

Elastic Storage with Postpaid Service

T. Y. J. Naga Malleswari¹, S. Usha Sukhanya², A. Nithya Kalyani³, S. Girija⁴ and Shashank⁵

ABSTRACT

Cloud computing model provides cloud computing resources as a RaaS. Turnaround time and cost is mainly concerned with this model. Amazon offers spot instances with less expense. Resource unavailability leads to ongoing execution interruption, thereby reducing the performance. EAIC model handles spot instance revocation to reduce cost and turnaround time. No Hardware failure occurs as the allocation of SI hardware is not done. EAIC skips delta compression of Checkpoint files. EAIC send checkpoint files periodically. Checkpointing avoids loss of the work by leveraging to utilize the resources until the completion of execution and with less cost.

Keywords: Cloud Computing, Checkpointing, Cloud Service Providers, RaaS, Spot instances.

1. INTRODUCTION

Cloud computing [1] is a pool of resources provided on demand as a utility to the user through the concept of virtualization. Elasticity is the main feature of cloud computing. Client's applications are hosted by other cloud service providers. Many cloud service providers like Amazon, Google, Yahoo, VMware, etc, providing the resources as a service for fewer prices based on SLA. Without much interference of human client can manage its task. Cloud provides three essential services.

- SaaS - CRM, video conference, accounting, etc., are the SaaS applications by which the customer is free of maintaining software by hosting his applications [2].
- PaaS – Google App Engine is PaaS provided by Google. In PaaS applications, web services are developed tested, hosted by the cloud without downloading and installing the software [3].
- IaaS - It is a model where cloud service providers sell server equivalent virtual machines as per user requirements. Amazon EC2 is an example.
- RaaS - Cloud service providers provide CPU, Memory, IO resources on demand for a period of time. Amazon EC2 provides resource as a service [4].

Amazon provides the following types of instances [5]:

- On Demand provides instances on an hourly basis. Computing capacity can be raised up or down based on user requirement. It is useful for a long term commitment and flexible applications with low cost. Interruptions of applications due to heavy workload are not done but sometimes, instances may not be in the specific zones.
- Reserved instances will be in a specific zone and are suitable for steady applications where the workload is predictable.
- Spot instances (SI) are on an hourly basis and less expensive than On-Demand instances. Instances have spot price based on the availability of zone and demand. These are suitable for applications that have a heavy workload for a specific period. Checkpointing plays a vital role in a better overall

^{1,2,3,4,5}Department of Computer Science and Engineering, SRM University, Chennai, *Emails: nagamalleswari.t, ushasukhanya.s, nithyakalyani.a, girija.s}@ktr.srmuniv.ac.in, shanu_2352@yahoo.com*

performance of resources during a hardware failure or SI revocation. Due to this, execution continues from the last checkpoint as the program states are saved.

In multi-level Checkpointing [6], [7], [8] local failures are handled by transferring check point files to the remote storage. Incremental Checkpointing [9], [10], [11], [12], [13] is a technique where modified pages are saved over a period of interval. Delta compression [14] is used to reduce the size of checkpoint files. These two techniques are used in multi-level Checkpointing. Adaptive Incremental Checkpointing (AIC) is implemented on single threaded programs on multi-cores by which turnaround time of an application is reduced. For SI RaaS cloud EAIC uses AIC [15] for multi-threaded programs. This paper gives the details of how enhanced Adaptive Incremental Checkpointing (EAIC) is used for multi-threaded programs on multi-cores provided from PAAS cloud with low cost by SI pricing [16].

2. EAIC

2.1. Enhanced Adaptive Incremental Checking

In the current scenario, the resource allocated to the user is limited and he cannot exceed his limit under any circumstances. At some point of time, the resource gets exhausted in the file that he wants to upload cannot be saved until a new subscription was purchased. Basic subscription provides basic subscription comes at a minimum amount for a minimum fixed size of storage. When the file size exceeds, the client has no option other than purchasing a bigger resource again for a greater price rather than what is needed. Adjusted Markov model is used for EAIC adaptive check pointing where hardware failures and spot instance revocation events are handled. SI revocation and hardware failures are treated with three failure levels and two checkpoint types. In a multi-level model, all the failures are recovered with checkpoint level. There is no allocated SI hardware hence assumed that no hardware failures. SI revocation is based on unavailability of storage or changes in the pricing strategy. Check pointing is highly essential for application based RaaS clouds to minimize the price and the cost involved. The enhanced EAIC reacts to network dynamicity by opportunistic network bandwidth measurement. Better accuracy is achieved with exponential smoothing. Delta compression is skipped. EAIC sends check point files at regular intervals to create its prediction models. It also adapts the changes in the allocation quickly till the last space available in the resource thereby allowing to extend the availability by Checkpointing and to let the current ongoing execution without interruption locking the final limited size of the resource without changing the actual allocated size on the storage. EAIC keeps track of exceeded size and SI will work over, and the price is calculated for the exceeded limit determined by the owner.

3. SYSTEM ARCHITECTURE

The architecture diagram represents the overall concept of the system that shows the relationship between different compositions of the system in which the principal parts or functions are represented by blocks and connected lines shows the relationship of the blocks.

The initial setup for the user registration and its authentication. The client will provide all the specific personal details that will be verified and authenticated by the owner. The administrator checks for the credentials and authenticates it. Next, the user can log in to the system after authentication. After successful login, then the user can allow reserving resource from the cloud, then the user can fill the requirements or resources. After resource reservation, requested user must deposit the initial amount to use the resource from the cloud. After getting the spot instance, the client has to pay the price. EAIC keeps track of exceeded size and SI will work over, and the price is calculated for the exceeded limit determined by the owner. The client has to fill up payment details. After finishing the payment details, a request will be forwarded to cloud owner The client can download the files at any point of the time. If the file is within the limit the file

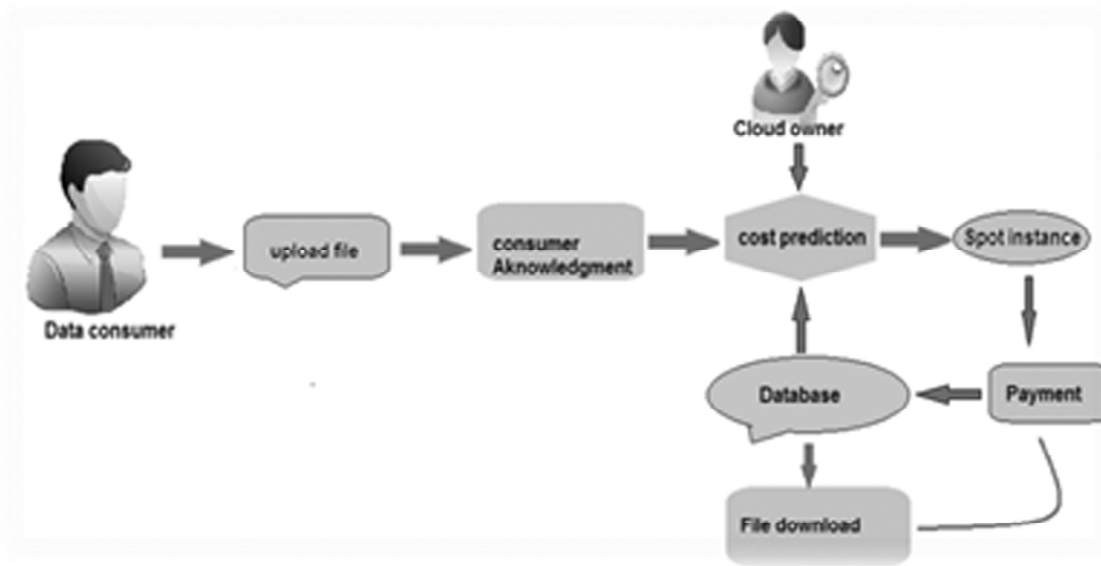


Figure 1: System Architecture

can directly be downloaded else they will be redirected to the payment page for the exceeded amount payment, and then the download will be available.

4. CONCLUSIONS

This paper has presented an adaptive multi-level Checkpointing scheme, EAIC for RaaS clouds under SI pricing which provides much reliable resource allocation and effective cost reduction. Checkpointing minimizes the cost and the turnaround time of the application. EAIC reduces the execution time by processing the client's request faster by saving the content and allowing them to explore the storage later. Thus, EAIC is beneficial concerning cost for the cloud systems.

REFERENCES

- [1] Naga Malleswari T Y J, Rajeswari D, Dr Jawahar Senthil Kumar V.: A Survey of Cloud Computing, Architecture & Services Provided by Various Cloud Service Providers. In: 2nd International Conference on Demand Computing, pp.201. Bangalore (2012).
- [2] M. Armbrust, A. Fox et al.: Above the clouds: A Berkeley View of Cloud Computing, EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS- pp.1 Berkeley (2009).
- [3] Bhaskar Prasad Rimal, Eunmi Choi, Ian Lumb: A Taxonomy and Survey of Cloud Computing Systems. In: Fifth International Joint Conference on INC,IMS and IDC, pp. 44-5. Washington (2009).
- [4] O.A.Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafirir.: The Resource-as-a-Service (RaaS) Cloud.4th USENIX Workshop Hot Topics in Cloud Computing (HotCloud' 12), pp. 12-12 (2012).
- [5] Amazon EC2 Pricing and Instance Purchase Options, <http://aws.amazon.com/ec2/pricing/>
- [6] N.H. Vaidya.:A Case for Two-Level Recovery Schemes. In: IEEE Trans Computers, vol. 47, no.6 pp.656-666, (1998).
- [7] A. Moody, G. Bronevetsky, K. Mohror, and B.R. de Supinski.: Design, Modeling, and Evaluation of a Scalable Multi-Level Checkpointing System. In: Proc. IEEE/ACM Int'l Conf. High-Performance Computing, Networking, Storage and Analysis (SC), pp. 1-11. (2010).
- [8] K. Sato et al.: Design and Modeling of a Non-Blocking Checkpointing System. In: Proc. IEEE/ACM Int'l Conf. HighPerformance Computing, Networking, Storage and Analysis (SC), pp. 1-10, (2012).
- [9] C. P. Tang, T. Y. Wong, and P. P. C. Lee.: CloudVS: Enabling Version Control for Virtual Machines in an Open-Source Cloud under Commodity Settings. In: Proc. IEEE Network Operations and Management Symp. (NOMS), pp. 188-195, (2012).
- [10] B. Nicolae and F. Cappello.: BlobCR: Efficient Checkpoint-Restart for HPC Applications on IaaS Clouds Using Virtual Disk Image Snapshots. In: Proc. IEEE/ACM Int'l Conf. High-Performance Computing, Networking, Storage and Analysis (SC), pp. 1-12, (2011).

- [11] M. Vasavada et al.: Comparing Different Approaches for Incremental Checkpointing: The Showdown. In: Proc. Linux Symp., pp. 69-80, (2011).
- [12] R Gioiosa, J. C. Sancho, S. Jiang, and F. Petrini.: Transparent, Incremental Checkpointing at Kernel Level: A Foundation for Fault Tolerance for Parallel Computers. In: Proc. IEEE/ACM Int'l Conf. High Performance Computing, Networking, Storage, and Analysis (SC'05), pp.9-23, (2005).
- [13] C. Wang, F. Mueller, C. Engelmann, and S. L. Scott.: Hybrid Checkpointing for MPI Jobs in HPC Environments. In: Proc. 16th IEEE Int'l Conf. Parallel and Distributed Systems, pp. 524-533, (2010).
- [14] J. S. Plank, J. Xu, and R.H.B. Netzer.: Compressed Differences: An Algorithm for Fast Incremental Checkpointing. In: Univ. of Tennessee, Technical Report CS-95-302, (1995).
- [15] I. Jangjaimon and N. F. Tzeng.: Adaptive Incremental Checkpointing via Delta Compression for Networked Multicore Systems. In: Proc. IEEE Int'l Parallel & Distributed Processing Symp. (IPDPS), pp. 7-18, (2013).
- [16] S. Yi, D. Kondo, and A. Andrzejak.: Monetary Cost-Aware Checkpointing and Migration on Amazon Cloud Spot Instances. In: IEEE Trans. Service Computing, vol. 5, pp. 512-524, (2012).