School of Informatics & IT
TEMASEK POLYTECHNIC

Specialist Diploma in Big Data Management

AY 2020/2021 April Semester Term A

Big Data Systems Architecture

CBG1C01

Individual Assignment

Meenakshi Dekshinamurthy

ID: 2080548G

Temasek Polytechnic School of Informatics & IT

Specialist Diploma in Big Data Management

BIG DATA SYSTEMS ARCHITECTURE CBG1C01

BDSA_Assignment_AY2021_Apr

Practical Class:	P01
Submitted by: (list all students)	Meenakshi Dekshinamurthy / 2080548G
Date:	21/ 06 / 2020

"By submitting this work, I am declaring that I am the originator(s) of this work and that all other original sources used in this work has been appropriately acknowledged.

I understand that plagiarism is the act of taking and using the whole or any part of another person's work and presenting it as without proper acknowledgement.

I also understand that plagiarism is an academic offence and that disciplinary action will be taken for plagiarism."

Name and Signature of Student: Meenakshi Dekshinamurthy

Contents

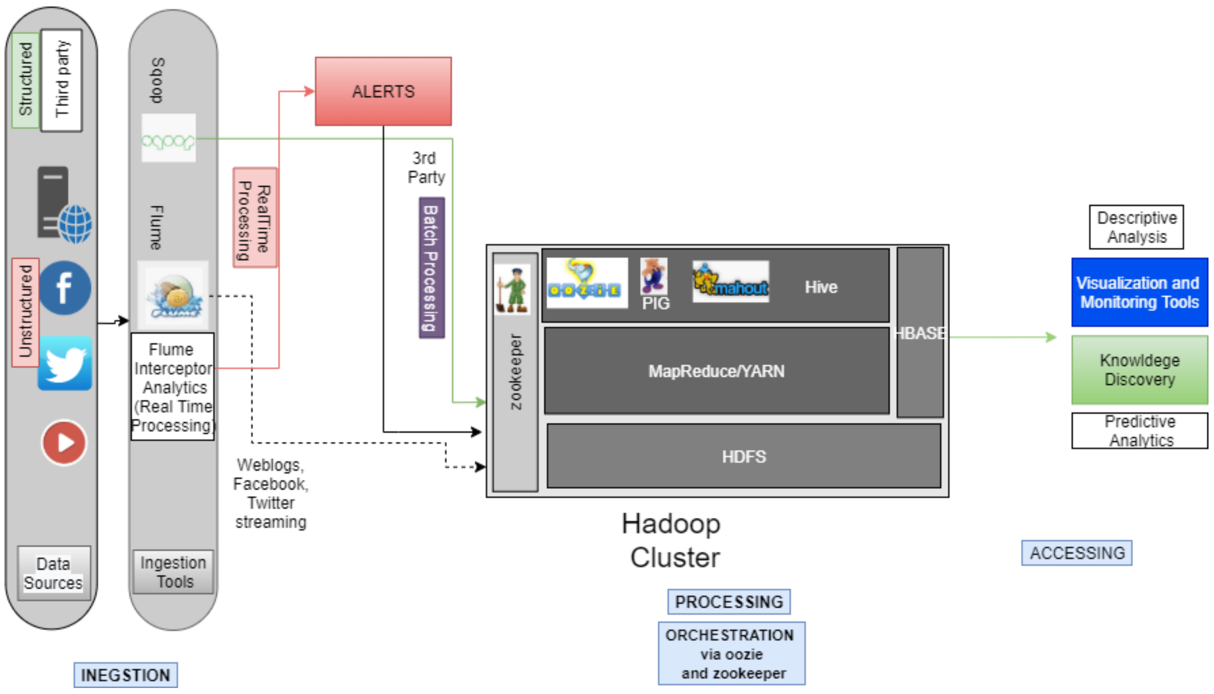
1. INTRODUCTION

A small IT business, venturing into Hyper casual games, has released a theme of freemium games in a Genre. They rely on Ads as their main source of income and are looking for effective monetization strategies to be implemented.

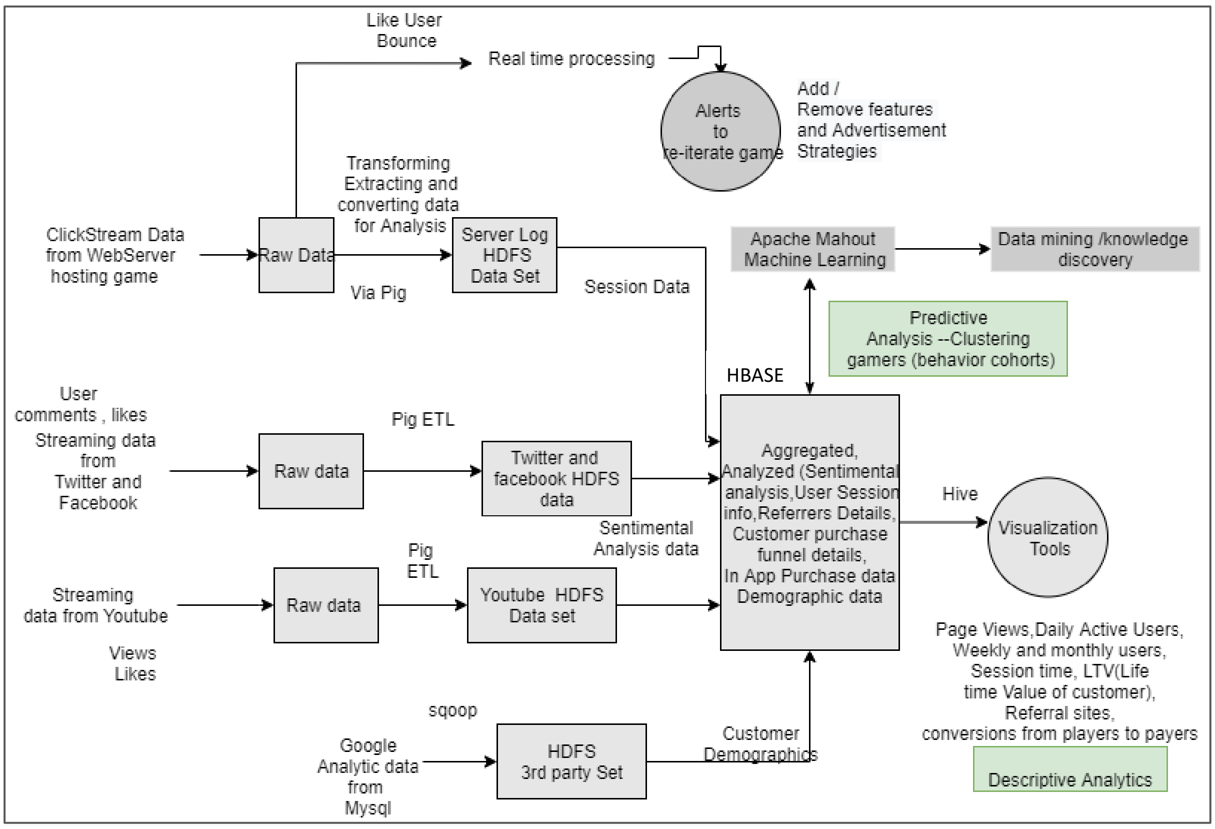
With more gamers playing the games, there is continuous growth of data. We recommend a big data solution to handle the velocity and volume of data, so that data can be collected on the gamer behavior, area of play, frequency of visits, social influences through real time and batch processing. This data can then be used for clustering the players into optimal segments by ML. The monetization strategies can then be personalized to the clusters so that more players are engaged, and Lifetime value of players can be increased. The real time analytics will help to iterate the games with better responsive times to engage the players. Personalized Strategies and gaming features with quick iterations to the game can help the company improve their revenue.

We recommend using cloud-based components to support scalability, reliability, cost effectiveness and content delivery. The following figure gives the overview of architecture Design.

2. DESIGN OVERVIEW



3. PROCESS FLOW AND DESIGN


Like User Bounce
Real time processing
Alerts to re-iterate game
Add / Remove features and Advertisement Strategies
Transforming Extracting and converting data for Analysis
ClickStream Data from WebServer hosting game
Raw Data
Server Log HDFS Data Set
Via Pig
Session Data
Apache Mahout Machine Learning
Data mining /knowledge discovery
Predictive Analysis --Clustering gamers (behavior cohorts)
HBASE
User comments , likes
Streaming data from Twitter and Facebook
Raw data
Pig ETL
Twitter and facebook HDFS data
Aggregated, Analyzed (Sentimental analysis,User Session info,Referrers Details, Customer purchase funnel details, In App Purchase data Demographic data
Hive
Visualization Tools
Sentimental Analysis data
Pig ETL
Streaming data from Youtube
Raw data
Youtube HDFS Data set
Views Likes
Page Views,Daily Active Users, Weekly and monthly users, Session time, LTV(Life time Value of customer), Referral sites, conversions from players to payers
sqoop
Google Analytic data from Mysql
HDFS 3rd party Set
Customer Demographics
Descriptive Analytics

4. DATA SOURCES

ClickStream data is the primary data source for this analysis.

ClickStream data is real time log data from server streamed into HDFS and this unstructured data is transformed, processed for useful information like user ip, user clicks and online activity and stored in a NO SQL Database HBASE

ClickStream Data from Webservers can be real time analysed to identify and respond with real time responsiveness to critical gamer behaviors like bounce rate and iterate the game features.

Facebook Data on the game from users can be captured, transformed, and processed to find the social influence of the customer and sentimental analysis

Every theme of game released has a play preview in Youtube. The Youtube data for the game video is captured to analyze the player interest, sentimental analysis

Google Analytic Data is captured periodically into mysql and imported using Sqoop and analyzed for further input to ML algorithm and Dashboard of visualization tools

Here the sample data taken for demonstration is webserver log in the combined log format.

https://github.com/elastic/examples/blob/master/Common%20Data%20Formats/apache_logs/apache_logs

The Data is containing ipAddr, identity of client, username ("-" for public sites), timestamp string, HTTP request, HTTP status, bytes transferred, /referrer, User Agent.

5. INGESTION

The Real time Streaming data from Weblogs, Facebook and Youtube is streamed using Flume, a Streaming tool for logs and events into HDFS.

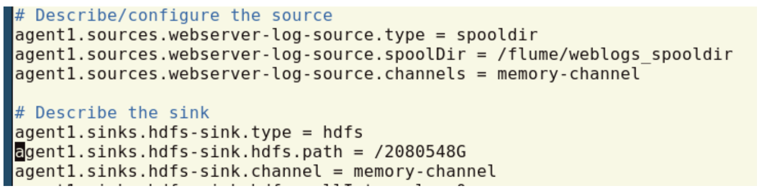
This demo shows simulation for ingestion of raw webserver log data into HDFS using Flume

The Flume source, sink and Channel configurations are to be done appropriately for the weblogs in spooldir.conf file depending on whether the source of data is weblogs, twitter, facebook. The Apis available in Facebook and Youtube can be used for streaming data using Flume into HDFS.

The weblog data is being demonstrated here:

Agent1.sources = webserver-log-source.

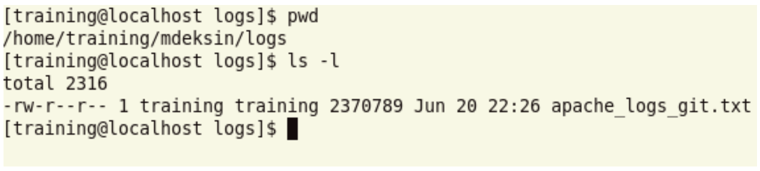
Source and Sink directories are configured as shown below



```
# Describe/configure the source
agent1.sources.webserver-log-source.type = spooldir
agent1.sources.webserver-log-source.spoolDir = /flume/weblogs_spooldir
agent1.sources.webserver-log-source.channels = memory-channel

# Describe the sink
agent1.sinks.hdfs-sink.type = hdfs
agent1.sinks.hdfs-sink.hdfs.path = /2080548G
agent1.sinks.hdfs-sink.channel = memory-channel
```

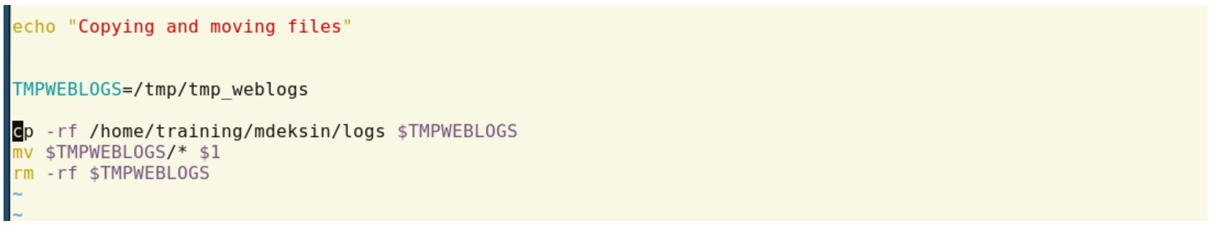
The sample log data is locally stored in



```
[training@localhost logs]$ pwd
/home/training/mdeksin/logs
[training@localhost logs]$ ls -l
total 2316
-rw-r--r-- 1 training training 2370789 Jun 20 22:26 apache_logs_git.txt
[training@localhost logs]$
```

The source and sink directories are verified

The Script copy-move-weblogs.sh to simulate server log creation is pointing to the sample data.

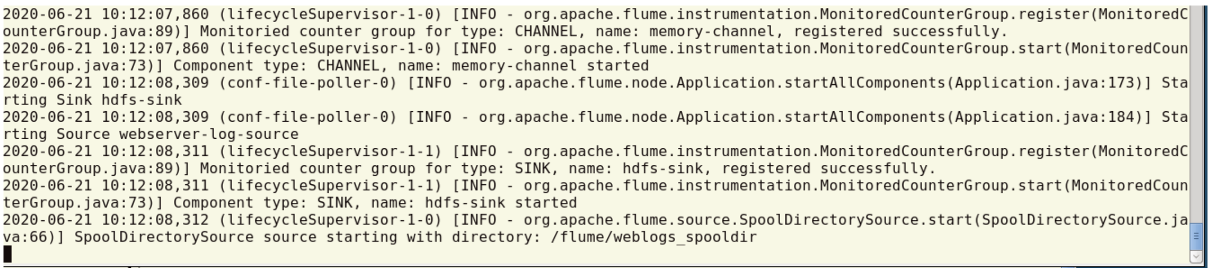


```
echo "Copying and moving files"

TMPWEBLOGS=/tmp/tmp_weblogs

cp -rf /home/training/mdeksin/logs $TMPWEBLOGS
mv $TMPWEBLOGS/* $1
rm -rf $TMPWEBLOGS
~
~
```

Once the flume agent is started, the memory channel, the sink and source are ready.



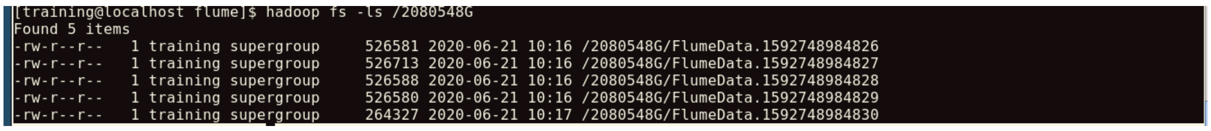
```
2020-06-21 10:12:07,860 (lifecycleSupervisor-1-0) [INFO - org.apache.flume.instrumentation.MonitoredCounterGroup.register(MonitoredC
ounterGroup.java:89)] Monitoried counter group for type: CHANNEL, name: memory-channel, registered successfully.
2020-06-21 10:12:07,860 (lifecycleSupervisor-1-0) [INFO - org.apache.flume.instrumentation.MonitoredCounterGroup.start(MonitoredCoun
terGroup.java:73)] Component type: CHANNEL, name: memory-channel started
2020-06-21 10:12:08,309 (conf-file-poller-0) [INFO - org.apache.flume.node.Application.startAllComponents(Application.java:173)] Sta
rting Sink hdfs-sink
2020-06-21 10:12:08,309 (conf-file-poller-0) [INFO - org.apache.flume.node.Application.startAllComponents(Application.java:184)] Sta
rting Source webserver-log-source
2020-06-21 10:12:08,311 (lifecycleSupervisor-1-1) [INFO - org.apache.flume.instrumentation.MonitoredCounterGroup.register(MonitoredC
ounterGroup.java:89)] Monitoried counter group for type: SINK, name: hdfs-sink, registered successfully.
2020-06-21 10:12:08,311 (lifecycleSupervisor-1-1) [INFO - org.apache.flume.instrumentation.MonitoredCounterGroup.start(MonitoredCoun
terGroup.java:73)] Component type: SINK, name: hdfs-sink started
2020-06-21 10:12:08,312 (lifecycleSupervisor-1-0) [INFO - org.apache.flume.source.SpoolDirectorySource.start(SpoolDirectorySource.ja
va:66)] SpoolDirectorySource source starting with directory: /flume/weblogs_spooldir
```

And the data is injected into HDFS in its raw format

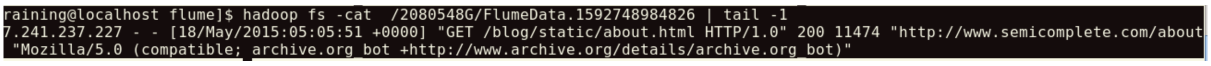
6. PROCESSING

The raw Data ingested into HDFS is cleaned, transformed, and aggregated into data sets that can be further used for analysis using Pig into HBASE

The ingested data files in HDFS are



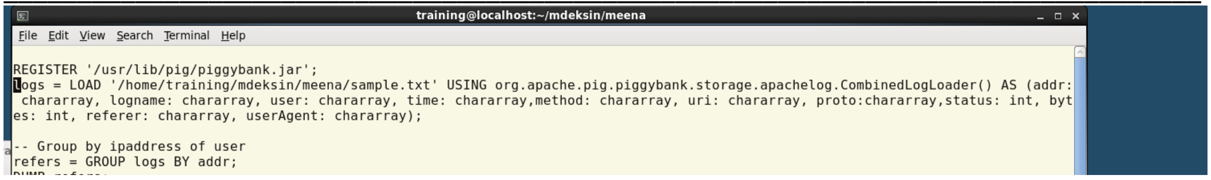
```
[training@localhost flume]$ hadoop fs -ls /2080548G
Found 5 items
-rw-r--r--   1 training supergroup     526581 2020-06-21 10:16 /2080548G/FlumeData.1592748984826
-rw-r--r--   1 training supergroup     526713 2020-06-21 10:16 /2080548G/FlumeData.1592748984827
-rw-r--r--   1 training supergroup     526588 2020-06-21 10:16 /2080548G/FlumeData.1592748984828
-rw-r--r--   1 training supergroup     526580 2020-06-21 10:16 /2080548G/FlumeData.1592748984829
-rw-r--r--   1 training supergroup     264327 2020-06-21 10:17 /2080548G/FlumeData.1592748984830
```



```
raining@localhost flume]$ hadoop fs -cat  /2080548G/FlumeData.1592748984826 | tail -1
7.241.237.227 - - [18/May/2015:05:05:51 +0000] "GET /blog/static/about.html HTTP/1.0" 200 11474 "http://www.semicomplete.com/about
 "Mozilla/5.0 (compatible; archive.org_bot +http://www.archive.org/details/archive.org_bot)"
```

The screen capture above shows the raw data format of the server log.

For demonstration purpose, I have used a small part of ingested data as sample.txt to explain the script. The Script is named 2080548G.pig. A component called piggybank.jar to handle combined log is used in the pig script.


training@localhost:~/mdeksin/meena
File Edit View Search Terminal Help

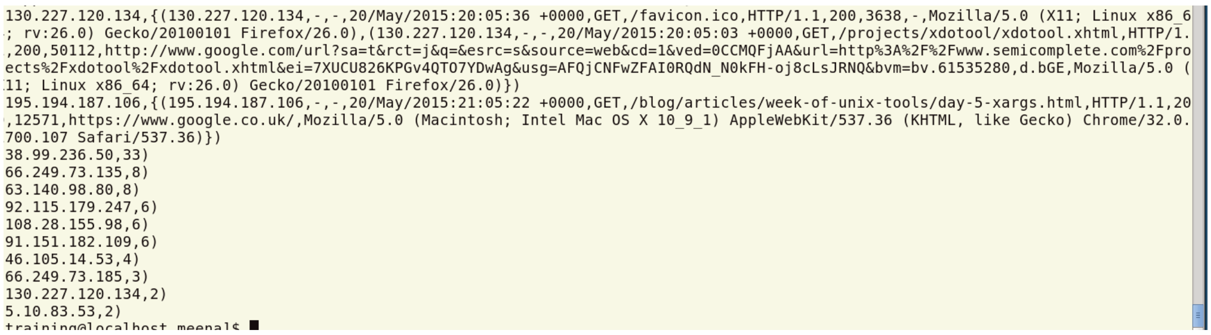
```
REGISTER '/usr/lib/pig/piggybank.jar';
logs = LOAD '/home/training/mdeksin/meena/sample.txt' USING org.apache.pig.piggybank.storage.apachelog.CombinedLogLoader() AS (addr:
 chararray, logname: chararray, user: chararray, time: chararray,method: chararray, uri: chararray, proto:chararray,status: int, byt
es: int, referer: chararray, userAgent: chararray);

-- Group by ipaddress of user
refers = GROUP logs BY addr;
```

The sample line shown above can be used to understand how data types are assigned in the script. Char array type chararray is used to store ipaddr, time, http method, URI, referral and int types are used to store HTTP status, bytes transferred.

The data is then GROUP by, say ipaddr -> GROUP logs by addr

The result will be a schema of two columns. Basically, a tuple with two fields. One integer and one bag. One will be the ipaddr and the other will be bag containing group of tuples containing the records with respective ipaddr



```
130.227.120.134,{(130.227.120.134,-,-,20/May/2015:20:05:36 +0000,GET,/favicon.ico,HTTP/1.1,200,3638,-,Mozilla/5.0 (X11; Linux x86_6
; rv:26.0) Gecko/20100101 Firefox/26.0),(130.227.120.134,-,-,20/May/2015:20:05:03 +0000,GET,/projects/xdotool/xdotool.xhtml,HTTP/1.
,200,50112,http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCMQFjAA&url=http%3A%2F%2Fwww.semicomplete.com%2Fpro
ects%2Fxdotool%2Fxdotool.xhtml&ei=7XUCU826KPGv4QTO7YDwAg&usg=AFQjCNFwZFAI0RQdN_N0kFH-oj8cLsJRNQ&bvm=bv.61535280,d.bGE,Mozilla/5.0 (
11; Linux x86_64; rv:26.0) Gecko/20100101 Firefox/26.0)})
195.194.187.106,{(195.194.187.106,-,-,20/May/2015:21:05:22 +0000,GET,/blog/articles/week-of-unix-tools/day-5-xargs.html,HTTP/1.1,20
,12571,https://www.google.co.uk/,Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/32.0.
700.107 Safari/537.36)})
38.99.236.50,33)
66.249.73.135,8)
63.140.98.80,8)
92.115.179.247,6)
108.28.155.98,6)
91.151.182.109,6)
46.105.14.53,4)
66.249.73.185,3)
130.227.120.134,2)
5.10.83.53,2)
```

In the above screenshot, we can see the tuple for ipaddr = 130.227.120.134

FLATTEN will unnest the tuple and count the tuples inside the bags

So, for each ipaddr, the no. of tuples will be counted, which in turn will give the no. of clicks from the ipaddr.

For (130.227.120.134, 2) shows there are 2 clicks which generated http methods from 130.227.120.134.

This type of data meaningful can be loaded into HBASE for further analysis.

7. ORCHESTRATION

Oozie and zookeeper are used for scheduling and co-ordinating with Pig and Hbase.

8. ANALYSIS

Once into HBASE, further the data is further normalized and used for Machine learning using Apache Mahout. The machine learning algorithms K-mode and K-mean help to process categorical and normalized data for clustering the gamers into optimal segments.

Also, various Descriptive analysis and visualization tools to monitor will help to develop better insights using data.

9. RESEARCH TOOL

a. Logstash (Ingestion Layer tool)

Logstash is a tool for managing events and logs. It is used to collect real time logs, parse them and store them for later use. The minimal Logstash installation has one Logstash instance and one Elasticsearch instance, which are directly connected. Logstash uses an input plugin to ingest data and an Elasticsearch output plugin to index the data in Elasticsearch, following the Logstash processing pipeline. Logstash Is used to gather logging messages, convert them into json documents and store into Elasticsearch cluster

Comparing Logstash and Flume

Logstash	Flume
Ingests and Transforms	Ingests
Dynamically prepare data regardless of complexity and format (all shape, sizes and sources)	Most suited to textual log data. Can also be images. Hard to manage multiple connections.
Parse, and transform data on the fly. Derive Structure from unstructured data	Minimal Reg Exp parsing to include and exclude events
Logstash has a pluggable framework /easy custom plugin generator	Custom flume component possible
Elasticsearch is the default output	Elasticsearch can be configured as output
At least once using filebeat	At least once using transactions
Flexible and interoperable	Best integration with HDFS

The logs are not of complex data types like json and Flume uses a simple extensible data model that allows for online analytic applications and integrates best with HDFS. Through flume-interceptor-analytics it is possible to apply real-time analytics to data flows. Analysing data in-flight reduces response times and allows consumers to view information as it happens.

b. Airflow (Orchestration Layer)

Air flow is used to programmatically author, schedule, and monitor data pipelines. It is an opensource tool with which DAGs are described how to run a workflow. Python features are used to create workflows. Dynamic pipelines are generated using Airflow.

Airflow	OoZie
Parametrizing scripts using Jinja templating engine	Workflows can be parametrized
Event based Scheduling	Time based and input based scheduling
Python script and UI to support	JAVA and XML
Capable of creating extremely complex workflows	Hard to build complex workflows
Compatible to HDFS, HIVE, MYSQL, Postgres	Native connections to HDFS, HIVE and PIG, Sqoop

Oozie is simple to use with pig and since the developer has better knowledge of java/xml oozie is chosen for the solution

10. Video Walkthrough

The link to my video is stored in the following Google drive.

https://drive.google.com/drive/folders/1UIdpwReCedwKoKQt5LvFI_yUBXR1f2LO?usp=sharing

11. APPENDIX

Practical Hive: A Guide to Hadoop's Data Warehouse System, Scott Shaw etnl..

Using Flume, Hari Shreedharan

http://fileformats.archiveteam.org/wiki/Combined_Log_Format

http://hadooptutorial.info/wp-content/uploads/2014/11/combined_access_log.txt

https://airflow.apache.org/

https://www.elastic.co/logstash