Projects in Big Data Software and Applications

Spring 2017

Bloomington, Indiana

Editor:
Gregor von Laszewski
Department of Intelligent Systems
Engeneering
Indiana University
laszewski@gmail.com

Contents

CDAP Cask Data Application Platform

AVADHOOT AGASTI[1,*,+]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: aagasti@indiana.edu*
[+] *HID - SL-IO-3000*

project-000, February 28, 2017

This paper explains CDAP - Cask Data Application Platform. CDAP provides abstraction layer on top of Apache Hadoop and other Apache Big Data Stack technologies. This paper explains CDAP technology, the kind of problems it can solve, the infrastructure and setup requirements, and its competitive landscape. The paper also provides links to learning material for CDAP.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: CDAP, Hadoop

https://github.com/avadhoot-agasti/sp17-i524/tree/master/paper1/S17-IO-3000/report.pdf

1. INTRODUCTION

CDAP stands for Cask Data Application Platform. CDAP is an application development platform using which developers can build, deploy and monitor applications on Apache Hadoop. In a typical CDAP application, a developer can ingest data, store and manage datasets on Hadoop, perform batch mode data analysis, and develop web services to expose the data. They can also schedule and monitor the execution of the application. This way, CDAP enables the developers to use single platform to develop the end to end application on Apache Hadoop. This paper introduces CDAP as application development platform and explains various use cases that can be solved using CDAP. The paper also explains the CDAP deployment options and infrastructure requirements. Finally we conclude by explaining the other similar platforms and their high level comparison with CDAP. The paper also provides references to the learning material.

2. WHY CDAP

Before we understand how CDAP helps in application development, lets understand how a typical application looks like in Hadoop.

2.1. Typical Application Architecture on Hadoop

Figure 1 shows a typical application architecture on Hadoop.

There are following layers/components -

- Data Ingestion - ingest the data from data source into Hadoop. Data Ingestion tools like Apache Sqoop, Apache Flume, Apache Kafka are popularly used for Data Ingestion
- Data Storage - The data is stored in HDFS.

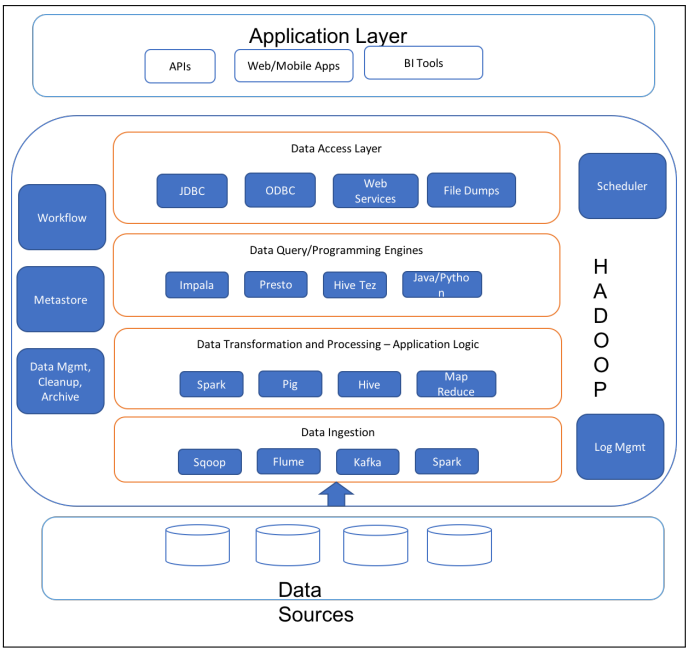


Fig. 1. Typical Application Architecture on Hadoop.

- Data Processing - The data is transformed and aggregated in Data Processing layer. The processing can involved various steps like cleansing, joining, aggregation and running machine learning algorithms. Many different tools and technologies are used to perform data processing operations - e.g. PIG, Hive, Spark are popular open source scripting technologies while Talend, Informatica are visual commercial products.
- Result Storage - The output of data processing step is again stored in HDFS
- Data Access - The end users can access the data (mainly results) using various data access mechanism like APIs, SQL interface or BI tool interface.

2.2. Why CDAP - CDAP Application Architecture

CDAP provides a common application development platform in which a developer can code all the application layers in a typical Hadoop application. CDAP provides abstractions to ingest data, store it in HDFS, process it using the application business logic, store the results in HDFS and expose web service APIs on the result data. User need not use different tools to code different layers. He can simply code all the layers in CDAP platform. He can use same coding language (Java) to do the coding across all the layers.

Further CDAP uses native Hadoop tools for actually performing the operations. For example, the data processing operation implemented in CDAP translate to Spark jobs. Due to this, CDAP users continue to leverage the new enhancements in Apache Hadoop.

3. IMPORTANT CDAP CONCEPTS

CDAP revolves around below important concepts:

- CDAP Datasets provide logical abstraction over the data stored in Hadoop. The data can be files in HDFS or tables in HBase. A dataset needs to be first declared in the CDAP. Any dataset declared in CDAP can be used in any CDAP applications or CDAP services.
- CDAP Applications provide containers to implement application business logic in open source processing frameworks like map reduce, Spark and real time flow. CDAP applications also provide standardize way to deploy and manage the apps
- CDAP Services provide services for application management, metadata management, and streams management

4. CDAP DEPLOYMENT

CDAP provides many deployment options. In standalone mode, it can be downloaded as a zip file and deployed. Alternatively it is available as a standalone virtual machine. For cluster mode deployment, CDAP provides Hadoop-distribution specific options as explained below

- Cloudera Hadoop Distribution (CDH) - Cloudera Manager is tool to deploy CDH cluster. As per CDAP documentation [1] CDAP provides CDAP-parcel which is plug in for Cloudera Manager. Once you add CDAP-parcel to your Cloudera Manager, CDAP can be deployed using Cloudera Manager and all CDAP services can be monitored using Cloudera Manager
- Amazon EMR (Elastic Map Reduce) - EMR is Amazon's Hadoop distribution for the Amazon Web Services cloud. EMR provides 'Create Cluster Wizard' to create EMR cluster. According the CDAP documentation [2], CDAP provides a bootstrap action which is executed when the EMR cluster is created . Using this mechanism, CDAP platform can be deployed on EMR when the EMR cluster is created.

CDAP can also be deployed on HortonWorks Hadoop Distribution, MapR Hadoop Distribution and Apache Hadoop.

5. CDAP INFRASTRUCTURE REQUIREMENTS

CDAP is deployed on edge nodes of the Hadoop cluster. CDAP communicates with Hadoop services like Yarn, HDFS and HBase. Hence CDAP needs to be installed in same network as that of Hadoop. However, none of the CDAP components are required to be installed on Hadoop Namenode or Hadoop datanodes. CDAP documentation [3] provide the CDAP deployment architecture.

6. EDUCATIONAL MATERIAL

- CDPA Applications code repository in Github [4] provide sample applications which are built on top of CDAP Platform.
- CDAP Documentation [5] provides introduction to CDAP platform.

7. REPRESENTATIVE USE CASES WHICH CAN LEVERAGE CDAP

CASK [6] is the company which provides commercial distribution for CDAP. CASK has developed several applications using CDAP. Some of the applications developed using CDAP are explained below

- CASK Hydrator [7] is interactive application for building, running and managing data pipelines for enterprise data lake. CASK Hydrator is UI driven tool using which users can ingest data from sources like traditional RDBMS, trasnform it, aggregate it and finally store the data into permanent storage like HDFS. CASK Hydrator provides UI drag-and-drop style abstraction to all of the above task.
- Customer 360 is another representative application which can be built using CDAP. Customer 360 applications analyzes customer behavior data on various interaction platforms like mobile apps, online communities, customer support portals, and social media. CDAP can be used to ingest the data from these sources and perform join, unification and aggregations to derive 360 degree view of customer.

8. LICENSING

CDAP is licensed [8]under Apache License, Version2.0.

9. OTHER HADOOP APPLICATION DEVELOPMENT PLATFORMS

- Cascading [9] is another application development platform on Apache Hadoop. Cascading has many similar features like CDAP. Cascading supports Java APIs, Data Processing APIs, Data Integration APIs, Scheduler APIs, Relational Operations and scriptable interface. Cascading also support many different Hadoop distributions.

- Talend Big Data Integration [10] : Talend is integration tool using which data can be extracted from source systems, stored on Hadoop and processed in Hadoop. Although Talend is not exactly an application development platform, lot of its features overlap with CDAP. Talend provides visual interface for performing data ingestion and processing operations on Hadoop

10. CONCLUSION

CDAP provides an application development platform over Apache Hadoop. Using CDAP developers can code multiple layers of thier data pipeline in one uniform language and tool. CDAP also can help to shield developers from different Hadoop deployment options like Cloudera, Hortonworks and EMR.

ACKNOWLEDGEMENTS

The authors thank Prof. Gregor von Laszewski for his technical guidance.

REFERENCES

[1] CASK, "Installation using cloudera manager," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://docs.cask.co/cdap/current/en/admin-manual/installation/cloudera.html#admin-installation-cloudera

[2] ——, "Installation on amazon emr using bootstrap actions," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://docs.cask.co/cdap/current/en/admin-manual/installation/emr.html

[3] ——, "System requirements," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://docs.cask.co/cdap/current/en/admin-manual/system-requirements.html

[4] "Cdap applications," Code Repository, May 2015, accessed: 2017-2-18. [Online]. Available: https://github.com/caskdata/cdap-apps

[5] CASK, "Getting started developing with cdap," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://docs.cask.co/cdap/current/en/developers-manual/getting-started/index.html

[6] ——, "Cask - the first unified integration platform for big data," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://cask.co/

[7] ——, "Cask - hydrator," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://cask.co/products/hydrator/

[8] ——, "Cdap product license," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://docs.cask.co/cdap/4.0.0/en/reference-manual/licenses/index.html#cdap-product-license

[9] Cascading, "Cascading," Web Page, online; accessed 18-Feb-2017. [Online]. Available: http://www.cascading.org/projects/cascading/

[10] Talend, "Talend products - big data integration," Web Page, online; accessed 18-Feb-2017. [Online]. Available: https://www.talend.com/products/big-data/

MySQL

CORY COULTER[1,*,+]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding Authors: cacoulte11@gmail.com*
[+] *HID - S17-IO-3004*

project-000, March 9, 2017

This paper covers various aspects relating to the MySQL. Topics covered include a basic overview of the technology, modern use cases in the area of Big Data, infrastructure needs, and further educational resources. © 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Database, Relational Database, Relational Database Management System, Structured Query LanguageL

https://github.com/cacoulte/sp17-i524/blob/master/paper1/S17-IO-3004/report.pdf

INTRODUCTION

MySQL is an "...Open Source SQL database management system [that] is developed, distributed, and supported by Oracle Corporation." [1] MySQL Server provides the means to manage data stored in a relational database. [1] The origins of MySQL can be traced back to 1979 when Micharl Widenius began creating an in-house tool for managing databases. [2]

DATABASE

A database provides a logical way to store information that can later be retrieved. [3] Databases also provide a structure to data that can facilitate the access to stored data.

RELATIONAL DATABASE

Relational databases, which were invented by E.F. Codd in 1970, are a type of database that allows for the access and reassembly of data in a database without the need to entirely reorganize the database. [4] A simple example of the flexibility of a relational database is the ability to add new data categories without having to modify existing applications. [4] The categories in the table correspond to the columns in the table while each row of a table defines a separate entry into that table. [5] In more abstract terms, the columns of a table are referred to as attributes and the rows or entries in a table are referred to as tuples. This framework provides the structure by which data can be logically organized, thus allowing for easy access and analysis of the data. The data in the various tables of the database can be analyzed and filtered based on various sets of criteria and the results presented in a view to an end user. [4]

A simple top down overview of the structure of, say, a database managed with MySQL is database>table>entry. A database consists of a table or tables. Every table contains columns that attempt to define the type of data that is to be stored in the column. Tables are in turn made up of rows which correspond to separate entries in the table. A potentially useful analogy is to think of a database as a warehouse. The warehouse can be partitioned into different sections. The different sections of the warehouse are like the tables in a database. Each section of the warehouse is responsible for storing a particular type of container. The different containers in the sections of the warehouse are analogous to the individual entries in a table. These containers are themselves partitioned further so that corresponding sections of different containers contain the same type or category of stuff (data). The various sections within the containers are similar to the different columns in a table. While the utility of this analogy may be limited, it may prove useful in providing a basic idea of the different parts that allow the structured storage of data in a database.

RDBMS

Relational Database Management Systems (RDBMS) are programs that allow for the creation, updating and administration of relational databases. [4] Many commercial RDBMS softwares use Structured Query Language to access and interact with the relational databases they manage. [6] SQL is a standardized programming language used to perform tasks in relational databases such as creating databases and tables, adding and deleting entries, and querying databases and tables. [7] Databases that utilize the SQL language are colloquially referred to as SQL databases. [7] According to [7], "an official SQL standard was adopted by the American National Standards Institute (ANSI) in 1986 and then by the International Organization for Standardization, known as ISO, in 1987." Although a standard SQL syntax is defined, many RDBMSs have added their own extensions to the language which, in many cases, can only be used with their own systems. [8]

MYSQL USE CASES

MySQL has been used by many different organizations to solve many different problems. [9]. The number of organizations that are listed in [9], and the wide array of problems that they use MySQL to solve, demonstrates the flexibility and utility of relational databases managed by MySQL. Some of the applications that MySQL has been applied to range from science to marketing to health-care. [9] These applications have a wide ranging impact on those that utilize their services.

CERN

MySQL has been employed at the European Organization for Nuclear Research (CERN). [10] Scientists at CERN use sensitive and complex instruments to study the fundamental building blocks and origin of our universe. [10] The IT department at CERN has offered a MySQL Database-as-a-Service to its employees and the scientists associated with its various projects to aid in the management of the data it collects. [10] The management of the database itself is handled by CERN's IT staff but is made available to those with responsibilities and interests in analyzing the data collected. [11]

A specific project at CERN that utilizes MySQL is the ATLAS project. [11] ATLAS uses MySQL to manage its Primary Numbers database. [11] These Primary Numbers are parameters that describe the "detector geometry and digitization in simulations, as well as certain reconstruction parameters, including the identifier maps of the detector elements." [11] An example of the type of data stored in the Primary Numbers database is are maps of magnetic fields associated with instrumentation used in the experiment. [11] One of the specific benefits that MySQL offers to the group at CERN, with relation to managing the Primary Numbers database with MySQL, is that the database can be used by a researcher although they are not currently connected to the central database. [11] Once they are connected to the database again, the database will be updated. [11] Other features that have been found useful include the transfer of data in binary form and MySQL's certificate authorization technology. [11]

boo-box

boo-box is an advertising network that primarily focuses on the South American market. [12] boo-box facilitates the coupling of marketing platforms with marketers to display approximately one billion advertisements per month. [12] boo-box connects marketers with marketing space on "web sites, blogs and social media properties." [12] MySQL is used by boo-box to log user activity such as what pages they view and their click-through rates. [12] Analyzing this data helps boo-box to accurately report and direct ads in marketing campaigns. [12]

MySQL has offered boo-box low latency query performance which allows them to place targeted advertisements in under 250 milliseconds. [12] Ruby and Python interact with boo-box's MySQL database to offer other services in their advertising platform. [12] boo-box is able to capture 2 TB of web logs and process 22 billion rows with MySQL. [12] boo-box is able to use MySQL to store over 8 TB of data and manage 1 TB in a Statistics database. [12]

Health-care

In the health-care industry, MySQL can be used to manage data related to scheduling, billing, prescriptions, and Electronic Medical Records. [13] A couple of the benefits of using MySQL in a health-care application include scalability and security options (with MySQL Enterprise Audit) allowing for compliance with applicable regulations. [13] Waiting Room Solutions (WRS) is a web-based service for physicians' offices that uses MySQL. [13] They offer solutions for the management of "electronic medical and health records, billing, scheduling, electronic prescriptions, [and] online patient registration" among many other services. [13]

EDUCATION

MySQL has been available for download for free since 2000. [14] As such, it should come as no surprise that there are plenty of educational resources available free of charge to those who wish to learn how to use MySQL. [15] provides a list of 50 different sources that offer lessons about various aspects of using MySQL. w3shools.com offers free interactive lessons that cover the basics of the SQL language, which of course is an essential skill if one would like to work with and understand MySQL. [16] An educational resource that focuses more on MySQL can be found at www.mysqltutorial.org. [15] The MySQL Tutorial website offers many lessons in using MySQL; from a Basic MySQL Tutorial to MySQL Administration to programming interfaces such as a Python MySQL Tutorial, this site is a near exhaustive resource to learn all things related to MySQL. [17]

ALTERNATIVES

MySQL is not the only RDBMS. Different RDBMSs have advantages and disadvantages in any given situation/application. [18] Two other RDBMSs that are worth comparing to MySQL are SQLite and PostgreSQL. [18] These three RDBMSs provide end users with a wide array of options, each possibly more suitable than the next in different situations.

SQLite

SQLite is an embedded RDBMS. [18] It is a file-based database that contains a many tools that enable it to deal with a wide array of data types. [18] SQLite is faster than server relational databases. [18] One of the main disadvantages of SQLite is that it doesn't offer user management. [18] Depending on the intended use, this can be a real concern if the application requires multiple users to access the database. [18]

PostgreSQL

PostgreSQL is another open-source RDBMS that is compliant with applicable SQL standards. [18] PostgreSQL is extensible, meaning that functions can be stored as procedures. [18] In addition to being a a relational database management system, PostreSQL is also objective and thus supports nesting. [18] One disadvantage of PostgreSQL is that it does not offer fast read operations. [18]

REFERENCES

[1] MySQL, "Mysql 5.7 reference manual, section 1.3.1 what is mysql?" Website. [Online]. Available: https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html

[2] G. Reese, R. J. Yarger, T. King, and H. E. Williams, *Managing and Using MySQL*, 2nd ed. O'Reilly, 2002, ch. 1. [Online]. Available: http://docstore.mik.ua/orelly/weblinux2/mysql/ch01_01.htm

[3] S. Suehring, *MySQL Bible*. Wiley Publishing, Inc., 2002, ch. 1, pp. 3–8.

[4] M. Rouse, "Relational database," Website, 2006. [Online]. Available: http://searchsqlserver.techtarget.com/definition/relational-database

[5] ibm, “Tables, rows, and columns,” Website. [Online]. Available: http://www.ibm.com/support/knowledgecenter/SSPK3V_6.3.0/com.ibm.swg.im.soliddb.sql.doc/doc/relational.databases.html
[6] M. Rouse, “Relational database mnagement system (rdbms),” Website, November 2005. [Online]. Available: http://searchsqlserver.techtarget.com/definition/relational-database-management-system
[7] ——, “Sql (structured query language),” Website, September 2016. [Online]. Available: http://searchsqlserver.techtarget.com/definition/SQL
[8] “What is sql?” Website. [Online]. Available: http://www.sqlcourse.com/intro.html
[9] MySQL, “Case studies,” Website. [Online]. Available: https://www.mysql.com/why-mysql/case-studies/
[10] B. Matthelie, “With its mysql database-as-a-service cern empowers scientists,” Website, October 2012. [Online]. Available: https://blogs.oracle.com/MySQL/entry/with_its_mysql_database_as
[11] A. Vaniachine, S. Eckmann, D. Malon, P. Nevski, and T. J. Wenaus, “Primary numbers database for ATLAS detector description parameters,” *CoRR*, vol. cs.DB/0306103, 2003. [Online]. Available: http://arxiv.org/abs/cs.DB/0306103
[12] “boo-box serves 1 billion advertisements per month with mysql and hadoop,” Website. [Online]. Available: https://www.mysql.com/why-mysql/case-studies/1billion-advertisements-mysql-hadoop.html
[13] “Mysql in healthcare,” Website. [Online]. Available: https://www.mysql.com/industry/healthcare/
[14] E. Rieuf, “History of mysql,” Website, December 2016, blogpost. [Online]. Available: http://www.datasciencecentral.com/profiles/blogs/history-of-mysql
[15] Code Conquest, “The 50 best websites to learn mysql,” Website, August 2015. [Online]. Available: http://www.codeconquest.com/blog/top-50-websites-to-learn-mysql/
[16] w3schools.com, “Introduction to sql,” Website. [Online]. Available: https://www.w3schools.com/sql/sql_intro.asp
[17] MySQL Tutorial, “Mysql tutorial,” Website. [Online]. Available: http://www.mysqltutorial.org/
[18] O. Tezer, “Sqlite vs mysql vs postgresql: A comparison of relational database management systems,” Website, February 2014. [Online]. Available: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems

RabbitMQ

ABHISHEK GUPTA[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: abhigupt@iu.edu*

project-001, February 27, 2017

RabbitMQ provides simple yet powerful messaging platform which allows applications pass messages in a reliable and fault tolerant way. RabbitMQ implements Advanced Message Queuing Protocol (AMQP) and is written in Erlang programming language. It runs on all major operating systems and supports SDK in all major programming languages[1] including objective-C, swift and node.js. When we look for messaging, we look for certain features: asynchronous messaging, large scale, reliability, clustering, multi-protocol, highly available, fault tolerant. RabbitMQ[2] fulfills these requirements and provides a distributed, persistent, highly available, fault tolerant messaging system which can scale as data grows.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524

https://github.com/cloudmesh/sp17-i524/blob/master/paper1/S17-IO-3005/report.pdf

1. INTRODUCTION

RabbitMQ based[2] off AMQP (Advanced Message Queuing Protocol[3]) Which defines how client applications connect to message brokers. Message brokers receive messages from producers and make it available to consumers. The producer posts messages to exchanges and broker agent reads these messages from exchange and writes to queues. Consumers can further read messages from the queue. It also supports a notion of acknowledgements where consumers can post an acknowledgement back to broker and which in turn can post messages back to the producer.

Looking back at the history[4] of development of rabbitMQ, it started with IBM MQseries in 1993, 1997 Microsoft MQ, 2001 java messaging service. It was in 2003 where JPMorgan created the first version AMQP which became the base for RabbitMQ technologies formed in 2006.

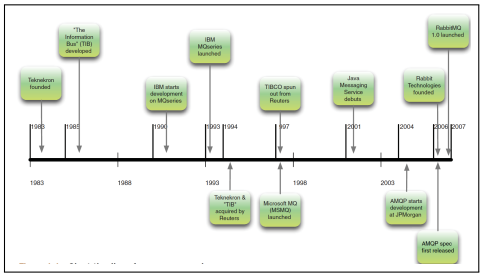


Fig. 1. Timeline for evolution of RabbitMQ [4]

2. ARCHITECTURE

[4]At high level, producer publishes the message to the broker. Consumer consumes or subscribes the messages from the broker. Broker is the middle man who has the knowledge of exchanges and queues. It maintains the bindings between exchanges and queues. The consumers read the messages from the queue. Following table shows different exchanges supported by RabbitMQ and corresponding default exchanges[2] .

Name	Default pre-declared names
Direct exchange	(Empty string) and amq.direct
Fanout exchange	amq.fanout
Topic exchange	amq.topic
Headers exchange	amq.match (and amq.headers in RabbitMQ)

3. TYPE OF EXCHANGES

The sections below explains various types of exchanges supported by RabbitMQ.

3.1. Default Exchange

It is created by the broker and all queues are bound to default exchange unless specified separately. For example we have a queue called demoqueue, all messages assigned to demoqueue will be routed by default exchange.

3.2. Direct exchange

It is used to route messages to queue with a given routing key. For example a message queue has routing key K. A message with same routing key K will be delivered to same message queue.

3.3. Fanout exchange

Delivers messages to all queues that are bound to a given exchange ignoring the routing key. For example if there are 5 queues bound to a exchange E. Messages to exchange E are delivered to all 5 queues.

3.4. Topic exchange

It delivers the messages to a given queue, not just based on the routing key but a pattern specified in the message called topic. It can be useful in scenario when consumer/application decides to which topic(s) it is interested in. Further, it can subscribe to those topics(or messages).

3.5. Header exchange

This exchange ignores the routing key but uses the header parameter to decide which messages goes to which queue. A message is matched when a value in header matches with the one specified in the queue binding. Exchanges have other important attributes apart from routing key and exchange type for example name, durability, auto-delete, arguments etc. Durability allow messages to persist on disk in case of broker restarts. Auto-delete deletes the exchange, once all queue have finished using the exchange.

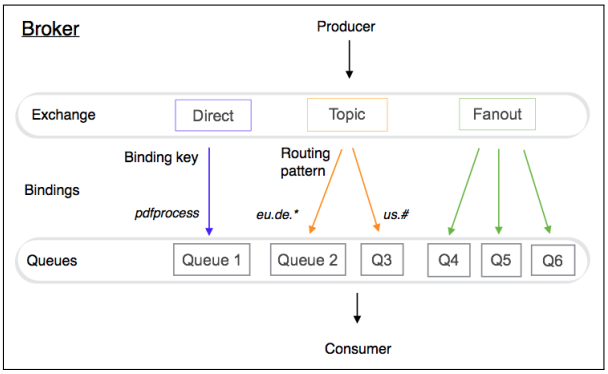


Fig. 2. Different type of exchanges supported by RabbitMQ [5]

4. BINDINGS

Bindings are the rules that define how exchanges will route messages to the queue. To route all messages from exchange E to queue Q, Q has to be bound to Exchange E. Bindings use routing key as one of the criteria to route messages to a queue. However, routing key is optional and not always applicable.

5. CONSUMERS

Messages from message queue are eventually used or consumed by consumers. Consumers can use push or pull mechanism to consume these messages. Push API have messages delivered to the consumer whereas a Pull API is used to fetch messages from the queue.

6. MESSAGE ACKNOWLEDGEMENT

It has a built-in mechanism to send and receive acknowledgements. Producer can send messages and wait to acknowledgement in the response queue from consumer. Consumer can receive messages and post acknowledgement to response queue. Here request queue and request exchange can be used to send and receive messages. Whereas response queue and response exchange can be used to send and receive acknowledgements. This mechanism makes RabbitMQ robust in case of failures.

7. CLUSTERING

[4]To achieve high availability and making sure producers and consumers send and receive data without knowing about node failures, clustering was introduced. RabbitMQ follows OTP(open telecom platform) framework provided by erlang to achieve high availability. RabbitMQ by default doesn't replicate the content of queues i.e. all queues are stored on one node in the cluster. To achieve clustering rabbitmq keeps track of metadata for queue, exchange, binding and vhost. In case of cluster, it only stores all information about the queue like metadata, state and contents on one node rather than all nodes in the cluster. However, it stores metadata and pointer to actual data on each node in the cluster. This is to optimize storage space and performance.

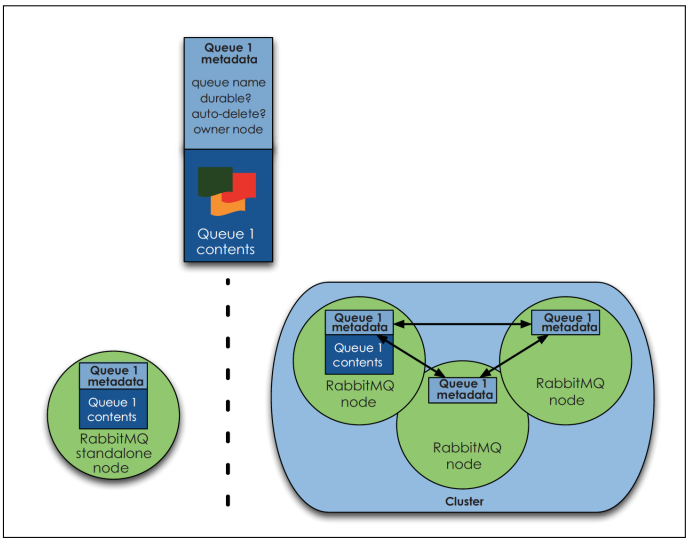


Fig. 3. Shows queue metadata for a cluster vs single node [4]

On the other hand exchanges are just bindings to the queues. So when you send a message to rabbitmq exchange, the channel checks the routing key of the message and compares it against the queue bindings. Further it sends the message to appropriate queue. Since, exchanges are just lookup tables for queue bindings, they are replicated across all nodes on the cluster.

8. MANAGEMENT

RebbitMQ provides all management using:

- web ui
- REST interface
- Rabbitmqctl command line utility

Web UI can be used by administrators to create user, monitor queues and exchanges, view statistics, add configurations etc. Similar functionality can be achieved by cli utility rabbitmqctl and REST API. Rabbitmqctl can be used to automate rabbitmq deployments and management. IT can also be used to write automated tests. REST Api can used for integrating with 3rd part UI and plugins. Using REST api you can monitor the number of connections, download or upload a configuration, list the nodes in the cluster, create or delete rabbitmq users, view or create virtual host, set permission for a user etc. You can also list all current APIs using following url: http://localhost:55672/api with some api documentation and explanations.

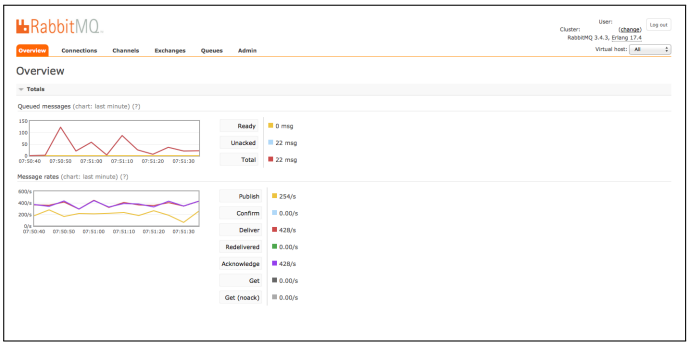


Fig. 4. RabbitMQ management UI showing messages queued and message data rates

[5]

9. LICENSING

RabbitMQ is licensed under Mozilla Public License(MPL) and GPL v2. [2]

10. USE CASES

RabbitMQ messaging can be useful in applications which require asynchronous messaging for example an application initiates a task by posting a message to RabbitMQ, it doesn't have to wait for the task to get completed. Rather it can periodically check the status of task. It can be useful to scale up as the data volume grows by adding additional nodes to RabbitMQ cluster. With distributed applications where applications run as micro-services in a container or virtual machine, RabbitMQ is useful to communicate and share data between different services. RabbitMQ can be managed separately with its management UI, CLI tool and REST api interface. This decouples the messaging layer from application and makes the overall design very robust.

11. CONCLUSION

RabbitMQ is an open source platform and provides a robust messaging platform for applications. It provides simple manageability decoupling with the application. RabbitMQ can scale well when application demand increases and can handle more data by adding more nodes to the cluster. Based on test [6]conducted on RabbitMQ with set of 4K(4096) and 16K(16384) messages, the performance of RabbitMQ on multi node cluster decreases in comparison with single node cluster to reach a threshold and then becomes stable. This decrease in performance can be primarily accounted due to replication between nodes in the cluster.These tests were conducted on combination of single publisher and single subscriber, multiple publishers and single subscriber, multiple publishers and multiple subscribers but there is no concrete conclusion to these test and numbers.

ACKNOWLEDGEMENTS

Special thanks to Professor Gregor von Laszewski, Dimitar Nikolov and all associate instructors for all help and guidance related to latex and bibtex, scripts for building the project, quick and timely resolution to any technical issues faced. The paper is written during the course I524: Big Data and Open Source Software Projects, Spring 2017 at Indiana University Bloomington.

REFERENCES

[1] Pivotal, "RabbitMQ, clients and developer tools," Web Page, accessed: 2017-02-26. [Online]. Available: https://www.rabbitmq.com/devtools.html

[2] ——, "RabbitMQ, components," Web Page, accessed: 2017-02-02. [Online]. Available: https://www.rabbitmq.com/

[3] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Computing*, vol. 10, no. 6, 2006. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4012603

[4] A. Videla and J. J. Williams, *RabbitMQ in action*. Manning, 2012.

[5] Lovisa Johansson, "Rabbitmq for beginners - what is rabbitmq?" WEb Page, May 2015, accessed: 02-15-2017. [Online]. Available: https://www.cloudamqp.com/blog/2015-05-18-part1-rabbitmq-for-beginners-what-is-rabbitmq.html

[6] B. Jones, S. Luxenberg, D. McGrath, P. Trampert, and J. Weldon, "Rabbitmq performance and scalability analysis," *project on CS*, vol. 4284, 2011. [Online]. Available: https://people.cs.vt.edu/butta/cs4284/spring2011/butta/RabbitMQPaper.pdf

Docker Container

VISHWANATH KODRE[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: vkodre@iu.edu

paper1, March 3, 2017

A portable lightweight packing and run time tool a platform for developers and administrators to build, ship, and run distributed application with Docker Engine. Docker can get code tested and deployed faster.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Docker Container, Docker Machine, Docker Swarm, I524

https://github.com/cloudmesh/sp17-i524/tree/master/paper1/S17-IO-3008/report.pdf

WHAT IS DOCKER

Dockers package the application into filesystem that contains software needed to run, runtime, system tools, system libraries. Anything that can be installed on server. These units are known as Docker containers which are also known as lightweight VMs, Docker containers are application images hosted on single machine. However Docker containers differs from the VM with its architecture. It runs on single machine shares the same Kernel. The images are constructed from layered filesystem and share common files, making disk usage and image download much more efficient. Virtual Machines also include application, binaries, libraries and entire guest operating system. "The ability to create multiple lightweight, self-contained execution environments on the same Linux host simplifies application deployment and management. By improving collaboration between developers and system administrators, container technology encourages a DevOps culture of continuous deployment and hyperscale, which is essential to meet current user demands for mobility, application availability, and performance."[1].

However Docker differs from conventional container based echo system. Docker enhances the container technology with its open source platform it makes it more accessible by creating simpler and more powerful tools. With the help of Docker the lifecycle of tens of thousands of containers can easily be managed.

DOCKER APPLICATION

Docker Machine controls the remote Docker-Engines as if they were locally installed. With Docker Swarm installing thousands of Docker Machine becomes the job running single commands. In practice many cloud platform providers are adapting to Docker containers, Docker Machine and Docker Swarm. Such Amazon AWS or Microsoft Azure working in collaborative form if the Docker host is setup with help of these user can access host remotely and one don't have to establish SSH connection to work with specific Docker Engine.

The highly scalable Amazon EC2 Container Service (ECS), supports Docker containers that allows run applications easily on cluster of Amazon EC2 instances with high performance container management service. "Amazon ECS eliminates the need for you to install, operate, and scale your own cluster management infrastructure. With simple API calls, you can launch and stop Docker-enabled applications, query the complete state of your cluster, and access many familiar features like security groups, Elastic Load Balancing, EBS volumes, and IAM roles. You can use Amazon ECS to schedule the placement of containers across your cluster based on your resource needs and availability requirements. You can also integrate your own scheduler or third-party schedulers to meet business or application specific requirements."[2]

INFRASTRUCTURE

Docker don't require any specific infrastructure to setup, unlike VM there is no requirement for guest OS or hypervisor require for container. Docker Container contains only necessary what is to build and run application. With this organization to reduce the cost over infrastructure as cost over storage and licensing of hypervisor gets reduced.

Docker container are portable which allows enterprises to leverage Docker containers across any infrastructure which allows IT operation teams to move workloads across different cloud service, physical servers or virtual Machines which don't demand for specific infrastructure. This way enterprises can optimize their infrastructure and reduce the maintenance cost.

DOCKER AND BIG DATA

For faster and easily available compute instance Docker plays vital role, by collaborating with OpenStack, Apache Spark. Docker Containers helps getting these environment ready running the instances in no time and don't consumer much resources.

Project is one of the example of which is community driven project represents the management of shared files as core service

to OpenStack. With use of Docker container Project Manila get flexibility to share the compute instances among the clusters which is as easy as copying the image. With Manila shared files user can create new instances (with configuration) which are not in use and publish with VM and Docker Container. Usage here is deploying Manila services into OpenStack container, deploying Big Data clusters/services using HEAT into containers. As the Big data processing requires shared file system which Manila provides.[3]

Talend Big Data Sandbox is another case which allows user to experiment and test real word Big Data scenarios. Which provides facility to users to work with Apache Spark and Hadoop distribution. With help of Docker containers SandBox provider free preconfigured, easy to use virtual environment. With SandBox user can perform real time analytics of data from multiple streaming sources. Personal recommendation, Visualization with heat map, monitoring of IT operations using Apache weblogs.[4]

Docker Inc is key player in provide Docker Container, and are in collaboration providing services along with Amazon EC2 Container Service, IBM Bluemix Container Service, Joyent Smart Data Center and MicroSoft Azure to integrate their offerings with Docker Swarm.

By providing online learning material in terms of tutorials, white papers, Admin guide, video tutorials and blogs and forums to increase the awareness and its usage to enterprises. By leveraging these learning and support enterprises are also adapting Docker and provide their services packaged with Docker Container.

Popularity of Docker container is increasing now a days and many enterprises are adapting to Docker container, Docker Machine and Docker Swarm. Few are case studies shows the adaptation Docker in running their business day to day. Such as SA Home Loan Adopts Microservices and deploys 20-30 times a day with Docker Datacenter.

As SA home loan started with using support and services as RabitMQ and nginx, and move all of their main application services over to Container. Which facilitate immutable, transferable development platform and deployment pipline. Also production ready Orchestration service that gives single point from which to manage and distribute containers on nodes.

"SA Home Loans now uses Docker Datacenter, the on-premises solution that brings container management and deployment services to the enterprise via a supported Container-as-a-Service platform that is hosted locally." [5]

SIMILAR TECHNOLOGY

As Docker gains it popularity and dominating the Container-as-a-Service market there are other similar container based technologies and providers are emerging over the period of time such as

1. Open Container Initiative (OCI) It is Open System Interconnection initiative similar to OSI layer network communication and integration model.

2. Kubernetes "One of the primary advantages of the system is that it provides a consistent object model and API for many underlying resources that vary between cloud providers, and has modules allowing it to run on most of the major ones including Amazon Web Services and, of course, Google Cloud."

3. CoreOS and rkt Simple command-line tool "supports multiple container formats, including Docker, and 'pluggable' levels of runtime container isolation, which is useful for certain kinds of system and server applications"

4. Apache Mesos and Mesosphere "Mesos is a cluster management system and control plane for efficient allocation of computing resources between application delivery platforms, called frameworks that are layered above it"

"Mesosphere is an enterprise software OEM that sells a 'data center operating system' also built on Mesos and providing cluster management, container orchestration, service discovery and build automation for elastic computing."

5. Canonical and LXD "LXD builds on the capabilities of LXC by adding to it a systemwide daemon with an API for LXC container management and an OpenStack Nova plug-in for managing virtual LXD hosts in the cloud" [6]

Though Docker has gain it popularity these are few technologies provides similar services, with companies like Amazon, Google, IBM in collaborating with few of these to provide cloud base solution to enterprise the service and support backed up by these big players.

TAKE AWAY

Docker is container based ecosystem growing rapidly is most promising disruptive solution on cloud platform. Community is leveraging multiple IaaS service provider for portability, scale on demand, fault tolerance and performance with continuous application delivery.

Docker containers has added the ease of installation to cloud platform over the heavy virtual machines because of its portability and one stop installation module many enterprises are moving towards container based services (Container-as-a-Service).

REFERENCES

[1] J. MSV, "Docker and the linux container ecosystem." [Online]. Available: www-gigaom.com/report/docker-and-the-current-linux-container-ecosystem
[2] "Ecs." [Online]. Available: www.aws.amazon.com/ecs/
[3] "Big data analytics and docker: The thrilla in manila." [Online]. Available: www.openstack.org/videos/vancouver-2015/big-data-analytics-and-docker-the-thrilla-in-manila
[4] M. Balkenende, "Choose your own big data adventure: Getting started with talend's new big data sandbox," September 2016. [Online]. Available: www.talend.com/blog/2016/09/20/choose-your-own-big-data-adventure-getting-started-with-talends-new-big-data
[5] "Sa home loans adopts microservices and deploys 20-30 times a day with docker datacenter." [Online]. Available: www.docker.com/sites/default/files/DOC_SA_CaseStudy_04252015_V2%20%282%29%5B2%5D.pdf
[6] M. Betz, "Five alternatives to docker you should consider." [Online]. Available: http://searchcloudapplications.techtarget.com/tip/Five-development-containers-to-consider-that-arent-Docker

AUTHOR BIOGRAPHY

Vishwanath Kodre received his Masters Degree in Computer Science from Pune University. He is currently studying Data Science at Indiana University Bloomington.

Couchbase Server: A Usable Overview

MATTHEW LAWSON[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: laszewski@gmail.com*

Paper1, March 14, 2017

Couchbase, Inc. develops Couchbase Server (CBS), an open-source, document-oriented, NoSQL database. Couchbase targets situations requiring high availability and high throughput of large amounts of data, i.e., big data. CBS integrates Couchstore, Memcache and ForestDB, as well as a host of maintenance, administration and querying tools, in order to attempt to meet its promises to its users. Corporate Couchbase users include General Electric (GE), LinkedIn Corp. and American Airlines, among others..

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Couchbbase, Memcache, CouchDB, Cloud, I524

https://github.com/eunosm3/classes/blob/master/docs/source/format/report/report.pdf

INTRODUCTION

Couchbase, Inc. offers Couchbase Server (CBS) to the marketplace as its entry in the NoSQL, *big data* database field. Salient features include a) an integrated cache tier which is essential to the product's operation; b) persistent storage in JSON document format, i.e. document-based storage, or simple key-value pairs; c) relatively uncomplicated scalability across clusters of commodity servers; d) sub-millisecond response times; e) a SQL-like query language; and, f) built-in cluster replication, failover and disaster recovery features. In addition, Coucbhase markets a mobile product, Couchbase Mobile, which uses a Couchbase-designed syncing system to extend CBS to mobile devices and offline use cases.

ARCHITECTURE

A Couchbase Server (CBS) system consists of at least one cluster of interconnected servers running a copy of CBS. By default, the CBS system's computers, referred to as nodes, work together in a master-master setup, which Couchbase calls a peer-to-peer topology [1]. In a master-master distributed cluster, the nodes co-exist in flat hierarchy, i.e., no node acts as the central authority. Despite the egalitarian nature of the cluster, the nodes still need to coordinate activities. Therefore, the nodes *elect* a node to coordinate cluster functions. If the node fails or is removed from the cluster, the remaining nodes elect a new *orchestrator*.

In addition, the database administrator can override the default peer-to-peer topology by taking advantage of CBS' *multi-dimensional scaling*. This functionality allows the administrator to customize nodes to perform tasks for which the node is best suited, e.g., memory-intensive processes or I/O-intensive processes, etc.

A complete CBS system physically consists of a) one or more server clusters running the couchbase daemon; b) high-speed connections between the servers and between the clusters; and, c) client computer applications utilizing memcached-compatible software development kits, also known as *devkits* or *SDKs*. **RESPONSE: I am not sure how to comply because this paragraph represents a portion of the mosaic of knowledge I now possess regarding CBS. That is, it is an original thought.**.

The main components of a Couchbase Server node consist of the following: a) the cluster manager; b) the data service; c) the query service; and, d) the index service, as well as the underlying managed cache and storage components. [1]

Cluster Manager The Cluster Manager, which runs on every node, manages each respective node's interaction and involvement with the other nodes in the cluster. The Cluster Manager configures and monitors the node, determines the layout for CBS' primary services, e.g., Data, Query and Index Services, controls data rebalancing amongst the cluster's nodes, gathers operational statistics, determines the nodes's membership in a cluster, authenticates connections to the cluster, responds to heartbeat requests and repairs itself if possible [1, 2].

Data Service The Data Service provides the core functionality of any type of database management service, i.e., data access. CBS organizes documents, or items, into *buckets* and *vBuckets* [3]. CBS distributes a bucket evenly across the cluster's nodes. CBS refers to the portions of a bucket on a single node as a vBucket, which conceptually resembles a RDBMS shard. Buckets typically have 1,024 vBuckets, so a three-node cluster with one bucket would have 341 vBuckets on two nodes and 342 vBuckets one the last node.

The Data Service provides an API for creating, retrieving, updating and deleting (CRUD) items in CBS. It operates on items with keys in buckets.

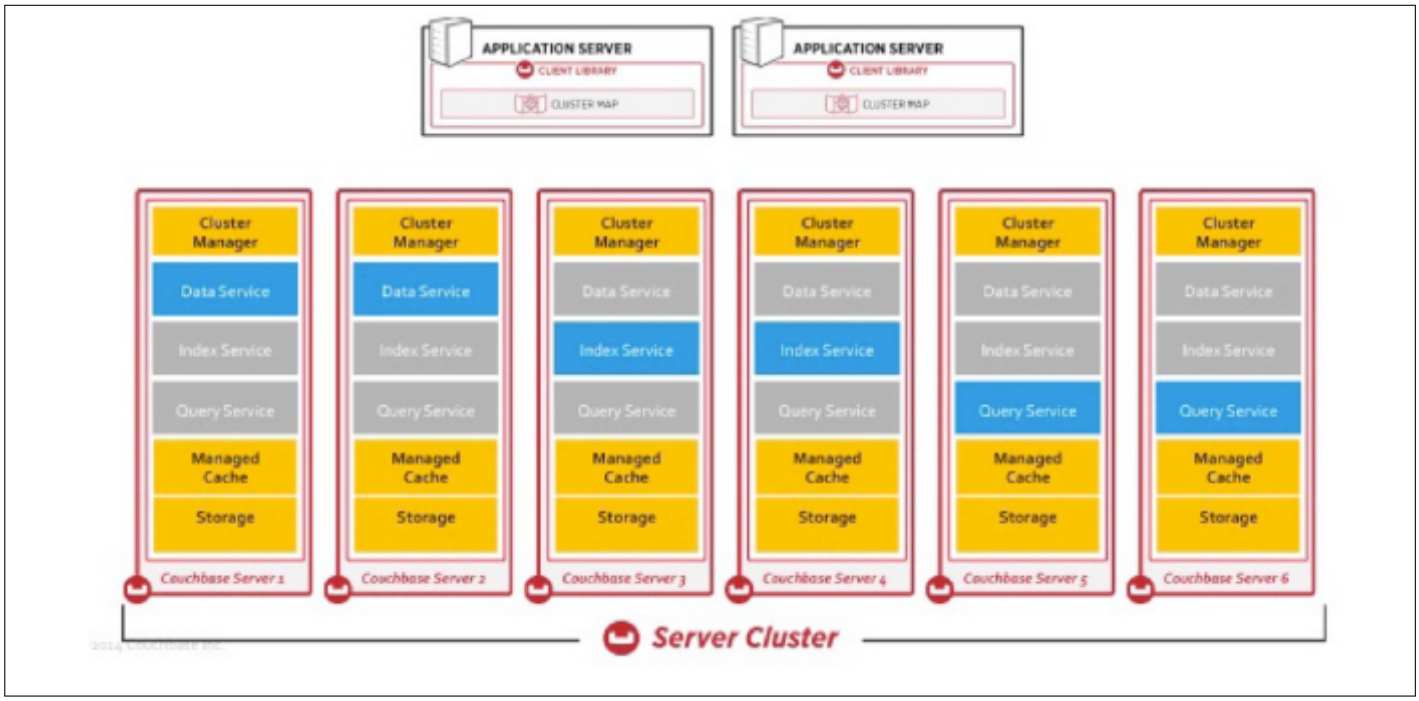


Fig. 1. Multidimensional Scaling with Couchbase [1]

Indexes and Index Services The index services create, maintain and destroy primary and secondary indexes of a bucket's keys for three index services. Couchbase refers to its index services as *Map-Reduce Views*, *Spatial Views* and *Global Secondary Indexes* or GSIs [4].

Views, which represent CBS's first generation index service, exist within CBS' Data Service. Map-Reduce Views return, or emit, document attributes as View keys after applying user-defined map-reduce functions to JSON documents. Spatial Views act in a similar fashion, except they process geographic information and emit geographic coordinates as View keys.[5]. Spatial Views for geospatial data equate to Map-Reduce Views non-geospatial data.

In contrast, CBS' Index Service represents the software's next-generation index, the GSIs. Couchbase developed GSIs in conjunction with, and in service of, its SQL-like query service.

As a result of their respective historical development paths, choosing to use a View or a GSI depends on the use case. For instance, Map-Reduce View indexes allow users to create arbitrarily complex indexes for later use. "[Map-Reduce][v]iews are typically useful for interactive reporting type queries where complex data processing and custom data reshaping is necessary [4]." Spatial View indexes allow users to create "multidimensional bounding box queries for location aware applications." [4]

Since Views, Map-Reduce or Spatial, exist as part of the Data Service, they are partition-aligned with the core data distribution. That is, CBS spreads Views across the cluster roughly proportionate to the underlying data. Therefore, performance slows as the number of nodes contacted increases due to network processing needs.

In contrast, Couchbase constrains a GSI to residence on a single node, i.e., *not* partition-aligned. This design allows GSIs to return results faster than Views. However, GSIs can handle only relatively simpler queries. In addition, users must manually create identical GSIs in order to use the index on multiple nodes for concurrent searches, or as a backup option.

Finally, a CBS system's *primary index* holds information for all of the data in a bucket, while its *secondary index* holds data for a pre-specified subset of the data. Couchbase encourages the use of secondary indexes since they avoid scanning the contents of an entire bucket index.

Query Service Couchbase Server provides four methods of querying the data. First, users can take advantage of the Data Service's key-value API. This method returns results faster the other methods, but it requires the user to know the item's key. The second and third methods complete query execution by accessing the Views API. Such queries operate on the map-reduce or spatial Views keys. These two methods provide the greatest query flexibility, including data reshaping, at the cost of increased elapsed query time. The fourth method provides query flexibility and speed between the key- value API and the Views API. Couchbase calls its newest method N1QL [4]. Although the company designed GSIs for use by N1QL, it can complete ad-hoc queries, i.e., queries without a pre-defined index. It can also exploit View indexes in a limited fashion.

Managed Cache "Since Couchbase built Couchbase Server on a memory-first architecture, achieving high performance and scalability requires effective memory management."[6]. CBS stores frequently accessed data items, such as documents and indexes, in its integrated cache tier. Couchbase opted for this setup as a method to provide high-performance, i.e., as fast as volatile memory allows, reads, writes and queries. CBS monitors the frequency with which users access items in order to determine which items to retain in cache and which items to write to disk. The various CBS services, e.g., Data Service, manage their respective cache usage to optimize their respective tasks. In addition, CBS administrators can allocate certain amounts of cache space by changing the system's Ram Quotas. [6]

Storage Components CBS utilizes two distinct storage engines, namely, Couchstore and ForestDB. Couchstore supports the Data Service, and, by extension, the View index service. It uses a B+tree structure for key-based access. It also captures changes to items via an append-only write model. In contrast, ForestDB uses a B+trie structure for key-based access. "B+trie provides a more efficient tree structure compared to B+trees and ensures a shallower tree hierarchy to better scale large item counts and very large index keys." [7]. ForestDB defaults to using an append-only write model, but can also utilize a "circular-reuse" model. The latter takes advantage of orphaned space the former ignores, thus reducing the frequency of compaction.

B-tree, B+tree and B+trie Databases and filesystems commonly utililze a B-tree structure because data access and manipulation occurs in logarithmic time. A B+tree structure increases data access performance for filesystems over a B-tree because a) each node only holds keys instead of a key-value pair like a B-tree and b) each node has an additional level of linked leaves associated with it; these leaves function as a kind of metadata for the nodes. Finally, a B+trie node does not store key-value pairs or keys. Instead, its position in a B+trie determines the key with which it is associated. As a result, a B+trie's data access speed exceeds that of a a B+tree.

Cross Data Center Replication Service [XDCR] Couchbase created a service for CBS, *Cross Data Center Replication* or XDCR [8], to enhance data availability and disaster recovery. XDCR syncs data between separate CBS clusters, which can co-exist within a single data center or can reside in entirely separate geographies. Besides data replication for disaster recovery, XDCR can be configured to immediately take over for a failed primary cluster. In addition, XDCR can reduce latency by moving the data closer to the end user. Companies using CBS can target "external applications (e.g. Elastic, Spark, Storm, etc.)."

USER INTERFACES

API Client applications interact with CBS through memcached-compatible SDKs, which support numerous programming languages. As of version 4.6, developers could choose from an SDK for the following languages: a) Node.js; b) Java; c) PHP; d) .NET; e) Python; f) Go and g) C [9]. Couchbase also provides a client library for JDBC/ODBC [10].

Shell Access Couchbase offers a variety of command line tools. The *cbc* tool operates on a node, a bucket or a vBucket (shard). It includes commands to create, retrieve or remove documents in a CBS system, list the buckets in a cluster, manage users, etc. **RESPONSE: I do not understand the critique. The name of the tool is cbc. I also provide a brief explanation.** In addition, each CBS installation includes the *cbq* tool to issue N1QL queries [11]. **RESPONSE: I do not understand the critique. The name of the tool is cbq. I mention its primary purpose.**. CBS includes a number of other command line tools to accomplish various tasks [12].

Graphical Interface Couchbase implements CBS' GUI via a web browser. Users access the web GUI by navigating to a cluster's url appended with the admin port number. The browser interface acts as the primary management tool for CBS. **RESPONSE: Couchbase openly publishes the admin port number, so mentioning it did not pose a security threat.**. It offers access to node management, queries, indexes, et cetera [13].

LICENSING

Couchbase, Inc. offers a community edition of Couchbase Server as well as an enterprise edition. Couchbase Server Enterprise Edition includes more features and better quality assurances, e.g., testing and bug fixes, versus Couchbase Server Community Edition. Couchbase targets "enterprise customers with large production deployments running in data centers and/or public clouds" [10] with the Server Edition. The remaining, primary differentiating factor of the Enterprise Edition over the Community Edition consists of Couchbase's 24x7 technical support. Community Edition users must rely on published material and the online CBS community forum instead of dedicated technical support.

ECOSYSTEM

CBS does not have a large ecosystem built around it, but Couchbase has developed a number of interfaces to software often used in conjunction with large data sets. The company offers the aforementioned client librariers, e.g., .NET, node.js, et al, as well as connectors and plugins for a) Spring Framework (connector); b) Spark (connector); c) Kafka (connector); d) Hadoop Sqoop (plugin); e) ElasticSearch (plugin); and, f) Solr LucidWorks Fusion (unspecified). Couchbase also maintains Moxi Server, a proxy for memcached traffic [10].

USE CASES

General Use cases include a) supporting / enabling real-time analytics; b) building mobile apps with offline support via Couchbase Lite; c) digital communication by enabling low-latency read / write access to messages; and, d) purportedly holistic views of client data via aggregation from multiple sources even when the sources have different data models.

Use Cases for Big Data Couchbase markets CBS to customers who desire high throughput / low latency response times from a so-called schema-less database managing data at scale, i.e., *big data*. In the context of NoSQL, big data databases, low latency translates to sub-millisecond response times. Other aspects of competitive products in this space include scalability, a flexible data model (as implied by the NoSQL tag), a SQL-like query language and simple administration.[14]

The company highlights a number of real-world business wins to support its assertions that CBS meets these criteria.

Equifax, Inc. For instance, Equifax chose Couchbase Server Enterprise Edition when it needed to meet a new customer need in a short amount of time. In October 2015, the Federal National Mortgage Association, a government-sponsored entity (GSE) more commonly referred to as Fannie Mae, announced it would begin providing 24 months of trended credit history on its industry-standard *Desktop Underwriter* software instead of a point-in-time snapshot. Fannie Mae promised this change by the end of the second quarter of 2016. Therefore, Equifax had less than three calendar quarters to scale up its trended data product for a customer that underwrote nearly 46% of all US residential mortgages at the time, when combined with its GSE-twin, Freddie Mac [15, 16]

Equifax needed a solution to handle the five petabytes (5Pb) of data plus the necessary throughput associated with trended data. In addition, it needed a) its new software to work with systems the company already used, like Hadoop and Spark; b) it needed the solution to facilitate application development; and, c)

it needed five millisecond (5ms) response times. CBS met those requirements for Fannie Mae. The mortgage underwriting GSE also found the ease of data replication offered by CBS' XDCR attractive, as well as the minimal Java coding needed to make CBS' Views useful to its operations teams. [17]

LinkedIn Corp. LinkedIn also opted for Couchbase as its data management needs grew. More specifically, the challenges of moving data across its hosts / clusters with its prior Memcache-only design prompted it to consider other solutions. The company currently utilizes CBS as a) a simple read-through cache; b) an ephemeral counter store, i.e., storage for temporary IDs; c) a temporary de-duplication store; and, d) a *source of truth* for internal tooling. LinkedIn's data expands across 148 buckets and 2,821 hosts. The largest cluster by nodes consists of 72 hosts, while the largest cluster by documents holds 1.4 billion items. Overall, its CBS system handles 10 million-plus queries per second (QPS) [18]

EDUCATIONAL MATERIAL

If you would like to learn more about Couchbase Server, visting developer.couchbase.com or the Couchbase Connect section of Couchbase's youtube.com channel should prove beneficial. In addition, perusing the works cited in the reference section may also prove beneficial.

CONCLUSION

Couchbase Server appears to offer the necessary features to succeed commercially as a *big data* database. That is, it scales well, it handles extremely large datasets well, it handles high-throughput transactions well and it has a SQL-like query interface. Whether or not CBS will succeed due to technical superiority, administrative ease or because Couchbase simply marketed better than the competition exceeds the scope of this write-up, though. Based on the feature set and the business wins, it appears to be a legitimate option for organizations interested in this type of general product.

ACKNOWLEDGEMENT

I would like to thank Dr. Gregor von Laszewski, the TAs for I524, Big Data Software and Projects in the Cloud and the other students in the class for their insights and assistance related to this paper.

I would also like to thank my employer, Indiana Farm Bureau, which funded this research, in part, via its employee education assistance program.

REFERENCES

[1] Couchbase, Inc., "Distributed data management," Web page, feb 2017, online; accessed 19-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/concepts/distributed-data-management.html

[2] ——, "Cluster manager," Web page, feb 2017, online; accessed 20-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/architecture/cluster-manager.html

[3] ——, "Couchbase server architecture," Web page, feb 2017, online; accessed 20-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.0/architecture/architecture-intro.html

[4] ——, "Views, indexing, and index service," Web page, feb 2017, online; accessed 20-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.0/architecture/views-indexing-index-service.html

[5] ——, "Query data and query data service," Web page, feb 2017, accessed 21-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/architecture/querying-data-and-query-data-service.html

[6] ——, "Managed caching layer architecture," Web page, feb 2017, online; accessed 19-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/current/architecture/managed-caching-layer-architecture.html

[7] ——, "Storage architecture," Web page, feb 2017, accessed 21-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/architecture/storage-architecture.html

[8] ——, "Cross datacenter replication (xdcr)," Web page, feb 2017, accessed 23-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/xdcr/xdcr-intro.html

[9] ——, "Start using the sdk," Web page, feb 2017, accessed 23-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/sdk/java/start-using-sdk.html

[10] ——, "Couchbase server & n1ql," Web page, feb 2017, accessed 23-feb-2017. [Online]. Available: https://www.couchbase.com/downloads

[11] ——, "Browser and cli access," Web page, feb 2017, accessed 23-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/sdk/webui-cli-access.html

[12] ——, "Cli reference," web page, feb 2017, accessed 24-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/cli/cli-intro.html

[13] Couchase, Inc., "Couchbase web console," Web page, feb 2017, accessed 23-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/current/admin/ui-intro.html

[14] Couchbase, Inc, "Why couchbase?" Web page, feb 2017, accessed 23-feb-2017. [Online]. Available: https://developer.couchbase.com/documentation/server/4.6/introduction/intro.html

[15] Equifax, Inc., "Fannie mae to introduce equifax trended data and verification service to underwriting platform," Web page, oct 2015, accessed 23-feb-2017. [Online]. Available: https://goo.gl/TCev5B

[16] ValueWalk Staff, "Fannie mae: Who owns the us.. mortgage markets?" Web page, mar 2016, accessed 23-feb-2017. [Online]. Available: http://www.valuewalk.com/2016/03/fannie-mae-who-owns-the-u-s-mortgage-markets/

[17] J. Duraisamy and G. Lee, "Connecting the dots with couchbase," nov 2016, accessed 23-feb-2017. [Online]. Available: https://m.youtube.com/watch?list=PLcspbWiU9RuunKnZwfE757B6-xsaiJV84&v=0dKXHy6vJRA

[18] M. Kehoe, "Linkedin: Going all in: from a single use case to many - couchbase connect 2016," Youtube Video, nov 2016, accessed 23-feb-2017. [Online]. Available: https://m.youtube.com/watch?list=PLcspbWiU9RuunKnZwfE757B6-xsaiJV84&v=1shb4UZON_I

AUTHOR BIOGRAPHIES

Matthew Lawson received his BSBA, Finance in 1999 from the University of Tennessee, Knoxville. His research interests include data analysis, visualization and behavioral finance.

WORK BREAKDOWN

The work on this project was distributed as follows between the authors:

Matthew Lawson. Researched Couchbase Server and related topics, wrote the paper and edited the paper.

Apache Airavata

SCOTT MCCLARY[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: scmcclar@indiana.edu*

paper-001, March 9, 2017

Apache Airavata provides an alternative to running and monitoring large-scale scientific applications from the command line. The Apache Software Foundation's Airavata software framework allows developers to create what are known as Science Gateways. These Graphical User Interfaces are desktop-based and/or web-based applications, which allow researchers to compose, manage, execute and monitor their research workflows in a user-friendly manner. Apache Airavata simplifies the process of accessing the large-scale computational power of local clusters, supercomputers, computational grids and computing clouds.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, Gateway, HPC, I524, Middlware, Workflow

https://github.com/scottmcclary1/sp17-i524/blob/master/paper1/S17-IO-3011/report.pdf

1. INTRODUCTION

Apache Airavata is an open-source software framework designed to diminish the learning curve and reduce the inherent complexity of conducting large-scale scientific computing. Therefore, scientific researchers leverage the Apache Airavata software framework in order to obscure the intricacies of running large-scale applications or workflows on local clusters, powerful supercomputers or distributed clouds. The expertise and interests of a given researcher likely revolves around their specific area of research (i.e. Computational Chemistry, Molecular Dynamics and etc.). The Apache Airavata technology allows these researchers to focus their time, effort and grant money on the science rather than the details of the computing. In addition to executing and monitoring large-scale scientific applications, managing the input and visualizing the output from the command line can be complicated on distributed compute resources. Apache Airavata provides the technological infrastructure to make these complicated tasks simple. The general idea is to wrap command line-driven applications (i.e. Gaussian, Amber, NAMD and etc.) with Apache Airavata in order to create simple, effective and efficient Science Gateways. The Apache Airavata software framework provides the infrastructure to allow Science Gateway developers to abstract away the described complexity so that end-users can simply "compose, manage, execute, and monitor large-scale applications and workflows" with the click of a button [1].

2. ARCHITECTURE

Figure 1 depicts the architectural details of Apache Airavata. From this diagram one can see how end-users are able to interact with the Apache Airavata services (API Server, Workflow Interpreter, Orchestrator, Messaging System and etc.) via Science Gateways. The Apache Software foundation provides a detailed description of the architectural diagram shown in figure 1 [2].

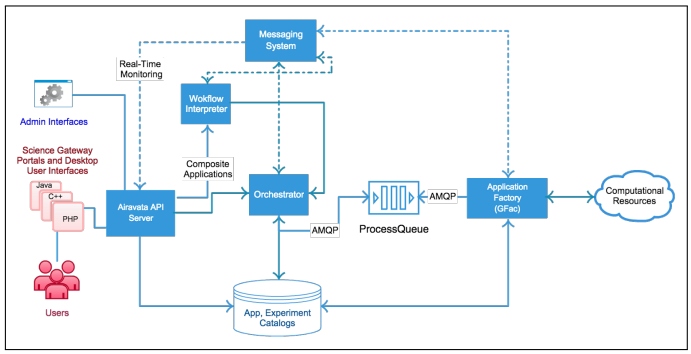


Fig. 1. The image above depicts the architectural details of Apache Airavata [2].

2.1. API

As introduced in section 4 and shown with multiple real-world examples in section 5, researchers leverage Science Gateways to interact with Apache Airavata. In order to promote simplicity, Apache Airavata's application programming interface (API) is intentionally obscured from these end-users. Therefore, the Airavata API is generally intended for Science Gateway developers who are specifically interested in using Apache Airavata as a middleware service between a user interface and one or more compute resources, as shown in figure 2. Airavata API is written

using apache thrift [2]. This allows Science Gateway developers to use the programming language of their choice (e.g. Java, PHP, JavaScript, C++, etc.). The Apache Software Foundation provides an in-depth overview of the API for those interested in learning the details of this service [3].

2.2. Shell Access

As section 1 thoroughly explained, Apache Airavata's purpose is to simplify the typical command line driven process of composing, managing, executing, and monitoring large-scale scientific applications on powerful distributed computing resources. Therefore, end-users of Apache Airavata should not rely on shell access. Instead, section 2.3 explains that end-users interact with Apache Airavata through graphical interfaces (i.e. Science Gateways). Shell access is contained to the Science Gateway developer level of the technological ecosystem, shown in figure 2.

2.3. Graphical Interface

Apache Airavata leverages a Graphical Workflow Composer, known as XBaya, which helps "create workflows, submit and manage multiple applications ... [and] also has a web-based interface ... where users can ... register, run and monitor applications " [2]. Furthermore, the Apache Airavata's thrift-based API, introduced and discussed in section 2.1, allows developers to create their own desktop and web interfaces using Airavata as the technological foundation.

2.3.1. Science Gateways

Science Gateways are the resulting reward of leveraging Apache Airavata. As explained in 4, Science Gateways are an instrumental piece of the Apache Airavata ecosystem that allow end users to compose, manage, execute, and monitor scientific research workflows on distributed (and potentially complex) compute resources. Science Gateways simplify workflows and allow researchers to focus their expensive time and effort on science. Furthermore, Science Gateways promote reproducibility, which is an important piece of scientific research as well as publication. In other words, it is beneficial to researchers to have the ability to easily reproduce results from an experiment in the past and Science Gateways typically ensure this functionality. Section 5 describes many of the prominent and currently available Science Gateways built on top of the Apache Airavata software framework.

3. LICENSING

Apache Airavata is open source software [4]. Therefore, anyone can download, install, modify and improve this software framework using the popular "fork and pull" model. As part of an introductory tutorial, one can create an Airavata Test Drive account, which is free as well. Note, this free Airavata Test Drive account will need to be approved by an Airavata Test Drive Administrator before one can use the system. Once this approval occurs, one can simply create and run "experiments" (i.e. Gaussian, Amber, Trinity and etc.) via a Science Gateway on local clusters, supercomputers, computational grids and computing clouds.

4. ECOSYSTEM

Figure 2 depicts the ecosystem developed around Apache Airavata for SEAGrid. Apache Airavata plays an important role as the middleware between the end-users (Science Gateways) and the compute resources. As discussed above, the placement of Apache Airavata in between the compute resources and the science gateways allows for the abstraction of the complexity of composing, running, executing and managing applications and workflows.

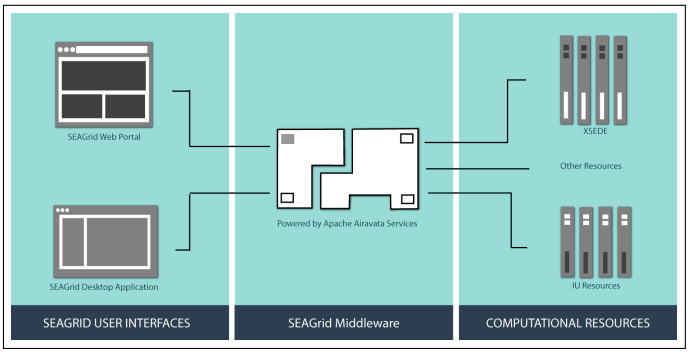


Fig. 2. The image above depicts an example of the technological ecosystem developed around Airavata [5].

5. USE CASES

Ultrascan [6], SEAGrid [7] and GenApp [8] are instances of Science Gateways that leverage Apache Airavata to perform computations [1]. Each of these services is in place to simply bridge the gap between scientific applications on large-scale compute resources and domain specialists. In other words, these Science Gateways are examples of Apache Airavata enabling science.

5.1. SEAGrid

The Apache Airavata services allow Science Gateways such as SEAGrid to simplify the use of "scientific applications deployed across a wide range of supercomputers, campus clusters and computing cloud" [7]. SEAGrids bridges the gap between domain specialists and scientific applications on large-scale compute resources using both a desktop client and web application. SEAGrid currently promotes research in Computational Chemistry (e.g. Gaussian, Gamess, etc.), Molecular Dynamics (Lammps, Amber, NAMD and etc.), Structural Mechanics (e.g. Abaqus and etc.), Fluid Dynamics (e.g. Nek5000, OpenFOAM and etc.) and much more. SEAGrid abstracts away the fine-grained details of running such scientific applications on large-scale compute resources and therefore allows the domain specialists to focus on the fine-grained details of their scientific research. Additionally, SEAGrid enables scientists to create model inputs, visualizations of outputs and archives for simulation data" [7].

5.2. Use Cases for Big Data

The One-Degree Imager (ODI) "is a gigapixel mosaic camera ... built by [the] WIYN Observatory with a pixel scale of 0.1 arcseconds for the 3.5-meter telescope" and is the newest instrument at the WIYN 3.5m Observatory in Sells, AZ [9, 10]. Similarly to the SEAGrid Science Gateway, the Apache Airavata software framework is at the foundation of ODI's Pipeline, Portal and Archive (PPA) system which "execute[s] the NOAO High Performance Pipeline System (NHPPS) pipelines on XSEDE resources" [9]. The large amount of data generated by the ODI demonstrates that the ODI-PPA Science Gateway and therefore

the Apache Airavata software framework can handle big data software projects.

6. EDUCATIONAL MATERIAL

There are multiple ways to find out more information about the Apache Airavata software framework [1]. In order to cater to varying audiences and learning styles there is online documentation [2], a course provided at Indiana University, Bloomington [11] as well as online tutorials [12]. The online documentation is entirely sufficient for motivated users to teach themselves. The online tutorials provide everything from quick-start to extended tutorials and everything in between. Historically, there has been a basic Airavata course at Indiana University offered during the Fall semester and a more advanced version of the course is offered during the Spring semester.

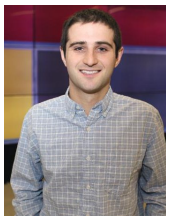
7. CONCLUSION

Apache Airavata appeals to a wide range of scientific researchers since the technology allows researchers to focus their time, effort and grant money on the science rather than the details of the computing. Apache Airavata has enabled researchers to compose, manage, execute and monitor workflows on large-scale systems with the click of a button. Gateways such as Ultrascan, SEAGrid and GenApp are clear and defined examples of how Apache Airavata has been leveraged to improve and optimize scientific research workflows. As compute resources get more complicated and/or distributed over time, the Apache Airavata software framework will continue to promote the ease of use with Science Gateways.

ACKNOWLEDGEMENTS

The authors would like to thank the School of Informatics and Computing for providing the Big Data Software and Projects (INFO-I524) course [13]. This paper would not have been possible without the technical support & edification from Gregor von Laszewski and his distinguished colleagues.

AUTHOR BIOGRAPHIES



Scott McClary received his BSc (Computer Science) and Minor (Mathematics) in May 2016 from Indiana University and will receive his MSc (Computer Science) in May 2017 from Indiana University. His research interests are within scientific application performance analysis on large-scale HPC systems. He will begin working as a Software Engineer with General Electric Digital in San Ramon, CA in July 2017.

WORK BREAKDOWN

The work on this project was distributed as follows between the authors:

Scott McClary. He completed all of the work for this paper including researching and testing Apache Airavata as well as composing this technology paper.

REFERENCES

[1] Apache Software Foundation, "Apache Airavata," Web Page, 2016, accessed: 2017-2-22. [Online]. Available: http://airavata.apache.org

[2] ——, "Apache Airavata Overview - Apache Airavata - Apache Software Foundation," Web Page, February 2016, accessed: 2017-2-22. [Online]. Available: https://cwiki.apache.org/confluence/display/AIRAVATA/Apache+Airavata+Overview

[3] ——, "Airavata API Overview - Apache Airavata - Apache Software Foundation," Web Page, May 2014, accessed: 2017-2-22. [Online]. Available: https://cwiki.apache.org/confluence/display/AIRAVATA/Airavata+API+Overview

[4] ——, "Airavata Source Code & Developers Guide," Web Page, 2014, accessed: 2017-2-22. [Online]. Available: https://airavata.apache.org/development/source.html

[5] Indiana University, Apache Airavata, XSEDE, and NSF, "SEAGrid Portal," Web Page, accessed: 2017-2-22. [Online]. Available: https://seagrid.org/themes/seagrid/assets/img/workflow.png

[6] UltraScan Project, "UltraScan Analysis Software," Web Page, April 2015, accessed: 2017-2-22. [Online]. Available: http://ultrascan.uthscsa.edu

[7] Indiana University, Apache Airavata, XSEDE, and NSF, "SEAGrid Portal," Web Page, accessed: 2017-2-22. [Online]. Available: https://seagrid.org

[8] Apache Software Foundation, "GenApp - Apache Airavata - Apache Software Foundation," Web Page, August 2014, accessed: 2017-2-22. [Online]. Available: https://cwiki.apache.org/confluence/display/AIRAVATA/GenApp

[9] ——, "ODI - Apache Airavata - Apache Software Foundation," Web Page, January 2017, accessed: 2017-2-22. [Online]. Available: https://cwiki.apache.org/confluence/display/AIRAVATA/ODI

[10] WIYN Observatory, "WIYN 3.5m Observatory," Web Page, February 2017, accessed: 2017-2-22. [Online]. Available: https://www.noao.edu/wiyn/ODI/

[11] M. Pierce and S. Marru, "I590 Course Structure," Web Page, accessed: 2017-2-22. [Online]. Available: http://courses.airavata.org/spring2016/public/slides/I590CourseStructure.pdf

[12] Apache Software Foundation, "Airavata Quick-Start Tutorials - Apache Airavata - Apache Software Foundation," Web Page, July 2016, accessed: 2017-2-22. [Online]. Available: https://cwiki.apache.org/confluence/display/AIRAVATA/Airavata+Quick-Start+Tutorials

[13] Gregor von Laszewski and Badi Abdul-Wahid, "Big Data Classes," Web Page, Indiana University, Jan. 2017. [Online]. Available: https://cloudmesh.github.io/classes/

Google Bigtable

MARK MCCOMBE[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: mmccombe@iu.edu*

paper1, March 7, 2017

Google's NoSQL database, Bigtable, is a critical technology in big data for its use internally at Google, as the external service Cloud Bigtable, and for inspiring open source technologies such as Hbase. An overview of Bigtable's storage model and architecture are presented, including available APIs, shell access, and the graphical user interfaces. Performance and security features of Bigtable are discussed along with technologies related to Bigtable. Internal and external use cases involving Bigtable are detailed. Finally, educational resources for learning more about Bigtable are identified.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Bigtable, Google, NoSQL, I524

https://github.com/cloudmesh/sp17-i524/tree/master/paper1/S17-IO-3012/report.pdf

INTRODUCTION

Google Bigtable is a NoSQL database developed by Google, built on several Google technologies, including Google File System, Chubby Lock Service, and SSTable[1]. One of the earliest NoSQL databases, development on Bigtable started in 2004 and Bigtable was introduced to the public in a paper published in 2006 [2]. Bigtable is important to Big Data both for its use internally at Google and externally as Cloud Bigtable, which was made available in May 2015. Google uses Bigtable to power many core Google products, such as Search, Analytics, Maps, Earth, Gmail, and YouTube. [1].

Bigtable has inspired other technologies, notably Hbase [3] an open source distributed, scalable database that was modeled after Bigtable and is typically used along with Hadoop and Hadoop Distributed File System (HDFS) as part of the Apache Big Data Stack.

While Bigtable is a significant technology in Big Data due to its use in Google products and role in the development of other NoSQL technologies, Cloud Bigtable itself is not one of the more popular databases available today. DBEngines ranks Cloud Bigtable only 6 of 9 among Wide Data Stores and 166 of 285 among databases overall for popularity [4].

STORAGE MODEL

Bigtable stores data in tables, which are sorted by key/value maps. Tables have rows, typically a single entity, and columns which contain values for the rows. Rows are index by a row key. Columns have both a family and a qualifier, which is unique within a family [5].

Figure 1 shows the Bigtable Storage model. It contains four rows (row keys - gwashington, jadams, tjefferson, and wmckinley), one column family (follows), and four column identifiers (also gwashington, jadams, tjefferson, and wmckinley). Tables in Bigtable are sparse, meaning that a cell will not take up space if it does not contain data (as in the case of jadams/wmckinley). Intersections may contain multiple cells with different timestamps (as in the case of tjefferson/gwashington and jadams/tjefferson) providing a historical record of data in Bigtable [5].

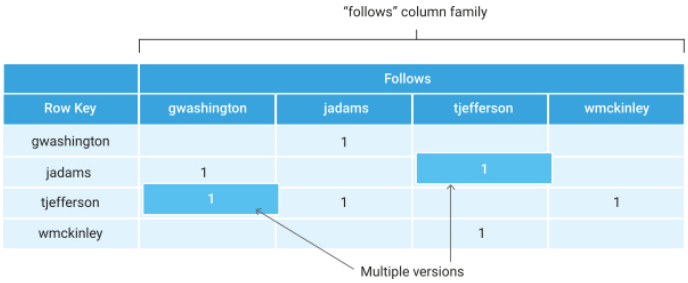


Fig. 1. Bigtable Data Model [5]

ARCHITECTURE

The architecture of Cloud Bigtable is depicted in Figure 2. As shown, client requests come through a front-end server pool and are directed to a Bigtable node (called tablet servers when Bigtable was introduced in 2006). Bigtable nodes are organized into Bigtable clusters, which in turn each belong to a Bigtable instance [5].

Tables in Bigtable are sharded tablets, which contain blocks of contiguous rows, to balance query workload. Tablets are stored in SSTable format, which provides a map from keys to values,

in Google's file system Colossus and housed in Google's data centers. Each tablet belongs to a specific node [5].

The approach of storing data in tablets rather than rows provides performance benefits and fault tolerance to Bigtable. Because no data needs to be copied, rebalancing tablets between nodes is fast. If a Bigtable node fails, no data is lost and recovery is quick because only metadata needs to be copied to the new node [5].

Bigtable balances data volume and workload across clusters automatically. Bigtable handles this automatically, reducing the administrative effort required[5].

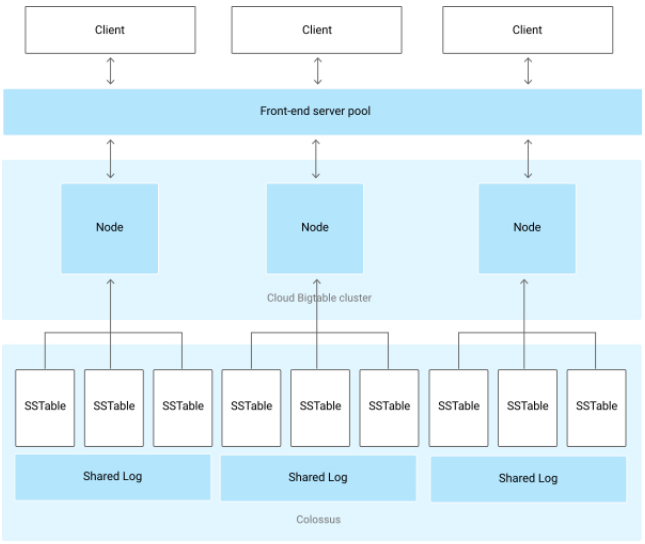


Fig. 2. Bigtable Architecture [5]

API

Bigtable APIs exist for several languages including HBase Client for Java, Go Client, Python Client, Bigtable-dotnet (.NET), Scio (Scala), and Dataflow Connector (for use in Pipelines) [6].

Bigtable can also integrate with Google Cloud Dataflow, a cloud based, big data programming model, and with Apache Hadoop through the use of Google Cloud Dataproc [6].

Shell Access

Shell access in Cloud Bigtable can be performed through the HBase Shell [7]. The HBase shell provides a Ruby environment that allows functions in BigTable to be executed on the command line and provides scripting capabilities.

HBase shell commands fall into four main categories [8]. First are general commands. Examples of general commands are version and status. Second are table management commands, which provide functionality to create, drop, and alter tables. The third category is data management commands which include commands like count, get, and delete. The final type of commands enabled by the shell are cluster replication commands, which enable stopping and starting replication and adding and removing peers.

In addition to HBase Shell, cbt, a command line tool written in Go, can be used to perform operations against Bigtable [9].

Graphical Interface

BigQuery Web UI is a graphical interface, designed to run in Google's Chrome browser, that allows users to interact with BigTable. Functionality provided by BigQuery Web UI includes the ability to load and export data, to run queries, to create, delete, copy and append to tables, and to view, add, delete and share datasets [10].

LICENSING

Bigtable is closed source software and is not available for free use outside of Google. Cloud Bigtable is available for public use on at a cost to the user. The current pricing structure in Figure 3.

Nodes	$0.65 node/hr (minimum 3 nodes)
SSD Storage	$0.17 (GB/Month)
HDD Storage	$0.026 (GB/Month)
Network Ingress	Free
Network Egress	Cross-region and Internet egress rates apply

Fig. 3. Cloud Bigtable Cost Structure [11]

Bigtable has inspired several open source projects which are based on the concepts of Bigtable, notably HBase, Hypertable, and Accumulo (all further discussed in *Bigtable Alternates*). Hypertable is licensed under the GNU General Public License Version 3 while HBase and Accumulo are licensed under the Apache License Version 2.0.

PERFORMANCE

According to Google, "Bigtable is designed to handle massive workloads at consistent low latency and high throughput" [11]. While Google does not release performance details of its internal applications, the fact that Google uses Bigtable as the data store for applications that successfully support extremely large data volumes such as Search, Gmail, and Maps supports this claim.

FIS Advanced Technology analyzed the performance of Cloud Bigtable for an application Consolidated Audit Trail (CAT), that will analyze over 100 billion financial market events and store over 30 petabytes of data in the coming years. They reached several conclusions regarding Bigtable's performance capabilities.

- Bigtable was able to handle the demands of the CAT application
- Scaling was linear with clusters of up to 300 nodes
- Data insertion scaled linearly for MapReduce jobs
- No tuning was needed to get sufficient performance from Bigtable

FIS found that Bigtable could write up to 2.7 Gigabytes per second and 10 Terabytes per hour and could process and insert 2.7 million FIX messages per second and 10 billion Fix messages per hour [12].

SECURITY

Security in Cloud Bigtable is at the cloud project level. If a user has access to a project, they have access to all tables within the project. Bigtable does not support security at the table, row, column, or cell level [5].

RELATED TECHNOLOGIES

Based on Bigtable

Multiple NoSQL databases have been built based on the specifications presented in the 2006 paper introducing Bigtable. Three important examples are Hbase, Hypertable and Accumulo.

Hbase [3] is the most well known database patterned after Bigtable, HBase is part of the Apache Big Data stack and "provides Bigtable-like capabilities on top of Hadoop and HDFS"[3]. Hbase is written in Java.

Hypertable [13] which is currently sponsored by Baidu, the Chinese search engine, was also inspired by Bigtable's design. It is written in C++.

Accumulo [14] developed by the National Security Agency and contributed to the Apache Software Foundation, extends Bigtable's data model with a new element called Column Visibility. Accumulo is written in Java.

Bigtable Alternatives

In addition to being a NoSQL database, Bigtable is classified as a wide column store. Popular wide column stores alternatives are Cassandra, HBase, Accumulo, Azure Table Storage, and Hypertable [15].

Bigtable is not well suited for all applications. Google recommends Bigtable for a applications that require "high throughput and scalability for non-structured key/value data, where each value is typically no larger than 10 MB" [5]. Additionally, Google recommends Bigtable for machine learning, stream processing/analytics, and MapReduce operations.[5]

For applications with other needs, Google recommends other databases in the Google suite [5]. For application needing online transaction processing (OLTP), Google recommends Google Cloud SQL. For applications requiring online analytical processing (OLAP), Google recommends Google BigQuery. For immutable blobs including images or movies greater than 10 MB, Google recommends Google Cloud Storage. Finally, for structured objects, SQL like queries, and ACID transactions, Google recommends Cloud Datastore.

USE CASES

Bigtable is used both internally by Google and externally as Cloud Bigtable by other companies. Use cases of each are discussed below.

Google Use Cases

Google uses Bigtable internally as the data store for many applications that deal with extremely large data volumes. While Google does not provide the proprietary implementation details of the Bigtable in these applications, their success handling large data volumes is evident. A partial list of applications that utilize Bigtable includes Web Search, Book Search, Search History, Analytics, Maps, Earth, Gmail, YouTube, and Blogger [1].

External Use Case - CAT Application

As discussed in the *Performance*, FIS Advanced Technology found that Cloud Bigtable was a viable technology for the extreme performance demands of the financial market auditing CAT system [12].

EDUCATIONAL MATERIAL

Three key resources exist for learning more about Bigtable. First is the paper *Bigtable: A Distributed Storage System for Structured Data* introducing Bigtable in 2006 [2]. It contains very detailed descriptions of Bigtable's storage model and architecture. Second is Google's documentation for Cloud Bigtable [16]. The documentation is current and covers all aspects of Cloud Bigtable. Finally, the GoogleCloudPlatform github repository contains many examples of how to use Cloud Bigtable. [17].

CONCLUSION

As the the storage layer for Google applications like Search, Gmail, and many others, Bigtable has been one of the most important database technologies in the Big Data Revolution. Based on Google's internal usage of Bigtable and the performance evaluation of Bigtable done by FIS Advanced Technology for the CAT system, it is clear that Bigtable provides excellent performance when processing large amounts of data. With Cloud Bigtable, this performance and scalability is now available to the public.

In addition to Bigtable's own impact, it has inspired other important open source NoSQL datastores, notably Hbase, Accumulo, and Hypertable. These technologies, particularly the Apache Software Foundations, Hbase, have become important players in their own right in the Big Data software stack.

REFERENCES

[1] Wikipedia, "Bigtable," Web Page, Jan. 2017, accessed 2017-01-29. [Online]. Available: https://en.wikipedia.org/wiki/Bigtable

[2] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, ser. OSDI '06. Berkeley, CA, USA: USENIX Association, 2006, pp. 205–218. [Online]. Available: http://dl.acm.org/citation.cfm?id=1298455.1298475

[3] Apache Software Foundation, "Welcome to apache hbase," Web Page, accessed 2017-02-21. [Online]. Available: https://hbase.apache.org/

[4] DB-Engines, "Google cloud bigtable," Web Page, accessed 2017-02-18. [Online]. Available: http://db-engines.com/en/system/Google+Cloud+Bigtable

[5] Google, "Overview of cloud bigtable," Web Page, accessed 2017-02-15. [Online]. Available: https://cloud.google.com/bigtable/docs/overview

[6] ——, "Apis & reference," Web Page, accessed 2017-02-20. [Online]. Available: https://cloud.google.com/bigtable/docs/apis

[7] ——, "nstalling the hbase shell for cloud bigtable," Web Page, accessed 2017-02-21. [Online]. Available: https://cloud.google.com/bigtable/docs/installing-hbase-shell

[8] Guru99, "Hbase shell and general commands," Web Page, accessed 2017-02-21. [Online]. Available: http://www.guru99.com/hbase-shell-general-commands.html

[9] Google, "cbt overview," Web Page, accessed 2017-02-24. [Online]. Available: https://cloud.google.com/bigtable/docs/go/cbt-overview

[10] ——, "Bigquery web ui," Web Page, accessed 2017-02-21. [Online]. Available: https://cloud.google.com/bigquery/bigquery-web-ui

[11] ——, "Cloud bigtable," Web Page, accessed 2017-01-29. [Online]. Available: https://cloud.google.com/bigtable/

[12] N. Palmer, M. Sherman, Y. Wang, and S. Just, "Scaling to build the consolidated audit trail: A financial services application of google cloud bigtable," FIS Advanced Technology, techreport, Dec. 2015, accessed 2017-02-224. [Online]. Available: https://cloud.google.com/bigtable/pdf/FISConsolidatedAuditTrail.pdf

[13] Wikipedia, "Hypertable," Web Page, Jan. 2017, accessed 2017-02-24. [Online]. Available: https://en.wikipedia.org/wiki/Hypertable

[14] ——, "Apache accumulo," Web Page, Oct. 2016, accessed 2017-02-24. [Online]. Available: https://en.wikipedia.org/wiki/Apache_Accumulo

[15] DB-Engines, "Db-engines ranking of wide column stores," Web Page, accessed 2017-02-15. [Online]. Available: http://db-engines.com/en/ranking/wide+column+store

[16] Google, "Cloud bigtable documentation," Web Page, accessed 2017-02-12. [Online]. Available: https://cloud.google.com/bigtable/docs/
[17] ——, "cloud-bigtable-examples," Code Repository, accessed 2017-02-24. [Online]. Available: https://github.com/GoogleCloudPlatform/cloud-bigtable-examples

AUTHOR BIOGRAPHY

Mark McCombe received his B.S. (Business Administration/Finance) and M.S. (Computer Information Systems) from Boston University. He is currently studying Data Science at Indiana University Bloomington.

WORK BREAKDOWN

All work on this paper was completed solely by Mark McCombe.

Apache Beam (Google Cloud Dataflow)

LEONARD MWANGI[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: lmwangi@iu.edu

March 2, 2017

Data has continued to grow in an exorbitant rate and consumers are now demanding real-time analytics for answers in order to make timely decisions, this has been challenging with batch-based systems due to unordered and unbounded dataset being generated and consumed thus requiring a paradigm shift to make it possible accommodate these datasets. This paper introduces Apache Beam previously Google Cloud Dataflow, a unified model for building data processing pipelines that handle both batch and stream processes for bound and unbound data.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524, Apache Beam, Google Cloud Dataflow

https://github.com/lmundia/sp17-i524/tree/master/paper1/S17-IO-3013/report.pdf

INTRODUCTION

In the last two decades, there has been a continuous data explosion in every organization causing them to rethink how to store and make it consumable so as to gain competitive edge without losing focus to the core business. This explosion is expected to continue accelerating with an estimated growth of 4300% by the year 2020 [1] thus becoming prudent for these organizations to identify solutions that will help keep this growth at bay and get actionable insights out of it. There are many solutions in the market that currently solve this growth and produce useful insights, MapReduce [2], Apache Spark [3] and Flink [4] being some of the leading ones though they have some major shortcomings. These technologies require either hardware upgrades [5] or rewriting pipelines to adopt engine-specific APIs which leads to throw away code especially when different stream or batch processing is involved. Google Cloud Dataflow offers an alternative to these technologies allowing you to run different types of analysis in a cost friendly manner. Cloud Dataflow is a fully managed service for creating data pipelines that ingest, transform and analyze data in both batch and stream mode [6]. Based on Millwheel [7] and Flume [8] technologies, it's posed as the successor of MapReduce and allows analysis of large volumes of data real-time in the cloud thus removing the need for deployment, maintaining and or scaling infrastructure. Cloud Dataflow has been submitted and accepted to Apache incubator, the project is now referred to as Apache Beam [9].

IMPLEMENTATION

Cloud Dataflow is language agnostic, it's first SDK was written in Java [10] but now available in Python [11] allows an entire pipeline to be written in a single program using intuitive Cloud Dataflow constructs to express application semantics [12]. The SDKs are portable allowing it to produce programs that can execute in many pluggable environments using "runners" which connect to the execution engines. At the moment, pluggable "runners" exist for Artisan, Apache Spark, single-node local execution runner by Google and Google hosted cloud Dataflow service execution engines

Figure 2

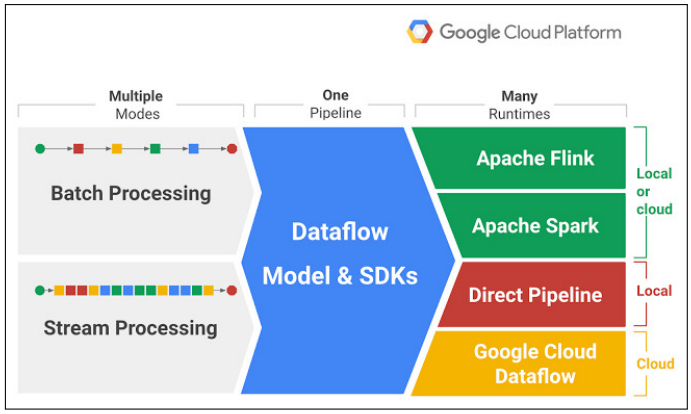


Fig. 1. Dataflow Execusion Engine [9]

When programing with Dataflow SDK, you essentially create a data processing job to be executed by one of the runner services. The model handles the low-level details like coordinating individual workers, sharding data sets amongst other tasks allowing focus to be on logical composition of data processing job.

Dataflow SDK

There are four major concepts in Dataflow SDK [13]:

- Pipelines – computation process that accepts data input from external sources, transforms it to provide some useful intelligence and produce some output data.
- PCollection–represents data in the pipeline, PCollection classes can represent virtually unlimited data set size.
- Transforms – it's the dataprocessing operation in the pipeline taking data from PCollection and producing output PCollection.
- I/O Sources and Sinks – provides data source and data sink APIs for pipeline I/O. Source API reads data into the pipeline and sink API writes output data from the pipeline. The source and sink represents root and endpoints of a pipeline.

In order to work with data in the pipeline, it has to be in form of PCollection. Each PCollection is owned by a specific pipeline object and only that Pipeline. PCollection has the following limitations:

- It's immutable, once created you cannot add, remove, or change individual elements
- It does not support random access to individual elements
- Cannot be shared between Pipeline objects

Dataflow SDK

As mentioned at the beginning of the paper, Cloud Dataflow was submitted for incubation to Apache and now the project is referred to as Apache Beam. Thus, in the example below Apache Beam is referenced.

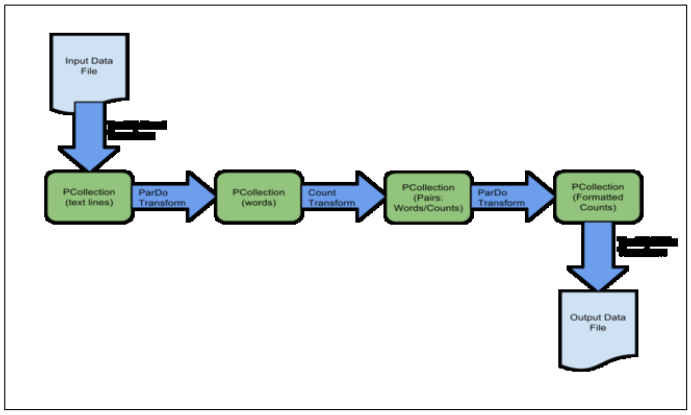


Fig. 2. Apache Beam dataflow [14]

Below is the processing pipeline code, which is accomplishable with a few lines of code.

RELATION TO BIG DATA

Collecting, transforming and analyzing big data in near real-time has become essential as form of getting instant feedback in order to solve customer needs or solve a problem quickly. Cloud Dataflow provides that capability through its real-time streaming platform. Cloud Dataflow allows processing of unbound, out-of-bound and global scale data [16].

```
# ... other imports ...
import google.cloud.dataflow as df

@df.typehints.with_output_types(df.typehints.Tuple[int,
  float])
def parse_sales_record(line):
  # Lines look like this:
# {"Timestamp": 1234.56, "Price": 10, "ProductName":
  "Name", "ProductID": 4}
record = json.loads(line)
return int(record['ProductID']), float(record['Price'])

p = df.Pipeline(...options...)
(p
| df.io.Read(df.io.TextFileSource('gs://SOMEBUCKET/
PATH/*.json'))
    | df.Map(parse_sales_record)
    | df.CombinePerKey(sum)
    | df.Map(lambda (product, value): {'ProductID':
      product,'Value': value})
    | df.io.Write(df.io.BigQuerySink('SOMEDATASET.
    SOMETABLE'
        schema='ProductID:INTEGER, Value:FLOAT',
        create_disposition=df.io.BigQueryDisposition.
        CREATE_IF_NEEDED,
        write_disposition=df.io.BigQueryDisposition.
        WRITE_TRUNCATE)))
```

Fig. 3. Processing Python Pipeline Code [15]

USE CASES

Financial Industry

With constant threats in financial industry, detecting and identifying anomalies in data flow is paramount to prevent fraud and financial crimes. By leveraging Cloud Dataflow real-time streaming, the industry can identify anomalies and notify necessary authorities for further investigations thus preventing catastrophic outcome.

Improve store layout

Sales are attributed to customers traffic, understanding the behavior of the customers when they are in a store and re-aligning to cater their needs helps increase sales. After Capturing these behaviors by use of RFIDs and QR code sensors, store owners can utilize Cloud Dataflow to analyze them in real-time and offer incentives like instant coupons to drive sales [17].

Sentiments Tracking

Every organization wants to know what their customers think about them and social media makes it easy for the customers to express themselves on how they feel about a brand. Collecting, quantifying and analyzing these sentiments becomes daunting task due to large amounts of data. Cloud Dataflow eases this task due to its ability to stream and analyze real-time data. Cloud Dataflow taps into social media outlets and analyzes the sentiments thus giving the organization a clear picture in real-time of what customers think and can also be used to re-align the marketing message for a better outcome.

ALTERNATIVE TECHNOLOGIES

Amazon Kinesis Stream

Amazon Kinesis Stream is an AWS data streaming offering that can capture and analyze data from different sources in real-time [18]. Using Kinesis Client Library (KCL) developers have the ability to write Amazon Kinesis powered applications that can generate and store data in other AWS offerings. A subscription is required in order to use Kinesis Stream. In comparison to Cloud Dataflow, Kinesis Stream has a limitation of 1MB/sec while Cloud Dataflow has a limitation of 10MB also Kinesis deployment locality is limited to regional while Dataflow is global [19].

Azure Stream Analytics (ASA)

Like Cloud Dataflow, Azure Stream Analytics is a fully managed real-time event processing engine capturing data from different sources and provide analytics [www-asa]. Stream Analytics is provided through Azure portal where analytics jobs can be authored. Stream Analytics connectivity is limited to Azure platform [www-asa] whereas Cloud Dataflow runners integrate with Flink, Spark and local runners for testing.

Apache Spark

Apache Spark also a competing solution to Cloud Dataflow, is a fast in-memory data processing engine with expressive development APIs that allow data workers to efficiently execute streaming, machine learning or SQL workloads [3]. The downside to Spark is that it's a batch-based processing framework [20] thus limiting true record-by-record processing also data arriving out-of-sequence possess a problem because it may be processed in the wrong batch.

CONCLUSION

Data growth is going to continue by staggering numbers and the consumers are becoming more aware of what they can accomplish with it thus demanding powerful platforms that can cater their needs as close to real-time as possible. Google Cloud Dataflow offers the solution to these needs by providing a stream and batch processing model thus not limiting the type of data being consumed. Making Cloud Dataflow available as an Apache Project increases reachability and enhances visibility to allow more runners to be incorporated to project. With more runners, it becomes cheaper and easier to migrate the existing on-premise and cloud solutions to Cloud Dataflow (Apache Beam) that may benefit from its processing capabilities. By handling the low-level details tasks in the system, Dataflow allows the developers to focus on the core processes of the pipeline which generates the desired performance, reliability and correctness. This makes Cloud Dataflow very attractive platform to handle big data needs.

ACKNOWLEDGEMENT

This research was done as part of course "I524: Big Data and Open Source Software Projects" at Indiana University. I thank Professor Gregor von Laszewski and associate instructors for their support throughout the course.

REFERENCES

[1] CSC, "The rapid growth of global data," PDF, 2013. [Online]. Available: http://assets1.csc.com/insights/downloads/CSC_Infographic_Big_Data.pdf

[2] Apache Hadoop, "Mapreduce tutorial," WebPage, 2013. [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

[3] S. Penchikala, "Big data processing with apache spark," *InfoQ*, no. 1, 2015. [Online]. Available: https://www.infoq.com/articles/apache-spark-introduction

[4] Apache Flink, "Introduction to apache flink," WebPage, 2017. [Online]. Available: https://ci.apache.org/projects/flink/flink-docs-release-1.2/index.html

[5] cloudera. (2017) Migrating from mapreduce (mrv1) to mapreduce (mrv2). S17-IO-3013. [Online]. Available: https://www.cloudera.com/documentation/enterprise/5-9-x/topics/cdh_ig_mapreduce_to_yarn_migrate.html#concept_zzt_smy_xl

[6] G. DeMichillie. (2014) Reimagining developer productivity and data analytics in the cloud - news from google io. Google. S17-IO-3013. [Online]. Available: https://cloudplatform.googleblog.com/2014/06/reimagining-developer-productivity-and-data-analytics-in-the-cloud-news-from-goog
html

[7] T. Akidau, A. Balikov, K. Bekiroglu, S. Chernyak, J. Haberman, R. Lax, S. McVeety, D. Mills, P. Nordstrom, and S. Whittle, "Millwheel: Fault-tolerant stream processing at internet scale," in *Very Large Data Bases*, 2013, pp. 734–746. [Online]. Available: https://research.google.com/pubs/pub41378.html

[8] Apache Flume. (2012) Apache flume. The Apache Software Foundation. [Online]. Available: https://flume.apache.org/

[9] F. Perry and J. Malone. (2016) Dataflow and open source - proposal to join the apache incubator. [Online]. Available: https://cloudplatform.googleblog.com/2016/01/Dataflow-and-open-source-proposal-to-join-the-Apache-Incubator.html

[10] Google, "Google cloud dataflow sdk for java," WebPage, Google Inc., 2016. [Online]. Available: https://cloud.google.com/dataflow/java-sdk/JavaDoc/

[11] Apache Beam. (2017) Apache beam python sdk. The Apache Software Foundation. [Online]. Available: https://beam.apache.org/documentation/sdks/python/

[12] F. Perry, "Sneak peek: Google cloud dataflow, a cloud-native data processing service," WebPage, Google Inc., 2014. [Online]. Available: https://cloudplatform.googleblog.com/2014/06/sneak-peek-google-cloud-dataflow-a-cloud-native-data-processing-service.html

[13] Google, "Dataflow programming model," Google, 2017. [Online]. Available: https://cloud.google.com/dataflow/model/programming-model

[14] ——, "Wordcount example pipeline," Web Page, 2017. [Online]. Available: https://cloud.google.com/dataflow/examples/wordcount-example

[15] S. Calinoiu, "Google announces cloud dataflow with python support," Web page, Google, 2016. [Online]. Available: https://cloud.google.com/blog/big-data/2016/03/google-announces-cloud-dataflow-with-python-support

[16] K. Knowles, "Stateful processing with apache beam," Web Page, 2017. [Online]. Available: https://beam.apache.org/blog/2017/02/13/stateful-processing.html

[17] P. Ciciora, "Study: Store layout an important variable for retailers," *NEWS BUREAU*, 2013. [Online]. Available: https://news.illinois.edu/news/13/0124storelayout_yunchuanliu.html

[18] Amazon. (2017) Amazon kinesis streams. [Online]. Available: https://aws.amazon.com/kinesis/streams/

[19] Google, "Google cloud platform for aws professionals: Big data," Web Page, Google, 2016. [Online]. Available: https://cloud.google.com/docs/compare/aws/big-data

[20] R. Beggs, "5 reasons why spark streaming batch processing of data streams is not stream processing," *SQL Stream*, 2015. [Online]. Available: http://sqlstream.com/2015/03/5-reasons-why-spark-streamings-batch-processing-of-data-streams-is-not-stream-p

Xen: A bare metal hypervisor

PIYUSH RAI[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: piyurai@iu.edu*
[+] *HID - S17-IO-3014*

March 9, 2017

Xen is an open-source baremetal hypervisor. This paper explores it's overview, architecture and contains a brief note on its alternatives.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Baremetal hypervisor, Xen

https://github.com/piyurai/sp17-i524/tree/master/paper1/S17-IO-3014/report.pdf

INTRODUCTION

To provide an isolated environment to different applications in terms of memory, process scheduling and disk access such that one process does not impact the other, used to be a major challenge for system administration [1]. Also, different applications may have different OS requirements. Running an application within its own OS environment should protect it from other malicious applications because of the guranteed memory and disk allocation to each guest VM (virtual machine).

Xen is a baremetal hypervisor and the only one to be available as open-source [2]. It's based on microkernel design with the advantage of small memory footprint and has a limited guest interface. It's responsibility is to manage CPU and memory, and to handle interrupts. Virtual machines are deployed in the guest domain called DomU which has no access privilege to hardware. A special virtual machine is deployed in the control domain called Domain 0. It contains hardware drivers and the toolstack to control the VMs and is the first VM to be deployed after the system comes up. The hypervisor is the first process to be started after bootloader.

ARCHITECTURAL OVERVIEW AND USE CASES

Xen has a small memory footprint and provides limited interface to guest VMs. It runs at the highest CPU privilege level while the guests are deployed in DomU domain [2]. It's primary responsibility is to manage CPU, memory and interrupts and does not contains the drivers for I/O functions such as networking and storage. The main device driver for a system can be run inside of a virtual machine. It protects the rest of the system from events such as driver crashes as the VM containing the driver can be rebooted independently. The control stack is generally deployed on a Linux VM running in domain 0. It is the first VM to be deployed by the system and can access the hardware directly. It handles system's I/O functions and provides interface to control the system. It's responsible for creating and configuring VMs, allocating resources and monitoring them and terminating the VMs when they are no longer required. The control stack interface can also be driven through a cloud orchestration stack such as OpenStack. Figure 1 shows the architectural overview of Xen.

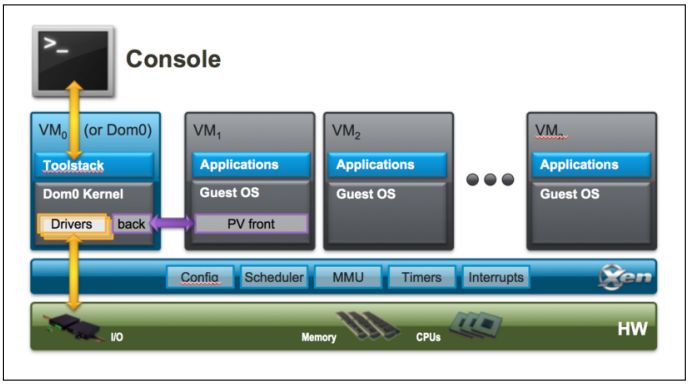


Fig. 1. Xen architecture overview [3].

The guest virtual machines are deployed in domain DomU which has no privilege to access the hardware. The guest VMs can either be deployed in Para virtualization (PV) mode or Hardware-assisted virtualization mode (HVM). It's also possible to use PV drivers inside of a HVM mode to improve its performance. The two types of modes can be deployed simultaneously on a single hypervisor. PV guests work without virtualization support from the hardware, and require PV-enabled kernel and drivers. The application binaries do not need any modification and run as in their native environments. HVM uses virtualization support available on the host CPU such as Intel VT or AMD-V hardware extensions. QEMU is used to emulate the remaining PC hardware. No changes to kernel are required for fully virtualized guests. HVM guests can also use PVHVM drivers i.e. special paravirtual device drivers optimized

for HVM environments to boost up the performance. PVH is a recently developed new virtualization mode where PV guests uses PV drivers for boot and I/O and hardware extensions otherwise. It is expected to simplify the Xen architecture and is the recommended virtualization mode to be used with newer Xen releases [4].

RESOURCES

The Xen Project wiki page presents an overview of the software architecture and host the links to dcocumentation and release notes for the different releases, e.g. the list of feature for release 4.8 can be found at [5]. David Chisnall has also witten a book explaining the hypervisor internals and it's important features along with their interfaces.

ALTERNATIVES

[6] talks about the limitations during the early days of Xen development like modification to the guest kernels and limited Linux support. The article here [7] states about later years into the development of Xen, when it remained based on the older version of Linux kernel as the developers waited for more recent kernels with support for modern hardware. This allowed KVM to overtake as preferred choice for virtualization. KVM overcame the problems faced by Xen by using the virtualization support in the newly available hardware and reusing the components from the Linux kernel. This gave it the advantage of automatically benefiting from the developments in the mainline kernel. Its incorporation into the Linux version 2.6.20 and contribution from companies like AMD and Intel increased KVM's adaptability among the Linux distributors.

KVM requires hardware assisted virualization with new releases containing paravirtualization support for certain devices [8].Xen was originally developed to overcome the problem of limited support for hardware assisted virtualization [4] by using Paravirtualization. Xen is a layer between the kernel and hardware whereas KVM turns the kernel itself into the hypervisor. Xen has it's own memory and power management system designed specifically for VMs where as KVM needs to have support for processes as well.

In his blog [9] Bredan Gregg has listed the observations made during comparisons of different virtualization technologies at Joyent using DTrace. It analyzed the stack trace for different system functionalities such as network and I/O calls stack trace along with the relevant overhead for Zones, Xen and KVM. It mentioned about Zones, with Linux Containers being its equivalent, to be the preferred choice for virualization for them as it has the lowest overhead for not having to run an entire OS for each application. They stopped using Xen in favor of KVM for provisioning fully virualized guest machines.

CURRENT STATUS AND THE FUTURE

Xen is actively maintained by Linux Foundation under the trademark "XEN Project". Some of the features included in the latest releases include "Reboot-free Live Patching" (to enable application of security patches without rebooting the system) and KCONFIG support (compilation support to create a lighter version for requirements such as embedded systems) [10]. The overheads in terms of passing data across several layers is being minimized by use of shared memory transports and buffers [9]. Xen Project also strives to minimize the foorptint of the change to Linux kernel. The codebase contains less than 150,000 lines of code [2]. Xen Project hosts a lisiting of vendors that uses Xen Project software to allow them to advertise their software and services. Xen is used by more than 10 million users [11] . Some of the major cloud hosting service providers using Xen include Amazon and Rackspace. Hypervisors are predictedd to become a core aspect of cloud based solutions. Containers are grwoing in popularity and there's an increasing interest in combination of isolation aspects of hypervisor and container runtime environments [12]. Bellani has witten a comprehensive blog [11] about the ongoing work within Xen community towards the development of hypervisor-based containers.

REFERENCES

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, ser. SOSP '03. New York, NY, USA: ACM, 2003, pp. 164–177. [Online]. Available: http://dl.acm.org/citation.cfm?id=945462

[2] "Xen Project Software Overview," Web Page, Mar. 2016. [Online]. Available: https://wiki.xenproject.org/wiki/Xen_Project_Software_Overview

[3] "Xen Architecture Diagram," Web Page, Apr. 2015. [Online]. Available: https://wiki.xenproject.org/wiki/File:Xen_Arch_Diagram.png

[4] David Chisnall, "Xen PVH," Web Page, Jun. 2014. [Online]. Available: http://www.informit.com/articles/article.aspx?p=2233978

[5] "Xen Project 4.8 Release Note," Web Page, Dec. 2016. [Online]. Available: https://blog.xenproject.org/2016/12/07/whats-new-with-xen-project-hypervisor-4-8/

[6] Amit Shah, "Ten years of KVM," Web Page, Nov. 2016. [Online]. Available: https://lwn.net/Articles/705160/

[7] Thorsten Leemhuis, "Rise of KVM," Web Page, Jun. 2011. [Online]. Available: http://www.h-online.com/open/features/Xen-lets-KVM-overtake-1262171.html

[8] "KVM - Wikipedia," Web Page, Feb. 2017. [Online]. Available: https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine

[9] Brendan Gregg, "Virtualization Performance: Zones, KVM, Xen," Web Page, Jan. 2013. [Online]. Available: http://dtrace.org/blogs/brendan/2013/01/11/virtualization-performance-zones-kvm-xen/

[10] "Xen Project 4.7 Feature List," Web Page, Jun. 2016. [Online]. Available: https://wiki.xenproject.org/wiki/Xen_Project_4.7_Feature_List

[11] Stefano Stabellini, James Bulpin, "Hypervisor in 2017," Web Page, Dec. 2016. [Online]. Available: goo.gl/S3X1oQ

[12] Stefano Stabellini, "Hypervisor-Based Containers," Web Page, Dec. 2016. [Online]. Available: https://thenewstack.io/hypervisors-container-era/

Apache Lucene

ROY CHOUDHURY, SABYASACHI[1,*,+]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: sabyroyc@indiana.edu*
[+] *HID - S17-IO-3015*

project-000, February 27, 2017

This paper gives an overview of Apache Lucene. We will go through the basic architecture and functionality of the library. We will see the advantages and downfalls of Lucene and conclude on its implementation.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Search Engine Library, Lucene

https://github.com/sabyasachi087/sp17-i524/tree/master/paper1/S17-IO-3015/report.pdf

INTRODUCTION

Apache Lucene is a search library that enables search facility to any application. Although it was initially written in Java but has been ported to many languages chiefly C-Sharp, c/c++ , Python etc. It is an active open source project and under Apache License. Its latest release is 6.4.1. It is important that we must understand that Lucene is just a search library and cannot handle other related stuffs like crawling, document filtering , administration etc.

CONCEPTS

To understand the purpose of Lucene we have to be familiar with two terms, one is "Information Overload" and "Information Retrieval"[1]. The term information overload means, difficulty that one can have in making decisions, because of the presence of too much of information. In another words, "Information Overload" can occur if the rate of feed/input into a system exceeds its processing capabilities. Imagine the situation of current world. We are living in the world where data has reached volume of zeta bytes. To extract some insight , first step is to collect all related data. This is known as information retrieval(IR). IR is the task of collecting relevant information from collection of data resources scattered across devices. Now with the above understanding we can say Lucene is a scalable Information Retrieval library.

Lucene and Search Engine

Google has set some base expectations for search engines, which if not available can cause dissatisfaction for the users. For example spell checker and response time of <= 1 second. Lucene should be able to met those expectations or else its not worth. But Lucene is not a full fledged search engine rather its a tool kit to achieve so. So lets jot down the steps for any search engine

Module : Raw Content -> Gather Data -> Analyze -> Index -> Query Support UseCase : User query -> Search Engine -> Build Query -> Extract Information from indexed data

ARCHITECTURE

Figure 1 explains a basic architecture of Lucene

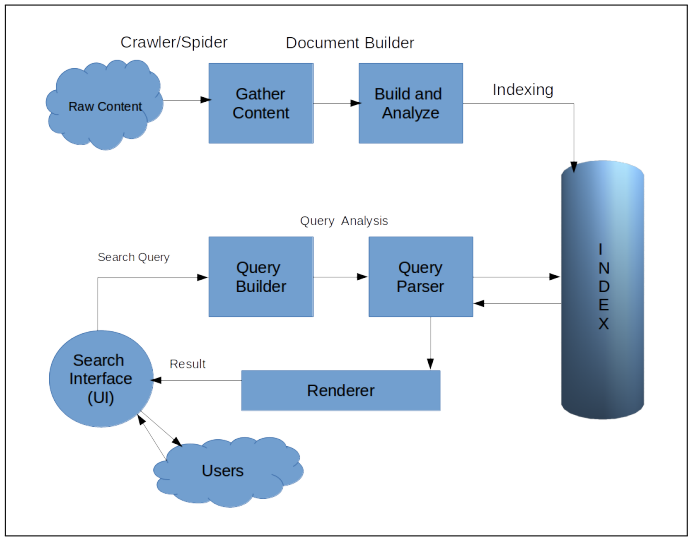


Fig. 1. Apache Lucene Architecture.

Lucene library is compact and does not have any external dependencies. But it can be plugged with other libraries for building a search engine. Fig 1[2] shows process flow of Lucene. On a high level , data collected from different sources are analyzed and converted into smaller chunks. These are known as documents. Documents are text entries and from these text

entries Lucene performs indexing and store it in local disk for future reference. The next step is to handle the search query from users. Lucene has a query parser to understand the query and search the index for the correct or relevant match. If found, returns the document back. We will elaborate the flow beneath.

LUCENE COMPONENTS

Document Analysis and Indexing

Data or contents which are available in different format and location needs to be gathered. This process is typically done by a crawler/spider. Core Lucene does not have these capabilities and can be considered to be a pre-requisite for Lucene to implement.Two of the crawler that are build on Lucene are Solr and Nutch. Once the data is collected , document has to be constructed from the contents. The design of constructing documents has to be decided by the user and implemented within Lucene. Lucene provides an API for building documents but logic has to be provided by the implementation layer. Lucene also does not provide any API for document filtering. But yet again we have Tika , which is built on Lucene, can be used for this purpose. But we cannot index the document yet. We cannot index the raw content within the document directly. Before that , we need to break the content into smaller chunks known as tokens. Each token is map to a "word". Analysis includes handling compound words, spell check, typo correction injection of synonyms, etc. Lucene has built in support of list of analyzers which gives a fine grain control over analysis. Once tokenization is done , now its time for indexing. Lucene takes care all of the need to cater this step. An API has been provide for this purpose, but has to be implemented carefully as the searching will solely depend upon how well the indexing has been done.

Searching

It is a look up process within the index to extract the most relevant (matching) documents. It is based on two matrices i.e. Precision and Recall. Recall measures how well the system finds the relevant documents and Precision measures the filtering out the irrelevant one. Lucene offers benchmarking technique for measuring these matrices. User Interface (UI) is equally important for a search application as that is what the end user is going to use. Lucene does not provide any UI support. When user inputs the search query the first step is to build the query. User inputs are human readable and need further processing before it can be used within the application. Lucene provides a powerful parser for this job known as QueryParser. Even it is default state it does the job pretty clean but often needs extension as per some advance requirements. Finally hitting the search query to index. Almost everything about this is catered by Lucene and also provide option for extension. It finds the result and returns all relevant document objects. Its the responsibility of the UI to render the results correctly.

ADVANCE USAGES

Lucene has some advance feature for administration and analytics. For example Lucene allows to configure the RAM buffer size , re-indexing ,commit and purge scheduler. It allows some fault tolerance mechanism in case a newly added document failed to index. Related to analytics it also provides some meta information regarding the search queries it receives and the results it renders. For example which kind of query are run , query hitting lowest relevance , query having no results and so on. One big problem within the list of advance usages that Lucene does not support is "Scaling". Scaling in terms of both through put and processing speed. In a clustered environment this is quiet an important part as data and resource all are distributed. But both Solr and Nutch provides data partitioning and sharding to achieve higher throughput if not speed. Elastic search is another option which is based upon Lucene and provides distributed computing.

CONCLUSION

Lucene is the standard library for search applications. It can be compared with 'C' (language) of computing which is small and powerful but requires much more effort to build an entire application. It must be remembered that Lucene is just a library which sits at the core of the functionality but it needs much more than that to build an application. Elastic Search , Solr and Nutch which are based on Lucene are preferred tools in terms of building enterprise level search engines.

READING SOURCES

- Lucene In Action [2] provides a startup guide to learn, build and implement search engines based on Lucene
- Lucene at tutorial point [3] provides introduction to Lucene Library.

ACKNOWLEDGEMENTS

Thanking Prof. Gregor von Laszewski for his technical help and support.

REFERENCES

[1] "Wikipedia," Web page. [Online]. Available: https://en.wikipedia.org/wiki/Information_retrieval
[2] O. G. Erik Hatcher, *Lucene In Action*, 2nd ed. Manning, 2004.
[3] "Tutorialpoint," Web page. [Online]. Available: https://www.tutorialspoint.com/lucene/index.htm

CoreOS

RIBKA RUFAEL[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: rrufael@umail.iu.edu HID: S17-IO-3016

paper-001, February 25, 2017

CoreOS is a minimal Linux operating system that allows application to run on containers. CoreOS Linux is an operating system designed for container cluster frameworks.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: CoreOS, Container, cloud, Linux

https://github.com/cloudmesh/sp17-i524/blob/master/paper1/S17-IO-3016/report.pdf

1. INTRODUCTION

CoreOS [1] is a light weight linux operating system that is designed to be used for container infrastructure. CoreOS allows applications to run on containers so that there is abstraction layer between applications and the operating system. The separation of applications and operating system allows to avoid dependencies. CoreOS can be run on clouds, virtual or physical servers. CoreOS allows the ability for automatic software updates in-order to make sure containers in cluster are secure and reliable. It also makes managing large cluster environments easier. One of the differences between CoreOS Linux and traditional Linux distribution is that, in the case of traditional Linux distribution operating system, utilities and software are puts together but in the case of CoreOS Linux only operating system and utilities are bundled together. In CoreOS Linux applications and software are run on containers. Figure 1 shows the CoreOS Linux layout:

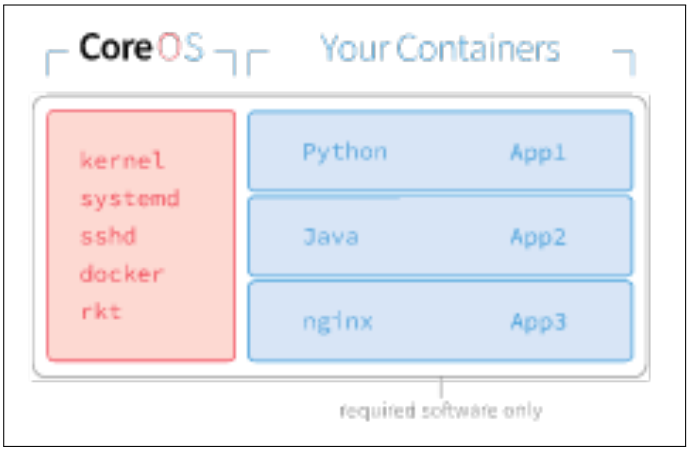


Fig. 1. CoreOS container layout. [1]

The company that provides CoreOS Linux also provides open source tools like etcd, rkt and flannel. CoreOS also has commercial products Kubernetes and CoreOS stack. In CoreOS linux service discovery is achieved by etcd, applications are run on Docker and process management is achieved by fleet.

Big data projects in science or business involve processing of large amount of data. These big data projects are hosted on bare metal servers or on clouds. Big data projects in science and in business sector can benefit from container frameworks that provide flexibility, ease of deployment, adding or removing container clusters based on demand. [2] Since CoreOS Linux is designed to be used in container frameworks, it makes it one of the candidates to be included into big data projects software stack that has container cluster infrastructure.

2. COREOS ARCHITECTURE AND INSTALLATION

CoreOS linux is a minimal operating system where the OS and utilities are one unit and applications are run on containers. CoreOS has 9 disk partitions. The disk partions are: 1) EFI-SYSTEM which contains the bootloader and the partition type is VFAT. 2) BIOS-BOOT, ROOT-C and the 7th partion these partions have no partion format and are reserved for future use. 3) USR-A and USR- B are the active and passive partions that container linux sits on. Only one of them can be active at a time and partion type is EXT4 depending on which one is active. 4) OEM has EXT4 as partition type and this is where OEM platform related configurations like custom networking and running an agent are stored. 5) OEM-CONFIG serves as an alternative place for an OEM. 6) ROOT may have EXT4, BTRFS, or XFS as partion type and it is used for keeping data which is persistent and this partion is stateful. [1]

CoreOS Linux comes bundled with etcd, Fleet and Docker. Service discovery in CoreOS Linux is achieved by etcd. Key value store on etcd is distributed and stored on all machines that run CoreOS Linux. This service discovery capability makes it easy to add or remove machines. There is a command line interface that comes preinstalled on CoreOS linux called etcdctl

that can be used to change and get key value data from etcd using etcdctl set and etcdctl get respectively. Another command that can be used for setting and reading key value is curl.

Container management in CoreOS linux is made possible by Docker. All applications run on Docker. Containers can be launched using docker run command line interface.

Fleet is the third component of CoreOS and it is used for management of containers with Docker installed. Fleet is init system and fleetctl command line interface can be used to check the status of containers, start and stop containers.[3]

Based on the information in this book [4], there is no package manager for installing, upgrading, configuring softwares in CoreOS . Softwares are installed as containers in CoreOS operating system. Inorder to apply updates, CoreOS makes use of two root partitions one is active and the other is passive.First the updates to CoreOS are applied to passive partion and upon reboot all the updates are applied to the active partition.

CoreOS can be installed on clouds from EC2, Rackspace, GCE or virtual machines such as Vagrant, VMware, OpenStack or on physical servers such as PXE, iPXE, ISO. [3]Based on the information provided on CoreOS site [5], CoreOS container can be setup using Vagrant. Vagrant, virtual machine manager, can be used to run CoreOS on a single machine with Windows, OS X or Linux operating systems. It is recommened to have Vagrant version 1.6.3 or latest. Virtual machines by VirtualBox or VMware are both supported by Vagrant.

3. LICENSING

CoreOS Container Linux is an open source operating system . It is comprised of other programs and documents developed by other individuals and companies. All original components that are part of CoreOS Container Linux are licensed under Apache 2.0. The latest version of CoreOS Container Linux is 1325.1.0. [1]

4. PERFORMANCE

In a study done by Purdue University [2] , performance tests between CoreOS 899.5.0 on Docker and Red Hat Enterprise Linux 6.5 were performed. In their study, the tests were run on 24-core AMD Opteron systems, with 2.1 GHz processors, 48 GB memory and 10Gbps Ethernet connection. For HPL performance tests the results were "The native RHEL 6.5 system showed an average performance of 7.839 GFLOPS, versus 7.811 GFLOPS in a containerized environment (approximately a 0.4% slowdown). " The netwrok throughput iperf was used and the results were: "final upload throughput was 8.43 Gbps under Docker (versus 8.26 Gbps in RHEL 6.5), with a download throughput of 9.37 Gbps (versus 9.38 Gbps in RHEL 6.5). " Network File System throughput was done using iozone and results are shown in Figure 2:

In another study done in Institute of Informatics – Federal University of Rio Grande do Sul [6], performance analysis of ClickOS, CoreOS and OS^v was performed. "NFV is a new networking paradigm where functions (e.g., firewalls, DNS, IDS), traditionally performed by dedicated physical devices, are virtualized and deployed on commodity hardware. " In their paper comarision of ClickOS, CoreOS and OS^v as NFV based tools was performed. They comapred boot time, response time and memory consumption for the three virtualization technologies. As per their evaluation result, ClickOS has the lowest boot time and response time followed by CoreOS and with regards to memory cosumption, CoreOS has the smallest memory usage followed

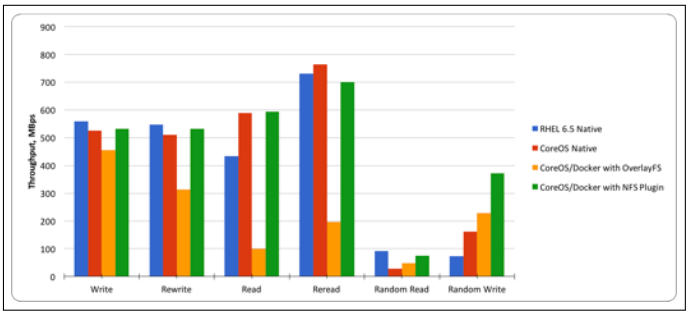


Fig. 2. File Server throughput. [2]

by ClickOS. Based on the result OS^v has the slowest boot time and response time as compared to CoreOS and ClickOS. Also for memory consumption, OS^v has the highest cosumption as compared to the other two technologies used.

5. USE CASES

In this section use cases from big data and other areas where CoreOS Linux is used are discused.

5.1. Use Cases for Big Data

According to [7] , Metagenomics RAST server(MG-RAST) is free access portal that can be used by researchers for accessing and analyzing metagenomics data. "Random community genomes (metagenomes) are now commonly used to study microbes in different environments. " As described on [8], CoreOS and fleet are used on MG-RAST container application servers. Researchers can input data into MG-RAST portal through script, web site or REST API.

5.2. Other Use Cases

According to the paper [9], CoreOS container linux on Amazon EC2 cloud was used to implement the project Two-stage Stochastic Programming Resource Allocator (2SPRA). The language used to implement was Python. In this project,they try to address the problem that exist in most datacenters of over provisioning resources in order to achieve performance service level objective. 2SPRA is a resource allocation scheme and it is able to optimize resource allocation for containerized web services based on varying workloads. 2SPRA analyses the relationship between change in workload, resource allocation and response latency inorder to calculate the the number of containers needed. In this experimental work, CoreOS was used inorder to simulate real word scenarios of n tier application servers running on containers. The test architecture has client Java based emulator which creates multiple user sessions at the same time, web hosting platform with where RUBis benchmark is installed on Virtual machine with CoreOS version stable r717.3.0 and the third component is 2SPRA implemented in Python running on Virtual machine.

6. EDUCATIONAL MATERIAL

CoreOS website [3] has detailed documentation and materials for anyone who is interested in setting up CoreOS Container linux on clouds, virtual or physical servers. Users who are interested can contribute to the CoreOS open source projects through github [10].

This book [4] also has information about CoreOS overview and its installation.

7. CONCLUSION

CoreOS Linux is a light weight Linux based operating system that is designed for containers. It provides abstraction by separating the operating system from application and softwares. Operating system updates to CoreOS are automatic without a need for user interaction. CoreOS can be installed on clouds from EC2, Rackspace, GCE or virtual machines such as Vagrant, VMware, OpenStack or on physical servers such as PXE, iPXE, ISO.

CoreOS Linux makes applications and microservices running on containers secure, easy to deploy and portable. Big data projects can leverage these benefits that comes with CoreOS Linux by adding it to their infrastructure.

As previous performance results has shown [6], CoreOS Linux has the lowest memory usage compared to other operating systems namely ClickOS and OS^{v}.

ACKNOWLEDGEMENTS

The author would like to thank Professor Gregor von Laszewski and associate instructors for their help and guidance.

REFERENCES

[1] CoreOS, "Why CoreOS," Web Page, Jan. 2017, accessed: 2017-01-23. [Online]. Available: https://coreos.com/why/

[2] S. Julian, M. Shuey, and S. Cook, "Containers in Research: Initial Experiences with Lightweight Infrastructure," in *Proceedings of the XSEDE16 Conference on Diversity, Big Data, and Science at Scale*, ser. XSEDE16. New York, NY, USA: ACM, 2016, pp. 25:1–25:6. [Online]. Available: http://doi.acm.org.proxyiub.uits.iu.edu/10.1145/2949550.2949562

[3] CoreOS, "CoreOS Quick Start," Web Page, Feb. 2017, accessed: 2017-02-17. [Online]. Available: https://coreos.com/os/docs/latest/quickstart.html

[4] R. Mocevicius, *CoreOS Essentials*. Packt Publishing Ltd, Jun. 2015.

[5] CoreOS, "Vagrant," Web Page, Feb. 2017, accessed: 2017-02-17. [Online]. Available: https://coreos.com/os/docs/latest/booting-on-vagrant.html

[6] L. Bondan, C. R. P. dos Santos, and L. Z. Granville, "Comparing virtualization solutions for NFV deployment: A network management perspective," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, Jun. 2016, pp. 669–674.

[7] F. Meyer, D. Paarmann, M. D'Souza, R. Olson, E. M. Glass, M. Kubal, T. Paczian, A. Rodriguez, R. Stevens, A. Wilke *et al.*, "The metagenomics RAST server–a public resource for the automatic phylogenetic and functional analysis of metagenomes," *BMC Bioinformatics*, vol. 9, no. 1, p. 386, 2008. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-9-386

[8] A. Wilke, J. Bischof, W. Gerlach, E. Glass, T. Harrison, K. P. Keegan, T. Paczian, W. L. Trimble, S. Bagchi, A. Grama, S. Chaterji, and F. Meyer, "The MG-RAST metagenomics database and portal in 2015," vol. 44, no. D1, pp. D590–D594, 2016. [Online]. Available: http://dx.doi.org/10.1093/nar/gkv1322

[9] O. Adam, Y. C. Lee, and A. Zomaya, "Stochastic Resource Provisioning for Containerized Multi-Tier Web Services in Clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2016.

[10] CoreOS, "CoreOS," Web Page, Jan. 2017, accessed: 2017-02-26. [Online]. Available: https://github.com/coreos/

MongoDB

NANDITA SATHE[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding author: nsathe@iu.edu*

paper-1, February 27, 2017

The data that Internet and various gadgets generate today has increased manifold than yesterday. This data is voluminous, complex and un-structured. This arises the need of having a flexible, scalable, and robust database that will not only store this data but also provide it as required in lightening speed. Many NoSQL databases came into existence. Out of them, MongoDB is the fastest-growing database ecosystem. It is the next-generation database that helps businesses transform their industries by harnessing the power of data.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524

https://github.com/nsathe/sp17-i524/blob/master/paper1/S17-IO-3017/report.pdf

1. INTRODUCTION

MongoDB is a non-RDBMS key-value data store. The data to be stored does not necessarily have to follow a fixed schema. In relational database data is stored primarily in tables, whereas in MongoDB data is stored in collections which act as container for a Document. A Document can be compared to a row of a table. The data in MongoDB Document is stored in JSON array like data structure. The database also supports large volume of data storage and offers very high data insert speed due to its schema-less design. The database system can be used for Big Data applications where volume of incoming data is huge and of varied structure.

2. INFRASTRUCTURE AND PERFORMANCE CONSIDERATIONS

Capacity planning is the most important thing when deciding about production deployments in every environment. Applications may change their percentage of writes versus reads over time. An increase in users typically lead to more queries and a larger working set.

System performance and capacity planning are two important topics that should be addressed in any deployment. The planning should involve establishing baselines on data volume, system load, performance (throughput and latency), and capacity utilization. These baselines should reflect the workloads you expect the database to perform in production, and they should be revisited periodically as the number of users, application features, performance SLA, or other factors change. Baselines will help you understand when the system is operating as designed, and when issues begin to emerge that may affect the quality of the user experience or other factors critical to the system. The following section discusses key deployment considerations, including hardware, scaling and HA. The section also discusses what you need to monitor to maintain optimum system performance.

2.1. Determine working set size

[1] When prioritizing hardware budget for MongoDB deployments, RAM should be at or near the top of the list.

MongoDB makes extensive use of RAM for low latency database operations. In MongoDB, all data is read and manipulated through memory-mapped files. Reading data from memory is measured in nanoseconds and reading data from disk is measured in milliseconds; and so reading from memory is approximately 100,000 times faster than reading from disk.

The set of data and indexes that are accessed most frequently during normal operations is called the working set, which ideally should fit in RAM. It may be the case that the working set represents a fraction of the entire database, such as applications where data related to recent events or popular products is accessed most commonly.

Page faults occur when MongoDB attempts to access data that has not been loaded in RAM. If there is free memory then the operating system will locate the page on disk and load it into memory directly. However, if there is no free memory the operating system must write a page that is in memory to disk and then read the requested page into memory. This process will be slower than accessing data that is already in memory.

Some operations may inadvertently purge a large percentage of the working set from memory, which adversely affects performance. For example, a query that scans all Documents in the database, where the database is larger than the RAM on the server, will cause documents to be read into memory and

the working set to be written out to disk. Ensuring you have defined appropriate index coverage for your queries during the schema design phase of the project will minimize the risk of this happening. The MongoDB **explain** method can be used to provide information on your query plan and indexes used.

A useful output resulted from the MongoDB's **serverStatus** command is a workingSet document that provides an estimated size of the MongoDB instance's working set. Database Administration team can track the number of pages accessed by the instance over a given period, and the elapsed time from the oldest to newest document in the working set. By tracking these metrics, it is possible to detect when the working set is approaching current RAM limits and proactively take action to ensure the system is scaled.

MongoDB Management Service and **mongostat** command help to monitor memory usage and are discussed in detail below.

2.2. Storage and Disk I/O

While your MongoDB system should be designed so that its working set fits in memory, disk I/O is still a key performance consideration. MongoDB regularly flushes writes to disk and commits to the journal, so under heavy write load, the underlying disk subsystem may become overwhelmed. The **iostat** command can be used to show high disk utilization and excessive queuing for writes.

2.3. CPU Selection – Speed or Cores?

MongoDB performance is typically not CPU-bound. As MongoDB rarely encounters workloads and is able to leverage large numbers of cores, it is preferable to have servers with faster clock speeds than numerous cores with slower clock speeds.

2.4. Scaling your Database

MongoDB provides horizontal scale-out for databases using a technique called sharding. Sharding distributes data across multiple physical partitions called shards. Sharding allows MongoDB deployments to address the hardware limitations of a single server, such as bottlenecks in RAM or disk I/O, without adding complexity to the application.

Figure 1 shows scaling of database.

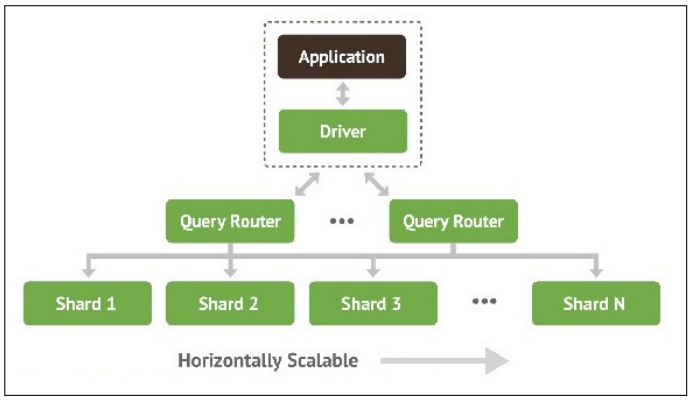


Fig. 1. MongoDB database scaling.

2.5. MongoDB Auto-Sharding with Application Transparency

It is far easier to implement sharding before the resources of the system become limited. That is why capacity planning and proactive monitoring are important elements in successfully scaling the application

Users should consider deploying a sharded MongoDB cluster in the following situations:

RAM Limitation: The size of the system's active working set will soon exceed the capacity of the maximum amount of RAM in the system.

Disk I/O Limitation: The system has a large amount of write activity, and the operating system cannot write data fast enough to meet demand; and/or I/O bandwidth limits how fast the writes can be flushed to disk.

Storage Limitation: The data set approaches or exceeds the storage capacity of a single node in the system.

One of the goals of sharding is to uniformly distribute data across multiple servers. If the utilization of server resources is not approximately equal there may be an underlying issue that is problematic for the deployment. For example, a poorly selected shard key can result in uneven data distribution. In this case, most if not all of the queries will be directed to the single mongodb that is managing the data. Furthermore, MongoDB may be attempting to redistribute the documents to achieve a more ideal balance across the servers. While redistribution will eventually result in a more desirable distribution of documents, there is substantial work associated with re-balancing the data and this activity itself may interfere with achieving the desired performance SLA. By running **db.currentOp()** you will be able to determine what work is currently being performed by the cluster, including re-balancing of documents across the shards. In order to ensure data is evenly distributed across all shards in a cluster, it is important to select a good shard key.

2.6. High Availability with MongoDB Replica Sets

MongoDB uses its native replication to maintain multiple copies of data across replica sets. Replica sets help prevent downtime by detecting failures (server, network, OS or database) and automatically initiating failover. It is recommended that all MongoDB deployments should be configured with replication.

Figure 2 shows replica sets maintained by MongoDB.

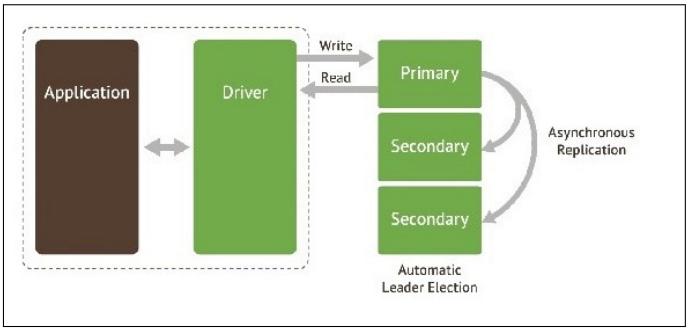


Fig. 2. MongoDB replica sets.

3. FEATURES

These are main features of MongoDB.[2]

- General purpose database, almost as fast as the key:value NoSQL type.
- High availability.

- Scalability: (from a standalone server to distributed architectures of huge clusters). This allows us to shard our database transparently across all our shards. This increases the performance of our data processing.
- Aggregation: batch data processing and aggregate calculations using native MongoDB operations.
- Load Balancing: automatic data movement across different shards for load balancing. The balancer decides when to migrate the data and the destination Shard, so they are evenly distributed among all servers in the cluster. Each shard stores the data for a selected range of our collection according to a partition key.
- Native Replication: syncing data across all the servers at the replica set.
- Security: authentication, authorization, etc.
- Advanced users management.
- Automatic failover: automatic election of a new primary when it has gone down.
- Zero downtime upgrades.
- There are no bottlenecks processing large volumes of data.
- MongoDB uses JSON objects to store and transmit information.
- We can do queries and geospatial operations in 2D and 3D.
- We can utilize Map-Reduce for information processing using JavaScript functions at the server side.
- JavaScript execution: Ability to store JavaScript functions on the server for queries and aggregation functions
- MongoDB Management Service. (MMS) is a powerful web tool that allows us tracking our databases and our machines and also backing up our data.
- Monitoring:
    1. MMS tracks the database and hardware metrics for managing a MongoDB deployment
    2. Performance is visualized in a rich web console to help you optimize your deployment
    3. Custom alerts: Discover issues before your MongoDB instance will be affected
- Backup
    1. Continuous backup with point-in-time recovery of replica sets and sharded clusters
    2. Multiple copies of every backup are archived across data centers (geographically distributed and fault-tolerant)

4. MONGODB FOR BIGDATA ANALYTICS

At the top level there are 3 Vs that define BigData [3]

Volume:

MongoDB supports storage of high volume of data which is complex in nature. The system is designed in such a way that for larger data loads the data can be distributed across multiple clusters, providing new levels of availability and scalability with no downtime.

Variety:

Many a times it so happens that the incoming data is unknown, not adhering to specific schema or sometimes data is ever evolving. This brings in many challenges to the developers as to how to store such data. MongoDB's dynamic schema provides a major advantage for businesses that need to ingest, store, and process rapidly evolving data streams from new sources.

Velocity:

The term velocity refers to high and volatile inbound data, faster query/read operations at low latency. MongoDB by default prefers high insert rates over transaction safety so insertion of high volume of data takes less time. The read operations on the other hand are also optimized by keeping frequently used data sets in memory rather reading from disk IO which is expensive.

4.1. Data Analytics

[4] In several scenarios the built-in aggregation functionality provided by MongoDB is sufficient for analyzing your data. However in certain cases, significantly more complex data aggregation may be necessary. This is where Hadoop and Apache Spark can provide a powerful framework for complex analytics.

In this scenario data is pulled from MongoDB and complex processing is performed in Hadoop/Spark, output of the processing can then be written back to MongoDB for later querying and ad-hoc analysis.

To be able to connect efficiently to MongoDB Hadoop and Spark connectors are available and can be configured easily.

The analytics can also be performed in python library **monary** which its developers are claiming to be superfast as opposed to **pymongo** driver.

5. REAL LIFE APPLICATION-PROJECTS USING MONGODB

This section lists some of the prominent companies currently using MongoDB. We will also see why MongoDB is becoming first choice for big data storage.

5.1. Addressing Problems using MongoDB

MongoDB can be used in situations where the volume, complexity, velocity of incoming data is huge due to the underlying nature of data inserts supported by the database system. Analysis of geospatial data becomes easy with MongoDB as it natively supports geospatial queries by which patterns can be identified in neighbourhood making it easy for organisations and government establishments to detect activities proactively.

The database system can be put to analyze and condense larger volume of data into a meaningful result by applying map-reduce mechanism which can efficiently scan through each contained document and return aggregated data. [5]

MongoDB supports horizontal scaling by which data can be scattered across different nodes in turn reducing the hardware cost as multiple machines can act as a single server, the process

is called shard clusters. This way organizations need not have to spend heavily on in fracture costs by way of using lower hardware configuration systems as shards.

[6] The system is also suitable for deployment to cloud computing platforms, since they support automatic replication and scaling your system to respond to higher user load by adding replicas. Since, they support sharding, you can easily scale up the system to influx of new data by adding shards.

5.2. Use Cases

MongoDB is considered as the next-generation database that helps businesses transform their industries by harnessing the power of data. The world's most sophisticated organizations, from cutting-edge startups to the largest companies, use MongoDB to create applications never before possible at a fraction of the cost of legacy databases. MongoDB is the fastest-growing database ecosystem, with over 9 million downloads, thousands of customers, and over 750 technology and service partners. [7]

[8] **Barclays:** Leading consumer, corporate and investment bank replaces three decades of relational databases with MongoDB, for increased agility, scalability, and cost-efficiencies.

Adobe: World's 1 Content Management solution relies on MongoDB for petabyte scale data management in the cloud.

Metlife: One of the largest global providers of insurance builds a single-view of 100M+ customers across 70 systems in just 90 days.

Viacom: Viacom built a data collection tool that integrates into 100+ sites, handles 15,000 1K writes/second, and has nearly zero downtime with MongoDB.

Pearson: Pearson, the global online education leader, has a simple yet grand mission: to educate the world; to have 1 billion students around the globe touching their content on a regular basis. The company has been able to leverage MongoDB for use cases such as:

- Identity and access management for 120 million user accounts, with nearly 50 million per day at peak;

- Adaptive learning and analytics to detect, in near real-time, what content is most effective and identify areas for improvement

Medtronic: Medical equipment maker Medtronic, last year served 9 million patients and this year the company announced that it serves a patient in some way every three seconds. In addition, Medtronic collects more than 30 million data samples about its devices every day.

Bosch: The massive volume and increasingly unstructured nature of IoT data has put new demands on Bosch SI's entire technology stack, especially the underlying database. Rigidly defined RDBMS data models have limited use in IoT. They lack the flexibility, scale and real-time analytics needed to quickly capture, share, process and analyze IoT data. IoT calls for a new mindset, and a new database. MongoDB helped Bosch SI reimagine what's possible. Here's how:

Manage complex data types. IoT data arrives at higher speeds, in greater volumes and variability of structure. MongoDB can easily handle the full spectrum of data: structured, semi-structured, unstructured. Efficient modelling of data using JSON makes it easy to map the information model of the device to its associated document in the database.

Support continuous innovation and business agility. Changes in IoT customer requirements, standards and use cases will require frequent data model changes. MongoDB's dynamic schema supports agile, iterative development methodologies and makes it simple to evolve an app. Adding new devices, sensors and assets is straightforward, even when you're dealing with multiple versions in the field concurrently. Instead of wasting time dealing with the mismatch between programming language and the database, MongoDB lets developers focus on creating rich, functional apps.

Create a unified view. Creating a single view of an asset or customer with a relational database is complicated. Source schema changes require additional changes to the single view schema. MongoDB makes it easy to aggregate multiple views of related data from different source systems into one unified view.

Power operational insight with real-time analysis. Apps handling fast-moving IoT data can't wait on ETL processes to replicate data to a data warehouse. They need to react and respond in real time. MongoDB's rich indexing and querying capabilities – including secondary, geospatial and text search indexes, the Aggregation Framework and native MapReduce – allow users to ask complex questions of the data, leading to real-time operational insight and business discovery.

Be enterprise-ready. MongoDB complements agility with enterprise-grade availability, security and scalability. Zero downtime with replica sets. Proven database security with authentication, authorization, auditing and encryption. Cost-effective scale-out across commodity hardware with auto-sharding. As IoT data volumes continue to explode, Bosch will be able to efficiently scale without imposing additional complexity on development teams or additional cost on the business.

6. COMPARISON

There are various alternatives for MongoDB which are good in their own arena, here we are going to compare 3 most widely used databases Cassandra, Hbase MongoDB. Please refer Table 1.

7. EDUCATIONAL MATERIAL

To get started on learning MongoDB, following resources can prove helpful.

- Helpful tutorials for beginner developers - https://www.tutorialspoint.com/mongodb/
- MongoDB's documents. You will find reference material for everything here - https://docs.mongodb.com/v3.2/getting-started
- Fixed schedule course on MongoDB - https://university.mongodb.com/
- MongoDB course specifically for Data Scientists - https://www.udacity.com/course/data-wrangling-with-mongodb–ud032

REFERENCES

[1] Mat Keep, "Preparing for your first mongodb deployment: Capacity planning and monitoring," Web Page, Oct 2013. [Online]. Available: https://www.infoq.com/articles/mongodb-deployment-monitoring

[2] Cesar Trigo Esteban, "Mongodb: Characteristics and future," Web Page, Aug 2014. [Online]. Available: http://www.mongodbspain.com/en/2014/08/17/mongodb-characteristics-future/

[3] MongoDB, Inc, "Real-time analytics," Web Page. [Online]. Available: https://www.mongodb.com/use-cases/real-time-analytics

[4] ——, "Hadoop and mongodb use cases," Web Page. [Online]. Available: https://docs.mongodb.com/ecosystem/use-cases/hadoop/

Table 1 shows comparison amongst MongoDB, HBase and Cassandra. [9]

Table 1. MongoDB vs. Cassandra Vs. HBase

Name	Cassandra	HBase	MongoDB
Description	Wide-column store based on ideas of BigTable and DynamoDB	Wide-column store based on Apache Hadoop and on concepts of BigTable	One of the most popular document stores
Supported server operating systems	BSD, Linux, OS X, Windows	Linux, Unix, Windows	Linux, OS X, Solaris, Windows
Datatype support	yes	no	yes
Secondary indexes	restricted	no	yes
APIs and other access methods	CQL and an API based on Apache Thrift	Java API, RESTful HTTP API, Thrift	proprietary protocol using JSON
Server-side scripts	no	yes	JavaScript
Triggers	yes	yes	no
Partitioning methods	Sharding	Sharding	Sharding
Replication methods	selectable replication factor	selectable replication factor	Master-slave replication
MapReduce	Yes	Yes	Yes
Concurrency	Yes	Yes	Yes
In-memory capabilities	No	No	Yes
Privileges	Access rights for users can be defined per object	Access Control Lists (ACL)	Access rights for users and roles

[5] ——, "Map-reduce," Web Page. [Online]. Available: https://docs.mongodb.com/manual/core/map-reduce/

[6] Edmond Lau, "What are the problems that a nosql database tries to solve?" Web Page, Jan 2011. [Online]. Available: https://www.quora.com/What-are-the-problems-that-a-NoSQL-database-tries-to-solve

[7] MongoDB, Inc, "Massive mongodb ecosystem flourishes with world's fastest growing database," Web Page. [Online]. Available: https://www.mongodb.com/press/massive-mongodb-ecosystem-flourishes-world%E2%80%99s-fastest-growing-database

[8] MongoDB, Inc., "Flexible enough to fit any industry," Web Page. [Online]. Available: https://www.mongodb.com/who-uses-mongodb

[9] solid IT, "System properties comparison cassandra vs. hbase vs. mongodb," Web Page. [Online]. Available: http://db-engines.com/en/system/Cassandra%3BHBase%3BMongoDB

vCloud and vSphere

MICHAEL SMITH[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: mls35@iu.edu*

Paper 1, February 26, 2017

vSphere and vCloud are categorized as infrastructure as a service(Iaas) products. They were developed by VMware and are a cloud computing virtualization platform.[1] vSphere is not one piece of software but a suite of tools that contains software such as vCenter, ESXi, vSphere client and a number of other technologies. Similarly, vCloud is also a suite of applications but for establishing an infrastructure for a private cloud.[2] The suite includes a variety of products such as the vSphere suite, site recovery management for disaster recovery, site networking and security, and many more. These specific services as well as Iaas in general are discussed. Advantages and disadvantages of utilizing such as service are presented followed by an analysis of vSphere and vCloud. Additionally, the big data extensions available for these services are examined. © 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: vCloud, vSphere, Iaas, I524

https://github.com/michaelsmith1983/sp17-i524/blob/master/paper1/S17-IO-3019/report.pdf

CLOUD COMPUTING

With the rise of high speed internet, connectivity amongst devices loc made cloud computing a practical alternative to the standard local solution. Cloud computing is defined as the utilization of remote servers to provide the computing power and data management instead of utilizing a local computer. During the late 1990's, companies began to see the potential benefits of moving to the cloud, one of the first companies to do so was salesforce.com who delivered applications to end users over the internet. [3] Today there are many companies in the business of cloud computing. The majority of business aqcuired coming from Microsoft, Google, and Amazon. Cloud computing business can be subdivided into three categories such as software as service(Saas), platform as a service(Paas), and infrastructure as a service(Iaas). Saas is software that runs remotely from a service provider and is utilized over the internet, some examples that fall into this category are google docs, gmail, office, and office 365. Paas are web and database services enabling the user to develop and run applications in the cloud, some examples are google app engine, microsoft azure, and amazon web services. Iaas provides the user virtualized computing infrastruture over the internet. Companies such as amazon web services, Microsoft azure, and vCloud provide this service. It is important to note that services offer more than just one type of cloud computing option. [4]

ADVANTAGES OF IAAS

Iaas is a business model in where a provider will host all components of infrastructure including hardware, software, servers, and data storage for their clients. The cost to the client is at a rate of usage which can range from various different rates such as per hour, week or month. Other charges come from levels of computing power and data storage. While it might seem costly to pay for such as service, utilization of Iaas provide several advantages for organizations. Management does not have the upfront costs their own IT infrastructure as well as maintaining hardware or replacing equipment that has failed or become obsolete. The cost of ensuring the network is and up and running through having a qualified team of IT on staff is also mitigated. Due to the cost of Iaas being metered the companies will only pay for what they need. [5] Depending on the application of an Iaas, there may be instances where a company may need to quickly scale up their usage. A great example is when a website produces an unexpected influx of visits in short period of time. If the IT infrastructure is maintained in house, it might not be able to quickly scale up to the demand. This would be detrimental to the business due to a slow online experience of the customer or possibly a website that crashes. [6] Depending on the company, it can be both costly and impractical to staff IT personnel that monitor demands twenty four hours a day seven days a week. The Iaas has the capacity to scale with demands and solutions are in place to help alleviate instances where a sudden increase in IT infrastructure is required. Disaster recovery is a major worry for companies, implementation of a safe backup solution can be both costly and difficult to execute. Iaas offer disaster recovery solutions, as long as an internet connection is available, the same environment can be accessed from leading to little to no downtime. With offsite storage of data via Iaas any form of data loss potential is limited to very little or none. [5]

DISADVANTAGES OF IAAS

Iaas is not a perfect solution, it does come with potential problems. Security is a major concern, becoming fully reliant on a service provider for IT solutions means their outages become a major problem for the client's business. Sensitive data that is being protected in the cloud always has the risk of possibly being stolen. As these Iaas services grow with a list of companies they support, this can lead them to become appealing targets for possible hackers due to the degree of damage that can be done. A common threat to all websites that Iaas are not immune to are called distributed denial of service (DDos) attacks. This is a type of cyber-attack where the goal is to make a network unavailable for normal users by disrupting service of a host on the internet. [7] The typical method to disrupt service is done so by flooding the bandwidth of a host with a plethora of requests through a botnet which is definied as a network of computers programmed to receive commands without the owners knowledge. These attacks have grown to such an extent that the department of homeland security has initiated funding new research leading to the prevention of DDos. [8] From a business perspective, this vulnerability can have disastrous effects to the users of Iaas, risks should be assessed and possible backup plans in place.

VCLOUD AND VSPHERE

vCloud falls into the cloud computing subcategory Iaas. It is a suite of multiple products that include vSphere, vCloud Director, vCloud, connector, vCloud networking and security, vCloud networking and security, vCenter site recovery and manager, vCenter operations management suite, vFabric application director, and vCloud automation center. vSphere is responsible for the physical hardware resource management and allocation of virtualization across a large group of infrastructure such as CPUs, data storage and networking.

vSphere utilizes ESXi which is an enterprise class type 1 hypervisor. A hypervisor is also known as a virtual machine manager defined as "a hardware virtualization technique that allows multiple guest operating systems (OS) to run on a single host system at the same time. The guest OS shares the hardware of the host computer, such that each OS appears to have its own processor, memory and other hardware resources." [**?**] The type 1 refers to a type of hypervisor that can run directly on the hosts pc and control its resources.

vCloud director is a tool for overall cloud management that helps the user build hybrid clouds through pooling resources into data centers. This product helps empower existing IT within an organization with the tools necessary to expand their infrastructure into the cloud. vCloud connector creates a single user interface that as a bridge between private and public clouds, this simplifies management by enabling the user to transfer workloads under a single hybrid cloud umbrella.

vCloud networking and security provides capabilities to protect virtual machines. A firewall is applied either encompassing a virtual datacenter or at the network interface. VPN or virtual private network is utilized to ensure safety for extensions of the virtual data center, as well as secure sockets layer (SSL) VPN which is an industry standard for security compliance. [9] With regards to data security, a feature included will scan file servers for sensitive data such as credit card or social security numbers and ensure proper measures are in place for protection of such critical data.

Cloud Disaster recovery is defined as "a backup and restore strategy that involves storing and maintaining copies of electronic records in a cloud computing environment". [10] This also addressed in the feature vCenter site recovery manager. It is an automated solution that will recover from downtimes in a timely manner. This is done so by using replication technology to migrate virtual machines to a different site. Users are able to test the migration process in order to safely address any potential migration issues.

LICENSING

Depending on the demand, the cost of vCloud can be quite expensive. It is important to note that there are current Iaas offerings that are free but with limit usage. Amazon web services offers a free tier consisting of certain limits such as 1 million requests per month on aws lambda, 25 gb of storage through dynamoDB, 100 million free events per month on amazon mobile analytics and many other limits. [11] Other services generally offer limited time services followed by a pay requirement.

VSPHERE BIG DATA EXTENSIONS

Within the vSphere suite is a feature that can support big data and Apache Hadoop workloads. A set of tools are available for the user to deploy and run Haddop within the virtual infrastructure. The following distributions of Hadoop are supported: apache Hadoop, cloudera, pivotal, hortonworks, and mapR. Customization options such deploying a specific version of Hadoop or even multiple types of Hadoop are supported. In order to automate the management of Hadoop, VMware initiated project Serengeti. It can quickly deploy a Hadoop cluster into vSphere, it will protect the master node by automatically starting a new virtual machine if there is a suspected failure. An important note is that graphical user interface of big data extensions is only supported on the web client 5.1 or later, if big data extension is installed on vSphere 5.0, all abilities of the administrator can only be performed within the command-line. [12]

VCLOUD API

The vcloud application program interface (API) clients and the director communicate via HTTP through an XML exchange. "You use HTTP GET requests to retrieve the current representation of an object, HTTP POST and PUT requests to create or modifty an object, and HTTP DELETE requests to delete an object." [13]

CONCLUSION

VMware is one of the oldest companies that have been involved in the virtualization market. Their service provides a suite of tools that assist companies who want to utilize IT virtual infrastructure. The Iaas can help businesses in a lot of areas that would be difficult in a private IT environment such as scalablity, cost, disaster recovery and backup. There are drawbacks to Iaas such as risk of downtime from the IT service provider and security risks of sensitive data, however the vCloud suite has features that can address these drawbacks with regards to security. Additionally, vcloud supports big data extensions including support for various versions of hadoop and have initiatives such as project serengeti which will automate its management. Vcloud is a potential option for clients interested in extending their IT infrastructure into the cloud.

REFERENCES

[1] vmware, "vcloud," Webpage. [Online]. Available: http://www.vmware.com/products/vcloud-suite.html

[2] Bipin, "Difference between vsphere, esxi and vcenter," Webpage, 08 2012. [Online]. Available: http://www.mustbegeek.com/difference-between-vsphere-esxi-and-vcenter/

[3] ECI, "History of cloud computing," Webpage. [Online]. Available: http://www.eci.com/cloudforum/cloud-computing-history.html

[4] B. Kepes, "Understanding the cloud computing stack," Webpage. [Online]. Available: https://support.rackspace.com/white-paper/understanding-the-cloud-computing-stack-saas-paas-iaas/

[5] Statetech, "5 important benefits of infrastructure as a service," Webpage. [Online]. Available: http://www.statetechmagazine.com/article/2014/03/5-important-benefits-infrastructure-service

[6] C. Loo, "3 things about scalability in iaas," Webpage, 01 2015. [Online]. Available: https://www.linkedin.com/pulse/3-things-scalability-iaas-charlie-loo

[7] wikipedia, "Denial of service attack," Webpage. [Online]. Available: https://en.wikipedia.org/wiki/Denial-of-service_attack

[8] J. Brown, "Dhs wants to stop the rise of large scale ddos attacks," Webpage, 02 2017. [Online]. Available: http://www.ciodive.com/news/dhs-wants-to-stop-the-rise-of-large-scale-ddos-attacks/436536/

[9] vmware, "vcloud networking and security overview," Webpage. [Online]. Available: http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/whitepaper/products/vcns/vmware-vcloud-networking-and-security-overview-whitepaper.pdf

[10] M. Rouse, "Cloud disaster recovery," Webpage. [Online]. Available: http://searchcloudstorage.techtarget.com/definition/cloud-disaster-recovery-cloud-DR

[11] amazon, "Aws free tier," Webpage. [Online]. Available: https://aws.amazon.com/s/dm/optimization/server-side-test/free-tier/free_np/

[12] vmware, "Vmware vsphere big data extensions," Webpage. [Online]. Available: http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vsphere/vmware-vsphere-big-data-extensions-faq.pdf

[13] ——, "vcloud api programming guide." [Online]. Available: http://pubs.vmware.com/vcloud-api-1-5/wwhelp/wwhimpl/js/html/wwhelp.htm

Google Fusion Table

MILIND SURYAWANSHI[1,2]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[2] *Electronics and Telecommunication Engineer,Pune University, 2010*
[*] *Corresponding authors: laszewski@gmail.com*

Paper-1 S17-IO-3020, March 13, 2017

Fusion Table is a web service provided by Google Inc. It is a data management tool, provides the feature like: data gathering, storing data in tables, visualizing stored data, merging tables, sharing table, view and download the tables.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Fusion Tables, data, query, I524

https://github.com/cloudmesh/classes/blob/master/docs/source/format/report/report.pdf

1. INTRODUCTION

Fusion Table is a "cloud Software as a Service (SaaS) application" [1]. It's a data visualization web application, which gathers, stored, share and visualize data. User can store data in tables and can represent it in the form of different charts, maps and plots. This information can be share, merged, published by individual user or website.

Fusion table provides capacity to import tables/files up to 250MB [2] of the type .csv (Comma Separated Value), .tsv (text-delimited files), .kml (KML), spreadsheets (like: .xls, .xsls, .ods) and google spreadsheet. Google provides quota of 1 GB per user to store tables. This 1 GB limit does not count the shared tables and tables in trash as a user quota [2]. Fusion table helps to visualize the stored data in unambiguous chart format, like: pie charts, bar charts, line plots, scatterplots, timelines and the geographical maps are provided. Fusion Tables service got launched in June 9 2009, and added to the Google Docs feature in 2011 [3]. Now they have released the v2 (version-2) of fusion table and v1 (version-1) has been deprecated from May 3rd, 2016 and be available till August 1st, 2017 [4].

Fusion tables uses HTTP request to perform the various tasks on tables. It allows users to [5]:

- create and delete tables
- read and modify the table column name and type
- insert, update and delete the rows in a table.
- Can change the access control of certain tables and its visualization.
- Fire quires on row to perform required tasks.

Data operation on table are performed through subset of SQL queries using HTTP request, also it used for retrieve the tables either in CSV or JSON file format. It uses the JSON format to make the changes in default behavior of table structure, metadata and visualization settings [5].

2. ARCHITECTURE

Fig-1 shows the architecture of Fusion Table; we are discussing the architecture component of Fusion Table in brief. Which explains the storage stack, Bigtable and user table schema information.

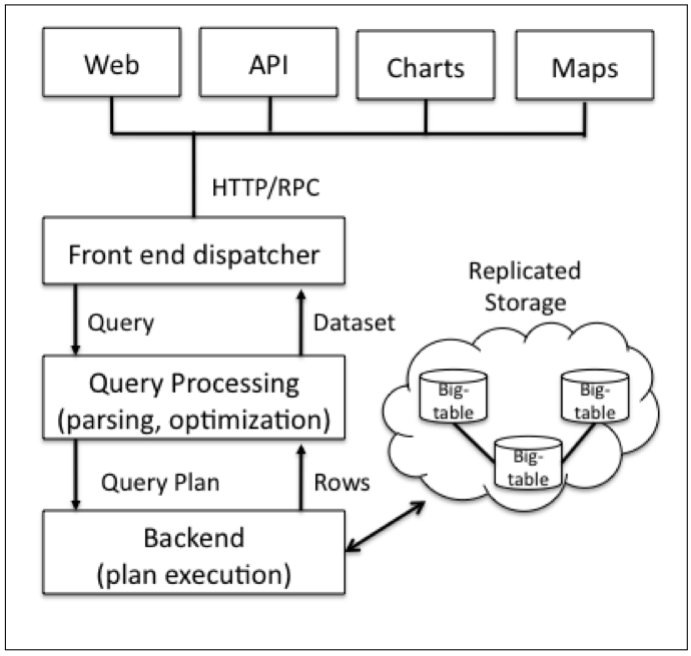


Fig. 1. Fusion Table Architecture

The service request in Fusion table is generated through Web, API, Charts and Maps. Web is a Fusion Table website, API will be from app which provides Fusion Table services, Charts to visualize the chart and to use spatial, the Maps service being used. These service request gets handled by the front end dispatcher. It will convert the above service request into required format and passes it to Query Processing. Front end dispatcher receives the dataset from Query Processing. The Query Processing convert the query into query plan. Here the query will be getting parsed and optimize to enhance the response of a query. Then Backend module's will execute the query plan received form Query Processing. Backend module will be responsible for fetching the diverse data of different table, sizes and queries from Storage. Also provides the requested (through optimize queries) data in rows [6].

3. STORAGE SYSTEM

Fusion Table provide large data storage capacity, it internally uses two-layer data storage, Bigtable and Magastore. It known as Google storage stack.

Bigtable: Bigtable stores data in tuples (it's a row, in a context of database), in key-value pair form. Its also stores the transaction timestamp history, the timestamp is a time when the tuples get written in table. Bigtable sorts the keys and stores the shard key in respective servers, according the key. Bigtable performs read and write operation. Write operation done automatically where read operation can be performed in 3 forms. Read by key prefix, read by key range or read key by exact key.

Megastore: It is "library on top of Bigtable" [6]. Megastore provide higher level operation like consistent secondary indexes, multi-row transaction and consistent replication.

Row Store: Fusion table stores all users table in to a single table called Rows, this being possible by Bigtable. Users table row gets saved in a table with key of a row with id of user's table. If a new user adds a table, then Bigtable splits the Rows table into sub-tables. Now this sub-tables are handled by separate server, which provide good performance in querying millions of tables. Bitables stored the property value of table in string format. Please see table 1.1.

Row Key (table Id, Row Id)	Index property	Non-Index property
(123, 1)	model=328i, color=red, type=sedan	notes=sells quickly
(123, 2)	model=330i, color=red	
(124, 1)	price=20, location=warehouse, UPC=500	
(124, 2)	price=32, location=shelf, UPC=430	notes=reorder needed
…	…	…

Fig. 2. As an example of sub-table

Observe the above table, it's a subset of Rows table. It holds the values for two different table, table 123, and 124, with its row Id e.g.1, 2. If we observe the row of table 123, with row id-1 holds the property model, color, type and notes. Where row-2 holds the value of model and color.

4. QUERY PROCESSING

As mentioned above, Fusion Table uses Bigtable and MegaStore for storage, and it useless subset of SQL, for query processing. In Fusion table we use different kind of queries to retrieve the results, following are the few common queries:

Prefix scan: This is the query type with prefix value. E.g. "select * from 111 limit 50". Here the result rows would be with prefix value = 111 up to the count 50;

Index prefix scan: Here the scan would be performing on index and the prefix value. E.g. "select * from 111 where color = 'blue'". This query will result the rows with prefix value 111 and the color is blue.

Index range scan: The queries which provides the result in range. E.g. "select * from 111 where price 10 and price 20". The query will result in rows, which have the prefix 111 and has the price in the range of 10 to 20.

Index join: Such queries particularly used in merging the table. It either merges the small table into big one or do the index merging. Parallel query will prefix the keys (in this example A.key and B.key) and compare the matching rows, such pairs then retrieved from the rows table. Query example: "select * from A.key = B.key".

5. VISUALIZATION FEATURE

Google provide powerful collection of visualization frameworks [7]. Which provides API to embed in website to show the real time data. Where the user can integrate the visualization in the website with data, which will show the refection of data changes directly in the graph/maps. It provides data set interface to obtain data, for visualization [7].

Fusion tables facilitate the integration of geo location in website as well. Where user can upload a table with street address, points, lines or polygons. This data then rendered on server side and that will send to the user's website where it has been integrated, in the form of tiles (images). Following is the example of map service integration using Fusion Table.

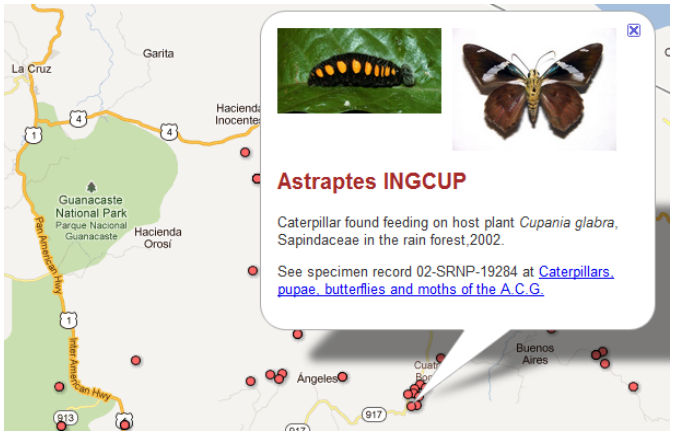


Fig. 3. Map service integration using Fusion Table [8]

Like the maps, user can also integrate the pie charts, line charts, bar carts, scatter chart as per the requirement .

6. LATEST FEATURES

1. Controlled sharing of data, user can share the specific data which should be shared with other user.
2. Chart tooltip layout: Shows the information about the values after one would hover over them.

3. Filtering option: "NOT" queries and regular expression matching.

4. Custom table and column properties.

5. Extended quota limits up to 1GB per user, and user can store 250 MB data per table.

6. Media download of large tables: provide large tables download facility without having server timeout problem.

7. CONCLUSION

Google Fusion Tables is a web application used for sharing, visualizing and publishing of data on websites. It is also known as SaaS app, hosted and maintained by Google cloud services. This service can be accessible to user through Internet (using browser) or by using application. User can upload files (CSV, KML, ODS, XLS or Google Spreadsheet) to the Fusion Tables table. This data can be then visualize using different kind of charts, for private use or embed in website. Fusion Tables table can be used for real time data changes and its reflection on charts/map. As in collaboration of all the feature, Fusion table is helpful in data management tool and data analytics operation, with good scalability. Also it provides a real time data collaboration with add, delete, edit and comment feature to the users. Integration to Google will provide our tables, maps are searchable to Google's crawlers.

8. ACKNOWLEDGEMENT

Thanks to Professor Gregor von Laszewski for giving me an opportunity and encouragement to create a technical paper. Also the entire class and TA's for their suggestion for paper.

REFERENCES

[1] Technopedia, "Google Fusion Tables," Web Page, Feb. 2017. [Online]. Available: https://www.techopedia.com/definition/26624/google-fusion-tables

[2] Google Developers, "Fusion Table Help: Type and size of files to import," Web Page, Feb. 2017. [Online]. Available: https://support.google.com/fusiontables/answer/171181?hl=en

[3] ——, "Type and size of files to import," Web Page, Feb. 2017. [Online]. Available: https://en.wikipedia.org/wiki/Google_Fusion_Tables

[4] ——, "Fusion tables Rest API: Get Start," Web Page, Feb. 2017. [Online]. Available: https://developers.google.com/fusiontables/docs/v2/getting_started

[5] ——, "Fusion tables Rest API: Release Notes," Web Page, Feb. 2017. [Online]. Available: https://developers.google.com/fusiontables/docs/release_notes

[6] W. S. HectorGonzalez, AlonHalevy, "Google fusion tables: Data management, integration and collaboration in the cloud," Tech. Rep., 2010.

[7] Google Developers, "Google Charts," Web Page, 2017. [Online]. Available: https://developers.google.com/chart/interactive/docs/?csw=1

[8] Gregor von Laszewski and Badi Abdul-Wahid, "Big Data Classes," Web Page, 2017. [Online]. Available: https://support.google.com/fusiontables/answer/2527132?hl=en

CUBRID RDBMS

ABHIJIT THAKRE[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[2] *Mechanical Engineer,Nagpur University, 2003*
[*] *Corresponding authors: abhijit.thakre@gmail.com*

project-000, February 28, 2017

With advanced techniques of data mining and analysis, bigdata processing has become a key in today's world. Many of the bigdata processing uses NOSQL data for storing. However in order to avail the ACID behavior of database, the focus is again back to the RDBMS databases. This paper focuses on one the similar ORDBMS CUBRID. It also highlight the architecture of CUBRID with it key component and features provided.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524

https://github.com/cloudmesh/classes/blob/master/docs/source/format/report/report.pdf

INTRODUCTION

CUBRID is open source RDBMS with object support developed by Navel corporation. Developed in C language CUBRID provides key features like scalability, high availability, higher performance, online and incremental backups. CUBRID is distributed under GNU general public license for the database server engine and BCD license for API and client tool.

ARCHITECTURE

CUBRID has distinguished 3 layer architecture. It consists of Database server, connection broker and application layer.

Database Server

Â It is the core component of the CUBRID Database Management System. The main function for the database server are as below

- Saving and managing the data.
- Processing of the queries from user.
- Providing smooth functioning for multiple users.

Broker

It acts as middleware between the Database Server and GUI application to provide seamless experience. It provides functions

- Connection pooling.
- Monitoring.
- Log tracing and analysis.

CUBRID Manager

Â It is a GUI tool that manages database and broker. It also provides the Query Editor for executing queries on database for users.

OVERALL ARCHITECTURE

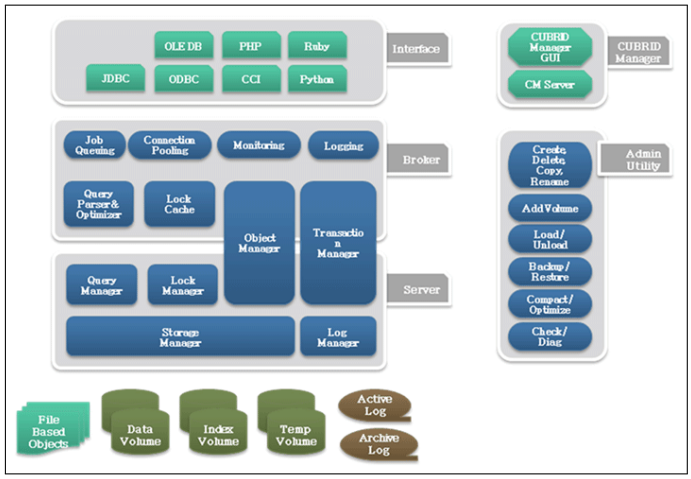


Fig. 1. [1]

DESCRIPTION

Database server and the broker work in co-ordination with each other as server and client respectively. The can be deployed on the different or same machine. The broker takes care of the queries from the users, it process it to the optimum level. On optimization it creates a query plan tree and sends the request to the server. The response from the server is via cursor navigation which is further returned to the user.

The client caches object instances from the database to its memory to provide fast access to data by using the query execution results or directly by users/applications. In addition, it caches locks as well as objects from the server for concurrency control. The execution of triggers or methods specified by users or applications is also performed in the client module.

References: [1]

MODULE CONFIGURATION

The CUBRID client and server modules consist of the following components:

- Transaction Management Component.
- Server Storage Management Component.
- Client Storage Management Component.
- Object Management Component.
- Client-Server Communications.
- Thread Management.
- Query Processing.

AUTHENTICATION IN CUBRID

CUBRID provides two levels of authentication.

User needs to enter credentials to login to the Host Server. On first login user need to set the admin credentials.

User needs to login to each database in the host server to access the individual database. Reference : [**?**]

KEY FEATURES

High Availability and Scalability

CUBRID uses heartbeat technology to provide automated accurate fail-over and fail-back features which makes the database continuously available. The server is available during the upgrades, replacement or even during the maintenance phase.

Database Sharding

CUBRID 8.4.3 provides free sharding feature where data can be divided on multiple instance. In addition to unlimited sharding it also provides the features like connection pooling and load balancing to all the shards.

Performance

CUBRID provides high performance to the users with feature like query caching, optimized algorithm for indexing, fast object access.

Function based indexing, filtering indexing and index skip scan provides various features to user for increasing the performance

Reliability

CUBRID is highly reliable with features like online incremental backup and restore. It provides the access restriction based on userip and databaseid.

Language Support

It provides 90 percent support to sql language support.

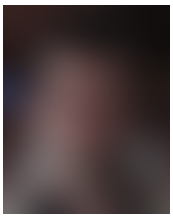
CUBRID SHARD

CUBRID shard is sharding is RDBMS specially targeted to address the problem on processing bigdata. CUBRID shard distributes the user data on multiple server to store it. So for fetching the data specific to user it needs to pass key information about the shard in the query. Parsing the query and finding the shard both things are taken care by the broker and does not needs the additional layer. This helps in increasing the performance for big data.

REFERENCES

[1] "CUBRID RDBMS," Web Page, Indiana University, Feb. 2017. [Online]. Available: http://www.cubrid.org

AUTHOR BIOGRAPHIES



Abhijit Thakre received his BE (Mechanical) in 2003 from The University of Nagpur.

Netty vs ZeroMQ in Realtime Analytics

SUNANDA UNNI[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

project-000, February 28, 2017

Comparison of Netty with other messaging and communication frameworks used in HPC-ABDS namely ZeroMQ for realtime analytics. © 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: I524, Netty, Storm, Distributed Computing System, Real time Analytics

https://github.com/cloudmesh/classes/suunni/master/docs/source/format/report/report.pdf

INTRODUCTION

Big Data upsurge has seen a better performance requirement from Data Processing Pipelines. Another goal for the organization is to get timely results for the data crunching. Structured and unstructured data generated by heterogeneous sources are collected and processed in batches or in realtime. To improve upon the processing time distributed computational cluster set ups are used. Distributed computational model requires a lot of communication between the processes running on different nodes. There are many messaging frameworks available in HPC-ABDS for realtime analytics and specifically for interprocess communication namely ZeroMQ, Kyro. We are comparing the data of experiments done at Yahoo and IBM for throughput using the different messaging frameworks.

Netty [1] is an asynchronous event-driven network application framework for rapid development of maintainable high performance protocol servers and clients. Netty book "Netty in Action" [2] mentions Netty to have a performance superior to standard Java NIO API thanks to optimized resource management, pooling and reuse and low memory copying.

DESIGN OF A DISTRIBUTED REALTIME COMPUTATIONAL SYSTEM

In a distributed cluster worker processes, executors and tasks makes a running topology required for data processing and analytics. A reliable messaging framework is crucial for the inter process, intra process and inter topology communication.

Worker Process Illustration

An illustration of worker process is as shown in Fig. 1.

Running Topology

A running topology as illustrated in Fig. 2.

Each of the above worker process is running on the same or different node in the distributed cluster. Worker process(executor or task) communicates with other Worker processes.

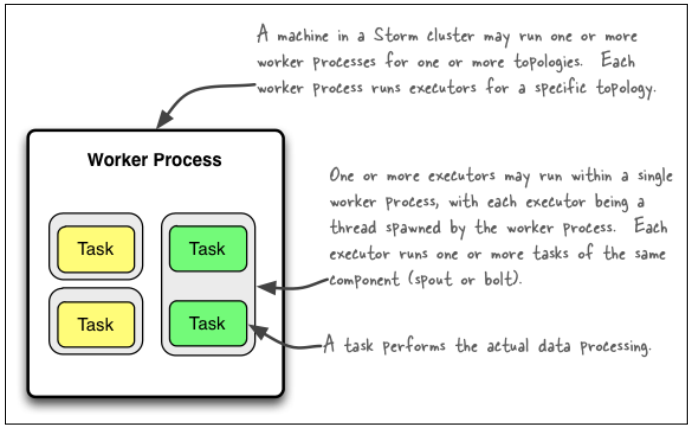


Fig. 1. The relationships of worker processes, executors (threads) and tasks in Storm, adopted from [3]

Apache Storm is a free and open source distributed realtime computation system. Storm makes it easy to reliably process unbounded streams of data, doing for realtime processing what Hadoop did for batch processing [4].

As per [5], Users are free to stitch together a directed graph of execution, with Spouts (data sources) and Bolts (operators). Architecturally, it consists of a central job and node management entity dubbed the Nimbus node, and a set of per-node managers called Supervisors. The Nimbus node is in charge of work distribution, job orchestration, communication, fault-tolerance, and state management (for which it relies on ZooKeeper).

The parallelism of a Topology can be controlled at 3 different levels: number of workers (cluster wide processes), executors (number of threads per worker), and tasks (number of bolts/spouts executed per thread)

Communication is handled at different levels in Storm using-

- Intra-worker communication in Storm (inter-thread on the same Storm node): using LMAX Disruptor

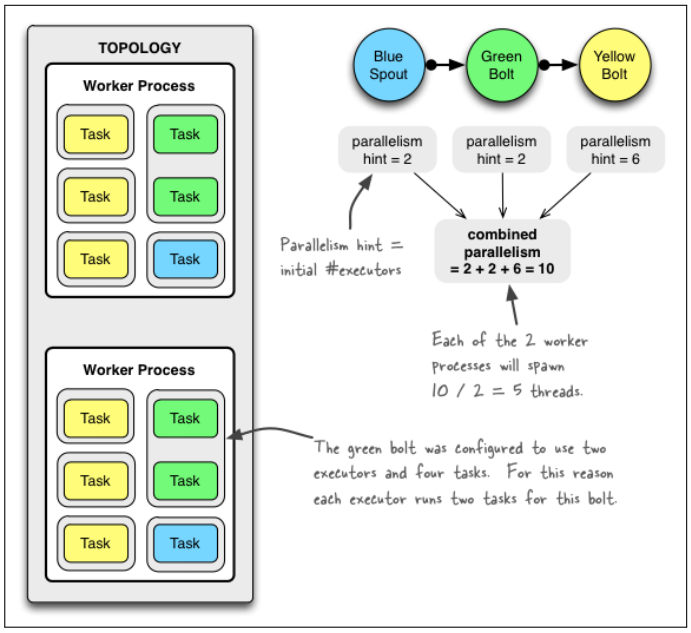


Fig. 2. Example of a Running Topology, adopted from [3]

- Inter-worker communication (node-to-node across the network): using ZeroMQ or Netty
- Inter-topology communication: nothing built into Storm, you must take care of this yourself with e.g. a messaging system such as Kafka/RabbitMQ, a database, etc.

For inter-worker communication Storm uses ZeroMQ by default in older versions and can use Netty in Storm versions later than 0.9, as the network messaging backend.

PERFORMANCE - YAHOO ENGINEERING EXPERIMENT

Yahoo experimented in 2013 for using a pluggable messaging layer for Inter-worker communication using Netty [6]. A simple speed of light test was run to benchmark and measure performance of messaging layer. This can be done by pushing messages between different Spouts and Bolts.

Setup and Configuration of Cluster

In the cluster setup 38 nodes, 3 nodes were dedicated to Zookeeper, 1 to Nimbus and the UI, and 34 for Supervisor and Logviewer processes. Each node had 2 x 2.4GHz Xenon processors each with 4 cores and Hyperthreading enabled for a total of 16 threads of execution.

Netty Behavior with large Topology

The Netty behavior with Topology of 3 workers running on 3 nodes. In the below Fig 3, the Y-axis represents number of messages sent and X-axis represent the size of messages. A 100% increase in number of messages is seen with Netty as compared to zeromq.

Netty Behavior with large Topology

The Netty behavior with Topology of 100 workers, 100 spouts, 500 bolts spread out in 5 levels of 100 bolts each, and 100 ackers is shown in Fig 4.

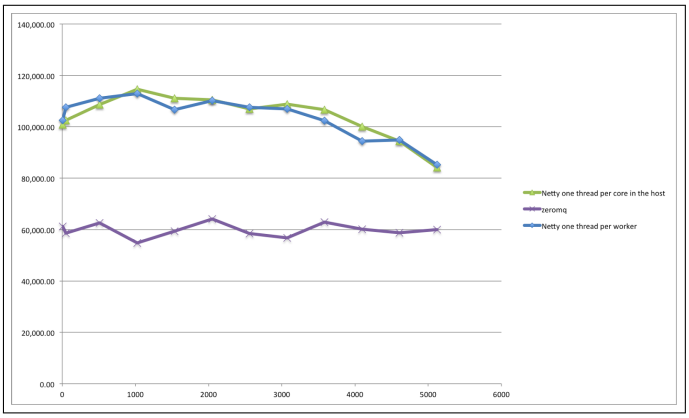


Fig. 3. Small Topology- Netty vs ZeroMQ, adopted from [6]

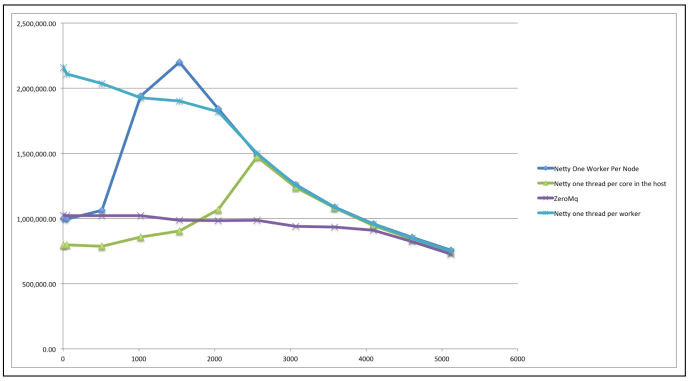


Fig. 4. Large Topology- Netty vs ZeroMQ, adopted from [6]

An increase is seen in number of messages as message size is increased however above behavior is due to less context switching as the message size increases. Post a peak at 2.5KB message size, network saturation occurs and we see a slope down where Netty performance matches ZeroMQ at 4KB message size.

To resolve and better Netty performance we need to-

- Reduce number of workers so single worker is running per node
- Make the number of threads configurable to default 1, which matches ZeroMQ default.

PERFORMANCE - IBM EXPERIMENT

Another verification of the numbers stated by Yahoo was done in IBM Whitepaper "Of streams and storms" [5].

In [5],the performance benchmarking is carried out by processing an Enron email dataset of half a million emails through data processing pipeline. The processing pipeline used IBM Infosphere Stream versus Apache Storm.

Setup and Configuration of Cluster

The cluster here consists of 6-nodes, 1 Nimbus node, 1 Zookeeper, 4 Supervisory nodes. Each node with 3GHz dual-core Xeon cluster (HS21), 2 CPUs per node, 8GB RAM, dual 146GB drives, RHEL 6.2 It is mentioned in the paper that Storm with Netty is around 1.65 times faster (in terms of throughput) than Storm with ZeroMQ for a multinode setup and “Restricted Benchmark” which consists of a simple 3-stage pipeline: source, count, and sink on a 4-node configuration

Report of Throughput

The below table shows the throughput number improvement for Storm with Netty(Storm 0.8.2) vs Storm with ZeroMQ(Storm 0.9.0.1) in Fig. 5.

Table 4: Storm 0.9 Results

System	Nodes	Dataset	Parallelism	Elapsed time	Throughput (emails/s)	CPU time
Storm 0.8.2	1	100%	4	21m 29s	1,270	1h 24m 5s
Storm 0.9.0.1				27m 28s	988.04	1h 28m 2s
Streams				3m 13s	8,438	0h 9m 48s
Storm 0.8.2	4	100%	4	5m 14s	5,213	1h 04m 34s
Storm 0.9.0.1				4m 43s	5,757	1h 04m 56s
Streams				1m 43s	31,637	0h 10m 12s
Storm 0.8.2	4	100%	8	9m 34s	5,659	2h 28m 56s
Storm 0.9.0.1				8m 58s	6,053	2h 18m 56s
Streams				1m 43s	31,637	0h 20m 47s

Fig. 5. Storm Results 0.9 vs 0.8.2, adopted from [p. 27 5]

CONCLUSION

Netty can be used as a pluggable Messaging framework for Distributed Computation System for Realtime Analytics for performance improvement however there is restriction of currently using it only for inter process communication. Netty is a pure Java solution which has a higher memory footprint of the application in comparison to ZeroMQ hence tuning is required for JVM heap size.

REFERENCES

[1] "Netty site," Web Page. [Online]. Available: http://netty.io/

[2] N. Maurer and M. A. Wolfthal, *Netty in Action*, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2015. [Online]. Available: http://www.ebook.de/de/product/21687528/norman_maurer_netty_in_action.html

[3] M. Noll, "Understanding the parallelism of a storm topology," Web Page, Oct. 2012. [Online]. Available: http://www.michael-noll.com/blog/2012/10/16/understanding-the-parallelism-of-a-storm-topology/

[4] Apache Software Foundation, "Apache storm," Web Page, 2015. [Online]. Available: http://storm.apache.org/index.html

[5] Z. Nabi, E. Bouillet, A. Bainbridge, and C. Thomas, "Of streams and storms," *IBM White Paper*, pp. 1–31, Apr. 2014. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.695.5752&rep=rep1&type=pdf

[6] B. Evans, "Making storm fly with netty," Web Page, Oct. 2013. [Online]. Available: https://yahooeng.tumblr.com/post/64758709722/making-storm-fly-with-netty

S17-IO-3023/report.pdf not submitted

LaTeX Linux Containers - Virtualization for Cloud Era

ASHOK VUPPADA[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: ashokmadhu66@gmail.com*

paper-1, February 26, 2017

In todays world , cloud is seen as delivery model for growing number of enterprises as the benefits of agility and efficiencies are better understood and available. The concept of virtualization is one of the building blocks for cloud infrastructure and services offering.The traditional VM work with hardware emulation causing inefficient resource and storage management. New technology 'Linux containers' build with the concept of 'Operating system Virtualization' aims at efficient resource management and customization. © 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: LXC,Cloud, I524

https://github.com/justbbusy/sp17-i524/tree/master/paper1/S17-IO-3024/report.pdf

1. INTRODUCTION

Traditional Hyperviser based Virtualization works with the idea of creating virtual hardware and running guest operating system on them. Applications are then expected to run on the guest operating system. The hypervisor layer would control the different guest operating systems.This model offers great deal of isolation,however the disadvantage of the this set up is each of the guest operating system would let a kernel process (memory and cpu manager processes) run on the host operating system and causing the performance overhead and resource constrain. It is also noted that each of the guest operating system runs from the ISO file (usually in GBs) causing limitations on the storage of the host. [1]

Container based virtualization works with the similar concept except eliminating the need of hypervisor layer. Each instance of the guest operating system can be assumed as any other processes running on the host system. There would be only one kernel processing running on the host operating system (that belong to the host OS) and it would shared with the guest operating systems. The hardware resources (CPU and memory) will also be shared. Since there is single process controlling the memory and CPU usage is running this model has performance advantage [1]

2. BUILDING BLOCKS

Linux containers are built on the modern kernel features like 'chroots' , 'namespaces' ,'cgroups' and Linux security models and mandatory access controls.[2]

- chroot : its a way of isolating a process to change only defined file system.The process running inside the chroot jail cannot act on the file system out side it. this is used to isolate the unstable applications to hamper the file system in use.[3]

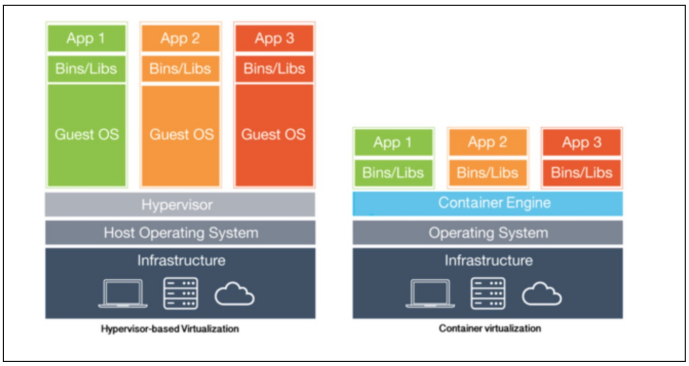


Fig. 1. Architecture Comparison of Hypervisor Based VM and containers.

- namespaces: namespaces provides the process level isolation for the global resources.It helps the process in each of the namespace to believe its the only process.[2]

- cgroups: its a kernel feature which allows to allocate resources to a given group of process(s).it helps one allocating and accounting CPU time, memory etc for a set of processes[4]

- Linux Security module: "Linux Security Modules (LSM) is a framework that allows the Linux kernel to support a variety of computer security models while avoiding favoritism toward any single security implementation".[5]

- Mandatory Access control: Its is process by which the security policies setup cannot gets overrides by a user. [6]

Linux containers are built on kernel features stated , as they are linux features the contender concept is not available in Microsoft windows.

3. PROJECT LXC AND DOCKER

Both LXC and Docker are the practical implementations of the linux containers. LXC is the original linux container developed.The project docker was initially started with the idea of extending LXC but ended up developing its own container. The key difference between LXC and a docker is in LXC each virtual container would allow multiple process to run where as in docker each process would need a separate docker container.Docker is designed to be stateless and credited for its portability. Because of the overwhelming popularity of docker it is often treated as synonym for linux containers .The below diagram depicts the main structural differences between LXC and docker.[7]

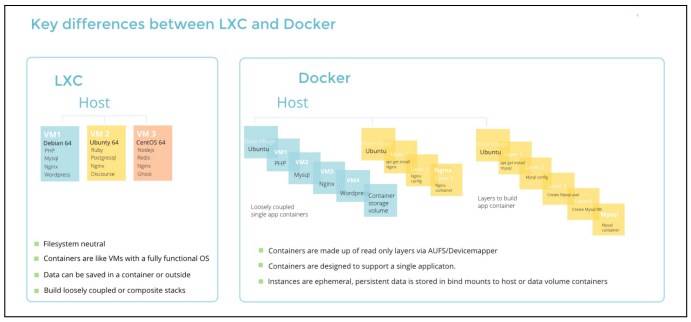


Fig. 2. Architecture Comparison of LXC and Docker.

4. PERFORMANCE COMPARISON

[8] In this study , Linux on KVM is studied for Hypervisor based VM , LXC and docker are compared as container based solutions. OSv is used as light weight guest OS.

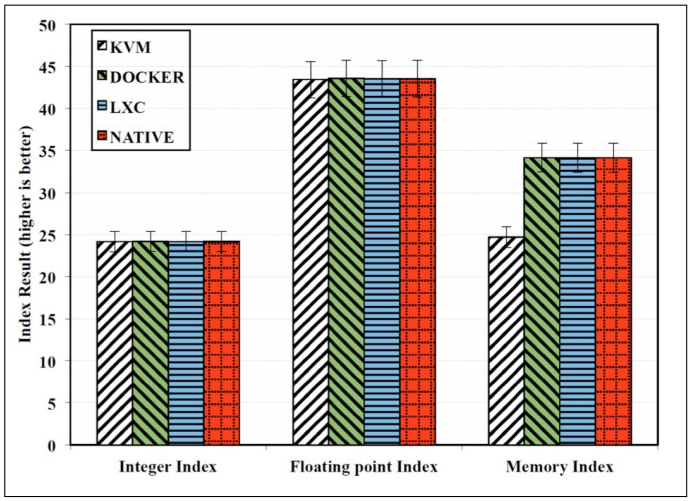


Fig. 3. Single Core Perf Comparison

1. CPU Performance
   CPU multithreaded performance is measured with a benchmark called 'Y Cruncher' which calculate the values of Pi the results can be seen from the table below NBENCH is used for single core comparison producing three different indexes: Integer Index, Floating Point Index, and Memory Index. Please see the blow diagram from the comparison

Platform	Multi-core Efficiency
Native	98.27%
LXC	98.19%
Docker	98.16%
KVM	97.51%

Fig. 4. Multi Core Perf Comparison

2. Memory Performance
   Below are test results using STREAM

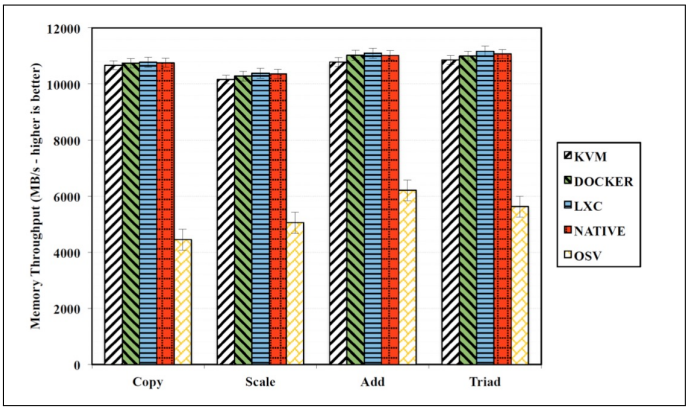


Fig. 5. Memory Perf Comparison

3. Disk I/O Performance
   Bonnie++ is used to measure the disk performance and below are the results.

Platform	Random Write speed (Kb/s)		Random Seeks	
Native	48254	%	1706	%
LXC	43172	**− 10.53%**	1517	**− 11.07%**
Docker	41170	**− 14.68%**	975	**− 42.84%**
KVM	23999	**− 50.26%**	125.7	**− 92.63%**

Fig. 6. Disk IO Perf Comparison

4. Network I/O Performance
   Network I/O is measured with Netperf

This test concludes that containers performed well but needs to improve on security and isolation.

Platform	TCP_STREAM (Mbps)		UDP_STREAM (Mbps)	
Native	9413.76	**%**	6907.98	**%**
LXC	9411.01	**− 0.00029%**	3996.89	**− 42.14%**
Docker	9412	**− 0.00018%**	3939.44	**− 42.97%**
KVM	6739.01	**− 28.41%**	3153.04	**− 54.35%**
OSv	6921.97	**− 26.46%**	3668.95	**− 46.88%**

Fig. 7. Network IO Comparison

5. USE CASES

[9]

1. Continous Integration
   With the concept of DevOps there is a tight integration between the development and Operations.The frequency of deploying new applications had increased tremondously.With multiple applications to be deployed with along with the dependencies,Op team used to have tough time, Container with the portability and packaging made the job easy. [9]

2. Container as a service
   This concept is similar to the PaaS, In a complex IT structure of an organization it was difficult to manage different existing technologies and add a new one,but with container as a service the new technology container image can be built and hosted on the existing host for the development to happen. This model eliminates the need of learning the dependencies associated with running multiple applications on the same host. [9]

3. Micro Services
   The idea of Micro service is to have the isolation between the application processes as much as possible so the each of the application can be built in the best technology suitable. This model is very helpful for modernization of the applications easily. Since container installation takes few seconds and doesn't come with the lot of dependencies they are the best choice for Microservices. [9]

6. LIMITATIONS

[10]

1. Containers are not suitable for all the purposes , some of the existing applications are designed best for sharing the physical layer of the machine.If scalability and fast deployment are not really applicable there is no good reason to use containers. [10]

2. Containers provides weaker isolation compare the hypervisor based virtual machines.Since they share the common kernel process, if there is some bug or virus on the host there would be chance to propgate to other containers. [10]

3. Containers are presently getting evolved,compared to the existing traditional VMs the tools available for monitoring the resources on containers are limited. [10]

4. containers are build on the kernel features build in Linux,Hence this cannot be directly used by other operating systems which does't have Linux kernel (Microsoft Windows)

7. CONCLUSION

In the era of cloud computing and micro services the need and popularity of containers are gradually growing up.Docker with its ability of "Build Ship Run" [11] ability made the container concept more popular and useful.But eventually its not going to replace the traditional hypervisor based virtual machines any time soon (as they are far more superior interns of security and integration compared to containers). Both the technologies needs to be used in conjunction to get the better of both the worlds.

REFERENCES

[1] "Hypervisor and container based virtulization," web page. [Online]. Available: http://www.slashroot.in/difference-between-hypervisor-virtualization-and-container-virtualization
[2] "Linux container building blocks," web page. [Online]. Available: http://bodenr.blogspot.com/2014/03/linux-containers-building-blocks.html
[3] "Basic chroot," web page. [Online]. Available: https://help.ubuntu.com/community/BasicChroot
[4] "Lxc and docker explained," web page. [Online]. Available: http://www.infoworld.com/article/3072929/linux/containers-101-linux-containers-and-docker-explained.html
[5] "Linux security modules," web page. [Online]. Available: https://en.wikipedia.org/wiki/Linux_Security_Modules
[6] "Linux madandatory access controls," web page. [Online]. Available: https://www.linux.com/news/securing-linux-mandatory-access-controls
[7] "Understanding the key differences between lxc and docker," web page. [Online]. Available: https://www.flockport.com/lxc-vs-docker/
[8] M. K. Roberto Morabito, Jimmy Kjällman, "Hypervisors vs. lightweight virtualization: a performance comparison," web page, 2015. [Online]. Available: https://pdfs.semanticscholar.org/551d/2b55067c06b5667cfc35b3b20096b9235cb4.pdf
[9] "Use cases for conatiners," web page. [Online]. Available: https://www.virtualizationpractice.com/container-use-cases-35048/
[10] "Cons for container technology," web page. [Online]. Available: http://searchservervirtualization.techtarget.com/feature/Five-cons-of-container-technology
[11] "Docker," web page. [Online]. Available: https://www.docker.com

HTCondor: Distributed Workflow Management System

NITEESH KUMAR AKURATI[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: akuratin@indiana.edu*

paper1, February 27, 2017

HTcondor is a distributed high throughput computing open-source workflow management system developed by the condor research team at University of Wisconsin-Madison Department of Computer Sciences. It is an unique and highly sophisticated Job Scheduler which has been changing and adopting dynamically in alignment with the users and the growing popularity of the distributed computing field. It is used for distributed parallelization of computing intensive tasks. The Condor project has become popular for its two key products, they are: The Condor high-throughput computing system, and the Condor-G agent for grid computing.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: HTCondor, ClassAd, Match Making, Condor-G, High-throughput Computing, Cycle Scavenging, Batch Systems

https://github.com/Niteesh01/sp17-i524/blob/master/paper1/S17-IR-2001/report.pdf

INTRODUCTION

An ideal computing environment provides ready access to huge scale of computing power. Over the course of years it is recognized that such immense computing power can be achieved at a very low cost by combining various small devices rather than using expensive supercomputers. This situation is addressed by the condor project. The core philosophy of the Condor project is flexibility.

"Like other full-featured batch systems, HTCondor provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their serial or parallel jobs to HTCondor, HTCondor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion"[1] Apart from providing the functionalities as other batch systems, HTCondor harnesses effectively wasted CPU cycles from idle desktops as well as workstations by using various effective mechanisms.

HTCondor is really unique because it can be used for managing workload on a set of dedicated clusters like the beowulf clusters, thereby allocating work to idle desktops/workstations the process being called as cycle scavenging. "HTCondor can seamlessly integrate both dedicated resources (rack-mounted clusters) and non-dedicated desktop machines (cycle scavenging) into one computing environment"[2]

ARCHITECTURE

At the core of HTCondor technology lies the kernel.The kernel shown in Fig.1 is the fundamental structure of HTCondor. Computing environments with wide variety can be constructed by making minor modifications to the kernel. The kernel workflow is as follows a user submits a job to an agent, the agent remembers the job in a persistent storage while looking for resources that are willing to run the job. Matchmaker is responsible for potential agents and resources. The agents and resources advertises themselves to the matchmaker. The matchmaker introduces the agent, the agent once introduced is responsible for contacting the resource and validating the match.To execute a job both agent and resource starts new processes called shadow and sandbox respectively. Shadow provides all the details necessary to execute a job. Sandbox creates a safe environment to run the job [3]

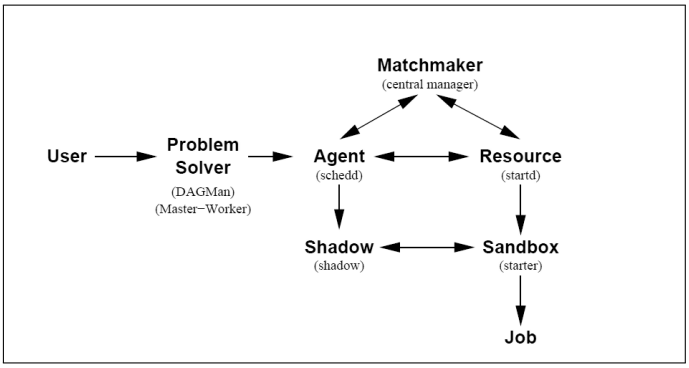


Fig. 1. The HTCondor Kernel

THE HTCONDOR SOFTWARE

HTCondor project supports a variety of computing systems deployed worldwide for various commercial and academic purposed. The HTCondor products: The Condor high-throughput computing system, and the Condor-G agent for grid computing are very popular among various domains. Lets discuss both these products in detail

HTCondor High Throughput Computing System

HTCondor being a high-throughput distributed batch computing system. It provides job management, scheduling, resource management and monitoring just like other batch systems. HTCondor is prominent in areas where other batch systems are weak like:high-throughput computing, and oppurtunistic computing. The idea of high-throughput computing is providing large amounts of computing power that is fault tolerant over prolonged periods of time by effectively utilizing the resources that are available in the network. Oppurtunistic computing is to use resources whenever they are available. High-throughput computing can be efficiently achieved through oppurtunistic computing. Therefore, HTCondor brings together high-throughput computing as well as oppurtunistic computing

To achieve high-throughput computing through oppurtunistic means, this requires several powerful as well as unique tools:

- **ClassAds.** "The ClassAd language in Condor provides an extremely flexible and expressive framework for matching resource requests(e.g jobs) with resource offers (e.g machines)"[3]. The ClassAd mechanism makes it easy for the job to state the requirements and preferences of the job. It also makes it easy for the machines to specify about the preferences and requirements for the jobs they are willing to run. Therefore, enabling these resources and requirements to be described in the form of powerful expressions thus resulting in HTCondor's adoption mostly any desired policy[3]

- **Job Checkpoint and Migration.** A Job checkpoint is a snapshot of the complete state of the job[4]. With certain types of jobs HTCondor can take a checkpoint of the job and later resume it. Job can continue its execution later exactly from it left off at the time of checkpointing. HTCondor also allows job migration from one pool of resources to an other pool of resources with the help of checkpointing[1]

- **Remote System Calls.** HTCondor preserves the local execution enviroment using remote system calls despite running jobs on remote machines. Users need not make files available on remote workstations by accessing the machines using remote login. Remote system calls is a mobile sandbox mechanism of HTCondor which is used for redirecting all the I/O related calls back to the machine which submitted the job. The program behaves as if it is running on the originally submitted workstation, regardless of where it really executes[3]

HTCondor with the help of above tools can also scavenge and manage wasted CPU power from otherwise idle workstations across the organization with minimal effort[5]. The same mechanisms enable preemptive-resume scheduling on compute cluster resources. Therefore, this allows HTCondor to support priority based scheduling on clusteres. HTCondor therefore can be used to combine all of organization's computing power into a single resource.

Condor-G

The Condor-G is combination of initiatives from Globus and HTCondor projects. Globus is about inter-domain communications as well as standardized access to a variety of remote batch systems[6]. In HTCondor comes the user concerns of fault tolerance, error recovery, creating a friendly execution environment, job submission and job allocation

"Condor-G can be used as the reliable submission and job management service for one or more sites, HTCondor can exist both at front end and back end of a grid. The HTCondor HTC system can be as the fabric management service for one or more sites and the Globus Toolkit can be used as the bridge between them"[3].

CLASSAD MECHANISM

The ClassAd mechanism is an unique, extremely flexible mechanism for handling jobs and resource requests. HTCondor uses matchmaking for matching an idle job with an available machine. ClassAds are used by users to specify which machines should service their jobs. Adminstrators uses it customize the sccheduling policy.

HTCondor's ClassAd mechanism is similar to the classifieds in the advertising section of the newspaper. Sellers advertise what they hope to sell to attract the buyers at the same time buyers advertise specifics whar they wish to purchase.

ClassAds consists of a unique set of named expressions. Each expression is an attribute. Each attribute has an attribute name as well as value[7].

DAGMAN

DAGMan is a problem solver which is a higher-level structure built on top of the HTCondor agent. It provides an unique programming model for managing large number of jobs. A problem solver relies/uses agent as a service for reliable execution of jobs. Therefore, the problem need not worry about failure of jobs as the agent assumes responsibility for hiding and retrying those jobs.

DAG Manager(DAGMan) is a service responsible for execution of multiple jobs with dependencies in a declarative form. DAGMan is a fault tolerant as well as distribute version of traditional tool make. DAG does not depend on the file system to record a DAG process unlike make. DAG keeps a set of private logs to act in the case of crashes or failures

APPLICATIONS

HTCondor has been adopted widely across various commercial and academia. Few of the notable applications in academia and commercial space are listed here: C.O.R.E Digital Pictures, NUG30 Optimization Challenge

- **C.O.R.E DIgital Pictures.** A highly successful Toronto-based computer animation studio. The studio primarily deals primarily with Photo-realisitic animation which is a compute intensive process. Each frame can take upto 1 hour. An animation requires 30 or more such frames. Today, HTCondor manages hundreds of linux and silicon Graphics machines at C.O.R.E Digital Pictures. On a average day 15000 jobs are submitted by the C.O.R.E animators to HTCondor. HTCondor has been successfully used by C.O.R.E for major productions such as X-Men, Blade II nd The Time Machine

- **NUG30 Optimization Challenge.** NUG30 is quadratic assignment problem first proposed in 1968 as one of the most difficult combinatorial optimization challenges, but remained unsolved for 32 years beacuse of its complexity. This problem is solved by four mathematicians from Argone National Laboratory, University of Iowa, and Northwestern University by using Condor-G and several other technologies

  The actual computation was managed by HTCondor's Master-Worker(MW) problem solver environment." MW submitted work to Condor-G, which provided compute resources from around the world by both direct flocking to other Condor pools and by gliding in to other compute resources accessible via the Globus GRAM protocol. Remote System Calls, part ofCondor's standard universe, was used as the I/O service between the master and the workers. Checkpointing was performed every 15 minutes for fault tolerance"[3]

  As a result a solution to NUG30 was discovered by utilizing Condor-G in a run of less than one week. Condor-G allowed the mathematicians to manage 2400 systems at 10 different sites seamlessly. Over 95,000 CPU hours are consumed in that week[3]

EDUCATIONAL MATERIAL

HTCondor is one of the most popular technology for high-computing distributed workflow management system. More information about HTCondor can be found here [8].

LICENSING

HTCondor is released under the open source Apache License, Version 2.0. You may not use this file except in compliance with the License. You may obtain a copy of the License here[9].

CONCLUSION

HTCondor is a powerful, unique and flexible high throughput computing distributed workflow management system which has evolved over years by keeping the end users,adminstrators and the growing demand in mind. HTCondor continues to standout from other batch management systems with the power of high-throughput computing and oppurtunistic computing.

REFERENCES

[1] T. Tannenbaum, D. Wright, K. Miller, and M. Livny, "Condor – a distributed job scheduler," in *Beowulf Cluster Computing with Linux*, T. Sterling, Ed. MIT Press, oct 2001.

[2] "Htcondor," Webpage. [Online]. Available: https://en.wikipedia.org/wiki/HTCondor

[3] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: the condor experience." *Concurrency - Practice and Experience*, vol. 17, no. 2-4, pp. 323–356, 2005.

[4] M. Litzkow, T. Tannenbaum, J. Basney, and M. Livny, "Checkpoint and migration of UNIX processes in the Condor distributed processing system," University of Wisconsin - Madison Computer Sciences Department, Tech. Rep. UW-CS-TR-1346, apr 1997.

[5] M. Litzkow, M. Livny, and M. Mutka, "Condor - a hunter of idle workstations," in *Proceedings of the 8th International Conference of Distributed Computing Systems*, jun 1988.

[6] I. Foster and C. Kesselman, "Globus: a metacomputing infrastructure toolkit," *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, no. 2, pp. 115–128, jun 1997. [Online]. Available: http://dx.doi.org/10.1177/109434209701100205

[7] N. Coleman, R. Raman, M. Livny, and M. Solomon, "Distributed policy management and comprehension with classified advertisements," University of Wisconsin - Madison Computer Sciences Department, Tech. Rep. UW-CS-TR-1481, apr 2003.

[8] "Htcondor high throughput computing," Webpage. [Online]. Available: https://research.cs.wisc.edu/htcondor/publications.html

[9] "Apache license 2.0 for htcondor." [Online]. Available: http://www.apache.org/licenses/LICENSE-2.0

Google Dremel: SQL-Like Query for Big Data

JIMMY ARDIANSYAH[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *jardians@indiana.edu - S17-IR-2002*

Research Article-01, March 14, 2017

Dremel is a scalable, interactive ad-hoc query system for analysis of read-only nested data. By combining multi-level execution trees and columnar data layout, it is capable of running aggregation queries over trillion-row tables in seconds.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Google Dremel, Big Data Query

https://github.com/jardians/sp17-i524/tree/master/paper1/S17-IR-2002/report.pdf

INTRODUCTION

Big data analytics nowadays has become more common across industries and government agencies partly due to fast and affordable commodity storage to keep pace with data and business growth.

Make the data sense at the fingertips of data scientists and analysts has become increasingly essential to compete in the business world. Having interactive tool with fast response times often make a difference in data exploration, rapid prototyping as well as designing of data pipelines. Performing interactive data analysis at scale demands a high degree of parallelism. That's where Dremel comes into play.

Dremel is a scalable, interactive ad-hoc query system for analysis of read-only nested data. By combining multi-level execution trees and columnar data layout, it is capable of running aggregation queries over trillion-row tables in seconds. The system scales to thousands of CPUs and petabytes of data. With Dremel, you get to write a declarative SQL-like query against data stored in a very efficient for analysis read-only columnar format. It's also possible to write queries that analyze billions of rows, terabytes of data, and trillions of records in seconds [1].

HISTORY DEVELOPMENT

When Google published the Dremel paper in 2010, it explained how this structure is preserved within column store. Every column, in addition to its value, also stores two numbers — definition and repetition levels.

This encoding ensures that the full or partial structure of the record can be reconstructed by reading only requested columns, and never requires reading parent columns (which is the case with alternative encoding). That same paper gives an exact algorithm for both encoding the data and reconstructing the record.

In 2014, Google published another paper — Storing and Querying tree-structured records in Dremel — which lays theoretical foundation and proves correctness of algorithms for filtering and aggregation operators, which take advantage of the above encoding. [2].

WHY DREMEL

This observation report discusses some characteristics of Dremel; a system that supports interactive analysis of very large datasets over shared clusters of commodity machines. Dremel can even execute a complex regular expression text matching on a huge logging table that consists of about 35 billion rows and 20 TB, in merely tens of seconds. This is the power of Dremel; it has super high scalability and most of the time it returns results within seconds or tens of seconds no matter how big the queried dataset is. Why Dremel can be as drastically fast as the examples show? The answer can be found in two core technologies which gives Dremel this unprecedented performance:

1. Columnar Storage. Data is stored in a columnar storage fashion which makes possible to achieve very high compression ratio and scan throughput.

2. Tree Architecture is used for dispatching queries and aggregating results across thousands of machines in a few seconds [1].

Columnar Storage.

Dremel stores data in its columnar storage, which means it separates a record into column values and stores each value on different storage volume, whereas traditional databases normally store the whole record on one volume; this is efficient for cases where many columns of the records need to be fetched. For example, if one analysis heavily relied on fetching all fields for records that belong to a particular time ranged, row-oriented storage would make sense.

To illustrate what columnar storage is all about, here is an example with three columns

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

Fig. 1. Typical row-oriented storage

In a row-oriented storage, the data is laid out one row at a time as follows:

A1	B1	C1	A2	B2	C2	A3	B3	C3

Fig. 2. Transform row to column-oriented format

Whereas in a column-oriented storage, it is laid out one column at a time:

A1	A2	A3	B1	B2	B3	C1	C2	C3

Fig. 3. Laid out the column

Dremel has introduced columnar storage, which provides several advantages over row-oriented system:

- Is generally very efficient in term of compression on columns because entropy within a column is lower that entropy within a block of rows. in other words, data is more similar within the same column, than it is in a block of rows. This can make a huge difference especially when the column has few distinct value.
- Work well for queries that only access a small subset of columns.
- I/O will be reduced as we can efficiently scan only a subset of the columns while reading the data. Better compression also reduces the bandwidth required to read the input.
- Is often well suited for data-werehousing applications where users want to aggregate certain columns over a large collection of records.
- As we store data of the same type in each column, we can use encoding better suited to the modern processors' pipeline by making instruction branching more predictable [3].

Tree Architecture

Dremel builds on ideas from web search and parallel DBMSs. First, its architecture borrows the concept of a serving tree used in distributed search engines. Just like a web search request, a query gets pushed down the tree and is rewritten at each step. The result of the query is assembled by aggregating the replies received from lower levels of the tree. Tree Architecture has enable Dremel to dispatch queries and collect results across tens of thousands of machines in a matter of seconds by using the Tree architecture. The architecture forms a massively parallel distributed tree for pushing down a query to the tree and then aggregating the results from the leaves at a blazing fast speed [4]. Consider a simple aggregation query below:

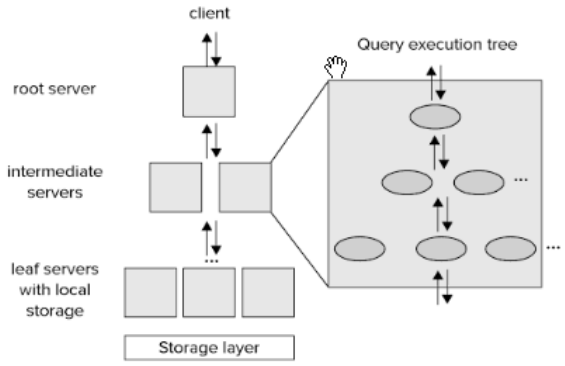


Fig. 4. Typical Tree Architecture [5]

A root server receives incoming queries, reads metadata from the tables, and routes the queries to the next level intermediate servers in the serving tree. The leaf servers communicate with the storage layer (based on the columnar model described earlier) to read data which is bubbling up for the aggregation and final result is return to the user or access the data on local disk.

However, after data is processed, one will be running aggregate queries and analysis on large chunks of data at a time, most probably only on a subsets of columns. Because many analytical queries only select a subsets of columns at a time for storing that will be analyzed later [5].

Overall, Dremel combine parallel query execution with the columnar format that supporting performance data access and also capable of operating on in situ nested data. In situ refers to the ability to access data 'in place', e.g., in a distributed file system like Veritas Cluster File System, General Parallel File System (GPFS), and Global File System (GFS).

IMPLEMENTATION OF GOOGLE DREMEL

Apache Drill is the open source version of Google's Dremel system which is available as an infrastructure service called Google BigQuery. One explicitly stated design goal is that Drill is able to scale to 10,000 servers or more and to be able to process petabytes of data and trillions of records in seconds. Drill is an Apache top-level project. Drill supports a variety of NoSQL databases and file systems. In addition, Drill supports data locality, so it's a good idea to co-locate Drill and the datastore on the same nodes [6].

CONCLUSION

Although query and process large volute of data in any system is a challenging task, especially in the big data ecosystem due to vast expense of option available, Dremel has been standing out as the right model for process and storing data with a lot of benefits as well as fitting as part of an entire big data stack which can be used against raw data, like log data. Choosing the right tool for your data is one of the most important decision one will make in the application, and everyone need to spend the appropriate amount of time and effort to get it right the first time. Because of it, I believed Dremel will be the future of interactive ad-hoc query system for analysis that require fast result.

ACKNOWLEDGEMENTS

this work was done as part of the course "I524: Big Data and Open Source Software Projects" at Indiana University during

Spring 2017

REFERENCES

[1] S. Melnik, A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, and T. Vassilakis, "Dremel: Interactive analysis of web-scale datasets," *Communications of the ACM*, vol. 54, pp. 114–123, Jun. 2011. [Online]. Available: http://cacm.acm.org/magazines/2011/6/108648-dremel-interactive-analysis-of-web-scale-datasets/fulltext

[2] Google, "Dinside capacitor, bigquery's next-generation columnar storage format," Web Page, Feb. 2017, accessed: 2017-02-20. [Online]. Available: https://cloud.google.com/blog/big-data/2016/04/inside-capacitor-bigqueys-next-generation-columnar-storage-format

[3] M. Grover, T. Malaska, J. Seidman, and G. Shapira, *Hadoop Application Architectures*. Sebastopol, California: O'Reilly Media Inc, 2015.

[4] Twittter, "Dremel made simple with parquet," Web Page, Feb. 2017, accessed: 2017-02-15. [Online]. Available: https://blog.twitter.com/2013/dremel-made-simple-with-parquet

[5] B. Lublinsky, K. T. Smith, and A. Yakubovich, *Professional Hadoop Solutions*. Indianapoli, Indiana: Wiley Publishing, 2013.

[6] wikipedia, "Apache drill," Web Page, Feb. 2017, accessed: 2017-02-20. [Online]. Available: https://en.wikipedia.org/wiki/Apache_Drill

Apache Samza

AJIT BALAGA, S17-IR-2004[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: abalaga@iu.edu, ajit.balaga@gmail.com*

project-000, February 27, 2017

Apache Samza is a stream processing framework which enables usesrs to analyze streaming datawith ease. The concepts required streaming data and the architecture behind Samza is presented here. Samza relies on Kafka and YARN, and its relationship with the two technologies is elaborated upon. The API is introduced as a precursor to future streaming projects.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524

https://github.com/argetlam115/sp17-i524/blob/master/paper1/S17-IR-2004/report.pdf

INTRODUCTION

Messaging systems are a method of implementing realtime asynchronous computation, where messages can be added to a message queue, pub-sub system, or log aggregation system whenever an event occurs. Consumers or subscribers read these messages from the queues or systems and take actions based on the message contents. A messaging system stores messages and waits for consumers to consume them. Samza, a stream processing system, is a higher level of abstraction on top of messaging systems.[1]

SAMZA

Samza uses YARN and Kafka to provide a framework for stage-wise stream processing and partitioning. Samza is a stream processing framework with the following features:[1]

- Simple API
- Managed state
- Fault tolerance
- Durability
- Scalability
- Pluggable
- Processor isolation

The Samza client uses YARN to run a Samza job: YARN starts and supervises one or more SamzaContainers, and your processing code runs inside those containers. The input and output for the Samza StreamTasks come from Kafka brokers that are co-located on the same machines as the YARN NMs.[2] Samza is made up of three layers:

- A streaming layer
- An execution layer
- A processing layer

Samza provides out of the box support for all three layers.

- Streaming: Kafka
- Execution: YARN
- Processing: Samza API

Samza's execution and streaming layers are differentiable from the processing layer allowing developers to implement alternatives of their choice.

STREAMING WITH KAFKA

Kafka is a distributed pub/sub and message queueing system that provides at-least once messaging guarantees, and highly available partitions. Each topic (streams in Kafka) is partitioned and replicated across multiple machines called brokers. When a producer sends a message to a topic, it provides a key, which is used to determine which partition the message should be sent to. The Kafka brokers receive and store the messages that the producer sends. Kafka consumers can then read from a topic by subscribing to messages on all partitions of a topic.[3] Kafka has some interesting properties:

- All messages with the same key are guaranteed to be in the same topic partition.
- A topic partition is a sequence of messages in order of arrival and are stored using a monotonically increasing offset. The consumer can keep track of the offset by storing the offset of the last message it has processed.

RESOURCE MANAGEMENT WITH YARN

YARN is Hadoop's next-generation cluster scheduler which allows you to allocate a number of containers in a cluster of machines, and execute arbitrary commands on them. Samza uses YARN to manage deployment, fault tolerance, logging, resource isolation, security, and locality. YARN has three important pieces: a ResourceManager, a NodeManager, and an ApplicationMaster. In a YARN grid, every machine runs a NodeManager, which is responsible for launching processes on that machine. A ResourceManager talks to all of the NodeManagers to tell them what to run. Applications talk to the ResourceManager when they wish to run something on the cluster. The third piece, the ApplicationMaster, is actually application-specific code that runs in the YARN cluster. It's responsible for managing the application's workload, asking for containers, and handling notifications when one of its containers fails.[4] Samza provides a YARN ApplicationMaster and a YARN job runner out of the box.The Samza client talks to the YARN RM when it wants to start a new Samza job. The YARN RM talks to a YARN NM to allocate space on the cluster for Samza's ApplicationMaster. Once the NM allocates space, it starts the Samza AM. After the Samza AM starts, it asks the YARN RM for one or more YARN containers to run SamzaContainers. Again, the RM works with NMs to allocate space for the containers. Once the space has been allocated, the NMs start the Samza containers.[5]

RELATED CONCEPTS

Streams

A stream is composed of immutable messages of a similar type or category. Messages can be appended to a stream or read from a stream. A stream can have any number of consumers, and reading from a stream doesn't delete the message. Messages can optionally have an associated key which is used for partitioning. [1]

Jobs

A Samza job performs a logical transformation on a set of input streams to append output messages to set of output streams. In order to scale the throughput of the stream processor, streams and jobs are cut into smaller units of parallelism: partitions and tasks.[1]

Partitions

Each stream is broken into a number of partitions, each with a few properties. Each partition is an ordered sequence of messages with an identifier called the offset, unique to its partition. When a message is appended to a stream, it is appended to only one of the stream's partitions.[1]

Tasks

A job is broken into multiple tasks. Each task consumes data from a partition. A task processes messages from each of its input partitions sequentially, in the order of message offset, in parallel for all partitions available. The YARN scheduler assigns each task to a machine, so the job as a whole can be distributed across many machines. The number of tasks in a job is determined by the number of input partitions (there cannot be more tasks than input partitions, or there would be some tasks with no input). However, you can change the computational resources assigned to the job (the amount of memory, number of CPU cores, etc.) to satisfy the job's needs. See notes on containers below. The assignment of partitions to tasks never changes: if a task is on a machine that fails, the task is restarted elsewhere, still consuming the same stream partitions.[1][6]

Dataflow Graphs

We can compose multiple jobs to create a dataflow graph, where the edges are streams containing data, and the nodes are jobs performing transformations. This composition is done purely through the streams the jobs take as input and output. The jobs are otherwise totally decoupled: they need not be implemented in the same code base, and adding, removing, or restarting a downstream job will not impact an upstream job. These graphs are often acyclic—that is, data usually doesn't flow from a job, through other jobs, back to itself. However, it is possible to create cyclic graphs if you need to.[1]

Containers

Partitions and tasks are both logical units of parallelism—they don't correspond to any particular assignment of computational resources. Containers are the unit of physical parallelism, and a container is essentially a Unix process. Each container runs one or more tasks. The number of tasks is determined automatically from the number of partitions in the input and is fixed, but the number of containers is specified by the user at run time and can be changed at any time.[1]

API

When writing a stream processor for Samza, you must implement the StreamTask interface. When you run your job, Samza will create several instances of your class. These task instances process the messages in the input streams. For each message that Samza receives from the task's input streams, the process method is called. The envelope contains three things of importance: the message, the key, and the stream that the message came from. The key and value are declared as Object, and need to be cast to the correct type. If you don't configure a serializer/deserializer, they are typically Java byte arrays. A deserializer can convert these bytes into any other type.[1] The getSystemStreamPartition() method returns a SystemStreamPartition object, which tells you where the message came from. It consists of three parts:

- The system: the name of the system from which the message came, as defined in your job configuration. You can have multiple systems for input and/or output, each with a different name.
- The stream name: the name of the stream (topic, queue) within the source system. This is also defined in the job configuration.
- The partition: a stream is normally split into several partitions, and each partition is assigned to one StreamTask instance by Samza.

The API looks like this:

```
/** A triple of system name, stream name and partition. */
public class SystemStreamPartition extends SystemStream
/** The name of the system which provides this stream. It is
defined in the Samza job's configuration. */
public String getSystem() ...
/** The name of the stream/topic/queue within the system. */
```

```
public String getStream() ...
/** The partition within the stream. */
public Partition getPartition() ...
```

CONCLUSION

Samza is a very powerful tool to work on streaming data. With its simple approach, it allows us to analyze large amounts of streaming data on the go. The architecture allows the developer to utilize their own resource manager and their message handling system. Samza's architecture is very similar to hadoop, enabling users to get started with their applications quickly and making the learning curve shallow. With its simpleapi, Samza is a comfortable technology for analyzing streaming data.

REFERENCES

[1] Web Page. [Online]. Available: http://samza.apache.org

[2] T. Feng, Z. Zhuang, Y. Pan, and H. Ramachandra, "A memory capacity model for high performing data-filtering applications in samza framework," in *Big Data (Big Data), 2015 IEEE International Conference on*. IEEE, 2015, pp. 2600–2605.

[3] M. Kleppmann and J. Kreps, "Kafka, samza and the unix philosophy of distributed data," *Bulletin of the IEEE CS Technical Committee on Data Engineering*, 2015.

[4] G. Wang, J. Koshy, S. Subramanian, K. Paramasivam, M. Zadeh, N. Narkhede, J. Rao, J. Kreps, and J. Stein, "Building a replicated logging system with apache kafka," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1654–1655, 2015.

[5] B. Srikanth and V. K. Reddy, "Efficiency of stream processing engines for processing bigdata streams," *Indian Journal of Science and Technology*, vol. 9, no. 14, 2016.

[6] J. Samosir, M. Indrawan-Santiago, and P. D. Haghighi, "An evaluation of data stream processing systems for data driven applications," *Procedia Computer Science*, vol. 80, pp. 439–449, 2016.

Apache Spark

SNEHAL CHEMBURKAR[1] AND RAHUL RAGHATATE[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: snehchem@iu.edu, rraghtate@iu.edu

project-001, February 28, 2017

Apache Spark, developed at UC Berkeley AMPLAB, is a high performance framework for analyzing large datasets [1]. The main idea behind the development of Spark was to create a generalized framework that could process diverse and distributed data as opposed to MapReduce which only support batch processing of data. Spark has multiple libraries built on top of its core computational engine which help process diverse data. This paper will discuss the spark runtime architecture, its core and libraries.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Spark, RDDs, DAG, Driver, Cluster, Worker, I524

https://github.com/snehalvartak/sp17-i524/paper1/S17-IR-2006/report.pdf

INTRODUCTION

Spark is an open source distributed cluster computing engine for processing the different types of data available these days. The distribution, scheduling and monitoring of clusters is done by the Spark core. The high level components required for processing the diverse workloads such as structured or streaming data are powered by the spark core. "These components are designed to inter-operate closely letting you combine them like libraries in a software project [2]."

The Spark core and the higher level libraries on top of the core are tightly integrated meaning when updates or improvements are implemented in the spark core help improve the spark libraries as well. Tight integration also makes it easier to write applications combining different workloads. This is explained nicely in the following example. One can build an application using machine learning libraries to process real time data from streaming sources and analysts can simultaneously access the data using SQL also in real time. In this example three different workloads namely SQL, streaming data and machine learning algorithms can be implemented in a single system which is a requirement in today's age of big data.

SPARK COMPONENTS

Figure 1 depicts the various building blocks of the spark stack. The Spark Core is computational engine which performs the task scheduling, distribution, and cluster monitoring tasks. Resilient Distributed Datasets(RDD) [3] and Directed Acyclic Graphs(DAG) are two important concepts in Spark. The libraries or packages supporting the diverse workloads are built on top of the Spark core. These packages include Spark SQL, Spark Streaming, MLlib (machine learning library) and GraphX.

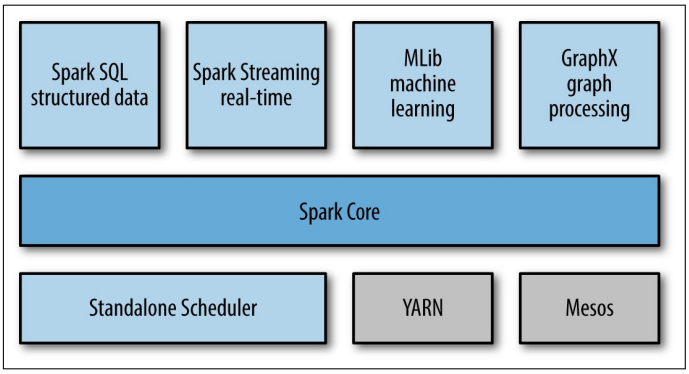


Fig. 1. Spark Components [2]

Spark Core

Spark Core is the foundation framework that provides basic I/O functionality, distributed task scheduling and dispatching. more [1]."

Resilient Distributed Datasets(RDD)

Resilient Distributed Datasets(RDD) [3] are Spark's primary abstraction, which are a fault-tolerant collection of elements that can be operated in parallel. RDDs are immutable once they are created but they can be transformed or actions can be performed on them [1]. Users can create RDDs through external sources or by transforming another RDD. Transformations and Actions are the two types of operations supported by RDDs.

1. Transformations: Since RDDs are immutable the transformations return a new RDD and not a single value." Transformations are lazily evaluated, i.e. they are not computed

immediately. They are executed only when an action runs on it. Some of the Transformation functions are map, filter, ReduceByKey, FlatMap and GroupByKey [1]."

2. Actions are operations that result in a return value after computation or triggers a task in response to some operation.Some Action operations are first, take, reduce, collect, count, foreach and CountByKey [1].

As such, RDDs are ephemeral disk, which means they do not persist data, however, users can explicitly persist RDDs to ease data reuse. Traditional distributed computing systems provide fault tolerance through checkpoint or data replication. "RDDs provide fault tolerance by logging the transformations used to build a data set through (its lineage) rather than actual data [3]." If one of the RDD fails, it has enough information about of its lineage so as to recreate the dataset from other RDDs, thus saving cost and time.

Directed Acyclic Graph(DAG)

Directed Acyclic Graph(DAG), which supports a cyclic data flow, "consists of finitely many vertices and edges, with each edge directed from one vertex to another, such that there is no way to start at any vertex v and follow a consistently-directed sequence of edges that eventually loops back to v again [4]." When we run any application in spark, the driver program converts the transformations and actions to logical directed acyclic graphs(DAG). The DAGs are then converted to physical execution plans with a set of stages which are distributed and bundled into tasks. These tasks are distributed among the different worker nodes for execution.

Spark SQL

Spark SQL[2] is a library built on top of the Spark Core to support querying structured data using SQL or Hive Query Language. It allows users to perform ETL (Extract, Transform and Load) operations on data from various sources such as JSON, Hive Tables and Parquet. Developers can "intermix SQL queries with programmatic data manipulations supported by RDDs in Python, Java, and Scala, all within a single application [2]."

Spark Streaming

Spark Streaming [2] library enables Spark to process real time data. Examples of streaming data are messages being published to a queue for real time flight status update or the log files for a production server. "Spark Streaming provides an API for manipulating data streams that closely matches the Spark Core's RDD API, making it easy for programmers to learn the project and move between applications that manipulate data stored in memory, on disk, or arriving in real time." Spark Streaming is designed such that it provides the same level of fault tolerance, throughput and scalability as the Spark Core.

MLlib

MLlib [2] is rich library of machine learning algorithms for Spark which can be accessed from Java, Scala as well as Python. It provides Spark with machine learning algorithms such as "classification, regression, clustering, and collaborative filtering". It also provides machine learning functionality such as "model evaluation and data import". The common machine learning algorithms include K-means, navie bayes, logistic regression, Principal component analysis and so on.

GraphX

GraphX introduces the Resilient Distributed Property Graph, which is directed multi-graph having properties attached to each edge and vertex. GraphX includes a set of operators like aggregateMessages, subgraph and joinVertices, and optimized variant of Pregel API. It also includes builders and graph algorithms to simplify graph analytics tasks [1].

RUNTIME ARCHITECTURE

The runtime architecture of Spark, illustrated in Figure 2, consists of a driver program, a cluster manager, workers or executors and the HDFS (Hadoop Distributed File System) [1]. Spark uses a master/slave architecture in which the driver program is the master and worker nodes or executors are the slaves. The driver runs the main() method of the user program which creates the SparkContext, the RDDs and performs transformations and actions [2].

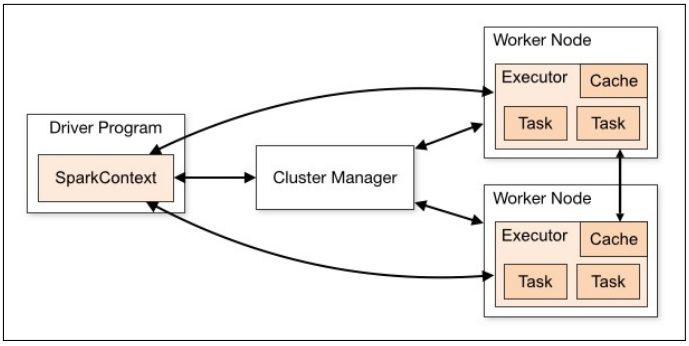


Fig. 2. Spark Architecture [5]

When we launch an application using the Spark Shell it creates a driver program which in turn initializes the SparkContext. Each spark application has its own SparkContext object which is responsible for the entire execution of the job. The SparkContext object then connects to cluster manager to request resources for its workers. The cluster manager provide executers to worker nodes, which are used to run the logic and also store the application data. The driver will send the tasks to the executors based on the data placement. The executors register themselves with the driver, which helps the driver keep tabs on the executors. Driver can also schedule future tasks by caching or persisting data. The following Cluster Managers are used in Spark based on the requirement-

- Standalone cluster manager is a simple cluster manager built into Spark to manage is own clusters [5].
- Apache Mesos is a dedicated cluster manager that provides Spark with rich resource scheduling capabilities [5].
- YARN is the only cluster manager in Spark that provides security support. "It allows dynamic sharing and central configuration of the same pool of cluster resources between various frameworks that run on YARN [6]".

Educational Resources

The Apache Spark website has a detailed documentation on the how to get started with spark [7]. It explains the concepts and shows examples to help us familiarize with Spark.

ACKNOWLEDGEMENTS

This paper is written as part of the I524: Big Data and Open Source Software Projects coursework at Indiana University. We would like to thank our Prof. Gregor von Laszewski, Prof. Gregory Fox and the AIs for their help and support

REFERENCES

[1] A. Bansod, "Efficient big data analysis with apache spark in hdfs," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 4, no. 6, pp. 313–316, aug 2015.

[2] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analytics*, 1st ed. O'Reilly Media, Inc., feb 2015. [Online]. Available: https://www.safaribooksonline.com/library/view/learning-spark/9781449359034/

[3] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauly, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012, San Jose, CA, USA, April 25-27, 2012*. Berkeley, CA, USA: USENIX Association, 2012, pp. 15–28. [Online]. Available: https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/zaharia

[4] "Directed acyclic graph - wikipedia," Article, accessed: 02-26-2017. [Online]. Available: https://en.wikipedia.org/wiki/Directed_acyclic_graph

[5] A. S. Foundation, "Cluster mode overview - spark 2.1.0 documentation," Article, accessed: 02-22-2017. [Online]. Available: http://spark.apache.org/docs/latest/cluster-overview.html

[6] "Apache spark ecosystem and spark components," Feb. 2016, apache Spark Ecosystem and Spark Components. [Online]. Available: https://www.dezyre.com/article/apache-spark-ecosystem-and-spark-components/219

[7] A. S. Foundation, "Spark programming guide - spark 2.1.0 documentation," Article, accessed: 02-22-2017. [Online]. Available: http://spark.apache.org/docs/latest/programming-guide.html

An overview of HadoopDB and its Architecture

KARTHIK ANBAZHAGAN[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: kartanba@iu.edu

February 27, 2017

With an explosion in data available for analysis, database management for the analytical applications is rapidly changing from high-end proprietary machines towards a cheaper, lower-end hardware, and virtual environment inside clouds. This paper speculates the feasibility of using a cost efficient system that is a hybrid between a parallel database and a MapReduce-based system that takes the best features of both systems.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, MapReduce, I524

https://github.com/kartanba/sp17-i524/blob/master/paper1/S17-IR-2008/report.pdf

1. INTRODUCTION

The proliferation of sensors and data capturing devices, coupled with the evolution and the increase in automation process has resulted in the capture and storage of data. The amount of data that is stored and processed by analytical database systems are growing at an immense rate. More and more historical data are being accumulated and kept online for future analysis. With the data explosion problem, companies are forced to upgrade themselves to a cheaper yet more efficient database management system. Since the data analysis workloads consist of many large-scale operations, complex aggregations, and star schema joins, parallelization of databases across nodes in a cluster and usage of a MapReduce system is increasingly getting more attention. MapReduce or its available alternate as open source Hadoop processes structured data, and can do so at tremendous scale. Hadoop is being used to manage Facebook's 2.5 petabyte data warehouse.

HadoopDB[1] is and open source component that serves as exactly such a hybrid system. It is a hybrid of a parallel database and MapReduce technologies. It approaches parallel databases in performance and efficiency, and also yields the scalability, fault tolerance, and flexibility of MapReduce systems. It uses PostgreSQL as the database layer and Hadoop as the communication layer, Hive as the translation layer. It has been demonstrated on clusters with 100 nodes and should scale as long as Hadoop scales.

2. COMPONENTS OF HADOOPDB

A HadoopDB [2] system consists of two main components whose best of features it tries to adapt and provide a hybrid system of the two. This section describes in brief about the components and describes the best features and the shortfalls of the two main components of HadoopDB.

2.1. Parallel DBMSs

The parallel DBMS are the most recent systems that are support standard relational tables and SQL. In this DBMS the data partition is transparent to the end-user. Parallel databases use an optimizer tailored for distributed workloads into a query plan whose execution is divided equally among multiple nodes.

Parallel databases can achieve especially high performance when administered by someone who can carefully design, deploy, tune, and maintain the system. It also has a very flexible query interface property which contains an implementation of SQL and ODBC which are an important part of the analytical data management picture. Parallel databases however generally do not score well on the fault tolerance and ability to operate in a heterogeneous environment properties.

2.2. MapReduce systems

MapReduce system processes data distributed (and replicated) across many nodes in a shared-nothing cluster via three basic operations. First, a set of Map tasks are processed in parallel by each node in the cluster without communicating with other nodes. Next, data is repartitioned across all nodes of the cluster. Finally, a set of Reduce tasks are executed in parallel by each node on the partition it receives. Unlike parallel database, MapReduce does not create a detailed query execution plan. It is determined at runtime. This allows MapReduce to assign more tasks to faster nodes and reassigning tasks from failed nodes.

One of the biggest issue with MapReduce is performance. By not requiring the user to first model and load data before processing, many of the performance enhancing tools used by database systems are not possible. MapReduce has a flexible query interface; Map and Reduce functions are just arbitrary

computations written in a general-purpose language. It best meets the fault tolerance and ability to operate in heterogeneous environment properties. It achieves fault tolerance by detecting and reassigning Map tasks of failed nodes to other nodes in the cluster.

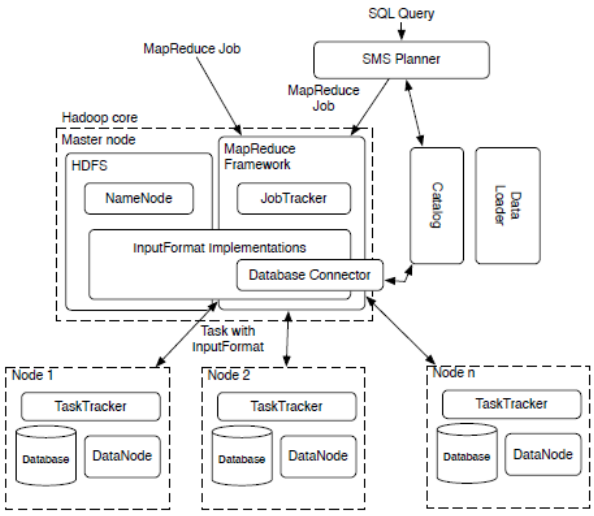


Fig. 1. Architecture of HadoopDB system

3. ARCHITECTURE

Fig 1. shows the architecture[3] of a model for using the HadoopDB system. It is very essential to understand every component of the architecture to understand how hadoopDB works. This section describes in brief about the components that are part of the hadoopDB architecture.

As shown in Figure 1., HadoopDB connects to multiple single node database systems using Hadoop as the task coordinator and network communication layer. Queries are parallelized across nodes using the MapReduce framework. HadoopDB achieves fault tolerance and the ability to operate in heterogeneous environments by inheriting the scheduling and job tracking implementation from Hadoop.

3.1. HadoopDB Layers

At the heart of HadoopDB is the Hadoop[4] framework. Hadoop consists of two layers:
(i) a data storage layer or the Hadoop [5] Distributed File System (HDFS)
(ii) a data processing layer or the MapReduce Framework.

HDFS[6] is a block-structured file system managed by a central NameNode. Individual files are broken into blocks of a fixed size and distributed across multiple DataNodes in the cluster. The NameNode maintains metadata about the size and location of blocks and their replicas. The MapReduce Framework follows a simple master-slave architecture. The master is a single JobTracker and the slaves or worker nodes are TaskTrackers. The JobTracker handles the runtime scheduling of MapReduce jobs and maintains information on each TaskTracker's load and available resources. Each job is broken down into Map tasks based on the number of data blocks that require processing, and Reduce tasks. The JobTracker assigns tasks to TaskTrackers based on locality and load balancing. It load-balances by ensuring all available TaskTrackers are assigned tasks. TaskTrackers regularly update the JobTracker with their status through heartbeat messages.

3.2. Other Components of HadoopDB

3.2.1. Database Connector

The Database Connector is the interface between independent database systems within the nodes in the multinode cluster and TaskTrackers. Each MapReduce job supplies the Connector with an SQL query and connection parameters such as which JDBC driver to use, query fetch size and other query tuning parameters. The Connector connects to the database, executes the SQL query and returns results as key-value pairs.

3.2.2. Catalog

The catalog maintains information about the databases:
(i) connection parameters such as database location, driver class and credentials,
(ii) metadata such as data sets contained in the cluster, replica locations, and data partitioning properties.

The current implementation of the HadoopDB catalog stores its information as an XML file in HDFS. This file is accessed by the JobTracker and TaskTrackers to retrieve information necessary to schedule tasks and process data needed by a query.

3.2.3. Data Loader

The Data Loader is responsible for
(i) globally repartitioning data on a given partition key upon loading,
(ii) breaking apart single node data into multiple smaller partitions or chunks and
(iii) finally bulk-loading the single-node databases with the chunks.

The Data Loader consists of two main components: Global Hasher and Local Hasher. The Global Hasher executes a custom made MapReduce job over Hadoop that reads in raw data files stored in HDFS and repartitions them into as many parts as the number of nodes in the cluster. The Local Hasher then copies a partition from HDFS into the local file system of each node and secondarily partitions the file into smaller sized chunks based on the maximum chunk size setting.

3.2.4. SQL to MapReduce to SQL (SMS) Planner

HadoopDB provides a parallel database front-end to data analysts enabling them to process SQL queries. The SMS planner extends Hive[7] which transforms HiveQL, a variant of SQL, into MapReduce jobs that connect to tables stored as files in HDFS. The MapReduce jobs consist of DAGs of relational operators such as filter, select, join, aggregation that operates as iterators. Since each table is stored as a separate file in HDFS, Hive assumes no collocation of tables on nodes.

4. CONCLUSION

This paper has given an overview and the architecture of HadoopDB which give a clear picture of the hybrid system. We see that HadoopDB does not replace Hadoop. Both systems coexist enabling the analyst to choose the appropriate tools for a given dataset and task. We also find that HadoopDB can approach the performance of parallel database systems while achieving similar scores on fault tolerance, an ability to operate in heterogeneous environments, and software license cost as Hadoop.

HadoopDB is, therefore, a hybrid of the parallel DBMS and Hadoop approaches to data analysis, achieving the performance and efficiency of parallel databases, yet still yielding the scalability, fault tolerance, and flexibility of MapReduce-based systems. The ability of HadoopDB to directly incorporate Hadoop and open source DBMS software makes HadoopDB particularly flexible and extensible for performing data analysis at the large scales expected of future workloads.

REFERENCES

[1] B. Lorica, "Hadoopdb: An open source parallel database," July 2009. [Online]. Available: http://radar.oreilly.com/2009/07/hadoopdb-an-open-source-parallel-database.html

[2] B. Pawlikowski, A. Abouzeid, D. Abadi, and A. Silberschatz, "Hadoopdb project," August 2009. [Online]. Available: db.cs.yale.edu/hadoopdb/hadoopdb.html

[3] HadoopDB, *HadoopDB Quick Start Guide*, Yale University, 2009 July. [Online]. Available: http://hadoopdb.sourceforge.net/guide/

[4] Wikipedia, "Apache hadoop." [Online]. Available: https://en.wikipedia.org/wiki/Apache_Hadoop

[5] ApacheFoundation, "Hadoop web page," Web Page, January 2017. [Online]. Available: hadoop.apache.org/core/

[6] ApacheFoundation, "Hadoop cluster setup," January 2017. [Online]. Available: https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/ClusterSetup.html

[7] ApacheFoundation, "Hive as a contrib project," December 2011. [Online]. Available: issues.apache.org/jira/browse/HADOOP-3601

Ansible

ANURAG KUMAR JAIN[1] AND GREGOR VON LASZEWSKI[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: laszewski@gmail.com

Paper 1, March 3, 2017

Ansible is a powerful open source automation tool which can be used in configuration management, deployment, and orchestration tool [1]. This paper gives an in-depth overview of Ansible.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, Ansible, Playbook, Roles

https://github.com/anurag2301/sp17-i524

INTRODUCTION

Ansible is an open source automation engine which can be used to automate cloud provisioning, configuration management, and application deployment. It is designed to be minimal in nature, secure, consistent, and highly reliable [2]. In many respects, it is unique from other management tools and aims to provide large productivity gains. It has an extremely low learning curve and seeks to solve major unsolved IT challenges under a single banner. Michael DeHaan developed the Ansible platform and Ansible, Inc. was the company set up to commercially support and sponsor Ansible. The company was later acquired by Red Hat Inc. It is available on Fedora, Red Hat Enterprise Linux, CentOS and other operating systems [1].

ARCHITECTURE

One of the primary differences between Ansible and many other tools in the space is its architecture. Ansible is an agentless tool, it doesn't requires any software to be installed on the remote machines to make them manageable. By default is manages remote machines over SSH or WinRM, which are natively present on those platforms [3].

Like many other configuration management software, Ansible distinguishes two types of servers: controlling machines and nodes. Ansible uses a single controlling machine where the orchestration begins. Nodes are managed by a controlling machine over SSH. The location of the nodes are described by the inventory of the controlling machine [2].

Modules are deployed by Ansible over SSH. These modules are temporarily stored in the nodes and communicate with the controlling machine through a JSON protocol over the standard output.[3]

The design goals of Ansible includes consistency, high reliability, low learning curve, security and to be minimalistic in nature. The security comes from the fact that Ansible doesn't requires users to deploy agents on nodes and manages remote machines using SSH or WinRM. Ansible doesn't require dedicated users or credentials - it respects the credentials that the user supplies when running Ansible. Similarly, Ansible does not require administrator access, it leverages sudo, su, and other privilege escalation methods on request when necessary [4]. If needed, Ansible can connect with LDAP, Kerberos, and other centralized authentication management systems [5].

ADVANCED FEATURES

The Ansible Playbook language includes a variety of features which allow complex automation flow, this includes conditional execution of tasks, the ability to gather variables and information from the remote system, ability to spawn asynchronous long running actions, ability to operate in either a push or pull configuration, it also includes a "check" mode to test for pending changes without applying change, and the ability to tag certain plays and tasks so that only certain parts of configuration can be applied [2]. These features allow your applications and environments to be modelled simply and easily, in a logical framework that is easily understood not just by the automation developer, but by anyone from developers to operators to CIOs. Ansible has low overhead and is much smaller when compared to other tools like Puppet [6].

PLAYBOOK AND ROLES

Playbook is what Ansible uses to perform automation and orchestration. They are Ansible's configuration, deployment and orchestration language. They can be used to describe policy you need your remote systems to enforce, or a set of steps in a general IT process [7].

At a basic level, playbooks can be used to manage configurations and deployments to remote machines. While at an advanced level, they can be utilized to sequence multi-tier rollouts which involves rolling updates, and can also be used to delegate actions to other hosts, interacting with monitoring servers and load balancers at the same time.

Playbooks consists of series of 'plays' which are used to define automation across a set of hosts, known as the 'inventory'. These 'play' generally consists of multiple 'tasks,' that can select one, many, or all of the hosts in the inventory where each task is a call to an Ansible module - a small piece of code for doing a specific task. The tasks may be complex, such as spinning up an entire cloud formation infrastructure in Amazon EC2. Ansible includes hundreds of modules which help it perform a vast range of tasks [3].

Similar to many other languages Ansible supports encapsulating Playbook tasks into reusable units called 'roles.' These roles can be used to easily apply common configurations in different scenarios, such as having a common web server configuration role which can be used in development, test, and production automation. The Ansible Galaxy community site contains thousands of customizable rules that can be reused used to build Playbooks.

COMPLEX ORCHESTRATION USING PLAYBOOKS

Playbook can be used to combine multiple tasks to achieve complex automation [7]. Playbook and Ansible can be easily used in implementing a cluster-wide rolling update that consists of consulting a configuration/settings repository for information about the involved servers, configuring the base OS on all machines and enforcing desired state. It can also be used in signaling the monitoring system of an outage window prior to bringing the servers offline and signaling load balancers to take the application servers out of a load balanced pool. The usage doesn't ends here and it can be utilized to start and stop server, running appropriate tests on the new server or even deploying/updating the server code, data, and content. Ansible can also be used to send email reports and as a logging tool [3].

EXTENSIBILITY

Tasks in Ansible are performed by Ansible 'modules' which are pieces of code that run on remote hosts. Although there are a vast set of modules which cover a lot of things which a user may require there might be a need to implement a new module to handle some portion of IT infrastructure. Ansible makes it simpler and by allowing the modules to be written in any language, with the only requirement that they are required to take JSON as input and product JSON as output [3].

We can also extend Ansible through it's dynamic inventory which allows Ansible Playbooks to run against machines and infrastructure discovered during runtime. Out of the box Ansible includes support for all major cloud providers, it can also be easily extended to add support for new providers and new sources as needed.

ONE TOOL TO DO IT ALL

Ansible is designed to make IT configurations and processes both simple to read or write, the code is human readable and can be read even by those who are not trained in reading those configurations. Ansible is different from genrally used software programming languages, it uses basic textual descriptions of desired states and processes, while being neutral to the types of processes described. Ansible's simple, task-based nature makes it unique and it can be applied to a variety of use cases which includes configuration management, application deployment, orchestration and need basis test execution.

Ansible combines these approaches into a single tool. This not only allows for integrating multiple disparate processes into a single framework for easy training, education, and operation, but also means that Ansible seamlessly fits in with other tools. It can be used for any of the above mentioned use cases without modifying existing infrastructure that may already be in use.

INTEGRATION OF CLOUD AND INFRASTRUCTURE

Ansible can easily deploy workloads to a variety of public and on-premise cloud environments. This capability includes cloud providers such as Amazon Web Services, Microsoft Azure, Rackspace, and Google Compute Engine, and local infrastructure such as VMware, OpenStack, and CloudStack. This includes not just compute provisioning, but storage and networks as well and the capability doesn't ends here. As noted, further integrations are easy, and more are added to each new Ansible release. And as Ansible is open source anyone can make his/her contributions [3].

AUTOMATING NETWORK

Ansible can easily automate traditional IT server and software installations, but it strives to automate IT infrastructure entirely, including areas which are not covered by traditional IT automation tools. Due to the Ansible's agentless, task-based nature it can easily be utilized in the networking space, and the inbuilt support is included with Ansible for automating networking from major vendors such as Cisco, Juniper, Hewlett Packard Enterprise, Cumulus and more [2].

By harnessing this networking support, network automation is no longer a task required to be done a separate team, with separate tools, and separate processes. It can easily be done by the same tools and processes used by other automation procedures you already have. Ansible tasks also include configuring switching and VLAN for a new service. So now users don't need a ticket filed whenever a new service comes in.

CONCLUSION

Ansible is an open source community project sponsored by Red Hat. It is one of the easiest way to automate IT. Ansible code is easy to read and write and the commands are in human readable format. The power of Ansible language can be utilized across entire IT teams – from systems and network administrators to developers and managers. Ansible includes thousands of reusable modules which makes it even more user friendly and users can write new modules in any language which makes it flexible too. Ansible by Red Hat provides enterprise-ready solutions to automate your entire application lifecycle – from servers to clouds to containers and everything in between. Ansible Tower by Red Hat is a commercial offering that helps teams manage complex multi-tier deployments by adding control, knowledge, and delegation to Ansible-powered environments.

REFERENCES

[1] Wikipedia, "Ansible (software)," Web Page, Feb. 2017, accessed: 2017-02-20. [Online]. Available: https://en.wikipedia.org/wiki/Ansible_(software)

[2] Red Hat Inc., "Ansible in depth," Web Page, Feb. 2016, accessed: 2017-02-23. [Online]. Available: https://www.ansible.com/ansible-in-depth-whitepaper

[3] Red Hat Inc., “How ansible works,” Web Page, Feb. 2016, accessed: 2017-02-14. [Online]. Available: https://www.ansible.com/how-ansible-works
[4] Michael DeHaan, “Ansible,” Code Repository, Feb. 2017, accessed: 2017-02-21. [Online]. Available: https://github.com/ansible/ansible
[5] Red Hat Inc., “Ansible documentation,” Web Page, Feb. 2016, accessed: 2017-02-15. [Online]. Available: https://docs.ansible.com/ansible/index.html
[6] UpGuard Inc., “Ansible vs puppet,” Web Page, Feb. 2017, accessed: 2017-02-24. [Online]. Available: https://www.upguard.com/articles/ansible-puppet
[7] Red Hat Inc., “Playbooks,” Web Page, Feb. 2016, accessed: 2017-02-24. [Online]. Available: https://docs.ansible.com/ansible/playbooks.html

Lustre File System

PRATIK JAIN

School of Informatics and Computing, Bloomington, IN 47408, U.S.A.

Corresponding authors: jainps@iu.edu

Paper-1, February 28, 2017

This paper talks about the Lustre file system and gives a brief overview on its applications in the industry, its architecture, and the steps required for successfully installing and configuring the file system. Alongside its various applications in various fields like HPC and Big Bata, the paper also throws light on the areas in which the Lustre file system is not recommended.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Lustre File System, Object based file system, Object based storage device

https://github.com/pratik11jain/sp17-i524/blob/master/paper1/S17-IR-2012/report.pdf

INTRODUCTION

Lustre is a type of parallel distributed file system, started as a research project in 1999 by Peter J. Braam, and is now, generally used for large-scale cluster computing. The name Lustre is a combination of Linux and cluster [1]. It is often used in supercomputers due to its high-performance capabilities and open licensing. Lustre file systems are scalable and can be part of multiple computer clusters with tens of thousands of client nodes, tens of petabytes of storage on hundreds of servers, and more than a terabyte per second of aggregate I/O throughput. A Lustre file system was first installed for production use in March 2003, on one of the largest supercomputers at the time, the MCR Linux Cluster at Lawrence Livermore National Laboratory. Lustre file system software is available under the GNU General Public License and can be utilized for computer clusters ranging in size from small workgroup clusters to large-scale, multi-site clusters. This makes Lustre file systems a popular choice for businesses with large data centers, including in various industries such as simulation, life science, meteorology, rich media, oil and gas, and finance.

ARCHITECTURE

The Lustre architecture is a storage architecture for clusters. Its central component, the Lustre file system, is supported on the Linux operating system and provides a POSIX standard-compliant UNIX file system interface. The architecture is used for many different kinds of clusters and is best known for powering many of the largest HPC clusters worldwide [2]. Lustre file system is used by many HPC sites as a site-wide global file system, serving dozens of clusters. Its ability to scale capacity and performance for any need reduces the need to deploy many separate file systems, such as one for each compute cluster and avoiding the need to copy data between compute clusters simplifies storage management. In addition to aggregating storage capacity of many servers, the I/O throughput is also aggregated and scales with additional servers. Also, throughput and/or capacity can be easily increased by adding servers dynamically. Lustre's scalable architecture has three main components, first, the Metadata Server that provides metadata services for a file system and manages a Metadata Target that stores the file metadata, second, the Object Storage Servers that manage the Object Storage Targets that store the file data objects and third, the clients that access and use the data. Lustre presents all clients with a unified namespace for all of the files and data in the file system and allows concurrent and coherent read and write access to the files in the filesystem [3].
Following are the principal foundations of Lustre:

Object-based storage devices

Unlike conventional storage devices, an Object Based Disk (OBD) or Object-Based Storage Device (OBSD) is one that works at the level of files, rather than at the level of individual blocks. The OBD keeps track of allocated objects, which blocks belong to each object, free space, etc. internally, rather than exposing these details to the operating system. This architecture looks at devices that can manipulate file objects. Typical commands executed as part of the object interface are create, destroy, read/write block X in object N and read/write attributes of objects [4].

A stackable object driver model

In addition to direct drivers which control storage, there are logical object drivers, client object drivers and associated target drivers. For example, RAID can be implemented by having a logical object driver that speaks with multiple direct drivers [5]. Other interesting logical drivers can perform HSM, parallel I/O and cryptographic operations. Lustre's logical object driver

manages snapshots of file systems. Client drivers are responsible for packing up object requests and shipping them to targets and this can exploit SAN's such as Fibre Channel, InfiniBand and Gigabit Ethernet. Since the interface is uniform, logical drivers can be stacked on top of direct drivers or clients.

Object-based file systems

There are at least three types of file systems that can be imagined in the object storage model. OBDFS is an object-based file system that is meant for use with non-shared storage devices. An inode file system provides direct access to objects named by object id. Third are cluster file systems. The traditional cluster file system can become significantly simpler than those implemented with shared block storage devices.

LUSTRE FILE SYSTEM AND STRIPING

The ability to stripe data across multiple OSTs in a round-robin fashion is one of the main factors leading to the high performance of Lustre file systems. Users can optionally configure for each file the number of stripes, stripe size, and OSTs that are used. Striping can be used to improve performance when the aggregate bandwidth to a single file exceeds the bandwidth of a single OST. The ability to stripe is also useful when a single OST does not have enough free space to hold an entire file.

IMPLEMENTATION

In a typical Lustre installation on a Linux client, the filesystem driver module is loaded into the kernel and the filesystem is mounted like any other local or network filesystem. Even though it may be composed of tens to thousands of individual servers and filesystems, client applications see a single, unified filesystem. On some massively parallel processor (MPP) installations, computational processors can access a Lustre file system by redirecting their I/O requests to a dedicated I/O node configured as a Lustre client [6]. Another approach used in the early years of Lustre is the user-level liblustre library which provided userspace applications with direct filesystem access. Liblustre allows computational processors to mount and use the Lustre file system as a client. Using liblustre, the computational processors could access a Lustre file system even if the service node on which the job was launched is not a Linux client. Liblustre allowed direct data movement between application space and the Lustre OSSs and did not require an intervening data copy through the kernel, thus providing access from computational processors to the Lustre file system directly in a constrained operating environment.

INSTALLATION

Following is the overview of steps needed for installing Lustre. The first step is to setup Lustre Filesystem Hardware. Lustre runs on most commodity hardware with any kind of block storage device including single disks, software and hardware RAID and logical volume manager. For servers, 64-bit architectures are recommended. Lustre allows for multiple MDSes for high availability. The size of the MDT's backing file system should be chosen based on the total number of files planned to be stored in the Lustre file system, and the aggregate OST space should be chosen based on the total amount of data planned to be stored in the file system. Estimating space requirements early can dictate hardware requirements. After this, the Lustre software is installed. Lustre runs on a variety of Linux kernels from Linux distributions including RHEL, CentOS, and SLES. When using the Lustre ldiskfs OSD only, it will be necessary to patch the kernel before building Lustre. The required Lustre RPMs or source can be downloaded here [7]. Metadata and Object Storage Server require the Lustre patched Linux kernel, Lustre modules, Lustre utilities and e2fsprogs installed. The clients require the Lustre client modules, client utilities and, optionally, the Lustre patched kernel. For configuring Lustre, the Lustre Networking (LNET) kernel modules have a variety of module parameters that can be set in the /etc/modprobe.d/lustre.conf file. The type of network used and globally-available networks can be specified along with routes in a Lustre configuration. To set up and tune the filesystem, Lustre provides a variety of configuration utilities that include mkfs.lustre to format a disk for a Lustre service, tunefs.lustre to modify configuration information on a Lustre target disk, lctl to directly control Lustre via an ioctl interface and mount.lustre to start a Lustre client or target service.

WHERE NOT TO USE IT?

Although a Lustre file system can function in many work environments, it is not necessarily the best choice for all applications [2]. It is best suited for uses that exceed the capacity that a single server can provide, though in some use cases, a Lustre file system can perform better with a single server than other file systems due to its strong locking and data coherency. A Lustre file system is not particularly well suited for "peer-to-peer" usage models where clients and servers are running on the same node, each sharing a small amount of storage, due to the lack of data replication at the Lustre software level. In such uses, if one client or server fails, then the data stored on that node will not be accessible until the node is restarted.

CONCLUSION

The Lustre file system is an open-source, parallel file system that supports many requirements of leadership class HPC simulation environments and enterprise environments worldwide. Because Lustre file systems have high performance capabilities and open licensing, it is often used in supercomputers. Lustre file systems are scalable and can be part of multiple computer clusters with tens of thousands of client nodes, tens of petabytes of storage on hundreds of servers, and more than a terabyte per second of aggregate I/O throughput. Lustre file systems a popular choice for businesses with large data centers, including those in industries such as meteorology, simulation, oil and gas, life science, rich media, and finance. Lustre provides a POSIX compliant interface and many of the largest and most powerful supercomputers on Earth today are powered by the Lustre file system. Its Architecture contains 3 main components - the Metadata Server, the Object Storage Servers and the clients. There are a few fields in which use of Lustre file system is not recommended. This paper covers the basic components of Lustre File system and gives an overview about the installation steps of lustre file system. It also talks about scenarios in which the use of lustre file system is not recommended. Thus the paper tries to briefly introduce the paradigm of lustre file system.

REFERENCES

[1] Aviso Legal, "Ungrid status report 2010," Web Page, Nov. 2010, accessed 2017-02-25. [Online]. Available: http://www.ungrid.unal.edu.co/cluster/status.htm

[2] Intel, “The lustre*software release 2.x operations manual,” Web Page, Dec. 2016, accessed 2017-02-25. [Online]. Available: http://doc.lustre.org/lustre_manual.xhtml
[3] OpenSFS, EOFS, “Getting started with lustre,” Web Page, Dec. 2016, accessed 2017-02-25. [Online]. Available: http://lustre.org/getting-started-with-lustre/
[4] Wikipedia, “Object storage,” Web Page, Feb. 2017, accessed 2017-02-13. [Online]. Available: https://en.wikipedia.org/wiki/Object_storage#Object-based_storage_devices
[5] OpenSFS, EOFS, “About the lustre file system,” Web Page, Dec. 2016, accessed 2017-02-25. [Online]. Available: http://lustre.org/about/
[6] Wikipedia, “Blue gene,” Web Page, Jan. 2017, accessed 2017-02-13. [Online]. Available: https://en.wikipedia.org/wiki/Blue_Gene
[7] OpenSFS, EOFS, “Download lustre,” Web Page, Dec. 2016, accessed 2017-02-25. [Online]. Available: http://lustre.org/download-lustre/

An overview of Flume and its Applications in BigData

SAHITI KORRAPATI[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: sakorrap@iu.edu, S17-IR-2013*

techpaper-1, March 7, 2017

This paper provides an overview of Apache Flume and its relevance in BigData. Data Analysis is only half the battle when it comes to Big Data; getting the huge volumes of data to Hadoop is the first step in any Big Data project. Apache Flume comes handy in bringing this data to a centralized store such as Hadoop Distributed File Systems.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Flume, Apache Software, BigData

https://github.com/sakorrap/sp17-i524/tree/master/paper1/S17-IR-2013/report.pdf

INTRODUCTION

Big Data consists of large volume, high speed, and wide variety of data generated from various sources [1]. Hadoop framework allows for the distributed processing of large data sets across clusters of computers using simple programming models [2]. Apache Flume is one way to get this large data into Hadoop Systems.

According to Apache, Flume is a distributed, reliable, and available service for moving large amounts of data soon after the data is produced. The primary use case for Flume is as a logging system that gathers a set of log files on every machine in a cluster and aggregates them to a centralized persistent store such as the Hadoop Distributed File System (HDFS) [3].

ARCHITECTURE

Flume is an application that allows collecting data from origin logs and sends it to a destination like Hadoop systems. This is achieved by defining dataflows consisting of sources, channels and sinks. These are the three primary structures which make up any Flume dataflow. The unit of data that flows through Flume is called an event, and the JVM process that runs the dataflow is called agent [3].

Data flow

Figure 1 shows the architecture of Flume. An external source delivers events to the Flume source. Flume source receives the event and stores it into its channels. The channel is a pathway between Flume Source and Flume Sink. The sink sends the event to an external repository like HDFS or forwards it to the Flume agent of the next hop in the flow. The source and sink run asynchronously with the events in the channel within the given agent. Events travel through agents of multiple hops before reaching the final destination. So, Flume allows fan-in and fan-out flows, contextual routing and fail-over routes for failed hops [4].

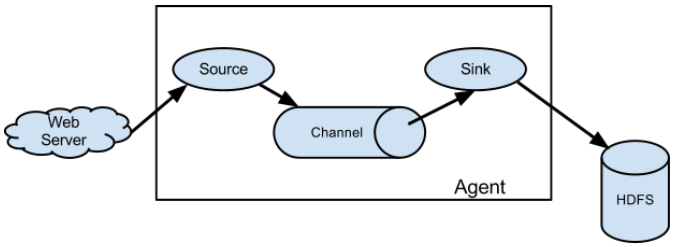


Fig. 1. Architecture of Flume [3]

Reliability and recover-ability

The events that are put in channel are removed from it only after they are stored in the channel of next agent or in the terminal repository, maintaining end-to-end reliability of the flow. The transactional approach of the Flume guarantees the reliable delivery of the events. The end-to-end reliability is guarantied by writing the event to disk in a 'write-ahead log' (WAL). When the agent crashes and restarts, knowledge of the event is not lost. After the event has successfully made its way to the end of its flow, an acknowledgment is sent back to the originating agent so that it knows it no longer needs to store the event on disk. This way, the set of events are reliably passed from point to point in the flow. In case of a multi-hop flow, sink from the previous hop and source from the next hop both run transactions to ensure that the data is safely transferred to the channel of the next hop.

Figure 2 shows the Transaction Interface of Flume.

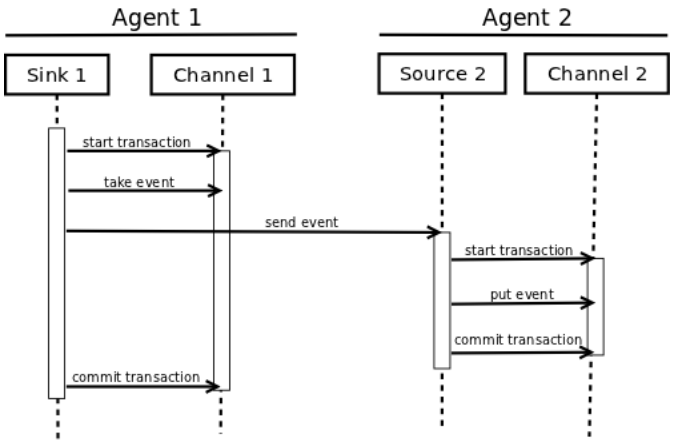


Fig. 2. Transaction Interface of Flume [5]

SETUP AND CONFIGURATION

Setting up an agent

Flume agent configuration is text file in Java properties file format. One or more agents can be configured in the same configuration file which specifies properties of each source, sink and channel in an agent [3].

Starting an Agent

Shell script called flume-ng in the bin directory of Flume starts an Agent. Agent name, the config directory, and the config file should be specified as arguments as given below [3]:

```
bin/flume-ng agent -n \$agent\_name -c conf
-f conf/flume-conf.properties.template
```

Properties of Sources, Sinks and Channels should be configured in configuration file. It can be configured as shown below [3]:

```
# Name the components on this agent
a1.sources = r1
a1.sinks = k1
a1.channels = c1

# Describe/configure the source
a1.sources.r1.type = netcat
a1.sources.r1.bind = localhost
a1.sources.r1.port = 41114

# Describe the sink
a1.sinks.k1.type = logger

# Use a channel which buffers events in memory
a1.channels.c1.type = memory
a1.channels.c1.capacity = 1600
a1.channels.c1.transactionCapacity = 100

# Bind the source and sink to the channel
a1.sources.r1.channels = c1
```

Given this configuration file, we can start Flume as follows [3]:

```
bin/flume-ng agent --conf conf
--conf-file example.conf --name a1
-Dflume.root.logger=INFO,console
```

Note that in a full deployment we would typically include one more option: –conf=<conf-dir>. The <conf-dir> directory would include a shell script flume-env.sh and potentially a log4j properties file. In this example, we pass a Java option to force Flume to log to the console and we go without a custom environment script [3].

Logging Raw Data

Flume does not log raw stream of data is not desired in many production environments. Flume attempts to provide clues for debugging the problems like broken pipeline. One way to debug is by connecting an additional Memory Channel connected to a Logger Sink which will output event data to the Flume logs. In some situations, however, this approach is insufficient. For this, Java system properties should be set in addition to log4j properties [3].

To enable configuration-related logging, set the Java system property $-Dorg.apache.flume.log.printconfig = true$. This can either be passed on the command line or by setting this in the $JAVA\_OPTS$ variable in flume-env.sh.

To enable data logging, set the Java system property -Dorg.apache.flume.log.rawdata=true in the same way described above. For most components, the log4j logging level must also be set to DEBUG or TRACE to make event-specific logging appear in the Flume logs.

Here is an example of enabling both configuration logging and raw data logging while also setting the Log4j loglevel to DEBUG for console output [3]:

```
bin/flume-ng agent --conf
conf --conf-file example.conf
--name a1 -Dflume.root.logger=DEBUG,
console -Dorg.apache.flume.log.printconfig=true
-Dorg.apache.flume.log.rawdata=true
```

EXPERIMENTAL FEATURE

Zoopkeeper based configuration

Flume supports Agent configurations via Zookeeper. The configuration file is stored in its Node data. Zookeeper Node tree for agents a1 and a2 will be as follows [3]:

```
- /flume
 |- /a1 [Agent config file]
 |- /a2 [Agent config file]
```

Once the configuration file is uploaded, start the agent with following options [3]

```
bin/flume-ng agent -conf
conf -z zkhost:2181,zkhost1:2181
-p /flume -name a1
-Dflume.root.logger=INFO,console
```

LICENSING

Apache Flume is an open source software licensed under Apache License 2 terms, can be downloaded at Flume website [6]. Source code is available on GitHub [7].

USE CASES

Application logs, GPS tracking, social media updates, and digital sensors all constitute fast-moving streams requiring storage in the Hadoop Distributed File System (HDFS). An example of

the same is logging twitter data using Flume. Flume helps in gathering data from Twitter API source and storing it in HDFS systems. Flume agent can be configured to catch all the new twitter feeds that appear and automatically transfer them to Hadoop [8].

USEFUL RESOURCES

Flume website has both user manual [3] and developers manual [5] for further reference which covers how to configure it and use it with examples.

CONCLUSION

Some data destined for Hadoop clusters comes from sporadic bulk loading processes, such as database and mainframe offloads and batched data dumps from legacy systems. But what has made data really big in recent years is that most new data is contained in high-throughput streams. Flume efficiently helps in capturing and moving data from these high speed high volume generators to HDFS.

ACKNOWLEDGEMENTS

The authors thank Professor Gregor Von Laszewski and all the AIs of big data class for the guidance and technical support.

REFERENCES

[1] Tutorials Point, "Hadoop - big data overview," Web Page. [Online]. Available: https://www.tutorialspoint.com/hadoop/hadoop_big_data_overview.htm

[2] The Apache Software Foundation, "Welcome to apache hadoop," Web Page, January 2017. [Online]. Available: http://hadoop.apache.org/#Getting+Started

[3] Apache Flume, *Flume User Manual*, 1st ed., The Apache Software Foundation. [Online]. Available: https://flume.apache.org/FlumeUserGuide.html

[4] Cloudera, "Flume user guide," Web Page, March 2013. [Online]. Available: http://archive.cloudera.com/cdh/3/flume/UserGuide/

[5] Apache Flume, *Flume Developer Manual*, 1st ed., The Apache Software Foundation. [Online]. Available: https://flume.apache.org/FlumeDeveloperGuide.html

[6] ——, Web Page. [Online]. Available: https://flume.apache.org/download.html

[7] ——, Web Page. [Online]. Available: https://git-wip-us.apache.org/repos/asf?p=flume.git;a=tree;h=refs/heads/trunk;hb=trunk

[8] J. Natkins, "Analyzing twitter data with apache hadoop, part 2: Gathering data with flume," Web Page, October 2012. [Online]. Available: http://blog.cloudera.com/blog/2012/10/analyzing-twitter-data-with-hadoop-part-2-gathering-data-with-flume/

AUTHOR BIOGRAPHIES

Sahiti Korrapati is pursuing her MSc in Data Science from Indiana University Bloomington

An Overview of Apache Sqoop

HARSHIT KRISHNAKUMAR[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: harkrish@iu.edu, S17-IR-2014

Project-01, March 13, 2017

Big Data is increasingly being used in conjunction with RDBMS systems to perform every day analyses. In light of these requirements, we need to have an efficient system to transfer data between RDMBS and Hadoop systems. Sqoop provides an interface to efficiently manage data movement activities between these systems. This paper describes the different components and the operation of Sqoop

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524

https://github.com/harkrish1/sp17-i524/blob/master/paper1/S17-IR-2014/report.pdf

INTRODUCTION

Hadoop systems are known to efficiently store transactional or log data in distributed file systems, as opposed to the traditional RDBMS systems which store relatively smaller volumes of data in database tables. Often times we would need to combine the transaction logs with the RDBMS data to perform analyses. Sqoop is the method of transferring data between RDBMS and Hadoop systems. Often there will be multiple RDBMS sources to operate on, Sqoop helps in automating these tasks by providing specific connectors to Teradata, Netezza, Oracle, MySQL, Postgres, and HSQLDB.

Sqoop also leverages the parallel processing capabilities of Hadoop to transfer data. Sqoop integrates with Oozie, allowing developers to schedule and automate import and export tasks. Sqoop uses a connector based architecture which supports plugins that provide connectivity to new external systems.

Sqoop supports daily incremental data loads, production workflows for division of roles and administrators and also supports different security compliances. The latest release of Sqoop provides a rest API and Java API for easy integration along with a Hue UI and a command line client.[1]

HIGH LEVEL ARCHITECTURE

Figure 1 shows the High Level Architecture of how Sqoop works. Sqoop keeps different clients separate so that none of them have access to the entire set data, or rather have access to a specific subset for security purposes. Each client sends requests to a common Sqoop server which acts as an interface between client face and the actual servers. When Sqoop receives multiple requests, it prioritizes and sends the requests. The next step is to get the metadata from tables in RDBMS and use MapReduce to split the job between nodes. Sqoop server does not actually handle any data, rather it is just an agent which handles the jobs or requests. The data flows between Hadoop and RDBMS servers as instructed by Sqoop server[2].

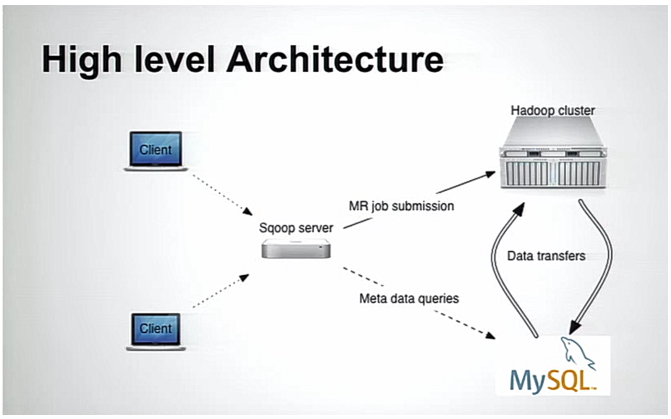


Fig. 1. High Level Architecture of Sqoop [2]

COMPONENTS OF SQOOP

Connector

Connectors are pluggable components that are used to communicate to RBDMS systems. There are pre defined connectors for many different type of databases, and there are generic JDBC connectors for any other new type of database. Connectors expose vital information like metadata to the Sqoop server. Connectors are responsible to move data in and out of RBDMs software.

Repository for Metadata

Connectors fetch metadata information from each RDBMS servers and store them in repositories. This helps the work

of Sqoop server in job management and to perform MapReduce on the requests.

Server

Servers have CRUD (create, read, update and delete) capabilities on the metadata repository. These capabilities are exposed via REST interface. Server also takes care of initiating data transfers. It is responsible for prioritising and managing the data transfer jobs. This also happens via REST interface. Sqoop server also monitors the jobs which are running to check their progress and failures. Sqoop server is independent of the actual data transfer and this is ideal for security purposes, since the actual data is not getting exposed. This also helps when Sqoop server is down for maintenance, the data transfers which are already in progress will continue to happen.

WORKFLOW OF SQOOP

Figure 1 shows the workflow of Sqoop. Sqoop server maintains the first two parts of the work flow - initializing and partitioning the data transfer jobs. The next step is handled by connectors which extract data from RDBMS systems. This happens in parallel since the partitioner works on MapReduce algorithms to divide the job into parallel streams. The output is then sent to different nodes of Hadoop systems to load the data. The destroyer works to clean up all the temporary tables left behind by the entire process. The workflow resembles an ETL flow without the transformation phase.

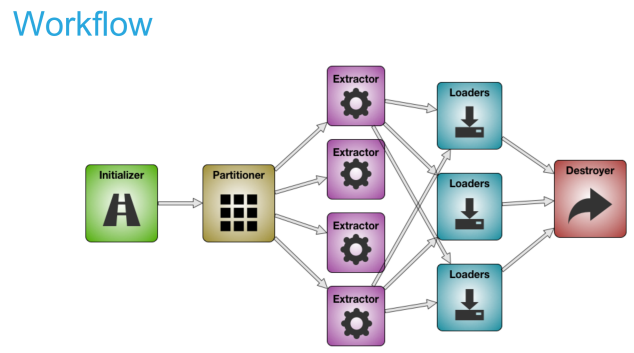


Fig. 2. Workflow Diagram [2]

LICENSING

Apache Sqoop is available as an open source software, provided to download via multiple mirrors in the Sqoop website[3]. The source code is also shared via GitHub [1] for developers to fork and modify it.

SHELL ACCESS

The shell code for importing data from MYSQL, to Hive, HBase and exporting data is given in Algorithms 1, 2, 3 and 4[1].

USE CASES

Importing data into Hadoop

One use case of Sqoop is to import data into Hadoop from MYSQL. In a typical banking scenario there can be a case where

Algorithm 1. MYSQL Import

```
sqoop import
--connect jdbc:mysql://localhost/acmedb
--table ORDERS
--username test
--password
```

Algorithm 2. Hive Import

```
sqoop import
--connect jdbc:mysql://localhost/acmedb
--table ORDERS
--username test
--password
--hive-import
```

Algorithm 3. HBase Import

```
sqoop import
--connect jdbc:mysql://localhost/acmedb
--table ORDERS
--username test
--password
--hbase-create-table
--hbase-table ORDERS
--column-family mysql
```

Algorithm 4. Export

```
sqoop export
--connect jdbc:mysql://localhost/acmedb
--table ORDERS
--username test
--password ****
--export-dir /user/asd
```

[1] https://git-wip-us.apache.org/repos/asf?p=sqoop.git;a=summary

each users' transaction records are stored in a Hadoop system and the users' attributes can be stored in an RDBMS database. If we need to perform analyses into each users' transactions and link it to the users' attributes to check for fradulent transactions, we need to get the RDBMS data into the Hadoop system. Sqoop can be used to do this, by automatically pushing new updates in the RDBMS database into Hadoop.

Exporting data from Hadoop

In the online search business, enormous amounts of data gets generated from user search queries, advertisements, clicks and views. In order to perform quick analyzes on the data like how many users clicked on a particular advertisement or how to price a particular slot, we need to aggregate the data from Hadoop and export it to RDBMS systems.

CONCLUSION

Sqoop is an opensource software that helps to move efficiently data between RDBMS and Hadoop systems. In traditional systems we can get data dumps from ETL systems and push them to Hadoop using shell scripts. This process is time consuming to operate and develop. There are a lot of manual configurations involved like the file path and names. Sqoop can be used to avoid this hassles and automate and schedule these transfers easily.

FURTHER EDUCATION

Further learning about Sqoop is encouraged and informative materials can be found at the Apache Sqoop homepage[4].

ACKNOWLEDGEMENTS

The author thanks Professor Gregor Von Lazewski for providing us with the guidance and topics for the paper. The author also thanks the AIs of Big Data Class for providing the technical support.

REFERENCES

[1] A. Prabhakar, "Apache sqoop -overview," Web Page, October 2011. [Online]. Available: http://blog.cloudera.com/blog/2011/10/apache-sqoop-overview/

[2] V. Basavaraj, "Sqoop presentations and blogs," Web Page, December 2014. [Online]. Available: https://cwiki.apache.org/confluence/display/SQOOP/Sqoop+Presentations+and+Blogs

[3] The Apache Software Foundation, "Apache sqoop," Web Page, November 2016. [Online]. Available: http://sqoop.apache.org/

[4] Atlassian Confluence, "Pages - apache sqoop," Web Page. [Online]. Available: https://cwiki.apache.org/confluence/collector/pages.action?key=SQOOP

AUTHOR BIOGRAPHIES

Harshit Krishnakumar is pursuing his MSc in Data Science from Indiana University Bloomington

Apache Spark's Machine Learning Library (MLlib)

ANVESH NAYAN LINGAMPALLI[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: anveling@umail.iu.edu*

February 27, 2017

This paper provides a summary about Apache Sparks Machine learning library (MLlib) and its functionality.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: MLlib, Apache Spark, machine learning

https://github.com/Anveling/sp17-i524/paper1/S17-IR-2016/report.pdf

1. INTRODUCTION

Apache Spark is a open source processing engine with consists of elegant APIs, for performing efficient data analytics. It provides a framework to process big data which are diverse in nature. Spark has many advantages when compared to other technologies such as Hadoop and Storm. Hadoop is also a big data processing technology which is proved to be a solution for processing large data sets. But it is not efficient in cases involving machine learning or streaming data. In these cases, Hadoop requires other tools such as Mahout or Storm to process the data. This is the most important advantage that the Apache Spark has on Hadoop. Spark is faster in run times than Hadoop MapReduce.[1]

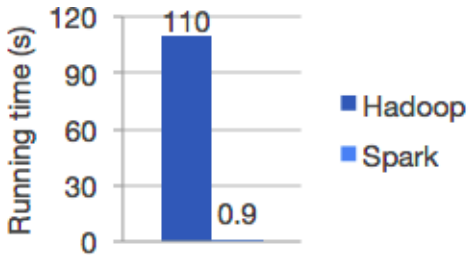


Fig. 1. Hadoop vs Spark

Spark in addition to Map and Reduce functions, also supports SQL queries and machine learning. It has many libraries in Big Data analytics and Machine Learning domains. MLlib is one of the top level libraries that Spark offers. MLlib (Machine Learning Library) is Apache Spark's scalable machine learning library with APIs in Java, Python, R and Scala. It has the algorithms and tools for performing various tasks on the data such as, clustering, classification, regression and dimensionality reduction. The main goal of this library is to make machine learning easy.

2. HISTORY AND DEVELOPMENT

Development of MLlib began in 2012 as a part of MLBase project (Kraska et al., 2013). It is an open source since September 2013. It has since then, been integrated into the Spark as an in-built package. The original version of MLlib was developed in UC Berkeley and provided a limited set of machine learning methods. Since it is an open source community, MLlib developed and now has additional functionality.

3. COMPONENTS OF MLLIB

MLib provides various linear models, Naive bayes and decision trees for classification and regression problems. With the help of these models, problems such as alternating least squares(ALS), k-means problem, PCA (principal component analysis) for clustering have been successfully implemented. Text mining, predictive analysis of data are certain areas where MLlib is being used as an efficient tool.

MLlib has a package named spark.ml, which provides APIs for the functionality of the pipelines. This package enables users to swap the existing algorithms with their own algorithms.[2]

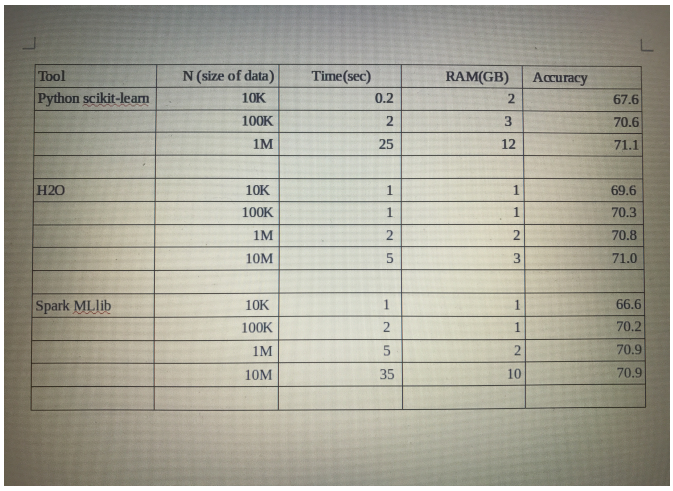
MLlib supports various methods for binary classification, multiclass classification, and regression analysis. Each type of problem has its own supported algorithms. Binary Classification has Linear SVMs, Logistic regression, descision trees and naive Bayes. Multiclass Classification also has decision trees and naive Bayes as its supported algorithms. Regression has linear least squares, Lasso and decision trees.

4. PERFORMANCE ANALYSIS BETWEEN MLLIB AND ITS ALTERNATIVES

Hadoop Mahout is one of the alternative choice for a machine learning library. Mahout uses Hadoop as underlying framework whereas in the case of MLlib, it is Spark. In terms of features, support and performance MLlib performs better. In 2014, Ma-

hout announced it would not accept Hadoop MapReduce and completely switched to Spark.

H2O, xgboost, python scikit-learn are few other alternatives to MLlib. Scalability, speed and performance are measured for these tools and are shown in the table below.



Tool	N (size of data)	Time(sec)	RAM(GB)	Accuracy
Python scikit-learn	10K	0.2	2	67.6
	100K	2	3	70.6
	1M	25	12	71.1
H2O	10K	1	1	69.6
	100K	1	1	70.3
	1M	2	2	70.8
	10M	5	3	71.0
Spark MLlib	10K	1	1	66.6
	100K	2	1	70.2
	1M	5	2	70.9
	10M	35	10	70.9

Fig. 2. Analysis of performance

For each tool and each size N, observations of the training tie, memory usage, and accuracy are presented. These tests have been carried out on a Amazon EC2 instance (32 cores, 60GB RAM).[3]

The graph for the results is shown below. H2O is memory efficient and faster than MLlib. But MLlib is the better choice of the two as it has variety of functionalities.

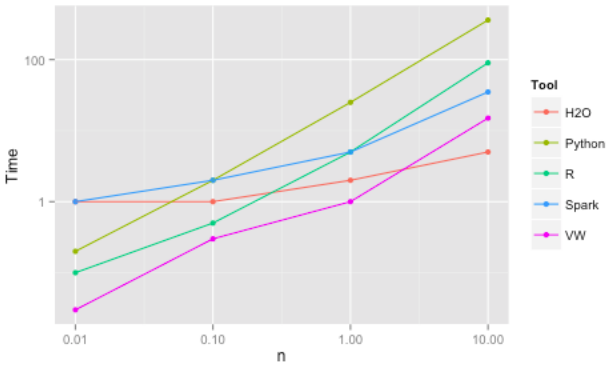


Fig. 3. Graph analysis

5. USE CASES

Apache Spark Machine Learning Library is used in wide range of applications in research and industry. Here two such applications are described briefly.

5.1. Movie Recommendation with MLlib

In this mini course project MLlib library is used to make personalized movie recommendations.[4]

5.2. Predict Telco Churn with Apache Spark MLlib

Churn prediction, is one of the most common applications of machine learning in the telecommunications industry, as well as many other subscriptions-based industries.MLlib is used here to fit a machine-learning model that can predict which customers of a telecommunications company are likely to stop using their service.[5]

6. USEFUL RESOURCES

[6] also has some good step by step tutorials on how to use Machine learning library to work on big data anlytics involving machine learning learning studio.

7. CONCLUSION

In conclusion, MLlib is one of the best libraries to perform machine learning as a part of big data analysis. It is still in active development phase, and there have been many improvements over the previous versions over time. MLlib provides developers with a wide range of tools to make machine learning easy and scalable.

8. ACKNOWLEDGEMENTS

This work was done as part of the course "I524: Big Data and Open Source Software Projects" at Indiana University during Spring 2017.

REFERENCES

[1] "Mllib," webpage. [Online]. Available: http://spark.apache.org/mllib/

[2] X. Meng, J. K. Bradley, B. Yavuz, E. R. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. B. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar, "Mllib: Machine learning in apache spark," *CoRR*, vol. abs/1505.06807, 2015. [Online]. Available: http://arxiv.org/abs/1505.06807

[3] "Analysis of various machine learning packages," webpage. [Online]. Available: https://github.com/szilard/benchm-ml

[4] "Movie-recommender-using-mllib," webpage. [Online]. Available: http://ampcamp.berkeley.edu/big-data-mini-course/movie-recommendation-with-mllib.html

[5] "Prediction-of-telco-churn," webpage. [Online]. Available: https://blog.cloudera.com/blog/2016/02/how-to-predict-telco-churn-with-apache-spark-mllib/

[6] "Guide for mllib," webpage. [Online]. Available: https://spark.apache.org/docs/latest/ml-guide.html

An Overview of OpenNebula Project and its Applications

VEERA MARNI[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: vmarni@umail.iu.edu

February 27, 2017

This paper provides insight into how OpenNebula can provide the right cloud services for unique needs of each organization by managing data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: opennebula, open cloud, cloud computing

https://github.com/narayana1043/sp17-i524/blob/master/paper1/S17-IR-2017/report.pdf

INTRODUCTION

OpenNebula[1]is a open cloud platform using which each organization setup the right cloud for its organizational needs. This quite natural like it happened with the databases and webservers. As one cloud could not solve all the needs of various work environments OpenNebula helps in setting up and deploying cloud platform based on needs[2]. It is a simple feature rich and flexible solution for the management of virtualized data centers.

It's interoperability makes cloud progress in developments of new methodologies to take advantage of the IT assets in turn saving investments and completely avoiding lock-in costs for the project. It's platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. It is a trunkey enterprise-ready solution that includes all features need to provide private and public cloud services. It is a open cloud architecture which is simple, open, reliable and flexible.

The paper is organized as follows. First OpenNebula's Architecture is discussed followed by Integration, API's and Language Binding. It is then followed by OpenNebula ecosystem and licensing. Finally, use cases are discussed with respect to its intended users which followed by key features and educational resources. It is then concluded by high lighting the major drawbacks that need attention in moving forward.

ARCHITECTURE OF OPENNEBULA

OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures. The OpenNebula platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. It provides features at two main layers of Data Center Virtualization and Cloud Infrastructure.

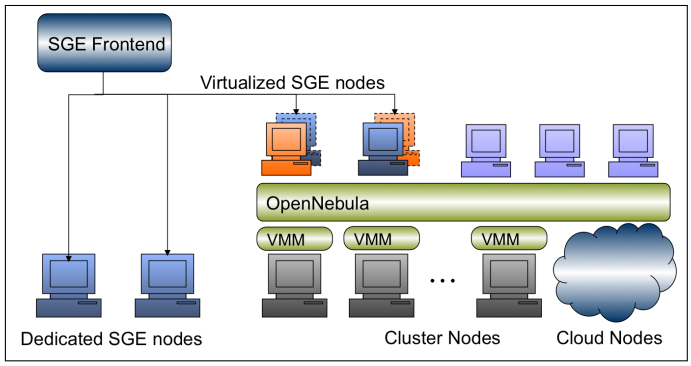


Fig. 1. The OpenNebula Engine for Data Center Virtualization and Cloud Solutions.

Data Center Virtualization Management

Many users use OpenNebula to manage data center virtualization[3], consolidate servers, and integrate existing IT assets for computing, storage, and networking. In this deployment model, OpenNebula directly integrates with hyper-visors and has complete control over virtual and physical resources, providing advanced features foe capacity management, resource optimization, high availability and business continuity.

Cloud Management

OpenNebula also provides a multi-tenant, cloud-like provisioning layer on top of an existing infrastructure management solution (like VMware vCenter). It helps in provisioning, elasticity

and multi-tenancy cloud features like virtual data centers provisioning, data center federation or hybrid cloud computing to connect in-house infrastructure is managed by already familiar tools for infrastructure management and operation.

INTEGRATION, API'S AND LANGUAGE BINDING

System Interfaces

OpenNebula has been designed to be easily adapted to any infrastructure and easily be extended with new components. The result is a modular system that can implement a variety of cloud architectures and can interface with multiple data center services. It interfaces can be classified into e categories

1. end-user cloud interfaces
2. system interfaces

Cloud interfaces are primarily used to develop tools to the end-user, and they provide a high level abstraction of the functionality provided by the cloud. They are designed to manage virtual machines, networks and images through a simple and easy-to-use REST API. The clod interfaces hide most of the complexity of a cloud and specially suited for end-users. OpenNebula features a EC2 interface, implementing the functionality offered by the Amazon's EC2 API, mainly those related to virtual machine management. In this way, you can use any EC2 Query tools to access your OpenNebula Cloud.

System interfaces[4] expose the full functionality of OpenNebula and are mainly used to adapt and tune the behavior of OpenNebula to the target infrastructure. The XML-RPC interface is the primary interface for OpenNebula, exposing all the functionality to interface the OpenNebula daemon. The OpenNebula cloud API provides a simplified and convenient way to interface with the OpenNebula core XMLRPC API. OpenNebula also includes 2 language bindings for OCA: Ruby and JAVA. The OpenNebula OneFlow API is a RESTful service to create, control and monitor service to create, control and monitor services composed of interconnected VMs with deployment dependencies between them.

Infrastructure Integration

The interactions between OpenNebula and the Cloud infrastructure[5] are performed by specific drivers. Each one addresses a particular area:

Storage The OpenNebula core issue abstracts storage operations that are implemented by specific programs that can be replaced or modified to interface special storage back-ends and file systems.

Virtualization The interaction with the hypervisors are also implemented with custom programs to boot, stop or migrate a virtual machine. This allows you to specialize each VM operation so to perform custom operations.

Monitoring Monitoring information is also gathered by external probes. You can add additional probes to include custom monitoring metrics that can later be used to allocate virtual machines or for accounting purposes.

Authorization OpenNebula can be also configured to use an external program to authorize and authenticate user requests. In this way, you can implement any access policy to Cloud resources.

Networking The hypervisor is also prepared with the network configuration for each Virtual Machine.

ECOSYSTEM

The OpenNebula Ecosystem[6] is formed by external tools and extensions that complement the functionality provided by the OpenNebula Cloud Management Platform. In addition, the ecosystems built around the cloud interfaces implemented by OpenNebula, Amazon AWS and OGC OCCI, can also be leveraged.

USE CASES OF OPENNEBULA

For the Infrastructure Manager

OpenNebula responds quickly to infrastructure needs for services with dynamic resizing of the physical infrastructure by adding new hosts and dynamic cluster partitioning to meet capacity requirement of services. It has centralized management of all the virtual and physical distributed infrastructure. It can improve the utilization of existing resources in the data center and infrastructure sharing between different departments managing their own production clusters, so removing application silos. It improves operational saving with server consolidation to a reduced number of physical systems, so reducing space, administration efforts, power and cooling requirements. It has also reduced infrastructure expenses with the combination of locate and remote cloud resources so eliminating the over purchase of systems to meet peak demands.

For the Infrastructure user

It is built for fast delivery and scalability of services to meet dynamic demands of service end-users. It supports heterogeneous execution environment with multiple, even conflicting, software requirements on the same shared infrastructure. It also provides full control of the life cycle of virtualized services management.

For System Integrators

It fits into any existing data center due to its open, flexible and extensible interfaces, architecture and components. It can build any type of cloud deployment. It is open sourced under Apache license and has seamless integration with any product and service in the virtualization/cloud ecosystem and management tool in the data center.

KEY FEATURES AND COMPONENTS OPENNEBULA

Features

There are several key features of OpenNebula[7] for the comprehensive management of virtualized data centers to enable private, public and hybrid clouds. Some of these key features are interfaces for cloud consumers, service management and catalog, interfaces for administrators and advanced users, appliance market place, chargeback, capacity and performance management, high availability and business continuity, virtual infrastructure management and orchestration, external cloud connector, platform independent, security, integration with third-party tools, fully open-sourced, automatic upgrade process, quality assurance and community support.

Components

OpenNebula has several advanced components[8] that can be easily integrated and deployed. Some of the important components that are extensively used are listed below:

1. Multi-VM Applications and Auto-scaling

2. Host and VM Availability
3. Data Center Federation
4. Cloud Bursting
5. Application Insight
6. Public Cloud
7. MarketPlace

EDUCATIONAL MATERIAL

A great place to start is by reading the OpenNebula documentation[9]. Future Systems has a tutorial on there webpage[10] to get started with OpenNebula. Handbook of Research on High Performance and Cloud Computing in Scientific Research and Education[11] by Marijana Despotovic-Zrakic is great book to get started from the basics of cloud computing and understanding the working of OpenNebula.

CONCULSION

OpenNebula is made impact in the way cloud services are offered before it was introduced by reducing costs of infrastructure, increasing the utilization of available resources, increase the computing power by integrating different machines and simplifying the development and deployment in industry. However, there are some disadvantages that needs attention. Some of them are Greater dependency on service providers, Risk of being locked into proprietary or vendor-recommended systems, Potential privacy and security risks of putting valuable data on someone else's system. Another important problem what happens if the supplier suddenly stops services. Even with this disadvantages the technology is still used greatly in various industries and many more are looking forward to move into cloud.

REFERENCES

[1] "Wikipedia-opennebula," webpage. [Online]. Available: https://en.wikipedia.org/wiki/OpenNebula
[2] "About-techonology," webpage. [Online]. Available: https://opennebula.org/about/technology/
[3] "Data center virtualization and solutions," webpage. [Online]. Available: https://opennebula.org/the-opennebula-engine-for-data-center-virtualization-and-cloud-solutions/
[4] "Opennebula system interfaces," webpage. [Online]. Available: http://docs.opennebula.org/5.2/integration/system_interfaces/index.html
[5] "Opennebula infrastructure integration," webpage. [Online]. Available: http://docs.opennebula.org/5.2/integration/infrastructure_integration/index.html
[6] "Opennebula ecosystem," webpage. [Online]. Available: https://opennebula.org/community/ecosystem/
[7] "Key features of opennebula," webpage. [Online]. Available: https://opennebula.org/about/key-features/
[8] "Opennebula adavanced components," webpage. [Online]. Available: http://docs.opennebula.org/5.2/
[9] "Opennebula," Web Page. [Online]. Available: https://opennebula.org/documentation/
[10] FutureSystems, "Opennebula online tutorial," Web Page. [Online]. Available: https://portal.futuresystems.org/tutorials/opennebula
[11] M. Despotovic-Zrakic, V. Milutinovic, A. Belic, and I. Global, *Handbook of Research on High Performance and Cloud Computing in Scientific Research and Education*, ser. Advances in Systems Analysis, Software Engineering, and High Performance Computing. Hershey, Pennsylvania (701 E. Chocolate Avenue, Hershey, Pa., 17033, USA), 2014.

Analysis of Pentaho

BHAVESH REDDY MERUGUREDDY[1,*]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: bmerugur@umail.iu.edu*

Paper 1, February 27, 2017

Pentaho is a leading business analytics and data integration tool that provides a qualified open source-based platform to assist a variety of big data deployments. It enables different organizations to utilize their data which helps them in delivering their services efficiently with minimum risk.Pentaho is often considered as an ideal application which can be used by businesses that desire to get the most out of their data and can also be used for embedded analytics.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Pentaho, Data Integration, Big Data, Community, ETL, MapReduce, SQL, Hadoop, OLAP

https://github.com/bhavesh37/sp17-i524/blob/master/paper1/S17-IR-2018/report.pdf

INTRODUCTION

Pentaho can be viewed as a business intelligence suite that provides data mining, reporting, dashboarding and data integration capabilities. Generally, organizations tend to obtain meaningful relationships and useful information from the data present with them. Pentaho addresses the obstacles that obstruct them from doing so [1]. The platform includes a wide range of tools that analyze, explore, visualize and predict data easily which simplifies blending the data. The sole objective of pentaho is to translate data into value. Being an open and extensible source, pentaho provides big data tools to extract, prepare and blend any data. Along with this, the visualizations and analytics will help in changing the path that the organizations follow to run their business. From spark and hadoop to noSQL, pentaho transforms big data into big insights.

PENTAHO COMMUNITY

Pentaho provides two different editions, community edition and enterprise edition. As the name suggests, the enterprise edition comes with more packages to provide addition support. The community edition enables the developers or users to create complex solutions for the problems pertaining to their business [2]. The pentaho community has a group of intellectual people and helps the users in becoming a part of them and benefit from the open source contributions. These open source projects are helpful in delivering reliable and faster products which are timely tested by the community. The community includes all users like developers, testers and managers. Generally, the community edition platform enables the developers to sketch their design and develop a rough version of their product after which they can upgrade to enterprise edition for final production.

Pentaho provides an interactive console to its users. With a few clicks of the mouse, users are allowed to interact with new data models and data. The platform hides the database connections and underlying application server and provides access to various data sources [3]. It provides metadata management capabilities and a dashboard to allow the administrators set security levels, monitor servers and set user access. There are many server plugins and desktop applications provided by pentaho.

Server applications

Business Intelligence platform is a basic service that provides reports, displays dashboards, reports business rules and performs OLAP analysis. The latest version comes with RESTful services and re-written scheduler along with a migration system. It generally runs in Apache java application server and can be embedded in any other java application server [1]. Pentaho analysis service is another server application that is written in java which primarily focuses on online analytical processing. It aggregates data into a memory cache by performing read operation from data sources like SQL. It comes with the pentaho platform in both the editions. These are some of the server applications provided by pentaho.

Desktop applications

Pentaho data mining is a desktop application that searches for patterns in data by performing knowledge analysis. All the techniques of data mining such as classification, clustering, regression and visualization are employed by this application along with some machine learning algorithms. This helps the users in predicting the trends in future. Pentaho metadata editor is an application that is used as an abstraction layer from the underlying data sources and helps the users in creating effective business models which can be used by other applications in

creating reports for the analytics. There are many more useful desktop applications.

Server plugins

Some of the important server plugins are community data access and data browser. Community data access is a pentaho server plugin that provides a common layer on the business analytics server for an easy data access. It runs the server by providing a REST interface and gets back the results in various forms such as xml, csv or json. Community data browser is a plugin that helps R in performing analytics on the data. It does the job of supplying queries to R by using online analytical processing browser.

DATA INTEGRATION

Extract, transform and load (ETL) are the basic operations that act as a tool for transforming data from one database and placing in other database. These processes can be carried out in pentaho with the help of a component called pentaho data integration, which is also referred as kettle [4]. The most useful functions of pentaho data integration include massive load of data into databases, data cleansing, migrating data between applications and integrating several applications. It is metadata oriented and can be used as a standalone application. The ability of transforming data is so high that the data can be manipulated with a very few limitations. Various input and output formats such as datasheets and text files are supported by pentaho data integration.

The transformation process undergoes three steps, input step, transformation step and output step. In the input step, data is ingested [5]. The data is then processed within pentaho data integration and the transformed data is given out in the output step. All these steps are carried out in parallel. The throughput of transformation process is restricted to speed of the step which is slowest. The slowest step is often referred as bottleneck. To improve the performance of transformation process, two steps are run in a loop which are, identification of the bottleneck and continued improvement of bottleneck until it is no longer a bottleneck.

Pentaho data integration has a set of components that contribute to its functionalities. They are spoon, kitchen, pan and carte [6]. Spoon can be considered as a desktop application that creates simple and even complex extract, transform and load (ETL) jobs without making the users write or read code. Spoon is the application that is used for transformations and jobs with the help of editor. So, it is the one that is used in most of the cases such as editing, debugging or running a transformation or a job. As the transformations are created in spoon, they can be executed with the help of a standalone command line process called pan. It is an engine that reads data, manipulates it and loads into various data sources. Kitchen is another standalone command line process that for executing jobs. It schedules different jobs to run at regular intervals. Carte provides remote execution capabilities and a medium for setting up a remote ETL server.

ARCHITECTURE

Pentaho architecture can be considered as a set of four components which are presentation layer, business intelligence platform, data and application integration and third party applications. Data can be provided to the presentation layer by reporting, analysis or process management. This data can then be accessed through a web service, portal or a browser [7]. The security and repository issues are dealt by the business intelligence platform. Data integration and third party applications are respectively, the integration layer and applications with database from various sources.

The architecture also includes a set of predefined layers such as data layer, server layer and client layer. Data layer allows an application to connect to a data source. Server layer serves as a middle layer and several applications run on the server. Dashboards are provided to the end users by deploying them on the server along with the required reports. As mentioned above, a user console is provided that is used for security and configuration purposes. Client layer is of two forms, thin client and thick client. Thin client generally runs on a server. Analyzer and dashboard editor can be considered as the examples. Report designer and data integration come under thick client which act as a standalone.

BIG DATA USE CASES

Big data refers to humongous volumes of data being taken from multiple data sources and put into data stores. A use case can be defined as a technology solution for business specific challenges. Big data use cases help in understanding the problems that big data addresses.

Cyber security analysis helps the end users such as data scientists and security analysts in quickly detecting the threats. Cyber security analytics allows the users to utilize most of the staff resources via automation [8]. It empowers the data scientists with predictive analytics with the help of machine learning tools. It also provides the automation of blending and reporting on a variety of data. Pentaho platform can be utilized for data processing, data ingestion and delivery of threat calls with minimal costs and complexity.

Pentaho optimizes data warehouse and speeds up the development and deployment processes. It employs a simplified process for offloading to Hadoop. The offloaded data is usually less frequent data. Hand coding in MapReduce jobs and SQL can be avoided by the usage of visual integration tools. It provides access to data sources ranging from relational to operational to NoSQL technologies. Pentaho MapReduce helps in achieving high performance in a cluster environment. It provides a graphical and intuitive big data integration.

Another use case identified by pentaho is the streamlined data refinery. Pentaho data integration processes and refines different data sets by using Hadoop as its data processing platform. It provides modelled, delivered and published data sets to the users for visual analytics just by a mouse click. It can be seen as an integration process that blends huge volumes of highly diversified data. It also supplies tools for in-cluster simplified data processing and is regarded as a highly practical approach.

Pentaho's big data support extends the 360-degree view to internal and external customer related data. Customer service teams are provided with time-sensitive and blended streams of data. This helps in making profitable decisions. The presence of an adaptive big data layer relieves several organizations from evolving technologies. Customers are given access to customizable, intuitive and interactive dashboards. Data scientists are provided with predictive analytics and data mining tools.

Monetizing the data is the final use case addressed by pentaho. It allows the users to capitalize on big data with the help of powerful data processing and embeddable data analytics [9]. Pentaho's big data analytics platform empowers easy big data

ingestion and transformation as it works as a no-code data integration environment. It is a flexible platform that supports security and deployments specific to customers.

COMPARISON

Pentaho products compete with some big names in current field such as SAP, IBM and oracle. Pentaho provides open source solutions and is considered to be much cheaper than the proprietary equivalents. Jaspersoft is an established open source rival of pentaho. Though both pentaho and jaspersoft offer similar features with similar costs, pentaho has got wider online presence and more followers in social media [10].

CONCLUSION

Pentaho is an open source based platform for diverse big data deployments. It empowers analytics in any environment by delivering governed data. It has unified data integration and analytics components which are comprehensive and embeddable. The primary aim of pentaho is to enable organizations to find new revenue streams with extraordinary service at minimum risk. It helps them in harnessing the value from their data in order to make their operations efficient and consistent.

REFERENCES

[1] "Pentaho," webpage. [Online]. Available: https://en.wikipedia.org/wiki/Pentaho
[2] "Community wiki home," Webpage. [Online]. Available: http://wiki.pentaho.com/display/COM/Community+Wiki+Home
[3] "Pentaho bi suite enterprise edition," Webpage, 2006. [Online]. Available: http://searchdatamanagement.techtarget.com/review/Pentaho-BI-Suite-Enterprise-Edition
[4] M. C. Roldán, "Pentaho data integration (kettle) tutorial," Webpage, 2008. [Online]. Available: http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+(Kettle)+Tutorial
[5] R. Haces, "Pentaho data integration performance tuning," Webpage. [Online]. Available: https://support.pentaho.com/hc/en-us/articles/205715046-Best-Practice-Pentaho-Data-Integration-%20Performance-Tuning-
[6] "Pentaho data integration architecture," Webpage. [Online]. Available: https://help.pentaho.com/Documentation/5.3/0L0/0Y0/010
[7] "Understanding pentaho architecture," Webpage. [Online]. Available: https://www.edureka.co/blog/understanding-pentaho-architecture/
[8] "What is big data?" Webpage. [Online]. Available: http://www.pentaho.com/what-is-big-data#tab-3
[9] "Monetize my data," Webpage. [Online]. Available: http://www.pentaho.com/Monetize-My-Data
[10] "Compare pentaho vs. jaspersoft," Webpage. [Online]. Available: https://comparisons.financesonline.com/jaspersoft-vs-pentaho

Twister: A new approach to MapReduce Programming

VASANTH METHKUPALLI[1,*]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: mvasanthiiit@gmail.com

Paper-1, February 27, 2017

MapReduce is a method to process vast sums of data in parallel without requiring the developer to write any other code other than the mapper and reduce functions. Starting with Google in 2004 there has been a lot of research going on in this particular field since the rate at which data is increasing is exponential. The need to store the data and analyze has become of paramount importance in the current situation. This has lead the researcher's to look for various parallel processing programming models to saturate the needs. Of these MPI, MapReduce are some of the examples which has produced good results to the scientific community. Here in this paper we examine an implementation of MapReduce programming model, Twister, which exhibits some improvements over the current model.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: MapReduce, Twister, Iterative, reduction, combine

https://github.com/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2019/report.pdf

INTRODUCTION

Emergence of massive data sets in many areas and settings have presented many challenges and opportunities in data storage and analysis[1][2]. Traditional analytic tools many a times cannot live up to the needs and demands at the ongoing rate at which data is produced to store and analyze it[3]. However, domain knowledge and research have given rise to new tools and technologies which has made many of these tasks easier[4]. Google's MapReduce falls in to one of these categories. The MapReduce programming framework uses two tasks common in functional programming: Map and Reduce, Map() procedure (method) that performs filtering and sorting, Reduce() method that performs a summary operation(merging the results)[5][6]. MapReduce is a new parallel processing framework and Hadoop is its open-source implementation on a single computing node or on clusters. Compared with existing parallel processing paradigms (e.g. grid computing and graphical processing unit (GPU)), MapReduce and Hadoop have two advantages:1) Fault-tolerant storage resulting in reliable data processing by replicating the computing tasks, and cloning the data chunks on different computing nodes across the computing cluster. 2) High-throughput data processing via a batch processing framework and the Hadoop distributed file system (HDFS)[7][8][3].

Data are stored in the HDFS and made available to the slave nodes for computation. A MapReduce program consists of the "MapReduce System" (also called "infrastructure" or "framework") orchestrates the processing by marshalling the distributed servers, running the various tasks in parallel, managing all communications and data transfers between the various parts of the system, and providing for redundancy and fault tolerance[6]. MapReduce programming model has simplified the implementations of many data parallel applications[3]. The simplicity of the programming model and the quality of services provided by many implementations of MapReduce attract a lot of enthusiasm among parallel computing communities[8][9]. It has been identified that MapReduce can be extended to many other applications by improving on the programming model and the architecture. In this regard, Twister attempts to extend the Google's MapReduce application to more class of applications by including more features[10][11].

TWISTER:

Twister was developed as a part of Ph.D. research and is an going research project by the SALSA team @IU[9][1]. Identifying many problems in MapReduce programming model, they have envisioned to develop a better version to apply it to various scientific applications[4]. A set of extensions to the programming model and improvements to its architecture that will expand the applicability of MapReduce to more classes of applications[7]. Twister is a lightweight MapReduce runtime that has been developed by incorporating these enhancements[8][3]. Twister provides the following features to support MapReduce computations. 1) Distinction on static and variable data 2) Configurable long running (cacheable) map/reduce tasks 3) Pub/sub messaging based communication/data transfers 4) Efficient support for Iterative MapReduce computations (extremely faster than Hadoop or Dryad/DryadLINQ) 5) Combine phase to collect all reduce outputs 6) Data access via local disks 7) Lightweight (5600 lines of

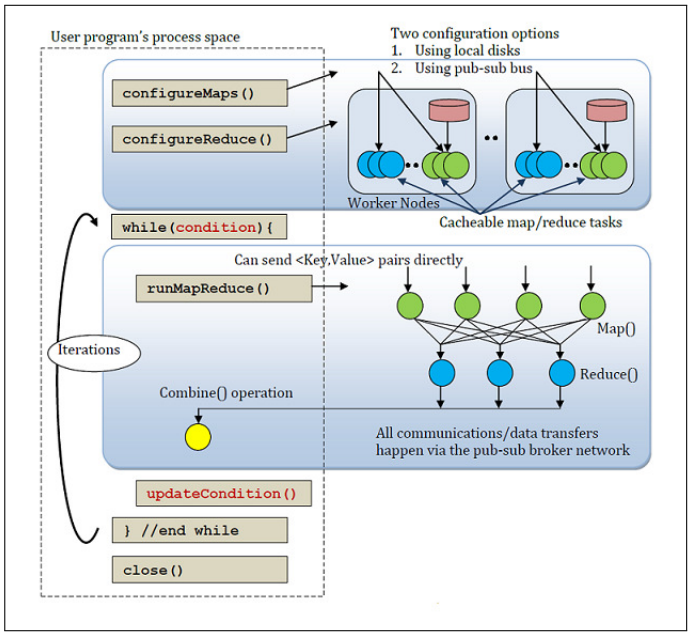


Fig. 1. Twister Programming model.

Java code) 8) Support for typical MapReduce computations 9) Tools to manage data[10][11].

TWISTER IMPROVEMENTS OVER MAPREDUCE PROGRAMMING MODEL:

Static vs. Dynamic Data

In all the iterative applications which were observed, they showed two types of data formats, one-static data, two-dynamic data[9]. Static data is data which is fixed throughout the computation whereas the variable data(Dynamic Data) is the computed results in each iteration and typically consumed in the next iteration in other applications, one of which example is an expectation minimization algorithms[11].

Canceable Mappers/Reducers

Although some of the typical MapReduce computations such as distributed sorting and information retrieval consume very large data sets, many iterative applications we encounter operate on moderately sized data sets which can fit into the distributed memory of the computation clusters[7]. This observation led us to explore the idea of using long running map/reduce tasks similar to the long running parallel processes in many MPI applications which last throughout the life of the computation. The long running (cacheable) map/reduce tasks allow map/reduce tasks to be configured with static data and use them without loading again and again in each iteration[3]. Current MapReduce implementations such as Hadoop and DryadLINQ do not support this behavior and hence they initiate new map/reduce tasks and load static data in each iteration incurring considerable performance overheads[10].

Supports "side-effect-free" Programming

Twister has long running map-reduce takes by which it does not encourage users to store state information in the map/reduce tasks. Thereby, achieving the side-effect-free nature of MapReduce. Caching the static data across map/reduce tasks helps in achieving better thoroughput[3]. This framework does not ensure the use of same set of map/reduce tasks throughout the life of an iterative computation[10].

Combine Step as Further Reduction

Twister also introduce an optional reduction phase named "combine", which is another reduction phase that can be used to combine the results of the reduce phase into a single value. The user program and the combine operation run on a single process space allowing its output directly accessible to the user program[9]. This enables the user to check conditions based on the output of the MapReduce computations[10].

Uses Pub/sub messaging

Twister uses pub/sub messaging for all the communication/data transfer requirements which eliminates the overhead in transferring data via file systems as in Hadoop orDryadLINQ[9]. The output <Key,Value> pairs produced during the map stage get transferred directly to the reduce stage and the output of the reduce stage get transferred directly to the combined stage via the pub-sub broker network[3]. Currently Twister uses publish-subscribe messaging capabilities of NaradaBrokering messaging infrastructure, but the framework is extensible to support any other publish-subscribe messaging infrastructure such as Active MQ [10].

Data Access via local disks

Two mechanisms about data access have been proposed in Twister; (i) Directly from the local disk of computer nodes, (ii) Directly from the pub-sub infrastructure[7]. For the simplicity of the implementation, Twister provided a file based data access mechanism for the map/reduce tasks. Unlike Hadoop, twister does not come with the built in file system. Instead it provides a tool to manage the data across these distributed disks. Apart from the above the use of streaming enables Twister to support features such as directly sending input <Key,Value> pairs for the map stage from the user program and configuring map/reduce stages using the data sent from the user program[6].

Fault Tolerance

Providing fault tolerance support for iterative computations with Twister is currently under development.

APPLICATION OF TWISTER TO PRESENT PROBLEMS:

K-Means Clustering

Kmeans clustering is a well-known clustering algorithm aiming to cluster a set of data points to a predefined number of clusters. In that each map function gets a portion of the data, and it needs to access this data split in each iteration. These data items do not change over the iterations, and it is loaded once for the entire set of iterations. The variable data is the current cluster centers calculated during the previous iteration and hence used as the input value for the map function[10][1].

All the map functions get this same input data (current cluster centers) at each iteration and computes a partial cluster centers by going through its data set. A reduce function computes the average of the partial cluster centers and produce the cluster centers for the next step. Main program, once it gets these new cluster centers, calculates the difference between the new cluster centers and the previous cluster centers and determine if it needs to execute another cycle of MapReduce computation[12][10].

Matrix Multiplication

Let A and B, produce matrix C, as the result of the multiplication process. Here they split the matrix B into a set of column blocks and the matrix A into a set of row blocks. In each iteration, all the map tasks process two inputs: (i) a column block of matrix B, and (ii) a row block of matrix A; collectively, they produce a row block of the resultant matrix C. The column block associated with a particular map task is fixed throughout the computation, while the row blocks are changed in each iteration. However, in Hadoop's programming model (a typical MapReduce model), there is no way to specify this behavior. Hence, it loads both the column block and the row block in each iteration of the computation. Twister supports the notion of long running map/reduce tasks where these tasks are allowed to retain static data in the memory across invocations, yielding better performance for "Iterative MapReduce" computations[1][10].

Page Rank

In Twister implementation of PageRank, we constructed web graphs with vertices where in-link degree of all pages comply with the power law distribution. These input data are partitioned into few parts and stored in the format of adjacency list. Each map function runs on one of the partitioned data, which are constant over the iterations. The variable data are the PageRank values calculated during previous iteration which in turns used as the input value for the next iteration. In each iteration, the MAP task updates the old PageRank values to new one by analyzing the local partial adjacency list file. The output of MAP task is partial of PageRank values. The reduce task receives all the partial output and produces the new PageRank values[10].

Graph Search

This algorithm tries to use Twister Framework to process breadth-first graph search problem in parallel. This algorithm is based on Cailin's Hadoop version of breadth-first graph search[10]. The basic idea of this algorithm is to exploring the nodes of the same level parallel, and then explore the next levels iteratively[10].

Word Count

In typical word count applications implemented using other MapReduce runtimes, the map task outputs (word,1) pairs for all the words it encounter. This approach is not optimized for performance rather simple to program. With small amount of more complexity we can simply convert this map task to produce a list of (word,count) pairs corresponding to the local data partion. This is the approach used in Twister word count application[10].

High Energy Physics(HEP) Data Analysis

The goal of the analysis is to execute a set of analysis functions on a collection of data files produced by high-energy physics experiments. After processing each data file, the analysis produces a histogram of identified features. These histograms are then combined to produce the final result of the overall analysis. This data analysis task is both data and compute intensive and fits very well for MapReduce computation model. First figure shows the program flow of this analysis once it is converted to a MapReduce implementation and the second figure compares the performance of Twister and Hadoop for this data analysis[12][10].

CONCLUSION

It has been observed that changes in the programming model and the way the nodes interact with each other(pub-sub messaging)[8], has produced very good results with Twister MapReduce. This has in turn lead it benificial to many applications of which many have been discussed in the present paper. So it can be safe to assume that Twister has a very high scope for commercial applications even thought it is limited to scientific applications as of now.

REFERENCES

[1] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," pp. 107–113, 2008.

[2] A. Mohebi, S. Aghabozorgi, T. Ying Wah, T. Herawan, and R. Yahyapour, "Iterative big data clustering algorithms: a review," *Software: Practice and Experience*, vol. 46, no. 1, pp. 107–129, 2016.

[3] K.-H. Lee, Y.-J. Lee, H. Choi, Y. D. Chung, and B. Moon, "Parallel data processing with mapreduce: a survey," *AcM sIGMoD Record*, vol. 40, no. 4, pp. 11–20, 2012.

[4] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets." *HotCloud*, vol. 10, no. 10-10, p. 95, 2010.

[5] "MapReduce," Feb. 2017, page Version ID: 764859583. [Online]. Available: https://en.wikipedia.org/w/index.php?title=MapReduce&oldid=764859583

[6] E. A. Mohammed, B. H. Far, and C. Naugler, "Applications of the mapreduce programming framework to clinical big data analysis: current landscape and future trends," p. 22, 2014.

[7] A. Elsayed, O. Ismail, and M. E. El-Sharkawi, "Mapreduce: State-of-the-art and research directions," *International Journal of Computer and Electrical Engineering*, vol. 6, no. 1, p. 34, 2014.

[8] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, S.-H. Bae, J. Qiu, and G. Fox, "Twister: a runtime for iterative mapreduce," in *Proceedings of the 19th ACM international symposium on high performance distributed computing*. ACM, 2010, pp. 810–818.

[9] K. Grolinger, M. Hayes, W. A. Higashino, A. L'Heureux, D. S. Allison, and M. A. Capretz, "Challenges for mapreduce in big data," in *Services (SERVICES), 2014 IEEE World Congress on*. IEEE, 2014, pp. 182–189.

[10] "Twister: Iterative MapReduce." [Online]. Available: http://www.iterativemapreduce.org/

[11] C. Doulkeridis and K. Nørvåg, "A survey of large-scale analytical query processing in mapreduce," *The VLDB Journal*, vol. 23, no. 3, pp. 355–380, 2014.

[12] J. Ekanayake, S. Pallickara, and G. Fox, "Mapreduce for data intensive scientific analyses," in *eScience, 2008. eScience'08. IEEE Fourth International Conference on*. IEEE, 2008, pp. 277–284.

Docker(Machine,Swarm)

SHREE GOVIND MISHRA[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors:shremish@indiana.edu

project-000, March 5, 2017

Docker is a container based technology which helps in the packaging and shipping of applications quickly and across networks. Docker is being well accepted and appreciated in the Agile Software Development workforce and among the DevOps because of their attributes of Continous Integeration and testing.To create a cluster of Docker Engines into a single docker virtual engine is used Docker-Swarm sed which is tested for more than 50,000 containers. © 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524, Big Data, Virtualisation, Docker, Docker Swarm, Docker Machine , Virtual Machines, VMs, Containers,LInux Containers, lightweight containers,LXC

https://github.com/Govind273/sp17-i524/blob/master/paper1/S17-IR-2021/report.pdf

INTRODUCTION

[1]Docker is an open-source container based technology. It is an extension of Linux Containers (LXC): which is a unique kind of lightweight, application-centric virtualization that drastically reduces overhead and makes it easier to deploy software on servers. A container allows a developer to package up an application and all its part includig it's code, stack it runs on, dependencies it is associated with,system tools and everything the application requirs to run within an isolated enviorment. This makes it easier for programmers and developers to run more apps on the same server and it even makes it easier to package and ship the apps very frequently. Docker has been able to popularize the container approach in part because it's improved the security and simplicity of container environments. Plus, interoperability is enhanced by its association with major companies – such as Google, Canonical, and Red Hat – on its open source element libcontainer.

HOW DOCKER DIFFERS FROM VMS

The virtualisation of application can be obtained with Hyervisors or Virtual Machine Manager(VMM) which makes it easy for applications to the run in isolation with one another while sharing the same underlying hardware articture. VM hypervisors, such as Hyper-V, KVM, and Xen, all are "based on emulating virtual hardware" which means they're fat in terms of system requirements.Instead of virtualizing hardware, containers rest on top of a single Linux instance.A full virtualized system gets its own set of resources allocated to it, and does a very minimal sharing of thoses resources. So, the developer gets more isolation but each Instance of the VM aslo has many "Junk VM files", which is not useful to the developers as it gets "heavy".Since they only require to keep the virtal machine image and it can be kept small.Thus, the smaller the image is the less we need to store and the less you need to send around the network which makes them fairly lightwight in comparison to the Virtal Machines. Thus, we can run even use 1000s of containers on a same host OS. [2]A full virtualized system usually takes time to start whereas the Docker contaier do take even less than seconds to up start and running with a lower overhead than the VMs. Containers are potentially much more efficient than VMs because they're able to share a single kernel and share application libraries. This can lead to substantially smaller RAM footprints even when compared to virtualisation systems that can make use of RAM overcommitment. Storage footprints can also be reduced where deployed containers share underlying image layers. IBM's Boden Russel has done

USES OF DOCKER

Docker for DevOps

Another major use of Docker is it's use in the DevOps community. Docker is not there to replace other configuration management tools and instead can be incoprporated with other configuration management tools like Chef, Puppet, Salt or ansible. The other major benifit of using Docker, Dockerfiles, the registry amd the whole Docker ecosystem is that the teams don't have to learn domain specific language as these are easier to learn than the domain specific Ecosystems. Though many a times Docker can be made to work with the other configuration management tools.[3]Docker can also the eliminate the need for a development team to have the same versions of everything installed on their local machine. The repeatable nature of Docker images makes it easier for them to standardize their production code and configurations. Their work has led to the creation of Helios,

an application that manages Docker deployments across multiple servers and that alerts them when a server isn't running the correct version of a container.

Docker as Virtualized Sandbox

[3] Docker allows sytems administrators and developers to build applications that can berun on any Linux distribution or hardware in a virtualized sandbox without making custom builds for all the different enviorments.Finally, it's easy to deploy Docker containers in a cloud scenario. You can easily integrate it with typical DevOps environments seamlessly (Ansible, Puppet, etc.) or use it as a standalone.

Docker for continuous integration

Docker can be used as git for continious integeration. Docker is similar as changes in the system can be tracked just like chaneges in the git. These collaboration features (docker push and docker pull) are one of the most disruptive parts of Docker. The fact that any Docker image can run on any machine running Docker is very much appreciated. But The Docker pull/push are the too new fot the first time developers and ops guys have ever been able to easily collaborate quickly on building infrastructure together.[1]The app guys can share app containers with ops guys and the ops guys can share MySQL and PosgreSQL and Redis servers with app guys.

DOCKER MACHINE

[4]Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands. You can use Machine to create Docker hosts on your local Mac or Windows box, on your company network, in your data center, or on cloud providers like AWS or Digital Ocean. Docker Machine is the only way to run Docker Engine on Mac or Windows previous to Dockerv1.12 and starting with the Dockerv1.12 we have Docker for Mac and Docker for Windows. Thus we can create a cluster of Docker hosts which is called a Swarm using Docker Machine.

DOCKER SWARM

[5]Docker Swarm provides native clustering capabilites to turn a group of Docker engines into a single, vrtual Docker Engine. These help you scale up the applications as if these all are running on a single,huge computer.It does so by providing a stabdard Docker API where if any tool which communicates with the Docker deamon can use Docker Swarm to transparently scale to multiple hosts: Dokku, Docker Compose, Krane, Flynn, Deis, DockerUI, Shipyard, Drone, Jenkins and, of course, the Docker client itself. Docker Networking, Volumes and plugins can also be used through their respective Docker commands via Swarm. Swarm has been tested and is production ready to scale up to one thousand (1,000) nodes and fifty thousand (50,000) containers with no performance degradation in spinning up incremental containers onto the node cluste Swarm also comes with a built-in scheduler, but you can easily plugin the Mesos or Kubernetes backend while still using the Docker client for a consistent developer experience. To find nodes in your cluster, Docker Swarm can use either a hosted discovery service, static file, etcd, consul and zookeeper depending on what is best suited for your environment.

ECOSYSTEM SUPPORT TO DOCKER

[3]Finally, it's easy to deploy Docker containers in a cloud scenario. You can easily integrate it with typical DevOps environments seamlessly (Ansible, Puppet, etc.) or use it as a standalone. The main reason it's so popular is simplification, says Ben Lloyd Pearson via opensource.com. You can do local development within a system that is identical to a live server; deploy various development environments from your host that each use their own software, OS, and settings; easily run tests on various servers; and create an identical set of configurations, so that collaborative work isn't ever hindered by parameters of the local host.Ecosystem support for Docker is improving with every passing day as it is gaing the popularity among the developers and the system operators.

Operating Systems suppport to Docker: Is compatible with virtually any distribution with a 2.6.32 + kernel RedHat Docker collaboration to work with across fedora and other(2.6.32)+.

It is compatible with Private PaaS(Platform as a Service) technolgies: OPen Shift, Solum(Rackspace and OPenstack).

Public PaaS technologies like Voxoz, Cocaine(Yandex),Biadu,etc.

Ecosystem Support for IaaS is present via Rackspace,Digital Ocean, AWS(Amazon Web Services), AMI,etc.

Orchestration tools Support and Integeration with: Chef,Puppt,Jerkins,Travis,Ansible..

In Open Stack Docker integeration into NOVA(compatibility with Glance,Horizon)are aslo present

SHORTFALLS OF DOCKER

Though dockers are leagcy virtualization techniques, they are not a go to solution for all kinds of virtualization and cannot be considered as a replacement of the VMs.

VMs are self constarined with as they have a unique operating system(OS), drivers and application components whereas the containers are dependent as containers under Docker cannot run on Windows server.

VMs provide high level of isolation as the system's underlying hardware resources are all virtualized and thus any bugs or viruses could not affetc the other VM in the virtualized enviorment.

The kinds of tools and technologies required to manage the containers are still lacking in the industry and thus only a few management tools from companies like Google and Docker like Kubernetes and SWarm respectively are present which is not a good situation for an open-source products

FUTURE WITH DOCKER

Docker Inc has set a clear path on the development of core capabilities (libcontainer), cross service management (libswarm) and messaging between containers (libchan). Docker's key open source project is libcontainers ,which is on it's way to becomiing the default standard for Linux-based technology. Libcontainers enables the containers to work with the Linux namespace, control groups, capabilities, AppArmor security profiles, network interfaces and firewalling rules in a consistent and predictable way . [3]Libcontainers is getting help from Google, RedHat, and Parallels to build the program asthey will work with docker as core maintainers of the code. libconatiner which is written natively in Google's Go, is also being ported into other languages. Microsoft may be porting it to ASP.NET. Parallels' libct, which includes libcontainer's functionality, has native C/C++

and Python bindings. Microsoft is even jumping on board by bringing Docker to their Azure platform, a development that could potentially make integration of Linux applications with Microsoft products easier than ever before.

CONCLUSION

[6]With more than 1200 Docker Contributors , 10,000 Dockerized Applications at index.docker.io 3 to 4 million Developers using Docker, 300 Million Downloads, 32,0000 Docker related Projects and 70 percent of enterprise using DockerIt is not an exaggeration that with Docker Ecosystem there is a greater potential for some advanced deployment tools that combine containers, configuration management, continuous integration, continuous delivery and service orchestration in the days ahead.

REFERENCES

[1] CenturyLink, "What is docker and when to use it," Web Page, Apr. 2014, accessed 2017-2-20. [Online]. Available: https://www.ctl.io/developers/blog/post/what-is-docker-and-when-to-use-it/

[2] Ken Cochrane, "How is docker different from a normal virtual machine?" Web page, apr 2013, accessed 2017-02-23. [Online]. Available: http://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-normal-virtual-machine

[3] opensource.com, "Who's using docker?" Web Page, Jul. 2014, accessed 2017-2-19. [Online]. Available: https://opensource.com/business/14/7/docker-through-hype

[4] Docker Inc, "What is docker machine," Web Page, accessed 2017-2-23. [Online]. Available: https://docs.docker.com/machine/overview/

[5] ——, "Docker swarm," Web Page, accessed 2017-2-21. [Online]. Available: https://www.docker.com/products/docker-swarm

[6] DATADOG, "8 surprising facts about real docker adoption," Web Page, accessed 2017-2-23. [Online]. Available: https://www.datadoghq/com/docker-adoption/

Triana

ABHISHEK NAIK[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: ahnaik@indiana.edu*

paper-1, February 26, 2017

Triana is an open source problem solving environment (PSE). It provides a powerful data analysis tool alongwith a intuitive visual interface. It is mainly used in the areas of signal, text and image processing. It comes with a set of inbuilt data analysis tools and also provides easy mechanisms for the integration of custom built ones.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Triana, Problem Solving Environment, I524

https://github.com/absnaik810/sp17-i524/blob/master/paper1/S17-IR-2022/report.pdf

INTRODUCTION

Triana is an open source Problem Solving Environment (PSE) that is supported with a powerful data analysis tool. It is predominantly used for image, text and signal processing; and comes with a host of different tools for analysis. Besides, it also provides features for easy integration of custom analysis tools. It thus focuses on supporting services from various environments, like peer-to-peer (P2P) and Grid.

BACKGROUND

[1] Triana has a graphical interactive environment that can be used by the users to create their applications and specify their behavior. In this, it is similar to tools like [2] draw.io and [3] IBM's Rational Rose that provide the user with a graphical user interface (GUI) to draw the UML diagrams. The users can model the work flow using the various units onto the workspace and then depict the relationship between them with some interconnection. The Triana PSE can also be extended to promote discovery of web services, their composition and decomposition.

FEATURES

The Triana framework that has been extended with the framework has the following set of features:

- Simple creation of the Web services: In case of other frameworks, the major challenge is that a composite service cannot be created easily. However, by allowing the discovery, composition, invoking and publishing of services in an atomic manner, Triana makes this easy.
- Execution of services in a distributed fashion: Services can be executed on a P2P or Grid middleware using Triana.
- 'What-if' analysis: 'What-if' analysis can be easily carried out using Triana, by monitoring the different resulting work flows.
- Annotation: Triana allows the work flows to be annotated for later use.

[4] Triana was originally developed for the GEO600 gravitation project. In this project, it is still being used for analysing the gravitational wave signals that emanate from the laser interferometer based out of Germany. [5] Triana uses a pluggable architecture, that is it checks the inputs coming to, and the outputs coming from various units in real time to perform data-type checking and conformation. It can also be used to monitor the work flow and run the executables in standalone mode. [6] It has a custom work flow language, although it can be integrated with others like the Business Process Execution Language (BPEL). This helps it in analysing large data sets and makes it particularly important in big data analysis.

INFRASTRUCTURE

[5] The Triana infrastructure consists of the Triana Controlling Service (TCS) that has a Triana Engine, implemented as a Triana service. The Triana engines are free to carry out the execution either locally or on distributed servers, as per the implementation policies in force. Communication in the later case can be carried out using pipelined work flow distributions.

The Triana Distributed implementation makes use of Triana Group units. They have the same features of the normal units (like input/output, etc.) and so they can be connected to the other Triana Units using the standard connection mechanism. The tools need to be group so that they can be distributed; and Triana has two distribution policies:

- Parallel: a 'farming-out' mechanism involving no communication between the hosts.

- Pipeline: this involves distributing the group 'vertically'.

Groups can in turn contain groups and each group can in turn, have its own distribution policy to be followed. The Triana distribution mechanisms are based upon the concept of GAT (Grid Application Toolkit). The GAT aims to shield the applications from the implementation details using a standard Application Programming Interface (GAT-API). It also provides a set of Grid services for carrying out tasks such as resource and information management.

Triana is further divided into a set of pluggable components that can actively interact with each other. The GUI connects to the engine either locally or via a network. Thus, the clients can log in into TCS and view the results onto their devices, although the unit in itself might be remote. They also have the option to log off and then log in again using a different device altogether. In this way, the Triana can be used as a monitoring system. Furthermore, there are three ways in which Triana can be used, namely:

- Using the GUI on top of an existing application. The various interfaces can be used to connect various applications with Triana. There are the work flow writers as well as the command writers.

- Using the remote control facility of an existing application. The Triana facility can be used in exactly the same way as highlighted above, but in addition to it, the facility of logging in and off can also be leveraged. Here, the scheduling would be implemented by a third party system.

- Forming various Triana units. This is the best and the most preferred way of usage since it becomes very easy to prototype and can be easily distributed using the Triana distribution mechanisms.

Triana has a pluggable architecture that can be extended using the Triana GUI as a front end to some standalone application. This can be carried out using the inbuilt task-graph or command writers. Custom writers can also be used. Similarly, it can also be extended by implementing the same set of interfaces that the Triana Control Service includes. Another way of extension is by implementing custom Triana units to achieve the desired functionality. This would require the building of a number of Triana components and their seamless integration with inbuilt components.

GAT provides simple communication mechanism with Triana. The calls are independent of the GAT middleware binding. This decoupling enables easy porting of Triana on different Grid middleware without any modification to the core Triana code. Furthermore, the Triana project is concurrently being developed with the GridLab project, that is developing the GridLab GAT. The Triana is precisely being used as a test application to develop requirements from the GridLab GAT. A prototype GAT API called the GAP Interface (Grid Application Prototype Interface) has been developed. This Interface is basically used for communicating with the remote servers, which was one of the main functionalities to be developed as a part of the GridLab GAT. JXTA currently has GAP interface bindings.

The GAP interface bindings implement the basic GAP functionality, which is locating and communicating the JXTA services. Such a service has two input nodes and may have multiple output nodes - either zero, one or more. The input and the output nodes are delegated as the JXTA pipes and a virtual communication channel is established between them. This virtual communication channel adapts a particular communication protocol depending upon the current operating environment.

PROJECTS

[7] Triana has been involved in a lot of projects (including some Big Data projects) like the [8] Data-Mining Grid, [9] Scalable Robust Self-organizing Sensor Network Project (SRSS) and [10] SHaring Interoperable Work flows for large-scale scientific simulations on Available DCIs (SHIWA), etc. As a part of the Data-Mining Grid, Triana was predominantly used to model and then manage the planning, development and the execution of data-mining work flows in grid computing environments. [9] The SRSS project is carrying out a research about the various communication protocols that can be leveraged in distributed and self-organizing networks; and they are using Triana's P2P binding to simulate various P2P networks. The European project, SHIWA, mainly focuses on the interoperability of myriad European Work flow Systems and has been intrinsically integrated with Triana to create [10] SHIWA bundles. Other than these, Triana is also being actively used in some other projects like EDGI, TRIACS, EDGES, WHIP (Work flows Hosted In Portals), DART (Distributed Audio Retrieval using Triana), GridOneD, GEO600, BiodiversityWorld, DIPSO, GEMMS, etc. many of which are related to Big Data.

Triana was released as an open source project on 30th May 2003. It included two GAT bindings - one that was implemented as the JXTA and the other as the Java socket based on the P2P mechanism (P2PS). How to run Triana, as JXTA or as a Java socket based on P2P mechanism is upto the user and the choice he makes at the system start.

CONCLUSION

Thus, to conclude, we presented Triana and the Triana PSE. We focused on the Triana infrastructure as well as its distributed implementation. We also presented some Big Data projects that used Triana for their development.

REFERENCES

[1] S. Majithia, I. Taylor, M. Shields, and I. Wang, "Triana as a graphical web services composition toolkit," in *Cardiff School of Computer Science, Cardiff University, Cardiff*, 2003. [Online]. Available: http://www.trianacode.org/papers/pdf/TrianaAHM03WS.pdf

[2] "Draw-io-website," Code Repository, JGraph, accessed: 2017-2-20. [Online]. Available: https://github.com/jgraph/draw.io

[3] "Ibm-rational-rose-website," Web Page, IBM, accessed: 2017-2-20. [Online]. Available: http://www-03.ibm.com/software/products/en/rosemod

[4] "Geo600-website," Web Page, Max Planck Gesellschaft, accessed: 2017-2-20. [Online]. Available: www.geo600.uni-hannover.de

[5] I. Taylor, M. Shields, I. Wang, and R. Philip, "Grid enabling applications using triana," in *Cardiff School of Computer Science, Cardiff University, Cardiff*, 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.4103&rep=rep1&type=pdf

[6] F. Pop, J. Kolodziej, and B. DiMartino, Eds., *Resource Management for Big Data Platforms*. Springer Nature, 2016, pp. 48–49.

[7] "Triana-project-list," Code Repository, Cardiff School of Computer Science & Informatics, accessed: 2017-2-20. [Online]. Available: https://github.com/CSCSI/Triana/wiki/Projects

[8] "Data-mining-grid," Web Page, University of Ulster, accessed: 2017-2-20. [Online]. Available: http://www.datamininggrid.org/

[9] "Srss," Web Page, US Naval Research Laboratory, accessed: 2017-2-20. [Online]. Available: http://srss.pf.itd.nrl.navy.mil
[10] "Shiwa-workflow," Web Page, Computer and Automation Research Institute, Hungarian Academy of Sciences - MTA SZTAKI, accessed: 2017-2-20. [Online]. Available: http://www.shiwa-workflow.eu/

LDAP

RONAK PAREKH[1] AND GREGOR VON LASZEWSKI[2]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: parekhr@indiana.edu

project-000, February 26, 2017

LDAP is a "lightweight" (smaller amount of code) version of Directory Access Protocol (DAP), which is part of X.500, a standard for directory services in a network. In a network, a directory tells you where in the network something is located. On TCP/IP networks (including the Internet), the domain name system (DNS) is the directory system used to relate the domain name to a specific network address (a unique location on the network).LDAP allows you to search for an individual without knowing where they're located (although additional information will help with the search).An LDAP directory can be distributed among many servers. Each server can have a replicated version of the total directory that is synchronized periodically.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: LDAP, directory, authentication, X.500, commands, design

https://github.com/ronak1182/sp17-i524/tree/master/paper1/S17-IR-2024/report.pdf

INTRODUCTION

LDAP (Lightweight Directory Access Protocol) is an open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol network.[1] LDAP is based on the client/server model of distributed computing. It has evolved as a lightweight protocol for accessing information in X.500 directory services.[2] LDAP is specified in a series of Internet Engineering Task Force(IETF) Standard Track publications called Request for Comments (RFC's), using the description language ASN.1. LDAP is based on a simpler subset of the standards contained within the X.500 standard.

BACKGROUND

LDAP was a result of the X.500 series of International Telecommunication Union (ITU) recommendations. X.500 is a set of recommendations about directories. [2] Because of this relationship, the structure of X.500 and LDAP is similar. LDAP directory implementations are often X.500 compliant and gateways between the two directories are plentiful. LDAP is defined by a set of published Internet standards, commonly referenced by their Request For Comment (RFC) number. The main reason for the emergence of LDAP can be attributed to the fact that X.500 was too tied to the OSI (Open Systems Interconnection) protocols, making it out of favour for the TCP-dominated world that was emerging. LDAP and Domain Naming System solved problems in a simpler way and thus, LDAP started becoming widely used.[2]

WHY LDAP OVER X.500?

The success of LDAP has been largely due to the characteristics that makes it simpler to implement and use as compared to X.500. LDAP runs over TCP/IP rather than the OSI protocol stack. The availability of TCP/IP is more and it consumes less resources. The functional model of LDAP is simpler i.e it removes less frequently used, duplicate and esoteric features, making it easier to implement and understand. LDAP uses strings to represent data rather than Abstract Syntax Notation One (ASN.1) which is considered much more complicated.[2]

LDAP ARCHITECTURE OVERVIEW

LDAP defines the content of messages exchanged between an LDAP client and an LDAP server. There are some operations such as search, modify, delete which are specified by the messsages requested by the client or responses from the server. The messages also describe the format of the data it carries [3]. The messages are carried over a TCP/IP protocol and thus, there are operations to establish and disconnect a session between the client and the server. The factors to be considered while designing the LDAP directory are the logical model which is defined by the messages and data types, the organization structure of the directory, the possible operations to be perfomed on the directory and the security of the information carried in the messages.[1]

The general interaction between an LDAP client and LDAP server is as follows:

The first step is known as Binding step. In this step, the client establishes a session with the LDAP server. The client specifies

the host name or the IP address and the TCP/IP port number where the LDAP server is listening. The client can provide a user name and a password to properly authenticate with the server. Data encryption can also be used while establishing a session to inmprove security. The client then performs operations on directory data. LDAP offers both read and update capabilities. Thus, the directory information can be managed as well as queried. Searching is a common LDAP operation where user can specify what part of the directory to search and what information to return. When the client is finished making requests, it closes the session with the server which is known as unbinding. The LDAP directory stores and organizes data structures known as entries. A directory entry usually descibes an object such as a person or a server and so on. Each entry has a distinguished name (DN) that uniquely identifies it. The DN consists of a sequence of parts called relative distinguished names (RDNs). The entries can be arranged into a hierarchical tree-like structure based on their distinguised names. The tree of directory entries is called the Directory Information Tree. [2]

Each entry contains one or more attributes that describe the entry. Each attribute has a type and a value. A directory entry describes some object which in general is called as a template. The following operations are defined by LDAP for accessing and modifying directory entries:

- Adding an entry
- Deleting an entry
- Modifying an entry
- Comparing an entry
- Moving an entry

A client starts an LDAP session by connecting to an LDAP server, called a Directory System Agen (DSA), by default on TCP and UDP port 389, or on port 636 for LDAPS. The client sends an operation request to the server, and the server sends responses in return. The client does not need to wait for a response before sending the next request, and the server may send the responses in any order. The client may request the following operations: StartTLS, Bind, Search, Compare, Add entry, Delete Entry, Modify Entry, Abandon, Extended Operation, Unbind. The server may send Unsolicited Notifications" that are not responses to any request.

DESIGNING AND MAINTAINING LDAP DIRECTORY

The designing of an LDAP directory are distributed into four phases [4]

- First Phase: The first phase is defining the directory content. It has two components: First component is to define the directory requirements which is to carefully analyse the main purpose of the directory and the consideration to arrive to a holistic approach for the directory plan. The second component is the designing of data which is to understand the source and nature of the data. The scope of the data within the directory is decided and its integration with external data is planned. [4]
- Second Phase: This phase is also divided into two components. First is designing the schema and determining the format in which the data is to be stored. The second component is desinging the namespace and determining the hierarchical structure of the directory.
- Third Phase: This phase involves securing the directory entries. The privacy and security of the data in the directory is the main area of focus. The applications using the directory should also be secured and their security is also considered in this phase.
- Fourth Phase: This phase involves in designing the underlying network infrastructure and designing the server. This phase involves the topology design which helps to determine the number of servers and the location of their directory services. It also considers about the distribution of data amongst the servers. Replication can also be considered in this phase, which enables multiple copies of the data to be deployed. [4]

LDAP COMMAND-LINE TOOLS

LDAP protocol operations are divided into three categories: Authentication, Interrogation, and Update and Control. The LDAP C-API provides a number of simple command-line tools that together covers all three categories [5].

- ldapbind: It authenticates to a directory server. It can also be used to check whether a particular server is running or not.
- ldapsearch: It searches for specific entries in a directory. It opens a connection to a directory, authenticates the user performing the operation, searches for the specified entry, and prints the result of the search operation.
- ldapadd: It adds an entry to the directory. It opens a connection to the directory and authenticates the user and opens the LDIF file supplied as an argument and adds each entry in the file in succession.
- ldapdelete: It removes the leaf entries from a directory. It deletes the specified entry by opening a connetion and authenticating the user.
- ldapmodify: It modifies the existing entries by opening a connection to the directory and opening the LDIF file supplied and then modifies the entry.
- ldapmoddn: It changes the RDN of an entry. It moves an entry or subtree to another location in the directory.

FUTURE OF LDAP

The future of LDAP lies in refinements to LDAPv3. The most recent improvements added include upgrades to managements GUIs that allow easier modification of users and their attributes. The greatest challenge is one shared with any other directory service which includes Active Directory. LDAP's ability to adapt to the changes in delivery of identity and access management which could be possible through new types of authentication such as biometrics or through Software as a Service (SaaS) models. The key features to LDAP's future are its ability to be flexible, scalable and adaptable with new technologies. [2]

CONCLUSION

By reviewing the features and implementation of LDAP, we can conclude that LDAP is lightweight and a Standard wire protocol. LDAP has emerged as a mature protocol for accessing directories. Its servers provide a number of security features such

as automatically encoding passwords with one-way digests, its support for extensible authentication via the SASL framework. It includes general purpose data storage which can hold information about users and groups. Alongwith a variety of features LDAP supports high availability, disaster recovery and logging options to boost its usability. Thus, making LDAP extremely effective.

REFERENCES

[1] Wikipedia, "Lightweight directory access protocol," Web Page, Feb. 2017, online; accessed 20-Feb-2017. [Online]. Available: https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

[2] S. Tuttle, A. Ehlenberger, R. Gorthi, J. Leiserson, R. Machbeth, N. Owen, S. Ranahandola, M. Storrs, and C. Yang, *Understanding LDAP*, 2nd ed. Greenwich, CT, USA: Redbooks, 2004. [Online]. Available: http://www.redbooks.ibm.com/pubs/pdfs/redbooks/sg244986.pdf

[3] Sun Microsystems, Inc., "How ldap servers organize directories," Web Page, Feb. 2006, accessed 2017-02-21. [Online]. Available: https://docs.oracle.com/cd/E19203-01/819-1160-13/ldap.html

[4] Chris Wilson, "Designing and maintaining ldap directory services," Web Page, Feb. 2012, accessed 2017-02-22. [Online]. Available: https://niccs.us-cert.gov/training/search/skillsoft/designing-and-maintaining-ldap-directory-services

[5] Sun Microsystems, Inc., "Ldap command line tools," Web Page, Feb. 2002, accessed 2017-02-21. [Online]. Available: https://docs.oracle.com/cd/B10501_01/network.920/a96579/comtools.htm

Ceph - Distributed Storage System

RAHUL RAGHATATE[1] AND SNEHAL CHEMBURKAR[1]

[1] *School of Informatics and Computing, Bloomington, IN 47408, U.S.A.*
[*] *Corresponding authors: rragahta@iu.edu, snehchem@iu.edu*

paper-1, February 28, 2017

Ceph is a unique storage solution that delivers all critical storage system capabilities:open-source, software-defined, enterprise-class and unified storage (object, block, file). Ceph being highly reliable, easy to manage and free, possesses power to transform company's IT infrastructure and ability to manage vast amounts of data. Ceph's extraordinary scalability has provided clients with accessibility to petabytes to exabytes of data. Moreover basic enterprise storage features including: replication (or erasure coding), snapshots, thin provisioning, auto-tiering (ability to shift data between flash and hard drives), self-healing capabilities has resulted in allowing it to be a reliable big data storage platform. This article explores these salient features Ceph as well as provides study of Ceph architecture and its uniqueness in comparison to few of the existing Large Scale Systems.

© 2017 https://creativecommons.org/licenses/. The authors verify that the text is not plagiarized.

Keywords: Ceph, scalability, storage, exabytes

https://github.com/rahulraghatate/sp17-i524/paper1/S17-IR-2026/report.pdf

INTRODUCTION

Developing a distributed file system(DFS) in today's world of exponentially growing data is not a handy job. However, if the right issues are addressed and resolved, it becomes immensely valuable and foundation stone in any business success. Although most of the DFS offer a similar set of features, they also can provide unique features which allow them to stand distinct.

For IT Decision Makers, inadequate storage infrastructure stand out to be the fourth out of the top ten pain points. It's interesting to know that 74% of IT decision makers are worried about their organization's ability to cope with an increasing volume of data, and 70% believe that their current storage systems will not be able to handle next generation workloads [1]. Hence, Enterprises in due struggle to manage the explosive growth of data while remaining agile and cost competitive are turning to cloud technology to store their data.

As a self-healing, self-managing platform with no single point of failure, Red Hat Ceph Storage significantly lowers the cost of storing enterprise data in the cloud and helps enterprises manage their exponential data growth in an automated fashion [1].

Ceph is open-source storage platform providing highly scalable object, block as well as file-based storage. It is a unified, distributed storage system designed for excellent performance, reliability and scalability [2].

Ceph has emerged as one of the best storage ecosystem which initially began as a PhD research project in storage systems by Sage Weil at the University of California, Santa Cruz (UCSC). The name Ceph comes from Cephalopod, a class of mollusks.

ARCHITECTURE

The Ceph architecture consists of four subsystems:

- File System Clients
- Cluster of metadata servers(MDS)
- RADOS which includes Monitor Services and object storage devices(OSDs)
- Data distribution system using CRUSH

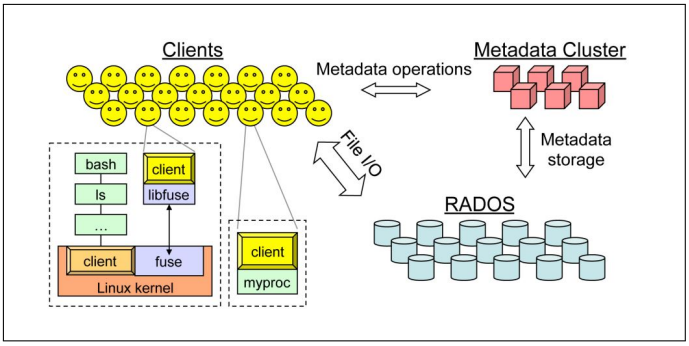


Fig. 1. System Layout of Ceph [3]

Client Operation

Ceph depends upon Ceph Clients and Ceph OSD Daemons having knowledge of the cluster topology, which is inclusive of 5 maps collectively referred to as the "Cluster Map".The Ceph Client is the user of the Ceph file system. The Ceph client runs on each host executing application code and exposes a file system interface to applications. Ceph has a user-level client as well as a kernel client. The user-level client is either linked directly to the application or used via FUSE. The kernel client allows the Ceph file system to be mounted. Each client maintains its own file data cache, independent of the kernel page or buffer caches, making it accessible to applications that link to the client directly [3].

The Ceph metadata server(MDS)

Ceph provides a cluster of metadata servers which continually load-balances itself using dynamic subtree partitioning [4]. The responsibility for managing the namespace hierarchy is adaptively and intelligently distributed among tens or even hundreds of metadata servers. The key to the MDS cluster's adaptability is that Ceph metadata items are very small and can be moved around quickly. To enable failure recovery, the MDS journals metadata updates to OSDs. The mapping of metadata servers to namespace is performed in Ceph using dynamic subtree partitioning, which allows Ceph to adapt to changing workloads (migrating namespaces between metadata servers) while preserving locality for performance. Rebalancing of the MDS at even extreme workload changes is usually accomplished within a few seconds. Clients are notified of relevant partition updates whenever they communicate with the MDS [5].

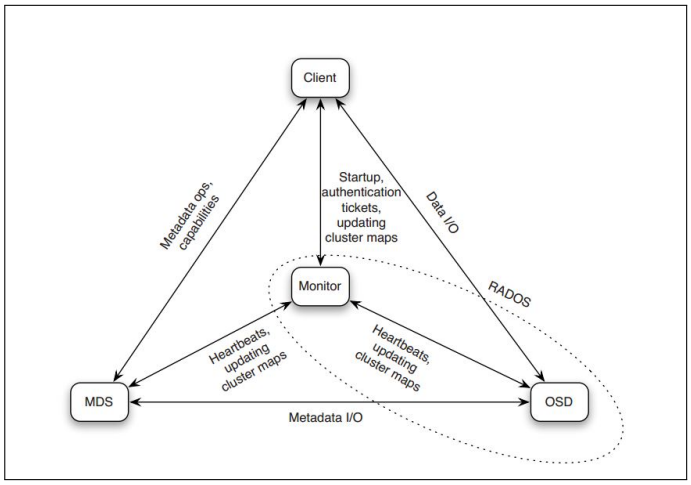


Fig. 2. Ceph components interaction [6].

Reliable Autonomic Distributed Object Storage (RADOS)

From a bird view, object storage cluster made of hundreds of thousands of OSDs as a single logical object store and namespace to the Ceph clients and metadata servers. Ceph's RADOS achieves linearity in both capacity and aggregated performance by delegating management of object replication, cluster expansion, failure detection and recovery to OSDs in a distributed fashion. RADOS can also be used as a stand-alone system.

Unlike other parallel file systems, replication is managed by OSDs instead of clients, which shifts replication bandwidth overhead to the OSD cluster, simplifies the client protocol, and provides fully consistent semantics in mixed read/write workloads. RADOS manages the replication of data using a variant of primary-copy replication and replicas are stored in placement groups which includes a primary OSD which serializes all requests to the placement group.

Writes are applied in two phases and this approach separates writing for the purpose of sharing with other clients from writing for the purpose of durability and makes sharing data very fast. Ceph's failure detection and recovery are fully distributed. The monitor service is only used to update the master copy of the cluster map. OSDs communicates the cluster map updates using epidemic-style propagation that has bounded overhead. This procedure is used to respond to all cluster map updates, whether due to OSD failure, cluster contraction, or expansion. OSDs always collaborate to realize the data distribution specified in the latest cluster map while preserving consistency of read/write access [6].

Data Distribution System

The small size of metadata items in the MDS and the compactness of cluster maps in RADOS are enabled by CRUSH (Controlled Replication Under Scalable Hashing) [7]. Ceph uses this hash function to calculate the placement of data instead of using allocation tables, which can grow very large and unwieldy. CRUSH is part of the cluster map and behaves like a consistent hashing function in that failure, removal, and addition of nodes result in near-minimal object migration to re-establish near-uniform distribution. CRUSH maps a placement group ID to an ordered list of OSDs, using a hierarchically structured cluster map and placement rules as additional input. Any list output by CRUSH meets the constraints specified by placement rules preventing two replicas being placed in same failure domain [3]. Knowledge of failure domains is important for overall data safety of very large storage systems where correlated failures are common.

SALIENT FEATURES

1. Ceph Clients include several service interfaces. These include:
   - Block Devices: The Ceph Block Device (a.k.a., RBD) service provides resizable, thin-provisioned block devices with snapshotting and cloning.
   - Object Storage: The Ceph Object Storage (a.k.a., RGW) service provides RESTful APIs with interfaces that are compatible with Amazon S3 and OpenStack Swift.
   - Filesystem: The Ceph Filesystem (CephFS) service provides a POSIX compliant filesystem usable with mount or as a filesytem in user space (FUSE).
2. Scalability and high availability: In traditional architectures there is single point of entry to a complex subsystem. This imposes a limit to both performance and scalability, while introducing a single point of failure. Ceph eliminates the centralized gateway using CRUSH algorithm to enable clients to interact with Ceph OSD Daemons directly. Ceph OSD Daemons create object replicas on other Ceph Nodes to ensure data safety and Ceph Monitors provide high availability [8].
3. Network Security: Ceph provides its cephx authentication system to authenticate users and daemons. Cephx uses

shared secret keys for mutual authentication, such that both parties can prove to each other they have a copy of the key without actually revealing it.

4. Dyanamic cluster management: Ceph uses CRUSH which enables modern cloud storage infrastructures to place data, re-balance the cluster and recover from faults dynamically. The Ceph storage system supports the notion of 'Pools', which are logical partitions for storing objects.

5. Smart Daemons enable hyperscale [9]: The ability of Ceph Clients, Ceph Monitors and Ceph OSD Daemons to interact with each other allows Ceph OSD Daemons to utilize the CPU and RAM of the Ceph nodes perform task easily that can bog down a centralized server.Leveraging this computing power leads to several major benefits:

    - OSDs Service Clients Directly: Ceph Clients can maintain a session when they need to, and with a certain Ceph OSD Daemon instead of a centralized server.
    - OSD Membership and Status: The Ceph OSD Daemon status reflects whether it is running and able to service Ceph Client requests. Ceph also empowers OSD Daemons with ability to check each other's heartbeats and report back to the Ceph Monitor reliving their burden.
    - Data Scrubbing: Ceph OSD Daemons insures data integrity by scrubbing placement groups.Light scrubbing (daily) catches bugs or filesystem errors. Deep scrubbing (weekly) finds bad sectors on a drive that weren't apparent in a light scrub.
    - Replication:Like Ceph Clients, Ceph OSD Daemons use the CRUSH algorithm, but the Ceph OSD Daemon uses it to compute where replicas of objects should be stored (and for rebalancing).

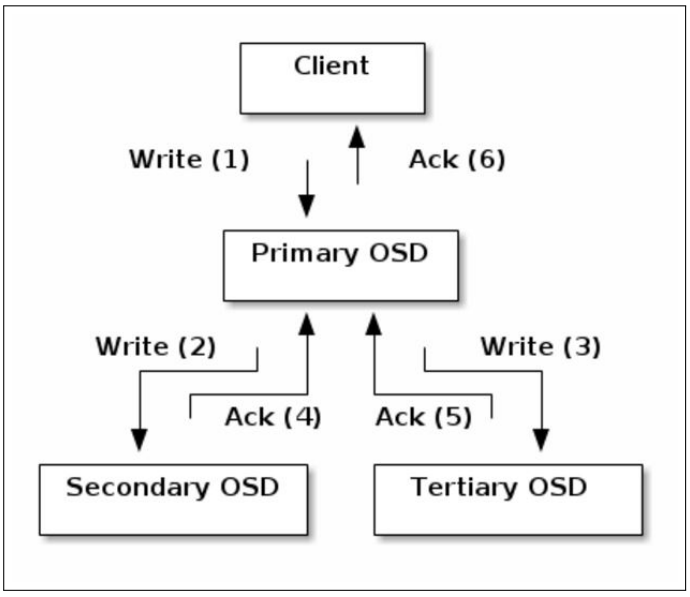


Fig. 3. Replication Process [9].

6. Ceph's provides a native interface to the Ceph Storage Cluster via librados, and a number of service interfaces built on top of librados.

RELATED WORK

Ceph scalability provide high-performance access to a small set of files by tens of thousands of cooperating clients in contrast to Largescale systems like OceanStore [10] and Farsite [11] which fails due to bottlenecks in subsystems such as name lookup. Ceph proves more reliable over other parallel file and storage systems such as Vesta [12], Galley [13], PVFS [14], and Swift [15] due to their lack of strong support for scalable metadata access or robust data distribution. These systems also typically suffer from block allocation issues: blocks are either allocated centrally or via a lock-based mechanism, preventing them from scaling well for thousands of write requests.

USE CASES

Ceph is being used in wide range of applications [16]. Few of them are listed below:

1. Red Hat Ceph Storage team worked with WDLabs and SuperMicro and built and tested a 504 node Ceph cluster with 4 PB of raw storage using these WDlabs Micro-Servers. [17].

2. Cloud Infrastructure for Microbial Bioinformatics (CLIMB) has selected and implemented Red Hat Ceph Storage for their large-scale extensive research needs [18].

3. Yahoo's deployment of the community version of Ceph software for its Flickr and Mail applications on its Cloud Object Store (COS) [19].

4. Red Hat Ceph Storage on Dell PowerEdge server

5. Red Hat Ceph Storage on Intel processors and SSDS

USEFUL RESOURCES

Ceph installation manual [20], provides Installation and Deployment guide which is excellent resource as starter kit. Tutorial on Ceph Deployment by Alan Johnson[21] , is a good tutorial about Ceph deployment.

CONCLUSION

Ceph provides unique solution for the three critical challenges of large scale storage systems—scalability, performance, and reliability. CRUSH and RADOS provides Ceph with improved data safety, ability to manage data replication, failure detection and recovery, low-level disk allocation, scheduling, and data migration without encumbering any central server(s). Ceph's metadata management architecture addresses one of the most vexing problems in highly scalable storage of providing a single uniform directory hierarchy obeying POSIX semantics [3]. Thus, Ceph has proven to be one stop solution for the large-scale storage system in today's Big Data World.

ACKNOWLEDGEMENTS

This work was done as part of the course "I524: Big Data and Open Source Software Projects" at Indiana University during Spring 2017. Many thanks to Professor Gregor von Laszewski and Prof. Geoffrey Fox at Indiana University Bloomington for their academic as well as professional guidance. We would also like to thank Associate Instructors for their help and support during the course.

REFERENCES

[1] Red Hat,Inc., *Red Hat Ceph Storage*, Red Hat,Inc., accessed: 2017-2-26. [Online]. Available: https://www.redhat.com/en/resources/red-hat-ceph-storage-datasheet

[2] Red Hat, Inc., "Ceph homepage-ceph," Web Page, Red Hat, Inc., 2017, accessed: 2017-2-26. [Online]. Available: https://ceph.com/

[3] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in *Proceedings of the 7th symposium on Operating systems design and implementation*, ser. OSDI '06. Berkeley, CA, USA: USENIX Association, Nov. 2006, pp. 307–320, accessed: 2017-1-26. [Online]. Available: https://www.usenix.org/legacy/event/osdi06/tech/full_papers/weil/weil.pdf

[4] S. A. Weil, K. T. Pollack, S. A. Brandt, and E. L. Miller, "Dynamic metadata management for petabyte-scale file systems," in *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing*, ser. SC '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 4–, accessed: 2017-2-26. [Online]. Available: https://doi.org/10.1109/SC.2004.22

[5] M. Jones, "Ceph: A linux petabyte-scale distributed file system," Jun 2010, accessed: 2017-2-26. [Online]. Available: https://www.ibm.com/developerworks/library/l-ceph/

[6] C. Maltzahn, E. Molina-Estolano, A. Khurana, A. J. Nelson, S. A. Brandt, and S. Weil, "Ceph as a scalable alternative to the hadoop distributed file system," *login: The USENIX Magazine*, vol. 35, pp. 38–49, 2010, accessed: 2017-2-26.

[7] R. Latham, N. Miller, R. Ross, P. Carns, Mathematics, and C. U. Computer Science, "A next-generation parallel file system for linux cluster." *LinuxWorld Mag.*, vol. 2, Jan 2004, accessed: 2017-2-26.

[8] Red Hat, Inc., "Ceph homepage-ceph," Web Page, Red Hat, Inc., 2017, accessed: 2017-2-26. [Online]. Available: http://docs.ceph.com/docs/master/architecture/#scalability-and-high-availability

[9] Red Hat,Inc., "Ceph homepage-ceph," Web Page, Red Hat, Inc., 2017, accessed: 2017-2-26. [Online]. Available: http://docs.ceph.com/docs/master/architecture/#smart-daemons-enable-hyperscale

[10] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An architecture for global-scale persistent storage," *SIGPLAN Not.*, vol. 35, no. 11, pp. 190–201, Nov. 2000, accessed: 2017-2-26. [Online]. Available: http://doi.acm.org/10.1145/356989.357007

[11] A. Adya, B. Bolosky, M. Castro, R. Chaiken, G. Cermak, J. J. Douceur, J. Howell, J. Lorch, M. Theimer, and R. a. Wattenhofer, "Farsite: Federated, available, and reliable storage for an incompletely trusted environment," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI)*. Boston, MA: USENIX, December 2002, p. 1–14, accessed: 2017-2-26. [Online]. Available: https://www.usenix.org/legacy/events/osdi02/tech/full_papers/adya/adya.pdf

[12] P. F. Corbett and D. G. Feitelson, "The vesta parallel file system," *ACM Trans. Comput. Syst.*, vol. 14, no. 3, pp. 225–264, Aug. 1996, accessed: 2017-2-26. [Online]. Available: http://doi.acm.org/10.1145/233557.233558

[13] N. Nieuwejaar and D. Kotz, "The galley parallel file system," in *Proceedings of the 10th International Conference on Supercomputing*, ser. ICS '96. New York, NY, USA: ACM, 1996, pp. 374–381, accessed: 2017-2-26. [Online]. Available: http://doi.acm.org/10.1145/237578.237639

[14] R. Latham, N. Miller, R. Ross, P. Carns *et al.*, "A next-generation parallel file system for linux cluster." *LinuxWorld Mag.*, vol. 2, no. ANL/MCS/JA-48544, 2004, accessed: 2017-2-26.

[15] L.-F. Cabrera and D. D. Long, *Swift: Using distributed disk striping to provide high I/O data rates*. University of California, Santa Cruz, Computer Research Laboratory, 1991, vol. 8523, accessed: 2017-2-26. [Online]. Available: https://www.researchgate.net/profile/Darrell_Long2/publication/2752148_Swift_Using_Distributed_Disk_Striping_to_Provide_High_IO_Data_Rates/links/09e415060773f188dd000000.pdf

[16] Red Hat, Inc., "Ceph use cases-ceph," Web Page, Red Hat, Inc., 2017, accessed: 2017-2-26. [Online]. Available: http://docs.ceph.com/use-cases/

[17] Red Hat,Inc., "First large scale ceph storage microserver cluster unveiled," Web Page, Red Hat, Inc., Oct. 2016, accessed: 2017-2-26. [Online]. Available: http://ceph.com/community/500-osd-ceph-cluster/

[18] Red Hat ,Inc., "Climb supports research collaboration with red hat ceph storage," Web Page, Red Hat, Inc., Oct. 2016, accessed: 2017-2-26. [Online]. Available: https://www.redhat.com/en/resources/climb-case-study

[19] N. P.P.S, S. Samal, and S. Nanniyur, "Yahoo cloud object store - object storage at exabyte scale |yahoo engineering," Web Page, Yahoo Engineering, Apr. 2015, accessed: 2017-2-26. [Online]. Available: https://yahooeng.tumblr.com/post/116391291701/yahoo-cloud-object-store-object-storage-at

[20] Red Hat, Inc., "Installation (manual)- ceph documentation," Web Page, Red Hat, Inc., 2017, accessed: 2017-2-26. [Online]. Available: http://docs.ceph.com/docs/master/install/

[21] Alan Johnson, "Ceph - hands-on guide |aj's data storage tutorials," Web Page, accessed: 2017-2-26. [Online]. Available: https://alanxelsys.com/ceph-hands-on-guide/