International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 20 Issue 3 – MARCH 2016.

AGGRESSIVE CONVENTION FOR SOURCE REGENERATION OF SCIENTIFIC DATASETS IN THE CLOUD

Mr.R.Ramesh Kannan [1], Ms V.Kiruthika[2], ,Ms G.Sangeetha[3]

[1]Assistant Professor, Department of CSE, Dhanalakshmi College of Engineering. Chennai- 601301

[2,3] Student Department of CSE, Dhanalakshmi College of Engineering. Chennai- 601301

kannanrrk@gmail.com
kiruthikasekar1594@gmail.com
sangee2895@gmail.com

***Abstract*--The aim of this study is to analysis and presentation of some ideas on performance testing in Cloud Computing. Performance is an important factor in testing a web application. Performance testing in cloud computing is different from that of traditional applications. Here the file which is send to receiver should be analyzed previously for cloud space network traffic using Benchmark. Our research methodology in this article includes an overview of existing works on testing performance in Cloud Computing, focusing on discussion that the traditional benchmarks are not sufficient to analyze performance testing in Cloud Computing. In this study we are focused mainly on analysis performance metrics in Cloud Computing, based on their characteristics such as elasticity, scalability, pay-per-use and fault tolerance. Then we discuss why needed new strategies for performance testing in Cloud Computing and creation of new benchmarks. From this study we conclude that the performance testing and evaluation should be performed using new models testing, which are created according to Cloud Computing characteristics and metrics.**

Keyword- Data Dependency Graph(DDG), Datasets storage and regeneration, Minimum cost Benchmark, Partitioned Solution Space(PSS),

I .INTRODUCTION

IaaS (Infrastructure as a Service) is a very popular way to deliver services in the cloud, where users can deploy their applications in unified cloud resources such as computing and storage services without any infrastructure investments. However, along with the convenience brought by using on-demand cloud services, users have to pay for the resources used according to the pay-as-you-go model, which can be substantial. Especially, nowadays applications are getting more and more data intensive , e.g. scientific applications, where the generated data are often gigabytes, terabytes, or even petabytes in size. These generated data contain important intermediate or final results of computation, which may need to be stored for reuse and sharing. Storing all the generated application datasets in the cloud may result in a high storage cost since some datasets (e.g. intermediate results) may be never reused but large in size. Hence, there is a trade-off between computation and storage for storing generated application datasets in the cloud. Based on this trade-off, different storage strategies are designed for the generated datasets in order to reduce the total application cost in the cloud.

The benchmark referred in this paper is the minimum cost for storage and regeneration of the datasets for the required users for the renewable of backup, which is used to evaluate the cost effectiveness of storage strategies used in cloud applications. Due to the dynamic provisioning mechanism in cloud computing, this minimum cost varies from time to time whenever new datasets are generated or the dataset's usage frequencies are changed. .In this paper, by studing the issue of computation and storage trade-off for the required users, we describe a novel Aggressive convention for source regeneration of scientific datasets in the cloud approach with highly efficient algorithms that can calculate the minimum cost for datasets storage in the cloud,so that a effective cloud would be created . In the approach we update a Partitioned Solution Space (PSS), which saves all the possible minimum cost storage strategies of the datasets in the cloud. Therefore, whenever the application cost changes at runtime, our benchmarking algorithm can dynamically derive the new minimum cost from the PSS to keep the benchmark updated from user to user. Hence this approach can be utilised on-the-fly to either proactively report the dynamic minimum cost benchmark to SaaS providers or instantly respond to their benchmarking requests. Experimental results show the excellent efficiency, scalability and practicality of our approach.

II .RELATED WORKS

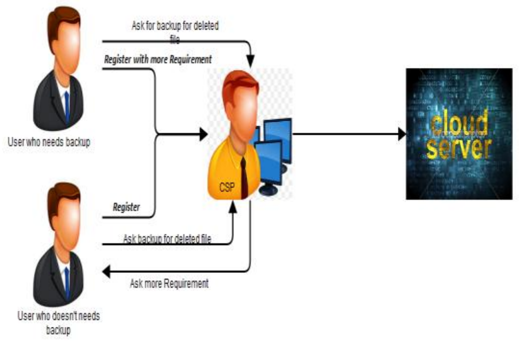
Cloud computing system for scientific applications, i.e. science cloud, has already commenced . This paper is mainly inspired by the work in two research areas: cache management and scheduling. With smart caching mechanism system performance can be greatly improved. The similarity is that both pre-store some data for future use, while the difference is that caching is to reducing data accessing delay but our work is to find the minimum application cost. Some works in scheduling focus on reducing various costs for either applications or systems. However these works mainly focus on resource utilisation rather than the perspective of the trade-off between computation and storage. This trade-off is a unique issue in the cloud due to the pay-as-you-go model, hence our benchmarking approach is brand new which is different from other existing benchmarking counterparts in the cloud .

Researchers are exploring the cost-effectiveness of the cloud, because comparing to the traditional computing systems like cluster and grid, a cloud computing system has a cost benefit in various aspects. Assunção et al demonstrate that cloud computing can extend the capacity of clusters with a cost benefit. With Amazon clouds' cost model and BOINC volunteer computing middleware, the work in analyses the cost benefit of cloud computing versus grid computing. The work by Deelman et al. also applies Amazon clouds' cost model and demonstrates that cloud computing offers a cost-effective way to deploy scientific applications. The above works mainly focus on the comparison of cloud computing systems and the traditional computing paradigms, which shows that applications running in the cloud have cost benefits, but they do not touch the issue of computation and storage trade-off for datasets in the cloud.

Due to the importance of data provenance in scientific applications, many works about recording data provenance of the system have been done . Recently, research on data provenance in cloud computing systems has also appeared . More specifically, Osterweil et al. present how to generate a data derivation graph for the execution of a scientific workflow, where one graph records the data provenance of one execution, and Foster et al. propose the concept of Virtual Data in the Chimera system, which enables the automatic regeneration of datasets when needed. Our DDG is based on data provenance in scientific applications, which depicts the dependency relationships of all the datasets in the cloud. With DDG, we know where the datasets are derived from and how to regenerate them.

In Deelman et al. present that storing some popular intermediate data can save the cost in comparison to always regenerating them from the input data. Adams et al. propose a model to represent the trade-off between computation cost and storage cost, but have not given the strategy to find this trade-off proposes practical cost-effective strategies for datasets storage of scientific applications, but it is not the minimum cost storage strategy in the cloud. Yuan et al. propose the CTT-SP algorithm with a polynomial time complexity (i.e. O(n9)) that can calculate the minimum cost of storing scientific datasets in the cloud with fixed usage frequencies. However, this algorithm is only suitable for static scenarios. Whenever datasets' usage frequencies are changed or new datasets are generated, we have to run the CTT-SP algorithm on all datasets stored in the cloud to calculate the new benchmark, which is not efficient. Hence, it can only be utilised for ondemand minimum cost benchmarking.

In this paper, we propose an entirely new approach for dynamic on-the-fly minimum cost benchmarking of datasets storage in the cloud. Significantly different from the on-demand benchmarking approach , we divide the DDG into small segments and pre-calculate all the possible minimum cost storage strategies (i.e. the solution space) of every DDG segment using the CTT-SP algorithm. By utilising the pre-calculated results, whenever new datasets are generated and/ or existing datasets' usage frequencies are changed, we develop efficient algorithms that can dynamically calculate the changing minimum cost benchmark at runtime by calling the CTT-SP algorithm only on the local DDG segment. By keeping the minimum cost benchmark updated on the fly, SaaS providers' benchmarking request can be instantly responded. Hence, it is a practical approach for dynamic minimum cost benchmarking of storing generated datasets in the cloud. The efficiency and scalability of the PSS based dynamic benchmarking algorithms comparing to the original CTT-SP algorithm for on-demand benchmarking .


Ask for backup for deleted file
Register with more Requirement
User who needs backup
CSP
cloud server
Register
Ask backup for deleted file
Ask more Requirement
User who doesn't needs backup

III . ALGORITHM

3.1:CTT-SP ALGORITHM:

The CTT SP algorithm compares all possible paths through the graph between each pair of vertices. It is able to do this with $\Theta(|V|^3)$ comparisons in a graph. This is remarkable considering that there may be up to $\Omega(|V|^2)$ edges in the graph, and every combination of edges is tested.

Consider a graph G with vertices V numbered from 1 through N.Further consider a function shortestPath(i,j,k) that returns the shortest possible path fron i to j using vertices only from the set{1,2,…,k} as intermediate points along the way. Now, given this function, our goal is to find the shortest path from each i to each j using only vertices from 1 to K+1.

For eachof these pair of vertices, the true shortest path could be either

(1) A path that only uses vertices in the set{1,2,….,k}
(2) A path that goes from i to K+1 and then from K+1 to j.

We know that the best path from i to j

only uses vertices 1 through k is defined by shortest path(i,j,k), and it is clear that if there were a better path from i to k+1 and k+1 to j,then the length of this path would be the concatenation of the shortest path from i to k+1(using vertices in {1,…k})and the shortest path from k+1 to j (also using vertices in{1,….k}).

If w(i,j) is the weight of the edge between vertices i and j, we candefine shortest Path(i,j,k+1) in terms of the following recursive formula:the base case is

shortestPath(i,j,0)=w(i,j)

and the recursive case is

shortestPath(i,j,k+1)=min(shortestPath(i,j,k),
shortestpath(i,k+1,k)+shortestPath(k+1,j,k))

3.2:BRUTE-FORCE ALGORITHM:

Brute-force:

Brute-force (checking every element with every other element) takes O(n^2).

Sorting:

Sorting takes O(n log n), which is generally considered to be a fairly decent running time.Sorting has the advantage above the below (hash table) approach in that it can be done in-place (O(1) extra space), where-as the below takes O(n) extra space.

Hash table:

An alternative is to use a hash table.
For each item:Check if that item exists in the hash table (if it does, all items are not distinct) and
Insert that item into the hash table
Since insert and contains queries run in expected O(1) on a hash table, the overall running time would be expected O(n), and, as mentioned above, O(n) extra space.

IV .PSEUDOCODE FOR BRUTE-FORCE

```
VertexCover(edges, k):
  if edges = {}
    # we win
    return true
  else if k = 0:
    # we lose, keep trying
    return false
  else:
    for each endpoint x of edges[1]:
      let edges' = { e in edges : x is not an endpoint of e }
      if VertexCover(edges', k-1) = true
        # we found one, stop looking
        return true
    # else we didn't find one, keep trying
return false
```

CLOUD SERVER:

In some respects cloud servers work in the same way as physical servers but the functions they provide can be very different. When opting for cloud hosting, clients are renting virtual server space rather than renting or purchasing physical servers. They are often paid for by the hour depending on the capacity required at any particular time.

Traditionally there are two main options for hosting: shared hosting and dedicated hosting. Shared hosting is the cheaper option whereby servers are shared between the hosting provider’s clients. Dedicated hosting is a much more advanced form of hosting, whereby clients purchase whole physical servers. Dedicated servers allow for full control over hosting.

CLIENT REGISTRATION:

In this module the new client register their details to be a member of the Cloud server. Client must give their username and password for login purpose. Using these username and password only they can able to login.

BACKUP AND DELETE:

Cloud has splitted into normal cloud and virtual cloud. Virtual cloud is created to hold the backup of the deleted file for future use purpose.

User may mention their details regarding their backup. If a user wants to register with the backup details, user need to pay more amout than required for space. Once they registered like that user may able to get the file backup even through they have deleted the file.Moreover tat user may not to pay more amount for that.And if the user have only registered with the required details and are deleted any file and to get it back they may have to pay more amount to get it back.

FILE UPLOADING:

In this module the Cloud user can take their Personal backup in Cloud Server and the data are stored in the Upload Folder. Server not has the authentication to delete uploaded files. Server can able to monitor the all the uploaded files and information of uploders and other Clients.

FILE DOWNLOADING:

In this module the user downloads their file from the cloud server by using their account. If the user is stored their backup in cloud server they can download only from cloud server if the requested file is there.

FILE SIZE FILTER:

Most of the files in the PC dataset are tiny files that less than 10KB in file size, accounting for a negligibly small percentage of the storage capacity.
To reduce the metadata overhead, ALG-Dedupe filters out these tiny files in the file size filter before the deduplication process, and groups data from many tiny files together into larger units of about 1MB each in the segment store to increase the data transfer efficiency over WAN.

APPLICATION-AWARE DEDUPLICATOR:

After data chunking in intelligent chunker module, data chunks will be deduplicated in the application-aware deduplicator by generating chunk fingerprints in the hash engine and detecting duplicate chunks in both the local client and remote cloud.

ALG-Dedupe strikes a good balance between alleviating computation overhead on the client side and avoiding hash collision to keep data integrity.

V .CONCLUSION

In this paper, we have examined the unique issues of storing application datasets in the cloud and analysed the re-quirements of Aggressive convention for source regeneration of scientific datasets in the cloud. We have updated a novel PSS (Partitioned So-lution Space) based practical approach with innovative algorithms that can dynamically calculate the minimum cost benchmark for storing generated application datasets in the cloud from user to user, which achieves the best trade-off between computation cost and storage cost of the cloud resources for renewable of backup. Both theoretical analysis and experimental results demon-strate that our novel dynamic minimum cost benchmark-ing approach is highly efficient and scalable. Hence, it can be practically utilised on the fly at runtime in the cloud, which was unavailable before.

REFERENCES

[1]"Amazon Cloud Services”,, http://aws.amazon.com/.

[2] "Eucalyptus", http://open.eucalyptus.com/.

[3] "Nimbus", http://www.nimbusproject.org/.

[4] "OpenNebula", http://www.opennebula.org/.

[5] I. Adams, D. D. E. Long, E. L. Miller, S. Pasupathy, and M. W. Storer, "Maximizing Efficiency by Trading Storage for Computation," in Workshop on Hot Topics in Cloud Computing (HotCloud'09), pp. 1-5, 2009.

[6] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Communication of the ACM, vol. 53, pp. 50-58, 2010.

[7] M. D. d. Assuncao, A. d. Costanzo, and R. Buyya, "Evaluating the Cost-Benefit of Using Cloud Computing to Extend the Capacity of Clusters," in 18th ACM International Symposium on High Performance Distributed Computing (HPDC'09), pp. 141-150, 2009.

[8] R. Bose and J. Frew, "Lineage Retrieval for Scientific Data Processing: A Survey," ACM Computing Surveys, vol. 37, pp. 1-28, 2005.

[9] A. Burton and A. Treloar, "Publish My Data: A Composition of Services from ANDS and ARCS," in 5th IEEE International Conference on e-Science (e-Science'09),, pp. 164-170, 2009.

[10] 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 5th Utility," Future Generation Computer Systems, vol. 25, pp. 599-616, 2009.

[11] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking Cloud Serving Systems with YCSB," in 1st ACM Symposium on Cloud Computing (SOCC'10), pp. 143-154, 2010.

[12] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and eScience: An Overview of Workflow System Features and Capabilities," Future Generation Computer Systems, vol. 25, pp. 528-540, 2009.

[13] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, "The Cost of Doing Science on the Cloud: the Montage Example," in ACM/IEEE Conference on Supercomputing (SC'08), pp. 1-12, 2008.

[14] I. Foster, J. Vockler, M. Wilde, and Z. Yong, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," in 14th International Conference on Scientific and Statistical Database Management, (SSDBM'02), pp. 37-46, 2002.

[15] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in Grid Computing Environments Workshop (GCE'08), pp. 1-10, 2008.

[16] J. Zhan, L. Wang, X. Li, W. Shi, C. Weng, W. Zhang, and X. Zang, "Cost-aware Cooperative Resource Provisioning for Heterogeneous Workloads in Data Centers," IEEE Transactions on Computers, vol. 62, pp. 2155-2168, 2013.