***Hadoop Based Logger Analytics***

Pearson Technical Challenge – 2014

Media & Information Services

**Team:**

* Abhishek Basak
  + [abhishek.basak@tcs.com](mailto:banerjee.avijit@tcs.com)
* Suchetana Basu
  + [suchetana.basu@tcs.com](mailto:suchetana.basu@tcs.com)
* Irfan Kapadiya
  + [irfan.kapadiya@tcs.com](mailto:irfan.kapadiya@tcs.com)

|  |
| --- |
| **Confidentiality Statement** |
| Include the confidentiality statement within the box provided. This has to be legally approved  **Confidentiality and Non-Disclosure Notice**  The information contained in this document is confidential and proprietary to TATA Consultancy Services. This information may not be disclosed, duplicated or used for any other purposes. The information contained in this document may not be released in whole or in part outside TCS for any purpose without the express written permission of TATA Consultancy Services. |

|  |
| --- |
| **Tata Code of Conduct** |
| We, in our dealings, are self-regulated by a Code of Conduct as enshrined in the Tata Code of Conduct. We request your support in helping us adhere to the Code in letter and spirit. We request that any violation or potential violation of the Code by any person be promptly brought to the notice of the Local Ethics Counselor or the Principal Ethics Counselor or the CEO of TCS. All communication received in this regard will be treated and kept as confidential. |

**Table of Content**

[Abstract 4](#_Abstract)

[Problem Statement 4](#_Toc377204913)

[Why solve this Problem? 4](#_Toc377204914)

[What is the challenge? 4](#_Toc377204914)

[Solution Outline 5](#_Toc377204914)

[Solution Outline 5](#_Toc377204914)

[Benefit of the Solution 6](#_Toc377204914)

[Technology Stack 7](#_Toc377204914)

[Configuration Instruction 7](#_Toc377204914)

[Known Issues and Limitation 9](#_Toc377204914)

[Reference 9](#_Toc377204914)

# Abstract

This paper is about *Hadoop Based Logger Analytics*. This paper will give basic idea about the problem domain of Big Data Analytics, possible implementation approach for streaming log data into Hadoop system, using one of the available analytical tool to extract meaning full information out of the unstructured log data and presenting information to user in graphical format.

The Code can be downloaded from: <https://github.com/abhishekbasak/LogAnalyzer>

# Problem statement

As part of this technical challenge we are going to make a Hadoop Based Logger Analytics. The proposed POC will try to collect application log from a simulated log generator, which is kind of unstructured data, will stream the collected log into HDFS System, and will employ an map-reduced framework to collect the data back from the system in more meaningful format and will present to the user in some informative textual or graphical format.

# Why solve this problem?

Now a days this is a very popular business case in Big Data domain and most of the business people are eying for implementing such system for different reasons like Usage Tracking, Trend Determination, System Monitoring, Disaster Control, Failure Analysis etc.

A simple example can be, for an Assessment Delivery System, an obvious monitoring system can be a real time system tracker, which will deliver various information to the business users like, daily frequency of assessment taken, which part of the day is the most busiest time for the application, which part of the geographical location is mostly hitting the application, where users are facing more problem etc. Now for this kind of system, this will be very expensive for the main application if this kind of data collection needs to done by them. So here our solution can fit very nicely. This will keep the main application server and database untouched and with very low investment this application can add enormous value to the business users.

# What is the “challenge”?

The first reason that this is a real challenge is the fact that this type of work is very nascent across the industry. Open Source tools have started coming out for different problem domains in Big Data platform but many of them have not been much evaluated. That means we have very little access to expertise and published works in this area.

Apart from the above fact, the width and depth of the tech stack involved (mentioned in the next section) also makes this POC a challenge. We don’t know how (if at all) all the pieces in the tech stack will work together.

# Solution Outline

The three important aspects of the problem, that has been targeted to solve in this challenge, are:

1. Collecting Log Data and Streaming into Hadoop System
2. Maintaining Unstructured Data into Hadoop
3. Extracting meaningful information out of unstructured data and presenting to the user.



Below is the brief description about the exploration done in each of these areas:

1. **Collecting Log Data and Streaming into Hadoop System**

Logger data being the key ingredients of the process, the foremost problem we have to deal with is getting the log into the Hadoop System. Ideally there can be multiple production instances of the application running and GBs of logs may get generated per hour. So 2 aspects of the problem are; a) collecting the log from distributed systems and sinking them into the targeted Hadoop system and b) continuity, i.e. when volume of the data generated becomes more than the consumption capability of the Hadoop system, to ensure logs are not getting lost or the either systems are not crashing due to the sudden heavy load. To address all these problems, Apache Flume has been used as the streaming tool.

Apache Flume is a distributed, reliable and available service for effectively collecting, aggregating, and moving large amount of log data. It has simple and flexible architecture based on streaming data flows. It also has effective failover and recovery mechanism, so ideal for fault tolerance.

1. **Maintaining Unstructured Data into Hadoop**

Next part of the problem is obviously how to maintain the unstructured data effectively, so that analytical information can be extracted out of that. HDFS is Hadoops underlying file system, which is distributed in nature. The log that has been collected from Hadoop, have been placed into the HDFS maintaining some convenient structure like logs from each day have been placed in same folder.

1. **Extracting Analytical Information**

Finally getting the information out of the unstructured data. Map reduce framework is the choice to mine the huge data volume. Hadoop comes up with other tools like Pig, Hive for convenient data access, in the bottom which are converted to map-reduce jobs finally. For this POC, Hive has been used.

Apache Hive is a data warehouse software facilities for querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto its data and querying the data using an SQL-like language called Hive-QL.

As part of this POC, a Hive table has been created which will contain different parts of the log in different column. Below is an example of a sample line from the log:

*2014-11-29 12:31:08 INFO ~ classname=SampleLogGenerator, method=doJob, LogGenerator=1.0-alpha, Test Started, params={STUDENT\_ID=28b6db28-d4d7-4b32-a9af-5961df5713e6}*

So the Hive table that has been configured will contain the columns like:

*Date | Time | Log\_Lavel | Class\_Name | Method\_Name | App\_Name | Release\_Version | Event\_Message | Event\_Parameters*

And from the above line data will be mapped like:

|  |  |
| --- | --- |
| **Date** | 11/29/14 |
| **Time** | 12:31:08 |
| **Log\_Lavel** | INFO |
| **Class\_Name** | SampleLogGenerator |
| **Method\_Name** | doJob |
| **App\_Name** | LogGenerator |
| **Release\_Version** | 1.0-alpha |
| **Event\_Message** | Test Started |
| **Event\_Parameters** | {STUDENT\_ID=28b6db28-d4d7-4b32-a9af-5961df5713e6} |

For better performance the table will contain a partition, to store the log for each day in the single partition.

# Benefits of the solution

1. This will enable user to leverage Log data, which gets generated by the application to extract out valuable information.
2. Will help user to maintain a pool of unstructured data and will enable them to use it as and when required without touching the main application process, so no overload on the application.

# Technology Stack

**Language:** Java

**Web Framework:** Play Framework

**Hadoop System:** Hortonwork Sandbox

**Log Streaming Tool:** Apache Flume

**Hadoop DB:** Hive

# Configuration Instruction

# Configuring Log Generator Application and Apache Flume

**loggen** is the sample application used for generating sample log which will be feeding into Hadoop System. This is an application built on Play Framework 1.2.x. Detail information on how to install and get a Play application up can be found in <https://www.playframework.com/documentation/1.2.x/install>

For stream the log into Hadoop System, Apache Flume needs to be setup. There needs two installation of Apache Flume, one in the server where **loggen** application is running, and another one in the Hadoop system. Detailed instruction about Apache Flume installation can be found in <https://flume.apache.org/FlumeUserGuide.html>

The package *Flume-Setup* contains two conf files for two of the servers. [*flume-source.conf*](https://github.com/abhishekbasak/LogAnalyzer/blob/master/Flume-Setup/flume-source.conf) and [*flume-server.conf*](https://github.com/abhishekbasak/LogAnalyzer/blob/master/Flume-Setup/flume-server.conf) is meant for Hadoop side installation.

# Hadoop and Hive Setup

Hortonwork Sanbox has been used for the Hadoop System, which comes up with the popular Hadoop system packages, like Hive, Pig, Oozie etc. Detailed installation instruction can be found in <http://hortonworks.com/products/hortonworks-sandbox/>

A Virtualbox or a VMware installation is required for the sanbox to be installed. This POC has been tested in MacOS X environment, with Hadoop Sandbox built on Oracle Virtual Box.

Instructions for Apache Flume setup in Hortonwork Sanbox can be found in <http://hortonworks.com/hadoop/flume/>

Required Hive scripts can be found in the package *Hive-Query*. The create query will create the required table which will map the unstructured log file stored in the HDFS, to the external table which can be queried to retrieve the required analytical info.

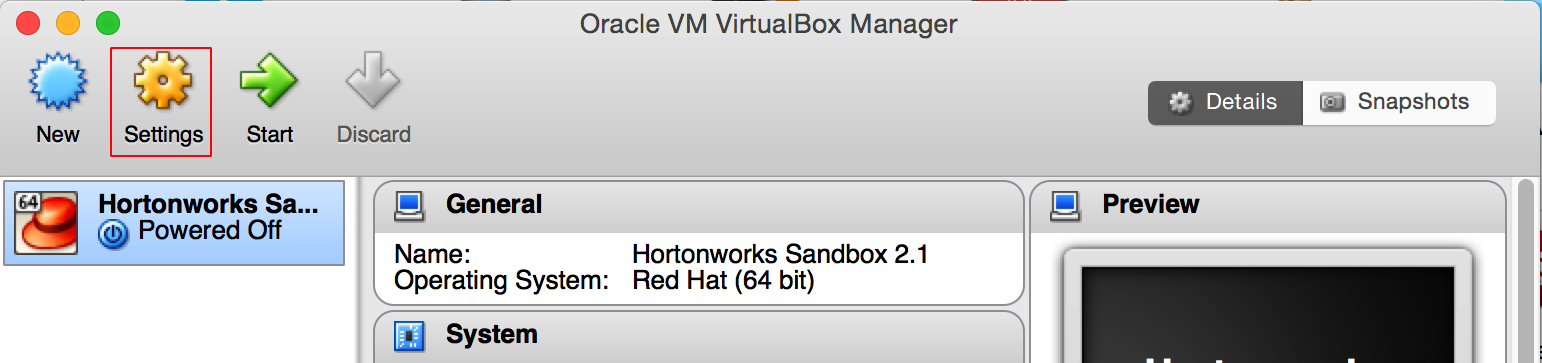
The table contains a partition for boosting performance. The partition is mapped for per day log, which is done by pointing the per day folder. When logs are streamed into the HDFS, Flume configuration takes care that each day logs are going into different folder.

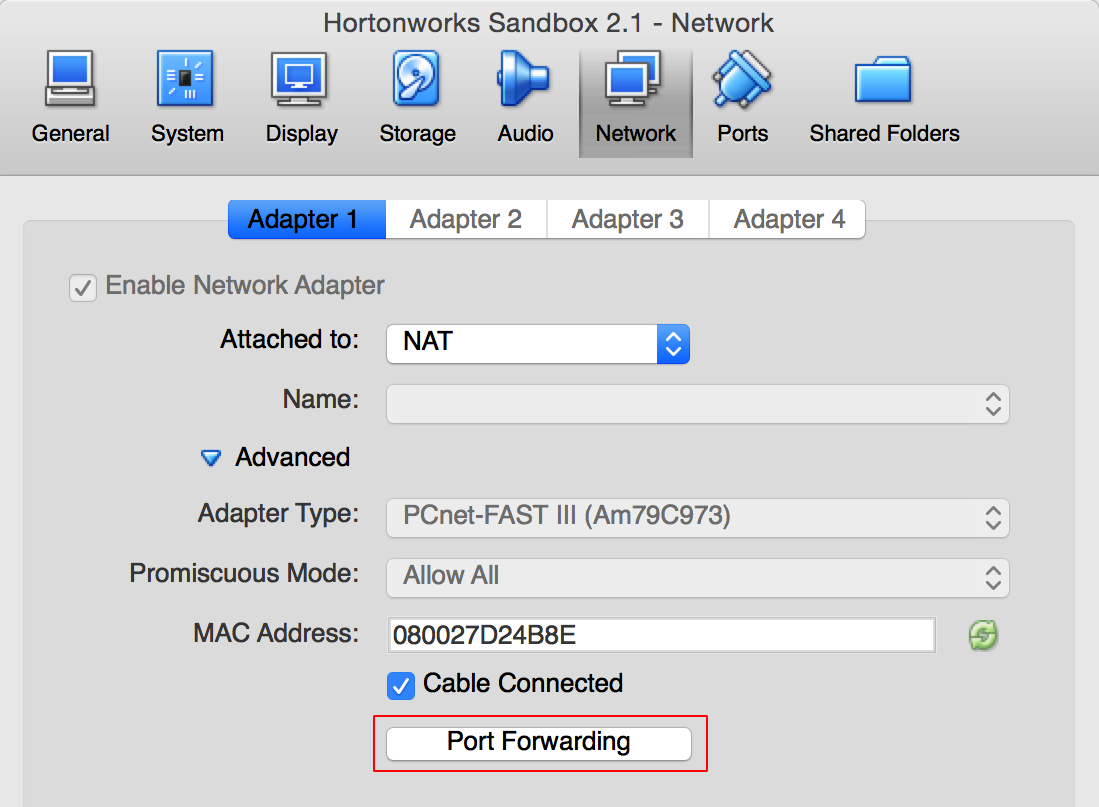
An alter statement is required to be executed to point the data to the required partition. This needs to be executed per day. An **Oozie** set up can be used to automate this process.

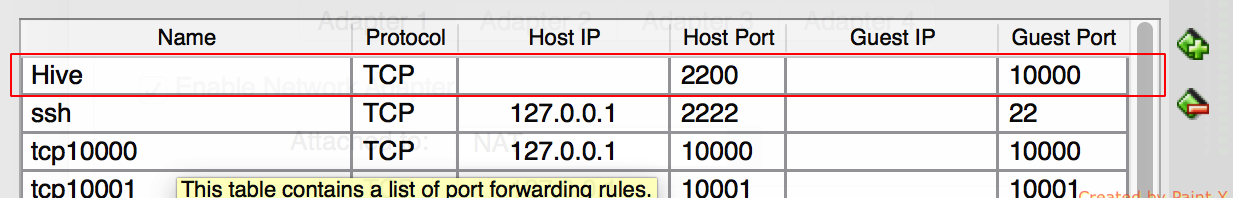
# Configuring Hive Connector Application

**hiveconnector** is the sample application where user can see the analytical information extracted from the log file stored in HDFS. This is also an application built on Play Framework. Same setup steps can be followed here what was mentioned earlier.

To reach Hive2 service from Java Program, we need to add a port-forwarding rule as shown in the image below:







# Known Issues & Limitations

1. An Oozie setup is required to automate the Hive table partition creation process. That has not been covered in this document.
2. This is a local system installation just meant for POC purpose. For production-based installation this has to be deployed in some Cloud server like AWS. That is also not covered in this doc.

# References

* Hortonwork: <http://hortonworks.com/products/hortonworks-sandbox/>
* Apache Flume: <http://flume.apache.org/>
* Hive: <https://cwiki.apache.org/confluence/display/Hive/GettingStarted>
* Play Framework: <https://www.playframework.com/documentation/1.2.x/home>

**About Tata Consultancy Services (TCS)**

Tata Consultancy Services is an IT services, consulting and business solutions organization that delivers real results to global business, ensuring a level of certainty no other firm can match. TCS offers a consulting-led, integrated portfolio of IT and IT-enabled infrastructure, engineering and assurance services. This is delivered through its unique Global Network Delivery ModelTM, recognized as the benchmark of excellence in software development. A part of the Tata Group, India’s largest industrial conglomerate, TCS has a global footprint and is listed on the National Stock Exchange and Bombay Stock Exchange in India.

For more information, visit us at **www.tcs.com**.

**IT Services**

**Business Solutions**

**Outsourcing**

All content / information present here is the exclusive property of Tata Consultancy Services Limited (TCS). The content / information contained here is correct at the time of publishing. No material from here may be copied, modified, reproduced, republished, uploaded, transmitted, posted or distributed in any form without prior written permission from TCS. Unauthorized use of the content / information appearing here may violate copyright, trademark and other applicable laws, and could result in criminal or civil penalties. **Copyright © 2011 Tata Consultancy Services Limited**