# A toolkit for modeling and simulating cloud data storage: an extension to CloudSim

Saiqin Long, Yuelong Zhao
School of Computer Science & Engineering
South China University of Technology
Guangzhou, China
longsaiqin@163.com, zngdhyn@sina.com

***Abstract*—With the popularity of cloud computing technology, cloud storage technology as a new kind of distributed storage technology is widely used to store and manage massive data. Large-scale data management and analysis is difficult to achieve in complex scenarios. In order to overcome this challenge, we extend the cloud simulation platform CloudSim by adding the file striping and data replica management function, making it into a simulation platform for computing and storage. The expanded CloudSim can be used to implement different data layout and replica management strategy. And it's easily to add functionality to meet other needs. This paper gives a detailed description of the design and proves the significance of it through examples.**

***Keywords- cloud simulation; data clouds;data striping; replica***

## I. INTRODUCTION

Cloud computing as a new type of distributed computing technology was first proposed in 2006 after and was widely used in astronomy[1], high energy physics[2], bioinformatics [3] and other areas of various data-intensive applications which need to deal with data with size in TB or even PB level[4], More and more enterprises prefer to choose the professional cloud storage service providers such as Amazon's S3[5], Google's GFS[6] and Microsoft's Live Services to store and manage their data. But as the development of web2.0 technology, application of cloud service users become more flexible, it is difficult to quantify the performance of the system by building such a complex, dynamic and meeting users' QoS requirements cloud environment under current conditions. Testing the performance of the system with the real infrastructure is usually limited by specific infrastructure. For example, Yahoo and HP established a cloud computing test platform OpenCirrus to support the integration of data centers from ten organizations. But it takes a lot of money to set up the experiment environment and is difficult to repeat the experiment with the change of resource conditions. Only the members can access it. Therefore, a feasible way for developers to evaluate the performance of the system is simulation tools. Simulation tools enable developers to repeat in a controllable and heterogeneous environment and predict the future bottlenecks of the system.

Considering there's no simulation platform on cloud storage, we extended the well-known cloud computing simulation platform CloudSim [7]. The extended CloudSim has the following new features: (1) it strips a file before storing it on the storage so that different data blocks of the file can be distributed on different data nodes which improves parallel access ability of the system; (2) the file is stored in multiple data centers in the form of replica which is benefit for resource sharing; (3) it supports querying the position of the replica; (4) it improves data availability through replica management; (5) it supports complex queries on the data attributes. These new features make it convenient for CloudSim users to integrate their replica management strategy and file striping policy into the cloud task scheduling in data center available resources. Therefore, the main contribution of this paper is adding data cloud storage function into CloudSim. Many cloud developers will be benefit form it as they needn't to implement the data cloud entities to test specific data layout and replica management strategy.

## II. ARCHITECTURE OF THE EXPANDED CLOUDSIM

The CloudSim architecture with a new Data Cloud layer is shown in Figure 1. Underlying of CloudSim are some core functions such as message or event queue, creation of cloud system entities and management of simulation clock. CloudSim consists of six layers. Network layer defines the network topology and Euclidean distance, link delay, link bandwidth between entities of cloud computing have been modeled. As CloudSim is a time-based emulator, different entities deliver messages between them by sending events. This layer simulates the delay generated by the interaction between entities. Cloud resources layer defines the various resource entities in the cloud storage system and processes events from entities. Cloud services layer allocates virtual machine, processor, memory capacity, bandwidth to cloud task. Data cloud layer is the expansion layer of CloudSim and it mainly used to manage data. Researchers can implement their own file layout strategy, file striping strategy, as well as replica management strategy based on this layer. Virtual service layer is mainly used to manage cloud task and virtual machine. Cloud providers can use their own host or virtual machine allocation strategy by extending the core functions of this layer. User interface structure layer is mainly responsible for the definition of the cloud tasks and virtual machine. The top-level structure in CloudSim is user code layer [7]. This layer

This research is financially supported by the National Natural Science Foundation of China (Grant No. 60573145), the Ph. D. Programs Foundation of Ministry of Education of China (Grant No. 200805610019) and the Guangzhou City Science and Technologies Program (No. 2010Y1-C681)

978-0-7695-4881-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICCECT.2012.160

provides the basic host entity classes, application, number of virtual machine, number of users, application type and agent scheduling policy. Cloud application developers can generate a mixed workload configuration of application through expanding the entities of the layer.

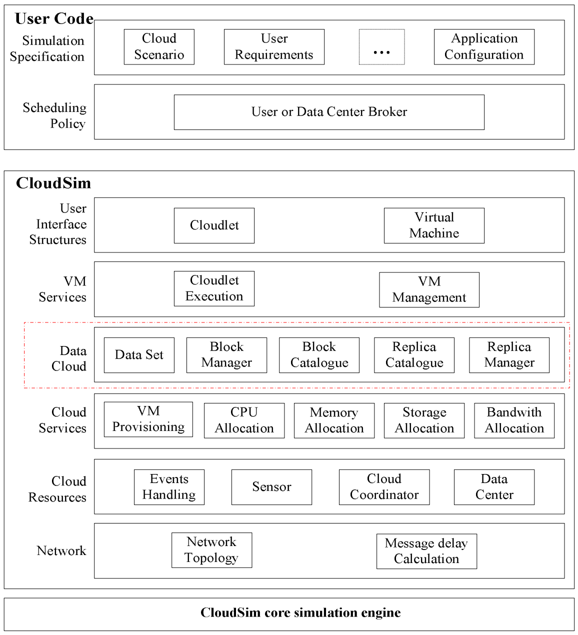


Figure 1. CloudSim architecture with the new data Clouds layer.

### *A. Data sets*

Data sets in the cloud environment can be used as a unit to access the collection of share data. It can be a series of files on the physical disk or only a single large file with specific format. We use files object to represent the data sets in CloudSim. Each file has attributes, such as file owner, file size, file checksum, last updated time, number of replicas, number of data blocks. Similarly, we use blocks object to represent data blocks which also have their attributes. Data Cloud supports two types of file striping operation: fixed length and non-fixed-length. Fixed-length means that it strips all the files with a fixed stripe parameter into sub-blocks, so all data blocks have the same length. Non-fixed-length means that it stripes files with a changeable stripe parameter which is based on the size and access frequency of the file. In addition, data cloud separates file object by setting two types named master or replica to management data. First created files are defined master, replica is a copy from master, and the difference between them is that master can not be automatically deleted. It is created by the user or user's program. Only the user has permissions to delete. On the contrary, replica can be dynamically created or deleted by data center according its storage constraints and access requirements.

### *B. Name Node*

Data cloud provides Replica or Block Catalogue (named RC or BC) entity, which is responsible for providing the location information of data sets in CloudSim. It supports two types of RC model: centralized RC and hierarchical RC. Only one RC component is used to process the registration and queries from all users or resources in centralized RC model. The RC maps file name to a list of resources which store this file. The hierarchical model is built into a directory tree. The root node in the model is TopRegionalRC used to save the route location, while the actual storage location of the file is stored in the leaf node named RegionalRC or LocalRC. Thus RegionalRC and LocalRC can share the queries from users or resources with the root node which overcomes bottleneck of the centralized model. BC Model is similar with RC model so we don't repeat. This idea originates from GridSim [8].

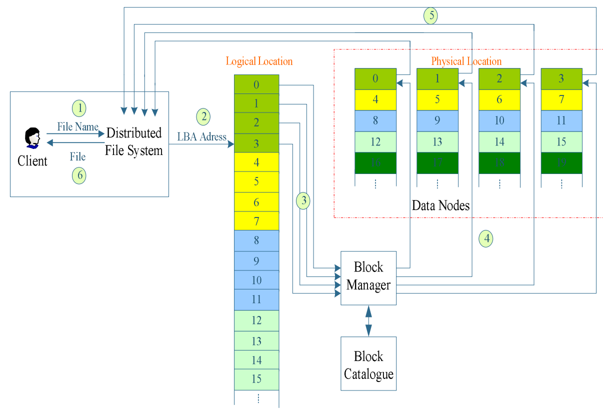


Figure 2. An example of file block.

We introduce Name Node entity in CloudSim, which is mainly used for replica and block management. Replica management is a component which is used to retrieve or query the available replica in Data Cloud. For example, when the replica manager receives a query or read request from a user, it first query RC to obtain the location information, and then return the result message to the user. Block management is mainly used to stripe large files and manage data block so as to improve the parallel access ability of the system. Figure 2 shows an example of storing data blocks in round-robin way in different data nodes.

### *C. Data Center*

Data center is the core component of the cloud system to provide users with hardware and software resources. It provides users with computing and storage service. Computing service allows users to perform the required application on it, while storage service allows users to access the data sets and store the result of computing application in data center. In this paper, we extend the data center in CloudSim, joined the large-scale storage system. The expansion can not only be used for high-performance computing, but also for efficient storage and access of large data sets.

### *D. Implementing data cloud operations in CloudSim*

Figure 3 describes the flow of communication between the core entities in the expanded CloudSim during the file writing operation. At the beginning of the simulation, each data center is first registered with the cloud information service center and then cloud information service center will provide query function for available resources to data center proxy. Agent query for the cloud service providers which meet its required hardware and software requirements, and then create the virtual machines used to perform the tasks submitted to the cloud from users on the data center. If the cloud task needs to store large files in data center, they are first submitted to block manager for striping into sub-blocks. The replica manager will store these blocks in accordance with a data layout strategy in the SAN storage devices of data center and register the specific storage locations of data blocks to the RC. So that when users need to read the files, they will know the specific locations of the files by querying the RC. In addition, the replica manager also implements one or more backup operations for files based on data blocks by a certain strategy. It is noteworthy that the traffic in the diagram only describes the simulation experiments related to the basic flow of an operation. During other operations, traffic may be different depends on the specific policy.

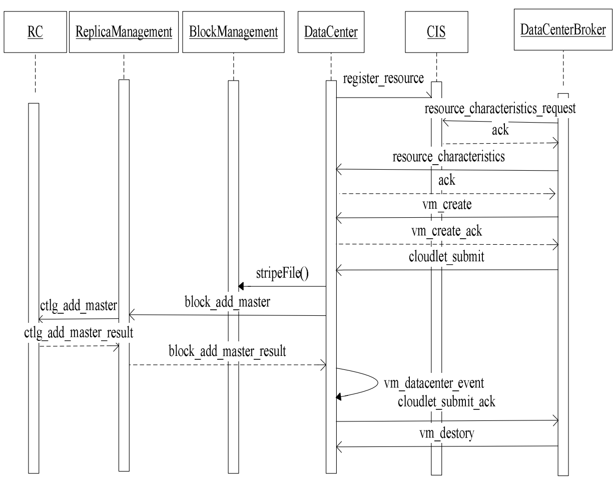


Figure 3. Simulation data flow.

## III. EXPERIMENTS AND EVALUATION

This section will focus on the construction of the simulation environment, doing data cloud storage with the simulation environment and evaluating the experimental result of the operation.

### *A. the Simulation set-up*

In this experiment, we created two data centers, each data center consisted of ten dual-core hosts and ten quad-core hosts with processing rate 1000MIPS and memory 16384MB. Each data center has SAN-based storage devices in the storage pool with capacity of 1PB. We random mapped the entities in CloudSim to the network topology shown in Figure 4 which made CloudSim had the ability to simulate the real network topology. Three parameters of the triple on the link in the graph represent the Euclidean distance between two nodes, link delay and link bandwidth. They are stored in a BRITE format file. We can easily simulate different network topology by changing that configuration file. In addition, we assume that the file size of the cloud task required obeying the Pareto distribution. Figure 5 shows the required file size of the experiment 70 cloud tasks which obey Pareto distribution. We can see from the dotted line in the graph that 80% of the cloud tasks require files to less than 3500MB, only a very small to 10,000 MB or above.

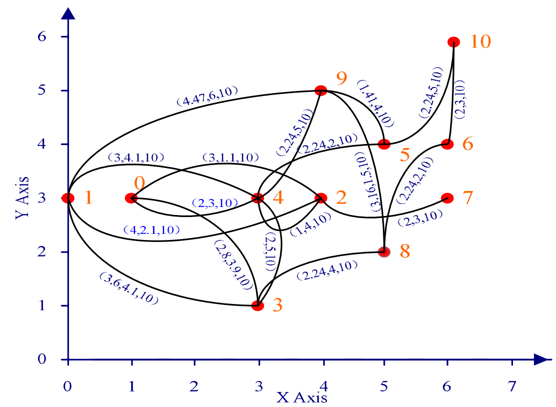


Figure 4. The simulated network topology.

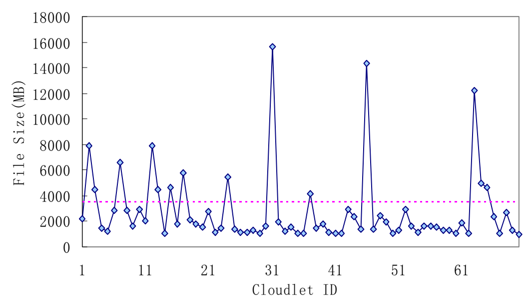


Figure 5. Size of cloud tasks required files.

### *B. Simulation results*

This section is mainly test the execution time of cloud tasks. We have adopted scheduling policy based on time-sharing when cloud task scheduled in virtual machine and virtual machine on the host. Figure 6 describes the life cycle of cloud tasks when we strip the required files in 64MB size including their start time, run time and completion time. We adopted a batch submitted way on Cloud task submission, so there is some cloud task start at the same time. Cloud task completion time is defined as sum of start time and running time.

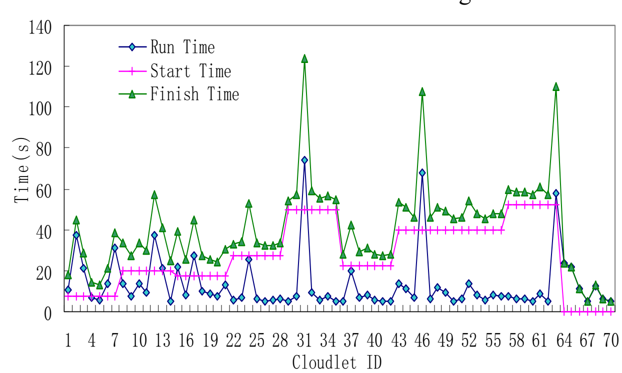


Figure 6. The life cycle of the cloud tasks

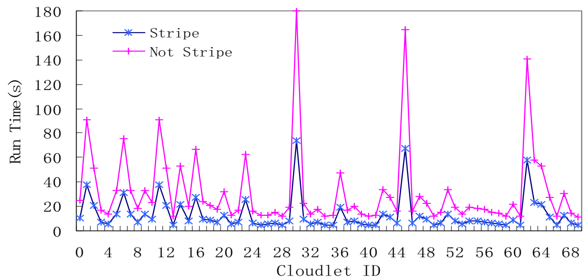


Figure 7. Impact of striping or not on run time of the cloud tasks.

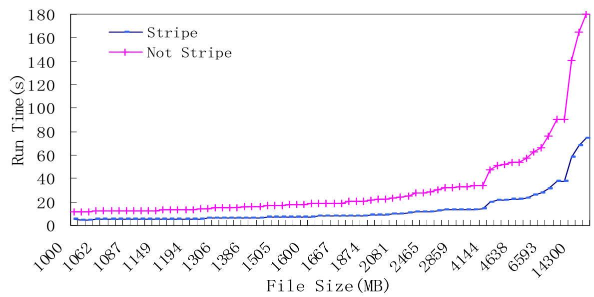


Figure 8. Impact of striping or not on cloud tasks sorted by file size.

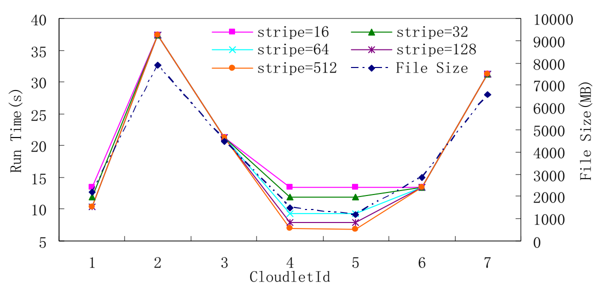


Figure 9. Impact of striping parameter on cloud tasks.

Figure 7 shows the impact of striping or not on run time of the cloud tasks. We can see that doing striping operation can reduce the run time of the cloud tasks more than half compared to not. Figure 8 shows when sorted by file size, contrast on run time of the cloud task in striping or non-striping case. It can be seen from the figure the larger the file, the more time can be saved for the execution of tasks joined the operation of the striping on the cloud. Next, we consider how to choose the striping parameters to optimize system performance as much as possible. We can see that the choice of the striping parameter makes greater impact on run time of cloud task when file size is less than 3000MB from Figure 9. Then the greater the striping parameters, the smaller the running time of the cloud task. When the file size is more than 5000MB, the choice of the striping parameter has little effect on run time of cloud task. This is because the larger the file size, the more the number of data blocks. But the number of storage devices in our layout strategy is fixed which leads large number of data blocks being assigned to a storage device, plus the delay of storage device, the performance of the system comes down.

# IV. CONCLUSION AND FUTURE WORK

In this paper, one can be used for the simulation of cloud data storage framework is proposed on the basis of well-known cloud computing simulation tools CloudSim. The extended CloudSim possesses the core functions of the cloud data storage, mainly in the following two aspects: First, the increased file striping function on CloudSim strips files into sub-blocks and stores data blocks to storage devices of the data center data in accordance with the data layout strategy which improves concurrent access ability of the system and saves the execution time of the cloud tasks. Second, the increased replica management function on CloudSim makes the data being stored in the form of replica in different storage devices which avoids the case that the entire file is not available due to a data block is not available and improves the availability of data. Replica management feature also supports queries for the replica location and access to a replica. Large number of experiments has proved that file striping operation can accelerate the implementation of the cloud task if all data blocks are available.

In addition, our work also facilitates the follow-up researchers to model and simulation cloud applications with CloudSim. They can easily achieve their own data layout strategy and replica management strategy in the cloud data storage framework. In future work, in order to better simulation of cloud data storage, we plan to propose a new replica management model and technology which can not only improve system performance, but also save energy and the cost of cloud service providers to satisfy customers' needs.

# REFERENCES


[1] E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.H. Su, K. Vahi and M. Livny, "Pegasus: Mapping scientific workflows onto the grid," Grid Computing Lecture Notes in Computer Science, Vol.3165, pp. 131-140, 2004.

[2] B. Ludascher, l. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A.Lee, J. Tao and Y. Zhao, "Scientific workflow management and the Kepler system," Concurrency and Computation: Practice and Experience, vol.18, pp.1039-1065, 2005.

[3] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat and P. Li, "Taverna: A tool for the composition and enactment of bioinformatics workflows," Bioinformatics, vol.20, pp.3045-3054, 2004.

[4] P. Zheng, L. Z. Cui, H.Y. Wang and M.Xu, "A data placement strategy for data-Intensive applications in cloud," Chinese Journal of Computers, vol.33, pp. 1472-1480, 2010.

[5] K.P.N. Puttaswamy, T. Nandagopal and M. Kodialam, "Frugal Storage for Cloud File Systems," Proceeding EuroSys '12 Proceedings of the 7th ACM european conference on Computer Systems," 2012, pp.71-84.

[6] S. Ghemawat, H. Gobioff and S.T. Leung, "The Google File System," ACM SIGOPS Operating Systems Review - SOSP '03, vol.37, pp. 29-43, 2003.

[7] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F. De Rose and R. Buyya, "CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services," Echnical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, March 13, 2009.

[8] A. Sulistio, U. Cibej and S. Venugopal, B. Robic and R. Buyya, "A toolkit for modeling and simulating data Grids:an extension to GridSim," Concurrency Computat.: Pract. Exper. Vol.20, pp. 1591-1609, 2008.