Visual Analytics Sandbox

Satya Katragadda

January 25, 2018

Agenda

- Why Big Data?
- Goals
- Visual Analytics Sandbox
- Traditional Workflow in a Big Data Environment
- VA Sandbox: Software Stack
- VA Sandbox: Execution Examples

Why Big Data?

- Reports, e.g.,
  - Track business processes, transactions
- Diagnosis, e.g.,
  - Why is user engagement dropping?
  - Why is the system slow?
  - Detect spam, worms, viruses, DDoS attacks
- Decisions, e.g.,
  - Decide what feature to add
  - Decide what ad to show
  - Block worms, viruses, …

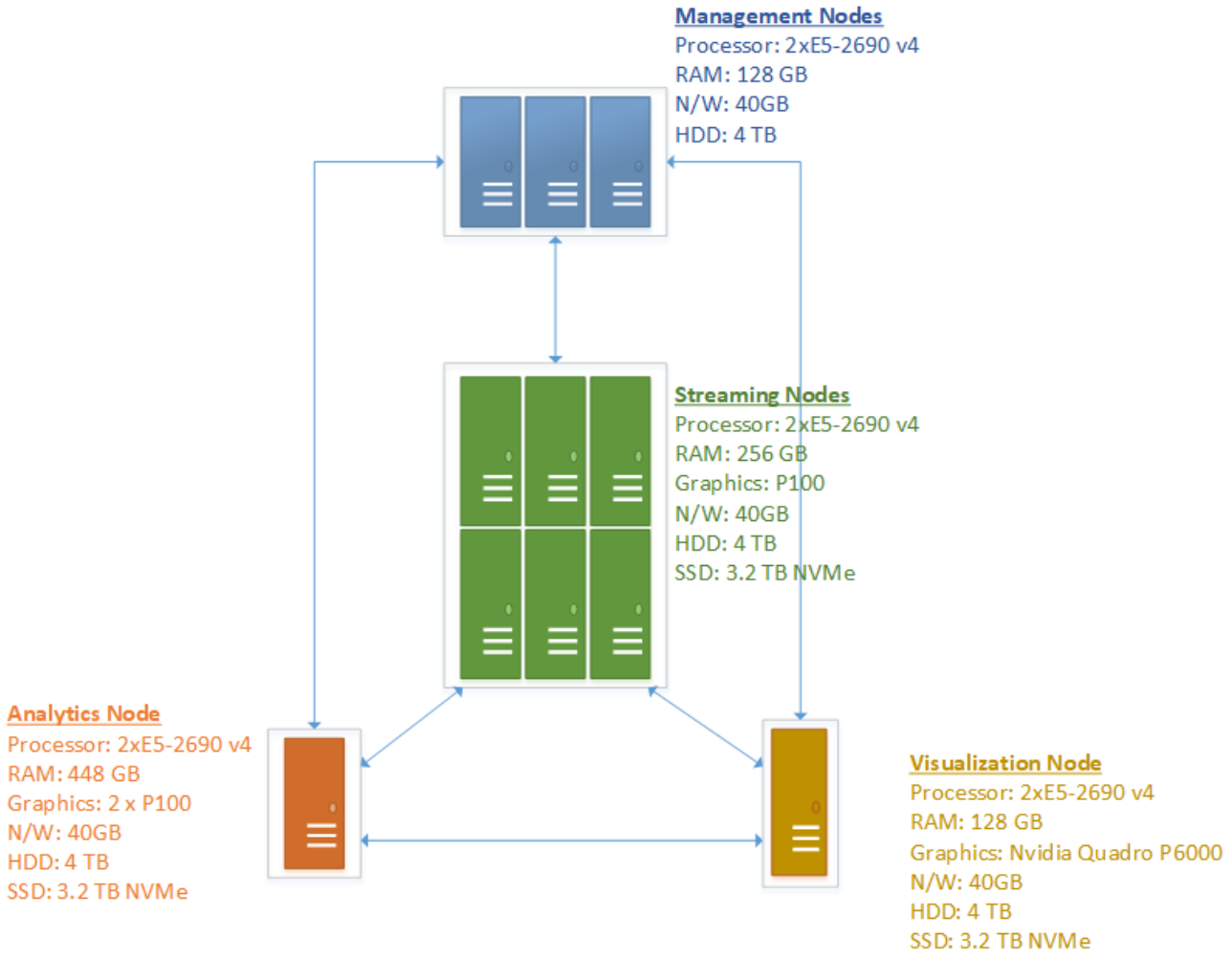
Goals

- **Low latency (interactive) queries on historical data:** enable faster decisions
  - E.g., identify why a site is slow and fix it

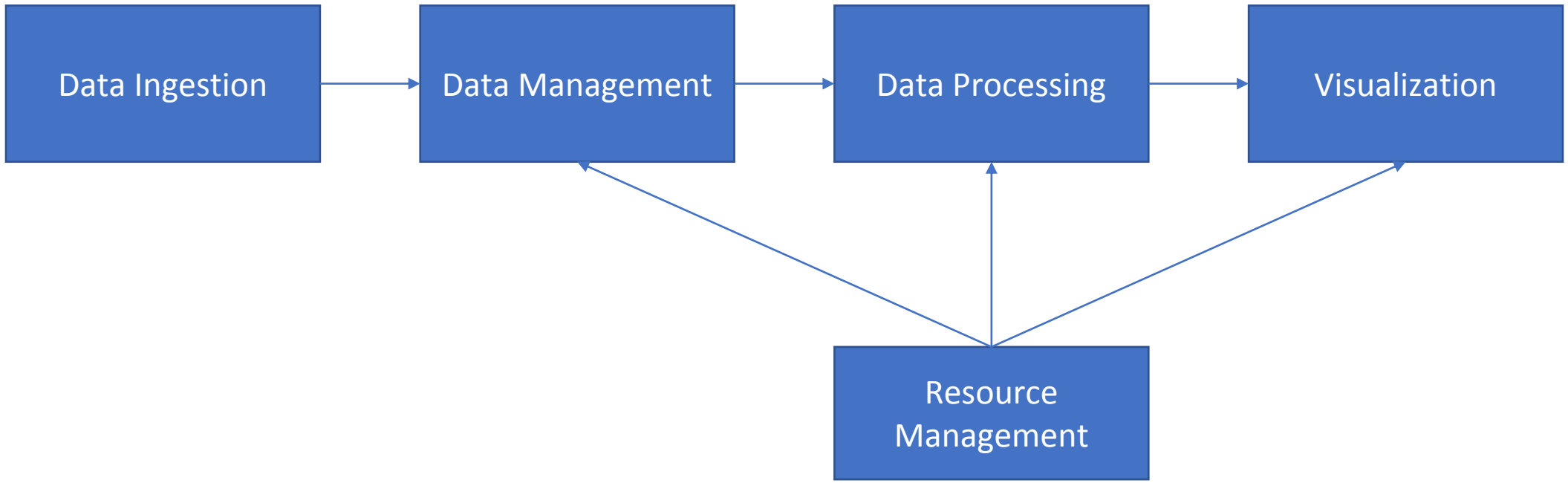
- **Low latency queries on live data (streaming):** enable decisions on real-time data
  - E.g., detect & block worms in real-time (a worm may infect **1mil** hosts in **1.3sec**)

- **Sophisticated data processing:** enable "better" decisions
  - E.g., anomaly detection, trend analysis

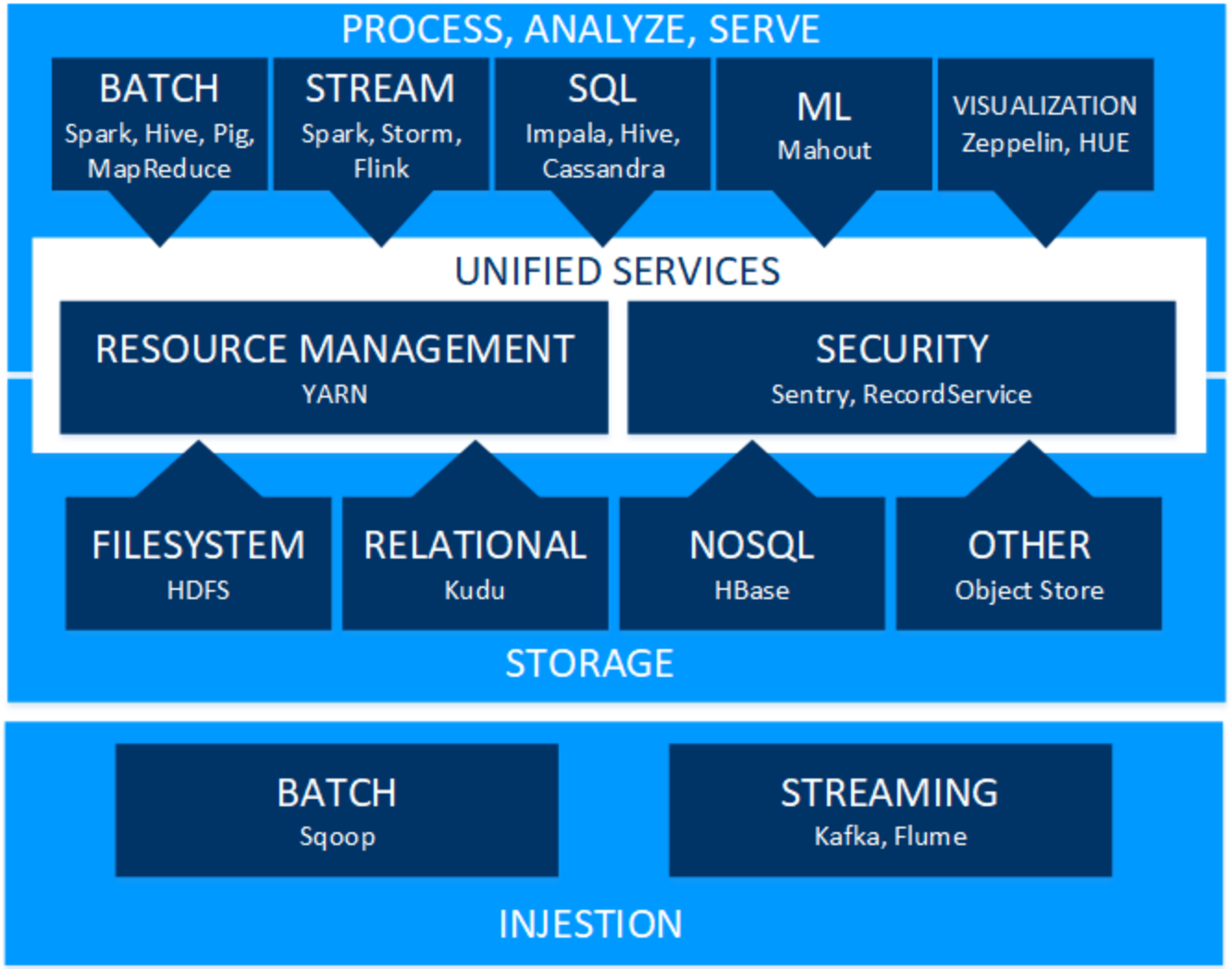
Visual Analytics Sandbox



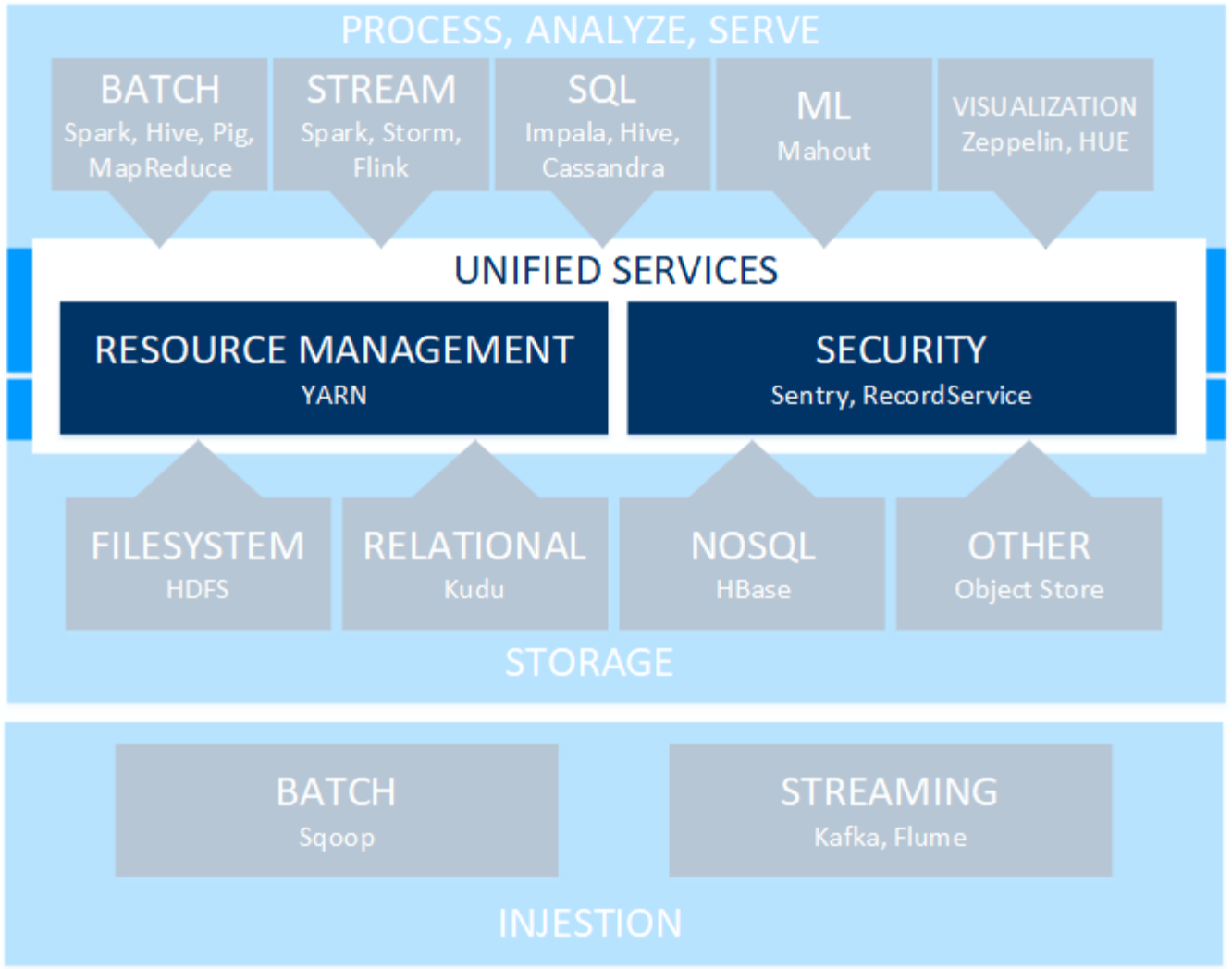
Big Data Workflow



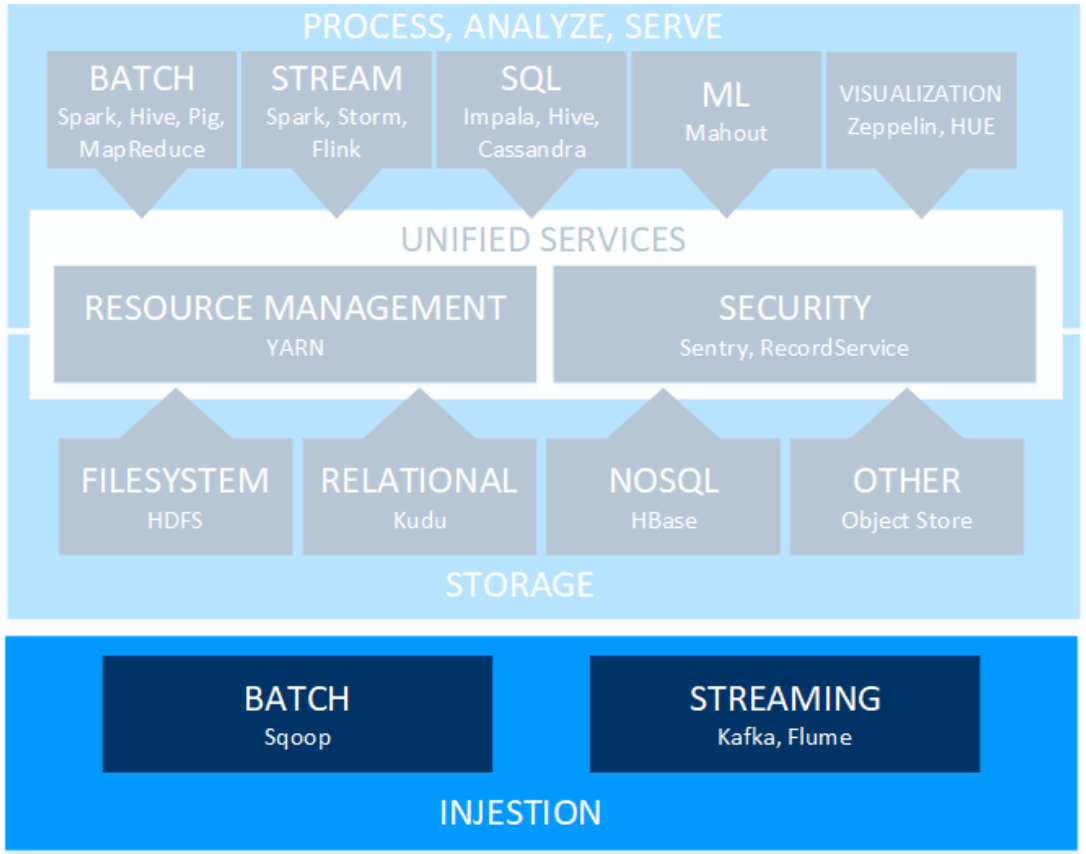
VA Sandbox: Software Stack



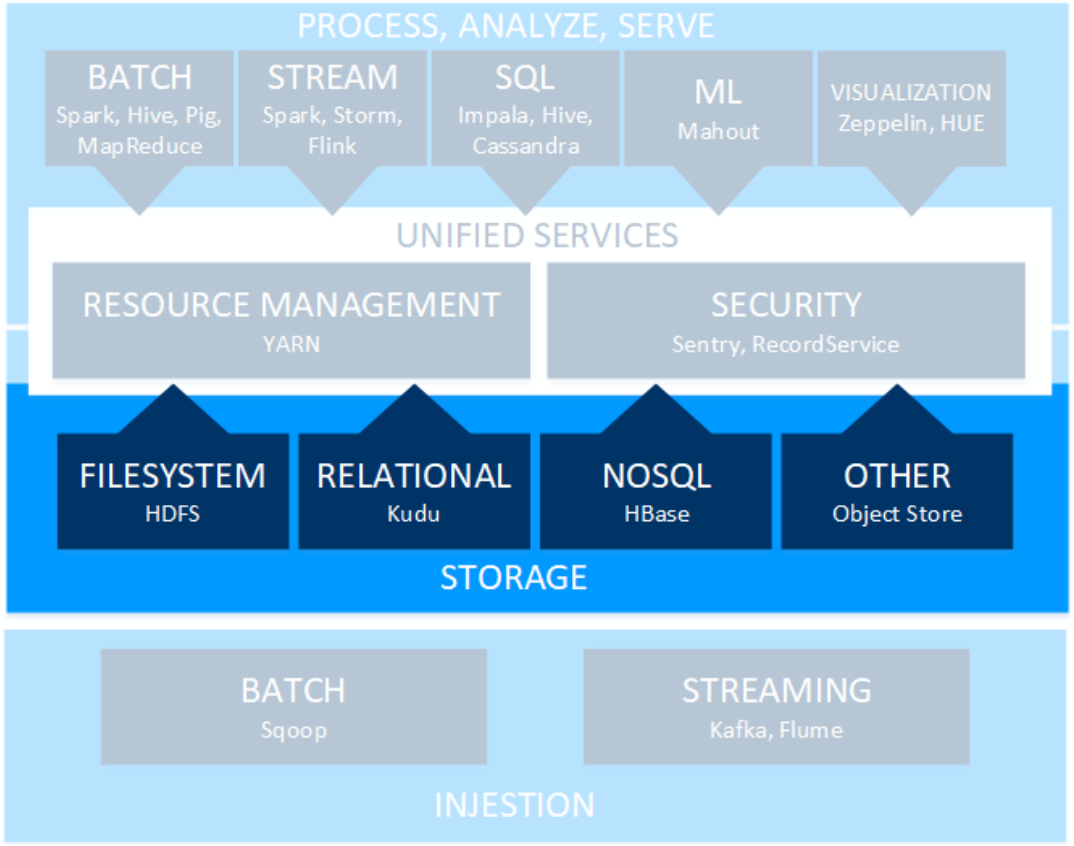
VA Sandbox: Resource Manager



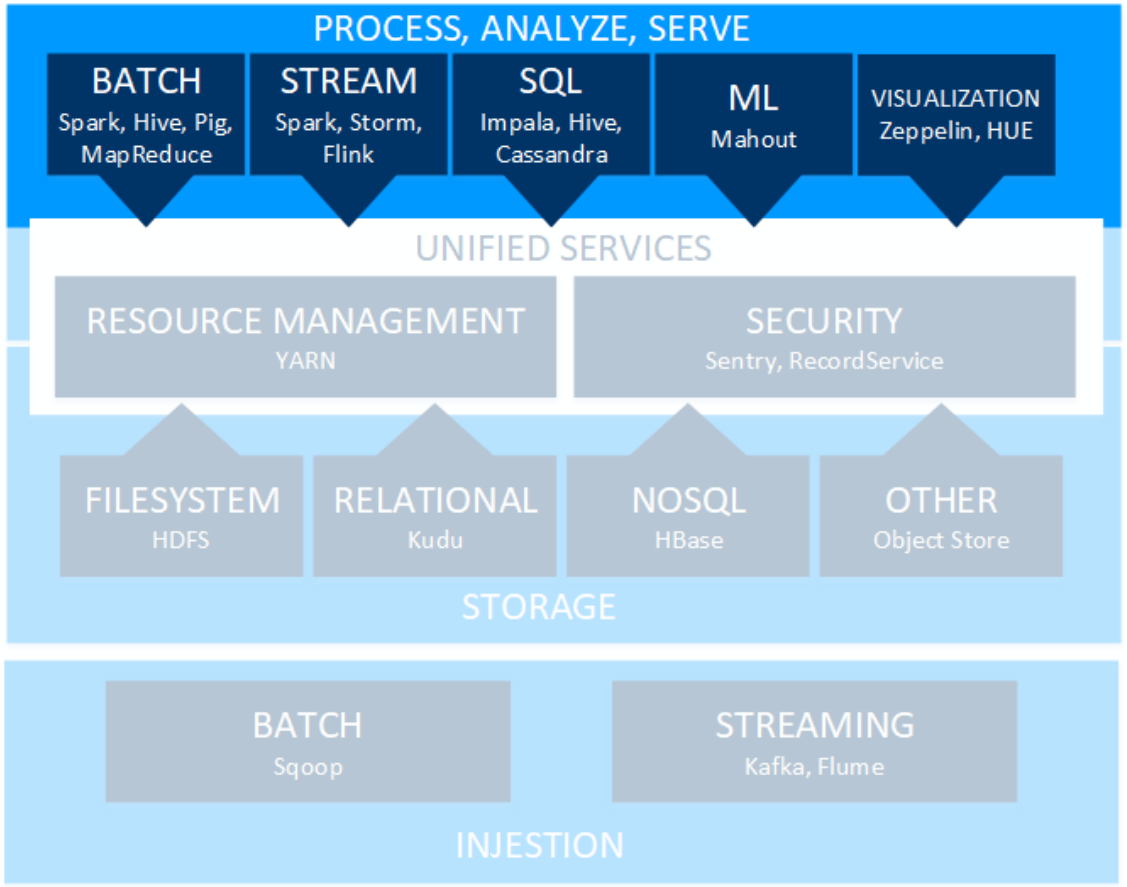
VA Sandbox: Data Injestion



VA Sandbox: Data Storage



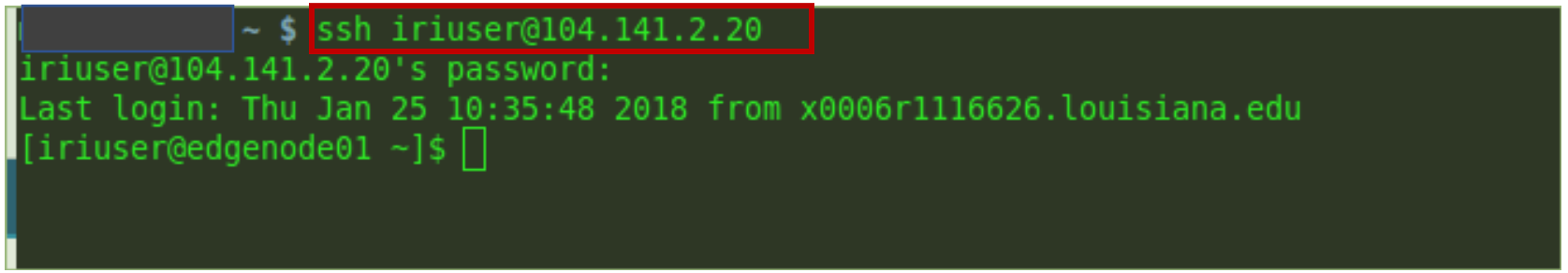
VA Sandbox: Processing and Visualization



VA Sandbox

- Stephens Hall
- Accessible through university network

VA Sandbox: Access



VA Sandbox: Execution

```
[iriuser@edgenode01 ~]$ pyspark
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkContext available as sc, HiveContext available as sqlContext.
>>> quit()
[iriuser@edgenode01 ~]$ spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Scala version 2.10.5 (Java HotSpot(TM) 64-Bit Server VM, Java 1.7.0_67)
Type in expressions to have them evaluated.
Type :help for more information.
Spark context available as sc (master = yarn-client, app id = application_1513254681785_0008).
SQL context available as sqlContext.

scala>
```

VA Sandbox: Input

```
[iriuser@edgenode01 ~]$ hdfs dfs -put sample input.txt hdfs://masternode01:8020/user/iriuser/
Warning: fs.defaultFS is not set when running "put" command.
[iriuser@edgenode01 ~]$ hdfs dfs -ls hdfs://masternode01:8020
Warning: fs.defaultFS is not set when running "ls" command.
Found 3 items
drwx------   - iriuser iriuser          0 2018-01-25 11:51 hdfs://masternode01:8020/user/iriuser/.Trash
drwxr-xr-x   - iriuser iriuser          0 2018-01-25 11:44 hdfs://masternode01:8020/user/iriuser/.sparkStaging
-rw-r--r--   3 iriuser iriuser       1339 2018-01-25 11:51 hdfs://masternode01:8020/user/iriuser/sample_input.txt
[iriuser@edgenode01 ~]$
```

VA Sandbox: Spark Script

```
[iriuser@edgenode01 ~]$ pyspark
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 1.6.0
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkContext available as sc, HiveContext available as sqlContext.
>>> myfile = sc.textFile("hdfs://masternode01:8020/user/iriuser/sample_input.txt")
>>> counts = myfile.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda v1,v2: v1 + v2)
>>> counts.saveAsTextFile("hdfs://masternode01:8020/user/iriuser/sample_output.txt")
```

VA Sandbox: Spark Output

```
iriuser@edgenode01 ~]$ hdfs dfs -cat hdfs://masternode01:8020/user/iriuser/sample_output.txt/part-*
Warning: fs.defaultFS is not set when running "cat" command.
u'', 4)
u'and', 2)
u'Latent', 1)
u'words', 1)
u'classification', 1)
u'predict', 1)
u'is', 11)
u'zipfian', 1)
u'topic', 1)
u'/usr/share/dict/linux.words.', 1)
u'including', 1)
u'file', 1)
u'(ALS)', 1)
u'different', 1)
u'from', 2)
u'collaborative', 1)
u'for', 3)
```

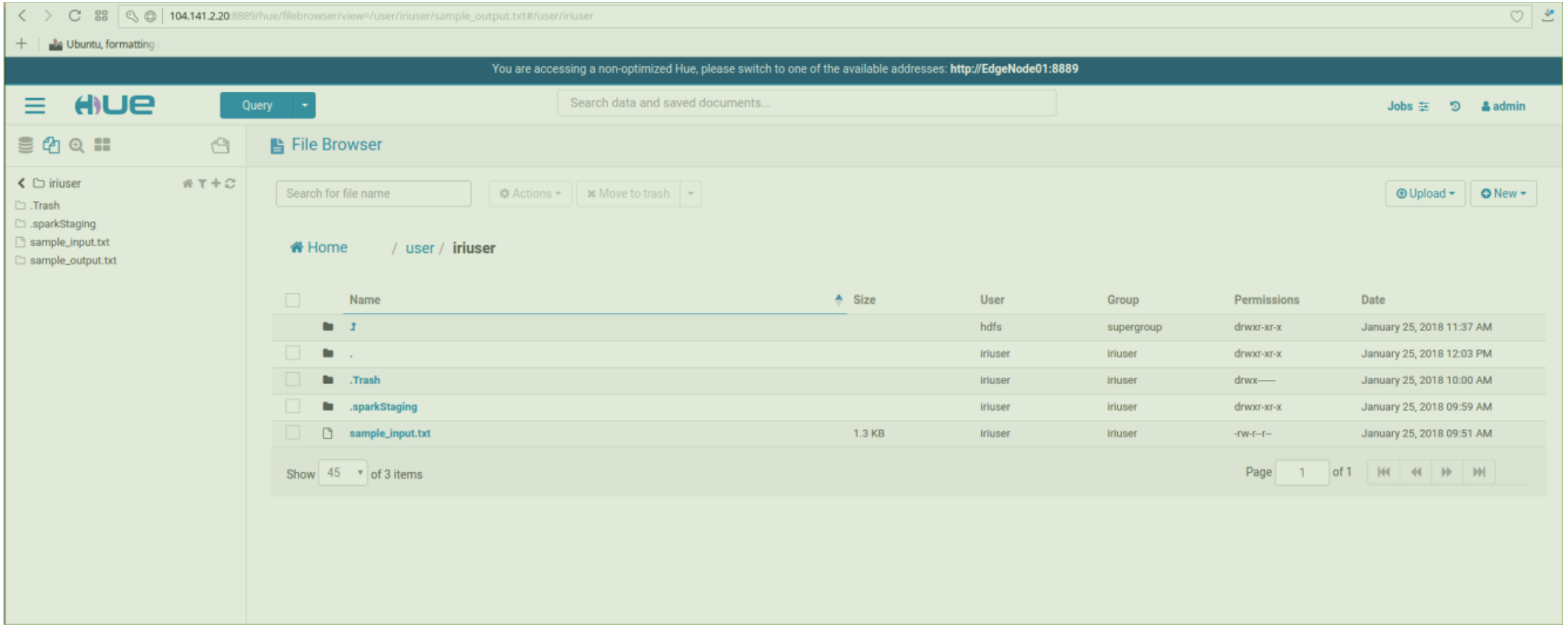
Alternative Execution Environment

HUE: Hadoop User Experience

An open-source Web interface that supports Apache Hadoop and its ecosystem

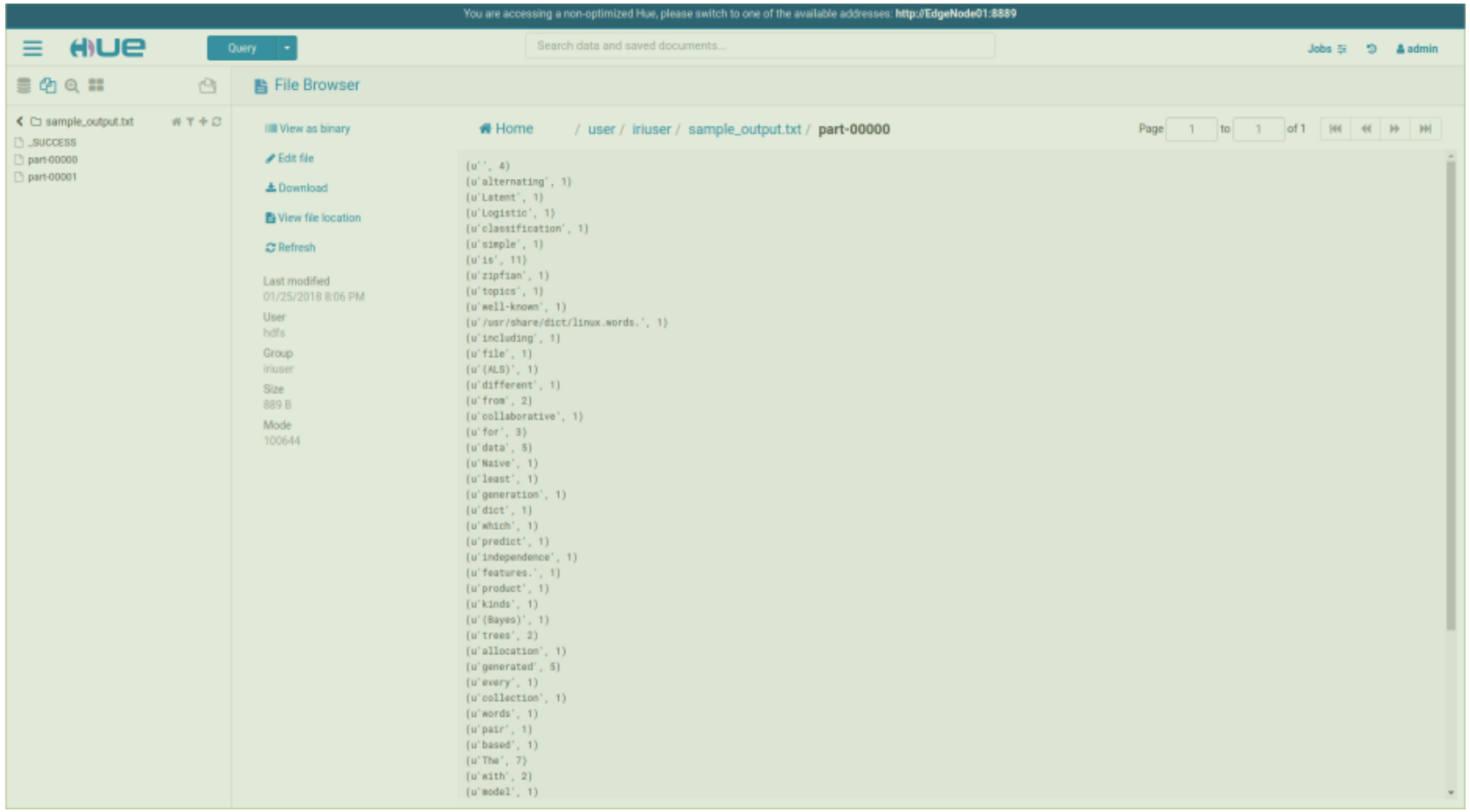
Component	Applications
Editor	SQL, Pig, Spark
Browsers	YARN, Oozie, Impala, HBase, Livy
Scheduler	Oozie
Dashboard	Solr, SQL (Impala, Hive...)

HUE: File Browser

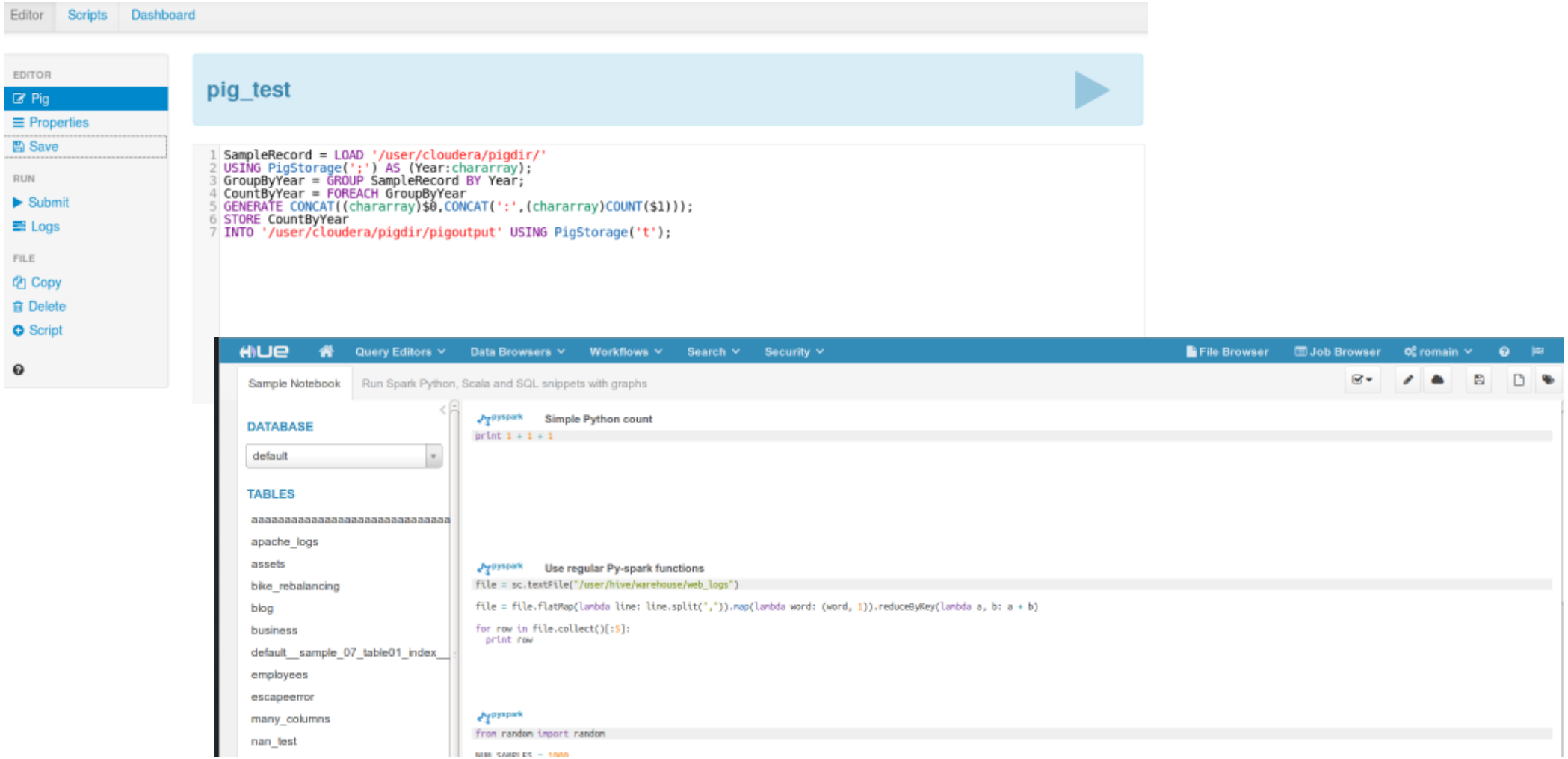


HUE: Job Execution

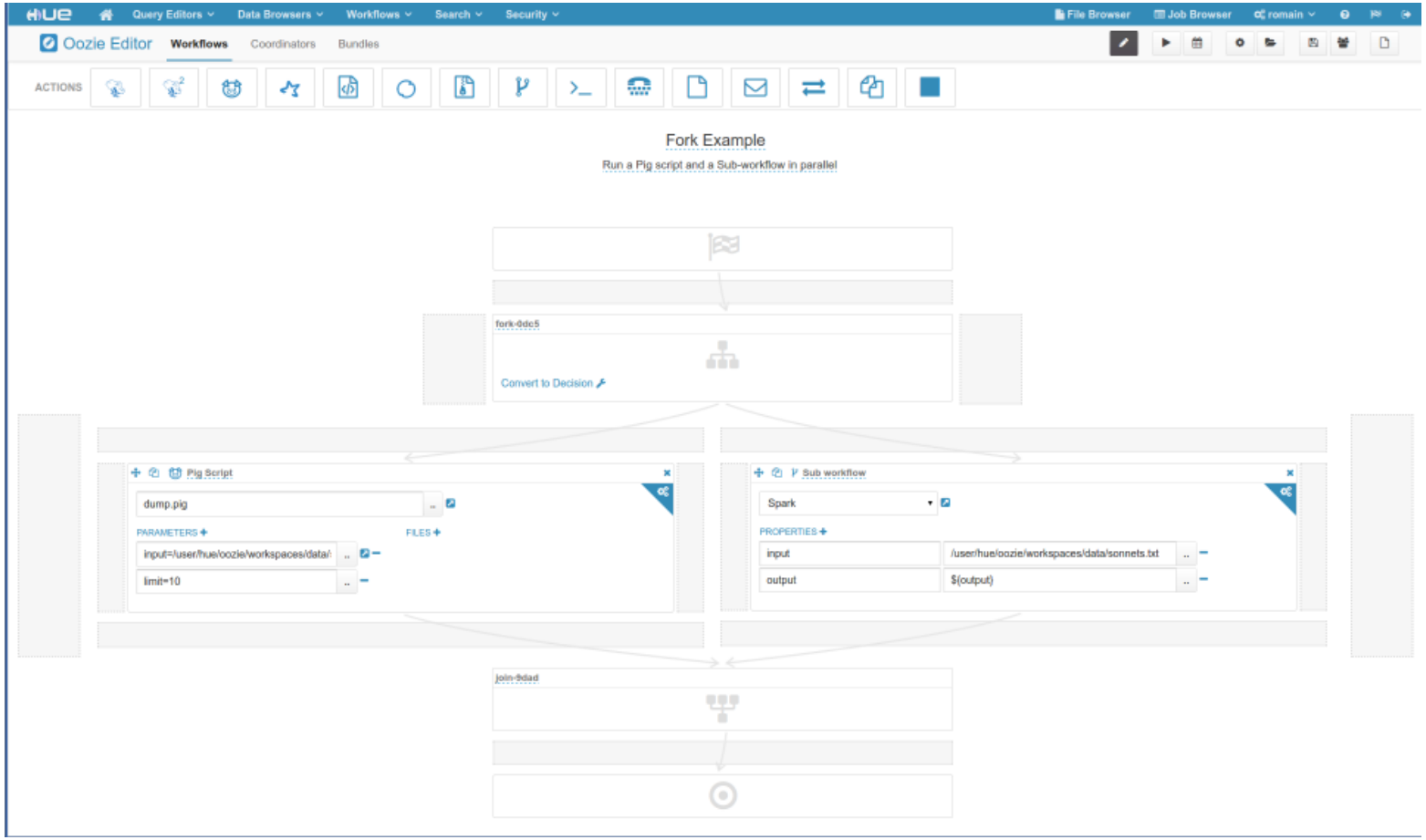
HUE: Output


You are accessing a non-optimized Hue, please switch to one of the available addresses: http://EdgeNode01:8889
HUE
Query
Search data and saved documents...
Jobs
admin
File Browser
sample_output.txt
_SUCCESS
part-00000
part-00001
View as binary
Edit file
Download
View file location
Refresh
Last modified
01/25/2018 8:06 PM
User
hdfs
Group
iriuser
Size
889 B
Mode
100644
Home / user / iriuser / sample_output.txt / part-00000
Page 1 to 1 of 1
(u'', 4)
(u'alternating', 1)
(u'Latent', 1)
(u'Logistic', 1)
(u'classification', 1)
(u'simple', 1)
(u'is', 11)
(u'zipfian', 1)
(u'topics', 1)
(u'well-known', 1)
(u'/usr/share/dict/linux.words.', 1)
(u'including', 1)
(u'file', 1)
(u'(ALS)', 1)
(u'different', 1)
(u'from', 2)
(u'collaborative', 1)
(u'for', 3)
(u'data', 5)
(u'Naive', 1)
(u'least', 1)
(u'generation', 1)
(u'dict', 1)
(u'which', 1)
(u'predict', 1)
(u'independence', 1)
(u'features.', 1)
(u'product', 1)
(u'kinds', 1)
(u'(Bayes)', 1)
(u'trees', 2)
(u'allocation', 1)
(u'generated', 5)
(u'every', 1)
(u'collection', 1)
(u'words', 1)
(u'pair', 1)
(u'based', 1)
(u'The', 7)
(u'with', 2)
(u'model', 1)

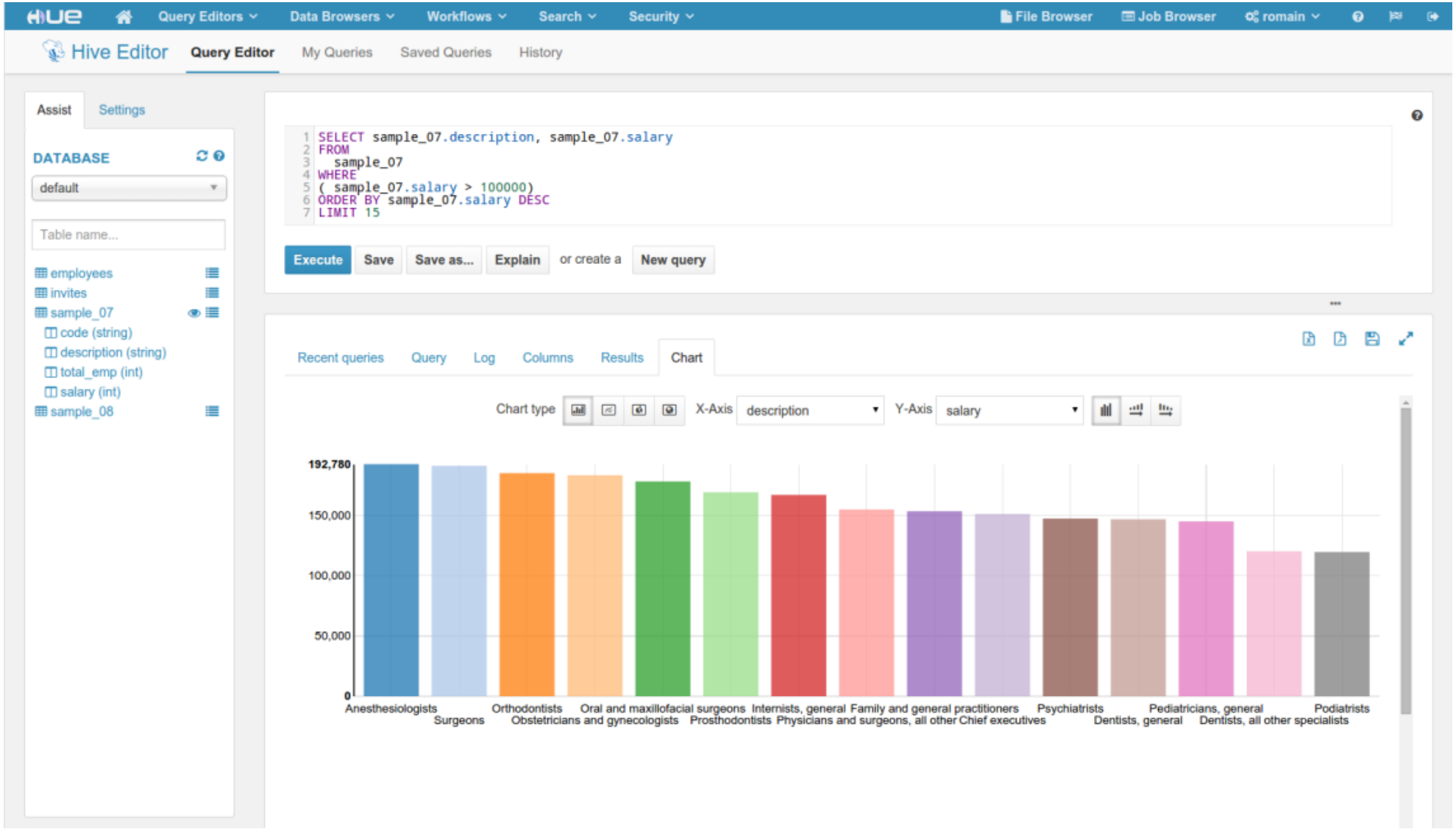
HUE: Editors


Editor
Scripts
Dashboard
EDITOR
Pig
Properties
Save
RUN
Submit
Logs
FILE
Copy
Delete
Script
pig_test
1 SampleRecord = LOAD '/user/cloudera/pigdir/'
2 USING PigStorage(';') AS (Year:chararray);
3 GroupByYear = GROUP SampleRecord BY Year;
4 CountByYear = FOREACH GroupByYear
5 GENERATE CONCAT((chararray)$0,CONCAT(':',(chararray)COUNT($1)));
6 STORE CountByYear
7 INTO '/user/cloudera/pigdir/pigoutput' USING PigStorage('t');
HUE
Query Editors
Data Browsers
Workflows
Search
Security
File Browser
Job Browser
romain
Sample Notebook
Run Spark Python, Scala and SQL snippets with graphs
DATABASE
default
TABLES
apache_logs
assets
bike_rebalancing
blog
business
default__sample_07_table01_index__
employees
escapeerror
many_columns
nan_test
Simple Python count
print 1 + 1 + 1
Use regular Py-spark functions
file = sc.textFile("/user/hive/warehouse/web_logs")
file = file.flatMap(lambda line: line.split(",")).map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)
for row in file.collect()[:5]:
print row
from random import random

HUE: Schedulers



HUE: Dashboards



Questions?

Satya Katragadda
RM 118, Abdalla Hall
satya@Louisiana.edu