A Comparative Analysis and Evaluation of MapReduce Cloud Computing Simulators

Ebenezer Komla Gavua*
*Institute of Information Technology
University of Miskolc
Miskolc, Hungary
Email: {gavua,kecskemeti}@iit.uni-miskolc.hu

Gabor Kecskemeti*†
†Department of Computer Science
Liverpool John Moores University
Liverpool, UK
Email: g.kecskemeti@ljmu.ac.uk

***Abstract*—The application of MapReduce cloud computing simulators for research and development is becoming popular, due to its efficiency and ease of utilization. This ignited the development of several cloud simulators for algorithm testing and performance analysis of dynamic MapReduce environments. The selection of appropriate simulator for a specific research remains a challenge. We have designed a MapReduce classification framework to guide cloud and big data researchers on suitable tools. We have reviewed eleven MapReduce specific simulators. Our evaluation first revealed thirty general functional requirements for more widely applicable cloud simulators. Then, we focused on specific concerns of MapReduce related simulations and filtered the general requirements down to the most relevant thirteen. Our evaluation highlighted the strengths and weakness of several MapReduce simulators. IoT-Based applications, stream processing and replaying of production cluster workloads are key criteria absent from many simulators. Therefore, we identified these as gaps, that simulator developers could focus on when extending their works towards MapReduce oriented simulations. Finally, researchers simulating dynamic behaviors of Hadoop clusters should select simulators efficient in parameter tuning.**

I. Introduction

MapReduce (MR) computing paradigm [1] and its open source implementation Hadoop [2] have become a de-facto standard to process big data in a distributed environment. This paradigm has been widely accepted because of its flexibility, reliability and robustness. To better understand Hadoop's computing model [3], it is also important to highlight its issues when dealing with big data. Effective handling of MapReduce requires in depth understanding of its capabilities. This enables its users to process data intensive applications, access IoT-Based applications with real-time stream processing capabilities and increase the performance of the infrastructure. Processing heterogeneous data has remained a major challenge in the computations of Hadoop MapReduce Workloads.

When addressing such challenges MapReduce Simulators are often used to understand the relevant underpinnings of MR. These simulators are often used because they are cost-effective and flexible. Past research with these simulators have already identified several issues. First, there could be weaknesses in the infrastructure which require addressing. Also, the absence of an appropriate data model to enhance data processing by simulators has also compounded the problems. These problems often lay hidden due the absence of a survey highlighting functional requirements expected of MapReduce simulations aiming at heterogeneous data processing.

Choosing an appropriate simulator could also be a problem for researchers. Past research [4], [5], [6] suggested ways to select a simulator for generic cloud simulations based on nine evaluation criteria. From these nine, the classifications most importantly focused on base platforms (i.e., whether a new simulator is built upon some pre-existing simulation framework) and energy modeling. Other, more MapReduce oriented works (e.g., [7]) focused on simulation and modeling performance. However, existing surveys provide little insight into features and functionalities relevant to users.

As the main contribution of this paper, we propose an alternative, functionality focused classification for MapReduce simulators. This fosters new simulator and simulation designs aiming at new heterogeneous data management models. Our thirteen classification criteria is the result of the systematic review of eleven MapReduce simulators. Our criteria helps to highlight the strengths and weaknesses of a given simulator from the point of view of functional requirements.

These requirements range from data-intensive applications, through Internet of Things (IoT) support to even parameter tuning techniques. This allows researchers to find simulators that enable their envisioned MR models: (i) to scale out to meet the demands of more than the normal inflows of data and workloads; (ii) to provide support for multitude of devices utilizing variety of protocols; (iii) to adjust specific MR and algorithmic configuration parameters to achieve an improved performance or quality.

As our final contribution, based on the functionality-based review of eleven state-of-the-art MR simulators, we have identified several potential issues, e.g.,: (i) lack of detailed processing of task trackers; (ii) stream processing deficiencies; (iii) limited support for IoT-Based applications; and finally, (iv) only partial support for replaying production cluster workloads and process real-time data.

The remainder of this paper is structured as follows. In Section II, we identified the past surveys surrounding the field of MR simulations. In Section III, we present our classification methodology to evaluate MapReduce cloud simulation tools. Then, we evaluate the latest MR simulators with our proposed classification technique and identify recommendations for future users and simulation designers in Section IV. Finally, Section V concludes the paper.

II. Related works

There have been many studies utilizing simulators to investigate distributed systems. The application of simulating techniques have proved to be very effective in understanding computer systems in a cost-effective and flexible environment. Over time, research has progressed to evaluating these simulators and highlight their benefits and disadvantages.

In most cases, classification frameworks have been proposed to assist in the selection of suitable simulators for research. These works, which are highly connected to ours, are discussed in this section to serve as a basis for our proposed MapReduce simulator classification framework.

Naicken et al. [8] analyzed nine peer-to-peer network simulators based on six criteria. The architecture and scalability were key criteria discussed. This was mainly due to the fact that peer-to-peer simulators were mainly designed to be scalable. Also, they found it important to highlight if a specific simulator supported structured/unstructured overlay simulation or both. They also discussed the usability of the simulators and pointed out few issues respect to underlying networking simulations. These authors limited their work to peer-to-peer network simulators. Therefore, their work is not directly applicable to our scope on MapReduce & cloud computing.

Pan et al. [9] evaluated the current development status of network simulators. They discussed key concepts of network simulators while analyzing them. The main features of these simulators were discussed also including their current status and future development prospects. Their work formed a base for comparative studies for other distributed system simulators. These authors focused on criteria such as if they are commercial or free, or if they are simple or complex ones. Most of the networking simulators were found to be open source which made them accessible for research and development. These authors limited their work to four network simulators and hence, their results needs revision in relation to computation models for MapReduce and cloud computing.

Sundani et al. [10] conducted a comprehensive survey on a number of Wireless Sensor Network simulators. They highlighted the strengths and weaknesses of the simulators. These highlights allow users to select the most appropriate for their testing. Fourteen wireless network simulators were evaluated based on different criteria. Comparative results we presented in tabular form to assist researchers in choosing the best simulator for a particular application environment.

A few research activities have been undertaken with respect to classifying cloud simulators. The use of cloud simulators provides a conducive environment for assessing of various scenarios under different workloads. Purchasing software costs less when compared to purchasing hardware and proprietary software. Also many simulators are available for free.

A few papers [4], [5], [6] classified various simulators based on nine evaluation criteria. These authors focused their work on evaluating various simulators to improve simulation efficiency and cost. As there are cloud simulators for specific purposes, it is critical to select appropriate ones for research. Among the criteria discussed, we shortly discuss the most relevant three in relation to our study. First, they identified the base platform, which refers to the preexisting frameworks on which these simulators were built upon. This is critical for the operation, performance and validity of the simulators. Next, they discuss the energy model offered to evaluate consumption of data centers. Finally, they highlighted the importance of federation policies that allow coordination among different cloud service providers. These policies often support simulated application level internetworking and workload migration for high QoS. These authors limited their work to popular cloud computing simulators neglecting the investigation towards the simulators MapReduce potential.

Ettikyala et al. [11] compared fifteen cloud simulators with five evaluation criteria. They undertook the study to evaluate cloud computing systems, their security, performance and application behaviors. Their focus was to analyze the characteristics which made them unique. The availability of the simulator was used to analyze, whether the simulator was available as open source or a commercial tool. The authors were also interested in whether the simulators included a packet level network simulation, or if the simulation framework was using continuous or event based models. Although these authors worked within the cloud computing domain; they did not discuss any MapReduce simulator. As MapReduce is commonly utilized in cloud contexts nowadays, such simulations require more attention.

Sinha et al. [12] evaluated fifteen cloud simulators with six evaluation criteria. They analyzed the performance of a cloud systems with reduced complexity. The outstanding criteria utilized in their classification was their focus on the software and/or hardware features that can be simulated. Another key feature was the presence or absence of graphical user interface (GUI). This was considered because some researchers preferred command line to GUI. These authors also limited their work on popular cloud computing simulators. Their work serves as a foundation for the our proposed classification, because although, they critically reviewed renown cloud simulators; MapReduce simulators were not reviewed.

A critical evaluation was performed by [13] on modeling and simulation frameworks for cloud computing environment. In their work, ten most popular simulation software frameworks and testbeds were evaluated according to six evaluation criteria. They ranked the most popular and available simulators in the cloud computing research community. The work of these authors is connected to our proposed classification framework, since they did a detailed analysis of the ten simulators. However, their lack of focus on MapReduce also highlighted the need for a more specific classification framework.

A few researchers have evaluated MapReduce oriented simulators. Some of the authors focused on studying problems that cannot be easily studied using real MapReduce systems. They have shown that some problems require a controllable environment to enable reproducible experiments.

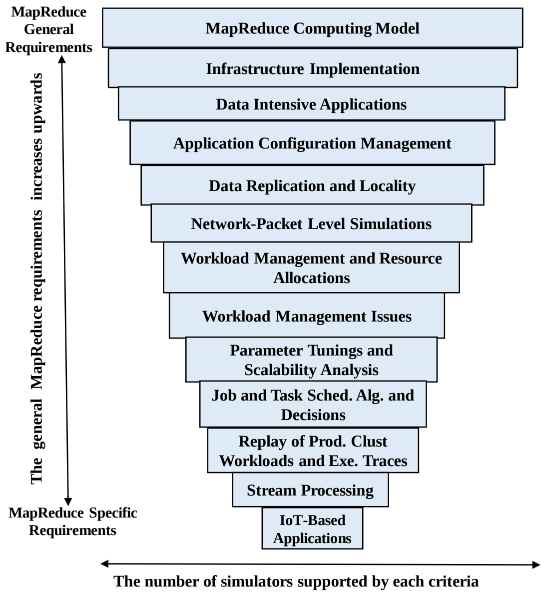

Fig. 1. Criteria for Classification

Wang [14] analyzed eight simulators with three evaluation criteria. The main criteria were whether the simulators were workload-aware or resource-contention-aware. Workload-aware simulators were able to predict performance of a MapReduce job that runs on a cluster when other jobs are also running. The simulators could model performance of an application in fine grain, i.e., with sub-task stages, so they can model resource contention where multiple tasks share the same resource and will run slower. Prior to their comparisons, the authors classified the simulators into two categories. Namely: simulators for evaluating schedulers and simulators targeting individual jobs. They concluded that each of the simulators was built upon slightly different performance model. However, these authors did not tackle detailed functional requirements for MapReduce simulators.

The above discussions shows, that the current body of research does not focus on surveying MapReduce simulators. As the MapReduce cloud computing model has become a standard to process big data in a distributed environment, simulating its behavior is critical to enhancing its functionalities. The absence of a survey to highlight its strengths and weakness affects the choice of appropriate simulator for research and could potentially limit possible future research directions on the simulators themselves.

III. Methodology

In order to critically analyze and evaluate the existing MapReduce cloud simulation and modeling tools, we proposed to develop a novel evaluation framework. This classification framework consists of a set of criteria which can be used to compare the various simulators. These criteria were selected after evaluating the functional requirements of MapReduce simulators. The criteria can potentially highlight the strengths and weaknesses of current and future MapReduce simulators. The proposed classification framework ensures that a suitable simulator can be selected for designing future data models.

Our evaluation work was done on two levels to enable the review to cover all necessary areas: general (cloud oriented) and specific (MapReduce focused) evaluation. Our general evaluation concentrated on the functional requirements necessary for running cloud computing simulators. This produced thirty themes such as network topology support and job completion time estimation. We have identified the themes based on past classifications of these simulators (as discussed in Section II), as well as based on our own research.

Then, based on these themes, we have done a MapReduce specific evaluation which refined our potential criteria to the thirteen themes most relevant to Hadoop/MapReduce simulations. The evaluation work showed that the absence of any of the thirteen criteria made the simulating of a detail MapReduce framework incomplete. Thus we focused our work on these. An overview of our criteria is presented in Figure 1, then we discuss each one of them in detail below.

A. Data Intensive MapReduce Applications

This refers to applications that enable computer systems to scale out to meet the demands of more than the normal inflows of data and workloads in an organization. As such, most existing implementations of the MapReduce programming model are designed around clusters of low-end commodity servers [15].

Additionally, simulators must possess certain properties before being classified into this category. These include: (i) to accommodate large quantities of data and manipulate it efficiently; (ii) efficient programming model for data computation and analysis of data; (iii) scalable underlying hardware and software; (iv) the simulator should be reliable and available.

B. Parameter Tunings and Scalability Analysis

This refers to adjusting specific parameters of either a device, model or algorithm upwards or downwards to achieve an improved or specific result in the cloud infrastructure. Furthermore, scalability analysis determines whether a proposed new algorithm is capable of increasing its capacity with increasing demand. The capacity is the maximum workload the algorithm can handle on a given device while still meeting its SLA (Service Level Agreement) [16].

In order for a simulator to be considered in this category, key parameters should be measured and increased when needed. Additionally, the simulator itself should be able to scale with the increasing complexity of the to be simulates system. In order to determine whether the scalability analysis criteria is available in a simulator, the next factors should be considered: load-, space-, space-time and structural scalability [17].

C. Workloads Management and Resource Allocation

A MapReduce workload is an abstraction of the actual work to be performed. These workloads require effective management to forestall processing challenges such as which processor to assign it to, which scheduling algorithm to utilize in the assignment and the estimation of its processing duration. Resource allocation could often times starve services if allocation is not managed precisely [18].

Simulators in this category should possess out of the box workload management and resource allocation strategies. Also, the following issues should be considered before classification: (i) they should ensure that no two applications try to access the same resource at the same time; (ii) the scarcity of resources does not lead to resource contention; (iii) resource fragmentation should be avoided [19].

D. Job and Task Scheduling Algorithms and Decisions

Job scheduling is the process of allocating system resources to many different tasks. The system handles prioritized job queues that are awaiting CPU time, and it should determine which job to be taken from which queue and the amount of time to be allocated for the job. This type of scheduling makes sure that all jobs are carried out fairly and on time. Task scheduling in Hadoop MapReduce involves allocating appropriate tasks of the jobs to the appropriate server. The Hadoop scheduling model is a master/slave cluster model [20].

Additionally, a simulator must be able to determine the following to qualify in this category: (i) the CPU utilization; (ii) the general throughput of all processes; and (iii) the turnaround time of various processes.

E. Data Replication and Locality

Data replication is the process of copying data from one location to another. The technology helps an organization to possess up-to-date copies of its data in the event of a disaster. In Hadoop, data locality is the process of moving the computation close to where the actual data resides on the node, instead of moving large data to computation [21].

Also, simulators should be able to perform certain functions before qualifying for this category. These include: (i) continuous replication with many recovery points; (ii) cross-platform replication (i.e., disk to cloud and vice versa); (iii) replication of synchronized data with zero data loss. Furthermore, a simulator should have the appropriate network topology and understand data locality.

F. Workload Management Issues

Workload management issues are closely related to the previous criterion and arise periodically while running of MapReduce applications. A MapReduce workload is an abstraction of the actual work that virtual machine instances are going to perform. The main issues are scalability of application platforms, data locality and replication [22].

In order to determine the presence of this feature, it is important to assess how all the workloads are distributed during data processing. Data distribution imbalances represent the presence of workload management issues. In most cases, there are challenges with monitoring about which data center has more data and which one has less.

G. Infrastructure Implementation

In computing, information technology infrastructure is composed of physical and virtual resources that support the flow, storage, processing and analysis of data. Virtual infrastructures may be centralized within a data center, or they may be decentralized and spread across several data centers that are either controlled by the organization or by a third party, such as a co-location facility or cloud provider [23].

Moreover, some simulators are built upon existing simulation frameworks. The features of the existing platform are inherited. For instance, MRSG simulator is built on SimGrid and MR-CloudSim is built on CloudSim. Simulators in this category, should have an underlying infrastructure that can interface with MapReduce framework. They should be capable of VM provisioning, scheduling policies, design and analysis of different hardware configurations.

H. Application Configuration Management

Application configuration management (ACM) allows a user to create templates in order modify and manage application configurations associated with server applications. ACM enables a user to manage, update, and modify those configurations from a central location, ensuring that applications in the facility are accurately and consistently configured [24].

Additionally, simulators must possess the following features to be classified into this category: (i) they should provide high performance; (ii) the applications should have a high-volume activation capabilities; (iii) the simulator should be scalable on and preferably possess event-driven bus architecture.

I. Stream Processing

Stream processing is a technology that allows users to query a continuous data stream and quickly detect conditions within a small time period from the time of receiving the data. The detection time period may vary from a few milliseconds to minutes. Stream processing is also known as real-time/streaming analytics, complex event processing [25].

Some of the simulators possessing this feature ensure that they can process data "in-stream". Such stream processing enabled simulators should guarantee predictable and repeatable outcomes [26]. The simulator should allow the modeling of storage, access and modification of state information, and its combination with live streaming media.

J. Replaying of production cluster workloads and Execution Traces

This explains how production cluster workload as well as execution traces are replayed on simulators to measure the efficiency of the designed simulation. Workload is an abstraction of the actual work that virtual infrastructures perform, and traces are created to track specific actions performed on MapReduce systems. They provide valuable information for troubleshooting MapReduce issues and tuning engine performance [27].

The role of that cluster workloads and execution traces play are crucial to determining the appropriate results from each simulator. The key properties observable in simulators in this category are (i) data characteristics, (ii) job submission patterns, and (iii) common jobs [28].

K. IoT-based Applications

IoT-Based applications refers to systems that build on and utilize various types typical IoT connections such as RFID, Wi-Fi, Bluetooth, and ZigBee in addition to allowing wide area connectivity using many technologies such as GSM, GPRS, 3G, and LTE. These systems allow seamless information sharing about the condition of things and the surrounding environment with people, software systems and other machines [29]. From the proposed classification only one simulator (i.e., IoTSim) supports this criteria. The capability to handle IoT-Based Applications is very important to enable effective processing of data emanating from IoT devices.

L. MapReduce Computing Model

MapReduce is a programming model developed for large-scale analysis. It takes advantage of the parallel processing capabilities of a cluster in order to quickly process very large data sets in a fault-tolerant and scalable manner. The core idea behind MapReduce is mapping the data into a collection of key/value pairs, and then reducing overall pairs with the same key. Using key/value pairs as its basic data unit, the framework is able to work with the less-structured data types and to address a wide range of problems. In Hadoop, data can originate in any form, but in order to be analyzed by MapReduce software, it needs to eventually be transformed into key/value pairs [30]. All the reviewed simulators support this feature. The simulators in this category should be able to generate big data sets, run with a parallel, distributed algorithms on a cluster.

IV. Evaluation

The proposed classification highlights the strengths and weaknesses of MapReduce simulators. The issues highlighted, determine the suitability of a MapReduce simulator for simulating various scenarios under different workloads. The classification will help selecting a suitable simulator for designing new data models and algorithms for MapReduce. This section comprises of two subsections, first it starts with the classification of eleven simulators then it follows up with our recommendations for simulation developers and other researchers in the field.

A. Classifying The Simulators

In this section, the features of the simulators are discussed in relation to our classification framework. This classification is also shown in a summarized form in Table I.

1) SimMR: The main goal for the development of the MapReduce simulator, called SimMR was to design an accurate and fast simulation environment [31]. It was also developed for evaluating workload management and resource optimization decisions in MapReduce environments. SimMR consists of the following three main components: (i) trace generator, (ii) simulator engine and (iii) scheduling policy. It is capable of replaying the scheduling decisions over a large workload (several months of job logs) in a few minutes on a single machine [41]. With respect to the proposed simulation framework, it is capable of handling data intensive MapReduce applications, replaying of production workload and tuning up of parameters like schedulers and job queues. However, it cannot process IoT-Based applications nor stream processing.

2) SimMapReduce: This simulator is designed to evaluate the performance of MapReduce applications under different scenarios. The system design of SimMapReduce applies multi-layer architecture. This has made the simulator a flexible toolkit and is convenient to extend by others [32]. It considers some special features such as data locality and dependence between Map and Reduce. It also provides essential entity services that can be predefined in XML [42]. The proposed classification framework has shown that, this simulator is capable of performing most MapReduce simulations. However, it lacks the capability to interface with IoT-Based Applications and replaying the production cluster workload and execute database traces.

3) IoTSim: This simulator is built on top of a widely used simulator, CloudSim, and has been extensively extended to improve the preexisting functions. IoTSim allows simulation of IoT applications and inherently supports big data processing. It is efficient for analyzing the impact and performance of IoT-based applications [33], [43]. The simulator has five main layers. These are: the CloudSim core simulation engine, CloudSim simulation layer, storage layer, big data processing layer, and the user code layer [41]. The proposed classification framework has highlighted that it is the only MapReduce Simulator capable of interfacing with IoT applications.

4) HSim: The HSim simulator is designed for Hadoop MapReduce applications. The key contributions of HSim lie in its high accuracy in simulating the dynamic behaviors of Hadoop environments and the large number of Hadoop parameters that can be modeled in the simulator, by tuning their values [16]. The key parameters for running this simulator include: node-, cluster-, Hadoop system- and HSIM parameters. It can also be used to study the scalability of MapReduce applications which might involve hundreds of nodes [44].

The proposed classification framework has shown that, the HSim simulator is an efficient simulator suitable for accurately simulating of Map Reduce Hadoop clusters. However, it lacks the capabilities to process streaming data and interfacing with IoT-based applications.

5) WaxElephant: This simulator is designed to help Hadoop operators in scalability analysis and parameter tuning. WaxElephant is composed of a pluggable job scheduling module, a load generator and a simulator engine [34]. The job scheduling module receives jobs from the load generator and monitors the status of computing nodes. The load generator is responsible for producing MapReduce jobs, either replaying jobs collected from a historical log, or synthesizing a set of jobs that follow a user-defined statistical properties [45], [46]. This simulator has

TABLE I
PROPOSED CLASSIFICATION FRAMEWORK

Evaluation Criteria	[16]	[31]	[32]	[33]	[34]	[35]	[36]	[37]	[38]	[39]	[40]
Data Intensive Applications	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Parameter Tunings and Scalability Analysis	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓
Workload Mgt and Resource Allocations	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓
Job and Task Scheduling Algorithms and Decisions	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓
Network- Packet level Simulations	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
Data Replication and Locality	✓	✓	✓	✓	✓	✓	×	✓	✓	✓	✓
Workload Management Issues	✓	✓	✓	✓	✓	✓	×	✓	✓	×	✓
Infrastructure Implementation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Application Configuration Management	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Stream processing	×	×	✓	✓	×	×	×	×	×	×	×
Replaying of Prod. Cluster Workloads and Exe. Traces	✓	✓	×	×	✓	×	×	✓	✓	×	✓
IoT-Based Applications	×	×	×	✓	×	×	×	×	×	×	×
MapReduce Computing Model	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

most of the features required to run MapReduce simulations according to the proposed classification framework. It also ensures scalability analysis and parameter tuning which are key metrics required in every efficient MapReduce simulator. However, the simulator is incapable of stream processing and interfacing with IoT-Based applications.

6) MRSim: The goal of the MRSim project was to design and implement a MapReduce simulator to model the behavior of data mining algorithms on Hadoop environments [35]. This allows the deduction of the best algorithm configuration values that tune a Hadoop cluster for best performance. MRSim was created by extending the discrete event engine used by SimJava to accurately simulate the Hadoop environment. SimJava enables interactions between different entities to be simulated within a cluster. GridSim package was also used for network simulation. GridSim was written in Java on top of SimJava to achieve efficiency on network simulation [47]. This simulator is effective in enabling users to estimate the best job configuration to get optimum performance for certain algorithms [48]. This simulator has most of the features required for an effective MapReduce simulation according to its evaluation by the proposed classification framework. However, the simulator lacks the capacity to process streaming data, and interface with IoT-based applications and replay production cluster workloads.

7) MR-CloudSim: This is an implementation of a bare bone structure of MapReduce on CloudSim environment. The goal of this implementation is to provide an easier and cheaper way to validate MapReduce operations with far less cost. In order to implement MapReduce in CloudSim, the MapReduce model was implemented outside of the original CloudSim code to minimize unpredictable behavior. Also, to provide a bridge between the two implementations, native CloudSim code were modified [36]. This simulator provides simplistic, single-state Map and Reduce computation. It lacks support for network link modeling [49]. The proposed classification framework rated this simulator low due to its features. However, it is capable of running a few data intensive applications. It cannot process streaming media nor IoT-based applications.

8) Mumak: The goal of Mumak is to build a discrete event simulator for conditions when a Hadoop MapReduce scheduler performs actions on a large-scale specific workload. Mumak is composed of the following entities: client, job tracker, task tracker and the main part named simulated engine [37]. As input, Mumak takes a reasonably large workload and simulates them in a matter of hours if not minutes on very few machines. Additionally, this simulator utilizes data from real experiments to estimate performance metrics (e.g., completion time) for Map and Reduce tasks with different scheduling algorithms [33]. However, in cases where data from real experiments does not exists, Mumak cannot estimate completion time for Map and Reduce tasks. The shuffle and sort phases are not modeled [50]. Furthermore, the classification framework has shown that the simulator is not prepared for IoT-based applications, stream processing and effective parameter tuning.

9) HaSim: This Hadoop simulator was designed to tune the performance of a Hadoop cluster and to study the behaviors of Hadoop applications [38]. The key contributions of HaSim lie in its high accuracy in simulating the dynamic behaviors of Hadoop environments and the large number of Hadoop parameters that can be modeled in the simulator. The main parameters involved in its design focuses on the node-, cluster-, Hadoop system-, and HaSim simulator level [51]. The proposed classification frame rated this simulator very high as capable of simulating MapReduce effectively. However, it cannot handle IoT-Based applications nor stream processing.

10) MPerf: The MPerf simulator was designed to understand the sensitivity of application performance to platform parameters [39]. MPerf provides fine-grained simulation at sub-phase level, models inter-and intra-rack network communications, as well as activities inside a single node, such as processor time consumed by a job, and disk I/O time for reading inputs and writing results. Although based on Hadoop, MPerf does not simulate important aspects of the platform such as speculative execution. Its implementation is able to simulate only a single device per node and allows one replica per chunk [52]. The proposed classification framework rated this simulator low; because it lacks a lot of features to simulate MapReduce completely. It cannot also process streaming media, interface with IoT-based applications and handle workload management issues.

11) MRSG: The simulator aims to facilitate research on the behavior of MapReduce platforms, and possible changes in the technology. This simulator seeks to provide: a complete API to translate theoretical algorithms, such as task scheduling and data distribution, into executable code; ease of change and test of different system configurations [40]. MRSG is developed on top of SimGrid, which is a simulation-based framework for evaluating cluster, grid and P2P (peer-to-peer) algorithms and heuristics [48]. The communication between MRSG and SimGrid is achieved through the use of SimGrid's MSG API. This simulator allows users to define tasks costs and intermediary data through [53]. The proposed classification framework rated this simulator as capable of simulating MapReduce effectively. However, this simulator cannot replay production cluster workloads and execute database traces, process streaming data and interface with IoT-Based Applications.

B. Recommendations

This research work has highlighted important features that are absent in most of the simulators evaluated. The key ones among them includes IoT-based applications and stream processing. Therefore, we recommend, that simulator developers should focus on incorporating some of these unavailable features into their works. Simulator developers are encouraged to also focus on some special features such as data locality and dependence between Map and Reduce, as it provides essential entity services that can be predefined. Simulator developers should dedicate much attention to parameter tuning and scalability analysis and job/task scheduling algorithms and decisions. These features are the distinguishing factors of the MapReduce programming model. Therefore, neglecting them when developing MapReduce simulators will generate inaccurate results.

The development of simulators on single computers is cheap but this research work has shown that most such simulators generate inaccurate results. We therefore recommend MapReduce clusters for simulator developers to prevent workload management issues.

Additionally, we recommended that researchers interested in simulating dynamic behaviors of Hadoop clusters should select simulators efficient in parameter tuning and scalability related decision making. These simulators can also investigate the impact of large number of Hadoop parameters. Those interested in monitoring speculative execution mechanism and data replication should select simulators running cluster of computers. Usually, those simulators are also capable of defining task costs and intermediary data. These features make the simulator's behavior closer to a real MapReduce deployment.

Simulating IoT applications in real time is gradually becoming an important area for researchers. Hence, we recommend simulators with IoT capabilities for such research work, since they have the ideal platform for stream processing. The capability of estimating job configurations to obtain optimum performance for various algorithms is a major concern in the cloud computing environment. Therefore, we recommend simulators that can capture the effects of different configurations of Hadoop setups on an algorithm's behavior in terms of speed and hardware utilization. Most MapReduce researchers prefer simulators that can replay production cluster workloads therefore, we recommend simulators that replay production cluster workloads with different scenarios of interest, assessing various what-if questions, and helping to avoid error-prone decisions.

V. Conclusion

The comparative analysis of these MapReduce simulators has revealed a lot of information that hitherto was not available. The various strengths and weaknesses of the simulators have been highlighted. This has the potential to make the appropriate MapReduce simulators easier to access for cloud researchers. The classification has revealed that all MapReduce simulators support data intensive applications, application configuration management and suitable infrastructural implementation. Additionally, the classification also highlighted that only one supports IoT-based applications and a few support the processing streaming media.

This study has given us enough information to begin focusing on the previously highlighted key issues and gaps by adding preliminary solutions to DIScrete event baSed Energy Consumption simulaTor for clouds and Federations - DISSECT-CF, [54]. Major issues such as MapReduce stream processing, workload management and job and task scheduling algorithms will be given immediate attention to enhance the efficiency of this simulator. Also, on top of the extended simulation framework, primary work will begin on designing a hybrid distributed data mining model to improve data retrieval on the Hadoop Distributed Filesystem (HDFS).

References

[1] A. Katal, M. Wazid, and R. Goudar, "Big data: issues, challenges, tools and good practices," in *Contemporary Computing (IC3), 2013 Sixth International Conference on*. IEEE, 2013, pp. 404–409.

[2] T. White, *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.

[3] J. Dean and S. Ghemawat, "Mapreduce: a flexible data processing tool," *Communications of the ACM*, vol. 53, no. 1, pp. 72–77, 2010.

[4] A. Ahmed and A. S. Sabyasachi, "Cloud computing simulators: A detailed survey and future direction," in *Advance Computing Conference (IACC), 2014 IEEE International*. IEEE, 2014, pp. 866–872.

[5] R. Kaur and N. S. Ghumman, "A survey and comparison of various cloud simulators available for cloud environment," *Int. J. Adv. Res. Comput. Commun. Eng*, vol. 4, no. 5, pp. 605–608, 2015.

[6] P. S. Suryateja, "A comparative analysis of cloud simulators." *International Journal of Modern Education & Computer Science*, vol. 8, no. 4, 2016.

[7] G. Kecskemeti, G. Casale, D. N. Jha, J. Lyon, and R. Ranjan, "Modelling and simulation challenges in internet of things," *IEEE cloud computing*, vol. 4, no. 1, pp. 62–69, 2017.

[8] S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai, "A survey of peer-to-peer network simulators," in *Proceedings of The Seventh Annual Postgraduate Symposium, Liverpool, UK*, vol. 2, 2006.

[9] J. Pan and R. Jain, "A survey of network simulation tools: Current status and future developments," *Email: jp10@ cse. wustl. edu*, vol. 2, no. 4, p. 45, 2008.

[10] H. Sundani, H. Li, V. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless sensor network simulators a survey and comparisons," *International Journal of Computer Networks*, vol. 2, no. 5, pp. 249–265, 2011.

[11] K. Ettikyala and Y. R. Devi, "A study on cloud simulation tools," *International Journal of Computer Applications*, vol. 115, no. 14, 2015.

[12] U. Sinha and M. Shekhar, "Comparison of various cloud simulation tools available in cloud computing," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 3, pp. 171–176, 2015.

[13] A. Bashar, "Modeling and simulation frameworks for cloud computing environment: A critical evaluation," in *International Conference on Cloud Computing and Services Science*, 2014, pp. 1–6.

[14] G. Wang, "Evaluating mapreduce system performance: A simulation approach," Ph.D. dissertation, Virginia Tech, 2012.

[15] J. Ekanayake, S. Pallickara, and G. Fox, "Mapreduce for data intensive scientific analyses," in *eScience, 2008. eScience'08. IEEE Fourth International Conference on*. IEEE, 2008, pp. 277–284.

[16] Y. Liu, M. Li, N. K. Alham, and S. Hammoud, "Hsim: a mapreduce simulator in enabling cloud computing," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 300–308, 2013.

[17] A. B. Bondi, "Characteristics of scalability and their impact on performance," in *Proceedings of the 2nd international workshop on Software and performance*. ACM, 2000, pp. 195–203.

[18] S. S. Manvi and G. K. Shyam, "Resource management for infrastructure as a service (iaas) in cloud computing: A survey," *Journal of network and computer applications*, vol. 41, pp. 424–440, 2014.

[19] V. Anuradha and D. Sumathi, "A survey on resource allocation strategies in cloud computing," in *Information Communication and Embedded Systems (ICICES), 2014 International Conference on*. IEEE, 2014, pp. 1–7.

[20] S. Ghanbari and M. Othman, "A priority based job scheduling algorithm in cloud computing," *Procedia Engineering*, vol. 50, no. 1, pp. 778–785, 2012.

[21] G. L. Stavrinides and H. D. Karatza, "The impact of data locality on the performance of a saas cloud with real-time data-intensive applications," in *Proceedings of the 21st International Symposium on Distributed Simulation and Real Time Applications*. IEEE Press, 2017, pp. 180–187.

[22] N. Antonopoulos and L. Gillam, *Cloud computing*. Springer, 2010.

[23] J. Peng, X. Zhang, Z. Lei, B. Zhang, W. Zhang, and Q. Li, "Comparison of several cloud computing platforms," in *Second International Symposium on Information Science and Engineering (ISISE 2009)*. IEEE, 2009, pp. 23–27.

[24] A. Khan, J. Hintsch, G. Saake, and K. Turowski, "Variability management in infrastructure as a service: Scenarios in cloud deployment models," in *Computing, Networking and Communications (ICNC), 2017 International Conference on*. IEEE, 2017, pp. 724–728.

[25] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, "S4: Distributed stream computing platform," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 170–177.

[26] M. Stonebraker, U. Çetintemel, and S. Zdonik, "The 8 requirements of real-time stream processing," *ACM Sigmod Record*, vol. 34, no. 4, pp. 42–47, 2005.

[27] W. Van der Aalst, A. Adriansyah, and B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 182–192, 2012.

[28] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating mapreduce performance using workload suites," in *2011 IEEE 19th International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS 2011)*. IEEE, 2011, pp. 390–399.

[29] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[30] T. Gunarathne, T.-L. Wu, J. Qiu, and G. Fox, "Mapreduce in the clouds for science," in *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*. IEEE, 2010, pp. 565–572.

[31] A. Verma, L. Cherkasova, and R. H. Campbell, "Play it again, simmr!" in *2011 IEEE International Conference on Cluster Computing*. IEEE, 2011, pp. 253–261.

[32] F. Teng, L. Yu, and F. Magoulès, "Simmapreduce: A simulator for modeling mapreduce framework," in *Multimedia and Ubiquitous Engineering (MUE), 2011 5th FTRA International Conference on*. IEEE, 2011, pp. 277–282.

[33] X. Zeng, S. K. Garg, P. Strazdins, P. P. Jayaraman, D. Georgakopoulos, and R. Ranjan, "Iotsim: A simulator for analysing iot applications," *Journal of Systems Architecture*, vol. 72, pp. 93–107, 2017.

[34] Z. Ren, Z. Liu, X. Xu, J. Wan, W. Shi, and M. Zhou, "Waxelephant: A realistic hadoop simulator for parameters tuning and scalability analysis," in *ChinaGrid Annual Conference (ChinaGrid), 2012 Seventh*. IEEE, 2012, pp. 9–16.

[35] S. Hammoud, M. Li, Y. Liu, N. K. Alham, and Z. Liu, "Mrsim: A discrete event based mapreduce simulator," in *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, vol. 6. IEEE, 2010, pp. 2993–2997.

[36] J. Jung and H. Kim, "Mr-cloudsim: Designing and implementing mapreduce computing model on cloudsim," in *ICT Convergence (ICTC), 2012 International Conference on*. IEEE, 2012, pp. 504–509.

[37] A. C. Murthy, "Mumak: Map-reduce simulator," *MAPREDUCE-728, Apache JIRA*, 2009.

[38] R. P. Kennedy and T. Gopal, "A mr simulator in facilitating cloud computing," *International Journal of Computer Applications*, vol. 72, no. 5, 2013.

[39] G. Wang, A. R. Butt, P. Pandey, and K. Gupta, "A simulation approach to evaluating design decisions in mapreduce setups," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS'09. IEEE International Symposium on*. IEEE, 2009, pp. 1–11.

[40] W. Kolberg, P. D. B. Marcos, J. C. Anjos, A. K. Miyazaki, C. R. Geyer, and L. B. Arantes, "Mrsg–a mapreduce simulator over simgrid," *Parallel Computing*, vol. 39, no. 4-5, pp. 233–244, 2013.

[41] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the hadoop ecosystem," *Journal of Big Data*, vol. 2, no. 1, p. 24, 2015.

[42] K. Chandrasekaran, *Essentials of cloud computing*. Chapman and Hall/CRC, 2014.

[43] F. Tian and K. Chen, "Towards optimal resource provisioning for running mapreduce programs in public clouds," in *Cloud Computing (CLOUD), 2011 IEEE International Conference on*. IEEE, 2011, pp. 155–162.

[44] H. Yang and M. Tate, "A descriptive literature review and classification of cloud computing research." *CAIS*, vol. 31, p. 2, 2012.

[45] A. E. Gencer, D. Bindel, E. G. Sirer, and R. van Renesse, "Configuring distributed computations using response surfaces," in *Proceedings of the 16th Annual Middleware Conference*. ACM, 2015, pp. 235–246.

[46] M. Wasi-Ur-Rahman, N. S. Islam, X. Lu, D. Shankar, and D. K. Panda, "Mr-advisor: A comprehensive tuning tool for advising hpc users to accelerate mapreduce applications on supercomputers," in *Computer Architecture and High Performance Computing (SBAC-PAD), 2016 28th International Symposium on*. IEEE, 2016, pp. 198–205.

[47] H. Yang, Z. Luan, W. Li, and D. Qian, "Mapreduce workload modeling with statistical approach," *Journal of grid computing*, vol. 10, no. 2, pp. 279–310, 2012.

[48] A. Rasooli and D. G. Down, "An adaptive scheduling algorithm for dynamic heterogeneous hadoop systems," in *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*. IBM Corp., 2011, pp. 30–44.

[49] H. Casanova, A. Giersch, A. Legrand, M. Quinson, and F. Suter, "Versatile, scalable, and accurate simulation of distributed applications and platforms," *Journal of Parallel and Distributed Computing*, vol. 74, no. 10, pp. 2899–2917, 2014.

[50] G. Wang, A. R. Butt, H. Monti, and K. Gupta, "Towards synthesizing realistic workload traces for studying the hadoop ecosystem," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*. IEEE, 2011, pp. 400–408.

[51] Z. Bian, K. Wang, Z. Wang, G. Munce, I. Cremer, W. Zhou, Q. Chen, and G. Xu, "Simulating big data clusters for system planning, evaluation, and optimization," in *Parallel Processing (ICPP), 2014 43rd International Conference on*. IEEE, 2014, pp. 391–400.

[52] G. Wang, A. R. Butt, P. Pandey, and K. Gupta, "Using realistic simulation for performance analysis of mapreduce setups," in *Proceedings of the 1st ACM workshop on Large-Scale system and application performance*. ACM, 2009, pp. 19–26.

[53] H. Casanova, A. Legrand, and M. Quinson, "Simgrid: A generic framework for large-scale distributed experiments," in *Computer Modeling and Simulation, 2008. UKSIM 2008. Tenth International Conference on*. IEEE, 2008, pp. 126–131.

[54] G. Kecskemeti, "DISSECT-CF: A simulator to foster energy-aware scheduling in infrastructure clouds," *Simulation Modelling Practice and Theory*, vol. 58, pp. 188 – 218, 2015, Special issue on Cloud Simulation.