Systems Engineering at MITRE
CLOUD COMPUTING SERIES

Leveraging Public Clouds to Ensure Data Availability

Toby Cabot
Lawrence Pizette

MITRE

January 2013

January 2013

Table of Contents

THE BIG PICTURE: Replicated databases in cloud environments are a cost-effective alternative to explore for ensuring the availability of data.

Leveraging Public Clouds to Ensure Data Availability

Toby Cabot
Lawrence Pizette

1.0 Problem

In the event of a disaster such as a hurricane, earthquake, or attack by an adversary, a data center hosting large amounts of U.S. Government data could become unavailable within seconds. This fragility requires that the U.S. Government establish strategies for safeguarding and restoring access to data commensurate with operational needs. As it is not possible to move many gigabytes (GB), terabytes (TB), or petabytes (PB) of information across a wide area network in the initial seconds of a disaster, data must be pre-positioned at alternate locations.

"White House [former] chief information officer Vivek Kundra … announced [in February 2010] an initiative to consolidate hundreds of redundant federal government databases," writes Byron Acohido. He adds, "Kundra also called for stepping up the federal government's reliance on cloud-based systems to deliver public services." These databases are widespread and permeate Government Information Technology. Forrester Research writes that databases are a "critical asset to any enterprise." It estimates the database market to reach $32 billion per year by 2013.

2.0 Analysis

System architects can consider several options for backing up data. First, replication can be employed by pre-positioning geographically separate but identical databases. As transactions are committed to the primary database, they are copied to the backup database. Second, very large data sets can be copied onto physical media, shipped to backup locations, and imported into standby databases. This process has the drawback of not being in real time, but captures a snapshot of when the media is copied. It can be faster than using a network to transfer very large, multi-TB datasets.[1] Third, databases can be copied to backup media and stored off site. When needed, the off-site storage can be used to recreate databases in a geographically separate environment. This process can be less costly because the data is simply stored offline unless needed. However, it has the drawback of being a snapshot in time and not quickly available for mission-critical systems.

If a conventional, dedicated data center approach is used, all these options may be problematic. The first two options require continual operation of two data centers with support staff, duplicate infrastructure, and software licensing costs. The first option also requires continual network connectivity. The third option requires the availability of a receiving data center with adequate spare capacity whenever it may be needed.

Many public cloud providers offer redundancy solutions to address these challenges. For example, commercial companies such as Microsoft, Rackspace, and Amazon Web Services (AWS) offer network-based replication of their public cloud database products.[2,3,4] And as an example of shipment-based back-up, Amazon provides a service, AWS Import/Export, that allows flash drives and hard drives to be shipped to Amazon for importing to their Simple Storage Service (S3).[5]

This paper addresses potential uses of cloud computing approaches to facilitate data access in the event of a failure in a primary location. We conclude that replicated databases in cloud environments are a cost-effective alternative for ensuring the availability of data.

There is nothing magic about cloud computing. It can provide a significant reduction in infrastructure cost by sharing costs among multiple users and leveraging the economies of scale of very large operations. This can reduce the cost of many operations, including database replication, whether populated with imported data (e.g., by shipping disk drives) or on-demand or real-time backups.

This paper will focus on a public cloud provider example to illustrate the costs for cloud-based database replication. As the government develops alternatives that provide analogous services and realize a degree of similar cost savings, these will offer more private means to the same end. Every situation requires it's own analysis based on the consideration in Section 3.

3.0 Cloud Computing Considerations

We recommend the following analyses:

Systems Engineering and Technical Analysis: Perform an analysis to determine whether cloud deployment is appropriate. Cloud computing has significant benefits that can be leveraged by many projects, but policy, security or technical obstacles can make cloud computing (shared or public) difficult or suboptimal for some projects. If cloud represents a viable solution, ensure that the provider's storage capacity limitations are greater than the planned database size. Run a pilot, if necessary.

Analyze and Perform a Return on Investment Calculation: Analyze the system to determine how tolerant it can be to unavailability or loss of data, and whether the system could benefit from a cloud computing approach. Some systems can tolerate outages of a day or two while a new hosting capability is provisioned, some can tolerate hours of downtime, while other systems cannot be offline for more than a few seconds without severe consequence. Another consideration is tolerance to lost transactions before and during failover. Some systems can tolerate the loss of some transactions that were processed shortly before the failover, while others cannot.

The typical government Web-based system is a three-tier architecture: presentation, business logic, and database. It often uses a browser and Web server to provide a graphical user interface. When a user enters data or clicks on the browser, a request is sent to the Web server and, for dynamic content, to the business logic tier. The business logic tier relies upon the database to read and write persistent data. One browser action from the user can result in many reads and writes to the database. For example, an inventory system may receive a request to provide a product to a user. The business logic tier may need to validate if the inventory is available, verify the shipping address, and decrement the requested amount from inventory. This persistent data is maintained in a relational database management system that may be a candidate for the cloud.

The implications of being "off-line" will determine the operational benefit of investing in different cloud strategies. The total costs should also be analyzed; these include software license and support agreements that can be affected by cloud deployments.

Replicate Database: If warranted by the return on investment, systems engineering, and technical analysis, pre-position a replicated copy of the data in a secondary cloud environment. For example, most enterprise-scale relational database management systems (RDBMSs) offer the capability to copy data from a primary database to an always-running secondary database (i.e., "hot" standby). The primary and secondary databases can be hosted at geographically separate locations, and can be configured as cloud-to-cloud replication or replication between a private datacenter and cloud service.

4.0 Prototype

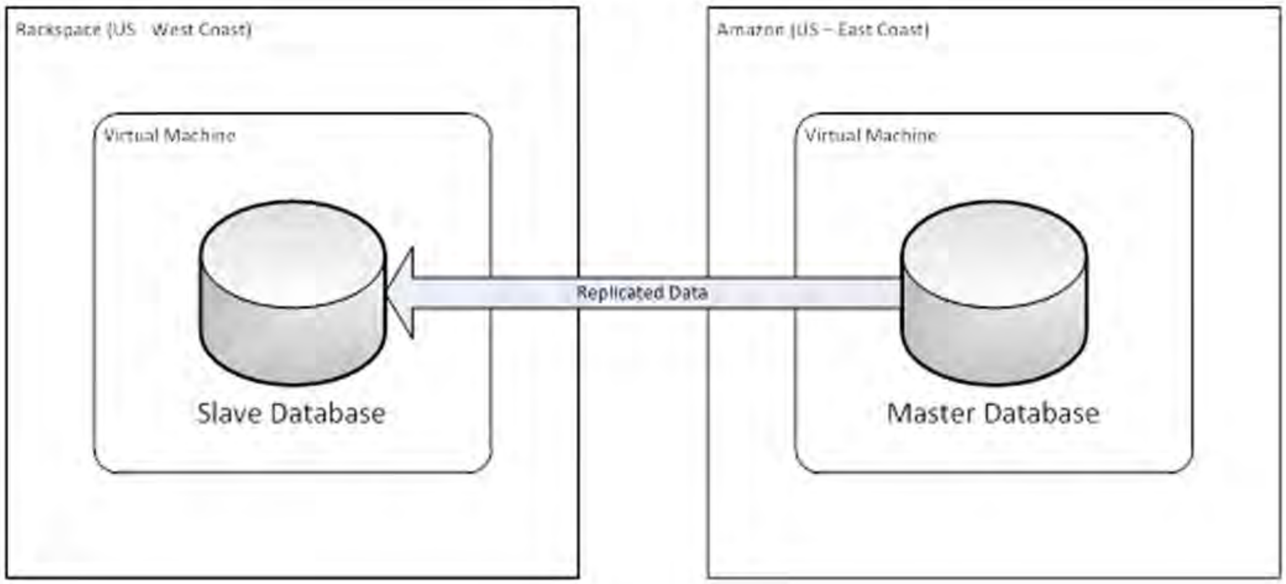

Figure 1. Prototype High-Level View

To demonstrate the capabilities and determine the cost effectiveness of cloud computing for backup, an exemplar MySQL RDBMS database hosted in AWS' Elastic Compute Cloud (EC2) environment was prototyped and replicated to Rackspace. We performed one test at "small" scale and one at "medium" scale. The medium scale test first wrote 1 billion database records—500 million names and 500 million address records—for a sample size of 37.5 GB of "raw" data. Subsequently, the test updated 50 million records of each type, then deleted 50 million records of each type. Update and delete activity drives additional communication between databases. Vendor monitoring tools were used to measure the amount of traffic in and out of the environments, and that data helped calculate costs for the cloud environments. The prototype took approximately 60 hours to run at medium scale. The outbound network traffic from AWS to Rackspace was 210.6 GB, and the inbound traffic from Rackspace to AWS (e.g., for polling of replicated transactions) was 5.6 GB.

The small-scale test was similar to the medium-scale test, but at an order of magnitude smaller scale: 100 million records were written and 10 million database records were modified and deleted.

The gathered prototype data helped extrapolate the cost of large databases. Thus, a large database will be one order of magnitude larger than the medium scale prototype (i.e., 375 GB of "raw" data). Because TB-scale and PB-scale databases can have additional processing needs and may not be appropriate for a single-instance type of environment, the analysis was not scaled to that degree, which would require a parallel processing, horizontally scalable, database management system (e.g., a sharded RDBMS, RDBMS cluster, "not only SQL" [NoSQL] database, or map/reduce cluster).

This prototype addressed the database only. There was no attempt to model failover at application tiers above the database, although this can be a significant systems engineering problem.

While each organization must perform its own cost of ownership analysis based upon its specific processing needs and costs, the hosting and input/output (I/O) costs of running a database in the cloud can be low. Assuming the equivalent volume of data from the prototype spans one month, the monthly costs are shown in Table 1. While the services leveraged are well-known cloud offerings, there are other services from these and other providers that could increase or decrease the costs.

Cost element[7]	Small Database (prototype)	Medium Database (prototype)	Large Database (calculated)
Raw data size	3.75 GB	37.5 GB	375 GB
AWS Elastic Block Store 10 GB @ $0.10/GB month (assume that 10 GB is allocated for each 3.75 GB of data, due to overhead of indexes, etc.) and $0.10 per million I/O requests[8]	$6.27 ($1.00 for 10 GB of EBS plus $0.67 for 6.67 million I/O requests)	$62.70 ($10.00 for 100 GB of EBS plus $6.90 for 69 million I/O requests)	$627.00 ($100.00 for 1,000 GB of EBS plus $69.00 for 690 million I/O requests)
Large EC2 On-Demand Linux instance running 24 hrs./day for 30 days @ $0.32/hr. (running in U.S. East Region)[9]	$230.40	$230.40	N/A
Extra-Large EC2 On-Demand Linux instance running 24 hrs./day for 30 days @ $0.64/hr. (running in U.S. East Region)[10]	N/A	N/A	$460.80
Rackspace Linux instance running 24 hrs./day for 30 days @ $0.24/hour (4,096 MB RAM, 160 GB Disk)[11]	$172.80	$172.80	N/A
Rackspace Linux instance running 24 hrs./day for 30 days @ $0.96/hour (15,872 MB RAM, 620 GB Disk)[12]	N/A	N/A	
Data transfer out of EC2 (1st GB free, then $0.12 per GB up to 10 TB)	21.5 GB costs $2.46	21.5 GB costs $2.46	2,106 GB costs $252.60
Data transfer out of Rackspace ($0.18/GB)	0.6 GB costs $0.11	5.6 GB costs $1.00	140 GB costs $25.20
Total	$412.71	$498.95	$2,125.80

Table 1. Cost Analysis

5.0 Conclusion

Given the criticality of many government services, adequate attention must be paid to disaster recovery considerations. This becomes even more crucial as database consolidation potentially increases the impact of the loss of single data center. Replicated databases in cloud environments are a cost-effective alternative to explore for ensuring the availability of data. Given that the capabilities of some commercial Infrastructure as a Service offerings are designed to meet many U.S. Government needs for Federal Information Security Management Act moderate systems, Federal Information Technology leadership has more options for providing cost-effective availability to its customers.

Acronyms

Acronym	Definition
AWS	Amazon Web Services
COTS	Commercial Off-the-Shelf
DB	Database
EBS	Elastic Block Store
EC2	Elastic Compute Cloud
GB	Gigabyte
I/O	Input/Output
NoSQL	Not only SQL
PB	Petabyte
RDBMS	Relational Database Management Systems
ROI	Return on Investment
TB	Terabyte

References

[1] A 1 TB database would take over 1,000 hours to move a network at the T1 connection speed of 1.544 million bits per second and almost 1 hour at the much faster fiber optic OC48 rate of 2,488 million bits per second.

[2] Amazon. (2012, July 26). "Amazon FAQs," Amazon Web Services. http://aws.amazon.com/rds/faqs/#86.

[3] Windows Azure. (2012, July 26). "Windows Azure SQL Database Overview," Windows Azure. http://msdn.microsoft.com/en-us/library/windowsazure/ee336241.aspx.

[4] Google. (2012, July 26). "Structuring Data for Strong Consistency," Google Developers. http://code.google.com/appengine/docs/python/datastore/hr/overview.html.

[5] Amazon. (2012, July 26). "AWS Import/Export," Amazon Web Services. http://aws.amazon.com/importexport/.

[6] The amount of disk storage required was significantly larger than 37.5 GB, due to database storage efficiency, log files, indexes, etc.

[7] Prices as of April 26, 2012

[8] Amazon. (2012, July 26). "Amazon EC2 Pricing," Amazon Web Services. http://aws.amazon.com/ec2/pricing/.

[9] Ibid.

[10] Ibid.

[11] Rackspace. (2012, July 26). "How We Price Cloud Servers," Rackspace Cloud Servers. http://www.rackspace.com/cloud/cloud_hosting_products/servers/pricing/.

[12] Ibid.

MITRE

www.mitre.org

Case Number: 12-0230
Document Number: MTR120400