



The Total Cost of (Non) Ownership of a NoSQL Database Cloud Service

Jinesh Varia and Jose Papo
March 2012

(Please consult <http://aws.amazon.com/whitepapers/> for the latest version of this paper)

Introduction

Weighing the financial considerations of owning and operating a data center or co-located facility versus employing a cloud infrastructure or a cloud service requires detailed and careful analysis. In practice, it is not as simple as just measuring potential hardware expense alongside utility pricing for compute and storage resources. The Total Cost of Ownership (TCO) is often the financial metric that is used to estimate and compare direct and indirect costs of a product or a service. While it is challenging to do the right apples-to-apples comparison between on-premises software and a cloud service, in this whitepaper, we attempt to explain the economic benefits of using a NoSQL (non-relational) database cloud service such as Amazon DynamoDB over equivalent NoSQL database software that is deployed on-premises or hosted in the cloud.

The goal of this whitepaper is to help you understand the different cost factors involved in deploying and managing a scalable NoSQL service or solution. We walk through an example scenario (a social game to support the launch of a new movie) and highlight the total costs for three different options. We state our assumptions in each option so you can adjust them based on your own research or quotes from your hardware vendors and co-location providers.

Major Cost Considerations that Are Often Overlooked

When determining the TCO of a cloud-based service, it's easy to overlook several cost factors such as administration and redundancy costs, which can lead to an inaccurate and incomplete comparison. Additionally, in the case of a NoSQL database solution, people often forget to include database administration costs.

First, it's important to understand what it takes to deploy NoSQL database software.

In a traditional data center, you will need to acquire physical servers, storage disks and software licenses (when they are not open source), power and cooling hardware, real estate space (or co-located space) and administration. To operate and maintain that same NoSQL storage solution, you will have to consider the cost of intra and inter datacenter redundant storage, maintenance of servers and storage arrays, overprovisioning of the procured storage, cost of redundant storage and replacement servers to ensure high-availability, and on-going hardware maintenance of servers, etc. Redundancy on its own typically increases these costs by at least 3x, depending on your redundancy levels. Furthermore, to operate, maintain, and scale that same NoSQL storage solution, you will quickly realize that the most significant cost of owning and managing a scalable NoSQL database solution is related to operating and maintaining the software, along with the hardware and infrastructure needed to support it. As your business grows, you will have to add processes in place so that you can quickly add more storage and compute capacity, and this adds more complexity, which further increases your costs.

Running NoSQL database software in the cloud significantly reduces infrastructure costs. In the cloud, those costs include instance hours, GB-month of storage, I/O requests, and data transfer. As you add more virtual servers and cloud storage to your solution, your costs increase. You will also have to manage the virtual servers and cloud storage yourself. As the use of your database grows, you will incur additional expense as you manage, operate, and scale the NoSQL database software and its infrastructure environment. This cost comes in the form of hours of time from expert data architects who perform complex scaling techniques like sharding and partitioning.

With Amazon DynamoDB, there are no direct acquisition costs of database hardware, and no indirect administration costs of managing and scaling your hardware environment. That's because Amazon DynamoDB isn't database software. It's a **database service** that handles all this heavy-lifting for you. It frees the IT department from the headaches of provisioning hardware and systems software, setting up and configuring a distributed database cluster, and managing

ongoing cluster operations such as patching the OS or NoSQL software. With a few clicks of a mouse in the AWS Management Console, you can create your table and then the Amazon DynamoDB service is ready to accept API requests from your applications. To scale, you do not need to deploy new infrastructure or perform database sharding. You tell the service how many requests it needs to be able to handle per second and it automatically spreads your data across enough hardware to provide consistent performance and to protect against down time.

Scenario

Let us assume that your organization wishes to leverage NoSQL database technologies for a new application - your new upcoming multi-player social game with characters from a future blockbuster movie. Your organization believes it will be a very successful game and realizes that they have multiple NoSQL database options:

1. Open source NoSQL database software hosted on-premises
2. Open source NoSQL database software hosted on Amazon Elastic Compute Cloud (Amazon EC2) with Amazon Elastic Block Storage (Amazon EBS)
3. Amazon DynamoDB (a NoSQL database service)

To get a complete picture of the total cost of ownership, assume three different moments in time with each of the three options above:

	Month 1 (Low)	Month 2 (High)	Month 3 (Medium)
Reads (per second)	50	5000 (peak) 2000 (off-peak)	2000 (peak) 1000 (off-peak)
Writes (per second)	25	5000 (peak) 2000 (off-peak)	2000 (peak) 1000 (off-peak)
Data accumulated (GB)	200	900	1,200

Table 1: Usage Profiles

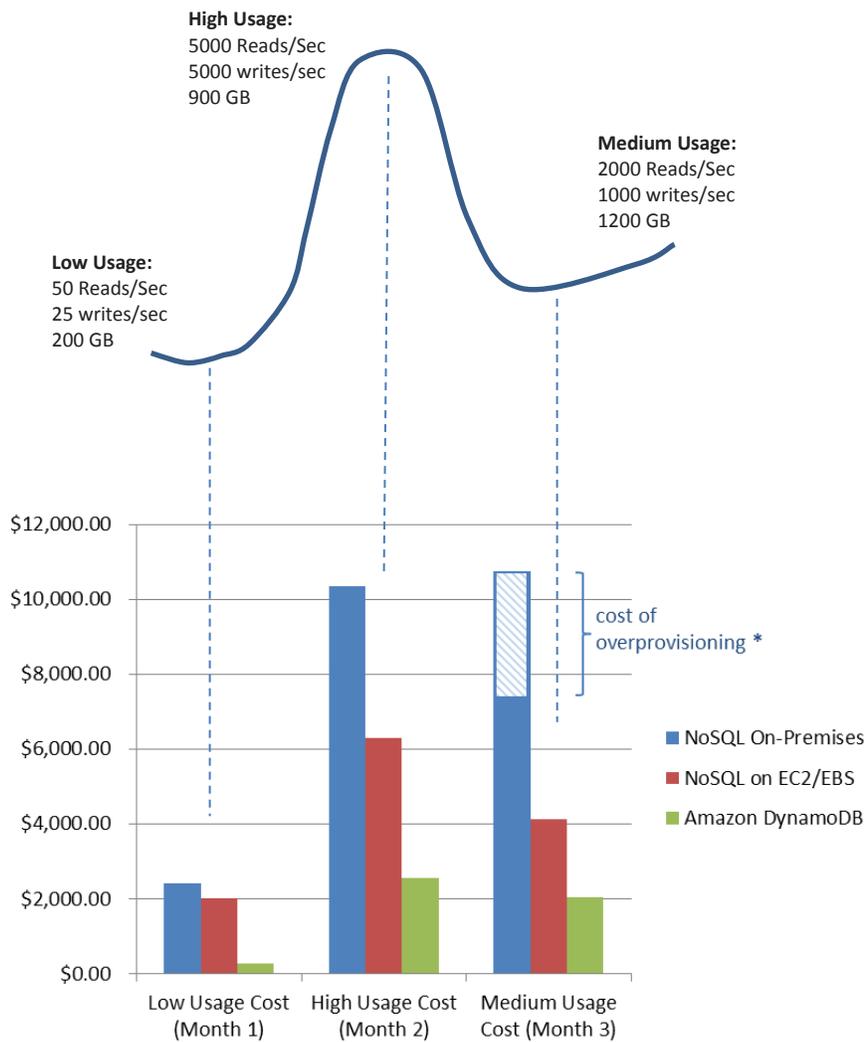
- Month 1: In the first month, since the game was launched with little marketing and the movie was still not released, the game did not require more than 50 reads per second and 25 writes per second. At the end of the month, the game accumulated approximately 200 GB of data.
- Month 2: In the second month, the movie was released and the game gained popularity and experienced a large spike in traffic with thousands of users accessing the game simultaneously. Users were consistently accessing the game at the rate of 5,000 reads and writes per second during peak times and 2,000 reads and writes per second during off-peak times. Data usage increased quickly to 900 GB (Application has more updates and overwrites than new row inserts).
- Month 3: In the third month, the movie buzz faded. As a result, the traffic subsided, and the demand decreased for the game. Reads and writes dropped to 2,000 per second during peak hours and 1000 per second during off-peak hours. At the end of the month, the game accumulated approximately 1,200 GB of data.

For the next several months, the game was experiencing uniform traffic similar to that of Month 3 traffic as it was accessed only by selected frequent visitors (fans). Hence the costs were similar to Month 3 costs.

Summary of TCO Analysis

When calculating for TCO, you should include the costs of servers and network hardware, costs of maintenance, costs of running 3-way replicated storage, costs of power and cooling and data center real estate and at the same time, not forget to include the costs for running redundant hardware and costs of administration (both hardware and database administration).

Since some of the above costs are upfront capital expenditure while others are operating expenditure, in order to simplify the calculations and cost comparison between options, we have amortized the costs over 3 year period for the on-premises option. For the above scenario as described in previous section, the graph shows the cost of running such a solution in each option for each month.



* Cost of overprovisioning (on-premises) is due to the (idle) infrastructure that once purchased cannot be relinquished.

Figure 1: Summary of TCO costs for the scenario

Breakdown of TCO costs – Month 1 (Low Usage)

In the first month, since the game was launched with little marketing and the movie was still not released, the game did not require more than 50 reads per second and 25 writes per second. At the end of the month, the game accumulated approximately 200 GB of data.

TCO – Month 1 (Low Usage)	NoSQL		
	On-Premises	Amazon EC2/EBS	Amazon DynamoDB
Compute Costs			
Server Hardware	\$75.72	\$0.00	\$0.00
Network Hardware	\$15.14	\$0.00	\$0.00
Hardware Maintenance	\$27.26	\$0.00	\$0.00
Power and Cooling	\$25.31	\$0.00	\$0.00
Data Center/Co-located Space	\$22.36	\$0.00	\$0.00
Hardware Administration	\$400.00	\$0.00	\$0.00
Cloud Resources	\$0.00	\$495.00	\$20.50
Total Compute Costs	\$565.79	\$495.00	\$20.50
Redundancy Costs (3x)	\$1,131.58	\$990.00	\$0.00
Storage (3-way replication)	\$300.00	\$95.33	\$219.50
Data Transfer	\$16.00	\$24.00	\$24.00
NoSQL Administration	\$400.00	\$400.00	\$0.00
Total	\$2,413.37	\$2,004.33	\$264.00

Table 2: TCO for Month 1 (Low Usage)

Month 1 Assumptions – Low Usage (200 GB, 50 reads per second, 25 writes per second)

On-premises NoSQL database:

- Compute costs: \$565.79 per server per month

The monthly cost of running one physical server with a high-CPU system configuration amortized over 3 years. This includes the cost of server hardware, network hardware, hardware maintenance, power and cooling and data center real estate. This number was calculated using the [Amazon EC2 Cost Comparison Calculator](#).

This also includes hardware administration costs: \$400 per server per month. The monthly amortized cost of administering 1 physical server assuming that one system administrator can manage 25 servers (based on a people to server ratio of 1:25 and an annual salary + benefits of \$120,000 in the United States. \$120,000 divided by 12 Months divided by 25 Servers = \$400 per server per month).

- Additional Redundancy Costs : \$1131.58 (two times above compute costs)

Assuming 3X redundancy for ensuring high reliability.

- Storage: \$300.00 per month for 300 GB per month at a rate of \$1 per GB per month in storage.

This cost is calculated at 150% of the allocated storage to accommodate growth and to allow time to purchase more hardware before the ceiling is reached. This number was calculated using the On-premise redundant storage cost based on the Forrester Report¹.

- Data Transfer Costs: \$16 per month for 200 GB at a rate of \$25.00 per Megabits per Month (0.6 Avg. Monthly Mbps). This number was calculated using the [Amazon EC2 Cost Comparison Calculator](#).
- NoSQL administration Costs: \$400 per server configuration per month

The monthly amortized cost of NoSQL administration assuming that one NoSQL administrator can manage 25 servers configurations (based on a people to server configuration ratio of 1:25 and an annual salary + benefits of \$120,000 in the United States. \$120,000 divided by 12 Months divided by 25 server configurations = \$400 per server configuration per month). The NoSQL administrator or consultant is assumed to have expertise in one of the following: MongoDB, CouchDB, Voldemort, Cassandra, or Riak, and can install, configure, patch, shard or partition, update, and maintain the server cluster. Note: we assume that NoSQL administrator is managing server configuration as opposed to physical servers.

The total cost of running NoSQL database on-premises for Month 1 is \$2,413.37.

NoSQL database on Amazon EC2 with Amazon EBS:

- Compute Costs: \$495 per instance per month

Instance used is 1 high-CPU Extra Large, On-Demand EC2 Instance (similar in configuration as the on-premises option) running in the US East region at a rate of \$0.68 per hour. The Reserved Instance rate will be much lower. For more information about Reserved Instances, go to <http://aws.amazon.com/ec2/reserved-instances>.

There are no hardware administration costs.

- Additional Redundancy Costs : \$990 (two times above system costs)

Assuming 3X redundancy for ensuring high reliability.

¹ [Forrester Report: "File Storage Costs Less In The Cloud Than In-House" \(August 25, 2011\)](#)

- Storage: \$95.33 per month (\$31.77 per month x 3 servers).

It costs \$24 for 240 GB of Amazon EBS storage at a rate of \$0.10 per GB per month (allocated at 120% of storage) plus \$7.77 for I/O requests for 75 I/O requests per second (200,880,000 I/O requests per month) and assuming 90% cache-hit ratio (leveraging built-in caching NoSQL Software systems)

- NoSQL administration costs: \$400 per server configuration per month

The amortized monthly cost of NoSQL administration assuming that one NoSQL administrator can manage 25 servers configurations (based on a people to server configuration ratio of 1:25 and an annual salary (+ benefits) of \$120,000 in the United States. \$120,000 divided by 12 Months divided by 25 server configurations = \$400 per server configuration per month). The NoSQL administrator or consultant is assumed to have expertise in one of the following: MongoDB, CouchDB, Voldemort, Cassandra, or Riak, and can install, configure, patch, shard or partition, update, and maintain the server cluster. Note: we assume that NoSQL administrator is managing server configuration as opposed to physical servers.

- Data Transfer costs: \$24 per month for 200 GB at a rate of \$0.12 per GB per month

The total cost of running a NoSQL database on Amazon EC2 with Amazon EBS for month 1 is \$2,004.33.

Amazon DynamoDB:

- Provisioned Throughput: \$20.50 for 25 write capacity units and 50 read capacity units and assuming 1 KB item size (Taking the [AWS Free Usage Tier](#) into consideration, 5 writes per second and 10 reads per second are at no charge).

There are no hardware or NoSQL database administration costs.

- Storage: \$219 for 200 GB per month (plus an additional cost of indexed data storage) at a rate of \$1 per GB per month (US East Region).
- Data Transfer: \$24 per month for 200 GB at a rate of \$0.12 GB per month.

The total cost of using Amazon DynamoDB for month 1 is \$264.00.

Breakdown of TCO costs – Month 2 (High Usage)

In the second month, the movie was released and the game gained popularity and experienced a large spike in traffic with thousands of users accessing the game simultaneously. Users were consistently accessing the game at the rate of 5,000 reads and writes per second during peak times and 2,000 reads and writes per second during off-peak times. Data usage increased quickly to 900 GB.

TCO – Month 2 (High Usage)	NoSQL		
	On-Premises	Amazon EC2/EBS	Amazon DynamoDB
Costs Items (Amortized)			
Compute Costs			
Server Hardware	\$378.60	\$0.00	\$0.00
Network Hardware	\$75.70	\$0.00	\$0.00
Hardware Maintenance	\$136.30	\$0.00	\$0.00
Power and Cooling	\$126.55	\$0.00	\$0.00
Data Center/Co-located Space	\$111.80	\$0.00	\$0.00
Hardware Administration	\$2,000.00	\$0.00	\$0.00
Cloud Resources	\$0.00	\$1,368.84	\$1,393.00
Total Compute Costs	\$2,828.95	\$1,368.84	\$1,393.00
Redundancy Costs (3x)	\$5,657.90	\$2,737.68	\$0.00
Storage (3-way replication)	\$1,350.00	\$1,581.77	\$987.89
Data Transfer	\$116.58	\$195.00	\$180.00
NoSQL Administration	\$400.00	\$400.00	\$0.00
Total	\$10,353.43	\$6,283.29	\$2,560.89

Table 2: Total Costs for Month 2 (High Usage)

Month 2 Assumptions – High Usage (900 GB of data, 5000 I/O per second at peak and 2000 I/O per second at off-peak)

On-premises NoSQL database:

- Compute costs: \$2828.95 (\$565.79 per server per month)

The monthly cost of running five physical servers with a high-CPU system configuration amortized. This includes the cost of server hardware, network hardware, power and cooling and data center real estate. This number was calculated using the [Amazon EC2 Cost Comparison Calculator](#).

This includes hardware administration costs: \$2,000 (\$400 per server per month). The monthly amortized cost of administering 5 physical servers assuming that one system administrator can manage 25 servers (based on a people to server ratio of 1:25 and an annual salary (+ benefits) of \$120,000 in the United States).

- Additional Redundancy Costs : \$5657.90 (two times above compute costs)

Assuming 3X redundancy for ensuring high reliability.

- NoSQL administration costs: \$400 per server configuration per month

Same as calculated above - Month 1 (Low usage).

- Storage: \$1350 for 1350 GB per month at the rate of \$1 per GB/month in storage

This cost is calculated at 150% of the allocated storage to accommodate growth and to allow time to purchase more hardware before the ceiling is reached. This number was calculated using the On-premise redundant storage cost based on the Forrester Report².

- Data Transfer: \$116.58 per month for 1500 GB at the rate of \$25.00 per Megabits per Month (4.7 Avg. Monthly Mbps).

The total cost of running NoSQL database On-premises – Month 2: \$10,353.43

NoSQL database on Amazon EC2 with Amazon EBS:

- Instances: \$1,368.84

The instance used is high-CPU extra-large, On-Demand EC2 Instance running in the US East region at a rate of \$0.68 per hour.

Peak workload: 3 instances at 75% utilization

Off-peak workload: 2 instances at 25% utilization

- Storage: \$1581.77 (\$527.26 for 5 volumes per month X 3 times for redundancy)

\$108 for 1080 GB of Amazon EBS at a rate of \$0.10 per GB per month (calculated at 120% allocated storage)

Peak workload: \$359.64 for 5,000 I/O requests per second (3596400000 requests per month)

Off-peak workload: \$59.62.6 for 2,000 I/O requests per second (596160000 requests per month) assuming 90% cache-hit ratio (leveraging built-in caching NoSQL Software systems).

There are no hardware administration costs.

- Additional Redundancy Costs : \$2,737.68 (two times above system costs)

² [Forrester Report: "File Storage Costs Less In The Cloud Than In-House" \(August 25, 2011\)](#)

Assuming 3X redundancy for ensuring high reliability

- Data Transfer: \$195 per month for 1500 GB at the rate of \$0.12 GB/Month + \$15 for 1500 GB at the rate of \$0.01 GB/Month of Regional Data Transfer.
- NoSQL administration Costs: \$400 per server configuration per month

Same as calculated above - Month 1 (Low usage).

The total cost of running a NoSQL database on Amazon EC2 with Amazon EBS - Month 2 is \$6,283.29.

Amazon DynamoDB:

- Provisioned Throughput: \$1,393

Peak Workload: \$1203.96 for 1500 writes/second and 3500 reads/second

Off-Peak Workload: \$189.04 for 800 writes/second and 1200 reads/second
(includes AWS Free Usage Tier)

There are no hardware or NoSQL database administration costs.

- Storage: \$987.89 for 900 GB per month (+ additional cost of indexed data storage) at the rate of \$1 per GB/month.
- Data Transfer: \$180 per month for 1500 GB at the rate of \$0.12 GB/Month.

The total cost of using Amazon DynamoDB - Month 2 is \$2506.89

Breakdown of TCO costs – Month 3 (Medium Usage)

In the third month, the movie buzz faded. As a result, the traffic subsided, and the demand decreased for the game. Reads and writes dropped to 2,000 per second during peak hours and 1000 per second during off-peak hours. At the end of the month, the game accumulated approximately 1,200 GB of data.

TCO – Month 3 (Medium Usage)	NoSQL		
	On-Premises	Amazon EC2/EBS	Amazon DynamoDB
Compute Costs			
Server Hardware	\$378.61	\$0.00	\$0.00
Network Hardware	\$75.72	\$0.00	\$0.00
Hardware Maintenance	\$136.30	\$0.00	\$0.00
Power and Cooling	\$126.54	\$0.00	\$0.00
Data Center/Co-located Space	\$111.80	\$0.00	\$0.00
Hardware Administration	\$2,000.00	\$0.00	\$0.00
Cloud Resources	\$0.00	\$866.32	\$646.36
Total Compute Costs	\$2,828.97	\$866.32	\$646.36
Redundancy Costs (3x)	\$5,657.94	\$1,732.64	\$0.00
Storage (3-way replication)	\$1,800.00	\$1,040.47	\$1,317.19
Data Transfer	\$46.60	\$78.00	\$72.00
NoSQL Administration	\$400.00	\$400.00	\$0.00
Total	\$10,733.51	\$4,117.43	\$2,035.55

Table 3: Total Costs for Month 3 (Medium Usage)

Month 3 Assumptions – Medium Usage (1,200 GB of data, 2000 I/O per second at peak and 1000 I/O per second at off-peak)

On-premises NoSQL database:

- Compute costs: \$2828.97 (\$565.79 per server per month)

The monthly cost of running five physical servers with a high-CPU system configuration amortized. The servers and hardware procured last month cannot be relinquished. Hence although there is no need to have five servers to meet the demand, there are still costs in running them this month (overprovisioned or idle capacity costs).

This includes hardware administration costs : \$2,000 (\$400 per server per month). The monthly amortized cost of administering 5 physical server assuming that one system administrator can manage 25 servers (based on a people to server ratio of 1:25 and an annual salary (+ benefits) of \$120,000 in the United States). Although there is no need for five servers to meet the demand, there are still costs involved in managing them this month because they are still running (administration costs to manage overprovisioned or idle capacity).

- Additional Redundancy Costs : \$5657.94 (two times above compute costs)

Assuming 3X redundancy for ensuring high reliability.

- NoSQL administration: \$400 per server configuration per month

Same as calculated above - Month 1 (Low usage).

- Storage: \$1,800 for 1800 GB per month at the rate of \$1 per GB/month in storage, calculated at 150% allocated storage to accommodate growth, and to allow time to purchase more hardware before the ceiling is reached. (Used On-premise redundant storage cost based on Forrester Report³).
- Data Transfer: \$46.60 per month for 600 GB at the rate of \$25.00 per Megabits per Month (1.9 Avg. Monthly Mbps).

The total cost of running NoSQL Software On-Premise – Month 3: \$ \$10,733.51

NoSQL database on Amazon EC2 with Amazon EBS:

With cloud, you have the ability to get rid of servers and storage when you don't need them and, as a result, there is no overprovisioned capacity.

- Instances: \$866.32

The instance used is high-CPU extra-large, On-Demand EC2 Instance running in the US East region at a rate of \$0.68 per hour.

Peak workload: 2 instances at 75% utilization

Off-peak workload: 1 instance at 25% utilization

- Additional Redundancy Costs : \$1,732.64 (two times above compute costs)

Assuming 3X redundancy for ensuring high reliability.

- Storage: \$1,040.47 (\$346.82 for 3 volumes per month X 3 times for redundancy)

\$144 for 1440 GB Elastic Block Store at the rate of 0.10 /GB per month (calculated at 120% allocated storage)

Peak workload: \$178.85 for 2000 I/O requests per second (1788480000 requests per month) at a rate of \$0.10 per 1,000,000 requests.

Off-peak workload: \$23.98 or 1000 I/O requests per second (239760000 requests per month) at a rate of \$0.10 per 1,000,000 requests.

³ [Forrester Report: "File Storage Costs Less In The Cloud Than In-House" \(August 25, 2011\)](#)

- NoSQL Admin: \$400 per server per month

Same as calculated above - Month 1 (Low usage).

- Data Transfer: \$78 per month for 600 GB at the rate of \$0.12 GB/Month + \$6 for 600 GB at the rate of \$0.01 GB/Month of Regional Data Transfer.

The total cost of running NoSQL software on EC2 with EBS – Month 3: \$4,117.43

Amazon DynamoDB:

- Provisioned Throughput: \$646.36

Peak Workload: \$567.12 for 800 writes/second and 1200 reads/second

Off-Peak Workload: \$79.24 for 300 writes/second and 700 reads/second

(includes AWS Free Usage Tier)

There are no hardware or NoSQL database administration costs.

- Storage: \$1317.19 for 1200 GB per month (+ additional cost of Indexed data) at the rate of \$1 per GB/month.
- Data Transfer: \$42 per month for 600 GB at the rate of \$0.12 GB/Month.

The total costs of using Amazon DynamoDB – Month 3: \$2,035.55

Scenario Summary

Summary of above usage scenario is provided in the table below:

TCO Savings	Option 1: NoSQL Software hosted on-premise	Option 2: NoSQL Software on Amazon EC2 with EBS	Option 3: Amazon DynamoDB
Low Usage Cost (Month 1)	\$2,413.37	\$2,004.33	\$264.00
High Usage Cost (Month 2)	\$10,353.43	\$6,283.29	\$2,560.89
Medium Usage Cost (Month 3)	\$10,733.51	\$4,117.43	\$2,035.55
Total – 3 Months	23,500.31	12,405.05	\$4,860
Savings over option 1	-	47%	79%
Savings over option 2	-	-	61%

Based on the conservative assumptions highlighted above, for this particular scenario (gaming application), you can clearly see that the Total Cost of Ownership of a NoSQL database service such as Amazon DynamoDB is 79% more cost-effective than running and managing an equivalent open source solution on an on-premises infrastructure and 61% more cost-effective than running and managing an equivalent solution on an on-demand cloud infrastructure. For a fair comparison, we have used amortized monthly costs, wherever possible. For example, hardware acquisition costs and administration costs are calculated per month. Typically, these costs are huge upfront costs with long-term contracts.

Other Intangible Costs

1. Lower cost of experimentation and lower barrier to entry. If you are trying to prototype or test your application or use case to perform against a NoSQL Database, you can get started quickly with a NoSQL Database Service such as Amazon DynamoDB since there is no hardware to install and no software to manage. If you decide to move away from the decision, you can do so without any residual charges or contracts.
2. NoSQL database admins are not that easy to find these days. It's not only difficult to hire the right candidate with NoSQL expertise, but it can also be extremely difficult to retain them.
3. Ability to scale up quickly within minutes. You can quickly dial up the scale without any change to your code, if your application is successful. Your customers will continue to have great user experience, irrespective of how many of them are accessing the application at a given time.

Conclusion

Amazon DynamoDB is a fully managed NoSQL database service that provides fast performance with seamless scalability. It frees the IT department from the headaches of provisioning hardware and systems software, setting up and configuring a distributed database cluster, and managing ongoing cluster operations. There are no hardware administration costs since there is no hardware to maintain. There are no NoSQL database administration costs such as patching the OS and managing the NoSQL cluster, since there is no software to maintain. You experience significant cost savings from both the elimination of hardware and software costs and from the elimination of manual database administration efforts. It handles all the complexities of scaling and partitions and re-partitions your data over more machine resources to meet your I/O performance requirements so you can focus on your application instead of worrying about infrastructure.

References

1. AWS Economics Center - <http://aws.amazon.com/economics>
2. Amazon DynamoDB - <http://aws.amazon.com/dynamodb>
3. Forrester Report - File Storage Costs Less in the Cloud than In-House - http://www.forrester.com/rb/Research/file_storage_costs_less_in_cloud_than/g/id/57696/t/2
4. Solid State Drive - http://en.wikipedia.org/wiki/Solid-state_drive
5. Choosing a key-value storage system - http://blog.medallia.com/2010/05/choosing_a_keyvalue_storage_sy.html
6. Amazon EC2 Cost Comparison Calculator - http://media.amazonwebservices.com/Amazon_EC2_Cost_Comparison_Calculator_042810.xls
7. AWS Simple Monthly Calculator - <http://aws.amazon.com/calculator>