PROJECT PROPOSAL

"Records web user activities into Cassandra which can scale out by adding more nodes to handle increased volume"

Fall 2013

Submitted To
Professor Samuel A. Ajila

By

Ferhan Jamal (100953487)
Nikhil Nayyar (100941327)

Carleton University

TITLE

Records web user activities into Cassandra which can scale out by adding more nodes to handle increased volume.

INTRODUCTION AND LITERATURE REVIEW

Due to exponential growth of Internet in this era, interactive web applications have changed dramatically over the last 20 years, and so have the needs of managing data of those applications. Three interrelated megatrends- Rapid Big Data Management, Big users and Cloud Computing are driving the adoption of NoSQL technology forward, which is an efficient alternative to relational databases. Today more organizations recognize that operating at a desired scale is better achieved on clusters of standard, commodity servers, and a schema-less data model which is often better for the variety and type of data captured and processed nowadays. [4, 13]

Providing scalability and reliability in this big user's trend is very crucially important these days. Gone are the days when, managing 1000-2000 daily users of an application was sufficient, these days we have to deal and talk about minimum of 70,000 or 1000,000 daily users of an application and if respective databases cannot support or handle many concurrent requests, then it's a big loss of revenue. The large number of users along with the dynamic behavior of the query patterns is driving the need of more reliable and efficient database which should be up for handling concurrent and all time available to proceed with the request. Such database to which we could easily call as *"Always Available Database"*. [2, 13]

As a part of this term paper, we are going to do some research on how Cassandra will scale as the web user activities keeps on increasing and how easy is to scale Cassandra database without having any single point of failure. We will be setting up a few nodes of Cassandra cluster on Amazon EC2 or some other cloud servers and will wire on lot of user traffic against it and will monitor how Cassandra is behaving in terms of throughput, CPU usage and latency. We will also try to identify what data serialization format we should follow while storing the data into Cassandra so that we don't always run out of space in Cassandra which will also be beneficial in terms of read / write performance while scaling the Cassandra database. We will try to measure the scalability mostly by throughput. If Cassandra writes cannot keep up with user traffic, then we may see timeout when writing to Cassandra or resource utilization increases on Cassandra side, then it's time to increase Cassandra capacity.

TERM PAPER GOAL AND OBJECTIVES

As we stated earlier our main goal for this term paper is to identify how easily we can scale Cassandra database under heavy load on the cloud server. We will be hosting few nodes of Cassandra cluster on Amazon EC2 or some other cloud servers and we will build a sample web application or some application that will generate lot of traffic against Cassandra cluster.

Afterwards, we will wire on the traffic to Cassandra database and mostly we will be storing user data in Cassandra database. We will monitor how Cassandra is behaving under various loads. If Cassandra writes cannot keep up with user traffic, then we may see timeout when writing to Cassandra or resource utilization increases on Cassandra side, then it's time to increase the Cassandra capacity.

NoSQL databases are proposed in order to overcome limitation of RDBMS systems, which allows horizontal scaling, effective replication and distribution, dynamically editing, weaker concurrency than relational model [5, 6].

REFERENCES

[1] The Little Engine(s) That Could: Scaling Online Social Networks: Josep M. Pujol, Vijay Erramilli, Member, IEEE, Georgos Siganos, Xiaoyuan Yang, Nikolaos Laoutaris, Member, IEEE, Parminder Chhabra, and Pablo Rodriguez, Member, IEEE.

[2] A request skew aware heterogeneous distributed storage system based on Cassandra : Zhen Ye, Shanping Li

[3] Issues in Handling Complex Data Structures with NoSQL databases : Santo Lombardo, Elisabetta Di Nitto and Danilo Ardagna

[4] A Study Into the Capabilities of NoSQL Databases in Handling a Highly Heterogeneous
Tree: Dileepa Jayathilake, Charith Sooriaarachchi, Thilok Gunawardena, Buddhika Kulasuriya and Thusitha Dayaratne

[5] Cattell, Rick. "Scalable SQL and NoSQL data stores." ACM SIGMOD Record39.4 (2011): 12-27.

[6] Survey on NoSQL Database: Jing Han, Haihong E, Guan Le

[7] Will NoSQL Databases Live Up to Their Promise? : Neal Leavitt

[8] A Novel Solution of Distributed Memory NoSQL Database for Cloud Computing: Han, Jing ; Song, Meina ; Song, Junde

[9] Assessing NoSQL Databases for Telecom Applications : Cruz, F. ; HASlab, Univ. do Minho, Braga, Portugal ; Gomes, P. ; Oliveira, R. ; Pereira, J.

[10] Designing performance monitoring tool for NoSQL Cassandra distributed database: Bagade, P.; Chandra, A.; Dhende, A.B.

[11] The NoSQL Principles and Basic Application of Cassandra Model : Guoxi Wang ; Sch. of Software Eng., Tongji Univ., Shanghai, China ; Jianfeng Tang

[12] Security Issues in NoSQL Databases : Okman, L. Deutsche Telekom Labs., Ben-Gurion Univ., Beer-Sheva, Israel Gal-Oz, N. ; Gonen, Y. ; Gudes, E. ; Abramov, J.

[13] Why NoSQL? [http://www.couchbase.com/why-nosql/nosql-database]

CONCLUSION

In this paper, we will be presenting how easily Cassandra can be scale as the throughput keeps on increasing without impacting the client application directly and what are the things we should avoid while scaling out the Cassandra database. We will also talk about what are the best practices that we should follow while storing the data in Cassandra so that read / write latency are not getting affected. We will be hosting Cassandra cluster on Amazon EC2 or some other cloud cluster.