

Big Data Software

Spring 2017

Bloomington, Indiana

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

Contents

1 S17-ER-1001		
Globus Toolkit		
Saber Sheybani,		6
2 S17-IO-3000		
CDAP Cask Data Application Platform		
Avadhoot Agasti		9
3 S17-IO-3004		
MySQL		
Cory Coulter		12
4 S17-IO-3005		
RabbitMQ		
Abhishek Gupta		15
5 S17-IO-3008		
Docker Container		
Vishwanath Kodre		18
6 S17-IO-3010		
Couchbase Server: A Usable Overview		
Matthew Lawson		20
7 S17-IO-3011		
Apache Airavata		
Scott McClary		24
8 S17-IO-3012		
Google Bigtable		
Mark McCombe		27
9 S17-IO-3013		
Apache Beam (Google Cloud Dataflow)		
Leonard Mwangi		31
10 S17-IO-3014		
Xen: A bare metal hypervisor		
Piyush Rai		34
11 S17-IO-3015		
Apache Lucene		
Roy Choudhury, Sabyasachi		36

12	S17-IO-3016		
	CoreOS		
	Ribka Rufael		38
13	S17-IO-3017		
	MongoDB		
	Nandita Sathe		41
14	S17-IO-3019		
	vCloud and vSphere		
	Michael Smith		45
15	S17-IO-3020		
	Google Fusion Table		
	Milind Suryawanshi		48
16	S17-IO-3021		
	CUBRID RDBMS		
	Abhijit Thakre		51
17	S17-IO-3022		
	Netty vs ZeroMQ in Realtime Analytics		
	Sunanda Unni		54
18	S17-IO-3023		
	H-Store		
	Karthick Venkatesan		57
19	S17-IO-3024		
	Not Submitted		
	Ashok Vuppada		61
20	S17-IR-2001		
	HTCondor: Distributed Workflow Management System		
	Niteesh Kumar Akurati		64
21	S17-IR-2002		
	Google Dremel: SQL-Like Query for Big Data		
	Jimmy Ardiansyah		67
22	S17-IR-2004		
	Apache Samza		
	Ajit Balaga, S17-IR-2004		70
23	S17-IR-2006		
	An Overview of Apache Spark		
	Snehal Chemburkar, Rahul Raghatate		73

24 S17-IR-2008	An overview of HadoopDB and its Architecture Karthik Anbazhagan	78
25 S17-IR-2011	Ansible Anurag Kumar Jain, Gregor von Laszewski	81
26 S17-IR-2012	Lustre File System Pratik Jain	84
27 S17-IR-2013	An overview of Flume and its Applications in BigData Sahiti Korrapati	87
28 S17-IR-2014	An Overview of Apache Sqoop Harshit Krishnakumar	90
29 S17-IR-2016	Apache Spark's Machine Learning Library (MLlib) Anvesh Nayan Lingampalli	93
30 S17-IR-2017	An Overview of OpenNebula Project and its Applications Author Missing	96
31 S17-IR-2018	Analysis of Pentaho Bhavesh Reddy Merugureddy	99
32 S17-IR-2019	Twister: A new approach to MapReduce Programming Vasanth Methkupalli	102
33 S17-IR-2021	Docker Machine and Swarm Shree Govind Mishra	105
34 S17-IR-2022	Triana Abhishek Naik	108
35 S17-IR-2024	LDAP Ronak Parekh, Gregor von Laszewski	111

36 S17-IR-2026	Ceph - Distributed Storage System Rahul Raghatare, Snehal Chemburkar	114
37 S17-IR-2027	Twitter Heron Shahidhya Ramachandran	118
38 S17-IR-2028	A Report on Kubernetes Srikanth Ramanam	122
39 S17-IR-2029	An overview of Azure Machine Learning and its Applications Naveenkumar Ramaraju	125
40 S17-IR-2030	Microsoft Azure Data Factory Sowmya Ravi	128
41 S17-IR-2031	Google Cloud storage: A journey towards Cloud storage Kumar Satyam	131
42 S17-IR-2034	Apache Drill Yatin Sharma	134
43 S17-IR-2035	Oracle PGX Piyush Shinde	137
44 S17-IR-2036	An Overview of the Java Message Service (JMS) Rahul Singh	141
45 S17-IR-2037	File Transfer Protocol - An Overview Sriram Sitharaman	144
46 S17-IR-2038	Introduction to H2O Sushmita Sivaprasad	148
47 S17-IR-2041	Tajo: A Distributed Warehouse System for Large Datasets Sagar Vora	151

48	S17-IR-2044 Allegro Graph Diksha Yadav	154
49	S17-TS-0003 TBD Tony Liu, Vibhatha Abeykoon, Gregor von Laszewski	156
50	S17-TS-0006 Not Submitted Author Missing	159

Globus Toolkit

SABER SHEYBANI^{1,*}

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

* Corresponding authors: sheybani@indiana.edu

paper-001, April 12, 2017

The Globus Toolkit is an open source software stack, developed to serve as the middleware for grid computing. It is organized as a collection of loosely coupled components consisting of services, programming libraries and development tools designed for building grid-based applications. GT components fall into five broad domain areas: Security, Data Management, Execution Management, Information Services, and Common Runtime.

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

Keywords: Globus Toolkit, grid computing, WSRF, OGSA, cluster resource management

<https://github.com/ssheybani/sp17-i524/tree/master/paper1/S17-ER-1001/report.pdf>

1. INTRODUCTION

Initially, work on Globus was motivated by the demands of "virtual organizations" in science. These organizations need access to resources and services which are not easily replicable locally. Examples of these needs include access and management of equipment and large amounts of data, located in various remote databases on a regular basis. Although each application may have different specific requirements, the demand for a few basic functions is shared between most applications. "They often need to discover available resources, configure a computing resource to run an application, move data reliably from one site to another, monitor system components, control who can do what, and manage user credentials". These functions can be well addressed in a grid computing framework [1].

According to [2], "*a grid is a system that coordinates resources that are not subject to centralized control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service*". The major difference between grids and other distributed systems is that grids have significantly larger extent of heterogeneity of resources. They also enable resource sharing among virtual organizations, whereas other distributed computing technologies only support a single organization [3]. The grid architecture consists of a number of layers that altogether serve the functions and purposes of the systems (See figure 1). EGEE: Enabling Grids for E-Science in Europe [4], NEESit: Network for Earthquake Engineering Simulation [5], TeraGrid [6] are examples of grid systems.

A grid system often comprises various services such as Job Management Service and Resource Discovery and Management Service, which constantly need to interact with each other. However, each of these services may have been implemented by different vendors, which is not necessarily compatible with other services. The Open Grid Services Architecture (OGSA) aims

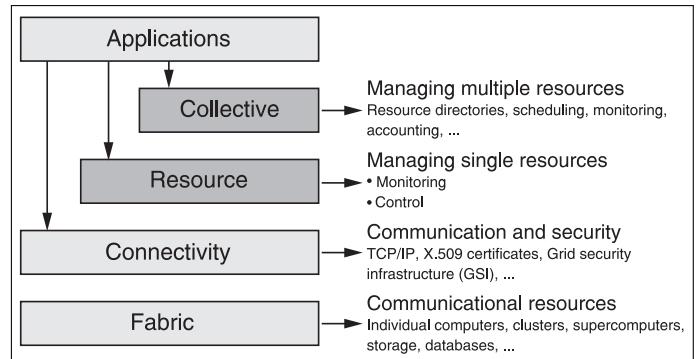


Fig. 1. Grid Architecture [2]

to standardize practically all the services that can commonly be found in a grid system (job management services, resource management services, security services, etc.) by specifying a set of standard interfaces for these services. In order to realize that goal, OGSA needs to choose some sort of distributed middleware on which to base the architecture. In other words, if OGSA (for example) defines that the JobSubmissionInterface has a submitJob operation, there has to be a common and standard way to invoke that operation. **Web Services** were chosen as the underlying technology. Web services are platform-independent and language-independent since they are message-oriented and rely on language-neutral XML dialects to send messages, to specify interfaces, etc. Hence, web services are well-suited for building loosely coupled systems such as grid systems.

However, although Web services can, in theory, be either stateless or stateful, they are usually stateless. This means that the Web service can't "remember" information, or keep state,

from one invocation to another. On the other hand, OGSA requires Stateful Services. The **Web Services Resource Framework (WSRF)** specifies how we can make our Web Services stateful, along with adding other features. In other words, while OGSA is the architecture, WSRF is the infrastructure on which that architecture is built on.

With this background, we can define Globus Toolkit as an open source toolkit that provides an implementation of OGSA, WSRF and a number of other standards for grid computing. Figure 2 displays the relationship between OGSA, Web service, WSRF, Globus Toolkit.

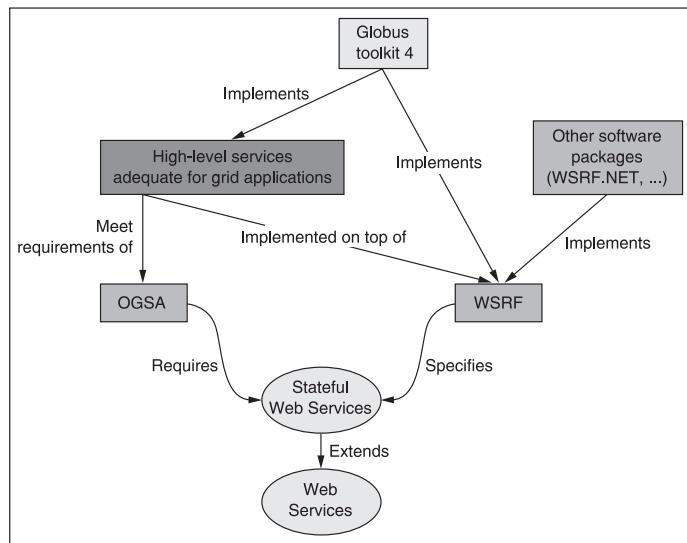


Fig. 2. Relationship between OGSA, GT4, WSRF, and Web Services [2]

2. COMPONENTS

The libraries and services in Globus Toolkit can be classified into five broad domain areas, namely security, data management, execution management, information services, and common runtime. Components in each of these areas will be discussed further in this section [2].

Security : Grid Security Infrastructure (GSI) is a collection of components in GT4 that facilitate secure communication in the system. These components include authentication and authorization (for controlling access to services and resources), delegation, community authorization, and credential management.

Data Management : These components provide for the discovery, transfer, and access to large data. They include GridFTP (an implementation of a protocol optimized for transferring large amounts of data between hosts), Reliable File Transfer (adds several features to GridFTP), Replica Location Service(keeping track of different replicas of a dataset in a virtual organization), Data Replication (uses RLS, RFT to guarantee that local copies of replicas are available to the hosts), and OGSA Data Access and Integration (a framework to access and integrate datasets on a Grid).

Execution Management : These components handle the deployment, scheduling, and monitoring of executable programs, referred to as jobs. They include Grid Resource

Allocation and Management (the heart of GT Execution Management, providing services to deploy and monitor jobs), Community Scheduler Framework (provides a single interface to different resource schedulers), Workspace Management (allows users to dynamically create and manage workspaces on remote hosts), and Grid Telecontrol Protocol (provides a WSRF-enabled service interface for control of remote instruments).

Information Services : Commonly referred to as the Monitoring and Discovery System (MDS), these services deal with monitoring and discovery of resources in a virtual organization. They include Index Service(for aggregation of resources of interest to a VO), Trigger Service (same as Index Service, but is configured to perform certain actions based on the data collected from resources), and WebMDS (provides a web browser-based view of data collected by GT4 aggregator services).

Common Runtime : These components provide a set of fundamental libraries and tools for hosting existing services as well as developing new services, in languages of C, Java, and Python (C Runtime, Java Runtime, and Python Runtime).

3. APPLICATIONS

Globus Toolkit has been used in many scientific and industrial applications. Examples of large-scale e-science projects relying on the Globus Toolkit include the Network for Earthquake Engineering and Simulation (NEES), FusionGrid, the Earth System Grid (ESG), the NSF Middleware Initiative and its GRIDS Center, and the National Virtual Observatory. In the design of the Large Hadron Collider at CERN, Globus-based technologies have been developed through the European Data Grid, and the U.S. efforts like the Grid Physics Network (GriPhyN) and Particle Physics Data Grid [7].

Globus Toolkit is helping to bridge the gap for commercial applications of Grid computing. Thus, companies like Avaki, Data-Synapse, Entropia, Fujitsu, Hewlett-Packard, IBM, NEC, Oracle, Platform, Sun, and United Devices have pursued Grid strategies based on the Globus Toolkit. A company named Univa Corporation is devoted to providing commercial support for Globus software. In addition, the Globus Consortium was formed by a group of companies with an interest in supporting Globus Toolkit enhancements for enterprise use [7].

4. ECOSYSTEM AND SUPPORT

Globus Toolkit is a part of the larger "Globus ecosystem", which provides a wide range of application-level functions in many domains. Other than the GT, the ecosystem includes a number of other tools, a resource provider and a community of users and developers [1].

Support for Globus ecosystem includes documentation, software updates, mailing lists, and training. The documentation includes manuals, specific for general use, system administrators, developers, and end user. In addition, many tutorials, papers, and presentations are accessible on the website [8]. Workshops and conferences for in-person learning are also available. For Java developers, [2] is a specifically useful resource for learning how to implement a grid using GT from scratch.

5. RELATED SOFTWARE

There is a number of software available for use with certain versions of GT, in order to extend the abilities of each of its components. Here are a few examples of each: GridShib and PERMIs for security, Weka4WS for data management, HandleSystem-GT Project for information services, Condor-G, GridWay and Sun Grid Engine for execution management, Grid Packaging Tools (GPT) for packaging, pyGridware, Java CoG Kit and WSRF.NET for programming, and Common TeraGrid Software Stack (CTSS) and LHC Computing Grid (LCG) for software distributions. The full list of software is available in [www-related-tools].

6. COMPARISON

Currently, there are multiple versions of grid middleware stacks. In [9], it is argued that none of the available stacks is generally accepted by all user communities. However, started in 1996, the Globus project is the first grid middleware created [10] and has managed to keep a large community of users. Figure 3 shows a comparison of GT and three other grid middleware technologies, namely UNICORE, Legion, and Gridbus. Efforts have been made to combine GT and UNICORE, in order to get the best of both of them [11].

Middleware Property	UNICORE	Globus	Legion	Gridbus
Focus	High level Programming models	Low level services	High level Programming models	Abstractions and market models
Category	Mainly uniform job submission and monitoring	Generic computational	Generic computational	Generic computational
Architecture	Vertical multi tiered system	Layered and modular toolkit	Vertically integrated system	Layered component and utility model
Implementation Model	Abstract Job Object	Hourglass model at system level	Object-oriented metasystem	Hourglass model at user level
Implementation Technologies	Java	C and Java	C++	C, Java, C# and Perl
Runtime Platform	Unix	Unix	Unix	Unix and Windows with .NET (Alchemi)
Programming Environment	Workflow environment	Replacement libraries for Unix & C libraries. Special MPI library (MPICH-G), CoG (Commodity Grid) kits in Java, Python, CORBA, Matlab, Java Server Pages, Perl and Web Services	Legion Application Programming Interfaces (API). Command line utilities	Broker Java API XML-based parameter-sweep language Grid Thread model via Alchemi.
Distribution Model	Open source	Open source	Not open source. Commercial version available	Open source
Some Users and Applications	EuroGrid [18], Grid Interoperability Project (GRIP) [20], OpenMolGrid [19], and Japanese NAREGI [22].	AppLeS [28], Ninf [30], Nimrod-G [29], NASA IPG [36], Condor-G [31], Gridbus Broker [32], UK eScience Project [33], GriPhyN [35], and EU Data Grid [34].	NPACI Testbed [42], Nimrod-L [41], and NCBioGrid [40]. Additionally, it has been used in the study of axially symmetric steady flow [39] and protein folding [38] applications.	ePhysics Portal [52], Belle Analysis Data Grid [50], NeuroGrid [48], Natural Language Engineering [53], HydroGrid [46], and Amsterdam Private Grid [47].

Fig. 3. Comparison of Grid Middleware Systems [12]

8. ACKNOWLEDGEMENT

The author would like to thank Professor Gregor von Laszewski and the teaching assistants for all the support and guidance. This paper is written during spring of 2017 semester course I524: Big Data and Open Source Software Projects at Indiana University Bloomington.

REFERENCES

- [1] I. Foster, "Globus toolkit version 4: Software for service-oriented systems," *Journal of computer science and technology*, vol. 21, no. 4, p. 513, 2006.
- [2] B. Sotomayor and L. Childers, *Globus® Toolkit 4: Programming Java Services*. Morgan Kaufmann, 2006.
- [3] F. Magoules, J. Pan, K.-A. Tan, and A. Kumar, *Introduction to grid computing*. CRC Press, 2009.
- [4] "Egee: Enabling grids for e-science in europe," Web Page, accessed: 2017-2-26. [Online]. Available: <https://eu-egee.org/>
- [5] "Neesit: Network for earthquake engineering simulation," Web Page, accessed: 2017-2-26. [Online]. Available: <https://nees.org/>
- [6] "Teragrid," Web Page, accessed: 2017-2-26. [Online]. Available: <https://en.wikipedia.org/wiki/TeraGrid>
- [7] "About the globus toolkit," Web Page, accessed: 2017-2-26. [Online]. Available: <http://toolkit.globus.org/toolkit/about.html>
- [8] "Globus toolkit presentations and tutorials," Web Page, accessed: 2017-2-26. [Online]. Available: <http://toolkit.globus.org/toolkit/presentations/>
- [9] U. Schwiegelshohn, R. M. Badia, M. Bubak, M. Danelutto, S. Dustdar, F. Gagliardi, A. Geiger, L. Hluchy, D. Kranzlmüller, E. Laure *et al.*, "Perspectives on grid computing," *Future Generation Computer Systems*, vol. 26, no. 8, pp. 1104–1115, 2010.
- [10] "Father of the grid," Web Page, accessed: 2017-2-26. [Online]. Available: <http://magazine.uchicago.edu/0404/features/index.shtml>
- [11] M. Rambadt and P. Wieder, "Unicore-globus interoperability: Getting the best of both worlds." in *HPDC*. Citeseer, 2002, p. 422.
- [12] P. Asadzadeh, R. Buyya, C. L. Kei, D. Nayar, and S. Venugopal, *Global grids and software toolkits: A study of four grid middleware technologies*. Wiley Press, New Jersey, USA, 2005. [Online]. Available: <http://gridbus.csse.unimelb.edu.au/papers/gridmiddleware.doc>

7. CONCLUSION

Globus toolkit is a collection of tools for building grid computing networks, which is a suitable solution for distributed computing. It provides APIs for languages of C, Python, and Java. It is the most widely used grid middleware, in comparison to its competitors and has been used in many notable scientific and industrial projects. Extensive documentations and tutorials are available for beginners to start using Globus Toolkit.

CDAP Cask Data Application Platform

AVADHOO 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, April 12, 2017

CDAP provides application development platform on top of Apache Hadoop. CDAP services enable users to automate the task of building, executing and managing data pipelines. The CDAP studio allows users to drag-and-drop data sources, transformation tasks, and data sinks. User can chain these tasks to create data pipelines. Furthermore, CDAP provides abstraction of logical data pipeline over execution environment. Using CDAP, user can execute a data pipeline either using MapReduce or Apache Spark depending on the capability of underlying Hadoop cluster.

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

Keywords: CDAP, Hadoop, Name Node, Edge Node, Yarn, Apache Sqoop, Apache Flume, Apache Flink, Apache Atlas, HDFS, Apache Kafka, Apache Spark, MapReduce, HBase

<https://github.com/cloudmesh/sp17-i524/raw/master/paper1/S17-IO-3000/report.pdf>

1. INTRODUCTION

CDAP stands for Cask Data Application Platform [1]. CDAP is an application development platform that can help developers build, deploy and monitor applications on Apache Hadoop. In a CDAP application, a developer can ingest a dataset, store and manage it on Hadoop, perform data analysis, and develop web services to expose the original and transformed dataset(s). He can also schedule and monitor the execution of the application. CDAP enables the developers to use single platform to develop the end-to-end application on Apache Hadoop.

This technology paper is structured as follows:

- Section 2 explains commonly used application architecture on top of Apache Hadoop without using CDAP. It then explains the application architecture using CDAP to emphasize the use of CDAP.
- Section 3 explains important CDAP concepts.
- Section 4 and Section 5 explain the CDAP deployment options and infrastructure requirements.
- Section 6 explains the representative use cases.
- Section 7 and 8 provide useful information about CDAP like licensing and educational material.
- Finally, in Section 9 and 10, we conclude by explaining the other similar platforms and their high level comparison with CDAP

2. CDAP: UNIFIED APPLICATION DEVELOPMENT PLATFORM

CDAP provides a unified application development platform on top of Apache Hadoop. In below sections, we explain a commonly used application architecture on top of Apache Hadoop. Then we explain the CDAP application architecture using which different components of the application architecture are unified in CDAP platform.

2.1. Application Architecture on Hadoop without CDAP

Figure 1 [2] shows a commonly used application architecture on Hadoop (without CDAP).

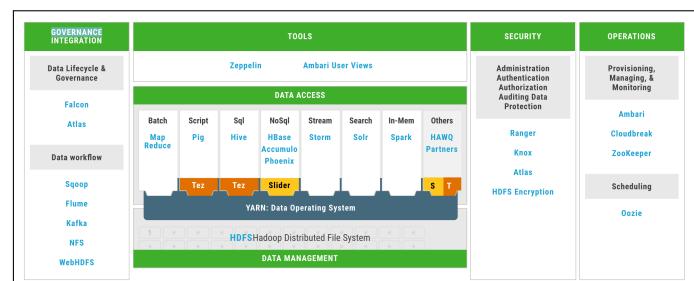


Fig. 1. Application Architecture on Hadoop.

There are following layers/components as depicted in the architecture diagram [2].

- Data Governance and Integration: This layer is responsible for ingesting the data from data source into Hadoop.

Data Ingestion tools like Apache Sqoop, Apache Flume and Apache Kafka are used for Data Ingestion while Apache Falcon and Apache Atlas are used for Data Lifecycle management.

- Data Storage: The data is stored in HDFS.
- Data Processing and Access: The data is transformed and aggregated in Data Processing layer. The processing can involve various steps like cleansing, joining, aggregation and running machine learning algorithms. Many different tools and technologies are used to perform data processing operations. PIG, Hive, Spark are open source scripting technologies which can perform Data Processing tasks.
- Tools: Tools like Apache Zeppelin provide user interface to visualize the data.
- Security and Operations: The data pipeline built using above tools can be secured using tools such as Ranger, Knox, and Atlas. The scheduling of the data pipeline is orchestrated using Oozie.

This application architecture explains that an application built on top of Apache Hadoop involve multiple tools and technologies. Further it is tightly dependent on the underlying deployment architecture and tools availability. Using this mechanism, it is difficult to segregate the application business logic from infrastructure components.

2.2. Application Architecture on Hadoop with CDAP

Figure 2 [3] explains how CDAP provides a common application development platform. 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. Developers need not use different tools to implement various layers. He can implement all the layers in CDAP platform. Further, he can use Java as a common coding language to implement all the coding tasks across all the layers.

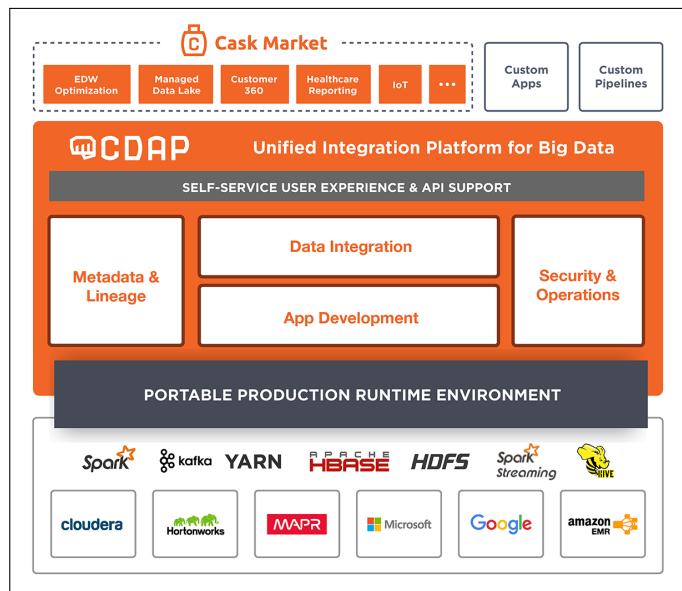


Fig. 2. CDAP Application Architecture.

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 MapReduce, 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 multiple 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 deployment in cluster mode, CDAP provides options which are specific to underlying Hadoop distribution as explained below:

- Cloudera Hadoop Distribution (CDH) - Cloudera Manager [4] is tool to deploy CDH cluster. CDAP provides CDAP-parcel [5] 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 [6]. EMR provides 'Create Cluster Wizard' to create EMR cluster. According the CDAP documentation [7], 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 [8] provide the CDAP deployment architecture.

6. REPRESENTATIVE USE CASES

CASK [1] 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 [9] 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, transform 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.

- CASK Customer 360 [10] is another representative application which is 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.

7. LICENSING

CDAP is licensed [11] under Apache License, Version 2.0.

8. EDUCATIONAL MATERIAL

Below given is the list of educational material on CDAP:

- CDPA Applications code repository in Github [12] provide sample applications which are built on top of CDAP Platform.
- CDAP Documentation [13] provides introduction to CDAP platform.

9. OTHER HADOOP APPLICATION DEVELOPMENT PLATFORMS

Below given are the two other application development platforms on top of Apache Hadoop:

- Cascading [14] 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 [15] : 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 a unified application development platform over Apache Hadoop. Using CDAP developers can implement multiple layers of their data pipeline in one uniform language and tool.

ACKNOWLEDGEMENTS

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

REFERENCES

- [1] CASK, "Cask - the first unified integration platform for big data," Web Page, online; accessed 18-Feb-2017. [Online]. Available: <http://cask.co/>
- [2] Hortonworks, "Hortonworks data platform," Web Page, online; accessed 20-Mar-2017. [Online]. Available: <https://hortonworks.com/products/data-center/hdp/>
- [3] CASK, "The first unified integration platform for bigdata that cuts down the time to production for data applications and data lakes by 80online; accessed 20-Mar-2017. [Online]. Available: <https://cask.co/products/cdap/>
- [4] Cloudera, "Simple administration for apache hadoop," Web Page, online; accessed 20-Mar-2017. [Online]. Available: <https://www.cloudera.com/products/product-components/cloudera-manager.html>
- [5] 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>
- [6] Amazon, "Amazon emr," Web Page, online; accessed 20-Mar-2017. [Online]. Available: <https://aws.amazon.com/emr/>
- [7] CASK, "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>
- [8] CASK, "System requirements," Web Page, online; accessed 18-Feb-2017. [Online]. Available: <http://docs.cask.co/cdap/current/en/admin-manual/system-requirements.html>
- [9] CASK, "Cask - hydrator," Web Page, online; accessed 18-Feb-2017. [Online]. Available: <http://cask.co/products/hydrator/>
- [10] CASK, "Customer 360," Web Page, online; accessed 20-Mar-2017. [Online]. Available: <http://cask.co/solutions/customer-360/>
- [11] CASK, "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>
- [12] "Cdap applications," Code Repository, May 2015, accessed: 2017-2-18. [Online]. Available: <https://github.com/caskdata/cdap-apps>
- [13] 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>
- [14] Cascading, "Cascading," Web Page, online; accessed 18-Feb-2017. [Online]. Available: <http://www.cascading.org/projects/cascading/>
- [15] 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, April 12, 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>

1. 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 Michael Widenius began creating an in-house tool for managing databases. [2]

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.

3. 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.

4. RDBMS

Relational Database Management Systems (RDBMS) are programs that allow for the creation, updating and administration of relational databases. [4] Many commercial RDBMS software 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]

5. 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.

5.1. 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 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]

5.2. 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]

5.3. Health-care

In the health-care industry, MySQL can be used to manage data related to scheduling, billing, prescriptions, and Electronic Medi-

cal 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]

6. 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. w3schools.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]

7. 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.

7.1. SQLite

SQLite is an embedded RDBMS. [18] It is a file-based database that contains 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]

7.2. 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 relational database management system, PostgreSQL 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 management system (rdbms)," Website, November 2005. [Online]. Available: <http://searchsqlserver.techtarget.com/definition/relational-database-management-system>
- [7] M. Rouse, "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. Mattheie, "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, April 12, 2017

RabbitMQ provides 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 [2] based 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 read these messages from the exchange and write to queues. Consumers can further read messages from the queue. It also supports a notion of acknowledgments where consumers can post an acknowledgment back to broker and which in turn can post messages back to the producer. Broker plays an important role, it eliminates the well-known producer-consumer problem [4]. It also provides persistence in case of failures.

Looking back at the history [5] of the development of RabbitMQ, it started with IBM Miseries in 1993, 1997 Microsoft MQ, 2001 java messaging service. These technologies were incubated in different companies with a common goal to create a messaging service for a distributed system. It was then in 2003 where JP Morgan created the first version AMQP which became the base for RabbitMQ technologies formed in 2006.

2. ARCHITECTURE

At high level, producers publish messages to the broker. Consumers consume messages from the broker. Broker acts like a middle man, who has knowledge of all available exchanges and queues [5]. It maintains bindings between exchanges and queues. Exchanges here are often compared to post offices or mailboxes. Following table shows different type of exchanges supported by RabbitMQ and corresponding default exchanges [2].

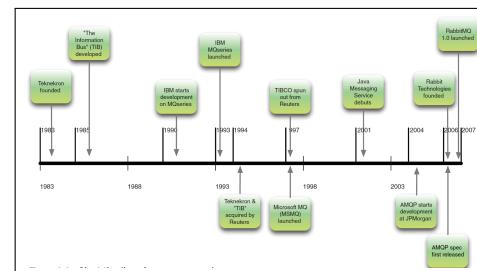


Fig. 1. Timeline for evolution of RabbitMQ [5]

An exchange is a binding to queue(s). So when you send a message to an exchange, the broker checks the routing key of the message and compares it against the queue bindings. Further it sends the message to appropriate queue.

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 bound to a given exchange. Fanout exchange ignores the routing key. For example, if there are 5 queues bound to an exchange 'E'. Messages to exchange 'E' are delivered to all 5 queues.

3.4. Topic exchange

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

3.5. Header exchange

This exchange ignores the routing key but rather uses a header parameter to decide which message 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 allows messages to persist on disk in case broker restarts. Auto-delete deletes the exchange once all queues have finished using the exchange.

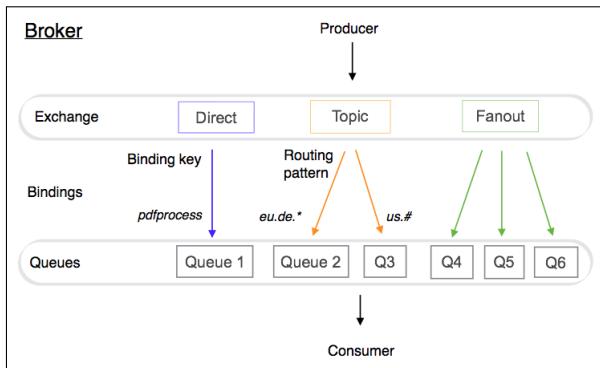


Fig. 2. Different type of exchanges supported by RabbitMQ[6]

4. BINDINGS

Binding is a rule that defines how exchanges will route messages to the queue. To route all messages from an exchange E to queue Q, Q has to be bound to Exchange E. Binding uses 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

RabbitMQ has a built-in mechanism to send and receive acknowledgements. Producer can send a message and wait for the acknowledgement in the response queue of the consumer. Once consumer successfully receives the message, it can post an acknowledgement to the response queue. Here request queue and request exchange can be used to send and receive messages. Response queue and response exchange can be used to send and receive acknowledgements. This mechanism makes RabbitMQ robust to handle failure scenarios.

7. CLUSTERING

To achieve high availability [5] 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 mode, where broker, producer and consumer run on different nodes, RabbitMQ only stores information about the queue like metadata, state and contents on one node rather than all nodes on the cluster. However, it stores metadata and pointer to actual data on each node on the cluster. This is to optimize storage space and performance.

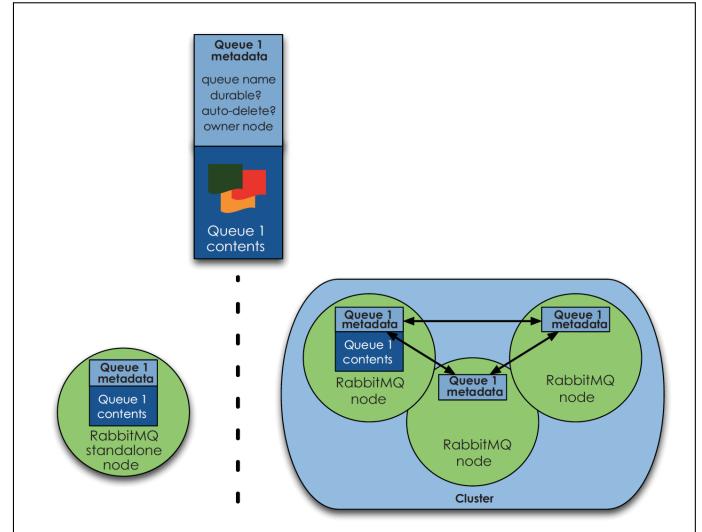


Fig. 3. Shows queue metadata for a cluster vs single node [5]

Since, exchanges act like lookup tables for queue bindings, they are replicated across all nodes on the cluster.

8. MANAGEMENT

RabbitMQ provides all management using:

- Web UI
- REST interface
- rabbitmqctl command line utility

Web UI can be used by the administrator to create user, monitor queues and exchanges, view statistics, add configurations etc. Similar functionality can be achieved by CLI(command line interface) utility called rabbitmqctl or using REST API. Rabbitmqctl can be used to automate RabbitMQ deployments and management. It can also be used to write automated tests. REST API can be 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 '/api' URI.



Fig. 4. RabbitMQ management UI showing messages queued and message data rates[6]

9. LICENSING

RabbitMQ is licensed under Mozilla Public License(MPL) and GPL v2 [7]. MPL license [8] allows "for free use, modification, distribution, and "exploit[ation]" of the work, but does not grant the licensee any rights to a contributor's trademarks".

10. USE CASES

RabbitMQ messaging can be useful for applications which require asynchronous messaging i.e. an application initiates a task by posting a message to RabbitMQ, it does not have to wait for the task to get completed. Application can periodically check for the status of completion of the task. RabbitMQ can scale up as the data needs of the application grow, by just adding additional nodes to the RabbitMQ cluster. When applications run as micro-services in a containerized environment, RabbitMQ is useful to communicate and share data between different services. RabbitMQ can be managed separately using management Web UI, CLI tool and REST API. This decouples the messaging layer from application and makes the overall design 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. It can handle more data by adding more nodes to the cluster. Based on tests [9] 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. Due to time constraints there is no concrete conclusion to these test and numbers. Further testing should be conducted on a bigger cluster i.e. more nodes and for longer duration to produce more consistent results.

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] Pivotal, "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] Wikipedia, "Producer-consumer problem," Web Page, accessed: 03-18-2017. [Online]. Available: https://en.wikipedia.org/wiki/Producer%20E2%80%93consumer_problem
- [5] A. Videla and J. J. Williams, *RabbitMQ in action*. Manning, 2012.
- [6] 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>
- [7] Pivotal, "RabbitMQ license," Web Page, accessed: 2017-02-02. [Online]. Available: <https://www.rabbitmq.com/mlp.html>
- [8] Wikipedia, "Mozilla public license," Web Page, accessed: 2017-03-18. [Online]. Available: https://en.wikipedia.org/wiki/Mozilla_Public_License
- [9] 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,*}

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

* Corresponding authors: vkodre@iu.edu

paper1, April 12, 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>

1. 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.

2. 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]

3. 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.

4. 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 RabbitMQ and nginx, and move all of their main application services over to Container. Which facilitate immutable, transferable development platform and deployment pipeline. 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]

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.

6. 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 thrill 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 LAWSON¹

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

* Corresponding authors: laszewski@gmail.com

Paper1, April 12, 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: Couchbase, Memcache, CouchDB, Cloud, I524

<https://github.com/cloudmesh/sp17-i524/raw/master/paper1/S17-IO-3010/report.pdf>

1. 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, Couchbase markets a mobile product, Couchbase Mobile, which uses a Couchbase-designed syncing system to extend CBS to mobile devices and offline use cases.

2. 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*. RE-SPONSE: 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 node'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 on 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.

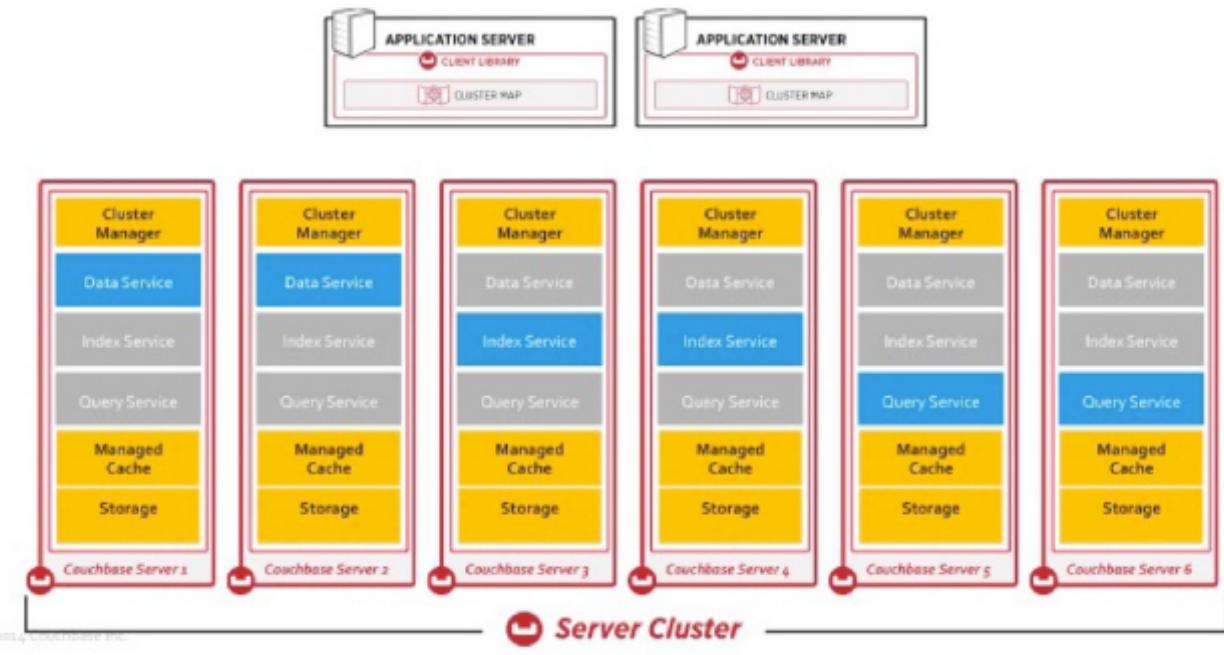


Fig. 1. Multidimensional Scaling with Couchbase [1]

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]Views 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 utilize 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 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.)."

3. 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].

4. 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.

5. 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 librarians, 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].

6. 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]

7. EDUCATIONAL MATERIAL

If you would like to learn more about Couchbase Server, visiting 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.

8. 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.

9. 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] Couchbase, Inc., "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, Inc., "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] Couchbase, Inc., "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] Couchbase, Inc., "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] Couchbase, Inc., "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] Couchbase, Inc., "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] Couchbase, Inc., "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] Couchbase, Inc., "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, Inc., "Couchbase server & n1ql," Web page, feb 2017, accessed 23-feb-2017. [Online]. Available: <https://www.couchbase.com/downloads>
- [11] Couchbase, Inc., "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] Couchbase, Inc., "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] Couchbase, 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=0dKXH6vJRA>
- [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_

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.

A. 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,*}

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

* Corresponding authors: scmcclar@indiana.edu

paper-001, April 12, 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, Middleware, Workflow

<https://github.com/cloudmesh/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 inter-

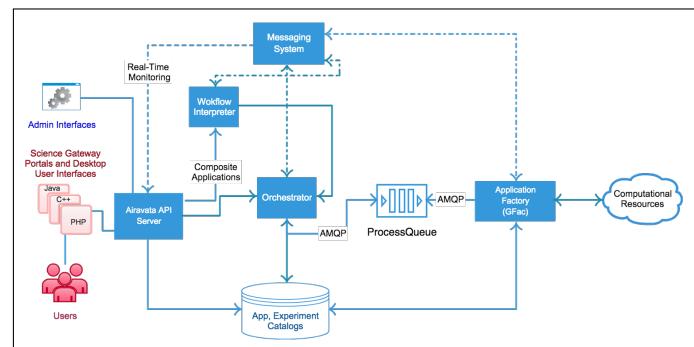


Fig. 1. The image above depicts the architectural details of Apache Airavata [2].

act 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].

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

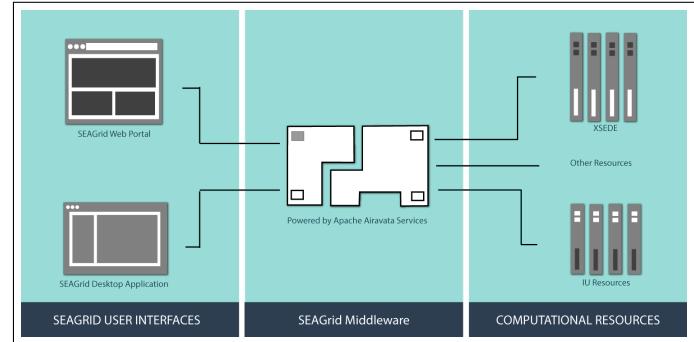


Fig. 2. The image above depicts an example of the technological ecosystem developed around Airavata [5].

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.

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 Software Foundation, "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] Apache Software Foundation, "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] Apache Software Foundation, "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] Apache Software Foundation, "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, April 12, 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>

1. 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].

2. 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 indexed 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].

ley), 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].

		Follows			
		gwashington	jadams	tjefferson	wmckinley
Row Key	gwashington		1		
	jadams	1		1	
tjefferson	1		1		1
wmckinley				1	

Multiple versions

Fig. 1. Bigtable Data Model [5]

3. 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].

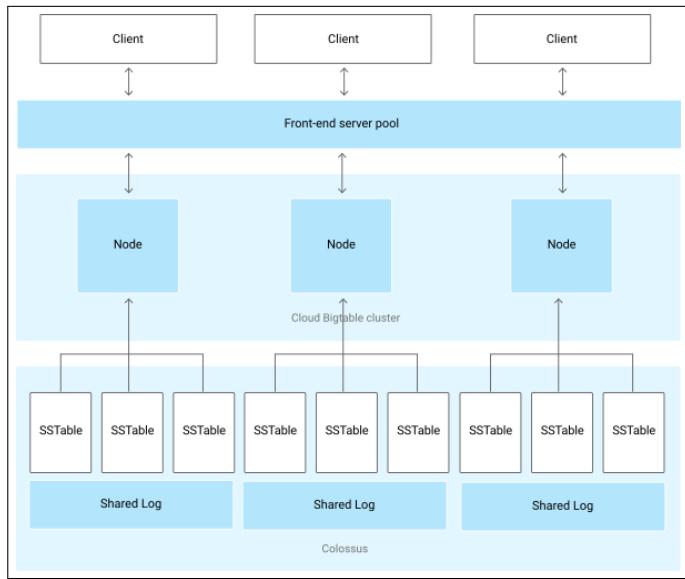


Fig. 2. Bigtable Architecture [5]

3.1. 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].

3.2. 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].

3.3. 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].

4. 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 Table 1.

Table 1. Cloud Bigtable Cost Structure [11]

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

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.

5. 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].

6. 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].

7. RELATED TECHNOLOGIES

7.1. 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.

7.2. 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 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.

8. USE CASES

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

8.1. 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].

8.2. 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].

9. 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].

10. CONCLUSION

As 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] Google, "Apis & reference," Web Page, accessed 2017-02-20. [Online]. Available: <https://cloud.google.com/bigtable/docs/apis>
- [7] Google, "Installing 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] Google, "Bigquery web ui," Web Page, accessed 2017-02-21. [Online]. Available: <https://cloud.google.com/bigquery/bigquery-web-ui>
- [11] Google, "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] Wikipedia, "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] Google, "cloud-bigtable-examples," Code Repository, accessed 2017-02-24. [Online]. Available: <https://github.com/GoogleCloudPlatform/cloud-bigtable-examples>

A. WORK BREAKDOWN

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

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.

Apache Beam (Google Cloud Dataflow)

LEONARD MWANGI¹

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

* Corresponding authors: lmwangi@iu.edu

April 12, 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. Apache Beam previously Google Cloud Dataflow, a unified model solves these problems by providing 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>

1. 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].

2. IMPLEMENTATION

Cloud Dataflow is language agnostic, its 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.

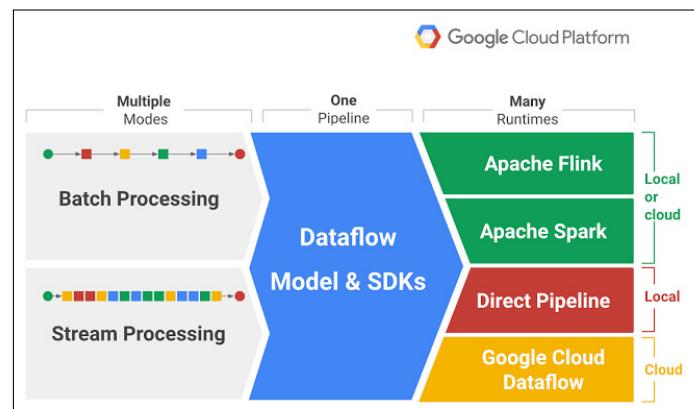


Fig. 1. Dataflow Execution Engine [9]

Figure 1 shows an image of Dataflow Execution Engine.

When programming 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.

2.1. 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

2.2. Dataflow SDK

As mentioned in introduction section, 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.

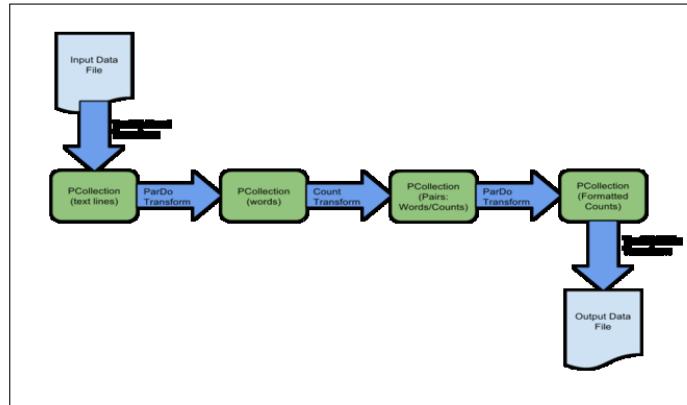


Fig. 2. Apache Beam dataflow [14]

Figure 2 shows an image of Beam dataflow.

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

```
# ... 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)))
p.run()
```

3. 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 [15].

4. USE CASES

4.1. 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.

4.2. 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 [16].

4.3. 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.

5. ALTERNATIVE TECHNOLOGIES

5.1. Amazon Kinesis Stream

Amazon Kinesis Stream is an AWS data streaming offering that can capture and analyze data from different sources in real-time [17]. 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 [18].

5.2. 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 [19]. 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.

5.3. 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.

6. 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.

7. 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-google.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] Google, "Wordcount example pipeline," Web Page, 2017. [Online]. Available: <https://cloud.google.com/dataflow/examples/wordcount-example>
- [15] K. Knowles, "Stateful processing with apache beam," Web Page, 2017. [Online]. Available: <https://beam.apache.org/blog/2017/02/13/stateful-processing.html>
- [16] P. Ciciora, "Study: Store layout an important variable for retailers," *NEWS BUREAU*, 2013. [Online]. Available: https://news.illinois.edu/news/13/0124storelayout_yunchuanliu.html
- [17] Amazon. (2017) Amazon kinesis streams. [Online]. Available: <https://aws.amazon.com/kinesisstreams/>
- [18] Google, "Google cloud platform for aws professionals: Big data," Web Page, Google, 2016. [Online]. Available: <https://cloud.google.com/docs/compare/aws/big-data>
- [19] J. Stokes, K. Whitlatch, A. Pasic, A. Jenks, T. Nevil, R. McAllister, P. McKay, S. Swift, and L. Franks, "What is stream analytics?" *Microsoft Azure*, 2017. [Online]. Available: <https://docs.microsoft.com/en-us/azure/stream-analytics/stream-analytics-introduction>
- [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 RAI¹

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

* Corresponding authors: piyurai@iu.edu

+ HID - S17-IO-3014

April 12, 2017

Xen is an open-source baremetal hypervisor. This paper explores its 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>

1. 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 guaranteed memory and disk allocation to each guest VM (virtual machine).

Xen is a baremetal hypervisor and is 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. A special virtual machine is deployed in the control domain called Domain 0. It contains hardware drivers and the toolstack to control the VMs.

2. ARCHITECTURAL OVERVIEW

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]. The hypervisor is the first process to be started after bootloader. It's primary responsibility is to manage CPU, memory and interrupts and does not contain 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.

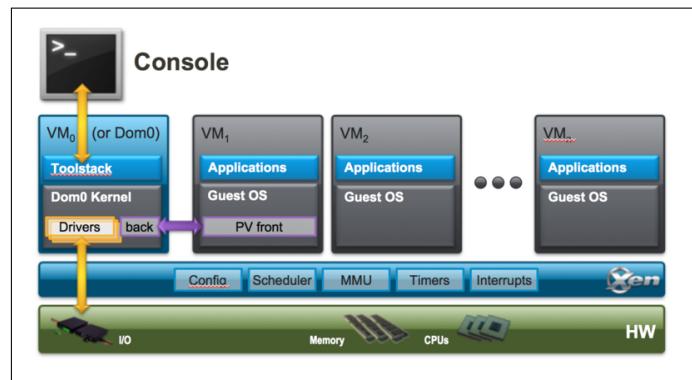


Fig. 1. Xen architecture overview [3].

3. VIRTUALIZATION MODES

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 for a VM 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 their kernel and drivers to be PV-enabled. However, 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, an

open source system emulator, 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, introduced in Xen Project 4.4 where PV guests uses PV drivers for boot and I/O and hardware extensions otherwise. It is meant to reduce the number of interfaces available to guest thereby reducing the attack surface. It is expected to simplify the Xen architecture and is the recommended virtualization mode to be used with newer Xen releases [4].

4. RESOURCES

[5] presents an overview of the software architecture and host the links to documentation and release notes for the different releases. [6] explains the hypervisor internals and it's important features along with their interfaces.

5. ALTERNATIVES

The early days of Xen development came across different limitations such as modification to the guest kernels and limited Linux support [7]. Into the later years of the development of Xen, it remained based on the older version of Linux kernel as the developers waited for more recent kernels with support for modern hardware [8]. 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 virtualization with new releases containing paravirtualization support for certain devices [9]. 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 its own memory and power management system designed specifically for VMs whereas KVM needs to have support for processes as well.

[10] lists the observations made during comparisons of different virtualization technologies in 2013 using DTrace. It analyzes 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. KVM took fewer steps than Xen in terms of I/O path. Zones, with Linux Containers being its equivalent, turned out to be the preferred choice for virtualization as it had the lowest overhead for not having to run an entire OS for each application.

6. 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) [11]. The overheads in terms of passing data across several layers is being

minimized by use of shared memory transports and buffers [10]. Xen Project also strives to minimize the footprint of the change to Linux kernel. The codebase contains less than 150,000 lines of code [2]. Xen Project hosts a listing 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 [12]. Some of the major cloud hosting service providers using Xen include Amazon and Rackspace. Hypervisors are predicted to become a core aspect of cloud based solutions. Containers are growing in popularity and there's an increasing interest in combination of isolation aspects of hypervisor and container runtime environments [13]. [11] talks 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 wiki," Web Page, Dec. 2016. [Online]. Available: https://wiki.xenproject.org/wiki/Main_Page
- [6] D. Chisnall, *The Definitive Guide to the Xen Hypervisor*, 1st ed., ser. Open Source Software Development Series, A. Robbins, Ed. Prentice Hall, nov 2007, no. ISBN-10: 0-13-358249-3.
- [7] Amit Shah, "Ten years of KVM," Web Page, Nov. 2016. [Online]. Available: <https://lwn.net/Articles/705160/>
- [8] Thorsten Leemhuis, "Rise of KVM," Web Page, Jun. 2011. [Online]. Available: <http://www.h-online.com/open/features/Xen-lets-KVM-overtake-1262171.html>
- [9] "KVM - Wikipedia," Web Page, Feb. 2017. [Online]. Available: https://en.wikipedia.org/wiki/Kernel-based_Virtual_Machine
- [10] 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/>
- [11] "Xen Project 4.7 Feature List," Web Page, Jun. 2016. [Online]. Available: https://wiki.xenproject.org/wiki/Xen_Project_4.7_Feature_List
- [12] Stefano Stabellini, James Bulpin, "Hypervisor in 2017," Web Page, Dec. 2016. [Online]. Available: goo.gl/S3X1oQ
- [13] Stefano Stabellini, "Hypervisor-Based Containers," Web Page, Dec. 2016. [Online]. Available: <https://thenewstack.io/hypervisors-container-era/>

Apache Lucene

Roy CHOWDHURY, 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, April 12, 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>

1. 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.

2. 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.

2.1. 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

3. ARCHITECTURE

Figure 1 explains a basic architecture of Lucene

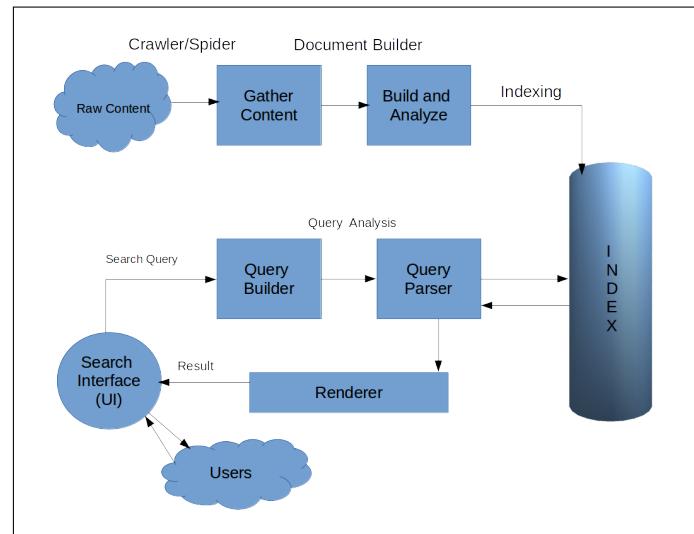


Fig. 1. Apache Lucene Architecture.

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.

4. LUCENE COMPONENTS

4.1. 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.

4.2. 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.

5. 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.

6. 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.

7. 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 can be used on clouds from EC2, Rackspace, GCE or virtual machines such as Vagrant, VMware, OpenStack or on physical servers such as PXE, iPXE, ISO. CoreOS makes use of it's disk partitions to achieve continous updates to the operating system without manual interaction. In CoreOS service discovery is achieved by etcd, applications are run on Docker and process management is achieved by fleet.

© 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/raw/master/paper1/S17-IO-3016/report.pdf>

1. INTRODUCTION

CoreOS also known as CoreOS Container Linux 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 helps 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(such as systemd) and software(such as Python and Java) are put together but in the case of CoreOS Linux only operating system and utilities are bundled together. In CoreOS Linux applications and software(such as Python and Java) are run on containers as shown in Figure 1 [1].

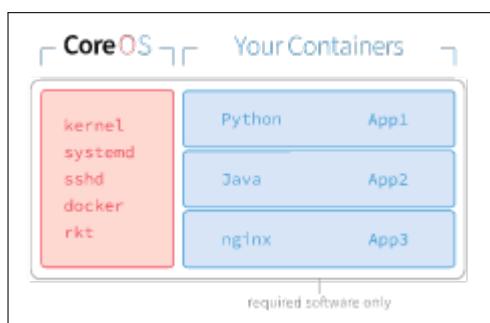


Fig. 1. CoreOS container layout [1].

CoreOS is an open source software that is provided by CoreOS, Inc. CoreOS, Inc also provides open source tools like etcd, rkt and flannel. CoreOS, Inc also has commercial products Kubernetes and CoreOS stack. These products can be used on top of CoreOS for additional functionalities like for example Kubernetes can be used for orchestration and management of containers that run on CoreOS [1].

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. Containers are multiple instances that contain applications and the softwares needed to run these applications sharing the same underlying operating system [2]. Big data projects in science or business involve processing of large amount of data. These big data projects are hosted on bare metal servers which are dedicated physical servers that are only used by one tenant 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].

2. COREOS ARCHITECTURE AND INSTALLATION

CoreOS has 9 disk partitions. CoreOS makes use of these partitions to achieve continous updates to the operating system. Having this disk layout aslo allows CoreOS to be able to reverse the updates in the event of unsuccessful updates. The disk partitions are: 1) EFI-SYSTEM which contains the bootloader and the partition type is VFAT. 2) BIOS-BOOT, ROOT-C and the 7th partition these partitions have no partition format and are reserved for future use. 3) USR-A and USR- B are the active and passive partitions that container linux sits on. Only one of them can be active at a time and partition 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 partition type and it is used for keeping data which is persistent and this partition is stateful [1].

CoreOS Linux comes bundled with etcd, fleet and Docker. In CoreOS Container Linux service discovery is achieved by etcd, applications are run on Docker and process management is achieved by fleet. 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. State and configuration of each container running on CoreOS is stored in etcd key value store and this information is used to identify which container is selected to receive service requests. 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 [3].

Container management in CoreOS linux is made possible by Docker. All applications run on Docker. Containers can be launched using Docker's command line interface [3].

the third component of CoreOS is fleet and it is used for management of containers with Docker installed. fleet is init system. init system is a process that gets started before any other process when a machine boots and runs till the machine is shutdown. fleet has fleetctl a command line interface that can be used to check the status of containers, to start containers and to stop containers [3, 4].

There is no package manager for installing, upgrading, configuring softwares in CoreOS. Softwares are installed as containers in CoreOS Linux operating system. In order 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 partition and upon reboot all the updates are applied to the active partition [5].

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, 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 recommended to have Vagrant version 1.6.3 or latest. Virtual machines by VirtualBox or VMware are both supported by Vagrant [6].

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. At the time of writing is paper, the latest version of CoreOS Container Linux is 1325.1.0 [1].

4. PERFORMANCE

In a study done by Purdue University, 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 native RHEL 6.5 system with performance measurement of 7.839 GFLOPS performed slower than the container environment with CoreOS on Docker with performance measurement of 7.811 GFLOPS. For the network throughput measurement iperf was used. The results for RHEL 6.5 system were 8.26 Gbps for upload throughput and 9.38 Gbps for download throughput. The result for container environment with CoreOS on Docker were 8.43 Gbps for upload throughput and 9.37 Gbps for download throughput. Network File System throughput was done using iozone and results are shown in Figure 2 [2].

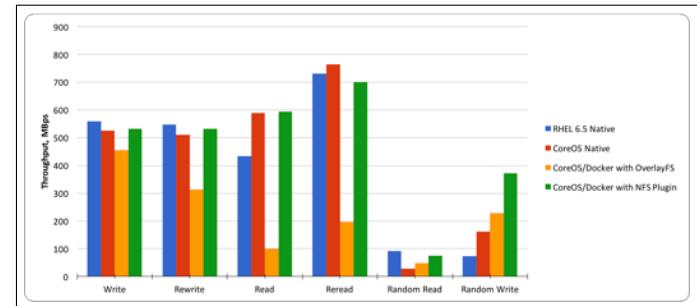


Fig. 2. File Server throughput [2].

In another study done in Institute of Informatics – Federal University of Rio Grande do Sul, performance analysis of ClickOS, CoreOS and OS^v was performed. Network Functions Virtualization(NFV) allows virtualization of network services such as firewalls and DNS. In their paper comparison of ClickOS, CoreOS and OS^v as NFV based tools was performed. They compared 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 consumption, CoreOS has the smallest memory usage followed 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 [7].

5. USE CASES

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

5.1. Use Cases for Big Data

Metagenomics RAST server(MG-RAST) is free access portal that can be used by researchers who study microbes for accessing and analyzing metagenomics data(Random community genomes) [8]. CoreOS Container Linux and fleet are the technologies used on MG-RAST container application servers. Researchers can input data into MG-RAST portal through script, web site or REST API [9].

5.2. Other Use Cases

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 tried to address the problem that exist in most data centers 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 analyzes the relationship between change in workload, resource allocation and response latency in order to calculate the number of containers needed. In this experimental work, CoreOS was used in order 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 [10].

6. EDUCATIONAL MATERIAL

CoreOS website contains detailed documentations and materials for anyone who is interested in setting up CoreOS Container Linux on clouds, virtual or physical servers [3]. Users who are interested can contribute to the CoreOS open source projects through GitHub [11].

The book titled CoreOS Essentials, also has information about CoreOS overview and its installation [5].

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

As previous performance results have shown, CoreOS Linux has the lowest memory usage compared to other operating systems namely ClickOS and OS^v [7].

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] Wikipedia, "init," Web Page, Mar. 2017, accessed: 2017-03-28. [Online]. Available: <https://en.wikipedia.org/wiki/Init>
- [5] R. Mocevicius, *CoreOS Essentials*. Packt Publishing Ltd, Jun. 2015.
- [6] CoreOS, "Vagrant," Web Page, Feb. 2017, accessed: 2017-02-17. [Online]. Available: <https://coreos.com/os/docs/latest/booting-on-vagrant.html>
- [7] 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.
- [8] 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>
- [9] 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>
- [10] 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.
- [11] 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, April 12, 2017

MongoDB is a NoSQL database. Instead of using tables and rows as in relational databases, MongoDB is built on an architecture of collections and documents. Documents comprise sets of key-value pairs and are the basic unit of data in MongoDB. Collections contain sets of documents and function as the equivalent of relational database tables. MongoDB database is used when data size is expected to be huge and schema is not stable.

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

Keywords: I524, MongoDB, NoSQL

<https://github.com/cloudmesh/sp17-i524/raw/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. In NoSQL database like MongoDB data is stored in collections. A collection acts as a container for a 'Document'. A Document is equivalent 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 [1].

2. INFRASTRUCTURE AND PERFORMANCE

Following sections discuss key deployment considerations.

2.1. Working set size

The set of data and indexes that are accessed most frequently during normal operations is called the working set. Working set resides in memory rather than on disk to ensure low latency database operations. Ideally working set should fit into RAM. Before MongoDB deployment it is necessary to assess the working set size based on the size of data and kind of operations that will be performed on data, so that adequate RAM size can be determined. Page faults occur in case working set size exceeds the available RAM. In this case either the RAM size should be increased or sharding should be done (see Section 2.4).

2.2. Storage and Disk I/O

If working set size is far larger than any available memory, then selecting the proper disk type for deployment is important. Local disks should be used as far as possible as the network storage can cause high latency and poor performance.

2.3. CPU Selection

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].

2.4. Sharding

MongoDB partitions data across servers using a technique called Sharding. Balancing of data across shards is automatic, and shards can be added and removed without taking the database offline [3]. Sharding allows the database to scale out on hardware deployed on-premises or in the cloud, enabling almost unlimited growth with higher throughput and lower latency than relational databases [4]. Figure 1 shows scaling of database.

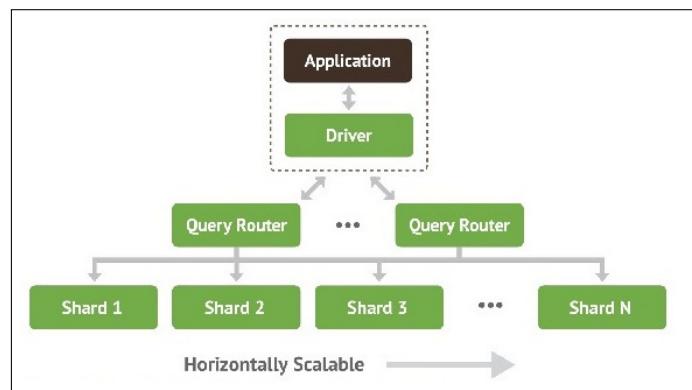


Fig. 1. MongoDB Auto-sharding [2].

Users should consider deploying a sharded MongoDB cluster in the following situations [2]:

RAM Limitation: If the working set size is expected to exceed the capacity of the maximum amount of RAM in the system.

Disk I/O Limitation: If the system has a large amount of write activity, and the operating system cannot write data fast enough to meet demand.

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

2.5. Replication

MongoDB maintains multiple copies of data called replica sets using native replication. A replica set consists of multiple replicas. At any given time, one member acts as the primary replica set member and the other members act as secondary replica set members. Reads and writes are issued to a primary copy of the data. If the primary member fails for any reason (e.g., hardware failure, network partition), one of the secondary members is automatically elected to primary and begins to process all writes [5].

Figure 2 shows replica sets maintained by MongoDB.

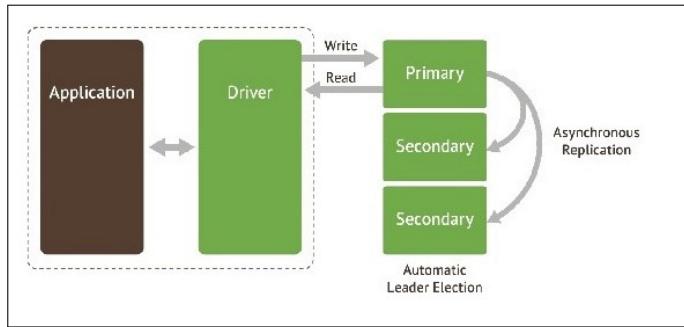


Fig. 2. MongoDB replica sets [2].

2.6. Indexing

Indexes are a crucial mechanism for optimizing system performance and scalability. MongoDB includes support for many types of secondary indexes that can be declared on any field in the document, including fields within arrays:

One can define compound, unique, array, partial, TTL, geospatial, sparse, hash and text indexes to optimize for multiple query patterns, multi-structured data types and constraints.

Index intersection enables MongoDB to use more than one index to optimize an ad-hoc query at run-time [4].

3. FEATURES

These are main features of MongoDB.[6]

- General purpose database, almost as fast as the key:value NoSQL type.
- 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: MMS tracks the database and hardware metrics for managing MongoDB deployment. Performance is visualized in a rich web console to help optimize your deployment. Discover issues via custom alerts before MongoDB instance will be affected.
- Backup: Continuous backup with point-in-time recovery of replica sets and sharded clusters. 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 - Volume, Variety and Velocity. MongoDB supports storage of high volume of data which is complex in nature. Its dynamic schema provides a major advantage for businesses that need to ingest, store, and process rapidly evolving data streams from new sources. The term velocity refers to high and volatile inbound data, faster query at low latency. MongoDB is designed to support insertion of high volume of data in less time and faster response to query [7].

For data analysis one can utilise built-in aggregation functionality provided by MongoDB. In case of more complex data analytics one can leverage Hadoop and Apache Spark framework. To be able to connect efficiently to MongoDB Hadoop and Spark connectors are available and can be configured easily. 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.

5. MONGODB USE CASES

Some of the common MongoDB use cases involve operational intelligence, Internet of Things, content management and real-time analytics. Following are some of the MongoDB customers [8].

BOSCH: The Bosch IoT Suite is a cloud-enabled software package for developing applications in the Internet of Things (IoT). Bosch has built its Internet of Things suite on MongoDB. MongoDB has been used to manage massive volume and unstructured nature of IoT data and perform real-time analytics.

Aadhar: India's Unique Identification project, Aadhar, maintains biometrics database. Aadhar is in the process of capturing demographic and biometric data of over 1.2 billion residents. Aadhar has used MongoDB as one of its databases to store this huge amount of data.

MetLife: MetLife uses MongoDB for its customer service application called 'The Wall'. It provides a consolidated view of MetLife customers, including policy details and transactions.

eBay: eBay has a number of projects running on MongoDB for search suggestions, metadata storage, cloud management and merchandizing categorization.

6. LICENSING

MongoDB database server and tools are available under GNU AGPL v3.0.

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>

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

The comparison shows that MongoDB supports the maximum operating systems. Secondary indexes makes it easy to query in MongoDB. MongoDB is a step ahead in case of ease of use, as it is easy to program in JSON.

REFERENCES

- [1] MongoDB, Inc, "Mongodb: Bringing online big data to business intelligence & analytics," PDF on Web, Jun 2016. [Online]. Available: http://s3.amazonaws.com/info-mongodb-com/MongoDB_BI_Analytics.pdf
- [2] 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>

Table 1. MongoDB vs. Cassandra Vs. HBase [9].

Name	Cassandra	HBase	MongoDB	
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 RESTful API, 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 (ACL)	Con-Lists	Access rights for users and roles

- [3] MongoDB, Inc, "Capacity planning and hardware provisioning for mongodb in ten minutes," Web Page. [Online]. Available: <https://www.mongodb.com/blog/post/capacity-planning-and-hardware-provisioning-mongodb-ten-minutes>
- [4] MongoDB, Inc, "Mongodb architecture," Web Page. [Online]. Available: <https://www.mongodb.com/mongodb-architecture>
- [5] IBM Business Partner Network, "Mongodb architecture explained," Web Page, Jun 2016. [Online]. Available: <https://www.ibmpn.com/linux-blog/mongodb-architecture>
- [6] Cesar Trigo Esteban, "Mongodb: Characteristics and future," Web Page, Aug 2014. [Online]. Available: <http://www.mongodbSpain.com/en/2014/08/17/mongodb-characteristics-future/>
- [7] MongoDB, Inc, "Real-time analytics," Web Page. [Online]. Available: <https://www.mongodb.com/use-cases/real-time-analytics>
- [8] Sudha Gopinath, "Real world use cases of mongodb," Web Page, Jan 2014. [Online]. Available: <https://www.edureka.co/blog/real-world-use-cases-of-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 SMITH¹

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

*Corresponding authors: mls35@iu.edu

Paper 1, April 12, 2017

vSphere and vCloud are categorized as infrastructure as a service(IaaS) offering. vSphere was developed by VMware and is a cloud computing virtualization platform[1]. vSphere is not one piece of software but a suite of tools 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 the vsphere suite, but also contains site recovery management for disaster recovery, site networking and security. These services as well as IaaS in general are discussed. Advantages of utilizing such as service are presented followed by an analysis of vSphere and vCloud. 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/cloudmesh/classes/blob/master/docs/source/format/report/report.pdf>

1. CLOUD COMPUTING

With the rise of high speed internet, connectivity amongst devices locally 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 started to see the potential benefits of moving to the cloud, one of the first companies to do so was salesforce.com, which delivered applications to end users over the internet [3]. Today there are many companies in the business of cloud computing. The majority of business coming from Microsoft, Google, and Amazon. There are subdivisions of cloud computing 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 resources over the internet such as amazon web services, Microsoft azure, and VMware. It is important to note that many providers offer more than just one type of cloud computing option[4].

IaaS is a business model in where a provider will host all components of infrastructure including hardware, software, servers, data storage for their clients. The cost to the client is at a rate of usage which can range from different rates such as per hour, week, or month. While it might seem costly to pay for such as

service, utilization of IaaS provide several advantages for organizations to use such as service. Management does not have the upfront costs their own IT infrastructure as well as maintaining hardware or replacing equipment that is obsolete. The responsibility of ensuring the network is up and running through having a qualified team of IT staff is also mitigated. The cost of IaaS is metered thus 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. 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]. It is both costly and impractical to staff IT personnel to monitor demands twenty four hours a day seven days a week. The IaaS has the capacity to scale with depends 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 is potential is limited to very little or none[5].

2. VCLOUD AND VSPPHERE

vCloud falls into the cloud computing subcategory IaaS. It is a suite of multiple products that include vSphere, vCloud Direc-

tor, 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 also known as a virtual machine manager is 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” [7]. The type 1 refers to a type of hypervisor that can run directly on the hosts pc to control its resources.

vCloud director is a tool for overall cloud management it helps the user build hybrid clouds through pooling resources into data centers. This product helps empower existing IT within an organization 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[8]. 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 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”[9]. This is 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.

3. LICENSING

Depending on the demand, the cost of vCloud can be quite expensive. It is important to note that there are currently 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[10]. Other services generally offer limited time services followed by a pay requirement.

4. 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 Hadoop 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 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 extensions is installed on vSphere 5.0, all abilities of the administrator can only be performed within the command-line[11].

5. VCLOUD API

The vCloud application program interface (API) enables the client and the director to communicate via HTTP through an XML exchange[12]. This type of workflow is referred to as a RESTful workflow. REST is defined as Representational State Transfer, which “describes an architectural style characteristic of programs that use the HTTP to exchange serialized representations of objects between a client and a server”[12]. These transfers are executed by a script to pertaining to the acronym CRUD which is create, retrieve, update or delete an object defined by the API.

The vCloud API is responsible for the definition of objects that are normally found in a cloud computing environment such as organizations, users/groups, catalogs, virtual data center or VDC, virtual systems and images, and tasks[12]. An organization is a combination of users, groups, and computing resources. The user must undergo a verification through an identity provider in vcloud director or externally. This verification is done so by using an exchange of data in security assertion markup language or SAML. The catalog will possess all vApp templates and media images configurable in a private local repository, or published content accessible publicly. The vds is the environment for virtual systems dictating allocation of resources such as storage, CPU, and memory.

As previously stated, the exchange of representations on the API are done so through the XML documents. The following attributes within the XML document are id, type, and href[12]. The id attribute is a unique identifier that is not reused and is context free. The type is a multipurpose internet mail extension(MIME). The href is the object reference which is a URL expression. This URL will also includes the id attribute thus it will not be reused. Along with attributes, elements pertaining to a specific object and its creation or modification are detailed in the document as well.

6. 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 scalability, 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] technopedia, "Hypervisor," Webpage. [Online]. Available: <https://www.techopedia.com/definition/4790/hypervisor>
- [8] 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>
- [9] M. Rouse, "Cloud disaster recovery," Webpage. [Online]. Available: <http://searchcloudstorage.techtarget.com/definition/cloud-disaster-recovery-cloud-DR>
- [10] amazon, "Aws free tier," Webpage. [Online]. Available: https://aws.amazon.com/s/dm/optimization/server-side-test/free-tier/free_np/
- [11] 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>
- [12] vmware, "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, April 12, 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.

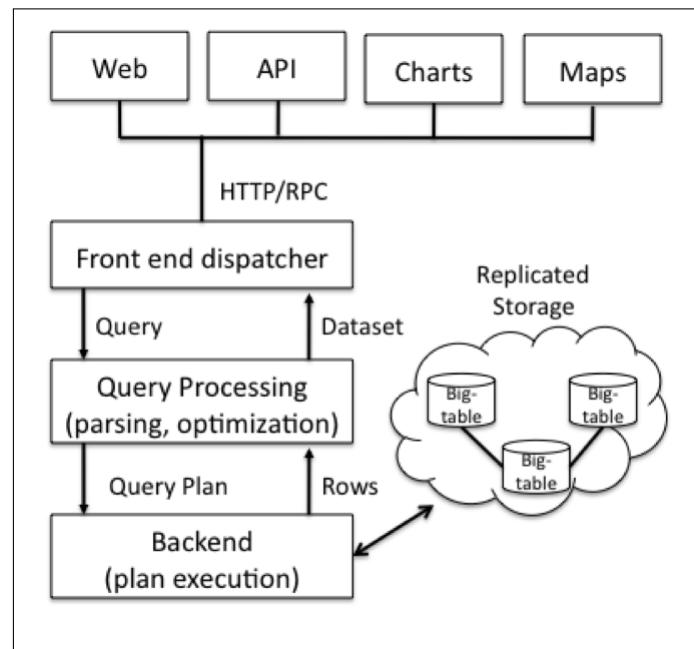


Fig. 1. Fusion Table Architecture

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 from 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 uselesss 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 has 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.

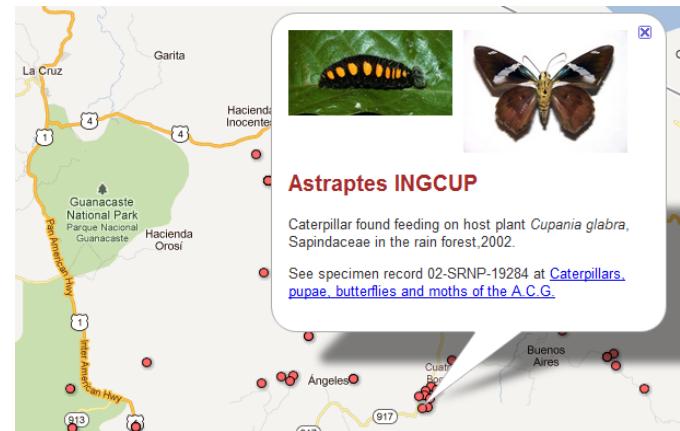


Fig. 3. Map service integration using Fusion Table [8]

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

6. LATEST FEATURES

- Controlled sharing of data, user can share the specific data which should be shared with other user.
- 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/1711181?hl=en>
- [3] Google Developers, “Type and size of files to import,” Web Page, Feb. 2017. [Online]. Available: https://en.wikipedia.org/wiki/Google_Fusion_Tables
- [4] Google Developers, “Fusion tables Rest API: Get Start,” Web Page, Feb. 2017. [Online]. Available: https://developers.google.com/fusiontables/docs/v2/getting_started
- [5] Google Developers, “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 THAKRE¹

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

²Mechanical Engineer,Nagpur University, 2003

* Corresponding authors: abhijit.thakre@gmail.com

project-000, April 12, 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/sp17-i524/blob/master/paper1/S17-IO-3021/report.pdf>

1. 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.

2. ARCHITECTURE

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

2.1. Database Server

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.

2.2. 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.

2.3. 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.

3. ARCHITECTURE DIAGRAM

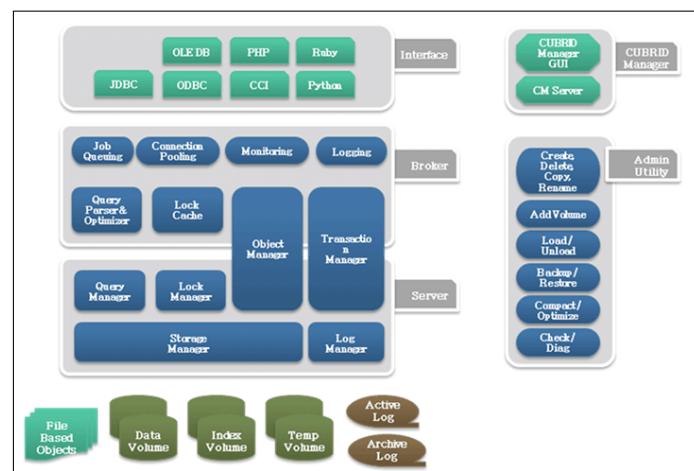


Fig. 1. [1]

4. DESCRIPTION

Database server and the broker work in co-ordination with each other as server and client respectively. They can be deployed on the different or same machine. The broker takes care of the queries from the users, it processes them 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]

5. 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.

6. 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.

References: [2]

7. KEY FEATURES

7.1. 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.

7.2. Database Sharding

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

7.3. Performance

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

CUBRID performance for hotspot read and special character for web service using logging technique have applied to disk which has resulted in increase in the overall performance.

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

7.4. Reliability

CUBRID is highly reliable with features like online incremental backup and restore. It provides the access restriction based on userip and databaseid. Naver is one of the top web portal in Korea rely on cubrid database.

7.5. Language Support

It provides 90 percent support to sql language support.

8. 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 servers 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 need the additional layer. This helps in increasing the performance for big data.

9. PERFORMANCE BENCHMARKING

9.1. CUBRID Vs MySQL

Test was performed by one of the social networking site on new 8.4.0 version of cubrid database to understand the time taken by cubrid database for managing the queries of 6 million active users. The queries mostly involved the IN and UNION operator.

For executing the test both Mysql and CUBRID database was made to store 3 user group table with 1520000 rows each. Each of the db had given 40 thread load for 10 min. The systems used for both the database are of same configuration. The test was conducted for five different scenarios.

9.2. Analysis Results

In most realistic scenario CUBRID IN query have performed twice as fast as Mysql UNION query and 8 times as fast as the Mysql IN query.

It also came out from the test that the effect of Key limit have considerably decreased the IO operations.

9.3. NBD BenchMark

The NBD (NHN Internet Bulletin Board Application Database) Benchmark is important benchmark to measure of performance of a DBMS used for a Bulletin Board System (BBS) type of web services.

This benchmark test was conducted to compare the performance of CUBRID R1.1 DBMS with OSS DBMS D1, Commercial DBMS D1, and Commercial DBMS D2. Factors considered for the benchmark includes database size, workload increase and cache functionality.

9.4. Conclusion of NBD Benchmark

In the small and medium-sized database benchmarking, CUBRID is at second place after commercial DBMS D1. In the small database benchmarking, CUBRID performance can be improved if provided more work load. For the given workload it showed the usage of 30% CPU.

For medium size however it showed 80% of CPU usage. DBMS D2 shows the highest CPU usage of 100% in both small and medium-sized databases. References: [3]

REFERENCES

- [1] "CUBRID RDBMS," Web Page, Indiana University, Feb. 2017. [Online]. Available: <http://www.cubrid.org>
- [2] "CUBRID RDBMS," Web Page, Indiana University, Feb. 2017. [Online]. Available: <http://www.methodsandtools.com/tools/cubriddatabase.php>
- [3] "CUBRID RDBMS," Web Page, Indiana University, Feb. 2017. [Online]. Available: http://www.cubrid.org/cubrid_mysql_sns_benchmark_test

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.

* Corresponding authors: suunni@indiana.edu HID: S17-IO-3022

project-000, April 12, 2017

Performance of realtime analytics on Big Data depends on efficiency of inter and intra process communication. There are many messaging frameworks supported by HPC-ABDS like Netty, ZeroMQ, Kyro, ActiveMQ, LMAX disruptor. Performance of messaging framework is measured by throughput of the messaging framework. We are comparing the throughput of Netty with ZeroMQ based on research done by IBM and Yahoo. © 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/sp17-i524/raw/master/paper1/S17-IO-3022/report.pdf>

1. 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.

2. 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.

2.1. Worker Process Illustration

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

2.2. Running Topology

A running topology as illustrated in Fig. 2.

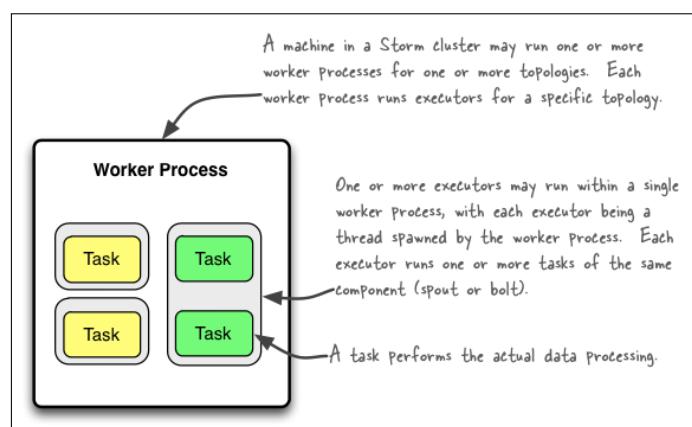


Fig. 1. The relationships of worker processes, executors (threads) and tasks in Storm, adopted from [3]

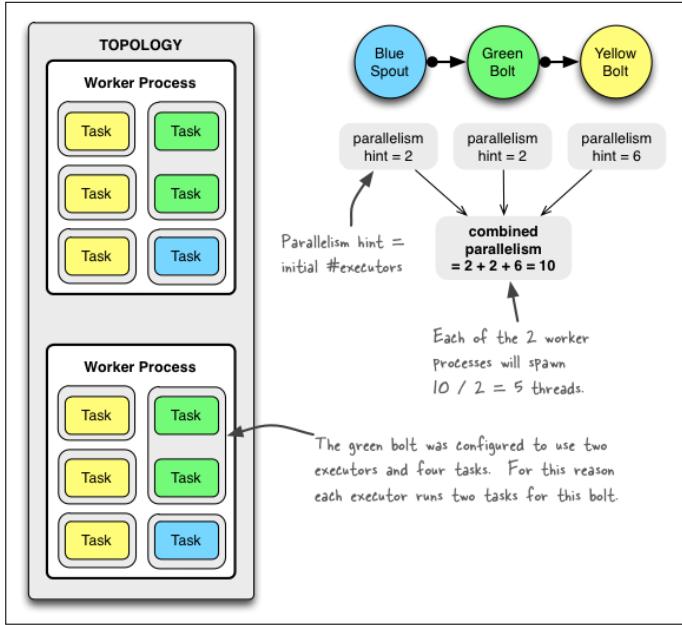


Fig. 2. Example of a Running Topology, adopted from [3]

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.

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
- 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.

3. 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.

3.1. 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.

3.2. Netty Behavior with small 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.

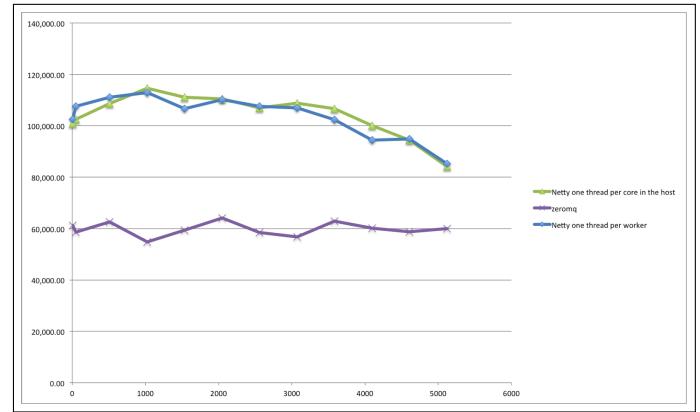


Fig. 3. Small Topology- Netty vs ZeroMQ, adopted from [6]

3.3. 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.

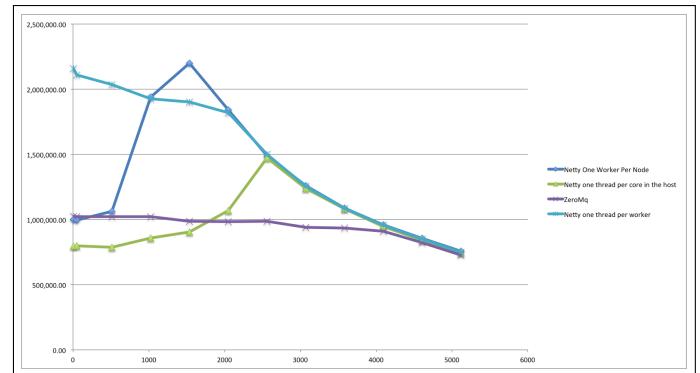


Fig. 4. Large Topology- Netty vs ZeroMQ, adopted from [6]

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.

- [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://yahoobeng.tumblr.com/post/64758709722/making-storm-fly-with-netty>

4. 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.

4.1. 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.

4.2. 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]

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>

H-Store

KARTHICK VENKATESAN^{1,*,+}

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

* Corresponding authors: vkarthickprabu@gmail.com

+ HID - S17-IO-3023

paper1, April 12, 2017

"NewSQL" denotes a class of modern (RDBMS) relational database management systems that seek to provide the same scalable performance of NoSQL systems for online transaction processing (OLTP) read-write workloads while still maintaining the ACID guarantees of a traditional database system. H-Store is one the first NewSQL system which is a complete redesign of the traditional RDMS database. © 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

Keywords: H-Store, OLTP, I524

<https://github.com/cloudmesh/sp17-i524/tree/master/paper1/S17-IO-3023/report.pdf>

1. INTRODUCTION

H-Store is an experimental database management system (DBMS) designed for online transaction processing applications that was developed by a team of researchers at Brown University, Carnegie Mellon University, the Massachusetts Institute of Technology, and Yale University.

Traditional SQL databases offer rich transaction capabilities, ad-hoc query and reporting capability, and vast amounts of standards-based tooling. Where they fall short is in the ability to scale out to meet the needs of modern, high-performance applications. NoSQL Databases are a solution to issues of scale that emerged as organizations began dealing with immense volumes, velocity, and variety of big data from a multitude of sources. NoSQL solutions offered availability, flexibility and scale. However, they came with difficult tradeoffs - sacrifice of data consistency and transactional (ACID) guarantees, loss of ad-hoc query capability and increased application complexity. By removing just the parts of SQL that hampered scalability and performance, it was able to increase both while maintaining key advantages of SQL systems. NewSQL technologies offer the best of both worlds: the scale of NoSQL with the ACID guarantees, strong consistency, minimized application complexity and interactivity of SQL. The H-Store DBMS belongs to this class of parallel database management systems, called NewSQL.

The current RDBMS systems were architected more than 25 years ago, when hardware characteristics were much different than today. Today processors are thousands of times faster and memories are thousands of times larger. Disk volumes have increased enormously, making it possible to keep essentially everything, if one chooses to in memory [1]. The current day DBMS include the following architectural features which are not fully utilising the technological advancements in the last 25 years .

- Disk oriented storage and indexing structures
- Multithreading to hide latency
- Locking-based concurrency control mechanisms
- Log-based recovery

H-Store is a complete redesign of the traditional RDMS database designed to fully leverage the advancements in hardware capabilities over the last 25 years and experimental results show that the OLTP processing on a H-Store database is faster by a factor of 82 [1].

2. SYSTEM ARCHITECTURE

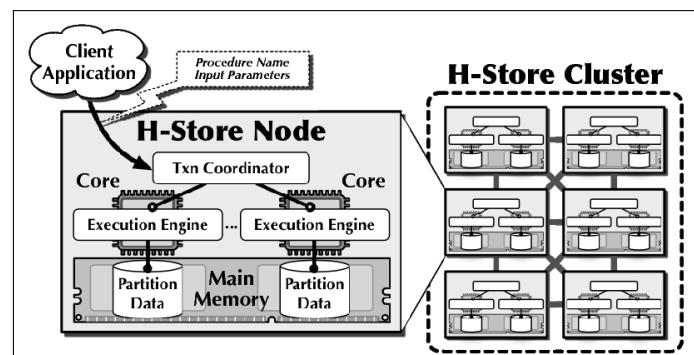


Fig. 1. H-Store Architecture
[2]

A single H-Store instance as in Figure 1 consists of a cluster of two or more computational nodes deployed within the same domain [2].

- A node is a single physical computer system that hosts one or more execution sites and one transaction coordinator.
- A site is the logical operational entity in the system; it is a single-threaded execution engine that manages some portion of the database and is responsible for executing transactions on behalf of its transaction coordinator.
- The transaction coordinator is responsible for ensuring the serializability of transactions along with the coordinators located at other nodes.
- A typical H-Store node contains one or more multi-core CPUs, each with multiple hardware threads per core. As such, multiple sites are assigned to nodes independently and do not share any data structures or memory.

3. STORAGE ARCHITECTURE

Every table in H-Store is horizontally divided into multiple shards that each consist of a disjoint sub-section of the entire table. The boundaries of a table's fragments are based on the selection of a column for that table; each tuple is assigned to a fragment based on the hash values of this attribute. H-Store also supports partitioning tables on multiple columns. Related fragments from multiple tables are combined together into a partition that is distinct from all other partitions. Each partition is assigned to exactly one site.

All tuples in H-Store are stored in main memory on each node; the system never needs to access a disk in order to execute a query. It replicates partitions to ensure both data durability and availability in the event of a node failure. Data replication in H-Store occurs in two ways: (1) replicating partitions on multiple nodes and (2) replicating an entire table in all partitions. For the former, H-Store adopts the k-safety concept, where k is defined by the administrator as the number of node failures a database can tolerate before it is deemed unavailable [2].

4. TRANSACTION PROCESSING

Client applications make calls to the H-Store system to execute pre-defined stored procedures. Each procedure is identified by a unique name and consists of user-written Java control code intermixed with parameterized SQL commands. The input parameters to the stored procedure can be scalar or array values, and queries can be executed multiple times.

Each instance of a stored procedure executing in response to a client request is called a transaction. Similarly as with tables, stored procedures in H-Store are assigned a partitioning attribute of one or more input parameters. When a new request arrives at a node, the transaction coordinator hashes the value of the procedure's partitioning parameter and routes the transaction request to the site with the same id as the hash. Once the request arrives at the proper site, the system invokes the procedure's Java control code, which will use the H-Store API to queue one or more queries in a batch. The control code invokes another command in H-Store to block the transaction while the execution engine executes the batched queries.

All operations within a transaction are atomic across all the partitions involved in the transaction. When a transaction executes, it has exclusive access to the data at the partitions that it locks, therefore all of its operations execute on a consistent view of the database and are isolated from all other transactions (since no other transactions execute concurrently at those partitions).

Finally, with snapshots and command logging, the state of the database is completely recoverable and all changes made by transactions are durable [2].

5. DOCUMENTATION

- Detailed documentation on H-Store Deployment , Configuration, Debugging and Development is available at [3]
- H-Store DBMS is open source and complete source code is available in the [4] github repository.

6. COMMERCIAL USE

A commercial implementation of H-Store is VoltDB. VoltDB was developed for production environments, and thus it is focused on high-performance throughput for single-partition transactions and provides robust handling of failures that are needed for a main memory system. H-Store uses VoltDB's VoltProcedure API, so any stored procedures written for VoltDB work in H-Store [5].

7. RELATED TECHNOLOGIES-OTHER IN MEMORY NEW SQL DATABASES

- **VoltDB:** VoltDB is an in-memory, scale-out SQL database purpose-built to power a new generation of applications that thrive on fast, smart data. Tapping the lightning speed and real-time analytics of VoltDB, organizations are able to add context and intelligence to data – the instant it arrives – to make real-time transactional decisions that maximize business value [6].
- **Clustrix:** Clustrix provides the leading scale-out SQL database engineered for the cloud and the database built specially to meet the unique growth, performance and availability demands of today's e-commerce sites. ClustrixDB, helps to build business critical applications that support massive transactional volume and realtime reporting of business performance metrics. ClustrixDB delivers more than one trillion transactions per month for customers [7].
- **NuoDB:** NuoDB is a scale-out SQL database for the cloud and the modern datacenter. It is the NewSQL solution designed to meet the needs of cloud-scale apps and over 50 billion connected devices in use worldwide [8].
- **MemSQL:** MemSQL is a leader in real-time databases for transactions and analytics. As a purpose built database for instant access to real-time and historical data, MemSQL uses a familiar SQL interface and a horizontally scalable distributed architecture that runs on commodity hardware or in the cloud [9].

8. COMPARISON OF NEW SQL DATABASES

A comparison of other NewSQL database features with H-Store is provided in Figure 2.

9. USE CASE

In Memory NewSQL databases such as H-Store are ideally suited for handling use cases related to Fast Data [11].

	Year Released	Main Memory Storage	Partitioning	Concurrency Control	Replication	Summary
Clustrix [6]	2006	No	Yes	MVCC+2PL	Strong+Passive	MySQL-compatible DBMS that supports shared-nothing, distributed execution.
CockroachDB[7]	2014	No	Yes	MVCC	Strong+Passive	Built on top of distributed key/value store. Uses softwares hybrid clocks for WAN replication.
Google Spanner[24]	2012	No	Yes	MVCC+2PL	Strong+Passive	WAN-replicated, shared-nothing DBMS that uses special hardware for timestamp generation.
H-Store[8]	2007	Yes	Yes	TO	Strong+Passive	Single-threaded execution engines per partition. Optimized for stored procedures.
Hy-Per[9]	2010	Yes	Yes	MVCC	Strong+Passive	HTAP DBMS that uses query compilation and memory efficient indexes.
MemSQL[11]	2012	Yes	Yes	MVCC	Strong+Passive	Distributed, shared-nothing DBMS using compiled queries. Supports MySQL wire protocol.
NuoDB[14]	2013	Yes	Yes	MVCC	Strong+Passive	Split architecture with multiple in-memory executor nodes and a single shared storage node.
SAPHANA[55]	2010	Yes	Yes	MVCC	Strong+Passive	Hybrid storage(rows + cols).Amalgamation of previous TRELX, P*TIME, and MaxDB systems.
VoltDB[17]	2008	Yes	Yes	TO	Strong+Active	Single-threaded execution engines per partition.Supports streaming operators.

Fig. 2. NewSQL DB Comparison

[10]

- Real-time recommendation engines or hyper-personalization applications that can detect and act on individual customer needs in real-time. Emagine International is an innovative company that provides a real-time, adaptive contextual marketing platform and managed marketing services for mobile service providers. To differentiate its offering and help operators take advantage of real-time analytics and decisioning, Emagine implemented H-Store's Commercial implementation VoltDB as the core of its Emagine Real-time Event Decisioning (ERED) platform. This has allowed Emagine to architect a fast data solution that requires 3 milliseconds for the ingest-analyze-decide journey through the ERED platform, enabling Emagine to deliver customized offers to subscribers in fewer than 250 milliseconds. Mobile operators that use the ERED platform achieved a measurable ROI in real-time personalization [12].

- Real-time, "down to the last dollar" resource management applications. More than 150,000 apps and the world's leading advertisers rely on Airpush, a mobile advertising company, to deliver the industry's highest performance, driven by exceptional ad formats and targeting technology. The company originally built a MySQL infrastructure to manage advertising transactions. Airpush's business grew exponentially, pushing the limits of MySQL. To ensure the system would be able to process transactions fast enough to align advertising purchases with customer budgets, Airpush began to look into alternatives. Airpush selected VoltDB (H-Store commercial implementation), which offered the performance of in-memory, the scalability of NoSQL, and the transactional consistency of traditional relational databases. With VoltDB, Airpush built a transactional, database-oriented application against data feeds that were previously limited to stream processing methods because of scale. Supporting hundreds of thousands of concurrent connections with round-trip latencies in milliseconds, VoltDB is an ideal platform for high-speed policy enforcement, authorization, rule evaluation, and quota management [13].

10. CONCLUSION

H-Store and its commercial implementation VoltDB are main memory DBMS designed for applications running on a cluster of nodes needing high transaction throughput. They are ideally suited for the Fast Data use case of Big Data. As an example of a "NewSQL" DBMS with superior performance, they are ideally positioned to take over the Modern Transaction Processing market from legacy disk-oriented RDBMS.

ACKNOWLEDGEMENTS

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

REFERENCES

- [1] M. Stonebraker, S. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland, "The end of an architectural era: (it's time for a complete rewrite)," in *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007, pp. 1150–1160. [Online]. Available: <http://hstore.cs.brown.edu/papers/hstore-endofera.pdf>
- [2] Brown University and Carnegie Mellon University and Massachusetts Institute of Technology and Yale University, "H-Store-Arch," Web Page, accessed 2017-02-19. [Online]. Available: <http://hstore.cs.brown.edu/documentation/architecture-overview/>
- [3] Brown University and Carnegie Mellon University and Massachusetts Institute of Technology and Yale University, "H-store documentation," Web Page, Dec. 2012, accessed 2017-02-19. [Online]. Available: <http://hstore.cs.brown.edu/documentation/>
- [4] Brown University and Carnegie Mellon University and Massachusetts Institute of Technology and Yale University, "H-store," Code Repository, Sep. 2010, accessed: 2017-2-19. [Online]. Available: <https://github.com/apavlo/h-store>
- [5] Brown University and Carnegie Mellon University and Massachusetts Institute of Technology and Yale University, "H-Store-Faq," Web Page, accessed 2017-02-19. [Online]. Available: <http://hstore.cs.brown.edu/documentation/faq>
- [6] VoltDB, Inc, "VoltDB," Web Page, accessed 2017-02-19. [Online]. Available: <http://voltdb.com/>
- [7] Clustrix, "ClustrixDB," Web Page, accessed 2017-02-19. [Online]. Available: <http://www.clustrix.com/summary-of-our-db/>
- [8] NuoDB, Inc, "NuoDB," Web Page, accessed 2017-02-19. [Online]. Available: <https://www.nuodb.com/product-overview>

- [9] MemSQL Inc, "MemSQL," Web Page, accessed 2017-02-19. [Online]. Available: <http://www.memsql.com/product/>
- [10] A. Pavlo and M. Aslett, "What's really new with newsql?" *SIGMOD Rec.*, vol. 45, no. 2, pp. 45–55, Sep 2016. [Online]. Available: <http://db.cs.cmu.edu/papers/2016/pavlo-newsq-sigmodrec2016.pdf>
- [11] TechTarget, "fast data," Web Page, accessed 2017-02-19. [Online]. Available: <http://whatis.techtarget.com/definition/fast-data>
- [12] VoltDB, Inc, "Emagine implements realtime telecommunications personalization with fast data," VoltDB, Inc, Case Study, 2015, accessed 2017-02-19. [Online]. Available: https://www.voltdb.com/hubfs/content/case_studies/hv_case_studies/hv-case-study-emagine-implements-real-time-telecommunications-presonalization-with-fast-data.pdf
- [13] VoltDB, Inc, "Airpush," VoltDB, Inc, Case Study, 2015, accessed 2017-02-19. [Online]. Available: https://www.voltdb.com/hubfs/content/case_studies/hv_case_studies/hv-case-study-airpush-on-a-mission-to-redefine-mobile-advertising.pdf

LAT_EX Linux Containers - Virtualization for Cloud Era

ASHOK VUPPADA¹

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

* Corresponding authors: ashokmadhu66@gmail.com

paper-1, April 12, 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 Hypervisor 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 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]

Figure below shows the comparison between the hypervisor based VM and container from the architecture.[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

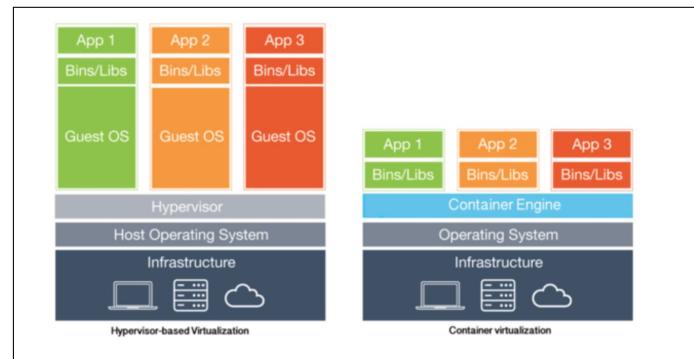


Fig. 1. Architecture Comparison of Hypervisor Based VM and containers.

jail cannot act on the file system outside it. This is used to isolate the unstable applications to hamper the file system in use.[3]

- namespaces: namespaces provides the process level isolation for the global resources. It helps the process in each of the namespace to believe it's the only process.[2]
- cgroups: it's a kernel feature which allows to allocate resources to a given group of process(s). It helps in 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]

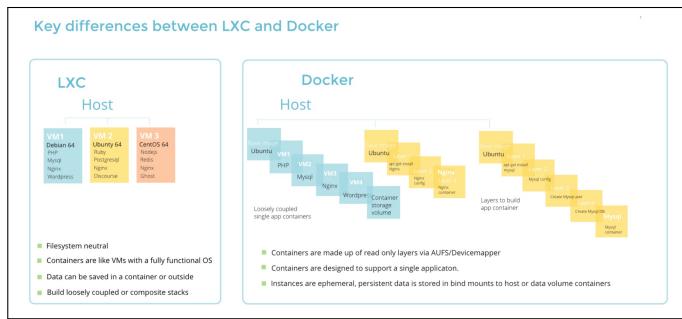


Fig. 2. Architecture Comparison of LXC and Docker.

4. PERFORMANCE COMPARISON

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. [8]

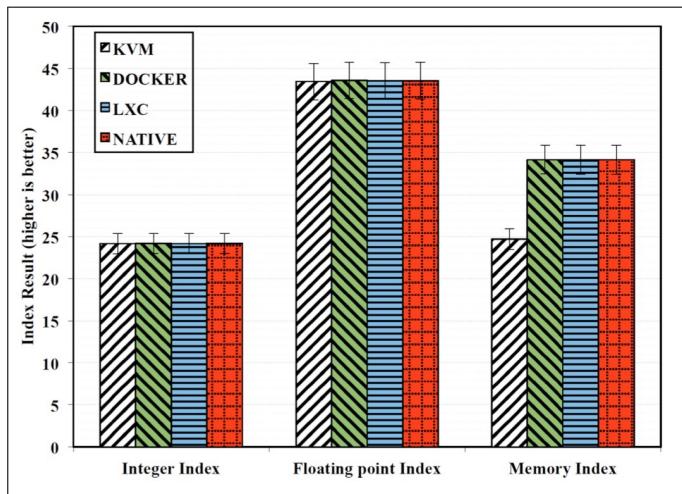


Fig. 3. Single Core Perf Comparison

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 [8]

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 [8]

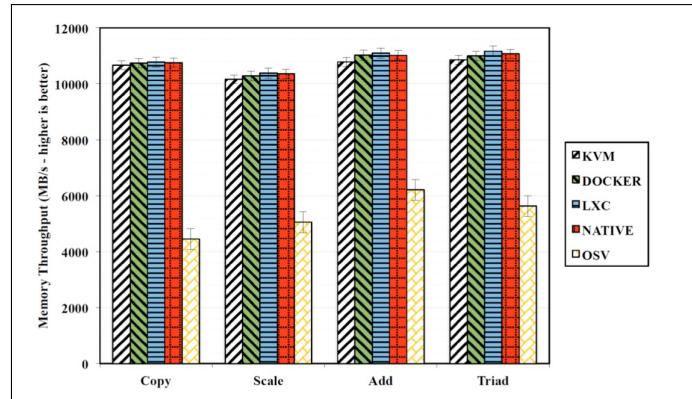


Fig. 5. Memory Perf Comparison

3. Disk I/O Performance

Bonnie++ is used to measure the disk performance and below are the results. [8]

Platform	Random Write speed (Kb/s)		Random Seek	
	Native	%	Native	%
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 [8]

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

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 tremendously. 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

- 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]
- 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 propagate to other containers. [10]
- Containers are presently getting evolved, compared to the existing traditional VMs the tools available for monitoring the resources on containers are limited. [10]
- Containers are build on the kernel features build in Linux, Hence this cannot be directly used by other operating systems which don'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 mandatory 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/2b55067c06b5667cf35b3b20096b9235cb4.pdf>
- [9] "Use cases for containers," 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, April 12, 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>

1. 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]

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]

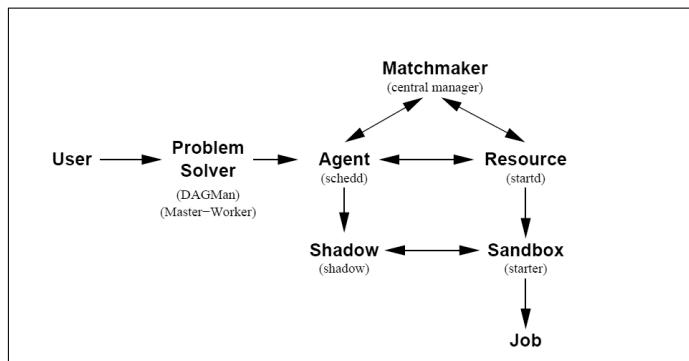


Fig. 1. The HTCondor Kernel

3. THE HTCONDOR SOFTWARE

HTCondor project supports a variety of computing systems deployed worldwide for various commercial and academic purposes. 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

3.1. 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 opportunistic 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. Opportunistic computing is to use resources whenever they are available. High-throughput computing can be efficiently achieved through opportunistic computing. Therefore, HTCondor brings together high-throughput computing as well as opportunistic computing

To achieve high-throughput computing through opportunistic 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.** 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 where it left off at the time of checkpointing. HTCondor also allow 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 environment 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 clusters. HTCondor therefore can be used to combine all of organization’s computing power into a single resource.

3.2. 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].

4. 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. Administrators uses it customize the scheduling 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].

5. 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

6. 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-realistic 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 and 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 because of its complexity. This problem is solved by four mathematicians from Argonne 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 of Condor'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].

7. 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].

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].

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, administrators and the growing demand in mind. HTCondor continues to standout from other batch management systems with the power of high-throughput computing and opportunistic 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, April 12, 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>

1. 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].

2. 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 [2] — which lays theoretical foundation and proves correctness of algorithms for filtering and aggregation operators, which take advantage of the above encoding.

3. DESIGN

Dremel 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 in merely tens of seconds. This is the power of Dremel; it has high scalability and most of the time it returns results within seconds or tens of seconds no matter how big the queried dataset is. These two core technologies gives Dremel high performance in analyzing large datasets:

1. Columnar Storage. Data is stored in a columnar storage fashion which makes possible to achieve high compression ratio and scan throughput.
2. Tree Architecture is used for dispatching queries and aggregating results across thousands of machines. [1].

3.1. 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 in the Figure-1

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 in the Figure-2:

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 in the Figure-3:

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:

- it is generally efficient in term of compression on columns because entropy within a column is lower than 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 big difference especially when the column has few distinct values.
- 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-warehousing 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].

3.2. 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. Figure- 4 shows a serving three execution model. 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 [4]. 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 fast speed [5]. Consider a simple aggregation query below:

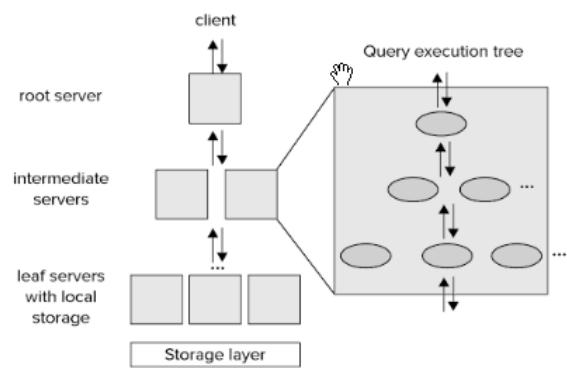


Fig. 4. Typical Tree Architecture [6]

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 [6].

Overall, Dremel combines parallel query execution with the columnar format that supporting performance data access and also capable of operating on in place nested data. In place refers to the ability to process and analyze large, fast-moving datasets where it resides e.g., in a distributed file system like Veritas Cluster File System, General Parallel File System (GPFS), and Global File System (GFS).

4. 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 [7]. 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 [8]. Drill is an Apache top-level project. Open source project must pass Apache incubator process which is the gateway for open source project intended to become top-level or fully fledged Apache Software Foundation (ASF) projects. 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.

5. CONCLUSION

Although query and process large volume of data in any system are a challenging task, especially in the big data ecosystem due to vast expense of option available, Dremel has shown 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, Dremel's design could be the future of interactive ad-hoc query system for analysis that require fast result.

6. 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] A. Y. Zomaya and S. Sakr, *Handbook of Big Data Technologies*. Cham Switzerland: Springer Interntional Publishing, 2017.
- [5] 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>
- [6] B. Lublinsky, K. T. Smith, and A. Yakubovich, *Professional Hadoop Solutions*. Indianapolis, Indiana: Wiley Publishing, 2013.
- [7] Radar Orelilly, "Big data, interactive access: How apache drill makes it easy," Web Page, Feb. 2017, accessed: 2017-03-24. [Online]. Available: <http://radar.oreilly.com/2015/07/big-data-real-time-access-how-apache-drill-makes-it-easy.html>
- [8] 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-2004¹

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

*Corresponding authors: abalaga@iu.edu, ajit.balaga@gmail.com

project-000, April 12, 2017

Apache Samza is a stream processing framework which enables users to analyze streaming data with ease. The concepts required for 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>

1. INTRODUCTION

Messaging systems are a method of implementing real-time 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]

2. 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.

3. STREAMING WITH KAFKA

Kafka is a distributed pub/sub and message queuing 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.

4. 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]

5. RELATED CONCEPTS

5.1. 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]

5.2. 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]

5.3. 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]

5.4. 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]

5.5. 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]

5.6. 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]

6. 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() ...
```

7. 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. Haghghi, "An evaluation of data stream processing systems for data driven applications," *Procedia Computer Science*, vol. 80, pp. 439–449, 2016.

An Overview of Apache Spark

SNEHAL CEMBURKAR¹ AND RAHUL RAGHATATE¹

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

*Corresponding authors: snehchem@iu.edu, rragnate@iu.edu

paper-1, April 12, 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, Spark SQL, MLlib, GraphX, I524

<https://github.com/cloudmesh/sp17-i524/raw/master/paper2/S17-IO-2006/report.pdf>

1. INTRODUCTION

Spark is an open source, easy-to-use distributed cluster computing engine for processing the different types of data available these days. Spark provides a generalized framework which can efficiently process MapReduce jobs (batch processing) as well as iterative algorithms, interactive data mining and streaming analytics. Iterative algorithms include many machine learning algorithms which iterate over the same dataset to optimize a parameter. Interactive data mining refers to executing ad-hoc queries to explore the dataset.

MapReduce can process iterative algorithms by splitting each iteration into a separate MapReduce job. Each job must then read data from a stable storage and write it back to a stable storage at each intermediate step. This repeated access to the stable storage systems like physical disks or HDFS increases the processing time while reducing efficiency of the system. For interactive data mining in Hadoop, the data is loaded in memory across a cluster, and queried repeatedly. Each query is executed as separate MapReduce job which reads data from the HDFS or hard drives thus incurring significant latency (tens of seconds) [2]. Spark is specialized to make data analysis faster in terms of data write speed as well as program execution. It supports in-memory computations which enables faster data querying compared to disk-based systems like Hadoop.

Spark is implemented in Scala, a high level programming language that runs on JVM. It makes programming easier by providing a clean and concise API for Scala, Java and Python [2]. Spark also provides libraries that allow for iterative, interactive, streaming and graph processing. Spark SQL library provides for interactive data mining in Spark. MLlib provides Spark with machine learning algorithms required for iterative computations. Similarly Spark streaming and GraphX libraries enables Spark to process real-time and graph processing data respectively. These

high level components required for processing the diverse workloads such as structured or streaming data are powered by the Spark Core. The distribution, scheduling and monitoring of clusters is done by the Spark Core.

The Spark Core and it's higher level libraries 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.

2. 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. RDDs are a collection of read-only Java or Python objects parallelized across a cluster. DAGs, as the name suggests are directed graphs with no cycles. The libraries or packages supporting the diverse workloads, built on top of the Spark Core, include Spark SQL, Spark Streaming, MLlib (machine learning library) and GraphX. The Spark Core runs atop cluster managers which are covered in section 4.

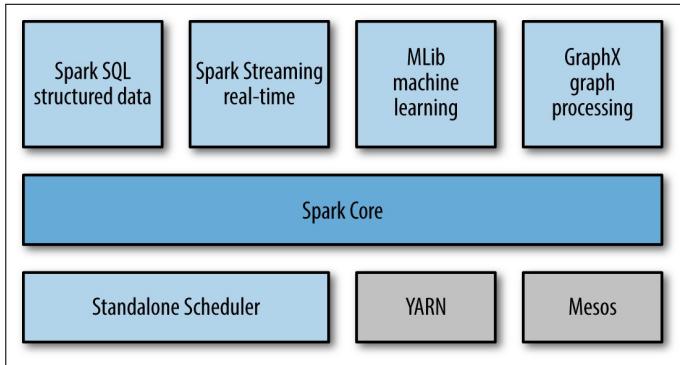


Fig. 1. Spark Components [4].

2.1. Spark Core

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

2.2. 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.

- *Transformations:* Since RDDs are immutable, the transformations return a new RDD and not a single value. RDDs are lazily evaluated i.e they are not computed immediately when a transformation command is given. They wait till an action is received to execute the commands. This is called Lazy evaluation. Examples of transformation functions are map, filter, ReduceByKey, FlatMap and GroupByKey [1].
- *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].

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. The transformations used to build a data set are logged and can be used to rebuild the original data set through its lineage [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.

2.3. Directed Acyclic Graph(DAG)

Directed Acyclic Graph(DAG), which supports acyclic 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 [5]." 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.

2.4. Spark SQL

Spark SQL [4] 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. Spark SQL provides developers with a seamless intermix of relational and procedural API, rather than having to choose between the two. It provides a DataFrame API that enables relational operations on both the in-built collections as well as external data sources. Spark SQL also provides a novel optimizer called Catalyst, to support the different data sources and algorithms found in big data [6].

2.5. Spark Streaming

Spark Streaming [4] 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's API for manipulating data streams is very similar to the Spark Core's RDD API. This similarity makes it easier for users to move between projects with stored and real-time data as the learning curve is short. Spark Streaming is designed to provide the same level of fault tolerance, throughput and scalability as the Spark Core.

2.6. MLlib

MLlib [4] is a rich library of machine learning algorithms for, which can be accessed from Java, Scala as well as Python. It provides Spark with various 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. It also provides basic utilities for feature extractions, optimizations and statistical analysis to name a few [7].

2.7. GraphX

GraphX is a graph processing framework built on top of Spark. It 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].

3. 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 whereas the 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 [4].

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 executors 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

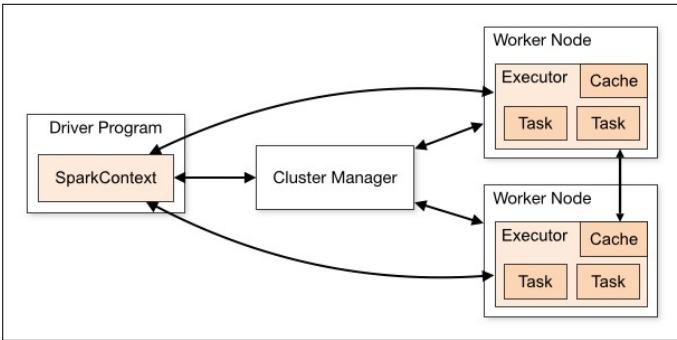


Fig. 2. Spark Architecture [8].

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.

4. DEPLOYMENT MODES

Spark can be deployed in local and clustering modes. The local mode of Spark runs a single node. In clustering mode, Spark can connect to any of the following cluster managers - standalone cluster manager, YARN or Mesos - explained in the following sections.

4.1. Standalone

Standalone cluster manager is the built-in cluster manager provided by spark in its default distribution. To run your Spark application in a standalone clustered environment, make sure that Spark must be installed on all nodes in the cluster. Once Spark is available on all nodes, follow the steps given in Spark documentation to start the master and workers. The master server has a web UI which is located at <http://localhost:8080> by default. This UI will give information regarding the number of CPUs and memory allocated to each node. The Standalone Master acts as the resource manager and allocates resources to the Spark application based on the number of cores. Spark standalone mode is not very popular in production environments due to reliability issues [9].

4.2. YARN

YARN (Yet Another Resource Negotiator) is the resource manager in Hadoop ecosystem. Like the Standalone cluster master, the YARN ResourceManager decides which applications get to run executor processes, where they run it and when they run it. The YARN NodeManager is the slave service that runs on every node and runs the executor processes. This service also helps monitor the resource consumption at each node. YARN is the only cluster manager for Spark that provides security support [9]. To run Spark on YARN, Spark distribution with YARN support must be downloaded from the Apache Spark download page.

4.3. Apache Mesos

Apache Mesos is an open source distributed systems kernel, using principles similar to the Linux kernel but at a different level of abstraction. This kernel provides Spark with APIs for resource management and scheduling across the cluster and runs on each node in the cluster. While scheduling tasks, Mesos considers the other frameworks that may coexist on the same

cluster. The advantage of deploying Spark with Apache Mesos include dynamic partitioning between Spark and other frameworks and scalable partitioning between multiple instances of Spark. Installation of Mesos for Spark is similar to its installation for use by any other frameworks. You can either download Mesos release from its source or from the binaries provided by third party projects like Mesosphere [9].

5. EASY INSTALLATION USING PRE-BUILT PACKAGES

To run Spark on your Windows, Linux or Mac systems you need to have Java 7+ installed on your system. To verify if you have java installed on your Linux machine type `java -version` command in the terminal. If you do not have Java installed on your system you can download it from the Oracle website [10]. The environment variable PATH or JAVA_HOME must be set to point to the Java installation. Now that the Java prerequisite is satisfied, go to the download page of Apache Spark, select the 2.1.0 (latest version as on 26th March 2017) version of Spark, select the pre-built Hadoop 2.7 package and download the `spark-2.1.0-bin-hadoop2.7.tgz` file. Then, go to the terminal and change the directory folder to where the file is located and execute the following command to unzip the file

```
tar -xvf spark-2.1.0-bin-hadoop2.7.tgz -C /
```

This will create a folder `spark-2.1.0-bin-hadoop2.7` in that directory. Move this folder to the `/usr/local/spark` using command `mv spark-2.1.0-bin-hadoop2.7 /usr/local/spark`. To set the environment variable for Spark, open the `.bashrc` file using command `sudo nano .bashrc` and add the following lines at the end of this file.

```
export SPARK_HOME = /usr/local/spark
export PATH = $PATH:
$SPARK_HOME/bin
```

Go back to your home directory and execute `. bashrc` using command `. . bashrc` for the changes to take effect. This change can be verified by executing command `echo $PATH`. The PATH variable should now reflect the path to the spark installation. To verify that Spark is installed correctly, execute command `$spark-shell` in the terminal. It will display the Spark version and then enter the Scala prompt.

6. BUILDING SPARK BASED APPLICATIONS

The first step to start building Spark based applications is exploring the data in Spark Local Mode and developing a prototype. Spark local mode runs on a single node and can be used by developers to learn Spark by building a sample application that leverages the functionalities of Spark API. The developer can use Spark Shells like Scala REPL or PySpark to develop a quick prototype. It can then be packaged as a Spark application using Maven or Scala Build Tool(SBT) [11]. The second step involves deploying the Spark application to production. To achieve this, the developer will fine tune the prototype by running it against a larger dataset. This involves running Spark in cluster mode on YARN or Mesos. Thus the prototype application created in the local mode of Spark will now be submitted as a Spark job to the production cluster [11].

7. PERFORMANCE MONITORING TOOLS

Spark provides a web interface to monitor its applications. By default, each `SparkContext` launches a webUI, at port 4040. This UI displays the memory usage statistics, list of scheduler

stages and tasks, environmental information and information about the executors. This interface can be accessed by opening <http://<driver-node>:4040> in the web browser. If multiple instances of SparkContext are running on the same machine, then they will bind to successive ports beginning with 4040 (4041, 4042, 4043, ...). This information is only available for the life of the application. To view this information after the life of the application, set `spark.eventLog.enabled` to true before starting the application. This will configure Spark to store the event log to persistent storage [12].

A REST API which enables the metrics to be extracted in JSON format, making it easier for developers to create visualizations and monitoring tools for Spark [12]. These metrics can also be extracted as HTTP, JMX, and CSV files by configuring the metric system in the configuration file present at `$SPARK_HOME/conf/metrics.properties`. In addition to these, external tools like Ganglia, dstat, iostat, iotop, jstack, jmap, jmap, and jconsole can also be used to monitor Spark performance [12].

8. USE CASES

In its early days, Spark was adopted in production systems by companies like Yahoo, Conviva, and ClearStory for personalization, analytics, streaming and interactive processing. These use cases are explained in further paragraphs. *Yahoo News Personalization*: This project implements machine learning algorithms on Spark to improve news personalization for their visitors. Spark runs on Hadoop Yarn to use existing data and clusters. In order to achieve personalization, the system will learn about users' interests from their clicks on the web page. It also needs to learn about each news and categorize it. The SparkML algorithm written for this project was 120 lines of Scala code as compared to the 15,000 lines of C++ code used previously [13].

Yahoo Advertisement Analytics: In this project Yahoo leverages Hive on Spark to query and visualize the existing BI analytic data that was stored in Hadoop. Since Hive on Spark (Shark) uses the standard Hive server API, any tools that can be plugged into Hive, will automatically work with Shark. Thus visualization tools like Tableau that are compatible with Hive can be used with Shark to interactively query and view their ad visit data [13].

Monitor Network Conditions in Real-time: Conviva is a video streaming company with a huge video feed database. To ensure quality service, it requires pretty sophisticated technology to be applied behind the scenes to ensure high quality service. With the increase in internet speeds, people's tolerance towards buffering or delays has plummeted. To keep up with the rising expectations of high quality and speed for streaming videos, Conviva implemented Spark Streaming to learn about the network conditions in real-time. This information is then feed to the video player running on the user's laptop to optimize the video speeds [13].

Merge Diverse Data Sources: ClearStory develops data analytics software with speciality in data harmonization. To merge data from internal and external sources for its business users, they turned to Spark, which is one of the core components of their interactive and real-time product [13].

Credit Card Fraud Detection: Using Spark Streaming on Hadoop, banks can detect fraudulent transactions in real-time. The incoming transactions are verified in real-time against a known database of fraudulent transactions. Thus a match against the known database will alert the call center personnel to instantly verify the transaction with the credit card owner.

The authentic transactions are stored to the Hadoop file system where they are used to continuously update the model using deep machine learning techniques [14].

Network Security: Spark can be used to examine network data packets for traces of malicious activity. Spark streaming checks the data packets against known threats and then forwards the unmatched data packets to the storage devices where it is further analyzed using the GraphX and MLlib libraries [14].

Genomic Sequencing: Genomic companies are leveraging the power of Spark to align chemical compounds with genes. Spark has reduced the genome data processing time from a few weeks to a couple of hours [14].

These are few of the real-world use cases of Spark. Real-world applications of Spark that incorporate MongoDB are Content Recommendations, Predictive Modeling, Targeted Ads and Customer Service [15].

9. WHEN NOT TO USE SPARK

Apache Spark is not the most suitable data analysis engine when it comes to processing (1) data streams where latency is the most crucial aspect and (2) when the available memory for processing is restricted. In cases where latency is the most crucial aspect we can get better results using Apache Storm. Since Spark maintains its operations in memory, Hadoop MapReduce should be preferred, when available memory is restricted [16].

10. EDUCATIONAL RESOURCES

The Apache Spark website has a detailed documentation on the how to get started with Spark [17]. It explains the concepts and shows examples to help us familiarize with Spark.

11. LICENSING

Apache Spark is an open-source software licensed under the Apache License 2.0 [18]. Under this license, it is free to download and use this software for personal or commercial purposes. It forbids the use of marks owned by the Apache Software Foundation in a way that might imply that you are the creator of the Apache Software. It requires that you copy the license in any redistribution made by you which includes the Apache Software. You need to provide acknowledgement for any distributions that include the Apache Software [18].

12. CONCLUSION

Apache Spark is an open source cluster computing framework, which has emerged as the next generation big data processing engine surpassing Hadoop MapReduce. Spark facilitates in-memory computations which help execute the diverse workloads efficiently. Its ability to join datasets across various diverse data sources is one of its major attributes. As mentioned in the previous section, Apache Spark is suitable for almost any kind of big data analysis except for the following scenarios: (1) where latency is the most crucial aspect and (2) when the available memory for processing is restricted. Spark finds place in almost all types of big data analysis projects, as seen from the wide range of use cases, due to its core features (RDDs and in-memory computation) and different libraries.

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] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, ser. HotCloud'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 10–10. [Online]. Available: https://www.usenix.org/legacy/event/hotcloud10/tech/full_papers/Zaharia.pdf
- [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] 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/>
- [5] Wikipedia, "Directed acyclic graph - wikipedia," Web Page, dec 2016, accessed: 02-26-2017. [Online]. Available: https://en.wikipedia.org/wiki/Directed_acyclic_graph
- [6] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, and M. Zaharia, "Spark sql: Relational data processing in spark," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '15. New York, NY, USA: ACM, 2015, pp. 1383–1394. [Online]. Available: <http://doi.acm.org/10.1145/2723372.2742797>
- [7] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. 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," *Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, Jan. 2016. [Online]. Available: <http://www.jmlr.org/papers/volume17/15-237/15-237.pdf>
- [8] 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>
- [9] E. Chan, "Configuring and deploying apache spark," Blog, Jul. 2015, accessed: 03-25-2017. [Online]. Available: <https://velvia.github.io/Configuring-Deploying-Spark/>
- [10] Oracle, "Java se development kit 8 - downloads," Web Page, accessed: 03-25-2017. [Online]. Available: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- [11] Hortonworks, "What is apache spark," Web Page, accessed: 03-25-2017. [Online]. Available: https://hortonworks.com/apache/spark/#section_8
- [12] A. S. Foundation, "Monitoring and instrumentation - spark 2.1.0 documentation," Web Page, accessed: 03-22-2017. [Online]. Available: <http://spark.apache.org/docs/latest/monitoring.html>
- [13] A. Woodie, "Apache spark: 3 real-world use cases," Article, Mar. 2014, accessed: 20-Mar-2017. [Online]. Available: https://www.datanami.com/2014/03/06/apache_spark_3_real-world_use_cases/
- [14] P. Kumar, "Apache spark use cases," Article, Sep. 2016, accessed: 20-Mar-2017. [Online]. Available: <https://www.linkedin.com/pulse/apache-spark-use-cases-prateek-kumar>
- [15] mongoDB, "Apache spark use cases," Web Page, accessed: 20-Mar-2017. [Online]. Available: <https://www.mongodb.com/scale/apache-spark-use-cases>
- [16] A. G. Shoro and T. R. Soomro, "Big data analysis: Apache spark perspective," *Global Journal of Computer Science and Technology*, vol. 15, no. 1, p. 9, 2015. [Online]. Available: https://globaljournals.org/GJCST_Volume15/2-Big-Data-Analysis.pdf
- [17] 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>
- [18] A. S. Foundation, "Apache license and distribution faq," Web Page, 2016, accessed: 24-Mar-2017. [Online]. Available: <http://www.apache.org/foundation/license-faq.html>

An overview of HadoopDB and its Architecture

KARTHIK ANBAZHAGAN¹

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

*Corresponding authors: kartanba@iu.edu

April 12, 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 [1].

HadoopDB[2] is an 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.

2. COMPONENTS OF HADOOPDB

A HadoopDB [3] 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 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.

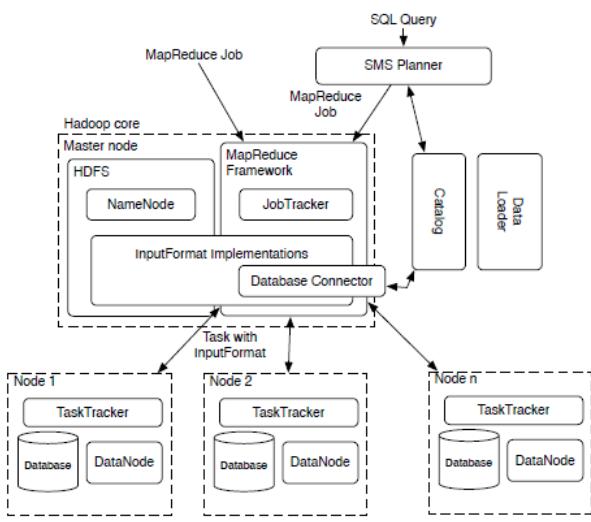


Fig. 1. Architecture of HadoopDB system

3. ARCHITECTURE

Fig 1. shows the architecture[4] 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[5] framework. Hadoop consists of two layers:

- (i) a data storage layer or the Hadoop [6] Distributed File System (HDFS)
- (ii) a data processing layer or the MapReduce Framework.

HDFS[7] 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[8] 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] A. McAfee and E. Brynjolfsson, "Big data: The management revolution," Oct 2012. [Online]. Available: <https://hbr.org/2012/10/big-data-the-management-revolution>
- [2] 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>
- [3] B. Pawlikowski, A. Abouzeid, D. Abadi, and A. Silberschatz, "Hadoopdb project," August 2009. [Online]. Available: db.cs.yale.edu/hadoopdb/hadoopdb.html
- [4] HadoopDB, *HadoopDB Quick Start Guide*, Yale University, 2009 July. [Online]. Available: <http://hadoopdb.sourceforge.net/guide/>
- [5] Wikipedia, "Apache hadoop." [Online]. Available: https://en.wikipedia.org/wiki/Apache_Hadoop
- [6] ApacheFoundation, "Hadoop web page," Web Page, January 2017. [Online]. Available: hadoop.apache.org/core/
- [7] ApacheFoundation, "Hadoop cluster setup," January 2017. [Online]. Available: <https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/ClusterSetup.html>
- [8] ApacheFoundation, "Hive as a contrib project," December 2011. [Online]. Available: issues.apache.org/jira/browse/HADOOP-3601

Ansible

ANURAG KUMAR JAIN¹ AND GREGOR VON LASZEWSKI^{1,*}

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

* Corresponding authors: laszewski@gmail.com

Paper 1, April 12, 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/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2011/report.pdf>

1. INTRODUCTION

Ansible is an open source automation engine which can be used to automate cloud provisioning, configuration management, and application deployment [1]. 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].

2. 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 require any software to be installed on the remote machines to make them manageable. By default it 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 require 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].

3. 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].

4. 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.

5. 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].

6. 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 produce JSON as output [3].

We can also extend Ansible through its 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.

7. 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 generally 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.

8. 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].

9. 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.

10. 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_\(software\)](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, April 12, 2017

We introduce 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. Not only its applications in various fields like HPC and Big Data are explored, but also the areas in which this file system is not recommended are discussed.

© 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/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2012/report.pdf>

1. INTRODUCTION

Lustre is a type of parallel distributed file system. It was started as a research project in 1999 by Peter J. Braam. It 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. The Lustre file system is 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 this file system 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.

2. 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 it 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 sim-

plifies 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].

Followings are the principal foundations of Lustre:

2.1. 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 internally keeps track of allocated objects, which blocks belong to each object, free space, etc. rather than exposing these details to the operating system. This architecture looks at devices that can manipulate file objects. Typical commands executed as a part of the object interface are create, destroy, read/write block X in object N and read/write attributes of objects [4].

2.2. 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. 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.

2.3. 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 the use with non-shared storage devices. An inode file system provides direct access to objects named by an object id. Third are cluster file systems. The traditional cluster file system can become significantly simpler than those implemented with shared block storage devices.

3. 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 each file the number of stripes, stripe size, and OSTs that are used. Striping can be used to improve performance when the aggregate the 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.

4. 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. This 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. It 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.

5. 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.

6. 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]. Although, it is best suited for uses that exceed the capacity that a single server can provide, 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.

7. 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 is 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. Here, we have covered the basic components of Lustre File system, the overview about the installation steps of lustre file system and also a few scenarios in which the use of lustre file system is not recommended. Thus the paradigm of lustre file system is briefly introduced.

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, April 12, 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>

1. 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].

2. 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].

2.1. 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-

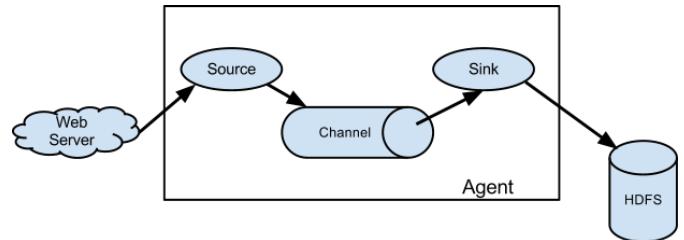


Fig. 1. Architecture of Flume [3]

out flows, contextual routing and fail-over routes for failed hops [4].

2.2. Reliability and recoverability

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 guaranteed 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.

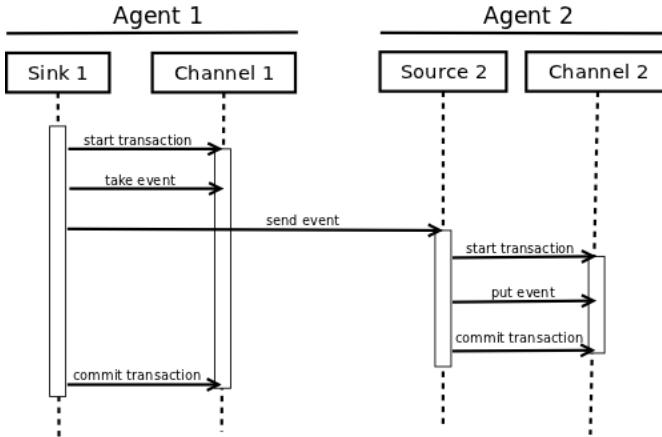


Fig. 2. Transaction Interface of Flume [5]

3. SETUP AND CONFIGURATION

3.1. 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].

3.2. 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]:

```

1 bin/flume-ng agent -n \$agent\_name -c conf
2 -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]:

```

1 # Name the components on this agent
2 a1.sources = r1
3 a1.sinks = k1
4 a1.channels = c1
5
6 # Describe/configure the source
7 a1.sources.r1.type = netcat
8 a1.sources.r1.bind = localhost
9 a1.sources.r1.port = 41114
10
11 # Describe the sink
12 a1.sinks.k1.type = logger
13
14 # Use a channel which buffers events in memory
15 a1.channels.c1.type = memory
16 a1.channels.c1.capacity = 1600
17 a1.channels.c1.transactionCapacity = 100
18
19 # Bind the source and sink to the channel
20 a1.sources.r1.channels = c1

```

Given this configuration file, we can start Flume as follows [3]:

```

1 bin/flume-ng agent --conf conf
2 --conf-file example.conf --name a1
3 -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].

3.3. 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]:

```

1 bin/flume-ng agent --conf
2 conf --conf-file example.conf
3 --name a1 -Dflume.root.logger=DEBUG,
4 console -Dorg.apache.flume.log.printconfig=true
5 -Dorg.apache.flume.log.rawdata=true

```

4. EXPERIMENTAL FEATURE

4.1. Zookeeper 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]:

```

1 - /flume
2   |- /a1 [Agent config file]
3   |- /a2 [Agent config file]

```

Once the configuration file is uploaded, start the agent with following options [3]

```

1 bin/flume-ng agent -conf
2 conf -z zkhost:2181,zkhost1:2181
3 -p /flume -name a1
4 -Dflume.root.logger=INFO,console

```

5. 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].

6. 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].

7. 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.

8. 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] Apache Flume, Web Page. [Online]. Available: <https://flume.apache.org/download.html>
- [7] Apache Flume, 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, April 12, 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>

1. 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]

2. 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].

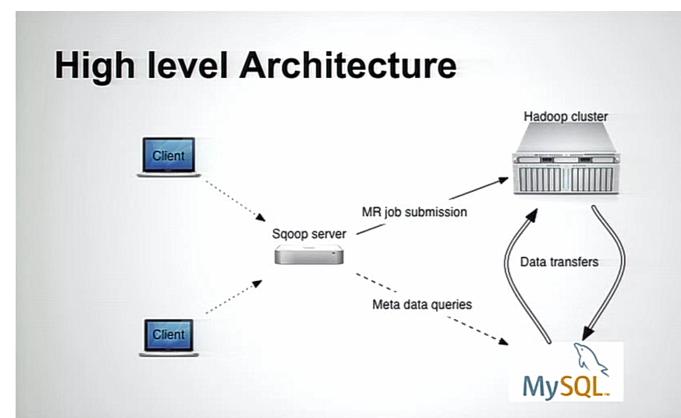


Fig. 1. High Level Architecture of Sqoop [2]

3. COMPONENTS OF SQQOP

3.1. Connector

Connectors are pluggable components that are used to communicate to RDBMS 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 RDBMs software.

3.2. 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.

3.3. 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.

4. WORKFLOW OF SNOOP

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.

Workflow

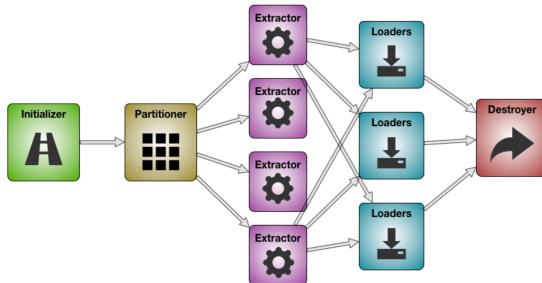


Fig. 2. Workflow Diagram [2]

5. 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 GitHub¹ for developers to fork and modify it.

6. 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].

7. USE CASES

7.1. 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

¹<https://git-wip-us.apache.org/repos/asf?p=sqoop.git;a=summary>

Algorithm 1. MYSQL Import

```

1  sqoop import
2  --connect jdbc:mysql://localhost/acmedb
3  --table ORDERS
4  --username test
5  --password
  
```

Algorithm 2. Hive Import

```

1  sqoop import
2  --connect jdbc:mysql://localhost/acmedb
3  --table ORDERS
4  --username test
5  --password
6  --hive-import
  
```

Algorithm 3. HBase Import

```

1  sqoop import
2  --connect jdbc:mysql://localhost/acmedb
3  --table ORDERS
4  --username test
5  --password
6  --hbase-create-table
7  --hbase-table ORDERS
8  --column-family mysql
  
```

Algorithm 4. Export

```

1  sqoop export
2  --connect jdbc:mysql://localhost/acmedb
3  --table ORDERS
4  --username test
5  --password ****
6  --export-dir /user/asd
  
```

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 fraudulent 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.

7.2. 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.

8. 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.

9. 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 LINGAMPALLI¹

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

*Corresponding authors: anveling@umail.iu.edu

April 12, 2017

MLlib is a machine learning library that runs on top of Apache Spark. With Spark and MLlib, jobs that reference a number of predefined machine learning algorithms are used to build applications.

© 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[1] is an 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. MLlib(Machine Learning Library) is Apache Spark's scalable machine learning library[2]. Spark has many advantages when compared to other technologies such as Hadoop and Storm. Hadoop[3] is a big data processing technology, which is proved to be a solution for processing large data sets. In cases involving machine learning or streaming data, Hadoop is not efficient. It requires other tools such as Mahout[4] or Storm[5] 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[6].

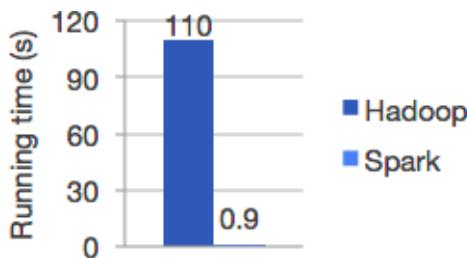


Fig. 1. Hadoop vs Spark

Apache 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[7] (Kraska et al., 2013)[8]. 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[9] and decision trees[10] for classification and regression problems. With the help of these models, problems such as alternating least squares(ALS), k-means problem[11], PCA (principal component analysis)[12] 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[13].

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[14], decision trees and naive Bayes. Multiclass Classification also has decision trees and naive Bayes as its supported algorithms. Regression has linear least squares[15], Lasso[16] 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, Mahout announced it would not accept Hadoop MapReduce and completely switched to Spark.

H2O[17], xgboost[18], python scikit-learn[19] are few other alternatives to MLlib. Scalability, speed and performance are measured for these tools and are shown in the table.

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 time, memory usage, and accuracy are presented. These tests have been carried out on a Amazon EC2 instance (32 cores, 60GB RAM)[20].

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.

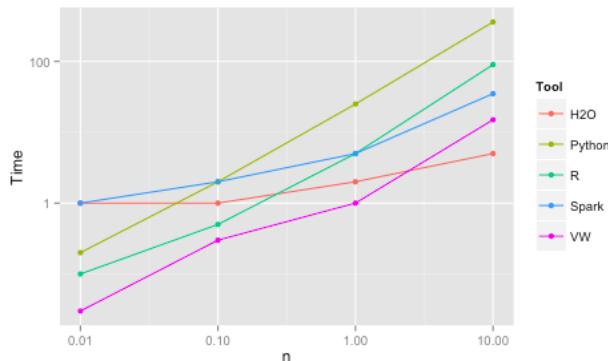


Fig. 3. Graph analysis

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.[21]

5.2. Predict Telco Churn with Apache Spark MLlib

Churn prediction[22], 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.[23]

6. USEFUL RESOURCES

[24] also has some good step by step tutorials on how to use Machine learning library to work on big data analytics involving machine learning learning studio.

7. CONCLUSION

In conclusion, MLlib is a library used to perform machine learning as a part of big data analysis. It is designed for simplicity, scalability, and easy integration with other tools. 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] "Apache spark," webpage. [Online]. Available: <http://spark.apache.org/>
- [2] "Apache spark's mllib," webpage. [Online]. Available: <http://spark.apache.org/mllib/>
- [3] "Apache hadoop," webpage. [Online]. Available: <http://hadoop.apache.org/>
- [4] "Apache mahout," webpage. [Online]. Available: <http://mahout.apache.org/>
- [5] "Storm," webpage. [Online]. Available: <http://storm.apache.org>
- [6] "Hadoop mapreduce," webpage. [Online]. Available: <https://en.wikipedia.org/wiki/MapReduce>
- [7] T. Kraska, A. Talwalkar, J. Duchi, R. Griffith, M. J. Franklin, and M. Jordan, "Mbase: A distributed machine-learning system," in *6th Biennial Conference on Innovative Data Systems Research(CIDR'13)*, Asilomar, CA, USA, 2013. [Online]. Available: http://cidrdb.org/cidr2013/Papers/CIDR13_Paper118.pdf
- [8] "Mbase project paper," webpage. [Online]. Available: http://cidrdb.org/cidr2013/Papers/CIDR13_Paper118.pdf
- [9] "Naive bayes," webpage. [Online]. Available: https://en.wikipedia.org/wiki/Naive_Bayes_classifier
- [10] "Decision trees," webpage. [Online]. Available: https://en.wikipedia.org/wiki/Decision_tree
- [11] "K-means problem," webpage. [Online]. Available: https://en.wikipedia.org/wiki/K-means_clustering
- [12] "Principal component analysis," webpage. [Online]. Available: https://en.wikipedia.org/wiki/Principal_component_analysis
- [13] 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, "Mlib: Machine learning in apache spark," *CoRR*, vol. abs/1505.06807, 2015. [Online]. Available: <http://arxiv.org/abs/1505.06807>
- [14] "Logistic regression," webpage. [Online]. Available: <http://www.statisticssolutions.com/what-is-logistic-regression/>
- [15] "Linear least squares," webpage. [Online]. Available: [https://en.wikipedia.org/wiki/Linear_least_squares_\(mathematics\)](https://en.wikipedia.org/wiki/Linear_least_squares_(mathematics))

- [16] "Lasso statistics," webpage. [Online]. Available: [https://en.wikipedia.org/wiki/Lasso_\(statistics\)](https://en.wikipedia.org/wiki/Lasso_(statistics))
- [17] "H2o," webpage. [Online]. Available: [https://en.wikipedia.org/wiki/H2O_\(software\)](https://en.wikipedia.org/wiki/H2O_(software))
- [18] "xgboost," webpage. [Online]. Available: <https://xgboost.readthedocs.io/>
- [19] "python scikit-learn," webpage. [Online]. Available: <http://scikit-learn.org/stable/>
- [20] "Analysis of various machine learning packages," webpage. [Online]. Available: <https://github.com/szilard/benchm-ml>
- [21] "Movie recommender using mllib," webpage. [Online]. Available: <http://ampcamp.berkeley.edu/big-data-mini-course/movie-recommendation-with-mllib.html>
- [22] "Churn prediction," webpage. [Online]. Available: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-azure-ml-customer-churn-scenario>
- [23] "Prediction of telco churn," webpage. [Online]. Available: <https://blog.cloudera.com/blog/2016/02/how-to-predict-telco-churn-with-apache-spark-mllib/>
- [24] "Guide for mllib," webpage. [Online]. Available: <https://spark.apache.org/docs/latest/ml-guide.html>

An Overview of OpenNebula Project and its Applications

VEERA MARNI¹

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

April 12, 2017

OpenNebula orchestrates storage, network, virtualization, combining both data center resources and remote cloud resources, according to allocation policies. This paper provides insights into how OpenNebula can provide the right cloud services for unique needs of each organization.

© 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>

1. INTRODUCTION

OpenNebula[1] is an open cloud platform using which each organization setup the right cloud for its organizational needs. This quite natural like it had happened with the databases and web-servers. 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 one of the solutions for the management of virtualized data centers with built-in features available for deployments in hybrid clouds.

Its 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. Its platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. Private cloud hosting, on the other hand, by definition is a single-tenant environment where the hardware, storage and network are dedicated to a single client or company.

The public cloud is defined as a multi-tenant environment, where consumers buy cloud services in a cloud computing environment that is shared with a number of other clients or tenants. Private cloud hosting, on the other hand, by definition is a single-tenant environment where the hardware, storage and network are dedicated to a single client or company. Hybrid cloud is mixture of both public and private clouds where one uses both public and private clouds. OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures.

2. ARCHITECTURE OF OPENNEBULA

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.

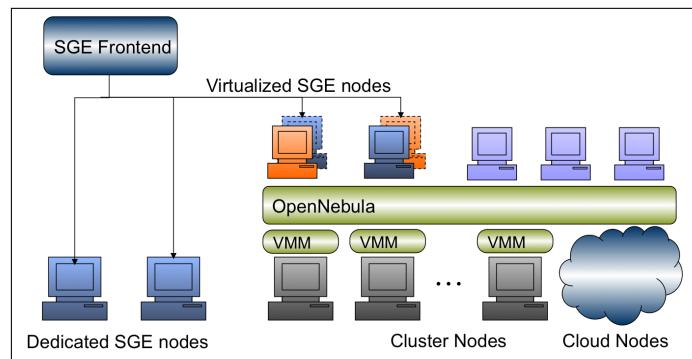


Fig. 1. The OpenNebula Engine for Data Center Virtualization and Cloud Solutions[3].

2.1. Data Center Virtualization Management

Many users use OpenNebula to manage data center virtualization[4], consolidate servers, and integrate existing IT assets for computing, storage, and networking. In this deployment model, OpenNebula directly integrates with virtual machine monitor(VMM) or hyper-visors and has complete control over virtual and physical resources, providing advanced features for capacity management, resource optimization, high availability and business continuity.

2.2. Cloud Management

OpenNebula also provides a multi-tenancy and cloud-like provisioning layer on top of an existing infrastructure management

solution where a single instance of a software application servers multiple customers. Each customer is called a tenant and each tenant may be given ability to customize some parts of the application. 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.

3. INTEGRATION, API'S AND LANGUAGE BINDING

OpenNebula has been designed to be adapted to most infrastructures and be extended with new components. Its interfaces hide most complexities of a cloud by and are suited to meet end user needs.

3.1. System Interfaces

It 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 2 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. 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[5] 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.

3.2. Infrastructure Integration

The interactions between OpenNebula and the Cloud infrastructure[6] are performed by specific drivers. Each one addresses a particular area:

Storage The OpenNebula core 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.

4. ECOSYSTEM

The OpenNebula Ecosystem[7] 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.

5. USE CASES OF OPENNEBULA

5.1. For the Infrastructure Manager

OpenNebula responds 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.

5.2. 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.

5.3. 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.

5.4. Usage in Industry

BIT is a business to business internet service provider in the Netherlands specialized in colocation and managed hosting. It has tested both OpenStack and OpenNebula in a lab environment where they found OpenNebula served them better than Openstack[8]. For more look into user testimonials on OpenNebula webpage[9].

6. KEY FEATURES AND COMPONENTS OPENNEBULA

6.1. Features

There are several key features of OpenNebula[10] 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.

6.2. Components

OpenNebula has several advanced components[11] 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

7. DISADVANTAGES

There are also some drawbacks that needs attention before one choose to use OpenNebula. 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.

8. EDUCATIONAL MATERIAL

A great place to start is by reading the OpenNebula documentation[12]. Future Systems has a tutorial on their webpage[13] to get started with OpenNebula. Handbook of Research on High Performance and Cloud Computing in Scientific Research and Education[14] by Marijana Despotovic-Zrakic is a book to get started from the basics of cloud computing and understanding the working of OpenNebula.

9. CONCULSION

The OpenNebula's architecture and its integration to adapt to most infrastructures make OpenNebula's ability to handle public, private and hybrid clouds with ease. OpenNebula has reduced efforts for handling cloud services 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.

REFERENCES

- [1] "Wikipedia-opennebula," webpage. [Online]. Available: <https://en.wikipedia.org/wiki/OpenNebula>
- [2] "About-technology," webpage. [Online]. Available: <https://opennebula.org/about/technology/>
- [3] "Wikipedia-opennebula," webpage. [Online]. Available: http://blog.dsa-research.org/wp-content/uploads/2008/08/one_amazon1.png
- [4] "Data center virtualization and solutions," webpage. [Online]. Available: <https://opennebula.org/the-opennebula-engine-for-data-center-virtualization-and-cloud-solutions/>
- [5] "Opennebula system interfaces," webpage. [Online]. Available: http://docs.opennebula.org/5.2/integration/system_interfaces/index.html
- [6] "Opennebula infrastructure integration," webpage. [Online]. Available: http://docs.opennebula.org/5.2/integration/infrastructure_integration/index.html
- [7] "Opennebula ecosystem," webpage. [Online]. Available: <https://opennebula.org/community/ecosystem/>
- [8] "Wikipedia-opennebula," webpage. [Online]. Available: <https://opennebula.org/using-opennebula-at-bit/>
- [9] "Wikipedia-opennebula," webpage. [Online]. Available: <https://opennebula.org/users/usertestimonials/>
- [10] "Key features of opennebula," webpage. [Online]. Available: <https://opennebula.org/about/key-features/>
- [11] "Opennebula adavanced components," webpage. [Online]. Available: <http://docs.opennebula.org/5.2/>
- [12] "Opennebula," Web Page. [Online]. Available: <https://opennebula.org/documentation/>
- [13] FutureSystems, "Opennebula online tutorial," Web Page. [Online]. Available: <https://portal.futuresystems.org/tutorials/opennebula>
- [14] 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, April 12, 2017

Pentaho is a 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 and it 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/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2018/report.pdf>

1. INTRODUCTION

Pentaho is 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 but data inconsistency and handling large amount of data are the factors that obstruct them from doing so. Pentaho addresses these obstacles [1]. The platform includes a wide range of tools that analyze, explore, visualize and predict data. It simplifies data blending, which is the process of combining data from multiple sources into a functioning dataset. Being an open and extensible source, Pentaho provides big data tools to extract, prepare and blend any data.

2. PENTaho COMMUNITY

Pentaho provides two different editions, Community edition and Enterprise edition. As the name suggests, the Enterprise edition provides more packages to provide additional 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 people and helps the users in becoming a part of them and benefitting from the open source contributions. 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. This shows the flexibility that the Community edition provides to innovate, find and experiment with the solution that the developers liked.

3. 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 [3]. 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 includes 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.

4. PLUGINS AND APPLICATIONS

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 [4]. 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.

4.1. Server applications

Business Intelligence platform is a basic Pentaho service that provides reports, displays dashboards, reports business rules and performs OLAP analysis. It generally runs in Apache Java application server and can be embedded in any other Java application server [1]. Pentaho analysis service is another Pentaho 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.

4.2. Desktop applications

Pentaho data mining is a desktop application that searches for patterns in data by performing knowledge analysis. Common data mining techniques 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 business models which can be used by other applications in generating reports for the analytics. There are many more useful desktop applications like Pentaho report designer, design studio and aggregate designer.

4.3. Server plugins

Pentaho provides certain core services in the form of 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.

5. 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 [5]. 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. 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 imported [6]. 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. Bottlenecks are eliminated by data conversion logic and character code-page conversion. For encoding a set

of characters, a character set is used along with a set of control characters. Code page is a table of values that describes this character set.

Pentaho Data Integration has a set of components that contribute to its functionalities. They are 'Spoon', 'Kitchen', 'Pan' and 'Carte' [7]. 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. It is used for transformations and jobs with the help of editor and 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.

6. BIG DATA USE CASES

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 enables the data scientists to perform 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. The presence of an adaptive big data layer relieves several organizations from evolving technologies. Customers are given access to customizable and interactive dashboards. Data scientists are provided with predictive analytics and data mining tools.

7. COMPARISON

Pentaho products compete with that of popular IT corporations like 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 fea-

tures with similar costs, Pentaho has got wider online presence and more followers in social media [9].

8. LICENSING

Pentaho Community edition is a free open source product licensed under the General Public License version 2.0 and Mozilla Public License 1.1. The Enterprise edition is available to the users on a commercial license. It needs to be purchased under a subscription model which includes services and support [1].

9. CONCLUSION

Pentaho is an open source based platform for diverse big data deployments. It enables analytics in any environment by delivering data with availability, usability, integrity and security. It provides unified data integration and analytics components which are embeddable. The server plugins and desktop applications provided by Pentaho play a major role in enabling data analytics. Pentaho Data Integration is the Pentaho component responsible for data transformation and loading the data. It helps organizations 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] "Understanding pentaho architecture," Webpage. [Online]. Available: <https://www.edureka.co/blog/understanding-pentaho-architecture/>
- [4] "Pentaho bi suite enterprise edition," Webpage, 2006. [Online]. Available: <http://searchdatamanagement.techtarget.com/review/Pentaho-BI-Suite-Enterprise-Edition>
- [5] M. C. Roldán, "Pentaho data integration (kettle) tutorial," Webpage, 2008. [Online]. Available: [http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+\(Kettle\)+Tutorial](http://wiki.pentaho.com/display/EAI/Pentaho+Data+Integration+(Kettle)+Tutorial)
- [6] 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->
- [7] "Pentaho data integration architecture," Webpage. [Online]. Available: <https://help.pentaho.com/Documentation/5.3/0L0/0Y0/010>
- [8] "What is big data?" Webpage. [Online]. Available: <http://www.pentaho.com/what-is-big-data#tab-3>
- [9] "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: mvasanthiit@gmail.com

Paper-1, April 12, 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 satuate 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>

1. 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].

2. 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

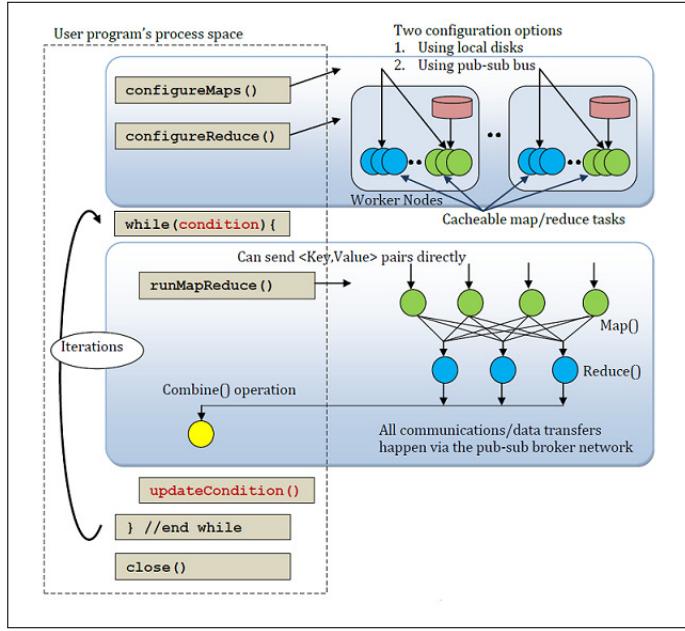


Fig. 1. Twister Programming model.

Java code) 8) Support for typical MapReduce computations 9) Tools to manage data[10][11].

3. TWISTER IMPROVEMENTS OVER MAPREDUCE PROGRAMMING MODEL:

3.1. 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].

3.2. Cacheable 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].

3.3. 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].

3.4. 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].

3.5. 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 or DryadLINQ[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].

3.6. 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].

3.7. Fault Tolerance

Providing fault tolerance support for iterative computations with Twister is currently under development.

4. APPLICATION OF TWISTER TO PRESENT PROBLEMS:

4.1. 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].

4.2. 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].

4.3. 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].

4.4. 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].

4.5. 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 partition. This is the approach used in Twister word count application[10].

4.6. 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].

5. 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 beneficial 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. Gunaratne, 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 and Swarm

SHREE GOVIND MISHRA¹

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

* Corresponding authors:shremish@indiana.edu

project-000, April 12, 2017

Docker is a container-based technology which helps in the packaging and shipping of applications quickly and across networks. The usage of Docker is improving in the Agile Software workforce and among the DevOps because of its attributes of Continous Integration and testing. To create a cluster of Docker Engines into a single docker virtual engine is used the Docker-Swarm 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/cloudmesh/sp17-i524/raw/master/paper1/S17-IR-2021/report.pdf>

1. INTRODUCTION

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 reduces overhead and makes it easier to deploy software on servers [1]. A container allows a developer to package up an application and all its part including its code, stack it runs on, dependencies it is associated with, system tools and everything the application requires running within an isolated environment. This makes it easier for programmers and developers to run more apps on the same server and also to package and ship the apps very frequently. Docker has been able to popularize the container approach in part because it has 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.

2. HOW DOCKER DIFFERS FROM VMS

The virtualization of application can be obtained with Hypervisors or Virtual Machine Manager(VMM) which makes it easy for applications to run in isolation with one another while sharing the same underlying hardware architecture. VM hypervisors such as Hyper-V, KVM, and Xen are all based on emulating virtual hardware which means they are expensive in terms of system requirements as each VM requires the underlying VMM, Memory, and RAM. Instead of virtualizing hardware, containers rest on top of a single Linux instances. The only requirement is to keep the virtual machine image small in containers. Thus, the smaller the image is the less is needed to be stored and the lesser information is needed to be sent around the network, which makes them lightweight in comparison to the Virtual Machines. Thus a large number of containers can be hosted on the same

host OS in comparison to the Virtual Machines. A full virtualized system usually takes a time to start whereas the Docker container does take even less than seconds to upstart and running with a lower overhead than the VMs [2]. Containers are potentially much more efficient than VMs because they are able to share a single kernel and share application libraries.

3. USES OF DOCKER

3.1. Docker for DevOps

Another major use of Docker is its use in the DevOps community.“DevOps is the practice of operations and development engineers participating together in the entire service lifecycle, from design through the development process to production support [3]”. Docker is not there to replace other configuration management tools and instead can be incorporated with other configuration management tools like Chef, Puppet, Salt or ansible. The other major benefit of using Docker, Dockerfiles, the registry and the whole Docker ecosystem is that the development teams don’t have to learn domain specific language, tools or technologies which are suitable to a specific platform. Docker will not be domain specific as it can be integrated with other configuration tools and software. Docker can also eliminate the need for a development team to have the same versions of everything installed on their local machine [4]. While working on to incorporate Docker to their development workflow, Spotify created Helios, an application that manages Docker deployments across multiple servers and that alerts the development teams when the server isn’t running the correct version of a container.

3.2. Docker as Virtualized Sandbox

Docker allows systems administrators and developers to build applications that can run on any Linux distribution or hardware in a virtualized sandbox without making custom builds for all

the different environments [4]. It is easy to deploy Docker containers in a cloud scenario. Which can easily integrate it with typical DevOps environments seamlessly (Ansible, Puppet, etc.) or use it as a standalone.

3.3. Docker for continuous integration

Docker can be used as git for continuous integration. Docker is similar as changes in the system as changes can be tracked just like changes in the git. These collaboration features (docker push and docker pull) are one of the features which are important. The Docker pull/push are features which help the DevOps teams to easily collaborate and in quickly building infrastructures together. The application teams can thus share app containers with operation teams and they can share MySQL, PostgreSQL, and Redis servers with application teams[1].

4. DOCKER MACHINE

Docker Machine is a tool that lets you install Docker Engine on virtual hosts, and manage the hosts with docker-machine commands[5]. You can use Machine to create Docker hosts on your local Mac or Windows Machine, 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.

5. DOCKER SWARM

Docker Swarm provides native clustering capabilities to turn a group of Docker engines into a single, virtual Docker Engine [6]. These helps to scale up the applications as if these are all running on a single, huge machine. It does so by providing a standard Docker API where if any tool which communicates with the Docker, the daemon 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, Volume, 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 cluster. 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 the cluster, Docker Swarm can use either a hosted discovery service, static file, etcd, consul and zookeeper depending on what is best suited for the environment [5]

6. ECOSYSTEM SUPPORT TO DOCKER

Finally, it is easy to deploy Docker containers in a cloud [4]. It can be integrated in a typical DevOps environments seamlessly (Ansible, Puppet, etc.) or use it as a standalone. You can do local development within a system that is identical to a live server, deploy various development environments from your host that each uses 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 time.

- Operating Systems support to Docker: Is compatible with virtually any distribution with a 2.6.32 + kernel Red-Hat Docker collaboration to work with across fedora and other(2.6.32)+.
- It is compatible with Private PaaS(Platform as a Service) technologies: Openshift, Solum(Rackspace and Openstack).
- Public PaaS technologies like Voxoz, Cocaine(Yandex), Baidu, etc.
- Ecosystem Support for IaaS is present via Rackspace, Digital Ocean, AWS(Amazon Web Services), AMI, etc.
- Orchestration tools Support and Integration with: Chef, Puppet, Jenkins, Ansible, etc.
- In Openstack Docker integration into NOVA(compatibility with Glance, Horizon)are also present.

7. SHORTFALLS OF DOCKER

Though dockers are legacy virtualization techniques, they are not the goto solution for all kinds of virtualization and cannot be considered as a replacement of the VMs.

- VMs are self-constrained 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 affect the other VM in the virtualized environment.
- 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 such as Kubernetes and Swarm respectively are available which are not appropriate for an open-source product.

8. FUTURE WITH DOCKER

Docker Inc has set a clear path to the development of core capabilities (libcontainer), cross service management (libswarm) and messaging between containers (libchan). Docker's key open source project is libcontainer. 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. Libcontainers is receiving help from Google, RedHat, and Parallels to build the program as they will work with docker as core maintainers of the code [4]. Libcontainer which is written natively in Google's Go is also being ported to other languages. Parallels' libct, which includes libcontainer's functionality, has native C/C++ and Python bindings. Microsoft is bringing Docker to their Azure platform, a development which would integrate the Microsoft products with Linux applications.

9. CONCLUSION

According to a survey by Docker community it is observed that there are more than 1200 Docker Contributors, 10,000 Dockerized Applications at index.docker.io, 300 Million Downloads,

and 32,000 Docker-related Projects [7]. At Docker Ecosystem there is a potential for some advanced deployment tools that combine containers, configuration management, continuous integration, continuous delivery, and service orchestration in the coming days.

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] the agile admin, "What is devops," Web Page, accessed 2017-3-13. [Online]. Available: <https://theagileadmin.com/what-is-devops/>
- [4] 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>
- [5] Docker Inc, "What is docker machine," Web Page, accessed 2017-2-23. [Online]. Available: <https://docs.docker.com/machine/overview/>
- [6] Docker Inc, "Docker swarm," Web Page, accessed 2017-2-21. [Online]. Available: <https://www.docker.com/products/docker-swarm>
- [7] DATADOG, "8 surprising facts about real docker adoption," Web Page, accessed 2017-2-23. [Online]. Available: <https://www.datadoghq.com/docker-adoption/>

Triana

ABHISHEK NAIK¹

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

* Corresponding authors: ahnaik@indiana.edu

paper-1, April 12, 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/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2022/report.pdf>

1. 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.

2. FEATURES

Triana has a graphical interactive environment that can be used by the users to create their applications and specify their behavior [1]. In this, it is similar to tools like draw.io and IBM's Rational Rose that provide the user with a graphical user interface (GUI) to draw the UML diagrams [2] [3]. Rational Rose and Draw.io are mainly used for modeling the control flow diagrams like the class diagram, sequence diagram, etc., while Triana can be used for modeling generic flows like an audio processing pipeline. Thus, while conventional flows like the sequence diagram can be modeled using Triana, it is better suited for the creation of generic flows. 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. A sample Triana GUI image has been shown in Figure 1.

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.

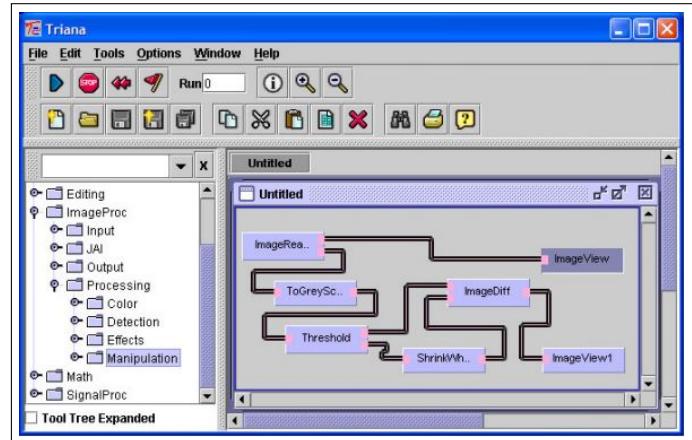


Fig. 1. Sample Triana Screen [4]

- 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.

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 [4]. It can also be used to monitor the work

flow and run the executables in standalone mode. It has a custom work flow language, although it can be integrated with others like the Business Process Execution Language (BPEL) [6]. This helps it in analysing large data sets and makes it particularly important in big data analysis.

3. INFRASTRUCTURE

The Triana infrastructure consists of the Triana Controlling Service (TCS) that has a Triana Engine, implemented as a Triana service [4]. 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 - (a) Parallel: a 'farming-out' mechanism involving no communication between the hosts; and (b) 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. The relationship between Triana and GAP is shown in Figure 2.

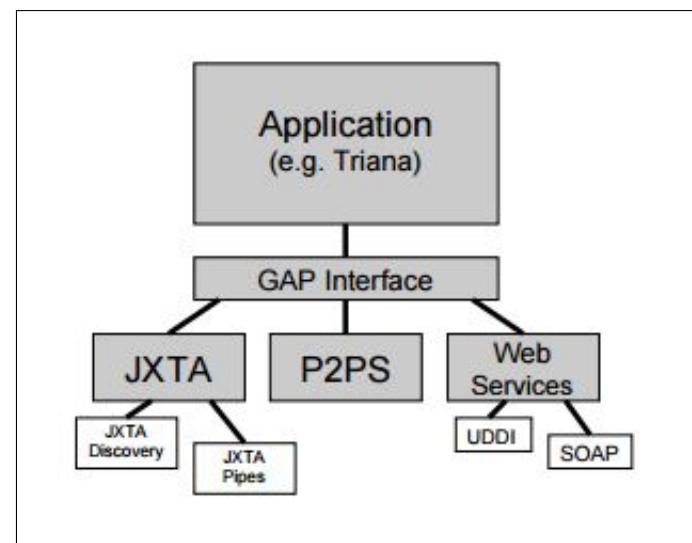


Fig. 2. Relationship between Triana and GAP [1]

4. PROJECTS

Triana has been involved in a lot of projects (including some Big Data projects) like the Data-Mining Grid, Scalable Robust Self-organizing Sensor Network Project (SRSS) and SHaring Interoperable Work flows for large-scale scientific simulations on Available DCIs (SHIWA), etc [7] [8] [9] [10]. 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. 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 [9]. The European project, SHIWA, mainly focuses on the interoperability of myriad European Work flow Systems and has been intrinsically integrated

with Triana to create SHIWA bundles [10]. 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.

5. 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/rosemob>
- [4] 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>
- [5] "Geo600-website," Web Page, Max Planck Gesellschaft, accessed: 2017-2-20. [Online]. Available: www.geo600.uni-hannover.de
- [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 PAREKH¹ AND GREGOR VON LASZEWSKI²

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

* Corresponding authors: parekhr@indiana.edu

project-001, April 12, 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/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2024/report.pdf>

1. 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.

2. 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]

3. 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]

4. 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 messages 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 performed 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 improve 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 describes 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 distinguished 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 Agent (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.

5. 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 designing 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]

6. 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 connection 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.

7. FUTURE OF LDAP

The future of LDAP lies in refinements to LDAPv3. The most recent improvements added include upgrades to management 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]

8. 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://nccs.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 RAGHATATE¹ AND SNEHAL CHEMBURKAR¹

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

*Corresponding authors: rraghta@iu.edu, snehchem@iu.edu

paper-1, April 12, 2017

Ceph is a storage solution that delivers four critical storage system capabilities:open-source, software-defined, enterprise-class and unified storage (object, block, file). Ceph is reliable, easy to manage, free and its scalability, provides 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 allowed it to be a reliable big data storage platform. This article explores these salient features of Ceph as well as studies Ceph's architecture and its comparison to few of the existing Large Scale storage systems.

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

Keywords: Ceph, scalability, storage, exabytes

<https://github.com/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2026/report.pdf>

1. INTRODUCTION

As the amount of data increases, the need to provide efficient, easy to use and reliable storage solutions has become one of the main issue for scientific computing. Even though widely used, the centralization inherent in the client/server model has proven a significant obstacle to scalable performance.

New distributed file systems(DFS) have emerged with architectural foundation on object-based storage having intelligent object storage devices (OSDs) instead of conventional hard disks. These OSDs provides CPU, network interface, and local cache with an underlying disk or RAID as one integrated solution. OSDs,replacing the traditional block-level interface with one unified platform,allowing clients to access data ranging from byte to varying sized(petabytes,exabytes) named objects. Capability of client of communicating directly with OSDs to perform I/O operations along with interaction with a metadata server (MDS) to perform metadata operations,hence improving scalability. However, limited or no metadata workload distribution, principles like allocation lists and inode tables and a reluctance to delegate intelligence to the OSDs leds most of the systems to suffer from scalability limitations.

To overcome these limitations, Ceph, a unified, distributed storage system designed for high performance, reliability and scalability [1], decouples data and metadata operations by replacing file allocation tables with generating functions allowing it to leverage the intelligence present in OSDs to distribute the complexity surrounding data access, update serialization and replication. Ceph utilizes a highly adaptive distributed metadata cluster architecture that dramatically improves the scalability of metadata access, and with it, the scalability of the entire system

[2].

Ceph is open-source storage platform providing highly scalable object, block as well as file-based storage. 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 [3].

Ceph initially began as a PhD research project in storage systems by Sage Weil at the University of California, Santa Cruz (UCSC). The name Ceph common nickname for pet octopuses, comes from Cephalopod, a class of mollusks. It suggests the highly parallel behavior of an octopus.

2. 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

2.1. Client Operation

Ceph architectural foundation depends relies mainly upon Ceph Clients and Ceph OSD Daemons(object storage daemon that runs on a cluster node and uses a local file system to store data objects) having knowledge of the cluster topology, which is inclusive of 5 maps collectively referred to as the "Cluster Map"

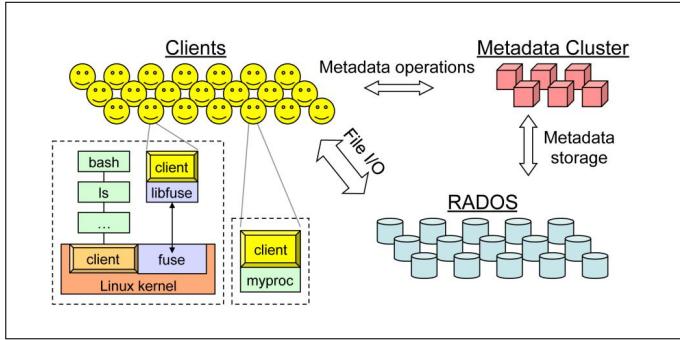


Fig. 1. System Layout of Ceph [2]

[4]. The Ceph Client is the user of the Ceph file system which runs on top of object storage system that provides object storage and block device interfaces.

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(a user-space file system interface). 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 [2].

2.2. The Ceph metadata server(MDS)

Ceph provides a cluster of metadata servers which continually load-balances itself using dynamic subtree partitioning [5]. 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 [6].

2.3. 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

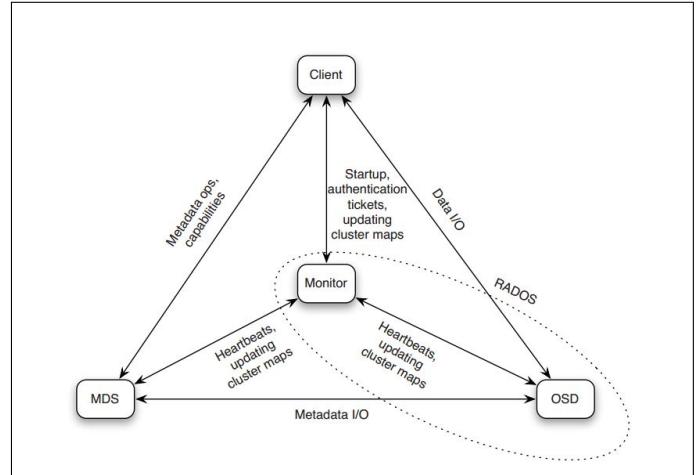


Fig. 2. Ceph components interaction [7].

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 communicate 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 [7].

2.4. 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) [8]. 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 [2]. Knowledge of failure domains is important for overall data safety of very large storage systems where correlated failures are common.

3. 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 filesystem 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 [9].
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. Dynamic 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 [10]: 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 relieving 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).
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.

4. RELATED WORK

Ceph scalability provide high-performance access to a small set of files by tens of thousands of cooperating clients in contrast to Large-scale systems like OceanStore [11] and Farsite [12] 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 [13], Galley [14], PVFS [15], and Swift [16] 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.

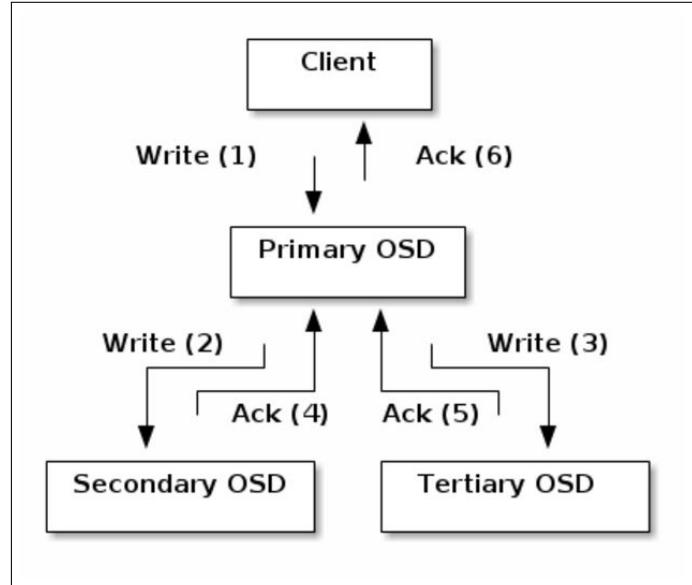


Fig. 3. Replication Process [10].

5. USE CASES

Ceph is being used in wide range of applications [17]. 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. [18].
2. Cloud Infrastructure for Microbial Bioinformatics (CLIMB) has selected and implemented Red Hat Ceph Storage for their large-scale extensive research needs [19].
3. Yahoo’s deployment of the community version of Ceph software for its Flickr and Mail applications on its Cloud Object Store (COS) [20].
4. Red Hat Ceph Storage on Dell PowerEdge server
5. Red Hat Ceph Storage on Intel processors and SSDs

6. USEFUL RESOURCES

Ceph installation manual [21], provides Installation and Deployment guide which is excellent resource as starter kit. Tutorial on Ceph Deployment by Alan Johnson [22], is a good tutorial about Ceph deployment.

7. 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 [2]. 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., "Ceph homepage-ceph," Web Page, Red Hat, Inc., 2017, accessed: 2017-2-26. [Online]. Available: <https://ceph.com/>
- [2] 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
- [3] 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>
- [4] Red Hat, Inc., "Ceph homepage-ceph," Web Page, Red Hat, Inc., 2017, accessed: 2017-3-22. [Online]. Available: <http://docs.ceph.com/docs/master/architecture>
- [5] 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>
- [6] 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/>
- [7] 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.
- [8] 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.
- [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/#scalability-and-high-availability>
- [10] 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>
- [11] 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>
- [12] 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
- [13] 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>
- [14] 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>
- [15] 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.
- [16] 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
- [17] 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/>
- [18] 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/>
- [19] 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>
- [20] 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://yahooblog.tumblr.com/post/116391291701/yahoo-cloud-object-store-object-storage-at>
- [21] 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/>
- [22] 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/>

Twitter Heron

SHAHIDHYA RAMACHANDRAN¹

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

* Corresponding authors: shahrama@iu.edu

Technology paper 1, April 13, 2017

Twitter Heron is a real-time analytics platform that was developed at Twitter for distributed streaming processing. The usage, architecture, features, performance and use cases of Twitter Heron are described in detail. Comparison of the capabilities of Twitter Heron and its predecessor Twitter Storm is summarized. A list of the other streaming technologies, which are used for streaming real-time data are discussed.

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

Keywords: Twitter Heron, Stream data, Real-time data Processing

<https://github.com/cloudmesh/sp17-i524/tree/master/paper1/S17-IR-2027>

1. INTRODUCTION

With the increase in the volume and diversity of data being generated in Twitter, the need to process these huge amounts of data in real-time along with an uprise in the number of use cases for real-time analytics, led to the necessity of having a real-time streaming system in place. This system should have the ability to process large amounts of data per minute; have sub-second latency and predictable behavior at scale; have high data accuracy in case of failure, ability to handle temporary traffic spikes and pipeline congestions; be easy to debug; and simple to deploy in a shared infrastructure as mentioned in [1].

Twitter Heron is a real-time analytics platform that was developed at Twitter for distributed streaming processing [2]. It was introduced at 'SIGMOD 2015' to overcome the shortcomings of Twitter Storm. Heron was open sourced under the Apache v2.0 license [3]. Twitter initially introduced 'Storm' which served as the real-time data processing system and was later replaced with 'Heron' that had enhanced capabilities and was architecturally more advanced than Storm [3]. Since extension of Storm would require redesigning and switching to a new platform may lead to prolonged migration periods, Twitter decided to reuse existing components of Storm and made Heron fully backward compatible with Storm.

2. ARCHITECTURE

Heron's architecture is a process-based system instead of the thread-based system used in Storm. It is implemented using standard languages like Java, C++ and Python for efficiency, maintainability, and easier community adoption since it is open sourced. Heron allows for deployment in clusters by integrating with open source schedulers like Apache Mesos, Apache Aurora, Apache REEF, Slurm [2].

2.1. Data Model

The data model and API for Heron and Storm are same in order to maintain the backward compatibility. As explained in [4], Heron contains topologies, each of which is a directed acyclic graph of spouts and bolts. Spouts generate the tuples that are given as input to the topology. Bolts are used for the computation. In the Heron topology, the number of tasks for every spout and bolt is specified. This decides the the degree of parallelism and grouping of the data as partitions when it moves through the spout and the bolt tasks. Heron's tuple processing semantics are similar to that of Storm: At most once – A tuple is not processed more than once. Some tuples may not be processes even once by the topology; At least once – Every tuple is processed at least once. Some tuples may be processed more than once [4].

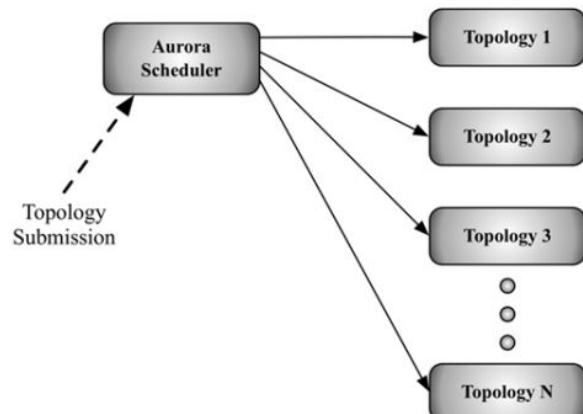


Fig. 1. Heron Architecture [4]

2.2. Overview

The architecture for Heron is shown in Figure 1. Heron API(spouts/bolts programming) is used to create and deploy topologies to the Aurora scheduler through a command line specific to Heron. As shown in Figure 2, each topology consists of several containers, Topology Master is run by the first container,each of the other containers run Stream Manager, Metrics Manager and Heron Instances. Instances are the spouts/bolts in which the user code is run. Multiple containers can be launched on a single node. Based on the resource availability these containers are allocated and scheduled by Aurora. The metadata for the topology which includes user information(who ever launched the job), time of launch, and the execution details are kept in Zookeeper. Protocol buffers are used for communication within the different processes.

2.3. Topology Master

As the name suggests the topology master is responsible for managing the entire topology and provides a consistent view of the topology throughout the process. A given topology can have only one TM and this is ensured by creating an ephemeral key in the Zoo keeper that can be easily identified by all processes run in that topology.

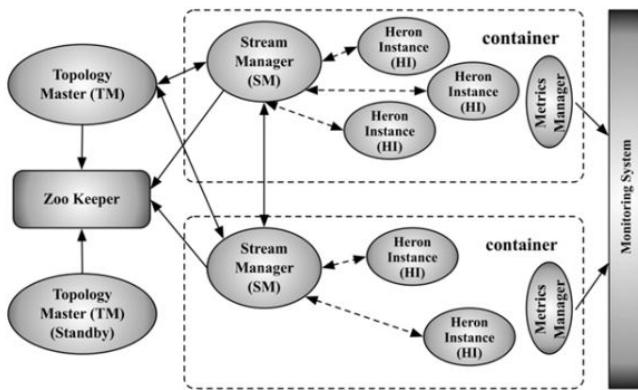


Fig. 2. Heron Topology Architecture [4]

2.4. Stream Manager

Stream Manager(SM) is used to route tuples that are sent and received by the Heron Instances. Heron uses a 'Backpressure' mechanism to control the flow of data into the system. If in a pipeline, all the stages do not process at the same rate then this may lead to tuples being dropped in order to clear the queued up buffers. This in turn will lead to loss of computational efficiency. Heron uses Spout Backpressure mechanism where the SM clamps the local spout to reduce the inflow of data. To maintain the processing rate across all SMs, the SM that is noticing slowed down performance sends a 'Start Backpressure' message to the other SMs. Once the other SMs receive this message they also clamp down their spouts until they get the 'Stop Backpressure' message from the SM that initialized the Backpressure [5].

2.5. Heron Instance

Each Heron Instance(HI) executes a single task of the spout/bolt. Heron uses the two-threaded approach that employs a Gateway thread and a task execution thread as shown in the Figure 3. The

Gateway thread is responsible for controlling all the communication and data movement in and out from the HI. It maintains TCP connections to the local SM and the metrics manager and receives the incoming tuples from the local SM. These tuples are sent to the Task Execution thread for processing. The task execution thread runs the user code. It evokes the 'execute' method in case of a bolt and the 'nextTuple' method in case of a spout.

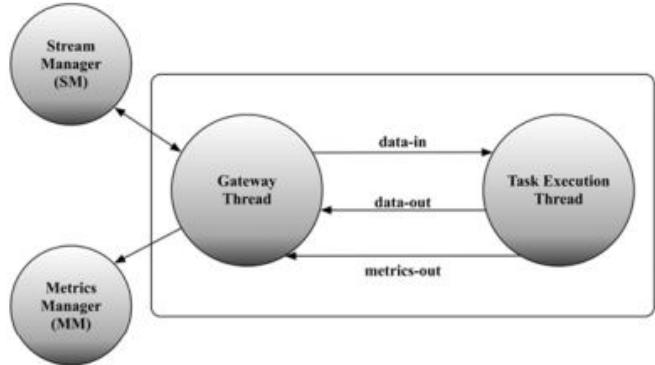


Fig. 3. Heron Instance [6]

2.6. Metrics Manager

The Metrics manager exports the system metrics and user metrics for the topologies. There is one metrics manager for each container, to which the Stream Manager and Heron Instances report their metrics.

2.7. Failure Handling

When TM dies, the failed process is restarted in the container and the TM gets its state from the Zookeeper. The standby TM becomes the master and the master becomes the standby. When an SM fails, it rediscovers the TM to initiate a connection and get the physical plan to check if there are any changes in its state. When an HI dies, it is restarted within the container and contacts the local SM to get the physical plan [6].

3. FEATURES

- API compatible with Storm: Back compatibility with Twitter Storm reduced migration time.
- Task-Isolation: Every task runs in process-level isolation, making it easy to debug/profile
- Use of main stream languages: C++, Java, Python for efficiency, maintainability, and easier community adoption
- Support for backpressure: dynamically adjusts the rate of data flow in a topology during run-time, to ensure data accuracy
- Batching of tuples: Amortizing the cost of transferring tuples
- Efficiency: Reduce resource consumption by 2-5x and Heron latency is 5-15x lower than Storm's latency [3]

4. USE CASES

Apart from being the primary streaming Engine of Twitter, Heron is being used for ETL, model enhancement, anomaly/fraud detection, IoT/IoE applications, embedded systems, VR/AR, advertisement bidding, financial, security, and social media[2]. Microsoft deployed Heron on a YARN scheduler that was implemented with the Apache REEF framework [7].

5. PERFORMANCE COMPARISON

The performance of Heron was empirically compared with Storm using word count topology. This topology counts the distinct words in a stream generated from a set of 150,000 words. According to [8] all experiments were run on machines with dual Intel Xeon E5645@2.4GHZ CPUs, each consisting of 12 physical cores with hyper-threading enabled, 72GB of main memory, and 500GB of disk space. There are no out-of-memory (OOM) crashes (or any other failure due to resource starvation during scheduling), or long repetitive GC cycles. The Storm topologies were run in isolation, i.e. no process besides the kernel, Mesos slaves, and metric exporter daemons were run in the system. Heron was run in a shared cluster, with Linux “cgroups” isolation. As shown in Figure 4, though the topology throughput increases linearly for both Storm and Heron, the throughput is 10–14x higher for Heron than that of Storm. The end-to-end latency, shown in Figure 5, increases far more gradually for Heron than it does for Storm. Heron latency is 5–15x lower than Storm’s latency. Twitter reported in [3] that with Heron, numerous topologies that aggregate data every second were able to achieve sub-second latencies and Heron was able to achieve the same with less resource consumption than Storm.

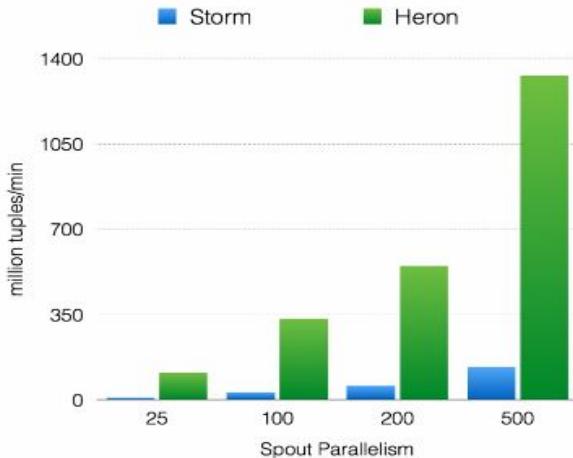


Fig. 4. Throughput with acknowledgements enabled [3]

6. OTHER STREAMING ENGINES

- Apache Flink: This streaming dataflow engine is programmed in Scala/Java/Python API. Flink offers batch processing, fault management, automated memory management, windowing features, etc [9].
- Apache Spark: It is a batch processing engine that emulates streaming via microbatching. It incorporates Scala, Java API, Python and R library [9].

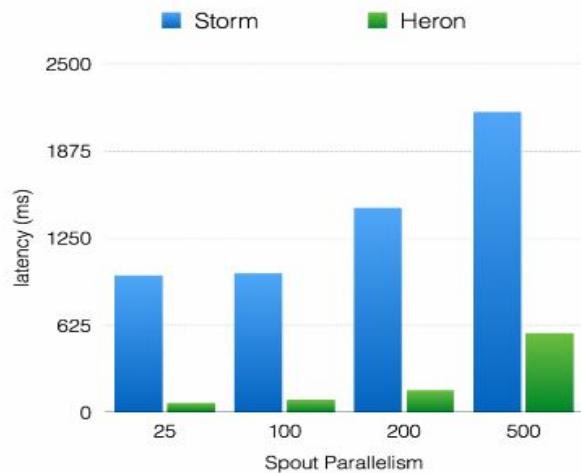


Fig. 5. Latency with acknowledgements enabled [3]

- Apache Storm: It is designed as a "topology" in the shape of a directed acyclic graph (DAG) with spouts and bolts acting as the graph vertices [10].
- Apache Samza: Apache Samza uses Apache Kafka for messaging, and Apache Hadoop YARN to provide fault tolerance, processor isolation, security, and resource management [11].
- Apache Apex: Apache Apex is a YARN-native platform that unifies stream and batch processing. It is said to be scalable, performant, fault-tolerant, stateful, secure, distributed.
- Apache GearPump: It is an event/message based real time streaming engine. Per initial benchmarks we are able to process 18 million messages per second (message length is 100 bytes) with a 8ms latency on a 4-node cluster [12].
- Amazon Kinesis: It is a platform for streaming data on AWS, to load and analyze streaming data and to build custom streaming data applications for specific needs [13].
- Google MillWheel: Framework at google to implement stream applications. Users specify a directed computation graph and application code for individual nodes, and the system manages persistent state and the continuous flow of records [14].
- Apache Beam: A framework to create pipelines. The pipelines itself will be executed on a streaming engine (such as Flink or Spark) [9].

7. CONCLUSION

Heron has been working effectively for large scale real-time data streaming at Twitter. The increase in throughput and reduction in latency of Heron has made it a useful tool for large scale processing. However it is essential to establish the required architecture to leverage the advantages of Heron. The effort and cost of setting up the Heron platform might be useful only for large scale systems.

REFERENCES

- [1] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja, "Twitter heron: Stream processing at scale," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM. Melbourne, Victoria, Australia: ACM New York, NY, USA ©2015, 2015, p. 239, accessed: 2017-2-24. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2742788>
- [2] K. Ramasamy, "Open sourcing twitter heron | twitter blogs," Web Page, May 2016, accessed: 2017-2-24. [Online]. Available: <https://blog.twitter.com/2016/open-sourcing-twitter-heron>
- [3] K. Ramasamy, "Flying faster with twitter heron | twitter blogs," Web Page, Jun. 2015, section: Our rationale and approach, Accessed: 2017-2-24. [Online]. Available: <https://blog.twitter.com/2015/flying-faster-with-twitter-heron>
- [4] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja, "Twitter heron: Stream processing at scale," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM. Melbourne, Victoria, Australia: ACM New York, NY, USA ©2015, 2015, p. 242, accessed: 2017-2-24. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2742788>
- [5] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja, "Twitter heron: Stream processing at scale," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM. Melbourne, Victoria, Australia: ACM New York, NY, USA ©2015, 2015, p. 243, accessed: 2017-2-24. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2742788>
- [6] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja, "Twitter heron: Stream processing at scale," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM. Melbourne, Victoria, Australia: ACM New York, NY, USA ©2015, 2015, p. 244, accessed: 2017-2-24. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2742788>
- [7] Twitter, "Heron documentation - apache hadoop yarn cluster(experimental)," Web Page, Aug. 2015, accessed: 2017-2-24. [Online]. Available: <http://twitter.github.io/heron/docs/operators/deployment/schedulers/yarn/>
- [8] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja, "Twitter heron: Stream processing at scale," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM. Melbourne, Victoria, Australia: ACM New York, NY, USA ©2015, 2015, p. 247, accessed: 2017-2-24. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2742788>
- [9] S. Papp, "What are the differences between apache spark, storm, samza, flink, beam, apex?" Web Page, May 2016, accessed: 2017-2-24. [Online]. Available: <https://www.quora.com/What-are-the-differences-between-Apache-Spark-Storm-Samza>
- [10] S. Kulkarni, N. Bhagat, M. Fu, V. Kedigehalli, C. Kellogg, S. Mittal, J. M. Patel, K. Ramasamy, and S. Taneja, "Twitter heron: Stream processing at scale," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, ACM. Melbourne, Victoria, Australia: ACM New York, NY, USA ©2015, 2015, p. 240, accessed: 2017-2-24. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2742788>
- [11] Samza, "Samza," Web Page, accessed: 2017-2-24. [Online]. Available: <http://samza.apache.org/>
- [12] T. A. S. Foundation, "Apache gearpump(incubating) : Overview," Web Page, accessed: 2017-2-24. [Online]. Available: <https://gearpump.apache.org/overview.html>
- [13] A. W. Services, "Amazon kinesis documentation," Web Page, 2016, accessed: 2017-2-24. [Online]. Available: <https://aws.amazon.com/documentation/kinesis/>
- [14] 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, accessed: 2017-2-24.

A Report on Kubernetes

SRIKANTH RAMANAM¹

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

* Corresponding authors: srikrama@iu.edu

April 12, 2017

Kubernetes is cluster management software. This paper explores the architecture, functioning and competition of Kubernetes.

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

Keywords: Cluster, Container, Pod, Kubernetes, I524

<https://github.com/cloudmesh/sp17-i524/raw/master/paper1/S17-IR-2028/report.pdf>

1. INTRODUCTION

Google Kubernetes is a cluster management platform developed by Google. Kubernetes is an open source system for “automating deployment, scaling and management of containerized applications” [1]. It primarily manages clusters through containers as they decouple applications from the host operating system dependencies and allowing their quick and seamless deployment, maintenance and scaling.

Docker tool, that can package applications and their dependencies into containers, was released in 2013, paving way for flexible and portable applications that can run on premises, public and private clouds. Google also decided to release key components of its orchestration, scheduling and load balancing software, called Borg as Open Source. This led to its release in 2015 as Kubernetes [2].

2. COMPONENTS

Kubernetes components are designed to be extensible primarily through Kubernetes API.

2.1. Pods

A pod is a basic unit of Kubernetes scheduling and provides a higher level of abstraction over containers. Pods are collections on containers and resources they are dependent on bundled together. Pods are not durable. They are easily created and deleted as needed.

2.2. Labels and selectors

Labels are Key-Value pairs associated with Pods that are queryable. This allows to manage a subset of nodes in a cluster selected through querying on the labels.

2.3. Services

A Kubernetes Service is a set of pods usually constitute a tier of an application. Such set of pods are identified by querying their

labels. A service usually has an external IP address, a port and a label selector.

2.4. Controllers

These are used for managing for managing the pods. Replication controller replicates specified number of pods to maintain availability. DaemonSet Controller is used for running one pod on each node. And Job Controller is used for running Pods to completion of a task like a batch job.

3. ARCHITECTURE

Kubernetes has a Master-Worker architecture. The worker agents were earlier called minions but are now known as Nodes. The Kubernetes Master is the controlling unit of the cluster and is responsible for scheduling, managing the workload and communications [2].

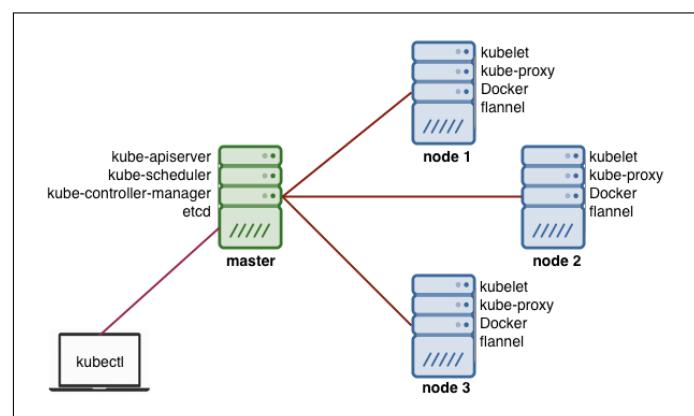


Fig. 1. Kubernetes Architecture [3]

3.1. Master

3.1.1. API server

All the tasks by master and workers are accomplished through API calls, which are handled by API server. This serves the Kubernetes API which provides external and internal interface to Kubernetes API [4].

- Get API version: /api/
- Get list of pods: /api/v1/namespaces/default/pods
- Get list of nodes: /localhost:8080//api/v1/nodes

New configurations for pods can be uploaded using JSON or yaml files.

3.1.2. Etcd

This is a persistent key-value data store used for storing configuration and overall state of the cluster. Etcd was developed by the CoreOS team.

3.1.3. Scheduler and Controller Manager

Controller manager communicates with the API server and creates, manages or deletes pods according to the API calls and/or preset configurations [5].

3.2. Worker Node

3.2.1. Kubelet

This is a special background process meant for implementing commands from master. The common commands include create, destroy and monitor its containers.

3.2.2. Proxy

This is a network proxy and a load balancer at the node level and is responsible for redirecting the incoming requests to the appropriate containers of the node.

3.2.3. cAdvisor

This is an agent that monitors resource usage and performance metrics of containers in nodes.

4. FUNCTIONING

Kubernetes applications are deployed over clusters as Services which comprise of a collection of pods. Pods are groupings of containers and their resources. These services are accessed through the endpoints they expose. Usually the target is to ensure high availability and scalability of services by managing the pods through controllers- creating, allocating and deleting when needed.

5. USER INTERFACE

Along with REST API, Kubernetes also provides a command line interface, kubectl [6], and a web based dashboard. This dashboard can be utilized to deploy, troubleshoot and manage containerized applications on Kubernetes. It also displays key metrics on resource usage and errors. Figure 2 shows a kubernetes dashboard.

6. LICENSE AND PRICING

Kubernetes is licensed under Apache License 2.0. [8].

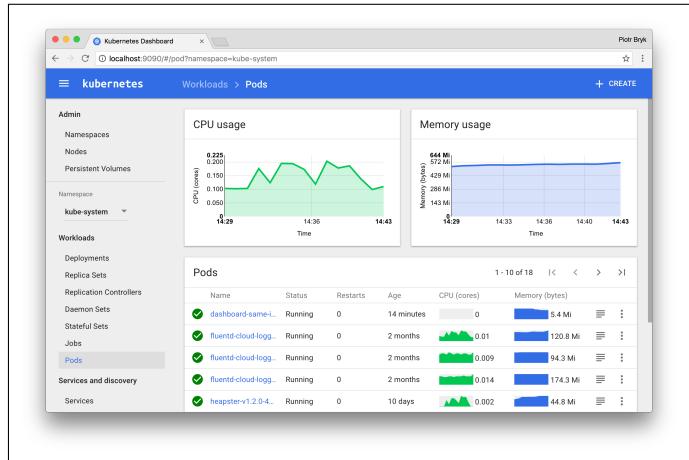


Fig. 2. UI-dashboard [7]

6.1. Pricing

While Kubernetes itself is free and open source, a paid kubernetes implementation is being offered by Google Cloud Platform. Nodes are priced between 0.010 USD and 0.240 USD per hour depending on the type of the machine [9]. 0 to 5 node clusters are free when you are paying for the nodes. 6 plus node clusters are being charged at 0.15 USD per hour on top of the price for nodes [10].

7. COMPARISON

Some of the competitors of Kubernetes that offer orchestration, clustering and cluster management are [11]:

7.1. Docker Swarm

Swarm uses the standard Docker interface. Though this makes it easy to integrate with existing work flows, this may make it difficult to implement complex scheduling with custom interfaces.

7.2. Fleet

Fleet is a low level orchestration tool that can be used run other high level orchestration systems such as Kubernetes.

7.3. Mesos

Apache Mesos is also a low level, but more stable, scheduler that supports several frameworks for container orchestrations such as Kubernetes, Marathon and Swarm. Usually large applications and in Big Data centers.

7.4. Kontena

Kontena a relatively newcomer to the field is aimed at simplifying the whole process and is targeting smaller companies like startups and quicker deployment [12].

8. USERS

Some of the Current users of Kubernetes are Pearson, ebay, wikipedia, box, SAP and ComCast [13].

9. CONCLUSION

Kubernetes is an open source cluster management software. Kubernetes comes in with the advantage of being the result of 15

years of research and development by Google and is considered stable. It also has an active community of developers. While embracing Kubernetes might involve some redesigning of existing applications, when utilized efficiently it offers a fault-tolerant and efficient system.

ACKNOWLEDGEMENTS

This paper has been written as part of a class assignment for the course: I524: Big Data Software and projects, Spring 2017, School of Informatics and computing, Indiana University, Bloomington. Special thanks to Professor Gregor von Laszewski, Dimitar Nikolov and all associate instructors for guiding through the process of writing this paper.

REFERENCES

- [1] Kubernetes, "Kubernetes site," Web Page, accessed: 2017-02-26. [Online]. Available: <https://kubernetes.io/docs/whatisk8s/>
- [2] D. Rensin, *Kubernetes*, 1st ed. O'Reilly Media, Inc., Jun. 2015, accessed: 2017-02-27.
- [3] S. Braun, "Kubernetes architecture," Web Page, 2016, accessed: 2017-04-04. [Online]. Available: <http://www.yet.org/2016/06/kubernetes/>
- [4] Kubernetes, "Kubernetes api," Web Page, accessed: 2017-04-04. [Online]. Available: <https://kubernetes.io/docs/tasks/access-kubernetes-api/http-proxy-access-api/>
- [5] Wikipedia, "Kubernetes wiki," Web Page, accessed: 2017-02-26. [Online]. Available: <https://en.wikipedia.org/wiki/Kubernetes>
- [6] Kubernetes, "kubectl documentation," Web Page, accessed: 2017-02-26. [Online]. Available: <https://kubernetes.io/docs/user-guide/kubectl-overview/>
- [7] Kubernetes, "Kubernetes ui dashboard," Web Page, accessed: 2017-02-26. [Online]. Available: <https://kubernetes.io/docs/user-guide/ui/>
- [8] Kubernetes, "Kubernetes license," Git hub file, 2015, accessed: 2017-02-26. [Online]. Available: https://github.com/kubernetes/kubernetes/blob/master/third_party/swagger-ui/LICENSE
- [9] Google, "Google container engine pricing," Web Page, Feb. 2017, accessed: 2017-02-26. [Online]. Available: <https://cloud.google.com/compute/pricing>
- [10] G. C. Platform, "Google cloud platform pricing," Web Page, Feb. 2017, accessed: 2017-02-26. [Online]. Available: <https://cloud.google.com/container-engine/pricing>
- [11] A. Mouat, "Kubernetes competition," Web Page, Oct. 2015, accessed: 2017-02-26. [Online]. Available: <https://www.oreilly.com/ideas/swarm-v-fleet-v-kubernetes-v-mesos>
- [12] Kontena, "Kontena website," Web Page. [Online]. Available: <https://www.kontena.io/>
- [13] Kubernetes, "Kubernetes users," Web Page, accessed: 2017-02-26. [Online]. Available: <https://kubernetes.io/case-studies/>

An overview of Azure Machine Learning and its Applications

NAVEENKUMAR RAMARAJU¹

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

* Corresponding authors: nramraj@umail.iu.edu

April 12, 2017

This paper provides a summary about Microsoft Azure Machine learning and features available in it. Some real world applications done using Azure machine learning studio is also explained.

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

Keywords: Azure, machine learning, predictive analytics, Microsoft

<https://github.com/naveenkumar2703/sp17-i524/paper1/S17-IR-2029/report.pdf>

1. INTRODUCTION

Azure machine learning is a cloud based service that could be used to build and deploy predictive models and analytics solutions. Azure machine learning studio makes it easier to build a predictive models with interactive user interface instead of programming for various operations like data cleaning, data transformation, feature selection and machine learning.

Azure also provides in built API to most of the popular machine learning algorithms like linear regression, anomaly detection, forecasting etc. A list of available machine learning APIs are discussed in section 2.3. Azure machine learning also offers pre-trained models for problems like speech recognition and image detection. Since building and deploying the predictive models are done in cloud based environments, overhead with installation and maintenance of software and hardware are reduced.

The paper is organized as follows. First the azure machine learning stack is explained with its features and APIs available. It is followed by licensing options available and associated useful Azure services are briefly discussed. Finally, overview of some of the applications built using Azure machine learning eco-system is discussed.

2. COMPONENTS OF AZURE MACHINE LEARNING

2.1. Azure Machine Learning Studio

"Microsoft Azure Machine Learning Studio is a collaborative, drag-and-drop tool you can use to build, test, and deploy predictive analytics solutions on your data. Machine Learning Studio publishes models as web services that can easily be consumed by custom apps or BI tools such as Excel." [1]

Azure Machine learning studio provides an interactive workspace to build predictive models with visual aids. Models are built using user interface. It allows data scientists to create an

experiment and iterate on model design by editing the models and data in a canvas with a drag and drop options. The models could be saved and published as webservice when ready.

2.2. Functionalities in Azure Machine Learning

Azure Machine Learning has many functionalities that are required to do end to end predictive analytics or machine learning [2]. The key features available in studio are: 1) importing raw data; 2) data preprocessing; 3) feature engineering and data labeling (for supervised learning such as classification); 4) training, scoring, and evaluating the model; 5) model comparison and selection; 6) saving the trained model; 7) creating a predictive experiment; and 8) publishing the web service in Azure Machine Learning.

2.3. Modules/APIs in Azure Machine Learning

In Azure machine learning a module is a building block for creating experiments. It comes with lot of inbuilt modules for various functionalities involved in predictive analytics. Full list of modules are available in Azure web site [3]. Some of the popular machine learning modules are grouped in one of four types: regression, classification, clustering and anomaly detection.

Anomaly detection includes algorithms like fraud detection, network intrusion, and abnormal clusters using PCA and one class SVM. Classification module has algorithms like decision trees, logistic regression, neural networks, SVM. Clustering module has K-means clustering algorithm. And regression module includes Bayesian, linear, poisson and ordinal regression algorithms.

Based on their functionality modules are available in following categories: Data Format conversions, Data Input and Output, Data Manipulation, Feature Selection, Machine learning, OpenCV modules, R and Python Language modules, Statistical

Funcitons, Text analytics and Time series module [4]

2.4. Cortana Intelligence Gallery

The Cortana Intelligence Gallery[5] is a site where the users and creators of Azure Machine learning and Cortana Intelligence Suite post their models and solutions. The Gallery contains a variety of resources that could be used to develop own analytics solutions. It also has experiments, tutorials and solutions built by leading users in the industry.

3. LICENSING

Microsoft Azure Machine learning studio is offered in two versions: a free version and a standard version.[6] Free version could be used without Azure subscription. It has limitations on maximum number of modules allowed per experiment, experiment duration, storage space and models cannot be published as API. Standard version comes at a monthly subscription charge of \$9.99 (USD) and \$1 per experimentaion hour.

In addition if models are published as web API in production they come in several tiers of pricing from free test tier to standard tiers such as S1, S2 and S3 based on number of calls and services required. The monthly charge range from \$0 to \$10000 based on the tier chosen.

4. MICROSOFT AZURE ECOSYSTEM

Machine learning studio is a part of Microsoft Azure, a collection of integrated cloud services that could be used for various purposes like computing, networking, Intellignece and Analytics etc. HDInsights, SQL data ware house and data factory are related services offered under Azure umbrella. These services enable storage and functioning of big data applications. Machine learning studio could be integrated with SQL data ware house for real time analytics on big data. The ecosystem is illustrated in Fig 1.

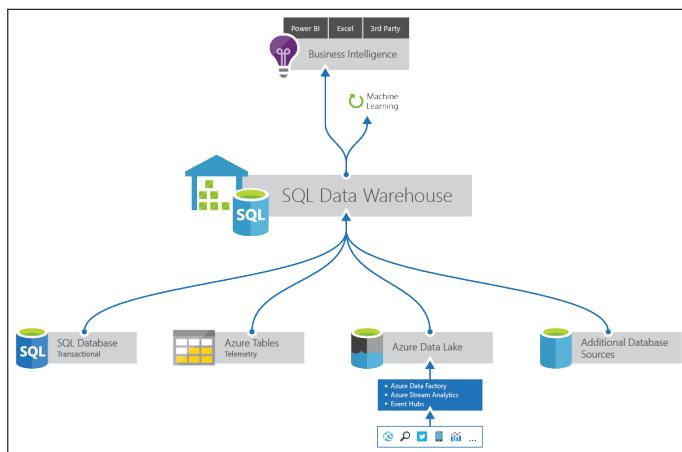


Fig. 1. Azure Machine Learning Ecosystem.

Source: Microsoft Azure website[7]

5. USE CASES

Microsoft Azure machine learning is used in wide range of aplications in research and industry. Here two such applications are described briefly.

5.1. Decoding Brain Signals

Decoding brain signals competition[8] was conducted Microsoft in which the obejctive is to train machine learning algoritm to interpret brain signals. The task is to learn a model that needs to predict whether the human subject is seeing a house image or a face image from the ECoG signals collected from the subtemporal cortical surfaces of four seizure patients.

The winning solution developed using Azure machine learning studio ensembeled 5 different models, two employing detection of evoked potential, and three using induced activity. All the models used data ranging from 100ms to 400ms after the onset of the stimulation. The model did not used any subject specific training or preprocessing.

Covariance matrices were used as feature for classification and logistic regression was used as classifier. The final results obtained using three fold cross validation gave accuracy of 0.921.

The solution also highlights some of the downsides of Azure machine learning studio as: "The Azure ML studio is not very adapted for exploratory data analysis and validation of the models. Mastering the cross-validation parameters, training subject specific model and ensembling them was a difficult task to setup. The scikit-learn distibution available on Azure ML was seriously outdated, and i could not install my own packages." [9]

5.2. Solving Business Decision Making Problem

In the study done by faculty at Zlin University[10], they have used the Azure Machine Learning platform to predict and compare the performance of telecommunication industry between Mexico and Sri Lanka. Data analysis was carried out in Azure Machine Learning's decision forest regression model. The study indicated that a decision forest regression model on Azure Machine Learning produced a relative accuracy of 70 percent and 60 percent on Mexico and Srilanka. Results of the model indicated the ability of the model in terms of forecasting information, can make predictions based on these models to make their decisions effectively at very high accuracy levels.

6. USEFUL RESOURCES

Predictive Analytics with Microsoft Azure Machine Learning [11] by Roger Barga is a wonderful text book that gives an introduction to basic machine learning concepts and how to apply them to real world data using Microsoft Azure Machine Learning and Studio.

Microsoft Azure website[12] also has some good step by step tutorials on how to use Azure Machine learning studio and real world applications done by Microsoft and users of Azure Machine learning studio.

7. CONCLUSION

Azure machine learning is a cloud based service that could be used for predictive analytics. Its user interface based studio provides capability to process data, build and test models iteratively with simple drag and drop options. The models developed using standard verison could be published as web API. Some down-sides of using studio are: to get full advantage users have to use standard version for monthly subscrition. As the softwares and tools are cloud based, getting access to latest versions could be trickier. Azure machine learning is used in research and industry to build predictive models quickly.

ACKNOWLEDGEMENTS

This work was done as part of the course "I524: Big Data and Open Source Software Projects" at Indiana University during Spring 2017. Thanks to our Professor Gregor von Laszewski and associate instructors for their help and support during the course.

REFERENCES

- [1] G. Ericson, L. Franks, and P. McKay, "What is azure machine learning studio?" Web Page, Jan. 2017, accessed: 2017-2-19. [Online]. Available: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-ml-studio>
- [2] H. Ning, G. Ericson, L. Franks, and P. McKay, "Machine learning studio capabilities," Web Page, Jan. 2017, accessed: 2017-2-19. [Online]. Available: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-studio-overview-diagram>
- [3] "Machine learning modules a-z," Web Page, Nov. 2016, accessed: 2017-2-19. [Online]. Available: <https://msdn.microsoft.com/en-us/library/azure/dn906033>
- [4] "Machine learning modules descriptions," Web Page, Feb. 2017, accessed: 2017-2-19. [Online]. Available: <https://msdn.microsoft.com/en-us/library/azure/dn906013>
- [5] "Cortona intelligence gallery," Web Page, Feb. 2017, accessed: 2017-2-19. [Online]. Available: <https://gallery.cortanaintelligence.com/>
- [6] "Machine learning pricing," Web Page, Feb. 2017, accessed: 2017-2-19. [Online]. Available: <https://azure.microsoft.com/en-us/pricing/details/machine-learning/>
- [7] "Azure SQL data warehouse," Web Page, Feb. 2017, accessed: 2017-2-19. [Online]. Available: <https://azure.microsoft.com/en-us/services/sql-data-warehouse/>
- [8] "Decoding brain signals," Web Page, Jul. 2016, accessed: 2017-2-19. [Online]. Available: <https://gallery.cortanaintelligence.com/Competition/Decoding-Brain-Signals-2>
- [9] A. Barachant, "Winning solution of decoding brain signals," Web Page, Jul. 2016, accessed: 2017-2-19. [Online]. Available: <https://github.com/alexandrebarachant/decoding-brain-challenge-2016>
- [10] L. A. B. Prieto and R. H. Kuruppuge., "Solving business decision-making problems with an implementation of azure machine learning," in *DOKBAT 12th Annual International Bata Conference*, ser. DOKBAT. Zlin, Czech Republic: Tomas Bata University, 2016, pp. 42–55. [Online]. Available: <http://dokbat.utb.cz/2016/wp-content/uploads/2015/04/DOKBAT-2016-Conference-Proceedings1.pdf#page=42>
- [11] V. Fontama, R. Barga, and W. H. Tok, *Predictive Analytics with Microsoft Azure Machine Learning 2nd Edition*. APRESS L.P., 2015. [Online]. Available: http://www.ebook.de/de/product/24095855/valentine_fontama_roger_barga_wee_hyong_tok_predictive_analytics_with_microsoft_azure_machine_learning_2nd_edition.html
- [12] C. Gronlund, G. Ericson, L. Franks, and P. McKay, "Azure machine learning," Web Page, Jan. 2017, accessed: 2017-2-19. [Online]. Available: <https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-what-is-machine-learning>

Microsoft Azure Data Factory

SOWMYA RAVI¹

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

* Corresponding authors: laszewski@gmail.com

project-000, April 12, 2017

Azure Data Factory is a cloud based data integration service that handles and automates data movement and data integration services. The Data Factory stands out in its ability to ingest data from multiple sources, efficiently build pipelines to manage processes and visualize the results in a pragmatic manner. The paper discusses the data movement and data transformation activities supported by Azure Data Factory

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

Keywords: Cloud, Data, Movement, Transformation

<https://github.com/sowravi>

1. INTRODUCTION

The cloud based data integration service is designed to pull data from various sources, transform/ process data and publish the result to the data stores. A data management gateway enables access to data on SQL Databases [1]. The data processing is done by It works by creating pipelines to transform the raw data into a format that can be readily used by BI Tools or applications. The services comes with rich visualization aids that aid data analysis. Data Factory supports two types of activities: data movement and data transformation. Data Movement [2] is a Copy Activity in Data Factory that copies data from a data source to a Data sink. Data Factory supports the following data stores. Data from any source can be written to any sink. Data Transformation: Azure Data Factory supports the following transformation activities such as Map reduce, Hive transformations and Machine learning activities. Data factory is a great tool to analyze web data, sensor data and geo-spatial data.



Fig. 1. Work flow in Azure Data Factory [3]

Fig. 1. illustrates the flow of activities in Data Factory. Data can be ingested from several data sources that reside either on-premises or on cloud platforms. Pre-processing steps can be applied to the data after data ingestion. Once the data preparation is complete, scheduling, transformation and analysis is carried out using data pipelines. The results of data analysis is

then published using the visual aids provided by Data factory[4].

2. DATASETS IN AZURE DATA FACTORY

The JSON format is used to store data in Azure Data Factory [3]. In Azure data factory creating a dataset essentially means creating a pointer to the data to be processed. The data store could be tables, files, folders, and documents. After a dataset is created, it can be used with activities in a pipeline which will be discussed later in the paper. For example, a dataset can be an input/output dataset of a Copy Activity or an HDInsightHive Activity. A visual layout of all your pipelines and data inputs and outputs is also provided by Azure. It enables one all the relationships and dependencies of pipelines across all sources. Thus, it is possible to keep track of all data sources and sinks.

3. PIPELINES AND ACTIVITIES IN AZURE DATA FACTORY

3.1. Data Pipeline

Azure Data Factory used pipelines to organize work flow and data processing [5]. A pipeline is an ordering of processes in accordance with the logic relation existing between the processes.

3.2. Activity

An activity is a process or action that is taken on the data [5]. An activity takes zero or more datasets as inputs and produces zero or more datasets as an output. The activities are grouped under data movement of data transformation activities.

3.2.1. Data Movement

The Data Movement is achieved by the Data Factory copy wizard. Data can be copied either from local or cloud sources with

ease for further transformation to be carried out. The copy activity is highly secure, reliable and scalable.

When the source as well as the sink data stores are in the cloud, Copy Activity goes through the following stages to copy data.

- Reads data from the source data store
- Performs serialization/de-serialization, compression/decompression, column mapping, and type conversion. It does these operations based on the configurations of the input dataset, output dataset, and Copy Activity.
- Writes data to the destination data store

In case of moving a data located on-premises to cloud, data management gateway must be installed in the local machine containing the data. The data management gateway is required to perform serialization/de-serialization, compression/decompression, column mapping, and type conversion. The data management gateway directly writes the data to the destination. The following figure shows the copy activity from a local machine

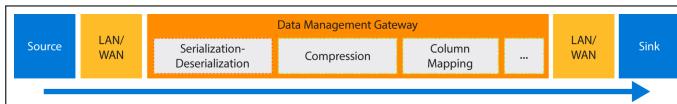


Fig. 2. Moving data between on-premises and cloud data stores [6]

The Azure Data Factory is connected to a variety of data stores and supports different data formats [6]. Copy Activity also reads from and writes to files in specified formats: text, Avro, ORC, Parquet, and JSON, and compression codec GZip, Deflate, BZip2, and ZipDeflate are supported. Furthermore, [6] pipelines could be created either using the copy wizard or using the JSON scripts. The scheduling facility make it possible to run multiple copy operations one after the other in a sequential manner. After copying, the data is ready for transformation purpose

3.2.2. Data Transformation

Data transformation activities transform the raw data to derive predictions and insights. The Data Factory comes with two types of computing environments

- On Demand: This type of environment is solely managed by Data Factory. The environment is created before a process begins and is automatically removed when the process is completed
- Bring your own: An individual can register his/her computing environment and has to be managed by the individual.

The following transformations are supported By Azure Data Factory [7]

- HDInsight Hive activity: The HDInsight Hive activity executes Hive queries on your own or on-demand Windows/Linux-based HDInsight cluster [8].
- HDInsight Pig activity: The HDInsight Pig activity in a Data Factory pipeline executes Pig queries on your own or on-demand Windows/Linux-based HDInsight cluster [9].

- HDInsight MapReduce activity: The HDInsight MapReduce activity in a Data Factory pipeline executes MapReduce programs on your own or on-demand Windows/Linux-based HDInsight cluster [10].

- HDInsight Streaming activity: HDInsightStreamingActivity Activity is used to invoke a Hadoop Streaming job from an Azure Data Factory pipeline. The HDInsight Streaming Activity in a Data Factory pipeline executes Hadoop Streaming programs on your own or on-demand Windows/Linux-based HDInsight cluster [11].
- Machine Learning activities: Azure Data Factory helps in building and deploying predictive models in the following three steps [12].

1. Create a training experiment: This is done using the Azure ML Studio. The ML studio is a collaborative visual development environment that can be used to train and test a predictive analytics model using training data.
2. Convert it to a predictive experiment: Once the model has been trained with existing data, it can be used to score new data, prepare and streamline the experiment for scoring.
3. Deploy it as a web service. The experiment can be published using an Azure web service.

- Stored procedure activity: A procedure that is already stored can be invoked in one of the following data stores: Azure SQL Database, Azure SQL Data Warehouse, SQL Server Database in your enterprise or on an Azure virtual machine (VM) [13].
- Data Lake Analytics U-SQL activity: Azure Data Lake Analytics can be linked to the compute service in the Data Factory [14].
- .NET custom activity: One can create an activity with a customized logic to be used in a pipeline [15].

4. MONITORING AND MANAGEMENT

The Data Factory has a monitor and manage app which helps keep tab of available resources, handle alerts and monitor various views. The resource explorer has a tree view and a diagram view of all the existing pipelines, datasets and linked services [16].

5. USE CASES

Data factory can be used for a wide range of applications such as Product recommendations, Customer profiling etc. The sample solution for a recommendation system is illustrated in Fig.3. In this case, the customer information, product information and the behavioural data of the customer are stored in azure blobs. The combined data is fed to the recommendation system which is modelled using any of the algorithms available in the Azure Machine Learning suite. The final result contains customer specific product recommendations that can be written to relational database for the consumption by the store's website [17].

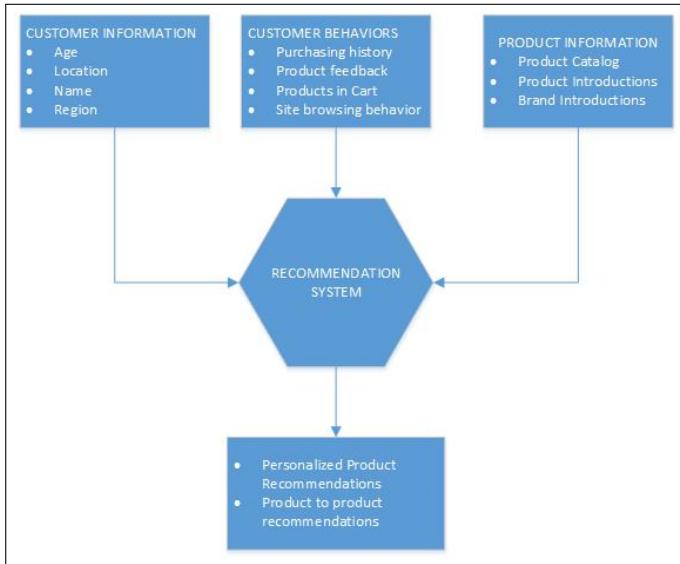


Fig. 3. A recommendation engine using Azure Data factory [17]

6. CONCLUSION

The paper talks about the data orchestration and management using Microsoft's Data Factory. Its ability to link diverse data stores makes it a versatile tool for data integration. In addition, the powerful pipeline based data transformation services embedded in it helps data scientists and developers to provide efficient and effective solutions using large datasets.

REFERENCES

- [1] J. Serra, "What is azure data factory?" Web Page, accessed: 02/03/2016. [Online]. Available: <http://www.jamesserra.com/archive/2014/11/what-is-azure-data-factory/>
- [2] L. Sharon, P. Sreedhar, and C. Caro, "Introduction to azure data factory service, a data integration service in the cloud," Web Page, accessed: 02/03/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-introduction>
- [3] L. Sharon, P. Sreedhar, W. Patricia, and M. Cody, "Datasets in azure data factory," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-create-datasets>
- [4] Microsoft, "Overview diagram," Web Page, accessed: 03/17/2016. [Online]. Available: <https://azure.microsoft.com/en-us/services/data-factory/>
- [5] L. Sharon, P. Sreedhar, and M. Patricia, Wagnerand Robert, "Pipelines and activities in azure data factory," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-create-pipelines>
- [6] W. Linda, P. Sreedhar, G. C J, P. Wagner, and A. Harish Kumar, "Move data by using copy activity," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-data-movement-activities>
- [7] L. Sharon, P. Sreedhar, W. Patricia, and M. Robert, "Transform data in azure data factory," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-data-transformation-activities>
- [8] L. Sharon, P. Sreedhar, and F. Iain, "Transform data using hive activity in azure data factory," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-hive-activity>
- [9] L. Sharon, P. Sreedhar, W. Patricia, and M. Robert, "Transform data using pig activity in azure data factory," Web Page, accessed:

02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-pig-activity>

- [10] L. Sharon, P. Sreedhar, F. Iain, J. Alma, and S. Jamie, "Invoke mapreduce programs from data factory," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-map-reduce>
- [11] L. Sharon, P. Sreedhar, W. Patricia, and M. Robert, "Transform data using hadoop streaming activity in azure data factory," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-hadoop-streaming-activity>
- [12] L. Sharon, P. Sreedhar, G. Brian, and F. Iain, "Create predictive pipelines using azure machine learning and azure data factory," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-azure-ml-batch-execution-activity>
- [13] P. Sreedhar, F. Iain, and A. Harish Kumar, "Sql server stored procedure activity," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-stored-proc-activity>
- [14] P. Sreedhar, F. Iain, and S. Jamie, "Transform data by running u-sql scripts on azure data lake analytics," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-usql-activity>
- [15] L. Sharon, P. Sreedhar, W. Patricia, and M. Robert, "Use custom activities in an azure data factory pipeline," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-use-custom-activities>
- [16] P. Sreedhar, "Monitor and manage azure data factory pipelines by using the monitoring and management app," Web Page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-monitor-manage-app>
- [17] L. Sharon, P. Sreedhar, E. Gary, and J. Alma, "Use case - product recommendations," Web page, accessed: 02/25/2016. [Online]. Available: <https://docs.microsoft.com/en-us/azure/data-factory/data-factory-product-reco-usecase>

AUTHOR BIOGRAPHIES



Sowmya Ravi pursuing Master's in Data Science at Indiana University, Bloomington. Her research interests include Machine learning, Data Mining and Big Data

Google Cloud storage: A journey towards Cloud storage

KUMAR SATYAM¹

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

* Corresponding authors: ksatyam@indiana.edu

paper-1, April 12, 2017

Now-a-days due to exponential growth of data, it is almost impossible to store the huge amount the data locally in our own systems or local storage boxes which are prone to data loss in case of hardware failure. With cloud storage one can save the data in cloud without any risk of data loss because of redundancy of data over cloud. It explains about the Google cloud storage technology which is one of cloud storage vendor. We will discuss what are the key elements of Google cloud storage and try to relate it with the general cloud storage technology[1].

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

Keywords: Cloud, Google Cloud Storage, BigQuery, Buckets, Objects, I524

<https://github.com/satyamsah/sp17-i524/blob/master/paper1/S17-IR-2031/report.pdf>

1. INTRODUCTION

Google cloud storage is a unified object storage mainly targeting enterprises and developers which offers capability of both live data serving with high I/O throughput to data archiving. As shown in Fig 1, it is a RESTful online file storage web service for accessing data on Google Cloud platform [2]. It is an IAAS cloud delivery model which provides good performance and scalability using Google's cloud platform.

3. KEY TERMS

Projects:

All the data in Google Cloud storage belongs inside a project. A project consists of a set of users, set of API and billing, authentication and monitoring setting for those APIs.[3]

Buckets:

Buckets are basic containers that hold the data. Everything that a user/organization stores in Google Cloud Storage must be contained in a bucket. Buckets have bucket name. In Google cloud storage, buckets are created to store the data. A Bucket has three properties, a location is where the bucket and its contents are stored, a globally unique name and, a default storage class. For example, if a user specifies the location 'eu' (European union) when he creates a bucket A, then bucket A and any objects of bucket A are stored on servers of European union region.

Objects:

2. ARCHITECTURE -GOOGLE CLOUD PLATFORM

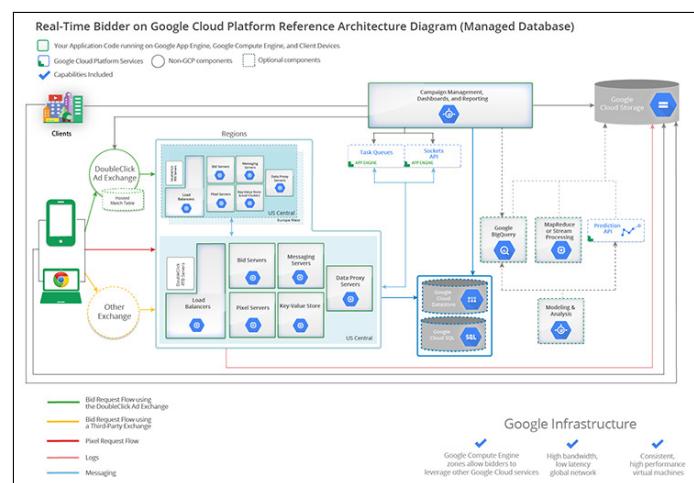


Fig. 1. Architecture of Google cloud platform using Google cloud storage as persistent layer [3]

Objects are individual pieces of data which are stored in Google cloud storage. A single object can go up to 5 TB in size. Objects have 2 components: Object data and object metadata. The object data component is usually a file that a user stores in Google cloud storage. Object name is just meta data to cloud storage. A common character in the file system is the slash (/). By slash an object name appears as they are stored in hierarchical structure. But Google cloud storage sees the object as independent objects with no hierarchy.

Data opacity:

An Object's data component is completely opaque to Google cloud storage. It is just a chunk of data to Google Cloud storage.

Object immutability:

Objects are immutable which means that an object cannot change throughout their storage lifetime. An object's storage life time is between the successful object creation and successful object deletion.

Hierarchy:

Google cloud storage uses a flat namespace to store objects. However, some tools show it in hierarchical manner.

Namespace:

There is only one google cloud storage namespace that means every bucket should have a unique name. Object name must be unique inside a bucket.

4. STORAGE CLASSES

In Google cloud storage bucket is the place where a user can store data. As explained earlier bucket needs to be in a 'Storage class'. One can think this a service tier of creating a bucket. There are 4 storage classes offered in Google cloud storage:

Multi-regional storage:

This offers 99.95 % availability with geo-redundant feature. It provides maximum availability of the data even in case of large scale disruption. It is used to store data that is frequently accessed around the world, such as serving website content. The cost of using this storage class is \$0.026 per GB per month. Till now there are three multi – region location

- asia — Asia Pacific
- eu — European Union
- us — United States

Regional storage:

It is used for high availability and redundancy within a single region. This offers 99.95 % availability. The cost of using this storage class is \$0.02 per GB per month. It is recommended to the users to use "Regional" storage to utilize durable reduced availability. A user can move data from DRA to other storage classes by performing a storage transfer. It is best when a user wants to optimized latency, availability and, network bandwidth within the same location. It can still be read globally , but if the organization's bucket is used primarily by client outside region , it is better to switch to multi-region location. It is best for application which require high data analysis and computation.Till now there are three multi–region location :

- asia-east1 — Eastern Asia-Pacific
- asia-northeast1 — Northeastern Asia-Pacific

- europe-west1 — Western Europe
- us-central1 — Central United States
- us-east1 — Eastern United States
- us-west1 — Western United States

Nearline Storage:

It is a storage offering which is suited for the archived data which the user access less than once a month .It is low cost, highly durable storage service for infrequently accessed data. It is ideal for back-up and serving long-tail multimedia. The cost of using it is \$ 0.01 per GB per month.

Coldline storage:

Coldline storage is the storage which is used for the data which the user access less than once a year. Typically this used in case of disaster recovery where it provides low latency to access data stored in Coldline storage ,or for the data that is archived and may or may not be needed at some future time. Cost is using this \$ 0.007 per GB per month.

Standard storage:

When a user creates a bucket without specifying storage class, the cloud storage will create a standard bucket which can be either be a multi-regional or a regional storage based on the location the user is creating the storage bucket. If a user has a standard storage in a multi-regional location like 'us' then it is equivalent to multi regional storage and continue to exist in 'us' location. If the standard storage is in the regional storage like 'us-east', it will be equivalent to 'regional' storage

Note: Each bucket has a default storage class which one can specify while creating the bucket. The objects also take the default storage unless specified otherwise. One can change the default storage class of his bucket anytime. But the objects which were created in the bucket before changing the storage class will remain in the same(earlier) storage class and will not change to new storage class. It will only affect the storage class of the objects which will be created after the change of the storage class of the bucket. One can specify the storage class of each object which is added to the bucket or can change their name afterwards using the API calls.

5. ACCESS CONTROL OPTIONS

This takes care of authentication and authorization mechanism. One can control who has access to organization's Cloud storage buckets and objects as well as what level of access they have.

Identity and access management(IAM):

IAM grants access to project's bucket and objects. For example, one can specify that a user has full control of all the objects in an organization's projects, but cannot create, modify, or delete any buckets in the projects. IAM gives broad control over the project but not fine-grained control over individual buckets or projects.

Access control list:

It is authorization mechanism of Google cloud storage in which read or write access is granted to users for individual buckets or objects.

Signed URLs:

These URLs give time limited read-write access to an object through a URL the owner can generate. This can be generated

using "gsutil" or by a program. The owner only need to share the url and recipient can access the object through the URL for a specific duration of time which the owner specifies.

Signed Policy Document:

This is a document which Specify the type, size and, other upload characteristics of data which can be uploaded and can be used by website owner to allow visitors to upload files to Google file storage

6. GOOGLE CLOUD PLATFORM CONSOLE

One can use Google cloud Platform Console to perform simple storage management tasks for cloud storage. It requires only a browser to work. Some of the task include: - Enable the Google Cloud storage API for a project - Creating and deleting buckets - Uploading, downloading and deleting objects - Managing ACLs for objects and buckets.

One can use other methods like using the "gsutil" command line tool or any of the client library that supports Google Cloud Storage.

6.1. Accessing Google Cloud Platform Console

Google Cloud Platform Console requires no setup or installation and one can access it directly through a browser. There are different ways to access cloud platform console depending on the type of user:

A project member:

In order to use Google Cloud Platform Console as a project member, the person account must be added to the list. A current project owner can give the person account privileges like owner, editor, or viewer access to the project which applies to all the buckets defined in the project.

A user granted read or write access to a bucket:

In this case , a project owner gives a user read or write access to a bucket name. This is useful for the people who are not the project member but need to access the bucket. This URL will prompt the user to authenticate with a Google account if the user is not already signed in.

A user is granted read access to an object:

In this case the project owner configures an object as publicly viewable and sends the user the URL to access the object. This URL will prompt the user to authenticate with a Google account if he is not already signed in. This is different from the process of "sharing the URL publicly". In the option of "sharing the URL publicly" the URL does not need any authentication.

6.2. Using the Google Cloud Platform Console to Manage the Data

An owner can perform various actions which can be performed while accessing Google Cloud storage using console.

- To create a bucket
- Uploading data to a bucket
- Downloading data from a bucket
- Creating and using folders
- Deleting objects, folders, and buckets
- Sharing the data publicly

- Setting bucket permissions
- Setting object permissions and metadata
- Filtering objects to view
- Adding a member to a project

7. GOOGLE CLOUD STORAGE IN BIGDATA

Google cloud platform and google cloud storage is apt to handle massive tasks working on both the structured and unstructured way. Analytics and machine intelligence at web-scale have been used to solve data intensive tasks. Google BigQuery [4] is cloud platform's managed data warehouse that lets an organization economically query massive volumes. It is ideal to run Spark and Hadoop.

8. GOOGLE CLOUD STORAGE COMPETITORS

In the cloud platform business where all the market leader like AWS, Microsoft Azure, IBM Bluemix etc., Google cloud platform is also giving good competition. It is spreading its services by coming up with more regions and availability zones all across the world with cheap rates.

9. CONCLUSION

This paper has shown the key aspects of the cloud storage systems, how it works and how it is an integral part of the Google cloud platform, some core concepts of Google Cloud storage. It also showed how Google cloud storage performing with respect to Big data technology. And how it is performing in comparison to other vendors who are in the market.

REFERENCES

- [1] Kun Liua and Long-jiang Dong, "Research on cloud data storage technology and its architecture implementation," Web Page, College of Oriental Application and Technology Beijing Union University, Beijing and Software Development Department Adobe (Beijing) Corporation, Beijing, jan 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877705811065192>
- [2] Wikipedia, "Google cloud storage," accessed: 02-25-2017. [Online]. Available: <https://cloud.google.com/storage/docs>
- [3] G. Inc., "Google cloud storage," accessed: 02-25-2017. [Online]. Available: <https://cloud.google.com/storage/docs>
- [4] G. Inc., "Google bigquery," accessed: 02-25-2017. [Online]. Available: <https://cloud.google.com/bigquery/>

Apache Drill

YATIN SHARMA^{1,*}, +

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

* Corresponding authors: yatins@indiana.edu

+ HID - S17-IR-2034

Paper-001, April 12, 2017

Apache Drill is a distributed SQL engine designed to enable users to explore and analyze data stored in non-relational datastores. It enables users to query the data using standard SQL and Business Intelligence(BI) tools without having to create schemas or transform data.

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

Keywords: Apache, Drill, DrillBit, NoSQL, Query

<https://github.com/cloudmesh/sp17-i524/tree/master/paper1/S17-IR-2034/report.pdf>

1. INTRODUCTION

Apache Drill[1], inspired by Google's Dremel System, is industry's first schema-free distributed SQL Engine which allows us to write SQL queries without having the need to define and maintain schemas or transform data. It automatically understands the structure of the data. It can integrate and combine from several data sources like Hive, Hbase, Mongodb, etc in a single query and export the data to any of the reporting tools like tableau[2], Excel, etc. Apache Drill can be used without the need of defining. It allows us to bypass schema maintenance tasks as well as Extract, Transform, Load (ETL) cycles and thus saves a lot time in data processing, which earlier consumed majority of projects time.

2. BENEFITS

Apache Drill allows us to conduct research on datasets without having to be an expert in everything. It is simple and straightforward to use and follows ANSI SQL standards, so there is no need for fresh learning. All we need is knowledge of SQL to get started. Additionally, Drill allows to couple various datasets with NoSQL[3] systems and provide direct access to keys of data which can then be used to query the data using Apache Drill whether it is stored in a file or NoSQL database. From a security perspective, Drill allows to create views for higher level access on the original data and give permission to others on these views. Drill comes with built in measures to protect and secure a database. It allows users to create a view and apply filesystem security to this view based upon their needs. Drill currently supports 3 layers of impersonation.

3. ARCHITECTURE

At the core, Drill consists of Daemon service called the Drillbit [4] which is responsible for accepting requests from clients,

processing them, and returning the results to the client. The drillbit service can be run on any or all nodes of a cluster. It uses ZooKeeper[5] for all sort of communication in the clusters. ZooKeeper is responsible for cluster combination, cluster membership, leadership selection, etc. So, when a user submits a query, the ZooKeeper finds the Drillbit instance, also called foreman, which will eventually do the parsing, optimization, execution and data aggregation. All of this takes fraction of seconds to execute. Just like MapReduce, Drill uses data locality such that there is no need to bring data over the network and instead bring code to the data. It comes with a distributed SQL MPP engine to execute queries in parallel on a cluster.

3.1. Query Execution Flow

The SQL query submitted by Drill client, any JDBC/ODBC client or Rest Client, is accepted by a drillbit in a cluster. The drillbit that accepts the query becomes the driving Drillbit node, also called as a foreman, for this particular request. As a client application we can either submit to a drillbit or talk to the zookeeper which in turn will route this to an available drillbit. Once the query is received, it parses the query and tries to figure out the most optimal way to execute this query. Drill allows to do a variety of rule based as well as cost based optimization in addition to being aware of data locality while doing the query planning. Once the query plan is determined, Drill splits the query plan into a number of pieces called 'Query Plan Fragments'. The coordinated drillbit talks to the ZooKeeper and finds out what are the other drillbits that are available in the cluster. It then gets the location of the data and combines this information to determine the drillbits that can handle this Query Plan Fragments. It distributes the work to other drillbits in the cluster. Each Drillbit then does its own processing of the Query Plan Fragment and returns the result to the original drillbit(foresman) which then combines and returns the result to the client. As

shown in the diagram below, Each of the drillbit, during the execution, would be interacting with the underlying Storage System like DFS, Hbase, MongoDB, etc.

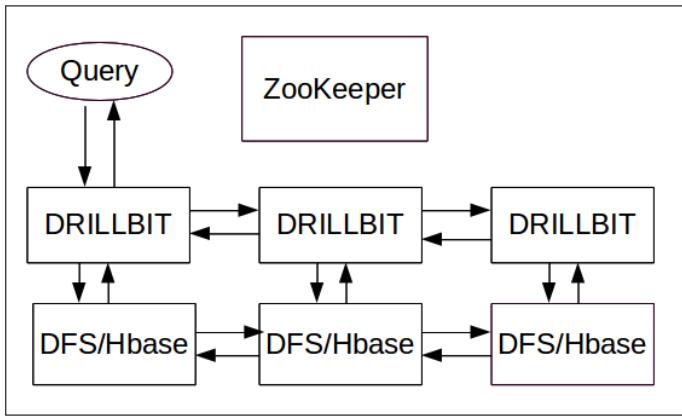


Fig. 1. Query Execution Workflow.

Figure 1. above describes the basic query execution workflow. Important thing to notice here is that each of the Drillbits are equal and there is no Master-Slave concept. So, the request from the client can go to any of the drillbit depending upon its availability. It is also a completely scalable architecture that allows to add in more drillbits as and when required[6].

3.2. Columnar Execution

Drill is Columnar in storage as well as memory. It means that it does not have to materialize the data into row format and can perform all SQL operations like join, sorting, etc directly on columnar data without having to change the data. The benefit of this is that it not only boosts the performance but also saves the memory footprint that Drill has to occupy during query execution.

3.3. Vectorized Processing

Drill provides a vectorized execution engine to offer high memory and CPU efficiencies along with rapid performance for a wide variety of analytic queries. Instead of working on single records at any given time, Drill allows CPU to work on vectors, which are arrays of values from many different set of records in table. So, Drill works on more than one record at a time to achieve efficiency.

Optimistic/In-Memory

Drill is very optimistic in the sense it assumes that the chances of failure like hardware or node failure is very uncommon during the short execution of the query. So, it tries to execute as much as possible in memory without writing anything to disk for checkpoint and recovery purposes. Drill uses the traditional Pipe-line execution model in which all the tasks are executed in parallel and in different stages in different parts of the cluster. This enables Drill to achieve high speed performance in seconds.

3.4. Datastores Support

Drill is build for non-relation datastores like Hadoop, NoSQL, and cloud storage. Currently following datastores are supported:

- **Hadoop:** All Hadoop distributions (HDFS API 2.3+), including Apache Hadoop, MapR, CDH and Amazon EMR-for Data Ingestion

- **NoSQL:** MongoDB, HBase
- **Cloud storage:** Amazon S3, Google Cloud Storage, Azure Blob Storage, Swift

3.5. Client Support

Drill currently supports following clients:

- **BI tools** via the ODBC and JDBC drivers (eg, Tableau, Excel, MicroStrategy, Spotfire, QlikView, Business Objects)
- **Custom applications** via the REST API
- **Java and C applications** via the dedicated Java and C libraries

4. COMPARISON

Drill supports various non-relational datastores including Hadoop. However, Drill takes a different approach compared to popular SQL-on-Hadoop technologies like Hive and Impala. For example, it lets users directly query self-describing data (eg, JSON) without having to create and manage schemas. Moreover, Hive is a typical batch processing framework best suitable for long-running jobs, Drill on the other hand is suitable for short-running jobs like data exploration and BI. Unlike Hive, Drill is not limited to Hadoop. Drill can also query NoSQL databases (eg, MongoDB, HBase) and cloud storage (eg, Amazon S3, Google Cloud Storage, Azure Blob Storage, Swift).

5. GETTING STARTED

Step-by-step procedures to download, install and get started with Drill can be found on-line.[7]

6. CONCLUSION

Apache Drill allows us to use familiar querying language (SQL) against different data sources. It provides a single SQL interface for self service data exploration for structured and semi-structured data Source without any IT intervention. Drill also allows us to use familiar BI tools like Tableau and Excel for data exploration without any time-consuming/expensive programming and data definition. Apache Drill is making big data analytics more accessible to wider groups of people.

ACKNOWLEDGEMENTS

The author thanks Prof. Gregor von Laszewski for his technical guidance.

REFERENCES

- [1] "Apache drill schema-free sql query engine for hadoop, nosql and cloud storage," Web Page, online; accessed 23-Feb-2017. [Online]. Available: <https://drill.apache.org/>
- [2] "Business intelligence and analytics | tableau software," Web Page, online; accessed 15-March-2017. [Online]. Available: <https://www.tableau.com/>
- [3] "Nosql databases explained | mongodb," Web Page, online; accessed 15-March-2017. [Online]. Available: <https://www.mongodb.com/nosql-explained>
- [4] "Architecture introduction - apache drill," Web Page, online; accessed 23-Feb-2017. [Online]. Available: <https://drill.apache.org/docs/architecture-introduction/>
- [5] "Apache zookeeper - home," Web Page, online; accessed 15-March-2017. [Online]. Available: <https://zookeeper.apache.org/>

- [6] "Drill query execution - apache drill," Web Page, online; accessed 23-Feb-2017. [Online]. Available: <https://drill.apache.org/docs/drill-query-execution/>
- [7] "Tutorials introduction - apache drill," Web Page, online; accessed 23-Feb-2017. [Online]. Available: <https://drill.apache.org/docs/tutorials-introduction/>

Oracle PGX

PIYUSH SHINDE^{1,*}

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

* Corresponding authors: piyushsshinde1992@gmail.com

Paper-1, April 12, 2017

Oracle PGX is a graph analysis toolkit. Graph analysis provides key insights, which play a vital role in business optimizations. Oracle PGX provides various graph analysis features like loading graphs into memory, running built-in and user-defined graph analysis algorithms, running graph pattern matching queries and mutating graphs [1]. In this paper we discuss Oracle PGX's various features, use cases, advantages and limitations.

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

Keywords: Graph Analysis, Algorithms, Shell

<https://github.com/cloudmesh/sp17-i524/blob/master/paper1/S17-IR-2035/report.pdf>

1. INTRODUCTION

A graph is a fundamental data structure that defines links between different data nodes. Graphs represent data sets in a wide range of applications. In a social graph, for example, nodes correspond to people while friendship relationships between them are represented as edges [2].

Graph analysis involves extracting information from a data-set, which is represented as a graph. Large amounts of data demands large amounts of computational power to analyse. Challenges faced during graph analysis include storage capacity for large data sets, performance variation with varying data set sizes, and efficient implementation of graph analyses algorithms.

Oracle PGX (Parallel Graph AnalytiX) is a fast, parallel, in-memory graph analytic framework. Its main features include loading graphs into memory, running built-in and user-defined graph analysis algorithms, running graph pattern matching queries and mutating graphs [1]. We will glance through these features in this paper.

Oracle PGX 2.3.1 is the latest version available as of 26th February 2017 [3].

2. INSTALLATION

To install Oracle PGX the platform requirements include a Linux or Mac OS X platform [3].

Oracle PGX can be installed in the following steps. Download Java SE 8 Development Kit (JDK8). Make sure the JAVA_HOME environment variable points to the JDK8 home directory, e.g. export JAVA_HOME=/usr/lib/jvm/java-8-oracle. Unzip the downloaded zip file e.g. unzip pgx-2.3.1-server.zip -d /opt/pgx. Change directory (cd) into the installation directory, e.g. cd \$PGX_HOME. Start the PGX shell using the command ./bin/pgx.

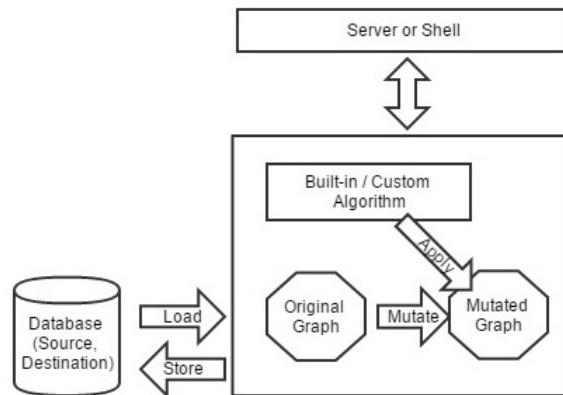


Fig. 1. PGX Overview [1]

3. USAGE MODES

PGX provides four different usage modes, depending on different user requirements [4].

3.1. Local Shell Mode

This mode enables the user to interact with a PGX shell on a local machine through a command-line shell interface. User commands are dynamically interpreted and executed by the PGX shell to invoke the PGX function. It can be used to run some quick graph analyses locally on small graph data. It can likewise be used to try different graph algorithms to explore the different interpretations of a data set.

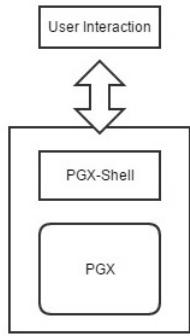


Fig. 2. PGX Local Shell Mode [4]

3.2. Remote Shell Mode

In the remote shell mode, the PGX execution engine is deployed as a RESTful application on a powerful PGX-Server which is remotely connected to the PGX-Shell from a client machine. This mode is useful for performing graph analysis on large data sets. This mode enables to connect multiple clients to the PGX-Server.

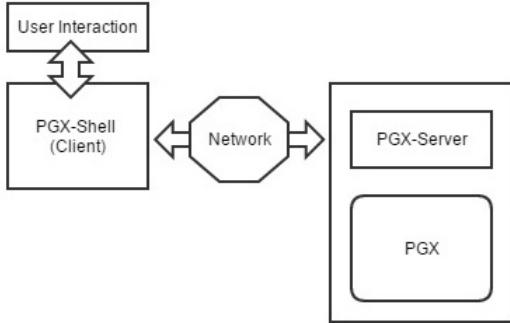


Fig. 3. PGX Remote Shell Mode [4]

3.3. Local Java Application Mode

This mode allows embedding PGX into a Java application, and using graph analysis as a part of the application's functionality.

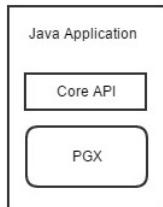


Fig. 4. PGX Locally Embedded Mode [4]

3.4. Remote Java Application Mode

This mode empowers the user to use the Java application to control the PGX-Server remotely, from a client machine. Similar to the remote shell model, this mode enables connections to the PGX-Server via multiple clients.

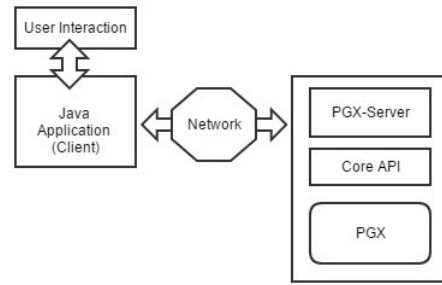


Fig. 5. PGX Remotely Embedded Mode.

4. GRAPH MUTATIONS

PGX helps users create mutated versions of the original graph, to perform algorithms for graph analyses. Since PGX does not allow mutation of the original graph, it always creates a new mutated version of the original graph.

A graph can be simplified by removing self-edges, duplicated edges, and trivial vertices.

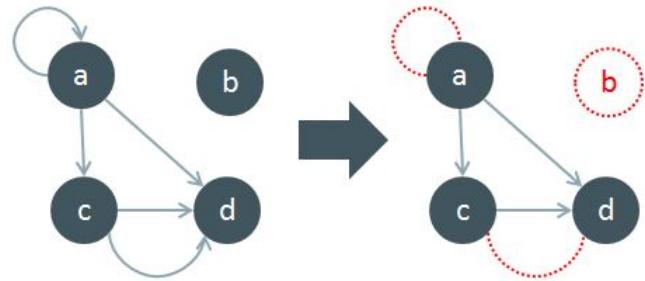


Fig. 6. Simplifying a graph. Source: Paper [5]

Graphs can also be simplified by undirecting its edges. Subgraphs and bipartite subgraphs can also be created in PGX.

5. RUNNING GRAPH ALGORITHMS ON PGX

PGX allows you to run graph algorithms in two ways.

Built-in graphs: PGX provides a set of built-in graph algorithms that are ready to use. They include computing various centrality measures, finding shortest paths, finding/evaluating clusters and components, and predicting future edges, etc. The detailed and complete list of the built-in algorithms, with their time complexity, space requirement and implementation code is available on the official website of Oracle Labs PGX.

Custom algorithms: User-defined algorithms can be programmed in PGX (excluding the ones from the built-in package), using the Green-Marl language (a Domain Specific Language) [6]. The Domain Specific Language approach provides benefits like productivity, performance and portability.

6. USE CASES

PGX can be used to solve real world graph analyses problems. Here are two such applications [7].

6.1. Detect Potential Fraud in a Healthcare System

In this example, anomalies in medical transactions were detected using PGX.

A public data set of medical transactions from CMS (United States Center for Medicare and Medicaid Services) for the year 2012 was used. It had transactions between over 880,000 medical providers and CMS, adding up to more than \$77B for 2012 [8].

To analyse the data, the data set was converted to a bipartite graph, separating health care providers from health services. An edge in the bipartite graph linked a specific health service to a health care provider. A two-hop (undirected) path existed between two health care providers, for each common health service they both provided. The more common services between two health care providers, the closer they would be to each other in the graph. The data set also specified the 'specialities' of medical providers. Vertices of medical providers of the same speciality would be closer to each other, since they would provide similar services. In case, a medical provider was exceptionally close to other providers with different speciality, it would be considered an anomaly since the medical provider would be providing services unfitting to its speciality.

Personalized PageRanks (PPR) were used to compute closeness between vertices. For each speciality a PPR score was computed. These scores were then cross-checked to see if they belonged to the current speciality. If not, the provider was considered an anomaly. This graph analysis was implemented using the built in Personalized PageRank algorithm.

The results were further investigated to identify potential frauds in the Healthcare System.

6.2. Using PGX as a Recommendation Engine

This example helps us discover the information implied by relationships through graph analytics.

A publicly available data set 'MovieLens' was used. Recommendations were generated using matrix factorization algorithm. Matrix factorization is a simple process, in which you factorize a matrix into two matrices, such that the newly generated matrices when multiplied with each other would generate back the original matrix [9].

We had a matrix of movies and users, where the values were the ratings given by the user to a movie. The aim was to recommend a bunch of movies for each user.

An array of features was synthesized, considering user information, the movies they rated and their ratings. The approach was to initialize each matrix with some values. Each user had an array of floating-point values that represented how strongly they were associated with that feature. To generate recommendations, the movie's feature array was multiplied to the user's feature array. The corresponding values were then sorted in a descending fashion and the top few items were the recommendations. This process was very fast using PGX. The results were then used for many recommendations, repeating the process periodically after new users or movies were added to the system.

7. BENEFITS AND LIMITATIONS

PGX executes graph algorithms in a fast, parallel, in-memory fashion. It provides built-in algorithms and supports user-defined algorithms. It offers an interleaved usage of graph analysis algorithms and graph pattern-matching. In addition, it provides an interactive shell application to easily execute commands from the shell command line.

PGX loads file-systems, exclusively in graph formats. It requires licensing, restricting its usage. Both of the above mentioned limitations can be removed by contacting Oracle Labs PGX [10].

8. USEFUL RESOURCES

The complete documentation of Oracle PGX, including instructions to download and install and several tutorials, in a systematic method are available on the main website of Oracle Labs PGX [11].

The paper titled "Graph Analysis – Do We Have to Reinvent the Wheel?", discusses about the alternative database technologies that provide optimized performance with regards to the dedicated graph databases. [12]

The paper on Green-Marl DSL approach titled "Green-Marl: A DSL for Easy and Efficient Graph Analysis", provides insight on the methods of using the Green-Marl DSL technique for optimizing graph analyses using user-defined algorithms. [2]

9. CONCLUSION

Numerous information is revealed from graphs. The analysis of graphs can unveil significant insights. Oracle PGX (Parallel Graph AnalytiX) is a toolkit for graph analysis. It allows users to load up their graph data and run analytic algorithms on them. It provides different usage modes which can be used as per client requirements. It allows users to implement various built-in and user-defined graphs and mutate graphs. Some classes of the built-in algorithms are community detection, path finding, page ranking, recommendation, pattern matching and influencer identification. Oracle PGX is easy to use. It can be used for precise and efficient graph analysis, and is scalable to implement graph analysis algorithms for large data sets.

10. ACKNOWLEDGMENTS

This project was a part of the Big Data and Software and Projects (INFO-I524) course. I would like to thank Professor Gregor von Laszewski and the associate instructors for their help and support during the course.

REFERENCES

- [1] Oracle Labs, "PGX Overview," Web Page, accessed: 2017-02-26. [Online]. Available: https://docs.oracle.com/cd/E56133_01/1.2.0/reference/overview/index.html
- [2] S. Hong, H. Chafi, E. Sedlar, and K. Olukotun, "Green-marl: a dsl for easy and efficient graph analysis," in *ACM SIGARCH Computer Architecture News*, vol. 40, no. 1. ACM, 2012, pp. 349–362.
- [3] Oracle Labs, "Oracle Labs PGX Downloads," Web Page, accessed: 2017-02-24. [Online]. Available: <http://www.oracle.com/technetwork/oracle-labs/parallel-graph-analytics/downloads/index.html>
- [4] Oracle Labs, "PGX Usage Modes," Web Page, accessed: 2017-02-24. [Online]. Available: https://docs.oracle.com/cd/E56133_01/2.3.1/reference/overview/usage.html
- [5] Oracle Labs, "Mutating Graphs," Web Page, accessed: 2017-02-24. [Online]. Available: https://docs.oracle.com/cd/E56133_01/2.3.1/reference/overview/mutate.html
- [6] Oracle Labs, "Running Graph Algorithms on PGX," Web Page, accessed: 2017-02-24. [Online]. Available: https://docs.oracle.com/cd/E56133_01/2.3.1/reference/overview/run.html
- [7] Oracle Labs, "Use Cases," Web Page, accessed: 2017-02-26. [Online]. Available: https://docs.oracle.com/cd/E56133_01/1.2.0/use-cases/index.html
- [8] Oracle Labs, "Detect Anomalies in a Healthcare System," Web Page, accessed: 2017-02-26. [Online]. Available: https://docs.oracle.com/cd/E56133_01/1.2.0/use-cases/healthcare-fraud/index.html
- [9] Oracle Labs, "Using PGX as a Recommendation Engine," Web Page, accessed: 2017-02-26. [Online]. Available: https://docs.oracle.com/cd/E56133_01/1.2.0/use-cases/recommendation/index.html

- [10] Oracle Labs, "About This OTN Release," Web Page, accessed: 2017-02-24. [Online]. Available: https://docs.oracle.com/cd/E56133_01/2.3.1/about/index.html
- [11] Oracle Labs, "Oracle Labs PGX," Web Page, accessed: 2017-02-24. [Online]. Available: https://docs.oracle.com/cd/E56133_01/1.2.0/index.html
- [12] A. Welc, R. Raman, Z. Wu, S. Hong, H. Chafi, and J. Banerjee, "Graph analysis: do we have to reinvent the wheel?" in *First International Workshop on Graph Data Management Experiences and Systems*. ACM, 2013, p. 7.

An Overview of the Java Message Service (JMS)

RAHUL SINGH¹

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

* Corresponding authors: rahpsing@iu.edu

April 12, 2017

Enterprise applications coordinate with each other through the exchange of asynchronous requests termed 'messages'. Messages contain precisely formatted data that describe specific business actions. Through the exchange of these messages each application tracks the progress of the enterprise. Messaging systems are expected to support fault tolerance, load balancing, scalability, and transactional support. JMS provides an intersection of messaging system common to all products. The Java Message Service (JMS) provides a standardized API that can be used for sending and receiving messages with many different messaging systems.

© 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-IR-2036/report.pdf>

1. INTRODUCTION

RMI (*Remote Method Invocation*) [1] and RPC (*Remote Procedure Call*) [2] were widely used as messaging systems in enterprise applications. With the advent of distributed systems, a need arose for a loosely coupled messaging API whose implementation wouldn't depend on knowing the details of the receiver components unlike RMI and RPC. Sun, along with several partner companies, introduced the *JMS API* [3] which defines a common set of interfaces and associated semantics that allow programs written in the Java programming language to communicate with other messaging implementations. JMS is a part of the Java Platform, Enterprise Edition, and is defined by a specification developed under the Java Community Process as *JSR 914* [4]. It is a messaging standard that allows application components based on the Java Enterprise Edition (Java EE) to create, send, receive, and read messages. It allows the communication between different components of a distributed application to be loosely coupled, reliable, and asynchronous [5]. The JMS API minimizes the set of concepts a programmer must learn in order to use messaging products but provides enough features to support sophisticated messaging applications.

2. JMS API ARCHITECTURE

A JMS application consists of the following major components. A JMS provider is a messaging system that implements the JMS interfaces and provides administrative and control features. An implementation of the Java EE platform includes a JMS provider [6]. JMS Clients are the Java language programs that send and receive messages. Any Java EE application component can act as a JMS client. Messages are the objects used to communicate

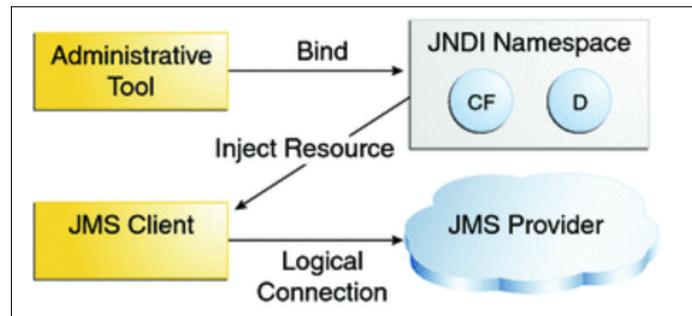


Fig. 1. Jms architecture [6]

information between the JMS clients. Administered objects are preconfigured JMS objects created by an administrator for the use of clients. JMS administered objects are 'destinations' and 'connection factories'. Destinations are the objects that a client uses to specify the destination of messages it sends and the source from where it receives them. A client uses a Connection Factory object to establish connection with a provider.

Administrative tools allow a user to bind destinations and connection factories into a *JNDI (Java Naming and Directory Interface)* [7] namespace. The JNDI is a directory service that allows Java clients to lookup services by a unique name bound to each service. A JMS client can then use resource injection to access the administered objects in the namespace and then establish a logical connection to the same objects through the JMS provider [6].

3. JMS COMMUNICATION MODELS

Before the advent of JMS, message provider systems supported either the point-to-point or the publish/subscribe approach to messaging implementations. The JMS specification provides common interfaces that enables a user to use the JMS API in a way that is not specific to either domain. A stand-alone JMS provider can implement one or both domains. A Java EE provider must implement both domains.

3.1. Point-to-point model

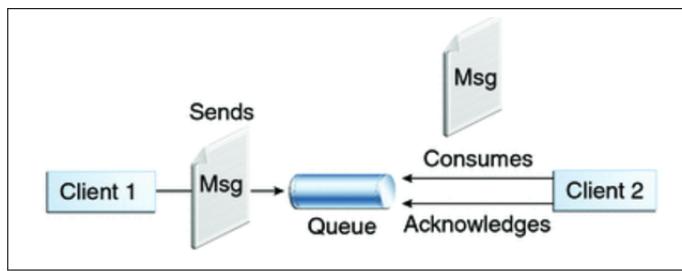


Fig. 2. Point-to-Point model [6]

Point-to-point (PTP) domains are built around the concept of message queues. Each message is addressed to a specific queue; clients extract messages from the queue(s) established to hold their messages. Queues retain all messages sent to them until the messages are consumed or expired.

3.2. Publish/Subscribe model

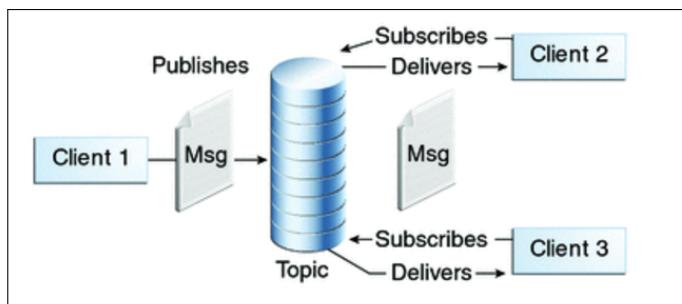


Fig. 3. Publish/Subscribe model [6]

In a publish/subscribe product or application, each message can have multiple consumers. Clients address messages to a topic, which is equivalent to a bulletin board. *Publishers* and *subscribers* are anonymous and can dynamically publish or subscribe to the content hierarchy. The system shall take care of distributing the messages arriving from a topic's multiple publishers to its multiple subscribers. Topics shall retain messages only for the time it takes to distribute them to current subscribers. Publishers and subscribers have a timing dependency. A client that subscribes to a topic can consume only messages published after the client has created a subscription, and the subscriber must continue to be active in order for it to consume messages [6].

4. IMPLEMENTING THE JMS PROGRAMMING MODEL

To implement the JMS specification, a series of objects must be created and registered to the system. The following steps comprise of developing a basic JMS system.

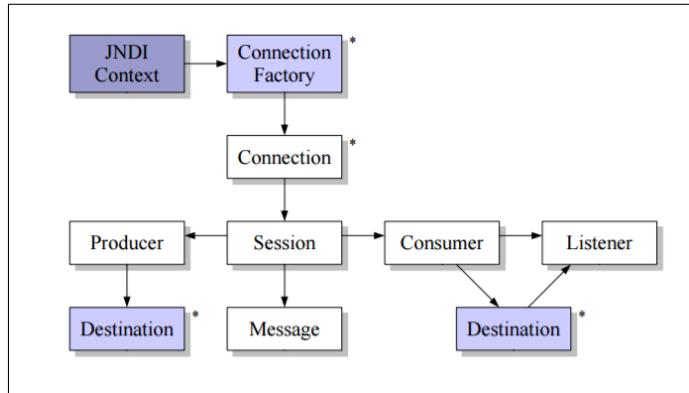


Fig. 4. Jms programming model [8]

4.1. Creating administered objects and binding resources

Creating and configuring connection factories and destinations are responsibilities of the system administrator. Alternatively, these administered objects shall also be created from the server admin console available in Glassfish or any other application servers. A JMS client can obtain access to connection factories and destinations by looking them up using JNDI. A connection encapsulates a virtual connection with a JMS provider and is used to create one or more sessions.

4.2. Creating a message object

The JMS API provides methods for creating messages of each type and for filling in their contents. For example, to create a message object the primitive `createMessage()` shall be used. Correspondingly, to set the content of a `TextMessage`, the primitive `setText()` shall be used.

4.3. Creating a producer and sending the message

A message producer is an object that is created by a session and used for sending messages to a destination. A `Session` object is used to create a `MessageProducer` for a destination. A Message producer can be created for a `Destination` object, a `Queue` object, or a `Topic` object.

Once a message producer has been created, it shall be used to send messages by using the `send()` method.

4.4. Consuming a message

A message consumer is an object that is created by a session and used for receiving messages sent to a destination. A message consumer allows a JMS client to register itself in a destination with a JMS provider. The JMS provider manages the delivery of messages from a destination to the registered consumers of the destination. Similar to a producer, a Message Consumer can be created for a `Destination` object, a `Queue` object, or a `Topic` object.

To support asynchronous operations, JMS defines the concept of a listener. A message listener is an object that acts as an asynchronous event handler for messages. The listener defines an `onMessage` method, where we shall define the actions that need to be taken once a message arrives.

A message listener shall be registered with a specific message consumer by using the `setMessageListener()` method defined by the API.

5. ANATOMY OF THE JMS MESSAGE

A JMS message carries application data and provides event notifications. A JMS message has three parts - the message headers provide metadata and routing information, the message properties are defined by the JMS client, the message body carries the payload of the message. When a message is delivered, the properties and the body of the message are made read-only [8].

5.1. Types Of Message

Message types

- TextMessage - contains String as payload.
- ObjectMessage - contains Java serializable object as its payload.
- Bytes Message -contains an array of bytes as its payload.
- Stream Message - carries stream of Java primitive types as its payload.
- Map Message - carries a set of name-value pairs as its payload.

6. MESSAGE ACKNOWLEDGEMENT AND GUARANTEED DELIVERY

Message acknowledgment is part of the protocol between the client runtime library of the JMS provider and the message server [8]. The acknowledgment protocol allows the JMS provider to guarantee delivery of messages to its consumers/subscribers.

In the point-to-point domain messages are always guaranteed to be delivered whereas, in the publish/subscribe domain messages are only guaranteed to be delivered to durable subscribers. If a consumer fails to acknowledge a message, the server considers the message undelivered and shall attempt to redeliver it.

JMS specifies three acknowledgment modes which are set when a session is created:

- AUTO-ACKNOWLEDGE - The session automatically acknowledges a client's receipt of a message after a successful return from the receive() or the onMessage() method.
- CLIENT-ACKNOWLEDGE - The client acknowledges a message by calling the message's acknowledge() method which gives the client a finer grained control.
- DUPS-OK-ACKNOWLEDGE - The session lazily acknowledges the delivery of messages which may result in the delivery of duplicate messages if the JMS provider fails [8].

7. JMS USE CASES

7.1. Multi client communication

The JMS API provides a standard API that can be implemented by any messaging system. JMS allows multiple clients, potentially written in different languages to access information and coordinate with each other.

7.2. Event Oriented Applications

Highly event oriented applications like the stock market require immediate action when the stock prices go high or low. In financial markets, it is a must for the customers to be notified whenever something important happens. The JMS API serves this purpose by providing configuration capabilities to act quickly on the occurrence of any important event.

7.3. Workflow of an entire web application

A fully functional web application has several components which at some point in time depend on each other's availability to perform their operations successfully. In a real-time environment, it wouldn't be ideal for a component to keep waiting on the result produced by another dependent component. Messaging attempts to fulfill this gap by notifying component say - A, about the status of another dependent component say - B on a timely basis, thus, allowing component A to proceed with other operations and notifying it once component B has produced a result.

7.4. Transactional Messaging

JMS API guarantees processing each message exactly once irrespective of system failures. When working with a system with multiple clusters of nodes, the API guarantees that "if any node goes offline then all the messages will be persisted for when the node comes back online" [9].

8. CONCLUSION

JMS provides a connection oriented approach to messaging in distributed systems. It provides loosely coupled communication by allowing objects to communicate with each other without knowing their implementation details, thus overcoming the drawbacks of RMI. It supports both one-to-one and many-to-many communication model and guarantees message delivery depending on the provider implementation.

REFERENCES

- [1] Wikipedia, "Java remote method invocation - wikipedia," webpage, accessed : 04-05-2017. [Online]. Available: https://en.wikipedia.org/wiki/Java_remote_method_invocation
- [2] Wikipedia, "Remote procedure call - wikipedia," webpage, accessed : 02-26-2017. [Online]. Available: https://en.wikipedia.org/wiki/Remote_procedure_call
- [3] Wikipedia, "Java message service - wikipedia," webpage, accessed : 02-26-2017. [Online]. Available: https://en.wikipedia.org/wiki/Java_Message_Service
- [4] "The java community process(sm) program," webpage, dec 2003, accessed : 02-24-2017. [Online]. Available: <https://jcp.org/aboutJava/communityprocess/final/jsr914/index.html>
- [5] "Java message service(jms)," webpage, accessed : 02-24-2017. [Online]. Available: <http://www.oracle.com/technetwork/java/jms/index.html>
- [6] "Java message service(jms)," webpage, accessed : 02-25-2017. [Online]. Available: <http://docs.oracle.com/javase/6/tutorial/doc/bncdr.html>
- [7] Wikipedia, "Java naming and directory interface - wikipedia," webpage, accessed : 04-05-2017. [Online]. Available: https://en.wikipedia.org/wiki/Java_Naming_and_Directory_Interface
- [8] S. Fischli, "Java message service," 2005. [Online]. Available: <http://www.sws.bfh.ch/~fischli/skripte/JMS.pdf>
- [9] "Apache activemq usecases," webpage, accessed : 04-05-2017. [Online]. Available: <http://activemq.apache.org/use-cases.html>

File Transfer Protocol - An Overview

SRIRAM SITHARAMAN^{1,*}

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

* Corresponding authors: srirsith@iu.edu

April 12, 2017

The advent of computer systems in the later part of 20th century facilitated the increase in the digitization of documents that were in existence. In order to share those documents and files among peers situated remotely, there raised a need for a mechanism which could facilitate file sharing. File Transfer Protocol developed in the late 1970s provided an efficient and reliable mechanism for sharing files among hosts. File Transfer Protocol (FTP) and its architecture is discussed along with the alternatives to FTP and how they fare when compared to FTP. FTP clients used for managing the file transfer at the client side is also discussed.

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

Keywords: Cloud, I524, FTP, FTP Client, Comparison, Architecture

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

CONTENTS

1	Introduction	1
2	Architecture	1
2.1	Components in FTP model	1
2.2	File System	2
2.3	File Request/Receive Process	2
2.4	FTP Commands	2
2.5	Response from Server DTP	2
3	Alternatives to FTP	3
3.1	Managed File Transfer (MFT)	3
3.2	Trivial File Transfer Protocol (TFTP)	3
3.3	Simple File Transfer Protocol (SFTP)	3
3.4	Secure Shell File Transfer Protocol (SFTP)	3
4	Customized File Transfer Protocol (FTP) mechanisms	3
5	FTP clients	3
5.1	Comparison of FTP Clients	3
6	Conclusion	4

1. INTRODUCTION

According to [1] FTP is an acronym for File Transfer Protocol. It is network protocol standard used for transferring files between two computer systems or between a client and a server. It is a part of Application layer of the Internet Protocol Suite and works along with HTTP/SSH and follows a client-server

model architecture. Secure systems asks the client to authenticate themselves using a Username and Password registered with the server to access the files via FTP. The specification for FTP was first written by Abhay Bhushan [2] in 1971 and is termed as RFC114. The current specification, RFC959 in use was written in 1985. Several other versions of the specification are available which provides firewall friendly FTP access, additional security extensions, support for IPV6 and passive mode file access respectively. FTP can be used in command line in most of the operating systems to transfer files. There are FTP clients such as WinSCP, FileZilla etc. which provides a graphical user interface to the clients to authenticate themselves (sign on) and access the files from the server.

2. ARCHITECTURE

Figure 1 shows the architecture of a model for using File Transfer Protocol. It is essential to understand the different components of the architecture to understand the flow of FTP. The next section describes in brief about the components that are part of the model architecture.

2.1. Components in FTP model

As shown in Figure 1., FTP model consists of two major components Server side FTP component and a client side FTP component. Each of those components consists of a Protocol Interpreter (PI) which takes of the respective protocol definitions for the user and server end, and a Data Transfer Process (DTP) which establishes and manages the connection between the server and the client. Apart from these two, client side FTP component has a User Interface through which the User authenticates himself with the server and requests for the files to be transferred.

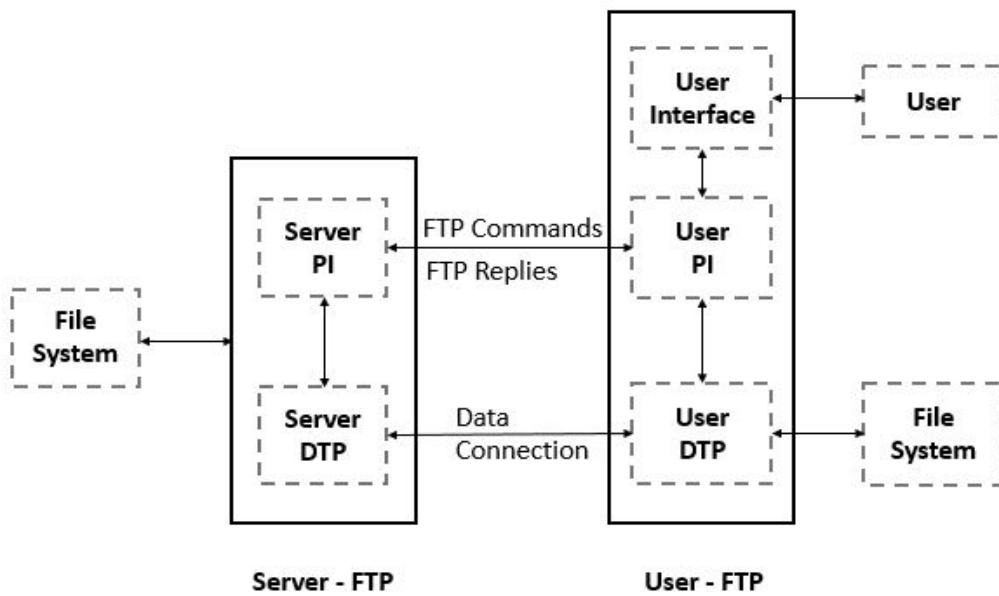


Fig. 1. Model for FTP usage [3]

2.2. File System

File system is the place where the files are located. It is present in both the client and the server end. When the user want to request a file from the server, a pathname (a character string) has to be specified as an input to the file system to locate the file. A location has be to specified by user/client in the client side for storing the file transferred by the client.

2.3. File Request/Receive Process

Once the user wants to request a file from the server, the user protocol interpreter is responsible for initiating the connection process to the server-FTP. Once the control connection is established by the user PI, it generates standard FTP commands to the server. User PI is also responsible for managing the data transfer process (DTP) as part of the file transfer process. In response to the commands sent from the user, replies are sent by the server which acts as an acknowledgement.

2.4. FTP Commands

The FTP commands specifies the following parameters for the data connection:

1. **Data port** - Port is a communication endpoint in an operating system [4]. Each computer system has ports numbered from 0 to 1023 each of which can be uniquely associated with a process. For File Transfer Protocol, the designated port number is 21.
2. **Transfer mode** - It defines the type of transfer process using which the data is sent between the client and the server. There are 3 different modes through which this can be accomplished:
 - (a) Stream mode - It is used when transferring data continuously from one end to another.

(b) Block mode - When there is a limitation on the size of data that can be sent, data is split into blocks of size (with a threshold on size) and sent across one by one according to the order of split.

(c) Compressed mode - data is compressed using a compression algorithm and then transferred to the other system.

3. **Representation type** - It defines the type of the data that is being transferred between the server to the client

- (a) ASCII mode - used if the data is of textual format.
- (b) Image mode - For transferring data in binary format as bytestreams.
- (c) EBCDIC mode - used for transferring text data between systems that follows the EBCDIC character format.
- (d) Local mode - used between systems that follow similar characteristics in data format.

4. **Structure** - Specifies the structure of the file that is sent across. It can be a: (1) file structure - when the data is continuous sequence of bytes, (2) record structure - when the data is a relational database table with sequential records on each line of the file and (3) page structure - when the data in each file is stored as independent indexed pages.

5. **Type of the file system operation** - store, retrieve, append, delete, etc.

2.5. Response from Server DTP

The user DTP should listen on the designated port mentioned in the parameters sent during the FTP commands and the server DTP initiates the data transfer process as per the specified parameters. The end of the data transfer process is indicated by the

Client	Creator	Software license	Over 2GB?	Interface	Windows	OS X	Linux	BSD	Unix
Cyberduck	David V. Kocher	GPL	Yes	GUI	Yes	Yes	No	No	No
FileZilla	Community	GPL	Yes	GUI	Yes	Yes	Yes	Yes	Yes
WinSCP	Martin Přikryl	GPL	Yes	GUI/CLI	Yes	No	No	No	No
CuteFTP	Globalscape	Proprietary	Yes	GUI	Yes	Yes	No	No	No
Classic FTP	NCH Software	Proprietary	Yes	GUI	Yes	Yes	No	No	No

Table 1. Comparisons of FTP clients [5]

Client	FTP	FTP over SSH	SFTP	Compression	Remote Compression	Resume Download	Passive mode
Cyberduck	Yes	No	Yes	No	Yes	Yes	Yes
FileZilla	Yes	Yes	Yes	No	No	Yes	Yes
WinSCP	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CuteFTP	Yes	Yes	Yes	Yes	No	Yes	Yes
Classic FTP	Yes	No	No	No	?	?	?

Table 2. Comparison of Functionalities - FTP clients [5]

End-of-File (EOF) sequence which indicates the end of the file being transferred. The data transfer process, in general is made through a full duplex connection. A duplex connection [6] is defined as a communication channel mechanism when the systems present at either end of the channel can communicate with each other in both directions. Full duplex is when the communication from both ends can occur simultaneously.

3. ALTERNATIVES TO FTP

After its arrival in 1971 termed under RFC114 [2], FTP has underwent multiple revisions in its usage specification and the current one is defined under RFC959 [3]. Though FTP is the standard file transfer mechanism that can be used in building applications, it is advisable to use modified versions of FTP when it comes to enterprise level applications.

3.1. Managed File Transfer (MFT)

MFT [7] is a secured file transfer protocol used by enterprises as an alternative to FTP. Some of the key features that makes MFT better than FTP include its ability to provide high secured connectivity, reporting of status of file transfer, auditability, payload management , monitoring of performance metrics [5] like time and latency of file transfer etc.

3.2. Trivial File Transfer Protocol (TFTP)

TFTP [8] is a type of file transfer protocol which the transfer of files between a system and multiple remote systems in parallel. It is mainly used for booting of systems remotely from a local area network.

3.3. Simple File Transfer Protocol (SFTP)

Simple File Transfer Protocol (SFTP) [1] (not to be confused with Secure Shell File Transfer Protocol) is a unsecured file transfer protocol with complexity between FTP and TFTP. On account of it being a simple protocol, it's command set comprises of only

11 commands. It is also known for supporting login with user id and password.

3.4. Secure Shell File Transfer Protocol (SFTP)

SSH File Transfer Protocol (SSH FTP or SFTP) [9] is a file transfer mechanism that offers the same set of commands and functions as FTP but enables the data transfer with a secured shell connection. SSH is a encrypted network protocol for operating over an unsecured network [10].

4. CUSTOMIZED FILE TRANSFER PROTOCOL (FTP) MECHANISMS

McWilliams in his patent [11] proposed a modified version of file transfer protocol which provides the advantages that the files are transferred at a time specified by the client and that only those files which have been selected by the client are transferred. Bahlmann et.al has specified an on the fly version of FTP [12] which enables the creation of files on the fly based on the parameters sent through the FTP connection.

5. FTP CLIENTS

An FTP client is a software which uses the FTP protocol to transfer files to and from a remote computer. FTP clients provide a Command Line Interface (CLI) or a Graphical User Interface (GUI) for the user to authenticate themselves with the server and request for file transfer.

5.1. Comparison of FTP Clients

As mentioned in the previous section, FTP client assists the user in the process of file transfer from the client. There are several FTP clients available in the market, open source as well as proprietary. Table 1 and Table 2 shows the five selected FTP

clients and the characteristics of them with respect to file transfer [13].

One of the important feature any FTP client is it's ability to transfer large files. According to [13], the five clients listed here as well most of the ftp clients available in the market supports large file transfer. And the next important factor is the ftp client's ability to incorporate different file transfer mechanisms. While it is clear from the Table 2 that all of the clients supports FTP, the basic transfer mechanism. According to [13], FTP is supported by almost all of clients available, while SSH FTP is supported only by a few. Since a secured shell connection for data transfer is required to support SSH FTP, it is understandable that most of these versions of ftp clients are sticking to the standard file transfer mechanisms. Table 2 also points out characteristics like if data compression is allowed (at both ends), if resume option is available for file transfer, and if passive mode (when the client is protected by a firewall) option is available. [13] lists several other ftp clients open source and proprietary and the characteristics of them.

6. CONCLUSION

A brief introduction was given to File transfer protocol and its architecture which gives a clear picture of the flow involved in the file transfer process between a client and a server. Parameters involved in the FTP command from the client to the server, and the range of values it can take depending on the use case was explained. Different alternatives to FTP was put forth, and how each of them differs and performs when compared to FTP is discussed. Finally, the notion of FTP client is introduced and FTP clients were compared across different parameters.

REFERENCES

- [1] Wikipedia, "Ftp wikipedia," Web Page, feb 2017, accessed: 2017-02-02. [Online]. Available: https://en.wikipedia.org/wiki/File_Transfer_Protocol
- [2] J. Reynolds and J. Postel, "Rfc114 specification," Web Page, oct 1985, accessed: 2017-02-02. [Online]. Available: <https://tools.ietf.org/html/rfc114>
- [3] A. Bhushan, "Rfc959 specification," Web Page, apr 1971, accessed: 2017-02-02. [Online]. Available: <https://tools.ietf.org/html/rfc959>
- [4] Wikipedia, "Port (computer networking)," Web Page, jan 2017, [Online; accessed 26-February-2017]. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Port_\(computer_networking\)&oldid=758985279](https://en.wikipedia.org/w/index.php?title=Port_(computer_networking)&oldid=758985279)
- [5] Wikipedia, "Performance metric," Web Page, dec 2016, [Online; accessed 26-February-2017]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Performance_metric&oldid=753635114
- [6] Wikipedia, "Duplex (telecommunications)," Web Page, feb 2017, accessed: 2017-02-20. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Duplex_\(telecommunications\)&oldid=767269511](https://en.wikipedia.org/w/index.php?title=Duplex_(telecommunications)&oldid=767269511)
- [7] Wikipedia, "Managed file transfer," Web Page, nov 2016, [Online; accessed 26-February-2017]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Managed_file_transfer&oldid=749730447
- [8] Wikipedia, "Trivial file transfer protocol," Web Page, feb 2017, [Online; accessed 26-February-2017]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Trivial_File_Transfer_Protocol&oldid=765170292
- [9] Wikipedia, "Ssh file transfer protocol," Web Page, nov 2016, [Online; accessed 26-February-2017]. [Online]. Available: https://en.wikipedia.org/wiki/SSH_File_Transfer_Protocol
- [10] Wikipedia, "Secure shell," Web Page, feb 2017, [Online; accessed 26-February-2017]. [Online]. Available: https://en.wikipedia.org/wiki/Secure_Shell
- [11] G. McWilliams, "High level control of file transfer protocol with capability for repeated transfer attempts," Web Page, aug 2001, uS Patent 6,272,547. [Online]. Available: <https://www.google.com/patents/US6272547>
- [12] B. Bahlmann, B. Field, and C. Melle, "On-the-fly trivial file transfer protocol," Web Page, jan 2001, uS Patent 6,170,008. [Online]. Available: <https://www.google.com/patents/US6170008>
- [13] Wikipedia, "Comparison of ftp client software," Web Page, feb 2017, [Online; accessed 26-February-2017]. [Online]. Available: https://en.wikipedia.org/wiki/Comparison_of_FTP_client_software

Introduction to H2O

SUSHMITA SIVAPRASAD¹

¹ School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
* sushsiva@umail.iu.edu

April 12, 2017

Machine learning and data mining have been used in many data driven industries. H2O is a platform using for performing machine learning and predictive analytics for large scale data using cloud. When the data that is generated is large scale and is in terabytes, H2O serves a very important purpose of being able to accurately predict using different algorithms and also using different programming languages through APIs. This paper gives an introduction to H2O, how this platform has impacted various industries across several domains with improved accuracy and reduced processing time. Different use cases of the H2O platform has been explained as well.

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

Keywords: machine learning, data mining,predictive analytics,cloud

<https://github.com/SushmitaSivaprasad/sp17-i524/tree/master/paper1/S17-IR-2038/report.pdf>

1. INTRODUCTION

H2O is an open source platform that is used to create machine learning and predictive analytics models on big datasets. It is mainly written in Javascript but have APIs for R , Python, Excel, Tableau and Flow[1]. The main algorithms implemented on the datasets are deep learning, gradient boosting , generalized linear model, distributed random forest and k-means. The algorithms implemented on the big datasets is read in a parallel manner and is then distributed and stored in memory in a compressed column format. H2O also has an inbuilt intelligence to detect and support the process of obtaining data and importing them for immediate use or storing in a database which can be obtained from various sources in different formats[1].

2. HOW DOES H2O WORK?

H2O is used on large dataset , usually in the range of terabytes due to it's speed in processing the data. A company might have their dataset stored on big data systems such as Hadoop. On analyzing a data we usually choose a smaller sample dataset rather than the entire dataset to build a model due to the large processing time involved. H2O has the advantage of being able to use the entire dataset to run the algorithm on a larger dataset, the more the data we are able to analyse better the predictions would be[2]. Suppose a business is trying to understand the best product placement for optimal customer enagement, the model would be created based on the dataset formed collecting information about the interactions of the customers on the website. H2O is used to model the data with multiple algorithms using more than one machine at the same time, this way we don't have

to sample the data[2]. H2O is also used to score hundreds of models in nano seconds to check which algorithm works better for that dataset.

The requirements for running H2O on a server can be obtained from the documentation provided on the H2O website[3].

3. ARCHITECTURE

The H2O architecture consists of different components which combine together to form the H2O software stack. We can divide the H2O architecture into 3 different components, top section includes all the REST API clients, middle includes the Network Cloud and the bottom section contains the different components that run within an H2O JVM process[4]. The top section contains the programming languages that can be used on the big dataset here. The REST API clients communicate with the H2O with the help of a socket connection[4]. The Network cloud consists of the different inbuilt algorithms to create the necessary model on the data, this can also contain a customized customer algorithm to analyze the required dataset and produce the desired outputs.

3.1. REST API Clients

- Javascript: The H2O Web UI is written in the javascript language.
- R: Using the H2O R package known as 'library(h2o)' the users can write their own R functions.
- Python: The Python scripts must use the REST API directly as a library for this has not been established yet but would

be released in the future.

- Excel: H2O provides an H2O worksheet for Microsoft Excel, which allows us to import the big datasets and run the algorithms.
- Tableau: The users of H2O may pull results from H2O to create visualizations in Tableau.
- Flow: This is a notebook style WEB UI for H2O[4].

3.2. JVM Components

The H2O cloud can consist of two or more nodes which can contain a single JVM process. Each JVM process consists of language, algorithm and infrastructure (manages the resources management such as memory and CPU)[4].

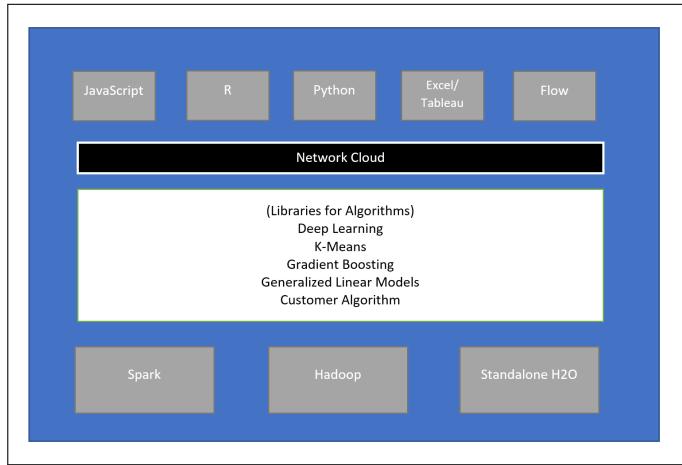


Fig. 1. H2O Architecture[4]

4. SUPPORTING PRODUCTS OF H2O

4.1. Sparkling Water

It is a spark application which is launched inside a Spark executor. It allows to initialize H2O service on every node inside a spark cluster resulting in the access of data which is stored in the data structure of Spark and H2O[5].

4.2. Steam

It is a platform that does the entire process of building and deploying predictive applications on a large scale dataset. It also provides an interface to monitor the memory usage and the cluster performance[6]. Thus Steam acts as an AI engine that adds cluster management, model management and model deployment to the H2O stack[7].

4.3. Deep Water

Deep Water integrates GPU deep learning frameworks such as TensorFlow, MXNet and Caffe and accelerates the deep learnings on the GPUs by 10-75x. It supports hyper-parameter optimization, cross validation and automatic model tuning[8].

5. USED CASES

5.1. CapitalOne

CapitalOne, a fortune 500 bank with 960 banks and 2000 ATMs accumulates terrabytes of information in real time on customer

information and financial processing. H2O software was implemented in their systems and was used to reduce the process time of the algorithm implementations over the large data sets[9]. Different algorithms can be applied to find which one works best due to reduction in time consumed in processing these large datasets. A large number of hard and soft metrics were evaluated as well using machine learning frameworks[9]. Using H2O they were able to process data received from credit cards the moment they were swiped, using this information new offerings were marketed to the customer based on their spending habits.

5.2. MarketShare

The company MarketShare have implemented H2O to optimize budgeting for marketing. Since marketing approaches are data driven features , predictive analytics under H2O was used to give a comparison on how the current state of the marketing budget is and how much is the predicted revenue[10]. Using H2O, solutions were generated as to what are the changes that can be made to improve the current projection and what an improved projection will look like. MarketShare was able to go on the cloud and use as much machines as required and get desired outputs on the large datasets. They use 25 machines for all of their clients to process the data and were able to expand the scalability of the dataset.If their datasize increases by x amount then they would add y more machines to solve the problem[10]. Scaling across lot of nodes is critical to their business as the company deals with digital log data and irrespective of the complexity of the modelling and the huge size of the data[10]. MarketShare was able to generate marketing plans and what-if cases based on the information from their customers using different predictive analytics models.

6. RELEVANT RESOURCES

H2O has an open source platform and hence has a community for support.

- Step by step instructions with documentation and videos have been provided for installation and to understand the work flow of H2O[1].
- Free online training videos are provided on the main webpage [11].
- H2O documentation is available on their website [1].
- h2ostream is an open source google group where H2O users can post questions and get answers.
- They have built an online community at[12] which is a discussion platform.
- They also conduct conferences around the year in United States for users to interact among one another and update new releases and happenings in the big data community[13].

7. CONCLUSION

Being an open source platform it gives user the flexibility to solve the problems. It is easy to set up and has a smooth installation and usage feature due to its interface with familiar programming environments using APIs[14]. Models can also be inspected during training. It can process any format of file , it can even connect to data from HDFS, S3 ,SQL and NoSQL

data sources[14]. It has a large scalability hence allowing large datasets to be analyzed by using multiple machines. It also conducts a real time data scoring for accurate predictions[14].

8. ACKNOWLEDGEMENT

A very special thanks to Professor Gregor von Laszewski and the teaching assistants Miao Zhang and Dimitar Nikolov for all the support and guidance in getting this paper done and resolving all the technical issues faced. The paper is written during the spring 2017 semester course I524: Big Data and Open Source Software Projects at Indiana University Bloomington.

9. AUTHOR BIOGRAPHY

Sushmita Sivaprasad is a graduate student in Data Science at Indiana University under the department of Informatics and Computing. She had completed her bachelors in Electronics and Communication from SRM University , India and her master's in International Business from Hult International Business School, UAE.

REFERENCES

- [1] "H2O Documentation," Web Page, Sep. 2016. [Online]. Available: <https://h2o-release.s3.amazonaws.com/h2o/rel-turing/7/docs-website/h2o-docs/welcome.html>
- [2] H2O.ai, "Oxdata H2O Explainer Video," Web Page, Aug. 2013. [Online]. Available: https://www.youtube.com/watch?v=UGW3cT_cZLc
- [3] "Requirements of H2O," Web Page, Sep. 2016. [Online]. Available: <https://h2o-release.s3.amazonaws.com/h2o/rel-turing/7/docs-website/h2o-docs/welcome.html#requirements>
- [4] "H2O Architecture," Web Page, Sep. 2016. [Online]. Available: <https://h2o-release.s3.amazonaws.com/h2o/rel-turing/7/docs-website/h2o-docs/architecture.html>
- [5]
- [6]
- [7] T. W. Dinsmore, "H2O.AI RELEASES STEAM," Web Page, Oct. 2016. [Online]. Available: <https://thomaswdinsmore.com/2016/10/12/h2o-ai-releases-steam/>
- [8]
- [9] Brendan Herger, "Capital One on Machine Learning using H2O , " Youtube Video, Jan. 2016. [Online]. Available: <https://www.youtube.com/watch?v=L6a8oITd2L8>
- [10] Prateem Mandal, "MarketShare turns to H2O for Digital Marketing Analytics," Youtube Video, Jan. 2016. [Online]. Available: <https://www.youtube.com/watch?v=L6a8oITd2L8>
- [11] "Learn H2O," Web Page, Sep. 2016. [Online]. Available: <http://learn.h2o.ai>
- [12] "H2O Community," Web Page, p. 32. [Online]. Available: <https://community.h2o.ai/index.html>
- [13] "H2O Meetups," Web Page. [Online]. Available: <http://www.h2o.ai/events/>
- [14] "Why H2O," Web Page, Sep. 2016. [Online]. Available: <http://www.h2o.ai/h2o/>

Tajo: A Distributed Warehouse System for Large Datasets

SAGAR VORA¹

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

April 12, 2017

There has been an increase in the volume of relational data generated today through a large number of sources. The large volume of data forces us to find solutions which can cope with them. Recently several hybrid approaches like HadoopDB, Hive, etc have been introduced to handle this large data. Although these have been successful in handling large data, but their architecture makes them inefficient to handle suboptimal execution strategies. Therefore, in order to solve the above problem, Apache has developed Tajo, a relational, distributed data warehouse system on large clusters. It uses Hadoop Distributed File System (HDFS) for storing data and has its own query execution engine instead of the MapReduce framework.

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

Keywords: HadoopDB, Hive, Distributed, MapReduce, Cluster, Query processing

<https://github.com/cloudmesh/sp17-i524/raw/master/paper1/S17-IR-2041/report.pdf>

1. INTRODUCTION

In this Big Data era, Hadoop [1] MapReduce [2], has been used for processing large-scale data sets. To handle a large amount of data, several hybrid approaches have been integrated with Hadoop and parallel databases. However, these cannot avoid the choice of suboptimal execution strategies because of their architecture, which led to the development of Tajo. Tajo [3] is a relational, distributed data warehouse system which runs on shared-nothing clusters. It uses Hadoop Distributed File System (HDFS) as the storage layer and has its own query execution engine instead of the MapReduce framework. A Tajo cluster consists of one master node and a number of workers. The master is responsible for the query planning and coordination among the workers.

The internals of Tajo has three main steps:-

1. Each worker has a local query engine that executes a directed acyclic graph (DAG) of physical operators. A DAG of operators can accommodate multiple sources of input and can be pipelined within the local query engine. Each worker generates query execution plan that can employ the existing query evaluation technique [4] [5] which lie in the database community.
2. Tajo can make use of various repartition methods specialized for specific queries. Consider joining two relations which are already sorted on the join key, Tajo needs to repartition only one relation to workers in which the corre-

sponding part of another relation resides.

3. In Tajo, a physical plan that a worker executes is generated at runtime according to the available resources like memory, processing capability, etc of the workers. Workers can simultaneously execute different physical plans in the same phase which allows Tajo to maximize the utilization of the resources.

2. ADVANTAGES OF TAJO OVER MAPREDUCE/HIVE

The limitations [6] of Hadoop MapReduce-Hive [7] technology due to its architecture led to the development of Tajo. These limitations are as follows:-

1. Single Source Input: The join operation in relational data warehouses integrates heterogeneous data sets from multiple sources, but MapReduce supports only a single input source, the join operation is performed by dividing input relations through one map reduce job only which doesn't produce optimal results.
2. Fixed Data Flow: The 3 phases of MapReduce which are the map, shuffle, and sort and reduce. The join operation and aggregation are performed in shuffle and sort phase. So it always follows a fixed execution flow leaving no room for any optimization needed in the processing. Sometimes processing huge datasets need to follow some hybrid map reduce flow which is not applicable to the Hadoop MapReduce framework.

3. Separate Storage: Some hybrid approaches using the database layer require separate data storages for data distribution and processing. The partitioned data must be loaded into the database layer for performance benefits. In Hadoop, it takes a long time for data loads causing overhead other running workloads in the cluster because of a separate data storage layer on the HDFS.

Tajo's system architecture has been designed in such a way to overcome the above shortcomings.

3. SYSTEM ARCHITECTURE

Figure 1 shows the system architecture of Tajo.

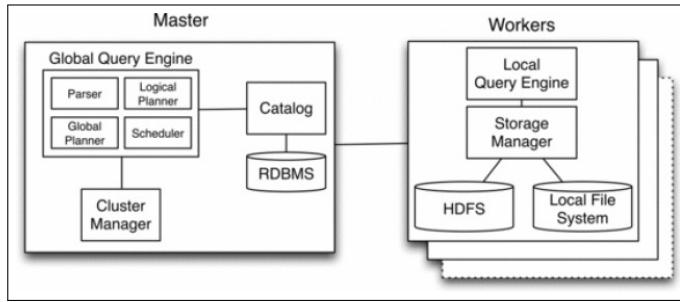


Fig. 1. [6] System Architecture of Tajo

A Tajo cluster consists of one master node and a number of workers. They are connected to one another via high-speed network. Tajo uses HDFS as the storage layer. HDFS consists of one name node (master) and a number of data nodes. A Tajo worker and HDFS data node can run on the same physical machine.

3.1. Storage Layer

Tajo uses Hadoop Distributed File System (HDFS) as the basic storage layer. The data in HDFS is distributed automatically among the cluster nodes. A local query engine of each worker scans data sets on HDFS. Tajo takes input data from HDFS and later outputs the results back into HDFS. Tajo even has a local file system and its storage manager provides interfaces to the operating system. The physical operators can process data sets on either HDFS or on local file system.

3.2. Master

The master is responsible for planning queries and coordinating the activities of workers. The master includes four components, cluster manager, catalog, global query engine, and history manager. All cluster nodes report their resource information like the number of available processors, memory usages, and remaining disk spaces periodically to the cluster manager. The catalog maintains various metadata, such as tables, schemas, partitions, functions, indices, and statistics. Since the metadata are frequently accessed by the global query engine, they are stored in a conventional RDBMS [8]. The global query engine builds a global plan based on the metadata of tables and cluster information which are provided from the catalog and the cluster manager respectively. Finally, the history manager records the metadata of the executed queries, including query statements, statistics, and logical plans.

3.3. Worker

A worker has a query execution engine that performs assigned query. A query unit contains a logical plan and fragments. A fragment is chunk information of an input relation. During execution, a worker sends periodically the reports of the running queries and the resource status to the master which aids in failure. The local query engine communicates with the storage manager which has its data sets stored on HDFS and has a local file system storage as well.

4. QUERY PROCESSING

Tajo has an SQL-like query language, called Tajo Query Language (TQL) [9] which supports most of the SQL commands. In addition to this, TQL also supports two kinds of variables to indicate a scala value and a temporary table respectively.

4.1. Query Planning

Figure 2 shows the steps involved in query transformation.

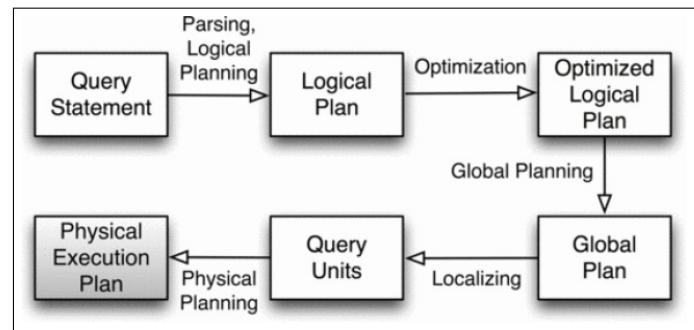


Fig. 2. [6] Query Transformation

Tajo has multiple steps to transform a query statement to physical execution plans. When a user submits a query, the global query engine parses the query into an abstract syntax tree and compiles it into a logical plan. The query optimizer finds the best logical plan which is similar to the original logical plan. This is the optimized logical plan which is transformed into a global query plan. In this step, some logical operators like group-by, sort, and join are transformed into two phases with appropriate repartition methods. Usually, the first phase computes local data on each node, and the intermediate data are range-partitioned or hash-partitioned on the specified keys (e.g., sort keys, grouping keys). The second phase computes the partitioned data on each node. As a result, a global query plan forms of a directed acyclic graph (DAG) of subqueries, which represents a data flow. Based on the physical information of the table, a subquery is localized into a number of query units, each of which is a basic unit of a query executed by a worker. Then, the global query engine schedules the query units along with the DAG of the global plan.

When a worker receives a query unit, it transforms the logical plan of the query unit into a physical execution plan according to its own computing resources. As a result, the query units of the same subquery can lead to different physical execution plans on different workers.

4.2. Query Execution

For input and output of data, Tajo has scanners and appenders. A scanner reads input data from HDFS or local file system,

whereas an appender writes output data to either of them. In the current implementation, the scanners/appenders of Tajo support CSV and row-based binary file formats. Since a worker executes a DAG of physical operators and Tajo can use various repartition methods, it can do more optimized and efficient query processing.

5. EXPERIMENT

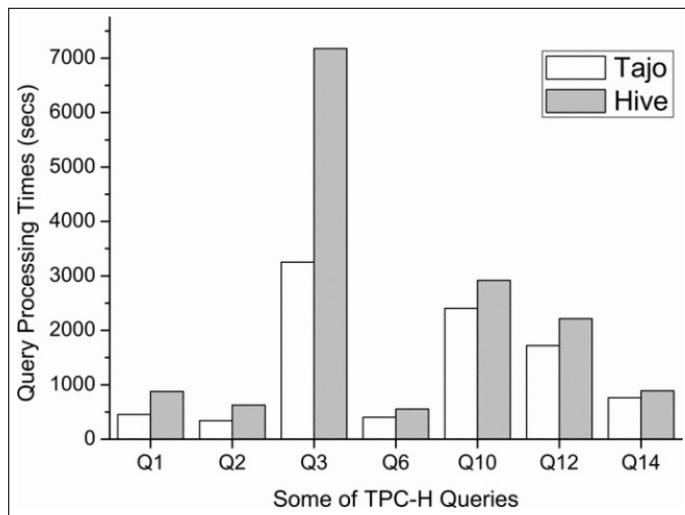


Fig. 3. [6] Performance Evaluation using TPC-H

In [6], the performance of Tajo and Hive has been compared by using 1TB TPC-H benchmark [10] set. Figure 3 shows the experimental results. For this experiment, an in-house cluster of 32 nodes, each of which is equipped with 16GB RAM, 4TB HDD, and an Intel i5 quad core CPU. The x-axis means the TPC-H queries and the y-axis indicates the processing times. The results show that the time take by Tajo to execute SQL queries is less than Apache Hive on the top of MapReduce.

6. TAJO SHELL

Tajo provides a shell utility named Tsql [9]. It is a command-line interface where users can create or drop tables, inspect schema and query tables and can execute other sql commands.

For example: bin/tsql [options] [database name]
If a database name is given, tsql connects to the database at startup time else connects to default database.

7. USE CASES OF TAJO

7.1. Data warehousing and analysis

Korea's SK Telecom firm [11] ran Tajo against 1.7 terabytes worth of data and found it could complete queries with greater speed than either Hive or Impala [12].

7.2. Data discovery

The Korean music streaming [11] service Melon uses Tajo for analytical processing. Tajo executes ETL (extract-transform-load process) jobs 1.5 to 10 times faster than Hive.

8. CONCLUSION

So, with the use of Tajo, efficient data processing and analysis is possible. It processes data faster than the MapReduce Hive framework and provides distributed data warehouse capabilities. By supporting SQL standards and advanced database techniques, Tajo allows direct control of distributed execution and data flow across a variety of query evaluation strategies and optimization opportunities. Lastly, being compatible with ANSI/ISO SQL standard, JDBC [13] driver and various file formats such as CSV, JSON, etc extends its capabilities further.

ACKNOWLEDGEMENTS

I would like to thank my professor Gregor von Laszewski and all the associate instructors for their constant technical support.

REFERENCES

- [1] "Apache hadoop," Web Page, accessed: 2017-2-26. [Online]. Available: <http://hadoop.apache.org/>
- [2] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008, published as an Article. [Online]. Available: <http://doi.acm.org/10.1145/1327452.1327492>
- [3] "Apache tajo," Web Page, accessed: 2017-2-26. [Online]. Available: <https://tajo.apache.org>
- [4] G. Graefe, "Query evaluation techniques for large databases," *ACM Comput. Surv.*, vol. 25, no. 2, pp. 73–169, Jun. 1993, published as an Article. [Online]. Available: <http://doi.acm.org/10.1145/152610.152611>
- [5] D. Kossmann, "The state of the art in distributed query processing," *ACM Comput. Surv.*, vol. 32, no. 4, pp. 422–469, Dec. 2000, published as an Article. [Online]. Available: <http://doi.acm.org/10.1145/371578.371598>
- [6] H. Choi, J. Son, H. Yang, H. Ryu, B. Lim, S. Kim, and Y. D. Chung, "Tajo: A distributed data warehouse system on large clusters," in *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. San Francisco, CA: Institute of Electrical and Electronics Engineers (IEEE), Apr. 2013, pp. 1320–1323. [Online]. Available: <http://ieeexplore.ieee.org/document/6544934/>
- [7] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, S. Antony, H. Liu, and R. Murthy, "Hive - a petabyte scale data warehouse using hadoop," in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*. San Francisco, CA: Institute of Electrical and Electronics Engineers (IEEE), Mar. 2010, pp. 996–1005. [Online]. Available: <http://ieeexplore.ieee.org/document/5447738/>
- [8] "What is rdbms," Web Page, accessed: 2017-4-11. [Online]. Available: <http://www.javatpoint.com/what-is-rdbms>
- [9] "Apache tajo," Web Page, accessed: 2017-2-26. [Online]. Available: <http://tajo.apache.org/docs/0.8.0/cli.html>
- [10] "What is tpc-h," Web Page, accessed: 2017-4-11. [Online]. Available: <http://www.tpc.org/tpch/>
- [11] "Apache tajo," Web Page, accessed: 2017-2-26. [Online]. Available: https://www.tutorialspoint.com/apache_tajo/apache_tajo_introduction.htm
- [12] M. Kornacker and J. Erickson, "Cloudera impala: Real-time queries in apache hadoop, for real," Blog, Oct. 2012, accessed: 2017-4-11. [Online]. Available: <http://blog.cloudera.com/blog/2012/10/cloudera-impala-real-time-queries-in-apache-hadoop-for-real/>
- [13] "What is jdbc," Web Page, accessed: 2017-4-11. [Online]. Available: <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>

Allegro Graph

DIKSHA YADAV^{1,*,+}

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

* Corresponding authors: yadavd@umail.iu.edu

+ HID - S17-IR-2044

Paper-001, April 12, 2017

Allegro Graph is a database technology that turns complex data into actionable business insights. It is a database technology that is used for analyzing complex information with relationships in the data which cannot be represented or analyzed by traditional database technologies. So, when our data contains relationships in it, we cannot use traditional database. We have use some advanced technology, one of which is "Allegro Graph".

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

Keywords: Allegro, Graph, triples, SPARQL, RDF

<https://github.com/diksha2112/sp17-i524/tree/master/paper1/S17-IR-2044/report.pdf>

1. INTRODUCTION

As stated by the developer of AllegroGraph, “Franz”, “AllegroGraph is a database technology that enables businesses to extract sophisticated decision insights and predictive analytics from their highly complex, distributed data that can’t be answered with conventional databases, i.e., it turns complex data into actionable business insights.” [1] It can be viewed as a closed source database that is used for storage and retrieval of data in the form of triples (triple is a data entity composed of subject-predicate-object like “Professor teaches students”).

Information in a triplestore is retrieved using a query language. Query languages can be classified into database query languages or information retrieval query languages. The difference is that a database query language gives exact answers to exact questions, while an information retrieval query language finds documents containing requested information. [2]

Triple format represents information in a machine-readable format. Every part of the triple is individually addressable via unique URLs — for example, the statement “Professor teaches students” might be represented in RDF(Resource Description Framework) as <http://example.namehashProfessor12http://xmlns.com/foaf/0.1/teacheshttp://example.namehashstudents>. Using this representation, semantic data can be queried and reasoned unambiguously [3]

2. ARCHITECTURE

AllegroGraph has the REST protocol architecture which is a superset of the Sesame HTTP Client. Adapters for various languages including Sesame Jena, Sesame Java, Python using the Sesame signatures, and Lisp are supported. Other than this,

Open Source Adapters for Csharp, Clojure, Ruby, Scala, and Perl are available. We can view the architecture of AllegroGraph in visualization 1 [1]

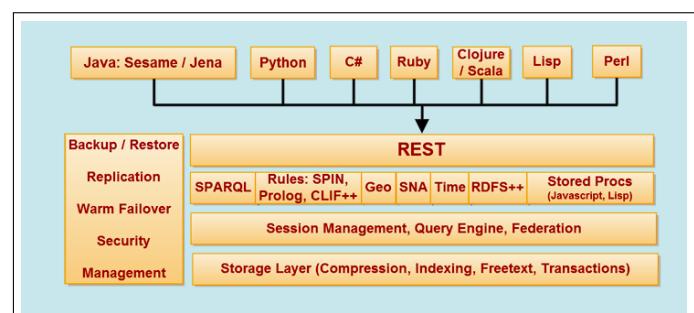


Fig. 1. Allegro Graph Architecture

3. FEATURES

1) RDF-Structure of data is maintained while exchange of information.

2) SPARQL-It is the standard query language used for writing complex structured, semi-structured or semi structured queries. It works excellently for changing databases or databases in which relationship is included in data.

3) Advanced queries-Advanced queries can be run on Allegro graph that cannot be executed by conventional relational databases and even difficult to execute with Hadoop like Geospatial (location), Temporal (time difference), Social Networking (relationships) and library (activities).

4) Linked Open Data-Allegro graph has the ability to link to the data elements of a large number of Publically- published semantic graph databases.

5) ACID Database model-Allegro graph adheres to full ACID database model and data integrity.

6) Magic predicates-Allegro graph uses predicates for better representation of real world which is helpful in processing of complex data and better predictions in business.

7) String support-Unlimited length is available for data elements to support lengthy URL references

8) Dereferenceable URLs-Reach of graph is extended to other database sources. So, entire web can be leveraged as information sources linked by graphical databases.

9) Enterprise Platform- Graph databases needs enterprise platform features (Enterprise scale, ACID, commit, roll-back, triple level security model, Cloud enabled, checkpoint, replication, auditing) to be reliable, robust and for high availability.

10) Cloud and on premise licensing- AllegroGraph has flexibility to run locally or from Amazon Web Services (AWS) whatever option suits the business and technical needs.

11) Predictive Analysis-AllegroGraph provides a unique and powerful platform for predictive analysis with combination of graph, semantics, rules/logic, geospatial, temporal, social networking etc.

12) Event Support-Supports for events in which machine learning is required along with other features of AllegroGraph.

13) HDFS file system support- Supports loading of files from Hadoop Distributed File System.

14) Compatible Semantic Technologies-AllegroGraph is compatible with several semantic technologies- like Data maestro, Talend, Callimachus, TopBraid Composer, Gruff, MongoDB, Solr, Semaphore, Allegro Set, Linked data, Racer Pro, Sentient Suite, KnowMed, Cogito etc [1]

4. SYSTEM REQUIREMENTS

The AllegroGraph database server runs on Linus x86-64 bit. To run AllegroGraph on other operating systems (i.e. Windows, Mac) Linux Virtual Machine or EC2 must be set up. [1]

5. BENEFITS

1)AllegroGraph semantic graph technology can automatically link new information irrespective of the format of the information. No manual user intervention is required for analysis of structure and relationships in information. Also, there is no need to pre structure the database.

2)Since AllegroGraph uses self-defined and schema less data, it is able to structure the new information automatically when it is ingested into AllegroGraph. So, if we have several databases, then automatic linking of data of all the databases can be done powerfully by AllegroGraph without any user intervention. For this task to take place in conventional relational databases, complex coding, pre-processing, data warehousing and expert knowledge of queries is required [1]

6. POPULARITY(ALLEGRO GRAPH VS BLAZEGRAPH)

Popularity of allegro graph in comparison to other similar technology can be seen from following plot by DB Engines ranking,

[4]

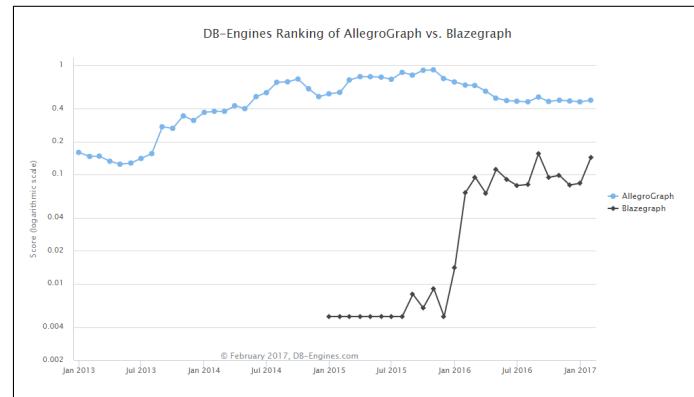


Fig. 2. Allegro Graph Popularity

7. RESOURCES FOR LEARNING ALLEGRO GRAPH

Allegro graph tutorial:

<http://franz.com/agraph/support/documentation/current/agraph-tutorial.html>

[5]

8. ACKNOWLEDGEMENT

I thank Franz, the developer of AllegroGraph for information on Allegro graph published on their website. I am also grateful to Dr. Gregor von Laszewski for providing the appropriate paper template.

9. CONCLUSION

Since Allegro Graph makes it easier for the users to analyze complex information using graphical format without much necessity for manual intervention, its popularity and utilization is increasing with time as compared to similar database technologies.

REFERENCES

- [1] "Allegro graph," Web Page, online; accessed 24-Feb-2017. [Online]. Available: <http://allegrograph.com/>
- [2] "Introduction- allegro graph," Web Page, online; accessed 24-Feb-2017. [Online]. Available: <https://en.wikipedia.org/wiki/AllegroGraph>
- [3] "Semantic triple," Web Page, online; accessed 24-Feb-2017. [Online]. Available: https://en.wikipedia.org/wiki/Semantic_triple/
- [4] "Popularity db engine," Web Page, online; accessed 23-Feb-2017. [Online]. Available: http://db-engines.com/en/ranking_trend/system/AllegroGraph
- [5] "Allegro graph tutorial," Web Page, online; accessed 24-Feb-2017. [Online]. Available: <http://franz.com/agraph/support/documentation/current/agraph-tutorial.html/>

Nagios (Example Paper for I524)

TONY LIU¹, VIBHATHA ABEYKOON¹, AND GREGOR VON LASZEWSKI¹

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

status: early draft, paper-1, S17-TS-000?, S17-TS-0003, S17-TS-0001April 12, 2017

Nagios is a system, network and infrastructure monitoring tool providing instant awareness of IT infrastructure. Nagios allows to monitor the infrastructure, alert the system admin, provide visualized reports, schedule downtime for maintenance, and plan upgrade in advance with trends and capacity diagrams. The design emphasizes highly on flexibility and scalability. We summarize details of Nagios and outline how it is useful as part of the services used in big data. © 2017 <https://creativecommons.org/licenses/>. The authors verify that the text is not plagiarized.

Keywords: Cloud, I524

<https://github.com/vibhatha/sp17-i524/blob/master/paper1/S17-TS-0003/report.pdf>

1. INTRODUCTION

Nagios [1, 2] is a system, network and infrastructure monitoring tool under open source license that provides instant awareness of mission-critical IT infrastructure. Nagios allows to monitor the infrastructure, alert the system admin, provide visualized reports, schedule downtime for maintenance, and plan upgrade in advance with trends and capacity diagrams. Its design emphasizes highly on flexibility and scalability. To provide such flexibility, Nagios is composed of different modules. Through the modular design Nagios can adapt easily to systems and networks monitoring needed by different users. It also allows integration of new components that allow adaptation towards service monitoring needs that have not yet been distributed with Nagios, allowing extensibility.

2. ARCHITECTURE

Nagios [3] flexible modular architecture allows users to customize modules to monitor the network and infrastructure. We show the elementary Nagios architecture in Figure 1 [4].

Nagios architecture consists of two main databases. They are MySQL and PostgreSQL. Nagios XI is directly connected with MySQL and PostgreSQL. There are two main components connected with MySQL, NDOUtils and NagioSQL are these two components. The databases keep the parameters and configurations needed for the Nagios Core configuration from the XI interface [5]. The NagioSQL refers to the configuration files to get configuration parameters. NagioSQL forms an advanced configuration section of the XI interface to enable the functionality. Nagios local installation is also connected with three main components. They are Local Checks, NRDP/NSCA and NRPE components.

Nagios has a number of components such as Nagios core, file system, Nagios daemon, services, plugins, event handlers,

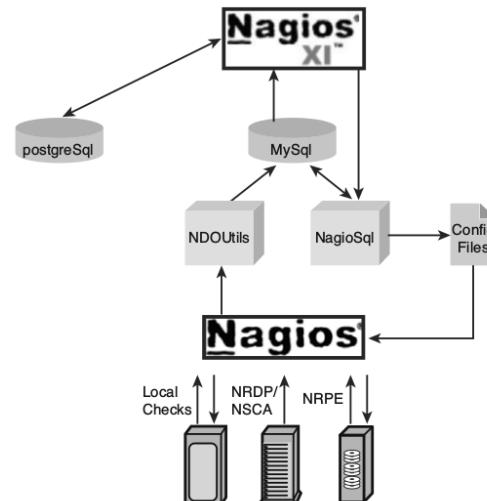


Fig. 1. Nagios Architecture [4].

performance processors and notification commands.

The main component is *Nagios core*, which is a scheduler daemon detecting network devices and services regularly. The core can alert the administrator through various notification methods like email, message and web interface about system and event changes to alert of issues and system states. Nagios plugin features allow to register events that can be triggered to execute actions and print status updates.

It has two types, check and notification. Both are used by Nagios core. The check plugin is used to check and monitor devices and services. The notification plugin is used to send out alerts if the check plugin detected any status change. Beside

these two, the users themselves can develop their own custom plugins.

Nagios uses *modules* that call through the Nagios API Nagios Events that are enacted upon by the Nagios Broker. The user can develop a modules with customized functionality and embed them within Nagios core. Whenever an event triggers the module, the module will be called to execute. The benefit of Nagios module is that the user can access all necessary information within the core process such as the Nagios status and check results.

Nagios' configuration file is text-based. It supports the sophistication of Nagios allowing to monitor large infrastructure. Also, the user must understand the configurable options. Management of the configuration has to be conducted carefully to avoid simple mistakes such as introduced by spelling errors. However, sample configurations and tools to generate configuration files by applying such configuration templates are also supported by a user-friendly web interface.

Nagios also contains a Web interface. It is developed using CGI technology in C programming language. Although it is not compatible with current popular web technologies like CSS, AJAX and JQuery, it provides sufficient scalability for larger infrastructures.

2.1. Comprehensive Monitoring

Using Nagios there are many structures that can be monitored, such as operating systems, networking protocols, system metrics and infrastructure components. There are many APIs to handle the easy monitoring of in-house applications, services and systems. In considering the network monitoring, basic use is the pinging and Nagios uses it in a complex way and configurations are complex as it is used to monitor many of the networking infrastructures. The feature is post query support is also there in Nagios, this is very important to monitor TCP or UDP connections with a particular host. The capability of pinging and keeping track on multiple ports is also enabled in this framework, so it is very easy to monitor a huge network in a particular organization. When it comes to service management, there are group service configurations that can be added to the system by Nagios, this provides a variety of options to a team to manage systems in a flexible way. There are a variety of centralized views provided. For instance General view provides a central dashboard which holds the keys to all other kinds of services and tools like monitoring performance, network outage, etc. Service details view provides the last check details, service type, host and many other information related to services. This is a vital tool, when it comes to managing thousands of services from a single organization. Host details and host groups can also be viewed by the tools in Nagios, this is very important to provide a more structured service when an organization have a number of clients, so the group management is a vital task when it comes to isolating services in cases of maintenance and development. Collapsed tree status map and marked up circular status map shows the connection between different hosts and routers in a network. This is an important tool to identify the service distribution and to identify breakdowns. By these views, the failures can be notified by a physical and geographical location, and this is a vital factor for a system management with less downtime [6].

2.2. Problem Remediation

Systems running on live environments generally associated with expected and unexpected problems. Tracking these problems

is a challenging task. Nagios has the capability of providing such events as alerts in an automatic way. Nagios consists of fast recovering capability with the support of automated tools. These tools has the capability of providing services with less downtime.

2.3. Visibility and Awareness

Visibility and awareness deal with a couple of factors. Notifications, real time updates, reporting, centralized dashboard are some of these main features. Nagios consists of a centralized dashboard which provides the access to all these information. The centralized architecture supports system administrators to deal with less problems in reaching vivid services in order to access information. In making decisions, it is vital to have access to all information in a quick manner. Nagios provides these features with the centralized dashboard.

2.4. Reporting

In referring to records and statistics, Nagios provide reporting tools to keep track on the outputs from the dashboards and central data collection systems, so that the data can be analyzed at the present moment and also the ability to check the data in the history. When it comes to tools like crystal reporting, casper reporting, etc, Nagios provide support to third party tools so the users can use the familiar reporting tools.

2.5. Open-Source

Nagios is an open source tool and the license for documentation and software is under different licenses. Nagios open software license [7], Nagios software license [8] and GPL are the licenses own by Nagios.

2.6. Multi-Tenant Capabilities

Concurrent access for multiple users is a vital factor as far as a scaled business is considered. The necessity of multiple user login, accessing data, sharing data and updating data is a vital task in such an organization. Nagios consists of a web interface which consists of all these features. The importance of the web interface is that, the installation cost and maintenance cost is less. Once it is deployed to a server, every user in that particular network containing the server has access to the web interface by means of web browser. As mentioned earlier, Nagios dashboard comes with a web interface enabling easy access to all users.

In referring to these factors, it is clear how Nagios can be used to implement a solid foundation to maintain the information technology based systems in an organization.

3. NAGIOS FOR BIG DATA

Big data is another word which refers to data sets generated by communication channels, networks, security details, etc. Nagios is a monitoring tool with major supporting tools to enhance network performance, failure detection, performance evaluation, etc. In referring to web services which access bulk data in form of JSON objects or SOAP format, network traffic occurred by these actions are very high. In such instances, Nagios can be used to monitor the network in order to provide better performance by analyzing the current status and providing suggestions on using the bandwith widely. Furthermore, the monitoring tools and reporting tools can provide a wide idea on the fact how data is being collected in a particular network. The rate of change of free capacity of local storages in servers is a vital factor in case of providing storage facility with less data damage and no data loss.

Nagios has the capability of providing an analytic understanding on the nature of big data distribution and management in a particular network.

3.1. Big Data Requirements for Monitoring

In monitoring of clouds and clusters, Nagios can be used as a powerful tool. The clouds and clusters always depend on the network. In such scenarios, network traffic monitoring, failure detection, network bottle-neck detection and similar tasks can be done using Nagios. And also the status monitoring such as memory, network bandwidth, CPU performance, GPU performance and similar activities can be tracked using Nagios.

Servers, disks and file systems are vital factors in big data management. The server performances and storage status is a vital factor for an organization providing cloud based software solutions. So these file system management becomes a very important task in order to provide less downtime services. Nagios has monitoring capability and visualizing capability, these dashboards can be used to provide information to the users in an easy manner. Real time performance update, storage status, server failures can be detected using Nagios. All these factors together form a solid background to provide cloud based software solutions [9].

4. NAGIOS FOR BIG DATA APPLICATION

Big data applications are the applications handling huge amount of data and thousands of users through out a network. In such a process, there are many factors that has to be considered. Performance is one of the main concerns in an organization. In order to achieve better performance, speed of the network, easy access to data, user management, security and many more factors contribute. Nagios has the capability of monitoring most of these factors and it provides updates, reports and centralized dashboard to system administrators. Big data always deal with very complex networks, so data flow must be smooth and fast. Nagios provides the network monitoring tools to understand the nature of a network in a given time. With the real time updates, the bottle-necks and network related problems can be traced quickly. Multi-user management is also a value added feature in Nagios, with big data applications, many users access same bulk of data in same time. Network monitoring tools and other performance tools in Nagios provide a wide understanding about user actions and network traffic distribution. Nagios provide a better infrastructure to this kind of applications.

5. CONCLUSION

Nagios provides a wide understanding to the network management, reporting, centralized dashboard and many more other features enabling a solid foundation for a scalable organization. It offers a variety of reports and vivid dashboards under the main panel to help system administrators to make decisions in an easier way. Nagios can be identified as a long term solution for big data related systems to manage both users and resources. Nagios shape up the organizational hierarchy in order to provide flexible and smooth services to the customers. In big data management, the network monitoring, file system monitoring, database monitoring are vital factors. Nagios provides infrastructure to all these components in order to provide better performance to the users. Cloud infrastructure relies on low downtime and better performance. Nagios provide the necessary tools to track the health of clouds and storages. Nagios is a useful tool to manage big data and big data related problems.

REFERENCES

- [1] Nagios Enterprises, "Nagios components," Web Page, accessed: 2017-1-11. [Online]. Available: <https://www.nagios.org/projects/>
- [2] Wikipedia, "Nagios," Web Page, Feb 2017. [Online]. Available: <https://en.wikipedia.org/wiki/Nagios>
- [3] C. Issariyapat, P. Pongpaibool, S. Mongkolluksame, and K. Meesublak, "Using nagios as a groundwork for developing a better network monitoring system," in *2012 Proceedings of PICMET '12: Technology Management for Emerging Technologies*, July 2012, pp. 2771-2777.
- [4] D. Josephsen, *Nagios: Building Enterprise-Grade Monitoring Infrastructures for Systems and Networks*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2013.
- [5] David Josephsen, "Nagios," Web Page, April 2013. [Online]. Available: <http://www.informit.com/articles/article.aspx?p=2033339&seqNum=2>
- [6] Network Startup Resource Centers, "Network management and monitoring," Web Page - Pdf, Feb 2017. [Online]. Available: <https://nsrc.org/workshops/2012/apricot-nmm/materials/nagios.pdf>
- [7] Assets Nagios, "Nagios open software license," Web Page, Feb 2017. [Online]. Available: https://assets.nagios.com/licenses/nagios_open_software_license.txt
- [8] Assets Nagios, "Nagios software license," Web Page, Feb 2017. [Online]. Available: https://assets.nagios.com/licenses/nagios_software_license.txt
- [9] Zachary Conger, "Answering big data questions: Monitoring at petabyte scale," Web Page, Feb 2017. [Online]. Available: <http://altitudedigital.com/tech-blog/when-big-data-gives-you-big-problems-and-how-to-fix-them/>

S17-TS-0006/report.pdf not submitted