DBA to Data Scientist with Oracle Big Data Appliance

November 09, 2013
Satyendra Kumar Pasalapudi

Practice Manager – Apps Associates
Co-Founder & Vice President – All India Oracle Users Group
14+ Years of Experience in Oracle Technologies
Exadata Certified Professional

@pasalapudi

Content Courtesy oracle.com, hortonworks,couchbase,apache
Agenda

• What is Big Data
• Big Data Growth
• 4 Phases of Big Data
• NoSQL Databases
• Hadoop Basics
• Big Data Appliance
• Skills Required for DBA Scientist
Big Data Growth

A new style of IT emerging

Every 60 seconds

- 98,000+ tweets
- 695,000 status updates
- 11 million instant messages
- 698,445 Google searches
- 168 million+ emails sent
- 1,820TB of data created
- 217 new mobile web users

Worldwide Data Growth

Source: IDC, BMC. 1EB = 1 Billion GB.
3 Macro Trends Driving Disruption

- Big Data
- Big Users
- SaaS/Cloud Computing
Gen X Stats

2+ Billion

35 Billion Hours

1+ Billion

Global Online Population

Hours Spent Online

Smartphone Users
Big Data – High Data Variety & Velocity

More Flexible Data Model Required

Unstructured and Semi-Structured Data
- Text, Log Files, Click Streams, Blogs, Tweets, Audio, Video, etc.

Structured Data

Trillions of Gigabytes (Zettabytes)
- 2000
- 2006
- 2011
- 0
- 0.50
- 1.00
- 1.50
- 2.00
$30B Database Market Being Disrupted
How Did Big Data Evolve?

• More people interacting with data
• Smartphones
• Internet

• Greater volumes of data being generated (machine-to-machine generation)
• Sensors
• General Packet Radio Services (GPRS)
What Is Big Data?

Big data is defined as voluminous unstructured data from many different sources, such as:

- Social networks
- Banking and financial services
- E-commerce services
- Web-centric services
- Internet search indexes
- Scientific searches
- Document searches
- Medical records
- Weblogs
Big Data

- Extremely large datasets that are hard to deal with using Relational Databases
  - Storage/Cost
  - Search/Performance
  - Analytics and Visualization
- Need for parallel processing on hundreds of machines
  - ETL cannot complete within a reasonable time
  - Beyond 24hrs – never catch up
Characteristics of Big Data

- Volume
- Variety
- Velocity
- Value

- Social Networks
- RSS Feeds
- Micro Blogs
The Four Phases of Data Conversion

1. Acquire
2. Organize
3. Analyze
4. Decide
Operational vs. Analytical Databases

Real-time, Interactive Databases

- NoSQL
- Fast access to data

Analytic Databases

- Get insights from data

Couchbase
- MongoDB
- Cassandra
- Hbase
- Cloudera
- Hortonworks
- Mapr
Growth is the New Reality

Instagram gained nearly 1 million users overnight when they expanded to Android
Draw Something Viral Growth

Draw Something by OMGPOP
Daily Active Users (millions)
How Do You Take This Growth?

Application Scales Out
Just add more commodity web servers

RDBMS Scales Up
Get a bigger, more complex server

RDBMS is good for many thing, but hard to scale
Scaling Out RDBMS

- Run Many SQL Servers
- Data could be sharded
  - Done by the application code
- Caching for faster response time
RDBMS are Not Enough?
NoSQL Technology Scales Out

Scaling out flattens the cost and performance curves
A New Technology

- Building new database to answer the following requirements
  - No schema required before inserting data
  - No schema change required to change data format
  - Auto-sharding without application participation
  - Distributed queries
  - Integrated main memory caching
  - Data synchronization (multi-datacenter)
## Use Cases

| Key Value               | • Session Management  
|                         | • User Profile/Preferences  
<table>
<thead>
<tr>
<th></th>
<th>• Shopping Cart</th>
</tr>
</thead>
</table>
| Document               | • Event Logging  
|                         | • Content Management  
|                         | • Web Analytics  
|                         | • E-Commerce Application  |
| Columns                | • Event Logging  
|                         | • Content Management  
|                         | • Counters  |
| Graph                  | • Connected Data / Social Networks  
|                         | • Routing, Dispatch  
|                         | • Recommendations based on Social Graph  |
Relational vs. Documental Data Model

**JSON or JavaScript Object Notation**, is a text-based open standard designed for human-readable data interchange. It is derived from the JavaScript scripting language for representing simple data structures and associative arrays, called objects. Despite its relationship to JavaScript, it is language-independent, with parsers available for many languages.
Brewer's CAP Theorem

- Requirements to distributed systems
  - **Consistency** – the system is in a consistent state after an operation
    - All clients see the same data
    - Strong consistency (ACID) vs. eventual consistency (BASE)
  - **Availability** – the system is “always on”, no downtime
    - Node failure tolerance – all clients can find some available replica
    - software/hardware upgrade tolerance
  - **Partition tolerance** – the system continues to function even when split into disconnected subsets (by a network disruption)
    - Not only for reads, but writes as well!

- **CAP Theorem** (E. Brewer, N. Lynch)
  - You can satisfy at most 2 out of the 3 requirements
Brewer's CAP Theorem

Visual Guide to NoSQL Systems

Availability:
Each client can always read and write.

Data Models
- Key-Value
- Column-Oriented/Tabular
- Document-Oriented

Relational (comparison)

CA
- RDBMSs (MySQL, Postgres, etc)
- Aster Data
- Greenplum
- Vertica

AP
- Dynamo
- Voldemort
- Tokyo Cabinet
- KAI
- Cassandra
- SimpleDB
- CouchDB
- Riak

Pick Two

Consistency:
All clients always have the same view of the data.

CP
- BigTable
- HBase
- MongoDB
- Terrastore
- Scalaris
- Berkeley DB
- MemcacheDB
- Redis

Partition Tolerance:
The system works well despite physical network partitions.
## NoSQL Technology Spectrum

<table>
<thead>
<tr>
<th>Persistence</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage + Processing</td>
<td></td>
</tr>
<tr>
<td>Hadoop (MapReduce + HDPS)</td>
<td>neo4j</td>
</tr>
<tr>
<td>Amazon S3</td>
<td>FlockDB (Twitter)</td>
</tr>
<tr>
<td>Amazon EMR</td>
<td>InfiniteGraph</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Graph</td>
<td></td>
</tr>
<tr>
<td>BigTable (Google)</td>
<td>HBase (BigTable)</td>
</tr>
<tr>
<td>Cassandra (Dynamo + BigTable)</td>
<td>Riak (Dynamo)</td>
</tr>
<tr>
<td>Hypertable (BigTable)</td>
<td></td>
</tr>
<tr>
<td>SimpleDB (AmazonAWS)</td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td></td>
</tr>
<tr>
<td>MongoDB (~BigTable)</td>
<td>CouchDB</td>
</tr>
<tr>
<td></td>
<td>Voldemort (Dynamo)</td>
</tr>
<tr>
<td>Document</td>
<td></td>
</tr>
<tr>
<td>Dynamo (Amazon)</td>
<td>Voldemort (Dynamo)</td>
</tr>
<tr>
<td></td>
<td>Hazeltcast</td>
</tr>
<tr>
<td>Persistent Key/Value</td>
<td></td>
</tr>
<tr>
<td>Volatile Key/Value</td>
<td>Membase (memcached)</td>
</tr>
<tr>
<td></td>
<td>Tokyo Cabinet</td>
</tr>
</tbody>
</table>
Operational vs. Analytical Databases

Real-time, Interactive Databases

- NoSQL
- Fast access to data

Analytic Databases

- Get insights from data

Couchbase
- Cassandra
- Hbase
- Cloudera
- Hortonworks
- Mapr
Hadoop Design Principles

• System shall manage and heal itself
  – Automatically and transparently route around failure
  – Speculatively execute redundant tasks if certain nodes are detected to be slow
• Performance shall scale linearly
  – Proportional change in capacity with resource change
• Compute should move to data
  – Lower latency, lower bandwidth
• Simple core, modular and extensible
• At Google MapReduce operation are run on a special file system called Google File System (GFS) that is highly optimized for this purpose.

• GFS is not open source.

• Doug Cutting and others at Yahoo! reverse engineered the GFS and called it Hadoop Distributed File System (HDFS).

• The software framework that supports HDFS, MapReduce and other related entities is called the project Hadoop or simply Hadoop.

• Projects Nutch and Lucene were started with “search” as the application in mind;
• Hadoop Distributed file system and mapreduce were found to have applications beyond search.
• HDFS and MapReduce were moved out of Nutch as a sub-project of Lucene and later promoted into a apache project Hadoop
Hadoop History

• Dec 2004 – Google GFS paper published
• July 2005 – Nutch uses MapReduce
• Feb 2006 – Starts as a Lucene subproject
• Apr 2007 – Yahoo! on 1000-node cluster
• Jan 2008 – An Apache Top Level Project
• Jul 2008 – A 4000 node test cluster
• May 2009 – Hadoop sorts Petabyte in 17 hours
What & Where is Hadoop Used For?

Search
• Yahoo, Amazon, Zvents

Log Processing
• Facebook, Yahoo, ContextWeb. Joost, Last.fm

Recommendation Systems
• Facebook

Data Warehouse
• Facebook, AOL

Video and Image Analysis
• New York Times, Eyelike
What & Where is Hadoop Used For?

Hadoop Ecosystem

- **Client Access**
  - Hue
  - Hive (Sql)
  - Pig (Pi/Sql)

- **Data Access**
  - Sqoop
  - Flume

- **Data Mining**
  - Mahout

- **Orchestration**
  - Oozie

**Networking**

- MapReduce (Job Scheduling/Execution System)
- HBase (key-value store)
- HDFS (Hadoop Distributed File System)

**Java Virtual Machine**

**OS**
- Redhat
- Suse
- Ubuntu
- Windows

**Commodity Hardware**

**Chukwa (Monitoring)**

**ZooKeeper (Coordination)**

© Copyright 2013. Apps Associates LLC.
HBase is an open source, non-relational, distributed database modeled after Google's BigTable and is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (Hadoop Distributed File system), providing BigTable-like capabilities for Hadoop. That is, it provides a fault-tolerant way of storing large quantities of sparse data.

HBase features compression, in-memory operation, and Bloom filters on a per-column basis as outlined in the original BigTable paper. Tables in HBase can serve as the input and output for MapReduce jobs run in Hadoop, and may be accessed through the Java API but also through REST, Avro or Thrift gateway APIs.
• HBase is not a direct replacement for a classic SQL database, although recently its performance has improved, and it is now serving several data-driven websites including Facebook's Messaging Platform.

• “Project's goal is the hosting of very large tables - billions of rows X millions of columns - atop clusters of commodity hardware”

• Column-oriented and Random access, real time read/write

• “Random access performance on par with open source relational databases such as MySQL”
Pig

- Compiled into a series of MapReduce jobs
  - Easier to program
  - Optimization opportunities
- grunt> A = LOAD 'student' USING PigStorage() AS (name:chararray, age:int, gpa:float);
- grunt> B = FOREACH A GENERATE name;
Hive

Managing and querying structured data

- MapReduce for execution
- SQL like syntax
- Extensible with types, functions, scripts
- Metadata stored in a RDBMS (MySQL)
- Joins, Group By, Nesting
- Optimizer for number of MapReduce required

```sql
hive> SELECT a.foo FROM invites a WHERE a.ds='<DATE>';
```
Sqoop

- Data import/export
- Sqoop is a tool designed to help users of large data import existing relational databases into their Hadoop clusters
- Automatic data import
- Easy import data from many databases to Hadoop
- Generates code for use in MapReduce applications

- It supports incremental loads of a single table or a free form SQL query as well as saved jobs which can be run multiple times to import updates made to a database since the last import
- Imports can also be used to populate tables in Hive or HBase
- Exports can be used to put data from Hadoop into a relational database
Flume

- Apache Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.
- Simple and flexible architecture based on streaming data flows.
- Robust fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms.
- The system is centrally managed and allows for intelligent dynamic management.
HDFS Architecture

- Master Node
  - Job Tracker
  - Data Node#1
    - Task Tracker
  - Data Node #N
    - Task Tracker
- Edge Node
  - Zookeeper
  - Hive
  - Pig
  - Oozie
  - Avro
  - HDFS Client
- HDFS
  - Name Node
  - Data Node
  - Data Node
  - Data Node

MapReduce

© Copyright 2013. Apps Associates LLC.
Namenode and Datanodes

- Master/slave architecture
- HDFS cluster consists of a single **Namenode**, a master server that manages the file system namespace and regulates access to files by clients
- There are a number of **DataNodes** usually one per node in a cluster
- The DataNodes manage storage attached to the nodes that they run on
- HDFS exposes a file system namespace and allows user data to be stored in files
- A file is split into one or more blocks and set of blocks are stored in DataNodes
- DataNodes: serves read, write requests, performs block creation, deletion, and replication upon instruction from Namenode
HDFS Architecture

- **Namenode**
  - Metadata ops
  - Metadata(Name, replicas..) (/home/foo/data,6...)

- **Datanodes**
  - Block ops
  - replication

- **Blocks**

  - Write
  - Read

- **Rack1**
- **Rack2**

**Client**

- **Metadata ops**
- **Block ops**
Architecture Overview
HDFS Distributions
• Oracle Big Data Appliance is an engineered system containing both hardware and software components. Oracle Big Data Appliance delivers:
  – A complete and optimized solution for big data
  – Single-vendor support for both hardware and software
  – An easy-to-deploy solution
  – Tight integration with Oracle Database
Hadoop 2.0

Hadoop 1.0

MapReduce
(cluster resource management & data processing)

HDFS
(redundant, reliable storage)

Hadoop 2.0

MapReduce
(data processing)

Others
(data processing)

YARN
(cluster resource management)

HDFS
(redundant, reliable storage)
Oracle Big Data Appliance: Where It Stands?

Data Variety
- Unstructured
- Schema-less

Big Data Appliance

Acquire
Organize
Analyze

Information
Oracle Big Data: Software Components

Oracle NoSQL Database

Oracle Big Data Connectors

Open Source R Distribution

Cloudera Manager & Cloudera’s Distribution Including Apache Hadoop

Oracle Linux 5.6 and Java Hotspot VM

Oracle Big Data Appliance
Oracle Big Data with Oracle Exadata

Source Stream

Oracle Big Data Appliance

Oracle Exadata

Oracle Exalytics

Acquire | Organize | Analyze & Decide

InfiniBand
Mapping the Phases with Software

Acquire Phase
– Hadoop Distributed File System
– Oracle NoSQL Database

Organize Phase
– Hadoop Software Framework
– Oracle Data Integrator

Analyze Phase
– R Statistical Programming Environment
– Oracle Data Warehouse
What Is a Key-Value Store?

- A KV Store is essentially a two-column table consisting of a key and a value associated with the key.
- The key acts as the index, and the value can be referenced as a look up.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>010101010</td>
<td></td>
</tr>
<tr>
<td>010101011</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
What Is Oracle Direct Connector for HDFS?

Oracle Direct Connector for HDFS (ODCH) is a connector which facilitates read access from HDFS to Oracle Database using external tables.

- It uses the ORACLE_LOADER access driver
- It enables you to:
  - Access big data without loading the data
  - Access the data stored in HDFS files
  - Access CSV (comma-separated values) files and Data Pump files generated by Oracle Loader for Hadoop
  - Load data extracted and transformed by Oracle Data Integrator
Analyze Phase

Analyze

Database + Oracle R Enterprise

Statistical Functions
Data Mining Algorithms
Query Capabilities
R is an open source statistical programming language and environment, which provides:

- An easy-to-use language
- A powerful graphical environment for visualization
- Several out-of-the-box statistical techniques
- R packages
- Several GUI front ends for analyzing data interactively

It was started in 1994 as an alternative to SAS, SPSS, and other statistical environments.

R’s widespread use, breadth of functionality, and quality of implementation have enabled it to establish itself as a new statistical software standard.
Oracle Big Data: Software Components

- **Oracle NoSQL Database**
- **Oracle Big Data Connectors**
- **Open Source R Distribution**
- **Cloudera Manager & Cloudera’s Distribution Including Apache Hadoop**
- **Oracle Linux 5.6 and Java Hotspot VM**

Oracle Big Data Appliance
Data Science

What is data science?

Data science can be broken down into four essential parts.

Mining data

Collecting and formatting the information

Statistics

Information analysis

Interpret

Leverage

Source :http://wikibon.org/blog/role-of-the-data-scientist/
Scouring
Their eyes search for information on the web
- Vectorized operations
- Algorithmic strategizing
- APIs

Defining a data scientist
A good data scientist understands:

Organization
Data Science: More Than Data Mining
Their voice asks questions about what they hope to accomplish at the end of the project, setting information goals.

Extraction
Takes information they want and organizing it using formulas. They organize the information in order to form educated, insightful conclusions using statistical and these mathematical methods:
- Factor Analysis
- Regression Analysis
- Correlation
- Time Series Analysis

Expansion & Application
The appropriate data flows out of the source in the form and other statistics.

Source: http://wikibon.org/blog/role-of-the-data-scientist/
Required skills for a data scientist

A successful data scientist must have a combination of skills that opens up possibilities both for that individual and their team. Visualization processes are often disjointed since each person is typically assigned to a specific part of the project. The designer depends on the information architect. The information architect depends on stats from the statistician, and so on. A true data scientist should be skilled in multiple areas.

Hacking and Computer Science
Expertise in Mathematics, Statistics, Data Mining
Creativity & Insight

- Knowing how to take advantage of computers and the internet to create data-mining formulas
- Pulling important statistics and coherently organizing them using mathematic prowess and computer formulas
- Knowing what statistics are important and how to leverage them

Hadoop
HDFS
Map Reduce
NoSQL Database
Hive
Pig

OR

All the above with Big Data Appliance
Oracle Big Data Solution

Stream

Oracle Event Processing
Apache Flume
Oracle GoldenGate

Decide

Oracle Real-Time Decisions
Endeca Information Discovery
Oracle BI Foundation Suite

Acquire – Organize – Analyze

Cloudera Hadoop
Oracle NoSQL Database
Oracle R Distribution

Oracle Big Data Connectors
Oracle Advanced Analytics
Oracle Spatial & Graph

Oracle Database
Oracle Data Integrator
Intelligence By Variety

Using big data, organizations can generate actionable insights that enable them to drive their business forward. Rapid integration of the ever-expanding pool of data sources and types is opening a whole new world of possibilities.

KEY

<table>
<thead>
<tr>
<th>SOME APIs</th>
<th>INTERNAL</th>
<th>EXTERNAL</th>
<th>BOTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO APIs</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TERMINOLOGY

SOME APIs
- Data that has a standard Web service
- Data that resides behind an organization’s firewall

NO APIs
- Data that has no standard Web service and requires alternative methods of integration

INTERNAL
- Data that resides outside of an organization’s firewall

EXTERNAL
- Data that does not have a pre-defined data model or is not organized in a pre-defined manner

UNSTRUCTURED
- Data that resides in a fixed field within a record or file

STRUCTURED
- The rate at which data is generated and changed

VELOCITY
- The number of different data sources and types

VARIETY
- The average quantity of data units per category

VOLUME

apps associates extreme expertise

© Copyright 2013. Apps Associates LLC.
Thank You!