From Batch to Real-time Big Data Analytics

Wenming Ye
Sr. Research Program Manager
Microsoft Research Connections

Agenda

What is Big Data?

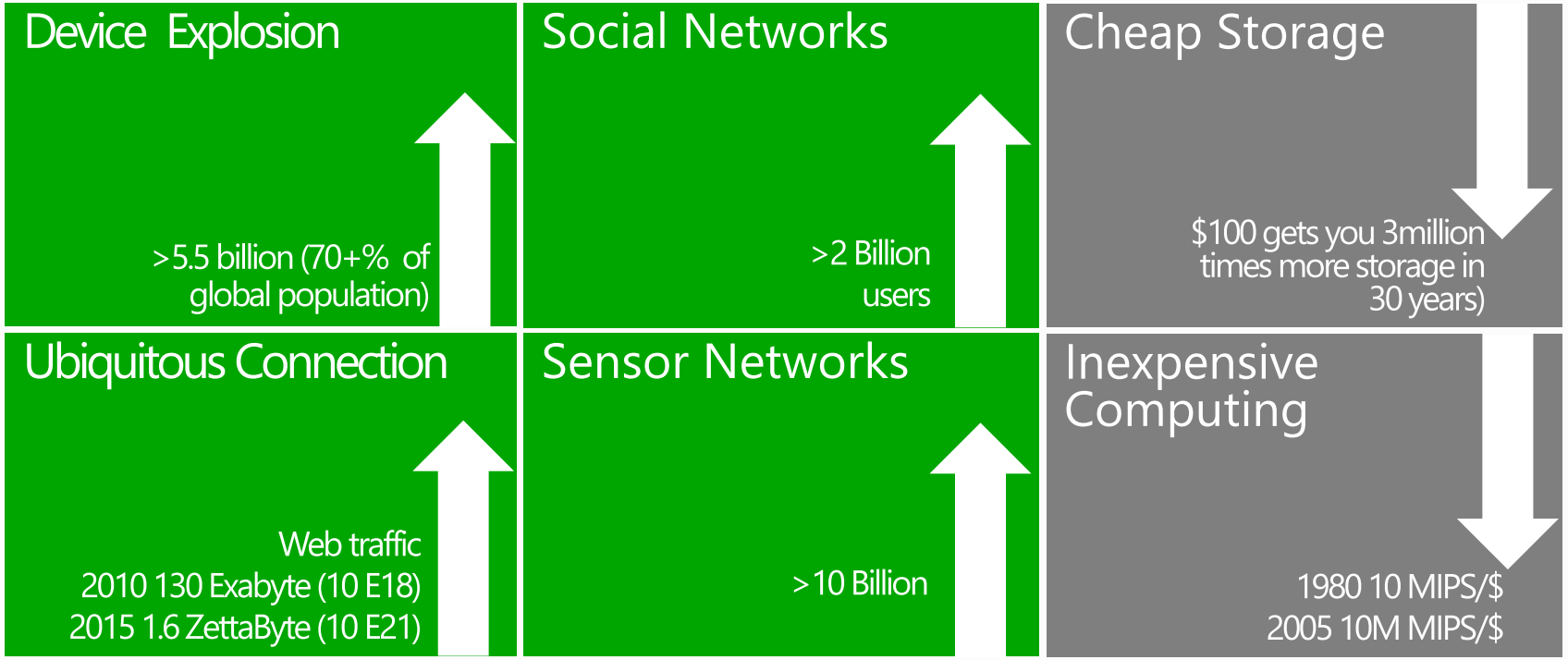
HDInsight: Windows Azure + Hadoop

Programming Hadoop

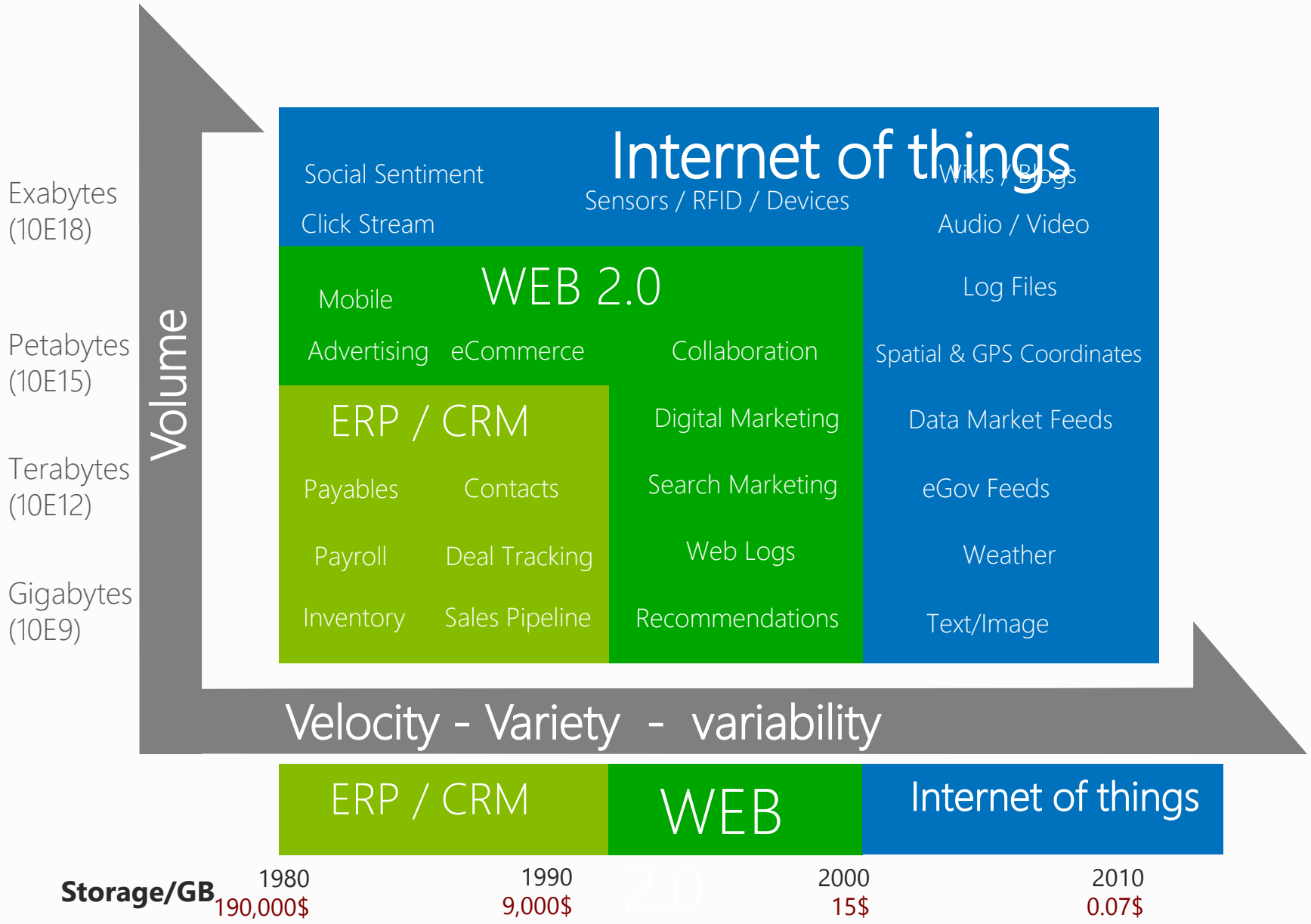
Realtime and Interactive Big Data Tools

Understanding Big Data

KEY TRENDS


Device Explosion
>5.5 billion (70+% of global population)
Social Networks
>2 Billion users
Cheap Storage
$100 gets you 3million times more storage in 30 years)
Ubiquitous Connection
Web traffic
2010 130 Exabyte (10 E18)
2015 1.6 ZettaByte (10 E21)
Sensor Networks
>10 Billion
Inexpensive Computing
1980 10 MIPS/$
2005 10M MIPS/$

What is Big Data?



Big Data, BIG OPPORTUNITY

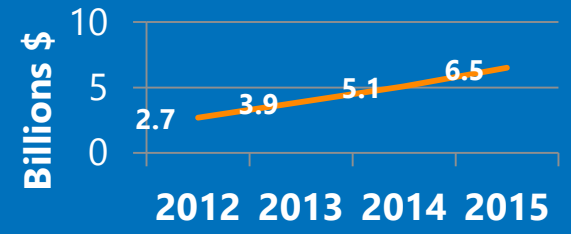
Big Data is a top priority for institutions

49% CEOs and CIOs are planning big data projects

Software Growth

34% compound annual growth rate[2]

Services Growth

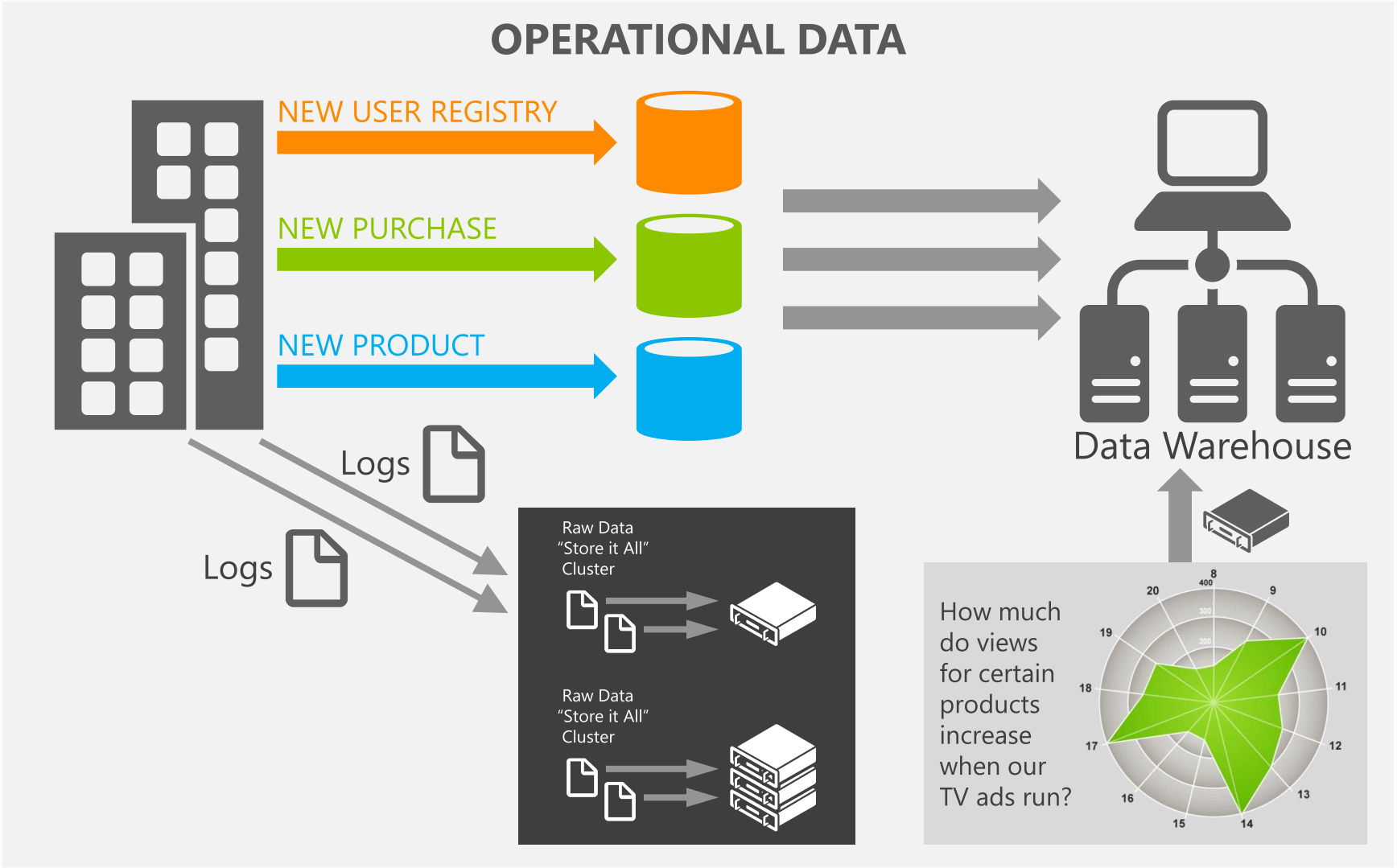


39% compound annual growth rate[2]

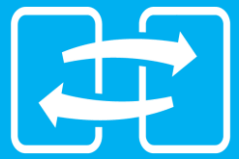
1. McKinsey&Company, McKinsey Global Survey Results, Minding Your Digital Business, 2012
2. IDC Market Analysis, Worldwide Big Data Technology and Services 2012–2015 Forecast , 2012

Big Data Scenarios

New workflow in Data Warehousing

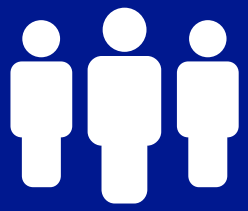


Devices: Internet and Internet of things

Internet of things	 Invisible devices	Trillions of computer-enabled devices which are part of the IoT Trillions of networked nodes	 100kBit/sec Low bandwidth last-mile connection		
 Mostly addressed by local schemes	 Machine-centric	 Sensing-focus	 Laptops / tablets / smartphones	6+billion people 1.5 billion use net US: 4.3 devices per adult Billions of networked devices	 Cable: 10Mbs+ Fiber: 50-100Mbs High-bandwidth access
	 Global addressing	 User-centric	 Communication-focus	Internet	

Collective Intelligence and Predictive analysis

What's the social sentiment of my product?

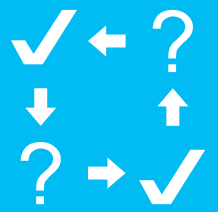


Social Analytics

Live Data Feed, Search

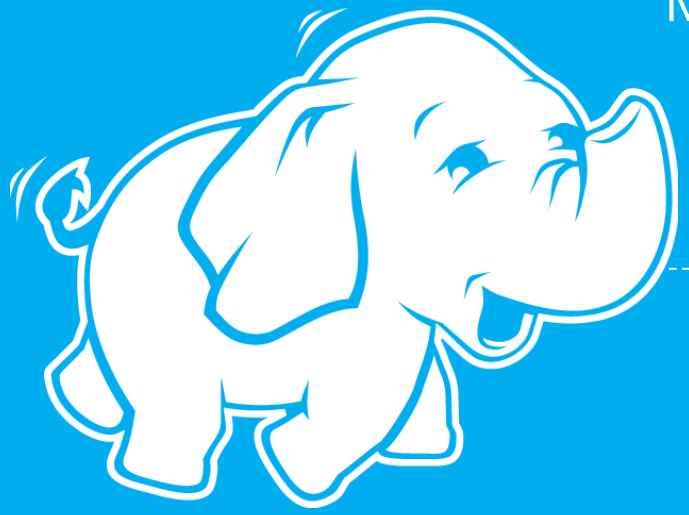
How do I optimize my services based on patterns of weather, traffic. How do I build a recommendation engine?

How do I better predict future outcomes?



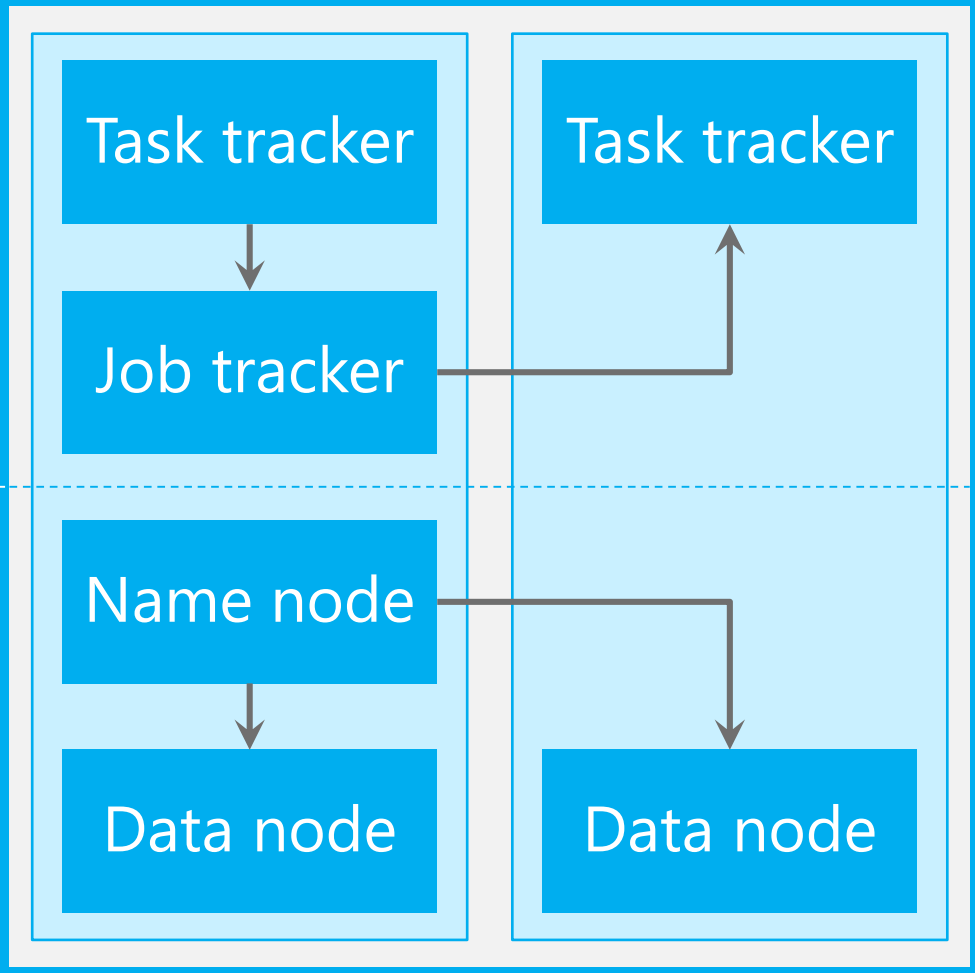
Advanced Analytics

Hadoop Distributed Architecture



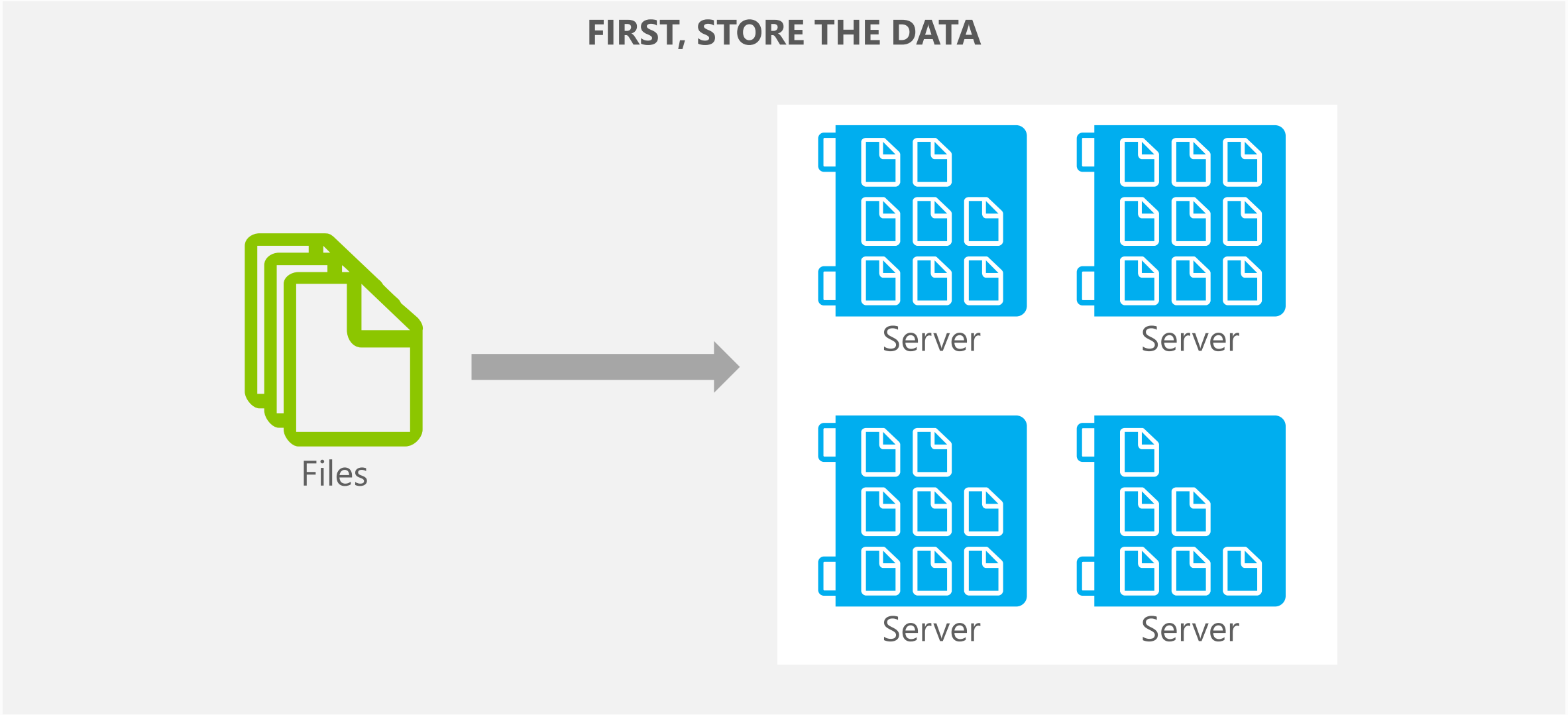
MapReduce Layer

HDFS Layer



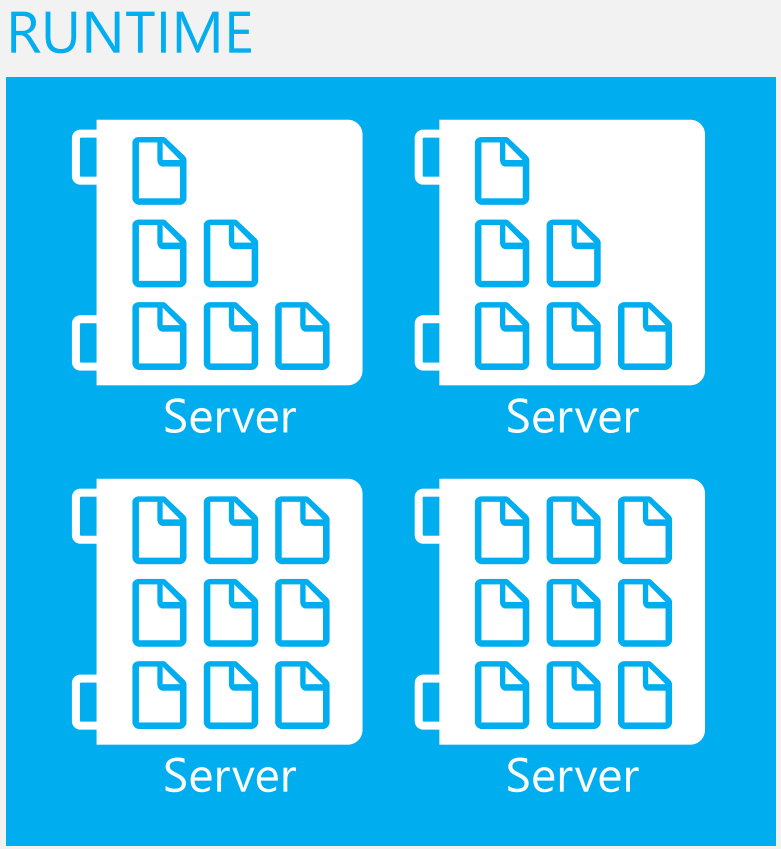
Reference: http://en.wikipedia.org/wiki/File:Hadoop_1.png

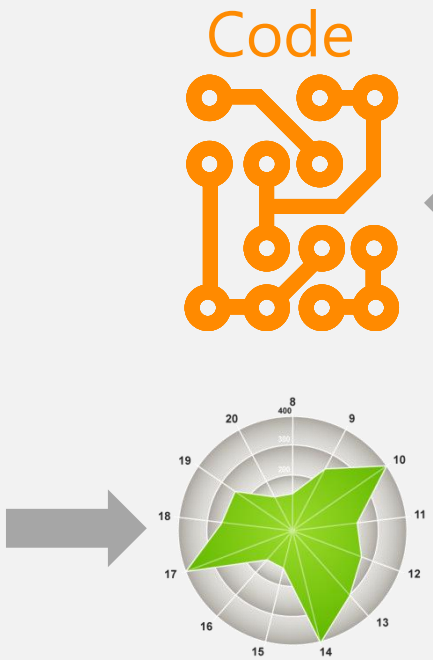
MapReduce: Move Code to the Data



So How Does It Work?

SECOND, TAKE THE PROCESSING TO THE DATA






```
// Map Reduce function in JavaScript

var map = function (key, value, context) {
var words = value.split(/[^a-zA-Z]/);
for (var i = 0; i < words.length; i++) {
          if (words[i] !== "")
context.write(words[i].toLowerCase(),
1);}
}};

var reduce = function (key, values, context)
{
var sum = 0;
while (values.hasNext()) {
sum += parseInt(values.next());
     }
context.write(key, sum);
};
```

MapReduce – Workflow

Data Acquisition & Modeling

Collaboration & Visualization

Analysis & Data Mining

Dissemination, Sharing, Preservation

"It *takes more time to hand a project from the seismic guys to me to the engineers in production than it does to figure out the oil field plays.*"

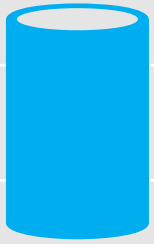
Geologist,
Major oil and gas company

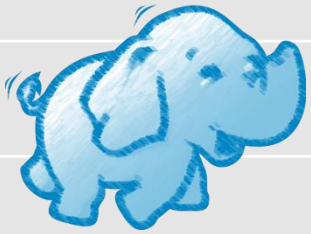
"*Our weather model and resulting data sets should be accessible to universities and other institutions.*"

Aerospace Development Manager,
U.S. Federal Government

Traditional RDBMS vs. NoSQL

	TRADITIONAL RDBMS	HADOOP
Data Size	Gigabytes *(Terabytes)*	Petabytes *(Hexabytes)*
Access	Interactive and Batch	Batch
Updates	Read / Write many times	Write once, Read many times
Structure	Static Schema	Dynamic Schema
Integrity	High (ACID)	Low
Scaling	Nonlinear	Linear
DBA Ratio	*1:40*	*1:3000*

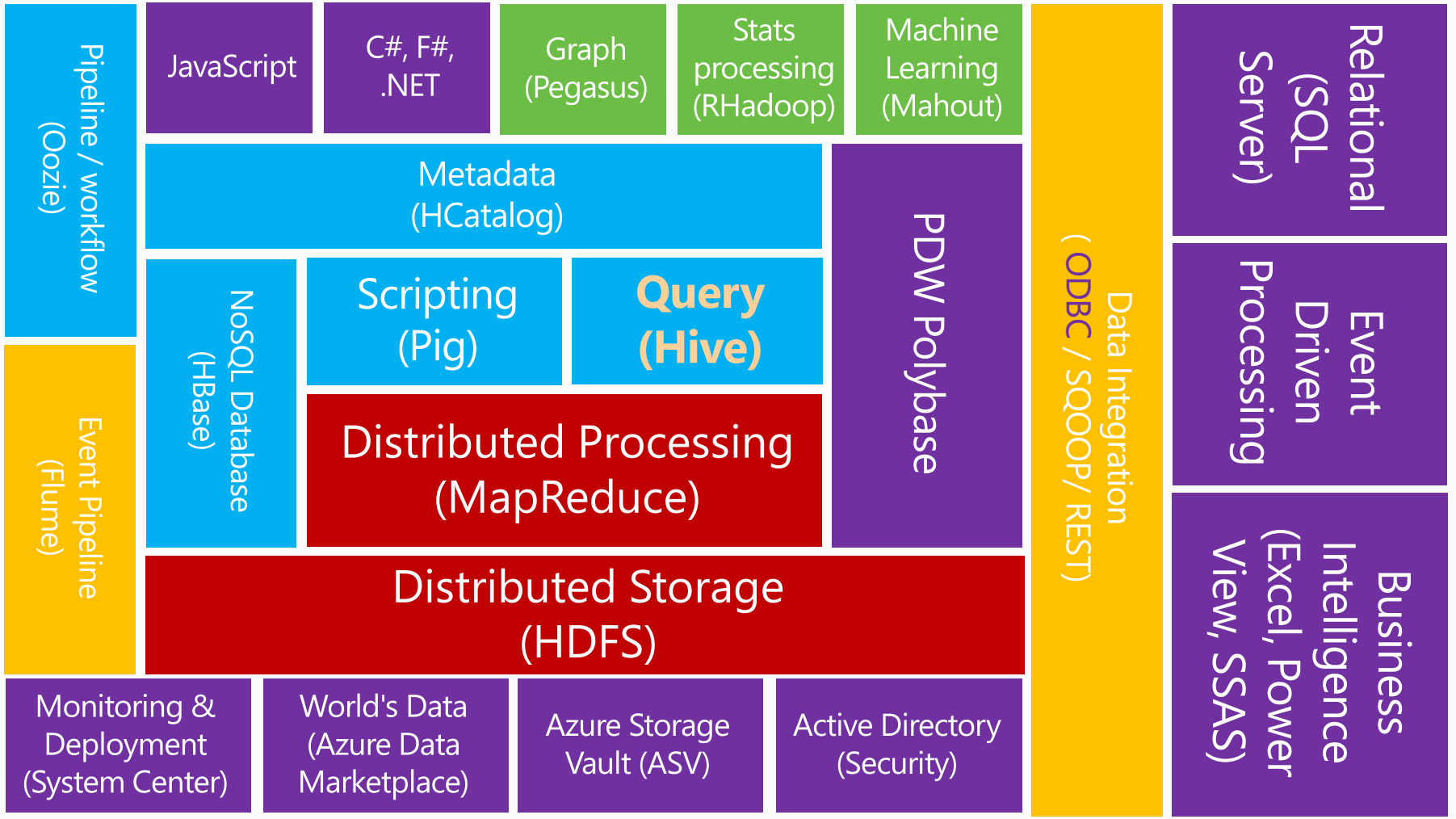




Reference: Tom White's Hadoop: The Definitive Guide

Windows Azure HDInsight Service

HDINSIGHT / HADOOP Eco-System



Legend

Red = Core Hadoop

Blue = Data processing

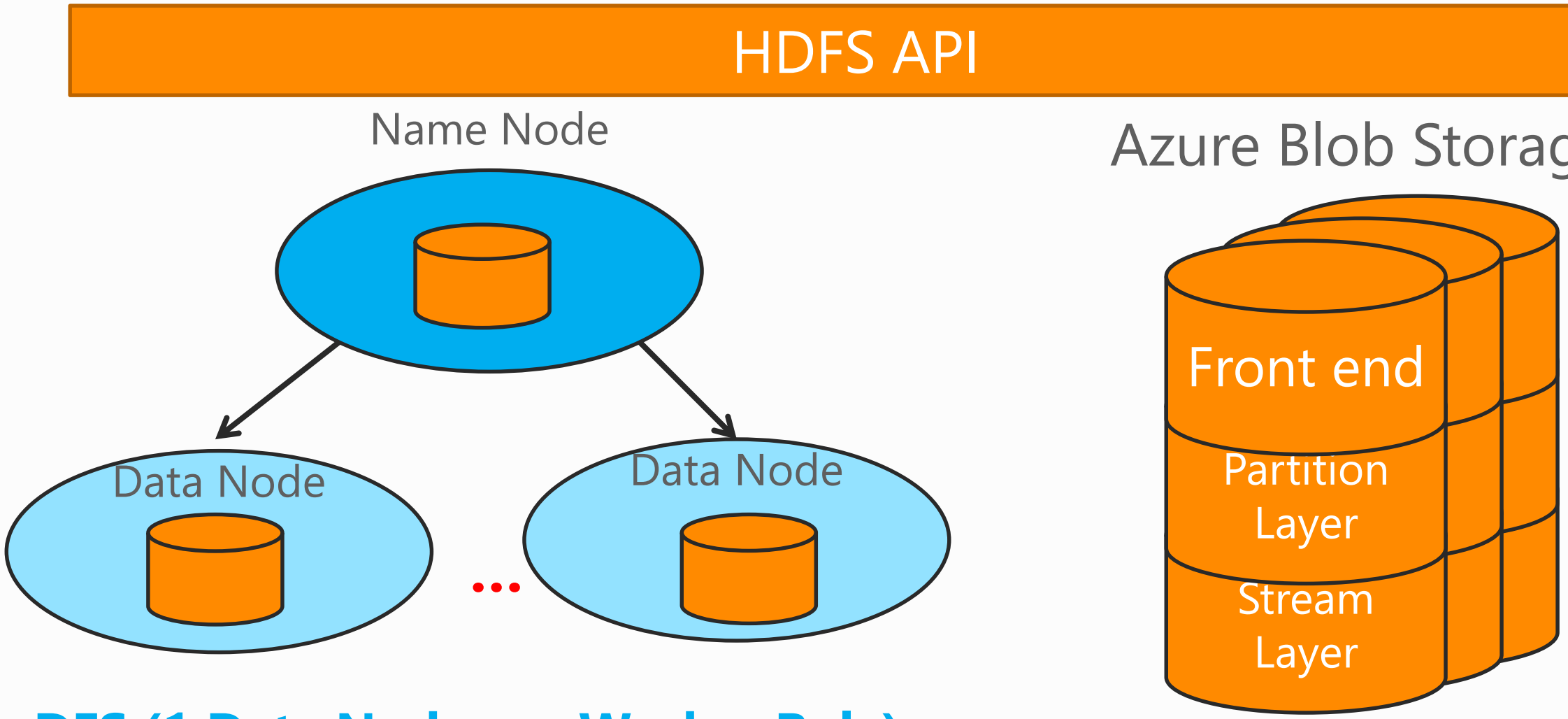
Purple = Microsoft integration points and value adds

Orange = Data Movement

Green = Packages

Storing Data with HDInsight

HDFS on Azure: Tale of two File Systems



DFS (1 Data Node per Worker Role) and Compute Cluster

Azure Storage (ASV)

Azure Storage (ASV)

- Default file system for HDInsight Service
- Provides sharable, persistent, highly-scalable Storage with high availability (Azure Blob Store)
- Azure storage itself does not provide compute
- Fast access from compute nodes to data in same data center
- Several file systems, addressable via:
  `asv[s]:<container>@<account>.blob.core.windows.net/<path>`
- Requires storage key in core-site.xml:

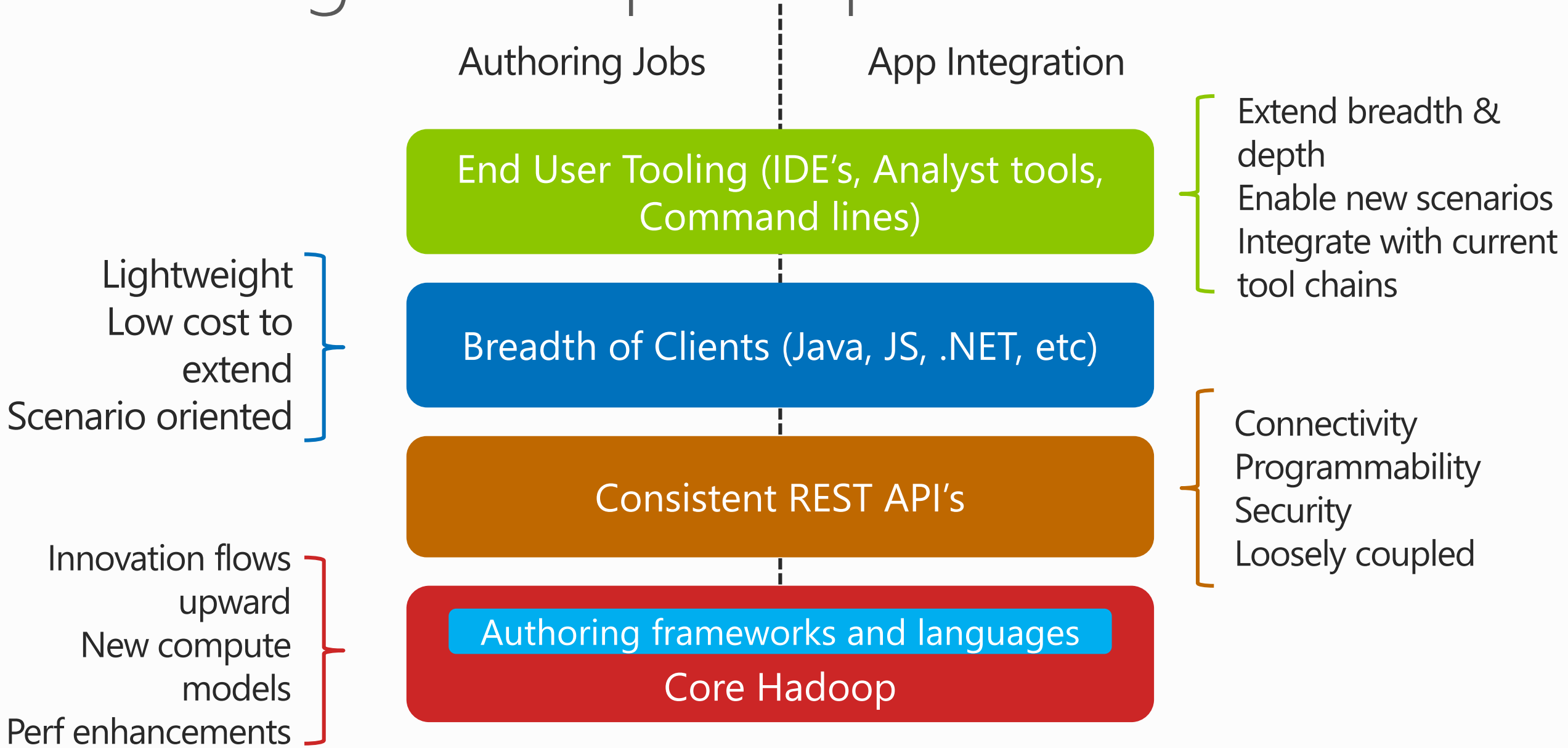
```
<property>
  <name>fs.azure.account.key.accountname</name>
  <value>enterthekeyvaluehere</value>
</property>
```

Programming HDInsight

Programming HDInsight

Existing Ecosystem	Hive, Pig, Mahout, Cascading, Scalding, Scoobi, Pegasus…
.NET	C#, F# Map/Reduce, LINQ to Hive, .NET management clients
JavaScript	JavaScript Map/Reduce, Browser hosted console, Node.js management clients
DevOps / IT Pros	PowerShell, Cross Platform CLI tools

Building Developer Experiences



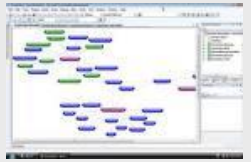
Microsoft Big Data Solution

FAMILIAR END USER TOOLS

Power View

Excel with PowerPivot

Predictive Analytics



Embedded BI

BI PLATFORM

SSAS

SSRS

HDInsight Service

Hadoop On Windows Server

Connectors

Microsoft PDW

UNSTRUCTURED & STRUCTURED DATA

Sensors

Devices

Bots

Crawlers

ERP

CRM

LOB

APPs

Deploying and Interacting With HDInsight Service

demo

Microsoft Hadoop Vision

Insights to all users by activating new types of data

INSIGHTS	Integrate with Microsoft Business Intelligence
ENTERPRISE READY	Choice of deployment on Windows Server + Windows Azure Integrate with Windows Components (AD, Systems Center)
BROADER ACCESS	Easy installation and configuration of Hadoop on Windows Simplified programming with . Net & Javascript integration Integrate with SQL Server Data Warehousing

Contributions proposed back to community distribution

Session Objectives And Takeaways

Session Objective(s):

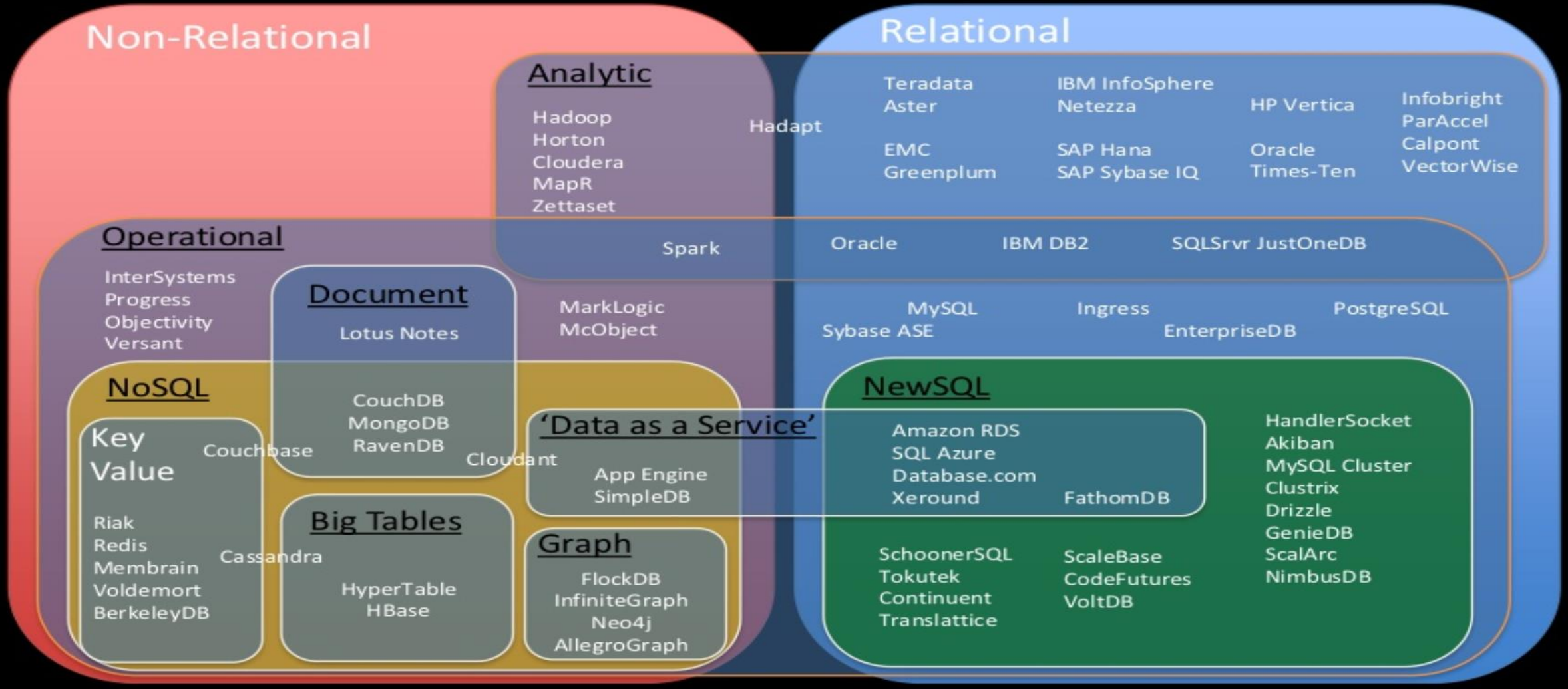
- Know today's Big Data Tools Landscape
- Understand the "Physics" of Big Data Tools
- Identify Batch, **Interactive, and Real-time tools**

Takeaways

- There's a proliferation of WMD(P)
- Identify and understand properties of up and coming tools

Problem

One Size Does Not Fit All



Big Data Processing

	Batch Processing	Interactive analysis	Stream processing
Query runtime	Minutes to hours	Milliseconds to minutes	Never-ending
Data volume	TBs to PBs	GBs to PBs	Continuous stream
Programming model	MapReduce	Queries	DAG
Users	Developers	Analysts and developers	Developers
Originating project	Google MapReduce	Google Dremel	Twitter Storm
Open source project	Hadoop / Spark	Drill / Shark /Impala Hbase	Storm / Apache S4 /Kafka

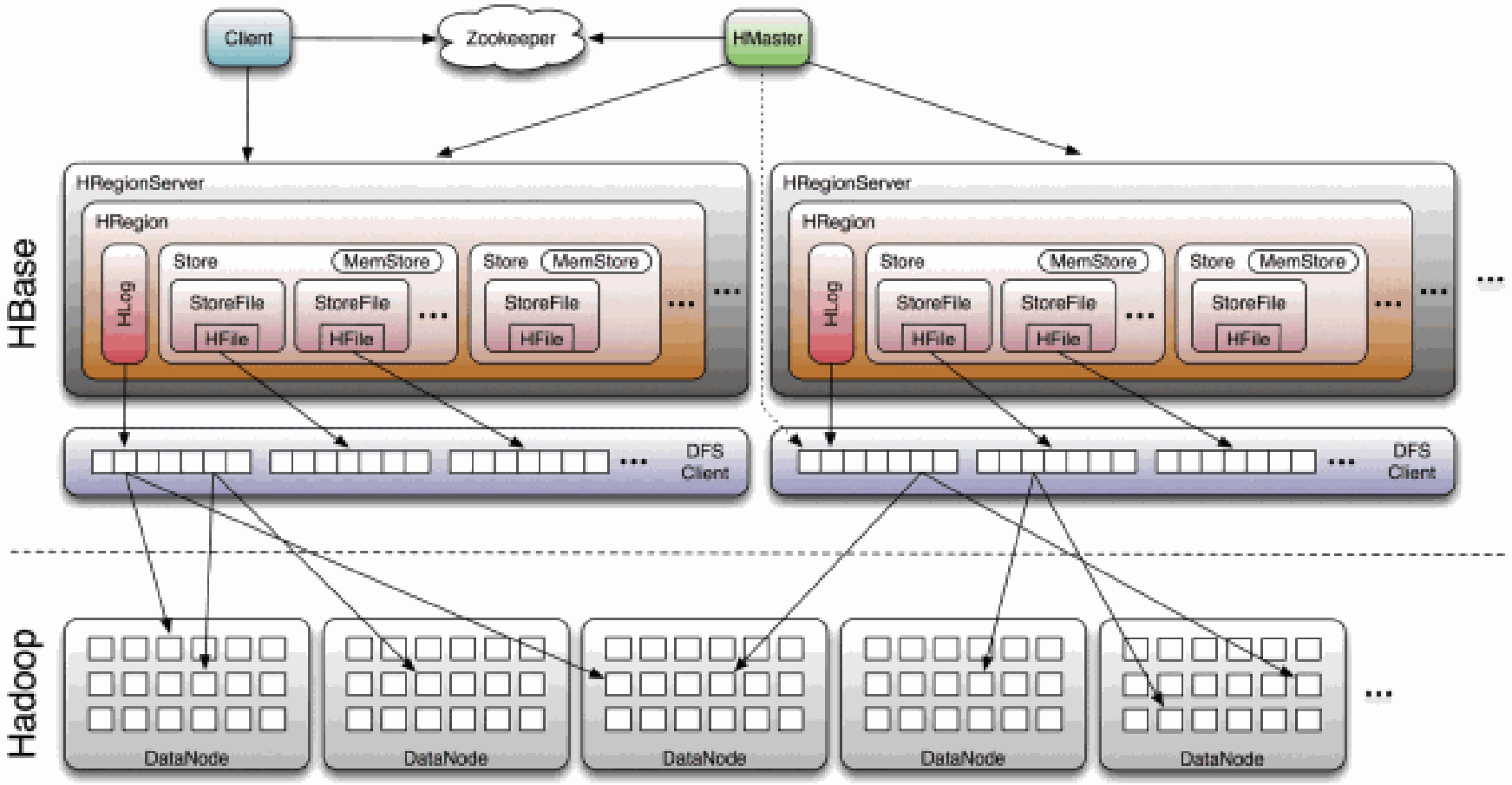
HBase:
Interactive OLTP

Concept: NoSQL (again)

- As David DeWitt described in the PASS 2011 Day 3 Keynote, the NoSQL is analogous to OLTP (as Hadoop is to BI)
- Comprised of many components: HBase, Cassandra, MongoDB, CouchBase, MemcacheD, Cache role (Azure) etc.
- Implementations of Google's BigTable – distributed storage system for managing structured data at very large sizes

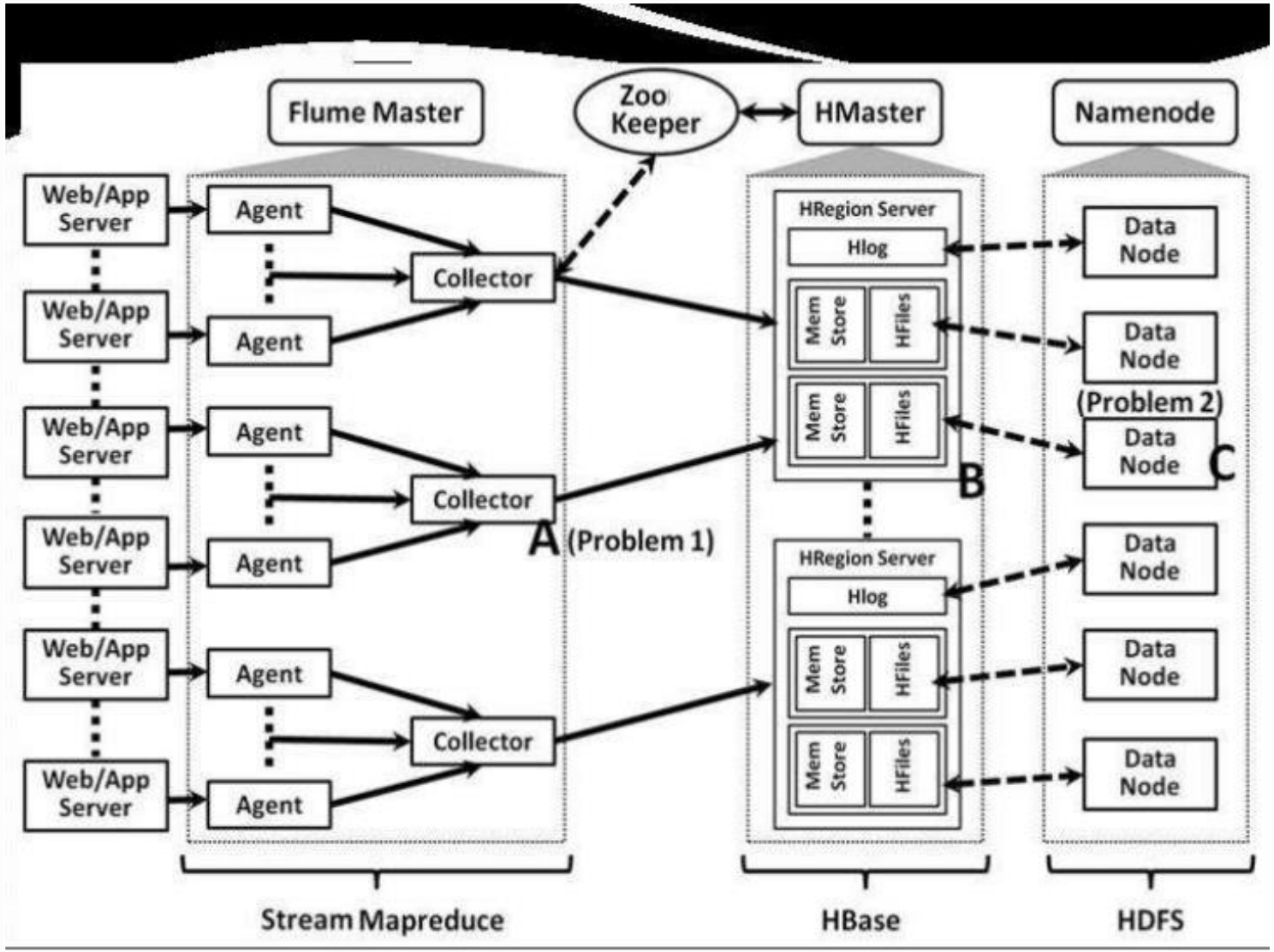
HBase

persistent | distributed



- In Memory
- Efficient at Random Reads/Writes
- Distributed, large scale data store
- Utilizes Hadoop for persistence
- Both HBase and Hadoop are distributed

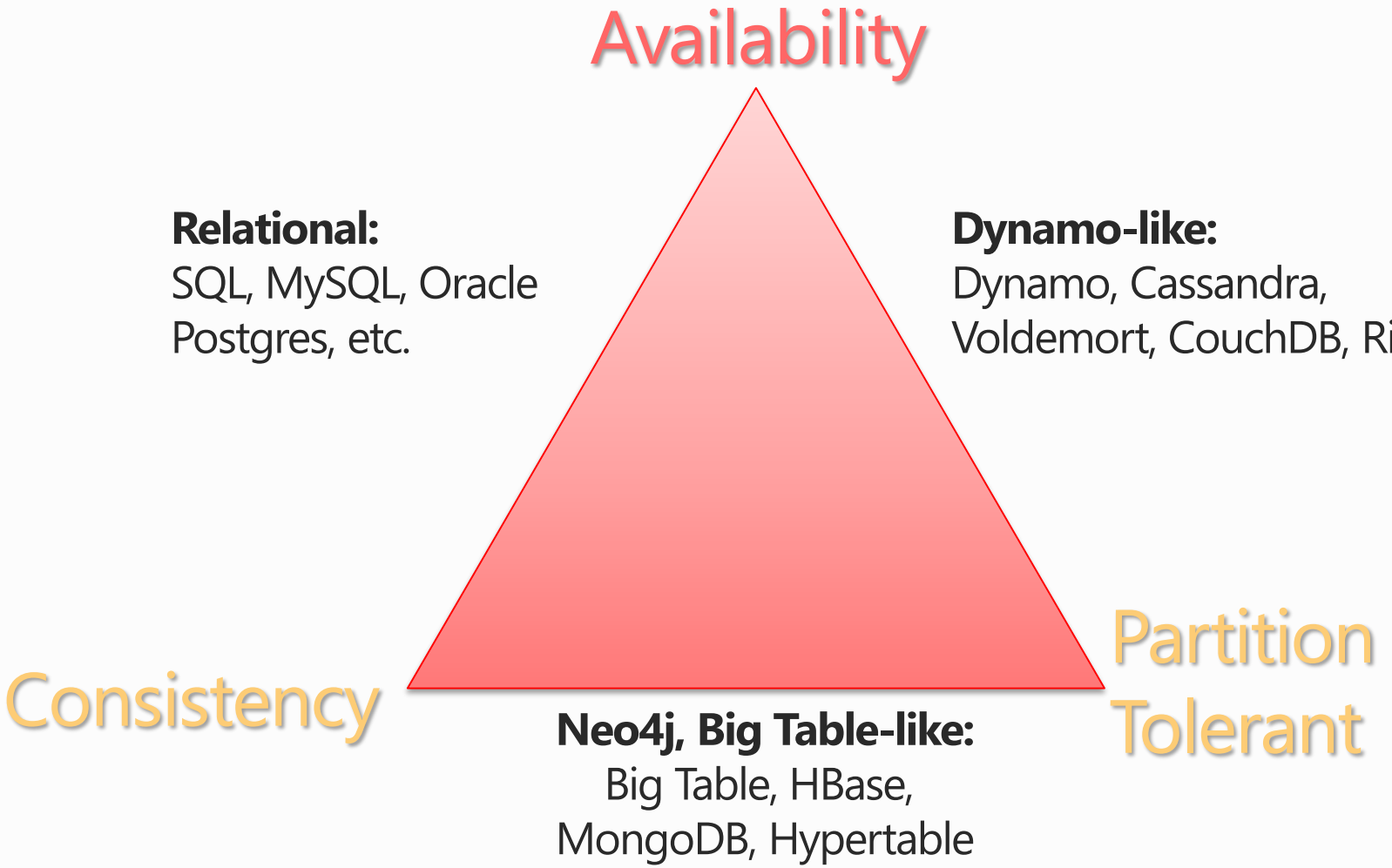
The "War" between HBase and Cassandra



- Cassandra was originally created at Facebook
- But for Facebook Message, they decided to use HBase
- To the left is Facebook's HBase / Hadoop / Flume architecture for Facebook Messages and Streaming MR

Reference: http://www.slideshare.net/danirayan/streaming-map-reduce

CAP Theorem



- Reference: Cassandra – The Definitive Guide
- Myth Rumors Fud Hate NoSQL Cassandra vs. HBase
- Both have their +ve and –ve
- Both have their own use cases:
  - Many gaming systems need consistency b/c they need the same information back
  - Netflix needs availability of videos at all times

HBase

Pros

- Perfect for handling large volumes of transactions, updates, volatile reads
- Facebook messages showcase
- Potential specific DW scenarios

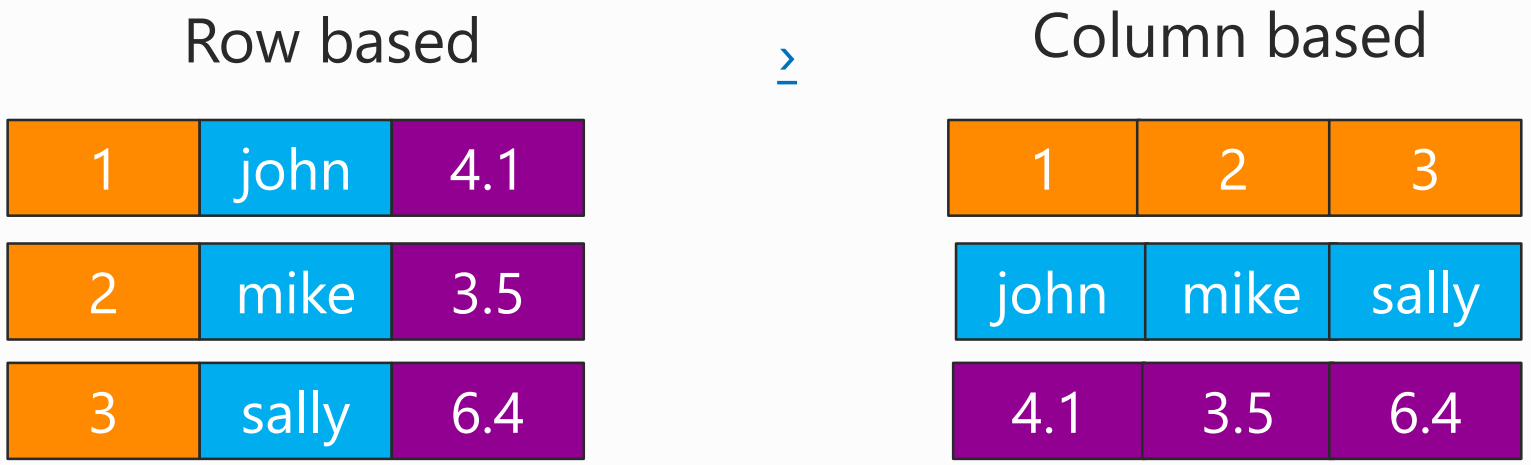
Cons

- Very hard to maintain and configure whether one or many
- Balancing regions requires deep understanding of data patterns
- A lot of logic must be done in middle tier though this is changing

Drill (Dremel)

Basic Info

- An Apache incubation project inspired by Dremel
- Designed to scale to 10,000 servers, query PBs in Minutes
- Supported by Jason Frantz from MapR
- ETL, Nested Columns, fast sequential scanning



APACHE DRILL

user

REST

CLI

native API

interfaces and apps

processing

SQL

DrQL

Mongo QL

DSL

...

parser - pluggable

Query Planner

Execution Engine

Storage Engine

data sources

Cassandra

APACHE HBASE

HDFS

mongoDB

hadoop

cluster or local - pluggable

Impala (Dremel)

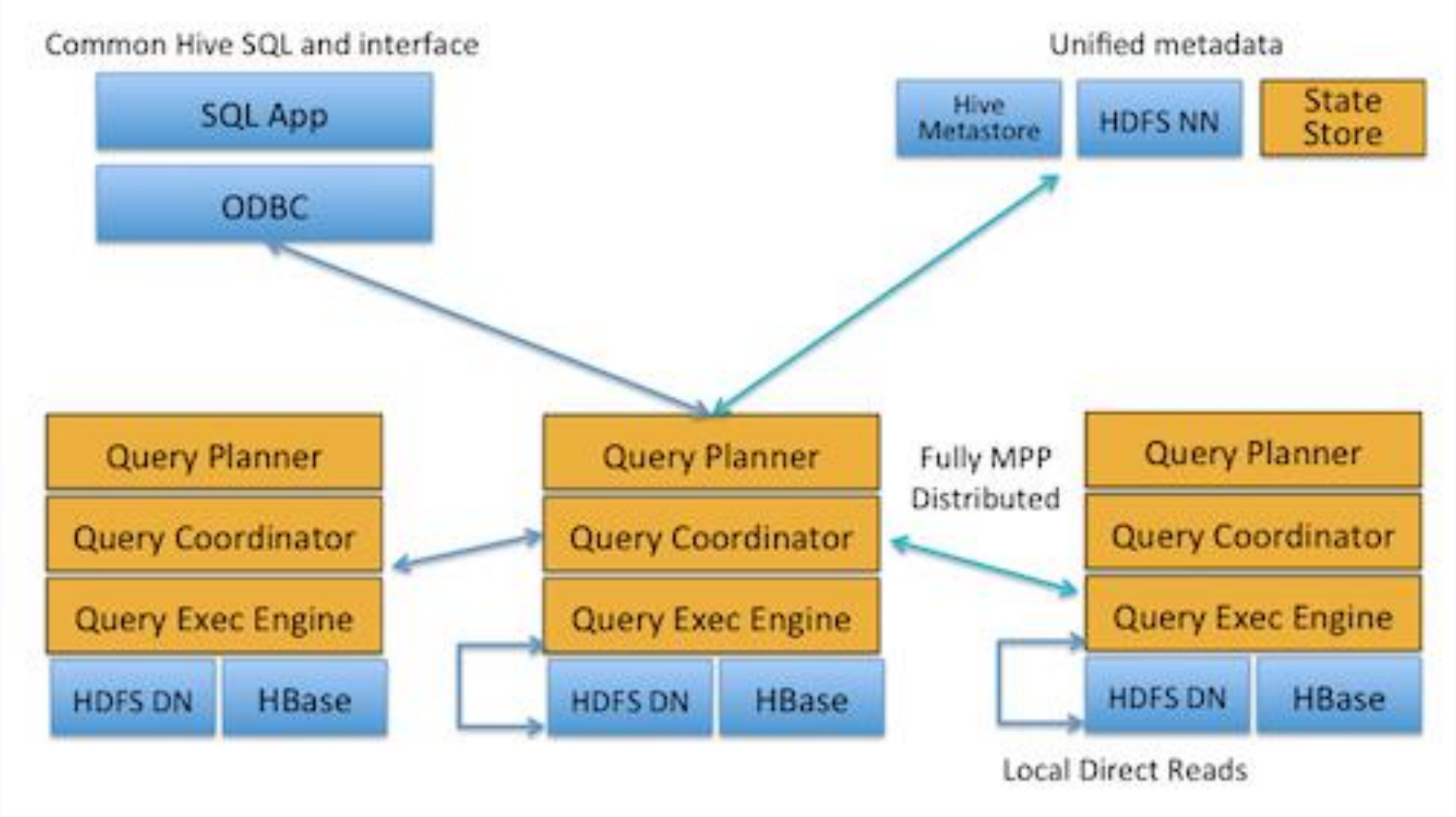
What is Impala

- Inspired by Dremel Paper (http://research.google.com/pubs/pub36632.html)
- Columnar Storage format: Trevni (by Doug Cutting, Hadoop creator)
- Distributed scalable aggregations, adhoc query system
- Utilizes HiveQL (SQL92 subset)
- Directly queries HDFS and Hbase
  - Text Files and Sequence Files
  - Compression Codecs: **Snappy**/Gzip/BZ2)
  - Avro, RCFile, LZO ext files, and Trevni

How does it do fast queries

- MPP like infrastructure, avoid using Map Reduce
  - Small-Large Joins vs. Large-Large Joins
- I/O Bound workloads faster by 3-4x
- Queries requiring multiple MR phases in Hive up to 45x faster
- Queries that run in-memory cached up to 90x faster
- Currently Missing
  - SerDes
  - UDFs
  - Joins are done in available memory space of smallest node
  - Views

Real-Time Queries in Apache Hadoop (for real)



HiveQL
Hive

```
[cloudera@localhost impalascripts]$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
Hive history file=/tmp/cloudera/hive_job_log_cloudera_201302031637_1593885508.txt
hive> select * from hivesampletable limit 10;
OK
8       18:54:20        en-US   Android Samsung SCH-i500        California      United States   13.9204007      0       0
23      19:19:44        en-US   Android HTC     Incredible      Pennsylvania    United States   NULL    0       0
23      19:19:46        en-US   Android HTC     Incredible      Pennsylvania    United States   1.4757422       0       1
23      19:19:47        en-US   Android HTC     Incredible      Pennsylvania    United States   0.245968        0       2
28      01:37:50        en-US   Android Motorola        Droid X Colorado        United States   20.3095339      1       1
28      00:53:31        en-US   Android Motorola        Droid X Colorado        United States   16.2981668      0       0
28      00:53:50        en-US   Android Motorola        Droid X Colorado        United States   1.7715228       0       1
28      16:44:21        en-US   Android Motorola        Droid X Utah    United States   11.6755987      2       1
28      16:43:41        en-US   Android Motorola        Droid X Utah    United States   36.9446892      2       0
28      01:37:19        en-US   Android Motorola        Droid X Colorado        United States   28.9811416      1       0
Time taken: 7.388 seconds
hive> select count(*) from hivesampletable;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_1359924783894_0002, Tracking URL = http://localhost.localdomain:8088/proxy/application_1359924783894_0002
/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=127.0.0.1:8021 -kill job_1359924783894_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2013-02-03 16:38:28,360 Stage-1 map = 0%,  reduce = 0%
2013-02-03 16:38:48,080 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.55 sec
2013-02-03 16:38:49,297 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.55 sec
2013-02-03 16:38:50,718 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.32 sec
2013-02-03 16:38:51,877 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.17 sec
MapReduce Total cumulative CPU time: 5 seconds 170 msec
Ended Job = job_1359924783894_0002
MapReduce Jobs Launched:
Job 0: Map: 1  Reduce: 1   Cumulative CPU: 5.17 sec   HDFS Read: 5015717 HDFS Write: 6 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 170 msec
OK
59793
Time taken: 47.289 seconds
hive>
```

HiveQL
Impala

```
[cloudera@localhost impalascripts]$ impala-shell
Welcome to the Impala shell. Press TAB twice to see a list of available commands.

Copyright (c) 2012 Cloudera, Inc. All rights reserved.

(Build version: Impala v0.4 (f360eba) built on Thu Jan 17 10:50:01 PST 2013)
[Not connected] > connect localhost:21000
Connected to localhost:21000
[localhost:21000] > select * from hivesampletable limit 10;
Query: select * from hivesampletable limit 10
Query finished, fetching results ...
8       18:54:20        en-US   Android Samsung SCH-i500        California      United States   13.9204007      0       0
23      19:19:44        en-US   Android HTC     Incredible      Pennsylvania    United States   NULL    0       0
23      19:19:46        en-US   Android HTC     Incredible      Pennsylvania    United States   1.4757422       0       1
23      19:19:47        en-US   Android HTC     Incredible      Pennsylvania    United States   0.245968        0       2
28      01:37:50        en-US   Android Motorola        Droid X Colorado        United States   20.3095339      1       1
28      00:53:31        en-US   Android Motorola        Droid X Colorado        United States   16.2981668      0       0
28      00:53:50        en-US   Android Motorola        Droid X Colorado        United States   1.7715228       0       1
28      16:44:21        en-US   Android Motorola        Droid X Utah    United States   11.6755987      2       1
28      16:43:41        en-US   Android Motorola        Droid X Utah    United States   36.9446892      2       0
28      01:37:19        en-US   Android Motorola        Droid X Colorado        United States   28.9811416      1       0
Returned 10 row(s) in 0.26s
[localhost:21000] > select count(*) from hivesampletable;
Query: select count(*) from hivesampletable
Query finished, fetching results ...
59793
Returned 1 row(s) in 0.23s
[localhost:21000] >
```

Hive vs. Impala

Query Type	Tech	Query
Select * [] limit 10;	Hive	7.388s
	Impala	0.26s
Select count(*) from []	Hive	47.289s
	Impala	0.23s

Storm / Kafka
Real-time Streaming

Basic Info

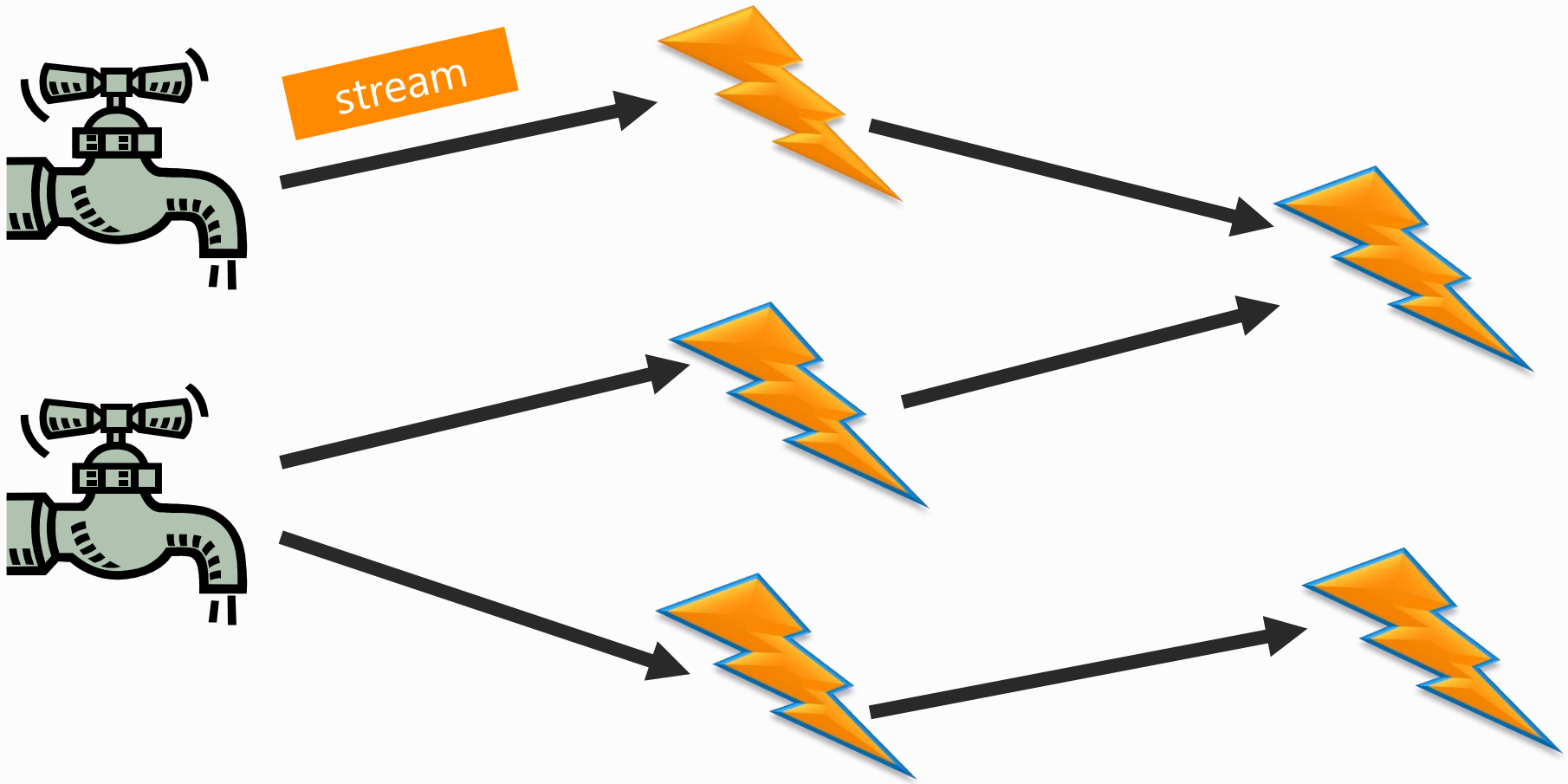
- Developed and currently used by twitter
- Guaranteed data processing (replay if incomplete)
- Horizontal scalability
- Fault-tolerance
- Higher level abstraction than message passing
- Experimental Windows Version upon request

STORM Concepts

- Streams

- Spouts: Source of streams
- Bolts: Functions, Filters, Aggregation, Joins, DB R/W
- Topologies: Grouping of Spouts and Bolts

Topology: Network of Spouts and Bolts



Demo: STORM

Stream grouping

- Shuffle grouping: pick a random task
- Fields grouping: mod hashing on a subset of tuple fields
- All grouping: send to all tasks
- Global grouping: pick task with lowest ID

Spark/Shark

Note, the next section of slides are almost verbatim copies of Matei Zaharia's slides [Spark and Shark: High-Speed In-Memory Analytics]

What is Spark / Shark?

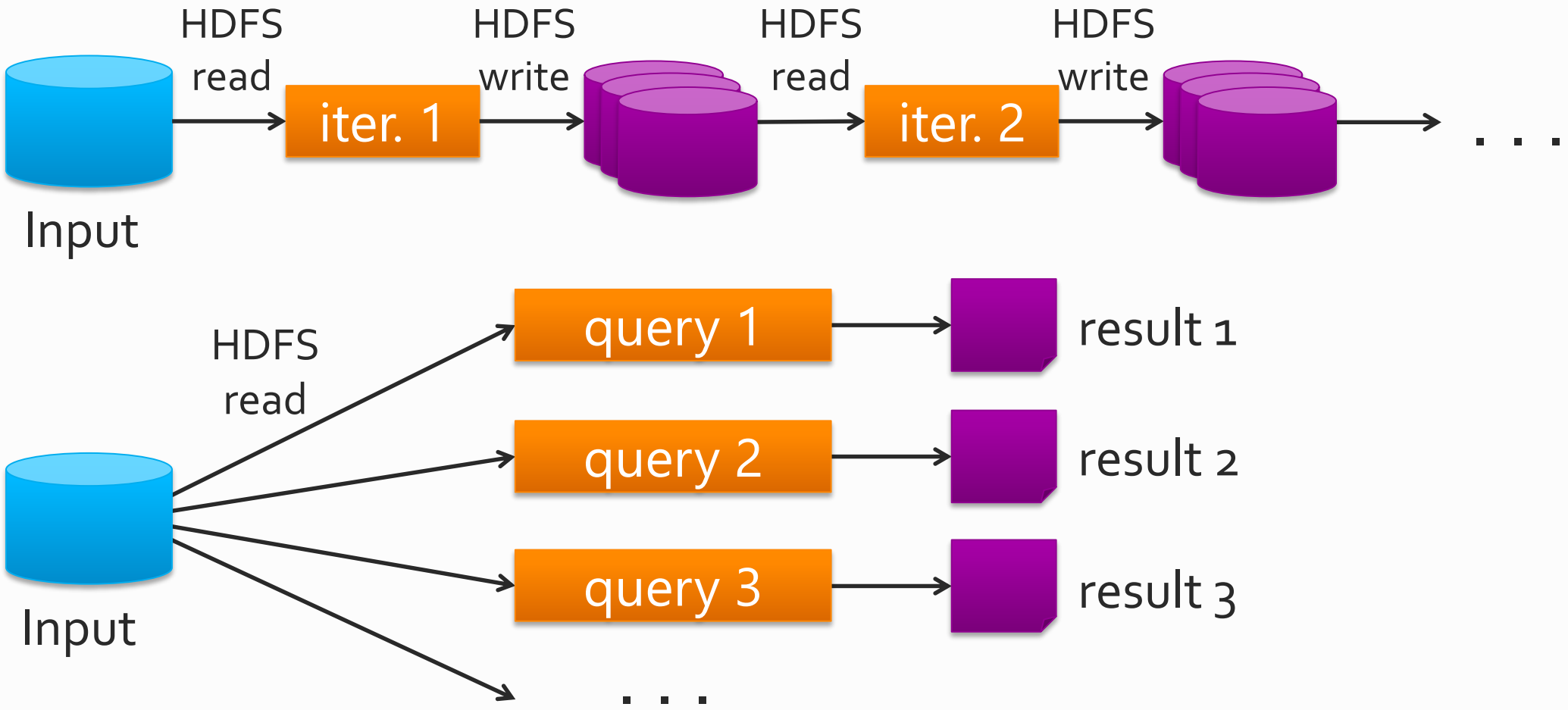
Spark

- Fast MapReduce-like engine (NOT Hadoop)
- In Memory data storage via RDDs (Resiliant Distributed Datasets)
- Up to 30x faster than Hadoop
- Compatible with Hadoop's storage APIs (HDFS, HBase, SequenceFiles, etc.)

Shark

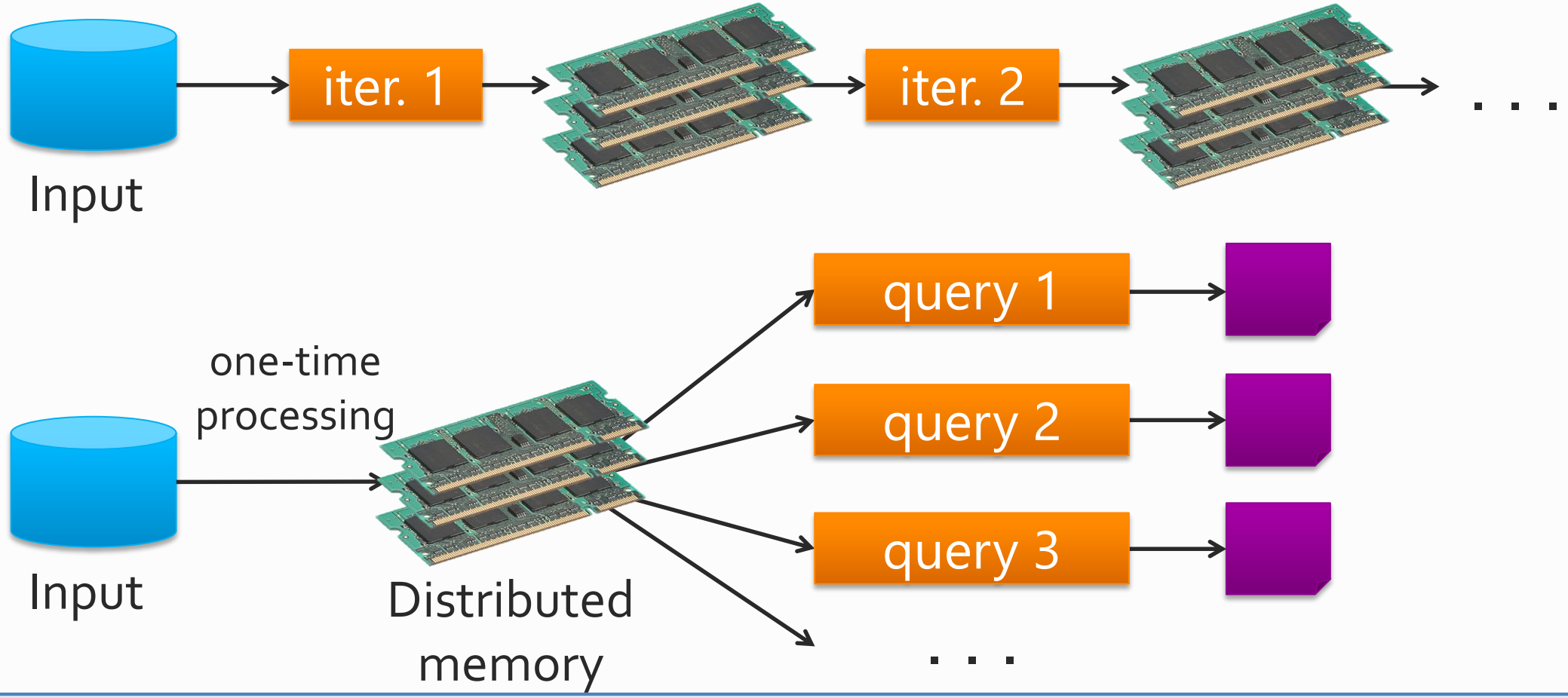
- Port of Apache Hive to run on Spark
- Compatible with existing Hive data, metastores, etc.
- Up to 40x faster than Hive (on Hadoop)

Data Sharing in MapReduce



Slow due to replication, serialization, and disk IO

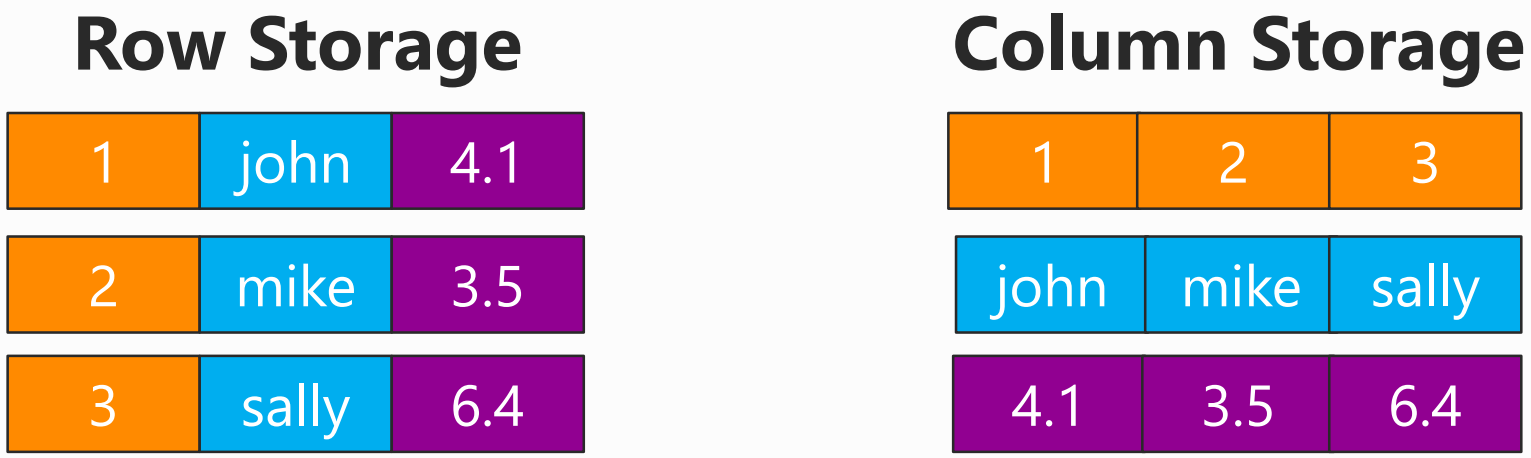
Data Sharing in Spark



10-100× faster than network and disk

Efficient In-Memory Storage

- Simply caching Hive records as Java objects is inefficient due to high per-object overhead
- Instead, Shark employs column-oriented storage using **arrays of primitive types**



Spark / Shark in Action

RowCount and WordCount via Pig

```
# Get rowcount from Mobile Sample Data
A = LOAD '/samples/HiveSampleData.txt' USING TextLoader as (line:chararray);
B = group A all;
C = foreach B generate COUNT(A);
dump C;

# Perform Wordcount
B = foreach A generate flatten(TOKENIZE((chararray)$0)) as word;
C = group B by word;
D = foreach C generate COUNT(B), group;
E = group D all;
F = foreach E generate count(D);
dump F;
```

RowCount and WordCount via Spark

```
# extract Mobile Sample data from HDFS
val mobile = sc.textFile("hdfs://localhost:9000/samples/HiveSampleData.txt")

# Get rowcount from Mobile Sample data
mobile.count

# Count the number of words in the Mobile Sample data, split via " "
var wc = mobile.flatMap(line => line.split(" ")).map(word => (word,
1)).reduceByKey(_ + _)

wc.count
```

Pig vs. Spark

Query Type	Technology	1st query	2nd query
RowCount	Pig	39s	
	Spark	0.53s	0.10s
Distinct WordCount	Pig	84s	
	Shark	1.43s	0.40s

Hive and Shark

```
# List top ten rows
select * from hivesampletable limit 10;

# Count
select count(*) from hivesampletable;

# Group by
select devicemake, count(*) from hivesampletable group by devicemake;
```

Hive vs. Shark

Query Type	Tech	1st Query	2nd Query
Select * [] limit 10;	Hive	4.7s	
	Shark	1.454s	0.343s
Select count(*) from []	Hive	34.115s	
	Shark	0.945s	0.449s
Select col, cnt, from [] group by col	Hive	31.593s	
	Shark	0.528s	0.381s

Spark / Shark

Pros

- Map Reduce functionality and flexibility
- Massively improves speed of MR and Hive queries in distributed fashion
- Scala / Hive
- Designed for EC2, mesos, standalone, cluster, etc.
- Smart data movement

Cons

- Still very much in beta
- Fast for MR, but we can go faster! (e.g. Count 1.28M rows took 9.928s)
- No Hive ODBC/JDBC … yet
- Potential Mesos dependency complicates things….a lot
- Scala

Session Objectives And Takeaways

Session Objective(s):

- Know today's Big Data Tools Landscape
- Understand the "Physics" of Big Data Tools
- Identify Batch, **Interactive, and Real-time tools**

Takeaways

- There's a proliferation of WMD(P)
- Use "Physics" to identify and understand new tools

Resources

- http://www.windowsazure.com/
- http://hadoop.apache.org/
- Nuget: http://nuget.org/packages?q=hadoop
- Hadoop SDK: http://hadoopsdk.codeplex.com

- Follow @wenmingye for Questions and latest info.