

# A Survey on Various Data Replication & Applications in Cloud

Supriya Kumari<sup>1</sup>, Prof. Deepti Dave<sup>2</sup>

**Abstract**— Cloud computing provides scalable computing and storage resources. More and more data intensive applications are developed in this computing environment. Different applications have different quality-of-service (QoS) requirements. To continuously support the QoS requirement of an application after data corruption in the existing technique two QoS aware based algorithm are implemented. The first algorithm adopts the intuitive idea of high-QoS first-replication (HQFR) to perform data replication. Here in this paper a complete survey of all the techniques are analyzed and discussed here.

## I. INTRODUCTION

The cloud computing paradigm [1, 2] has emerged as an efficient and cost effective way for managing and delivering services over the Internet. Cloud computing allows customers to acquire resources (i.e., computing power, storage, memory) in a very short time on a pay-per-use basis obtaining a high degree of elasticity. This allows minimizing startup costs and to rapidly scale up or down resources avoiding performance degradation in case of peak load and over-provisioning in case of scarce demanding.

Among all the challenges that cloud computing poses, we focus our attention on the dynamic QoS provisioning problem. QoS delivering is not a different matter in networked and distributed schemes; but the cloud and service computing standards proliferation the structure complication and measure, consequently self-importance innovative tasks. Cloud computing is architected in a stack composed by three main service models or abstract layers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Each layer offers to the upper layer its resources as a set of services, in the spirit of earlier distributed computing and network architectures.

According to the layers of the cloud service, the dynamic QoS provisioning problem can be managed at infrastructure and platform levels (e.g., [3]–[4]). Despite of the success of cloud computing and of the level and development of cloud-based descriptions, the problem of service level delivering in cloud systems is still an open issue. Some IaaS providers (e.g., Amazon Web Services [5], Rackspace [6]) offer simple autoscaling services that are still far away from allow to an application service provider or Software as a Service provider to efficiently allocate resources minimizing costs and promising the aspiration level of presentation in circumstances of changeable and bursty traffic. Additionally, now existing deal level agreements frequently deliver guarantees only on arrangement obtainability entirely overlooking other high level presentation metrics such as regular response time or throughput [7]

The field of distributed computing has understood technologies quickly develop from desktop computing, completed Grid computing, and currently to Cloud computing.

All these technologies focus on delivering computing power to a large number of end-users in a reliable, efficient and scalable manner. Cloud computing has raised the delivery of IT services to a new level that brings the comfort of traditional utilities such as water and electricity to its users. The advantages of Cloud computing, such as cost effectiveness, scalability, and ease of management, encourage more and more companies and service providers to adapt it and offer their solutions via Cloud computing models. According to a recent survey of IT decision makers of large companies, 68% of the respondents expect that by 2014, more than 50% of their company's IT services will be migrated to Cloud platforms [8]. Cloud computing has become a scalable service consumption and delivery platform.

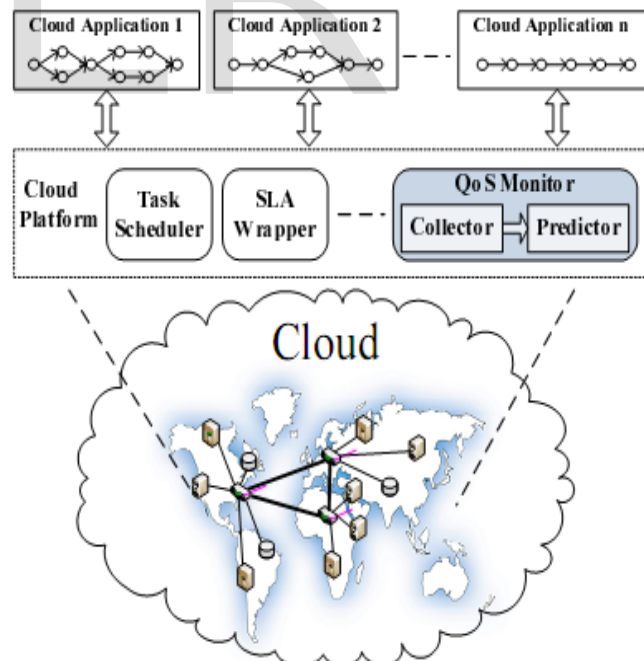


Figure 1.1: System Architecture of cloud computing.

Figure 1.1 shows the system architecture in cloud computing. In a cloud environment, the cloud provider holds a large number of distributed services (e.g. databases, servers, Web services, etc.), which can be provided to designers for developing various cloud applications. Designers of cloud applications can choose from a broad pool of distributed services when composing cloud

applications. These services are usually invoked remotely through communication links and are dynamically integrated into the applications. The cloud application designers are located in different geographic and network environments. Since the users invoke services via different communication links, the quality of services they observed are diverse.

The idea of our approach is to share local cloud component usage experience from different component users, to combine this local information to get global QoS information of all components, and to make personalized QoS value prediction based on both global and local information. As shown in Figure 1.1, each component user keeps local records of QoS usage experiences on cloud components. Since cloud applications are running on an identical cloud platform, QoS information can be collected by an identical interface on the platform side. If a component user would like to get personalized QoS information service from the cloud provider, authorization should be given to Collector for accessing its local QoS records. Collector then collects those local QoS records from different component users. Based on the collected QoS information, Predictor can perform personalized QoS value prediction and forward the prediction results to component users for optimizing the design of cloud applications.

Quality-of-Service (QoS) is usually employed to describe the non-functional characteristics of services. It becomes a major concern for application designers when making service selection. Moreover, for the existing cloud applications, by replacing low quality services with better ones, the overall quality of cloud application can be improved.

## II. QUALITY-OF-SERVICE AWARENESS IN CLOUD COMPUTING

Quality-of-Service (QoS) is usually employed to describe the non-functional characteristics of services. It becomes a major concern for application designers when making service selection [9]. Moreover, for the existing cloud applications, by replacing low quality services with better ones, the overall quality of cloud applications can be improved.

In recent year, a number of research tasks have been focused on optimal service selection [10] in distributed systems or service computing. Typically, evaluations on the service candidates are required to obtain their QoS values. In cloud environment, due to their various locations and communication links, different users will have different QoS experiences when invoking even the same service. Personalized QoS evaluation is required for each user at the user-side. However, a service user in general only invoked a limited number of services in the past and only received QoS performance information of these invoked services. In practice, therefore, conducting real-world evaluations on services to obtain their QoS information from the users' perspective is quite difficult, because: (1) executing invocations

for evaluation purposes becomes too expensive, since cloud providers who maintain and host services (e.g., Amazon EC2, Amazon S3, etc.) may charge for invocations; (2) with the growing number of available services over the Internet, it is time-consuming and impractical to conduct QoS evaluations on all accessible services; (3) component users need to focus on building cloud applications on top of various services. While conducting evaluation on a large number of service candidates would introduce extra cost and effort, and sharply slow down the application development progresses. Therefore, collecting historical usage records and conducting QoS prediction, which requires no additional invocation, is becoming an attractive approach. Based on the above analysis, in order to provide QoS information to application designers, we need to provide comprehensive investigation on QoS prediction approaches. In cloud computing, however, users can access multiple functional equivalent services via Internet at a very low cost. These services are usually developed and provided by different organizations, and can be dynamically composed to build fault tolerance systems.

## III. THEORETICAL BACKGROUND

Cloud computing [11] is Internet-based computing, whereby shared resources, software, and information are provided to computers and other devices on demand. With the exponential growth of cloud computing as a solution for providing flexible computing resources, more and more cloud applications emerge in recent years. The systems design of the software arrangements comprised in the distribution of cloud computing (named as cloud applications in this chapter), usually includes multiple cloud elements cooperating with each other finished application programming interfaces, usually Web services [12]. How to build high-quality cloud applications becomes an urgent and crucial research problem.

In the cloud environment, designers of cloud applications, denoted as component users, can choose from a broad pool of cloud components when creating cloud applications. These cloud components are usually invoked remotely through communication links.

Quality of the cloud applications is greatly influenced by the quality of communication links and the distributed cloud components. To build a high-quality cloud application, non-functional Quality-of-Service (QoS) performance of cloud components becomes an important factor for application designers when making component selection [9]. Moreover, for the existing cloud applications, by replacing low quality components with better ones, the complete excellence of cloud applications can be increased.

In recent year, a number of research tasks have been focused on optimal component selection [10] and recommendation [8] in distributed systems or service computing. Typically, evaluations

on the component candidates are required to obtain their QoS values.

In cloud environment, due to their various locations and communication links, different users will have different QoS experiences when invoking even the same cloud component. Personalized QoS evaluation is required for each user at the user-side. However, a cloud component user in general only invoked a limited number of cloud components in the past and only received QoS performance information of these invoked cloud components.

#### IV. DATA REPLICATION AND STORAGE ON CLOUD COMPUTING

A Data Grid is a geographically-distributed teamwork in which all participants necessitate admission to the datasets yield within the relationship. Replication of the datasets is therefore a key requirement to ensure scalability of the cooperation, dependability of data access and to reservation bandwidth. Replication is constrained by the size of storage existing at altered positions inside the Data Grid and the bandwidth amongst these sites. A replica management system therefore confirms admittance to the necessitated data while management the essential storage.

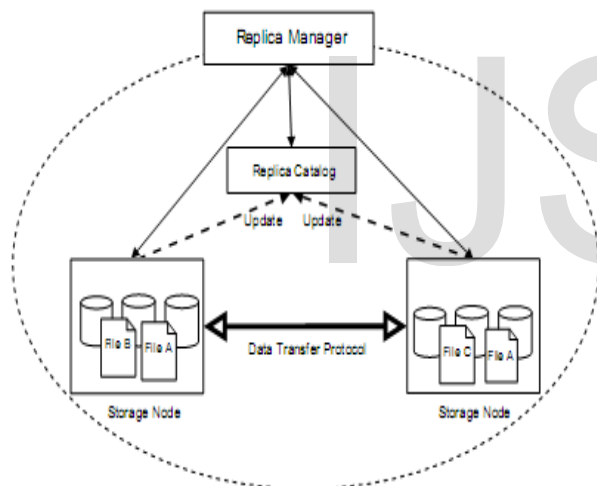


Figure 1.2: Replica Management Architecture.

A replica management system, shown in Figure 1.2, consists of storage nodes which are linked to each other via high-performance data transport protocols. The replica manager uninterrupted the formation and supervision of reproductions allowing to the requests of the customers and the accessibility of storage, and a collection or a directory keeps pathway of the duplications and their sites. The collection can be demanded by presentations to determine the number and the positions of existing duplications of a specific dataset. In some systems, the manager and the catalog are merged into one entity. Client-side

software generally consists of a library that can be integrated into applications and a set of commands or GUI utilities that are built on top of the libraries. The client libraries allow querying of the catalog to discover datasets and to request replication of a particular dataset.

#### V. LITERATURE SURVEY

In this paper [13] here author has been build upon the development of distributed computing, grid computing and virtualization on using Cloud computing. Different cloud resources has different scheduling cost of each task in cloud resources is different with one another, development of user assignments in cloud is not the equivalent as in conventional scheduling techniques. The intention of this paper design is to schedule task collections in cloud computing proposal, where resources have different resource costs and calculation performance. Due to job combination, communication of rude-small pieced jobs and resources optimizes computation/announcement relative amount. For this reason, an algorithm foundation on both costs with user task grouping is recommended. The proposed [13] scheduling come within reach of in cloud utilizes a get bettered cost-based scheduling algorithm for making well-organized drawing of tasks to available resources in cloud. This scheduling algorithm determines both resource cost and computation presentation, it also get better the working out and contact proportion by grouping the user tasks according to a scrupulous cloud resource's processing competence and launches the collection jobs to the resource.

In this paper [14] author has facing a task scheduling difficulties are of principal consequence which communicate to the effectiveness of the whole cloud computing services. In Ha doop, the open-source accomplishment of Map Reduce, scheduling policies, for instance FIFO or delay scheduling in FAIR scheduler is utilized by the master node to allocate waiting tasks to computing nodes i.e. slaves in answer to the class messages of these nodes it accepts. Even though delay scheduling guiding principle has declared to get better the throughput and response times by a factor of 2 evaluated to FIFO policy, it can unmoving accomplish more enhancement by taking into consideration a holistic observation of all the tasks in the making to be developmental. Consequently, this paper author tries to suggest a new scheduler which makes a scheduling pronouncement by estimating the complete group of tasks in the job queue. A genetic algorithm is planed as the optimization technique for the new scheduler. The preface replication consequences show that our scheduler can get a shorter make extent for jobs than FIFO and delay scheduling guidelines and accomplish an enhanced reasonable load transversely all the nodes in the cloud.

In this paper [15] author suggests a market-oriented hierarchical scheduling approach in cloud workflow methods. In particular, the service-level scheduling arrangements with the Task-to-Service job where tasks of entity workflow illustrations are planned to cloud services in the global cloud markets supported on their functional and non-functional QoS constraints; the task-level scheduling contracts with the optimization of the Task-to-VM (virtual machine) task in local cloud data centers where on the whole running cost of cloud workflow schemes will be reduced given the approval of QoS constraints for entity tasks. Supported on our hierarchical scheduling approach, a package based random scheduling algorithm is offered as the candidate service-level scheduling algorithm and three agent meta-heuristic based scheduling algorithms including genetic algorithm (GA), ant colony optimization (ACO), and particle swarm optimization (PSO) are become accustomed, put into practiced and examined as the aspirant task-level scheduling algorithms. The hierarchical scheduling strategy is individual put into practiced in our SwinDeW-C cloud workflow organization and representing acceptable show. For now, the investigational consequences show that the taken as a whole presentation of ACO based scheduling algorithm is enhanced than additional on three basic amounts: the optimization rate on make span, the optimization speed on expenditure and the CPU time.

Day-by Day the Cloud computing popularity has increased in modern times [16]. As a cloud must make available services to many customers at the same time and different customers have different QoS prerequisites, the scheduling approach should be expanded for various workflows with different QoS conditions. In this paper author has suggested a new method multiple QoS constrained scheduling approach of multi-workflows (MQMW) to concentrate on this difficulty. The approach can program multiple workflows which are established at any time and the QoS conditions are full into account. On applying this approach experimentation give you an idea about that our plan is proficient to enhance the scheduling accomplishment rate extensively.

In this paper [17], author has to present two work of fiction scheduling algorithms for a restricted number of varied processors with an intention to concurrently get together high performance and fast scheduling time, which are called the Heterogeneous Earliest-Finish-Time (HEFT) algorithm and the Critical-Path-on-a-Processor (CPOP) algorithm. The HEFT algorithm picks the task with the maximum increasing rank value at each step and allocates the chosen task to the processor, which reduces its most primitive end time with an insertion-based approach. Alternatively, the CPOP algorithm uses the outline of upward and downward rank importances for prioritizing assignments. Another difference is in the processor selection stage, which plans the critical tasks onto the processor

that decreases the total execution time of the critical assignments. With the intention of provide a forceful and impartial evaluation with the interrelated effort; a parametric graph initiator was proposed [17] to produce weighted directed acyclic graphs with a variety of distinctiveness. The evaluation learning, based on both indiscriminately generated graphs and the graphs of some authentic purposes, demonstrates that our scheduling algorithms extensively exceed preceding move towards in terms of both quality and price of schedules, which are generally accessible with schedule length ratio, speedup, frequency of most excellent consequences, and average scheduling time metrics.

## VI. CONCLUSION

Cloud computing has revolutionized the Information and Communication Technology (ICT) industry by enabling on-demand provisioning of elastic computing resources on a pay-as-you-go basis. An organization can either outsource its computational needs to the Cloud avoiding high up-front investments in a private computing infrastructure and consequent costs of maintenance and upgrades, or build a private Cloud data center to improve the resource management and provisioning processes.

## ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression, “One of us (R. B. G.) thanks . . .” Instead, try “R. B. G. thanks”. Put applicable sponsor acknowledgments here; DO NOT place them on the first page of your paper or as a footnote.

## REFERENCES

- [1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5<sup>th</sup> utility,” *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
- [2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the Clouds: A Berkeley view of Cloud computing,” EECS Dep., Univ. of Berkeley, Tech. Rep. UCB/EECS-2009-28, Feb 2009.
- [3] M. Andreolini, S. Casolari, M. Colajanni, and M. Messori, “Dynamic load management of virtual machines in cloud architectures,” in *Proc of 1st Int’l Conf. on Cloud Computing (ICST CLOUDCOMP ’09)*, ser. LNICST, vol. 34. Springer, 2010, pp. 201–214.
- [4] Y. Chen, T. Wo, and J. Li, “An efficient resource management system for online virtual cluster provision,” in *Proc of IEEE Int’l Conf. on Cloud Computing*, 2009, pp. 72–79.
- [5] “Amazon web services,” [aws.amazon.com](http://aws.amazon.com).
- [6] “Rackspace,” [www.rackspace.com](http://www.rackspace.com).

- [7] S. A. Baset, "Cloud slas: Present and future," SIGOPS Oper. Syst. Rev., vol. 46, no. 2, pp. 57–66, Jul. 2012. [Online]. Available: <http://doi.acm.org/10.1145/2331576.2331586>.
- [8] B. Narasimhan and R. Nichols, "State of cloud applications and platforms: The cloud adopters' view," Computer, vol. 44, no. 3, pp. 24–28, 2011.
- [9] B. Hayes. Cloud computing. Communications of the ACM, 51(7):9–11, 2008.
- [10] T. Yu, Y. Zhang, and K. Lin. Efficient algorithms for Web services selection with end-to-end QoS constraints. ACM Transactions on the Web (TWEB), 1(1):6, 2007.
- [11] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. Communications of the ACM, 53(4):50–58, 2010.
- [12] Wikipedia. <http://en.wikipedia.org/wiki/cloudcomputing>.
- [13] Improved cost-based algorithm for task scheduling in cloud computing - ResearchGate. Available from: [http://www.researchgate.net/publication/241180034\\_Improved\\_costbased\\_algorithm\\_for\\_task\\_scheduling\\_in\\_cloud\\_computing](http://www.researchgate.net/publication/241180034_Improved_costbased_algorithm_for_task_scheduling_in_cloud_computing) Jul 7, 2015.
- [14] GA-Based Task Scheduler for the Cloud Computing Systems. Available from: [http://www.researchgate.net/publication/251979305\\_GA-Based\\_Task\\_Scheduler\\_for\\_the\\_Cloud\\_Computing\\_Systems](http://www.researchgate.net/publication/251979305_GA-Based_Task_Scheduler_for_the_Cloud_Computing_Systems) Jul 7, 2015.
- [15] Zhangjun Wu, Xiao Liu, Zhiwei Ni, Dong Yuan, Yun Yang, "A market-oriented hierarchical scheduling strategy in cloud workflow systems" January 2013, Volume 63, Issue 1, pp 256-293.
- [16] Lizhen Cui ; Haiyang Wang ; Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing" Meng Xu Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on 10-12 Aug. 2009.
- [17] Topcuoglu, H. Hariri, S. ; Min-You Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing Parallel and Distributed Systems, IEEE Transactions on (Volume:13 , Issue: 3 ) 260 – 274 Mar 2002.

IJSER