Computer Engineering and Applications Vol. 3, No. 3, September 2014

Data Management Challenges in Cloud Environments

M.M javidi[1], N. Mansouri[2], A. Asadi

Department of Computer Science, Shahid Bahonar University of Kerman, Postal Code 97175-569, Kerman, Iran.
[1] javidi@uk.ac.ir,
[2]najme.mansouri@gmail.com

ABSTRACT

Recently the cloud computing paradigm has been receiving special excitement and attention in the new researches. Cloud computing has the potential to change a large part of the IT activity, making software even more interesting as a service and shaping the way IT hardware is proposed and purchased. Developers with novel ideas for new Internet services no longer require the large capital outlays in hardware to present their service or the human expense to do it. These cloud applications apply large data centers and powerful servers that host Web applications and Web services. This report presents an overview of what cloud computing means, its history along with the advantages and disadvantages. In this paper we describe the problems and opportunities of deploying data management issues on these emerging cloud computing platforms. We study that large scale data analysis jobs, decision support systems, and application specific data marts are more likely to take benefit of cloud computing platforms than operational, transactional database systems.

Keywords: Cloud computing, Taxonomy, Architecture.

1. INTRODUCTION

We live in the data days [1-2]. Today, the continuous increase of computational power has presented an overwhelming flow of data. According to IDC, the size of the digital universe was about 1.8 zettabyte in 2011. The result of this is the appearance of an obvious gap between the amount of data that is being generated and the capacity of old systems to store, process and make the best use of this data. Cloud computing is good solution for this problem in recent years due to its economic advantages [3].

Cloud computing is a hot discussed topic, and many big players of the software activity are entering the development of cloud services[4-6]. In the recent years, the continuous increase of computational power has produced an overwhelming flow of data. Moreover, the main advances in Web technology has made it easy for any user to provide and consume content of any form. This has called for a paradigm shift in the computing architecture and large scale data processing operations. Cloud computing is related with a new paradigm for the provision of computing infrastructure. This paradigm shifts the location of this infrastructure to the network to decrease the costs related with the management of hardware and software resources.

However, with the amount of cloud computing services increasing quickly, the need for a taxonomy framework rises. Table-based comparisons of cloud computing services have been proposed in [7], however, they are typically for commercial use and the degree of detail varies greatly. In [8], a taxonomy has been presented. However, [8] aims to recognize the strengths and weaknesses in current cloud systems, rather than compare existing and future cloud computing services. Further, also the industry has provided white papers describing cloud computing taxonomies, like [9] by Intel Cooperation. Intel's white paper presents five categories and describes possible applications and services that can be offered for each. More distinctive characteristics of these services are not provided.

Based on existing taxonomies, this taxonomy provides more detailed characteristics and hierarchies. Additionally, the taxonomy gives a comprehensive survey of different approaches and operations of deploying data-intensive applications in the cloud which are gaining a lot of momentum in both research and industrial communities. We study the numerous design decisions of each strategy and its suitability to support certain classes of applications and end-users. A discussion of some open issues and future challenges pertaining to scalability, consistency, economical processing of large scale data on the cloud is provided. Several organizations want to consider the possibilities and advantages of cloud computing, but with the amount of cloud computing services increasing fast, the need for a taxonomy rises. This paper presents the advantages and disadvantages of deploying database systems in the cloud environment. We show how the general properties of commercially available cloud computing platforms affect the choice of data management applications to use in the cloud.

The rest of the paper is organized as follows: Section 2 presents a Cloud computing overview. Section 3 introduces advantages and disadvantages of cloud computing. Section 4 presents the data management in Cloud. We show and analyze goals and challenges in cloud data management in section 5. Finally, section 6 concludes the paper and suggests some directions for future work.

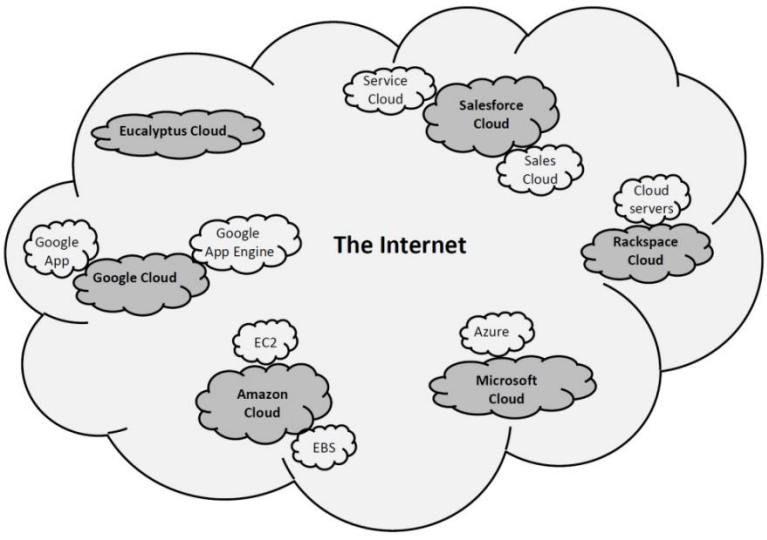

FIGURE 1. Cloud computing services.

2. CLOUD COMPUTING OVERVIEW

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

A cloud can be presented as a scalable infrastructure that supports and interconnects various cloud computing services; see Fig. 1. The cloud itself consists "of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s)" [10]. The clients that are the users of the cloud computing services use their home or work computer or any other Internet-enabled device to connect and apply the cloud computing services.

The important features that distinguish cloud computing from traditional computing solutions have been introduced in [11-16] and generally comprise the following:

- Underlying infrastructure and software is abstracted and presented as a service.
- Build on a scalable and flexible infrastructure.
- Presents on-demand service provisioning and quality of service (QoS) guarantees.
- Pay for apply of computing resources without up-front commitment by cloud users.
- Shared and multitenant.
- Accessible over the Internet by any device.

2.1. Service Models

This section shows the levels of abstraction (Figure 2) which are referred to as service models:

1) Infrastructure as a Service (IaaS): Provision resources like servers (often in the form of virtual machines), network bandwidth, storage, and related things necessary to build an application environment from scratch (e.g. Amazon EC210).

2) Platform as a Service (PaaS): presents a higher-level environment where users can deploy customized applications (e.g. Microsoft Azure11, Google AppEngine12). The maintenance, load-balancing and scale-out of the platform are performed by the service provider and the developer can focus on the fundamental functionalities of his application.

3) Software as a Service (SaaS): provides special-purpose software that are made available through the Internet (e.g. Sales- Force13). So it does not need each end-user to manually download, install, deploy, execute or apply the software applications on their own computing environments.

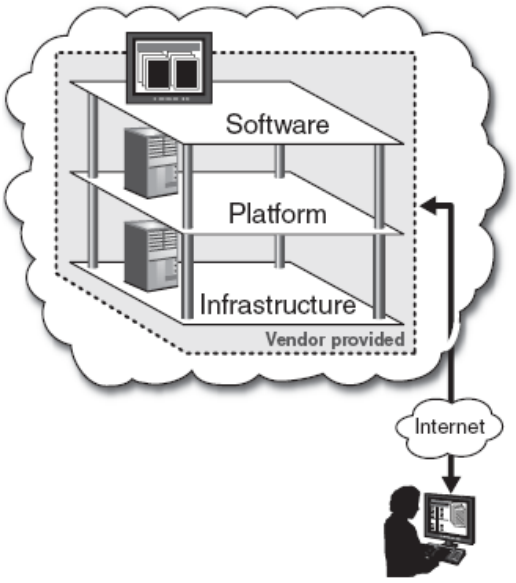

FIGURE 2. Models of Cloud Services.

2.2. Cloud Computing: Evolution over the Years

In 1999 Sales force started providing applications to users using a simple website. The applications were presented to enterprises over the Internet, and this way the dream of computing sold as utility started being reality. Although the service was attractive, some more time would proceed until it would become widespread.

In 2002 Amazon started Amazon Web Services, delivering services like storage, computation and even human intelligence. However, only starting with the launch of the Elastic Compute Cloud in 2006 a really commercial service open to everybody existed.

2009 known as a key turning point in the evolution of cloud computing, by providing the browser based cloud enterprise applications, with the best known being Google Apps.

Of course, all the important players are present in the cloud computing evolution, some earlier, some later. In 2009 Microsoft launched Windows Azure, and organizations like Oracle and HP have all joined the game. This shows that today, cloud computing has become mainstream computing.

In practice, the idea of renting computing power goes back decades to the days when organizations would share space on a single mainframe with big spinning tape drives and it has been visualized that computing facilities will be presented to the general public like a utility [17]. Today, the technology industry has grown to the point where there is now an emerging mass market for this rental approach. Therefore, cloud computing is not a revolutionary new development. However, it is an evolution that has occurred over several years. Figure 3 shows the evolution towards cloud computing in hosting software applications.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

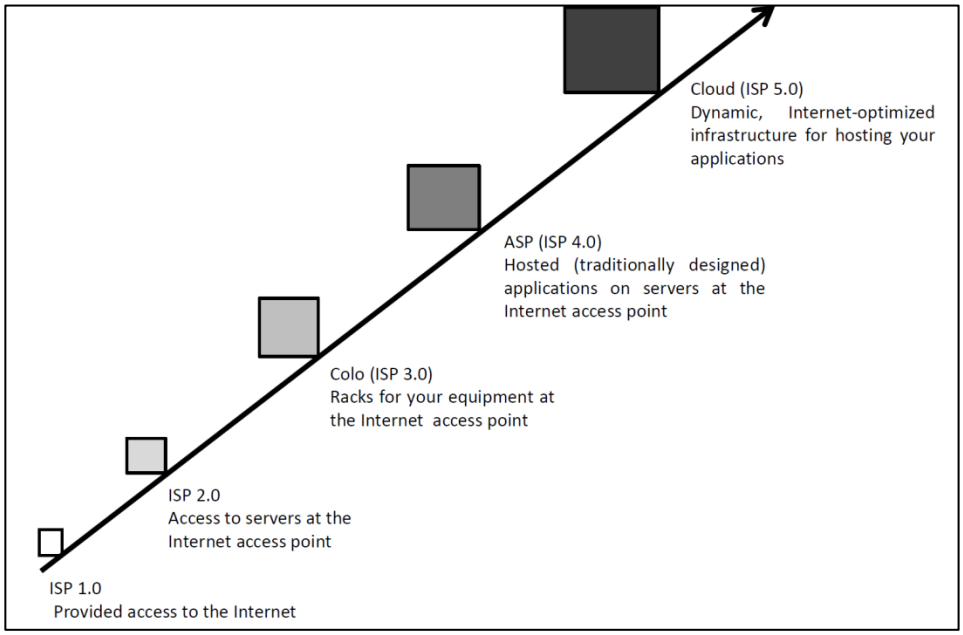

FIGURE 3. The evolution towards cloud computing in hosting software applications.

3. ADVANTAGES AND DISADVANTAGES OF CLOUD COMPUTING

3.1. Advantages

- Reduced Cost: Cloud technology is paid incrementally, reducing organizations cost.
- Increased Storage: Organizations can keep more data than on private computer systems.
- Highly Automated: No longer do IT operators need to worry about checking software up to date.
- Flexibility: Cloud computing provides much more flexibility than past computing strategies.
- More Mobility: workers can get information wherever they are, rather than having to remain at their desks.
- Bonus advantage: In a cloud computing environment resources can be shared between different applications as well as customers resulting in greater use of the resources for a similar energy cost.

3.2. Disadvantages

- Security and privacy: The main concerns about cloud computing environment are security and privacy. Clients might not be comfortable controlling over their data to a third party. This is an even greater problem when it comes to companies that wish to store their critical information on cloud servers situated somewhere in the other part of the world. Privacy is

another concern with cloud servers. After all, the usefulness of a cloud service depends on its reputation, and any sign of a security breach would result in a loss of users and business.

- Getting locked-in with a particular vendor: With a company using the services of a certain cloud vendor under the respective SLA (Service Level Agreement) it's not easy for the company to migrate to another cloud service provider.

The National Institute of Standards and Technology (NIST) presented the following definition of cloud computing: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provided and released with minimal management method or service provider interaction. This cloud model promotes availability." [18].

This description will be applied as the basis for distinguishing cloud computing services and traditional Internet services. Next, three service models are currently being differentiated— Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS). Some previous study considers additional service models, such as Service-as-a-Service, and Data-as-a-Service [19, 20] or Storage-as-a-Service [19], but normally it is possible to group these with the existing three service models [21].

4. DATA MANAGEMENT IN CLOUD

Data management has always been a problem - for individuals, for big businesses and companies, for decentralized enterprises like higher education institutions. The subject of data management is not novel - it started way back as records management, when paper files and folders were the data collection medium of choice. Unfortunately, legacy approaches based on paper often remain in place today for campus information, even when transformed to electronic data formats and processes. Once data management structure is set and operating well, it is ready to take on the new frontier of data management in the cloud.

Therefore a main problem in the Cloud computing environment is the management of very large volumes of data. This is thoroughly independent of the type of resource which is shared in the Cloud – databases are either directly accessible to users as part of the infrastructure, or are hidden behind service interfaces. In the context of data management, QoS guarantees typically encompass a high level of availability. For presenters of Cloud services, this means that data need to be partitioned and replicated across several data centers. Although traditional high throughput OLTP applications are most likely not to become the main applications placed in a Cloud computing environment, replicated data management nevertheless needs to take into account updates which are done on replicated data. Replicated data management in the context of simultaneous updates to different replicas can be provided either by using well-established protocols like Strict Two-Phase Locking (2PL) in combination with Two- Phase Commit (2PC), or by relaxing ACID properties to enhance the overall performance. The latter is used in different today's Cloud environments (e.g., PNUTS).

Today, mainly motivated by the need of applications in eScience where huge amounts of data are produced by specialized simulations and need to be collaboratively accessed, used and analyzed by a large number of researchers around the world, Grid computing has become increasingly popular. The Grid started as a

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

vision to distribute potentially unlimited computing power over the Internet to solve main issues in a distributed way – a first generation of Grids, so-named computational Grids, focused on CPU cycles as resources to be shared. Recent advances in Grid computing want to virtualize several types of resources such as data, instruments, computing nodes, and tools and making them transparently available. Motivated by the success of computational Grids, next generation of Grids, namely Data Grids, has provided as an idea for distributed data management in data-intensive applications. The size of data requested by these applications may be up to petabytes. In the earth observation organizations, for example, data are obtained from satellites, sensors and other data acquisition tools, archived along with metadata, catalogued and validated. According to the report [22], by the year 2010 the earth observation data sets around the world will grow to around 9.000 Terabytes and by the year 2014 to around 14.000 Terabytes. In many applications, Data Grids not only keep raw data generated by tools, but need to take into account also information derived out of these raw data and image interpretations that are periodically produced and potentially concurrently updated by researchers at different locations.

Similar to Cloud-based environments, a high degree of availability of data can only be achieved by means of replication technique across several sites in the Grid. In terms of replication management, most Data Grids have problem from several limitations as they merely deal with files as the replication granularity, do not allow replicated data to be updated, and/or require the manual placement of data files. In the past years, researchers have developed the Re:GRIDiT system that presents advanced data and replication management in the Grid [23, 24]. Re:GRIDiT follows a truly distributed idea to replication management in the Grid by providing together replication management, initially presented for database clusters, and distributed transaction management that does not rely on a global coordinator. In practice, it has been proposed to be independent from any underlying Grid middleware.

In this study, we compare requirements and state of the art in the Data Grid and in Cloud Data Management.

4.1. Distributed data management: Cloud vs. Grid

Data Grids and Cloud Data Management have similar goals. However, the developments of the Grid and of the Cloud have only been loosely coupled for various reasons. First, they both focused on particular user communities: scientific groups (eScience) in case of the Grid vs. the outsourcing of ICT services for commercial users in case of the Cloud. Second, both systems have different sources: the first driver for the Grid has been the High Energy Physics organization, while the proliferation of the Cloud has been dominated by large suppliers of IT services that already had the main computing resources in place and were heading towards a more efficient utilization of their capabilities. Third, the main requirements which have been provided were different. In the Data Grid, initial solutions have focused on the managed sharing of files within Virtual Organizations (VOs). Data management at a granularity finer than files, replication management and updates have only recently been added on the list of main requirements due to new applications. For Cloud-based systems, analytical data management has been considered as the predominant application. However, in the presence of QoS

limitations that need to be met by Cloud service suppliers, data need to be copied across different data centers.

Although the percentage of updates will be rather low in comparison with traditional OLTP settings, Cloud Data Management nevertheless needs to present consistent data management in the presence of conflicting updates.

Table 1 shows the relationship between Cloud Data Management and Data Grids. The table presents that differences between both fields still exist.

TABLE 1.
A Comparison of Cloud Data Management and Data Grids.

	Cloud Data Management	**Data Grids**
Distribution	Few data centers	Many Grid nodes (presumably much larger than the number of data centers in the Cloud)
Environment	Homogeneous resources in data centers	Heterogeneous Grid sites
Operations for Data Access	Usually SQL-based access to relational databases	Distinction between mutable and immutable data, materialized in several sets of access services.
Replication	Needed as a consequence of QoS guarantees (availability); must be transparent to customers	Needed in order to provide a high level of availability; must be transparent to Grid users and developers
Replication Granularity	Fine-grained replication, due to multi-tenancy	Course-grained replication sufficient
Updates	No traditional OLTP workloads, but (concurrent) updates to replicated data need to be provided	First generations considered read-only data; novel eScience applications also demand updates to replicated data
Global Control	Most solutions have some global component which might lead to a single point of failure or performance bottleneck	Most Data Grids consider global replica catalogs
Global Correctness	Relaxation of ACID attributes	Most Data Grids do not support concurrent updates

Also in this section we present the suitability of moving the two largest components of the data management market into the cloud: transactional data management and analytical data management.

4.2. Transactional data management

By “transactional data management”, we mention to the bread-and-butter of the database industry, databases that back banking, airline reservation, and supply chain management applications.

These applications generally rely on the ACID guarantees that databases provide, and tend to be justly write-intensive. We speculate that transactional data management applications are not likely to be used in the cloud, at least in the near future, for the following points:

Transactional data management systems do not generally apply a shared-nothing architecture. The transactional database market is controlled by Oracle, IBM DB2, Microsoft SQL Server, and Sybase [25]. Of these four approaches, neither Microsoft SQL Server nor Sybase can be deployed using a shared-nothing architecture.

IBM proposed a shared-nothing implementation of DB2 in the mid-1990s which is now known as a “Database Partitioning Feature” (DPF) add-on to their flagship product [26], but is presented to help scale analytical applications executing on data warehouses, not transactional data management [27].

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Oracle had no shared-nothing implementation until September 2008 with the release of the Oracle Database Machine that applies a shared-nothing architecture at the storage layer, but again, this implementation is proposed only to be employed for data warehouses [28]. Implementing a transactional database system using a shared-nothing architecture is non-trivial, because data is divided across sites and, in general, transactions cannot be limited to accessing data from a single site.

These outcomes in complex distributed locking and commit protocols, and in data being shipped over the network leading to increased latency and high network bandwidth bottlenecks. Furthermore the key advantage of a shared-nothing architecture is its scalability [29]; however this benefit is less relevant for transactional data processing for which the overwhelming majority of deployments are less than 1 TB in size [30].

It is hard to keep ACID guarantees in the face of data replication over distributed locations. The CAP theorem [31] presents that a shared-data system can only choose at most two out of three features: consistency, availability, and tolerance to partitions. When data is copied over a wide area, this mainly leaves just consistency and availability for a system to choose between. Thus, the 'C' (consistency) part of ACID is generally compromised to yield acceptable system availability.

In order to show a sense of the inherent challenges in building a replicated database over a distributed environment, it is interesting to refer the design approaches of some systems. Amazon's SimpleDB [20] and Yahoo's PNUTS [32] both use shared-nothing databases over a wide-area network, but solve the problems of distributed replication by relaxing the ACID guarantees of the system. In particular, they weaken the consistency model by deploying several forms of eventual/timeline consistency so that all replicas do not have to accord the current value of a kept value.

There are several risks in storing transactional data on an untrusted host. Transactional databases generally keep the complete set of operational data needed to power mission-critical business processes. This data contain the detail at the lowest granularity, and often includes critical information like customer data or credit card numbers. Any increase in potential security breaches or privacy violations is commonly unacceptable. We thus conclude that transactional data management applications are not well suited for cloud environment.

4.3. Analytical data management

By "analytical data management", we mention to applications that request a data store for use in business planning, problem solving, and decision support. Historical data along with data from several operational databases are all generally involved in the analysis. Consequently, the size of analytical data management systems is typically larger than transactional systems. Also analytical systems tend to be read-only, with occasional batch inserts.

Analytical data management systems are well-suited to use in a cloud environment, and will be among the first data management applications to be implemented in the cloud, for the following reasons:

Shared-nothing architecture is a good solution for analytical data management. Teradata, Netezza, Greenplum, DATAllegro, Vertica, and Aster Data all apply a shared-nothing architecture in their analytical DBMS products, with IBM. The ever increasing amount of data involved in data analysis workloads is the first driver

behind the selection of a shared- architecture, as the architecture is mainly believed to scale the best. Also, data analysis workloads usually involve many large scan scans, multidimensional aggregations, and star schema joins, all of which are fairly easy to parallelize across sites in a shared-nothing network. Finally, the few writes in the workload removes the need for complex distributed locking and commit protocols.

ACID guarantees are typically not needed. The few writes in analytical database workloads, along with the fact that it is typically sufficient to do the analysis on a recent snapshot of the data makes the 'A', 'C', and 'I' (atomicity, consistency, and isolation) of ACID easy to obtain.

Specially sensitive data can often be left out of the analysis. In many cases, it is possible to determine the data that would be most damaging should it be accessed by a third party, and either move it out of the analytical data store, include it only after using an anonymization function, or include it only after encrypting it. Also, less granular models of the data can be used instead of the lowest level, most detailed data.

Therefore the properties of the data and workloads of common analytical data management applications are well-suited for cloud environment.

5. CLOUD DATA MANAGEMENT: GOALS AND CHALLENGES

In this section, we present an overview of the important goals and challenges for applying data-intensive computing application in cloud environments.

5.1. Goals

Typically, successful cloud data management systems are proposed to satisfy as much as possible from the following list [32-34]:

Availability: They must be always accessible even on the events where there is a network failure or a whole datacenter has gone offline. Towards this plan, the concept of Communication as a Service (CaaS) appeared to provide such requirements, as well as network security, dynamic provisioning of virtual overlays for traffic isolation or provided bandwidth, guaranteed message delay, communication encryption, and network monitoring. For instance, [35-36] propose different architectural design decisions, protocols and solutions to present QoS communications as a service.

Scalability: They must be able to use huge databases with very high request rates at very low latency.

They should be able to take on new tenants or control growing tenants without much work beyond that of adding more hardware. In particular, the system must be able to dynamically redistribute data to take value of the new hardware.

Elasticity: They must be able to provide changing application requirements in both directions (scaling up or scaling down). Also, the system must be able to wisely reply to these changing requirements and immediately recover to its steady state.

Performance: On public cloud computing systems, pricing is set in a way such that one pays only for what one uses, so the vendor price grows linearly with the requisite storage, network bandwidth, and compute power. Hence, the system performance has an impressive effect on its costs. Thus, efficient system performance is a main requirement to save money.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

Multitenancy: They must be capable to provide many applications on the same hardware and software infrastructure. However, the performance of this tenant must be separated from each another. Adding a new tenant should require little work beyond that of guaranteeing that sufficient system capacity has been provided for the new load.

Load and Tenant Balancing: They must be able to dynamically distribute load between servers so that most of the hardware resources are effectively utilized and to keep away from any resource overloading situations.

Fault Tolerance: For transactional workloads, a fault tolerant cloud data management system needs to be able to recover from a failure without removing any data or updates from lately committed transactions. For analytical workloads, a fault tolerant cloud data management system should not need to reset a query if one of the sites involved in query processing fails.

Ability to run in a heterogeneous environment: On cloud computing environment, there is a strong trend towards increasing the number of sites that operate in query execution. It is nearly unfeasible to get homogeneous performance across hundreds or thousands of compute sites. Part failures that do not cause complete node failure, but result decreased hardware performance become more usual at scale. A cloud data management system should be presented to use in a heterogeneous environment and must take suitable measures to avoid performance degrading due to parallel processing on distributed sites.

Flexible query interface: They should use both SQL and non-SQL interface languages (e.g Map Reduce). Moreover, they should present process for allowing the user to write user defined functions (UDFs) an queries that apply these UDFs should be automatically parallelized during their processing. It is possible to see that even some more application specific goals can influence the networking processes of the cloud. For instance, to make possible the load and tenant balancing, it is essential to guarantee that the migration of the applications will take a time which will not decrease the quality of the service presented to the user of these migrated applications. This time is directly related to network parameters such as available bandwidth and delay. A flexible query interface can also influence the network configuration of the cloud because it can require the parallelism of some operations, which needs the efficient transfer of data from one location to the various sites.

5.2. Challenges

Deploying data-intensive applications on cloud environment is not a simple process. We presented a list of problems to the growth of cloud computing applications as follows.

Availability of a Service: In general, a distributed system is a system that works robustly over a wide network. A specific characteristic of network computing is that the network connections can potentially disappear. Organizations concern about whether cloud computing services will have sufficient availability. High availability is one of the most important objectives because even the slightest outage can have notable financial consequences and impacts customer trust.

Data Confidentiality: In practice, moving data off premises increases the number of probable security risks and suitable precautions must be made. Transactional databases usually involve the complete set of operational data needed to power mission-critical business procedures. This data contains detail at the lowest granularity, and often has critical information like customer data or credit card

numbers. Therefore, unless such critical data is encrypted using a key that is not placed at the host, the data may be used by a third party without the customer's knowledge.

Data Lock-In: APIs for cloud computing has not been topic of active standardization. Thus, customers cannot easily obtain their data and programs from one site to run on another. The worry about the difficulties of getting data from the cloud is preventing some organizations from using cloud computing. Customer lock in may be interesting to cloud computing developer but cloud computing users are vulnerable to price increases, to reliability issues, or even to developers going out of business.

Data Transfer Bottlenecks: Cloud users and cloud developers have to consider the implications of placement and traffic at every level of the system if they want to reduce costs.

Application Parallelization: Computing power is flexible but only if workload is parallelizable. Obtaining additional computational resources is not as simple as just upgrading to a bigger and more powerful site on the fly. However, the additional resources are generally got by allocating additional server instances to a task.

Shared-Nothing Architecture: Data management applications proposed to use on top of cloud system should follow a shared-nothing architecture where each site is self-supporting and there is no single point of contention across the system. Most of transactional data management systems do not generally apply a shared-nothing architecture.

Performance Unpredictability: Many HPC applications need to guarantee that all the threads of a program are executing simultaneously.

Application Debugging in Large-Scale Distributed Systems: A challenging issue in cloud computing programming is the deletion of errors in these very large scale distributed systems. A usual event is that these bugs cannot be regenerated in smaller configurations, so the debugging must take place at the same scale as that in the production datacenters.

6. CONCLUSIONS

The taxonomy not only helps to map a cloud computing service, but it also helps potential customers and developers to point out what features the service they seek or wish to develop should have.

Cloud computing has presented a number of benefits for its hosting to the deployments of data-intensive applications like:

- Minimized time-to-market by removing or simplifying the time-consuming hardware providing, buying, and deployment procedures.
- Minimized cost by using a pay-as-you-go business model.
- Minimized operational cost and pain by automating IT operations like security patches and fail-over.
- Unlimited throughput by using several servers if the workload increases.

Although having started as key ways for different communities and with different sets of requirements, Cloud Data Management and Data Grids are more and more converging. In this work, we have presented the commonalities between both areas and the differences that still exist. We focused on the important goals and main challenges of deploying data intensive applications in cloud environments. In practice, there are several main classes of existing applications that seems to be

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

more compelling with cloud environments and contribute further to its momentum in the near future like: Mobile interactive applications and Parallel batch processing.

The comprehensive list of features makes it possible to find a great variety of cloud computing services. To allow more accurate comparisons, the taxonomy could be expanded to incorporate more details for some of the features. As mentioned above, the security presented by the taxonomy only considers security measures between the client and the cloud. An important addition to the taxonomy will be to also provide the security processes used within the cloud.

There is a need for further progress on different issues such as standards, portability, mappings to business architecture, security and privacy, multi-supplier and hybrid sourcing, management and governance with business analytics for cloud etc. If the service goes down for hour, customers are hit with the crises and customers are handicapped. In future work we plan to study the online data migration mechanisms to deal with the dynamic changes in file access characteristics.

REFERENCES

[1]. N. Mansouri, "Improve the performance of data grids by cost-based job scheduling strategy," *Computer Engineering and Applications Journal*, vol. 3 (2), pp. 101- 111, 2014

[2] N. Mansouri, "A hybrid approach for scheduling based on multi-criteria decision method in data grid," *Computer Engineering and Applications Journal*, vol. 3 (1), pp. 1-11, 2014.

[3] I.A. Targio Hashem, I. Yaqoob, N. Badrul Anuar, S. Mokhtar, A. Gani, S.U. Khan, " The rise of "big data" on cloud computing: Review and open research issues," *Information Systems*, vol. 47, pp. 98-115, 2015.

[4] S. Long, Y. Zhao, W. Chen," MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster," *Journal of Systems Architecture*, vol. 60, pp. 234–244, 2014.

[5]. A. Abdelmaboud, D. N.A. Jawawi, I. Ghani, A. Elsafi, B. Kitchenham, " Quality of service approaches in cloud computing: A systematic mapping study," *Journal of Systems and Software*, vol. 101, pp. 159-179, 2015

[6] J. M. Alcaraz Calero, J. Gutiérrez Aguado, " Comparative analysis of architectures for monitoring cloud computing infrastructures," *Future Generation Computer Systems*, vol. 47, pp. 16-30, 2015.

[7] Tippit Inc. (2008) WebHostingUnleashed: Cloud-computing services comparison guide, 2008 .

[8] B.P. Rimal, E. Choi, I. Lumb, 'A taxonomy and survey of cloud computing systems," *In: NCM'09: proceedings of the 2009 fifth international joint conference on INC, IMS and IDC. IEEE Comp. Society*, pp 44–51, 2009.

[9] H. Li, C. Spence, R. Armstrong, R. Godfrey, R. Schneider, J. Smith, R.White "Intel cloud computing taxonomy and ecosystem analysis," IT-Intel Brief (Cloud Computing), 2010.

[10] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for

delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25(6), pp. 599–616, 2009.

[11] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, M. Zaharia, "Above the clouds: a Berkeley view of cloud computing," *Technical Report*, No. UCB/EECS-2009-28, Dept, Uni of California, Berkeley, 2009.

[12] H. Li, C. Spence, R. Armstrong, R. Godfrey, R. Schneider, J. Smith, R. White, "Intel cloud computing taxonomy and ecosystem analysis," *IT-Intel Brief* (Cloud Computing), 2010.

[13] Y. Demchenko, D. Bernstein, A. Belloum, A. Oprescu, T.W. Wlodarczyk, C. de Laat, "New instructional models for building effective curricula on cloud computing technologies and engineering,", *5th International Conference on Cloud Computing Technology and Science*, vol.2, pp. 112 – 119, 2013.

[14] S.M.parikh,"Asurveyon cloud computing resourceallocation techniques, " *Nirma University International Conference on Engineering*, pp. 1 – 5, 2013.

[15] Y.Amanatullah, C.Lim,H.P.Ipung,A.Juliandri, "Toward cloud computi ng reference architecture: Cloud service management perspective," *International Conference on ICT for Smart Society*, pp. 1 – 4, 2013.

[16] Q. Zhang, L. Cheng, R. Boutaba, "Cloud computing: state-of the- art and research challenges," *Journal of Internet Services and Applications*, vol. 1(1), pp. 7–18, 2010.

[17] D. Parkhill, "The challenge of the computer utility," Addison-Wesley, 1966.

[18] P. Mell, T. Grance, "The NIST definition of cloud computing (v15), *Technical Report*, National Institute of Standards and Technology, 2009.

[19] L. Wang, G. Laszewski, A. Younge, X. He, M. Kunze, J. Tao, C. Fu, "Cloud computing: a perspective study," *New Generation Computing*, vol. 28(2), 2010.

[20] D. Hilley, "Cloud computing: a taxonomy of platform and infrastructure-level offerings," *Technical Report*, GIT-CERCS-09-13, CERCS, Georgia Institute of Technology, 2009.

[21] M.G. Avram, "Advantages and challenges of adopting cloud computing from an enterprise perspective," *Procedia Technology*, vol. 12, pp. 529-534, 2014.

[22] R. Harris,N.Olby, "Archives for Earth observation data," *Space Policy*, vol. 16(13), pp. 223–227, 2007.

[23] L. C. Voicu, H. Schuldt, Y. Breitbart, H. -J. Schek, "Replicated data management in the grid: The Re:GRIDiT approach," *In Proc. ACM DaGreS'09*, 2009.

[24] L.C.Voicu,H.Schuldt, F. Akal, Y. Breitbart, H. -J. Schek, "Re:GRIDiT – coordinating distributed update transactions on replicated data in the grid," *In Proc. Grid'09*, 2009.

[25] C. Olofson, "Worldwide RDBMS 2005 vendor shares," *Technical Report*, 201692, IDC, May 2006.

[26]. http://en.wikipedia.org/wiki/IBM_DB2

[27] http://www.ibm.com/developerworks/db2/library/techarticle/dm-0608

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

mcinerney/ index.html.
[28] http://www.oracle.com/solutions/business_intelligence/exadata.html
[29] C.Zou, H.Deng, Q.Qiu"designandimplementationofhybrid cloud comp uting architecture based on cloud bus," *Ninth International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 289 – 293, 2013.
[30] M. Stonebraker, S. R. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, P. Helland, The end of an architectural era (it's time for a complete rewrite), In *33rd international conference on Very large data bases*, pp. 1150-1160, 2007.
[31] S. Gilbert, N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *SIGACT News*, vol. 33(2), pp. 51–59, 2002.
[32] B. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H. Jacobsen, N. Puz, D. Weaver, R. Yerneni, "PNUTS: Yahoos hosted data serving platform," *In Proceedings of 34th VLDB*, 2008.
[33] V.H. Pardeshi, "Cloud computing for higher education institutes: architecture, strategy and recommendations for effective adaptation," *Procedia Economics and Finance*, vol. 11, pp. 589-599, 2014.
[34] A.Abouzeid,K.Bajda-Pawlikowski,D.Abadi, A. Rasin, A. Silberschatz, "Hadoopdb: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads," *Proceedings of the VLDB Endowment*, vol. 2(1), pp. 922–933, 2009.
[35] J.Hofstader.Communicationsas a service. http://msdn.microsoft.com/en us/library/bb896003.aspx.
[36] Connectedservicesframework.http://www.microsoft.com/serviceprovid ers/solutions/connectedservicesframework.mspx.