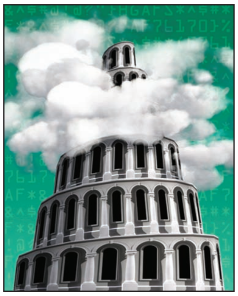
Editor: George Pallis • *gpallis@cs.ucy.ac.cy*



Comparing Public-Cloud Providers

Ang Li and Xiaowei Yang • *Duke University*
Srikanth Kandula and Ming Zhang • *Microsoft Research*

As cloud computing becomes increasingly popular among enterprises, developers, and organizations, it's time to consider the practical problem: how do you choose from the growing number of providers? To help users make this decision, researchers first determined the basic types of services that providers offer. They then devised a set of metrics related to application performance in a cloud and created tools for measuring them.

The public-cloud-computing market has grown tremendously. Different from private clouds, which organizations use internally, public-cloud platforms are open to virtually anyone with a credit card. Public-cloud customers can easily spin up tens or even hundreds of virtual machines (VMs) in a few minutes; they pay for only what they actually use, without any up-front investment. Such flexibility has motivated more and more organizations, including governments, schools, enterprises, and content providers, to migrate their applications to public-cloud platforms.

Following the trend, more and more companies are rolling out public-cloud-computing services, such as Amazon Web Services (AWS), Google's AppEngine, Microsoft's Windows Azure, and Rackspace's Cloud Servers and Cloud Sites. Although the variety of public-cloud providers fosters healthy competition, it makes choosing the best cloud for an application difficult.

Imagine that you want to migrate your email service to a cloud. Identifying the providers that offer the necessary features to support the application is easy because most of them have similar feature sets. However, telling which cloud offers the best email-processing performance is difficult, mainly because no detailed, comprehensive performance comparison between cloud providers exists. The performance specifications that providers publish are usually vague (for example, "every cloud server gets four virtual CPUs"). Also, the numerous industrial reports and blog posts on this topic[1–3] are rather ad hoc and don't cover all cloud aspects. Many focus on VM computation speed, ignoring other aspects such as the storage and network services' performance.

We recently performed a comprehensive comparison of public-cloud providers' performance.[4] The study led us to systematically determine which metrics best characterize cloud performance. Proper application of these metrics can help users quickly choose the best-performing provider for their application.

Cloud Services

To support diverse applications, cloud providers typically offer a spectrum of services in categories such as computation, storage, and networking. For instance, AWS offers the Elastic Compute Cloud (EC2) computation service, SimpleDB storage service, Simple Storage Service (S3), and Simple Queue Service (SQS), and Amazon's internal networking service that connects the other services. We've concluded that current major cloud providers offer the following four common types of services.

Elastic Compute Clusters

A compute cluster includes a set of virtual instances that run a customer's application code. Each virtual instance can be a bare-metal VM (in an infrastructure-as-a-service provider, such as AWS and Cloud Servers) or a sandbox environment (in a platform-as-a-service provider, such as AppEngine). Clusters are elastic in that the number of instances can scale dynamically with the application's workload. For instance, in a cloud-based Web application, the number of front-end server instances can scale according to the incoming request rate, so that each server instance won't be overwhelmed by too many simultaneous requests.

Persistent Storage Services

These services store application data and nonephemeral state; all instances in the cluster can access them. They're different from the local storage (for example, the local hard drive) in each virtual instance, which is temporary and can't be directly accessed by other instances. They're also different from block storage services that some providers offer (for example, Amazon's Elastic Block Storage). The latter can't be accessed by multiple instances simultaneously and serves primarily as backup.

There are several common types of storage services. *Table* storage (SimpleDB, Google's DataStore, and Azure's Table Storage) is similar to a traditional database. *Blob* storage (S3, Rackspace's Cloud Files, and Azure's Blob Storage) keeps binary objects such as user photos and videos. *Queue* storage (SQS and Azure's Queue Storage) is a special type of storage service. It implements a global message queue that enables synchronization across different tiers of virtual instances. Persistent storage services are usually implemented as RESTful Web services[5] (REST stands for Representational State Transfer) and are highly available and scalable compared to their noncloud siblings.

Intracloud Networks

These networks connect virtual instances with each other and with storage services. All clouds promise high-bandwidth and low-latency networks in a datacenter. This is because network performance is critical to the performance of distributed applications such as multitier Web services and MapReduce jobs.

Wide-Area Networks

Unlike intracloud networks, which connect an application's components, wide-area networks (WANs) connect the cloud datacenters, where the application is hosted, with end hosts on the Internet. For consumer applications such as websites, WAN performance can affect a client's response time significantly. All cloud providers operate multiple datacenters at different geographical regions so that a user's request can be served by a nearby datacenter to reduce WAN latency.

Putting Them All Together

These four types of services are common because they're fundamental in building a generic online computation platform. Imagine a typical online cloud application, such as a social network website. Its servers can run in the compute cluster, leveraging the scaling feature to absorb flash-crowd events. Its user data can be stored in the various storage services and accessed through the intracloud network. Its Web content can be delivered to users with just a short delay, with a WAN's help. Other important cloud services, such as MapReduce (Hadoop) services and backup services, aren't as common, probably because they aren't essential to most cloud applications.

Comparing Clouds

To compare clouds along the four common services, we must first choose the metrics for each service. There are two main rationales.

First, we want the metrics to be directly related to application performance. Our primary goal is to make selecting cloud providers easier. So, the metrics should be easy to understand and directly reflect some aspects of the application's performance. For instance, the storage metrics should reflect a storage-intensive application's I/O efficiency. We also don't compare service features that are unrelated to performance. Although some of them are important, choosing them is largely orthogonal to optimizing the application's performance.

Second, we want the metrics to be fair across different providers, regardless of how they implement the services. Different providers might implement services in drastically different ways. For example, some providers use Xen VMs to implement virtual instances, whereas others use Hyper-V or other proprietary virtualization technologies. The metrics should abstract away these implementation details and focus on the services' end-to-end performance impact.

Following these rationales, we chose 10 metrics that together comprehensively depict cloud performance (see Table 1). Some of the metrics are standard and are already widely used in noncloud contexts:

- a benchmark task's runtime, to measure a virtual instance's efficiency,
- a storage service's latency and throughput, and
- a network path's latency and capacity.

On the other hand, clouds' unique characteristics led us to choose the following novel and unconventional metrics.

Performance versus Cost

Many cloud services aren't free; users must pay for the resources consumed. A budget-aware customer might choose to not go with the cloud provider that performs the best but is expensive, instead choosing the most cost-effective one.

To help users make this choice, we developed two metrics to capture the cost-effectiveness of the two main billable cloud services: the compute cluster and the storage services. We

Table 1. Cloud performance metrics.

Service	Metric
Elastic compute cluster	Benchmark runtime
	Cost per benchmark
	Scaling latency
Persistent storage	Operation latency
	Operation throughput
	Cost per operation
	Time to consistency
Intracloud network	Path latency
	Path capacity
Wide-area network	Wide-area-network latency

first use the monetary cost of finishing a benchmark task to capture a virtual instance's cost-effectiveness. Then, we use the cost of each storage operation to compare different storage services' cost-effectiveness.

Scaling Speed

A unique feature of a cloud is its highly scalable compute cluster. Customers can easily scale a cluster from a few instances to hundreds of instances to cope with, for instance, a website's sudden surge in popularity. Scaling speed can be crucial to both the performance and cost of an application with a variable workload. When the workload increases, the scaling speed needs to catch up so that the quality of service won't degrade. When the workload is low, the customer still needs to keep a few backup instances in case the workload suddenly increases. So, the faster the cluster can scale, the fewer backup instances the customer needs, which decreases costs.

We use the scaling-latency metric to compare compute clusters' scaling speed. We measure latency as the time it takes to spin up a new instance, from when the instance is requested to when it's ready to serve the application.

Storage Consistency

Clouds usually implement storage services in a distributed fashion to improve availability and scalability. However, as a trade-off, some services don't provide a strong consistency guarantee, which means a read operation might return outdated data for a short period of time.

To compare data consistency across different providers' storage services, we chose the time-to-consistency metric. This metric measures the time a data item (a table row, binary object, or queue message) takes to become available after it's added to the service.

WAN Latency

We compare WAN performance by measuring the minimum latency from a vantage point to all a provider's datacenters. Suppose a provider has a datacenter in the US and one in Europe. For a vantage point in the US, we measure the network latency to both datacenters. The latency to the US datacenter is smaller owing to geographic proximity, so we choose it as the minimum latency. Such latency also corresponds to the actual latency for that vantage point under a perfect load-balancing algorithm that always directs a request to the closest datacenter. We perform the measurements from a set of geographically diverse vantage points on PlanetLab (www.planet-lab.org).

Measuring the Metrics

We developed a set of tools to measure these metrics under four major providers: AWS, AppEngine, Azure, and Cloud Servers. Each tool is simple and can be easily extended to other cloud platforms. We're releasing the tools on our project website (www.cloudcmp.net) so that customers can use them to collect comparison results for the providers they're interested in. Also, some third-party companies could use the tools to offer real-time cloud performance comparison services. Customers would only need to subscribe to such services and download the comparison data for the time frame and providers they're interested in.

From Cloud Performance to Application Performance

Ultimately, a cloud performance comparison aims to enable fast, accurate provider selection. Customers care about their application's performance on the different cloud providers and want to choose the provider that offers the best performance for that application. Here, we discuss two ways customers can use the comparison results to choose a provider.

Direct Performance Projection

Customers can directly use the comparison results along one or a few metrics to project an application's performance with different cloud providers. For instance, for a computation-intensive application, such as document processing, a customer could directly use the benchmark runtime to discover which cloud's instance will likely run the application most efficiently. Similarly, for a storage-intensive application, such as an e-commerce website, a customer could use the storage service latency and throughput to find the provider with the most efficient storage services. We've demonstrated this approach's usefulness with three realistic applications.[4]

Trace-Based Performance Prediction

The previous approach is simple but might be limited because in some cases it's unclear how to combine the results of multiple relevant metrics. For instance, a complex application, such as video processing, can be both computation- and storage-intensive; combining the computation and storage comparison results is a nontrivial task.

One possible way to address this problem is to instrument the application locally and combine its local execution trace with the measurement results to predict its overall performance in a cloud. An execution trace shows how the application runs locally and the resource consumption at each step. Figure 1 shows an execution trace of a request in a standard three-tier Web application. The trace shows how the different components (the Web server, application server, network, and back-end database) process the request. It also shows the resources consumed at each component (the CPU time consumed at each server component, the amount of data sent through the network, and the database queries).

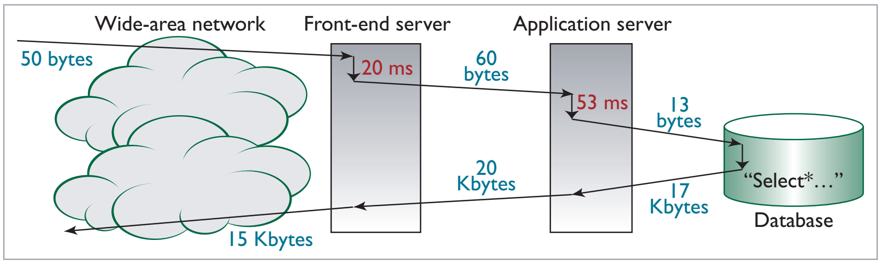


Figure 1. The execution trace of a request in a standard three-tier Web application. The red numbers show the CPU time each server component consumed to process the request; the blue numbers show the amount of data sent through the network (both over a wide-area network and inside a cloud). The trace also includes the query issued to the back-end database.

We can then use the measurement results to predict the time spent at each cloud component. For a server component, we can scale the CPU time consumed locally by the speed difference between the local and the cloud instances. For the time spent in network transmission, we can use the cloud network latency and capacity. For the time spent in database queries, we can directly use the storage services' operation latency. We then combine all the predicted times; the resulting value is a good estimation of the application's real runtime in the cloud.

This approach, however, has challenges. One major challenge is how to automatically extract an application's execution trace, given that the traces can differ widely across different types of applications. Another challenge is how to accurately predict the time spent on a virtual instance, because simple scaling might introduce excessive error in some cases. An interesting and open research question is, how can we design a generic cloud-performance-prediction framework with high accuracy and minimum customer involvement?

We believe our approach is a significant step toward enabling fast, accurate provider selection for thousands of emerging cloud applications. However, there's still much to do in this area. For example, each cloud service deserves its own thorough, systematic performance analysis. This can provide insight in how to optimize the existing applications to better fit the cloud environment. Furthermore, performance might not be the only criterion for selecting a provider. Aspects such as manageability, availability, and data redundancy also concern the customers. How to integrate these factors into a complete, easy-to-use provider-selection framework remains an open challenge.

References

1. "Comparing Amazon EC2 Performance with Other Cloud/VPS Hosting Options ... and Real Hardware," blog, 14 Apr. 2009; www.paessler.com/blog/2009/04/14/prtg-7/comparing-amazon-ec2-performance-with-other-cloudvps-hosting-options-and-real-hardware.
2. "Rackspace Cloud Servers versus Amazon EC2: Performance Analysis," blog; www.thebitsource.com/featured-posts/rackspace-cloud-servers-versus-amazon-ec2-performance-analysis.
3. J.S. Ward, "A Performance Comparison of Clouds: AmazonEC2 and Ubuntu Enterprise Cloud," presented at 2009 Scottish Informatics and Computer Science Alliance Demofest, 2009; www.cs.st-andrews.ac.uk/files/PerformanceComparison.pdf.
4. A. Li et al., "CloudCmp: Comparing Public Cloud Providers," *Proc. 10th Ann. Conf. Internet Measurement*, ACM Press, 2010, pp. 1–14; http://research.microsoft.com/apps/pubs/?id=136448.
5. R.T. Fielding and R.N. Taylor, "Principled Design of the Modern Web Architecture," *ACM Trans. Internet Technology*, vol. 2, no. 2, 2002, pp. 115–150.

Ang Li is a PhD student in Duke University's Department of Computer Science. Contact him at angl@cs.duke.edu.

Xiaowei Yang is an assistant professor at Duke University's Department of Computer Science. Contact her at xwy@cs.duke.edu.

Srikanth Kandula is a researcher in Microsoft Research's Networking Research Group. Contact him at srikanth@microsoft.com.

Ming Zhang is a researcher in Microsoft Research's Networking Research Group. Contact him at mzh@microsoft.com.