The Evolving Apache Hadoop Eco-System –
What it means for Big Data Analytics and Storage

Sanjay Radia
Architect/Founder, Hortonworks Inc.
Outline

• Hadoop and Big Data Analytics
  - Data platform drivers and How Hadoop Changes the Game
  - Data Refinery
  - Hadoop in Enterprise Data Architecture

• Hadoop (HDFS) and Storage
  - RAID disk or not?
  - HDFS’s Generic storage layer
  - Archival
  - Upcoming
Next Generation Data Platform Drivers

**Business Drivers**
- Need for new business models & faster growth (20%+)
- Find insights for competitive advantage & optimal returns

**Technical Drivers**
- Data continues to grow exponentially
- Data is increasingly everywhere and in many formats
- Legacy solutions unfit for new requirements growth

**Financial Drivers**
- Cost of data systems, as % of IT spend, continues to grow
- Cost advantages of commodity hardware & open source
Big Data Changes the Game

Transactions + Interactions + Observations = BIG DATA
How Hadoop Changes the Game

• **Cost**
  - Commodity servers, JBod disks
  - Horizontal scaling – new servers as needed
    - Ride the technology curve

• **Scale from small to very large**
  - Size of data or size of computation

• **The Data-Refinery**
  - Traditional Datamarts optimize for time and space
  - Don’t need to pre-select the subset of “schema”
    - Can experiment to gain insights into your whole data
    - You insights can change over time
  - Don’t need to throw your old data away

• **Open and Growing Eco-system**
  - You aren’t locked into a vendor
  - Ever growing choice of tools
# Vertical Specific Uses of Hadoop

<table>
<thead>
<tr>
<th>Vertical</th>
<th>Analytical Use Cases</th>
<th>Operational Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail &amp; Web</td>
<td>• Social Network Analysis&lt;br&gt;• Log Analysis/Site Optimization</td>
<td>• Session &amp; Content Optimization&lt;br&gt;• Dynamic Pricing</td>
</tr>
<tr>
<td>Retail</td>
<td>• Brand and Sentiment Analysis&lt;br&gt;• Loyalty Program Optimization</td>
<td>• Dynamic Pricing/Targeted Offers</td>
</tr>
<tr>
<td>Intelligence</td>
<td>• Person of Interest Discovery&lt;br&gt;• Threat Identification</td>
<td>• Cross Jurisdiction Queries</td>
</tr>
<tr>
<td>Finance</td>
<td>• Risk Modeling &amp; Fraud Identification&lt;br&gt;• Trade Performance Analytics</td>
<td>• Surveillance and Fraud Detection&lt;br&gt;• Customer Risk Analysis</td>
</tr>
<tr>
<td>Energy</td>
<td>• Smart Grid: Production Optimization&lt;br&gt;• Gris Failure Prevention</td>
<td>• Smart Meters&lt;br&gt;• Individual Power Grid</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>• Customer Churn Analysis&lt;br&gt;• Supply Chain Optimization</td>
<td>• Dynamic Delivery&lt;br&gt;• Replacement parts</td>
</tr>
<tr>
<td>Healthcare &amp; Payer</td>
<td>• Electronic Medical Records (EMPI)&lt;br&gt;• Clinical Trials Analysis</td>
<td>• Insurance Premium Determination</td>
</tr>
</tbody>
</table>
Transaction + Interactions + Observations

Big Data Refinery

1. Classic ETL processing
2. Store, aggregate, and transform multi-structured data to unlock value
3. Share refined data and runtime models
4. Retain runtime models and historical data for ongoing refinement & analysis
5. Retain historical data to unlock additional value

Business Transactions & Interactions

Business Intelligence & Analytics

Audio, Video, Images
Docs, Text, XML
Web Logs, Clicks
Social, Graph, Feeds
Sensors, Devices, RFID
Spatial, GPS
Events, Other

Web, Mobile, CRM, ERP, SCM, ...
Dashboards, Reports, Visualization, ...
Next-Generation Big Data Architecture

Big Data Refinery

Web, Mobile, CRM, ERP, SCM, ...

Business
Transactions & Interactions

SQL
NoSQL
NewSQL

Business
Intelligence & Analytics

Dashboards, Reports, Visualization, ...

Audio, Video, Images
Docs, Text, XML
Web Logs, Clicks
Social, Graph, Feeds
Sensors, Devices, RFID
Spatial, GPS
Events, Other

EDW
MPP
NewSQL
Hadoop in Enterprise Data Architecture

**Existing Business Infrastructure**
- IDE & Dev Tools
- ODS & Datamarts
- Applications & Spreadsheets
- Visualization & Intelligence

**Web**
- Web Applications
- Low Latency/NoSQL

**New Tech**
- Datameer
- Tableau
- Karmasphere
- Splunk

**Data Sources**
(transactions, observations, interactions)
- CRM
- ERP
- financials
- Social Media
- Exhaust Data
- logs
- files

**Operations**
- Custom
- Existing
Metadata Services

Apache HCatalog provides flexible metadata services across tools and external access

- **Consistency** of metadata and data models across tools (MapReduce, Pig, HBase and Hive)
- **Accessibility**: share data as tables in and out of HDFS
- **Availability**: enables flexible, thin-client access via REST API
Outline

• Hadoop and Big Data Analytics
  - Data platform drivers and How Hadoop Changes the Game
  - Data Refinery
  - Hadoop in Enterprise Data Architecture

• Hadoop (HDFS) and Storage
  - RAID disk or not?
  - HDFS’s Generic storage layer
  - Archival
  - Upcoming
HDFS: Scalable, Reliable, Manageable

Scale IO, Storage, CPU
- Add commodity servers & JBODs
- 6K nodes in cluster, 120PB

- Fault Tolerant & Easy management
  - Built in redundancy
  - Tolerate disk and node failures
  - Automatically manage addition/removal of nodes
  - One operator per 3K node!!

- Storage server used for computation
  - Move computation to data

- Not a SAN
  - But high-bandwidth network access to data via Ethernet

- Scalable file system
  - Read, Write, rename, append
    No random writes

Simplicity of design
why a small team could build such a large system in the first place
HDFS Architecture: Two Key Layers

• **Namespace layer**
  › Multiple independent namespace
    • Can be mounted using client side mount tables
  › Consists of dirs, files and blocks
  › Operations: create, delete, modify and list dir/files

• **Block Storage layer – Generic Block service**
  › DataNode cluster membership
  › Block Pool – set of blocks for a Namespace volume
    • Storage shared across all volumes – no partitioning
    • Namespace Volume = Namespace + Block Pool
  › Block operations
    • Create/delete/modify/getBlockLocation operations
    • Read and write access to blocks
  › Detect and Tolerate faults
    • Monitor node/disk health, periodic block verification
    • Recover from failures – replication and placement
File systems Background (3):
Leading to Google FS and HDFS

> **Separation of metadata from data - 1978, 1980**
  - “Separating Data from Function in a Distributed File System” (1978)
    - by J E Israel, J G Mitchell, H E Sturgis
  - “A universal file server” (1980) by A D Birrell, R M Needham

> **Horizontal scaling of storage nodes and io bandwidth (1999-2003)**
  - Several startups building scalable NFS – late 1990s
  - (Hadoop’s HDFS, pNFS)

> **Commodity HW with JBODs, Replication, Non-posix semantics**
  - Google FS (2003) - *Not using RAID a very important decision*

> **Computation close to the data**
  - Parallel DBs
  - Google FS/MapReduce
Significance of not using Disk RAID

• **Key new idea was to not use RAID on the local disk and instead replicate the data**
  - Uniformly Deal with device failure, media failures, node failures etc.
  - Nodes and clusters continue run, with slightly lower capacity
    - Nodes and disks can be fixed when convenient
  - Recover from failures in parallel
    - Raid5 disk take over half a day to recover from a 1TB failure
    - HDFS recovers 12TB in minutes – recover is in parallel
      - Faster for larger clusters
  - Operational advantage: system recovers automatically from node and disk failures very rapidly
    - **1 operator for 4K servers at mature Hadoop installations**
  - Yes there is the storage overhead
  - File-RAID is available (1.4 -1.6 efficiency)
    - Practical to use for portion of data otherwise node recover takes long
HDFS’ Generic Storage Service Opportunities for Innovation

- **Federation - Distributed (Partitioned) Namespace**
  - Simple and Robust due to independent masters
  - Scalability, Isolation, Availability

- **New Services – Independent Block Pools**
  - New FS - Partial namespace in memory
  - MR Tmp storage, HBase directly on block storage
  - Shadow file system – caches HDFS, NFS, S3

- **Future: move Block Management in DataNodes**
  - Simplifies namespace/application implementation
  - Distributed namenode becomes significantly simple
Archival data – where should it sit?

• **Hadoop encourages old data for future analysis**
  - Should it sit in a separate cluster with lower computing power?
  - Spindles are one of the bottlenecks in Big data system.
    - Can’t waste precious spindles to another cluster where they are underutilized
  - Better to keep archival data in the main cluster where the spindles can be actively used
  - As data grows over time, a Big data cluster may have smaller percentage of hot data
    - Should we arrange data on the disk platter based on access patterns?

• **Challenge – growing data**
  - Do tapes play a role?
## HDFS in Hadoop 1 and Hadoop 2

### Hadoop 1 (GA)
- Security
- Append/Fsync (HBase)
- WebHDFS + Spnego
- Write pipeline improvements
- Local write optimization
- Performance improvements
- Disk-fail-in-place
- HA Namenode
- Full Stack HA

### Hadoop 2 (alpha)
- New Append
- Federation
- Wire compatibility
- Edit logs rewrite
- Faster startup
- HA NameNode (hot)
- Full Stack HA – *in progress*
Hadoop Full Stack HA Architecture

Slave Nodes of Hadoop Cluster

Apps Running Outside

Apps pause/retry

Failover

JT into Safemode

N+K failover

HA Cluster for Master Daemons
Upcoming

- **Continue improvements**
  - Performance
  - Monitoring and Management
  - Rolling upgrades
  - Disaster Recovery

- **Snapshots (prototype already published)**

- **Support for heterogeneous storage on DataNodes**
  - Will allow Flash disks to be used for specific use cases

- **Block grouping - allow large number of smaller blocks**

- **Working-set of namespace in memory – large # of files**

- **Other protocols – NFS**

- …
Which Apache Hadoop Distro?

• Stable
• Reliable
• Well supported
• But do not want to lock into a vendor
Hortonworks Data Platform

Delivers enterprise grade functionality on a proven Apache Hadoop distribution to ease management, simplify use and ease integration into the enterprise

- **Simplify deployment** to get started quickly and easily
- **Monitor, manage any size cluster** with **familiar console** and tools
- Only platform to include **data integration services** to interact with any data source
- **Metadata services** opens the platform for integration with existing applications
- Dependable **high availability** architecture

The only 100% open source data platform for Apache Hadoop
Hortonworks Data Platform

Hadoop 1.0.x = stable “kernel”

Integrates and tests all necessary Apache component projects

Most stable and compatible versions of all components are chosen

Closely aligned with each Apache component project code line with closedown process that provides necessary buffer

QE/Beta process that produced every single stable Apache Hadoop release

- Apache Hadoop 2 is being QE’ed and alpha/beta tested through the same process by Hortonworks

Tested, Hardened & Proven Distribution Reduces Risk
Summary

• Hadoop changes Big Data game in fundamental ways
  - Cost, storage and compute capacity
  - Horizontally scale from small to very very large
  - Data Refinery – keep all your data, new business insights
  - Hooks to integrate with existing enterprise data architecture
  - Open, growing eco-system – no lock in

• Hortonworks and HDP Distribution
  - The team that originally created Apache Hadoop at Yahoo
  - The team that is driving key developments in Apache Hadoop
  - QE’ed and alpha/beta tested every stable Hadoop release
    including Hadoop 1 and Hadoop 2 (alpha)
  - HDP is fully open source to apache – nothing held back
    - close/identical to Apache releases
Thank You!

Questions & Answers