RESEARCH REVIEW JOURNALS
Volume-04
Issue-06
June-2019

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com[UGC Listed Journal]

A Study of Cost Optimization of Cloud Data Storage

[1]AlokTuli & [2]Dr. C. Kavitha

[1]*Research Scholar, Sri Satya Sai University, Sehore, Bhopal (India)*
[2]*Research Guide, Sri Satya Sai University, Sehore, Bhopal (India)*

ARTICLE DETAILS

Article History
Published Online: 12 June 2019

Keywords
Cloud Computing, Storage Systems, Cost Models, Cloud services and environments.

ABSTRACT

Cloud becomes more attractive and easy to use for everyone and people are more familiar with provided services and environments. One of the main aims of large-scale distributed systems such as the Cloud is to solve the problem of storage, data availability, and especially their security in public and shared environments where providers impose costs for end-users. In this paper we propose a cost minimization model based of Cloud Data Storage. The main scope is to find a cost efficient storage scheme for many heterogeneous data blocks using multiple public Cloud storage providers. The proposed model is validated through simulation using a realistic scenario: CyberWater -cyber infrastructure for natural resource management.

1. Introduction

The most significant point of cloud computing is that the resources and data are accumulated into data centers on the internet. These days, the cloud services like IaaS, PaaS & SaaS, have been improved in execution as application execution environments are aggregated at several levels for sharing. The ad-hoc data is stored in cash registers. Then, this stored data is analyzed with the help of the time-series. Hence, the behavior like purchasing behavior of individuals is analyzed from this ad-hoc data. According to a report, about 7 million pieces per second are accumulated at cloud centers. This ad-hoc data is not equivalent to that is obtained in reality because of the fact that much of the data is lost while moving to the cloud centers. Many research are going on in order to reduce this data leakage. In today's world, several kinds of data are accumulated in a cloud environment as the cost of devices of information and communication technology is decreasing day by day. There is an urgent need to analyze this massive data so that it can be helpful for the business and society.

A new technology needs to be adapted as the quantity of data is so massive which is far more than tens of terabytes or tens of petabytes. Also, these days, social infra structure services run for 24 hours and 7 days a week. Hence, there is an urgent need to change the configuration of system dynamically. Many laboratories are developing fundamental technologies for processing ad-hoc data in a cloud environment. A new methodology has been introduced to create cloud by aggregating data. So now there is a need to change the role of cloud from application aggregation to ad-hoc data aggregation and utilization. A new technology other than information and communication technology is needed to use this kind of ad-hoc data which is of more than tens of petabytes. Now, the scenario of cloud environment has expanded from information & communication technology applications to business processes to innovation. The aim is to increase sales by identifying valuable information via data analysis aggregated into clouds.

The most significant point of cloud computing is that the resources and data are accumulated into data centers on the internet. These days, the cloud services like IaaS, PaaS & SaaS, have been improved in execution as application execution environments are aggregated at several levels for sharing. Ad-hoc data processing is a powerful abstraction for mining terabytes of data. Systems for massive parallel data processing, such as MapReduce and Dryad allow Internet companies, e.g., Google, Yahoo, and Microsoft, to mine large web crawls, click streams, and system logs across shared-nothing clusters of unreliable servers.In today's world, technology is growing at a very faster speed. A variety of data needs to be processed as the applications like social network analysis; semantic web analysis and bio-informatics network analysis are growing rapidly. It is like a big challenge to analyze ad-hoc data. Several Governments and industries have shown their interest in ad-hoc data. This research work introduces several ad-hoc data processing techniques from system as well as application aspects. There are many big social environments like online shopping sites, social sites etc. Companies need to track the activities of users. There are many issues like computing platform, cloud architecture, cloud database and data storage scheme. These issues need to be solved by analyzing ad-hoc data. In this research work, we discuss the data processing in cloud computing environments and issues & challenges related to this.

2. Review of literature

Baron et al. (2012)[1] proposed a method, Pregel, which is used to implement a programming model. In this model, each node has its own input and transfers only some messages which are needed for the next iteration to other nodes.

Ranger et al. (2014)[2] proposed a 3-stage approach for end-to-end set-similarity joins. They efficiently partition the data across nodes in order to balance the workload and minimize the need for replication.

Kraska et al. (2012)[3] investigated how to perform kNN join using MapReduce. Mappers cluster objects into groups, then Reducers perform the kNN join on each group of objects separately.

Cordeiro et al. (2013)[4] described that to reduce shuffling and computational costs, they design an effective mapping mechanism that exploits pruning rules for distance filtering. In addition, two approximate algorithms minimize the number of replicas to reduce the shuffling cost.

J. Ekanatake et al. (2012)[5] proposed a method, Twister, which is an incremented MapReduce runtime which supports Repetitive MapReduce calculations efficiently.

Jain et al. (2010)[6] highlighted that MapReduce is used to add an extra Combine stage after Reduce stage. Thus, the output of data moves from Combine stage to next iteration's map stage.

John et al. (2013)[7]proposed another method called, HaLooP, which is quite similar to Twister. HaLoop is in fact, a modified version of the MapReduce framework which supports the iterative applications by adding a 'Loop Control'. It permits to save more input and outputs during iterations. There exist a lot of iterations during the processing of ad-hoc data.

D. Vernica et al. (2012)[8] presented four different architectures which were based on classic multi-tier database application architecture. These four architectures are: Partitioning, Replication, Distributed Control and Caching Architecture.

Robert et al. (2012)[9] observed that different providers have different business models and different kinds of applications are targeted by them. For example, Google, mostly, launches small applications having light work load whereas Azure launches the applications which are efficient for medium to large services.

Nylael et al. (2011)[10] described that most of the cloud service providers utilize hybrid architecture. This hybrid architecture has the potential to satisfy the actual service requirements.

Tonetta et al. (2013) [11] proposed BoW methods. In this method, MapReduce is used to cluster very large and multi-dimensional datasets.

Cimatti et al. (2012)[12] described that BoW method permits the automatic and dynamic communication between Disk Delay and Network Delay. MapDup Reducer is a MapReduce based system which has the capability to detect near duplicates over massive datasets effectively.

Darimont et al. (2012)[13] implemented the MapReduce framework on a number of processors in a single device.

Fisman at al. (2011)[14] developed Mars which is a MapReduce framework and is based on GPS. It enhances the efficiency of the system.

Fantechi et al. (2012)[15] proposed a sharing framework which is known as MRShare. MRShare is used to convert a new group that can be executed more effectively by aggregating tasks into groups and evaluating each group as a single query.

Liu et al. (2012)[16]proposed a method to reduce the data transfer cost. This method divides a MapReduce task into two sub-tasks : Sampling MapReduce Task and Expected MapReduce Task.

Carlo et al. (2013)[17] described that in first task, input data is obtained, keys are distributed and a good partition scheme is prepared. In second task, expected MapReduce task is used to perform the partition scheme to group the intermediate keys quickly.

Jeffords et al. (2014)[18] described that the biggest feature of innovation is that the users don't know what to do which differentiates it from traditional ICT application.

Jackson et al. (2014)[19] described that there are many methods to analyze ad-hoc data. The process of data analysis must be repeated a number of times from several prospective. Also, a processing having high speed and low cost is needed in all stages of development and operation.

Mannaet al. (2014)[20] described that cloud environment is the better option to analyze ad-hoc data as it offers benefits like temporary availability of a large number of computational resources and cost reduction by allowing resources to share data.

3. Definitions and explanation terms

Ad-hoc data processing architecture: Ad-hoc data processing architecture will be used for the current research work. Since much of the ad-hoc processing tasks programmers' uses are agreeable to incremental calculation, there is a hefty prospect to reprocess preceding computations and shun intermediary outcome. However, current ad-hoc data processing architectures require the programmer to overtly split modules into sub-modules to make pipelines.

MapReduce: For MapReduce, even these can abscond big events for use again on the table. In this research, we deal with these challenges by implementing a trendy ad-hoc data processing abstraction, MapReduce, over a distributed stream processor.

4. Cost optimization in cloud computing:

As in cloud computing there are two main actors involved, there are two sides of cost optimization: cost optimization performed by providers and cost optimization performed by users. Cost optimization performed by cloud providers mainly focuses on minimizing the cost to maintain a physical data center. The cost minimization is typically achieved by reducing electricity consumption. A proposed approachinvolves dynamically halting network devices. Another study proposes architectural principles, algorithms, and resource allocation policies for energy savings. Conversely, one of the most popular techniques for cost optimization executed by cloud users is to choose the correct balance the types of instances, i.e. cloud infrastructure planning.

5. Cost optimization for data storage:

More and more people from around the world produce digital data in their daily life (such as photo, sounds and videos, documents, etc.), which leads to an exponential growth creating a challenge for storage and management. Moreover, at the business level, organizations such as a company need to store and access their data and making a part of it public to the customers. A global vision of smart cities place the Public Cloud Storage services becomes a fundamental part of platform architecture for such systems. In any scenarios, data must be available 24x7 in any location and from any device. A possible solution for this challenge is represented by Cloud Computing Services. In Cloud computing everything is provided as a service to end-user, in a functional, usable and extremely powerful manner, permitting to use software resources in a pay-per-use manner. In this way the user has a great flexibility to adapt to changes.

With the adoption of Cloud storage, there are two sides of the cost optimization problem: first the Cloud storage provider must calculate his total cost of ownership and adequately put price on his services in order to have profit and amortize his investment and, second, a Cloud user must calculate total cost of storing data in the Cloud and minimized it as much as possible. Different components and their owners with claims for Storage as a Service are presented in Table 1.

Table 1. Total cost of ownership perspective

Component	Storage as a Service
Business Process	Customer
Business Logic	
Middleware Management	Cloud Storage Provider
Application Licensing/ Support	
OS Management	
OS Licensing / Support	
Server/Storage/Networks HW / Maintenance	
Domestic Utilities	
Maintenance Equipment	
Real Estate	

A major problem with Cloud storage services is represented by the "vendor lock-in" problem, which refers to dependence solely on a particular Cloud storage provider. Switching from one provider to another can be expensive as Cloud storage providers charge inbound, outbound bandwidth and requests of data. A client who wants to move from one provider to another must pay twice the bandwidth and in addition for the actual cost of online storage. The authors of propose a secured cost-effective multi-Cloud storage model in Cloud computing which holds an economical distribution of data among the available in the market, to provide customers with data availability as well as secure storage. Also the authors of propose application of RAID-like techniques at Cloud storage level, meaning striping of user data across multiple providers. We came with a model based on binary linear programming, which use real information and real scenarios, aiming to offer the best storage scheme with minimum cost.

6. Cost optimization in multi-site multi-cloud environments:

Cloud computing is growing in size and scope, with enterprises developing, testing and deploying their services on multiple sites and cloud providers. Such a distribution entails distribution of data, functionality or both. This model has resulted in a large increase in the number of cloud providers, whose data centers are also geographically distributed. The range of requirements has also resulted in a multitude of service offerings with multiple flexible (and dynamic) pricing schemes from these providers. At the same time, recent outages from cloud providers such as Amazon's EC2 has also reminded users that it is important to select multiple cloud providers so as to minimize dependency on a single provider. The complexity of managing multi-site multi cloud deployments has driven the growth of a host of service providers that "manage" multi-cloud deployments like enStratus, New Relic, RightScale, Standing Cloud. However none of these service providers do deployment planning to optimize the cost of resource acquisition and usage - their focus lies in operational management of a given deployment. This paper therefore formalizes the selection of cloud providers and the matching of a deployment site to a cloud provider and looks to optimize the Total Infrastructure Cost(TIC) under the conditions of dynamic pricing for a term deployment (of one year). Our placement model includes existing cloud price and resource requirements of sites with constraints. The problem is formulated as a linear optimization problem which is NP complete. We present a heuristic that comes close to the optimal solution.

7. Conclusion

Cost optimization when buying cloud storage capacity from public Cloud providers. We can conclude that the final gain depends on initial demand of data, however in some cases is surprisingly high (e.g. 80%), so this method can be successfully used for any scenarios considering public Clouds.For this current research, multiple crawlers discover displace areas of the web while an incessant MapReduce job builds each local index. The system converts user queries into continuous, in-network, MapReduce tasks to query the global index. A hallucination module describes the outcome of user queries, the rate of growth of the index across the system, and the impact of node failure on outcomes. Experience with our prototype signifies that wide-area incremental MapReduce is an influential method for handling data in the cloud.A

MapReduce job breaks data processing modules into two phases: map and reduce. The map function works on individual key-value pairs, and outputs a new pair. The system creates a list of values, for each key k2. The reduce function then creates a final valuefrom each key-value list pair.

References

1. Baron, “The importance of ‘ad-hoc data’: A definition,” 2012.
2. C. Ranger, “Ad-hoc data: science in the petabyte era,” Nature 455 (7209): 1, 2014.
3. D. Kossmann, T. Kraska, and S. Loesing, “An evaluation of alternative architectures for transaction processing in the cloud,” in Proceedings of the 2012 international conference on Management of data. ACM, 2012, pp. 579–590.
4. F. Cordeiro, J. Dean, S. Ghemawat, W. Hsieh, D. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber, “Bigtable: A distributed structured data storage system,” in 7th OSDI, 2013, pp. 305–314.
5. J. Ekanatake, “The hadoop distributed file system: Architecture and design,” Hadoop Project Website, vol. 11, 2012.
6. Jain, “A survey of large scale data management approaches in cloud environments,” Communications Surveys & Tutorials, IEEE, vol. 13, no. 3, pp. 311–336, 2010.
7. John Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2013.
8. R. Vernica, “Es2: A cloud data storage system for supporting both oltp and olap,” in Data Engineering (ICDE), 2012 IEEE 27th International Conference on. IEEE, 2012, pp. 291–302.
9. Robert and R. Katz, “Chukwa: A system for reliable large-scale log collection,” in USENIX Conference on Large Installation System Administration, 2012, pp. 1–15.
10. T. Nylael, “The Google file system,” in ACM SIGOPS Operating Systems Review, vol. 37, no. 5. ACM, 2011, pp. 29–43.
11. Cimatti, M. Roveri, and S. Tonetta. PSL Symbolic Compilation. IEEE Trans. on CAD of Integrated Circuits and Systems, 27(10):1737–1750, 2013.
12. Cimatti, M. Roveri, and S. Tonetta. Requirements Validation for Hybrid Systems. In CAV 2009, LNCS, pages 188–203. Springer, 2012.
13. R. Darimont, E. Delor, P. Massonet, and A. van Lamsweerde. GRAIL/KAOS: an environment for goal-driven requirements engineering. In ICSE’97, pages 612–613. ACM, 2012.
14. Eisner and D. Fisman. A Practical Introduction to PSL. Springer-Verlag, 2011.
15. Fantechi, S. Gnesi, G. Ristori, M. Carenini, M. Vanocchi, and P. Moreschini. Assisting Requirement Formalization by Means of Natural Language Translation. Formal Methods in System Design, 4(3):243–263, 2012.
16. Fuxman, L. Liu, J. Mylopoulos, M. Roveri, and P. Traverso. Specifying and analyzing early requirements in Tropos. Requirements Engineering, 9(2):132–150, 2012.
17. Carlo Ghezzi, Dino Mandrioli, and Angelo Morzenti. Trio: A logic language for executable specifications of real-time systems. Journal of Systems and Software, 12(2):107–123, 2013.
18. L. Heitmeyer, R. D. Jeffords, and B. G. Labaw. Automated consistency checking of requirements specifications. Trans. Softw. Eng. Methodol., 5(3):231–261, 2014.
19. Jackson. Alloy: a lightweight object modeling notation. ACM Trans. Softw. Eng. Methodol., 11(2):256–290, 2014.
20. Z. Manna and A. Pnueli. The Temporal Logic of Reactive and Concurrent Systems, Specification. Springer, 2014.

