International Journal of Advanced Research in Engineering Technology & Science
ISSN: 2349-2819
Impact Factor: 7.10
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Email: editor@ijarets.org Volume-10 Issue-10 October-2023 www.ijarets.org

STUDY ON COMMUNITY CLOUD AND ORACLE CLOUD INFRASTRUCTURE

Sonmali Sandip Shivaji

Research Scholar, NIILM University, Kaithal, Haryana

Dr. Lokendra

Department of Computer Science And Engineering

NIILM University, Kaithal, Haryana

ABSTRACT

The concept of cloud computing is now the most prominent new idea. This model involves a wide range of services, and it is possible to provide these services to customers on a pay-per-use basis. It is possible for it to be entirely private, wholly public, or a hybrid of the two. Because of this, businesses are able to boost their processing capacity while simultaneously lowering the likelihood of experiencing downtime and losing data. In spite of this, there are still a great deal of obstacles to overcome in multi-cloud computing, such as the facilitation of a common API for various resources, a straightforward pricing model for various resources, and hypervisors in a number of providers. Cloud computing environments offer low-cost services or resources, which are determined by the requirements of the users. Because of the ever-increasing requirements of users, it is the duty of the service provider to distribute resources to consumers in an effective manner. Additionally, the efficiency of the cost of resources ought to be optimized in accordance with the demand of the individual.

Keywords : *Community cloud, oracle cloud, infrastructure.*

INTRODUCTION

The concept of cloud computing is now the most prominent new idea. This model involves a wide range of services, and it is possible to provide these services to customers on a pay-per-use basis. Some examples of these services are water and energy delivery to customers' homes. It places an emphasis on the manner in which we link computer systems, develop applications, and provide services that are already in existence. During the process of providing services to customers, it is anticipated that the services will be scalable and flexible. The fundamental idea behind cloud computing is dynamic provisioning, which enables services to be administered to consumers in accordance with the workload that is currently being performed. Prior to the advent of cloud computing, local servers experienced an unexpectedly high volume of traffic. A method known as virtualization is used to replace these local servers, which ultimately leads to the development of cloud computing. Cloud computing has emerged as a rescuer for a great number of enterprises in the modern day.

The opportunities for growth that are available to organizations that are not progressing toward cloud computing are limited.

In Figure 1, we can see that there are four main kinds of clouds that may be used to develop cloud-based services. Cloud computing typically builds infrastructure, which in turn allows for the creation and distribution of a wide range of services to end users. The following is an overarching description of them:

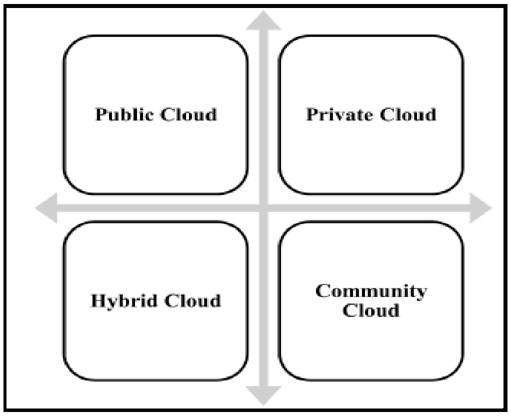

Figure 1: Types of cloud

Community cloud

Located anywhere in the globe and accessible from any internet-connected device, this is the first cloud service to go live. Multi-tenancy is its main trait that sets it apart. Multi-tenancy refers to the capacity to provide services to several users concurrently in a virtual environment. Public clouds that provide software, platforms, and infrastructure as a service include Google Apps and Amazon's Elastic Compute Cloud (EC2).

1. Public cloud

It has many similarities with public clouds, but the resources it offers are limited to certain organisations. This kind of cloud setup guarantees the Service Level Agreement (SLA) and keeps all security settings up to date. The private cloud, on the other hand, lacks the public cloud's elastic on-demand scalability. Concerning the private cloud, this is a major downside.

2. Hybrid cloud

There are still security vulnerabilities in the public cloud, even if it can deliver enough services to customers. However, although private clouds are great for on-premises data storage, they can't adapt to fluctuating demand, which is a major downside. The result is the creation of the hybrid cloud, which aims to blend the best features of public and private clouds while avoiding their drawbacks. One way to address the issue of scalability is by using a hybrid cloud, which offers resources on demand. Nevertheless, these resources have a use window and are relinquished after that. Cloud bursting is the name given to this process.

3. Community cloud

This entails integrating the services provided by many clouds to address the needs of the relevant community or industry. The funding for this kind of cloud infrastructure may usually come from a combination of sources, including private companies, research groups, and government bodies.

OCI, OR ORACLE CLOUD INFRASTRUCTURE

Oracle Cloud Infrastructure (OCI) is a cloud computing platform from Oracle Corporation. It helps organisations build, deploy, and manage cloud applications and workloads with its entire infrastructure services. Services include data processing, storage, networking, and database administration. Due to its scalability, security, and speed, Open Compute Infrastructure (OCI) lets enterprises run many workloads. Small online applications to business software are covered by these workloads. Finally, it provides AI, analytics, and security services to improve app development and deployment.

OCI for trades

OCI offers several benefits and capabilities for businesses:

1. **Infrastructure Services**: Open Cloud Infrastructure (OCI) offers virtual machines, bare metal servers, backup and recovery, virtual networks, load balancers, DNS services, and virtual and object storage. Businesses may construct and maintain their IT infrastructure in the cloud utilizing these services, saving money on hardware.

2. **Scalability and Elasticity**: By using OCI, businesses may modify their resource use in reaction to fluctuations in demand. Integrated autoscaling features enable seamless adjustment of resources to handle spikes or dips in workloads. This guarantees that the system's performance and cost-effectiveness are maximized.

3. **High Performance**: intended to achieve results, OCI strives for task and program excellence. It uses contemporary networking and hardware technologies like RDMA (Remote Direct Memory Access) to boost application performance by providing low-latency, high-bandwidth communication across instances.

4. **Security**: OCI uses strong security to protect commercial applications and data. This package includes identity and access management, data encryption, network security, and security monitoring tools. Mission-critical applications may be secure with OCI since it respects all legislation and industry requirements.

5. **Integrated Database Services**: OCI offers Oracle Autonomous Database, MySQL, and NoSQL databases. With these scalable, high-performance, and fully managed database systems, businesses may expedite data management and reduce administrative labour.

6. **Hybrid Cloud Capabilities**: OCI allows organizations to integrate their on-premises infrastructure to the cloud via hybrid cloud deployments. This allows smooth data transmission, task transfer, and central administration of both environments.
7. **Developer Tools and Services**: Open Container Infrastructure (OCI) offers container services, serverless computing, and DevOps automation tools for developers. These technologies speed up application development and deployment, making companies nimbler and more innovative.
8. **Artificial Intelligence (AI) and Analytics:** OCI provides ML, analytics, and AI. These include data analytics platforms, integration tools, and pre-built AI models. These abilities help organisations understand their data and leverage technology to improve creativity and decision-making. They also enhance decision-making.

Open Cloud Infrastructure lets companies use cloud computing's scalability, flexibility, security, and cost-effectiveness. Thus, firms may improve operational efficiency, digital transformation, and growth.

Bouache, Mourad. (2020), Cloud computing has shortened the amount of time it takes to bring a product to market, and it has also made it possible for businesses to dynamically change their provisioning in order to suit the ever-evolving requirements of customers all over the world. Amazon, which is one of the top three cloud computing companies in the world, has sixteen regions, forty-four availability zones, and one hundred edge sites all over the land. Taking into account the cost, the client experience is the most crucial factor to consider while developing applications.

Liu, Henry. (2011), Oracle performance and scalability material is data-driven. The basic explanation of database principles and theories in Oracle's context helps users rapidly understand how to exploit Oracle's performance and scalability capabilities in business application design and deployment. Based on the author's 10 years of Oracle expertise, this book offers dependable, tried, and proven performance optimisation strategies.

Mulia, WiraD & Acken, Johnm & Fritz, (2013), Several different types of computer system workloads are discussed in this article, along with the resource usage that underlies each of those workloads. To be more specific, the study focuses on the characterization of cloud workloads based on the issues, capabilities, and technologies that surround the categories. This is done from the numerous perspectives of the various stakeholders involved in cloud computing. The dynamic and quantifiable low-level metrics and measures that are utilized to detect and decrease resource contention, as well as identify category changes during run-time, are used to establish the relationship between the categories and the major limiting technologies that are underlying them. A usage case example involving high performance computing (HPC) clusters is presented, along with research questions pertaining to dynamic low-level measurements.

Bindu Madavi, (2020), Cloud computing has grown rapidly in recent years. Many businesses employ SaaS, PaaS, and IaaS. Cloud performance is crucial to enhancing future IT infrastructure service quality. Performance analysis is crucial to help users pick the best option. Cloud computing performance studies must address scalability, fault tolerance, storage, and security. This report surveys the most essential cloud performance factors. We also examine alternative cloud computing performance-boosting strategies.

OBJECTIVES

1. To study on OCI, or oracle cloud infrastructure
2. To study on Community cloud

RESEARCH METHODOLOGY

Performance analysis methodologies may be used in various mapping contexts to evaluate the communicative behaviour of the application. The academic literature has used many metrics to evaluate task mappings and establish a correlation with the network's behaviour. These measures include dilation, hop-bytes, and maximum load or congestion. This document provides a comprehensive investigation of the correlation between various task mappings and metrics.
Analysing the network and communication behaviour under various mappings facilitates the identification of near-optimal mappings and elucidates the basic reasons for performance improvements. This research examines three distinct indicators that influence communication performance.

- Aggregate duration of various MPI procedures
- Alongside the maximum and minimum documented quantities of hops used by the network
- Total number of packets sent across all network connections simultaneously, including both average and maximum values

Three metrics indicate the network's performance and the severity of congestion. Moreover, each of these metrics correlates with the application's messaging performance to varying degrees. Through an iterative procedure, we may evaluate various mappings and assess their performance to approach the best performance attainable on a certain architecture.

Performance optimization: mapping techniques

Given that pF3D's all-to-all and send messages represent a scalability barrier, we should endeavour to optimise these two processes in our mappings. Two separate mappings, ABCDET and TABCDE, were first evaluated. Within the node, ABCDET monitors all-to-all communications in the X direction. Conversely, the Sends mapping is infamously inefficient since it use a single connection for 32 processes on one node to attempt communication with analogous operations on another node. The TABCDE mapping alleviates congestion and minimises duration in both all-to-all and transmit operations by distributing the pF3D XY planes around the torus. To facilitate the aggregation of communicative tasks on the torus, we will now implement the tiling mapping procedure using pF3D. The whole code necessary for this task is seen in Figure 2. The torus dimensions for the designated partition may be automatically acquired by Rubik during execution; we only need to provide the number of MPI tasks given to each node. Subsequently, we tile the torus and the application, and finally, we execute the map-related operation. We experimented with several tile sizes for pF3D with differing node counts, all of which can be managed by the same script.

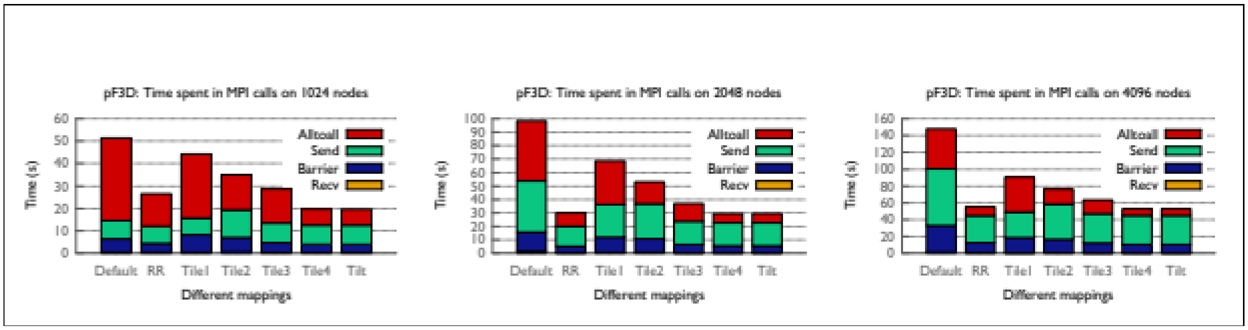

Figure 2: Using alternative task mappings for pF3D operating on 1,024, 2,048 and 4,096 nodes of Blue Gene/Q reduce the amount of time spent in various MPI functions (Note: y-axis has a different range in each figure).

Table 1: Details of communication patterns.

Mapping	Torus tile ($A \times B \times C \times D \times E \times T$)	pF3D tile
Tile1	Use fewest possible dimensions	$8 \times 8 \times 8$
Tile2	$4 \times 4 \times 4 \times 4 \times 2 \times 1$	$8 \times 8 \times 8$
Tile3	Use fewest possible dimensions	$32 \times 16 \times 1$
Tile4	$4 \times 4 \times 4 \times 4 \times 2 \times 1$	$32 \times 16 \times 1$

RESULT

MPI's total duration and the average and maximum number of 512-byte packets sent via network lines. Since point-to-point communication dominates MILC, the average hop curves and total average packet curves are similar.

No correlation exists between maximum hops and wait times. This is expected as MILC transmissions are unconstrained by latency and sometimes just a few kilobytes. The per-hop latency overhead of BG/Q is low. Except for a few outliers, maximum packets and average hops trend similarly to wait and MPI times. Our performance-marker correlation analysis matches these findings.

For 1,024 and 2,048 nodes, the default mapping yields similar average hop count and maximum packet detection rates. This suggests that MPI and wait times are similar. With 4,096 nodes, average hops quadruple, increasing wait time. The all-reduce time has dropped, suggesting better communication equilibrium, while the MPI duration remains unchanged. On 2,048 nodes, TABCDE has higher maximum packets and average hops. This is supported by both measurements. This causes high wait and MPI times for TABCDE's 2,048 nodes. ommunication time and maximum packet count are 50% lower with TABCDE than with default mapping.

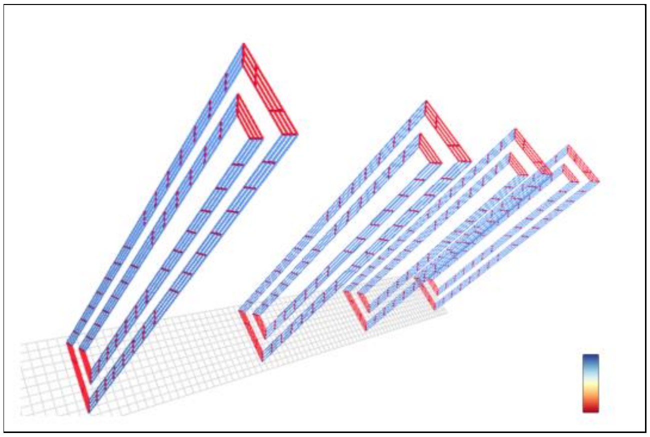

Figure 3: Four sub-tori depicting E's D (blue), C (red, long), and B (red, short, diagonal) linkages. The colours show how many packets transit over each connection.

With 2,048 nodes on average and 4,096 nodes at the most, Node mapping produces bigger packets than the default mapping. In addition, both the wait time and the MPI time are longer than expected. We don't see any such trends at 1,024 nodes. We should follow Jain et al.'s lead and do a more comprehensive study on this subject. Reduced wait and MPI times are the result of a mapping that reduces average hops and maximizes packets per node count. All four of these tile-based mappings are consistent with this finding.

Table 2: Dimensions of the allocated job partitions on BG/Q.

Nodes	A	B	C	D	E
1024	4	4	4	8	2
4096	4	4	8	16	2
4096	4	8	4	16	2

Table 3: Sizes of the input datasets in terms of the number of executions or samples for the different codes.

	2D Halo	3D Halo	Sub A2A	MILC	pF3D	Total
#Nodes	16 KB 4 MB	16 KB 4 MB	16 KB 4 MB			
1024	84 84	84 84	84 84	208	94	806
4096	84 84	84 84	84 84	103	103	710
Total	168168	168168	168168	311	197	1516

Prediction using ensemble methods

Our goal is to predict the application's performance (running time) for different mappings by utilizing supervised learning techniques, which are frequently used in statistics and machine learning. Section 3.4.3 explains these methods in depth. The learning method can infer a model or function from a training set that contains n samples (mappings) and one or more input features (raw and/or calculated, like average bytes) for each sample. Using the training set allows you to do this. Every sample will be subjected to the target, which is also called the desired output value (execution time). Then, we utilize the trained algorithm to forecast the results for a testing set, which includes new mappings for which we want to determine the runtime. Achieving the best possible results requires normalizing the values of the attributes and the target.

Various supervised learning algorithms were tested, encompassing both statistical and machine learning approaches, such as decision trees and support vector machines, and linear regression. These estimate and algorithmic tools are available in the scikit-learn package for Python. Predictions concerning unseen samples are made possible by fitting the estimator, the first stage as previously stated. For the benchmarks covered in this chapter, the ensemble learning method worked best. The goal of using ensemble approaches is to create a stronger model or estimator by combining the predictions of several models, each of which may be less accurate than the others.

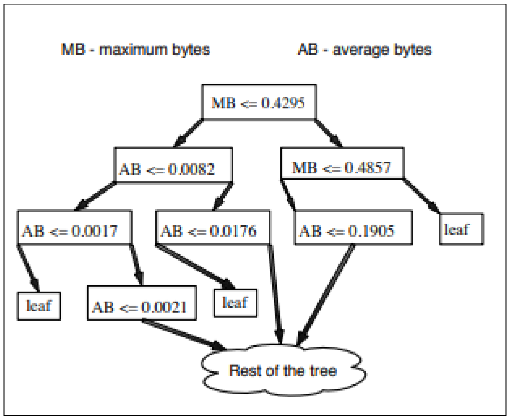

Figure 4 a) Decision tree. Based on the training set and the learning scheme, conditions are computed to guide prediction based on features, e.g., maximum bytes and average bytes. To predict, beginning at

the root, the tree is traversed based on the features of a test case until a leaf is reached. The leaf determines the predicted value.

CONCLUSION

Machine learning systems require continuous updates to learn from fresh data instead of simply the initial training instances. Consider a single-cloud simulation experiment. The system can only predict runtimes on that platform since it doesn't know how to estimate them on others. Improvising estimations while using a different source might fool the system. This might lead to a greater error, like Project B. Enhancing data when there aren't enough training instances and when you want to provide a machine learning system real-world examples is an excellent concept. When tables are full with data, machine learning algorithms may surpass humans in decision-making. Structured data boosts machine performance. Data augmentation and CML's benefits are to blame. Adding additional data and retraining the algorithm with the new entries might fix some of the problems, but the benefits are more exciting. The drawbacks are obvious, but the benefits are more interesting.

REFERENCES

[1.] G. Faanes, A. Bataineh, D. Roweth, T. Court, E. Froese, B. Alverson, T. Johnson, J. Kopnick, M. Higgins, and J. Reinhard, “Cray cascade: A scalable hpc system based on a dragonfly network,” in High Performance Computing, Networking, Storage and Analysis (SC), 2012 International Conference for, Nov 2012.

[2.] B. Arimilli, R. Arimilli, V. Chung, S. Clark, W. Denzel, B. Drerup, T. Hoefler, J. Joyner, J. Lewis, J. Li, N. Ni, and R. Rajamony, “The PERCS High-Performance Interconnect,” in 2010 IEEE 18th Annual Symposium on High Performance Interconnects (HOTI), August 2010, pp. 75–82.

[3.] Michalakes, J., J. Dudhia, D. Gill, T. Henderson, J. Klemp, W. Skamarock, and W. Wang, “The Weather Research and Forecast Model: Software Architecture and Performance,” in Proceedings of the 11th ECMWF Workshop on the Use of High Performance Computing In Meteorology, October 2004.

[4.] C. Bernard, T. Burch, T. A. DeGrand, C. DeTar, S. Gottlieb, U. M. Heller, J. E. Hetrick, K. Orginos, B. Sugar, and D. Toussaint, “Scaling tests of the improved KogutSusskind quark action,” Physical Review D, no. 61, 2000.

[5.] C. H. Still, R. L. Berger, A. B. Langdon, D. E. Hinkel, L. J. Suter, and E. A. Williams, “Filamentation and forward brillouin scatter of entire smoothed and aberrated laser beams,” Physics of Plasmas, vol. 7, no. 5, p. 2023, 2000.

[6.] X. Ni, L. V. Kale, and R. Tamstorf, “Scalable asynchronous contact mechanics using charm++,” in Proceedings of the IEEE International Parallel & Distributed Processing Symposium (to appear), ser. IPDPS ’15. IEEE Computer Society, May 2015, lLNLCONF-663041.

[7.] R. V. Vadali, Y. Shi, S. Kumar, L. V. Kale, M. E. Tuckerman, and G. J. Martyna, "Scalable fine-grained parallelization of plane-wave-based ab initio molecular dynamics for large supercomputers," Journal of Comptational Chemistry, vol. 25, no. 16, pp. 2006–2022, Oct. 2004.

[8.] J. Phillips, G. Zheng, and L. V. Kal´e, "Namd: Biomolecular simulation on thousands of processors," in Workshop: Scaling to New Heights, Pittsburgh, PA, May 2002.

[9.] F. Gygi, E. W. Draeger, M. Schulz, B. R. de Supinski, J. A. Gunnels, V. Austel, J. C. Sexton, F. Franchetti, S. Kral, C. W. Ueberhuber, and J. Lorenz, "Large-scale electronic structure calculations of high-z metals on the bluegene/l platform," in Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, ser. SC '06. New York, NY, USA: ACM, 2006. [Online]. Available: http://doi.acm.org/10.1145/1188455.1188502

[10.] F. Ercal and J. Ramanujam and P. Sadayappan, "Task allocation onto a hypercube by recursive mincutbipartitioning," in Proceedings of the 3rd conference on Hypercube concurrent computers and applications. ACM Press, 1988, pp. 210–221.

[11.] S. Wayne Bollinger and Scott F. Midkiff, "Processor and Link Assignment in Multicomputers Using Simulated Annealing," in ICPP (1), 1988, pp. 1–7.

[12.] Soo-Young Lee and J. K. Aggarwal, "A Mapping Strategy for Parallel Processing," IEEE Trans. Computers, vol. 36, no. 4, pp. 433–442, 1987.

[13.] G. Bhanot, A. Gara, P. Heidelberger, E. Lawless, J. C. Sexton, and R. Walkup, "Optimizing task layout on the Blue Gene/L supercomputer," IBM Journal of Research and Development, vol. 49, no. 2/3, pp. 489–500, 2005.

[14.] H. Yu, I.-H. Chung, and J. Moreira, "Topology mapping for Blue Gene/L supercomputer," in SC '06: Proceedings of the 2006 ACM/IEEE conference on Supercomputing. New York, NY, USA: ACM, 2006, p. 116.

[15.] T. Agarwal, A. Sharma, and L. V. Kal´e, "Topology-aware task mapping for reducing communication contention on large parallel machines," in Proceedings of IEEE International Parallel and Distributed Processing Symposium 2006, April 2006.