Project plan for degree projects

DV2572: Masterarbete i datavetenskap

June 11, 2017

Thesis	Tentative title	PERFORMANCE EVALUATION OF MONGODB ON MICROSOFT AZURE.
	Classification	INFORMATION SYSTEMS~ STORAGE ARCHITECTURE INFORMATION SYSTEMS~ STORAGE MANAGEMENT INFORMATION SYSTEMS~ PARALLEL AND DISTRIBUTED DBMSs
Student 1	Name	NEERAJ REDDY AVUTU
	e-Mail	Neav16@student.bth.se
	Social security nr	9411053375
	Visa expiration date	June 30th 2017
Supervisor	Name and title	Leave blank if no supervisor is assigned yet
	e-Mail	
	Department	
External	Name and title	Leave blank if no external** is assigned
	e-Mail	
	Company/HEI	

[*] *2012 ACM Computing Classification System: www.acm.org/about/class/2012*

[**] *Co-advisor from industry or a higher education institution (HEI).*

1 Introduction

1.1 MongoDB:

MongoDB is a document oriented database where the NoSQL DBMS is developed by 10gen Company. BSON is the data storing format for MongoDB where it is the binary format for JSON. Of the two BSON is a better data type as it also supports Date data type [1]. In MongoDB, a record is called a document and a table is called a collection [2]. It also allows the usage of arrays and objects inside its documents. MongoDB is written in C++. Memory-mapped storage engine is the storage type used in MongoDB which makes the operating system to take

responsibility to flush the data to disk and page in/page out the data [2]. MongoDB supports segregation and it can be configured to run in multiple data centers. A shell is being provided by MongoDB for admin console and graphical programs can also be used for the same which are available as free downloads [2].
MongoDB uses the concept of sharding for scaling out. Sharding is used for storing data among multiple machines. A single machine may be insufficient due to the increase in size of data with good performance. MongoDB uses horizontal scalability.

MongoDB uses an asynchronous master-slave replication model for replication of data through datasets even though traditional model is available which isn't recommended for MongoDB [3]. It provides automatic failover and data redundancy [4].

MongoDB supports rich query language in order to support read and write operations i.e. CRUD (create, read, update, delete). MongoDB provides high performance, high availability and automatic scaling [4].

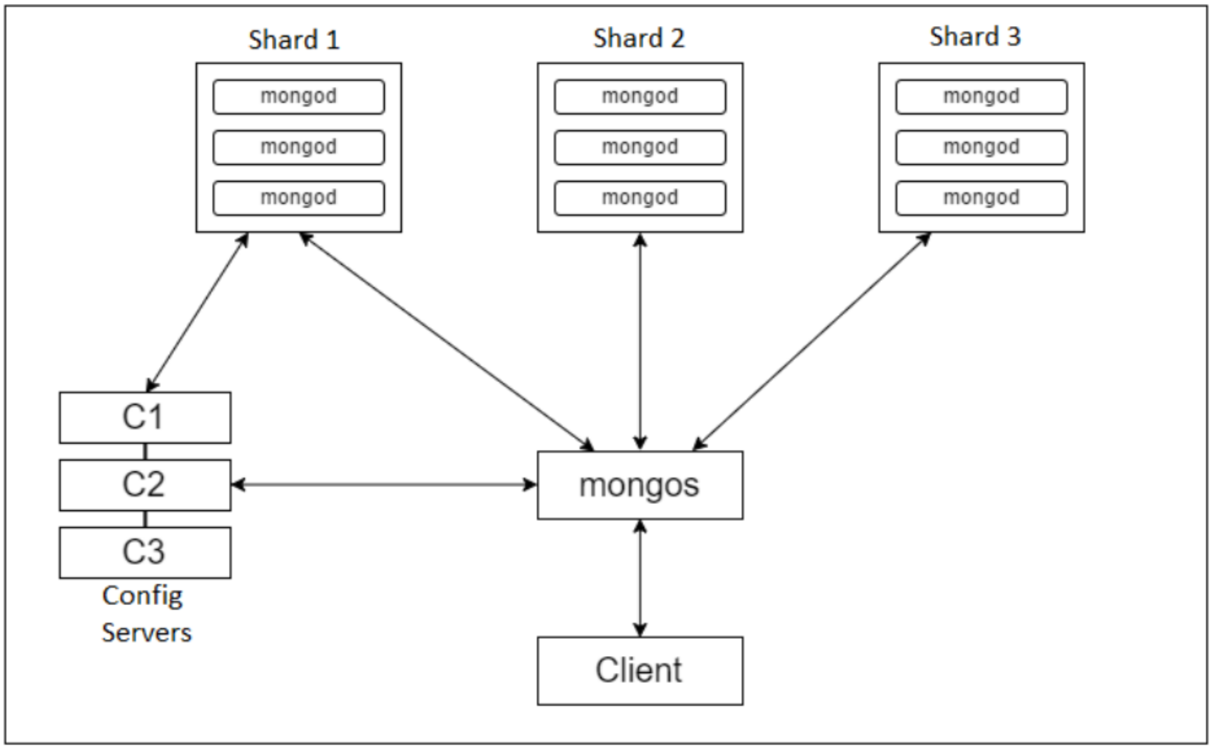

Figure1: MongoDB Architecture [5]

Why Mongo DB?

As MongoDB is a document oriented database, it provides richest query functionality as the ability to query data efficiently is the biggest difference between NoSQL DBMSs. MongoDB also provides tuneable consistency which is defined at query level, this provides flexibility for application and development teams who expect consistent systems [6]. Due to the varying maturity and functionality of APIs across API products, MongoDB's idiomatic drivers will be minimizing the on-board time and simplify the task for new developers [6]. Due to commercial backing of MongoDB it provides extensive documentation. MongoDB also provides the easiest way to run on cloud (MongoDB Management Service). There is Production Support provide by MongoDB engineers where they will be providing support in any aspect of the support [6]. These are the aspects which led into the selection of MongoDB as the No SQL DBMS.

1.2 Microsoft Azure, Azure Virtual Machines (VMs), Azure Container Service and Azure Auto-Scaling

Microsoft Azure provides cloud computing services. It supports Software as a service, Platform as a service and Infrastructure as a service. Microsoft Azure provides Microsoft Azure VMs and Azure Container service where virtual servers allow users in deploying and maintaining OS and server software, container management service which will support Docker containers and allow user for running applications on managing instance cluster respectively [7][8][9]. Azure AutoScaling will be automatically changing the number of instances that provide a particular compute workload where metric and thresholds are defined which will determine adding or removing of the instances [10]. Azure Load balancer provides high availability and network performance where it will distribute the incoming application traffic for scaling out automatically and then handling failover and routing a collection of services [11].

1.3 Introduction to Docker Container

Docker containers are becoming one of the trending applications where Docker is providing more flexibility, scalability and resource more efficient than VMs. In this process, applications and their libraries are bundled in lightweight Linux containers which are offered to public via cloud. It provides an extra layer of abstraction and automation of virtualization [12] [13].

2 Aim and objectives

Aim:

The main aim of this research is to analyse the performance of MongoDB on Microsoft Azure. Specifically, by measuring the latencies of auto-scaling on Azure VMs and Azure Container Service. This comparison shows the performance difference between the usage of the containers and the otherwise case.

Furthermore, there are certain objectives to be set and reached in order to fulfil the aim specified. The following are the objectives of this research.

Objectives:

1. Studying and researching about MongoDB, Virtualization on Azure and Azure Container Services
2. Experimenting so as to identify the latency associated when Mongo DB instances are scaled out on Azure Virtual nodes.
3. Understanding Sharding and Replication Factors in the Mongo DB instances and their effects while scaling out.
4. Repeating the experiment for i) Reliable results ii) To compare the results with the latencies associated with that of the Container Services.
5. Analyzing the data yielded and putting forth the results with an appropriate data analysis method.

3 Research questions

The following are the research questions taken into consideration. These questions are formulated to meet the aim of this research.

1. What is the latency when MongoDB instances are added to Azure nodes and how do we measure this latency?

Rationale: This research question is framed to identify and quantify the latency when the Mongo DB instances are added to the Azure instances while scaling out.

2. What are the effects of Sharding and Replication Factors when Mongo DB instances are scaled out on Azure Virtual Machines and Azure Container Services?

Rationale: The goal of this research question is to understand the impacts of Sharding and Replication Factors during scaling out. Since these factors can be tuned, there is a definite effect on the latency associated. Hence, to address the same, this research question is framed.

3. How is the Azure Container Services more efficient than Azure datacenters when MongoDB is scaled out on each of these platforms?

Rationale: This research question is framed to compare the results of latency with and without containerizing the Mongo instances.

4 Method

Literature review:

For understanding and gaining knowledge on MongoDB, Microsoft AZURE and Docker containers a literature study is done. For successfully performing an experiment, in order to meet all the requirements and designing an experiment extensive literature review is to be done. The official documentations of MongoDB, Microsoft Azure and Docker will be used for literature review. Further, literature which include performance evaluations of the technologies used will also be referred.

Experimentation:

Experiments require proper plan and design which can done through literature review done initially on experiments. Proceedings will take place initially by the installing a tarball file of MongoDB which will be run on Microsoft Azure. Then, scaling out the Mongo DB instances for measuring the latency on Microsoft Azure VMs and Container Services.

Latency is also to be measured by adjusting the replication factors and concurrency levels on each of these platforms to know the effects of the mentioned factors. Furthermore, controlled experimentation is the type of experimentation chosen for this research.

Selection and Rejection Criteria:

Literature review and experimentation are the methods selected for performing this research and answering the research questions. Other research methods like Survey and Case Study are rejected because it is not possible to predict scalability of MongoDB nodes and the latency in adding the MongoDB nodes using survey or case study. This led to the selection of experimentation along with literature review as an apt method for performing this research.

Dependent Variables: The factors affecting the MongoDB performance on Azure VMs and Container Services are considered to be dependent variables. The dependent variables are **availability, efficiency, replication** and the **concurrency** levels that are to be well maintained.

Independent Variables: The independent variables were identified in order to add MongoDB nodes for the performance of Azure VMs and Azure Containers services, where these were identified as **latency** and **horizontal scaling** through the literature review performed.

5 Expected Outcomes

- Gain knowledge and practical hands-on experience with MongoDB and Microsoft Azure.
- Effects of sharding and replication when MongoDB instances are scaled out on Microsoft Azure.
- Measure of latency when the nodes are added to Azure with and without containerization.
- Containerization reduces latency in adding MongoDB nodes.

6 Time and Activity Plan

Start Date	End Date	Activity
17th May 2017	25th May 2017	Literature study
28th May 2017	1st June 2017	Identifying the problem domain and research gap.
3rd June 2017	9th June 2017	Proposal
11th June 2017	Project plan submission	
13th June 2017	20th June 2017	Literature research related to Azure
22nd June 2017	23rd June 2017	Setting up Azure educator account.
24th June 2017	3rd July 2017	Literature research and gaining experience on MongoDB

4th July 2017	6th July 2017	Installation and setup- MongoDB
7th July 2017	11th July 2017	Literature review on how the experiments are to be performed
15th July 2017	30th July 2017	Performing the experiments with replication factors and consistency levels on auto scaling
1st August 2017	8th August 2017	Analyzing the experimental results.
5th August 2017	4th January 2018	Documentation
20th August 2017	20th October 2017	Quantifying the latency
21st October 2017	23rd October 2017	Verification of the results
25th October 2017	20th November 2017	Performing remaining experiments
25th November 2017	27th December 2017	Putting forth a proactive auto scaling mechanism
28th December 2017	4th January 2018	Analyzing the results of the proactive auto scaling mechanism
5th January 2018	13th January 2018	Final documentation of thesis work.
16th January 2018	23th January 2018	Submission of thesis draft
21st January 2018	Submission of opposition report	
22nd to 24th January 2018	Thesis presentation and defense	
4th February 2018	Submit thesis final draft for grading	

7 Risk management

Risk	Impact	Probability	Risk Mitigation Strategy

Time Constraints	Moderate	Moderate	Working according to the plan and meeting the deadlines in each activity
Ample Literature Unavailability	Moderate	low	Different literature on MongoDB and Azure shall be referred.
Lack of knowledge in research area	High	Moderate	Proper Literature Study
Non-productive days due to personal issues	High	Low	Leaving few days aside for personal issue in order to stick with the plan with few altercations

References

[1] Herrnansyah, Y. Ruldeviyani, and R. F. Aji, "Enhancing query performance of library information systems using NoSQL DBMS: Case study on library information systems of Universitas Indonesia," in *2016 International Workshop on Big Data and Information Security (IWBIS)*, 2016, pp. 41–46.

[2] V. Anand and C. M. Rao, "MongoDB and Oracle NoSQL: A technical critique for design decisions," in *2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS)*, 2016, pp. 1–4.

[3] V. Abramova and J. Bernardino, "NoSQL Databases: MongoDB vs Cassandra," in *Proceedings of the International C* Conference on Computer Science and Software Engineering*, New York, NY, USA, 2013, pp. 14–22.

[4] "MongoDB Documentation." [Online]. Available: https://docs.mongodb.com/. [Accessed: 10-Jun-2017].

[5] "The Definitive Guide to NoSQL Databases," *Toptal Engineering Blog*. [Online]. Available: https://www.toptal.com/database/the-definitive-guide-to-nosql-databases. [Accessed: 10-Jun-2017].

[6] "Top 5 Considerations" to learn why MongoDB is more widely used by organizations, than any other NoSQL database. [Online]. Available: https://www.ascent.tech/wp-content/uploads/documents/mongodb/10gen-top-5-nosql-considerations-february-2015.pdf. [Accessed: 11-Jun-2017].

[7] tysonn, "Microsoft Azure Documentation." [Online]. Available: https://docs.microsoft.com/en-us/azure/. [Accessed: 10-Jun-2017].

[8] carolz, “Azure Container Service Documentation - Tutorials, API Reference.” [Online]. Available: https://docs.microsoft.com/en-us/azure/container-service/. [Accessed: 10-Jun-2017].

[9] carolz, “Azure Virtual Machines Documentation - Tutorials, API Reference.” [Online]. Available: https://docs.microsoft.com/en-us/azure/virtual-machines/. [Accessed: 10-Jun-2017].

[10] lbrader, “Azure and AWS services compared - multicloud.” [Online]. Available: https://docs.microsoft.com/en-us/azure/architecture/aws-professional/services. [Accessed: 10-Jun-2017].

[11] carolz, “Azure Load Balancer Documentation - Tutorials, API Reference.” [Online]. Available: https://docs.microsoft.com/en-us/azure/load-balancer/. [Accessed: 10-Jun-2017].

[12] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah, Philippe Merle. Autonomic Vertical Elas- ticity of Docker Containers with ElasticDocker. 10th IEEE International Conference on Cloud Computing, IEEE CLOUD 2017, Jun 2017, Honolulu, Hawaii, United States. Proceedings of the 10th IEEE International Conference on Cloud Computing, IEEE CLOUD 2017.

[13] “Docker Documentation,” *Docker Documentation*, 10-Jun-2017. [Online]. Available: https://docs.docker.com/. [Accessed: 10-Jun-2017].