

JUDCon

JBoss Users & Developers Conference

2012: Boston

Big Data : Experiments with Apache Hadoop and JBoss Community projects

About the speaker

- Anil Saldhana is Lead Security Architect at JBoss.
- Founder of PicketBox and PicketLink.
- Interested in using Big Data and Analytics in PicketLink

About the talk

- Talk is based on personal experiments during free time.
- Project PicketLink generates logs.
- Log analysis and security analytics involves Big Data.
- Hence have to learn big data stuff.

About the talk

- Talk is based on personal experiments during free time.
- If you already use Hadoop projects, you will be disappointed. *Share your experience with us. :-)*
- If you are new to Hadoop ecosystem, then you will forgive me.

Milestones for the talk

- Understand what Apache Hadoop ecosystem is. (*Milestone 1*)
- Use Apache Hadoop Map Reduce using jboss.org projects. (*Milestone 2*)
- Use Apache Hadoop for PicketLink Analytics. (*Milestone 3*)

Define The Problem Space

- Big Data is a growing reality.
- Data, data, data is everywhere.
- Distinguish between structured and unstructured data.
 - Structured Data → Customer Data
 - Unstructured Data → web clicks, logs, social data

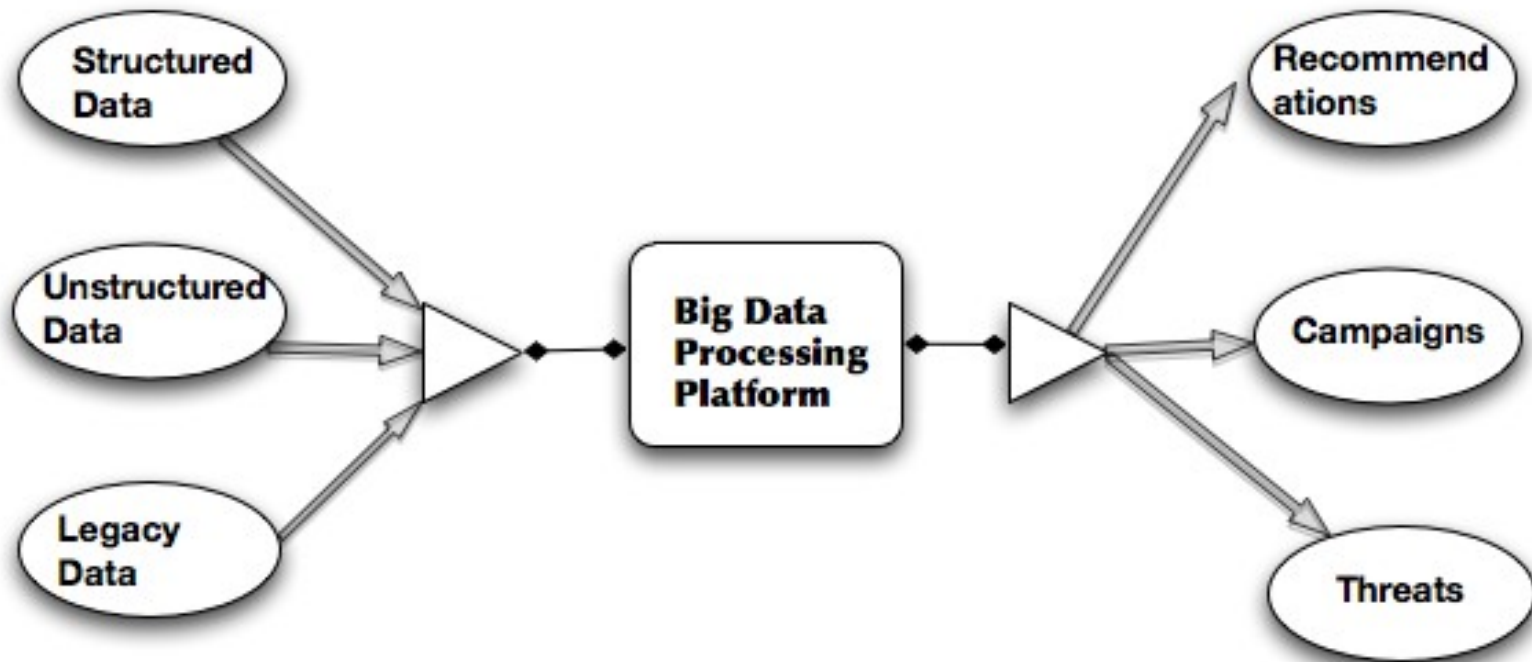
Big Data Analytics

- Data Warehouses hold unprocessed data.
- Data Marts hold processed/analyzed data.

Big Data Analytics

- Structured Data
 - Customer Records, Ordering System, News Feeds
- Semi-structured Data
 - Parsed Logs, Sales History, Click Analytics
- Unstructured Data
 - Raw Log Files, Images, Documents, Social Media Posts.

Big Data Analytics

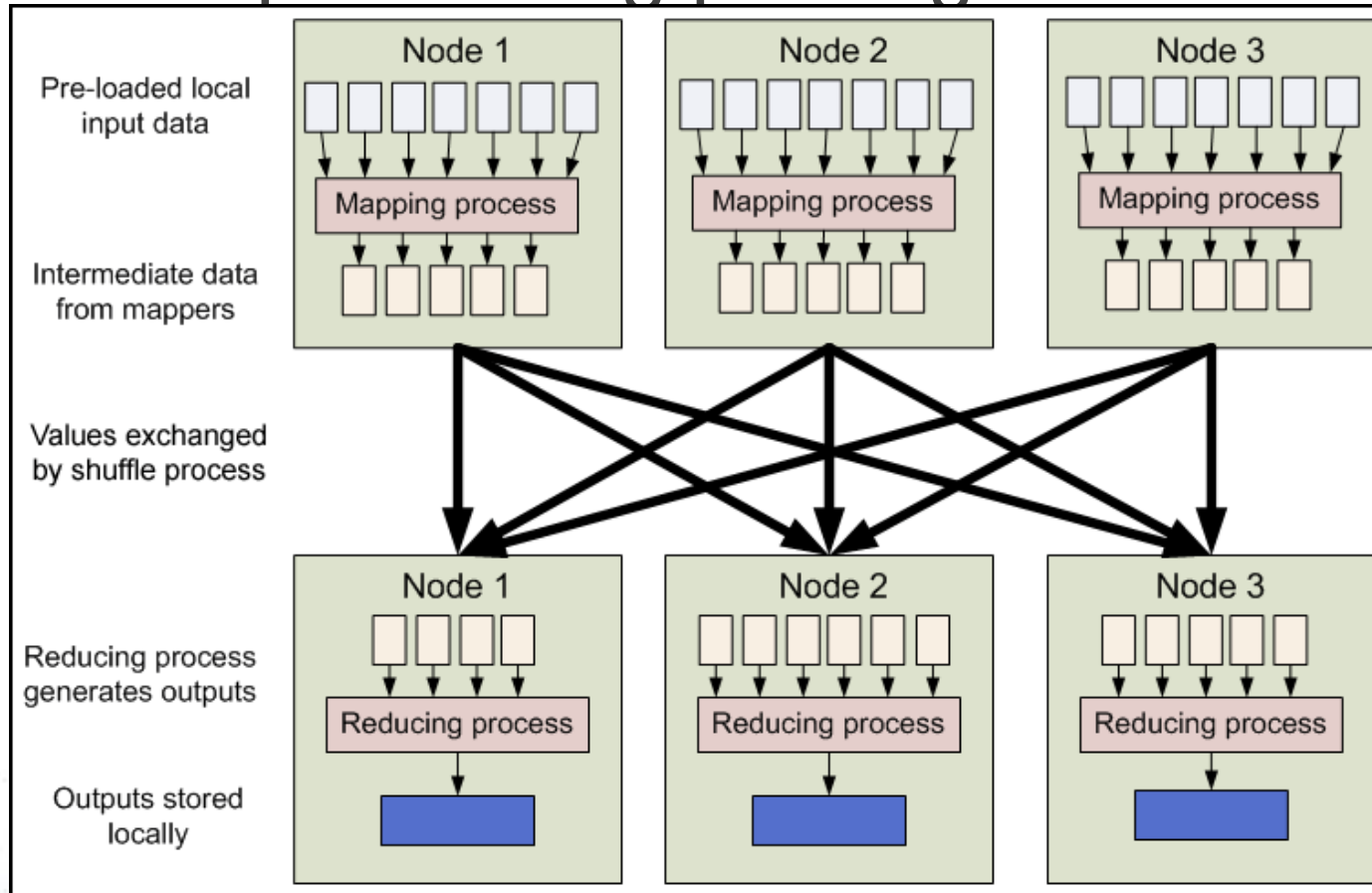


Apache Hadoop Ecosystem

- Open Source Project at the Apache Software Foundation.
- Hadoop Common, HDFS
- Other projects such as Pig, HBase, Zookeeper are relevant

Hadoop Map Reduce

- Core processing paradigm.



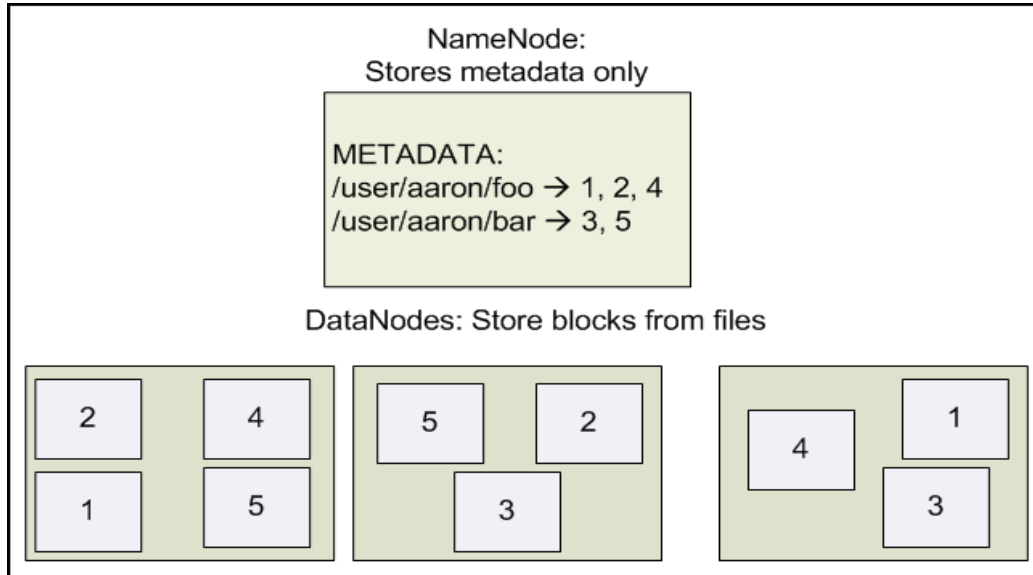
Hadoop Map Reduce

- Three modes of operation.
 - Local (standalone) mode
 - Pseudo-distributed mode
 - Fully distributed mode

Hadoop HDFS

- Core storage paradigm.
 - Eg: 100TB file as 1 file.
- Distributed File System.
- Stores files as blocks.
- Default block size is 64MB.
- Random Reads, Parallel Reads.
- Redundancy.
 - Each block stored 3 times

Hadoop HDFS



Source: <http://developer.yahoo.com/hadoop/tutorial/module2.html>

Hadoop HDFS

- NameNode
 - Directory of data blocks
 - High Memory Consumption.
 - State is saved to disk.
- Secondary NameNode
 - Backup for NameNode.
 - Takes time to become active.

Apache Pig

- Excellent ETL (Extract, Transform, Load) paradigm.
- Can use Map Reduce internally.
- Slightly slower than M/R direct.
- Saves programming needs.

Apache HBase

- Column Database.
- Uses HDFS for storage.
- Map Oriented/ Key-value pairs.
- Similar to Google BigTable.

Apache Zookeeper

- Management platform for Hadoop.

Tinkering with Hadoop

- Hadoop core code base is undergoing massive changes.
- Released.
 - V0.20
 - V1.0.x
 - V2.0 alpha (in development)
- Pointless to fork Hadoop code base.
 - It is going to change anyway. :(

Example of running Hadoop in Pseudo-distributed Mode

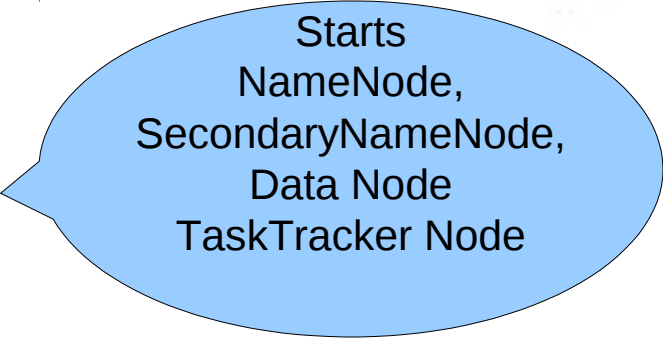
Hadoop Pseudo-distributed Mode

- One Node setup.
- Configure core-site.xml, hdfs-site.xml, mapred-site.xml
- Setup passphraseless ssh
- Format a DFS
 - `/bin/hadoop namenode -format`

http://hadoop.apache.org/common/docs/r1.0.3/single_node_setup.html

Hadoop Pseudo-distributed Mode

- Startup hadoop daemons
 - /bin/start-all.sh
- Open your browser to two tabs
 - HDFS Health <http://localhost:50070/dfshealth.jsp>
 - Job Tracker <http://localhost:50030/jobtracker.jsp>



Starts
NameNode,
SecondaryNameNode,
Data Node
TaskTracker Node

Hadoop Pseudo-distributed Mode

- Copy input files into HDFS
 - `/bin/hadoop fs -put input input`
- Run the wordcount M/R
 - `/bin/hadoop jar ../hadoop-examples-*.jar wordcount input output`

Hadoop Pseudo-distributed Mode

- Copy output files from HDFS into local
 - `/bin/hadoop fs -get output pseudoOutput`
- Now peek inside pseudoOutput dir
- If you want to delete any HDFS directory,
 - `/bin/hadoop fs -rmr output`
- Stop all Hadoop Daemons
 - `/bin/stop-all.sh`

Summary : We have
understood Apache Hadoop
ecosystem
(Milestone 1)

Experiments using JBoss Community Projects

Hadoop on JBoss AS 7

- Not a good fit.
- Hadoop is primarily used for batch processing.
- JavaEE currently has no direct batch processing capabilities.
 - JSR 352 in the works.

Drools on Hadoop

- Drools Expert is a rules engine from JBoss community.
- Drools Core can be used to introduce rules in Map Reduce programs.

Infinispan on Hadoop

- Infinispan is an extremely scalable, highly available data grid platform.
- Infinispan is a good data grid infrastructure for Map Reduce programs.

Use Case: Hadoop M/R, Twitter Feeds, Drools and Infinispan

Use Case

- Offline obtained a few twitter feeds with search terms – JBoss, JUDCon, Aerogear, Infinispan etc
- Map Reduce uses drools to see if a particular tweet contains desired search terms.
- If a tweet matches, it is put on a distributed Infinispan clustered cache.

Use Case

- Demo
 - Basically rule based batch processing
 - Distributed cache for distribution

Other Possibilities

- PicketBox XACML Engine
 - Access control inside your Map/Reduce programs.
- Hibernate
 - JPA stuff for your M/R.

Summary: Run Apache
Map Reduce using
JBoss.org projects.
(Milestone 2)

Use Case: PicketLink Log Analysis using Apache Pig

PicketLink Log Analysis

- PicketLink Log
 - Logs generated at the Service Provider
 - Logs generated at the Identity Provider

PicketLink Log Analysis

- PicketLink Log

- Logs generated at the Service Provider

- 13:14:06,862 INFO [org.picketlink.identity.federation.audit] (http--127.0.0.1-8080-1) PLFED000200: [PicketLink Audit] /sales-post-sig REQUEST_TO_IDP [Info]

- 13:14:09,042 INFO [org.picketlink.identity.federation.audit] (http--127.0.0.1-8080-2) PLFED000200: [PicketLink Audit] /idp-sig CREATED_ASSERTION ID_d4aaa7be-d19c-4136-853f-6b016d17570b [Info]

- 13:14:09,056 INFO [org.picketlink.identity.federation.audit] (http--127.0.0.1-8080-2) PLFED000200: [PicketLink Audit] /idp-sig RESPONSE_TO_SP http://localhost:8080/sales-post-sig/ [Info]

- 13:14:09,092 INFO [org.picketlink.identity.federation.audit] (http--127.0.0.1-8080-2) PLFED000200: [PicketLink Audit] /sales-post-sig RESPONSE_FROM_IDP tomcat [Info]

- 13:14:11,012 INFO [org.picketlink.identity.federation.audit] (http--127.0.0.1-8080-2) PLFED000200: [PicketLink Audit] /sales-post-sig REQUEST_TO_IDP [Info]

- 13:14:11,044 INFO [org.picketlink.identity.federation.audit] (http--127.0.0.1-8080-2) PLFED000200: [PicketLink Audit] /idp-sig RESPONSE_TO_SP http://localhost:8080/sales-post-sig/ [Info]

- 13:14:11,120 INFO [org.picketlink.identity.federation.audit] (http--127.0.0.1-8080-2) PLFED000200: [PicketLink Audit] /sales-post-sig REQUEST_TO_IDP [Info]

PicketLink Log Analysis

- PicketLink Log
 - Logs generated at the Service Provider
 - Using Apache Pig to generated reports

PicketLink Log Analysis

-- Load the PicketLink Log file

```
file = LOAD 'picketlink.log' USING PigStorage('\n') AS (entry: chararray);
```

-- Trim the entries loaded

```
trimmedfile = FOREACH file GENERATE TRIM(entry) as entry;
```

-- Filter the log entries to the desired pattern

```
selectedrows = FILTER trimmedfile BY (entry matches '.*RESPONSE_FROM_IDP.*');  
dump selectedrows;
```

-- Store the data into intermediate file

```
STORE selectedrows into 'selectedrows' USING PigStorage('');
```


PicketLink Log Analysis

-- Load the intermediate data

```
data = load 'selectedrows/part*' USING PigStorage(' ') AS (timestamp: chararray, info1:  
chararray, audit : chararray, plnum: chararray, thread: chararray, bracks1: chararray, bracks2:  
chararray, ignore: chararray, endpoint: chararray, event: chararray, username: chararray,  
info2: chararray);  
describe data;  
dump data;
```

-- Generate a tuple of endpoint vs. username

```
mytuple = FOREACH data GENERATE TOTUPLE(username,endpoint);  
dump mytuple;
```

-- Store into results

```
STORE mytuple into 'results' using PigStorage(' ');
```

PicketLink Log Analysis

- PicketLink Log Results

- (tomcat,/sales-post-sig)

- (tomcat,/sales-redirect-sig)

- (tomcat,/sales-post)

- (tomcat,/sales-redirect)

Summary: PicketLink Log
Analysis can be done using
Apache Pig
(*Milestone 3*)

Closing Thoughts

- If you just need Map Reduce, the Data Grid API from Infinispan has a much cleaner API.
- You can use any JBoss Community project as part of your Map/Reduce programs.
- GlusterFS for NameNode robustness.

Resources

- Blogs:
 - <http://everythingbigdata.blogspot.com>
 - <http://anil-identity.blogspot.com>
- Apache Hadoop Website.
 - <http://hadoop.apache.org>
- Project Infinispan
 - <http://www.jboss.org/infinispan/>
- Project Drools
 - <http://www.jboss.org/drools/>