

# Enabling Strategies for Big Data Analytics in Hybrid Infrastructures

Julio C. S. Anjos, Kassiano J. Matteussi, Paulo R. R. De Souza, Claudio F. R. Geyer

*Institute of Informatics*

*Federal University of Rio Grande do Sul*

*Porto Alegre, Brazil 91509-900*

Email: {jcsanjos, kjmatteussi, prrsjunior, geyer}@inf.ufrgs.br

Alexandre S. Veith, Gilles Fedak

*INRIA, LIP*

*ENS Lyon, France*

Email: alexandre.veith@ens-lyon.fr, gilles.fedak@inria.fr

Jorge Luis Victória Barbosa

*Applied Computing Graduate Program*

*University of Vale do Rio dos Sinos - São Leopoldo, Brazil*

Email: jbarbosa@unisinobr

**Abstract**—A huge volume of data is produced every day by social networks (e.g. Facebook, Instagram, Whatsapp, etc.), sensors, mobile devices and other applications. Although the Cloud computing scenario has grown rapidly in recent years, it still suffers from a lack of the kind of standardization that involves the resource management for Big Data applications, such as the case of MapReduce. In this context, the users face a big challenge in attempting to understand the requirements of the application and how to consolidate the resources properly. This scenario raises significant challenges in the different areas: systems, infrastructure, platforms as well as providing several research opportunities in Big Data Analytics. This work proposes the use of hybrid infrastructures such as Cloud and Volunteer Computing for Big Data processing and analysis. In addition, it provides a data distribution model that improves the resource management of Big Data applications in hybrid infrastructures. The results indicate the feasibility of hybrid infrastructures since it supports the reproducibility and predictability of Big Data processing by low and high-scale simulation within Hybrid infrastructures.

**Index Terms**—*Big Data; MapReduce; Hybrid Infrastructures; Distributed System; Data-intensive Computing;*

## I. INTRODUCTION

Mankind is producing an ever-increasing amount of data. According to the National Security Agency (NSA)<sup>1</sup>, by 2020 there will be around 40 Zettabytes (40,000,000 Petabytes) of data that will require processing of some sort. This volume of data requires processing capabilities beyond those that the current IT infrastructure can provide. In addition, the data input can originate from different sources, such as social interaction, scientific research, business activities and government decisions [1].

Although, Cloud Computing (Cloud) has grown rapidly in recent years, it still suffers from a lack of standardization and the availability of management resources [2]. *Private clouds* are used exclusively by a single organization, that is able to keep careful control of its performance, reliability and security, but might have low scalability for Big Data processing

requirements. *Public clouds*, on the other hand, have an infrastructure that is based on a particular Service Level Agreement which provides services and quality assurance requirements with resources regarding processing, storage and bandwidth. The Cloud Service Provider (CSP) manages its own physical resources. This interface might vary depending on the provider, but maintains properties like elasticity, isolation and flexibility [3].

*MapReduce* (MR) [4] is a programming framework proposed by Google that is currently adopted by many large companies and has been employed as a successful solution for data processing and analysis. The most popular open-source implementation of MR is the Hadoop [5], [6], which is used for deploying Hadoop clusters as the basis for several Big Data analytics.

The emerging systems are highly heterogeneous environments with variable structures, where resources can be added or removed if necessary [7]. The hybrid infrastructure in this work can be defined as a mix of public or private Cloud with Volunteer Computing (VC), similar to the environment explored by [8]. This work proposes the use of hybrid infrastructures such as Cloud and Volunteer Computing for Big Data processing and analysis. In addition, the work provides a data distribution model to improve resource management for Big Data applications in hybrid infrastructures. Further, a new platform for the *hybrid infrastructures*, proposed in previous work [9], called SMART, enables the deployment of VC and Cloud. Besides, a simulator called BIGhybrid [10] was designed by the features required for this infrastructure. It can be considered as a toolkit for Big Data analytics. Hence, a set of strategies will be defined and evaluated for the use of VC and Cloud in hybrid infrastructures.

This work is structured as follows. Section II carries out a review of related works. Section III describes the proposed model and its structure. In addition, the strategies for data and task distribution are discussed. Section IV describes the methodology, the experiments and the obtained results. The conclusions and suggestions for future work are outlined in

<sup>1</sup>The Next Wave, Vol. 20, No. 4, November 2014

## II. RELATED WORK

This section aims to examine the infrastructure initiatives for implementing Big Data in homogeneous and heterogeneous environments. Some of these initiatives lead to hybrid infrastructures in private and public Cloud. However, it is very unusual to find an evaluation of hybrid environments which include a Volunteer Computing platform, in the context to Big Data and particularly that of Geo-distributed environments. Thus, the strategies adopted for different Big Data platforms are summarized and evaluated regarding the related work, which is closer to the presented hybrid model.

### A. MapReduce in a Heterogeneous Environment

The *MapReduce* model was originally conceived for large homogeneous cluster environments. As a result, some simplifications were made to the model aiming to optimizing the task distribution. However, these adjustments may entail system degradation in heterogeneous environments. The work of Zaharia *et al.* [11] was the first study to detect these problems. Their study points out that there are some concerns over the changes of the *MapReduce* model since it may lead to an excessive number of speculative tasks. To overcome this problem, the authors proposed LATE (*Longest Approximate Time to End*), which is a new task scheduler. Although LATE does not completely avoid speculative tasks, it considerably reduces performance degradation in heterogeneous environments.

Some works focus on adapting *MapReduce* algorithms to volatile and heterogeneous environments. One of the first proposals was the MOON project (*MapReduce On Opportunistic eNvironments*) [12], a hybrid model for voluntary computing which assumes that the system consists of volatile and reliable non-volatile machines. The aim of the authors was to investigate whether there was a hybrid design which would be able to provide a high degree of data availability in volatile environments. Their solution involves applying the LATE algorithm and loss of data from the volatile machines is overcome by replicating data in reliable machines.

BitDew-MapReduce (BitDew-MR) [13] is a *MapReduce* implementation adapted to a volatile environment, that has already been combined with Cloud like a hybrid infrastructure to improve performance and reduce costs through the bag-of-tasks application. The fault-tolerance system needs a synchronization schema, as pointed out by [14], where transient and permanent failures can be handled. A barrier-free computation is implemented to mitigate the host churn behavior [13].

*MRA++ (MapReduce with Adapted Algorithms for Heterogeneous Environments)* adapted *MapReduce* algorithms to heterogeneous environments with the aim of addressing the main problems originating from the simplification of the *MapReduce* model [15]. Thus, the developed algorithms allow the use of data-intensive applications in large-scale environments through the use of the Internet. The strategy adopted in this work is to examine three areas of the *MapReduce* implementation: grouping, data distribution and task scheduling.

The data is distributed in accordance with the heterogeneity of the machines to prevent a large increase of execution time. The machines are grouped according to their computational capabilities.

### B. MapReduce across Multiple Clouds

Organizations are increasingly relying on an infrastructure from multiple providers as a means of increasing fault tolerance and avoiding provider lock-in. When multiple Clouds (hereafter also described as *Multi-Cloud*) are involved, the application deployment becomes complex. A *Multi-Cloud* infrastructure contains various configuration choices and can change its requirements and workloads dynamically at the time of execution. In this context, multiple cloud infrastructures, like clouds in heterogeneous environments, need different levels of configuration, such as the operational system, service containers and configuration capabilities [16].

The study of Tudoran *et al.* [17] argues that there are two methods for modeling complex infrastructures. The *analytical model* uses low-level details like workloads and characterizes their ability to predict performance. The *sampling method* as an active approach which does not require any previous knowledge of the infrastructure of the users in public clouds. Because of this, the authors introduce the GEO-DMS a sample-based category for modeling that monitors the environment through agents. The model registers the correlation between performance (execution time) and cost-effectiveness (finance), and imposes budgetary constraints in the interests of safety.

The system used for the geo-distributed data analytics in the work of Ji and Li [18] involves adopting a centralized approach where a significant amount of bandwidth is consumed, which leads to a poor performance. In addition, this approach must be subject to new constraints, such as concerns about privacy. The distributed execution is a strategy that transfers computations to local data centers and then aggregates the intermediate results for further processing. Thus, this execution method can lead to low latency since it involves a distributed execution, although several approaches fail to take account of the constraints in data movement.

### C. Big Data in Hybrid Systems

In a *Hybrid Cloud* environment, the data centers are interconnected by means of slow links. The data is moved from the private to public Cloud when a new VM allocation is required to improve a task performance. The data locality and data movement are a challenge for accelerating iterative *MapReduce* in *Hybrid Clouds* because iterative applications reuse invariant input data [19]. Furthermore, since the extra resources represent an additional cost for data movement, the trade-off between performance gains and benefits must be evaluated.

These issues are evaluated in the work by Clement *et al.* [20] which addresses iterative *MapReduce* problems in a Hybrid IaaS Cloud environment. The authors argue that it is essential to improve data locality in a hybrid Cloud environment. The aim of the strategy is to extend the original fault-tolerance

mechanism of HDFS and deploy data replicas from an on-premise VM in a private Cloud to another allocated off-premise VM in a public Cloud as if it were an external rack in the HDFS.

In the work of Tang *et al.* [8] the authors implement a layer to create a hybrid distributed file system (*HybridDFS*) with reliable data storage clusters in Cloud and unreliable data storage in desktop grid. Each data chunk in volatile nodes has at least one replica distributed for different volunteer PCs or cluster nodes. This means, it is possible to check the feasibility of using hybrid infrastructures such as Cloud and VC.

A next generation data processing engine has been designed that is based on *Lambda Architecture* [21]. This architecture makes it possible to build Big Data systems as layers that can satisfy the properties of getting low latency readings and updates without impairing the robustness of the system.

The Apache Flink, previously called Stratosphere, is a data analytic framework that follows the *Lambda Architecture* and enables the extraction, analysis and integration of heterogeneous datasets [22]. Flink has a flexible pipeline that enables several map-reduce and extended functions.

Although there have been several works that address this issue, there are still many open-ended problems about Big Data in Cloud environments. In addition, the geographically distributed environment has not been explored yet in hybrid infrastructures. The combination of VC resource allocation with Cloud environments for data-intensive applications, is a possible line of inquiry that could be explored to allow the available resources to be freely exploited. This work seeks to explore this and define what the reasonable constraints are for the development of a hybrid Infrastructure model that can make it possible to achieve this goal.

### III. THE HYBRID INFRASTRUCTURE MODEL

The Cloud infrastructure is comprised of providers with heterogeneous hardware and the environments may need the specifications of configuration parameters at several levels such as the operating systems, service containers, and network capabilities [16]. When there are many cloud providers with heterogeneous environments and configurations, a hybrid infrastructure often has to use an orchestrator to manage the results and data input from users. This infrastructure enables highly heterogeneous machines to be used.

On the basis of previous work carried out in MR for hybrid infrastructures [10], Figure 1 illustrates the solution proposed here that involves a hybrid model which depicts a *Global Dispatcher* and *Global Aggregator*. This concept can be used as an infrastructure for services that rely on multiple data abstractions. The *Global Dispatcher* that is located outside the cloud, is a layer that handles task assignments and the management of user-provided data. The layer decouples the data storage system and manages policies for splitting and distributing data in accordance with each system. The *Global Aggregator* obtains data output from both systems and merges them to obtain the final dataset.

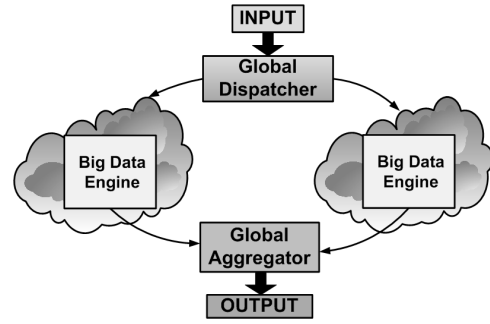


Fig. 1. Hybrid infrastructure from a high-level view

The *Global Dispatcher* handles the task assignments and input data from users. It is a centralized data-driven subsystem that manages remote data localization, and policies for the splitting and distribution of data, in accordance with the needs of each system. The working principle is similar to the publish/subscribe service where the system obtains data and publishes the computing results. This approach is simple, but risks causing a network bottleneck.

Concerning the proposed hybrid infrastructure, we modeled strategies based on off-the-shelves frameworks (e.g., Apache Flink, BitDew-MR, and Hadoop) which establish a feasible platform to evaluate data distribution and task allocation. For instance, BitDew-MR improves computational performance with the use of VC in a hybrid infrastructure. This means the hybrid environment will be simulated through the features from the Flink for the Cloud and BitDew-MR for the Volunteer Computing. The Flink framework and BitDew-MR are considered like subsystems from the hybrid infrastructure. This environment must make it possible to evaluate the best strategies for data distribution and task allocation, and hence make it feasible to establish the platform for Big Data executions.

#### A. Model of strategies for data and task distribution

This section provides a conceptual basis for strategies to use hybrid infrastructures. The evaluation is presented in Section IV. The hybrid infrastructures have several features that are related to distinct resources. So, consider that a user must launch his applications in an infrastructure where himself must choose the best resources for Cloud and Volunteer Computing (VC) for composing a hybrid infrastructure. Thus, each resource can have a different abstract specification, the user will be able to know which resource is faster, through a simulated environment, before choosing the resource, as follows.

A CSP offers resources that represent VM instances of a hardware  $H_{Vm}(i)$ , where  $i \in VM$ . Each VM represents a set of heterogeneous resources that are related to a number of CPU cores, memory, and storage. Each resource, called  $r_c$ , represents a VM and the total amount of the available resources represent a vector  $\hat{R}_c(n_r) = \{r_{c_1}, r_{c_2}, \dots, r_{c_n}\}$  where  $n_r$  is the number of available resources, i.e.,  $n_r = [1, 2, \dots, n]$  resources. The resource allocation defines the computational capacity in Flops/s of the machine set. The resources are

homogeneous from the standpoint of the user, because the CSP only enables access to the virtual instances of machines. However, the hardware of the CSPs might be heterogeneous and in general, the user does not have any knowledge of them.

Hence, an allocated resource may have a poorer performance than a single hardware component, when two or more users are running their applications on virtual machines with the same leased hardware. The performance may have a different time rate of execution for each VM, called  $\rho$ , that is caused by the hardware contention. Its value can be between an interval of  $(0,1)$ . Thus, the resource allocation ( $A_{Vm} \subseteq \hat{R}_C$ ) of a user ( $u$ ) in Cloud can be expressed in the Equation III.1, where  $n \leq n_r$ .

$$A_{Vm}(u) = \sum_{i=1}^n (1 - \rho_i) r_{ci} \quad (\text{III.1})$$

In VC, a donor offers a hardware  $H_l(t)$  that will be dedicated to the processing of tasks for an unspecified time period ( $t$ ), where the number of offers to a volunteer resource pool are defined by  $l = [1, 2, 3, \dots, m]$  resources. Any free hardware is related to heterogeneous resources that represent the number of CPU cores, memory, and storage available within a time period. The pool is a set of stable and unstable nodes. The volunteer resources are a vector  $\hat{R}_v(H_l(t)) = \{h_1 t_1, h_2 t_2, \dots, h_m t_m\}$  where  $t_k$  is the time availability in each machine for the users to make use of a free hardware resource in the resource pool. The user cannot define the availability time for a resource because this issue is handled for the user internally by the fault-tolerance mechanism of the platform in a transparent way. However, the user can define a variable  $\xi$  which determines the number of unstable nodes in the pool. The variable  $\xi$  is within the range of an interval  $(0,1)$ . The VC maintains connections through the Internet. In addition, data transference will depend on the bandwidth  $\beta_l$  between these nodes and the total bandwidth  $\beta_{ch}$  available in the CSP channel. The allocation of VC resources ( $A_{VC} \subseteq \hat{R}_v$ ) of a user ( $u$ ) is represented by Equation III.2.

$$A_{VC}(u) = \sum_{k=1}^m (1 - \xi) h_k \quad (\text{III.2})$$

The performance model has a component that represents the Cloud Allocation with stable VMs and a VC allocation that depends on a rate corresponding to  $\xi$  that defines the ratio of the unstable nodes and might be influenced by the available bandwidth ratio. This model shows the effects of the job execution on the data-intensive applications. Choosing the best features of the platform and data distribution for the environment are crucial factors that are required to mitigate these effects. Thus, the performance of an allocation  $P(A_u)$  of a user  $u$  is defined by Equation III.3.

$$P(A_u) = \underbrace{\sum_{i=1}^n (1 - \rho_i) r_{ci}}_{\text{Cloud allocation}} + \underbrace{\sum_{k=1}^m (1 - \xi) h_k}_{\text{VC allocation}} \quad (\text{III.3})$$

The hybrid infrastructure scenario in this study consists of five key components: workers, orchestrators, the dispatcher, aggregators and CSPs. Figure 2 represents these components and shows the data flow for this scenario. The workers are grouped according to their computational capacity to integrate a VC that provides a pool of computational resources for a user in a private mode. This private mode means that no other user can employ the allocated resources until the user himself releases them to the resource pool.

An orchestrator (built in the dispatcher module) receives the data which is assigned as user tasks and transformed into a ‘‘job’’. After this, the dispatcher partitions the data and tasks on the basis of their computational capacities and assigns them to VC and Cloud. In a hybrid infrastructure, each system processes its own data by grouping the intermediate data in a last *Reduce* phase within the Cloud to produce a single result in the aggregation stage. The *Shuffle* occurs in parallel with the *Map* phase. This begins when the *Map* tasks that have been completed achieve at least 5% (a Hadoop standard) and in the case of Cloud, assume that the execution of *Reduce* begins after all the *Map* tasks have been fulfilled.

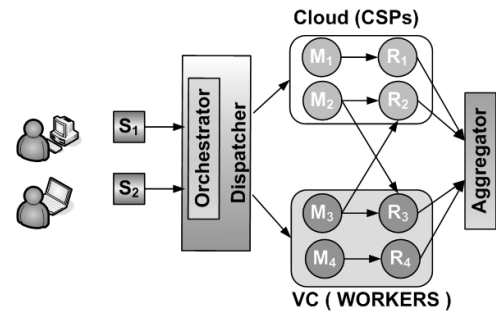


Fig. 2. An example of the data flows in the hybrid infrastructure

The time  $T_i$  (in Equation III.4) of a *MapReduce* job in a hybrid infrastructure (inspired by the equations of Ohnaga *et al.* [23] and Khan *et al.* [24]) consists of a maximum value between the execution time in Cloud ( $T_c$ ) and VC ( $T_d$ ) added to the time of the data integration (called  $T_i$ ). The  $T_i$  time includes the transfer of intermediate data ( $T_e$ ) from the VC to Cloud through an Internet link, and the time of the additional *Reduce* phase to produce a single *Reduce*. Our system is different from other approaches because it includes the integration time to produce a single and consistent result for Big Data processing.

$$T_i = \max(T_c, T_d) + T_i \quad (\text{III.4})$$

The user submits a job ( $J$ ) that will execute a workload across a set of selected resources ( $S$ ) from all of the available resources in Cloud and VC ( $R$ ) (where  $S \subseteq R$ ). When pending tasks occur for a job launch, the system scheduling allocates one free slot per time to each machine in the whole resource pool, and after this, will allocate the other pending tasks in one free slot for each machine and so on. This process sets off execution waves ( $w_j$ ) which are limited to the maximum

number of allocated resources since the workload is larger than the computation resources available. These waves are rounds of successive executions  $G = \{1, 2, 3, 4, 5, \dots, n\}$  that continue until all the  $n$  tasks have been completed. The execution waves are defined as follows  $w_j \subseteq G \subseteq J$ .

The concurrent tasks define execution spaces, called slots, in each machine and the number of free slots determines the amount of parallel work. As discussed previously, the *Shuffle* phase occurs at the same time as the *Map* phase if there is more than one execution wave. Thus, the time spent on data transfer (called  $T_S$ ) must be reduced to the max time of the last *Map* phase in the last wave. Therefore, the sum of the max time for the *Map* and *Reduce* phase in each wave is added to the  $T_S$  time.

The strategy of data distribution, in the Global Dispatcher module seeks to maximize the use of the VC resources, while at the same time, reducing the Cloud allocation resources. This is carried out without increasing the execution time when compared with a single Cloud implementation and the data-split will include the execution waves in the dispatcher to achieve this aim.

The *MapReduce* execution occurs in waves because the machines begin the tasks and finish almost at the same time. Then, the total resources ( $R_{Tc}$ ) are the sum of Vm instances ( $Vm_c$ ) multiplied by the *Map* free slots of a machine ( $\tau_{M_c}$ ). Slots represent the simultaneous execution tasks on the machine. The Equation III.5 introduces the  $\Phi$  relation that is the result of the total workload in Cloud ( $W_c$ ) (equivalent to chunk numbers) divided by the available resources in Cloud ( $R_c$ ). When  $\Phi_C = 1$ , then all the resources are busy, and there is a full execution wave. That is,  $\Phi$  relation summarizes the workload for each infrastructure.

$$\Phi_C = \frac{W_c}{R_c}, \text{ where } R_c = \sum_i^M Vm_c(i) \text{ and } R_c \subseteq R_{Tc} \quad (\text{III.5})$$

In finding the best data distribution, it is necessary to take half of the machines to Cloud and VC. The job in hybrid systems has two executions and to achieve a good load balance the  $\Phi_{Ch}$  relation in Cloud must be balanced to  $\Phi_{VC}$  relation. The best strategy is to use Equation III.5 to find the load balancing. Equation III.6 reflects the  $\Phi$  relation in Cloud hybrid and VC, where  $Ck_{Ch}$  and  $Ck_{VC}$  are the chunk size in Cloud hybrid and VC respectively, and  $Ck_c$  the chunk size for a single execution in Cloud.

$$\Phi_{Ch} = \frac{Ck_{Ch}}{Ck_c} * \frac{W_{Ch}}{\sum_i^M Vm_c(i)} \Phi_{VC} = \frac{Ck_{VC}}{Ck_c} * \frac{W_{VC}}{\sum_i^M M_{VC}(i)} \quad (\text{III.6})$$

Finally, the best load balance is achieved when Equation III.7 is verified, this relation defines the data distribution related to the number of Cloud and VC resources which is the goal of the *data load balance* strategy.

$$\Phi_{Ch} \leq \Phi_C \iff \Phi_{VC} \leq \Phi_C \quad (\text{III.7})$$

## IV. EVALUATION

This section provides the evaluation of data distribution model strategy for Big Data processing and analysis in hybrid infrastructures.

The strategies seek a minimal rate  $\xi$  to define the ratio between VC and Cloud nodes, also the data load balance to improve the computation resource allocation in a hybrid infrastructure. The workloads are real-world executions that have their origin in Big Data applications of companies like Yahoo (YH) and Facebook (FB) [25]. The YH and FB clusters have 2,000 and 3,000 machines respectively. Although the type of jobs at Facebook can change significantly from one year to another, the purpose of this scenario is to cover real-world environments. On the basis of this scenario, around 1,800 simulated experiments were conducted with the BIGHybrid simulator with the aims of finding compatible low and large-scale patterns in hybrid infrastructures.

### A. Simulated Experiments in Hybrid Infrastructures

This subsection describes a simulated environment setup to validate strategies in hybrid infrastructures. The evaluation of the strategies for the use of hybrid infrastructures in Big Data is divided into low-scale, middle-scale and high-scale experiments.

The computational capacity of the processors in the simulated experiments is equivalent to an Intel Xeon E5506 - 2 Cores, 4M Cache, 2.13 GHz  $\approx$  5 GFlops and the computational capacity in a volatility environment represents a distributed value between 4 to 6 GFlops, for all of the experiments. The configuration is similar to what was found in Yahoo and Facebook by the evaluation made by Yanpei [25]. For analytical purposes, the computational consumption is defined by workloads of 64, 32 and 16 MB chunk size. The network, workload and the machines numbers vary in each experiment. The number of the *Reduce* tasks is equal to twice the machine number.

The first experiment has more tasks than resources ( $T > R$ ) with a workload of 512 tasks (chunk size 64 MB) and 128 machines. The hardware and task configurations are the same as in the earlier experiment. The aim of this experiment is to consolidate the observations with regard to the relation of data distribution between Cloud and VC. Figure 3 shows eight executions ranging from **A** to **H** at the top of the chart. The chunk size and input data are described in the graph for each execution for Cloud (C) and VC. The red line indicates the time for a single Cloud deployment, equivalent to 1,232 seconds. The concurrent task execution in the y-axis is measured in seconds, and the number of the machines for Cloud and VC (Cloud/VC) in the x-axis is measured in units. The experiment shows that the data distribution for this scenario achieves the best load balancing when Equation III.7 is true. The **A**, **B**, **C** and **D** executions shown in Figure 3 have a performance that is acceptable in this case.

However, when there is more resources than tasks ( $R > T$ ) a lower limit ( $\Phi_C = 1$ ) is established, this makes sense if we imagine that all tasks are running in a single wave. Table I

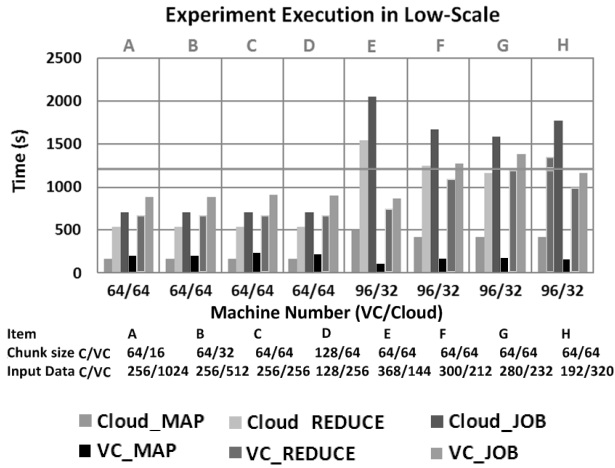


Fig. 3. Low Scale - 128 Machines - 10 Mbps - Tasks > Resources

shows that is possible to establish an upper limit for the  $\Phi$  relation for Cloud and VC which is independent of the time execution and has a value of  $\Phi \leq \Phi_C$ . Thus, the relationship for the resource distribution can be summarized as  $1 \leq \Phi \leq \Phi_C$ . This relation enables the machine quantities to be defined in a hybrid infrastructure through an initial simulation of Cloud execution and then to determine the data distribution for the hybrid infrastructure. This means both load balance from the Cloud and Volunteer Computing are bound by the  $\Phi$  relation.

TABLE I  
THE  $\Phi$  RELATIONSHIP FOR DATA DISTRIBUTION IN HYBRID INFRASTRUCTURES

Relation	Cloud	A	B	C	D	E	F	G	H
$\Phi_{Ch}/\Phi_{VC}$		4/4	4/4	4/4	4/4	4/5	3/6	3/7	3/7
$\Phi_{Ch}/\Phi_{VC}$	4								

Figure 4 shows the impact of volatility on performance for the same execution profile of Figure 3. The letters (a) and (b) correspond to the A and B executions in Figure 3 respectively. The y-axis measures the time in seconds and the x-axis represents the percentage of volatile machines. The first time execution on the left (for all the Figures) represents an implementation without volatility, for comparative purposes, which is the basis of the incremental volatility nodes made in the offsets of 5% to 35%. The experiments show the behavior of the task executions with a networking of 10 Mbps bandwidth. The line red represents the execution time of 1,232 seconds in a single Cloud.

The impact of volatility is sensitive for tasks with a chunk size ranging from 32 to 64 MB due to the overhead of data copy when a machine has a shutdown. The low bandwidth of the network affects the data transfers considerably. The experiments suggest there is an operational flexibility in volatile environments when 20% to 25% of machines have a shutdown without any serious degradation of performance. The chunk sizes for this scenario is 32. However, losing 1/4 of machines in VC can produce a high latency in a slow Internet link. The VC environments can remain relatively stable in some

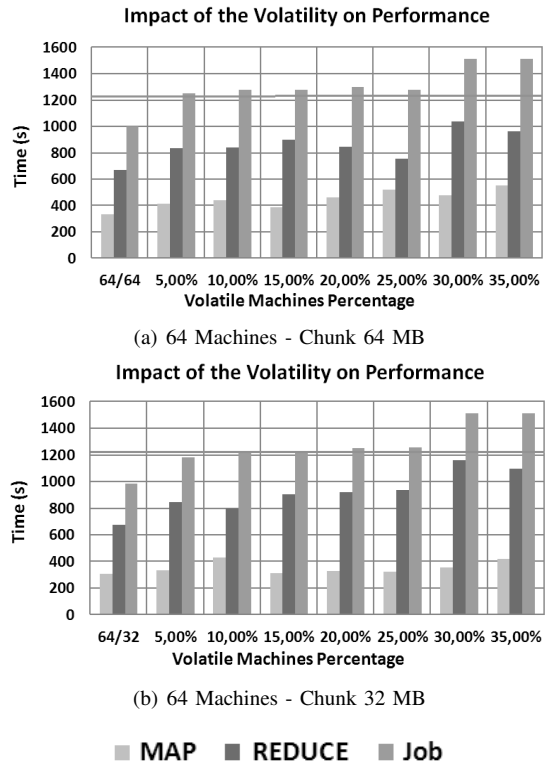


Fig. 4. Impact of Volatility on Performance - 10 Mbps - Tasks > Resources

scenarios but do not have a behavior that is easily predictable. Another factor is that is necessary to evaluate the strategy of the mechanism to improve the performance of Big Data in a hybrid infrastructure, as discussed earlier. In a volatile environment, the machines may have an overhead with a data copy to rebuild replicas, since they might experience long timeout periods, which can allow for the Fault-Tolerance Mechanism (FTM) to take account of false-negatives in the execution behavior. The FTM was detailed in previous work [10].

Figure 5 shows several VC job executions with the aim of determining the behavior in a high-scale and Table II shows the number of machines and workloads. The executions run with a 64 MB chunk size. The experiment in a single Cloud runs in 901 seconds on 2000 machines and a workload of 9088 chunks. The red line in the Figure 5 indicates the time of this execution in a high-scale. The y-axis measures the parallel tasks in a VC environment in seconds for each execution and the x-axis measures the bandwidth. The different jobs are represented with colors, as indicated in the caption. The different jobs are represented with colors as indicated in the caption of the Figure, and each Job shows its  $\Phi$  relation to Cloud and VC respectively. The  $\Phi_{VC}$  relation (VC relation) is maintained according to the earlier estimate.

The experiment in high scale shows that the behavior is different between operations of medium and low-scale. This can be explained by the large number of machines in the network and the administrative overhead needed to manage the

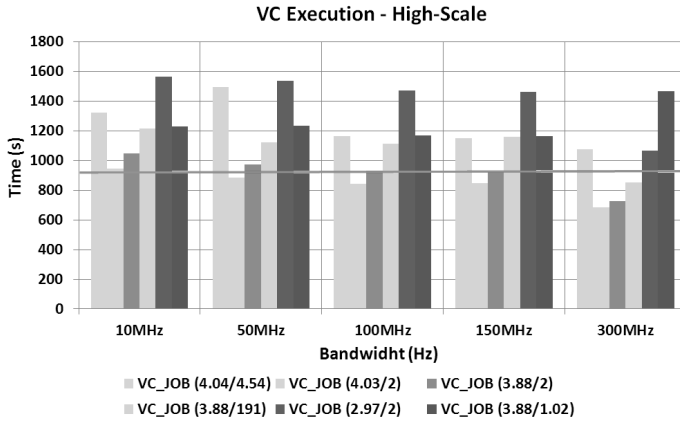


Fig. 5. Simulation in High-Scale (2,000 machines)

TABLE II  
RELATION WORKLOAD VS. NUMBER OF MACHINES

Relation ( $\Phi_{Ch}/\Phi_{VC}$ )	#Chunks (C/VC)	Machine (C/VC)
VC_JOB (4.04/4.54)	7270/1818	1800/400
VC_JOB (4.03/2)	7270/1818	1800/909
VC_JOB (3.88/2)	7000/2088	1800/1044
VC_JOB (3.88/1.91)	7000/2088	1800/1800
VC_JOB (3.88/1.02)	7000/2088	1800/2044
VC_JOB (2.97/2)	5000/4088	1680/2044

data transference in the Internet. In some case, the execution is possible from 50 MHz to 300 MHz bandwidth, but the best performance occurs with 300 MHz bandwidth. However, a thorough cost analysis can determine other data distributions where the borderline for the Cloud execution may be exceeded without any loss of quality in the solution. The results indicate a cost reduction ranging from 10% to 50% in the worst and best case scenarios respectively, in comparison with a single Cloud execution.

This scenario demonstrates that a slight variation in workload can lead to behavioral changes in the VC environment. Thus, hybrid infrastructure with high workloads require orchestrators to control data transfers and the task synchronization.

### B. Discussion

The results of the simulated experiments established strategies for using hybrid infrastructures based on behavior patterns of job executions in Cloud and VC.

Since an optimal allocation is difficult to achieve, strategies to obtain an approximation can be accepted, as argued by Mashayekhy *et.al.* [26]. The  $\Phi_{Ch}$  and  $\Phi_{VC}$  parameters used to find Cloud and VC resources, establish a suitable number of machines to achieve an acceptable performance and good approximation. As well as this, these settings help inexperienced users to locate the number of Cloud and VC machines without the need for a previous knowledge of the CSP infrastructure, which can be considered to be one of the benefits of this study.

The relation between workload, number of machines and load balance can be viewed as a significant contribution to

providing a data load balance and reducing the data transfers between machines from 39.1% to 57.14% in the worst and best case scenarios, respectively. These values are compatible with the work of Tudoran *et.al.* [17] who achieved a reduction of 50% with a relative error from 10% to 15%. The relation where  $1 \leq \Phi \leq \Phi_C$  is particularly stable in all the scales regardless of bandwidth.

## V. CONCLUSIONS

Cloud computing has changed the way applications are developed and ported in geographically distributed infrastructures. Clouds can offer several services, by providing dynamic features as well as being a suitable scenario for building Big Data applications and their range of components. However, the resource management for intensive applications, such as *MapReduce* applications, remains an opened topic due to the wide use of resources at run time [27], [28]. This problem can be even worse if the users do not manage their application resources properly, and can leading to problems of cost effectiveness.

The proposed evaluation found behavioral patterns which enable their deployment in hybrid infrastructures (in low and high scale) and established a set of strategies such as: i) The relationship between the execution time and the number of volatile nodes; ii) The relationship between resource distribution in Cloud and VC; iii) The relationship  $\Phi$  between Cloud and VC; iv) The relationship between the load balance and number of machines; The results are summarized in Table III and suggest that the hybrid infrastructures have an operational continuity in an environment with up to 25% of unstable nodes in the worst case scenario without a loss of performance and with 3 replicas which is the Hadoop standard.

The  $\Phi_{Ch}$  and  $\Phi_{VC}$  parameters designed to find Cloud and VC resources, established that there were a number of suitable machines that could achieve an acceptable performance. The  $\Phi$  relationship between Cloud and VC can help the users to find resources that are adapted to their workloads.

TABLE III  
STRATEGIES AND ACHIEVED AIMS

Parameters	Strategies	Aim
Volunteer hosts	$5\% < \xi < 25\%$	Achieved
Cloud resources	$\Phi_{Ch} = \frac{Ck_{Ch}}{Ck_c} * \frac{W_{Ch}}{\sum_i^M V_{m_c}(i)}$	Achieved
VC resources	$\Phi_{VC} = \frac{Ck_{VC}}{Ck_c} * \frac{W_{VC}}{\sum_i^M M_{VC}(i)}$	Achieved
$\Phi$ relation between Cloud and VC	$1 \leq \Phi \leq \Phi_C$	Achieved
Data load balance	$\Phi_{Ch} \leq \Phi_C \iff \Phi_{VC} \leq \Phi_C$	Achieved

In future works, several activities are needed to build the SMART platform in a real-world environment. First, it is necessary to complete the cost model analysis and the I/O interference of the strategies defined in this work. The mechanism of the decision engine can assist some strategies of this work when deployed as a decision heuristic model.

## ACKNOWLEDGMENT

This work was partly sponsored by CAPES, PROPESQ-UFRGS and by projects: "GREEN-CLOUD: Computação em Cloud com Computação Sustentável" (#16/2551-0000 488-9) and SmartSent (#17/2551-0001 195-3) from FAPERGS and CNPq Brazil, program PRONEX 12/2014. The experiments in this paper were carried out using the Grid'5000 experimental testbed, and conducted under the INRIA ALADDIN development scheme with the support of CNRS, RENATER and several Universities as well as other funding bodies (see <https://www.grid5000.fr>).

## REFERENCES

- [1] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *International Journal of Information Sciences*, vol. 275, pp. 314–347, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025514000346>
- [2] A. N. Toosi, R. N. Calheiros, and R. Buyya, "Interconnected cloud computing environments: Challenges, taxonomy, and survey," *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, pp. 7:1–7:47, May 2014.
- [3] C. Rista, D. Griebler, C. A. F. Maron, and L. G. Fernandes, in *Proceedings of the International Conference on High Performance Computing Simulation (HPCS)*, title=Improving the Network Performance of a Container-Based Cloud Environment for Hadoop Systems, year=2017, volume=, number=, pages=619-626, doi=10.1109/HPCS.2017.97, ISSN=, month=July.
- [4] J. Dean and S. Ghemawat, "MapReduce - a flexible data processing tool," *Journal of the Communications of the ACM (CACM)*, vol. 53, no. 1, pp. 72–77, Jan. 2010.
- [5] T. White, *Hadoop - The Definitive Guide*, 3rd ed. O'Reilly Media, Inc., 2012, vol. 1.
- [6] I. H. A. Meddah and K. Belkadi, "Parallel distributed patterns mining using hadoop mapreduce framework," *International Journal of Grid and High Performance Computing (IJGHPC)*, vol. 9, no. 2, pp. 70–85, 2017.
- [7] M.-A. Vasile, F. Pop, R.-I. Tutueanu, V. Cristea, and J. Kolodziej, "Resource-aware hybrid scheduling algorithm in heterogeneous distributed computing," *Journal of Future Generation Computer Systems (FGCS)*, vol. 51, pp. 61–71, 2015.
- [8] B. Tang, H. He, and G. Fedak, "HybridMR: a new approach for hybrid MapReduce combining desktop grid and cloud infrastructures," *Journal of Concurrency and Computation: Practice and Experience (CCPE)*, vol. 27, no. 16, pp. 4140–4155, May 2015.
- [9] J. C. S. Anjos, M. D. Assuncao, J. Bez, C. F. R. Geyer, E. P. de Freitas, A. Carissimi, J. P. C. L. Costa, G. Fedak, F. Freitag, V. Markl, P. Fergus, and R. Pereira, "SMART: An Application Framework for Real Time Big Data Analysis on Heterogeneous Cloud Environments," in *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, Oct 2015, pp. 199–206.
- [10] J. C. S. Anjos, G. Fedak, and C. F. R. Geyer, "BIGHybrid: a simulator for mapreduce applications in hybrid distributed infrastructures validated with the grid5000 experimental platform," *Journal of Concurrency and Computation: Practice and Experience (CCPE)*, vol. 28, no. 8, pp. 2416–2439, June 2016, cpe.3665.
- [11] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica, "Improving MapReduce performance in heterogeneous environments," in *Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation (OSDI)*. USENIX Association, 2008, pp. 29–42.
- [12] H. Lin, X. Ma, J. Archuleta, W.-c. Feng, M. Gardner, and Z. Zhang, "MOON: MapReduce on opportunistic environments," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC)*, June 2010, pp. 95–106.
- [13] L. Lu, H. Jin, X. Shi, and G. Fedak, "Assessing MapReduce for internet computing: A comparison of Hadoop and BitDew-MapReduce," in *Proceedings of the 13th ACM/IEEE Int. Conference on Grid Computing*, 2012, pp. 76–84.
- [14] B. Tang and G. Fedak, "Analysis of data reliability tradeoffs in hybrid distributed storage systems," in *Proceedings of the 26th IEEE International Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW)*, May 2012, pp. 1546–1555.
- [15] J. C. Anjos, I. Carrera, W. Kolberg, A. L. Tibola, L. B. Arantes, and C. R. Geyer, "MRA++: Scheduling and data placement on MapReduce for heterogeneous environments," *Journal of Future Generation Computer Systems (FGCS)*, vol. 42, pp. 22–35, Jan 2015.
- [16] D.-H. Le, H.-L. Truong, G. Copil, S. Nastic, and S. Dustdar, "SALSA: A framework for dynamic configuration of cloud services," in *Proceedings of the 6th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, Dec. 2014, pp. 146–153.
- [17] R. Tudoran, A. Costan, R. Wang, L. Bouge, and G. Antoniu, "Bridging data in the clouds: An environment-aware system for geographically distributed data transfers," in *Proceedings of the 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, May 2014, pp. 92–101.
- [18] S. Ji and B. Li, "Wide area analytics for geographically distributed datacenters," *Journal of Tsinghua Science and Technology (Tsinghua Sci Technol)*, vol. 21, no. 2, pp. 125–135, April 2016.
- [19] A. M. Chacko, A. Gupta, S. Madhu, and S. M. Kumar, "Improving execution speed of incremental runs of mapreduce using provenance," *International Journal of Big Data Intelligence*, vol. 4, no. 3, pp. 186–194, 2017.
- [20] F. J. Clemente-Castelló, B. Nicolae, K. Katrinis, M. M. Rafique, R. Mayo, J. C. Fernández, and D. Loreti, "Enabling big data analytics in the hybrid cloud using iterative MapReduce," in *Proceedings of the 8th IEEE/ACM International Conference Utility and Cloud Computing (UCC)*, Dec. 2015, pp. 290–299.
- [21] C. Wu, R. Buyya, and K. Ramamohanarao, "Book chapter: Big Data analytics = machine learning + cloud computing," in *Big Data: Principles and Paradigms*, R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, Eds. Morgan Kaufmann, 2016, pp. 3–38.
- [22] A. Alexandrov, R. Bergmann, S. Ewen, J. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. J. Sax, S. Schelter, M. Höger, K. Tzoumas, and D. Warneke, "The stratosphere platform for big data analytics," *International Journal on Very Large Data Bases (VLDB)*, vol. 23, no. 6, pp. 939–964, Dec. 2014.
- [23] H. Ohnaga, K. Aida, and O. Abdul-Rahman, "Performance of Hadoop application on hybrid cloud," in *Proceedings of the International Conference on Cloud Computing Research and Innovation (ICCCRI)*, Oct. 2015, pp. 130–138.
- [24] M. Khan, Y. Jin, M. Li, Y. Xiang, and C. Jiang, "Hadoop performance modeling for job estimation and resource provisioning," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 27, no. 2, pp. 441–454, Feb. 2016.
- [25] Y. Chen, "Workload-driven design and evaluation of large-scale data-centric systems," Ph.D. dissertation, Electrical Engineering and Computer Sciences, May 2012. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/Eecs-2012-73.html>
- [26] L. Mashayekhy, M. Nejad, and D. Grosu, "A PTAS mechanism for provisioning and allocation of heterogeneous cloud resources," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 26, no. 9, pp. 2386–2399, 2015.
- [27] M. Xavier, I. De Oliveira, F. Rossi, R. Dos Passos, K. Matteussi, and C. De Rose, "A performance isolation analysis of disk-intensive workloads on container-based clouds," in *Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, March 2015, pp. 253–260.
- [28] M. G. Xavier, K. J. Matteussi, F. Lorenzo, and C. A. De Rose, "Understanding performance interference in multi-tenant cloud databases and web applications," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, Dec. 2016, pp. 2847–2852.