ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Practical Approach for Achieving Minimum Data Sets Storage Cost In Cloud

M.Sasikumar[1], R.Sindhuja[2], R.Santhosh[3]

ABSTRACT— Traditionally, computing has meant calculating results and then storing those results for later use. Unfortunately, committing large volume of rarely used data to storage wastes space and energy, making it a very expensive strategy. Cloud computing, with its readily available and flexibly allocatable computing resources, suggests an alternative: storing the provenance data, and means to recomputing results as needed. It is used to deploy computation and data intensive application without infrastructure investment. Large application datasets can be stored in the cloud. They are based on Pay as you go model. It is used for Cost Efficient Storage of large volume of generated datasets in the cloud. All these are done for achieving the minimum cost Benchmark in cloud. The main focus of this strategy is the local-optimization for the trade off between computation and storage, while secondarily also taking users' (optional) preferences on storage into consideration. Both theoretical analysis and simulations conducted on general (random) data sets as well as specific real world applications with Amazon's cost model show that the cost effectiveness of our strategy is close to or even the same as the minimum cost benchmark, and the efficiency is very high for practical runtime utilization in the cloud.

KEYWORDS— cloud computing, computation-storage, data intensive application, dataset storage, minimum cost benchmark, trade off, data set cost

I. INTRODUCTION

The latest emergence of Cloud computing is a significant step towards realizing this utility computing model since it is heavily driven by industry vendors. Cloud computing promises to deliver reliable services through next-generation data centers built on virtualized compute and storage technologies. Users will be able to access applications and data from a "Cloud" anywhere in the world on demand and pay based on what they use. Many high-performance computing (HPC) and scientific workloads (i.e., the set of computations to be completed) in cloud environment, such as those in bioinformatics, biomedical informatics, chem. informatics and geo informatics, are complex workflows of individual jobs. The workflow is usually organized as a directed acyclic graph (DAG), in which the constituent jobs (i.e., nodes) are either control or data dependent (i.e., edges).Control-flow dependency specifies that one job must be completed before other jobs can start. In contrast, dataflow dependency specifies that a job cannot start until all its input data (typically created by previously completed jobs) is available. Control-flow is the more commonly used abstraction to reason about the relationship between different jobs, but we show how dataflow information is more valuable to effectively utilize the storage.

II. BACKGROUND REVIEW

Data centres (DCs) are a crucial component of the cloud computing paradigm. They house the computational resources and associated equipment required to provide the services available on the cloud. Some clouds are shared environments where multiple cloud users utilize the same equipment. Hence, there is potential for both unintentional and malicious service interference between users. The effects of service interference can be seen on current clouds. The throughput of medium instances on Amazon's EC2 can vary by 66% and it has been conjectured, based on anecdotal evidence that the reason for this is a lack of algorithms which manage bandwidth allocation between users. There are numerous mechanisms which can be used for controlling and managing computational, memory and disk re- sources. We compare and contrast the performance and monetary cost-benefits of clouds for desktop grid applications, ranging in computational size and storage. We address the following questions: (i) What are the performance tradeoffs in using

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

one platform over the other? (ii) What are the specific resource requirements and monetary costs of creating and deploying applications on each platform? (iii) In light of those monetary and performance cost-benefits, how do these platforms compare? (iv) Can cloud computing platforms be used in combination with desktop grids to improve cost-effectiveness even further? The scientific analyses are usually computation intensive, hence taking a long time for execution. Workflow technologies can be facilitated to automate these scientific applications. Accordingly, scientific workflows are typically very complex. They usually have a large number of tasks and need a long time for execution. During the execution, a large volume of new intermediate data will be generated. They could be even larger than the original data and contain some important intermediate results. After the execution of a scientific workflow, some intermediate data may need to be stored for future use.

III. FORECASTING DATA SETS

We need a highly practical cost-effective runtime storage strategy in the cloud, which can solve the following two problems: 1) store the generated application data sets with a cost close to or even the same as the minimum cost benchmark, and 2) take users' (optional) preferences on storage into consideration. We utilize a algorithm, which was used for static on-demand minimum cost benchmarking of data sets storage in the cloud. We enhance the algorithm by incorporating users' (optional) preferences on storage that can offer users some flexibility. Based on the enhanced algorithm, we propose a runtime local-optimization-based strategy for storing the generated application data sets in the cloud.

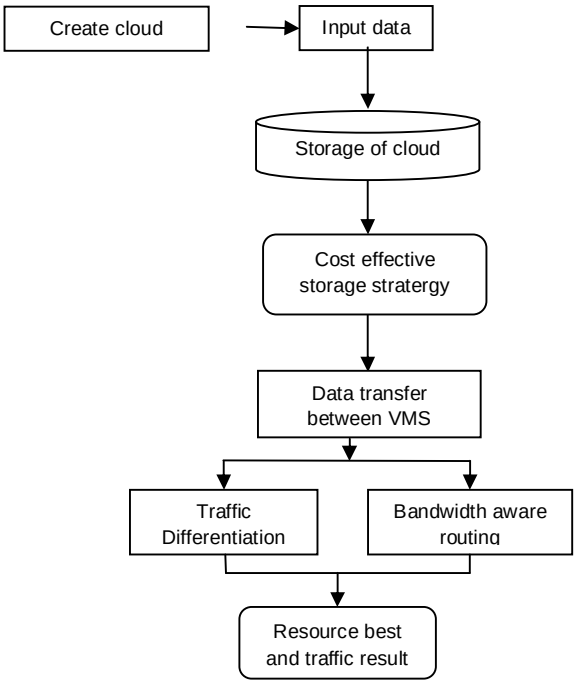

Fig 1: Data set storage strategy

3.1 Local Optimization

We introduce our local-optimization-based data sets storage strategy, which is designed based on the algorithm. The philosophy is to derive localized minimum costs instead of a global one, aiming at approaching the minimum cost benchmark with highly practical time complexity.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

3.2 Cost Transitive Tournament

We utilize a Cost algorithm, which was used for static on-demand minimum cost benchmarking of data sets storage in the cloud. We enhance an algorithm by incorporating users' (optional) preferences on storage that can offer users some flexibility. Based on the algorithm, we propose a runtime optimization-based strategy for storing the generated application data sets in the cloud.

3.3 Data dependency graph

Our DDG is based on data provenance, which depicts the dependency relationships of all the generated data sets in the cloud. With DDG, we can manage where the data sets are derived from and how to regenerate them. It is a acyclic graph based on data provenance in scientific application. Dataset once generated, whether it should be stored or deleted it should be stored in the data dependency graph.

IV. DATA SET COST REDUCTION ALGORITHM

We use the computation cost and storage cost to implement the algorithms. The communication cost also should be included in this, for including this cost the factors such as Jitter, Delay , Path availability and link availability they should find the minimum cost by using cost transitive algorithm , then in a cost effective way they should find the shortest path .

- This algorithm should have less time complexity.
- Should have a good practical runtime computation complexity
- Generation cost-based strategy, in which we store the data sets that incur the highest generation costs.
- Cost rate-based strategy reported in which we store the data sets by comparing their own generation cost rate and storage cost rate.

SPF Algorithm

Algorithm : SPF Algorithm
Input : Start dataset s , end dataset s_j
Output : Set of dataset

```
01  begin
02  genCost = 0;
03  for ( every dataset s_j, where s_i → s_j ;
04  create an edge
05  weight =0;
06  genCost = genCost+ d_n;
07  weight = weight + ( d_n +gencost )
08  set s_i , s_j >= weight
09  x = Set of dataset
10  return x
```

V. RELATED WORK

The work mentioned mainly focuses on the comparison of cloud computing systems and the traditional distributed computing paradigms, which shows that applications running in the cloud have cost benefits. They did not touch the issue of computation and storage tradeoffs in the cloud.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

5.1 Evaluation Setting

A simulation toolkit enables modeling and simulation of Cloud computing systems and application provisioning environments. The CloudSim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited effort. Currently, it supports modeling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). Moreover, it exposes custom interfaces for implementing policies and provisioning techniques for allocation of VMs under inter-networked Cloud computing scenarios. In this module we are creating cloud users and datacenters and cloud virtual machines as per our requirement

5.2 Problem analysis

Users can deploy their applications in unified resources without any infrastructure investment, where excessive processing power and storage can be obtained from commercial cloud service providers. With the pay-as-you-go model, the total application cost in the cloud highly depends on the strategy of storing the application data sets, e.g., storing all the generated application data sets in the cloud may result in a high storage cost, because some data sets may be rarely used but large in size; in contrast, deleting all the generated data sets and regenerating them every time when needed may result in a high computation cost.

5.3 Overall performance

It is introducing two new parameters that can represent users' preferences and provide users some flexibility in using the storage strategy. The two parameters are denoted as T and λ. T is the parameter used to represent users' tolerance on data accessing delay. Users need to inform the cloud service provider about the data sets that they have requirements on their availabilities. For a data set d_i, this needs regeneration, T_i is the delay time that users can tolerant when they want to access it.

λ is the parameter used to adjust the storage strategy when users have extra budget on top of the minimum cost benchmark to store more data sets for reducing the average data sets accessing time. Based on users' extra budget, we can calculate a proper value of λ[7], which is between 0 and 1. We multiply every data set d_i's storage cost rate (i.e., yi) by λ, and use it to compare with d_i's regeneration cost rate (i.e., genCost(d_i)*v_i) for deciding its storage status. Hence, more data sets tend to be stored, and literally speaking, data sets will be deleted only when their storage cost rates are (1/ λ) times higher than their regeneration cost rates.

$$(\forall d_i, d_j \in DDG \wedge d_i \rightarrow d_j) \Rightarrow \exists e<d_i, d_j>$$

To incorporate the parameter of data accessing delay tolerance

$$e<d_i, d_j> \Rightarrow \forall d_k \in DDG \wedge (d_i \rightarrow d_k \rightarrow d_j) \wedge \left(\frac{genCost(d_k)}{Price_{cpu}} < T_k\right).$$

Design uses two main techniques: traffic differentiation and bandwidth-aware routing. The first allows bulk transfers to exploit spare bandwidth without interfering with existing traffic, while the second achieves efficient use of the spare resources.

Traffic differentiation: Traffic differentiation is necessary to allow bulk transfers to use left-over bandwidth without affecting best-effort traffic. It separates traffic into best-effort traffic that is delay sensitive and bulk traffic that is delay tolerant. Best-effort traffic is forwarded without any change, while bulk traffic is forwarded with strictly lower priority, i.e., bulk traffic is sent only when there is no best-effort traffic waiting to be sent.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Bandwidth-aware routing: To achieve efficient use of spare resources, transit ISPs would have to modify the default routing within their networks. Intra-domain routing today is not optimized for bandwidth: ISPs do not use all possible paths between a pair of nodes, they do not necessarily pick the paths with the most available bandwidth, and they do not adapt their paths dynamically as the available bandwidths on the links vary. It addresses these limitations by employing bandwidth-aware routing that is optimized to pick potentially multiple paths between a pair of nodes to deliver the most data, independent of the paths' latencies. It also periodically recomputes its routes to account for changes in network conditions.

Performance Evaluation Compare the existing and proposed system with the following parameters Cost-effectiveness, Efficiency, data transferred.

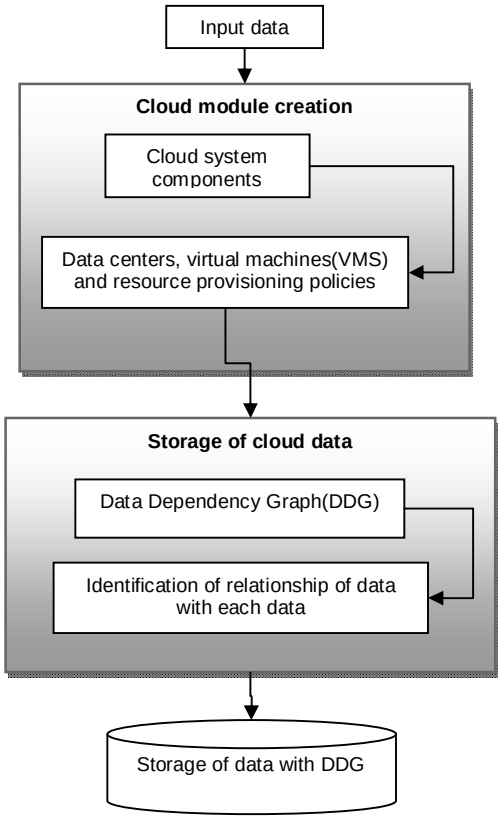

Fig 2 : Dataset storage storage diagram

VI. CONCLUSION

We have investigated the unique features and requirements of data sets storage in computation- and data intensive applications in the cloud. Toward practically achieving the minimum data sets storage cost in the cloud, we have developed a novel runtime local-optimization based storage strategy. The strategy is based on the enhanced linear CTT-SP algorithm used for the minimum cost benchmarking by taking into the consideration of users' (optional) preferences. Theoretical analysis, general random simulations, and specific case studies indicate that our strategy is

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

very cost-effective by achieving close to or even the same as the minimum cost benchmark with highly practical runtime efficiency.

Our current work is based on Amazon clouds' cost model and assumes that all the application data be stored with a single cloud service provider. However, sometimes large-scale applications have to run in a more distributed manner because some application data may be distributed with fixed locations.

REFERENCE

[1] S.K. Garg, R. Buyya, and H.J. Siegel, (2010)"Time and Cost Trade-Off Management for Scheduling Parallel Applications on Utility Grids," Future Generation Computer Systems, vol. 26, pp. 1344-1355

[2] P.K. Gunda, L. Ravindranath, C.A. Thekkath, Y. Yu, and L.Zhuang,(2010) "Nectar: Automatic Management of Data and Computation in Datacenters," Proc. Ninth Symp. Operating Systems Design and Implementation, pp. 1-14

[3] D. Yuan, Y. Yang, X. Liu, and J. Chen,(2010)"A Cost-Effective Strategy for Intermediate Data Storage in Scientific Cloud Workflows,"Proc. 24th Int'l Parallel and Distributed Processing Symp.

[4] D. Yuan, Y. Yang, X. Liu, and J. Chen(2011), "On-Demand Minimum Cost Benchmarking for Intermediate Data Sets Storage in Scientific Cloud Workflow Systems," J. Parallel and DistributedComputing, vol. 71, pp. 316-332

[5] D. Yuan, Y. Yang, X. Liu, G. Zhang, and J. Chen, (2012)"A Data Dependency Based Strategy for Intermediate Data Storage in Scientific Cloud Workflow Systems," Concurrency and Computation:Practice and Experience, vol. 24, pp. 956-976

[6] M. Zaharia, A. Konwinski, A.D. Joseph, R. Katz, and I. Stoica (2008) "Improving MapReduce Performance in Heterogeneous Environments," Proc. Eighth USENIX Symp. Operating Systems Design and Implementation, pp. 29-42

[7] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D.P. Anderson,(2009)"Cost-Benefit Analysis of Cloud Computing versus Desktop Grids," Proc. 23th Int'l Parallel and Distributed Processing Symp.

[8] D. Warneke and O. Kao,(2011) "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 6, pp. 985-997.

[9] L. Young Choon and A.Y. Zomaya,(2011) "Energy Conscious Scheduling for Distributed Computing Systems under Different Operating Conditions," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 8, pp. 1374-1381

[10] Ashutosh Ingole Sumit Chavan Utkarsh Pawde (2011) An Optimized Algorithm for Task Scheduling based on Activity based Costing in Cloud Computing ," 2nd National Conference on Information and Communication Technology (NCICT) Proceedings published in International Journal of Computer Applications® (IJCA).

[11] Van den Bossche, R., Vanmechelen, K., Broeckhove, J(2010).: Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In: 3rd International Conference on Cloud Computing, pp. 228–235

[12] Agrawal, p., Kifer, d., and Olston(2008),, C. Scheduling shared scans of large data files. Proc. VLDB Endow. 1, 1, 958–969.

[13] Agrawal, s., Chaudhuri, s., and Narasayya, (2000), V. R.Automated selection of materialized views and indexes in SQLdatabases. In VLDB, pp. 496–505.

[14] K. Kalpakis, K. Dasgupta, and O. Wolfson,(2001) "Optimal Placement of Replicas in Tree with Read, Write, and Storage Costs," IEEE Trans. Parallel and Distributed Systems, vol. 12, no. 6, pp. 628-636.

[15] D. Anderson. Boinc(2004): A system for public-resource computing and storage. In Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, Pittsburgh, USA

[16] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan.(2005) Interpreting the Data: Parallel Analysis with Sawzall. Sci. Program., 13(4):277–298

[17] Raicu, I. Foster, and Y. Zhao. (2008) Many-Task Computing for Grids and supercomputers. In Many-Task Computing on Grids and Supercomputers. MTAGS Workshop.

[18] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, (2008)"TheCost of Doing Science on the Cloud: the Montage example," in ACM/IEEE Conference on Supercomputing, Austin, Texas

[19] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, (2009)"Cost-benefit analysis of Cloud Computing versus desktop grids," in IEEE International Symposium on Parallel & Distributed Processing, IPDPS'09

[20] A.Malik, A. Park, and R. Fujimoto,(2009) "Optimistic Synchronization of Parallel Simulations in Cloud Computing Environments," Proc. IEEE Int'l Conf. Cloud Computing (CLOUD '09), pp. 49-56