Cloud Computing: Literature Review

Rakibul Hassan
Electrical and Computer Engineering Department
George Mason University
Fairfax, Virginia
rhassa2@gmu.edu

***Abstract*—Cloud computing has recently emerged as a new paradigm for hosting and delivering services over the Internet. Cloud computing is attractive to business owners as it eliminates the requirement for users to plan ahead for provisioning, and allows enterprises to start from the small and increase resources only when there is a rise in service demand. However, despite the fact that cloud computing offers huge opportunities to the IT industry, the development of cloud computing technology is currently at its infancy, with many issues still to be addressed. In this paper, we present a survey of cloud computing, highlighting its key concepts, architectural principles, state-of-the-art implementation as well as research challenges. The aim of this paper is to provide a better understanding of the design challenges of cloud computing and identify important research directions in this increasingly important area.**

***Index Terms*—cloud, performance, cloud computing, architecture, scale-up, big-data**

I. Introduction

Cloud computing provides a large variety of architectural configurations, such as the number of cores, amount of memory, and the number of nodes. The performance of a workload an application and its input can execute up to 20 times longer or cost 10 times more than optimal. The ready flexibility in cloud offerings has created a paradigm shift. Whereas before an application was tuned for a given cluster, in the cloud the architectural configuration is tuned for the workload. Furthermore, because the cloud has a pay-as-you-go model, each configuration (cluster size VM type) has running cost and execution time. Therefore, a workload can be optimized for least cost or shortest time which are different configurations.

Choosing the right cloud configuration for an application is essential to service quality and commercial competitiveness. For instance, a bad cloud configuration can result in up to 12 times more cost for the same performance target. The saving from a proper cloud configuration is even more significant for recurring jobs [5], [9] in which similar workloads are executed repeatedly. Nonetheless, selecting the best cloud configuration, e.g., the cheapest or the fastest, is difficult due to the complexity of simultaneously achieving high accuracy, low overhead, and adaptivity for different applications and workloads.

II. What is a Cloud

A cloud can be defined as the software and services that run on the Internet, instead of locally on a local host system. These software and services can be accessed remotely. Example of cloud services include Netflix, Google Drive, and Microsoft Onedrive.

Also Amazon Web Service, Microsoft Azure, IBM cloud service, Google Cloud service are among the top cloud service provider. Most Common questions that's need to answered in Cloud Computing are:

- How to evaluate and choose the right cloud solution?
- How to design applications which is optimized for the cloud?
- How to integrate public cloud applications with in-premise and private cloud applications?
- How to integrate different cloud solutions?
- How to setup new infrastructures so that applications running on them can easily interoperate and move to public cloud, if required?

In this work a brief survey is presented to discuss the present research work addressing all these questions and their probable solutions.

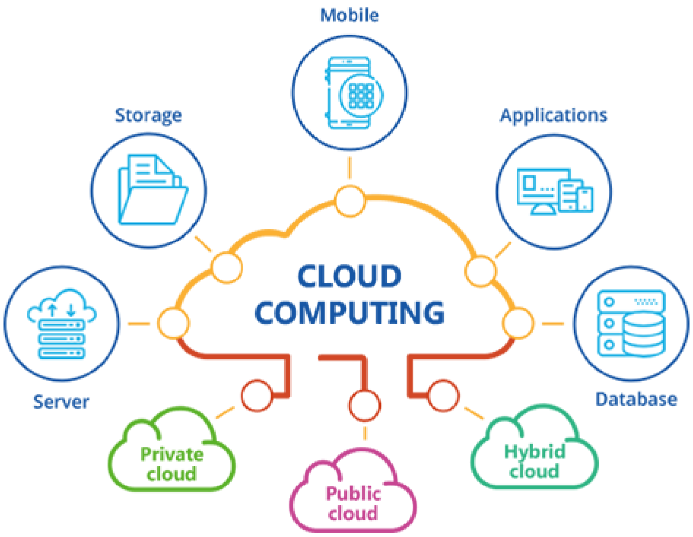


Fig. 1. Cloud Computing.

III. What is Cloud Computing

A. The NIST Definition of Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage,

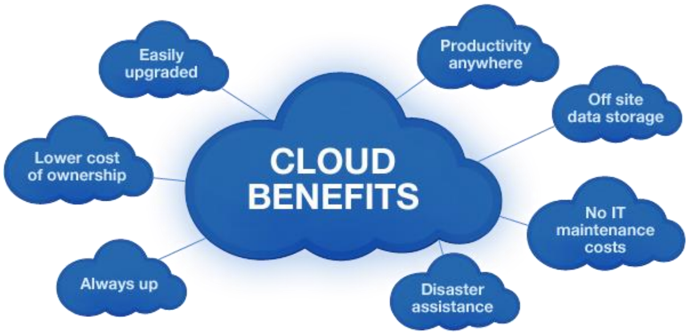


Fig. 2. Cloud-Computing-Benefits

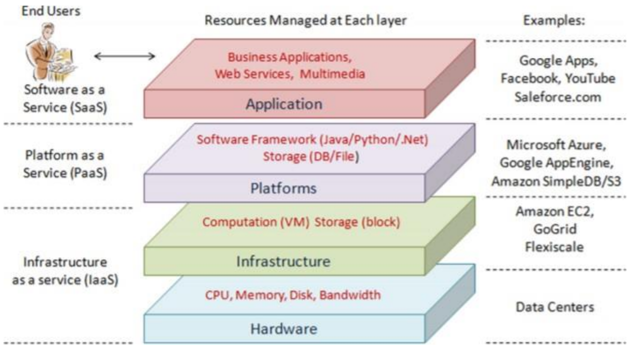

End Users
Resources Managed at Each layer
Examples:
Software as a Service (SaaS)
Business Applications, Web Services, Multimedia
Application
Google Apps, Facebook, YouTube Saleforce.com
Platform as a Service (PaaS)
Software Framework (Java/Python/.Net) Storage (DB/File)
Platforms
Microsoft Azure, Google AppEngine, Amazon SimpleDB/S3
Computation (VM) Storage (block)
Infrastructure
Amazon EC2, GoGrid Flexiscale
Infrastructure as a service (IaaS)
CPU, Memory, Disk, Bandwidth
Hardware
Data Centers

Fig. 3. Cloud-Computing-Architecture

applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [13]. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

NISTs definition identified self-service, accessibility from desktops, laptops, and mobile phones, resources that are pooled among multiple users and applications, elastic resources that can be rapidly reapportioned as needed, and measured service as the five essential characteristics of cloud computing. When these characteristics are combined, they create cloud computing infrastructure that contains both a physical layer and an abstraction layer. The physical layer consists of hardware resources that support the cloud services (i.e. servers, storage and network components). The abstraction layer consists of the software deployed across the physical layer, thereby expressing the essential characteristics of the cloud per NISTs definition.

B. *Characteristics of Cloud Computing:*

- On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- Resource pooling. The providers computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.
- Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
- Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

According to Amazon, clouds enable 7 transformation of how applications are designed, built and used.

- Cloud makes distributed architectures easy
- Cloud enables users to embrace the security advantages of shared systems
- Cloud enables enterprises to move from scaling by architecture to scaling by command
- Cloud puts a supercomputer into the hands of every developer
- Cloud enables users to experiment often and fail quickly
- Cloud enables big data without big servers
- Cloud enables a mobile ecosystem for a mobile-first world

C. *Service Models:*

- Software as a Service (SaaS). The capability provided to the consumer is to use the providers applications running on a cloud infrastructure2 . The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited userspecific application configuration settings.
- Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastruc-

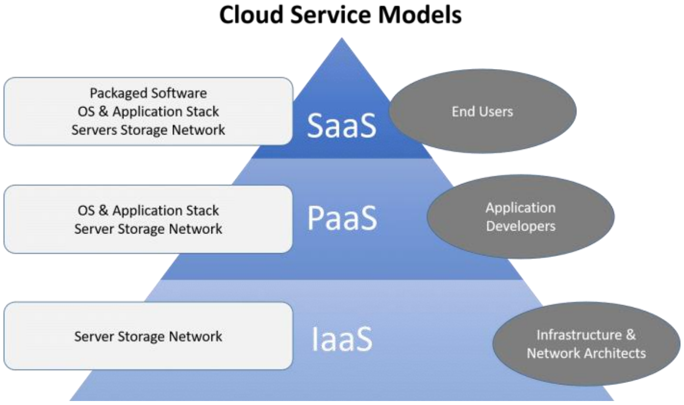


Fig. 4. Different Cloud Computing Models.

ture consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.3 The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

- Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

Fig. 4 shows the different cloud service models.

D. Deployment Models:

- Private cloud. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.
- Community cloud. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.
- Public cloud. The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.
- Hybrid cloud. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

E. Challenges in Cloud Computing

There are several challenges for picking the best cloud configurations for big data analytics jobs.

Complex performance model: The running time is affected by the amount of resources in the cloud configuration in a non-linear way. For instance a regression job on SparkML (with fixed number of CPU cores) sees a diminishing return of running time at 256GB RAM. This is because the job does not benefit from more RAM beyond what it needs. Therefore, the running time only sees marginal improvements. In addition, performance under a cloud configuration is not deterministic. In cloud environments, which is shared among many tenants, stragglers can happen. [6] measured the running time of TeraSort-30GB on 22 different cloud configurations on AWS EC2 five times. In [6] they then computed the coefficient of variation (CV) of the five runs. Their results show that the median of the CV is about 10% and the 90 percentile is above 20%. This variation is not new [9].

Cost model: The cloud charges users based on the amount of time the VMs are up. Using configurations with a lot of resources could minimize the running time, but it may cost a lot more money. Thus, to minimize cost, we have to find the right balance between resource prices and the running time.

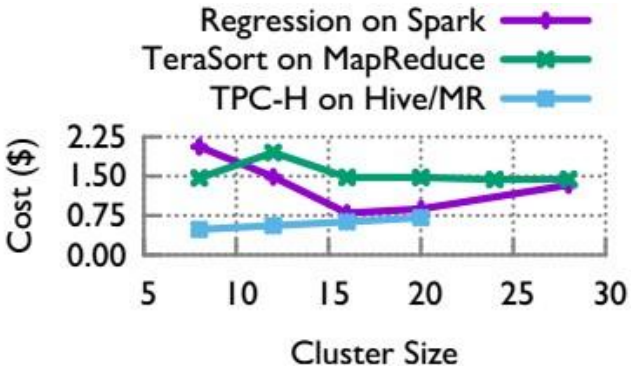


Fig. 5. Regression and TeraSort cost with varying cluster size (M4)

Fig. 5 shows the cost of running Regression on SparkML on different cluster sizes where each VM comes with 15 GBs of RAM and 4 cores in AWS EC2. We can see that the cost does not monotonically increase or decrease when we add more resources into the cluster. This is because adding resources may accelerate the computation but also raises the price per unit of running time.

The heterogeneity of applications: Diverse set of big-data applications and their resource requirement is another challenge for the cloud computing platform to maximize the performace by providing generalized resources.

IV. Literature Survey on Cloud Computing Research

Cloud Computing is a hot research are since couple of years. Good number of papers have been published in this domain. Some state-of-the-art work in cloud computing are described below.

Performance Prediction: There have been a number of recent efforts at modeling job performance in datacenters to support SLOs or deadlines. Techniques proposed in Jockey [9] and ARIA [15] use historical traces and dynamically adjust resource allocations in order to meet deadlines. In Ernest we build a model with no historic information and try to minimize the amount of training data required. Bazaar [12] proposed techniques to model the network utilization of MapReduce jobs by using small subsets of data. In Ernest we capture computation and communication characteristics and use high level features that are framework independent. Projects like MRTuner [16] model MapReduce jobs at very fine granularity and set optimal values for options like memory buffer sizes etc. In Ernest we use few simple features and focus on collecting training data will help us maximize their utility. Finally scheduling frameworks like Quasar [7] try to estimate the scale out and scale up factor for jobs using the progress rate of the first few tasks. Ernest on the other hand runs the entire job on small datasets and is able to capture how different stages of a job interact in a long pipeline.

Query Optimization: Database query progress predictors [14] solve a performance prediction problem similar to Ernest. Database systems typically use summary statistics of the data like cardinality counts to guide this process. Further, these techniques are typically applied to a known set of relational operators. Similar ideas have also been applied to linear algebra operators [11]. In Ernest we use advanced analytics jobs where we know little about the data or the computation being run. Recent work has also looked at providing SLAs for OLTP and OLAP workloads in the cloud and some of the observations in [6] about variation across instance types in EC2 are also known to affect database queries.

Tuning and Benchmarking: Ideas related to experiment design, where we explore a space of possible inputs and choose the best inputs, have been used in other applications like server benchmarking [17]. Related techniques like Latin Hypercube Sampling have been used to efficiently explore file system design space [8]. Autotuning BLAS libraries like ATLAS [19] also solve a similar problem of exploring a state space efficiently

V. Overview of State-of-the-Art Cloud Computing Literature

In this section a brief discussion is presented on recently published work on cloud computing.

A. Clearing the Clouds

In this work [8], they observe that scale-out workloads share many inherent characteristics that place them into a distinct workload class from desktop, parallel, and traditional server workloads. They perform a detailed micro-architectural study of a range of scale-out workloads, finding a large mismatch between the demands of the scale-out workloads and todays predominant processor microarchitecture. They observe significant over-provisioning of the memory hierarchy and core micro-architectural resources for the scale-out workloads. Moreover, continuing the current processor trends will result in further widening the mismatch between the scale-out workloads and server processors. Conversely, they find that the characteristics of scale-out workloads can be leveraged to gain area and energy efficiency in future servers.

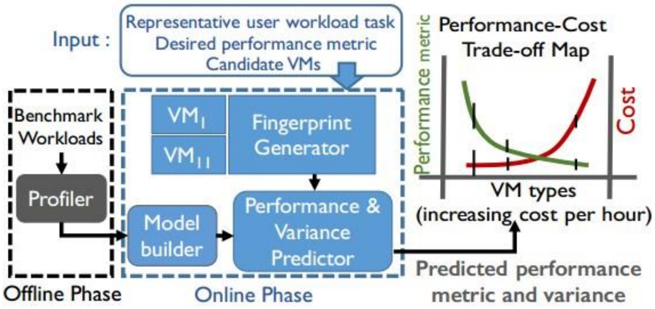

Input :
Representative user workload task
Desired performance metric
Candidate VMs
Benchmark Workloads
Profiler
Offline Phase
VM_I
VM_{II}
Fingerprint Generator
Model builder
Performance & Variance Predictor
Online Phase
Performance metric
Performance-Cost Trade-off Map
Cost
VM types
(increasing cost per hour)
Predicted performance metric and variance

Fig. 6. Architecture of PARIS

The key contributions of this paper are:

- an experimental characterization of performance trade-off of various VM types for realistic workloads across Amazon AWS and Microsoft Azure.
- a novel hybrid offline and online data collection and modeling framework which eliminates the O(n2) data collection overhead while providing accurate performance predictions across cloud providers.
- a detailed experimental evaluation demonstrating that PARIS accurately estimates multiple performance metrics and their variabilities (P90 values), for several real-world workloads across two major public cloud providers, thereby reducing user cost by up to 45 percent relative to strong baseline techniques.

B. Scout: An Experienced Guide to Find the Best Cloud Configuration

Selecting the best cloud configuration from the service provider is a challenge. Several methods have been proposed to find the best cloud configuration [15], [6], [17], [18],

[20], [10]. These methods can be broadly classified into (1) predictionwhich uses elaborate offline evaluation to generate a machine learning model that predicts the performance of workloads and (2) search-based techniqueswhich successively evaluate configurations looking for one that is near optimal [8, 16]. Prediction, as proposed in PARIS [30], is not reliable because of high variance in prediction results. A search-based method does not require an accurate model but can have a high evaluation cost (measured in terms of configurations evaluated). They choose the search-based method because it better tolerates prediction error and delivers effective solutions. Any search-based method has two aspects.

- Exploration: Gather more information about the search space by executing a new cloud configuration.
- Exploitation: Choose the most promising configuration based on information collected.

Additional exploration incurs higher search cost, and insufficient exploration may lead to sub-optimal solutions. This is the exploration-exploitation dilemma appeared in many machine learning problems. For example, CherryPick requires a good exploration strategy to characterize the search space [6]. In this paper, They argue that it is possible to trade exploration with exploitation without settling for a suboptimal configuration. The central insight of this paper is that the cost of the search for the right cloud configuration can be significantly reduced if They could learn from the historical dataexperiences of finding the right cloud configuration for other workloads. In this paper, They present a SCOUT, which uses historical data to find the best cloud configuration for a workload. In doing so, They (1) enable practitioners to find a near-optimal cloud configuration (2) with a lower search cost than state of the art. Additionally, They answer the following questions about improving the performance of the search-based method and reducing the search-cost. Their key contributions are:

- They propose a novel method, SCOUT, that finds (near) optimal solutions and solves the shortcomings of the prior work.
- They present a novel way to represent the search space, which can be used to transfer knowledge from historical measurements
- They evaluate SCOUT and other state-of-the-art methods using more than 100 workloads on three different data processing systems. and
- They make their performance data available for encouraging research of system performance.

C. CherryPick: Adaptively Unearthing the Best Cloud Configurations for Big Data Analytics

CherryPick is a system that leverages Bayesian Optimization to build performance models for various applications, and the models are just accurate enough to distinguish the best or close-to-the-best configuration from the rest with only a few test runs. Our experiments on five analytic applications in AWS EC2 show that CherryPick has a 45-90% chance to find optimal configurations, otherwise near-optimal, saving up to 75% search cost compared to existing solutions.

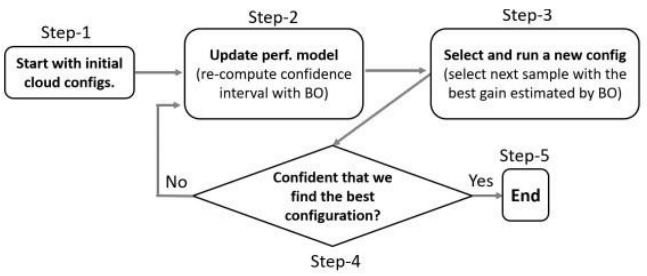

Step-1
Start with initial cloud configs.
Step-2
Update perf. model
(re-compute confidence interval with BO)
Step-3
Select and run a new config
(select next sample with the best gain estimated by BO)
No
Confident that we find the best configuration?
Step-4
Yes
Step-5
End

Fig. 7. CherryPick workflow

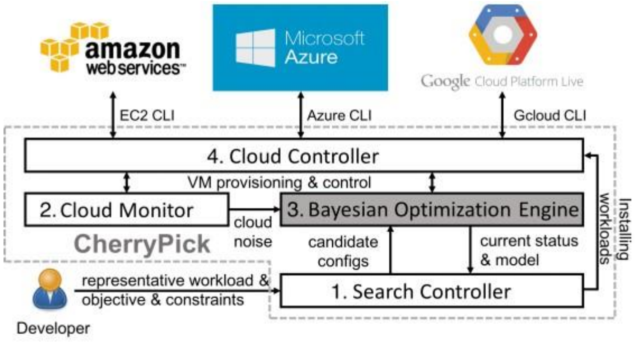


Fig. 8. Architecture of CherryPicks implementation.

The implementation details of CherryPick as shown in Fig. 8. It has four modules.

VI. Popular Cloud Computing Platform

In this section,a survey of some of the dominant cloud computing products will be discussed.

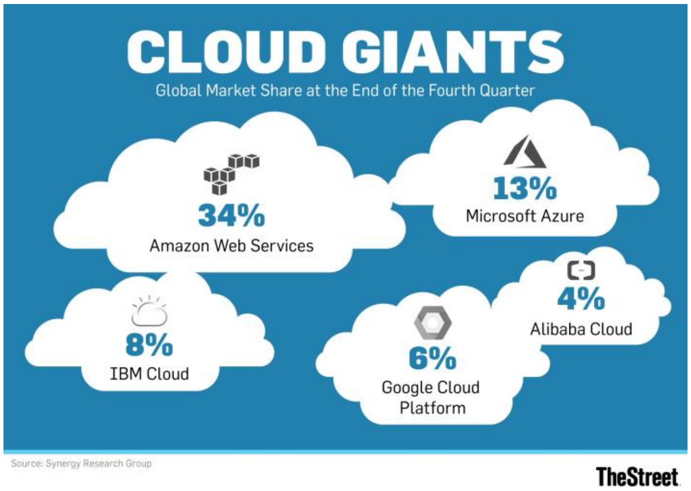


Fig. 9. Cloud Platform Market Share

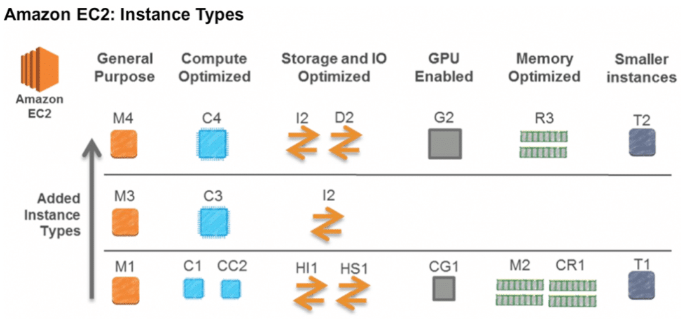
A. Amazon Web Services (AWS)

Amazon Web Services (AWS) [1] is a set of cloud services, providing cloud-based computation, storage and other functionality that enable organizations and individuals to deploy applications and services on an on-demand basis and at commodity prices. Amazon Web Services offerings are accessible over HTTP, using REST and SOAP protocols. Amazon Elastic Compute Cloud (Amazon EC2) enables cloud users to launch and manage server instances in data centers using APIs or available tools and utilities.

amazon webservices
Amazon EC2
Amazon RDS
AWS Direct Connect
Amazon EBS
Amazon S3
Elastic Load Balancing
Amazon Route 53
Amazon VPC
Elastic IP

Fig. 10. Amazon Web Services

EC2 instances are virtual machines running on top of the Xen virtualization engine [4]. After creating and starting an instance, users can upload software and make changes to it. When changes are finished, they can be bundled as a new machine image. An identical copy can then be launched at any time. Users have nearly full control of the entire software stack on the EC2 instances that look like hardware to them. On the other hand, this feature makes it inherently difficult for Amazon to offer automatic scaling of resources.


Amazon EC2: Instance Types

Amazon EC2	General Purpose	Compute Optimized	Storage and IO Optimized	GPU Enabled	Memory Optimized	Smaller instances
	M4	C4	I2 D2	G2	R3	T2
Added Instance Types	M3	C3	I2			
	M1	C1 CC2	HI1 HS1	CG1	M2 CR1	T1

Fig. 11. Amazon Elastic Cloud Computing(EC2) Instances

EC2 provides the ability to place instances in multiple locations. EC2 locations are composed of Regions and Availability Zones. Regions consist of one or more Availability Zones, are geographically dispersed. Availability Zones are distinct locations that are engineered to be insulated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region.

EC2 machine images are stored in and retrieved from Amazon Simple Storage Service (Amazon S3). S3 stores data as objects that are grouped in buckets. Each object contains from 1 byte to 5 gigabytes of data. Object names are essentially URI pathnames. Buckets must be explicitly created before they can be used. A bucket can be stored in one of several Regions. Users can choose a Region to optimize latency, minimize costs, or address regulatory requirements. Amazon Virtual Private Cloud (VPC) is a secure and seamless bridge between a companys existing IT infrastructure and the AWS cloud. Amazon VPC enables enterprises to connect their existing infrastructure to a set of isolated AWS compute resources via a Virtual Private Network (VPN) connection, and to extend their existing management capabilities such as security services, firewalls, and intrusion detection systems to include their AWS resources. For cloud users, Amazon CloudWatch is a useful management tool which collects raw data from partnered AWS services such as Amazon EC2 and then processes the information into readable, near real-time metrics. The metrics about EC2 include, for example, CPU utilization, network in/out bytes, disk read/write operations, etc.

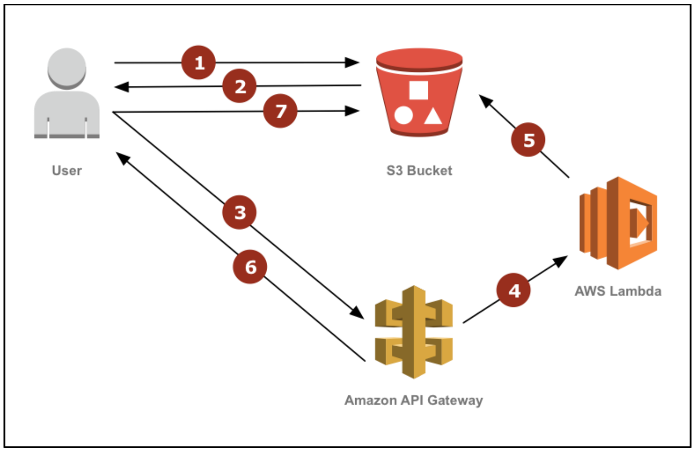


Fig. 12. Amazon Simple Sotrage Service(S3) Architecture

B. Microsoft Windows Azure platform

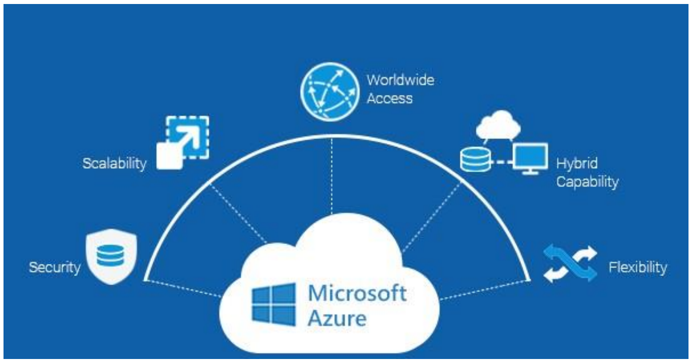


Fig. 13. Microsoft Azure

Microsofts Windows Azure platform [3] consists of three components and each of them provides a specific set of services to cloud users. Windows Azure provides a Windows-based environment for running applications and storing data on servers in data centers; SQL Azure provides data services in the cloud based on SQL Server; and .NET Services offer distributed infrastructure services to cloud-based and local applications. Windows Azure platform can be used both by applications running in the cloud and by applications running on local systems. Windows Azure also supports applications built on the .NET Framework and other ordinary languages supported in Windows systems, like C, Visual Basic, C++, and others. Windows Azure supports general-purpose programs, rather than a single class of computing. Developers can create web applications using technologies such as ASP.NET and Windows Communication Foundation (WCF), applications that run as independent background processes, or applications that combine the two. Windows Azure allows storing data in blobs, tables, and queues, all accessed in a RESTful style via HTTP or HTTPS. SQL Azure components are SQL Azure Database and Huron Data Sync. SQL Azure Database is built on Microsoft SQL Server, providing a database management system (DBMS) in the cloud. The data can be accessed using ADO.NET and other Windows data access interfaces. Users can also use on-premises software to work with this cloud-based information. Huron Data Sync synchronizes relational data across various on-premises DBMSs.

C. *Google Cloud*

Google Cloud [2] is a platform for traditional web applications in Google-managed data centers. Currently, the supported programming languages are Python and Java. Web frameworks that run on the Google App Engine include Django, CherryPy, Pylons, and web2py, as well as a custom Google-written web application framework similar to JSP or ASP.NET. Google handles deploying code to a cluster, monitoring, failover, and launching application instances as necessary. Current APIs support features such as storing and retrieving data from a BigTable non-relational database, making HTTP requests and caching. Developers have readonly access to the filesystem on App Engine.

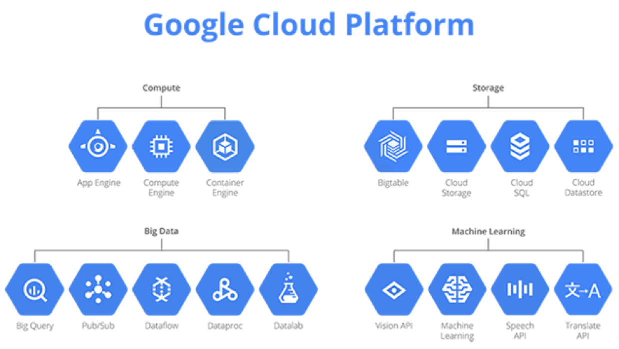


Fig. 14. Google Cloud Computing

VII. Conclusion

Several papers studied the performance of big data applications on scale-out platform and clouds [21, 22, 23, 24, 25, 26]. All of these works use performance counters to monitor the performance and behavior of applications. In [27, 28, 29, 30, 31], authors perform a set of comprehensive experiments to analysis the impact of memory subsystem on the performance of data intensive applications running on cloud environment. In [32, 33], author uses compress sensing to improve data movement after finding the performance bottleneck using performance counters. Performance counters also can be used to trace the applications behavior in order to find the malicious behavior [34, 35, 36, 37, 38]. Moreover, there are new approaches to improve the performance of modern computing systems such as hardware acceleration [39, 40, 41, 42], and cloud computing.

Cloud computing has recently emerged as a compelling paradigm for managing and delivering services over the Internet. The rise of cloud computing is rapidly changing the landscape of information technology, and ultimately turning the long-held promise of utility computing into a reality. However, despite the significant benefits offered by cloud computing, the current technologies are not matured enough to realize its full potential. Many key challenges in this domain, including automatic resource provisioning, power management and security management, are only starting to receive attention from the research community. Therefore, I believe there is still tremendous opportunity for researchers to make groundbreaking contributions in this field, and bring significant impact to their development in the industry.

In this paper, I have surveyed the state-of-the-art of cloud computing, covering its essential concepts, architectural designs, prominent characteristics, key technologies as well as research directions. As the development of cloud computing technology is still at an early stage, I hope this work will provide a better understanding of the design challenges of cloud computing, and pave the way for further research in this area.

References

[1]Amazon web services, aws.amazon.com.
[2]Google cloud, cloud.google.com.
[3]Windows azure, www.microsoft.com/azure.
[4]Xensource inc, xen, www.xensource.com.

[5] Agarwal, Sameer, et al. "Reoptimizing data parallel computing." Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12). 2012.

[6] Alipourfard, Omid, et al. "Cherrypick: Adaptively unearthing the best cloud configurations for big data analytics." 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17). 2017.

[7] Christina Delimitrou and Christos Kozyrakis. Quasar: resource-efficient and qos-aware cluster management. In *ACM SIGARCH Computer Architecture News*, volume 42, pages 127–144. ACM, 2014.

[8] Michael Ferdman, Almutaz Adileh, Onur Kocberber, Stavros Volos, Mohammad Alisafaee, Djordje Jevdjic, Cansu Kaynak, Adrian Daniel Popescu, Anastasia Ailamaki, and Babak Falsafi. Clearing the clouds: a study of emerging scale-out workloads on modern hardware. In *ACM SIGPLAN Notices*, volume 47, pages 37–48. ACM, 2012.

[9] Andrew D Ferguson, Peter Bodik, Srikanth Kandula, Eric Boutin, and Rodrigo Fonseca. Jockey: guaranteed job latency in data parallel clusters. In *Proceedings of the 7th ACM european conference on Computer Systems*, pages 99–112. ACM, 2012.

[10] Chin-Jung Hsu, Vivek Nair, Vincent W Freeh, and Tim Menzies. Low-level augmented bayesian optimization for finding the best cloud vm. *arXiv preprint arXiv:1712.10081*, 2017.

[11] Botong Huang, Matthias Boehm, Yuanyuan Tian, Berthold Reinwald, Shirish Tatikonda, and Frederick R Reiss. Resource elasticity for large-scale machine learning. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 137–152. ACM, 2015.

[12] Virajith Jalaparti, Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. Bridging the tenant-provider gap in cloud services. In *Proceedings of the Third ACM Symposium on Cloud Computing*, page 10. ACM, 2012.

[13]Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011)

[14] Kristi Morton, Magdalena Balazinska, and Dan Grossman. Paratimer: a progress indicator for mapreduce dags. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pages 507–518. ACM, 2010.

[15] Dejan Novakovi c´, Nedeljko Vasic´, Stanko Novakovic´, Dejan Kostic´, and Ricardo Bianchini. Deepdive: Transparently identifying and man- aging performance interference in virtualized environments. In Presented as part of the 2013 USENIX Annual Technical Conference (USENIX ATC 13), pages 219–230, 2013.

[16] Juwei Shi, Jia Zou, Jiaheng Lu, Zhao Cao, Shiqiang Li, and Chen Wang. Mrtuner: a toolkit to enable holistic optimization for mapreduce jobs. Proceedings of the VLDB Endowment, 7(13):1319–1330, 2014.

[17] Lingjia Tang, Jason Mars, Neil Vachharajani, Robert Hundt, and Mary Lou Soffa. The impact of memory subsystem resource sharing on datacenter applications. In ACM SIGARCH Computer Architecture News, volume 39, pages 283–294. ACM, 2011.

[18] Shivaram Venkataraman, Zongheng Yang, Michael Franklin, Benjamin Recht, and Ion Stoica. Ernest: efficient performance prediction for large-scale advanced analytics. In 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), pages 363–378, 2016.

[19] R Clint Whaley, Antoine Petitet, and Jack J Dongarra. Automated empirical optimizations of software and the atlas project. Parallel computing, 27(1-2):3–35, 2001.

[20] Neeraja J Yadwadkar, Bharath Hariharan, Joseph E Gonzalez, Burton Smith, and Randy H Katz. Selecting the best vm across multiple public clouds: a data-driven performance modeling approach. In Proceedings of the 2017 Symposium on Cloud Computing, pages 452–465. ACM, 2017.

[21] Makrani, Hosein Mohammadi, et al. "Evaluation of software-based fault-tolerant techniques on embedded OS's components." Proceedings of the International Conference on Dependability (DEPEND'14). 2014.

[22] Makrani, Hosein Mohammadi, et al. "Energy-aware and Machine Learning-based Resource Provisioning of In-Memory Analytics on Cloud." SoCC. 2018.

[23] Sayadi, Hossein, et al. "Machine learning-based approaches for energy-efficiency prediction and scheduling in composite cores architectures." 2017 IEEE International Conference on Computer Design (ICCD). IEEE, 2017.

[24] Malik, Maria, Dean M. Tullsen, and Houman Homayoun. "Co-Locating and concurrent fine-tuning MapReduce applications on microservers for energy efficiency." 2017 IEEE International Symposium on Workload Characterization (IISWC). IEEE, 2017.

[25] Malik, Maria, et al. "ECoST: Energy-Efficient Co-Locating and Self-Tuning MapReduce Applications." Proceedings of the 48th International Conference on Parallel Processing. ACM, 2019.

[26] Sayadi, Hossein, et al. "Power conversion efficiency-aware mapping of multithreaded applications on heterogeneous architectures: A comprehensive parameter tuning." 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2018.

[27] Makrani, Hosein Mohammadi, et al. "Understanding the role of memory subsystem on performance and energy-efficiency of Hadoop applications." 2017 Eighth International Green and Sustainable Computing Conference (IGSC). IEEE, 2017.

[28] Makrani, Hosein Mohammadi, and Houman Homayoun. "MeNa: A memory navigator for modern hardware in a scale-out environment." 2017 IEEE International Symposium on Workload Characterization (IISWC). IEEE, 2017.

[29] Makrani, Hosein Mohammadi, and Houman Homayoun. "Memory requirements of hadoop, spark, and MPI based big data applications on commodity server class architectures." 2017 IEEE International Symposium on Workload Characterization (IISWC). IEEE, 2017.

[30] Makrani, Hosein Mohammadi, et al. "A comprehensive memory analysis of data intensive workloads on server class architecture." Proceedings of the International Symposium on Memory Systems. ACM, 2018.

[31] Makrani, Hosein Mohammadi, et al. "Main-memory requirements of big data applications on commodity server platform." 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE, 2018.

[32] Namazi, Mahmoud, et al. "Mitigating the Performance and Quality of Parallelized Compressive Sensing Reconstruction Using Image Stitching." Proceedings of the 2019 on Great Lakes Symposium on VLSI. ACM, 2019.

[33] Makrani, Hosein Mohammadi, et al. "Compressive Sensing on Storage Data: An Effective Solution to Alleviate I/0 Bottleneck in Data-Intensive Workloads." 2018 IEEE 29th International Conference on Application-specific Systems, Architectures and Processors (ASAP). IEEE, 2018.

[34] Sayadi, Hossein, et al. "Ensemble learning for effective run-time hardware-based malware detection: A comprehensive analysis and classification." 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC). IEEE, 2018.

[35] Sayadi, Hossein, et al. "Customized machine learning-based hardware-assisted malware detection in embedded devices." 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, 2018.

[36] Sayadi, Hossein, et al. "Comprehensive assessment of run-time hardware-supported malware detection using general and ensemble learning." Proceedings of the 15th ACM International Conference on Computing Frontiers. ACM, 2018.

[37] Dinakarrao, Sai Manoj Pudukotai, et al. "Lightweight Node-level Malware Detection and Network-level Malware Confinement in IoT Networks." 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019.

[38] Sayadi, Hossein, et al. "2SMaRT: A Two-Stage Machine Learning-Based Approach for Run-Time Specialized Hardware-Assisted Malware Detection." 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2019.

[39] Neshatpour, Katayoun, et al. "Design Space Exploration for Hardware Acceleration of Machine Learning Applications in MapReduce." 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). IEEE, 2018.

[40] Makrani, Hosein Mohammadi, et al. "XPPE: cross-platform performance estimation of hardware accelerators using machine learning." Proceedings of the 24th Asia and South Pacific Design Automation Conference. ACM, 2019.

[41] Neshatpour, Katayoun, et al. "Architectural considerations for FPGA acceleration of Machine Learning Applications in MapReduce." Proceedings of the 18th International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation. ACM, 2018.

[42] Makrani, Hosein Mohammadi, et al. "Pyramid: Machine Learning Framework to Estimate the Optimal Timing and Resource Usage of a High-Level Synthesis Design." arXiv preprint arXiv:1907.12952 (2019).