A NEW PLATFORM FOR A NEW ERA

Pivotal

What is Hadoop?

Pivotal

What is Hadoop

- **Hadoop –** "The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models"

- Core Hadoop components: **HDFS**, **Map Reduce (M/R)** and **YARN.**
- **HDFS** is a read-only file system that is massively scalable and reliable. HDFS can store various file types regardless if they are structured or unstructured – SCHEMA ON READ
- **M/R** is a distributed processing framework built for very large batch processing.
- **YARN** is a distributed application framework with job scheduling and resource management components.

***initial ODP charter to standardize these among participants**

Pivotal

What is the Hadoop Ecosystem

- **Hadoop Ecosystem –** Components in and out of the Apache software foundation that either run on top of or have the ability to interact with Hadoop

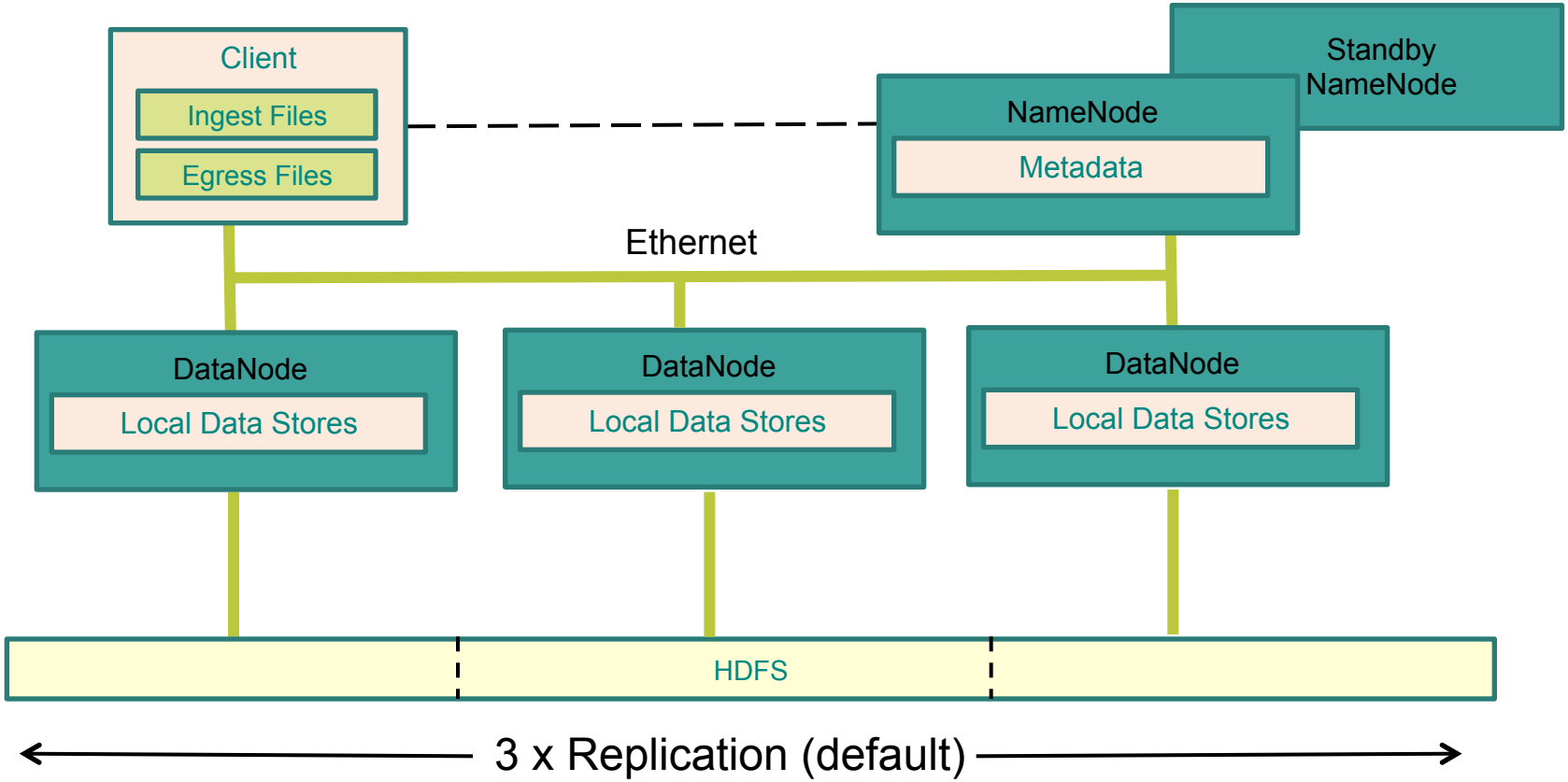
Examples

- **HBase** - A scalable, distributed database that supports structured data storage for large tables
- **Pig** - A high-level data-flow language and execution framework for parallel computation.
- **Spark** - A fast and general compute engine for Hadoop data. Spark supports a wide range of applications, including ETL, machine learning, stream processing, and graph computation
- **Zookeeper** - A high-performance coordination service for distributed applications

Pivotal

Pivotal

HDFS Basic Architecture



How do I access Hadoop (HDFS)?

- **Open a terminal**
- **Execute the following:**
  - hdfs dfs –ls / ***(List what directories are on HDFS)***
  - hdfs dfs –mkdir /user/sample ***(Create a directory for our demo)***
  - hdfs dfs –ls /user/ ***(Verify the directory was successfully created)***
- **Now let's load some data!**
  - cd /home/gpadmin/Labs/PHDHAWQ/ ***(You should see a .csv)***
  - hdfs dfs –put sample.csv /user/sample/
- **Let's check to see what was loaded**
  - hdfs dfs –ls /user/sample
  - hdfs dfs –tail /user/sample/sample.csv

Pivotal

What are the key things I need to know about Hadoop (HDFS)?

- HDFS stores files as blocks and each block is replicated 3 times (1 GB file = 3 GB file on Hadoop)
- The base architecture of Hadoop revolves around the namenode (brain) and the datanodes (workers)
- HDFS is best at handling very big files (1 TB file vs 1000 1 GB files)
- Horizontally scalable…just add datanodes
- Hadoop was originally designed for large batch processing. Without massive scale, quick results are not always attainable.

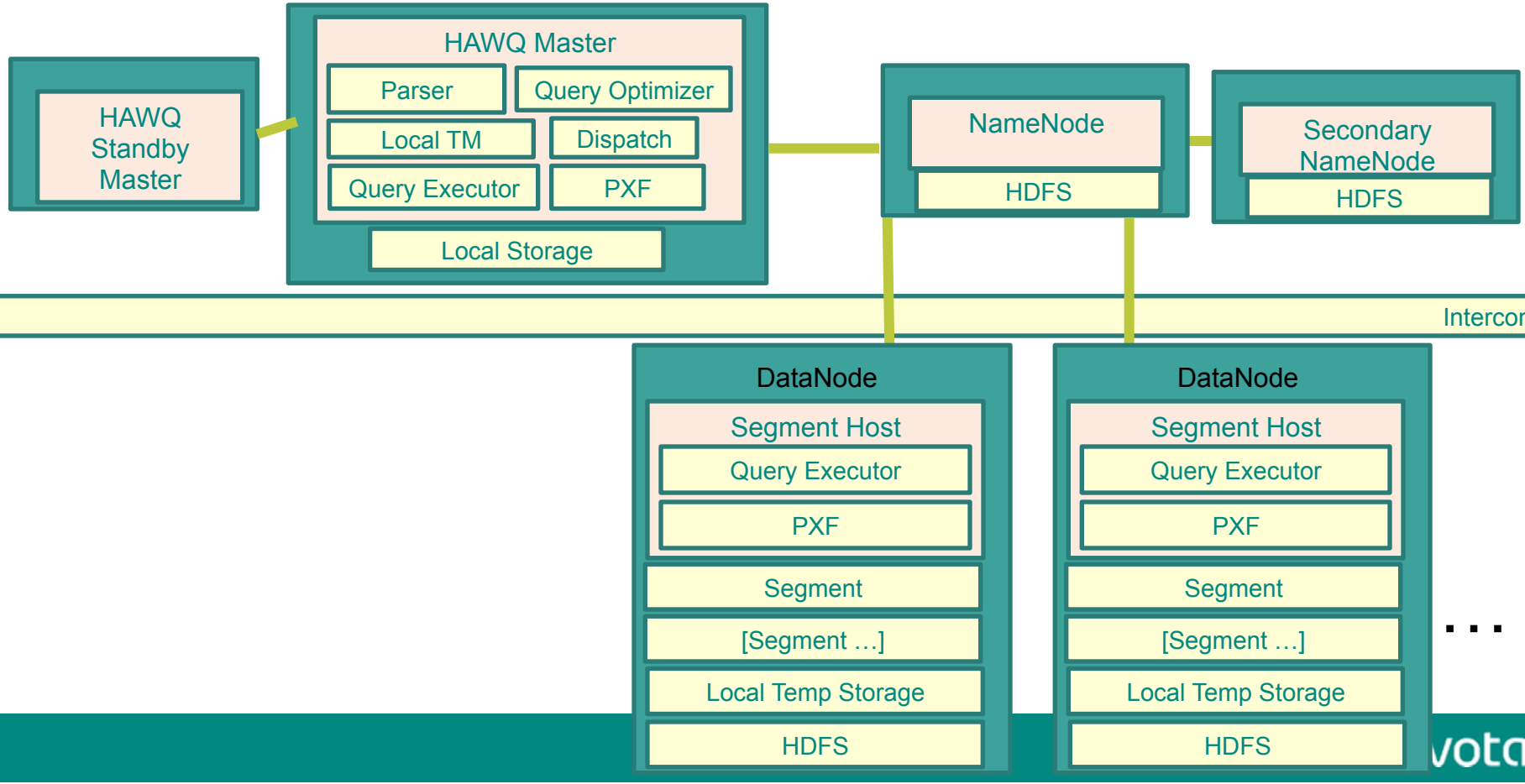
Pivotal

What is HAWQ?

- HAWQ is a Massively Parallel Processing RDBMS that sits on top of HADOOP
- SQL on Hadoop or more importantly ANSI SQL on Hadoop
- What about Hive?
- HAWQ stores all data directly on HDFS
  - Open Terminal
  - Execute the following
    - hdfs dfs -ls –R /hawq_data

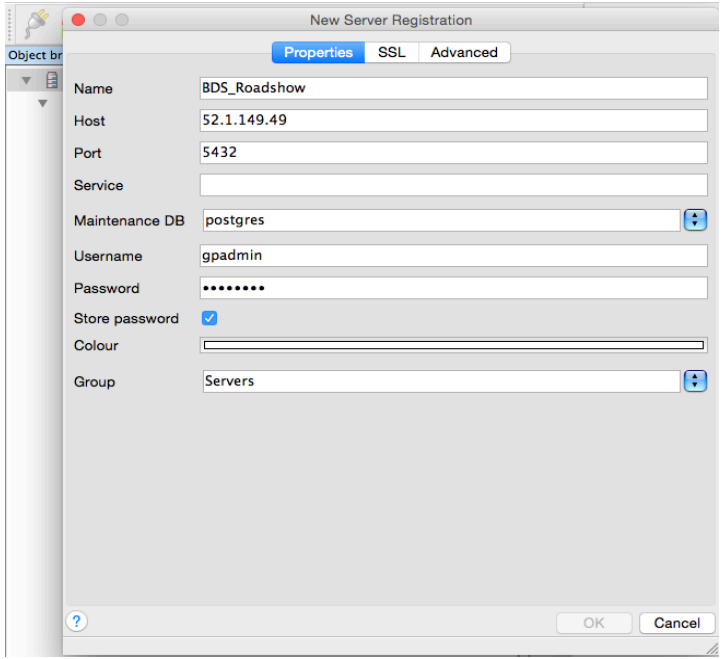
Pivotal

Basic HAWQ Architecture



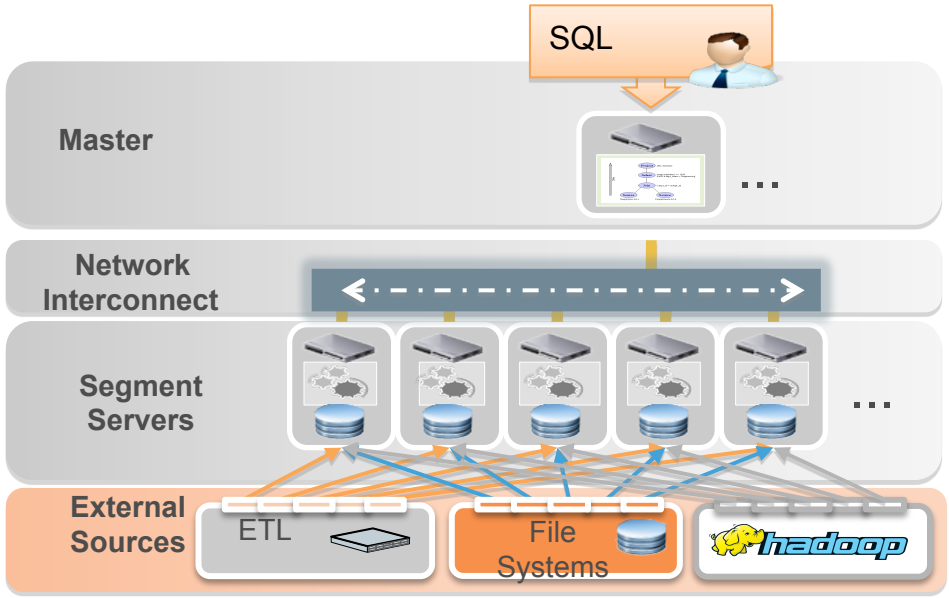
How do I access HAWQ

- **HAWQ can be accessed via PSQL command line client or pgadminIII. We're going to use pgadminIII**
  - Under Applications, click on pgadminIII
  - Click the plug in the upper right hand corner
- **Now that we're connected we're going to**:
  - Create an external table
  - Create a table in HAWQ
  - Load data into HAWQ



What do I need to know about external tables?

- External tables are the most efficient way to load data into HAWQ
- Loading speed is achieved by taking advantage of parallelization. A distributed protocol (PXF or GPFDIST) is used to load the data in parallel.
- External tables are a logical view of the file on HDFS.
- External tables are simple, use SQL and can be used to transform the data in flight

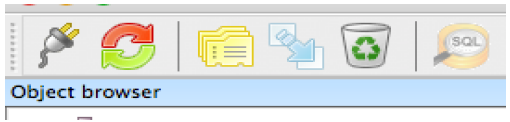


Pivotal

Create an external table

- **First thing we need to do is identify the exact location of the file we wish to load**
  - hdfs dfs -ls /user/sample/ (/user/sample/sample.csv)
  - This will be used by our location
- **In pgadminIII, click on the**

the SQL glass



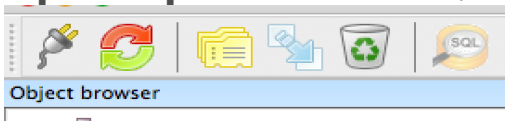
- **Open the following file from the lab**
  - 1_create_ext_table.sql 
  - Click the play button to execute

```
drop external table if exists stage.car_sample_data_ext;
create external table stage.car_sample_data_ext (
 time timestamp,
 vehicleSpeed numeric,
 instantFuelEconomy numeric,
 totalFuelEconomy numeric,
 fuelRate numeric,
 calculatedLoadValue numeric,
 engineCoolantTemperature numeric,
 shortTermFuelTrimBank1 numeric,
 longTermFuelTrimBank1 numeric,
.........
 longitude numeric,
 latitude numeric,
 altitude numeric,
 bearing numeric,
 gpsSpeed numeric,
 gpsAccuracy numeric
 )
LOCATION ('pxf://pivhdsne.localdomain:50070/user/sample/sample.csv?profile=HdfsTextSimple')
FORMAT 'CSV' (HEADER)
LOG ERRORS INTO stage.sample_err SEGMENT REJECT LIMIT 10;
```

Pivotal

Create an HAWQ table

- **We created the external table to read the data from hdfs, now we need to create a place to send the data**
- **Open up another SQL window**



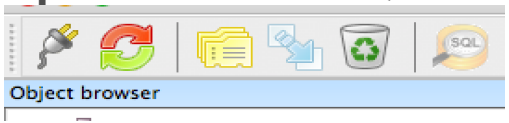
- **Open the following file from your desktop**
  - **2_create_hawq_table.sql**
  - **Click the play button to execute**

```
--2. Create table in HAWQ that will house the data
drop table if exists stage.car_sample_data;
create table stage.car_sample_data (
 time timestamp,
 vehicleSpeed numeric,
 instantFuelEconomy numeric,
 totalFuelEconomy numeric,
 fuelRate numeric,
 ......
 latitude numeric,
 altitude numeric,
 bearing numeric,
 gpsSpeed numeric,
 gpsAccuracy numeric
 )
WITH (appendonly=true, compresstype=quicklz)
DISTRIBUTED RANDOMLY;
```

Pivotal

Load data into HAWQ

- **We created the external table to read the data from hdfs, we created the HAWQ table to load it in to, now we need to load the data.**
- **Open another SQL window**



- **Open the following file from your desktop**
  - **3_load_hawq_table.sql**
  - **Click the play button to execute**

```
--3. Load data into HAWQ table
insert into stage.car_sample_data
select * from stage.car_sample_data_ext;
analyze stage.car_sample_data;
```

Pivotal

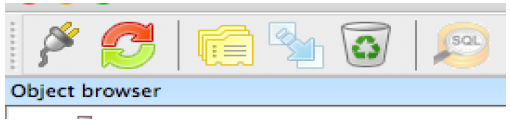
What are the key things I need to know about HAWQ?

- Distribution, Distribution and Distribution
- No Updates and Deletes
- Parallelism is defined by the number of segments

Query data in HAWQ

- **We'll now run a SQL query on the data we have in CSV in HDFS and additionally in HAWQ binary format**

- **Open another SQL window**



- **Open the following file from your desktop**
  - **4_rpm_bands.sql**
  - **Click the play button to execute**

```
--4 Analyze engine rpm in 5% chunks
\timing
SELECT MIN(enginerpm)::int,
       AVG(enginerpm)::int,
       MEDIAN(enginerpm)::int,
       MAX(enginerpm)::int,
       band
FROM (
  SELECT enginerpm,
         NTILE(20) OVER ( ORDER BY enginerpm ) AS band
  FROM stage.car_sample_data_ext ) AS a
GROUP BY band
ORDER BY band;

SELECT MIN(enginerpm)::int,
       AVG(enginerpm)::int,
       MEDIAN(enginerpm)::int,
       MAX(enginerpm)::int,
       band
FROM (
  SELECT enginerpm,
         NTILE(20) OVER ( ORDER BY enginerpm ) AS band
  FROM stage.car_sample_data ) AS a
GROUP BY band
ORDER BY band;
```

Pivotal

Summary

- **What have we done?**
  - We loaded a file onto hdfs
  - We navigated through the hdfs file system
  - We created an external table
  - We created a table in HAWQ
  - We loaded data into HAWQ
  - We queried HAWQ

Pivotal

Data Processing

Spring XD

Spark

Pivotal HD

Advanced Analytics

Pivotal Greenplum Database

Pivotal HAWQ

Apps at Scale

Pivotal GemFire

Redis

RabbitMQ

Pivotal Cloud Foundry

Pivotal Big Data Suite
on Pivotal Cloud Foundry

Pivotal