Hadoop, Spark and Flink Explained to Oracle DBA and Why They Should Care

Kuassi Mensah, Director Product Management
Oracle Corporation
@kmensah – db360.blogspot.com

Oracle BIWA Summit 2017

Speaker Bio

- Director of Product Management at Oracle
  (i) Java products for the Oracle database (OJVM, JDBC, UCP, App Cont, TG, etc)
  (ii) Oracle Datasource for Hadoop (OD4H), and upcoming Spark, Flink and so on
  (iii) JavaScript in the Oracle database (Nashorn with OJVM)
- Graduate MS in CS from the Programming Institute of University of Paris
- Frequent speaker
  Oracle Open World, JavaOne, BIWA , Collaborate/IOUG, RMOUG, Data Summit, Node Summit, UKOUG, DOAG,OUGN, BGOUG, OUGF, OTN LAD (GUOB, ArOUG, ORAMEX), OTN APAC (Sangam,OTNYathra, China, Thailand, etc),
- Author: *Oracle Database Programming using Java and Web Services*
- Social network handles
  @kmensah, http://db360.blogspot.com/, https://www.linkedin.com/in/kmensah

Agenda

- ✓ From Big Data to Fast Data
- Apache Hadoop
- Apache Spark
- Apache Flink
- Why Should Oracle DBA Care

Objectives

- Understand the evolution of Big Data and massive scale data processing
- Understand Hadoop, Spark and Flink: their strengths and limitations
- Understand why Oracle DBA should care and learn more about these frameworks

MapReduce: Where It All Begins

- Scalable Data Processing

  Google's paper *"MapReduce: Simplified Data Processing on Large Clusters"*

  https://research.google.com/archive/mapreduce-osdi04.pdf

- Simplicity

- Present: obsolete!

Big Data

- Google Search: 40,000+ search queries per second
- Exponential growth of data volume
  - 44 Zetabytes (44 trillion GB) of data in the digital universe by 2020
  - Every online user will generate ~1.7 MB of new data per second
- Massive-scale data processing requirements

Streaming Data

- Streaming data or unbound data is ubiquitous and emit continuous flows of events
  IoT, mobile devices data, sensors data, financial transactions, stock feeds, logs, retail, call routing, etc
- Business shift from reactive to proactive interactions; must process data as it enters the system
- How fast can you analyze your data and gain insights is more important than how big your data is
- New processing model: stream processing of unbound data
- New Big Data processing frameworks: Spark streaming,Flink, Storm, Stanza

Streaming Data Processing Concepts

Stream processing: analyze a fragment/window of data stream

- *Low Latency:* sub-second
- *Timestamp*: event-time, ingestion time, or processing time -- cf Star Wars
- *Windowing*: fixed/tumbling, sliding, session
- *Watermark:* defines when a window is considered done and GCed
- *Out-of-order processing*
- *In-order processing*
- *Triggers:* when to run the computation = watermark progress |event time progress| processing time progress| punctuations
- *Iterative, incremental and interactive processing*
- *Delivery guarantee*: at least once, exactly once, end to end exactly once
- *Repeatability*
- *Event prioritization*
- *Backpressure support*

Program Agenda

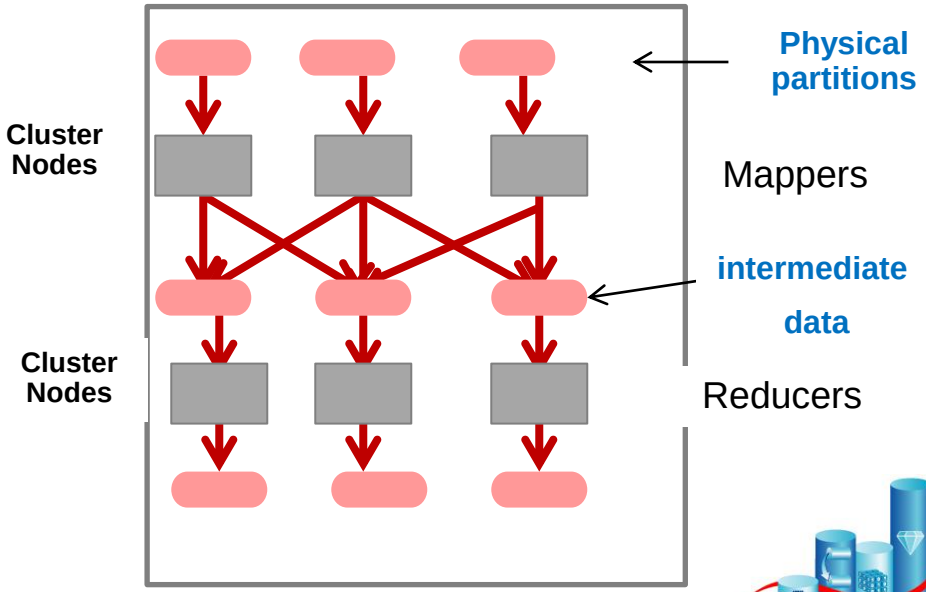
1. From Big Data to Fast Data
2. Apache Hadoop
3. Apache Spark
4. Apache Flink
5. Why Should Oracle DBA Care

Apache Hadoop 1.0

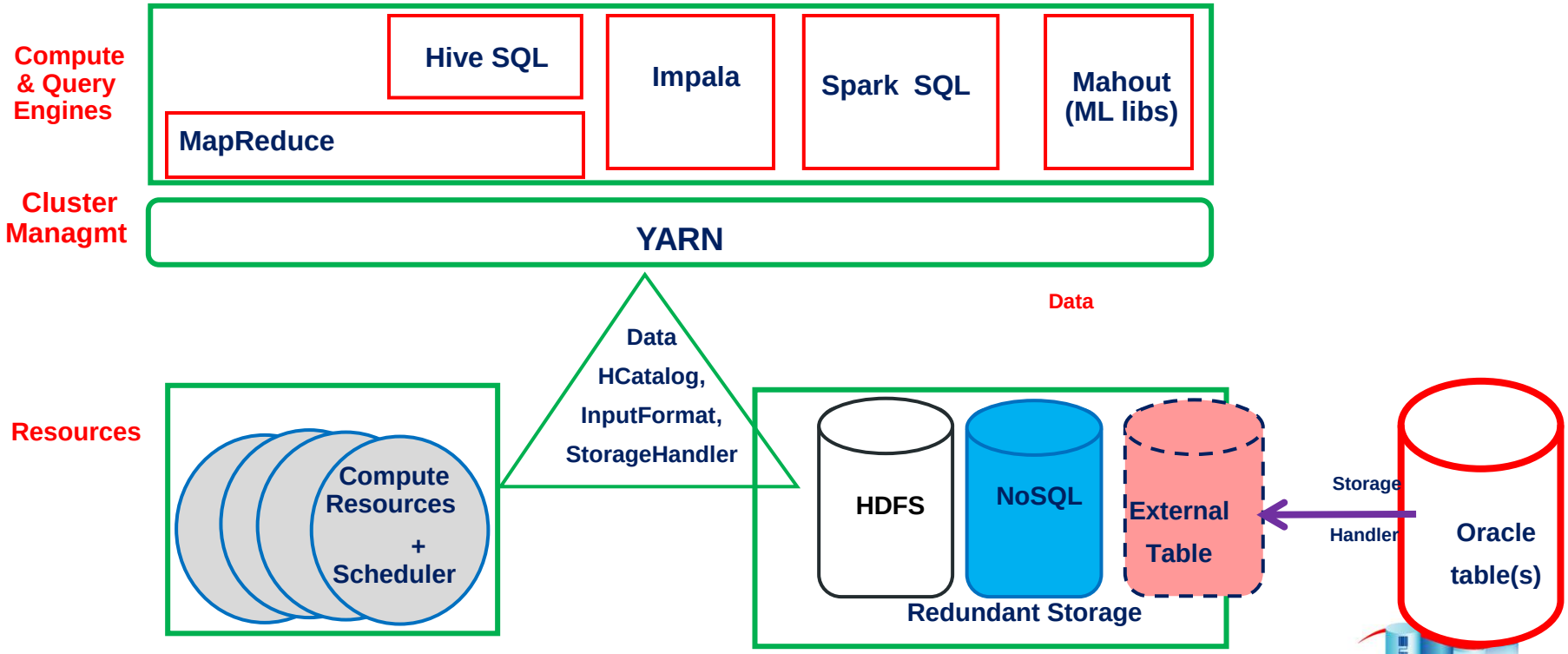
First Open-source MapReduce framework & ecosystem

- Processing model: batch
- 2004: HDFS + MapReduce (Python)
- 2006: Apache Hadoop (Java)
- 2009: 1 TB Sort in 209 sec
- 2010: 100TB sort in 173 min
- 2014: 100TB sort in 72 min

Hadoop Cluster (e.g., Big Data Appliance)



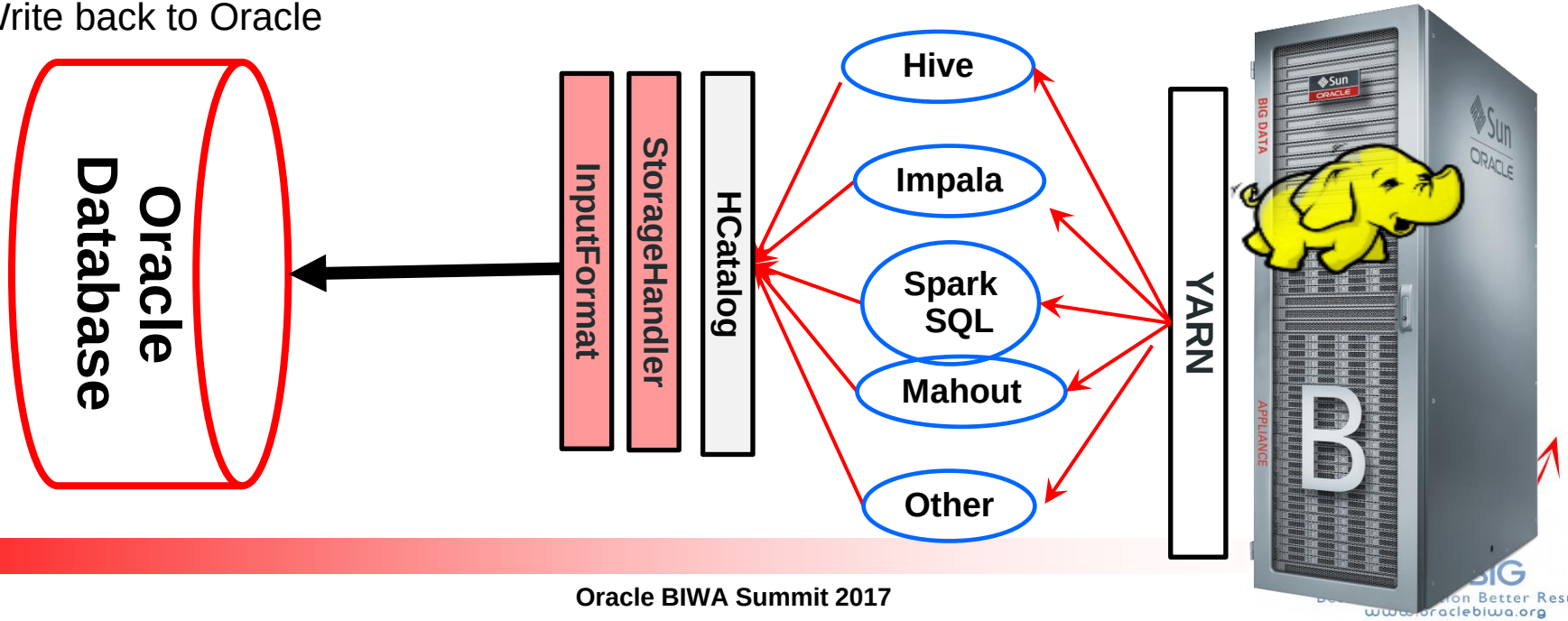
Apache Hadoop 2.0



Oracle Datasource for Hadoop (OD4H)

Released

- Direct, parallel, fast, secure and consistent access to Oracle database
- Join Big Data and Master Data
- Write back to Oracle



Hadoop Real-World Use Cases

1. Airbus Uses Big Data Appliance to Improve Flight Testing
2. BAE Systems Choose Big Data Appliance for Critical Projects
3. AMBEV chose Oracle's Big Data Cloud Service to expedite their database integration needs.
4. Big Data Discovery Helps CERN Understand the Universe
5. See more use cases @ http://bit.ly/1Oz2jCF

Strengths & Limitations of Apache Hadoop

- Strengths
  - Good for batch processing of data-at-rest i.e., Association Rules Mining
  - Inexpensive disk storage -> can handle enormous datasets

- Limitations:

  - Limited to batch processing: not suitable for streaming data processing
  - Static partitioning
  - Materialization on each job step
  - Complex processing requires multi-staging
  - Disk-based operations prevents data sharing for interactive ad-hoc queries

Program Agenda

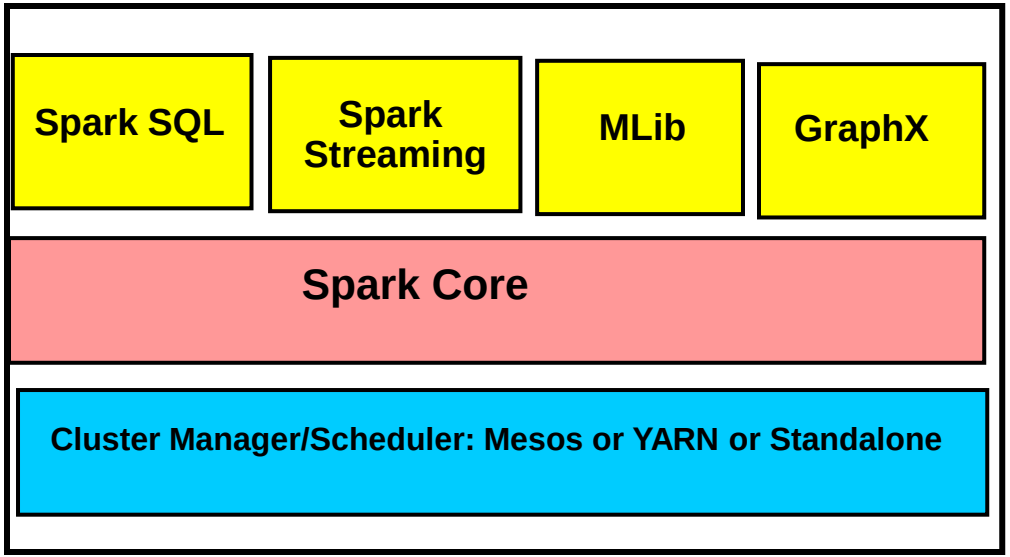
1. From Big Data to Fast Data
2. Apache Hadoop
3. Apache Spark
4. Apache Flink
5. Why Should Oracle DBA Care

Apache Spark Concepts

- 2009: AMPLab -> hybrid engine for batch and streaming processing
- Spark Streaming: for real-time streaming data processing; based on micro batching.
- RDD: partitioned datasets, fault tolerance abstraction for in-memory data sharing
  - Immutable, two types of operations: (i) Transformation -> new RDD; (II) Action -> new value
- Dataframe: conceptually equivalent to a table.
  - Registering a DataFrame as a table allows Spark-SQL queries over its data.
- Spark apps on Hadoop clusters run up to 100 times faster in memory and 10 times faster on disk.

Spark Architecture

- The largest known Spark cluster has 8000 nodes.
- More than 1000 organizations are using Spark in production
- Sort 100 TB of data 3X faster than Hadoop MapReduce on 1/10th of the machines



How Apache Spark Works

- All work expressed as
  (i) transformations : creating new RDDs, or transforming existing RDDs
  (ii) actions: calling operations on RDDs
- Execution plan as a Directed Acyclic Graph (DAG) of operations
- Every Spark program and shell session will work as follows:

  1) Create some input RDDs from external data.
  2) Transform them to define new RDDs using transformations like filter().
  3) Ask Spark to persist any intermediate RDDs that will need to be reused.
  4) Launch actions such as count() and first() to kick off a parallel computation, which are optimized and executed by Spark executor.

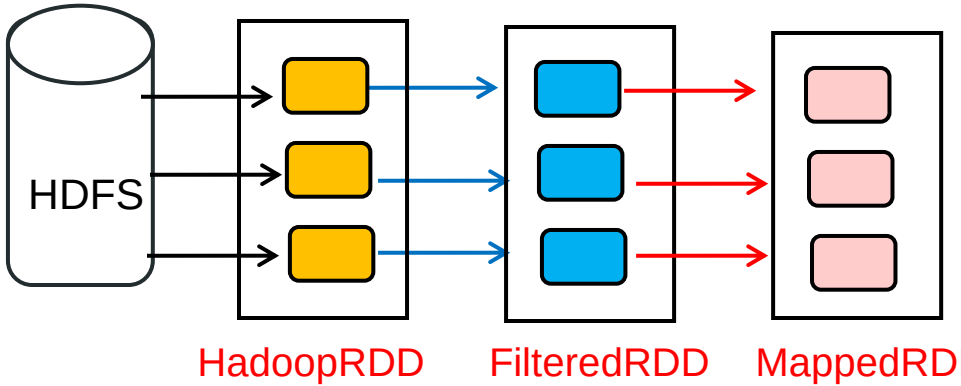
Basic Spark Example

```
Lines =spark.textFiles("hdfs://…")
Errors = ines.filter(_.startswith("ERROR")
messages = error.map(_.split('\t') (2))
message.persist()
```

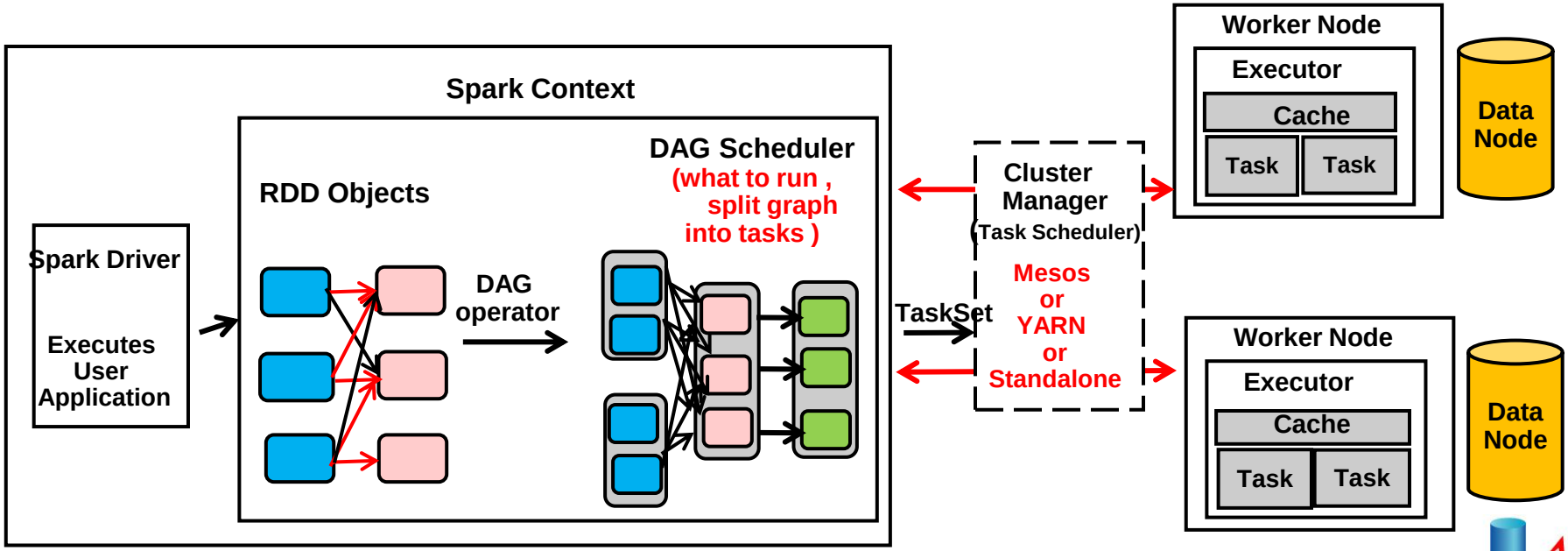
HadoopRDD

FilteredRDD

MappedRDD



Spark Workflow



Spark Real World Use Cases

- Security, finance: fraud or intrusion detection or risk-based authentication
- Log processing, BI/reporting/ETL
- Mobile usage patterns analysis
- Predictive analytics, data exploration
- Game industry: real-time discovering of patterns in-game events
- e-Commerce: real-time transaction information passed to a streaming clustering algorithm like k-means or collaborative filtering

Spark Strengths and Limitations

- Strengths
  - Speed: in-memory processing
  - High throughput
  - Correct under stress: strongly consistent
  - Event processing-time (in-order processing)
  - Spark Streaming: sub-second buffering increments
- Limitations
  - Latency of micro-batch (batch first)
  - Inability to fit windows to naturally occurring events
  - Supports only tumbling/sliding windows
  - No event-time windowing (out-of-order processing)
  - No watermarks support
  - Triggers (when to compute the window): at the end of the window only with Spark

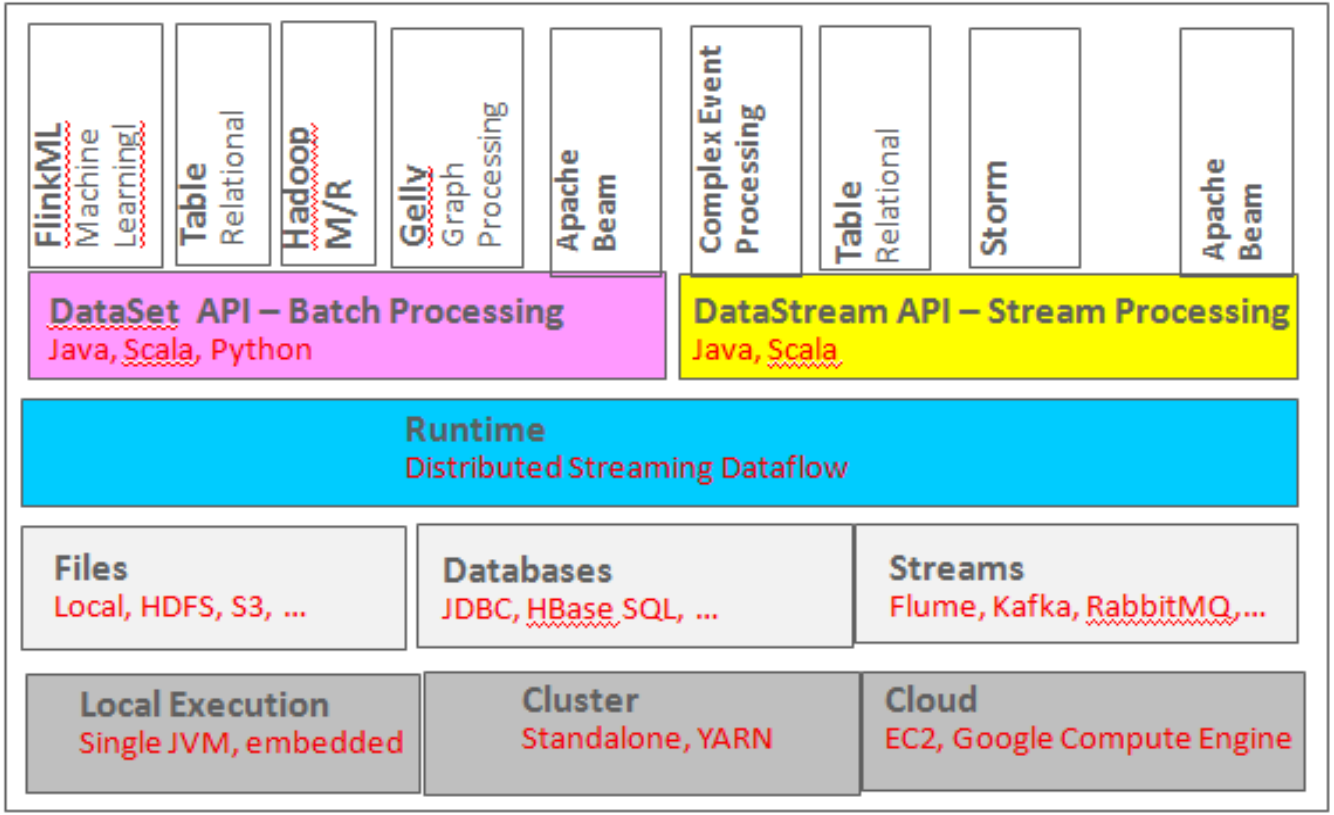
Program Agenda

1. From Big Data to Fast Data
2. Apache Hadoop
3. Apache Spark
4. Apache Flink
5. Why Should Oracle DBA Care

Apache Flink

- 2009: real time, high performance, very low latency streaming
- Single runtime for both streaming (stream-first) and batch processing
- Continuous flow: processes data when it comes
- High throughput fault tolerance
- Correct state upon failure
- Correct time/window semantics
- Supports Event-Time and Out-of-Order Events
- Processing time: pipelined execution is faster
- Own Memory management; no reliance on JVM GC -> no spike

Flink Architecture

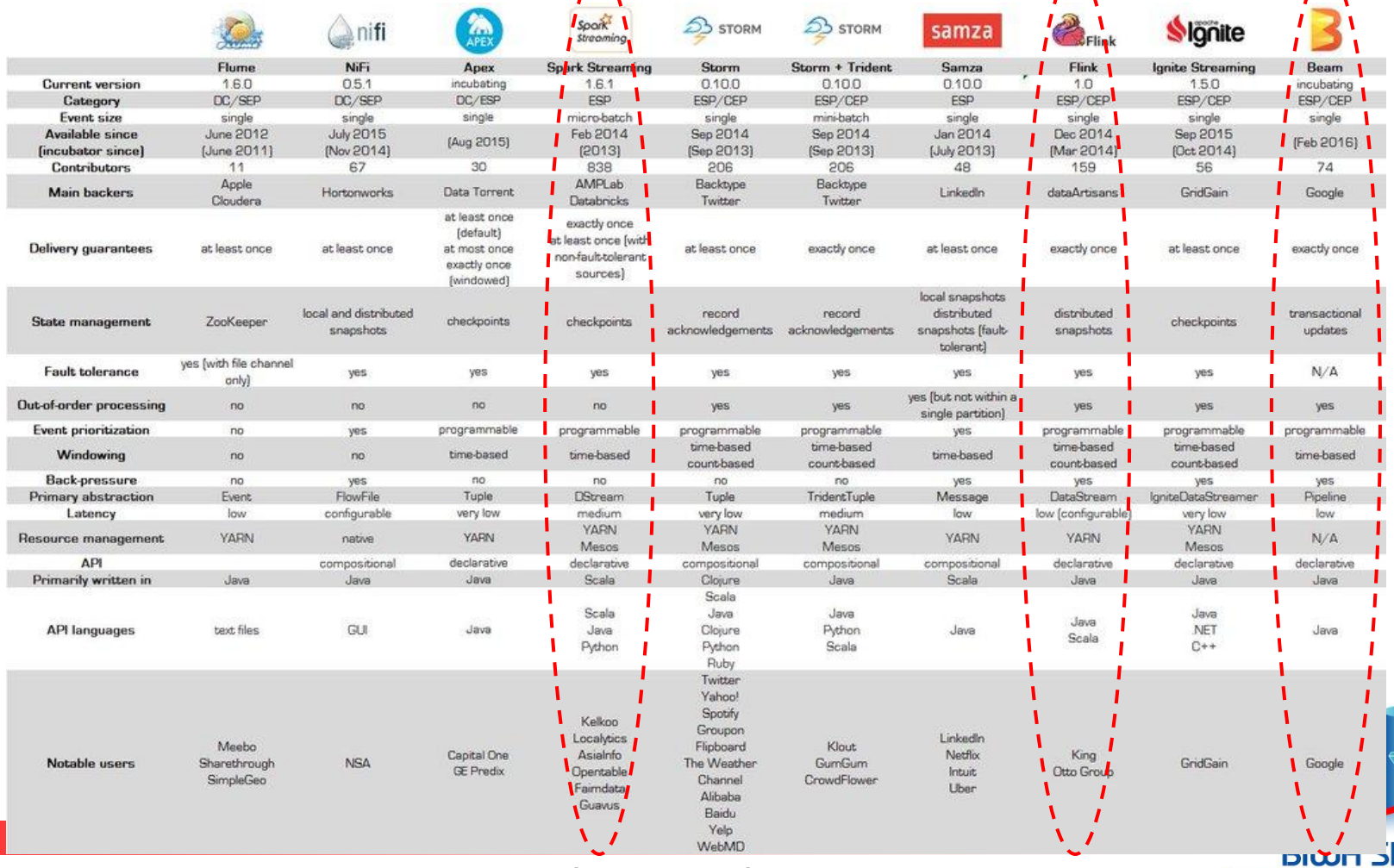


Flink Real World Use Cases

- Advertizing: real-time one/one targetting
- Financial Services: real-time fraud detection
- Retail: smart logistics, real-time monitoring of items and delivery
- Healthcare: smart hospitals, biometrics
- Telecom: real-time service optimization and billing based on location and usage
- Oil and Gaz: real-time monitoring of rigs and pumps

S.No	Category	Apache Spark Streaming	Apache Flink
1	Data processing	Micro Batch	Streaming
2	Windowing	Limited	Flexible
3	Memory Management	Automatic memory	Automatic memory
4	Batch and Stream Processing	Possible	Possible
5	Exactly-once guarantees	Yes	Yes
6	Data Flow	Procedural	Distributed
7	Intermediate Results	No	Yes
8	Processing Time	Comparatively Slow	Faster
9	Event level granularity	No	Yes
10	Iterations	Batch (slow)	Streaming(fast)
11	Fault tolerance	Low	Low
12	Throughput	High	High
13	Optimization	Automatic	Manual
14	Latency	Low	Very low
15	Separation of application logic from fault tolerance	No	Yes

But, this is not yet the end:
here is Apache Beam, ….!

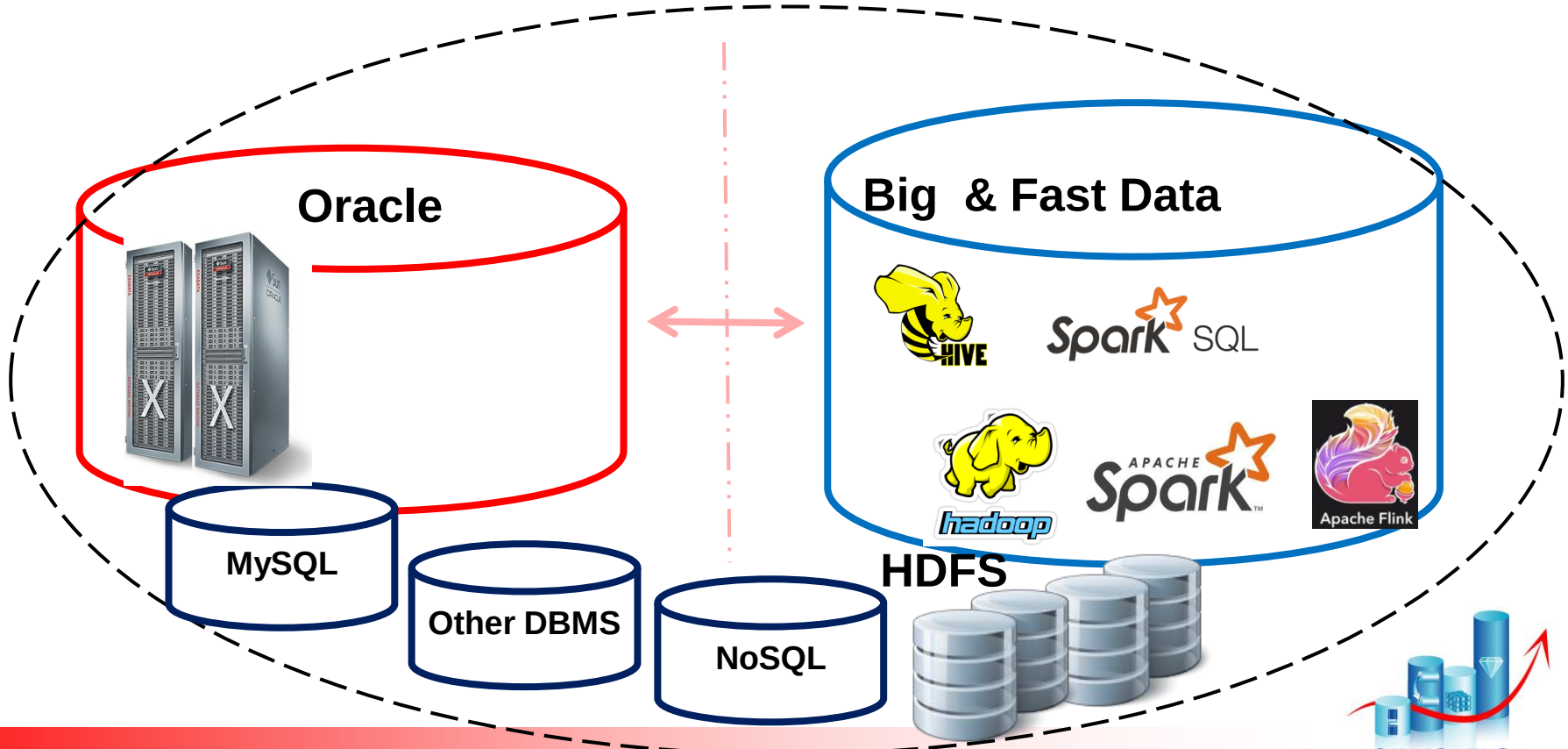


	Flume	NiFi	Apex	Spark Streaming	Storm	Storm + Trident	Samza	Flink	Ignite Streaming	Beam
Current version	1.6.0	0.5.1	incubating	1.6.1	0.10.0	0.10.0	0.10.0	1.0	1.5.0	incubating
Category	DC/SEP	DC/SEP	DC/ESP	ESP	ESP/CEP	ESP/CEP	ESP	ESP/CEP	ESP/CEP	ESP/CEP
Event size	single	single	single	micro-batch	single	mini-batch	single	single	single	single
Available since (incubator since)	June 2012 (June 2011)	July 2015 (Nov 2014)	(Aug 2015)	Feb 2014 (2013)	Sep 2014 (Sep 2013)	Sep 2014 (Sep 2013)	Jan 2014 (July 2013)	Dec 2014 (Mar 2014)	Sep 2015 (Oct 2014)	(Feb 2016)
Contributors	11	67	30	838	206	206	48	159	56	74
Main backers	Apple Cloudera	Hortonworks	Data Torrent	AMPLab Databricks	Backtype Twitter	Backtype Twitter	LinkedIn	dataArtisans	GridGain	Google
Delivery guarantees	at least once	at least once	at least once (default) at most once exactly once (windowed)	exactly once at least once (with non-fault-tolerant sources)	at least once	exactly once	at least once	exactly once	at least once	exactly once
State management	ZooKeeper	local and distributed snapshots	checkpoints	checkpoints	record acknowledgements	record acknowledgements	local snapshots distributed snapshots (fault-tolerant)	distributed snapshots	checkpoints	transactional updates
Fault tolerance	yes (with file channel only)	yes	yes	yes	yes	yes	yes	yes	yes	N/A
Out-of-order processing	no	no	no	no	yes	yes	yes (but not within a single partition)	yes	yes	yes
Event prioritization	no	yes	programmable	programmable	programmable	programmable	yes	programmable	programmable	programmable
Windowing	no	no	time-based	time-based	time-based count-based	time-based count-based	time-based	time-based count-based	time-based count-based	time-based
Back-pressure	no	yes	no	no	no	no	yes	yes	yes	yes
Primary abstraction	Event	FlowFile	Tuple	DStream	Tuple	TridentTuple	Message	DataStream	IgniteDataStreamer	Pipeline
Latency	low	configurable	very low	medium	very low	medium	low	low (configurable)	very low	low
Resource management	YARN	native	YARN	YARN Mesos	YARN Mesos	YARN Mesos	YARN	YARN	YARN Mesos	N/A
API		compositional	declarative	declarative	compositional	compositional	compositional	declarative	declarative	declarative
Primarily written in	Java	Java	Java	Scala	Clojure	Java	Scala	Java	Java	Java
API languages	text files	GUI	Java	Scala Java Python	Scala Java Clojure Python Ruby	Java Python Scala	Java	Java Scala	Java .NET C++	Java
Notable users	Meebo Sharethrough SimpleGeo	NSA	Capital One GE Predix	Kelkoo Localytics AsiaInfo Opentable Fairmdata Guavus	Twitter Yahoo! Spotify Groupon Flipboard The Weather Channel Alibaba Baidu Yelp WebMD	Klout GumGum CrowdFlower	LinkedIn Netflix Intuit Uber	King Otto Group	GridGain	Google

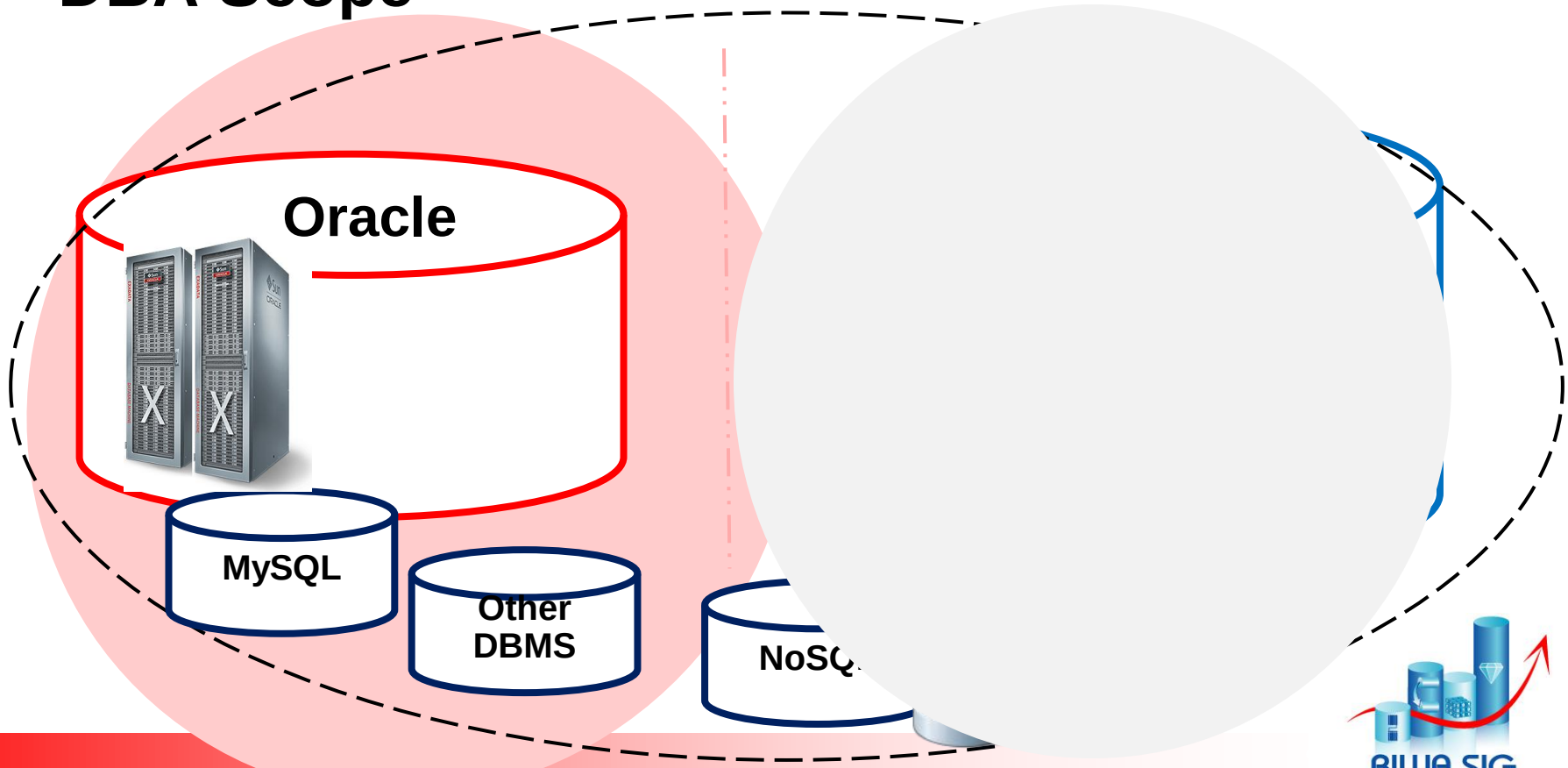
Program Agenda

1. From Big Data to Fast Data
2. Apache Hadoop
3. Apache Spark
4. Apache Flink
5. Why Should DBA Care

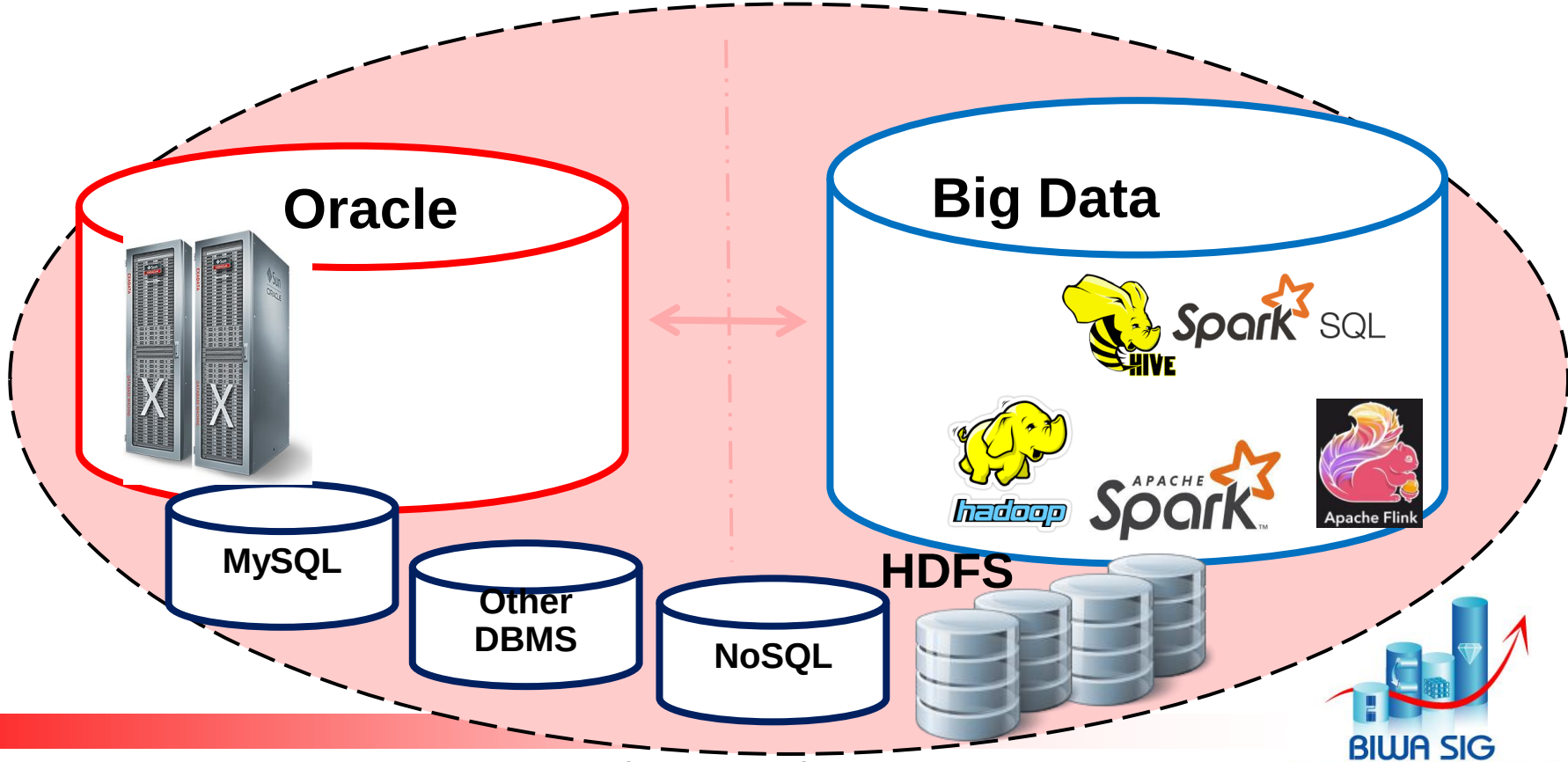
Your Data Center: today or tomorrow



DBA Scope



Data Architect and Chief Data Officer Scope



Which Big Data Job to Aim For? (1/2)

- Big Data Visualizer
  Make data visually understandable by senior management, sales & marketing,

- Data Scientist, Big Data Researcher
  Integrate multiple systems and data sets.
  Design the algorithms & products.
  Turn statistics and data into productive information.

- Big Data Developer, ETL Developer
  Implement the design from data scientists and solutions architects.
  Develop, maintain, test and evaluate big data solutions within organizations.

Which Big Data Jobs to Aim For (2/2)

- Big Data Administrator, Data Architect

  Manages the Big Data Clusters
  Monitors data and network traffic, prevents glitches
  Integrates, centralizes, protect s and maintains data sources
  Grants and revokes permissions to various clients and nodes.

- Chief Data Officer (beyond Big Data)

  Responsible for the overall data strategy within an organization
  Accountable for whatever data is collected, stored, shared, sold or analyzed as well as how the data is collected, stored, shared, sold or analyze
  Ensures that the data is implemented correctly, securely and comply with customers' privacy, data privacy, government and ethical policies
  Defines company standards and policies for data operation, data accountability, and data quality

Roadmap to Big Data Architect and Chief Data Officer

- Get your hands on Big Data platform, Cloud services or VMs
  e.g., Oracle BDALite Vbox, Oracle Big Data Cloud Services, Oracle BDA
- Leverage your Oracle background and notions: clusters, nodes, Big Data SQL, Big Data Connectors (e.g., Oracle Datasource for Hadoop)
- Get familiar with Big Data databases & storages: HDFS, NoSQL, DBMSes
- Get familiar with key Big Data Frameworks: Hadoop, Spark, Flink and streaming frameworks Kafka, Storm, and integration with Oracle
- Get familiar with Big Data tools and programming : Oracle SQL, Hive SQL, Spark SQL, visualization tools, R, Java, Scala
- Read, Practice and Get involved in Big Data projects
-

Key Takeaways

- Big Data is growing exponentially
- This is the era of Fast Data requiring new processing models
- Hadoop is good for some use cases but cannot handle streaming data
- Spark brings in-memory processing and data abstraction (RDD, etc) and allows real-time processing of streaming data however its micro batch architecture incurs high latency
- Flink brings low latency and promise to address Spark limitations
- DBA should embrace Big Data frameworks and expand their skills and coverage within the data center or in the Cloud.

Resources

- Road Map for Careers in Big Data

http://www.slideshare.net/MichTalebzadeh1/road-map-for-careers-in-big-data-65157691

https://www.linkedin.com/pulse/possible-next-career-move-relational-dbas-mich-talebzadeh-ph-d-?published=t

- An Enterprise Architect's Guide to Big Data

http://www.oracle.com/technetwork/topics/entarch/articles/oea-big-data-guide-1522052.pdf

BIWA SIG
Better Information Better Results
www.oraclebiwa.org