IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

Performance Benchmarking and Metrics Evaluation of Apache Accumulo

Dr. Vijay G. R[1], Archana P[2], Bhagyalakshmi M[3], Chaitra Reddy R[4]
Associate Professor, Department of Information Science and Engineering[1]
Students, Department of Information Science Engineering[2,3,4]
S. J. C. Institute of Technology, Chickaballapura, Karnataka, India

Abstract: *This paper aims at measuring the performance of Apache Accumulo by using Storage Benchmark Kit (SBK). Apache Accumulo is based on the Google's BigTable design with addition of two unique features. One feature is the iterator framework that embeds user-programmed functionality (server-side programming) into different Log-structured Merge Tree (LSM-tree) stages. The second is the cell-level security that enables fine-grain data access control. The SBK (Storage Benchmark Kit) is an open source software framework for the performance benchmarking of any storage system. We can measure the maximum throughput performance of any storage device/system using SBK. The SBK itself is a very high-performance benchmark tool/framework. It massively writes the data to the storage system and reads the data from the storage system. The SBK supports multi writers and readers and also the End to End latency benchmarking.*

Keywords: Apache Accumulo

I. INTRODUCTION

Accumulo is based on the Google BigTable design but includes several unique features that distinguish it from other, existing BigTable implementations. One feature is the Accumulo iterator framework, which embeds user-programmed functionality into different Log-Structured Merge Tree (LSM-tree) stages. Accumulo also provides cell level security that enables fine-grained data access control. An Accumulo cluster runs on a pool of machines and operates over the Apache Hadoop Distributed File System (HDFS). Data is manipulated in Accumulo using a client application programming interface (API) or is used as input or output for Hadoop MapReduce jobs . An Accumulo cluster consists of a single master server and many tablet servers, write-ahead logger servers, a garbage collector process, monitoring service, and many clients. The Accumulo master server is responsible for managing and coordinating tablet server activities, including responding to failures.

The SBK is an open source software framework for the performance benchmarking of any storage system. If anyone wants to measure the maximum throughput performance of their storage device/system, then SBK is the right software for them. The SBK itself is a very high-performance benchmark tool/framework. It massively writes the data to the storage system and reads the data from the storage system. The SBK supports multi writers and readers and also the End to End latency benchmarking. The SBK is not specific to particular type of storage system, it can be used for performance benchmarking of any storage system, let it be file system, databases , any distributed storage systems or message queues by adding SBK driver which specifies the IO operations of storage system. The latency quartiles and percentiles are calculated for complete data written/read without any sampling; hence the percentiles are 100% accurate. The design principle of SBK is the Performance Benchmarking of 'Any Storage System' with 'Any Type of data payload' and 'Any Time Stamp', because, the SBK is not specific to particular type of storage system, it can be used for performance benchmarking of any storage system.

II. PROBLEM IDENTIFICATION AND DEFINITION

2.1 Identification

In order to perform or to run any task successfully, the most essential thing required is, storage. To evaluate the working conditions of any storage, we measure its performance and benchmark standards. The SBK (Storage Benchmark Kit) is an open source software framework for the performance benchmarking of any storage system. So we are collaborating with SBK and working on Apache Accumulo.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

2.2 Definition

To identify the performance issues and demonstrate the performance of Apache Accumulo using Storage Benchmark Kit(SBK)

2.3 Objectives

- The main objective of this project is performance benchmarking of Apache Accumulo using Storage Benchmark Kit (SBK).
- It aims at measuring the maximum throughput performance of Apache Accumulo using SBK software.
- The performance benchmarking is done by adding SBK driver which specifies the IO operations of storage system.
- SBK delivers the through put and latency values for every specific interval for live performance analysis

III. METHODOLOGY

3.1 SBK Benchmark

The SBK benchmark parses and processes the application/user supplied or command line arguments, configures the multiple writers, readers, and the component "SBK performance processor

Data Type Writers and Readers/Callback (Push) Asynchronous Readers

These components initiate the performance benchmarking of write and read operations. These components implement Burst Mode/Max Throughput Mode to measure the maximum write/read throughput of a storage system, Throughput Mode and Rate Limiter Mode to analyze the latency variations under controlled throughput or rate of events/records and End to End Latency Mode to determine the total time duration between time to write a data record and reading the same data record.

Handler

The Data Type handler defines the type of data and methods/operations to operate on the data. Example data type handlers are Byte Array, Java NIO Byte Buffer, Java String, and Protocol buffers

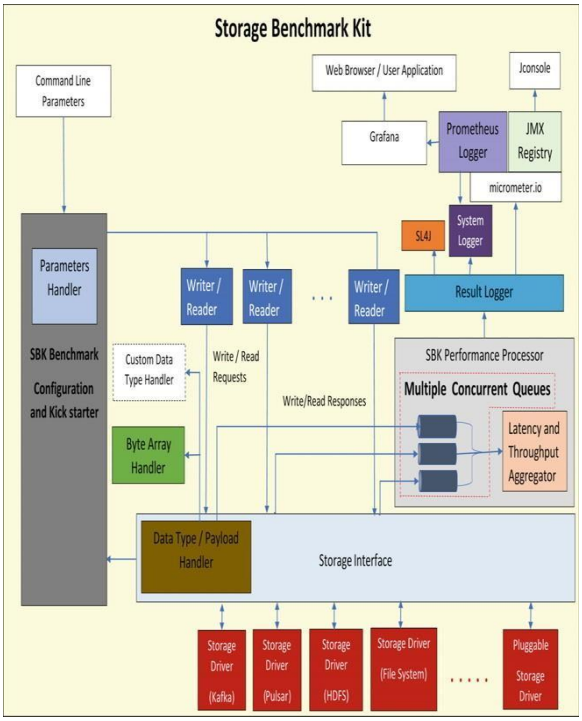

Fig 1: Storage Benchmark Kit

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

Result Logger

This component receives the benchmark results such as throughput values, average, and maximum latency values and latency percentiles for every predetermined time interval from the SBK performance processor component. These benchmark results are logged to a local output device. The SBK uses the micrometer software interface to log the results to the Prometheus monitoring system and SL4J (Simple Logging façade For Java) logging system. The Grafana analytics platform receives these benchmark results from Prometheus monitoring system

Storage Interface and Driver

The SBK defines and implements default methods for the storage interface which are extended and used to implement a custom and pluggable storage driver for any storage device/client. This pluggable storage driver component defines the write and read operations of the storage device/client. A single storage driver component implements a single or multiple instances storage device/client. The pluggable storage driver either chooses one of the available data type handlers or defines a new custom data type handler.

Apache Accumulo

- An Accumulo table is a sparse, multidimensional, sorted map of key–value pairs. The multidimensional keys, as shown in Figure. consist of a row ID, column key, and timestamp.
- Tables are sorted in ascending order by the row ID and column keys and in descending order by the timestamps
- To build an index to the rows based on their values, it is possible to construct an index table with the value as a row ID an the columns as the row IDs of main table

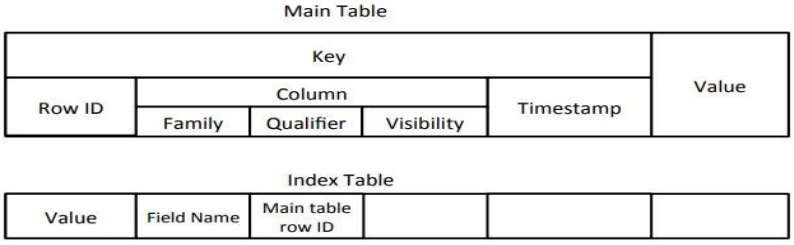

Fig 2: Apache Accumulo table structure

- This schematic diagram shows the relationship between the different storage components.
- Accumulo provides table features such as locality groups, constraints, bloom filters, and iterators. Locality groups enable sets of column families to be stored separately on disk to allow clients to scan over columns that are frequently used together efficiently
- This is similar to the Reduce step in MapReduce, which applies some function to all the values

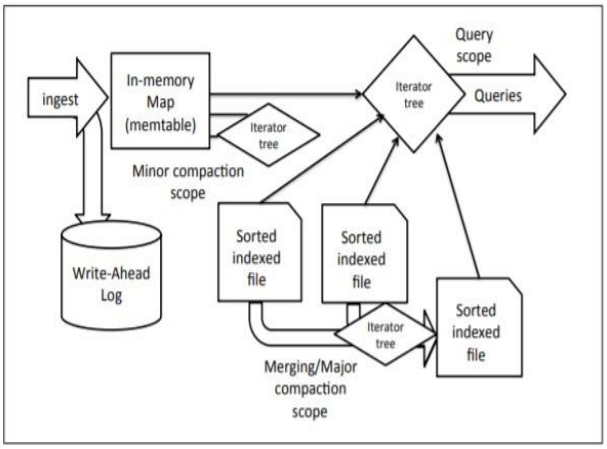

Fig 3: Apache Accumulo Tablet Storage Components

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

IV. IMPLEMENTATION

TO build Storage Benchmark kit,

STEP 1

INSTALL JDK 17+ IN UBUNTU

- java --version # To check the installed java version.
- sudo apt install openjdk-17-jdk openjdk-17-jre # command to install jdk-17.
- Java --version #to confirm the installed java version.

STEP 2:

INSTALL GRADLE 7+ IN UBUNTU

- wget -c https://services.gradle.org/distributions/gradle-7.4.2-bin.zip -P /tmp
  #To download the package of Gradle from its official website in the tmp directory of Ubuntu using the wget command.
- ls /tmp #To list down the downloaded package .
- sudo unzip -d /opt/gradle /tmp/gradle-7.4.2-bin.zip #The downloaded package is zipped, to unzip it using this command.
- ls /opt/gradle #Now again list down the unzipped files by executing the ls command.
- sudo nano /etc/profile.d/gradle.sh #Set up environment variables.
- export GRADLE_HOME=/opt/gradle/gradle-7.4.2
- export PATH=${GRADLE_HOME}/bin:${PATH}
  #In the opened file, add the below-mentioned two lines and then exit the editor by saving the changes made in the file
- sudo chmod +x /etc/profile.d/gradle.sh # Change the permissions of file.
- source /etc/profile.d/gradle.sh # Install the Gradle.
- gradle --version #To confirm the installation, to check the version of installed Gradle.

STEP 3:

INSTALL GIT IN UBUNTU

- git --version #to check the installed version of git.
- sudo apt-get install git #Install the git.
- git --version #to confirm the installed git version.

STEP 4:

BUILDING SBK

- git clone https://github.com/kmgowda/SBK.git
- cd SBK #to check whether the sbk files are listed.
- ./gradlew build # Buliding sbk.
- tar -xvf ./build/distributions/sbk-0.991.tar -C ./build/distributions/.
- #untar the SBK to local folder

V. CONCLUSION

It aims at measuring the maximum throughput performance of Apache Accumulo using SBK software. The performance benchmarking is done by adding SBK driver which specifies the IO operations of storage system. SBK delivers the throughput and latency values for every specific interval for live performance analytics. The SBK benchmark parses and processes the application/user supplied or command line arguments, configures the multiple writers, readers, and the component “SBK performance processor.” The design of SBK exports the standard storage interface APIs which can be extended to include a storage driver to conduct the performance benchmarking of any custom storage device/client. The SBK currently supports benchmarking of a wide category of storage systems such as local mounted file systems,

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Volume 2, Issue 9, June 2022
Impact Factor: 6.252

distributed file systems, streaming storage platforms, key-value storage systems, database systems, and object storage systems.

ACKNOWLEDGEMENT

We would like to express our gratitude to our College, SJC Institute of technology our Guide Dr. Vijay G.R and our project coordinator Aravind Tejas Chandra who have provided us with the opportunity to work on thisproject and given us support with guidance to make this project a success. We would also like to thank our teammates for their contribution andcontinued support and zeal towards this project.This project wouldn't be a success without them.

REFERENCES

[1]. Design And Implementation of Storage Benchmark Kit: K. Munegowda and N. V. Sanjay Kumar, https://github.com/kmgowda/SBK/blob/master/docs/sbk.pdf

[2]. Storage Performance metrics And Benchmarks: Peter M. Chen And David A. Pamrson, FELLOW, IEEE Proceedings of the IEEE, Vol. 81. No. 8, August 1993, https://Ieeexplore.Ieee.Org/Document/236192

[3]. Evaluating Accumulo Performance for a scalable cyber data processing pipeline: Scott M.Sawyer and B.David O'Gwyn https://ieeexplore.ieee.org/document/6597155

[4]. Benchmarking Apache Accumulo BigData Distributed Table store using its continuous test suit: Ranjan Sen,Andrew Farris,Peter Guerra, https://ieeexplore.ieee.org/document/704097