

# 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

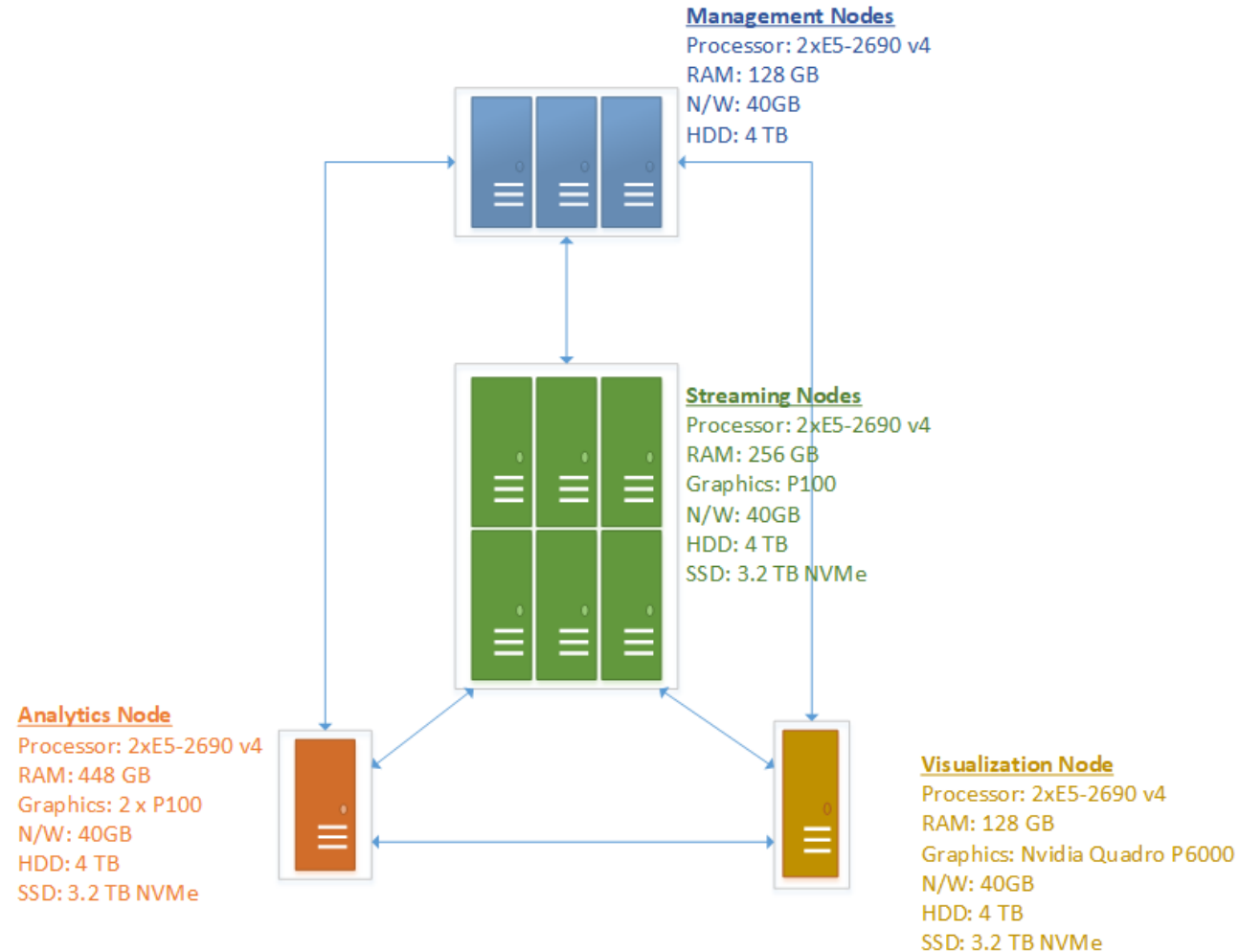
# Why Big Data?

- 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, ...

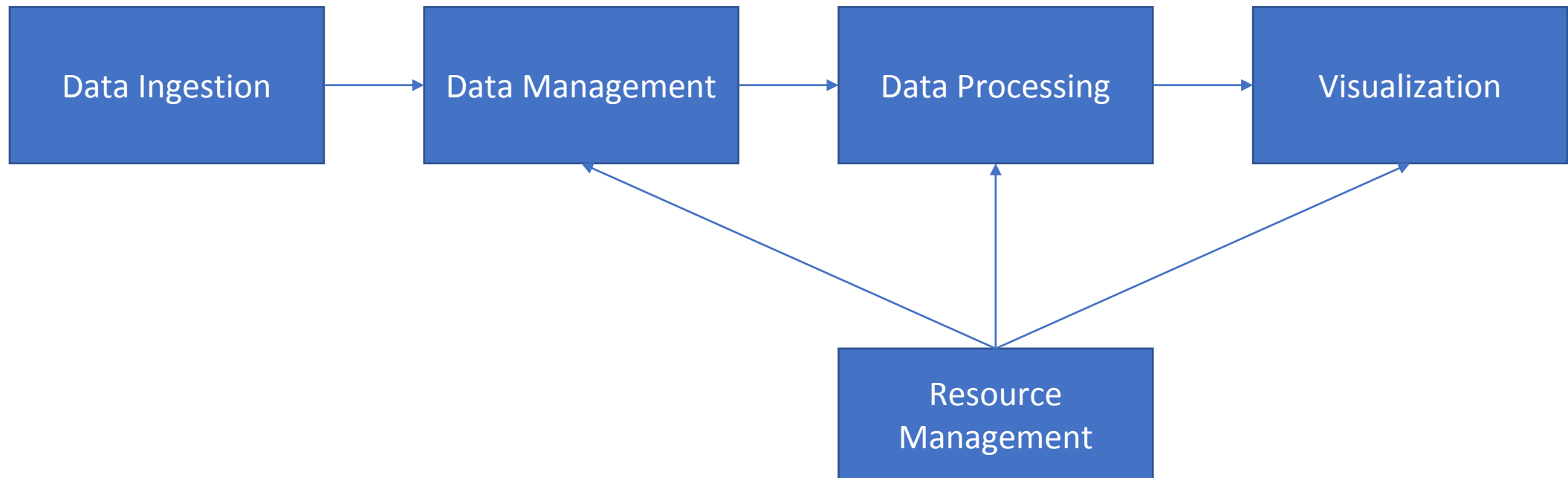
# Goals

- **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

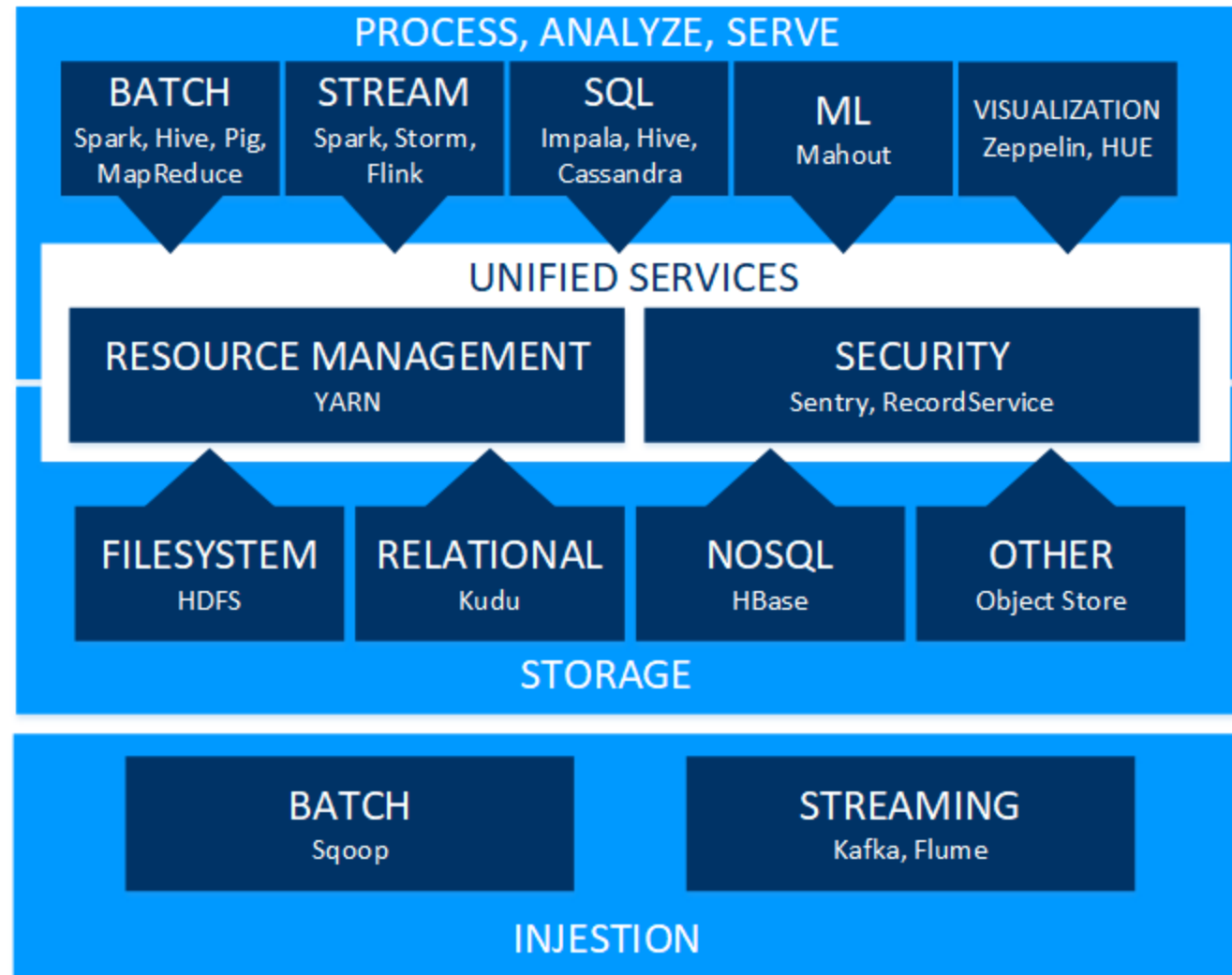
# Visual Analytics Sandbox



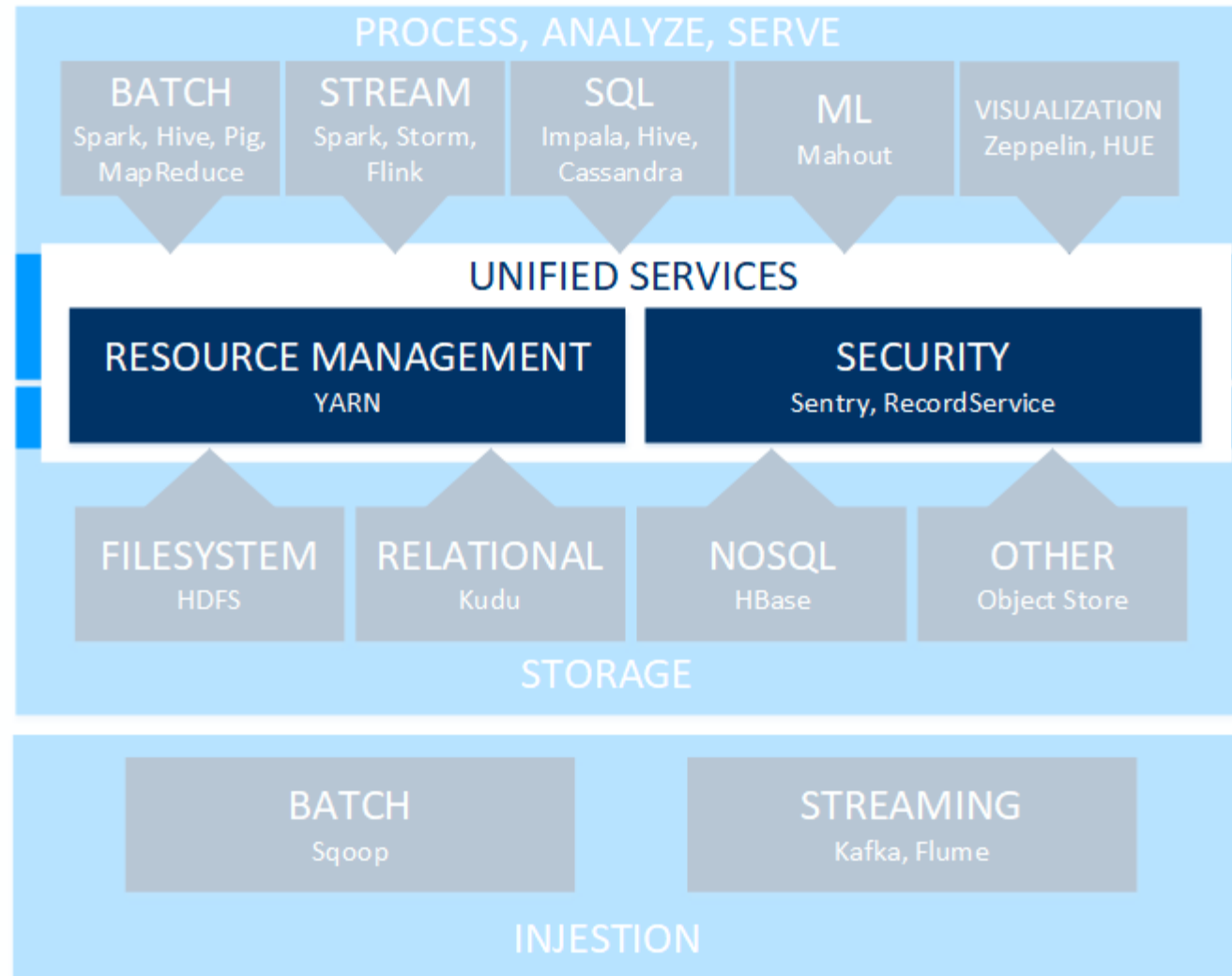
# Big Data Workflow



# VA Sandbox: Software Stack

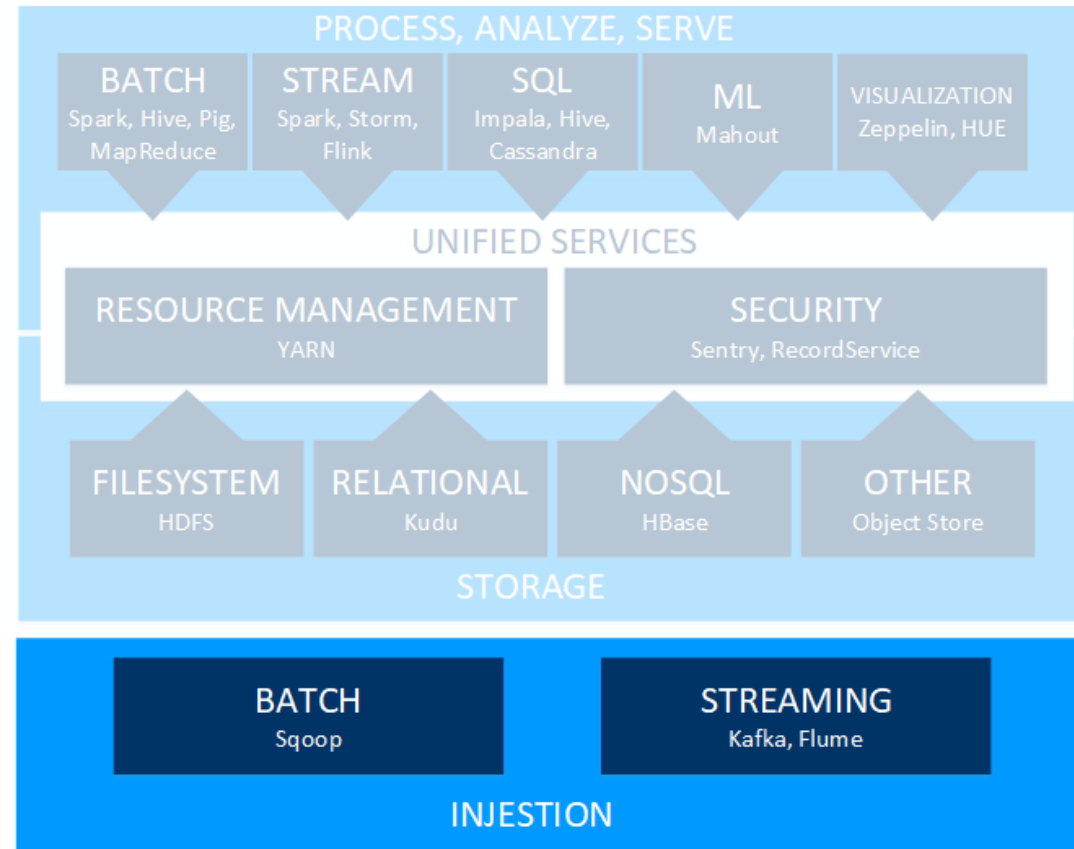


# VA Sandbox: Resource Manager

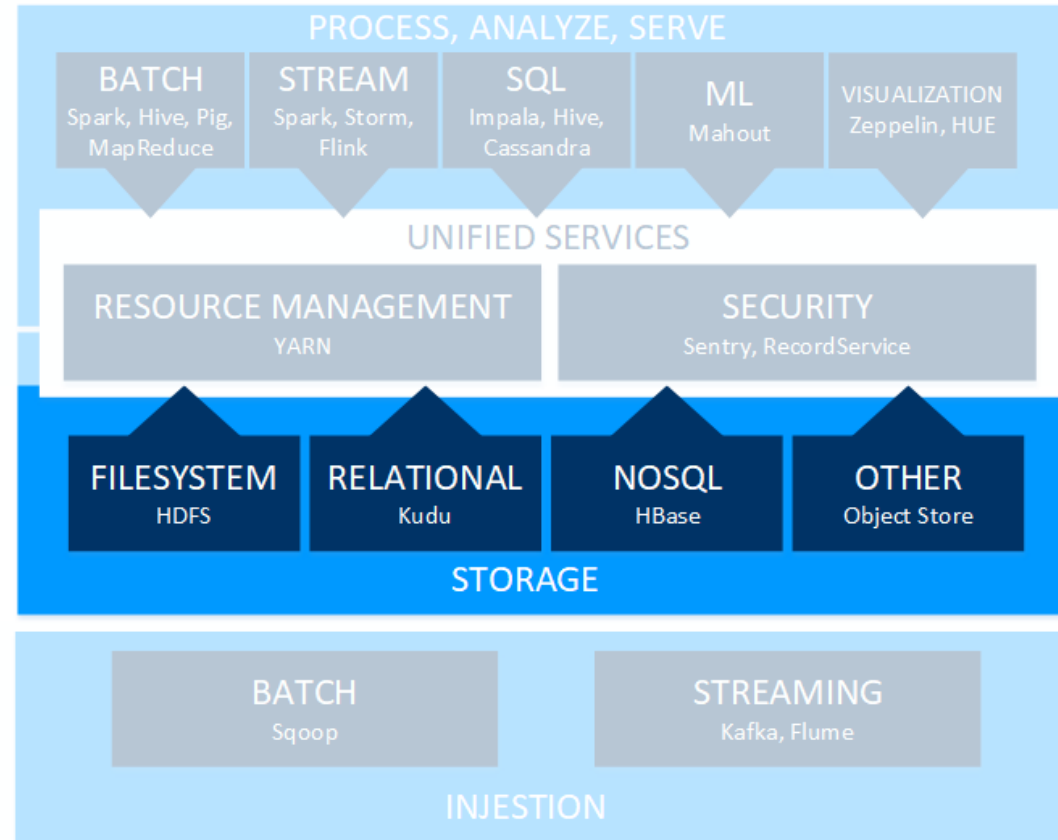




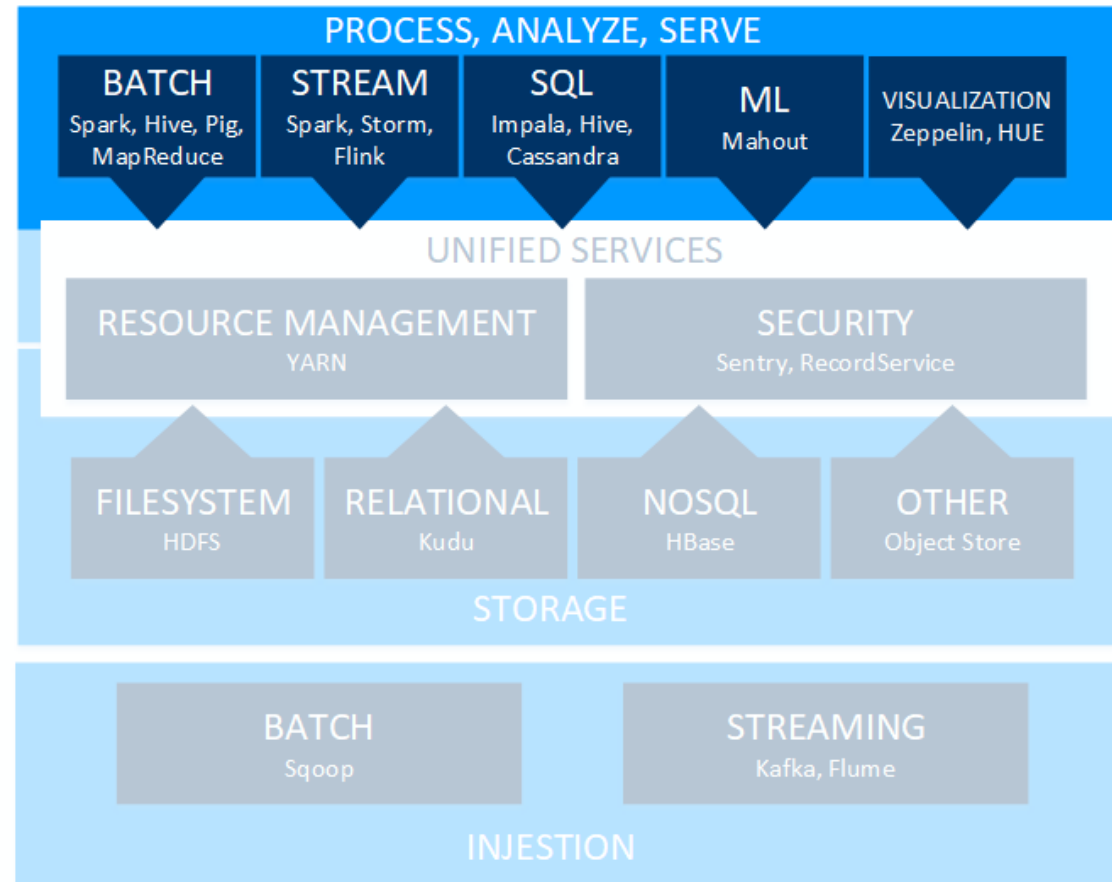
# VA Sandbox: Data Ingestion



# VA Sandbox: Data Storage



# VA Sandbox: Processing and Visualization



# VA Sandbox

- Stephens Hall
- Accessible through university network

# VA Sandbox: Access

```
~ $ ssh iriuser@104.141.2.20  
iriuser@104.141.2.20's password:  
Last login: Thu Jan 25 10:35:48 2018 from x0006r1116626.louisiana.edu  
[iriuser@edgenode01 ~]$
```

# 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

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

# HUE: File Browser

The screenshot displays the Hue File Browser interface. At the top, a browser address bar shows the URL `104.141.2.20:8889/hue/filebrowser/view=/user/irouser/sample_output.txt#/user/irouser`. Below the browser, a dark blue banner contains the Hue logo and a message: "You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://EdgeNode01:8889>".

The main interface features a navigation sidebar on the left with a tree view of the file system under the user 'irouser'. The main content area shows a directory listing for the path `/ user / irouser`. At the top of this area, there is a search bar for file names and several action buttons: "Actions", "Move to trash", "Upload", and "New".

The directory listing is presented as a table with the following columns: Name, Size, User, Group, Permissions, and Date. The items listed are:

<input type="checkbox"/>	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	<a href="#">J</a>		hdfs	supergroup	drwxr-xr-x	January 25, 2018 11:37 AM
<input type="checkbox"/>	<a href="#">.</a>		irouser	irouser	drwxr-xr-x	January 25, 2018 12:03 PM
<input type="checkbox"/>	<a href="#">.Trash</a>		irouser	irouser	drwx---	January 25, 2018 10:00 AM
<input type="checkbox"/>	<a href="#">.sparkStaging</a>		irouser	irouser	drwxr-xr-x	January 25, 2018 09:59 AM
<input type="checkbox"/>	<a href="#">sample_input.txt</a>	1.3 KB	irouser	irouser	-rw-r--	January 25, 2018 09:51 AM

At the bottom of the listing, there is a pagination control showing "Show 45 of 3 items" and "Page 1 of 1" with navigation arrows.

# HUE: Job Execution

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

SQL Add a name... Add a description...

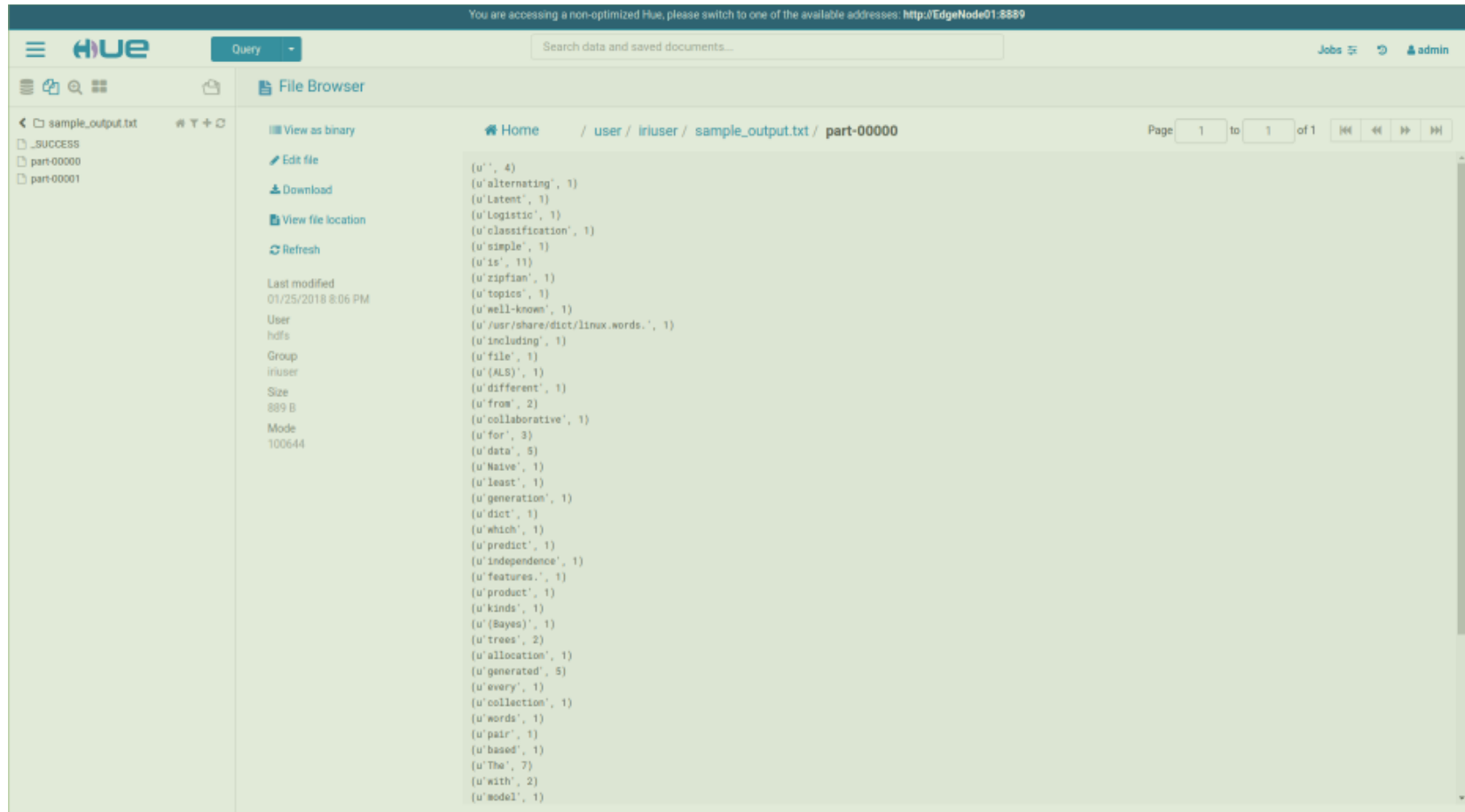
11.34s ?

```
1 myfile = sc.textFile("hdfs://masternode01:8020/user/irouser/sample_input.txt")
2 counts = myfile.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda v1,v2: v1 + v2)
3 counts.saveAsTextFile("hdfs://masternode01:8020/user/irouser/sample_output.txt")
```

Query History Saved Queries Results (1)

Done.

# HUE: Output



You are accessing a non-optimized Hue, please switch to one of the available addresses: <http://EdgeNode01.8889>

Query

Search data and saved documents...

Jobs admin

File Browser

sample\_output.txt

- part-00000
- part-00001

View as binary

Edit file

Download

View file location

Refresh

Last modified: 01/25/2018 8:06 PM

User: iruser

Group: iruser

Size: 889 B

Mode: 100644

Home / user / iruser / 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

The screenshot displays the HUE web interface. At the top, there are navigation tabs for 'Editor', 'Scripts', and 'Dashboard'. On the left side, a sidebar menu is visible with sections for 'EDITOR' (containing 'Pig', 'Properties', and 'Save'), 'RUN' (containing 'Submit', 'Logs'), and 'FILE' (containing 'Copy', 'Delete', and 'Script').

The main content area is titled 'pig\_test' and contains a Pig script:

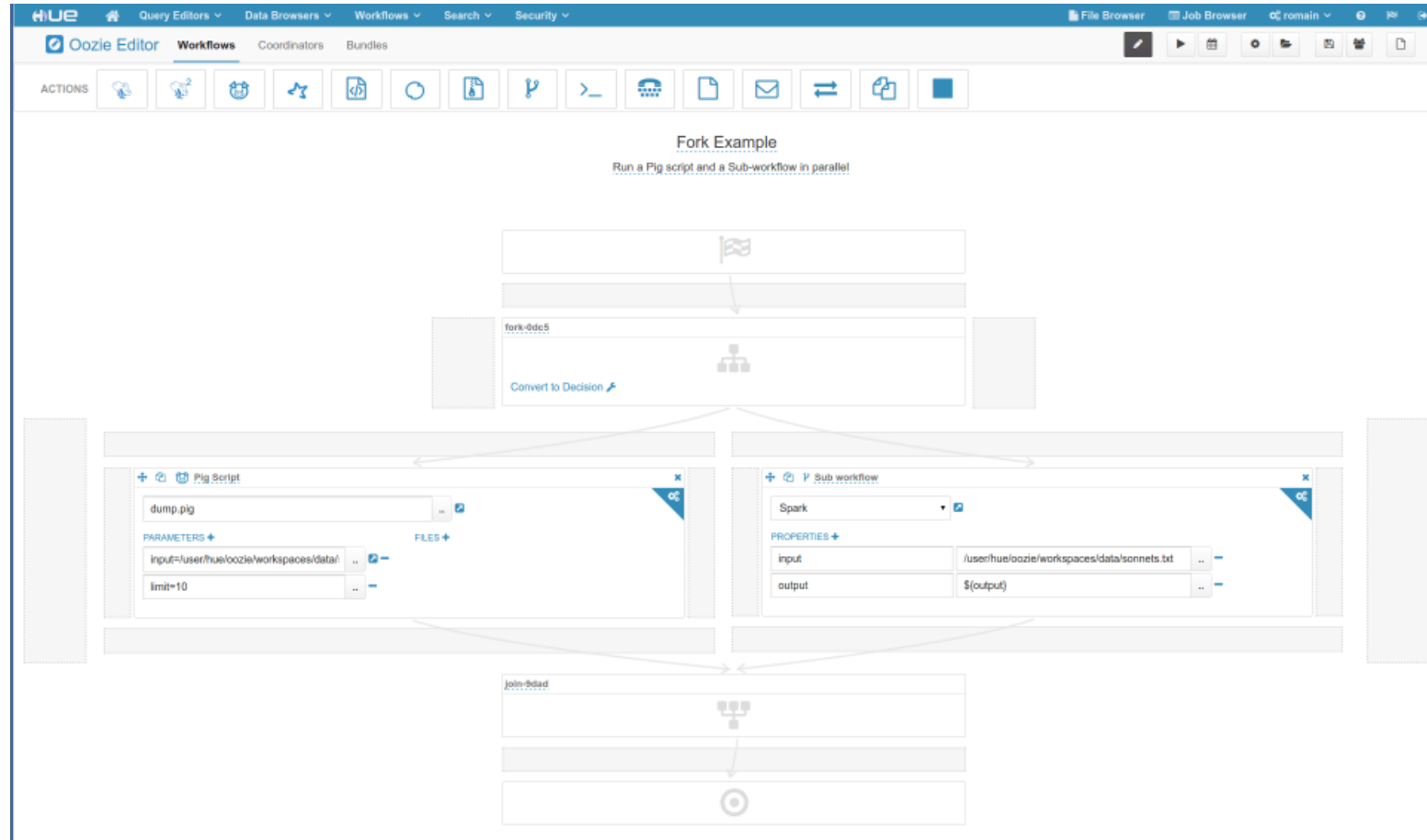
```
1 SampleRecord = LOAD '/user/cloudera/pigdir/'
2 USING PigStorage(',') AS (Year:chararray);
3 GroupByYear = GROUP SampleRecord BY Year;
4 CountByYear = FOREACH GroupByYear
5 GENERATE CONCAT((chararray)$0,CONCAT(':', (chararray)COUNT($1)));
6 STORE CountByYear
7 INTO '/user/cloudera/pigdir/pigoutput' USING PigStorage('t');
```

Below the script editor, a PySpark notebook is open. The notebook has a header 'Sample Notebook' and a description 'Run Spark Python, Scala and SQL snippets with graphs'. The notebook content is organized into sections:

- DATABASE**: A dropdown menu showing 'default'.
- TABLES**: A list of tables including 'apache\_logs', 'assets', 'bike\_rebalancing', 'blog', 'business', 'default\_sample\_07\_table01\_index', 'employees', 'escapeerror', 'many\_columns', and 'nan\_test'.
- Simple Python count**: A code snippet `print 1 + 1 + 1`.
- Use regular Py-spark functions**: A code snippet that reads a file, processes it with `flatMap`, `map`, and `reduceByKey`, and prints the results.
- from random import random**: A code snippet for importing the `random` module.

The bottom of the notebook shows a status bar with '100%' zoom and a '100%' indicator.

# HUE: Schedulers



# HUE: Dashboards

The screenshot displays the HUE (Hadoop User Experience) interface. At the top, there is a navigation bar with options like Query Editors, Data Browsers, Workflows, Search, Security, File Browser, Job Browser, and a user profile for 'romain'. Below this, the 'Hive Editor' section is active, showing a 'Query Editor' with a SQL query:

```
1 SELECT sample_07.description, sample_07.salary
2 FROM
3   sample_07
4 WHERE
5   ( sample_07.salary > 100000)
6 ORDER BY sample_07.salary DESC
7 LIMIT 15
```

Buttons for 'Execute', 'Save', 'Save as...', 'Explain', and 'New query' are visible below the query editor. On the left sidebar, the 'DATABASE' section shows a tree view of tables including 'employees', 'invites', 'sample\_07', and 'sample\_08'. The 'sample\_07' table is selected, showing its columns: 'code (string)', 'description (string)', 'total\_emp (int)', and 'salary (int)'. Below the query editor, the 'Chart' view is active, displaying a bar chart. The chart type is set to 'Bar', the X-axis is 'description', and the Y-axis is 'salary'. The chart shows the salary for 15 different professions, with Anesthesiologists having the highest salary at approximately 192,780.

Profession	Salary (approx.)
Anesthesiologists	192,780
Surgeons	185,000
Orthodontists	180,000
Obstetricians and gynecologists	175,000
Oral and maxillofacial surgeons	170,000
Prosthodontists	165,000
Internists, general	160,000
Physicians and surgeons, all other	155,000
Family and general practitioners	150,000
Chief executives	145,000
Psychiatrists	140,000
Dentists, general	135,000
Pediatricians, general	130,000
Dentists, all other specialists	125,000
Podiatrists	120,000



# Questions?

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