DataBench

Evidence Based Big Data Benchmarking to Improve Business Performance

D1.2 DataBench Framework – with Vertical Big Data Type benchmarks

Abstract

This document – DataBench Framework – with Vertical Big Data Type benchmarks – focuses on the classification of benchmarks according to different aspects, including Big Data types. The mappings from industry sectors and application types is made through their usage of various combinations of these six Big Data types. The document provides the introduction to the objectives of the deliverable, through a section which describes the DataBench Framework – based on the partners contributions to the BDVA Big Data Reference architecture – and the extensions for various Big Data types and thus usage of these within different application scenarios. Existing and new benchmarking approaches and challenges are being continuously mapped into the DataBench Framework matrix showing the relationship to the focus aspects of these. Further the focus is to present different Benchmarking approaches, including types of technical benchmarks and the relationship to business benchmarks. The document provides an overview of different benchmarking organisations, like TPC, SPEC, STAC, LDBC, BenchCouncil, BDVA-TF6-SG7 and Hobbit. Application benchmarks and Big Data types, use cases and application domains, Big Data standards (ISO SC42), Challenges, Competitions and Inducement prizes are also discussed. The document describes technical benchmarks as mapped into vertical benchmark groups, following the Big Data type dimensions, Structured data - IoT/Time series - Geo Spatial Temporal - Media, Images, Audio - Text, Language, Genomics - Web, Graph, Metadata.

This document "D1.2 DataBench Framework – with Vertical Big Data Type benchmarks" has been extended through the two documents " D1.3 Horizontal Benchmarks – Analytics and Processing" and "D1.4 Horizontal Benchmarks – Data Management" that are being provided at the same time as this document.

Deliverable D1.2	DataBench Framework with Vertical Big Data Type benchmarks
Work package	WP1
Task	1.2
Due date	31/12/2019 (Public version)
Submission date	23/12/2019
Deliverable lead	SINTEF
Version	2.1
Dissemination level	Public
Authors	SINTEF (Arne Berre, Volker Hoffman, Kasia Michalowska, Bushra Nazir, Chaudhry Rehan Ikram, Muhammad Shah Zaib, Afroditi Tsalgatidou) GUF (Todor Ivanov, Timo Eichhorn) ATOS (Tomás Pariente, Iván Martínez and Ricardo Ruiz) POLIMI (Chiara Francalanci) JSI (Marko Grobelnik)
Reviewers	Barbara Pernici (POLIMI), David Wells (IDC)

Keywords

Benchmarking, big data, big data technologies, BDVA Reference Model, Vertical benchmarks, Horizonal benchmarks, architecture, performance metrics

Disclaimer

Copyright Notice

Table of Contents

Table of Figures

Table of Tables

Executive Summary

This document focuses on the DataBench Framework – with Vertical Big Data Type benchmarks – based on a classification of benchmarks according to the six main Big Data types of the DataBench Framework. The mappings from industry sectors and application types is made through their usage of various combinations of these six Big Data types.

The document provides first an introduction to the objectives of the work package 1 and the deliverable and then a section which describes the DataBench Framework - based on the partners contributions to the BDVA Big Data Reference architecture – and the extensions for various Big Data types as a basis for mappings to application scenarios within the various industry sectors.

Existing and new benchmarking approaches and challenges are being continuously mapped into the DataBench Framework matrix showing the relationship to the focus aspects of

these. Further the focus is to present different Benchmarking approaches, including types of technical benchmarks and the relationship to business benchmarks.

Different benchmarking organisations, like TPC, SPEC, STAC, LDBC, BenchCouncil, BDVA-TF6-SG7 and Hobbit are described, together with Application benchmarks and Big Data types, use cases and application domains, Big Data standards (ISO SC42), Challenges, Competitions and Inducement (Motivational) prizes.

The document describes technical benchmarks as mapped into vertical benchmark groups, following the Big Data type dimensions, Structured data - IoT/Time series - Geo Spatial Temporal - Media, Images, Audio - Text, Natural Language, - Web, Graph/Linked Data and Metadata.

The conclusions of the document also describes the content of the companion deliverables D1.3 and D1.4 on Horizontal benchmarks and the relationship to other work packages.

Annex 1 contains structured descriptions of identified technical benchmarks – sorted by the year that they were introduced. The intention is that this Benchmark Annex will be provided in an online Knowledge Graph structure and will continue to be updated separately, and thus serve as a source of detailed information for identified and referred benchmarks.

The version 2.1 of this document has added more details on the further refinements of the categories of the DataBench Framework areas, based on recent definitions from the ISO SC42 AI and Big Data standardisation work, and an extended set of benchmarks introduced during 2019 in particular with more benchmarks in the AI/Machine Learning area. In addition, some of this is will become available through the DataBench Toolbox and web support.

1. Introduction and Objectives

The DataBench Framework is based on a combination of both the vertical and horizontal dimensions of the BDVA Reference Model, which uses a set of six different Big Data types to focus on end-to-end support along the horizontal layers of visualisation, analytics, processing and data management.

Existing Big Data benchmarks have primarily focused on the commercial/retail domain related to transaction processing (TPC benchmarks and BigBench) or to applications suitable for graph processing (Hobbit and LDBC – Linked Data Benchmark Council). The analysis of different sectors in the BDVA has concluded that they all use different mixes of the different Big Data Types (Structured data, Time series/IoT, Spatial, Media, Text and Graph). Industrial sector specific benchmarks will thus relate to a selection of important data types, and their corresponding vertical benchmarks, adapted for this sector. The existing holistic industry/application benchmarks have primarily been focusing on structured data and Graph data types and DataBench will in addition be focusing on also supporting the newer benchmarks related to the industry requirements for time series/IoT, spatial and media and text, from the requirements of different industrial sectors such as manufacturing, transport, bio economies, earth observation, health, energy and many others.

The D1.2 document – DataBench Framework – with Vertical Big Data Type benchmarks – focuses on the classification of benchmarks according to the six main Big Data types. The mappings from industry sectors and application types is made through their usage of various combinations of these six Big Data types.

The D1.2 document is structured as follow:

- Section 1 provides the introduction to the objectives of WP1 and the deliverable.
- Section 2 describes the DataBench Framework - based on the partners contributions to the BDVA Big Data Reference architecture – and the extensions for various Big Data types and thus usage of these within different application scenarios. Existing and new benchmarking approaches and challenges are being continuously mapped into the DataBench Framework matrix showing the relationship to the focus aspects of these.
- Section 3 presents different Benchmarking approaches, including types of technical benchmarks and the relationship to business benchmarks. This is followed by a description of different benchmarking organisations, like TPC, SPEC, STAC, LDBC, BenchCouncil, BDVA-TF6-SG7 and Hobbit. Further the section presents application benchmarks and Big Data types, use cases and application domains, Big Data standards (ISO SC42), Challenges and inducement prices for Big Data application problems.
- Section 4 describes technical benchmarks as mapped into vertical benchmark groups, following the Big Data type dimensions, Structured data - IoT/Time series - Geo Spatial Temporal - Media, Images, Audio - Text, Language, Genomics - Web, Graph, Metadata.
- Section 5 provides the conclusions of the document as well as outlines the future work of the soon forthcoming deliverables D1.3 and D1.4 on Horizontal benchmarks and the relationship to other work packages.
- Annex 1 contains structured descriptions of all of the technical benchmarks – sorted by the year that they were introduced. The intention is that this Annex will be continued to be updated separately, and serve as a source of detailed information for all of the identified and referred benchmarks.

The **objective 1** of the DataBench project is to provide the BDT Stakeholder communities with a comprehensive framework to integrate Business and Technical benchmarking approaches for Big Data Technologies. This includes developing a BDT framework bringing together diverse BDT benchmarking solutions to provide a comprehensive benchmarking system able to respond to the real needs of European businesses, technology providers and the research community. DataBench will identify and unify the numerous existing BDT benchmarking initiatives and their business and technical metrics into a common structure based on the BDVA reference model. DataBench will investigate and deliver a single model to import and assess the technical requirements and data coming from existing benchmarking tools and platforms based on the BDVA reference model and provide recommended benchmarks for dimensions from Big Data Analytics through processing to data management, covering various Big Data types from structured data through time series/real-time streaming. The objective is to provide a model which correlates technical benchmarks to performance and business needs of different sectors and domains.

The specific objectives of WP1 are being addressed by D1.2 and other WP1 deliverables as follows:

- Identify Industrial Requirements from different industry sectors – Interviews for priorities and metrics (which has been addressed through the previous D1.1 deliverable).
- Establish the *Big Data Benchmarking Community (BDBC*) - which is being organised through Task 6.2 in WP6 – and related to the different technical benchmarks and communities – identified and described in this deliverable.
- Establish vertical holistic benchmarks – end-to-end for different Industry sectors – which are based on the mappings/usage of the different Big Data types in the various application areas for these.
- *Establish vertical benchmarks – Big Data Type specific – which is being addressed in this deliverable D1.2 through the groupings of relevant benchmarks based on their respective Big Data type focus aspects.*
- Establish vertical benchmarks related to Data Privacy/ Security – which is being addressed in D1.4 related to how this can be related to each of the horizontal benchmarking areas.
- Analyse and adapt horizontal benchmarks for Analytics/AI/Machine Learning and Processing - which is being addressed by D1.3.
- Analyse and adapt horizontal benchmarks for Data Management – which is being addressed by D1.4.

This D1.2 document extends various areas from the D1.1 document "Industry Requirements with benchmarks metrics and KPIs". The D1.1 document has provided an initial overview of Big Data Benchmarking, with a focus on the DataBench ecosystem of Key Performance Indicators Classifications, including Business features and Business indicators. The D1.1 document has further provided the basis for further work towards a methodological integration framework for business and technical benchmarks, which will be further continued in other work packages.

The results of WP1 and D1.1 and D1.2 will feed into WP2 for further detailing of business requirements related to economic, market and business analysis, it will feed into the WP3 DataBench Toolbox for the implementation support for the DataBench Framework, to WP4 for the evaluations of business performance and to WP5 for the technical evaluations with the DataBench Toolbox. Further support and consensus building with the involved communities will be managed by WP6.

This document "D1.2 DataBench Framework – with Vertical Big Data Type benchmarks" has been extended through the documents " D1.3 Horizontal Benchmarks – Analytics and Processing" [65] and "D1.4 Horizontal Benchmarks – Data Management" [66] that are being provided at the same time as this document.

2. DataBench Framework

The DataBench Framework is based on the structure of the BDVA Architecture Model – and is focusing on both vertical and horizontal benchmarks according to this model – further related to business-oriented benchmarks.

The industry-based use cases are analysed in order to derive examples and metrics that can be related to each of the Big Data types. The focus is on reusing and adapting the established benchmarks for structural data (BigBench, BigDataBench, TPC and others) and graph data/linked data (Hobbit I-IV and LDBC 1-3) and in particular on incorporating benchmark proposals related to Time series/IoT (Yahoo Stream Benchmark, RIoTBench, StreamBench and others) and also input from DataBench partners research benchmarks on streaming sensor data, ABench (UFRA) and SenseMark (SINTEF).

Similarly, there will be a focus on the data types of Image/Audio/Media and Text/NLP where also analytic and processing benchmarks for machine learning (DeepBench, DeepMark and others) are relevant. A final relevant area for vertical benchmarks is on the effect of technology support for data privacy and security. A set of projects related to how to support data privacy has been started under the Big Data PPP ICT18 and a benchmark approach for analysing and understanding the use of these techniques has been requested from the user community.

The vertical dimension is based on benchmarks according to the following Big Data types:

- Structured Data Benchmarks
- IoT/Time Series Benchmarks
- SpatioTemporal Benchmarks
- Media/Image Benchmarks
- Text/NLP Benchmarks
- Graph/Metadata Benchmarks

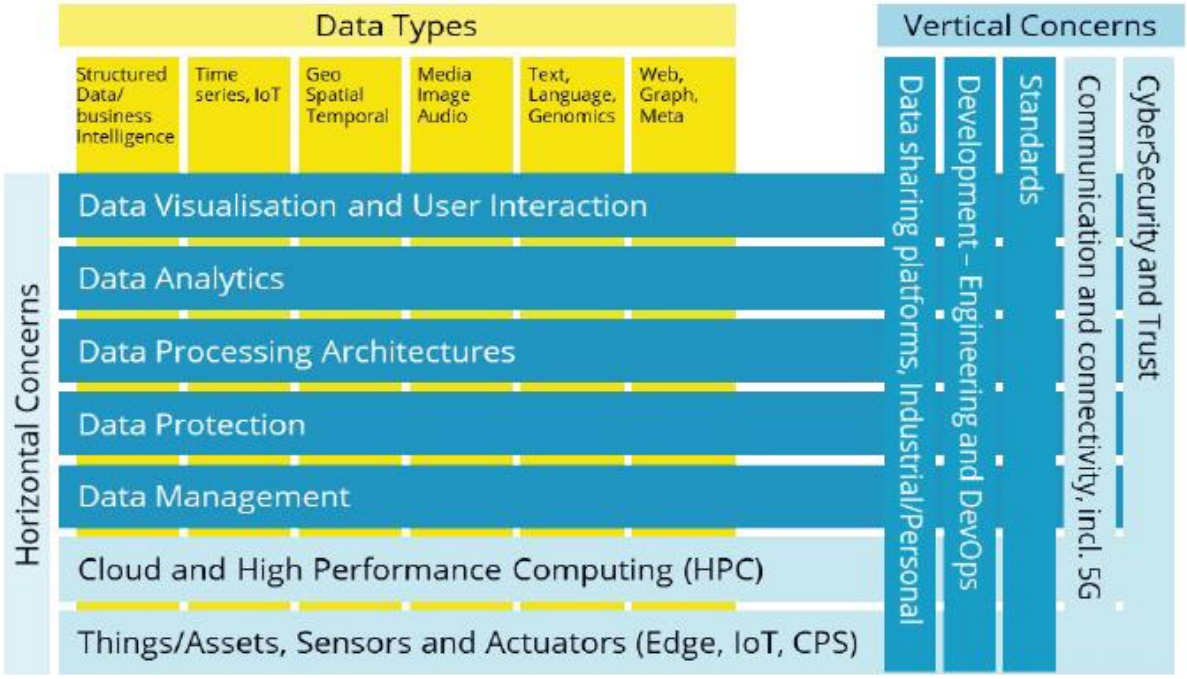


Figure 1 - BDV Reference Model as a foundation for the DataBench Framework

The BDV Reference Model[1] shown in Figure 1 has been developed by the BDVA, taking into account input from technical experts and stakeholders along the whole Big Data Value chain as well as interactions with other related PPPs. An explicit aim of the BDV Reference Model in the SRIA 4.0 document is to also include logical relationships to other areas of a digital platform such as Cloud, High Performance Computing (HPC), IoT, Networks/5G, CyberSecurity etc.

The BDV Reference Model may serve as common reference framework to locate Big Data technologies on the overall IT stack. It addresses the main concerns and aspects to be considered for Big Data Value systems.

The BDV Reference Model is structured into horizontal and vertical concerns.

- **Horizontal concerns** cover specific aspects along the data processing chain, starting with data collection and ingestion, reaching up to data visualization. It should be noted, that the horizontal concerns do not imply a layered architecture. As an example, data visualization may be applied directly to collected data (data management aspect) without the need for data processing and analytics. Further data analytics might take place in the IoT area – i.e. Edge Analytics. This shows logical areas – but they might execute in different physical layers.
- **Vertical concerns** address cross-cutting issues, which may affect all the horizontal concerns. In addition, verticals may also involve non-technical aspects (e.g., standardization as technical concerns, but also non-technical ones).

[1] http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf (page 37)

Given the purpose of the BDV Reference Model to act as a reference framework to locate Big Data technologies, it is purposefully chosen to be as simple and easy to understand as possible. It thus does not have the ambition to serve as a full technical reference architecture. However, the BDV Reference Model is compatible with such reference architectures, most notably the emerging ISO JTC1 WG9 Big Data Reference Architecture – now being further developed in ISO JTC1 SC42 Artificial Intelligence.

The following technical priorities as expressed in the BDV Reference Model are elaborated in the remainder of this section:

Horizontal concerns:

- **Big Data Applications**: Solutions supporting Big Data within various domains will often consider the creation of domain specific usages and possible extensions to the various horizontal and vertical areas. This is often related to the usage of various combinations of the identified Big Data types described in the vertical concerns.
- **Data Visualisation** and User Interaction: Advanced visualization approaches for improved user experience.
- **Data Analytics**: Data analytics to improve data understanding, deep learning, and meaningfulness of data.
- **Data Processing Architectures**: Optimized and scalable architectures for analytics of both data-at-rest and data-in- motion with low latency delivering real-time analytics.
- **Data Protection**: Privacy and anonymisation mechanisms to facilitate data protection. It also has links to trust mechanisms like Blockchain technologies, smart contracts and various forms for encryption. This area is also associated with the area of CyberSecurity, Risk and Trust.
- **Data Management**: Principles and techniques for data management including both data life cycle management and usage of data lakes and data spaces, as well as underlying data storage services.
- **Cloud and High Performance Computing (HPC):** Effective Big Data processing and data management might imply effective usage of Cloud and High Performance Computing infrastructures. This area is separately elaborated further in collaboration with the Cloud and High Performance Computing (ETP4HPC) communities.
- **IoT, CPS, Edge and Fog Computing**: A main source of Big Data is sensor data from an IoT context and actuator interaction in Cyber Physical Systems. In order to meet real-time needs it will often be necessary to handle Big Data aspects at the edge of the system.

Vertical concerns:

- **Big Data Types and semantics**: The following six Big Data types have been identified – based on the fact that they often lead to the use different techniques and mechanisms in the horizontal concerns, which should be considered, for instance for data analytics and data storage: *1) Structured data; 2) Times series data; 3) GeoSpatial data, 4) Media, Image, Video and Audio data; 5) Text data, including Natural Language Processing data and Genomics representations; 6) Graph data, Network/Web data and Meta data*. In addition, it is important to support both the syntactical and semantic aspects of data for all Big Data types.
- **Standards**: Standardisation of Big Data technology areas to facilitate data integration, sharing and interoperability.

- **Communication and Connectivity**: Effective communication and connectivity mechanisms are necessary for providing support for Big Data. This area is separately elaborated further with various communication communities, such as the 5G community.
- **Cybersecurity**: Big Data often need support to maintain security and trust beyond privacy and anonymisation. The aspect of trust frequently has links to trust mechanisms such as blockchain technologies, smart contracts and various forms of encryption. The CyberSecurity area is separately elaborated further with the CyberSecurity PPP community.
- **Engineering and DevOps**: for building Big Data Value systems. This area is also elaborated further with the NESSI (Networked European Software and Service Initiative) Software and Service community.
- **Data Platforms**: Marketplaces, IDP/PDP, Ecosystems for Data Sharing and Innovation support. Data Platforms for Data Sharing include in particular Industrial Data Platforms (IDPs) and Personal Data Platforms (PDPs), but also include other data sharing platforms like Research Data Platforms (RDPs) and Urban/City Data Platforms (UDPs). These platforms include efficient usage of a number of the horizontal and vertical Big Data areas, most notably the areas for data management, data processing, data protection and CyberSecurity.
- **AI platforms**: In the context of the relationship between AI and Big Data there is an evolving refinement of the BDV Reference Model – showing how AI platforms typically include support for Machine Learning, Analytics, visualization, processing etc. in the upper technology areas supported by data platforms – for all of the various Big Data types.

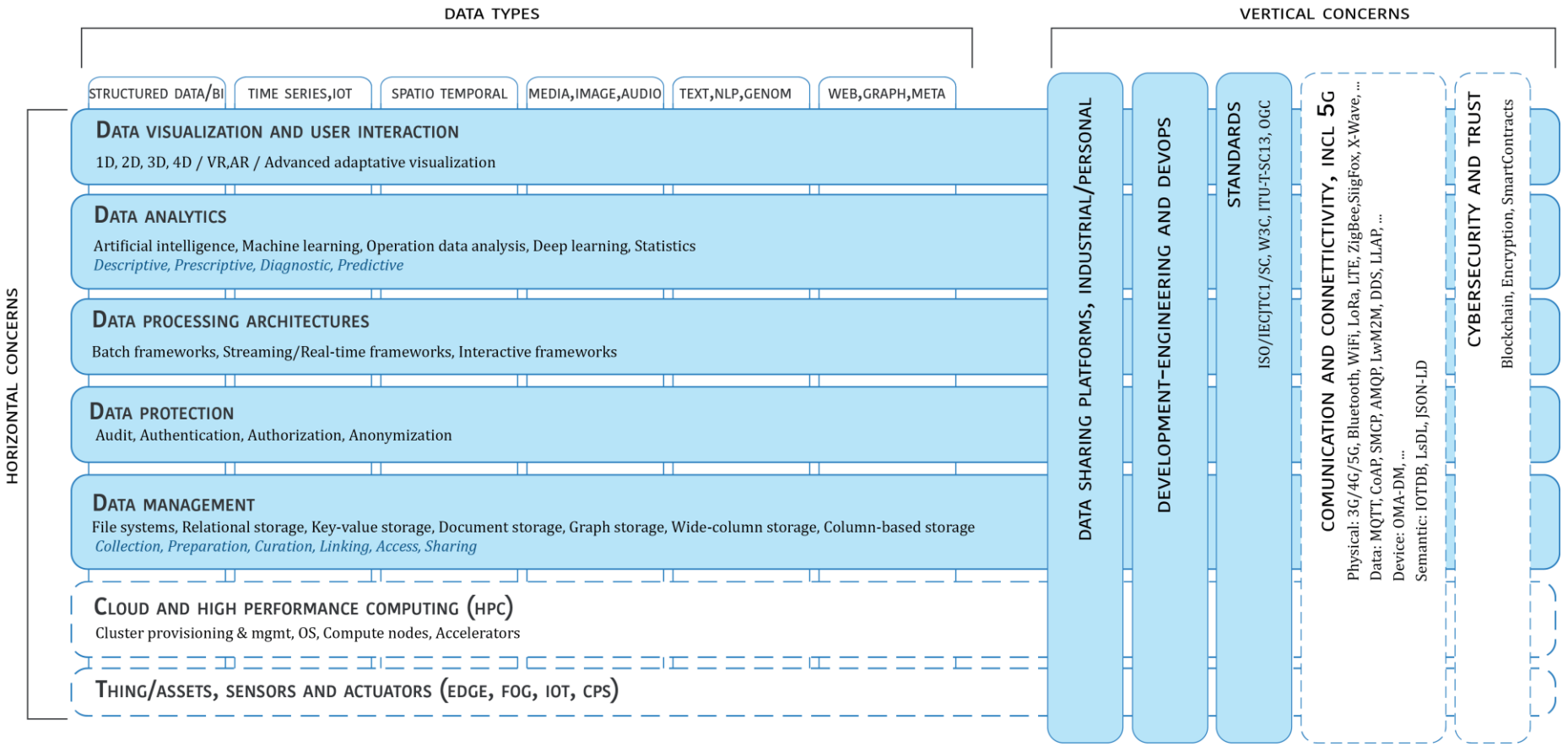


Figure 2 - Refinement of the BDVA Reference Model

The BDV Reference Model may serve as common reference framework to locate Big Data technologies on the overall IT stack. BDV Reference Model is compatible with such reference architectures, most notably the ISO JTC1 WG9 Big Data Reference Architecture which now has become part of the ISO SC42 AI (and Big Data) standard ISO 12345 XX.

The refinement of the BDVA Reference Model has been based on defining sub-categories within each of the reference model areas based on the refinement of the respective areas in the ISO SC42 suite of standards and technical reports currently in progress. The sub-categories describe typical technology types within each of the areas, relevant in benchmarking context.

The modeling approach in the figure is on the top level to describe logical technical areas within a wider Big Data and AI platform, and within each of the areas, relevant subcategories within this area. In addition to technical subcategories it has also been identified typical process steps in a Big Data pipeline relevant for the various areas. Work has started to consolidate and unify the models, metamodels and ontologies from D1.1, D3.1, D5.1 and D1.2 and the companion D1.3 and D1.4 public deliverables.

Data Visualization and User Interaction Layer:

This layer incorporates the research areas related to science of analytical reasoning assisted by advanced visualization and user interaction approaches. Major concern areas include:

> ***Visual data discovery***_ Proactive extraction of relevant information through visual data discovery techniques.
>
> ***Interactive visual analytics of multiple scale data***_ Facilitating empirical search for acceptable scales of analysis and the verifications of results.
>
> ***Collaborative, intuitive and interactive visual interfaces***_ Exploiting advanced discovery aspects of Big Data Analytics to enable collaborative decision-making processes. Carefully designed presentations and digital visualizations (including zooms, dynamic filtering, annotation*)* for quick and correct interpretation of data, Focus on relevance and relatedness of information for efficient search and exploration.
>
> ***Cross-platform mechanisms for data exploration, discovery and querying***_ Uniform data visualization on a range of devices.
>
> ***Innovating reporting***_ Innovative multi-device reports and dashboards (including dynamic, 3D, augmented-reality dimensions, etc.).
>
> ***Domain-specific data visualization techniques***_ Innovative techniques and approaches to visualize data coming from specific domain (e.g. graphs, geospatial, sensor, mobile data, etc.).

Sub-categories based on the ISO SC42 Big Data reference model:

This layer corresponds to the *Big Data Application Layer (**Visualization functional component**)* of ISO SC42 Big Data reference model. *The visualization functional component* is a part of Big Data architecture that is used to present analysed data in a meaningful manner, where data can be easily navigated, is comprehensible, with the possibility of distributed parallel operation on data.

Exploratory data visualization: multi-dimension (2D/3D), multi-resolution, interaction, animation, simulation, statistical graphics, surface rendering, volume rendering.

[SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.2.5]

Explanatory data visualization: reports and customer summarization presentation.

[SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.2.5]

Data Analytics Layer:

This layer incorporates data analytics to improve data understanding, deep learning and the meaningfulness of data. Major concern areas include:

Analytics frameworks and processing frameworks and APIs for batch and stream processing analytics. Improvement of scalability and speed of analytical algorithms.

Descriptive analytics methods for historical analysis.

Diagnostic analytics methods for diagnostic analysis (anomaly detection, fraud detection, condition monitoring).

Predictive analytics machine learning, clustering, pattern mining, network analysis.

Prescriptive analytics hypothesis testing techniques, recommendation systems.

Hybrid analytics combining data-driven analytics with first-order modelling – simulation/optimization.

Advanced business analytics and intelligence simplification and automation of these techniques.

Extreme analytics applying high performance computing (HPC) techniques to the processing of extremely huge amounts of data (data centre optimization, efficient resource allocation, quality of service provisioning).

Data analytics and Artificial Intelligence (AI) development of efficient and reliable data analytics processes for advanced and complex applications. This includes machine learning algorithms for deep learning and reinforcement learning techniques based on neural networks, and distribution of processing steps close to data sources (distributed deep learning).

Semantic and knowledge-based analysis near real-time interpretation of data, ontology engineering for Big Data sources, interactive visualization and exploration, real-time interlinking and annotation of data sources, scalable and incremental reasoning, linked data mining and cognitive computing.

Content validation validating content and exploiting content recommendations from unknown users.

Sub-categories based on the ISO SC42 Big Data reference model:

This layer corresponds to the *Big Data Application Layer (**Analytics functional component**)* of ISO SC42 Big Data reference model. *Big Data Application Layer* mainly deals with collection, preparation, analytics, visualization and access of Big Data.

Analytics component within Big Data Application Layer, is a part of Big Data architecture that is used to encapsulate the specialized computations that need to take place on the data for information finding and/or knowledge extraction to meet the applications requirements by using specified algorithms.

[SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.2.4]

Major types of analytical algorithms are as follows:

Classes of algorithms for machine learning: correlation, classification, data fusion, data integration, data mining, artificial intelligence, pattern recognition, predictive modelling, regression, cluster analysis, spatial analysis, audio analysis, visual analysis, textual analysis, etc.

Classes of algorithms for text analysis: sentiment analysis, named entity recognition, and theme detection.

Numerical analysis algorithm: Fast Fourier Transforms, Linear Algebra, and N-Body methods.

Graph algorithms: Community detection, Subgraph/motif finding, Finding diameter, Clustering coefficient, Page rank, Maximal cliques, Connected component, Betweenness centrality, Shortest path.

Operation data analysis: analysis of log text files, systems status data, alert information, etc. for system operation and maintenance.

Workflows: combination of several types of algorithms.

Artificial Intelligence (AI)

AI is the capability of a system to solve problems by emulating concepts (such as are reasoning, learning, planning, cooperation, perception, and communication) that are generally associated with intelligent behavior. Reasoning, machine learning, and problem solving are essential abilities in various AI systems.

[SOURCE: ISO/IEC JTC 1/SC 42/SG – Computational Approaches and AI Systems]

ISO/IEC JTC SC 42 is the effort of standardization in the area of AI, serving as a guidance to developing AI systems. Artificial Intelligence requires modern and heterogeneous hardware, including parallelism and distribution to boost performance. A variety of AI approaches, techniques, algorithms, and common characteristics are listed as below, with the cross reference to the corresponding section in the standard.

Major Characteristics of AI systems

Some of the common characteristics that may appear in AI systems are [SOURCE: ISO/IEC JTC 1/SC 42/SG – 6.3]:

Adaptable_ *adapt to changes in itself and its environment, depending on factors like domain data, architecture, etc.*

Constructive_ *generate static or dynamic output(s) based on input criteria.*

Coordinated_ *provide coordination between agents.*

Dynamic_ *exhibit dynamic decision-making based on external data sources.*

Explainable_ *provide mechanism to explain what precipitated a decision or output.*

Generative or Discriminative_ *able to either distinguish between categories by probability exclusion or represent data aspects it is designed for by probability inclusion technique.*

Introspective_ *self-monitor to adapt to its environment or provide insight into its functionality.*

Trained or Trainable_ *either trained on a dataset before deployment or trained dynamically as system is used.*

Variety of data handling_ *system may generate output(s) based on input criteria.*

Existing specialized AI systems

Some major existing specialized AI systems can be categorized as follows:

[SOURCE: ISO/IEC JTC 1/SC 42/SG – 8]

1. *Intelligent speech systems*
2. *Computer vision system*
3. *Natural Language processing systems*
4. *Knowledge graph systems*
5. *Anomaly detection systems*
6. *Autonomous systems*
7. *Recommender systems*

Major techniques used by AI systems

AI systems use the following four main approaches, which can overlap with each other and with evolutionary systems.

1. Formal logic
2. Bayesian inference
3. Discriminators
4. Artificial neural networks

[SOURCE: ISO/IEC JTC 1/SC 42/SG – 7.1]

Problem solving techniques

Problem solving is the process where an AI system perceives and tries to find a desired solution from a present situation. Problem solving also includes decision making, which is the process of selecting the best suitable alternative to reach the desired goal.

[SOURCE: ISO/IEC JTC 1/SC 42/SG – 7.2]

1. ***Complex mapping_*** establish complex mappings from a raw input signal to some rich information.

2. ***Search algorithms_*** use search algorithms to explore a state space in order to find a solution to accomplish a task.

Reasoning techniques

[SOURCE: ISO/IEC JTC 1/SC 42/SG – 7.3]

1. Logic programs
2. Rule engines
3. Deductive classifier
4. Cased-based reasoning
5. Procedural reasoning

Machine Learning Techniques

Machine learning techniques are characterized by the ability to learn and act without being explicitly programmed. ML draws on results from many fields including: AI, probability and statistics, computational complexity theory, control theory, information theory, philosophy, psychology, and neurobiology.

[SOURCE: ISO/IEC JTC 1/SC 42/SG – 7.4]

Methods in ML include:

1. *Artificial neural networks (feed forward, recurrent)*
2. *Bayesian network*
3. *Decision tree*
4. *Deep learning (convolutional neural network, deep convolutional neural network)*
5. *Reinforcement learning*
6. *Transfer learning*
7. *Genetic learning*
8. *Support vector machine*

Categories of Machine Learning Algorithms:

[SOURCE: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/]

The most common and meaningful way for grouping algorithms is grouping by similarity in terms of their function. There are algorithms that could fit into multiple categories. We could handle these cases by selecting the group that subjectively is the best fit.

Categories related to SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.2.4

ML Algorithm Category	Description	Sub-category
Regression algorithms	Modelling the relationship between modeling the relationship between variables that is iteratively refined using a measure of error in the predictions made by the model.	• Ordinary Least Squares Regression (OLSR) • Linear Regression • Logistic Regression • Stepwise Regression • Multivariate Adaptive Regression Splines (MARS) • Locally Estimated Scatterplot Smoothing (LOESS)
Instance-based algorithms	Such methods typically build up a database of example data and compare new data to the database using a similarity measure in order to find the best match and make a prediction.	• k-Nearest Neighbor (kNN) • Learning Vector Quantization (LVQ) • Self-Organizing Map (SOM) • Locally Weighted Learning (LWL) • Support Vector Machines (SVM)
Regularization algorithms	An extension made to another method (typically regression methods) that penalizes models based on their complexity, favoring simpler models that are also better at generalizing.	• Ridge Regression • Least Absolute Shrinkage and Selection Operator (LASSO) • Elastic Net • Least-Angle Regression (LARS)
Decision tree	Decision tree methods construct a model of decisions made based on actual values of attributes in the data. Decisions fork in tree structures until a prediction decision is made for a given record. Decision trees are trained on data for classification and regression problems.	• Classification and Regression Tree (CART) • Iterative Dichotomiser 3 (ID3) • C4.5 and C5.0 (different versions of a powerful approach) • Chi-squared Automatic Interaction Detection (CHAID) • Decision Stump • M5 • Conditional Decision Trees
Bayesian algorithms	Bayesian methods are those that explicitly apply Bayes' Theorem for problems such as classification and regression.	• Naive Bayes • Gaussian Naive Bayes • Multinomial Naive Bayes • Averaged One-Dependence Estimators (AODE) • Bayesian Belief Network (BBN) • Bayesian Network (BN)

Clustering algorithms	Clustering, like regression, describes the class of problem and the class of methods. Clustering methods are typically organized by the modeling approaches such as centroid-based and hierarchal. All methods are concerned with using the inherent structures in the data to best organize the data into groups of maximum commonality.	• k-Means • k-Medians • Expectation Maximisation (EM) • Hierarchical Clustering
Associate rule learning algorithms	Association rule learning methods extract rules that best explain observed relationships between variables in data.	• Apriori algorithm • Eclat algorithm
Artificial neural network algorithms	Artificial Neural Networks are models that are inspired by the structure and/or function of biological neural networks. They are a class of pattern matching that are commonly used for regression and classification problems but are really an enormous subfield comprised of hundreds of algorithms and variations for all manner of problem types. In contrast to Deep Learning, these algorithms are concerned with the more classical methods.	• Perceptron • Multilayer Perceptrons (MLP) • Back-Propagation • Stochastic Gradient Descent • Hopfield Network • Radial Basis Function Network (RBFN)
Deep Learning algorithms	Deep Learning methods are a modern update to Artificial Neural Networks that exploit abundant cheap computation.	• Convolutional Neural Network (CNN) • Recurrent Neural Networks (RNNs) • Long Short-Term Memory Networks (LSTMs) • Stacked Auto-Encoders

	They aim to build much larger and more complex neural networks and are usually concerned with very large datasets of labelled analog data, such as image, text, audio, and video.	• Deep Boltzmann Machine (DBM) • Deep Belief Networks (DBN)
Dimensionality reduction algorithms	Like clustering methods, dimensionality reduction seek and exploit the inherent structure in the data, but in this case in an unsupervised manner or order to summarize or describe data using less information	• Principal Component Analysis (PCA) • Partial Least Squares Regression (PLSR) • Sammon Mapping • Multidimensional Scaling (MDS) • Projection Pursuit • Principal Component Regression (PCR) • Partial Least Squares Discriminant Analysis (PLSDA) • Mixture Discriminant Analysis (MDA) • Quadratic Discriminant Analysis (QDA) • Regularized Discriminant Analysis (RDA) • Flexible Discriminant Analysis (FDA) • Linear Discriminant Analysis (LDA)
Ensemble algorithms	Ensemble methods are models composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction.	• Boosting • Bootstrapped Aggregation (Bagging) • AdaBoost • Weighted Average (Blending) • Stacked Generalization (Stacking) • Gradient Boosting Machines (GBM) • Gradient Boosted Regression Trees (GBRT) • Random Forest
Other	Algorithms from speciality task within ML process	• Feature selection algorithms • Algorithm accuracy evaluation • Performance measures • Optimization algorithms

Other	Algorithms from specialty subfields of machine learning, such as Computational intelligence, Computer Vision (CV), Natural Language Processing (NLP), Recommender Systems, Reinforcement Learning, Graphical Models, etc.	• Evolutionary algorithms • … • …

Data Processing Architectures Layer:

This layer deals with optimized and scalable architecture for analytics of both data-at-rest and data-at-motion, with low latency delivering real-time analytics.

Major concern areas include:

Heterogeneity_ handle Big Data's variety and uncertainty over several dimensions like syntactic formats, sematic representations, granularity, heterogenous hardware. Some data types are structured semi-structured or un-structured, multi-media, audio-visual, or textual data. Techniques for transformation and migration for data originated from heterogenous sources.

Scalability_ scalable analytical techniques, adjusting to increase of streams and volume of data.

Data processing techniques_ Real-time analytics through Event Processing and Stream Processing, spanning inductive reasoning (machine learning), deductive reasoning (inference), High Performance Computing and statistical analysis. Integrated processing of data-in-motion and data-at-rest

Decentralization_ Parallelization and distributed placement of data and data processing nodes.

Modern Architectures _ Integrating processing of data-at-rest and data-in-motion for robust and efficient analytics. Using modern architectures like Lamdba and Kappa architectures.

Performance_ Scaling performance of algorithms by reducing energy consumption, utilizing high performance computing, hardware-oriented technologies like main memory, software defined storage like built-in functionality for computation near the data, data availability guarantees and data reduction for efficient data processing.

High performance computing architectures_ novel architectures for computing-intensive applications with big and complex workloads and distributed workflows. Use of efficient energy consumption models.

New hardware_ increasing computing capacity using new hardware capabilities like deep learning processors.

Sub-categories based on the ISO SC42 Big Data reference model:

This layer corresponds to *Big Data Processing Layer (**Batch Frameworks & Streaming frameworks functional components**)* of ISO *SC42 Big Data reference model.* The focus here is primarily on performance (e.g. producing results of computations within the requisite period of time). *Big Data Processing Layer* adopts different processing engines which provide abstraction functionalities for the operations of the Big Data Application Layer. User operation is abstracted as data source, filter, map, window, aggregation, etc. The Big Data Processing Layer completes the execution process with data flowing from one operator to another, and from input to output. ISO *SC42 Big Data reference model* has categorized frameworks within this layer into following two categories:

1. ***Batch (Offline and Interactive) Frameworks*** main aim is to solve the problem of volume of Big Data, taking batch of elements as a basic unit to process. Batch frameworks provide two types of processing, either offline processing (when response time is in the minute or hour or longer range level) or interactive processing (when response time is in seconds level or sub-seconds level).

   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.3.2]

2. ***Streaming Frameworks*** mainly aims to solve the problem of velocity. The process model is pipelined, and every element is forwarded to next operator with ideally minimal possible latency. Ideally, the data flows continuously through the processing pipeline. ***Complex Event Processing*** is an advanced form of streaming which is queryable and adds more practical characteristics to pure streaming. The four characteristics include: event ordering, event processing guarantee, state store and stream partitioning / operator parallelism.

   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.3.3]

Data Protection Layer:

This layer deals with privacy and anonymization mechanisms to facilitate data protection. It has relation to data management and processing and area of CyberSecurity.

Data protection mechanisms Auditability for data usage, distributed trust technologies for distributed application scenarios.

Data privacy methods Anonymization techniques, data encryption techniques, quantifying privacy loss and data utility. Some effective methods of data privacy include differential privacy, private information retrieval, syntactic anonymity, homomorphic, encryption, secure search encryption, and secure multiparty computation. Advances in data protection may help in designing advanced privacy-preserving data-mining algorithms and pattern hiding techniques.

Sub-categories based on the ISO SC42 Big Data reference model

This layer corresponds to the *Security and Privacy Layer (**Audit framework, Authentication framework, Authorization framework**, and **Anonymization framework functional***

components) of ISO SC42 Big Data reference model. *Security and Privacy Layer* is responsible to main privacy, confidentiality, and integrity among different components in Big Data architecture.

1. ***Audit frameworks_*** used by other components to record events within the system. The audit trails and logs are used to help track provenance of data, for data/state recovery or forensic analysis of a system crash or incursion.
   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.6.3.2]

2. ***Authentication frameworks_*** provide access control to underlying data and services within other components and also to the system as a whole from external elements.
   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.6.3.3]
3. ***Authorization frameworks_*** supports mapping of a user or component within a system to the privileges (Read or Access, Write, Delete, Execute, Traverse and Terminate) they have in accessing resources (both data and processing) within the cluster.
   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.6.3.4]

4. ***Anonymization frameworks_*** supports maintaining privacy or security for data by obfuscating one or more data elements so that they cannot be easily associated with other data elements. A primary example of this is the anonymization of personally identifiable information (PII) about individuals to protect their privacy. These components frequently implement 1-way hash functions in order to create unique values that cannot easily be reversed to their original values.
   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.6.3.5]

Data Management Layer:

This layer deals with principles and techniques of data management. Data management is a set of activities aimed to implement the Big Data architecture that best meet business goals by following the strategic plan for data management assessment[2]. Interconnected data management is required for real-time business and next-generation applications. Major challenges in this area are:

Semantic annotation_ Pre-processing and enhancement of unstructured and structured data with semantic annotations.

Semantic interoperability_ Promote interoperability by usage of standards, efficient storage and exchange of semantic data, user-driven or automated annotations and transformations.

Data quality_ Improving and assessing data quality with improved data filtering techniques, human-data interaction, standardized data curation models and vocabularies.

[2] Data management assessment is a document specifying how data management is to be aligned to organizational strategy [SOURCE: ISO/IEC DIS 20547-3:2019(E), 4.5]

Data lifecycle management and data governance_ Methods and tools for data curation and cleaning (including pre-processing veracity, velocity integrity and quality of the data), Big Data transformations approaches (including aspects of automatic, interactive, sharable and repeatable transformations), and long-term storage and data access.

Data handling_ Tools and techniques for handling structured and unstructured data (including automatic measuring, tools for pre-processing and analyzing sensor, social, geospatial, genomics, proteomics and other domain-orientated data), as well as, standardized annotation frameworks for different sectors supporting the technical integration.

Data-as-a-service_ Bundle data, analytics and software in a single package.

Sub-categories based on the ISO SC42 Big Data reference model:

This layer corresponds to the *Big Data Platform Layer (**File system**, **Relational storage**, **Key-value storage**, **Wide-column storage, column-based storage, document storage,** and **graph storage functional components**)* of ISO SC42 Big Data reference model. It also corresponds to *Service Management (**Big Data Lifecycle Management**)* and *Big Data Application Layer (**Collection** and **Preparation functional components**).*

1. ***Big Data Platform Layer*** provides for the logical data organization and distribution combined with the associated access application programming interfaces (APIs) or methods. This may also include data registry and metadata services along with semantic data descriptions such as formal ontologies or taxonomies.
   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.4]

   This layer includes file system, relational and non-relational data storage systems as different functional components.

   a. ***File systems***_ *File systems* organize chunks of data accessed (typically defined as records) as a named entity within a defined namespace.

      Local filesystems are often used within Big Data systems for storing intermediate data local to a processing node. *Distributed file systems* manage the distributions and replication of data blocks across nodes and the namespace, rather than being stored with the data, is managed through a central name service often running in a master/slave or multi-master manner to provide fault tolerance. Distributed file systems seek to overcome the throughput issues presented by the volume and velocity characteristics of Big Data by combining I/O throughput across multiple devices n each node, with redundancy and failover mirroring or replicating data at the block level across multiple nodes. The replication prevents data lost in case of system/node failures, and allows for high levels of concurrency for reading data and for initial writes.

      Distributed object stores (DOSs) are a unique example of distributed file system organization. Unlike traditional file system hierarchy namespace approaches, DOSs present a flat namespace with a globally unique identifier

(GUID) for any given chunk of data. Generally, data in the store is located through a query against a metadata catalog that returns the associated GUIDs. These stores are commonly designed and used for storing high volumes of static and unstructured data, especially for AI and cloud-native applications.

b. ***Relational storage***_ provides data storage as rows with each field representing a column organized into a table based on the logical data organization. The actual storage of the data can be flat files where each record/line in the file represents a row in a table. Most Big Data relational storage models are batch oriented systems designed for very complex queries, which can generate very large intermediate cross-product matrices from joins. New implementations are focusing in improving response time. Some implementations are adopting binary storage formats optimized for distributed file systems. These formats use block level indexing and column-oriented organization of data to access individual fields in records without needing to read the entire record. Another approach to increase response time is to scale relational queries, distributing across multiple nodes (often as a Map/Reduce job).

a. ***Key-value storage***_ represents random access memory models. Key-value stores tend to work best when each key relates to a single value (1-1 relationships), but can also be effectively used for keys mapping to lists of homogeneous values (1-M relationships). In case of 1-M key/value structure, custom application logic is required. Distributed key-value stores are the most frequent implementation utilized in Big Data applications. One problem that is always addressed (not unique to key-value implementations) is the distribution of keys over the space of possible key-values.
There are various concerns related to the choice of keys for an implementation. Keys should be chosen carefully to avoid skew in the distribution of the data across the cluster. Heavily skewed data can result in computation hot spots across the cluster. For dynamic data (with new keys being added) there might be need for an occasional rebalancing across the cluster.

b. ***Wide-column storage***_ organizes data in groups of like values. The value of every column is a key and like column values point to the associated rows. In many ways, columnar data stores look very similar to indexes in relational databases. In addition, implementations of wide columnar stores that follow the sparse, distributed multi-dimensional sorted map model (where arbitrary byte arrays are indexed/accessed based on row and column keys) introduce an additional level of segmentation beyond the table, row and column model of the relational model, that is called the column family. Wide columnar stores add an additional dimension known as the column family.

c. ***Column-based storage***_ organizes and stores data by columns (unlike row-based stores where data is stored by rows), columnar. databases are well-suited for Big Data applications which require a wide spectrum of analysis,

such as multi-dimensional OLAP (online analytic processing) query, big and small scan query. Various column. based sorting, indexing and compression techniques, e.g. multi-dimensional indexing, dictionary coding etc., can be applied to increase the query performance.

d. ***Document storage***_ includes extensive search and indexing capabilities for structured data and metadata and why they are often referred to as semi-structured data stores. Within a document-oriented data store each document encapsulates and encodes the metadata, fields, and any other representations of that record. Their popularity lies in the fact that most implementations do not enforce a fixed or constant schema. That is one reason that document stores are frequently popular for datasets which have sparsely populated fields since there is far less overhead normally than traditional RDBMS systems where null value columns in records are actually stored.

e. ***Graph storage***_ represents data as a series of nodes, edges, and properties on those. In addition to social networking domain, Graph stores have been a critical part of many problem domains from military intelligence and counter terrorism to route planning/navigation and the semantic web for years. Analytics against graph stores include very basic shortest path and page ranking to entity disambiguation and graph matching. Unlike, relational and other data storage approaches most graph databases tend to use artificial/pseudo keys or guides to uniquely identify nodes and edges. This allows attributes/properties to be easily changed due to both actual changes in the data (someone changed their name) or as more information is found out (e.g. a better location for some item or event) without needing to change the pointers to/from relationships.
Typically, distributed architectures for processing graphs assign chunks of the graph to system nodes then the system nodes use messaging approaches to communicate changes in the graph or the value of certain calculations along a path. Even small graphs quickly elevate into the realm of Big Data when one is looking for patterns or distances across more than one or two degrees of separation between graph nodes.
A specialized implementation of a graph store known as the *Resource Description Framework* (RDF) is part of a family of specifications from the World Wide Web Consortium (W3C) that is often directly associated with Semantic Web and associated concepts. RDF triples as they are known consist of a Subject (Mr. X), a predicate (lives at), and an object (Mockingbird Lane). Thus, a collection of RDF triples represents a directed labeled graph. The contents of RDF stores are frequently described using formal ontology languages like OWL or the RDF Schema (RDFS) language, which establishes the semantic meanings and models of the underlying data. Graph data stores currently lack any form of standardized APIs or query languages. However, the W3C has developed the SPARQL query language for RDF which is currently in a recommendation status and there are several systems such as

Sesame which are gaining popularity for working with RDF and other graph oriented data stores.

2. ***Big Data Lifecycle Management components*** provide the functions to manage the Big Data lifecycle from the moment data are ingested into the system via the data import component, until they are processed or removed from the system. It includes:
    a. ***Metadata management*** refers to the management capabilities and functions of metadata generated in each Big Data life-cycle stage – from ingestion, pre-processing, processing, analysis, storage, to destruction or removal. The proper management of metadata is instrumental to data mining and analytics process, as metadata provides information on how data can be treated or utilized.

    b. ***Data quality management*** refers to the coordinated activities to direct and control an organization with regard to data quality [SOURCE: ISO 8000-2:2017(en), 3.4.9].
    Data quality is the degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions [SOURCE: ISO/177 IEC 25024:2015, 4.11].
    Data validation and data cleaning (transform, validate, cleanse, aggregate) should be guided by the application of Data quality management.

3. ***Collection components*** are used to establish connection to data provider, import data and store data. Systems under this category are concerned with getting the data into the system.

    [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.2.2]

4. ***Preparation components*** are used to prepare data for analytics. Common functions include data aggregation, cleansing, transformation, data calculation filed creation, data optimization, data partition, data summarization, data alignment, data validation, data virtualization and storage of prepared data.

    [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.2.3]

Cloud and High Performance Computing:

This layer deals with effective usage of Cloud and High Performance Computing architectures. Major technical requirements may include highly scalable performance, high memory bandwidth, low power consumption and excellent short arithmetic performance. This layer raises technical challenges in its subsequent uppers layers in BDV reference model.

Sub-categories based on the ISO SC42 Big Data reference model

This layer corresponds directly to *Big Data Infrastructure Layer* of ISO *SC42 Big Data reference model* . This layer integrates the architectural concerns regarding performance of

analytical techniques, processing frameworks, physical computing resources and architectures and data management.

Big Data Infrastructure Layer include functional components for resource abstraction and control and physical resources.

a. ***Resource abstraction***_ software abstraction over physical computing resources. Examples include elasticity, resource pooling, on-demand self-service, software elements (such as hypervisors, virtual machines, virtual data storage, time sharing), automated deployment, provisioning capabilities, infra-structure wide monitoring agents.
[SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.5.2]

b. ***Physical resources***_ hardware resources to run Big Data applications, such as computers, routers, firewalls, storage components, plant resources, etc.
[SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.5.3]

Things/Assets, Sensors and Actuators:

This layer deals with the handling of Big Data aspects at the edge of a system. A main source of Big Data is sensor data from an IoT context and actuator interaction in Cyber Physical Systems. Edge, Fog, IoT and CPS are typical systems that reside in this layer.

Communication and Connectivity Layer

This layer corresponds to *Integration layer* of ISO *SC42 Big Data reference model.*

1. ***Integration Layer*** provides services to connect the functionality of the components in the same layer or across different layers.
   It may include:
   a. ***Messaging frameworks***_ message routing and exchange between nodes in a horizontally scaled cluster, or components in the same or across different layers of application, processing, storage and computing components.
   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.6.2.2]

   b. ***State management frameworks***_ persist state across nodes in a distributed environment to ensure state consistency and persistency in case of system or resource failures.
   [SOURCE: ISO/IEC DIS 20547-3:2019(E), 10.2.6.2.3]

2.1 Big Data Types and Industry Sectors

Figure 2-b below shows on top various relevant industry sectors for the use of Big Data technologies. It is also shown how the documents D1.2 and D1.3/1.4 are addressing the different areas of this through the various mappings to the Big Data Types vertically in D1.2 - through the horizontal benchmarking areas in DataBench D1.3 and D1.4.

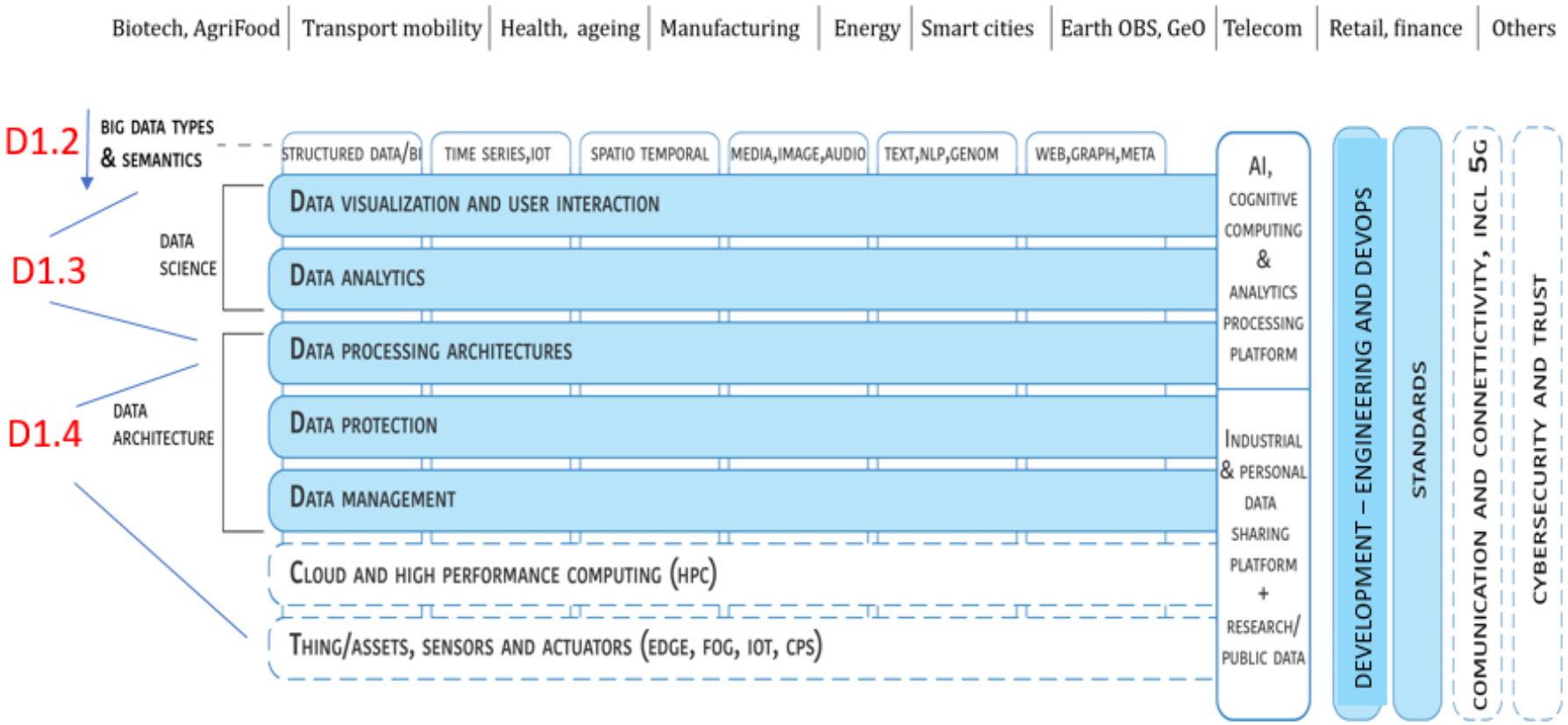


Figure 2-b - Industry Sectors and the BDVA Reference Model and D1.x focus

Figure 2-b shows on top various relevant industry sectors for the use of Big Data technologies. It is also shown how the documents D1.2 and D1.3/1.4 are addressing the different areas of this through the various mappings to the Big Data Types vertically in D1.2 - through the horizontal benchmarking areas in DataBench D1.3 and D1.4.

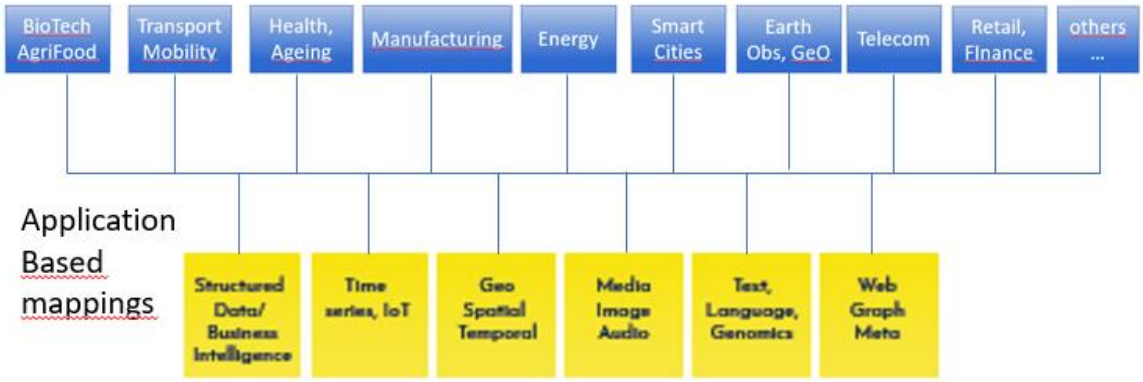


Figure 3 - Industry sectors mapped to Big Data types through Applications

Figure 3 illustrated how the large number of industry sectors can be mapped into various combinations of the six identified Big Data types, which typically will have different technology support in the various horizontal areas from analytics/machine learning through data processing and data management. The use of the Big Data types can help to reduce the number of application scenarios for benchmarking support – by instead of having very application specific benchmarks, reduce this to relevant combinations of benchmarks for the six different Big Data types.

2.2 DataBench and European AI Framework

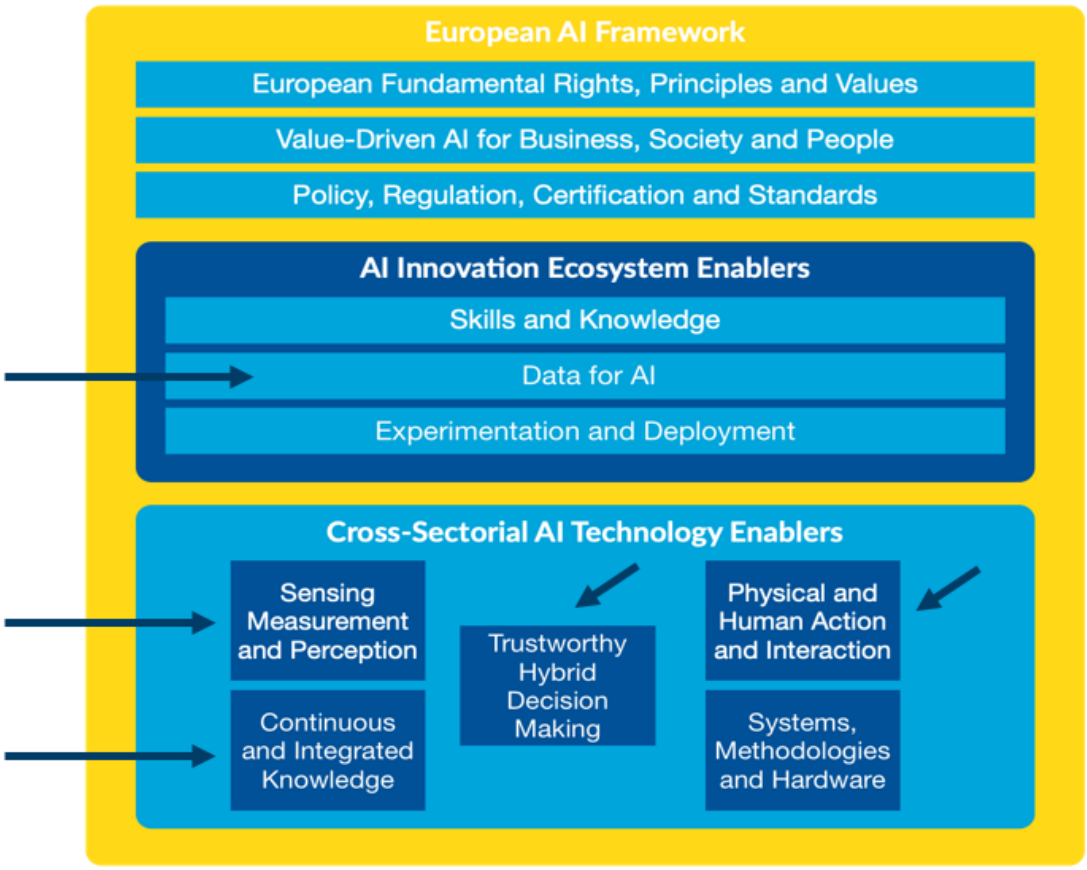


Figure 4 - European AI Framework from the BDVA/euRobotics Future AI SRIDA

The recent Future AI PPP – European AI Framework, shown in Figure 4, – contains areas that are also related to the DataBench benchmarking areas – in particular related to Data for AI and the cross sector AI technology enablers. These areas can be mapped into the DataBench Matrix by Sensing and Perception being mapped into the IoT/Cloud level, Data for AI and Continuous and Integrated Knowledge being mapped into the Data management level, Decision making being mapped into the processing and analytics level and Human interaction being mapped into the visual analytics level.

2.3 Overview of the DataBench Framework Matrix

The DataBench Framework for benchmarks is extended from the dimensions in the BDVA Reference Model – with additional aspects of industry sectors and application areas.

The DataBench Framework Matrix shown in Tables 1-6, classifies different existing benchmarks according to the various aspect of the DataBench Framework of Industry sector/domain, benchmark type, Big Data type out of the different data types: Metadata, Graph, Text, NLP, Image/Audio, Spatio temporal, Time series/IoT and structured data, and the different areas of the BDVA Reference Model: Analytics, Machine Learning/AI,

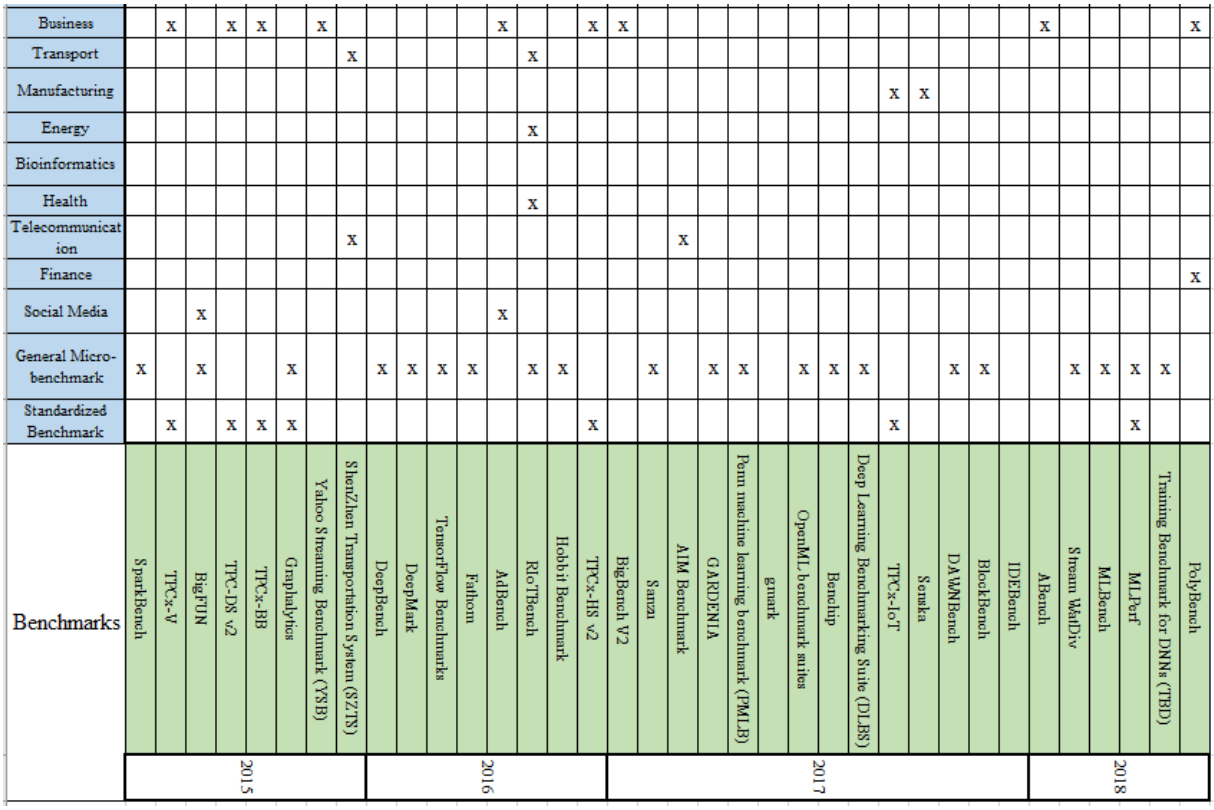
Processing (Streaming, Interactive, Batch), Data Privacy/Security, Data Governance, Data Storage and Communication/Connectivity and Cloud/HPC/Edge.

The Benchmarks are listed chronologically according to when they have been introduced. It is expected that more benchmarks will be introduced during the time of the project, and they will then be added incrementally to this matrix and to the corresponding descriptions.

The benchmarks are described according to a common template in Annex 1. The content of this Annex is being integrated with the DataBench Toolbox and the online Knowledge graph structure to represent a live version of all of the benchmark descriptions with their respective references.

Business	x	x					x									x		x			x			x				
Transport			x																									
Manufacturing																												
Energy																												
Bioinformatics																x												
Health																												
Telecommunication																					x	x						
Finance																												
Social Media																x				x	x	x		x				
General Micro-benchmark				x	x	x			x				x	x					x						x	x	x	x
Standardized Benchmark	x	x													x								x	x				x
Benchmarks	TPC-H	TPC-DS v1	Linear Road	Hadoop Workload Examples	GridMix	PigMix	MRBench	CALDA	HiBench	Liquid	YCSB	SWIM	CloudRank-D	PUMA Benchmark Suite	CloudeSuite	MRBS	AMP Lab Big Data Benchmark	BigBench	BigDataBench	LinkBench	BigFrame	PRIMEBALL	Semantic Publishing Benchmark (SPB)	Social Network Benchmark	ALOJA	Convnet	WadDiv	StreamBench	TPCx-HS
	1999	2002	2004	2007		2008		2009	2010			2011	2012			2013							2014					

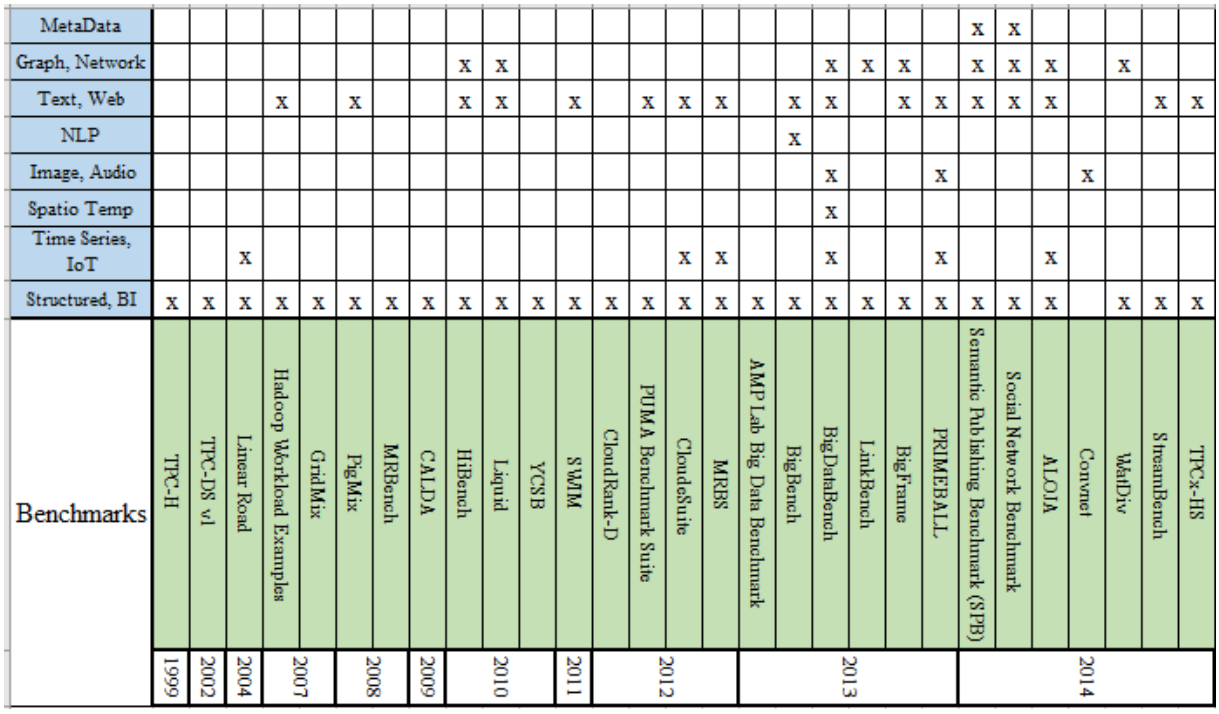
Table 1 - Domains for Big Data Benchmarks – 1999-2014



Benchmarks	Year	Business	Transport	Manufacturing	Energy	Bioinformatics	Health	Telecommunication	Finance	Social Media	General Micro-benchmark	Standardized Benchmark
SparkBench	2015		x								x	
TPCx-V		x										x
BigFUN										x	x	
TPC-DS v2		x										x
TPCx-BB		x										x
Graphalytics											x	x
Yahoo Streaming Benchmark (YSB)		x										
ShenZhen Transportation System (SZTS)			x					x				
DeepBench	2016										x	
DeepMark											x	
TensorFlow Benchmarks											x	
Fathom											x	
AdBench		x			x					x		
RIoTBench			x		x		x				x	
Hobbit Benchmark							x				x	
TPCx-HS v2		x										x
BigBench V2	2017	x										
Sanza											x	
AIM Benchmark								x				
GARDENIA											x	
Penn machine learning benchmark (PMLB)											x	
gmark												
OpenML benchmark suites											x	
Benchip											x	
Deep Learning Benchmarking Suite (DLBS)											x	
TPCx-IoT				x								x
Senska				x								
DAWNBench											x	
BlockBench											x	
IDEBench												
ABench	2018	x										
Stream WatDiv											x	
MLBench											x	
MLPerf											x	x
Training Benchmark for DNNs (TBD)											x	
PolyBench		x							x			

Table 2 - Domains for Big Data Benchmarks – 2015-2018

Table 1 and 2 describes how different Big Data benchmarks have been addressing different domains, introduced in the years 1999-2014 and 2015-2018, respectively.



Benchmarks	Year	MetaData	Graph, Network	Text, Web	NLP	Image, Audio	Spatio Temp	Time Series, IoT	Structured, BI
TPC-H	1999								x
TPC-DS v1	2002								x
Linear Road	2004							x	x
Hadoop Workload Examples	2007			x					x
GridMix									x
PigMix	2008			x					x
MRBench									x
CALDA	2009								x
HiBench	2010		x	x					x
Liquid			x	x					x
YCSB									x
SWIM	2011			x					x
CloudRank-D	2012								x
PUMA Benchmark Suite				x					x
CloudeSuite				x				x	x
MRBS				x				x	x
AMP Lab Big Data Benchmark	2013								x
BigBench				x	x				x
BigDataBench			x	x		x	x	x	x
LinkBench			x						x
BigFrame			x	x					x
PRIMEBALL				x		x		x	x
Semantic Publishing Benchmark (SPB)	2014	x	x	x					x
Social Network Benchmark		x	x	x					x
ALOJA			x	x				x	x
Convnet						x			
WatDiv			x						x
StreamBench									x
TPCx-HS				x					x

Table 3 - Big Data Types Benchmarks – 1999-2014

Benchmarks	SparkBench	TPCx-V	BigFUN	TPC-DS v2	TPCx-BB	Graphalytics	Yahoo Streaming Benchmark (YSB)	ShenZhen Transportation System (SZTS)	DeepBench	DeepMark	TensorFlow Benchmarks	Fathom	AdBench	RIoTBench	Hobbit Benchmark	TPCx-HS v2	BigBench V2	Sanzu	AIM Benchmark	GARDENIA	Penn machine learning benchmark (PMLB)	gmark	OpenML benchmark suites	Benchip	Deep Learning Benchmarking Suite (DLBS)	TPCx-IoT	Senska	DAWNBench	BlockBench	IDEBench	ABench	Stream WatDiv	MLBench	MLPerf	Training Benchmark for DNNs (TBD)	PolyBench
MetaData																																				
Graph, Network	x					x			x	x		x			x					x		x	x					x				x				x
Text, Web	x		x		x	x	x		x	x				x	x		x	x			x		x	x	x			x						x	x	x
NLP					x												x							x	x								x		x	
Image, Audio									x	x	x	x												x	x			x						x	x	
Spatio Temp								x							x																					
Time Series, IoT							x	x					x	x	x			x	x		x		x			x	x									x
Structured, BI	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			x	x	x	x	x	x	x	x	x		x
	2015								2016								2017														2018					

Table 4 - Big Data Types Benchmarks – 2015-2018

Table 3 and 4 describe how the different benchmarks have been addressing different Big Data types, introduced in the years 1999-2014 and 2015-2018, respectively.

Benchmarks	TPC-H	TPC-DS v1	Linear Road	Hadoop Workload Examples	GridMix	PigMix	MRBench	CALDA	HiBench	Liquid	YCSB	SWIM	CloudRank-D	PUMA Benchmark Suite	CloudSuite	MRBS	AMP Lab Big Data Benchmark	BigBench	BigDataBench	LinkBench	BigFrame	PRIMEBALL	Semantic Publishing Benchmark (SPB)	Social Network Benchmark	ALOJA	Convnet	WatDiv	StreamBench	TPCx-HS
Visual Analytics																													
Industrial Analytics (Descriptive, Diagnostic, Predictive, Prescriptive)																x			x				x						
Machine Learning, AI, Data Science									x				x			x		x	x				x			x			
Streaming/ Realtime Processing									x				x						x									x	
Interactive Processing	x	x					x				x					x	x	x	x	x	x	x	x	x		x	x		
Batch Processing	x	x		x	x	x	x	x	x			x	x	x		x	x	x	x	x	x	x	x	x		x	x		x
Data Privacy/Security																													
Data Governance/Mgmt																			x										
Data Storage	x	x		x	x	x	x	x	x		x		x	x	x	x	x	x	x	x	x	x	x	x				x	x
Communication & Connectivity				x																									x
Cloud Services & HPC, Edge										x			x		x		x		x			x			x				
	1999	2002	2004	2007		2008		2009	2010			2011	2012				2013							2014					

Table 5 - Big Data Analytics and Technology Benchmarks – 1999-2014

Benchmarks	SparkBench	TPCx-V	BigFUN	TPC-DS v2	TPCx-BB	Graphalytics	Yahoo Streaming Benchmark (YSB)	ShenZhen Transportation System (SZTS)	DeepBench	DeepMark	TensorFlow Benchmarks	Fathom	AdBench	RIoTBench	Hobbit Benchmark	TPCx-HS v2	BigBench V2	Sanzu	AIM Benchmark	GARDENIA	Penn machine learning benchmark (PMLB)	gmark	OpenML benchmark suites	Benchip	Deep Learning Benchmarking Suite (DLBS)	TPCx-IoT	Sensika	DAWNBench	BlockBench	IDEBench	ABench	Stream WstDiv	MLBench	MLPerf	Training Benchmark for DNNs (TBD)	PolyBench
Visual Analytics															x						x									x						
Industrial Analytics (Descriptive, Diagnostic, Predictive, Prescriptive)														x					x		x													x		
Machine Learning, AI, Data Science	x				x				x	x	x	x	x	x	x		x	x		x	x		x	x	x			x			x		x	x	x	
Streaming/ Realtime Processing	x						x							x					x								x				x	x				x
Interactive Processing	x	x	x	x	x	x	x		x	x	x	x	x	x	x		x		x	x		x				x	x			x	x	x				x
Batch Processing	x	x	x	x	x	x			x	x	x	x	x		x	x	x		x	x		x				x	x	x	x	x	x	x	x			x
Data Privacy/Security																																				
Data Governance/Mgmt																																				
Data Storage	x	x	x	x	x	x							x	x	x	x	x	x	x	x		x				x	x				x					
Communication & Connectivity																																				
Cloud Services & HPC, Edge		x													x																x					
	2015								2016								2017														2018					

Table 6 - Big Data Analytics and Technology Benchmarks – 2015-2018

Table 5 and 6 describe how the different benchmarks have been addressing different Big Data Analytics and technical architecture areas, introduced in the years 1999-2014 and 2015-2018, respectively.

3. Benchmarking Approaches

In this chapter we present different aspects of benchmarking including benchmarking terms and definitions, different types of benchmarks, benchmarking organisations, application benchmark perspectives, Big data standards and challenges/inducement prices.

3.1 Benchmarking Terms and Definitions

The meaning of the word benchmark defined in [1]:

" A predefined position, used as a reference point for taking measures against."

Jim Gray back in 1992 [2] described technical benchmarking as follows:

"This quantitative comparison starts with the definition of a benchmark or workload. The benchmark is run on several different systems, and the performance and price of each system is measured and recorded. Performance is typically a throughput metric (work/second) and price is typically a five-year cost-of-ownership metric. Together, they give a price/performance ratio."

In this context, a software benchmark is defined as a ***program used for comparison of software products/tools executing on a pre-configured, or configurable, hardware environment***.

Business benchmarking [3] focuses on the improvement of business activity, processes and management in companies. It differs from technical benchmarking and has multiple relevant definitions:

- "A continuous systematic process for evaluating the products, services and work of organisations that are recognised as representing best practices for the purpose of organisational improvement" [4].
- "A continuous search for, and application of, significantly better practices that lead to superior competitive performance" [5].
- "A disciplined process that begins with a thorough search to identify best-practice-organisations, continues with the careful study of one's own practices and performance, progresses through systematic site visits and interviews, and concludes with an analysis of results, development of recommendations and implementation" (Garvin, 1993).
- "Benchmarking is an external focus on internal activities, functions, or operations in order to achieve continuous improvement" [6].
- "Benchmarking is systematic and continuous measurement process: a process of continuously measuring and comparing an organisations business processes against process leaders anywhere in the world to gain information which will help the organisation to take action to improve its performance" (APQC/IBC cited in [5], p. 3).

In summary the characteristics that emerge from this definitions are 1) measurement via comparison; 2) continuous improvement and 3) systematic procedure in carrying out benchmarking activity. All three points are very relevant for technical benchmarking, which is the focus of this deliverable.

3.2 Type of Technical Benchmarks

Micro-benchmarks are either a program or routine to measure and test the performance of a single component or task [7]. They are used to evaluate either individual system components or specific system behaviors (or functions of codes) [8]. Micro-benchmarks report simple and well-defined quantities such as elapsed time, rate of operations, bandwidth, or latency [7]. Typically, they are developed for a specific technology, which reduces their complexity and development overhead. Popular micro-benchmark examples also part of the Hadoop binaries are WordCount, TestDFSIO, Pi, K-means, HiveBench and many others.

Application-level benchmarks also known as End-to-end benchmarks are designed to evaluate the entire system using typical application scenarios, each scenario corresponds to a collection of related workloads [8]. Typically, this type of benchmarks are more complex and are implemented using multiple technologies, which makes them significantly harder to develop. For example application level Big Data benchmarks are the one standardized by the Transaction Processing Performance Council (TPC) [9] such as TPC-H [10], TPC-DS [11], BigBench (TPCx-BB) [12] and many others.

Benchmark suites are combinations of different micro and/or end-to-end (application level) benchmarks and these suites aim to provide comprehensive benchmarking solutions [HJZ18]. Examples for Big Data benchmark suites are HiBench [13], SparkBench [14], CloudSuite [15], BigDataBench [16], PUMA [17] and many others.

Another important distinction between benchmarks is if they are standardized by an official organization (like SPEC [18] or TPC [9]) or not standardized (typically developed by a vendor or research organization). Also data and analytics driven challenges, such as those typically provided through the Kaggle Competitions[3], and/or the EU Big Data for Energy Inducement prize can be viewed as an approach to benchmarking of different solutions.

3.3 Relating Business Benchmarks and Technical Benchmarks

It is a main objective of DataBench to relate Business Benchmarks with Technical Benchmarks. D1.1 from WP1 and work in WP2 and WP4 is in particular focusing on this.

[3] https://www.kaggle.com/

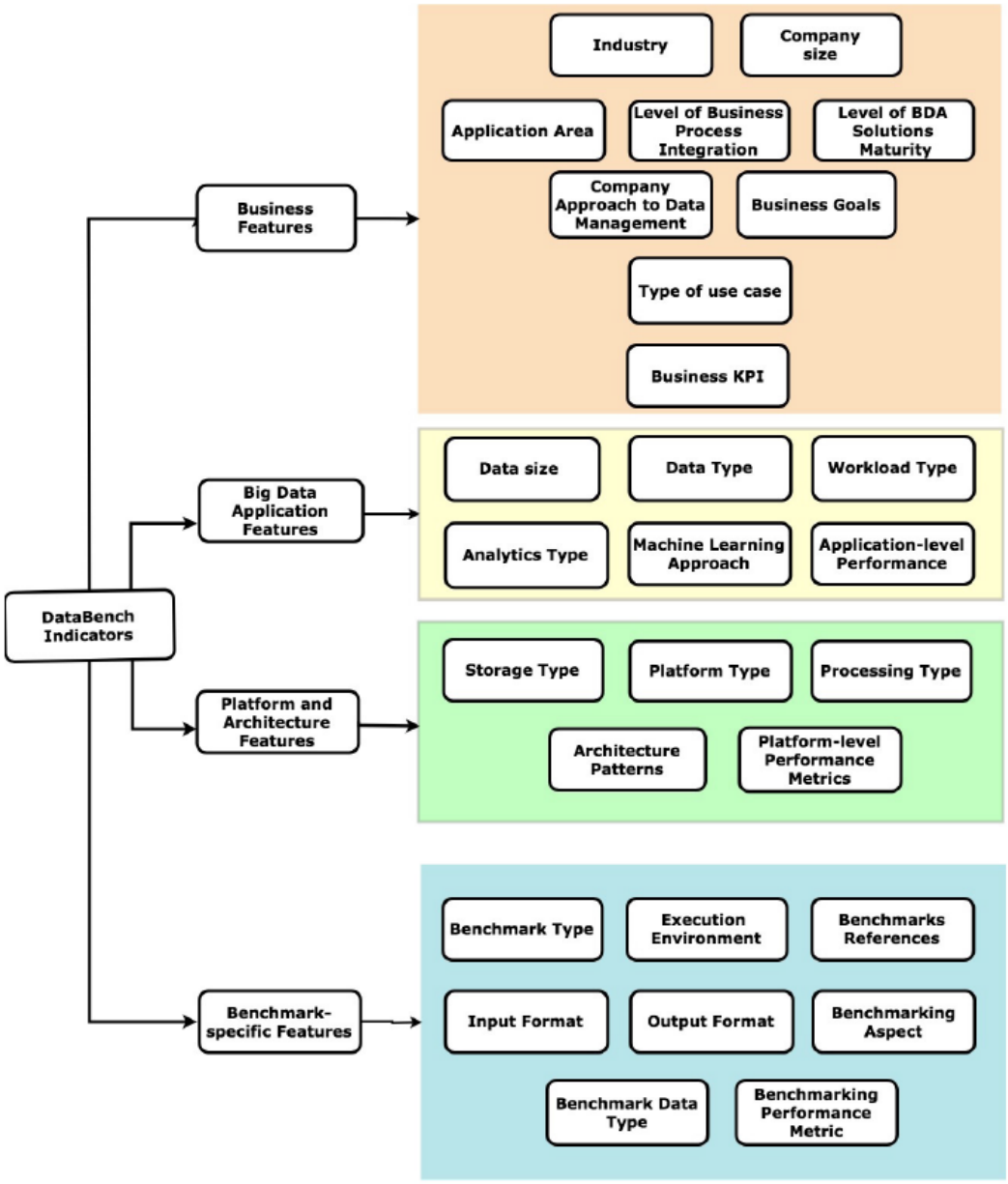


Figure 5 - DataBench Indicators Ecosystem (from D1.1)

Figure 5 shows the DataBench indicators ecosystem, which is further described in D1.1 – showing type of use cases as an element of business features, and data types as an element of Big Data application features. It further shows the aspect platform and architecture features, in particular the processing type and storage type (RDB, NoSQL, NewSQL, File etc), as well as platform type (Distributed, Centralised, Spark, Flink, …) , architectural patterns (Data pipelines, Lambda/Kappa architecture, ….) and platform-level performance metrics. These aspects are important to associate with each benchmark in addition to benchmark specific features, such as the particular execution environment and the specific performance metrics (D1.1 – tables 5 and 6).

Figure 6 - Big Data Application features (from D4.1)

Figure 6 shows Big Data Application features which is further described in D4.1 – showing the data types as an important element of Big Data application features, with further links to workload types, analytics types, data size, application level performance and machine learning approaches.

3.4 Benchmarking Organisations

This section describes relevant benchmarking organisations:

- Transaction Processing Performance Council (TPC)
- Standard Performance Evaluation Corporation (SPEC)
- Securities Technology Analysis Center (STAC)
- Linked Data Benchmark Council (LDBC)
- BDVA Big Data Benchmarking Sub Group (BDVA TF6 SG7)
- International Open Benchmarking Council (BenchCouncil)
- Hobbit platform and community (Hobbit)

Transaction Processing Performance Council

TPC (Transaction Processing Performance Council) [9] is a non-profit corporation operating as an industry consortium of vendors that define transaction processing, database and Big Data system benchmarks. TPC was formed on August 10, 1988 by eight companies convinced by Omri Serlin [9]. In November 1989 was published the first standard benchmark TPC-A with 42-pages specification [2]. By late 1990, there were 35 member companies. As of 2018, TPC has 21 company members and three associate members. There are six obsolete benchmarks (TPC-A, TPC-App, TPC-B, TPC-D, TPC-R and TPC-W), 14 active benchmarks TPC-C [19], TPC-E [20], TPC-H [21], TPC-DS [22]–[24], TPCDI [25], TPC-V [26], TPCx-HS [27], TPCx-BB [28] and two common specifications (Pricing and Energy) used across all benchmarks. Table A.1 lists the active TPC benchmarks grouped by domain.

Benchmark Domain	Specification Name
Transaction Processing (OLTP)	TPC-C, TPC-E
Decision Support (OLAP)	TPC-H, TPC-DS, TPC-DI
Virtualization	TPC-VMS, TPCx-V, TPCx-HCI
Big Data	TPCx-HS V1, TPCx-HS V2, TPCx-BB, TPC-DS V2
IoT	TPCx-IoT
Common Specifications	TPC-Pricing, TPC-Energy

Table 7 - Active TPC Benchmarks

Standard Performance Evaluation Corporation (SPEC)

The SPEC (Standard Performance Evaluation Corporation) [18] is a non-profit corporation formed to establish, maintain and endorse standardized benchmarks and tools to evaluate performance and energy efficiency for the newest generation of computing systems. It was founded in 1988 by a small number of workstation vendors. The SPEC organization is umbrella organization that covers four groups (each with their own benchmark suites, rules and dues structure): the Open Systems Group (OSG), the High-Performance Group (HPG), the Graphics and Workstation Performance Group (GWPG) and the SPEC Research Group (RG). As of 2018, there are around 19 active SPEC benchmarks listed in Table A.2.

Benchmark Domain	Specification Name
Cloud	SPEC Cloud IaaS 2016
CPU	SPEC CPU2006, SPEC CPU2017
Graphics and Workstation Performance	SPECapc for SolidWorks 2015, SPECapc for Siemens NX 9.0 and 10.0, SPECapc for PTC Creo 3.0, SPECapc for 3ds Max 2015, SPECwpc V2.1, SPECviewperf 12.1
High Performance Computing, OpenMP, MPI, OpenACC, OpenCL	SPEC OMP2012, SPEC MPI2007, SPEC ACCEL
Java Client/Server	SPECjvm2008, SPECjms2007, SPECjEnterprise2010, SPECjbb2015
Storage	SPEC SFS2014
Power	SPECpower ssj2008
Virtualization	SPEC VIRT SC 2013

Table 8 - Active SPEC Benchmarks

Securities Technology Analysis Center (STAC)

The STAC Benchmark Council [29] consists of over 300 financial institutions and more than 50 vendor organizations whose purpose is to explore technical challenges and solutions in financial services and to develop technology benchmark standards that are useful to financial organizations. Since 2007, the council is working on benchmarks targeting Fast Data, Big Data and Big Compute workloads in the finance industry. As of 2018, there are around 11 active benchmarks listed in Table A.3.

Benchmark Domain	Specification Name
Feed handlers	STAC-M1
Data distribution	STAC-M2
Tick analytics	STAC-M3
Event processing	STAC-A1
Risk computation	STAC-A2
Backtesting	STAC-A3
Trade execution	STAC-E
Tick-to-trade	STAC-T1
Time sync	STAC-TS
Big Data	in-development
Network I/O	STAC-N1, STAC-T0

Table 9 - Active STAC Benchmarks

Linked Data Benchmark Council (LDBC)

The Linked Data Benchmark Council (LDBC) [30] is a non-profit organization dedicated to establishing benchmarks, benchmark practices and benchmark results for graph data management software. As of 2018, there are three standardized benchmarks listed with more details in Table A.4 and 9 active member companies and organizations.

Benchmarks	Workload Description
Graphalytics benchmark	Breadth-first search, PageRank, weakly connected components, community detection using label propagation, local clustering coefficient, and single-source shortest paths
Semantic Publishing Benchmark (SPB)	Testing the performance of RDF engines inspired by the Media/Publishing industry
Social Network Bench mark	Interactive Workload Business Intelligence Workload Graph Analytics Workload

Table 10 - Active LDBC Benchmarks

BenchCouncil[4]

The International Open Benchmarking Council (BenchCouncil) is a non-profit benchmarking organization, which aims to promote multi-disciplinary benchmarking research and practice and foster collaboration and interaction between industry and academia.

BenchCouncil is a new initiative since 2018 with a Chinese foundation and elements of US participation and a plan for further international participation recruitment.

Current and planned activities of the BenchCouncil include:

- Establish and maintain a repository of benchmark specifications for quantitative system and algorithm evaluation and analysis.
- Review, shepherd, and release open-source benchmark implementations.
- Publish newsletters and research articles in the area of benchmarking.
- Organize conferences, workshops, and teleconferences fostering the transfer of knowledge between industry and academia in the areas of benchmarking.
- Organize challenges and competition using released benchmarks.

BenchCouncil publishes a journal: BenchCouncil Transaction on Benchmarking, measuring, and Optimizing (in short, TBench). BenchCouncil organizes BenchCouncil main conference (Bench) and BenchCouncil annual System Technology conference (BenchCouncil ATC).

BDVA TF6 SG7: Big Data Benchmarking Sub Group

The BDVA Task Force 6 – TF6[5] focuses on the BDVA Technical Priority areas for Big Data, including technical aspects and standards for Big Data. The Big Data Benchmarking sub group was established in 2018.

[4] http://www.benchcouncil.org/
[5] http://www.bdva.eu/task-force-6

SG7 within TF6 will focus on Big Data benchmarking and will merge the efforts of several data benchmarking projects under one umbrella within which HOBBIT will take the lead of linked data benchmarking and the DataBench project on other areas of Big Data benchmarking.

The subgroup description, objectives and activity plan have been officially presented during the BDVA Activity Group Meeting in March, 2018 by the SG7 leads Axel Ngonga (from the Hobbit Community) and Arne Berre (from DataBench). The main objectives of the group were presented as follows:

- Researching business and technical Big Data benchmarks
- Monitoring European performance in Big Data technologies
- Provision of benchmarks, performance indicators, tools and services

Hobbit platform and community

The Hobbit platform [6]6 is an open source distributed benchmarking platform for comparable benchmarking of Big Linked data solutions across the Big Linked Data lifecycle. The benchmark is designed for:

- benchmarking any step of linked data lifecycle (generation & acquisition, analytics & processing, storage & curation, visualization and services).
- ensuring that evaluation results can be found, accessed, integrated and reused easily.

There are two types of versions available offline and online. The offline version can be downloaded and executed locally. The online platform instance provides the opportunity to run public benchmarking challenges, and also to be able to run benchmarks even if one does not have required infrastructure available. The detailed information about its features and its usage is available at project deliverable available online7 and the platform wiki8 .

The platform is written in Java and RabbitMQ has been used for internal communication between the components. The HOBBIT platform has been used to carry out many benchmarking challenges for Linked Data. The different colors in the diagram stand for different parts of the platform. The blue components on the right side are the platform components that offer the core functionality. The orange components belong to a benchmark system and are instantiated when the benchmark is running. The grey component is the system that is benchmarked by the orange benchmark. The benchmarked system might comprise multiple distributed components.

The strategy of HOBBIT is to work very closely with **industry partners** that will participate in open calls and help to define Key Performance Indicators (KPIs) in order to evaluate within the Linked Data life cycle through benchmarks **related to use cases and specific industry solutions**.

[6] https://project-hobbit.eu
[7] https://project-hobbit.eu/wp-content/uploads/2017/04/D2.2.1.pdf
[8] https://github.com/hobbit-project/platform/wiki

Open calls involve the industry with use case and solution providers as well as scientific partners to work on the Linked Data KPI benchmarks. By this close collaboration it is an aim to establish a benchmarking community where all participants can benefit from the reported results.

3.5 Application (Domain) Benchmarks and Big Data Types

The application benchmarks are typically focused on an industry, domain specific use case, which represents common operations and challenges.

Historically, the TPC (Transaction Processing Performance Council) was one of the first organizations that started standardizing application benchmarks for measuring database systems performance and feature capabilities in 1988. The first benchmarks (TPC-A and TPC-B) were simulating scenarios from the banking industry with entry bookkeeping (credits and debits) with focus on the speed of the performed transactions. The later benchmarks like TPC-C, TPC-D and TPC-H simulate a business model of a wholesale parts supplier that operates out of a number of warehouses and their associated sales districts. However, all these benchmarks were designed to work with highly structured table data, whereas currently with all the emerging Big Data scenarios the variety of data formats increases on a daily basis. The BDVA Reference Model identifies six different data types that cover all major categories of industry data representations. These data types are as follows:

- Structured Data/ Business Intelligence
- Time Series, IoT
- Geo Spatial Temporal
- Media, Images, Audio
- Text, Language, Genomics
- Web, Graph, Meta

The diversity of data types requires for the new standardized benchmarks to cover not only the industry use case specifics but also the various data formats and representations that are processed by the company data platforms. Therefore, in the latest Big Data related benchmarks multiple data types are represented and stress the platform capabilities to handle the storage, processing and management of heterogeneous data formats.

3.6 Use Cases in Application Domains/Industrial Sectors

A number of different approaches for the categorisation of application domains and different industrial sectors have been proposed in different contexts. The DataBench project has through the initial work in D1.1 and WP2 started with the industry classifications used in international statistics. Mapping approaches to other classifications like the ones used in the ISO SC42 AI and Big Data standardisation and in the European Big Data Value community has shown that it can be useful to support different mappings – and this work will further continue in WP2.

The WG4 of ISO SC42 AI (and Big Data) described below has created an overview of more than 60 different application scenarios from more than 12 different industry sectors/application domains which we also will follow up for further mappings to relevant

benchmarking areas. A further enhancement of these industry use cases is now being worked on by the BDVA community.

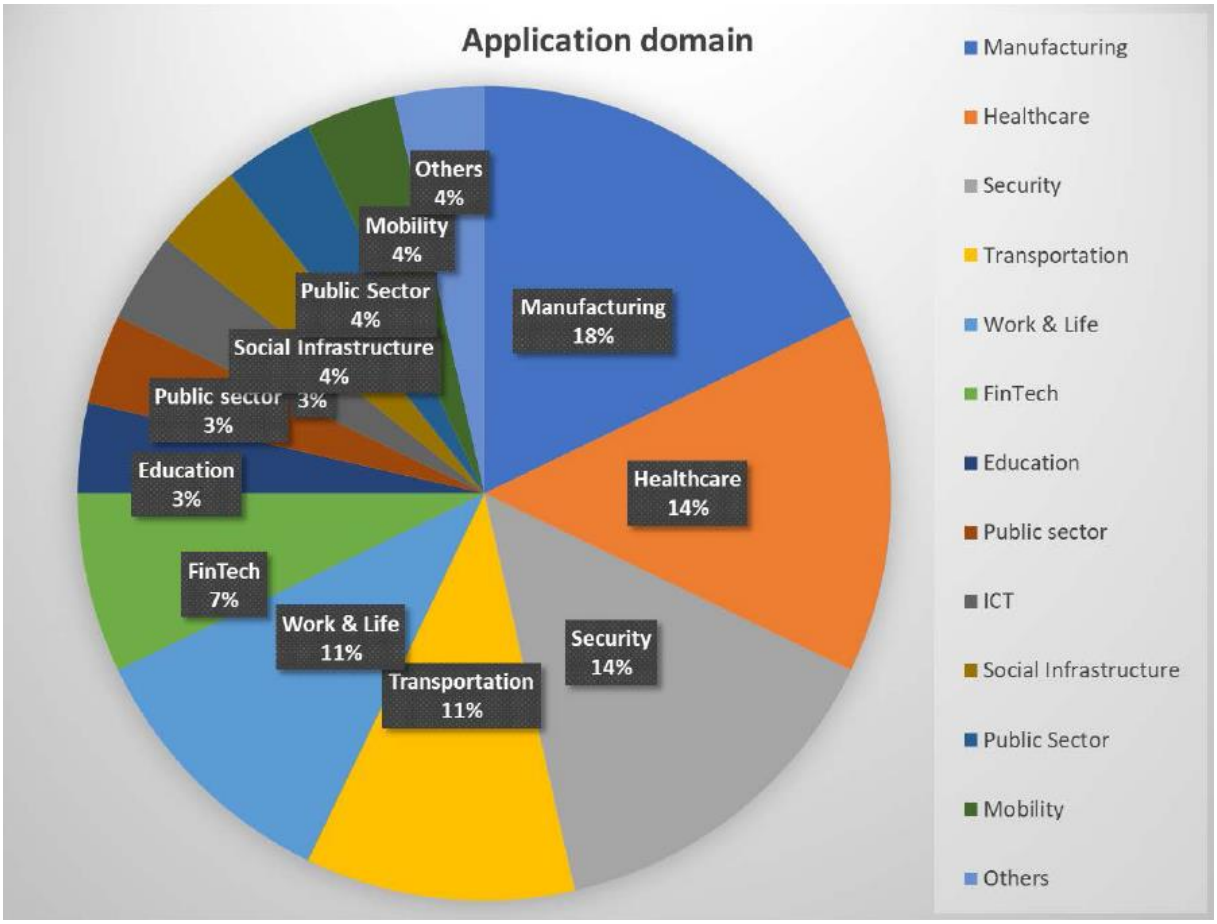


Figure 7 - Application domains/Industry sectors with use cases in ISO SC42

3.7 Big Data Standards

The initial international standardisation work on Big Data was initiated through the ISO JTC1 WG9 Big Data work. This has during 2018 and 2019 migrated into the new ISO SC42 Artificial Intelligence as a separate working group – WG2 – Big Data.

The following ISO/IEC SC42 Standards are in progress – including Big Data standards:

- Artificial intelligence -- Concepts and terminology (ISO 22989),
- Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML) (ISO 23053),
- Bias in AI systems and AI aided decision making (ISO 24027),
- Governance implications of the use of artificial intelligence by organizations (ISO 38507),
- Big data reference architecture -- Part 3: Reference architecture (ISO/IEC DIS 20547-3).

The following ISO/IEC SC42 Technical Reports (TRs) are in progress:

- AI -- Overview of trustworthiness in Artificial Intelligence (ISO TR 24028),
- AI -- Assessment of the robustness of neural networks -Part 1: Overview (ISO TR 24029-1),
- AI - Artificial Intelligence Use cases (ISO TR 24030),
- Big Data reference architecture -- Part 1: Framework and application process (ISO TR 20547-1)

The Big Data standards and technical reports are now embedded together with the AI standards and technical reports.

CEN CENELEC has started a European initiative on AI to follow up with a European strategy for the relationship and input to ISO SC42 from a European perspective.

In BDVA this is since 2018 being followed up by a dedicated subgroup SG6 of TF6 on Big Data Standardisation.

The SC42 WG2 Big Data reference architecture maps well into the BDVA Reference Architecture and thus to the DataBench Framework benchmarking categorisation.

3.8 Challenges and Inducement Prizes

In recent years, with the emergence of many new Big Data scenarios and applications, the system requirements have changed drastically guided by the challenges of the 3V's Big Data characteristics (Volume/Variety/Velocity). However, the existing software frameworks and data platforms are not able to address these requirements opening the space for new technologies and innovative solutions. In order to inspire such creative solutions, many companies and public organizations have started organizing challenges with money prizes for the best solutions. A typical challenge consists of very precise descriptions of the application scenario together with sample data sets, requirements for the execution environment including data formats, storage and processing. The provided data sets are anonymized and designed specifically to represent the application challenges. In some cases requirements for particular technology libraries are set or pre-configured cloud environments are available for developing and testing purposes. The application requirement and challenges to solve are clearly defined usually in the form of questions and additional descriptions that in some cases are automatically checked on submission. There are no strict rules on how the challenges should be defined or executed, which makes them very popular and motivating for the participants.

The Kaggle[9] platform was one of the first platforms to offer assistance and infrastructure for the organization and execution of challenges. Its major features are the list of competitions (listing top 20 active competitions), list of datasets uploaded and used in the competitions, kernels which represent machine learning code that can be executed directly in the platform to reproduce results, list of data science courses and additional resources like discussions and documentation on how to use the platform. A big advantage of the platform is that everything is open and the available datasets and code solutions of the challenges can be downloaded and used for other challenges and similar problems. For example MLBench [31] benchmark uses Kaggle competition datasets and their best solutions as a baseline of both feature engineering and machine learning models. What is even more interesting is

[9] https://www.kaggle.com

that the authors propose a novel performance metric based on the notion of "quality tolerance" that measures the performance gap between a given machine learning system and top-ranked Kaggle performers. MLBench demonstrates that results from challenge competitions are very suitable for other purposes like basis for benchmarks, because they include very well documented industry scenario, realistic dataset and finally proved and validated solution in form of execution code.

The Hobbit project has organized a number of benchmark challenge[10] representing industry-relevant Big Linked Data use cases as part different conferences and workshop. The Hobbit platform was used as the main technology platform on which were implemented and executed all team solutions. The best solutions to the challenges are available on the platform as example benchmarks.

The EU Commission organized during 2018 a Big Data Inducement Prize Challenge. The so-called "Horizon prize for Big Data technologies"[11].

Horizon Prizes are challenge prizes (also known as inducement prizes) offering a cash reward to whoever can most effectively meet a defined challenge. The aim is to stimulate innovation and come up with solutions to problems that matter to European citizens.

The challenge for the applicants for the Big Data inducement prize was to create software that could predict the likely flow of electricity through a grid taking into account a number of factors including the weather and the generation source (i.e. wind turbines, solar cells, etc). Using a large quantity of data from electricity grids combined with additional data such as weather conditions, applicants had to develop software that could predict the flow of energy through the grid over a six-hour period. The winners were selected based on combined rank of accuracy and speed, with greater weight being given to accuracy.

Many domains of societal or industrial significance, from epidemiology, to climate change, to transportation to energy production and transmission benefit from our ability to examine historical records and predict how the system under study will evolve.

In all these cases, it is not sufficient for predictions be accurate: they also need to be delivered fast enough for corrective action to be applied on the system observed.

This inducement prize also complements the activities of the Big Data Value Public Private Partnership (PPP) which aims to develop Europe's data driven economy and the prospects offered by Big Data technologies.

The solution selected demonstrates the ability to analyse extremely large scale collections of structured or geospatial temporal data in a way that is sensitive to the trade-off between the consumption of computational resources and the practical value of the predictions obtained. This not only results in the more efficient management of those domains in which spatio-temporal predictions are already used, but also in the applications of such predictive methods where today they are not, due to current limitations of speed, scalability, accuracy and resource efficiency. The analytics tasks and computational environment of the challenge

[10] https://project-hobbit.eu/challenges/
[11] https://ec.europa.eu/info/research-and-innovation/funding/funding-opportunities/prizes/horizon-prizes/big-data-technologies_en

were developed in the framework of SEE.4C-688356 Horizon 2020 Coordination and Support Action.

The first prize of €1.2 million went to Professor José Vilar from Spain, while data scientist Sofie Verrewaere and post-doctoral researcher Yann-Aël Le Borgne, both from Belgium, came in joint second place and won €400,000 each.

Currently, there are many organizations (Frankfurt Big Data Lab data challenges[12]) and companies (Ekipa Challenges[13]) organizing in data challenges which provide clear problem statements and solutions after that can be eventually adapted and used as inspirations and basis for new benchmarks.

4. Vertical Benchmarks

In the following sections we categorize the vertical benchmarks according to different data types specified in the BDVA reference model. However, we only focus on most appropriate and relevant benchmarks that satisfy a set of criteria. First, they need to be publicly available in the form of source code or/and execution binaries. Second, they should be regularly updated in terms of bug fixing, usability improvements and new functional extensions. Third, there should be available user documentation, installation and usage guides that accurately describe how to apply the benchmark. Finally, the benchmark should be popular among users in terms of reported results, vendor comparisons and scientific papers, which basically suggests that the benchmark offers a good baseline for comparison and is accepted as a standardized measurement tool.

4.1 Structured Data Benchmarks

TPC (Transaction Processing Performance Council) has defined various benchmarks over the time for structured data applications. TPC benchmarks are mainly designed to model real-world transaction processing applications. TPC-H, TPC-C, TPC-DS and TPC-DS v2 are the most popular benchmarks. New TPC benchmarks such as TPCx-HS and TPCx-BB ,covering more data types and testing new system features, were introduced.

TPC-H benchmark involves a decision support system, providing ad-hoc queries as a workload over relational database tables. The benchmark has a synthetic data generator fo generating different datasets for benchmarking. *TPC-DS benchmark* is also focused on business transaction applications. The workload involves transactional queries over tables. The data generator is capable of generating data of variable volumes by providing different scale factors. *TPCx-BB benchmark* covers structured, semi-structured and un-structured data. The structured part of the benchmark is based on TPC-DS which mimics a business retail model application. TPCx-BB workloads span three types of tests_ Load test, Power test and Throughput test, for evaluating performance at data, system and operational level.s

Other than TPC benchmarks, *Pavlo benchmark* is worth mentioning. It was developed to specifically compare the capabilities of Hadoop with those of commercial parallel Relational

[12] http://www.bigdata.uni-frankfurt.de/data-challenges/
[13] https://app.ekipa.de/challenges

Database Management Systems (RDBMS). It concentrates mainly on comparison of Hadoop-based data analysis systems, using structured data sets.

4.2 IoT/Time Series and Stream processing Benchmarks

The expansion of IoT world has a ripple effect in Big Data ecosystem. As the world of IoT is expanding, the number of Big Data solutions, dealing with real-time streaming and timeseries datasets and applications, are also increasing. There are many real-time distributed stream processing systems (DSPSs) available, Apache Spark Streaming, Apache Flink, Apache Storm and SANZA to name a few. For the storage of IoT/time series datasets, a wide range of storage solutions (NoSQL systems, Time-series databases, etc.) have emerged. This has given rise to the need for Big Data benchmark solutions, targeted towards evaluating the performance of IoT and stream processing and storage systems.

Stream Processing Benchmarks

Yahoo Streaming Benchmark (YSB) is a benchmark for evaluating stream processing systems. It is based on an end-to-end processing pipeline composed of a distributed streaming platform (Apache Kafka), a key-value database (Redis) and computation engines (Flink, Storm and Spark Streaming). The idea of this pipeline is to simulate a real-world advertisement analytics pipeline. The set of results are the comparison of the latency that a particular processing system can produce at a given input load.

SparkBench evaluates Apache Spark performance against different workload configurations. The benchmark provides four categories of workloads: Machine Learning, Graph Computation, SQL Query, Streaming Application. The metrics calculated are: Job execution time in seconds, Data process rate in MB/Second.

StreamBench experimental comparison of two stream processing frameworks platforms, namely Apache Spark and Apache Storm. One unique aspect of the benchmark is that the generated data is loaded into a queue, which decouples the data generation and data consumption processes. The workload varies according to three factors (data type, historical data usage, workload complexity). The metrics measured are: Latency, throughput, durability and fault tolerance.

IoTAbench was developed for evaluating Big Data analytics platforms for the Internet of Things. The benchmark has used smart metering as a use case, and involves generating,

loading, repairing and analyzing synthetic meter readings. The work evaluated the HP Vertica 7 Analytics platform for a scenario involving an electric utility with 40 million meters. The benchmark consists of three components: a scalable synthetic data generator; a set of SQL queries; and a test harness. It uses a Markov chain-based synthetic data generator to generate sensor datasets. The workload can be mainly categorized as loading, repairing and analyzing tasks. Metrics calculated are performance in terms of Query Execution Times (in seconds or milliseconds).

RioTBench is a real-time IoT Benchmark suite to evaluate distributed stream processing systems for IoT applications. The benchmark includes 27 common IoT tasks classified across various functional categories and implemented as reusable micro-benchmarks. Further, there are four IoT application benchmarks composed from these tasks, and that leverage various dataflow semantics of DSPS. The applications are based on common IoT patterns for data pre- processing, statistical summarization and predictive analytics. These

are coupled with four stream workloads sourced from real IoT observations on smart cities and fitness, with peak streams rates (ranging from 500 – 10, 000 messages/sec) and diverse frequency distributions. The benchmark has been validated for Apache Storm on the Microsoft Azure public Cloud.

NoSQL for IoT/Time Series

Yahoo! Cloud Serving Benchmark (YCSB) is a one well-known benchmarking system originally designed for direct performance evaluation of NoSQL stores. It has become de-facto standard benchmark for evaluating performance characteristics of NoSQL database systems (Apache HBase, Cassandra, Riak, MongoDB, etc.). The benchmark has a workload generator and a basic database interface, which cen be extended to support various NoSQL and relational database systems. It provides six pre-defined workloads, which simulate a cloud OLTP application (read and update operations). The reported metrics are: Execution time, Latency if request under load, Throughput (operations per second), Scaling. There have been several research efforts to extend YCSB. Currently more than 20 types of NoSQL systems have been tested by extending YCSB original benchmark.

4.3 Spatio-Temporal Benchmarks

This category of benchmarks aim to evaluate spatial databases such as the Sequoia 2000 benchmark [33] and the Paradise Geo-Spatial DBMS benchmark [34]. Werstein [35] provides a good overview of the popular spatio-temporal benchmarks and an extension with 36 queries of the two most common one, the Sequoia 2000 and Paradise Geo-Spatial DBMS benchmarks.

The *COST* benchmark [36] is focused on comparing the different spatio-temporal indexes, where the *BerlinMOD benchmark* [37] is one of the latest approaches with scalable data generator and simulating complete histories of movements, allowing for complex analyses of movements in the past.

Systems like GeoSpark [38], Spatial Spark[14], and Spatial Hadoop[15] have emerged to cope with challenges related to data with spatio-temporal dimensions. This requires a deeper understanding and evaluation [32], [39] of these new technologies in order.

4.4 Media/Image Benchmarks

MLPerf is the first industry standard machine learning benchmark suite for measurement of system performance of ML software frameworks (TensorFlow, PyTorch, etc.) and power efficiency of ML hardware platforms (Intel CPUs, NVIDIA GPUs, etc.). The benchmark covers a wide range of applications including natural language processing and autonomous driving. The benchmark measures inference which provides insight about the efficiency of a trained neural network to process new data. *MLPerf Inference v0.5* suite consists of five benchmarks, focusing on image classification, object detection and machine translation. The provided reference implementations are available in TensorFlow, PyTorch and ONNX frameworks.

[14] https://github.com/syoummer/SpatialSpark
[15] http://spatialhadoop.cs.umn.edu/

DeepBench benchmarks evaluates the performance of hardware platforms by benchmarking the basic underlying involved in the process of training a deep learning model, using neural network libraries. One of the workload type is calculating latency in speech recognition systems.

DAWNBench provides a suite for end-to-end learning training and inference. The benchmark provides a set of common deep learning workloads including image classification workload with image database (ImageNet[16]) and image datasets (CIFAR10[17]), and question answering workloads with reading comprehension dataset (SQuAD[18]) . The metrics calculated are (a)training time, (b)training cost, (c)inference cost and (d)inference latency for various software frameworks (Moxing, TensorFlow, PyTorch, nCluster, Caffe 1.0, ifx, TensorRT, horovod), cloud solutions, hardware platforms, model architectures and optimization strategies. Training time is the time taken to train an image classification model on an image database. Training Cost is the total cost (in US dollars) of public cloud instances to train an image classification model. Inference Latency is the latency required to classify one image using a model with an specific accuracy percentage or greater. Average cost on public cloud instances to classify a set of validation images using an image classification model with a specified accuracy % or greater.

Deep Learning Benchmarking Suite (DLBS) is a collection of command line tools for running deep learning benchmark experiments on different hardware and software platforms. Supported frameworks are TensorFlow, BVLC Caffe, NVIDIA Caffe, Intel Caffe, Caffe2, MXNet, TensorRT, and PyTorch. The suite has been tested with various operating systems (Ubuntu, ReadHat, CentOs) with and without NVIDIA GPUs. The benchmark can be tuned by configuring parameters like batch size, framework, model name, type of data, data path, etc. The metrics calculate are (a) inference/training times and (b) average training/inference time.

DeepMark is a benchmark for evaluation of deep learning frameworks (Caffe, Chainer, CNTK, MXNet, Neon, Theano, TensorFlow, Torch) against set of hardware platforms (multi-GPU with titan cards). Datasets are a range of images, video, audio and text. Metrics calculated are (a) round-trip time for 1 epoch of training and (b) maximum batch-size that fits (to depict the extra memory consumption that the framework uses)

Other benchmarks with workloads for manipulating image/media data types include Convnet benchmark, Fathom, TBD (Training Benchmark for DNNs), BENCHIP and BigDataBench.

BigDataBench is an open source Big Data benchmark suite. The current version BigDataBench 5.0 provides 13 representative real-world data sets and 27 Big Data benchmarks. The benchmarks cover a wide range of workload types including online services, offline analytics, graph analytics, data warehouse, NoSQL, and streaming from three important application domains, Internet services (search engines, social networks, e-commerce), recognition sciences, and medical sciences. For offline analytics, the suite provides Hadoop, Spark, Flink and MPI implementations. For graph analytics, Hadoop, Spark GraphX, Flink Gelly and GraphLab implementations are provided, and for AI

[16] http://www.image-net.org
[17] https://www.cs.toronto.edu/~kriz/cifar.html
[18] https://rajpurkar.github.io/SQuAD-explorer/

implementations for TensorFlow and Caffe are available. Additionally, for data warehouse Hive, Spark-SQL and Impala implementations, for NoSQL stores MongoDB and HBase implementations, and for streaming applications Spark streaming and JStorm implementations are provided.

Categorisation and Summary of Media/Image Benchmarks

The table 6 shown below categorizes and provides a summary of all the deep learning benchmarks related to Image analytics.

Name	Compares	Workload type	Metrics	Frameworks
DeepBench	Hardware	DNN libraries	Milisec,Flops,GB/s	
TF Benchmark	Hardware	Classification	Images/second	**Tensorflow**
DeepMark	Models	Classification	Training time/ epoch	**Torch**
Convnet - benchmark	Frameworks	Classification	training time	**Tensorlow,Torch Chainer/Caffe,**
Fathom	Models	Classification	time	**Tensorflow**
Dawnbench	Hardware Cloud	Classification	inference/training time, cost	**Tensorflow,Pytorh**
MLPerf	Hardware	Classification, Object detection	training time	**Tensorflow,PyTorc**
TBD	Hardware, memo	Classification, Object detection	throughput,CPU /GPU utilization	**Tensorflow,MXNet**
BENCHIP	Hardware	Classification, Object detection	accuracy, energy, performance	**Caffe**
DLBS	Models	Classification	images/sec	**All popular**
BigDataBench v 3.2 and greater	Hardware	Classification, Image generation	Utilization, frontend bound, backend bound	**Tensorflow,Caffe**

Table 6 - Deep learning benchmarks related to Image Analytics

4.5 Text/NLP Benchmarks

The area of natural language processing (i.e. NLP) transitioned from very model driven type of techniques (typically with manually written rules) in 1980ties into almost completely data driven statistical type of techniques after year 2000. The transition period (so called "statistical revolution" in the late 1980s and during 1990s significantly changed the landscape of the area of text processing and language modelling along a series of tasks relevant as academic problems or for industry.

Conceptually, the key breakthrough in data driven text processing was machine learning with sparse representations which allowed seamless mapping of textual document into standard machine learning problems. Techniques such as Support Vector Machine (mid 1990s) and Logistic Regression contributed a lot to approach and solve problems like text classification and similar on the datasets which became de facto standards in 1990s. In particular, among the key datasets which were used at that time was the Reuters-21578 dataset, and later in early 2000s Reuters RCV1 dataset served as key benchmarks for the areas of text mining, information retrieval, and broader the area of NLP. The enabler at that time was availability of labeled datasets where Reuters served an important role in contributing a sample of its data to the community. Both Reuters datasets were just a list of news articles (the first one 21578 articles, and the second one over 800,000 articles) manually categorized into 120 Reuters flat classification schema. All the early text processing research groups were competing in how good automatic classifiers to build to reconstruct manual work done by Reuters editors.

In terms of standardized tasks to be attacked by (mostly) academics, the central tasks were supervised text categorization and unsupervised text segmentation (clustering). An important class of tasks was 'information extraction' where the goal was to extract various fragments of information from unstructured texts. The most popular and useful task of this kind was 'named entity extraction' where the goal was to extract names of people, places and organizations (with no semantic alignment yet). Other tasks includes text summarization (compressing information from a long document into a shorter versions).

In the meantime, fairly disconnected from machine learning and text mining community there was a transforming linguistic community which formed an area of 'computational linguistics', where some of the basic linguistic tasks were slowly transformed from rule/model driven into statistical/data driven version. The key tasks in this class to be solved by the community were 'Part-of-Speech tagging' (PoS) and 'Deep-Parsing'. Supporting tasks included word lemmatization, morphological analysis, word sense disambiguation, sentiment analysis, coreference resolution, language modelling. Techniques such as Support Vector Machines, Logistic Regression, Random Forests and later Conditional Random Field (CRF) significantly contributed to transitioning of the computational linguistic community entirely to statistical/data driven approaches.

In 2000s, especially after 9/11 events when US government invested significant amounts of funding in all types of data analytics (including text processing), there was a significant push in all directions to solve text related problems in English and also in other languages of interest (e.g. Arabic). As a consequence of US funding and EU research programs (e.g. FP5 and later) there was an increased interest in dealing with multiple languages and consequently with machine translation. In its initial stages, machine translation was a rule driven machine, later supported by so called translation memories (database of translated

text fragments), but no significant progress was done until mid-2000s when larger datasets became available for training and evaluation and when machine learning techniques reached its maturity. Machine translation benefited especially with deep learning revolution after 2011 where large datasets (mostly collected from the web) were processed with deep neural networks on large clusters of GPUs. This approach contributed to very efficient general machine translation systems while terminologically or context specific documents are still an an open problem.

In the future, the area of NLP/computational linguistics/text mining is heavily moving in the direction of text semantics. In that respect we can say, today's text processing is fairly shallow in comparison with what is to be expected in the future, where the primary will be towards deep and contextual understanding of the textual content.

As part of evolution of the NLP research area, there was a series of benchmarking initiatives, mostly with varying success. The main reason was the availability of the large textual datasets which were typically under copyright by corresponding publishers. In its initial phases (as mentioned before), the key datasets were contributed by Reuters. Later in 2000s more and more datasets appeared. Maybe the key contribution to the area of text processing was made by U.S. National Institute of Standards and Technology (NIST) with its two series of annual challenges:

- Text Retrieval Conference (TREC) <https://trec.nist.gov/>, and
- Text Analysis Conference (TAC) <https://tac.nist.gov/>

…where each year a series of relevant benchmarks were selected by the research community. This series of challenges (running for over 20 years) landscaped many of the areas in text processing.

There are far too many benchmarks in the history of TREC and TAC to be listed here, but as of 2019, the following carefully prepared challenges are available as part of the TREC efforts. These benchmarks/challenges reflect the current state of the technology and interests by the corresponding academic and commercial communities:

- Complex Answer Retrieval Track (http://trec-car.cs.unh.edu/)
- Conversational Assistance Track (http://www.treccast.ai/)
- Deep Learning Track (https://microsoft.github.io/TREC-2019-Deep-Learning/)
- Fair Ranking Track (https://fair-trec.github.io/)
- Incident Streams Track (http://dcs.gla.ac.uk/~richardm/TREC_IS/)
- News Track (http://trec-news.org/)
- Precision Medicine Track (http://www.trec-cds.org/)

In parallel, the TAC initiative has a current 2019 set of challenges/benchmarks for the following tasks:

- Entity Discovery and Linking (EDL) (http://nlp.cs.rpi.edu/kbp/2019/)
- Streaming Multimedia Knowledge Base Population (SM-KBP) (https://tac.nist.gov/2019/SM-KBP/index.html)
- Drug-Drug Interaction Extraction from Drug Labels (DDI) (https://bionlp.nlm.nih.gov/tac2019druginteractions/)

Since the area of NLP is progressing on many fronts, there is hard to point to a particular reference set of challenges/benchmarks which last for a longer period of time. TREC and

TAC are maybe the most representative, while many others (e.g. The New York Times Annotated Corpus at https://catalog.ldc.upenn.edu/LDC2008T19) are less systematic and researchers are selecting them for the evaluation of new methods on per-need basis, based on the problem they are solving.

4.6 Graph/Metadata/Ontology-Based Data Access Benchmarks

For various applications a graph is the native underlying data structure. In the context of Big Data one automatically thinks of social networks where persons are represented by nodes connected to their friends or the structure of the Web where the links between them are the edges (used for example to calculate page rank). However, the more you start to think about graphs and their expressive power, you realize that there are even more applications where graphs can be used. In fact the object oriented world with objects as nodes and their associations building edges indicates that graphs could be used for nearly any modern application. However, whether to choose a graph as underlying structure also highly depends on the aims of the application. Will you need to do graph processing, like traversing the graph, finding the shortest path between nodes and so forth? If yes, a graph structure and according system might be useful.

Lissandrini et al. [40] categorizes the graph management systems into two different classes of functionalities:

- ***Graph processing systems*** analyse graphs to discover characteristic properties like density, average connectivity degree, and modularity. They also perform batch analytics at large scale. Typically systems are: SNAP, GraphLab, Giraph, Graph Engine, and GraphX [41].
- ***Graph databases*** focus on storage and querying tasks with the priority on high-throughput and transactional operations. Examples in this category are: Neo4j, OrientDB, Sparksee, JanusGraph, ArangoDB, and BlazeGraph.

Not only there is great variety of different graph management systems but also as graph data is becoming prevalent, larger, and more complex, the need for efficient and effective graph management is becoming apparent. Therefore, there is a need for benchmarks to compare and understand the differences of the systems [40].

The first class of benchmarks, falling under category of ***Graph processing systems*** observe a graph/network as a data structure to be processed using different analytic methods. Typical property of the graphs analysed in this area of research is to be poor in meta-data / annotation (i.e., nodes and edges typically don't have labels or extra rich meta data). Such representations allow simple transformation of such a graph into a matrix representation (typically sparse) which allows the whole spectrum of linear algebra and other data analytic operators to be applied. The scale of such networks can range from fairly small (few tens of nodes) to very large. In the early days of social network analysis a popular data set was "Zachary's karate club" (https://en.wikipedia.org/wiki/Zachary%27s_karate_club) where most of the early social network methods were developed, tested and compard. After 9/11 events in US, network analysis was an area which progressed a lot and several tools and datasets appeared. Probably the most relevant collection of real-world networks is at Stanford Large Network Dataset Collection (https://snap.stanford.edu/data/) which are often used in academic research.

The second class of benchmarks are under category of ***Graph databases***, where a graph is treated as an annotated database of interconnected nodes with additional metadata. The area overlaps a lot with semantic web efforts and gained significantly in its importance after 2015 along with the popularity of knowledge graphs, where graph databases serve as a typical software data infrastructure. The increase in adoption of semantic technologies with data storage systems, have create a new category of data stores commonly called as ***Ontology Based Data Access (OBDA) systems***. OBDA systems propose state-of-art solutions based on recent innovations in semantic technologies. These systems support representation and reasoning through ontology modelling, providing effective means of key data extraction from huge volumes of complex data. The ontologies are connected to external data stores through a mapping, establishing a sematic relation between data queries issued over the underlying databases and ontology elements. To access data, semantic query language, like SPARQL, is used. **Graph databases** can be categorized as OBDA systems.

Through the last years several benchmarks were created to evaluate graph and metadata based processing systems and databases. *WatDiv* measures how an RDF data management system performs across a wide spectrum of SPARQL queries with varying structural characteristics and selectivity classes [42]. *gMark* is a more holistic domain- and query language-independent framework targeting highly tuneable generation of both graph instances and graph query workloads based on user-defined schemas [43]. The *Linked Data Benchmark Council* (LDBC) published the Semantic Publishing Benchmark for RDF database engines, the Social Network Benchmark for interactive, business intelligence and graph analysis workloads [44], and *Graphalytics* for the comparison of graph analysis platforms [45].

As mentioned earlier, RDF Query languages (like SPARQL) and protocols are being implemented by an increasing number of storage systems, including Graph databases. A growing number of organizations and open web implementations have been adopting these technologies. A wide variety of benchmarks have been created to compare performance of systems that expose linked data service endpoints (commonly for SPARQL endpoints). *Berlin SPARQL Benchmark* (BSBM) compares the performance of RDF and Named Graph stores as well as RDF-mapped relational databases and other systems that expose SPARQL endpoints. Designed along an e-commerce use case. SPARQL and SQL version available *Lehigh University Benchmark* (LUBM) facilitates the performance evaluation of semantic Web repositories with respect to extensional queries over a large data set that commits to a single realistic ontology. *SP2Bench* is a SPARQL performance benchmark. It provides a scalable RDF data generator and a set of benchmark queries, designed to test typical SPARQL operations and RDF data access patterns. *Social Network Intelligence Benchmark* is a benchmark suite developed by CWI and Openlink. The schema is taken from Social Networks for generating test areas ideal for RDF/SPARQL models, and workloads include query processing over highly connected graph. *DBPedia SPARQL Benchmark* (DBPSB) is an SPARQL performance benchmark over DBPedia knowledge base, using query- log mining, clustering and SPARQL feature analysis. The benchmark provides implementation for triple stores (Virtuoso, Sesame, Jena-TDB, and BigOWLIM) with respect to two metrics (a) over-all performance of a tripe store in terms of computing query mixes per hour and (b) query completion before timeout (c) query based performance in terms of query per second.

Some other benchmarks that fall under this category are FedBench, Linked Data Integration Benchmark (LODIB), JustBench, A Benchmark for Spatial Semantic Web Systems, Linked Open Data Quality Assessment (LODQA), FEASIBLE, OntoBench, Fishmark Benchmark, NPD Benchmark, and Texas Benchmark.

The yearly Semantic Web Challenge (https://iswc2019.semanticweb.org/call-for-challenge/) addresses a number of topics as challenges involving Semantic Web data management and processing tasks, such as

- Ontology alignment, Fact checking
- Sentiment and Emotion analysis, Entity resolution
- Link prediction, Attribute prediction and validation
- Query and reasoning scalability
- Energy efficiency of computation (e.g. green processing)
- Stream processing

These challenges, and variations around these, has since 2006 been a target of yearly competitions with awarded winners and corresponding explanations of solutions and technologies.

5. Concluding Summary

The DataBench Framework is based on a combination of both the vertical and horizontal dimensions of the BDVA Reference Model, which uses a set of six different Big Data types to focus on end-to-end support along the horizontal layers of visualisation, analytics, processing and data management.

This D1.2 document – DataBench Framework – with Vertical Big Data Type benchmarks – focuses on the classification of benchmarks according to the six main Big Data types. The mappings from industry sectors and application types is made through their usage of various combinations of these Big Data types, and has thus in particular presented technical benchmarks mapped into vertical benchmark groups, following the Big Data type dimensions, Structured data - IoT/Time series - Geo Spatial Temporal - Media, Images, Audio - Text, Language, Genomics - Web, Graph, Metadata. Existing and new benchmarking approaches and challenges are being continuously mapped into the DataBench Framework matrix showing the relationship to the focus aspects of these.

This document has also presented different Benchmarking approaches, including types of technical benchmarks and the relationship to business benchmarks. Different benchmarking organisations, like TPC, SPEC, STAC, LDBC, BenchCouncil, BDVA-TF6-SG7 and Hobbit., Application benchmarks and Big Data types, use cases and application domains, Big Data standards (ISO SC42), Challenges and inducement prices,

Annex 1 contains structured descriptions of all of the technical benchmarks – sorted by the year that they were introduced. The intention is that this Annex will be continued to be updated separately, and serve as a source of detailed information for all of the identified and referred benchmarks.

5.1 Introduction to D1.3 and D.1.4

This document "D1.2 DataBench Framework – with Vertical Big Data Type benchmarks" has been extended through the documents " D1.3 Horizontal Benchmarks – Analytics and Processing" and "D1.4 Horizontal Benchmarks – Data Management" that are being provided at the same time as this document.

The deliverable **D1.3 - Horizontal Benchmarks – Analytics and Processing** [65] presents the various horizontal benchmarks in the area of analytics and processing.

The deliverable D1.3 is focusing on benchmarks in the horizontal layers according to the BDVA reference model with data visualisation (visual analytics), data analytics and data processing. Visual Analytics is an area that has been less focused in existing benchmarks, but an existing starting point for this can be found in the Hobbit-IV benchmark on visualisation and services, which also focuses on question answering and faceted browsing. Data analytics include a level of industrial analytics with descriptive, diagnostic, predictive and prescriptive analytics and the support for this with the use of machine learning. Machine Learning including supervised and unsupervised learning as well as reinforcement learning and has a strong focus in many ongoing ICT14 and ICT15 projects. Analytics is addressed for graph representations in the Hobbit-II benchmark on Graphalytics, but is also a focus in benchmarks on deep learning like DeepMark and DeepBench. Different analytic benchmarks will typically address different Big Data types such as time series, spatial, image and text. The area of data processing architectures includes benchmarks for real time

processing with stream processing, batch processing and interactive processing and main memory architectures. These are areas covered in many benchmarks such as BigBench, BigDataBench and SparkBench – benchmarking different processing architectures such as MapReduce (Hadoop), SPARK and Flink and others.

The D1.3 deliverable includes a focus on the the following areas for benchmarking:

- Data Visualization (visual analytics),
- Data Analytics - (including Machine Learning and AI benchmarks)
- Data Processing
-

The deliverable **D1.4- Horizontal Benchmarks – Data Management** [66] presents the various horizontal benchmarks in the area of data management, including data acquisition and curation and data storage for various classes of storage systems.

The deliverable D1.4 is responsible for the initial classic layer of Big Data benchmarks related to data management – both for data acquisition and curation and for data storage. This is the classical area of database benchmarking and will include a number of existing database benchmarks for various types of SQL and NoSQL storage types and file systems. Different indexing and retrieval schemes will be benchmarked for the various Big Data types. The historical successful benchmarks such as the TPC-series of benchmarks with BigBench and BigDataBench and many others. The Linked Data/Graph database benchmarks focuses here on the performance of Graph databases and RDF storage. The suite of horizontal benchmarks adapted for this will be representative of all relevant data management solutions relevant for the industrial requirements.

The D1.4 deliverable includes the following benchmark areas:

- Data Protection: Privacy/Security Management Benchmarks related to data management
- Data Management: Data Storage and Data Management Benchmarks
- Cloud/HPC
- Edge and IoT Data Management Benchmarks

Relevant elements from the current D1.2 document has been a basis for D1.3 and D1.4, in particular with a mapping of the vertical Big Data types into the various horizontal benchmarks.

5.2 Further Work

The results of D1.2 will be further used in WP2 related to the relationship to business benchmarks and KPIs and business requirements related to economic, market and business analysis, it will feed into the WP3 Data Bench Toolbox for the implementation support for the DataBench Framework, to the WP4 for the evaluations of business performance and to the WP 5 for the technical evaluations with the DataBench Toolbox. The further support and consensus building with the involved communities will be managed by WP6.

With the current priority work on the future AI PPP within BDVA resulting in a new AI Strategic Research, Innovation and Deployment Agenda (SRIDA) and proposals for a new AI PPP (Public Private Partnership) in December e 2019, the DataBench project will introduce

an extra priority on the AI related benchmarks in the area of data analytics, machine learning and AI, further also in 2020-

The active startup of the ISO SC42 Artificial Intelligence standardisation activities in 2019, including the embedding of the WG2 Big Data group, the DataBench project will also ensure a contribution into this community related to reference models/frameworks and benchmarking.

DataBench will continue the active lead and involvement of the BDVA TF6 SG7 group on Big Data benchmarking, and also here include the area of AI technology benchmarking, related to the priorities and wishes of the BDVA community.

The emphasis for the continued work in 2020 will be on the selection and execution and performance analysis of releveant benchmarks identiied in the DataBench Framework, as related to both the business and technical KPIs relevant for various projects and activities.

6. References

[1] B. Andersen and P.-G. Pettersen, *Benchmarking Handbook*. Springer Netherlands, 1995.

[2] "The Benchmark Handbook, Second Edition." [Online]. Available: https://jimgray.azurewebsites.net/BenchmarkHandbook/TOC.htm. [Accessed: 04-Jun-2019].

[3] P. K. Ahmed and M. Rafiq, "Integrated benchmarking: a holistic examination of select techniques for benchmarking analysis," *Benchmarking Qual. Manag. Technol.*, vol. 5, no. 3, pp. 225–242, Sep. 1998.

[4] M. J. Spendolini and M. J. Spendolini, *The benchmarking book*, vol. 4. Amacom New York, NY, 1992.

[5] G. H. Watson, *Strategic benchmarking: How to rate your company's performance against the world's best*. Wiley, 1993.

[6] C. J. McNair and K. H. Leibfried, *Benchmarking: A tool for continuous improvement*. John Wiley & sons, 1992.

[7] N. Poggi, "Microbenchmark," in *Encyclopedia of Big Data Technologies.*, S. Sakr and A. Y. Zomaya, Eds. Springer, 2019.

[8] R. Han, L. K. John, and J. Zhan, "Benchmarking big data systems: A review," *IEEE Trans. Serv. Comput.*, vol. 11, no. 3, pp. 580–597, 2017.

[9] "TPC-Homepage V5." [Online]. Available: http://www.tpc.org/. [Accessed: 04-Jun-2019].

[10] "TPC-H - Homepage." [Online]. Available: http://www.tpc.org/tpch/. [Accessed: 04-Jun-2019].

[11] "TPC-DS - Homepage." [Online]. Available: http://www.tpc.org/tpcds/. [Accessed: 04-Jun-2019].

[12] "TPCx-BB - Homepage." [Online]. Available: http://www.tpc.org/tpcx-bb/default.asp. [Accessed: 04-Jun-2019].

[13] Intel, "HiBench," 04-Jun-2019. [Online]. Available: https://github.com/Intel-bigdata/HiBench. [Accessed: 04-Jun-2019].

[14] IBM, "SparkBench — Bitbucket." [Online]. Available: https://bitbucket.org/lm0926/sparkbench/src/master/. [Accessed: 04-Jun-2019].

[15] M. Ferdman *et al.*, "Clearing the Clouds: A Study of Emerging Scale-Out Workloads on Modern Hardware," Aug. 2018.

[16] "BigDataBench | A Scalable Big Data and AI Benchmark Suite, ICT, Chinese Academy of Sciences." .

[17] "pumabenchmarks - Faraz Ahmad." [Online]. Available: https://engineering.purdue.edu/~puma/pumabenchmarks.htm. [Accessed: 04-Jun-2019].

[18] "SPEC - Standard Performance Evaluation Corporation." [Online]. Available: https://www.spec.org/. [Accessed: 04-Jun-2019].

[19] F. Raab, "TPC-C - The Standard Benchmark for Online transaction Processing (OLTP)," in *The Benchmark Handbook for Database and Transaction Systems (2nd Edition).*, J. Gray, Ed. Morgan Kaufmann, 1993.

[20] T. Hogan, "Overview of TPC Benchmark E: The Next Generation of OLTP Benchmarks," in *Performance Evaluation and Benchmarking, First TPC Technology*

Conference, TPCTC 2009, Lyon, France, August 24-28, 2009, Revised Selected Papers, 2009, vol. 5895, pp. 84–98.

[21] M. Pöss and C. Floyd, "New TPC Benchmarks for Decision Support and Web Commerce," *SIGMOD Rec.*, vol. 29, no. 4, pp. 64–71, 2000.

[22] M. Poess, T. Rabl, and H.-A. Jacobsen, "Analysis of TPC-DS: the first standard benchmark for SQL-based big data systems," in *Proceedings of the 2017 Symposium on Cloud Computing, SoCC 2017, Santa Clara, CA, USA, September 24-27, 2017*, 2017, pp. 573–585.

[23] M. Pöss, R. O. Nambiar, and D. Walrath, "Why You Should Run TPC-DS: A Workload Analysis," in *Proceedings of the 33rd International Conference on Very Large Data Bases, University of Vienna, Austria, September 23-27, 2007*, 2007, pp. 1138–1149.

[24] R. O. Nambiar and M. Poess, "The Making of TPC-DS," in *Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006*, 2006, pp. 1049–1058.

[25] M. Poess, T. Rabl, H.-A. Jacobsen, and B. Caufield, "TPC-DI: The First Industry Benchmark for Data Integration," *PVLDB*, vol. 7, no. 13, pp. 1367–1378, 2014.

[26] P. Sethuraman and H. R. Taheri, "TPC-V: A Benchmark for Evaluating the Performance of Database Applications in Virtual Environments," in *Performance Evaluation, Measurement and Characterization of Complex Systems - Second TPC Technology Conference, TPCTC 2010, Singapore, September 13-17, 2010. Revised Selected Papers*, 2010, vol. 6417, pp. 121–135.

[27] R. Nambiar, "Benchmarking Big Data Systems: Introducing TPC Express Benchmark HS," in *Big Data Benchmarking - 5th International Workshop, WBDB 2014, Potsdam, Germany, August 5-6, 2014, Revised Selected Papers*, 2014, vol. 8991, pp. 24–28.

[28] A. Ghazal *et al.*, "BigBench: towards an industry standard benchmark for big data analytics," in *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, 2013, pp. 1197–1208.

[29] "Home | STAC - Insight for the Algorithmic Enterprise | STAC." [Online]. Available: https://www.stacresearch.com/. [Accessed: 04-Jun-2019].

[30] "LDBCouncil |." [Online]. Available: http://www.ldbcouncil.org/. [Accessed: 04-Jun-2019].

[31] Y. Liu, H. Zhang, L. Zeng, W. Wu, and C. Zhang, "MLbench: benchmarking machine learning services against human experts," *Proc. VLDB Endow.*, vol. 11, no. 10, pp. 1220–1232, 2018.

[32] S. Karim, T. R. Soomro, and S. M. A. Burney, "Spatiotemporal Aspects of Big Data," *Appl. Comput. Syst.*, vol. 23, no. 2, pp. 90–100, Dec. 2018.

[33] M. Stonebraker, J. Frew, K. Gardels, and J. Meredith, "The SEQUOIA 2000 Storage Benchmark," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 1993, pp. 2–11.

[34] J. Patel *et al.*, "Building a Scaleable Geo-spatial DBMS: Technology, Implementation, and Evaluation," in *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 1997, pp. 336–347.

[35] P. Werstein, "A performance benchmark for spatiotemporal databases," in *In: Proc. of the 10th Annual Colloquium of the Spatial Information Research Centre*, 1998, pp. 365–373.

[36] C. S. Jensen, D. Tiešytė, and N. Tradišauskas, "The COST Benchmark—Comparison and Evaluation of Spatio-temporal Indexes," in *Database Systems for Advanced Applications*, 2006, pp. 125–140.

[37] C. Düntgen, T. Behr, and R. H. Güting, "BerlinMOD: a benchmark for moving object databases," *VLDB J.*, vol. 18, no. 6, p. 1335, Apr. 2009.

[38] J. Yu, Z. Zhang, and M. Sarwat, "Spatial data management in apache spark: the GeoSpark perspective and beyond," *GeoInformatica*, vol. 23, no. 1, pp. 37–78, Jan. 2019.

[39] S. Hagedorn, P. Götze, and K.-U. Sattler, "Big Spatial Data Processing Frameworks: Feature and Performance Evaluation," in *EDBT*, 2017.

[40] M. Lissandrini, M. Brugnara, and Y. Velegrakis, "Beyond macrobenchmarks: microbenchmark-based graph database evaluation," *Proc. VLDB Endow.*, vol. 12, no. 4, pp. 390–403, Dec. 2018.

[41] D. Yan, Y. Bu, Y. Tian, A. Deshpande, and J. Cheng, "Big Graph Analytics Systems," in *Proceedings of the 2016 International Conference on Management of Data*, New York, NY, USA, 2016, pp. 2241–2243.

[42] G. Aluç, O. Hartig, M. T. Özsu, and K. Daudjee, "Diversified Stress Testing of RDF Data Management Systems," in *The Semantic Web – ISWC 2014*, 2014, pp. 197–212.

[43] G. Bagan, A. Bonifati, R. Ciucanu, G. H. Fletcher, A. Lemay, and N. Advokaat, "gMark: Schema-Driven Generation of Graphs and Queries," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 856–869, Apr. 2017.

[44] R. Angles *et al.*, "The Linked Data Benchmark Council: A Graph and RDF Industry Benchmarking Effort," *SIGMOD Rec*, vol. 43, no. 1, pp. 27–31, May 2014.

[45] A. Iosup *et al.*, "LDBC Graphalytics: A Benchmark for Large-scale Graph Analysis on Parallel and Distributed Platforms," *Proc VLDB Endow*, vol. 9, no. 13, pp. 1317–1328, Sep. 2016.

[46] M. Stonebraker, U. Cetintemel and S. Zdonik, "The 8 Requirements of Real-Time Steam Processing," ACM SIGMOD Record, vol. 34, pp. 42-47, 2005..

[47] Yahoo!, "Github," [Online]. Available: https://github.com/yahoo/streaming-benchmarks. [Accessed May 2019].

[48] M. Li, Y. Wang, Z. Li, v. Salapura and A. Biven, "SparkBench: A Comprehensive Spark Benchmarking Suite Characterizing In-memory Data Analytics".

[49] R. Lu, G. Wu, B. Xie and J. Hu, "Stream Bench: Towards Benchmarking Modern Distributed Stream Computing Frameworks," in IEEE/ACM 7th International Conference on Utility and Cloud Computing, London, UK, 2014.

[50] M. Arlitt, M. Marwah, G. Bellala, A. Shah, J. Healey og B. Vandiver, «IoTAbench: an Internet of Things Analytics Benchmark,» ICPE´15 Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering, pp. 133-144, Jan 28 - Feb 04 2015.

[51] A. Shukla, S. Chaturvedi and Y. Simmhan, "RIoTBench: A Real-time IoT Benchmark for Distributed Stream Processing Platforms," Concurrency and Computation; Software Practice and Experience, Volume 29, Issue 21, 10 November 2017

[52] N. Shalom, *The Common Priniciples Behind The NoSQL Alternatives,* 2009. Blog: https://natishalom.typepad.com/nati_shaloms_blog/2009/12/the-common-principles-behind-the-nosql-alternatives.html (Accessed May 2019).

[53] Benchmarking Deep Learning Operations on Different Hardware: Baidu-Research/DeepBench. baidu-research. URL: https://github.com/baidu-research/DeepBench.

[54] Tensorflow Benchmarks. URL: https://www.tensorflow.org/performance/benchmarks. (Accessed May 2019)

[55] The Deep Learning Benchmarks. Contribute to DeepMark/Deepmark Development by Creating an Account on GitHub. DeepMark. URL: https://github.com/DeepMark/deepmark.

[56] Soumith Chintala. Easy Benchmarking of All Publicly Accessible Implementations of Convnets: Soumith/Convnet-Benchmarks. URL: https://github.com/soumith/convnet-benchmarks.

[57] Robert Adolf et al. 'Fathom: Reference Workloads for Modern Deep Learning Methods'. In: 2016 IEEE International Symposium on Workload Characterization (IISWC) (Sept. 2016), pp. 1–10. DOI: 10.1109/IISWC.2016.7581275. arXiv: 1608.06581. URL: http://arxiv.org/abs/ 1608.06581.

[58] Cody Coleman et al. 'DAWNBench: An End-to-End Deep Learning Benchmark and Competition', https://cs.stanford.edu/~deepakn/assets/papers/dawnbench-sosp17.pdf

[59] Reference Implementations of Training Benchmarks. Contribute to Mlperf/Training Development by Creating an Account on GitHub. MLPerf. URL: https://github.com/mlperf/training.

[60] MLPerf. URL: https://mlperf.org/.

[61] Hongyu Zhu et al. 'TBD: Benchmarking and Analyzing Deep Neural Network Training'. In: (16th Mar. 2018). arXiv: 1803.06905 [cs, stat]. URL: http://arxiv.org/abs/1803.06905.

[62] Jinhua Tao et al. 'BENCHIP: Benchmarking Intelligence Processors'. In: (23rd Oct. 2017). arXiv: 1710.08315 [cs]. URL: http://arxiv.org/abs/1710.08315.

[63] Getting Started - Deep Learning Bechmarking Suite. URL: https://hewlettpackard.github.io/ dlcookbook-dlbs/#/index?id=deep-learning-benchmarking-suite.

[64] Wanling Gao et al. 'Data Motif-Based Proxy Benchmarks for Big Data and AI Workloads'. Published in IEEE International Symposium on Workload, 2018, DOI: 10.1109/IISWC.2018.8573475

[65] DataBench Deliverable "D1.3 Horizontal Benchmarks – Analytics and Processing" - available at https://www.databench.eu/public-deliverables/

[66] DataBench Deliverable "D1.4 Horizontal Benchmarks – Data Management" - available at https://www.databench.eu/public-deliverables/

7. Annex – Benchmark Descriptions

The benchmarks are listed chronologically according to the year of their first public introduction and described according to the following template:

Template for description:

1. Benchmark Name

2. Short Description
3. Web references
4. Date of last description update:
5. Originating group
6. Time – first version, last version
7. Type/Domain
8. Workload
9. Data type and generation/data sets
10. Technology stack and Implementation
11. Metrics
12. Reported results and usage
13. Reference papers

This template will also be used as the common reference structure for the online Knowledge Graph representation of the benchmarks.

The benchmarks are in the following listed chronologically, starting from 1999 until 2018. New benchmarks arriving during 2019 will be added and then the benchmark list Annex might be lifted out to a separate document (to be referred to from other documents, like D1.3 and D1.4) as well as being available online.

7.1 Year 1999

TPC-H

Benchmark description
Benchmark Name

TPC-H
Short Description
TPC-H is the de facto benchmark standard for testing data warehouse capability of a system. Instead of representing the activity of any particular business segment, TPC-H models any industry that manages, sells, or distributes products worldwide (e.g., car rental, food distribution, parts, suppliers, etc.). The benchmark is technology-agnostic.
Web references
http://www.tpc.org/tpch/
Date of last description update
12.06.2018
Originating group
Transaction Processing Performance Council (TPC)
Time – first version, last version
1999 – 2018
Type/Domain
Decision support benchmark
Workload
The core of the benchmark is comprised of a set of 22 business queries designed to exercise system functionalities in a manner representative of complex decision support applications.
Data type and generation/datasets
Structured data, generates data from a sample file.
Technology stack and implementation
C++ Dataset generator, SQL Engine
Metrics
The primary metrics calculated are: TPC-H Composite Query-per-Hour metric (QphH@Size), (b) Price-performance metric ($/QphH/@Size), and (c) Availability Date of the system.
Reported results and usage
http://www.tpc.org/tpch/results/tpch_results.asp ftp://ftp.hp.com/pub/c-products/servers/benchmarks/tpch_on_hp_proliant.pdf
Reference papers

Boncz, Peter, Thomas Neumann, and Orri Erling. "TPC-H analyzed: Hidden messages and lessons learned from an influential benchmark." Technology Conference on Performance Evaluation and Benchmarking. Springer, Cham, 2013.

7.2 Year 2002

TPC-DS v1

Benchmark description
Benchmark Name
TPC-DS v1
Short Description
TPC-DS is a decision support benchmark that models several generally applicable aspects of a decision support system, including queries and data maintenance. The main focus areas: Multiple snowflake

schemas with shared dimensions, 24 tables with an average of 18 columns, 99 distinct SQL 99 queries with random substitutions, More representative skewed database content,Sub-linear scaling of non-fact tables,Ad-hoc, reporting, iterative and extraction queries, ETL-like data maintenance.
Web references
http://www.tpc.org/tpcds/default.asp http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-ds_v2.5.0.pdf
Date of last description update
January 2019
Originating group
Transaction Processing Performance Council (TPC)
Time – first version, last version
2002 - 2019
Type/Domain
Decision support benchmark
Workload
TPC-DS defines 99 distinct SQL-99 (with OLAP amendment) queries and twelve data maintenance operations covering (a) typical DSS like query types such as ad-hoc, reporting, iterative (drill down/up), and (b) extraction queries and periodic refresh of the database.
Data type and generation/datasets
Synthetic data set.
Technology stack and implementation
SQL Databases
Metrics
Measures query response time in single user mode, query throughput in multi user mode and data maintenance performance for a given hardware.
Reported results and usage
http://www.tpc.org/tpcds/results/tpcds_advanced_sort.asp https://medium.com/hyrise/a-summary-of-tpc-ds-9fb5e7339a35
Reference papers
Nambiar, Raghunath Othayoth, and Meikel Poess. "The making of TPC-DS." Proceedings of the 32nd international conference on Very large data bases. VLDB Endowment, 2006.

7.3 Year 2004

Hadoop Workload Examples

Benchmark description
Benchmark Name
Hadoop Workload Examples
Short Description
Set of commonly used Hadoop applications like WordCount, Grep, Pi and Terasort.
Web references
https://wiki.apache.org/hadoop/Grep

http://hadoop.apache.org/docs/r3.2.0/api/org/apache/hadoop/examples/
Date of last description update
22.01.2019
Originating group
Apache
Time – first version, last version
2004 - 2019
Type/Domain
Microbenchmark
Workload
Different micro benchmarks like WordCount, Grep, Pi and Terasort.
Data type and generation/datasets
Synthetic data generation
Technology stack and implementation
Java / MapReduce
Metrics
Execution time
Reported results and usage
https://medium.com/ymedialabs-innovation/hadoop-performance-evaluation-by-benchmarking-and-stress-testing-with-terasort-and-testdfsio-444b22c77db2 http://udspace.udel.edu/bitstream/handle/19716/17628/2015_LiuLu_MS.pdf?sequence=1
Reference papers
Ivanov, Todor, et al. "Big data benchmark compendium." Technology Conference on Performance Evaluation and Benchmarking. Springer, Cham, 2015

Linear Road

Benchmark description
Benchmark Name

Linear Road
Short Description
The Linear Road Benchmark compares relational database systems with stream data management systems, computes performance characteristics of different stream data management systems relative to each other. is an application benchmark simulating a toll system for the motor vehicle expressways of a large metropolitan area. It specifies a fictional urban area including such features as accident detection and alerts, traffic congestion measurements, toll calculations and historical queries. The benchmark reports an L-rating metric, which is the number of expressways the system can process in real-time.
Web references
https://www.cs.brandeis.edu/~linearroad/
Date of last description update
2004
Originating group
Brandeis University, Brown University, Massachusetts Institute of Technology, Stanford University
Time – first version, last version
2004
Type/Domain
Streaming
Workload
The benchmark manages statistics about the number of vehicles and average speed on each segment of each expressway for every minute. It executes continuously queries while detecting accidents and notifying the other vehicles for these. At the same time the dynamic tolls are being calculating and assessing, which are dependent on segment statistics and proximate accidents and keep track of all assessed tolls. The toll must be calculated every time a vehicle sends a position report in a new segment or the driver needs a notification. The requirements for the response time is 5 seconds between the dispatch of the position report and the time the toll notification is sent. Furthermore, the system must process the historical queries. For account balance queries, it must return the sum of all tolls in a response time of 5 seconds and an accuracy of 60 seconds prior to the time the request is issued. For daily expenditure queries it must return the sum of tolls which are spent on an expressway at a given day of the last 10 weeks.
Data type and generation/datasets
The input data is stored in flat files and generated through a simulation traffic model by the traffic simulator MITSIM. It generates a set of vehicles and each completes a vehicle trip with focus on the downtown area. The input stream data are tuples which re split in position reports and historical query requests. These historical query requests can be for Account Balances, Daily Expenditures and Travel Time Estimations. The position reports are tuples which contains an integer timestamp, the vehicle entifier and some information about the vehicle trip. There is a probability of 1% for position reports that they contain additionally a historical query request, which is in 50% of the cases an account balance request, in 10% of the cases a daily tolls request and in 40% of the cases a travel time request. Systems must maintain all assessed tolls always to answer historical query requests. Furthermore, the historical data generator constructs two files which

contain the toll history for the previous 10 weeks. The first file saves tuples which contain information about the vehicle, day, expressway and tolls. The second file is a segment history file which contains information about each segment like number of vehicles, toll and average speed. 10 weeks of tolling history must be available.
Technology stack and implementation
To implement Linear Road, it is necessary to generate 10 weeks of historical data with the Historical Data generator; generate 286 Chapter A Classification of Big Data Benchmarks L flat files, each containing 3 hours traffic data and historical query requests from a single expressway with the traffic simulator. Then the system must generate the output files which contain the response to the queries. Then the validation tool is used to check the response times and accuracy of generated output.
Metrics
The performance is measured through the L-rating, whereby L means L expressways worth of input. It measures the supported query load, represented by the historical and continuous queries, which the stream processing system can process while the constraints of response time and accuracy are still fulfilled. To determine the performance, the benchmark will be run with increasing scale factors, until there is one for which the requirements can no longer be met.
Reported results and usage
http://www.it.uu.se/research/group/udbl/lr.html
Reference papers
Arvind Arasu, Mitch Cherniack, Eduardo F. Galvez, et al. "Linear Road: A Stream Data Management Benchmark"

7.4 Year 2007

GridMix

Benchmark description
Benchmark Name
GridMix
Short Description
The benchmark suite emulates different users sharing the same cluster resources and submitting different types and number of jobs. This includes also the emulation of distributed cache loads, compression, decompression and job configuration in terms of resource usage.
Web references

https://hadoop.apache.org/docs/stable1/gridmix.html
Date of last description update
07.09.2018
Originating group
Apache
Time – first version, last version
2007 - 2018
Type/Domain
Benchmark Suite
Workload
Mix of traced synthetic jobs and basic operations to stress job scheduler and compression and decompression.
Data type and generation/datasets
Synthetic data generation
Technology stack and implementation
Trace of recorded MapReduce jobs.
Metrics
Execution time, Memory, Throughput
Reported results and usage
https://hadoop.apache.org/docs/r1.2.1/gridmix.html
Reference papers
Ivanov, Todor, et al. "Big data benchmark compendium." Technology Conference on Performance Evaluation and Benchmarking. Springer, Cham, 2015

7.5 Year 2008

PigMix

Benchmark description
Benchmark Name
PigMix
Short Description
A set of 17 queries, written in Pig Latin, specifically created to test the latency and scalability performance of Pig systems with different operations like data loading, different types of joins, group by clauses, sort clauses, as well as aggregation operations.
Web references
https://cwiki.apache.org/confluence/display/PIG/PigMix
Date of last description update
2013
Originating group

Apache
Time – first version, last version
2007 - 2013
Type/Domain
Microbenchmark
Workload
Different queries testing like data loading, different types of joins, group by clauses, sort clauses, as well as aggregation operations.
Data type and generation/datasets
Synthetic and structured data.
Technology stack and implementation
Pig Latin, Hadoop
Metrics
Execution time.
Reported results and usage
https://cwiki.apache.org/confluence/display/pig/PigMix
Reference papers
--

MRBench

Benchmark description
Benchmark Name
MRBench
Short Description
Implementing the TPC-H benchmark queries directly in map and reduce operations.
Web references

https://markobigdata.com/2016/07/13/hadoop-benchmark-test-mrbench/
Date of last description update
2008
Originating group
--
Time – first version, last version
2008
Type/Domain
Decision support benchmark
Workload
The core of the benchmark is comprised of a set of 22 business queries designed to exercise system functionalities in a manner representative of complex decision support applications.
Data type and generation/datasets
Structured data, generates data from a sample file.
Technology stack and implementation
C++ , Hadoop MapReduce
Metrics
Execution time.
Reported results and usage
--
Reference papers
Kim, Kiyoung, et al. "Mrbench: A benchmark for mapreduce framework." 2008 14th IEEE International Conference on Parallel and Distributed Systems. IEEE, 2008.

7.6 Year 2009

CALDA

Benchmark description
Benchmark Name
CALDA
Short Description
The benchmark consists of five tasks defined as SQL queries among which is the original MR Grep task, which is a representative for most real user MapReduce programs. The benchmark was developed to specifically compare the capabilities of Hadoop with those of commercial parallel Relational Database Management Systems (RDBMS).
Web references
http://www.cs.umd.edu/~abadi/papers/benchmarks-sigmod09.pdf
Date of last description update
2018

Originating group
Brown University, University of Wisconsin, Yale University, Microsoft Inc., M.I.T. CSAIL
Time – first version, last version
2008 - 2008
Type/Domain
Microbenchmark
Workload
5 SQL queries among MapReduce grep task.
Data type and generation/datasets
Synthetic structured data.
Technology stack and implementation
Hadoop, MapReduce.
Metrics
Execution time.
Reported results and usage
--
Reference papers
Andrew Pavlo, Erik Paulson, Alexander Rasin, et al. "A Comparison of Approaches to Large-Scale Data Analysis". In: SIGMOD. 2009, pp. 165–178

HiBench

Benchmark description
Benchmark Name
HiBench
Short Description

A comprehensive benchmark suite consisting of multiple workloads including both synthetic micro-benchmarks and real-world applications. HiBench features several ready-to-use benchmarks from 4 categories: micro benchmarks, Web search, Machine Learning, and HDFS benchmarks.
Web references
https://github.com/Intel-bigdata/HiBench http://www.odbms.org/wp-content/uploads/2014/07/hibench-wbdb2012-updated.pdf
Date of last description update
31.01.2018
Originating group
Intel
Time – first version, last version
2009 – 2019
Type/Domain
Benchmark Suite
Workload
Micro-benchmark suite including 6 categories which are micro, ML (machine learning), SQL, graph, websearch and streaming.
Data type and generation/datasets
Most workloads use synthetic data generated from real data samples. The workloads use structured and semi-structured data, including graph, network, text and web data types.
Technology stack and implementation
HiBench can be executed in Docker containers. It is implemented using the following technologies: (1) Hadoop: Apache Hadoop 2.x, CDH5, HDP; (2) Spark: Spark 1.6.x, Spark 2.0.x, Spark 2.1.x, Spark 2.2.x; (3) Flink: 1.0.3; (4) Storm: 1.0.1; (5) Gearpump: 0.8.1; and (6) Kafka: 0.8.2.2.
Metrics
The measured metrics are execution time (latency), throughput and system resource utilizations (CPU, Memory, etc.).
Reported results
--
Reference papers

Huang, Shengsheng, et al. "The HiBench benchmark suite: Characterization of the MapReduce-based data analysis."

7.7 Year 2010

Liquid

Benchmark description
Benchmark Name
Liquid benchmarks
Short Description
Liquid benchmarking platform is an online cloud-based platform for democratizing the performance evaluation and benchmarking process. The primary objective of the Liquid Benchmarking platform is to provide a cloud-based and social platform which can simplify and democratize the job of computer science scientific scholars in conducting solid experimental evaluations with high quality. The service allows building repositories of competing research implementations, sharing testing computing platforms, collaboratively building the specifications of standard benchmarks and allowing end-users to create and run testing experiments and share their results.

Web references
https://pdfs.semanticscholar.org/bf88/c9e10c0cf40698eeaa778d753f42b250c06b.pdf https://thesai.org/Downloads/Volume7No2/Paper_68-A_Cloud_Based_Platform_for_Democratizing.pdf
Date of last description update
--
Originating group
NICTA and University of South Wales (Sydney, Australia), Univerity of Trento (Trento, Italy)
Time – first version, last version
2010 – xx
Type/Domain
Benchmarking platform
Workload
Four benchmarks have been realized using the Liquid Platform: (1) *XML compression*, (2) Graph indexing and querying, (3) String Similarity Join, and (4) Reverse K Nearest Neighbors (RkNN). The tasks include: (a) creation of centralized repositories for software implementation and results, (b) establish shared resources environment, (c) generate a standard workable environment for experiments, (d) maintenance of experimental studies, (e) maintenance of feedbacks over results, (f) create platform for scientific crediting process, (e) provide provenance services for scientific experimental results and time analysis services for research methods.
Data type and generation/datasets
Structured BI, Text/Web and Graph/Network. Datasets used by the platform are dependent on the benchmark case study that is being used. Data can be of different types and different formats (e.g. image files, database records, XML files) depending on the context of the benchmark.
Technology stack and implementation
Current implementation of included benchmarks are using AWS for shared computing resources and open source social network platform, elgg. Reference implementation includes evaluation of XML compressors (like Gzip, Bzip, XMill), graph indexing and querying techniques (like ClosureTree, gIndex, TreePi, etc.) using iGraph framework and algorithms for string similarity joins and RkNN.
Metrics

XML compression benchmark metrics: compression ratio, compression time and de-compression time. Graph indexing and quering benchmark metrics: indexing time, index size and query processing time. String Similarity Join metrics: running time, size of candidate results.
Reported results
--
Reference papers
Sakr, S., Casati, F.: Liquid benchmarks: Towards an online platform for collaborative assessment of computer science research results. In: Proceedings of the Second TPC Technology Conference on Performance Evaluation, Measurement and Characterization of Complex Systems. pp. 10–24. TPCTC'10, Springer-Verlag, Berlin,Heidelberg (2011) Sakr, S., Shafaat, A., Bajaber, F., Barnawi, A., Batarfi, O., Altalhi, A.H.: Liquidbenchmarking: A platform for democratizing the performance evaluation process. In: Proceedings of the 18th International Conference on Extending Database Tech-nology, EDBT 2015, Brussels, Belgium, March 23-27, 2015. pp. 537–540 (2015)

YCSB

Benchmark description
Benchmark Name
YCSB
Short Description
A benchmark designed to compare emerging cloud serving systems like Cassandra, HBase, MongoDB, Riak and many more, which do not support ACID. It provides a core package of 6 pre-defined workloads A-F, which simulate a cloud OLTP application.
Web references
https://github.com/brianfrankcooper/YCSB
Date of last description update
August 2018
Originating group
Yahoo!
Time – first version, last version
2010-2018
Type/Domain

Collection of cloud OLTP related workloads representing a particular mix of read/write operations, data sizes, request distributions, and similar that can be used to evaluate systems at one particular point in the performance space.
Workload
YCSB provides a core package of 6 pre-defined workloads A-F, which simulate a cloud OLTP applications. The workloads are a variation of the same basic application type and using a table of records with predefined size and type of the fields.
Data type and generation/datasets
The benchmark consists of a workload generator and a generic database interface, which can be easily extended to support other relational or NoSQL databases.
Technology stack and implementation
Currently, YCSB is implemented and can be run with more than 14 different engines like Cassandra, HBase, MongoDB, Riak, Couchbase, Redis, Memcached, etc. The YCSB Client is a Java program for generating the data to be loaded to the database, and generating the operations which make up the workload.
Metrics
The benchmark measures the latency and achieved throughput of the executed operations. At the end of the experiment, it reports total execution time, the average throughput, 95th and 99th percentile latencies, and either a histogram or time series of the latencies.
Reported results
https://scalegrid.io/blog/how-to-benchmark-mongodb-with-ycsb/
Reference papers
Cooper, Brian F., et al. "Benchmarking cloud serving systems with YCSB."

7.8 Year 2011

SWIM

Benchmark description
Benchmark Name
SWIM
Short Description
It consists of a framework which is able to synthesize representative workload from real MapReduce traces taking into account the job submit time, input data size, shuffle/input and output/shuffle data ratio. The result is a synthetic workload which has the exact characteristics of the original workload.
Web references
http://barbie.uta.edu/~jli/Resources/MapReduce&Hadoop/the%20case%20for%20evaluating%20mapreduce.pdf https://github.com/SWIMProjectUCB/SWIM/wiki
Date of last description update

2016
Originating group
Cloudera, Brown University, UC Berkeley AMP Lab
Time – first version, last version
--
Type/Domain
Collection of MapReduce Jobs.
Workload
Synthetic MapReduce workloads.
Data type and generation/datasets
Synthetic workload generation.
Technology stack and implementation
Hadoop MapReduce
Metrics
Job completion time, Ratio of failed jobs.
Reported results and usage
https://github.com/SWIMProjectUCB/SWIM/wiki/Workloads-repository
Reference papers
Chen, Yanpei, et al. "The case for evaluating MapReduce performance using workload suites."

CloudRank-D

Benchmark description
Benchmark Name
CloudRank-D
Short Description
CloudRank-D is a benchmark suite for evaluating the performance of cloud computing systems running Big Data applications. The suite consists of 13 representative data analysis tools, which are designed to address

a diverse set of workload data and computation characteristics (i.e. data semantics, data models and data sizes, the ratio of the size of data input to that of data output)
Web references
http://prof.ict.ac.cn/ http://prof.ict.ac.cn/jfzhan/papers/Luo_FCS_12.pdf
Date of last description update
2013
Originating group
ICT
Time – first version, last version
2011 – 2013
Type/Domain
Benchmark Suite
Workload
The suite consists of 13 representative data analysis tools, which are designed to address a diverse set of workload data and computation characteristics. Workload set is a mix of basic operations (Sort, WordCount, Grep), Classification (Naïve bayes, Support vector machine), Clustering (K-means), Recommendation (item-based collaborative filtering), Association rule mining (frequent pattern growth), Sequence learning (Hidden Markov), and Data warehouse operations (Grep select, Ranking select, User-visits aggregation, User-visits ranking join).
Data type and generation/datasets
Depending on the workload synthetic or real data sets
Technology stack and implementation
Hadoop, Hive, Mahout
Metrics
Data processed per Second / Joule
Reported results
http://prof.ict.ac.cn/DComputing/uploads/2013/DC_5_2_CloudRank-D.pdf
Reference papers
Chunjie Luo, Jianfeng Zhan, Zhen Jia, et al. "CloudRank-D: benchmarking and ranking cloud computing systems for data processing applications".

7.9 Year 2012

PUMA Benchmark Suite

Benchmark description
Benchmark Name
PUMA Benchmark Suite
Short Description
A set of 13 common Hadoop micro-benchmarks, very similar to the Hadoop Workload Examples.
Web references
https://engineering.purdue.edu/~puma/datasets.htm https://engineering.purdue.edu/~puma/pumabenchmarks.htm
Date of last description update

2012
Originating group
Perdue University
Time – first version, last version
2012
Type/Domain
Microbenchmark
Workload
MapReduce workloads
Data type and generation/datasets
Benchmark supports structured data and unstructured data. Different datasets, of varying scales, are provided for different workloads. For example, K-means and classification use movies data while Term-vector and word-count use data from Wikipedia.
Technology stack and implementation
Hadoop MapReduce
Metrics
Execution time, MapReduce statistics
Reported results
--
Reference papers
Ahmad, Faraz, et al. "Puma: Purdue MapReduce benchmarks suite." (2012).

CloudSuite

Benchmark description
Benchmark Name
CloudSuite
Short Description

CloudSuite is a benchmark suite consisting of both emerging scale-out workloads and traditional benchmarks. The goal of the benchmark suite is to analyze and identify key inefficiencies in the processor's core micro-architecture and memory system organization when running today's cloud workloads.
Web references
https://cloudsuite.ch/
Date of last description update
2019
Originating group
CALCM, EcoCloud
Time – first version, last version
2012 – 2019
Type/Domain
Benchmark Suite
Workload
Scale-out workloads and traditional benchmarks. Workload includes operations related to data serving (operations over Cassandra 0.7.3 with YCSB 0.1.3), MapReduce (Bayesian classification from Mahout 0.4 lib), Media Streaming (Darwin Streaming Server 6.0.3 with Faban Driver), SAT Solver (Klee SAT solver), Web frontend (Olio, Nginx, CloudStone), Web search Nutch 1.2/Lucene 3.0.1), Web backend (MySQL 5.5.9), and traditional benchmarks (PARSEC 2.1, SPEC CINT2006, SPECweb09, TPC-C, TPC-E).
Data type and generation/datasets
Real-world data samples.
Technology stack and implementation
Docker container
Metrics
The micro-architectural behavior of scale-out workloads is examined through the commit-time execution breakdown. Each cycle of execution is classified as Committing if at least one instruction was committed during that cycle or as Stalled otherwise. Overlapped with the execution-time breakdown, it shows the Memory cycles bar, which approximates the number of cycles when the processor could not commit instructions due to outstanding long-latency memory accesses.
Reported results
--
Reference papers

Michael Ferdman, Almutaz Adileh, Yusuf Onur Koçberber, et al. “Clearing the clouds: a study of emerging scale-out workloads on modern hardware”.

MapReduce Benchmark Suite (MRBS)

Benchmark description
Benchmark Name
MapReduce Benchmark Suite (MRBS)
Short Description
A comprehensive benchmark suite for evaluating the performance of MapReduce systems in 5 areas: recommendations, BI (TPC-H), Bioinformatics, Text Processing & Data Mining.
Web references
http://sardes.inrialpes.fr/research/mrbs/index.html
Date of last description update
2012
Originating group
--
Time – first version, last version
2012
Type/Domain
Benchmark Suite
Workload
Two execution modes are supported: interactive mode and batch mode. The benchmark run consists of three phases dynamically configurable by the end-user: warm-up phase, run-time phase, and slow-down phase. The user can specify the number of runs and the different aspects of load: dataload and workload. The dataload is characterized by the size and the nature of the data sets used as inputs for a benchmark, and the workload is characterized by the number of concurrent clients and the distribution of the request type. Workload categories span over different domains, including Recommendation (benchmark based on real movie database), Business Intelligence (TPC-H), Bio-Informatics (DNA sequencing), Text processing (search patterns, word occurrences, sorting on randomy generated text files), and Data mining (classifying newsgroup documents into categories, canopy clustering operations).
Data type and generation/datasets

Depending on the executed benchmark synthetic or real data is used.
Technology stack and implementation
Hadoop MapReduce
Metrics
The high-level metrics reported by the benchmark are client request latency, throughput and cost. Additionally, low-level metrics like size of read/written data, throughput of MR jobs, and tasks are also reported.
Reported results
--
Reference papers
Sangroya, Amit, Damián Serrano, and Sara Bouchenak. "MRBS: Towards dependability benchmarking for Hadoop MapReduce."

7.10 Year 2013

AMP Lab Big Data Benchmark

Benchmark description
Benchmark Name
AMP Lab Big Data Benchmark
Short Description
Benchmark based on CALDA and HiBench, implemented on 5 SQL-on-Hadoop engines (RedShift, Hive, Stinger/Tez, Shark and Impala).
Web references
https://amplab.cs.berkeley.edu/benchmark/
Date of last description update

2014
Originating group
Berkeley
Time – first version, last version
2013 – 2014
Type/Domain
Data warehousing
Workload
It consists of four queries involving scan, aggregation, join, and bulk UDF query. It supports different data sizes and scaling to thousands of nodes.
Data type and generation/datasets
Synthetic structured data, unstructured data
Technology stack and implementation
RedShift, Hive, Stinger/Tez, Shark, and Impala, HDFS.
Metrics
Execution time.
Reported results
https://amplab.cs.berkeley.edu/benchmark/
Reference papers
Ivanov, Todor, et al. "Big data benchmark compendium." Technology Conference on Performance Evaluation and Benchmarking

BigBench

Benchmark description
Benchmark Name
BigBench
Short Description

It is an end-to-end Big Data benchmark that represents a data model simulating the volume, velocity and variety characteristics of a Big Data system, together with a synthetic data generator for structured, semi-structured and unstructured data, consisting of 30 queries.
Web references
http://www.tpc.org/tpcx-bb/
Date of last description update
2019
Originating group
TPC
Time – first version, last version
2013 – 2019
Type/Domain
BigBench is an end-to-end, technology agnostic, application-level benchmark that tests the analytical capabilities of a Big Data platform. It is based on a fictional product retailer business model.
Workload
The business model and a large portion of the data model's structured part is derived from the TPC-DS benchmark. The structured part was extended with a table for the prices of the retailer's competitors, the semi-structured part was added represented by a table with website logs and the unstructured part was added by a table showing product reviews. The simulated workload is based on a set of 30 queries covering the different aspects of Big Data analytics proposed by McKinsey.
Data type and generation/datasets
Synthetic / un-, semi-, and structured data.
Technology stack and implementation
Hadoop, using the MapReduce engine and other components like Hive, Mahout, Spark SQL, Spark MLlib and OpenNLP from the Hadoop Ecosystem.
Metrics
TPCx-BB defines the following primary metrics: (1) BBQpm@SF, the performance metric, reflecting the TPCx-BB Queries per minute throughput; where SF is the Scale Factor; (2) $/BBQpm@SF, the price/performance metric; and (3) System Availability Date as defined by the TPC Pricing Specification.
Reported results
http://www.tpc.org/tpcx-bb/results/tpcxbb_perf_results.asp http://msrg.utoronto.ca/publications/pdf_files/2013/Ghazal13-BigBench:_Towards_an_Industry_Standa.pdf

Reference papers
Ahmad Ghazal, Tilmann Rabl, Minqing Hu, et al. "BigBench: towards an industry standard benchmark for Big Data analytics".

BigDataBench

Benchmark description
Benchmark Name
BigDataBench
Short Description
It is an open source Big Data benchmark suite consisting of 15 data sets (of different types) and more than 33 workloads.
Web references
http://prof.ict.ac.cn/ http://www.benchcouncil.org/BigDataBench/
Date of last description update
2019
Originating group
Chinese Academy of Sciences & BenchCouncil / ICT
Time – first version, last version
2013 – 2019
Type/Domain
Benchmark Suite
Workload
Seven workload types including AI, online services, offline analytics, graph analytics, data warehouse, NoSQL, and streaming.
Data type and generation/datasets
Real world data.
Technology stack and implementation

For offline analytics, we provide Hadoop, Spark, Flink and MPI implementations. For graph analytics, we provide Hadoop, Spark GraphX, Flink Gelly and GraphLab implementations. For AI, we provide TensorFlow and Caffe implementations. For data warehouse, we provide Hive, Spark-SQL and Impala implementations. For NoSQL, we provide MongoDB and HBase implementations. For streaming, we provide Spark streaming and JStorm implementations.
Metrics
Wall clock time and energy efficiency.
Reported results
http://prof.ict.ac.cn/wp-content/uploads/2013/10/Wang_BigDataBench.pdf
Reference papers
"Lei Wang, Jianfeng Zhan, Chunjie Luo, et al. „BigDataBench: A Big Data Benchmark Suite from Internet Services".

LinkBench

Benchmark description
Benchmark Name
LinkBench
Short Description
LinkBench is a benchmark, developed by Facebook, using synthetic social graph to emulate social graph workload on top of databases such as MySQL.
Web references
https://github.com/facebookarchive/linkbench
Date of last description update
2015
Originating group
Facebook
Time – first version, last version
2013 – 2015
Type/Domain

Graph benchmark
Workload
Set of standard insert, update, and delete operations to modify data, along with variations on key lookup, range, and count queries.
Data type and generation/datasets
Synthetic social graph with key properties similar to the real graph.
Technology stack and implementation
MySQL, MongoDB
Metrics
Latency of requests.
Reported results
http://people.cs.uchicago.edu/~tga/pubs/sigmod-linkbench-2013.pdf
Reference papers
Timothy G. Armstrong, Vamsi Ponnekanti, Dhruba Borthakur, and Mark Callaghan. "LinkBench: A Database Benchmark Based on The Facebook Social Graph".

BigFrame

Benchmark description
Benchmark Name
BigFrame
Short Description
BigFrame is a benchmark generator offering a benchmarking-as-a-service solution for Big Data analytics.
Web references
https://github.com/bigframeteam/BigFrame/wiki
Date of last description update
2013
Originating group
--

Time – first version, last version
2013
Type/Domain
Benchmark generator
Workload
The benchmark distinguishes between two different analytics workload, 1) offline-analytics and 2) real-time analytics.
Data type and generation/datasets
Structured / semi-structured synthetic data adapted from TPC-DS.
Technology stack and implementation
Java and Hadoop
Metrics
Execution time.
Reported results
--
Reference papers
Mayuresh Kunjir, Prajakta Kalmegh, and Shivnath Babu. „Thoth: Towards Managing a Multi-System Cluster".

PRIMEBALL

Benchmark description
Benchmark Name
PRIMEBALL
Short Description
PRIMEBALL is a novel and unified benchmark specification for comparing the parallel processing frameworks in the context of Big Data applications hosted in the cloud. It is implementation- and technology-agnostic, using a fictional news hub called New Pork Times, based on a popular real-life news site.
Web references

https://hal.archives-ouvertes.fr/hal-00921822/document
Date of last description update
2013
Originating group
--
Time – first version, last version
2013
Type/Domain
Parallel processing frameworks in the context of Big Data applications hosted in the cloud.
Workload
Various use-case scenarios made of both queries and data-intensive batch processing.
Data type and generation/datasets
Structured XML and binary audio and video files.
Technology stack and implementation
Implementation- and technology-agnostic.
Metrics
Throughput and price performance.
Reported results
https://hal.archives-ouvertes.fr/hal-00921822/document
Reference papers
Jaume Ferrarons, Mulu Adhana, Carlos Colmenares, et al. “PRIMEBALL: A Parallel Processing Framework Benchmark for Big Data Applications in the Cloud”

OpenML Benchmark Suites

Benchmark description
Benchmark Name
OpenML benchmark suites

Short Description
The suite offers (a) ease of use through standardized data formats, APIs, and existing client libraries; (b) machine-readable meta-information regarding the contents of the suite; and (c) online sharing of results, enabling large scale comparisons. The OpenML100 is a machine learning benchmark suite of 100 classification datasets carefully curated from the thousands of datasets available on OpenML.org.
Web references
https://docs.openml.org/benchmark/ https://github.com/openml
Date of last description update
2019
Originating group
--
Time – first version, last version
--
Type/Domain
Machine Learning
Workload
Benchmark suite with different workloads.
Data type and generation/datasets
Different types of datasets depending of the used benchmark,
Technology stack and implementation
REST, Python, R, Java, .NET
Metrics
Depending on the executed Benchmark.
Reported results and usage
https://github.com/openml
Reference papers
Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. SIGKDD Explorations 15(2), pp 49-60, 2013.

7.11 Year 2014

Semantic Publishing Benchmark (SPB)

Benchmark description
Benchmark Name
Semantic Publishing Benchmark (SPB)
Short Description
It is a LDBC benchmark for RDF database engines inspired by the Media/Publishing industry, particularly by the BBC's Dynamic Semantic Publishing approach.
Web references
http://ldbcouncil.org/developer/spb https://github.com/ldbc/ldbc_spb_bm_2.0

Date of last description update
2018
Originating group
LDBC
Time – first version, last version
2014 – 2019
Type/Domain
Graph benchmark
Workload
Basic: Consisting of an interactive query-mix for evaluation RDF systems in most common use-cases Advanced: Consisting of interactive and analytical query-mixes, adding additional complexity to the query workload e.g. faceted, analytical and drill-down queries
Data type and generation/datasets
Synthetic RDF data
Technology stack and implementation
Graph DB, Apache Ant
Metrics
Execution time
Reported results and usage
http://ceur-ws.org/Vol-1700/paper-01.pdf https://evert.meulie.net/various/spb-benchmark-results/ http://ldbcouncil.org/benchmarks/spb https://github.com/ldbc/ldbc_spb_bm_2.0
Reference papers
Angles, Renzo, et al. "The linked data benchmark council: a graph and RDF industry benchmarking effort." ACM SIGMOD Record 43.1 (2014): 27-31.

Social Network Benchmark

Benchmark description

Benchmark Name
Social Network Benchmark
Short Description
It consists of a data generator that generates a synthetic social network, used in three workloads: Interactive, Business Intelligence and Graph Analytics.
Web references
http://ldbcouncil.org/benchmarks/snb
Date of last description update
2018
Originating group
Linked Data Benchmark Council (LDBC)
Time – first version, last version
2014 – 2018
Type/Domain
Graph benchmark
Workload
Interactive, Business Intelligence and Graph Analytics.
Data type and generation/datasets
Synthetic social network.
Technology stack and implementation
GraphDB
Metrics
Operations/minute
Reported results and usage
http://ldbcouncil.org/benchmarks/snb
Reference papers

Angles, Renzo, et al. "The linked data benchmark council: a graph and RDF industry benchmarking effort." ACM SIGMOD Record 43.1 (2014): 27-31.

ALOJA

Benchmark description
Benchmark Name
ALOJA
Short Description
The ALOJA research project is an initiative from the Barcelona Supercomputing Center (BSC) to produce a systematic study of Hadoop configuration and deployment options. The project provides an open source platform for executing Big Data frameworks in an integrated manner facilitating benchmark execution and evaluation of results. ALOJA currently provides tools to deploy, provision, configure, and benchmark Hadoop, as well as providing different evaluations for the analysis of results covering both software and hardware configurations of executions. ALOJA-ML is an extension to the platform for predictive analytics, providing an automated system which allows knowledge discovery.
Web references
http://minerva.bsc.es:8099 https://aloja.bsc.es/ https://github.com/Aloja/aloja-mlb
Date of last description update
2019
Originating group
BSC-Microsoft Research Centre
Time – first version, last version
2014 – 2019
Type/Domain
Benchmark platform
Workload
Different workloads depending on the system to be tested (Big Data applications/algorithms, frameworks, systems/clusters and data centers). MPI-based profiling workloads, cluster configuration workloads, Machine Learning and predictive analytic workloads.
Data type and generation/datasets

Different types of datasets based on workload type. Datasets for machine learning are time-series traces of system executions with different properties/attributes.
Technology stack and implementation
Vagrant, VirtualBox, Bash-Scripts, Hadoop Ecosystem.
Metrics
Depending on the system under test. Main metrics include execution time, cost/performance efficiency, Job-execution time for not-benchmarked configurations,
Reported results and usage
https://www.berralgarcia.com/documents/poggi-BigData15.pdf http://minerva.bsc.es:8099/metrics
Reference papers
Nicolás Poggi, David Carrera, Aaron Call, et al. "ALOJA: A systematic study of Hadoop deployment variables to enable automated characterization of cost effectiveness". Josep LI. Berral, Nicolás Poggi, David Carrera, Aaron Call "ALOJA: A framework for benchmarking and predictive analytics in Big Data deployments".

WatDiv

Benchmark description
Benchmark Name
WatDiv
Short Description
WatDiv measures how an RDF data management system performs across a wide spectrum of SPARQL queries with varying structural characteristics and selectivity classes. It consists of two components: the data generator and the query (and template) generator.
Web references
http://dsg.uwaterloo.ca/watdiv/
Date of last description update
2014
Originating group
University of Waterloo

Time – first version, last version
2014
Type/Domain
Graph benchmark
Workload
These tests consist of queries in four categories, namely, linear queries (L), star queries (S), snowflake-shaped queries (F) and complex queries (C) with a total of 20 query templates.
Data type and generation/datasets
Synthetic data generation. Default schema is created around products and users scenario.
Technology stack and implementation
C++ and GraphDB.
Metrics
Execution time.
Reported results and usage
--
Reference papers
Aluç, Güneş, et al. "Diversified stress testing of RDF data management systems." International Semantic Web Conference. Springer, Cham, 2014. G. Aluç, O. Hartig, M. T. Özsu and K. Daudjee. Diversified Stress Testing of RDF Data Management Systems. In Proc. The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, 2014, pages 197-212. WatDiv available from http://dsg.uwaterloo.ca/watdiv/.

StreamBench

Benchmark description
Benchmark Name
StreamBench
Short Description
It covers 7 micro-benchmark programs that intend to address typical stream computing scenarios, implemented in Spark Streaming and Storm.
Web references

--
Date of last description update
2014
Originating group
--
Time – first version, last version
2014
Type/Domain
Streaming
Workload
The benchmark consists of four different workload suites. The Performance workload suite uses all seven programs and reads and processes the data from the messaging system as fast as it can. The Multi-recipient performance workload suite also uses the seven benchmarks on only one dataset. It defines three different cluster configurations called reception ability, to be the proportion of nodes that receive input data out of the whole cluster. The Fault tolerance workload suite also includes the seven micro-benchmarks and considers failure of only one cluster node intentionally failing in the middle of the execution. The Durability workload suite contains only the Wordcount program and two data scale sizes (factors).
Data type and generation/datasets
The benchmark suite uses different data scale sizes generated from two datasets. The AOL Search Data set is a collection of real query log data from real users, whereas the CAIDA Anonymized Internet Traces Dataset consists of statistical information of an hour-long internet package traces. The datasets cover both text and numerical data, but have different number of attributes and number of records.
Technology stack and implementation
The benchmark suite is implemented and evaluated with the Apache Storm and Apache Spark Streaming frameworks. Apache Kafka is used as a messaging system.
Metrics
There are different metrics for the different workload suites. The main metrics are: (1) throughput (in bytes processed per second) (2) latency (the average time span from the arrival of a record until the record is processed). (3) The throughput penalty factor (TPF) and latency penalty factor (LPF) are both defined and reported in the fault-tolerance workload suite.
Reported results and usage
--

Reference papers
Lu, Ruirui, et al. "Stream bench: Towards benchmarking modern distributed stream computing frameworks." 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. IEEE, 2014.

TPCx-HS

Benchmark description
Benchmark Name
TPCx-HS
Short Description
It stresses both the hardware and software components including the Hadoop run-time stack, Hadoop File System and MapReduce layers. The benchmark is based on the TeraSort workload, which is part of the Apache Hadoop distribution.
Web references
http://www.tpc.org/tpcx-hs/ http://www.tpc.org/tpc_documents_current_versions/pdf/tpcx-hs_v2.0.3.pdf
Date of last description update
2018
Originating group
Transaction Processing Performance Council (TPC)
Time – first version, last version
2014 – 2018
Type/Domain
The benchmark is based on the TeraSort workload, which is part of the Apache Hadoop distribution.
Workload
It consists of four modules: HSGen, HSDataCkeck, HSSort, and HSValidate. The HSGen is a program that generates the data for a particular Scale Factor (see Clause 4.1 from the TPCx-HS specification) and is based on the TeraGen, which uses a random data generator. The HSDataCheck is a program that checks the compliance of the dataset and replication. The HSSort is a program, based on TeraSort, which sorts the data into a total order. Finally, HSValidate is a program, based on TeraValidate, that validates the output is sorted.
Data type and generation/datasets

The scale factor defines the size of the dataset, which is generated by HSGen and used for the benchmark experiments. In TPCx-HS, it follows a stepped size model. The data is synthetically generated.
Technology stack and implementation
Hadoop MapReduce
Metrics
The benchmark reports the total elapsed time (T) in seconds for both runs. This time is used for the calculation of the TPCx-HS performance metric also abbreviated with HSph@SF. The run that takes more time and results in lower TPCx-HS performance metric is defined as the performance run. On the contrary, the run that takes less time and results in TPCx-HS performance metric is defined as the repeatability run. The benchmark reported performance metric is the TPCx-HS performance metric for the performance run.
Reported results and usage
http://www.tpc.org/tpcx-hs/results/tpcxhs_perf_results.asp
Reference papers
Raghunath Othayoth Nambiar, Meikel Poess, Akon Dey, et al. "Introducing TPCx-HS: The First Industry Standard for Benchmarking Big Data Systems".

gMark

Benchmark description
Benchmark Name
gMark
Short Description
gMark is a domain- and query language-independent framework targeting highly tunable generation of both graph instances and graph query workloads based on user-defined schemas. It provides a query translator for SPARQL, openCypher, PostgreSQL and Datalog.
Web references
https://github.com/graphMark/gmark http://www.info.univ-tours.fr/ICVL/doc/jirc-2017/slides-radu-jirc.pdf
Date of last description update
2019

Originating group
--
Time – first version, last version
--
Type/Domain
Graph benchmark
Workload
gMark generates Unions of Conjunctions of Regular Path Queries (UCRPQ). UCRPQ contains recursive path queries for applications like social networks, bio-informatics, etc.
Data type and generation/datasets
Synthetic data graph data.
Technology stack and implementation
Shell, GraphDB
Metrics
Main metrics include: (a) Query execution times for diverse graph sizes and query workloads, and (b) Query execution times for simple recursive queries on various small graph.
Reported results and usage
http://www.info.univ-tours.fr/ICVL/doc/jirc-2017/slides-radu-jirc.pdf
Reference papers
Bagan, Guillaume, et al. "gMark: Schema-driven generation of graphs and queries." IEEE Transactions on Knowledge and Data Engineering 29.4 (2016): 856-869.

7.12 Year 2015

SparkBench

Benchmark description
Benchmark Name
SparkBench
Short Description
SparkBench, developed by IBM, is a comprehensive Spark specific benchmark suite developed for in-memory data analysis to provide insights into Spark system design and performance optimization and cluster provisioning. The benchmark provides automatic generation of data sets with various scale factors. There are four main workload categories: machine learning, graph processing, streaming and SQL queries.
Web references
https://bitbucket.org/lm0926/sparkbench https://github.com/CODAIT/spark-bench

https://research.spec.org/fileadmin/user_upload/documents/wg_bd/BD-20150401-spark_benchmark-v1.3-spec.pdf
Date of last description update
2018
Originating group
IBM TJ Watson Research Center
Time – first version, last version
--
Type/Domain
Benchmark Suite
Workload
Four categories: ML(Logistic Regression, Support Vector Machine, Matrix factorization), Graph computation (PageRank, SVD++, TriangleCount), SQL query(Hive, RDD Relation), Streaming application (Twitter, Page review)
Data type and generation/datasets
The data type and generation is depending on different workloads. The LogRes and SVM use the Wikipedia data set. The MF, SVD++, and TriangleCount use the Amazon Movie Review data set. The PageRank uses Google Web Graph data. Twitter uses Twitter data. The SQL Queries workloads use E-commerce data. Finally, the PageView uses PageView DataGen to generate synthetic data.
Technology stack and implementation
Apache Spark >= 2.1.1
Metrics
SparkBench defines a number of metrics facilitating users to compare between various Spark optimizations, configurations and cluster provisioning options: (1) Job Execution Time(s) of each workload; (2) Data Process Rate (MB/seconds); and (3) Shuffle Data Size.
Reported results and usage
https://research.spec.org/fileadmin/user_upload/documents/wg_bd/BD-20150401-spark_benchmark-v1.3-spec.pdf
Reference papers
Min Li, Jian Tan, Yandong Wang, Li Zhang, and Valentina Salapura. "SparkBench: a spark benchmarking suite characterizing large-scale in-memory data analytics"

IoTABench

Benchmark description
Benchmark Name
IoTABench
Short Description
IoTABench is a benchmark toolkit for IoT Big Data scenarios, facilitating apples-to-apples comparisons between different sensor data and analytics platform. The benchmark can be extended to multiple IoT use-cases, including a user´s specific needs, interests or datasets.
Web references
http://marwah.org/publications/papers/icpe2015.pdf
Date of last description update
--
Originating group
HP Laboratories
Time – first version, last version
--
Type/Domain
IoT analytics benchmark
Workload
The workload is a smart-metering use case which involves generating, loading, repairing and analyzing synthetic meter readings. A set of relevant business queries are created that stress the database system under test.
Data type and generation/datasets
The benchmark provides Markov chain-based synthetic data generator which creates time series smart meter data. A large experimental study is provided, where 22.8 trillion smart meter readings totaling 727 TB of data is stored in an eight-node cluster.
Technology stack and implementation
HP Vertica Analytics Platform.

HP ProLiant DL380p g8 servers
Metrics
The benchmark reports metrics: (1) Generator performance_ million readings/sec, (2) Load and Repair performance_ million readings/sec, (3) Analysis performance_ query time in seconds
Reported results and usage
http://marwah.org/publications/papers/icpe2015.pdf
Reference papers
Martin Arlitt, Manish Marwah, Gowtham Bellala, Amip Shah, Jeff Healey, and Ben Vandiver. 2015. IoTAbench: an Internet of Things Analytics Benchmark. In Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering (ICPE '15). ACM, New York, NY, USA, 133-144. DOI: https://doi.org/10.1145/2668930.2688055

BigFUN

Benchmark description
Benchmark Name
BigFUN
Short Description
BigFUN is based on a social network use case with synthetic semi-structured data in JSON format. The benchmark focuses exclusively on micro-operation level and consists of queries with various operations such as simple retrieves, range scans, aggregations, joins, as well as inserts and updates.
Web references
https://github.com/pouriapirz/bigFUN https://www.ics.uci.edu/~pouria/bigfun/BigFUN_extended.pdf
Date of last description update
2016
Originating group
University of California, Oracle Labs
Time – first version, last version
--
Type/Domain

Social Network
Workload
Simple retrieves, range scans, aggregations, joins, as well as inserts and updates.
Data type and generation/datasets
Synthetic JSON data.
Technology stack and implementation
AsterixDB, MongoDB and Hive.
Metrics
Execution time.
Reported results and usage
https://www.ics.uci.edu/~pouria/bigfun/BigFUN_extended.pdf
Reference papers
Pouria Pirzadeh, Michael J. Carey, and Till Westmann. "BigFUN: A Performance Study of Big Data Management System Functionality"

TPCx-DSv2

Benchmark description
Benchmark Name
TPCx-DSv2
Short Description
TPCx-DSv2 is the version 2 of TPCx-DS benchmark which is an industry standard for benchmarking SQL based big data systems. TPC-DS is a decision support benchmark that models several generally applicable aspects of a decision support system, including queries and data maintenance. The benchmark provides a representative evaluation of the System Under Test's (SUT) performance as a general-purpose decision support system. In addition to TPCx-DS v1 workloads, this v2 benchmark provides a work stream to extend support for non relational (Hadoop etc.) systems.
Web references

http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-ds_v2.11.0.pdf
Date of last description update
2019
Originating group
TPC
Time – first version, last version
--
Type/Domain
Decision support benchmark
Workload
Queries of various operational requirements and complexities (e.g., ad-hoc, reporting, iterative OLAP, data mining).
Data type and generation/datasets
Benchmark provides synthetic data generator for generating structured datasets.
Technology stack and implementation
RDBMS as well as Hadoop/Spark based systems.
Metrics
A benchmark result measures query response time in single user mode, query throughput in multi user mode and data maintenance performance for a given hardware, operating system, and data processing system configuration under a controlled, complex, multi-user decision support workload.
Reported results and usage
--
Reference papers
--

CityBench

Benchmark description
Benchmark Name

CityBench
Short Description
CityBench is a configurable benchmark for evaluation of RDF Stream Processing (RSP) engines, by comparing them in terms of their capability to fulfil application-specific requirements (for smart city applications with smart city datasets).
Web references
http://iswc2015.semanticweb.org/sites/iswc2015.semanticweb.org/files/93670319.pdf https://github.com/CityBench/Benchmark
Date of last description update
2015
Originating group
--
Time – first version, last version
2015
Type/Domain
Stream processing
Workload
The workload contains a set of continuous queries covering a variety of data- and application-dependent characteristics and performance metrics, to be executed over RSP engines.
Data type and generation/datasets
CityBench includes real-time IoT data streams generated from various sensors deployed within the city of Aarhus, Denmark. The benchmark includes vehicle traffic dataset, parking dataset, weather dataset, pollution dataset, cultural event dataset, library events dataset, and user location stream. All of these datasets are semantically annotated and interlinked using the CityPulse information model[19] .
Technology stack and implementation
Two RSP engines, CQELS and C-SPARQL, are supported.
Metrics

[19] http://iot.ee.surrey.ac.uk:8080/info.html

The RSP engines are evaluated with respect to: (1) *Latency (ms)*_ by increasing the number of input streams within a query and by increasing the number of concurrent queries executed. (2) *Memory consumption (MB)*_ by observing the usage of system memory during the concurrent execution of an increasing number of queries and increasing size of background data. (3) *Completeness of results (%)*_ by executing Query Q1 with variable input rate of data streams.
Reported results and usage
https://github.com/CityBench/Benchmark
Reference papers
Muhammad Intizar Ali, Feng Gao and Alessandra Mileo, "CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets", The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, October 11-15, 2015, Bethlehem, PA, USA.

Graphalytics

Benchmark description
Benchmark Name
Graphalytics
Short Description
It is an industrial-grade benchmark for graph analysis platforms such as Giraph. It consists of six core algorithms, standard datasets, synthetic dataset generators, and reference outputs, enabling the objective comparison of graph analysis platforms.
Web references
http://ldbcouncil.org/ldbc-graphalytics https://graphalytics.org
Date of last description update
2019
Originating group
Linked Data Benchmark Council (LDBC)
Time – first version, last version
2016 – 2019
Type/Domain

Graph benchmark
Workload
Six core algorithms: breadth-first search, PageRank, weakly connected components, community detection using label propagation, local clustering coefficient, and single-source shortest paths.
Data type and generation/datasets
Synthetic data for graph queries (Wikipedia talk communication network dataset, Citation network among US patents, Game traces from the KGS Go Server, etc.)
Technology stack and implementation
Graph analysis platforms (Giraph, GraphX, OpenG, PowerGraph, GraphMat, Gelly, GraphBLAS, Gunrock, mvGRAPH).
Metrics
Execution time.
Reported results and usage
https://github.com/ldbc/ldbc_graphalytics https://graphalytics.org/competition
Reference papers
Alexandru Iosup, Tim Hegeman, Wing Lung Ngai, et al. "LDBC Graphalytics: A Benchmark for Large-Scale Graph Analysis on Parallel and Distributed Platforms"

Yahoo Streaming Benchmark (YSB)

Benchmark description
Benchmark Name
Yahoo Streaming Benchmark (YSB)
Short Description
It is an end-to-end pipeline that simulates a real-world advertisement analytics pipeline. Currently implemented in Kafka, Storm, Spark, Flink and Redis.
Web references
https://github.com/yahoo/streaming-benchmarks
Date of last description update
2019

Originating group
Yahoo!
Time – first version, last version
2015 – 2019
Type/Domain
Streaming
Workload
The job of the benchmark is to read various JSON events from Kafka, identify the relevant events, and store a windowed count of relevant events per campaign into Redis.
Data type and generation/datasets
The data schema consists of seven attributes and is stored in JSON format: (1) user-id: UUID; (2) page-id: UUID; (3) ad-id: UUID; (4) ad-type: String in banner, modal, sponsored-search, mail, mobile; (5) event-type: String in view, click, purchase; (6) event-time: Timestamp; (7) IP-address: String.
Technology stack and implementation
The YSB benchmark is implemented using Apache Storm, Spark, Flink, Apex, Kafka and Redis.
Metrics
The reported metrics by the benchmark are: (1) Latency in milliseconds that a particular system can produce at a given input load (calculated as *window.finalevent- latency = (window.last-updated-at – window.timestamp) – window.duration*; (2) Aggregate System Throughput.
Reported results and usage
https://developer.yahoo.com/blogs/135370591481/
Reference papers
Sanket Chintapalli, Derek Dagit, Bobby Evans, et al. “Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming”.

7.13 Year 2016

DeepBench

Benchmark description
Benchmark Name
DeepBench
Short Description
DeepBench is an open source benchmarking tool that measures the performance of basic operations (dense matrix multiplies, convolutions and communication) involved in training deep neural networks. The benchmark includes operations and workloads for both training and inference, These operations are executed on different hardware platforms using neural network libraries.
Web references
https://github.com/baidu-research/DeepBench
Date of last description update
2018
Originating group
--

Time – first version, last version
--
Type/Domain
Deep Learning
Workload
Workloads differs for each benchmarked operation: (1) Dense Matrix Multiply: Matrices with specified sizes (2) Convolution: Data in NCHW format (3) Recurrent Layers: Networks with set hidden units (4) All Reduce: Data with a set number of oat numbers
Data type and generation/datasets
No real data is used for the benchmarks. For all benchmarks random numbers are generated in a fitting format. For Matrix Multiply, this would be a matrix filled with random numbers for example. The used data is fundamentally very basic and small and results in fast benchmark times. Data is generated at run-time.
Technology stack and implementation
The main library used to implement the operations are NVIDIA's cuDNN and OpenMPI. The code itself was written in C++. Communication type operations are implemented with MPI. Note that not every hardware is supported by this benchmark.The main library used to implement the operations are NVIDIA's cuDNN and OpenMPI. The code itself was written in C++. Communication type operations are implemented with MPI. Note that not every hardware is supported by this benchmark.
Metrics
The benchmark measures time in milliseconds, FLOPS and bandwidth in GB/s.
Reported results and usage
https://github.com/baidu-research/DeepBench/tree/master/results
Reference papers
--

DeepMark

Benchmark description
Benchmark Name
DeepMark
Short Description
DeepMark or convnet-benchmarks is an open-source framework for benchmarking a collection of Convolutional Neural Networks. Convolutional Neural Networks are a special kind of neuronal networks

which are specifically designed for processing data that has a known grid-like topology. For instance time-series data, that can be seen as a simple one dimensional grid (1-Dgrid), or image data, that is a more complex two dimensional grid (2-D grid) which consists of pixels. For processing those grid-like data Convolutional Networks, a mathematical operation called convolution is used.
Web references
https://github.com/DeepMark/deepmark
Date of last description update
2016
Originating group
--
Time – first version, last version
--
Type/Domain
Deep Learning
Workload
For every use case, that DeepMark covers, a different workload is chosen. For image training the ImageNet data set is used. For video recognition the data set Sports-1M is used.
Data type and generation/datasets
Most of the data are publicly available data sets like ImageNet that use NCHW as structure. ImageNet consists 14,197,122 images and Sports-1M provides 1,133,158 video-url's.
Technology stack and implementation
Every application of deep learning models has different neural networks attached to it. For example, Recurrent Neural Networks Machine and Deep Learning Benchmarks are well suited for Speech recognition while Convolutional Neural Networks are especially good at image recognition.
Metrics
While the time measurement is mostly written in python or bash script, it will track, for each network defined epoch-time, the round-trip time for a single epoch of training. Also maximum batch-size will be defined according to the memory consumption, each framework uses.
Reported results and usage
--
Reference papers
--

TensorFlow Benchmarks

Benchmark description
Benchmark Name
TensorFlow Benchmarks
Short Description
A selection of image classification models is tested across multiple platforms to create a point of reference for the TensorFlow community.
Web references
https://www.tensorflow.org/performance/benchmarks
Date of last description update
2019
Originating group
Google
Time – first version, last version
2016 – 2019
Type/Domain
Deep Learning
Workload
Image classification workloads.
Data type and generation/datasets
ImageNet data set.
Technology stack and implementation
TensorFlow, Python
Metrics
Images/sec
Reported results and usage

https://www.tensorflow.org/guide/performance/benchmarks
Reference papers
--

Fathom

Benchmark description
Benchmark Name
Fathom
Short Description
Fathom: a collection of eight archetypal deep learning workloads for study. Each of these models comes from a seminal work in the deep learning community.
Web references
https://github.com/rdadolf/fathom
Date of last description update
2019
Originating group
Harvard University
Time – first version, last version
2016
Type/Domain
Deep Learning
Workload
Image classification, Speech recognition, language-to-language sentence translation.
Data type and generation/datasets
ImageNet, WMT15, bAbl, MNIST, TIMIT.
Technology stack and implementation
TensorFlow, Python, Docker

Metrics
Differences in similarity, their execution time, their performance and the effects of parallel scalability.
Reported results and usage
https://arxiv.org/pdf/1608.06581.pdf
Reference papers
Adolf, Robert, et al. "Fathom: Reference workloads for modern deep learning methods." 2016 IEEE International Symposium on Workload Characterization (IISWC). IEEE, 2016.

AdBench

Benchmark description
Benchmark Name
AdBench
Short Description
It combines Ad-Serving, Streaming Analytics on Ad-serving logs, streaming ingestion and updates of various data entities, batch-oriented analytics (e.g. for Billing), Ad-Hoc analytical queries, and Machine learning for Ad targeting. While this benchmark is specific to modern Web or Mobile advertising companies and exchanges, the workload characteristics are found in many verticals, such as Internet of Things (IoT), financial services, retail, and healthcare.
Web references
http://www.tpc.org/tpctc/tpctc2016/presentations_2016/session%20009-adbench.pdf
Date of last description update
2017
Originating group
Ampool
Time – first version, last version
--
Type/Domain
Application/ Data pipeline
Workload

Streaming Analytics on Ad-serving logs, streaming ingestion and updates of various data entities, batch-oriented analytics (e.g. for Billing), Ad-Hoc analytical queries, and Machine learning for Ad targeting. Workload characteristics are found in many verticals, such as Internet of Things (IoT), financial services, retail, and healthcare.
Data type and generation/datasets
Synthetic data of relational and streaming models.
Technology stack and implementation
Apex, Trafodion, HDFS, ampool
Metrics
Throughput, Query concurrency, Execution time for batch computation & Ad-Hoc queries, End-to-end latency, Operational complexity, Cost to meet SLAs
Reported results and usage
--
Reference papers
Milind Bhandarkar. "AdBench: A Complete Benchmark for Modern Data Pipelines".

RIoTBench

Benchmark description
Benchmark Name
RIoTBench
Short Description
A Real-time IoT Benchmark suite, consisting of 27 IoT micro-benchmarks and 4 real-application benchmarks reusing the micro-benchmark components, along with per-formance metrics. The goal of the benchmark suite is to evaluate the efficacy and performance of Distributed Stream Processing Systems (DSPS) in cloud environ-ments.
Web references
https://github.com/dream-lab/riot-bench
Date of last description update
2018
Originating group

Department of Computational and Data Sciences (Indian Institute of Science, India)
Time – first version, last version
--
Type/Domain
Streaming
Workload
The benchmark contains 27 micro-benchmarks (like: parse, filter, statistical analytics, predictive analytics, pattern detection and I/O operations) and 4 real-workload streaming application benchmarks (extract-transform-load (ETL) and archival, prediction and pattern detection, classification and notification, and summarization and visualization).
Data type and generation/datasets
Real data sets.
Technology stack and implementation
Storm
Metrics
The main metrics are throughput (messages per second), latency (processing time), jitter (deviation of the output throughput from the ideal throughput) and CPU and Memory utilization.
Reported results and usage
https://arxiv.org/pdf/1701.08530.pdf
Reference papers
Anshu Shukla, Shilpa Chaturvedi, and Yogesh Simmhan. "RIoTBench: A Realtime IoT Benchmark for Distributed Stream Processing Platforms". In: CoRR abs/1701.08530 (2017).

Hobbit Benchmark

Benchmark description
Benchmark Name
Hobbit Benchmark
Short Description
The HOBBIT evaluation platform is a distributed FAIR benchmarking platform for the Linked Data lifecycle. This means that the platform was designed to provide means to: (1) benchmark any step of the Linked Data lifecycle, including generation and acquisition, analytics and processing, storage and curation as well as

visualization and services;(2) ensure that benchmarking results can be found, accessed, integrated and reused easily (FAIR principles); (3) benchmark Big Data platforms by being the first distributed benchmarking platform for Linked data.
Web references
https://project-hobbit.eu/
Date of last description update
2019
Originating group
H2020 Project
Time – first version, last version
2017 – 2019
Type/Domain
Benchmark Platform
Workload
Real-world application workloads.
Data type and generation/datasets
Linked data.
Technology stack and implementation
Java
Metrics
Depending on the executed benchmark.
Reported results and usage
https://hobbit-project.github.io/index.html https://github.com/hobbit-project
Reference papers
Axel-Cyrille Ngonga Ngomo and Michael Röder. “HOBBIT: Holistic benchmarking for big linked data”. In: ERCIM News 2016.105

TPCx-BB (BigBench)

Benchmark description
Benchmark Name
TPCx-BB (BigBench)
Short Description
TPCx-BB is a measure the performance of Hadoop based Big Data systems systems. Based on BigBench, it measures the performance of both hardware and software components by executing 30 frequently performed analytical queries in the context of retailers with physical and online store presence.
Web references
http://www.tpc.org/tpcx-bb/
Date of last description update
2016
Originating group
Transaction Processing Performance Council (TPC)
Time – first version, last version
2015 – 2016
Type/Domain
BigBench is an end-to-end, technology agnostic, application-level benchmark that tests the analytical capabilities of a Big Data platform. It is based on a fictional product retailer business model.
Workload
The business model and a large portion of the data model's structured part is derived from the TPC-DS benchmark. The structured part was extended with table for the prices of the retailer's competitors, the semi-structured part was added represented by a table with website logs and the unstructured part was added by a table showing product reviews. The simulated workload is based on a set of 30 queries covering the different aspects of Big Data analytics proposed by McKinsey.
Data type and generation/datasets
Synthetic data generator for structured, semi-structured and unstructured data.
Technology stack and implementation
Since the BigBench specification is general and technology agnostic, it should be implemented specifically for each Big Data system. The initial implementation of BigBench was made for the Teradata Aster platform. It was done in the Aster's SQL-MR syntax served - additionally to a description in the English language - as

an initial specification of BigBench's workloads. Meanwhile, BigBench is implemented for Hadoop, using the MapReduce engine and other components like Hive, Mahout, Spark SQL, Spakr MLlib and OpenNLP from the Hadoop Ecosystem.
Metrics
TPCx-BB defines the following primary metrics: (1) BBQpm@SF, the performance metric, reflecting the TPCx-BB Queries per minute throughput; where SF is the Scale Factor; (2) $/BBQpm@SF, the price/performance metric; and (3) System Availability Date as defined by the TPC Pricing Specification.
Reported results and usage
http://www.tpc.org/tpcx-bb/results/tpcxbb_result_detail.asp?id=119071401
Reference papers
Ahmad Ghazal, Tilmann Rabl, Minqing Hu, et al. "BigBench: towards an industry standard benchmark for Big Data analytics".

7.14 Year 2017

Sanzu

Benchmark description
Benchmark Name
Sanzu
Short Description
It is a data science benchmark for evaluating systems for data processing and analytics tasks such as Anaconda, PySpark, MADlib and R. The benchmark consists of micro (basic file I/O, data wrangling, descriptive statistics, distribution and inferential statistics, time series and machine learning) and macro (smart grid analytics and sport analytics) benchmark suites.
Web references
http://bigdata.cs.unb.ca/projects/sanzu/
Date of last description update
2018
Originating group
--
Time – first version, last version

--
Type/Domain
Machine Learning
Workload
It contains a micro benchmark and a macro benchmark. The micro benchmark consists of six workloads which are Basic File I/O, Data Wrangling, Descriptive Statistical, Distribution and Inferential Statistics, Time Series and Machine Analyses. The macro benchmark contains two applications that are modelled based on real-world use cases, namely Smart Grid Analytics and Sport Analytics, both of which involve reading data from files, data wrangling and model building.
Data type and generation/datasets
In the micro benchmark, datasets are generated from a synthetic data generator. Each dataset is generated under a given scale factor ranging from 1 million rows to 100 million rows. The types and schema of data contains time series, string, integer, float, even sequential time series. Some of the columns are chosen uniformly from a list while others are chosen from a normal distribution, an exponential distribution. For the macro benchmark, it uses real-world data resources.
Technology stack and implementation
In order to run the Sanzu benchmark, one should first install the five platforms that are Anaconda Python, R, Dask, PostgreSQL with MADLib and Spark. The following step is generating datasets by running a shell script file (create-dataset.sh). Next, one can run the tasks in python console, the results of which will be stored in the directory under benchmark/benchmark.csv.
Metrics
The metric used to evaluate the performance and functionality of 5 popular data science platforms is the execution time measured from the completion of a set of tasks. The scale factors range from 1 million data rows per table, 10 million data rows per table, to 100 million data rows per table.
Reported results and usage
--
Reference papers
Alex Watson, Deepigha Shree Vittal Babu, and Suprio Ray. “Sanzu: A data science benchmark”. In: 2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017

AIM Benchmark

Benchmark description
Benchmark Name
AIM Benchmark

Short Description
The AIM benchmark simulates a challenging use case of how to store and analyze billing data of subscribers and make marketing campaigns immediately available. The task is to process single events like phone calls or messages and to do real-time analytics which are represented by seven analytical queries. The workload is well-defined and fits perfectly in the class of analytics on fast data.
Web references
https://github.com/tellproject/aim-benchmark
Date of last description update
2018
Originating group
TUM Living Lab Connected Mobility (TUM LLCM) project, University of Munich, Oracle Labs
Time – first version, last version
2017 – 2018
Type/Domain
Streaming Benchmark
Workload
The workload of the AIM Benchmark is based on Analytics Matrix and is divided into two parts: (1) First, the system must process the stateful streaming workload depicted by the events which generate sales and marketing information through phone calls. This is called Event Stream Processing (ESP) and is divided in two phases. The ESP should update the Analytics Matrix immediately after an event arrives. As a default there are 10.000 events per second and each consist of a subscriber id (to identify the subscriber) and call-dependent details such as the call's duration, cost and type. (2) Then in the second phase the updated Analytics Matrix is made available for analytical queries and the updated record and event are checked against a set of triggers. There are 7 standard queries which are continuous queries from one or multiple clients which are answered by the Analytical Matrix. Each query is executed with the same probability. The state of the Matrix should not be older than 1 second. For example, one query selects all local and long-distance calls per region with the category "eat".
Data type and generation/datasets
The main part of the data for the benchmark is the Analytics Matrix. This Matrix contains aggregated data for each subscriber, identified by the subscriber id. Each row of the matrix represents a subscriber. The columns represent the aggregated data for each combination of the aggregation functions like sum, min, max and aggregation window like the day and several event attributes. By default, the Matrix consists of 546 columns and 10 million rows. Furthermore, there are links into the dimension table through foreign keys. The dimension tables contain the information and structure of the data in the Analytical Matrix like Region Info and Subscription Type. There is an open-source AIM schema generator which can be used to generate the aggregated structure.

Technology stack and implementation
The AIM Benchmark can be implemented on several systems like multimedia databases (MMDBs) such as HyPer or Tell, modern streaming systems like Flink and hand-crafted systems. There is a Tell implementation available in GitHub. In HyPer the analytics matrix can be implemented as a regular database table and the real time analytics as SQL queries on this table. The handcrafted AIM System is designed for the AIM Benchmark and so it achieves the best performance on the workload and can be used as a performance orientation for other implementations.
Metrics
Main metrics are: (a) Overall performance, (b) Read performance, (c) Write performance, (d) query response times, and (e) Impact of number of aggregates. The performance of the implemented system is measured as the query throughput dependent on the available amount of threads and can be distinguished in the overall performance, the read and write performance. The read performance focuses on the analytic queries, while the write performance on the event processing measured as the response time, with and without concurrent writes, for the events per second. To test the performance the number of clients and maintained aggregates can be varied.
Reported results and usage
https://www.db.ics.keio.ac.jp/seminar/2019/20190607_tasuku/Scalable%20Analytics%20on%20Fast%20Data.pdf
Reference papers
Andreas Kipf, Varun Pandey, Jan Böttcher, et al. "Analytics on Fast Data: Main-Memory Database Systems versus Modern Streaming Systems". In: Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017. 2017, pp. 49–60.

GARDENIA

Benchmark description
Benchmark Name
GARDENIA
Short Description
Gardenia is a domain-specific benchmark suite consisting of irregular graph workloads. These workloads mimic actual machine learning and Big Data applications running on modern datacenter accelerators using state-of-the-art optimization techniques.
Web references
https://github.com/chenxuhao/gardenia
Date of last description update
2018

Originating group
National University of Defence Technology
Time – first version, last version
2017 – 2018
Type/Domain
Graph benchmark
Workload
Breadth-First Search (BFS), Single-Source Shortest Paths (SSSP), Betweenness Centrality (BC), PageRank (PR), Connected Components (CC), Triangle Counting (TC), Stochastic Gradient Descent (SGD), Sparse Matrix-Vector Multiplication (SpMV), and Symmetric Gauss-Seidel smoother (SymGS).
Data type and generation/datasets
Datasets from the UF Sparse Matrix Collection, the SNAP datasetCollection, and the Koblenz Network Collection.
Technology stack and implementation
OpenMP, CUDA
Metrics
Exeution time, IPC(Instructions per cycle)
Reported results and usage
https://arxiv.org/pdf/1708.04567.pdf
Reference papers
Xu, Zhen, et al. "GARDENIA: A Domain-specific Benchmark Suite for Next-generation Accelerators." arXiv preprint arXiv:1708.04567 (2017).
Benchmark description
Benchmark Name
GARDENIA
Short Description
Gardenia is a domain-specific benchmark suite consisting of irregular graph workloads. These workloads mimic actual machine learning and Big Data applications running on modern datacenter accelerators using state-of-the-art optimization techniques.
Web references

https://github.com/chenxuhao/gardenia
Date of last description update
2018
Originating group
National University of Defence Technology
Time – first version, last version
--
Type/Domain
Graph benchmark
Workload
Breadth-First Search (BFS), Single-Source Shortest Paths (SSSP), Betweenness Centrality (BC), PageRank (PR), Connected Components (CC), Triangle Counting (TC), Stochastic Gradient Descent (SGD), Sparse Matrix-Vector Multiplication (SpMV), and Symmetric Gauss-Seidel smoother (SymGS).
Data type and generation/datasets
Datasets from the UF Sparse Matrix Collection, the SNAP datasetCollection, and the Koblenz Network Collection.
Technology stack and implementation
OpenMP, CUDA
Metrics
Exeution time, IPC(Instructions per cycle)
Reported results and usage
https://arxiv.org/pdf/1708.04567.pdf
Reference papers
Xu, Zhen, et al. "GARDENIA: A Domain-specific Benchmark Suite for Next-generation Accelerators." arXiv preprint arXiv:1708.04567 (2017).

Penn Machine Learning Benchmark (PMLB)

Benchmark description
Benchmark Name

Penn Machine Learning Benchmark (PMLB)
Short Description
It includes most of the real-world benchmark datasets commonly used in ML benchmarking studies such as UCI ML repository, Kaggle, KEEL and the meta-learning benchmark.
Web references
https://github.com/EpistasisLab/penn-ml-benchmarks
Date of last description update
2018
Originating group
Institute for Biomedical *Informatics(University* of Pennsylvania, USA), Department of Automatics and Biomedical Engineering (AGH University of Science and Technology, Poland)
Time – first version, last version
--
Type/Domain
Machine Learning
Workload
The main part of the workload is to compare the datasets in PMLB, which are clustered based on their meta-features, and to analyze the datasets based on ML performance, which identifies which datasets can be solved with high or low accuracy.
Data type and generation/datasets
PMLB is initialized with 165 real-word, simulated and toy benchmark datasets and evaluate the performance of 13 standard statistical methods from scikit-Learn over the full set of PMLB datasets.
Technology stack and implementation
When each ML method is evaluated, the features of the datasets are scaled by subtracting the mean and scaling the meta-features of the datasets into 5 clusters. To find the best parameters for each ML method on each dataset, a comprehensive grid search of each of the ML method's parameters is performed using 10-fold cross-validation. All clusters are compared in more detail according to the mean values of the dataset meta-features in each cluster. Using a spectral bi-clustering algorithm Kluger et al., the 13 ML models and 165 datasets are bi-clustered according to the balanced accuracy of the models using their best parameter setting.
Metrics
ML methods are evaluated using balanced accuracy as the scoring metric. This is a normalized version of accuracy that accounts for class imbalance by calculating accuracy on a per-class basis then averaging the per-class accuracies.

Reported results and usage
https://biodatamining.biomedcentral.com/articles/10.1186/s13040-017-0154-4
Reference papers
Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. "PMLB: a large benchmark suite for machine learning evaluation and comparison". In: BioData Mining 10.1 (2017)

BenchIP

Benchmark description
Benchmark Name
BenchIP
Short Description
BENCHIP focuses on benchmarking intelligent processors and system optimization. It is ultilized for comparison of optimization and bottleneck analysis of hardware platforms.
Web references
https://arxiv.org/pdf/1710.08315.pdf
Date of last description update
2017
Originating group
ICT CAS, Cambricon, Alibaba Group, AMD, RDA Microelectronics, JD, IFLYTEK
Time – first version, last version
2017
Type/Domain
Benchmark Suite
Workload
Two sets of bechmarks: (1) Micro benchmarks containing 12 single layer networks. (2) Macro benchmarks focussing on entire neural networks (LeNet-5, RNN, AlexNET, VGG, ResNet, Faster R-CNN, Deep Face Recognition, DeconvNet, FCLN, S2VT, SyntaxNet).
Data type and generation/datasets
Depending on the workload synthetic or real data sets.

Technology stack and implementation
Caffe, BLAS, cuDNN, IPLib.
Metrics
The metrics used for evaluation are accuracy, performance, and energy of deep learning intelligence processors.
Reported results and usage
https://arxiv.org/pdf/1710.08315.pdf
Reference papers
Tao, Jin-Hua, et al. "BenchIP: Benchmarking Intelligence Processors." Journal of Computer Science and Technology 33.1 (2018): 1-23.

Deep Learning Benchmarking Suite (DLBS)

Benchmark description
Benchmark Name
Deep Learning Benchmarking Suite (DLBS)
Short Description
Deep Learning Benchmarking Suite (DLBS) is a collection of command line tools for running deep learning benchmark experiments on various hardware/software platforms.
Web references
https://hewlettpackard.github.io/dlcookbook-dlbs/#/index?id=deep-learning-benchmarking-suite https://github.com/HewlettPackard/dlcookbook-dlbs
Date of last description update
2019
Originating group
Hewlett Packard Labs (HPL)
Time – first version, last version
2018
Type/Domain

Benchmark Suite
Workload
Training of eighteen deep learning models.
Data type and generation/datasets
Synthetic and real image datasets that can be used with convolutional neural networks. The actual format of datasets is dependent on framework type.
Technology stack and implementation
Caffe, Caffe2, MXNet, PyTorch, TensorRT.
Metrics
Number of data samples per second.
Reported results and usage
--
Reference papers
--

TPCx-IoT

Benchmark description
Benchmark Name
TPCx-IoT
Short Description
The TPC Benchmark IoT (TPCx-IoT) benchmark workload is designed based on Yahoo Cloud Serving Benchmark (YCSB). It is not comparable to YCSB due to significant changes. The TPCx-IoT workloads consists of data ingestion and concurrent queries simulating workloads on typical IoT Gateway systems. The dataset represents data from sensors from electric power station(s).
Web references
http://www.tpc.org/tpcx-iot/
Date of last description update
2018
Originating group

Transaction Processing Performance Council (TPC)
Time – first version, last version
2015 – 2018
Type/Domain
IoT gateway system.
Workload
The System Under Test (SUT) must run a data management platform that is commercially available and data must be persisted in a non-volatile durable media with a minimum of two-way replication. The workload represents data injected into the SUT with analytics queries in the background. The analytic queries retrieve the readings of a randomly selected sensor for two 30 second time intervals, TI1 and TI2. The first time interval TI1 is defined between the timestamp the query was started Ts and the timestamp 5 seconds prior to Ts , i.e. TI1 =[Ts-5,Ts]. The second time interval is a randomly selected 5 seconds time interval TI2 within the 1800 seconds time interval prior to the start of the first query, Ts-5. If Ts <=1810, prior to the start of the first query, Ts-5.
Data type and generation/datasets
Each record generated consists of driver system id, sensor name, time stamp, sensor reading and padding to a 1 Kbyte size. The driver system id represents a power station. The dataset represents data from 200 different types of sensors.
Technology stack and implementation
The benchmark currently supports the HBase 1.2.1 and Couchbase-Server 5.0 NoSQL databases. A guide providing instructions on how to add new databases is also available.
Metrics
TPCx-IoT was specifically designed to provide verifiable performance, price-performance and availability metrics for commercially available systems that typically ingest massive amounts of data from large numbers of devices. TPCx-IoT defines the following primary metrics: (1) IoTps as the performance metric; (2) $/IoTps as the price-performance metric; and (3) system availability date.
Reported results and usage
http://www.tpc.org/tpcx-iot/results/tpcx-iot_result_detail-5757.asp
Reference papers
Raghunath Nambiar and Meikel Poess. “Reinventing the TPC: From Traditional to Big Data to Internet of Things”. In: Performance Evaluation and Benchmarking: Traditional to Big Data to Internet of Things - 7th TPC Technology Conference, TPCTC 2015, Kohala Coast, HI, USA, August 31 - September 4, 2015. Revised Selected Papers. 2015, pp. 1–7.

Senska

Benchmark description
Benchmark Name
Senska
Short Description
It is an enterprise streaming benchmark. It consists of three major components: data feeder, system under test and result validator.
Web references
http://materials.dagstuhl.de/files/17/17441/17441.GünterHesse.Slides.pdf
Date of last description update
2017
Originating group
Hasso Platter Institute (University of Potsdam, Germany)
Time – first version, last version
--
Type/Domain
Streaming
Workload
Senska is following the domain-specific criteria defined by Gray. While keeping relevance, portability, scalability, and simplicity in focus, Senska do have three main components: the data feeder, system under test (SUT) and the result validator. As these components have to interact, there are some additional components which are responsible for the communication between the main components. The main queries handle nine main testing aspects: windowing, transformation, merging (union), filtering (selection/projection), sorting/ranking, correlation/enrichment (join), machine learning, and combination with DBMS data. In the first described query set seven of these nine aspects are already fulfilled within five use cases. Only merging (union) and sorting/ranking are not yet covered by the actual query definitions.
Data type and generation/datasets
Senska takes as input data a csv-file which should contain representative data for a manufacturing context (e.g. sensor data). These data will be handled by the data feeder which puts the data into the SUT via communication channels. Here, only Apache Kafka can be used as data feeder, therefore the SUT has to be able to communicate with this message broker. The SUT is split into two parts, the Benchmark Query Implementation and the DBMS. The benchmark query implementation executes the actual benchmark queries, while the DBMS is used to feed the benchmark with historical data whenever needed. This behavior is unique to the Senska benchmark, as all other benchmarks do only use sensor data for their results.

Technology stack and implementation
Apache Kafka Streaming application, toolkit is using a JVM language.
Metrics
During the execution of the benchmark queries, the results are send to the result validator. This component will then ensure the correctness of the query implementation and calculate the benchmark metrics. Unfortunately, there is no information about which metrics the benchmark will handle after its release.
Reported results and usage
https://hpi.de/plattner/publications/all-publications.html?tx_extbibsonomycsl_publicationlist%5BuserName%5D=import_epic&tx_extbibsonomycsl_publicationlist%5BintraHash%5D=ae95a69252021219a69f354955c4176c&tx_extbibsonomycsl_publicationlist%5BfileName%5D=HesseGuenter_TowardsEnterpriseStreamingBenchmark.pdf&tx_extbibsonomycsl_publicationlist%5Bcontroller%5D=Document&cHash=237b9b6fdc1891272a6f3f335cb921ae
Reference papers
Hesse, Guenter, et al. "Senska–Towards an Enterprise Streaming Benchmark." Technology Conference on Performance Evaluation and Benchmarking. Springer, Cham, 2017.

DAWNBench

Benchmark description
Benchmark Name
DAWNBench
Short Description
DAWNBench is a benchmark suite for end-to-end deep learning training and inference. It provides a reference set of common deep learning workloads for quantifying training time, training cost, inference latency, and inference cost across different optimization strategies, model architectures, software frameworks, clouds, and hardware.
Web references
https://dawn.cs.stanford.edu/benchmark/ https://cs.stanford.edu/~deepakn/assets/papers/dawnbench-sosp17.pdf
Date of last description update
2019
Originating group

Stanford University
Time – first version, last version
--
Type/Domain
End-to-End Deep Learning
Workload
DAWNBench offers the possibility to define custom values for the choice of optimizer, batch size, multi-GPU training and stochastic depth.The available optimizers are Adam, Single-node multi-GPU training and Stochastic Depth. Adam is an adaptive optimizer for gradient descent. At Stochastic Depth entire layers are randomly dropped during training in order to prevent co-adaptation. They investigate three different batch sizes: 32, 256 and 2048. For the models can be chosen different ResNet architectures: ResNet20, ResNet56 and ResNet164. It can be tested on different hardware platforms: GPUs and CPUs with different kernel sizes.
Data type and generation/datasets
It uses the ImageNet and CIFAR10 databases and question and answer on SQuAD.
Technology stack and implementation
The current implementations are on PyTorch and TensorFlow.
Metrics
Four metrics are available: 1) training time, to a specified validation accuracy: 2) cost; 3) average latency of performing inference on a single item (image or question) and 4) average cost of inference for 10 000 items.
Reported results and usage
https://dawn.cs.stanford.edu/benchmark/ https://github.com/stanford-futuredata/dawn-bench-entries
Reference papers
Cody Coleman, Daniel Kang, Deepak Narayanan, et al. “Analysis of DAWNBench, a Time-to-Accuracy Machine Learning Performance Benchmark”.

BlockBench

Benchmark description
Benchmark Name
BlockBench

Short Description
BlockBench is the first benchmarking framework for private blockchain systems. It serves as a fair means of comparison for different platforms and enables deeper understanding of different system design choices. It comes with both macro benchmark workloads for evaluating the overall performance and micro benchmark workloads for evaluating performance of individual layers.
Web references
https://www.comp.nus.edu.sg/~dbsystem/blockbench/
Date of last description update
2017
Originating group
National University of Singapore (NUS) Computing
Time – first version, last version
--
Type/Domain
Blockchain
Workload
Macro benchmarks to evaluate the overall performance and specific micro workloads.
Data type and generation/datasets
Real and synthetic smart contracts.
Technology stack and implementation
Ethereum, Parity and Hyperledger Fabric. Other private blockchains can be integrated by using an API.
Metrics
Throughput, Latency, Scalability, Fault tolerance.
Reported results and usage
https://www.comp.nus.edu.sg/~dbsystem/blockbench/
Reference papers
Tien Tuan Anh Dinh, Ji Wang, Gang Chen, et al. “BLOCKBENCH: A Framework for Analyzing Private Blockchains”. In: Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017. 2017, pp. 1085–1100.

IDEBench

Benchmark description
Benchmark Name
IDEBench
Short Description
IDEBench measures the performance of interactive data exploration systems over the course of entire user-centered workflows, where queries are built and refined incrementally and executed with delays (thinktime) between queries, rather than being processed back-to-back. Each workflow comprises a sequence of interactions performed by users: Creating a visualization (i.e., the starting query), filtering/selecting, linking, and discarding a visualization.
Web references
https://idebench.github.io
Date of last description update
2018
Originating group
MIT
Time – first version, last version
--
Type/Domain
Interactive data exploration.
Workload
Performance of interactive data exploration systems over the course of entire user-centered workflows.
Data type and generation/datasets
Synthetic data sets.
Technology stack and implementation

Python, MonetDB
Metrics
Time Requirement (TR) Violated: Boolean whether a query violated the time requirement Time Requirement (TR). Missing Bins: The ratio of the number of bins for which no result has been delivered and the total number of bins in the ground-truth. Mean Relative Error: The mean relative error of all bins returned in the result (see definition below). Cosine Distance: A measure of how much the "shape" of a result resembles the ground-truth. Mean Margin of Error: The mean of all relative margins of error for all bins. Out of Margin: The number of approximate results that were outside of the return confidence interval. Bias: The sum of all returned values in a result divided by the sum of all true results for the bins returned.
Reported results and usage
https://arxiv.org/abs/1804.02593
Reference papers
Eichmann, Philipp, et al. "Idebench: A benchmark for interactive data exploration." arXiv preprint arXiv:1804.02593 (2018). Eichmann, Philipp, Emanuel Zgraggen, Zheguang Zhao, Carsten Binnig, and Tim Kraska. "Towards a Benchmark for Interactive Data Exploration." IEEE Data Eng. Bull. 39, no. 4 (2016): 50-61.

TPCx-V

Benchmark description
Benchmark Name
TPCx-V
Short Description
The TPCx-V benchmark measures the performance of a server running virtualized databases. It simulate a mix of On Line Transaction Processing (OLTP) and Decision Support Systems (DSS) workloads in cloud computing environment.
Web references
http://www.tpc.org/tpcx-v/default.asp

Date of last description update
2018
Originating group
Transaction Processing Performance Council (TPC)
Time – first version, last version
2015 – 2016
Type/Domain
OLTP and OLAP, structured data
Workload
OLTP / DSS workloads.
Data type and generation/datasets
Synthetic data.
Technology stack and implementation
VMs, relational DBs.
Metrics
The Performance Metric reported by TPCx-V istpsV, which is a "business throughput" measure of the number of completed Trade-Result transactions per second.
Reported results and usage
http://www.tpc.org/tpcx-v/results/tpcx-v_result_detail-5301.asp
Reference papers
Andrew Bond, Douglas Johnson, Greg Kopczynski, and H. Reza Taheri. "Profiling the Performance of Virtualized Databases with the TPCx-V Benchmark"

Stream-WatDiv

Benchmark description
Benchmark Name
Stream Watdiv
Short Description

Stream-WatDiv is an open-source benchmark for streaming RDF data management systems, to evaluate streaming RDF processing engines. It is extended from Waterloo SPARQL Diversity Test Suite (WatDiv), and includes a streaming data generator, a query generator that can produce a diverse set of SPARQL queries, and a testbed to monitor correctness and latency.
Web references
https://uwspace.uwaterloo.ca/bitstream/handle/10012/12886/Gao_Libo.pdf?sequence=1&isAllowed=y https://dsg.uwaterloo.ca/watdiv/stream-watdiv https://github.com/nosrepus/Stream-WatDiv
Date of last description update
2018
Originating group
University of Waterloo
Time – first version, last version
2017
Type/Domain
Streaming
Workload
Streaming RDF queries involving both streaming and static data.
Data type and generation/datasets
Stream WatDiv generates two datasets of varying scale factor, streaming dataset and static dataset. These two datasets together describe an e-commerce website database.
Technology stack and implementation
Stream WatDiv supports evaluation of two popular streaming RDF engines: C-SPARQL and CQUELS.
Metrics
Average latency (ms)
Reported results and usage
https://dsg.uwaterloo.ca/watdiv/stream-watdiv
Reference papers
Libo Gao (2018). Stream WatDiv - A Streaming RDF Benchmark. UWSpace. http://hdl.handle.net/10012/12886

7.15 Year 2018

ABench

Benchmark description
Benchmark Name
ABench
Short Description
ABench is as a big data architecture stack benchmark. It aims to evaluate big data system across multiple layers of big data architecture, including cloud services, data storage, batch processing, interactive processing, streaming and machine learning. The benchmark supports re-using of existing benchmarks such as BigBench.
Web references
https://www.slideshare.net/DataBench/abench-big-data-architecture-stack-benchmark-125706043 https://www.researchgate.net/publication/324364983_ABench_Big_Data_Architecture_Stack_Benchmark
Date of last description update
2019
Originating group
--
Time – first version, last version
--

Type/Domain
Big Data architecture stack benchmark
Workload
The workloads could include graph queries, operational system workload, machine learning analytics, continuous queries and stream analytics. Current implementation extends BigBench workloads for stream processing and machine learning.
Data type and generation/datasets
Structured, Text, JSON logs. Benchmark provides both data generator and public datasets to stress the architecture.
Technology stack and implementation
Benchmark implementation in year 2018 supports Spark Streaming, Kafka, SparkMLlib.
Metrics
Main metrics are: execution time of a task for a SUT and end-to-end execution time of an application scenario. The benchmark has proposed to measure scalability, reliability, throughput, energy efficiency and cost in future implementations.
Reported results and usage
--
Reference papers
ABench: Big Data Architecture Stack Benchmark by Todor Ivanov (University of Frankfurt, Germany) and Rekha Singhal (TCS Research, India)

TERMinator Suite

Benchmark description
Benchmark Name
TERMinator Suite
Short Description
TERMinator is a benchmark suite for evaluating and comparing encrypted computer architectures based on homomorphic operations, avoiding termination problems while maintaining data privacy.
Web references
https://eprint.iacr.org/2017/1218.pdf

Date of last description update
2019
Originating group
Authors from University of Athens, Greece and New York University, New York. Benchmarks are maintained by the Modern Microprocessors Architecture Lab at New York University Abu Dhabi and the Trustworthy Computing Group at University of Delaware.
Time – first version, last version
--
Type/Domain
Benchmark Suite
Workload
Benchmarks are categorized in four classes, depending on the type and features of the main loop iteration: • Synthetic: This class comprises of primitive recursive benchmarks (such as Tak Error! Reference source not found. and N-Queens), which allow assessing the universality of an abstract machine with respect to encrypted computation, as well as the performance of encrypted data structures (e.g., a stack). • Microbenchmarks: This class evaluates the performance of homomorphic addition and multiplication, which are critical micro-operations of encrypted abstract machines. Examples in this class include Factorial (multiplication intensive), Fibonacci (addition-intensive) and PIR (both addition- and multiplication-intensive). • Kernels: This class focuses on evaluating essential core loops of different real-life applications, which combine memory swaps, branch decisions and arithmetic operations. Example benchmarks in this class include Insertion Sort, PSI, Deduplication (i.e., Set Union), Matrix Multiplication, Primes (i.e., Sieve of Eratosthenes) and Permutations. • Encoder Benchmarks: This class comprises three real-life cryptographic and hash applications (namely Speck, Simon and Jenkins), which are demanding in terms of bitwise operations and branch decisions on encrypted values, and allow assessing the BRO of the target abstract machine.
Data type and generation/datasets
Structured, BI.
Technology stack and implementation
Privacy-preserving computer architectures based on homomorphic operations; Cryptoleq architecture with Cryptoleq Enhanced Assembly Language (CEAL).
Metrics
Performance over different security configurations. Baseline runtime measurements and execution statistics

Reported results and usage
http://eprint.iacr.org/2017/1218.pdf https://github.com/momalab/TERMinatorSuite
Reference papers
D. Mouris, N. G. Tsoutsos and M. Maniatakos, "TERMinator Suite: Benchmarking Privacy-Preserving Architectures." IEEE Computer Architecture Letters, Volume: 17, Issue: 2, July-December 2018.

HERMIT

Benchmark description
Benchmark Name
HERMIT
Short Description
HERMIT (HEalthcaRe Monitoring for the Internet of Things) is a benchmark suite for IoT applications in healthcare industry. The goal of benchmark is to facilitate research into new microarchitectures and optimizations that will enable efficient execution of emerging Internet of Medical Things (IoMT) applications. HERMIT comprises of applications spanning various domains in the healthcare industry, including computerized tomography scan, ultrasound, magnetic resonance imaging, implantable heart monitors, wearable devices. HERMIT also includes supplementary applications for security and data compression. In addition, HERMIT is compared to three commonly used benchmark suites: 1) MiBench; 2) SPEC CPU2006; and 3) PARSEC, which indicates that IoMT applications' characteristics differ from existing benchmarks.
Web references
http://www2.engr.arizona.edu/~tosiron/papers/2018/HERMIT_IoTJ18.pdf https://github.com/ankurlimaye/HERMIT-BenchmarkSuite
Date of last description update
2018
Originating group
--
Time – first version, last version
2018
Type/Domain
Edge computing

Workload
The workload comprises of a selected set of ten applications, categorized in two parts: (1) computation applications and (2) communication protocol. *Computation applications* include Physical activity estimation (activity), Sleep apnea detection (apdet), Heart rate variability calculation (hrv), Histogram equalization (imghist), Inverse Radon Transform (iradon), k-means clustering (kmeans), ECG-QRS detection (sqrs), and Blood pressure monitoring (wabp). *Communication protocol* category includes Advanced Encryption Standard (aes) and Lempel-Ziv compression (lzw) to represent security and compression functions, respectively.
Data type and generation/datasets
HERMIT
Technology stack and implementation
HERMIT applications are executed on the Raspberry Pi 3 platform.
Metrics
The benchmark focuses on the execution characteristics that are observable from the hardware performance counters: (1) *IPC (Instructions Per Cycle)* _ average number of instructions that are executed by the processor in each clock cycle. (2) *Memory characteristics* _ cache accesses per thousand instructions and cache miss rates. *(3) Branch characteristics* _ function of the percentage of branch instructions and the branch miss rates.
Reported results and usage
https://github.com/ankurlimaye/HERMIT-BenchmarkSuite
Reference papers
Ankur Limaye and Tosiron Adegbija, "HERMIT: A Benchmark Suite for the Internet of Medical Things", Published in: IEEE Internet of Things Journal (Volume: 5 , Issue: 5 , Oct. 2018).

MLBench Services

Benchmark description
Benchmark Name
MLBench Services
Short Description
The MLBench services benchmark, inspired by Kaggle, consists of datasets with a best-effort baseline of both feature engineering and machine learning models. It uses a novel metric based on the notion of "quality tolerance" that measures the performance gap between a given machine learning system and top-ranked

Kaggle performers. Currently are available 7 binary classification datasets, 5 multi-class classification datasets and 5 regression datasets.
Web references
https://www.microsoft.com/en-us/research/publication/mlbench-benchmarking-machine-learning-services-human-experts/ http://www.vldb.org/pvldb/vol11/p1220-liu.pdf
Date of last description update
2018
Originating group
ETH Zürich and Microsoft
Time – first version, last version
2018
Type/Domain
Machine Learning
Workload
Currently are available 7 binary classification datasets, 5 multi-class classification datasets and 5 regression datasets.
Data type and generation/datasets
18 real world datasets of the Kaggle competition.
Technology stack and implementation
Cloud infrastructure: Microsoft Azur, Amazon.
Metrics
It uses a novel metric based on the notion of "quality tolerance" that measures the performance gap between a given machine learning system and top-ranked Kaggle performers.
Reported results and usage
http://www.vldb.org/pvldb/vol11/p1220-liu.pdf
Reference papers
Yu Liu, Hantian Zhang, Luyuan Zeng, Wentao Wu, and Ce Zhang. "MLBench: Benchmarking Machine Learning Services Against Human Experts". In: PVLDB 11.10 (2018), pp. 1220–1232.

MLBench Distributed

Benchmark description
Benchmark Name
MLBench_Distributed ML benchmark
Short Description
This is a benchmark suite for distributed machine learning algorithms, frameworks and systems. The focus is on standard supervised ML, including standard deep learning tasks as well as classic linear ML models.
Web references
https://mlbench.readthedocs.io/en/latest/ https://github.com/mlbench
Date of last description update
2019
Originating group
--
Time – first version, last version
2018-2019
Type/Domain
Deep learning
Workload
The tasks provided by benchmark are selected to be representative of relevant machine learning workloads. *(1) Image classification workload* benchmarks two model architectures of Deep residual Networks (ResNets) *(2) Linear learning* workload benchmarks Logistic Regression with L2 regularization.
Data type and generation/datasets
Benchmark uses the CIFAR-10 dataset (containing a set of images used to train machine learning and computer vision models)and epsilon dataset (an artificial and dense dataset which is used for Pascal large scale learning challenge in 2008).
Technology stack and implementation
Current benchmark implementation supports two deep learning frameworks (PyTorch and TensorFlow), deep learning models (ResNet-20, Logistic Regression), and GPU hardware.

Metrics
The two basic metrics for comparison are: (1) *Accuracy after Time* _ accuracy of the final model after running for certain amount of time for training. The higher the better. (2) *Time to Accuracy* _ training time of a system until certain accuracy (e.g. 97%) is reached and measured. (where accuracy will be test and/or training accuracy).
Reported results and usage
https://mlbench.readthedocs.io/en/latest/benchmark-tasks.html#benchmark-task-results https://github.com/mlbench
Reference papers
--

MLPerf

Benchmark description
Benchmark Name
MLPerf
Short Description
The MLPerf effort aims to build a common set of benchmarks that enables the machine learning (ML) field to measure system performance for both training and inference from mobile devices to cloud services.
Web references
https://mlperf.org/ https://github.com/mlperf/training https://arxiv.org/pdf/1910.01500.pdf
Date of last description update
2019
Originating group
--
Time – first version, last version
2018 – 2019

Type/Domain
Benchmark Suite
Workload
MLPerf set of benchmarks tries to cover the most important areas of machine learning tasks.
Data type and generation/datasets
It aims to collect publicly available data sets and models for the following problems: Image classification, Object detection, Translation, Recommendation, Reinforcement Learning, Speech to text and Sentiment Analysis.
Technology stack and implementation
Different machine learning libraries.
Metrics
End to end time for training a model.
Reported results and usage
https://mlperf.org/results/
Reference papers
Peter Mattson, Christine Cheng, Cody Coleman, and Greg Diamos. "MLPerf: Training Benchmark"

Training Benchmark for DNNs (TBD)

Benchmark description
Benchmark Name
Training Benchmark for DNNs (TBD)
Short Description
TBD is an open source benchmark suite. It covers 6 application domains with 8 deep learning models.
Web references
http://tbd-suite.ai/ https://github.com/tbd-ai/tbd-suite
Date of last description update
2019

Originating group
University of Toronto, Microsoft Research
Time – first version, last version
2018 – 2019
Type/Domain
Benchmark Suite
Workload
Different deep learning workloads categories including Image classification, Machine translation, Object detection, Speech recognition, Adversarial learning, Reinforcement learning.
Data type and generation/datasets
Depending on the workload synthetic or real data sets.
Technology stack and implementation
Different frameworks like: Tensorflow, MXNet, and CNTK.
Metrics
Throughput, GPU utilization, F32 utilization, CPU utilization, Memory consumption.
Reported results and usage
http://tbd-suite.ai
Reference papers
Zhu, Hongyu, et al. "Tbd: Benchmarking and analyzing deep neural network training." arXiv preprint arXiv:1803.06905 (2018).

PolyBench

Benchmark description
Benchmark Name
PolyBench
Short Description
Polybench is the first benchmark for heterogeneous analytics systems, especially for polystores, providing a complete evaluation environment. Polybench is an application-level benchmark that simulates a banking business model. It focuses on banking, since it features heterogeneous analytics and data types. The

benchmark suite consists of three main use-cases and two test scenarios. The use-cases operate with structured, semi-structured, and unstructured data types and support relational, stream, array, and graph data processing paradigms. The benchmark is not tied to a specific polystore technology, rather, it is generic and high level. PolyBench provides a benchmark suite with evaluation metrics and workloads, which will eventually lead to better baselines.
Web references
https://research.spec.org/fileadmin/user_upload/documents/wg_bd/BD-20180730-PolyBench.pdf
Date of last description update
2018
Originating group
DFKI German research Center for AI, TU Berlin
Time – first version, last version
--
Type/Domain
Benchmark Suite
Workload
Different depending on the use case.
Data type and generation/datasets
The use-cases operate with structured, semi-structured, and unstructured data types and support relational, stream, array, and graph data processing paradigms.
Technology stack and implementation
Technology agnostic.
Metrics
Query runtime, data load time.
Reported results and usage
https://research.spec.org/fileadmin/user_upload/documents/wg_bd/BD-20180730-PolyBench.pdf
Reference papers
Jeyhun Karimov, Tilmann Rabl, and Volker Markl. "PolyBench: The First Benchmark for Polystores". In: Performance Evaluation and Benchmarking for the Era of Artificial Intelligence - 10th TPC Technology Conference, TPCTC 2018, Rio de Janeiro, Brazil, August 27-31, 2018, Revised Selected Papers. 2018, pp. 24–41

7.16 Year 2019

NNBench-X

Benchmark description
Benchmark Name
NNBench-X
Short Description
NNBench-X is a benchmark for understanding and evaluating Neural Network workloads for accelerator designs. The benchmark aims to facilitate hardware-software co-designs to achieve significant performance improvements and energy saving, by dividing benchmarking process into three stages: (1) application set selection, (2) benchmark suite generation, (3) hardware evaluation. NNBench-X takes as input an application candidate pool and conducts an operator-level analysis and application-level analysis to understand the performance characteristics of both basic tensor primitives and whole applications.
Web references
https://www.emc2-workshop.com/assets/docs/hpca-19/paper5.pdf
Date of last description update
2019
Originating group
--
Time – first version, last version
2019

Type/Domain
Deep learning
Workload
Application feature extraction and similarity analysis is done by k-means clustering. *Benchmark suite generation* is performed with three configurations: no compression (for GPU), quantized 16-bit fixed-point (for DianNao), and 16-bit fixed-point quantized and 90%/95% pruned (for Cambricon-X). *Hardware evaluation* is derived using an analytical model based on the Roofline model to estimate the performance of each supported tensor operator on accelerators.
Data type and generation/datasets
Datasets used by the benchmark as input is a combination of NN application candidate pool, model compression methods and set of hardware designs.
Technology stack and implementation
Hardware evaluated by NNBench-X include GPU, Neurocube, DianNao, and Cambricon-X**.**
Metrics
Metrics for hardware evaluation include speedups over CPU baseline of applications on (a) GPU without any model compression (b) Neurocube with models quantized into 16-bit fixed-point (c) DianNao with models quantized into 16-bit fixed-point, Cambricon-X (90%) with models further pruned 90% weights, and Cambricon-X (95%) with models further pruned 95% weights.
Reported results and usage
https://www.emc2-workshop.com/assets/docs/hpca-19/paper5.pdf
Reference papers
Paper: NNBench-X: A Benchmarking Methodology for Neural Network Accelerator Designs by *Xinfeng Xie, Xing Hu, Peng Gu, Shuangchen Li, Yu Ji, and Yuan Xie (University of California, Santa Barbara).* Published in: IEEE Computer Architecture Letters (Volume: 18 , Issue: 1 , Jan.-June 1 2019)

GDPRbench

Benchmark description
Benchmark Name
GDPRbench
Short Description
The General Data Protection Regulation (GDPR), introduced in Europe, was a set of rules and regulations that offered new rights and protections to people concerning their personal data. GDPRbench is an open-source benchmark designed specifically to assess the GDPR compliance of database systems which means how well a storage solution responds to common GDPR queries. The benchmark provides workloads and

metrics to understand and assess personal-data processing database systems by providing provide quantifiable measurements concerning correctness and performance of databases under GDPR .
Web references
https://www.gdprbench.org https://github.com/GDPRbench/GDPRbench
Date of last description update
2019
Originating group
--
Time – first version, last version
2019
Type/Domain
Data privacy benchmark
Workload
GDPRbench extends YCSB by creating new workloads aligned with the four core entities of GDPR: controller, customer, processor and regulator.
Data type and generation/datasets
Personal data (structured Data)
Technology stack and implementation
Redis, an in-memory NoSQL store, and PostgreSQL, a fully featured RDBMS.
Metrics
GDPRbench characterizes a database system's GDPR compliance using three metrics: (1) correctness against GDPR workloads, (2) time taken to respond to GDPR queries, and (3) storage space overhead.
Reported results and usage
https://www.gdprbench.org/research https://github.com/GDPRbench/GDPRbench
Reference papers
Understanding and Benchmarking the Impact of GDPR on Database Systems. *Supreeth Shastri, Vinay Banakar, Melissa Wasserman, Arun Kumar and Vijay Chidambaram.* In submission at VLDB 2020.

Analyzing the Impact of GDPR on Storage Systems. *Aashaka Shah, Vinay Banakar, Supreeth Shastri, Melissa Wasserman, and Vijay Chidambaram.* HotStorage 2019.

BenchIoT

Benchmark description
Benchmark Name
BenchIoT
Short Description
BenchIoT is a benchmark suite and evaluation framework for evaluating micro-controllers (IoT-µCs) security mechanisms. The applications run on either bare-metal or a real-time embedded operating system, and are evaluated through security, performance, memory usage, and energy metrics. The BenchIoT evaluation framework consists of three parts: (1) a collection of Python scripts to automate running and evaluating the benchmarks; (2) a collection of Python scripts to measure the static metrics; (3) a runtime library which we refer to hereafter as the metric collector library written in C, that is statically linked to every benchmark to measure the dynamic metrics.
Web references
https://hexhive.epfl.ch/publications/files/19DSN.pdf
Date of last description update
2019
Originating group
Purdue's HexHive and DCSL research groups
Time – first version, last version
2019
Type/Domain
Data protection/security benchmark
Workload
µVisor, Remote Attestation (RA), Data Integrity (DI) benchmarks for bare metal and OS. BenchIoT defines 5 workloads stressing one or more of these fundamental task characteristics of an IoT application: network connectivity, sense, process, and actuate.
Data type and generation/datasets
Time series, IoT

Technology stack and implementation
OS and Bare Metal.
Metrics
BenchIoT provides 14 metrics spanning four categories, namely: Security, Performance, Memory and Energy metrics. (1) *Security*: Total privileged cycles, Privileged thread cycles, SVC cycles, Maximum code region ratio, Maximum global data region ratio, Data Execution Prevention, Number of available ROP gadgets, Number of indirect calls (2) *Performance*: Total runtime cycles , Sleep cycles (3) *Memory*: Total Flash usage, Stack and heap usage, Total RAM usage (4) *Energy*: Total energy consumption.
Reported results and usage
https://github.com/embedded-sec/BenchIoT https://hexhive.epfl.ch/publications/files/19DSN.pdf
Reference papers
BenchIoT: A Security Benchmark for the Internet of Things. *Naif Saleh Almakhdhub, Abraham A. Clements, Mathias Payerk, Saurabh Bagchi*. Published in 2019 49th Annual IEEE/IFIP Internatioonal Conference on Dependable Systems and Networks (DSN)

IoTBench

Benchmark description
Benchmark Name
IoTBench
Short Description
IoTBench is a benchmark suite targeting at IoT edge-device applications, focusing on architecture performance of IoT devices.
Web references
https://ieeexplore.ieee.org/document/8802949
Date of last description update
2019
Originating group
--

Time – first version, last version
--
Type/Domain
Edge computing
Workload
This suite includes seven representative programs from three major IoT categories.
Data type and generation/datasets
--
Technology stack and implementation
IoT edge devices.
Metrics
Metrics calculated are computational demand and execution efficiency of workloads running on IoT device platform. Energy consumption of IoTBench is also measured.
Reported results and usage
--
Reference papers
IoTBench: Benchmark Suite for Intelligent Internet of Things Edge Devies. *Chien-I Lee, Meng-Yao Lin, Chia-Lin Yang, and Yen-Kuang Chen*. Published in 2019 IEEE International Conference on Image Processing (ICIP). Chien-I Lee, Meng-Yao Lin, Chia-Lin Yang, Yen-Kuang Chen:Iotbench: A Benchmark Suite for Intelligent Internet of Things Edge Devices. ICIP 2019: 170-174

VisualRoad

Benchmark description
Benchmark Name
VisualRoad
Short Description
VisualRoad is a benchmark for evaluating video data management systems (VDBMSs). The benchmark comes with a data generator and a suite of queries over cameras positioned within a simulated metropolitan environment. Visual Road's video data is automatically generated and annotated using a simulation and

visualization engine. This allows for VDBMS performance evaluation while scaling up the size of the input data. Visual Road is designed to be implementable across a wide variety of VDBMS architectures, including those that perform video querying at scale (e.g., Scanner, Optasia, Chameleon), operate on emerging forms of video data (e.g., LightDB), and perform deep learning inference (e.g., NoScope, BlazeIt, Focus). A VDBMS can execute the benchmark either offline or online. Offline processing simulates batch processing of historical video streams, where the VDBMS has random access to entire video files on persistent storage. Online processing simulates real-time video processing, where data is exposed via a forward-only iterator with unknown total duration.
Web references
https://db.cs.washington.edu/projects/visualroad/ https://db.cs.washington.edu/projects/visualroad/p300-haynes.pdf https://github.com/uwdb/visualroad
Date of last description update
2019
Originating group
--
Time – first version, last version
2019
Type/Domain
Visual analytics
Workload
Workload set is comprised of microbenchmark and composite microbenchmark queries. (1)*Microbenchmark queries* measure a VDBMS´s ability to repeatedly perform small operations over input videos. Queries include spatial and temporal selection, transformation and sub-query, interpolation and re-sampling, and union operations. (2)*Composite queries* utilize two or more microbenchmarks to implement more complex tasks. This category includes object detection, vehicle tracking, panoramic stitching and tile-based encoding.
Data type and generation/datdasets
Benchmark data is a set of video frames. These frames are periodic temporal samples of visual data. Video generator is adapted version of CARLA 0.84, an open-source simulator designed for autonomous driving research. CARLA includes resources, textures, and geometry, which form the basis of the tiles used in Visual Road. It also exposes a configuration-driven API that facilitates camera placement, rendering, and other convenience functionality.
Technology stack and implementation
Visual Road is executed on three open-source VDBMSs: Scanner, LightDB, and NoScope.

Metrics
Metrics calculated for System comparison are**:** (1) *log-scale total runtime* for each system and query combination at various scale factor and (2) *Lines of Code* required to execute benchmark query. Metric for video quality is (3) *average precision*, and for video generation time are: (4) *performance by scale/resolution* and (5) *performance by number of nodes*. (6) Performance differences between benchmark execution in write and streaming modes are calculated in percentages.
Reported results and usage
https://db.cs.washington.edu/projects/visualroad/p300-haynes.pdf https://github.com/uwdb/visualroad
Reference papers
Brandon Haynes, Amrita Mazumdar and Magdalena Balazinska, Luis Ceze, Alvin Cheung. 2019. Visual Road: A Video Data Management Benchmark. In 2019 International Conference on Management of Data (SIGMOD '19), June 30-July 5, 2019, Amsterdam, Netherlands. ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3299869.3324955.

VisualRoad

Benchmark description
Benchmark Name
VisualRoad
Short Description
VisualRoad is a benchmark for evaluating video data management systems (VDBMSs). The benchmark comes with a data generator and a suite of queries over cameras positioned within a simulated metropolitan environment. Visual Road's video data is automatically generated and annotated using a simulation and visualization engine. This allows for VDBMS performance evaluation while scaling up the size of the input data. Visual Road is designed to be implementable across a wide variety of VDBMS architectures, including those that perform video querying at scale (e.g., Scanner, Optasia, Chameleon), operate on emerging forms of video data (e.g., LightDB), and perform deep learning inference (e.g., NoScope, BlazeIt, Focus). A VDBMS can execute the benchmark either offline or online. Offline processing simulates batch processing of historical video streams, where the VDBMS has random access to entire video files on persistent storage. Online processing simulates real-time video processing, where data is exposed via a forward-only iterator with unknown total duration.
Web references
https://db.cs.washington.edu/projects/visualroad/ https://db.cs.washington.edu/projects/visualroad/p300-haynes.pdf https://github.com/uwdb/visualroad
Date of last description update

2019
Originating group
--
Time – first version, last version
2019
Type/Domain
Visual analytics
Workload
Workload set is comprised of microbenchmark and composite microbenchmark queries. (1)*Microbenchmark queries* measure a VDBMS´s ability to repeatedly perform small operations over input videos. Queries include spatial and temporal selection, transformation and sub-query, interpolation and re-sampling, and union operations. (2)*Composite queries* utilize two or more microbenchmarks to implement more complex tasks. This category includes object detection, vehicle tracking, panoramic stitching and tile-based encoding.
Data type and generation/datasets
Benchmark data is a set of video frames. These frames are periodic temporal samples of visual data. Video generator is adapted version of CARLA 0.84, an open-source simulator designed for autonomous driving research. CARLA includes resources, textures, and geometry, which form the basis of the tiles used in Visual Road. It also exposes a configuration-driven API that facilitates camera placement, rendering, and other convenience functionality.
Technology stack and implementation
Visual Road is executed on three open-source VDBMSs: Scanner, LightDB, and NoScope.
Metrics
Metrics calculated for System comparison are**:** (1) *log-scale total runtime* for each system and query combination at various scale factor and (2) *Lines of Code* required to execute benchmark query. Metric for video quality is (3) *average precision*, and for video generation time are: (4) *performance by scale/resolution* and (5) *performance by number of nodes*. (6) Performance differences between benchmark execution in write and streaming modes are calculated in percentages.
Reported results and usage
https://db.cs.washington.edu/projects/visualroad/p300-haynes.pdf https://github.com/uwdb/visualroad
Reference papers
Brandon Haynes, Amrita Mazumdar and Magdalena Balazinska, Luis Ceze, Alvin Cheung. 2019. Visual Road: A Video Data Management Benchmark. In 2019 International Conference on Management of Data (SIGMOD '19), June 30-July 5, 2019, Amsterdam, Netherlands. ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3299869.3324955.

AdaBench

Benchmark description
Benchmark Name
AdaBench
Short Description
AdaBench is an effort towards an industry standard machine learning benchmark. It aims to evaluated advanced analytics systems and cover an end-to-end ML pipeline for industry-relevant application domains.
Web references
https://ssc.io/publication/adabench-towards-an-industry-standard-benchmark-for-advanced-analytics-tpctc/
Date of last description update
2019
Originating group
--
Time – first version, last version
2019
Type/Domain
Advanced analytics, ML
Workload
--
Data type and generation/datasets
--
Technology stack and implementation
--
Metrics
--
Reported results and usage
--

Reference papers
Tilmann Rabl, Christoph Brücke-Wendorff, Philipp Härtling, Stella Stars, Rodrigo Escobar Palacios, Hamesh Patel, Satyam Srivastava, Christoph Boden, Jens Meiners, Sebastian Schelter. AdaBench - Towards an Industry Standard Benchmark for Advanced Analytics. TPC Technology Conference on Performance Evaluation & Benchmarking (TPCTC), 2019.

MiDBench

Benchmark description
Benchmark Name
MiDBench
Short Description
MiDBench is a multi-modal industrial big data benchmark. It focuses s on big data systems in crane assembly, wind turbines monitoring and simulation results management scenarios, which correspond to bills of materials (a.b.a BoM), time series and unstructured data format respectively.
Web references
https://link.springer.com/chapter/10.1007/978-3-030-32813-9_15 https://github.com/dbiir/MiDBench.
Date of last description update
2019
Originating group
--
Time – first version, last version
2019
Type/Domain
Time series and unstructured data
Workload
Benchmark provides eleven typical workloads within crane assembly, wind turbines monitoring and simulation results management application domains.
Data type and generation/datasets
Graph and time series data
Technology stack and implementation

Benchmark evaluates performance of IoTDB, MongoDB, FileDB and Elastic Search.
Metrics
--
Reported results and usage
https://github.com/dbiir/MiDBench
Reference papers
--

CBench-Dynamo

Benchmark description
Benchmark Name
CBench-Dynamo
Short Description
Dynamo-based databases are designed to run in a cluster while offering high availability and eventual consistency to clients when subject to network partition events. CBench-Dynamo is a consistency benchmark for NoSQL Database system. The benchmark correlates properties, such as performance, consistency, and availability, in different consistency configurations while subjecting the System Under Test to network partition events.
Web references
https://eg.uc.pt/bitstream/10316/87987/1/main.pdf
Date of last description update
2019
Originating group
University of Coimbra
Time – first version, last version
2019
Type/Domain
NoSQL benchmarks / Data management
Workload

Benchmark provides a customized YCSB workload, including read and write queries.
Data type and generation/datasets
Structured data
Technology stack and implementation
Benchmark framework is tested with Cassandra.
Metrics
Metrics calculated are availability, read performance and write performance.
Reported results and usage
https://eg.uc.pt/bitstream/10316/87987/1/main.pdf https://github.com/migueldiogo/cbench-analyser
Reference papers
CBench-Dynamo: A Consistency Benchmark for NoSQL Database Systems by Miguel Diogo, Miguel Diogo and Jorge Bernardino.

Edge AI Bench

Benchmark description
Benchmark Name
Edge AI Bench
Short Description
Edge AIBench is a benchmark suite for end-to-end edge computing spanning all three layers: **client-side devices, edge computing layer, and cloud servers.** Benchmark includes four typical application scenarios: ICU Patient Monitor, Surveillance Camera, Smart Home, and Autonomous Vehicle, which consider the complexity of all edge computing AI scenarios. In addition, Edge AIBench provides an end-to-end application benchmarking framework, including train, validate and inference stages.
Web references
http://www.benchcouncil.org/EdgeAIBench/index.html http://www.benchcouncil.org/EdgeAIBench/files/EdgeAIBench-Bench18.pdf
Date of last description update
2019
Originating group

BenchCouncil
Time – first version, last version
2019
Type/Domain
AI, Edge Computing
Workload
Benchmark includes train, inference, data collection and data compression/de-compression using a general three-layer edge computing frame- work. Workload includes four typical AI use cases: (1) ICU patient monitor_ heart failure prediction and endpoint prediction application (2) surveillance camera_ person re-identification (3) smart home_ speech recognition and face recognition (4) autonomous vehicle_ road sign
Data type and generation/datasets
Structured and unstructured data
Technology stack and implementation
Benchmark focuses on components at cloud server, edge computing layer, and client-side devices. Included deep learning models are: two level neural attention model, LSTM model, DeepSpeech2 model, and FaceNet model.
Metrics
--
Reported results and usage
http://125.39.136.212:8090/tanya_hao/edge-aibench
Reference papers
Edge AIBench: Towards Comprehensive End-to-end Edge Computing Benchmarking. [PDF] By Tianshu Hao, Yunyou Huang, Xu Wen, Wanling Gao, Fan Zhang, Chen Zheng, Lei Wang, Hainan Ye, Kai Hwang, Zujie Ren, and Jianfeng Zhan. *2018 BenchCouncil International Symposium on Benchmarking, Measuring and Optimizing (Bench18).*

AIBench

Benchmark description
Benchmark Name

AIBench
Short Description
AIBench is an industry standard Internet service AI benchmark suite, designed specifically for modern Internet services with microservice-based architecture. The benchmark spans sixteen AI problem domains from three most widely used Internet service domains: search engine, social network, and e-commerce. The benchmark has loosely coupled modules which can be configured and extended, including data input, AI problem domain, online inference, offline training, and deployment tool modules.
Web references
http://www.benchcouncil.org/AIBench/files/AIBench-Bench18.pdf http://www.benchcouncil.org/AIBench/index.html http://www.benchcouncil.org/testbed/index.html
Date of last description update
2019
Originating group
BenchCouncil
Time – first version, last version
2019
Type/Domain
AI
Workload
AIBench consists of 12 micro benchmarks, 16 component benchmarks and 2 end-to-end application benchmarks. (1) Microbenchmarks support various data motifs: transform, matrix, logic, sampling and basic statistics. Supported algorithms or methods are: Convolution, fully-connected, Relu, Sigmoid, Tanh, MaxPooling, AvgPooling, CosineNorm, BatchNorm, Dropout, Element-wise multiply, and Softmax (2) Component benchmarks involve complex workloads related to image classification, image generation, text-to-text translation, image-to-text, image-to-image, speech-to-text, face embedding, 3D face recognition, object detection, recommendation, video prediction, image compression, 3D object reconstruction, text summarization, spatial transformer, and learning to rank features. Supported algorithms are: ResNet50, WassersteinGAN, Transformer, Neural image caption model, CycleGAN, DeepSpeech2, Facenet, 3D face models, Faster R-CNN, Collaborative filtering, Motion-focused predictive models, Recurrent neural network, Convolutional encoder-decoder network, Sequence-to-sequence model, Spatial transformer networks and Ranking distillation. (3) Application benchmark (DCMix) is an end-to-end e-commerce search application, mimicking complex Internet services workloads. Specification is available at http://www.benchcouncil.org/AIBench/specification.html .

Data type and generation/datasets
Benchmark uses diverse datasets from various industry partners, including Cifar, ImageNet, LSUN, WMT English-German, Microsoft COCO, Cityscapes, Librisspeech, LFW, VGGFace2, samples from face IDs, MovieLens, Robot pushing dataset, ShapeNet, Gigaword, MNIST, and Gowalia. Summary of datasets is provided at http://www.benchcouncil.org/AIBench/download.html.
Technology stack and implementation
Software stack supported by the benchmark includes TensorFlow, PThreads, and PyTorch. Hardware stack includes multiple types of NIVIDIA GPUs, Inter CPUs, AI accelerator chips.
Metrics
Training metrics are the wall clock time to train the specific epochs, the wall clock time to train a model achieving a target accuracy, and the energy consumption to train a model achieving a target accuracy. *Inference metrics* are the wall clock time, accuracy, and energy consumption. *Performance metrics* are reported to measure the training and inference speeds of different hardware platforms, and to measure the performance of different software stacks.
Reported results and usage
http://www.benchcouncil.org/AIBench/number.html http://www.benchcouncil.org/AIBench/download.html
Reference papers
AIBench: An Industry Standard Internet Service AI Benchmark Suite. [PDF] by Wanling Gao, Fei Tang, Lei Wang, Jianfeng Zhan, Chunxin Lan, Chunjie Luo, Yunyou Huang, Chen Zheng, Jiahui Dai, Zheng Cao, Daoyi Zheng, Haoning Tang, Kunlin Zhan, Biao Wang, Defei Kong, Tong Wu, Minghe Yu, Chongkang Tan, Huan Li, Xinhui Tian, Yatao Li, Gang Lu, Junchao Shao, Zhenyu Wang, Xiaoyu Wang, and Hainan Ye. *Technical Report, 2019.* AIBench: Towards Scalable and Comprehensive Datacenter AI Benchmarking. [PDF] by Wanling Gao, Chunjie Luo, Lei Wang, Xingwang Xiong, Jianan Chen, Tianshu Hao, Zihan Jiang, Fanda Fan, Mengjia Du, Yunyou Huang, Fan Zhang, Xu Wen, Chen Zheng, Xiwen He, Jiahui Dai, Hainan Ye, Zheng Cao, Zhen Jia, Kent Zhan, Haoning Tang, Daoyi Zheng, Biwei Xie, Wei Li, Xiaoyu Wang, and Jianfeng Zhan. *2018 BenchCouncil International Symposium on Benchmarking, Measuring and Optimizing (Bench18)*

HPC A1500

Benchmark description
Benchmark Name
HPC A1500
Short Description
HPC A1500 is a benchmark suite for evaluating HPC systems that run specific Deep Learning workloads. HPC1500 workloads are based on real scientific DL applications and cover representative scientific fields,

namely climate analysis, cosmology, high energy physics, gravitational wave physics, and computational biology.
Web references
http://www.benchcouncil.org/HPCAI500/index.html http://www.benchcouncil.org/HPCAI500/specification.html http://www.benchcouncil.org/bench19/file/slides/HPCAI500_tutorial.pdf
Date of last description update
2019
Originating group
BenchCouncil
Time – first version, last version
2019
Type/Domain
High Performance Computing (HPC), AI, Deep Learning
Workload
HPC1500 contains micro- and component benchmarks. (1) *Microbenchmarks* support Convolution, Fully connected, and Pooling operators in CNN. (2) *Component benchmarks* involve complex workloads related to image recognition (identifying particle signal), image generation (generating cosmological images), object detection (extreme weather detection) and sequence prediction (predicting spectrum of peptides). Models supported are: ResNet-50, DCGAN, Faster-RCNN, BiLSTM, Convolution pooling and Fully connected.
Data type and generation/datasets
Benchmark uses diverse datasets from various industry use cases. Datasets used by component benchmarks include HEP (High Energy Physics) dataset, Cos (Cosmology) dataset, Extreme Weather Dataset and pDeep dataset. Main data schema is matrix, with 2D dense, 2D sparse and 3D matrix data formats.
Technology stack and implementation
The benchmark suite supports MKL, CUDNN, TensorFlow and PyTorch. Currently, Extreme Weather Analysis workload, with TensorFlow-GPU version, and microbenchmarks, of MKL and CUDA version, are open sourced.
Metrics
The adopted metric for component benchmarks is *time-to-accuracy,* which means the total training time needed to a target validation accuracy, where accuracy is measured by mAP (mean Average Precision). Metrics for microbenchmarks are *FLOPS (Floating-point Operations Per Second)* and *images per second*.

Reported results and usage
http://125.39.136.212:8090/hpc-ai500/EWA
Reference papers
HPC AI500: a benchmark suite for HPC AI systems by *Zihan Jiang, Wanling Gao, Lei Wang, Xingwang Xiong, Yuchen Zhang, Xu Wen, Chunjie Luo, Hainan Ye, Yunquan Zhang, Shengzhong Feng, Kenli Li, Weijia Xu, Jianfeng Zhan.*

SparkAIBench

Benchmark description
Benchmark Name
SparkAIBench
Short Description
SparkAIBench is a benchmark to generate AI workloads on Apache Spark, supporting a variety of algorithms, configurable data input size, as well as parametric method for submission.
Web references
http://www.benchcouncil.com/bench19/file/slides/paper17.pdf https://github.com/harryandlina/SparkAIBench
Date of last description update
2019
Originating group
Beijing institute of technology
Time – first version, last version
2019
Type/Domain
AI
Workload
The benchmark provides workloads from different categories, including regression analysis, text mining, classification, frequent item-set mining, clustering, recommendation, image classification, and natural language processing.

Workloads support various algorithms: Linear Regression, LDA, Bayes, SVM, FP-Growth, k-means, ALS, LeNet, Inception, VGG ResNet, RNN and Auto-enoder.
Data type and generation/datasets
SparkMLlib uses SELF, LibSVM, BDGS, and MINST data. BigDL uses MINST, ImageNet, CIFAR-10 and text data.
Technology stack and implementation
Supported deep learning libraries are SparkMLlib and BigDL.
Metrics
The metric calculated is average job latency.
Reported results and usage
--
Reference papers
SparkAIBench: A Benchmark to Generate AI Workloads on Spark by *Zifeng Liu, Xiaojiang Zuo, Zeqing Li and Rui Han* (Beijing institute of technology)

AIMatrix

Benchmark description
Benchmark Name
SparkAIBench
Short Description
AI Matrix is a benchmark suite for testing AI software frameworks and hardware platforms.
Web references
https://aimatrix.ai/en-us/index.html https://aimatrix.ai/en-us/docs/contents.html
Date of last description update
2019
Originating group
AliBaba
Time – first version, last version

2019
Type/Domain
AI
Workload
The AI Matrix suite consists of four categories of tests: micro benchmarks, layer-based benchmarks, macro benchmarks, and synthetic benchmarks (StatsNet); **(1)** Microbenchmarks_ basic hardware level GEMM computation **(2)** Layered-based benchmarks_ evaluating basic elements of Neural Network **(3)** Macro-benchmarks_ evaluating complete models **(4)** Synthetic benchmarks_ matching statistical characteristics of input models. Macro-benchmarks cover major application categories: image classification, object detection, neural machine translation, deep speech, and deep interest network.
Data type and generation/datasets
Unstructured data.
Technology stack and implementation
Supported models are CNN and RNN and supported frameworks include TensorFlow and Caffe.
Metrics
Basic metric measured is *wall clock*.
Reported results and usage
--
Reference papers
--