SUMMIT

PRESENTED BY RED HAT

LEARN. NETWORK. EXPERIENCE OPEN SOURCE.
Large Scale/Big Data Federation & Virtualization: A Case Study

Vamsi Chemitiganti, Chief Solution Architect
Derrick Kittler, Senior Solution Architect
Bill Kemp, Senior Solution Architect
Red Hat
06.29.12
Red Hat Big Data – Week In Retrospective

- Big Data, Volume, Speed & Benefits with Red Hat JBoss Data Grid
- Red Hat's Big Data Strategy Overview & Optimizing Apache Hadoop with Red Hat Enterprise MRG Grid
- JBoss Enterprise Middleware & Big Data
- NoSQL & Big Data at Red Hat
- Large Scale / Big Data Federation & Virtualization: A Case Study
Goals

- “Big” Data by Example
- Significant Learning/s
- Solution Architectures
Background – **Big Data**

- Data growing at 40% per year (McKinsey Global Institute) – a lot of analyst reports exist...
- **Big Data**
  - Data size and performance requirements become significant design and decision factors for implementing a data management and analysis system.
  - In this definition, there’s not an absolute size milestone between “data” and “big data.”
- 4V's
  - How much? formats? speed? change?
Background – Where is it coming from?

- **Volume** - “old” and “new” types of data
  - transaction volumes and other traditional data types

- **Variety** – more types of information to analyze
  - social media, mobile (context-aware)
    - tabular data (databases), hierarchical data, documents, e-mail, metering data, video, still images, audio, stock ticker data, financial transactions, etc...

- **Velocity** – how fast is data produced?
  - how fast must the data be processed?

- **Variability** - data unpredictability, new forms, risk!
  - UPC barcodes, RFID scanners, sensors (HVAC), etc...
Background – **Big Data Tech Landscape**

- NoSQL (Not only SQL)
- Brewers CAP Theorem
  - Cassandra, MongoDB, Neo4J, Hive, Pig, JDG, etc...
- Data Federation and Virtualization
  - JBoss Enterprise Data Services Platform
- MapReduce and batch processing
- And the list goes on ...
Background – Financial Services Industry

- Large commercial bank
  - History of large acquisitions
    - Different sources of data that capture only parts of their overall data lifecycle

- Fast moving business and compliance environment
  - Dodd-Frank and Basel 2 regulations

- Need agility on top of all their data challenges
The Problem – **Case Study Problem Domain**

- Acquisitions and partner integration
  - Many sources of financial data with different origins and formats.
- Primary Business Drivers
  - Business Intelligence (predictive analytics and forecasting)
  - How to harness the volumes of Data; 'Big' Data
  - Compliance and Risk Management
  - Provide exposure to clients via Multiple Channels
  - Large pains around ETL
Reference Architecture

Open Source SOA, Jeff Davis, Manning Press
Reference Architecture

Open Source SOA, Jeff Davis, Manning Press
<table>
<thead>
<tr>
<th>App Name</th>
<th>Format</th>
<th>Size</th>
<th># Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>MERCURY</td>
<td>MF – ASCII Extract</td>
<td>~ 450 GB</td>
<td>1 tbl/day</td>
</tr>
<tr>
<td>ATLAS</td>
<td>M/F ASCII Extract</td>
<td>~ 54 GB</td>
<td>1 tbl/day</td>
</tr>
<tr>
<td>ARES</td>
<td>Oracle DB</td>
<td>~ 350 GB</td>
<td>8-10 tbs</td>
</tr>
<tr>
<td>HERCULES</td>
<td>Oracle DB</td>
<td>~ 200 GB</td>
<td>11-12 tbs</td>
</tr>
<tr>
<td>APOLLO</td>
<td>ASCII File</td>
<td>~ 10 GB</td>
<td>1 tbl</td>
</tr>
<tr>
<td>ZEUS</td>
<td>ASCII File</td>
<td>~ 10 GB</td>
<td>1 tbl</td>
</tr>
<tr>
<td>ATHENA</td>
<td>Oracle DB</td>
<td>~ 20 GB</td>
<td>8-10 tbs</td>
</tr>
<tr>
<td>HADES</td>
<td>Sybase Table</td>
<td>~ 50 GB</td>
<td>8 tbs</td>
</tr>
<tr>
<td>HERA</td>
<td>XML Dump</td>
<td>~ 10GB</td>
<td>9 tbs</td>
</tr>
<tr>
<td>DEMETER</td>
<td>Oracle DB</td>
<td>~ 10 GB</td>
<td>10-12 tbs</td>
</tr>
</tbody>
</table>
Mercury Trades

- Data from Mercury flows through to ZEUS
- If international data – ZEUS pass to International Settlement systems
- Data from MERCURY is also logged to Trademart (ARES) system
- Position information are captured in the GPDW (HERCULES) application

ATLAS Data

- Trades from ATLAS flows through to APOLLO (domestic trade)
- If international data – ATLAS passes to ARES and then to HADES/ATHENA
- Position information are captured in the GPDW (HERCULES) application
The Problem – **views of data...**

- **Trade View**
  - Rationalization, data agility, data source flexibility
  - Easily and rapidly augment models with new sources and/or data attributes

- **Account View**
  - Federate across multiple account sources and create a virtualized canonical mode
  - Data augmentation, federation and data source flexibility
    - Service real time data integration challenges

- **Instrument View**
  - Rationalization, virtualization, data source flexibility
The Problem – data hierarchies

**Trade Source Data systems**

<table>
<thead>
<tr>
<th>Priority</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ARES – Trade DM</td>
</tr>
<tr>
<td>2</td>
<td>HERCULES</td>
</tr>
<tr>
<td>3</td>
<td>APOLLO</td>
</tr>
<tr>
<td>4</td>
<td>ATHENA</td>
</tr>
<tr>
<td>5</td>
<td>HADES</td>
</tr>
</tbody>
</table>

**Instrument Source Data systems**

<table>
<thead>
<tr>
<th>Priority</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SMC Master</td>
</tr>
<tr>
<td>2</td>
<td>ARES – Trade DM</td>
</tr>
<tr>
<td>3</td>
<td>HERCULES</td>
</tr>
<tr>
<td>4</td>
<td>APOLLO</td>
</tr>
<tr>
<td>5</td>
<td>ATHENA</td>
</tr>
<tr>
<td>6</td>
<td>HADES</td>
</tr>
</tbody>
</table>

**Account Source Data systems**

<table>
<thead>
<tr>
<th>Priority</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AMC Master</td>
</tr>
<tr>
<td>2</td>
<td>ARES – Trade DM</td>
</tr>
<tr>
<td>3</td>
<td>HERCULES</td>
</tr>
<tr>
<td>4</td>
<td>APOLLO</td>
</tr>
<tr>
<td>5</td>
<td>ATHENA</td>
</tr>
<tr>
<td>6</td>
<td>HADES</td>
</tr>
</tbody>
</table>
Significant Learnings – not as easy as you think

- Source database performance matters
  - Database tuning is important for push-down model
- Materialization is a read-only solution
  - Write back is disabled!
- Unstructured data is not query-ready
  - Pre-processing, indexing and optimization is required
- Data virtualization provides the ability to discover nuances in the source data and relationships
  - Learn more about your data
Solution

- Speed Layer
- Serving Layer
- Batch Layer
The Solution – **Batch Layer**

- High Latency
  - Lots of data
  - Continual compute
- Pre-compute Views of Data
  - Master data set / all data
  - Constantly growing!
- MapReduce is a Canonical Example
  - Red Hat Storage, Hadoop, Etc...

```java
private static void runBatchLayer() {
    while (true) {
        recomputeBatchViews();
    }
}
```
The Solution – **Serving Layer**

- Loads the batch views
  - Indexes for efficient querying
    - Continual compute
- Pre-compute Views of Data
  - Master data set / all data
  - Constantly growing!
- NoSQL is a Canonical Example
  - MongoDB, Cassandra, Neo4J, Etc...
    - Key-Value, Graph, Document, Column??
The Solution – **Speed Layer**

- Similar to Service Layer but Faster!
  - Doesn't look at all new data at once
  - Updates real-time view as it receives new data
- Incremental Updates vs. Re-computation Updates
  - Produces views only on **RECENT** data vs. entire dataset
  - Random reads/writes
  - Way more complex than batch and serving layer
- Data Federation/Virtualization, Caching, etc..
  - Enterprise Data Services Platform, JBoss Data Grid
Solution Architecture

Red Hat Storage, m:r

NoSQL

Big Data, Nathan Martz, Manning Press
Solution Architecture

- Enterprise Data Services Platform
  - Speed layer; real time data and batch data access
- JBoss Data Grid
  - Speed layer and Service layer; in-memory data
- Cassandra
  - Built for analytics; fast writes, highly consistent
    - Maintains indexes for Speed layer
- MapReduce/Red Hat Storage
  - Continual processing of master data
    - Many jobs / continual processing
Enterprise Data Services Platform

How It Works

Data Services Platform

Common Data Models

Exposed Data Services

SOAP/JMS

Hibernate

JDBC/ODBC

CRM, Employee

Logistics, Supply Chain

Data Sources

Applications
Red Hat Storage – applicability

- Out-of-box compatible with MapReduce apps
- Superior Storage Economics
- NAS, NFS, CIFS, HTTP Access to data
- Unify Data Storage
- Eliminate need for NameNode
JBoss Data Grid – Core Architecture
Improving Integration of Big Data into Enterprise Application Architectures

• Red Hat’s Solution
  • Existing data producers, standards-based interfaces into the ETL pipeline.
Achieving Greater Transparency into Enterprise Data and Big Data Assets

• Red Hat’s Solution

  • A virtualized view of enterprise data, regardless of the source or type. It copes with large amounts of unstructured data via an ETL intake pipeline that extracts and store metadata in a fully customizable manner, increasing overall data visibility.