Understanding The Performance And Potential Of Cloud Computing For Scientific Applications

Talluri Lavanya Posi , Miss G.Keerthana, Sri.V.Bhaskara Murthy
MCA Student, Assistant Professor, Associate Professor
Dept Of MCA
B.V.Raju College, Bhimavaram

ABSTRACT: Commercial clouds bring a great opportunity to the scientific computing area. Scientific applications usually require significant resources, however not all scientists have access to sufficient high-end computing systems. Cloud computing has gained the attention of scientists as a competitive resource to run HPC applications at a potentially lower cost. But as a different infrastructure, it is unclear whether clouds are capable of running scientific applications with a reasonable performance per money spent. This work provides a comprehensive evaluation of EC2 cloud in different aspects. We first analyze the potentials of the cloud by evaluating the raw performance of different services of AWS such as compute, memory, network and I/O. Based on the findings on the raw performance, we then evaluate the performance of the scientific applications running in the cloud.Finally, we compare the performance of AWS with a private cloud, in order to find the root cause of its limitations while running scientific applications. This paper aims to assess the ability of the cloud to perform well, as well as to evaluate the cost of the cloud in terms of both raw performance and scientific applications performance. Furthermore, we evaluate other services including S3, EBS and Dynamo DB among many AWS services in order to assess the abilities of those to be used by scientific applications and frameworks. We also evaluate a real scientific computing application through the Swift parallel scripting system at scale. Armed with both detailed benchmarks to gauge expected performance and a detailed monetary cost analysis, we expect this paper will be a recipe cookbook for scientists to help them decide where to deploy and run their scientific applications between public clouds, private clouds, or hybrid clouds.

I. INTRODUCTION

HE idea of using clouds for scientific applications has been around for several years, but it has not gained traction primarily due to many issues such as lower net-work bandwidth or poor and unstable performance. Sci-entific applications often rely on access to large legacy data sets and pre-tuned application software libraries. These applications today run in HPC environments with low latency interconnect and rely on parallel file systems. They often require high performance systems that have high I/O and network bandwidth. Using commercial clouds gives scientists opportunity to use the larger re-sources on-demand. However, there is an uncertainty about the capability and performance of clouds to run scientific applications because of their different nature. Clouds have a heterogeneous infrastructure compared with homogenous high-end computing systems (e.g. su-
percomputers). The design goal of the clouds was to pro-vide shared resources to multi-tenants and optimize the cost and efficiency. On the other hand, supercomputers are designed to optimize the performance and minimize latency.

However, clouds have some benefits over supercom-puters. They offer more flexibility in their environment. Scientific applications often have dependencies on unique libraries and platforms. It is difficult to run these applications on supercomputers that have shared resources with pre-determined software stack and platform, while cloud environments also have the ability to set up a customized virtual machine image with specific platform and user libraries. This makes it very easy for legacy applications that require certain specifications to be able to run. Setting up cloud environments is significantly easier compared to supercomputers, as users often only need to set up a vir-tual machine once and deploy it on multiple instances. Furthermore, with virtual machines, users have no issues with custom kernels and root permissions (within the virtual machine), both significant issues in non-virtualized high-end computing systems.

There are some other issues with clouds that make them challenging to be used for scientific computing. The network bandwidth in commercial clouds is significantly lower (and less predictable) than what is available in supercomputers. Network bandwidth and latency are two of the major issues that cloud environments have for high-performance computing. Most of the cloud re-sources use commodity network with significantly lower bandwidth than supercomputers [13].

The virtualization overhead is also another issue that leads to variable compute and memory performance. I/O is yet another factor that has been one of the main issues on application performance. Over the last decade the compute performance of cutting edge systems has improved in much faster speed than their storage and I/O performance. I/O on parallel computers has always been slow compared with computation and communication. This remains to be an issue for the cloud environment as well. Finally, the performance of parallel systems including networked storage systems such as Amazon S3 needs to be evaluated in order to verify if they are capable of run-ning scientific applications [3]. All of the above men-tioned issues raise uncertainty for the ability of clouds to effectively support HPC applications. Thus it is important to study the capability and performance of clouds in sup-port of scientific applications. Although there have been early endeavors in this aspect [10] [14] [16] [20] [23], we de-velop a more comprehensive set of evaluation. In some of these works, the experiments were mostly run on limited types and number of instances [14] [16] [17]. Only a few of the researches have used the new Amazon EC2 cluster instances that we have tested [10] [20] [24]. However the performance metrics in those papers are very limited. This paper covers a thorough evaluation covering major performance metrics and compares a much larger set of EC2 instance types and the commonly used Amazon Cloud Services. Most of the aforementioned above men-tioned works lack the cost evaluation and analysis of the cloud. Our work analyses the cost of the cloud on differ-ent instance types.

The main goal of this research is to evaluate the per-formance of the Amazon public cloud as the most popular commercial cloud available, as well as to offer some con-text for comparison against a private cloud solution. We run micro benchmarks and real applications on Amazon AWS to evaluate its performance on critical metrics in-cluding throughput, bandwidth and latency of processor, network, memory and storage [2]. Then, we evaluate the performance of HPC applications on EC2 and compare it with a private cloud solution [27]. This way we will be able to better identify the advantages and limitations of AWS on the scientific computing area. Over the past few years, some of the scientific frame-works and applications have approached using cloud services as their building blocks to alleviate their compu-tation

processes [12] [31]. We evaluate the performance of some of the AWS services such as S3 and DynamoDB to investigate their abilities on scientific computing area.

Finally, this work performs a detailed price/cost analy-sis of cloud instances to better understand the upper and lower bounds of cloud costs. Armed with both detailed benchmarks to gauge expected performance and a de-tailed monetary cost analysis, we expect this paper will be a recipe cookbook for scientists to help them decide where to deploy and run their scientific applications be-tween public clouds, private clouds, or hybrid clouds.

This paper is organized as follows: Section 2 provides the evaluation of the EC2, S3 and DynamoDB perfor-mance on different service alternatives of Amazon AWS. We provide an evaluation methodology. Then we present the benchmarking tools and the environment settings of the testbed in this project. Section 2.4 presents the bench-marking results and analyzes the performance. On 2.5 we compare the performance of EC2 with FermiCloud on HPL application. Section 3 analyzes the cost of the EC2 cloud based on its performance on different aspects. In section 4, we review the related work in this area. Section 5 draws conclusion and discusses future work.

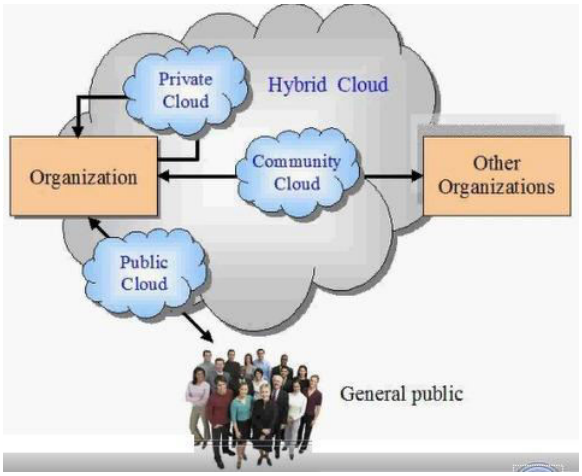
II. EXISTING SYSTEM

The cloud computing paradigm is successfully converging as the fifth utility , but this positive trend is partially limited by concerns about information confidentiality and unclear costs over a medium-long term .We are interested in the Database as a Service paradigm (DBaaS) that poses several research challenges in terms of security and cost evaluation from a tenant's point of view. Most results concerning encryption for cloud-based services are in applicable to the database paradigm. Other encryption schemes, which allow the execution of SQL operations over
encrypted data, either suffer from performance limits or they require the choice of which encryption scheme must be adopted for each database column and
SQL operations .

III. PROPOSED SYSTEM

The proposed architecture guarantees in an adaptive way the best level of data
confidentiality for any database workload, even when the set of SQL queries dynamically changes. The adaptive encryption scheme, which was initially proposed for applications not referring to the cloud, encrypts each plain column into multiple encrypted columns, and each value is encapsulated into different layers of encryption, so that the outer layers guarantee higher confidentiality but support fewer computation capabilities with respect to the inner layers. we propose the first analytical cost estimation model for evaluating cloud database costs in plain and encrypted instances from a tenant's point of view in a medium-term period. It takes also into account the variability of cloud prices and the possibility that the database workload may change during the evaluation period. This model is instanced with respect to several cloud provider offers and related real prices. As expected, adaptive encryption influences the costs related to storage size and network usage of a database service. However, it is important that a tenant can anticipate the final costs in its period of interest, and can choose the best compromise between data confidentiality and expenses.

IV. SYSTEM ARCHITECTURE



V. IMPLEMENTATION MODULES:

1. Adaptive encryption
2. Metadata structure
3. Encrypted database management
4. Cost Estimation of cloud database services
5. Cost model
6. Cloud pricing models
7. Usage Estimation

Adaptive encryption:

The proposed system supports adaptive encryption methods for public cloud database service, where distributed and concurrent clients can issue direct SQL
operations. By avoiding an architecture based on one [or] multiple intermediate servers between the clients and the cloud database, the proposed solution guarantees the same level of scalability and availability of the cloud service. Figure 1 shows a scheme of the proposed architecture where each client executes an encryption engine that manages encryption operations. This software module is accessed by external user applications through the encrypted database interface. The proposed architecture manages five types of information.

• plain data is the tenant information;
• encrypted data is stored in the cloud database;
• plain metadata represent the additional information that is necessary to execute SQL operations on encrypted data;
• encrypted metadata is the encrypted version of the metadata that are stored in the cloud database;
• master key is the encryption key of the encrypted metadata that is distributed to legitimate clients.

Metadata structure:

Metadata include all information that allows a legitimate client knowing the master key to execute SQL operations over an encrypted database. They are organized and
stored at a table-level granularity to reduce communication overhead for retrieval, and to improve management of concurrent SQL operations. We define all metadata information associated to a table as table metadata. Let us describe the structure of a table metadata .Table metadata includes the correspondence between the plain table name and the encrypted table name because each encrypted table name is randomly generated. Moreover, for each column of the original plain table it also includes a column metadata parameter containing the name and the data type of the corresponding plain column (e.g., integer, string, timestamp). Each column metadata is associated to one or more onion metadata, as many as the number of onions related to the column.

Encrypted database management:

The database administrator generates a master key, and uses it to initialize the architecture metadata. The master key is then distributed to legitimate clients. Each table creation requires the insertion of a new row in the metadata table. For each table creation, the administrator adds a column by specifying the column name, data

type and confidentiality parameters. These last are the most important for this paper because they include the set of onions to be associated with the column, the starting layer (denoting the actual layer at creation time) and the field confidentiality of each onion. If the administrator does not specify the confidentiality parameters of a column, then they are automatically chosen by the client with respect to a tenant's policy. Typically, the default policy assumes that the starting layer of each onion is set to its strongest encryption algorithm.

Cost Estimation of cloud database services:

A tenant that is interested in estimating the cost of porting its database to a cloud platform. This porting is a strategic decision that must evaluate confidentiality issues and the related costs over a medium-long term. For these reasons, we propose a model that includes the overhead of encryption schemes and variability of database workload and cloud prices. The proposed model is general enough to be applied to the most popular cloud database services, such as Amazon Relational Database Service.

Cost model:

The cost of a cloud database service can be estimated as a function of three main parameters:

Cost = f(T ime, Pricing,Usage) where:

• Time: identifies the time interval T for which the tenant requires the service.

• Pricing: refers to the prices of the cloud provider for subscription and resource usage; they typically tend to diminish during T .

• Usage: denotes the total amount of resources used by the tenant; it typically increases during T .In order to detail the pricing attribute, it is important
to specify that cloud providers adopt two subscription policies: the on-demand policy allows a tenant to payper-use and to withdraw its subscription anytime; the reservation policy requires the tenant to commit in advance for a reservation period. Hence, we distinguish between billing costs depending on resource usage and reservation costs denoting additional fees for commitment in exchange for lower pay-per-use prices. Billing costs are billed periodically to the tenant every billing period.

Cloud pricing models:

Popular cloud database providers adopt two different billing functions, that we call linear L and tiered T . Let us consider a generic resource x, we define as xb its usage at the b-th billing period and px b its price. If the billing function is tiered, the cloud provider uses different prices for different ranges of resource usage. Let
us define Z as the number of tiers, and [ˆx1, . . . , ˆxZ−1] as the set of thresholds that define all the tiers. The uptime and the storage billing functions of Amazon RDS are linear, while the network usage is a tiered billing function. On the other hand, the uptime billing functions of Azure SQL is linear, while the
storage and network billing functions are tiered.

Usage Estimation:

The uptime is easily measurable, it is more difficult to estimate accurately the usage of storage and network , since they depend on the database structure, the workload and the use of encryption. We now propose a methodology for the estimation of storage and network usage due to encryption. For clarity, we define sp, se, sa as the storage usage in the plaintext, encrypted, and adaptively encrypted databases for one billing period. Similarly, np, ne, na represent network usage of the three configurations. We assume that the tenant knows the database structure and the query workload and we assume that each column a A stores ra values. By denoting as vp a the average storage size of each plaintext value

stored in column a, we estimate the storage of the plaintext database.

VI. CONCLUSION

In this paper, we present a comprehensive, quantitative study to evaluate the performance of the Amazon EC2 for the goal of running scientific applications. We first evalu-ate the performance of various instance types by running micro benchmarks on memory, compute, network and storage. In most of the cases, the actual performance of the instances is lower than the expected performance that is claimed by Amazon. The network bandwidth is rela-tively stable. The network latency is higher and less stable than what is available on the supercomputers. Next, based on the performance of instances on micro-benchmarks, we run scientific applications on certain in-stances. We finally compare the performance of EC2 as a commonly used public cloud with FermiCloud, which is a higher-end private cloud that is tailored for scientific for scientific computing.

We compare the raw performance as well as the per-formance of the real applications on virtual clusters with multiple HPC instances. The performance and efficiency of the two infrastructures is quite similar. Their only dif-ference that affects their efficiency on scientific applica-tions is the network bandwidth and latency which is higher on FermiCloud. FermiCloud achieves higher per-formance and efficiency due to having InfiniBand net-work cards. We can conclude that there is need for cloud infrastructures with more powerful network capacity that are more suitable to run scientific applications.

We evaluated the I/O performance of Amazon instanc-es and storage services like EBS and S3. The I/O perfor-mance of the instances is lower than performance of dedi-cated resources. The only instance type that shows prom-ising results is the high-IO instances that have SSD drives on them. The performance of different parallel file sys-tems is lower than performance of them on dedicated clusters. The read and write throughput of S3 is lower than a local storage. Therefore it could not be a suitable option for scientific applications. However it shows promising scalability that makes it a better option on larger scale computations. The performance of PVFS2 over EC2 is convincible for using in scientific applications that require a parallel file system.

Amazon EC2 provides powerful instances that are ca-pable of running HPC applications. However, the per-formance a major portion of the HPC applications are heavily dependent on network bandwidth, and the net-work performance of Amazon EC2 instances cannot keep up with their compute performance while running HPC applications and become a major bottleneck. Moreover, having the TCP network protocol as the main network protocol, all of the MPI calls on HPC applications are made on top of TCP protocol. That would add a signifi-cant overhead to the network performance. Although the new HPC instances have higher network bandwidth, they are still not on par with the non-virtualized HPC systems with high-end network topologies. The cloud instances have shown to be performing very well, while running embarrassingly parallel programs that have minimal in-teraction between the nodes [10]. The performance of em-barrassingly parallel application with minimal communi-cation on Amazon EC2 instances is reported to be compa-rable with non-virtualized environments [21] [22]. Armed with both detailed benchmarks to gauge ex-pected performance and a detailed price/cost analysis, we expect that this paper will be a recipe cookbook for scien-tists to help them decide between dedicated resources, cloud resources, or some combination, for their particular scientific computing workload.

REFERENCE

[1] Amazon EC2 Instance Types, Amazon Web Services,[online]2013,http://aws.amazon.com/ec2/instance-types/ (Accessed: 2 November 2013)

[2] Amazon Elastic Compute Cloud (Amazon EC2), Amazon Web Services, [online] 2013, http://aws.amazon.com/ec2/ (Accessed: 2 November 2013)

[3] Amazon Simple Storage Service (Amazon S3), Amazon Web Services, [online] 2013, http://aws.amazon.com/s3/ (Accessed: 2 November 2013)

[4] Iperf, Souceforge, [online] June 2011, http://sourceforge.net/projects/iperf/ (Accessed: 2 November 2013)

[5] A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary. "HPL", (netlib.org), [online] September 2008, http://www.netlib.org/benchmark/hpl/ (Accessed: 2 November 2013)

[6] J. J. Dongarra, S. W. Otto, M. Snir, and D. Walker, \An intro-duction to the MPI standard," Tech. Rep. CS-95-274, University of Tennessee, Jan. 1995

[7] Release: Amazon EC2 on 2007-07-12, Amazon Web Services, [online] 2013, http://aws.amazon.com/releasenotes/Amazon-EC2/3964 (Accessed: 1 November 2013)

[8] K. Yelick, S. Coghlan, B. Draney, and R. S. Canon, "The Magel-lan report on cloud computing for science," U.S. Department of Energy, Tech. Rep., 2011

[9] L. Ramakrishnan, R. S. Canon, K. Muriki, I. Sakrejda, and N. J. Wright. "Evaluating Interconnect and virtualization perfor-mance for high performance computing", ACM Performance Evaluation Review, 2012

[10] P. Mehrotra, et al. 2012. "Performance evaluation of Amazon EC2 for NASA HPC applications" In Proceedings of the 3rd work-shop on Scientific Cloud Computing (ScienceCloud '12). ACM, New York, NY, USA, pp. 41-50

[11] A. J. Younge, R. Henschel, J. T. Brown, G. von Laszewski, J. Qiu, and G. C. Fox, "Analysis of virtualization technologies for high performance computing environments," International Conference on Cloud Computing, 2011

12] Y. Zhao, M. Hategan, B. Clifford, I. Foster, G. von Laszewski, I. Raicu, T. Stef-Praun, and M. Wilde. "Swift: Fast, reliable, loose-ly coupled parallel computation", IEEE Int. Workshop on Scien-tific Workflows, pages 199–206, 2007

[13] I. Raicu, Z. Zhang, M. Wilde, I. Foster, P. Beckman, K. Iskra, B. Clifford. "Towards Loosely-Coupled Programming on Pet-ascale Systems", IEEE/ACM Supercomputing 2008

[14] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema. "A Performance Analysis of EC2 Cloud Comput-ing Services for Scientific Computing". In Cloudcomp, 2009

[15] Q. He, S. Zhou, B. Kobler, D. Duffy, and T. McGlynn. "Case study for running HPC applications in public clouds," In Proc. of ACM Symposium on High Performance Distributed Computing, 2010

[16] G. Wang and T. S. Eugene Ng. "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center". In IEEE INFOCOM, 2010

[17] S. L. Garfinkel, "An evaluation of amazon's grid computing services: Ec2, s3 and sqs," Computer Science Group, Harvard University, Technical Report, 2007, tR-08-07

[18] K. R. Jackson et al. "Performance and cost analysis of the su-pernova factory on the amazon aws cloud". Scientific Pro-gramming, 19(2-3):107-119, 2011

[19] J.-S. Vockler, G. Juve, E. Deelman, M. Rynge, and G.B. Berri-man, "Experiences Using Cloud Computing for A Scientific Workflow Application," 2nd Workshop on Scientific Cloud Computing (ScienceCloud), 2011

[20] L. Ramakrishnan, P. T. Zbiegel, S. Campbell, R. Bradshaw, R. S. Canon, S. Coghlan, I. Sakrejda, N. Desai, T. Declerck, and A. Liu. “Magellan: experiences from a science cloud”. In Proceed-ings of the 2nd international workshop on Scientific cloud computing, pages 49–58, San Jose, USA, 2011