DataBench

Evidence Based Big Data Benchmarking to Improve Business Performance

D1.4 Horizontal Benchmarks – Data Management

Abstract

Considering that horizontal benchmarking is focusing on a specific process or activity, this document - deliverable D1.4- Horizontal Benchmarks – Data Management (M24) will present 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 including data acquisition and curation and data storage for various classes of storage systems. The deliverable D1.4 is focusing on classical big data benchmarks related to data management covering both for data acquisition and curation, and for data storage. It will be examined a number of existing database benchmarks for various types of SQL and NoSQL storage types and file systems as part of this classical area of database benchmarking. Different indexing and retrieval schemes will be benchmarked for the various big data types. The historically successful benchmarks such as the TPC-series of benchmarks with BigBench, BigDataBench and many others will be analyzed. 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.

This document will classify the benchmarks to four categories:

- 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

This "D1.4 Horizontal Benchmarks – Data Management" document is relating to the public version of the document "D1.2 DataBench Framework – with Vertical Big Data Type benchmarks" which have been provided at the same time as this document. It is also complementary to the " D1.3 Horizontal Benchmarks – Analytics and Processing" document.

Deliverable D1.4	DataBench Horizontal Benchmarks – Data Management
Work package	WP1
Task	1.4
Due date	31/12/2019
Submission date	17/01/2020
Deliverable lead	ATOS
Version	1.1
Dissemination level	Public
Authors	ATOS (Tomás Pariente, Iván Martínez and Ricardo Ruiz) SINTEF (Arne Berre, Bushra Nazir) GUF (Todor Ivanov, Timo Eichhorn) JSI (Marko Grobelnik)
Reviewers	Gabriella Cattaneo, David Wells

Keywords

Benchmarking, big data, big data technologies, BDVA Reference Model, Horizontal benchmarks, architecture, performance metrics, data management.

Disclaimer

Copyright Notice

Table of Contents

Table of Figures

Table of Tables

Executive Summary

The goal of this document - deliverable D1.4 - Horizontal Benchmarks – Data Management (M24) is to classify and give details about various horizontal benchmarks in the areas of data management (including data acquisition and curation and data storage for various classes of storage systems), Data Protection (benchmarks about privacy and security in relation to data management), and Cloud/HPC/Edge and IoT Data Management Benchmarks.

The area of data management and storage is one of the classical layers of big data benchmarks. It covers benchmarks for data acquisition and curation and for data storage and includes many of existing database benchmarks for various types of SQL and NoSQL storage types and file systems. Some of the historical successful benchmarks such as the TPC-series of benchmarks with BigBench and BigDataBench and many others are part of this layers, as well as the Linked Data/Graph database benchmarks that focus on the performance of Graph databases and RDF storage systems. The suite of horizontal benchmarks adapted for this will be representative of all relevant data management solutions relevant for the industrial requirements.

This document refers to the public version of deliverable D1.2, which provides an introduction to the objectives of the Work Package 1 and an extensive catalog of most of the existing benchmarking initiatives and tools. All benchmarks collected in the annexes of D1.2 have been therefore referenced from this document.

1. Introduction and Objectives

The DataBench Framework is based on a combination of both the vertical and horizontal dimensions of the BDVA Reference Model [3], which includes several horizontal layers out of which the bottom four while the other remaining horizontal layers (data visualization/interaction, analytics and processing) are considered in the parallel deliverable D1.3 [2] sharing the structure and the approach.

This document is therefore viewing and cataloguing benchmarks from the horizontal point of view, focusing on the horizontal layers related to data management. These areas include benchmarks for data acquisition and curation and data storage for various classes of storage systems, for data protection, privacy and security, and Cloud/HPC/Edge/IoT data management.

These areas cover some of the classical big data benchmarks for data acquisition and curation and for data storage and include many of existing database benchmarks for various types of SQL and NoSQL storage types and file systems. Some of the historical successful benchmarks such as the TPC-series of benchmarks with BigBench and BigDataBench and many others are part of this layers, as well as the Linked Data/Graph database benchmarks that focus on the performance of Graph databases and RDF storage systems. The suite of horizontal benchmarks adapted for this will be representative of all relevant data management solutions relevant for the industrial requirements.

This document refers to the public version of deliverable D1.2 [1], which provides an introduction to the objectives of the work package 1 and an extensive catalog of most of the existing benchmarking initiatives and tools. All benchmarks collected in the annexes of D1.2 have been therefore referenced from this document.

The document is structured as follow:

- Section 1 provides the introduction to the deliverable.
- Section 2 summarizes the DataBench Framework and positions the layers that are covered in the document.
- Section 3 describes technical benchmarks as mapped into horizontal benchmark groups, following the horizontal bottom layers of the BDVA Reference Model.
- Section 4 provides the conclusions of the document.

2. DataBench Framework

The DataBench Framework is further described in the DataBench D1.2 document - public version [1].

In this section we categorize the horizontal benchmarks according to different horizontal layers specified in the BDVA reference model (Figure 1). These layers cover specific aspects ranging from infrastructure (benchmarks related to the use of underlying HPC, Cloud, Edge and IoT infrastructures), to aspects of the data value chain (covering aspects related to data management, data processing, analytics and visualization, as well as issues related to data protection). Deliverable D1.3 [2] is covering benchmarks located in the upper horizontal layers, while this document is focusing in the four horizontal bottom layers shown in Figure 1:

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

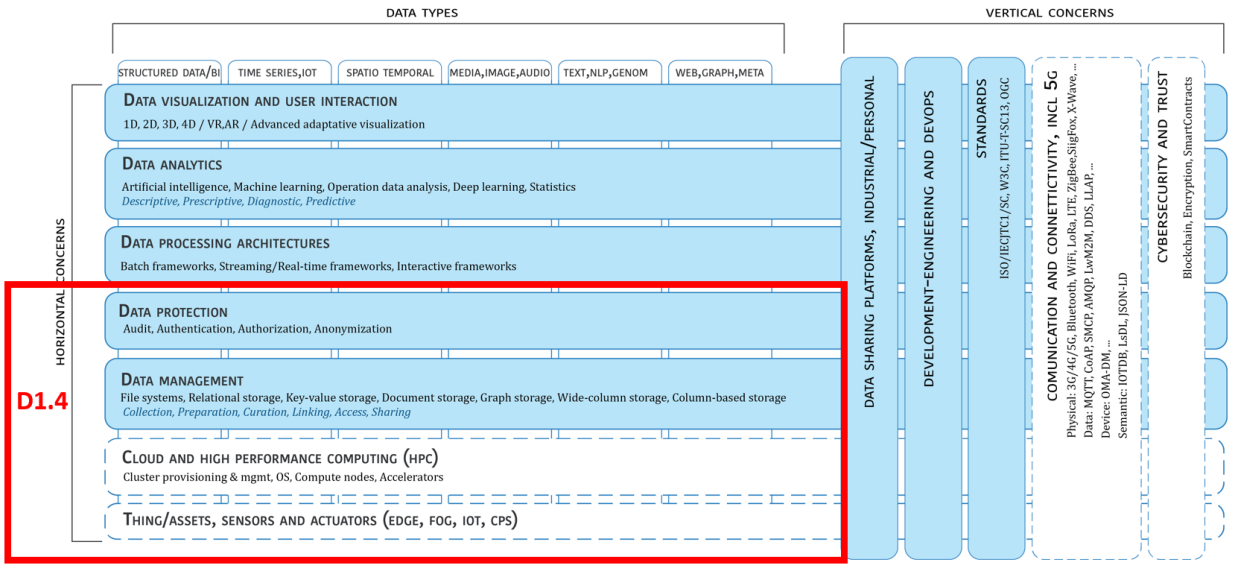

Figure 1 DataBench Framework and the layers covered in the document

However, we only focus on the most appropriate and relevant benchmarks that meet a set of criteria described below:

- First criterion: they must be publicly available in one of these two formats: source code and / or execution binary.
- Second criterion: they must be updated periodically in terms of error correction, usability improvements and new functional extensions.
- Third criterion: user documentation, installation and use guides must be available that accurately describe how to apply and run the benchmark so that these processes are facilitated to the end user.
- Fourth criterion: 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.

It is worth mentioning that this document does not contain extensive descriptions of the benchmarks, as most of them have already been described in detail in the annexes of deliverable D1.2. Therefore, **this document provides pointers to those description in the public version of D1.2** and complements them with extra information.

3. Horizontal Benchmarks for data management

3.1. Data Protection

This section presents benchmarks related to privacy and anonymisation mechanisms to facilitate data protection, including benchmarks related to blockchain, data encryption, or cybersecurity related to data management.

Deliverable D1.2 identified aspects of data protection related to the ISO SC42, in particular **Audit** (audit trails and logs to track provenance of data, for data/state recovery or forensic analysis of a system crash or incursion), **Authentication** (access control to data and services), **Authorization** (managing privileges to access to data or data services) and **Anonymization** (data obfuscating to avoid reidentification of personal or sensitive data) Frameworks.

Table 1 summarizes the benchmarks stressing different data protection, privacy and anonymization related data processing and storage system feature.

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
TERMinator Suite	Computer architectures based on homomorphic operations.		Performance over different security configurations of privacy-preserving versions of fourteen algorithms from four benchmark classes (namely synthetic, microbenchmarks, kernels and encoder benchmarks).	Cryptoleq Enhanced Assembly Language (CEAL) [13]	Structured, BI	https://github.com/momalab/TERMinatorSuite
	Described in D1.2. Section 7.15 [1]					
GDPRbench	GDPR compliance of database systems.	GDPRbench defines workloads aligned with the four core entities of GDPR: controller, customer, processor and regulator	GDPRbench characterizes a database system's GDPR compliance using three metrics: correctness against GDPR workloads, time taken to respond to GDPR queries, and storage space overhead.	Redis [10], an in-memory NoSQL store, and PostgreSQL [9], a fully featured RDBMS.	Personal data (structured data).	https://www.gdprbench.org/
	Described in D1.2. Section 7.16 [1]					
BenchIoT	Micro-controllers (IoT-µCs) security.		Security, Performance, Memory and Energy metrics.	BenchIoT evaluation framework [14]	Time Series, IoT	https://github.com/embedded-sec/BenchIoT
	Described in D1.2. Section 7.16 [1]					
AIBench	Sixteen prominent AI problem domains, including classification, image generation, text-to-text translation, image-to-text,	Workloads from Internet services	Training time, Training cost, Question answering Inference latency and Inference cost.	Java Application framework, Maven.	Real-world data sets from Internet services	http://www.aibench.org/

	image-to- image, speech-to-text, face embedding, 3D face recognition, object detection, video prediction, image compression, recommendation, 3D object reconstruction, text summarization, spatial transformer, and learning to rank.					
	Described in D1.2. Section 7.16 [1]					

Table 1 Summary of Data Protection benchmarks

3.2. Data Management and Storage

This section includes benchmarks related to pure data management aspects, such as data life cycle management, storage (i.e. NoSQL, SQL, data lakes, data spaces, etc.).

This is a traditional area for benchmarks, especially in relation with structured data in tables and databases, including the majority SQL databases (i.e. MySQL, Oracle, etc.). Popular and long-lasting benchmarking initiatives in this area include the database benchmarks of the Transaction Processing Performance Council (TPC-H, TPC-C and TPC-DS). More recently subsets of these benchmarks are more focused in big data, which include for instance TPCx-HS and TPCx-BB.

Deliverable D1.2 describes different aspects of this layer according to the several layers of the ISO SC42 Big Data Reference Model. In particular this layer comprises a set of technologies that can be coupled to storage data management (i.e. file system, SQL, NoSQL - Key-value, Wide-column, column-based, document, and graph-), data collection (i.e. data acquisition, ETL systems, data aggregation, data fusion, data virtualization, etc.), and data lifecycle management (i.e. metadata management, data quality, data cleaning, data curation).

Table 2Table 2 summarizes the data management and storage related benchmarks, which are targeting one of the most dynamically changing horizontal layers with a great number of emerging new technologies.

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
TPC-H	Data warehouse capability of a system.	22 business queries designed to exercise system functionalities in a manner representative of complex decision support applications [15]	TPC-H Composite Query-per-Hour metric, Price-performance metric, and Availability Date of the system.	C++ Dataset generator, SQL Engine	Structured data, generates data from a sample file	http://www.tpc.org/tpch
	Described in D1.2. Section 7.1 [1]					
TPC-DS v2	Decision Support Systems	99 distinct SQL-99 (with OLAP amendment) queries and twelve data maintenance operations. [16]	Query response time in single user mode, query throughput in multi-user mode and data maintenance performance for a given hardware.	SQL Databases	Synthetic data set.	http://www.tpc.org/tpcds/default.asp
	Described in D1.2. Section 7.2 [1]					
Yahoo! Cloud Serving Benchmark (YCSB)	Cloud serving systems	6 pre-defined workloads, which simulate a cloud OLTP application (read and update operations). and workload generator [17]	Execution time, Latency if request under load, Throughput (operations per second),	Support various NoSQL and relational database systems (i.e. Apache HBase, Cassandra, Riak, MongoDB, etc.)	The benchmark consists of a workload generator and a generic database interface, which can be easily extended to support other relational or NoSQL databases.	https://github.com/brianfrankcooper/YCSB
	Described in D1.2. Section 7.7 [1]					
Hadoop Workload Examples	Hadoop workloads	Different micro benchmarks like WordCount, Grep, Pi and Terasort [18]	Execution time	Java / MapReduce	Synthetic data generation	https://wiki.apache.org/hadoop/Grep

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
	Described in D1.2. Section 7.3 [1]					
GridMix	Cluster resources	Mix of traced synthetic jobs and basic operations [18]	Execution time, Memory, Throughput	MapReduce jobs	Synthetic data generation	https://hadoop.apache.org/docs/stable1/gridmix.html
	Described in D1.2. Section 7.4 [1]					
PigMix	Pig Systems	Different queries testing like data loading, different types of joins, group by clauses, sort clauses, as well as aggregation operations.	Execution time.	Pig Latin, Hadoop	Synthetic and structured data	https://cwiki.apache.org/confluence/display/pig/PigMix
	Described in D1.2. Section 7.5 [1]					
MRBench	Map and reduce operations.	22 business queries designed to exercise system functionalities [19]	Execution time.	C++ , Hadoop MapReduce	Structured data, generates data from a sample file.	https://markobigdata.com/2016/07/13/hadoop-benchmark-test-mrbench/
	Described in D1.2. Section 7.5 [1]					
CALDA	Relational Database Management Systems parallelization.	5 SQL queries among Map Reduce grep task [20] [20]	Execution time.	Hadoop, MapReduce.	Synthetic structured data.	
	Described in D1.2. Section 7.6 [1]					

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
HiBench	Micro benchmarks, Web search, Machine Learning, and HDFS benchmarks [21]	Micro-benchmark suite including 6 categories which are micro, ML (machine learning), SQL, graph, websearch and streaming.	Execution time (latency), throughput and system resource utilizations (CPU, Memory, etc.).	Hadoop: Apache Hadoop 2.x, CDH5, HDP; Spark: Spark 1.6.x, Spark 2.0.x, Spark 2.1.x, Spark 2.2.x; Flink: 1.0.3; Storm: 1.0.1; Gearpump: 0.8.1; and Kafka: 0.8.2.2.	Synthetic data generated from real data samples	https://github.com/Intel-bigdata/HiBench
	Described in D1.2. Section 7.6 [1]					
PUMA Benchmark Suite	Hadoop micro-benchmarks [22]	MapReduce workloads	Execution time, MapReduce statistics	Hadoop MapReduce	Predefined datasets	https://engineering.purdue.edu/~puma/pumabenchmarks.htm
	Described in D1.2. Section 7.9 [1]					
MRBS	MapReduce systems	Two execution modes are supported: interactive mode and batch mode [23]	Client request latency, throughput and cost	Hadoop MapReduce	Real-world data samples.	http://sardes.inrialpes.fr/research/mrbs/index.html
	Described in D1.2. Section 7.9 [1]					
BigBench v2	Big Data platform [24]	The business model and a large portion of the data model's structured part is derived from the TPC-DS benchmark.	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	Hadoop Ecosystem	Synthetic un-, semi-, and structured data.	http://www.tpc.org/tpcx-bb/results/tpcxbb_perf_results.asp

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
			(3) System Availability Date as defined by the TPC Pricing Specification.			
	Described in D1.2. Section 7.10 [1]					
BigDataBench	Big Data benchmark suite [25]	Seven workload types including AI, online services, offline analytics, graph analytics, data warehouse, NoSQL, and streaming	Wall clock time and energy efficiency.	Hadoop, Spark, Flink and MPI implementations	Real world data.	http://www.benchcouncil.org/BigDataBench/
	Described in D1.2. Section 7.10 [1]					
LinkBench	Social graphs	Set of standard insert, update, and delete operations to modify data, along with variations on key lookup, range, and count queries [25]	Latency of requests.	MySQL, MongoDB	Synthetic social graph with key properties similar to the real graph.	
	Described in D1.2. Section 7.10 [1]					
BigFrame	Big Data analytics [26]	Offline-analytics and Real-time analytics.	Execution time.	Java and Hadoop	Structured semi-structured synthetic data adapted from TPC-DS.	https://github.com/bigframeteam/BigFrame/wiki
	Described in D1.2. Section 7.10 [1]					
PRIMEBALL	Parallel processing frameworks in the context of Big Data	Various use-case scenarios made of both queries and	Throughput and price performance.	Technology agnostic.	Structured XML and binary audio and video files.	https://hal.archives-ouvertes.fr/hal-00921822/document

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
	applications hosted in the cloud [27]	data-intensive batch processing.				
	Described in D1.2. Section 7.10 [1]					
Semantic Publishing Benchmark (SPB)	RDF database engines inspired by the Media/Publishing industry [28]	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	Execution time	Graph DB, Apache Ant	Synthetic RDF data	http://ldbcouncil.org/developer/spb
	Described in D1.2. Section 7.11 [1]					
Social Network Benchmark	Data generators [28]	Interactive, Business Intelligence and Graph Analytics.	Operations/ minute	GraphDB	Synthetic social network.	http://ldbcouncil.org/benchmarks/snb
	Described in D1.2. Section 7.11 [1]					
TPCx-HS v2	Hadoop MapReduce operations [30]	HSGen, HSDataCkeck, HSSort, and HSValidate.	The benchmark reports the total elapsed time (T) in seconds for both runs. This time is used for the	Hadoop MapReduce	The scale factor defines the size of the dataset, which is generated by HSGen and used for the	http://www.tpc.org/tpcx-hs/

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
			calculation of the TPCx-HS performance metric also abbreviated with HSph@SF		benchmark experiments. The data is synthetically generated.	
	Described in D1.2. Section 7.11 [1]					
SparkBench	Spark system design and performance optimization and cluster provisioning [31]	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)	(1) Job Execution Time(s) of each workload; (2) Data Process Rate (MB/seconds); and (3) Shuffle Data Size	Apache Spark >= 2.1.1	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.	https://github.com/CODAIT/spark-bench
	Described in D1.2. Section 7.12 [1]					
TPCx-V	Server running virtualized databases [32]	OLTP / DSS workloads.	The Performance Metric *istpsV* is a "business throughput" measure of the number of completed Trade-Result transactions per second.	VMs, relational DBs.	OLTP and OLAP, structured data	http://www.tpc.org/tpcx-v/default.asp
	Described in D1.2. Section 7.14 [1]					
BigFUN	Micro-operations [33]	Simple retrieves, range scans, aggregations, joins, as well as inserts and updates.	Execution time.	AsterixDB, MongoDB and Hive.	Synthetic JSON data.	https://github.com/pouriapirz/bigFUN

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
	Described in D1.2. Section 7.12 [1]					
TPCx-BB	Analytical capabilities of a Big Data platform [34][34]	The business model and a large portion of the data model's structured part is derived from the TPC-DS benchmark	(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	Technology agnostic	Synthetic data generator for structured, semi-structured and unstructured data.	http://www.tpc.org/tpcx-bb/
	Described in D1.2. Section 7.13 [1]					
Graphalytics	Graph analysis platforms [35]	Six core algorithms: breadth-first search, PageRank, weakly connected components, community detection using label propagation, local clustering coefficient, and single-source shortest paths.	Execution time.	Graph analysis platforms (Giraph, GraphX, OpenG, PowerGraph, GraphMat, Gelly, GraphBLAS, Gunrock, mvGRAPH).	Synthetic data for graph queries	https://graphalytics.org
	Described in D1.2. Section 7.12 [1]					
AdBench	Data pipelines [36]	Streaming Analytics on Ad-serving logs, streaming ingestion and updates of various data	Throughput, Query concurrency, Execution time for batch computation & Ad-Hoc queries, End-to-end latency, Operational	Apex, Trafodion, HDFS, ampool	Synthetic data of relational and streaming models.	

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
		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.	complexity, Cost to meet SLAs			
	Described in D1.2. Section 7.13 [1]					
GARDENIA	Big Data applications running on modern datacenter accelerators [37]	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).	Execution time, IPC (Instructions per cycle)	OpenMP, CUDA	Datasets from the UF Sparse Matrix Collection, the SNAP datasetCollection, and the Koblenz Network Collection.	https://github.com/chenxuhao/gardenia

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
	Described in D1.2. Section 7.14 [1]					
gmark	Schema-driven generation of graphs and queries [38]	Unions of Conjunctions of Regular Path Queries (UCRPQ). UCRPQ contains recursive path queries for applications like social networks, bio-informatics, etc.	(a) Query execution times for diverse graph sizes and query workloads, and (b) Query execution times for simple recursive queries on various small graph.	Shell, GraphDB	Synthetic data graph data.	https://github.com/graphMark/gmark
	Described in D1.2. Section 7.11 [1]					
TPCx-IoT	IoT gateway systems.	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	(1) IoTps as the performance metric; (2) $/IoTps as the price-performance metric; and (3) system availability date	HBase 1.2.1 and Couchbase-Server 5.0 NoSQL databases	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.	http://www.tpc.org/tpcx-iot/
	Described in D1.4. Section 7.12 [1]					
Senska	Enterprise streaming benchmark [39] [39]	The data feeder, system under test (SUT) and the result validator	No metrics info.	Apache Kafka Streaming application, toolkit is using a JVM language.	Senska takes as input data a csv-file which should contain representative data for a manufacturing context.	
	Described in D1.2. Section 7.14 [1]					

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
ABench	Data processing and analytics apps [40][40]	The workloads could be business problem dependent if only specifications are given or could be in form of SQL queries which may access data across different data models.	Execution time, scalability, reliability, throughput, energy efficiency and cost of the solution.	Message platform (e.g. Kafka), streaming engine (e.g. Spark, Storm, Flink), in-memory store (e.g. MemSQL, MongoDB) and persistent data store (HDFS, HBase)	The framework needs to have different types of data generators for stream messages, structured, graph, unstructured, documents et	
	Described in D1.2. Section 7.15 [1]					

Table 2 Summary of Data Management and storage benchmarks

It is important to note that the following benchmarks have also functionalities belonging to the Data Management and Storage layer but are also classified in other horizontals where they present more functionalities. Therefore, these benchmarks are not listed in Table 2, but in the tables corresponding to their main coverage. The benchmarks are the following: Hobbit, Sanzu, AIM, RIotBench, CloudRank-D, CloudSuite and AMP Lab Big Data Benchmark.

3.3. Cloud/HPC Data Management Benchmarks

This subsection covers benchmarks for the second layer from the bottom of the BDVA Reference Model related to data management. Benchmarks for Cloud and HPC infrastructures. As pointed out in the BDVA SRIA v4.0 [3]: "Effective Big Data processing and data management might imply the 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".

This is therefore a very important aspect to take into account while selecting the use of big data solutions in the scope of these two different infrastructures. Traditionally big data solutions have been developed for data management in cloud environments. Most of the big data processing engines, frameworks and tools are therefore fine-tuned for cloud environments. However, the use of HPC is increasingly appealing for environments where the use of HPC could become a clear advantage. To this extent, the ETP4HPC has released a document highlighting what the big data computing stack and HPC stack can learn from each other [4]. The conclusion of this study is that big data system can profit from HPC on several aspects such as the fast communication between nodes, more efficient algorithms in HPC for linear algebra, introduction of new hardware (such as FPGA for Deep Learning processors), etc. However, several issues remain to make use in HPC environments of existing big data stacks, stream processing or how many big data applications may fit in the current resource management policies of HPC centres.

Table 3 lists benchmarks that are specifically designed and targeting Cloud and HPC related features and workloads.

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
Liquid	Concepts, models, metrics, and tools for an efficient, effective and sustainable way of creating, disseminating, evaluating, and consuming scientific knowledge [41]	XML Compressors, SPARQL query processors and Graph query processors.	Gain of performance.	Cloud Computing, Software As A Service (SAAS) and Collaborative and Social Web.	Structured data.	https://github.com/Shopify/liquid/blob/master/performance/benchmark.rb
	Described in D1.2. Section 7.7 [1]					
ALOJA	Big Data frameworks [42]	MPI-based profiling workloads, cluster configuration workloads, Machine Learning and predictive analytic workloads.	Main metrics include execution time, cost/performance efficiency, Job-execution time for not-benchmarked configurations.	Vagrant, VirtualBox, Bash-Scripts, Hadoop Ecosystem.	Synthetic social network.	https://github.com/Aloja/aloja-mlb
	Described in D1.2. Section 7.11 [1]					
Hobbit	Linked Data [43]	Real-world application workloads.	Depending on the executed benchmark	Java	Linked data.	https://project-hobbit.eu/
	Described in D1.2. Section 7.13 [1]					
CloudRank-D	Private cloud systems [44]	Scalable applications and input datasets, Tunable submission patterns and Configurable runtime systems.	Performance of Software & Hardware	Hadoop framework	Data models and Data semantics.	
	Described in D1.2. Section 7.8 [1]					
CloudSuite	Analyse and identify key inefficiencies in the processor's core micro-	Scale-out workloads and traditional benchmarks	Micro-architectural	Docker container	Real-world data samples	https://cloudsuite.ch/

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
	architecture and memory system organization [45] [45]		behavior of scale-out			
	Described in D1.2. Section 7.9 [1]					
AMP lab big data Benchmark	SQL-on-Hadoop engines [46]	Four queries involving scans, aggregations, joins, and UDFs.	Execution time.	RedShift, Hive, Stinger/Tez, Shark, and Impala, HDFS.	Synthetic structured data	https://amplab.cs.berkeley.edu/benchmark/
	Described in D1.2. Section 7.10 [1]					
PRIMEBALL	Parallel processing frameworks in the context of Big Data applications hosted in the cloud [47]	Various use-case scenarios made of both queries and data-intensive batch processing.	Throughput and price performance.	Technology agnostic.	Parallel processing frameworks in the context of Big Data applications hosted in the cloud.	https://hal.archives-ouvertes.fr/hal-00921822/document
	Described in D1.2. Section 7.10 [1]					
TPCx-V	Server running virtualized databases	OLTP / DSS workloads.	The Performance Metric istpsV is a "business throughput" measure of the number of completed Trade-Result transactions per second.	VMs, relational DBs.	OLTP and OLAP, structured data	http://www.tpc.org/tpcx-v/default.asp
	Described in D1.2. Section 7.14 [1]					
HPC AI500	Deep Learning algorithms to solve many important problems, such as extreme	Convolution Pooling Fully-Connected, ResNet, Faster-RCNN and DCGAN.	HPC AI500 metrics for component	CUDA MKL, TensorFlow, and Pytorch	Scientific data: Matrix, HEP Dataset Cos	http://www.benchcouncil.org/HPCAI500/

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
	weather analysis, high energy physics, and cosmology [48]		benchmarks include both accuracy and performance. For micro benchmarks, HPC AI500 provides metrics such as FLOPS to reflect the upper bound performance of the system.		Dataset, EWA Dataset and Cos Dataset.	
	Described in D1.2. Section 7.16 [1]					

Table 3 Summary of Cloud and HPC benchmarks

As in the previous case, some of the benchmarks have also functionalities belonging to the Cloud and HPC layer have not been listed in Table 3, as they are classified in other horizontals where they fit better in terms of coverage. These are the following benchmarks: Hobbit, Hermit, ABench and Edge AIBench.

3.4. IoT, Edge, Fog Data Management Benchmarks

This subsection covers benchmarks for the bottom layer of the BDVA Reference Model related to data management. Benchmarks for IoT, Edge and Fog are relatively new. The use of IoT devices creates the requirement of having real-time streaming engines that need to handle and analyse the data either on the fly or in specific storage modules, and come with the use of timestamped data (time series approaches). From the data management perspective many of them are therefore covering aspects related to distributed data storage, data processing and data analytics, and time series.

As a consequence, it is important to point out that many of the benchmarks used for this purpose have some overlapping with the ones used for Cloud and HPC covered in Section 3.3, but mostly with the ones covered by the Data Processing layer in deliverable D1.3. Therefore, in this section we will mention briefly benchmarks already covered in other parts of this document or in other documents, and we will go in more details with the benchmarks not covered so far. Table 4 lists these benchmarks.

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
Yahoo Streaming Benchmark (YSB)	Latency that a particular processing system can produce at a given input load. Execution Performance, Volume, Velocity, Fault-tolerance [49]	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.	Latency, Average throughput.	Apache Kafka, Redis and three computation engines (Flink, Storm and Spark Streaming)	Structured, Time Series	https://github.com/yahoo/streaming-benchmarks
	Described in D1.2. Section 7.12 [1]					
SparkBench	Spark deployments in different systems [50][50]	four categories: ML, Graph Computation, SQL Query, Streaming Application.	Job execution time in seconds Data process rate in MB/Second.	Spark	Structured data.	https://codait.github.io/spark-bench/
	Described in D1.2. Section 7.12 [1]					
IoTAbench		The workload can be mainly categorized as loading, repairing and analyzing tasks [51]	Performance in terms of Query Execution Times (in seconds or milliseconds)	HP Vertica 7 Analytics platform	Real data.	https://git.fortiss.org/pmwt/PIOT-Benchmark/Benchmark/wikis/iotabench
	Described in D1.2. Section 7.12 [1]					
RioTBench	27 common IoT tasks implemented as reusable micro-benchmarks [52]	4 IoT app. benchmarks composed from these tasks, plus 4 stream workloads from real IoT observations on smart cities and fitness (peaks from 500 – 10000 messages/sec and	Evaluate performance of DSPS for streaming IoT applications	Apache Storm DSPS on the Microsoft Azure public Cloud,	IoT data streams.	https://github.com/dream-lab/riot-bench

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
		diverse frequency distributions)				
	Described in D1.2. Section 7.13 [1]					
StreamBench	Stream computing [29]	The benchmark consists of four different workload suites: Performance, Multi-recipient performance, Fault tolerance and Durability workloads.	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.	The benchmark suite is implemented and evaluated with the Apache Storm and Apache Spark Streaming frameworks. Apache Kafka is used as a messaging system.	The benchmark suite uses different data scale sizes generated from two datasets. The AOL Search Data and CAIDA Anonymized Internet Traces Dataset.	
	Described in D1.2. Section 7.11 [1]					
LinearRoad	Streaming data management systems (SDMS) [53]	Continues and historical queries generator.	Performance.	Python, Java, Aerospike	Microscopic traffic simulator dataset.	https://github.com/walmartlabs/linearroad
	Described in D1.2. Section 7.3 [1]					
CityBench	Tool to scan the potentials of European cities [54]	A set of continuous queries covering a variety of data- and application-dependent characteristics.	RDF stream processing query performance.	City Bench framework, JVM 1.7, Webserver (JBoss,Tomcat etc.) and Java IDE	Smart City Datasets	https://github.com/CityBench/Benchmark

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
	Described in D1.2. Section 7.12 [1]					
AdBench	Tools for automatic differentiation (also called algorithmic differentiation is a set of techniques to numerically evaluate the derivative of a function specified by a computer program) [55]	Operator overloading (OO) or source transformation (ST)	Absolute runtimes as well as runtimes normalized with respect to individual languages.	Ceres , Autograd and Theano.		https://github.com/awf/ADBench
	Described in D1.2. Section 7.13 [1]					
Sanzu	5 popular data science platforms [55]	It contains a micro benchmark and a macro benchmark	Execution time	Anaconda Python, R, Dask, PostgreSQL with MADLib and Spark	Datasets are generated from a synthetic data generator	http://bigdata.cs.unb.ca/projects/sanzu/
	Described in D1.2. Section 7.14 [12]					
AIM Benchmark	How to store and analyse billing data of subscribers and make marketing campaigns [56]	Based on Analytics Matrix and divided into two parts.	Main metrics are: (a) Overall performance, (b) Read performance, (c) Write performance, (d) query response times, and (e) Impact of number of aggregates.	Multimedia databases (MMDBs) such as HyPer or Tell, and modern streaming systems like Flink and hand-crafted systems.	Analytics Matrix	https://github.com/tellproject/aim-benchmark
	Described in D1.2. Section 7.14 [12]					
Penn Machine Learning Benchmark (PMLB)	Supervised machine learning algorithms [57][57]		Number of instances. number of features, the number of categorical features, the number of discrete features, the number of continuous-valued features , endpoint type, the	ML benchmark suites including the UCI ML repository, Kaggle, KEEL and the meta-	A set of data sets cover a broad range of applications, and include binary/multi-class classification problems and regression problems, as well as combinations of	https://github.com/EpistasisLab/penn-ml-benchmarks

Name	Compares	Workload type	Metrics	Frameworks	Data Types	URL
			number of classes to predict in each dataset's endpoint and the class imbalance.	learning benchmark.	categorical, ordinal, and continuous features.	
	Described in D1.2. Section 7.14 [1]					
Edge AIBench	Edge AI benchmarks consist of 4 typical edge computing AI scenarios which covers the complicities of the most edge computing AI scenarios and 8 application benchmarks. These four scenarios includes: ICU Patient Monitor, Surveliance Camera, Smart Home, Autonomous Vehicle [59]	Edge computing scenarios workloads.	Performance, security, and privacy edge computing scenarios dependent.	Edge Computing, Cloud computing and AI frameworks.	Real-world datasets: MIMIC-III, Market-1501, LibriSpeech LFW, Tusimple and German Traffic Sign Recognition.	http://www.benchcouncil.org/EdgeAIBench/index.html
	Described in D1.2. Section 7.16 [1]					

Table 4 Summary of IoT, Edge and Fog benchmarks

As in the previous cases, the following benchmarks belong to this category, but have not been listed in Table 4 as they fit better in previous layers: Hobbit, Hermit and BenchIoT.

4. 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 D1.4 document – DataBench Framework – with Horizontal Data Management benchmarks – focuses on the classification of benchmarks according to the four bottom layers of the BDV Reference Model related to data management, data protection and HPC/Cloud/Edge/IoT. The document has listed the benchmarks under the four categories. The information collected information will serve as input for the DataBench platform, to be made operational and accessible via the DataBench Toolbox .

It is worth noticing that the data management and storage layer contains the majority of the benchmarks studied, as it is one of the classical areas where benchmarking has been applied. This layer covers benchmarks for data acquisition, curation and storage, therefore benchmarking file systems and different types of databases (SQL and NoSQL, Graph databases, RDF storage systems, etc.). To this layer belong some of the most successful historical benchmarks such as the TPC-series of benchmarks BigBench and BigDataBench. There is a huge community around these benchmarks. It is therefore of interest for DataBench to position the offering and integrate some of the benchmarks from this layer into the DataBench Toolbox, as maximize the impact in an existing community.

On the other side of the spectrum, Data Protection related to big data, including aspects such as auditing, authentication and anonymization, has been less represented in terms of benchmarking than other layers. It is foreseen that with the introduction of GDPR in 2018 and the advent of new projects related to data spaces (such as the ones recently funded for personal and industrial data spaces under the BDV PPP umbrella), this would change.

In recent years there has been also many efforts to benchmark big data in HPC and Cloud systems. The BDVA SRIA already pointed out that the usage of Cloud and HPC is essential for big data processing and AI. In fact there are efforts of convergence between Cloud and HPC, especially in what is related to analytic at scale and AI, especially for training models that need huge quantities of data and powerful processing power. However, the benchmarks related to HPC are less prominent so far than in cloud, as most of the efforts related to big data have been for cloud infrastructures and fine-tuned for cloud computing. Most of the typical big data engines (i.e. Hadoop, SPARK) were born for the cloud environment and although there has been efforts to port them to HPC, still this is used in a minority of the cases. This is probably changing in the coming years, with more benchmarks being released in the area of HPC, as it can offer some advantages for big data such the fast communication between HPC nodes compared to distributed cloud resources, use of new hardware (i.e. FPGA, specific deep learning processors), or the good performance for linear algebra, among others.

The bottom layer of the BDVA Reference Model showcases benchmarks for IoT, Edge and Fog. These are relatively new, but most of the data used in many big data settings are coming from IoT devices, and therefore benchmarks in this layer are key to understand the benefits of big data. There are overlaps between IoT, Edge and Cloud benchmarks, although some of the benchmarks studied in the document are related to IoT. IoT benchmarks are related in many cases to data processing, as they are handling real-time streams, specific data storage

for timestamped data or in-memory computing. There are less benchmarks related to Edge and Fog, as the use of big data in these settings is relatively newer.

5. References

[1]. DataBench Deliverable D1.2 White Paper - DataBench Framework – with Vertical Big Data Type benchmarks, – available at https://www.databench.eu/public-deliverables/

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

[3]. BDVA SRIA version 4. [Online]. Available: http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf (15 December 2019).

[4]. The Technology Stacks of HPC and Big Data Computing:: What they can learn from each other. [Online]. Available: https://www.etp4hpc.eu/pujades/files/bigdata_and_hpc_FINAL_20Nov18.pdf

[5]. TERMinator Suite: Benchmarking Privacy-Preserving Architectures. Dimitris Mouris, Nektarios Georgios Tsoutsos, and Michail Maniatakos.

[6]. R.Beaulieu, D.Shors, J.Smith, S.Treatman-Clark, B.Weeks, and L.Wingers, "The SIMON and SPECK lightweight block ciphers," in Design Automation Conference (DAC). ACM, 2015, pp. 1–6.

[7]. D. E. Knuth, "Textbook examples of recursion," Artificial Intelligence and Mathematical Theory of Computation, pp. 207–230, 1991.

[8]. Understanding and Benchmarking the Impact of GDPR on Database Systems. Supreeth Shastri, Vinay Banakar, Melissa Wasserman, Arun Kumar, Vijay Chidambaram.

[9]. PostgreSQL: The World's Most Advanced Open Source Relational Database. https://www.postgresql.org/,Accessed Mar312019.

[10]. Redis Data Store. https://redis.io,AccessedJan2019.

[11]. BenchIoT: A Security Benchmark for the Internet of Things. Naif Saleh Almakhdhub, Abraham A. Clements, Mathias Payerk, Saurabh Bagchi.

[12]. ABench: Big Data Architecture Stack Benchmark. - Ivanov, Todor. Singhal, Rekha.

[13]. 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.

[14]. 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)

[15]. 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.

[16]. 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.

[17]. Cooper, Brian F., et al. "Benchmarking cloud serving systems with YCSB."

[18]. Ivanov, Todor, et al. "Big data benchmark compendium." Technology Conference on Performance Evaluation and Benchmarking. Springer, Cham, 2015

[19]. Kim, Kiyoung, et al. "Mrbench: A benchmark for mapreduce framework." 2008 14th IEEE International Conference on Parallel and Distributed Systems. IEEE, 2008.
[20]. Andrew Pavlo, Erik Paulson, Alexander Rasin, et al. "A Comparison of Approaches to Large-Scale Data Analysis". In: SIGMOD. 2009, pp. 165–178
[21]. Huang, Shengsheng, et al. "The HiBench benchmark suite: Characterization of the MapReduce-based data analysis."
[22]. Ahmad, Faraz, et al. "Puma: Purdue MapReduce benchmarks suite." (2012).
[23]. Sangroya, Amit, Damián Serrano, and Sara Bouchenak. "MRBS: Towards dependability benchmarking for Hadoop MapReduce."
[24]. Ahmad Ghazal, Tilmann Rabl, Minqing Hu, et al. "BigBench: towards an industry standard benchmark for Big Data analytics".
[25]. "Lei Wang, Jianfeng Zhan, Chunjie Luo, et al. „BigDataBench: A Big Data Benchmark Suite from Internet Services".
[26]. Mayuresh Kunjir, Prajakta Kalmegh, and Shivnath Babu. „Thoth: Towards Managing a Multi-System Cluster".
[27]. Jaume Ferrarons, Mulu Adhana, Carlos Colmenares, et al. „PRIMEBALL: A Parallel Processing Framework Benchmark for Big Data Applications in the Cloud"
[28]. Angles, Renzo, et al. "The linked data benchmark council: a graph and RDF industry benchmarking effort." ACM SIGMOD Record 43.1 (2014): 27-31.
[29]. 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.
[30]. Raghunath Othayoth Nambiar, Meikel Poess, Akon Dey, et al. "Introducing TPCx-HS: The First Industry Standard for Benchmarking Big Data Systems".
[31]. Min Li, Jian Tan, Yandong Wang, Li Zhang, and Valentina Salapura. "SparkBench: a spark benchmarking suite characterizing large-scale in-memory data analytics"
[32]. Andrew Bond, Douglas Johnson, Greg Kopczynski, and H. Reza Taheri. "Profiling the Performance of Virtualized Databases with the TPCx-V Benchmark"
[33]. Pouria Pirzadeh, Michael J. Carey, and Till Westmann. „BigFUN: A Performance Study of Big Data Management System Functionality"
[34]. Ahmad Ghazal, Tilmann Rabl, Minqing Hu, et al. „BigBench: towards an industry standard benchmark for Big Data analytics".
[35]. Alexandru Iosup, Tim Hegeman, Wing Lung Ngai, et al. "LDBC Graphalytics: A Benchmark for Large-Scale Graph Analysis on Parallel and Distributed Platforms"
[36]. Milind Bhandarkar. "AdBench: A Complete Benchmark for Modern Data Pipelines".
[37]. Xu, Zhen, et al. "GARDENIA: A Domain-specific Benchmark Suite for Next-generation Accelerators." arXiv preprint arXiv:1708.04567 (2017).
[38]. Bagan, Guillaume, et al. "gMark: Schema-driven generation of graphs and queries." IEEE Transactions on Knowledge and Data Engineering 29.4 (2016): 856-869.
[39]. Hesse, Guenter, et al. "Senska–Towards an Enterprise Streaming Benchmark." Technology Conference on Performance Evaluation and Benchmarking. Springer, Cham, 2017.

[40]. Ivanov, Todor & Singhal, Rekha. (2018). ABench: Big Data Architecture Stack Benchmark. 13-16. 10.1145/3185768.3186300.

[41]. Sakr, Sherif & Casati, Fabio. (2010). Liquid Benchmarks: Towards an Online Platform for Collaborative Assessment of Computer Science Research Results. 6417. 10-24. 10.1007/978-3-642-18206-8_2.

[42]. Josep LI. Berral, Nicolás Poggi, David Carrera, Aaron Call "ALOJA: A framework for benchmarking and predictive analytics in Big Data deployments".

[43]. Axel-Cyrille Ngonga Ngomo and Michael Röder. „HOBBIT: Holistic benchmarking for big linked data". In: ERCIM News 2016.105

[44]. Chunjie Luo, Jianfeng Zhan, Zhen Jia, et al. „CloudRank-D: benchmarking and ranking cloud computing systems for data processing applications".

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

[46]. Ivanov, Todor, et al. "Big data benchmark compendium." Technology Conference on Performance Evaluation and Benchmarking

[47]. Jaume Ferrarons, Mulu Adhana, Carlos Colmenares, et al. „PRIMEBALL: A Parallel Processing Framework Benchmark for Big Data Applications in the Cloud"

[48]. W Gao, F Tang, L Wang, J Zhan, C Lan, C Luo. AIBench: an industry standard internet service AI benchmark suite.

[49]. Sanket Chintapalli, Derek Dagit, Bobby Evans, et al. "Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming".

[50]. Min Li, Jian Tan, Yandong Wang, Li Zhang, and Valentina Salapura. "SparkBench: a spark benchmarking suite characterizing large-scale in-memory data analytics"

[51]. 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.

[52]. Anshu Shukla, Shilpa Chaturvedi, and Yogesh Simmhan. „RIoTBench: A Realtime IoT Benchmark for Distributed Stream Processing Platforms". In: CoRR abs/1701.08530 (2017).

[53]. Arvind Arasu, Mitch Cherniack, Eduardo F. Galvez, et al. „Linear Road: A Stream Data Management Benchmark"

[54]. Muhammad Intizar Ali, Feng Gao and Alessandra Mileo. CityBench: A Configurable Benchmark to Evaluate RSP Engines using Smart City Datasets

[55]. Mil 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 ind Bhandarkar. "AdBench: A Complete Benchmark for Modern Data Pipelines".

[56]. 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.

[57]. 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)

[58]. AIBench: An Industry Standard Internet Service AI Benchmark Suite

[59]. 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, Junchao Shao, Zhenyu Wang, Xiaoyu Wang, Hainan Ye