



Cloudera and Open Source with Kudu, Spark

Graham Pymm | System Engineer @ Cloudera



Agenda

- Open Source Software and Open Standards
- Cloudera's Involvement in the Apache Hadoop Ecosystem
- Cloudera's Open Source Software Platform
- Apache Kudu (incubating)
- Apache Spark

Benefits of Open Source Software and Open Standards

Top 3 Reasons Open Source is Good for Business

1

Free Evaluation

Install, test, inspect, and evaluate open source code in perpetuity, with no financial obligation.

2

Freedom from Lock-in

Use open source software in production without paying royalties or for support.

3

Scalable Innovation

The collective work of a global, passionate community keeps the code base evolving.

What is the ASF?



“The Apache Software Foundation, a US 501(c)(3) non-profit corporation, provides organizational, legal, and financial support for a broad range of over 150 open source software projects.”

Most (not all) of the open source Hadoop platform is collaboratively developed within the ASF by a diverse community (including Cloudera employees), with all participants starting with equal status and earning more influence as they contribute.

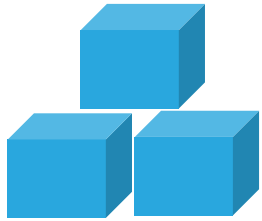
(See Appendix Slide 29 for more details.)

What the ASF is Not

- A guarantor of production-ready code
- A guarantor of predictable, regular releases
- A guarantor of community diversity (although it's encouraged)
- The arbiter of “winners” (standards) and “losers”
- The exclusive approach to open source governance
- An SLA-constrained support resource for production deployments

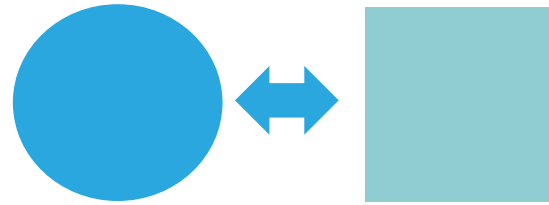
Open Standards and Open Source are Equally Important

A “de facto standard” is an open source component that, because of widespread grass-roots adoption, is supported by multiple vendors.



Long-Term Architecture

Only open standards get continuing, long-term investment from across the ecosystem.



Avoidance of Lock-in

Thanks to multi-vendor support, open standards give customers choices and prevent lock-in.



Ecosystem Compatibility

Open standards attract more connectors/certifications due to their broad adoption.

Cloudera and the Apache Hadoop ecosystem

How Customers Benefit from Cloudera's Open Source Credentials



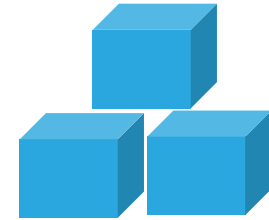
Support Across the Platform

With ~100 committer seats across all projects (not just Core Hadoop), Cloudera fixes all customer issues upstream/permanently.



Impact on the Roadmap

With the most PMC members and project founders, Cloudera shapes the platform roadmap for your needs through its contributions.

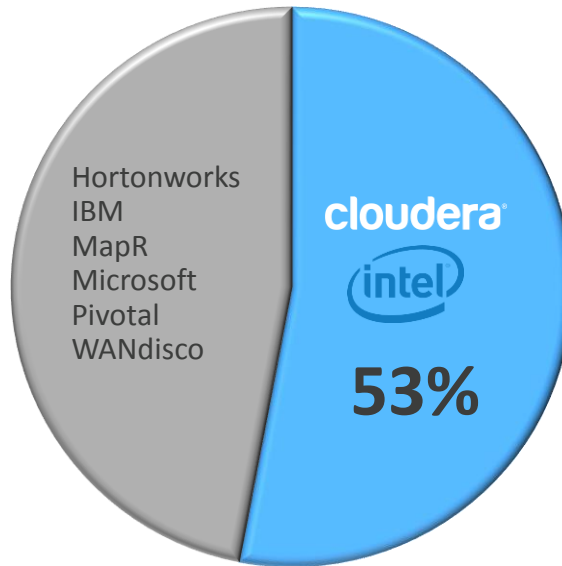


Mainstream Architecture

With its commitment to open standards and its curation of them, Cloudera ensures your architectural investments are safe.

Support for the Entire Platform (Not Just the Core)

- Nearly 100 committer seats on active projects* means faster fixes across the ecosystem for customers!
- Collectively, Cloudera and Intel committers resolve more than half of all tickets (JIRAs) on multi-vendor-supported projects that are assigned to platform vendor employees:



Source: Apache JIRA
Jan. 2012 – Nov. 2015

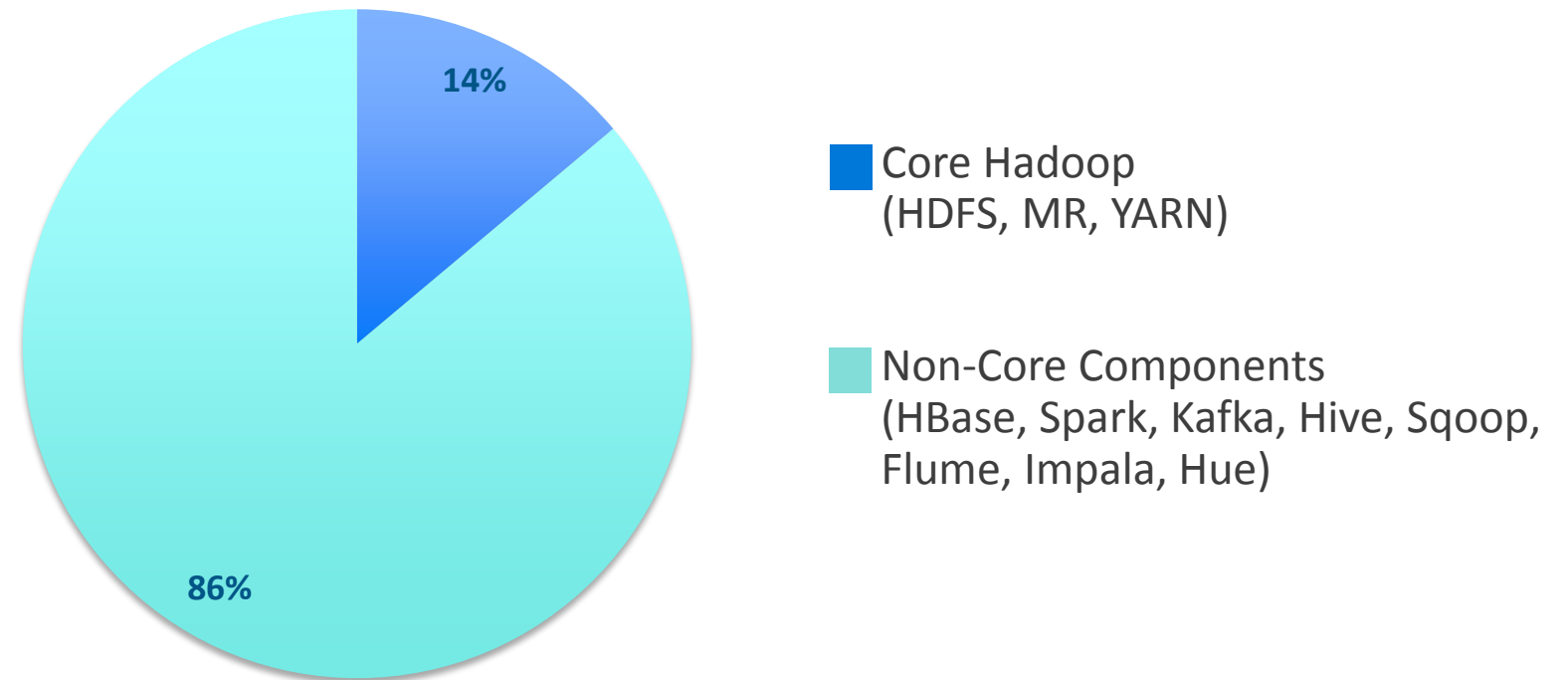
Projects Included:

Accumulo	Oozie
Avro	Parquet
Bigtop	Pig
Crunch	Solr
Flume	Spark
Hadoop Core	Sqoop
HBase	Tez
Hive	ZooKeeper
Kafka	

* “Committer” = A developer who has earned the privilege to commit patches

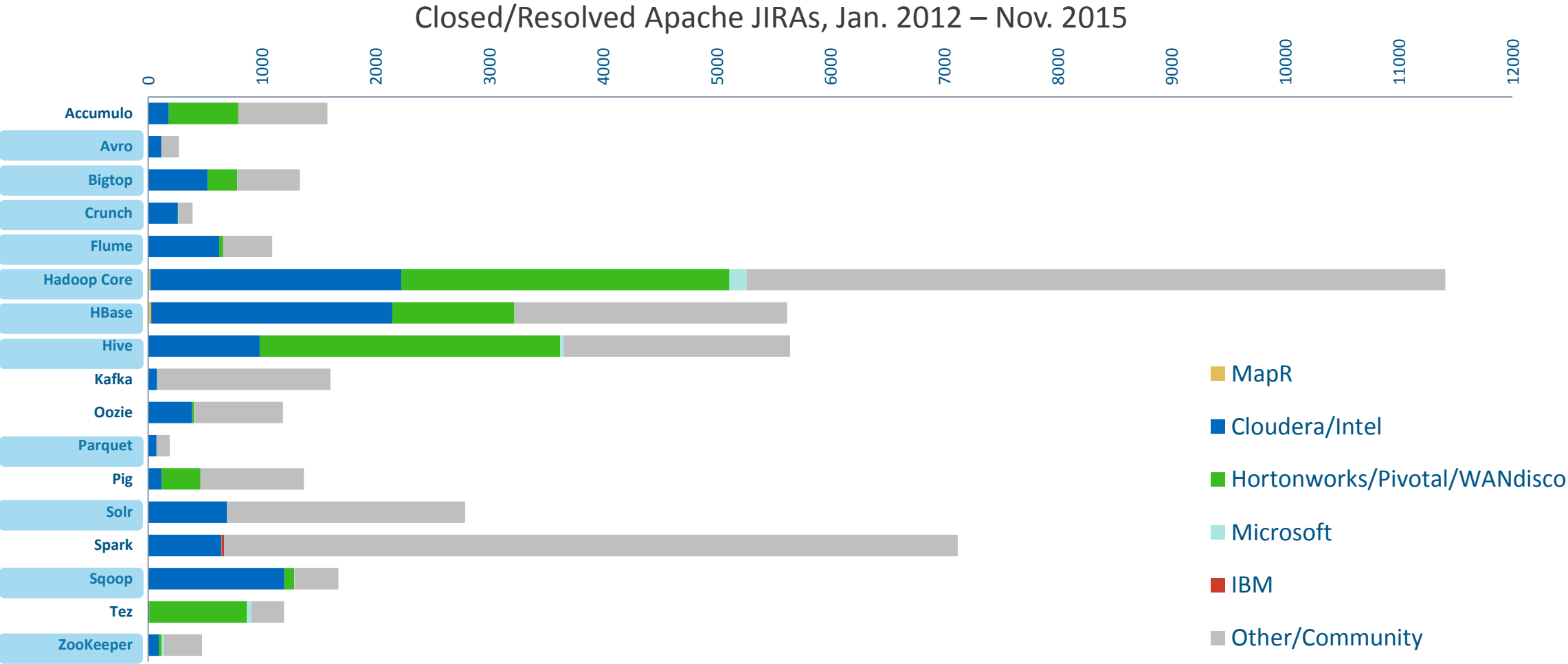
Most Innovation is Happening Outside the Core!

Commits in Past Year



Cloudera is the Main Founder, Shipper, and Supporter of Standards

(Components shipped by 1+ vendors)



Projects founded/co-founded by current/former employees are highlighted

Shaping (& Shipping) the Roadmap

- Cloudera has contributed, and shipped, more major open source features than any other platform vendor.

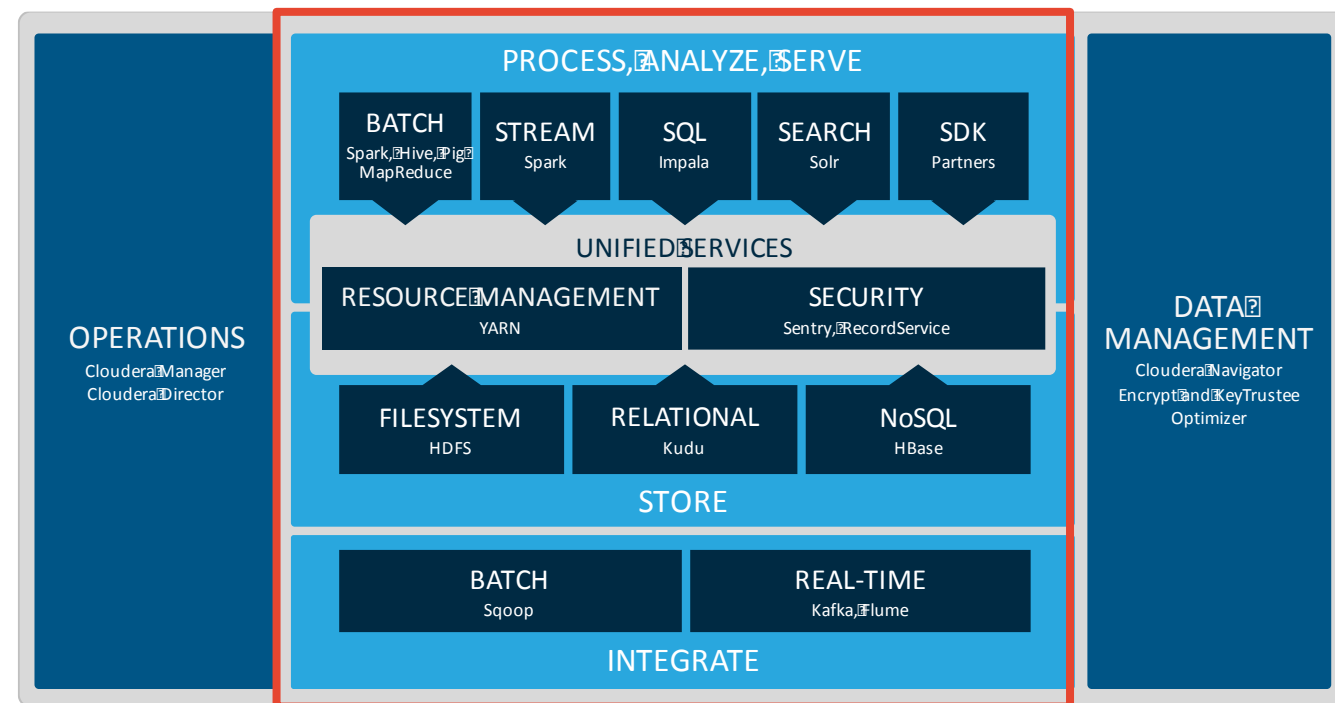
Examples:

HDFS NameNode HA
HDFS secure short-circuit local reads
HttpFS
Network encryption
HBase snapshots
Hive authentication (HiveServer2)
Solr+Hadoop Integration
HDFS caching
Transparent data encryption for HDFS
WITH GRANT OPTION for Hive and Impala
Kafka + Flume integration

Cloudera's Open Source Platform

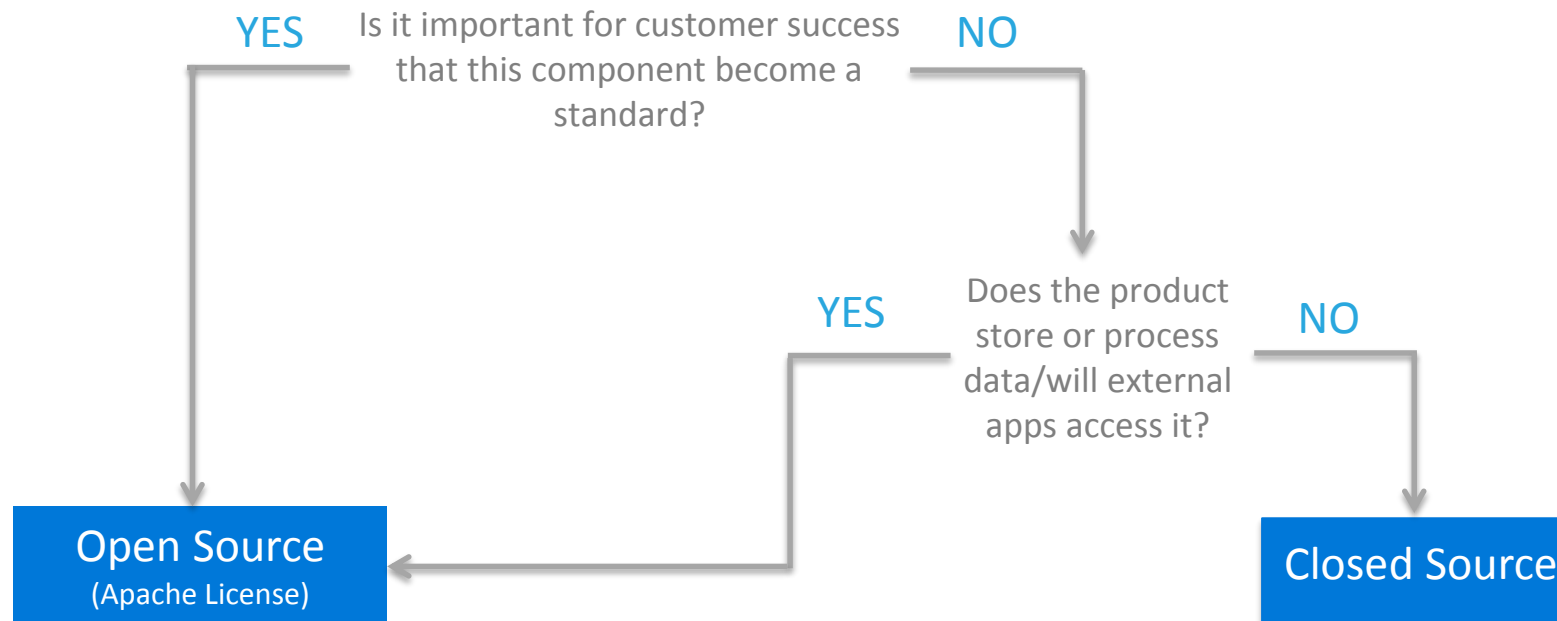
Cloudera's Platform: What Is/Is Not OSS

- Cloudera's platform contains the full ecosystem of standard Hadoop components (CDH) plus closed-source, vendor-specific software (on-prem or in cloud) for:
 - Cluster/system/encryption key management
 - Data management/governance
- The vendor-specific modules do not store or process data and are separately installable.



Inside the Open/Closed Source Decision Process

How does Cloudera decide which software should be open or closed source?



CDH: It's 100% Apache Hadoop, And More

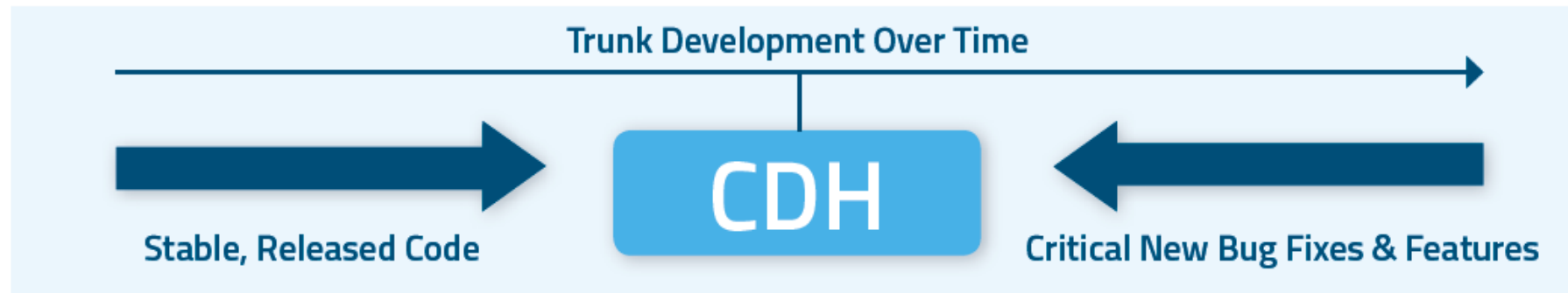
The latest stable releases
of open source
components in the
ecosystem



Curated bug fixes and features
(backports) from project
trunks that are not yet
available in an upstream
release



A combination of stable,
released code and new
features and fixes, all tested
and certified

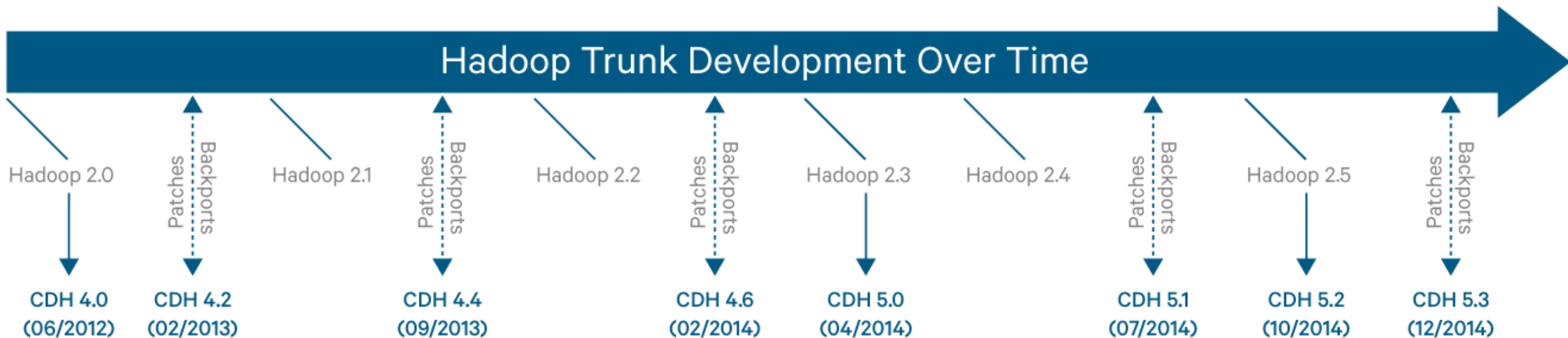


Backport: A code patch originally designed for one release but then retrofitted to another

Trunk: The project repository for all source code; releases are “branches” (subsets) of the trunk

Upstream: Shorthand for “the trunk” (“committed upstream” = “committed to trunk”)

Detailed View of the CDH Life Cycle



(Note: Some releases -- e.g. point releases -- omitted for clarity)

Customer Benefits from the CDH Life Cycle

- They can confidently access **stable, certified Apache code** after extensive testing and integration work.
- They can count on their issues being **fixed permanently** upstream.
- They can **access the most critical new upstream bug fixes and innovations** at a regular, predictable cadence.
- **Interoperability** is ensured across releases, as well as with upstream project trunks (which ensures application portability).
- Upgrades are **much easier**.

Key Takeaways

1

The Apache License, which all of Cloudera's open source software carries, has multiple business benefits.

2

Open standards are as important as open source for preventing lock-in and ensuring a safe investment in long-term, mainstream architecture.

3

With its deep/wide involvement in the open source community, Cloudera can support the entire platform, as well as impact/ship the roadmap.

4

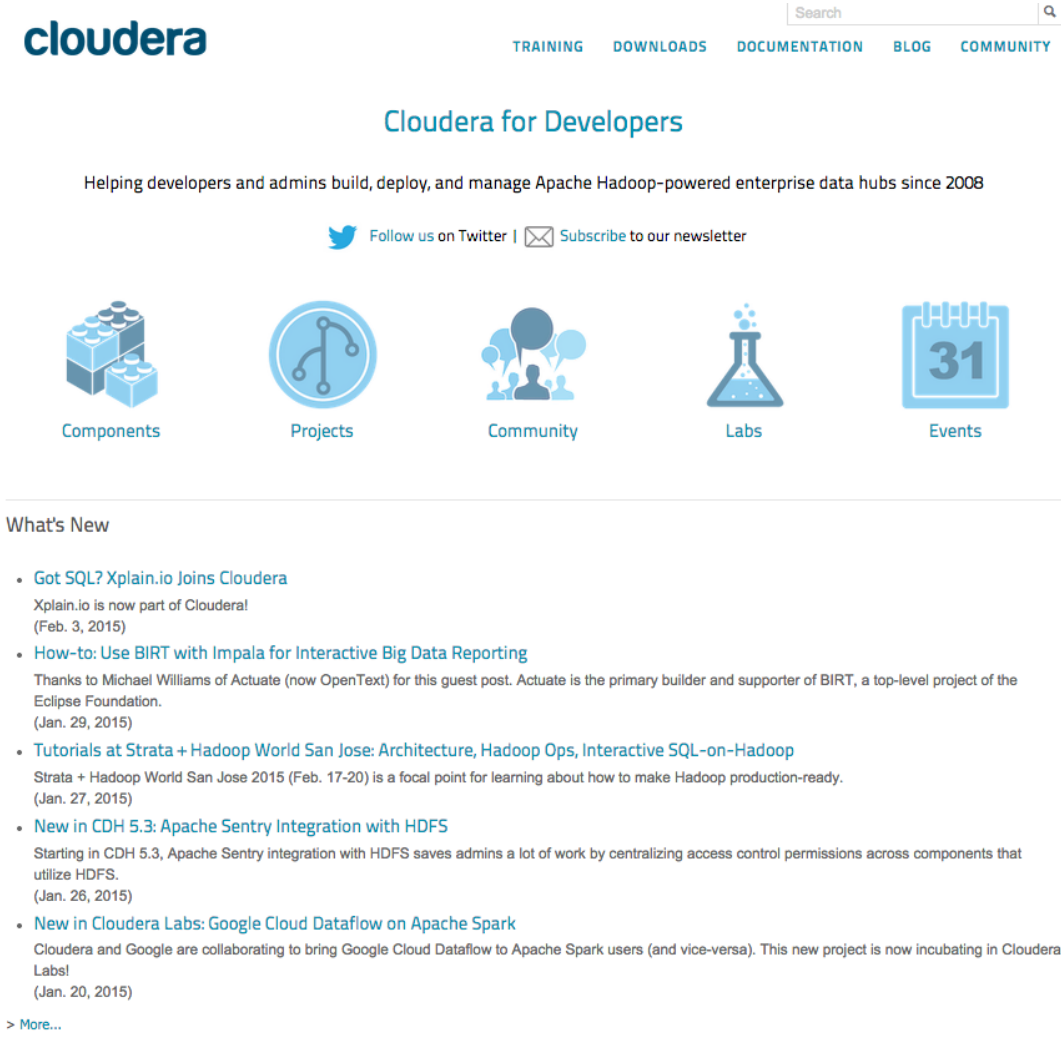
Cloudera's open source platform contains the same code as Apache releases, along with added stability, functionality, and predictability.

Cloudera's Developer Offerings

cloudera.com/developer

- Downloads and trials
 - Cloudera Live
 - QuickStart VM & Docker Image
 - CDH
- Cloudera Community
- Conferences and Meetups
- Cloudera Labs
- Cloudera Developer Program
- Cloudera Developer Newsletter
- Cloudera Engineering Blog

cloudera



The screenshot shows the Cloudera Developer website. At the top is the Cloudera logo and a navigation bar with links for TRAINING, DOWNLOADS, DOCUMENTATION, BLOG, and COMMUNITY. A search bar is located on the right. Below the navigation bar is the heading "Cloudera for Developers" and a subheading "Helping developers and admins build, deploy, and manage Apache Hadoop-powered enterprise data hubs since 2008". There are social media links for Twitter and a newsletter subscription. A row of five icons represents different offerings: Components (LEGO blocks), Projects (a circular diagram), Community (a group of people), Labs (a flask with bubbles), and Events (a calendar showing the number 31). Below this is a "What's New" section with a list of recent updates, including "Got SQL? Xplain.io Joins Cloudera", "How-to: Use BIRT with Impala for Interactive Big Data Reporting", "Tutorials at Strata + Hadoop World San Jose: Architecture, Hadoop Ops, Interactive SQL-on-Hadoop", "New in CDH 5.3: Apache Sentry Integration with HDFS", and "New in Cloudera Labs: Google Cloud Dataflow on Apache Spark". A link for "> More..." is at the bottom of the list.

cloudera


TRAINING DOWNLOADS DOCUMENTATION BLOG COMMUNITY

Search


Cloudera for Developers

Helping developers and admins build, deploy, and manage Apache Hadoop-powered enterprise data hubs since 2008


Follow us on Twitter | Subscribe to our newsletter




Components




Projects



Community



Labs



Events

What's New

- [Got SQL? Xplain.io Joins Cloudera](#)
Xplain.io is now part of Cloudera!
(Feb. 3, 2015)
- [How-to: Use BIRT with Impala for Interactive Big Data Reporting](#)
Thanks to Michael Williams of Actuate (now OpenText) for this guest post. Actuate is the primary builder and supporter of BIRT, a top-level project of the Eclipse Foundation.
(Jan. 29, 2015)
- [Tutorials at Strata + Hadoop World San Jose: Architecture, Hadoop Ops, Interactive SQL-on-Hadoop](#)
Strata + Hadoop World San Jose 2015 (Feb. 17-20) is a focal point for learning about how to make Hadoop production-ready.
(Jan. 27, 2015)
- [New in CDH 5.3: Apache Sentry Integration with HDFS](#)
Starting in CDH 5.3, Apache Sentry integration with HDFS saves admins a lot of work by centralizing access control permissions across components that utilize HDFS.
(Jan. 26, 2015)
- [New in Cloudera Labs: Google Cloud Dataflow on Apache Spark](#)
Cloudera and Google are collaborating to bring Google Cloud Dataflow to Apache Spark users (and vice-versa). This new project is now incubating in Cloudera Labs!
(Jan. 20, 2015)

> More...

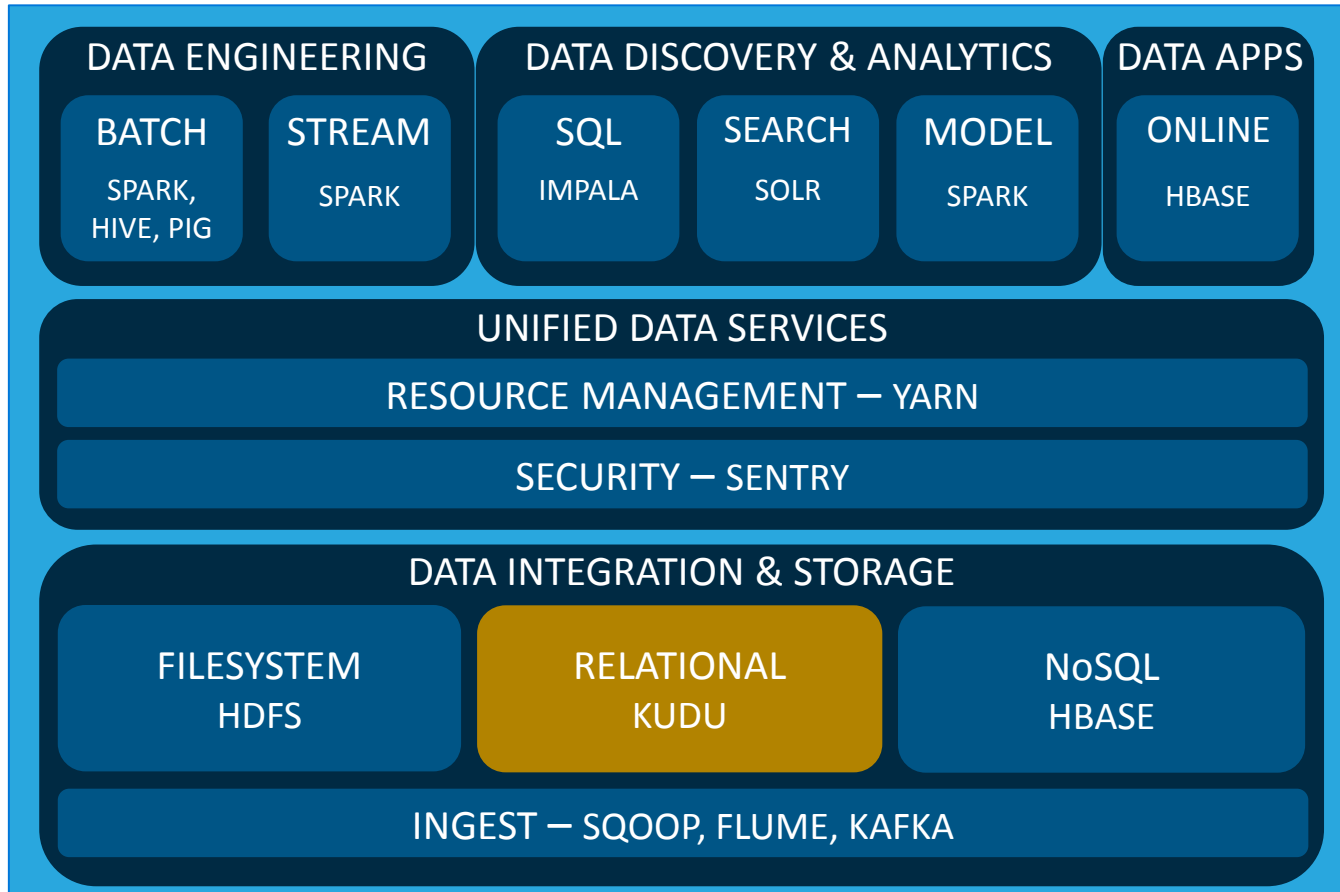
Apache Impala (incubating)

Apache Kudu (incubating)

Beta

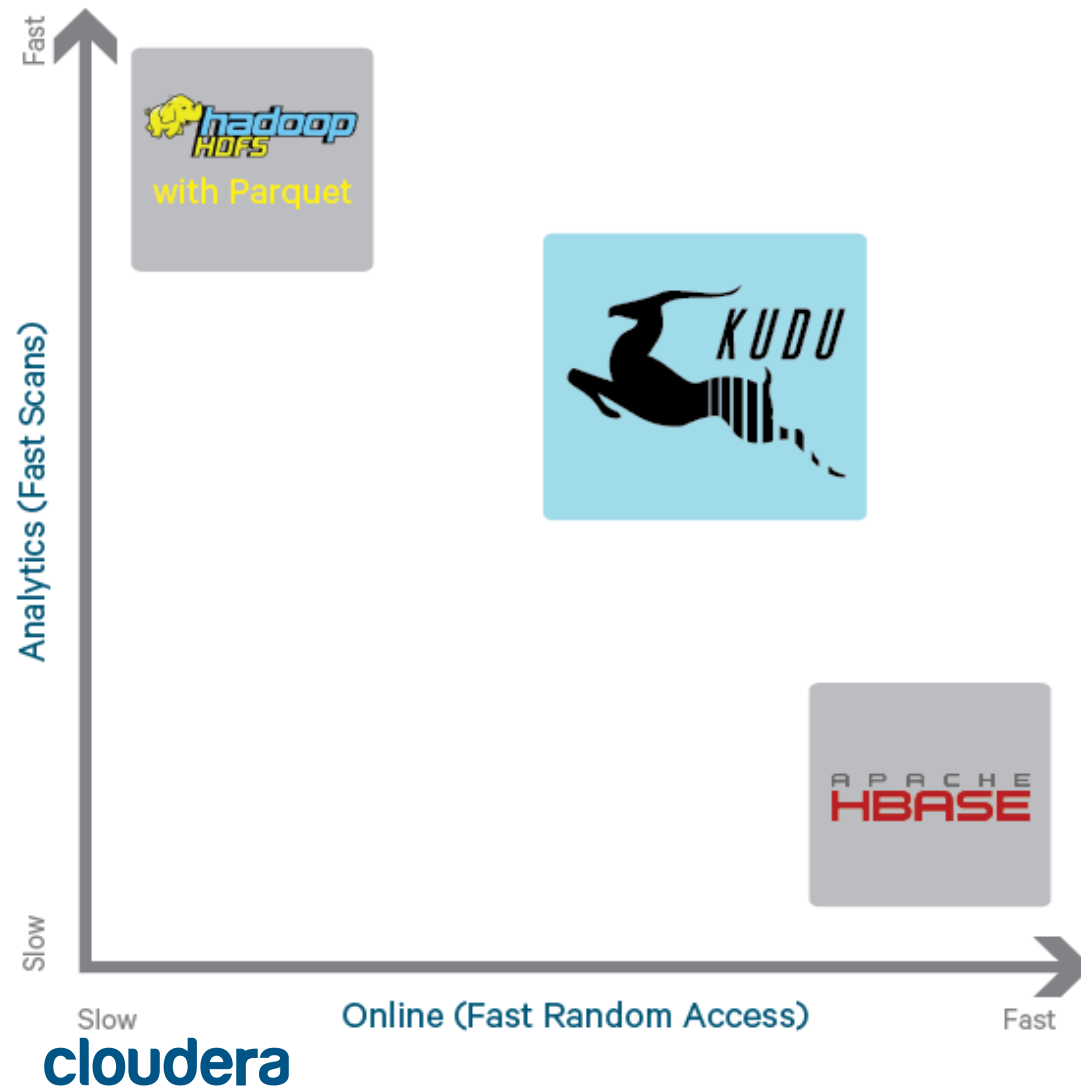
Kudu

Storage for Fast Analytics on Fast Data



- New updating column store for Hadoop
 - Simplifies the architecture for building analytic applications on changing data
 - Designed for fast analytic performance
 - Natively integrated with Hadoop
- Apache-licensed open source (intent to donate to ASF)
- Beta now available

Kudu Design Goals



- **High throughput** for big scans (columnar storage and replication)
Goal: Within 2x of Parquet
- **Low-latency** for short accesses (primary key indexes and quorum design)
Goal: 1ms read/write on SSD
- **Database-like** semantics (initially single-row ACID)
- **Relational data model**
 - SQL query
 - “NoSQL” style scan/insert/update (Java client)

Kudu Use Cases

Kudu is best for use cases requiring a *simultaneous* combination of sequential and random reads and writes

- **Time Series**

- Examples: Stream market data; fraud detection & prevention; risk monitoring
- Workload: Insert, updates, scans, lookups

- **Machine Data Analytics**

- Examples: Network threat detection
- Workload: Inserts, scans, lookups

- **Online Reporting**

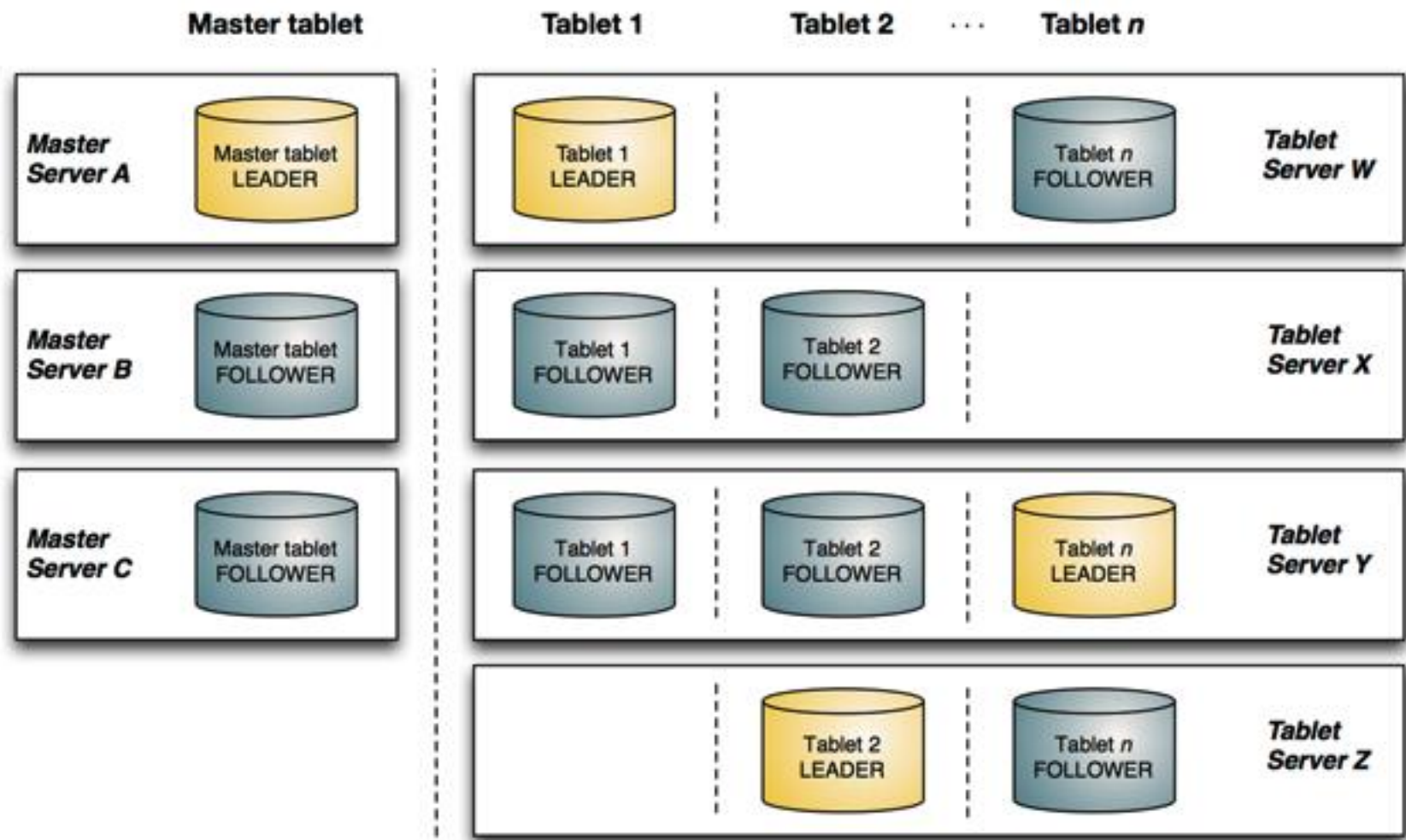
- Examples: ODS
- Workload: Inserts, updates, scans, lookups

Kudu Basic Design

- Basic Construct: Tables
 - Tables broken down into Tablets (roughly equivalent to partitions)
- Typed storage
- Maintains consistency via:
 - Multi-Version Concurrency Control (MVCC)
 - Raft Consensus¹
- Architecture supports geographically disparate, active/active systems
 - Not the initial design goal

¹<http://thesecretlivesofdata.com/raft/>

Kudu Architecture



What Kudu is *NOT*

- Not a SQL interface itself
 - It's just the storage layer
- Not an application that runs on HDFS
 - It's an alternative, native Hadoop storage engine
- Not a replacement for HDFS or HBase
 - Select the right storage for the right use case
 - Cloudera will continue to support and invest in all three

Kudu Trade-Offs

- Random updates will be slower
 - HBase model allows random updates without incurring a disk seek
 - Kudu requires a key lookup before update, Bloom lookup before insert
- Single-row reads may be slower
 - Columnar design is optimized for scans
 - **Future:** may introduce “column groups” for applications where single-row access is more important

What Impala SQL can I run on Kudu?

Single Tablet
(illustration
purposes only)

Create table and populate from HDFS

```
CREATE TABLE passenger_data
TBLPROPERTIES
('storage_handler' = 'com.cloudera.kudu.hive.KuduStorageHandler',
'kudu.table_name' = 'passenger_data', 'kudu.master_addresses' =
'127.0.0.1', 'kudu.key_columns' = 'id')
AS
SELECT * FROM passenger_data_raw;
```

13,901
records in
1,180 ms

UPDATE records

```
UPDATE passenger_data
SET    operating_airline = "United Airlines"
WHERE  operating_airline LIKE "United Airlines - Pre%";
```

2,500
records in
130 ms

INSERT records (from HDFS)

```
INSERT INTO passenger_data
SELECT new_id + id, [rest of cols...]
FROM    passenger_data_raw t1
CROSS JOIN
        (SELECT max(id) as new_id from passenger_data_raw) dt1;
```

13,901
records in
890 ms

DELETE records

```
DELETE FROM passenger_data;
```

27,802
records in
320 ms

Kudu Schema

- Kudu has a structured data model similar to DBMS
- Requires non-NULL, unique primary key column(s)
- Data distribution:
 - Range partitioning (default) – ordered by column values, each tablet gets a split of contiguous values e.g. a-l, m-z
 - Useful where insert/update and query is uniform across column(s)
 - Hash bucketing – rows assigned to buckets according to column(s) hash.
 - Useful to even out potentially skewed writes
- WARNING: you have to pre-set the distribution, it cannot be changed after creation

Kudu Schema

- DDL changes permissible:
 - Rename the table
 - Rename, add, or drop columns
 - Rename (but not drop) primary key columns
- Supports different per-column encodings:
 - Plain, Run Length, Dictionary, and Pre-fix
- Supports different per-column compression:
 - [LZ4](#), Snappy or [ZLib](#)

Current Status

- ✓ Completed all components core to the architecture
- ✓ Java and C++ API
- ✓ Early Impala integration done, working on Spark integration
- ✓ Support for SSDs and spinning disk
- ✓ Fault recovery
- ✓ Open to beta customers

Limitations

Target GA

- No authentication, No RBAC via Sentry
- No integration with Flume, Kafka, Sqoop
- Initial integration with Spark – see <https://github.com/tmalaska/SparkOnKudu>
- Impala scans are READ LATEST – cannot refer to a TIMESTAMP snapshot yet
- Multiple master not yet supported

Beyond GA

- Multi-table write transactions not supported
- Geographically dispersed tablets

Resources

Join the community

<http://getkudu.io>

Download the Beta

cloudera.com/downloads

Read the Whitepaper

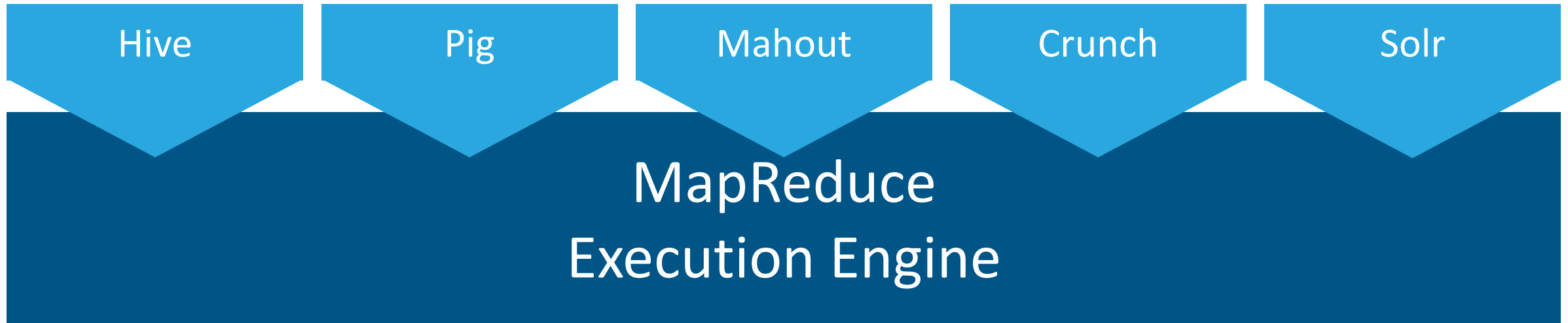
getkudu.io/kudu.pdf



Apache Spark

One Platform Initiative

MapReduce: A great tool for its day



The original scalable, general, processing engine of Hadoop ecosystem

- Useful across diverse problem domains
- Fueled initial ecosystem explosion

Enter Apache Spark

General purpose computational framework that
substantially improves on MapReduce

Key Properties:

- Leverages distributed memory
- Full Directed Graph expressions for data parallel computations
- Simpler developer experience

Yet Retains:

- Linear scalability
- Fault-tolerance
- Data locality-based computations

Apache Spark

Flexible, in-memory data processing for Hadoop

Easier

- Rich APIs for Scala, Java, and Python
- Interactive shell

More Powerful

- APIs for different types of workloads:
 - Batch
 - Streaming
 - Machine Learning
 - Graph

Faster

- In-Memory processing and caching

Easy Development

High Productivity Language Support

Python

```
lines = sc.textFile(...)
lines.filter(lambda s: "ERROR" in s).count()
```

Scala

```
val lines = sc.textFile(...)
lines.filter(s => s.contains("ERROR")).count()
```

Java

```
JavaRDD<String> lines = sc.textFile(...);
lines.filter(new Function<String, Boolean>() {
    Boolean call(String s) {
        return s.contains("error");
    }
}).count();
```

- Native support for multiple languages with identical APIs
 - Scala, Java, Python
- Use of closures, iterations, and other modern language constructs to minimize code
 - 2-5x less code

Easy Development

Use Interactively

```
$ ./bin/spark-shell --master local[*]
...
Welcome to

  ____      __
 / ___/____/ /_  ___
/ /  /_  __/ __/ __ \
/ ___/ _  / /_ / ___/
/ /  /_  / /_ / ___/
/___/_/___/_/___/___/

version 1.5.0-SNAPSHOT

Using Scala version 2.10.4
(Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_51)
Type in expressions to have them evaluated.
Type :help for more information.
...

scala> val words = sc.textFile("file:/usr/share/dict/words")
...
words: org.apache.spark.rdd.RDD[String] =
  MapPartitionsRDD[1] at textFile at <console>:21

scala> words.count
...
res0: Long = 235886

scala>
```

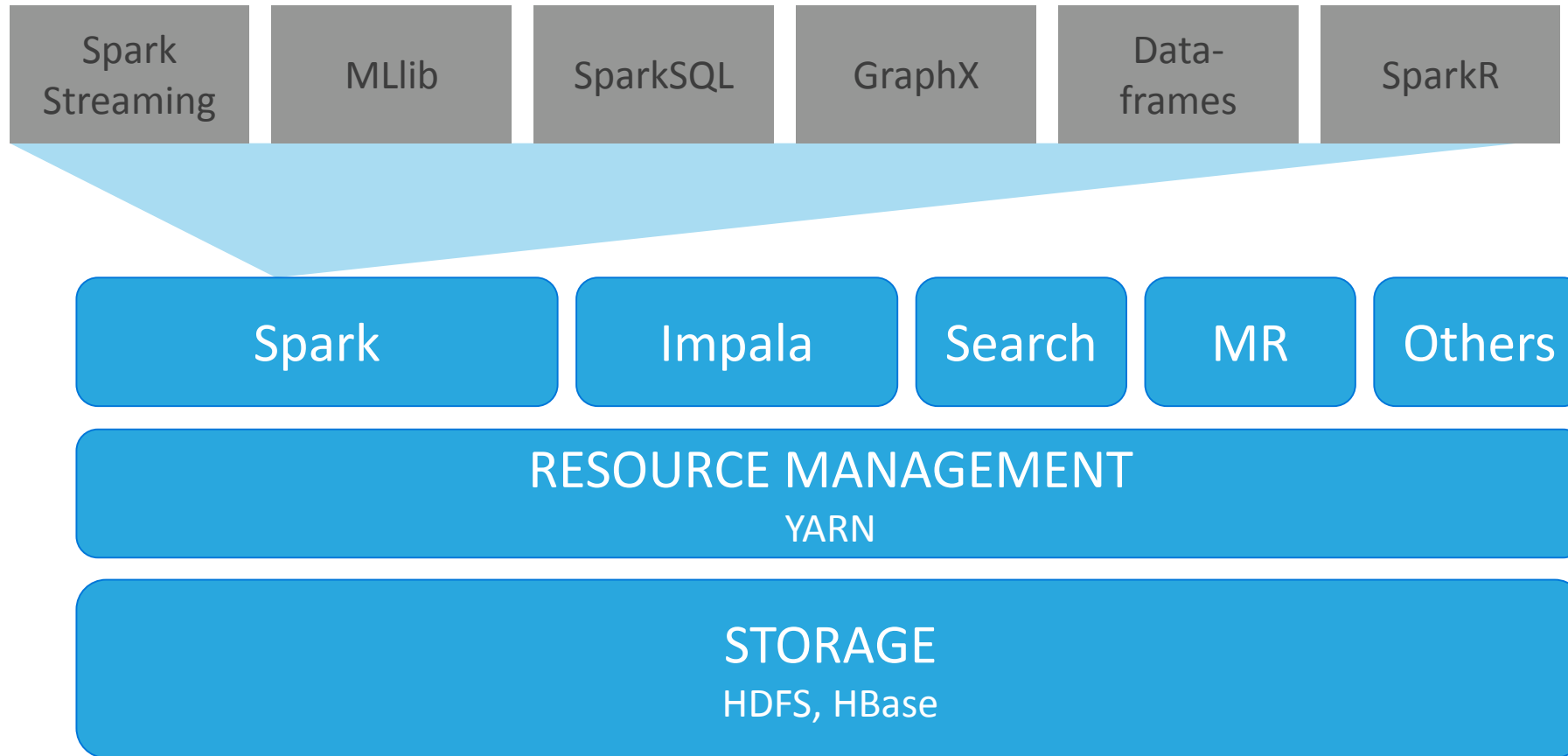
- Interactive exploration of data for data scientists
 - No need to develop “applications”
- Developers can prototype application on live system

Spark Takes Advantage of Memory

Resilient Distributed Datasets (RDD)

- Memory caching layer that stores data in a distributed, *fault-tolerant* cache
 - Can fall back to disk when data-set does not fit in memory
- Created by parallel transformations on data in stable storage
 - Provides fault-tolerance through concept of *lineage*

The Spark Ecosystem & Hadoop



Cloudera is Driving the Spark Movement



Identified Spark's early potential

Added Spark on YARN integration



Launches first Spark training

Added security integration

Driving effort to further performance, usability, and enterprise-readiness

2013

2014

2015

2016

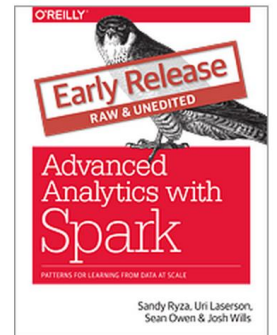


Ships and Supports Spark with CDH 4.4



Announces initiative to make Spark the standard execution engine

Cloudera engineers publish O'Reilly Spark book



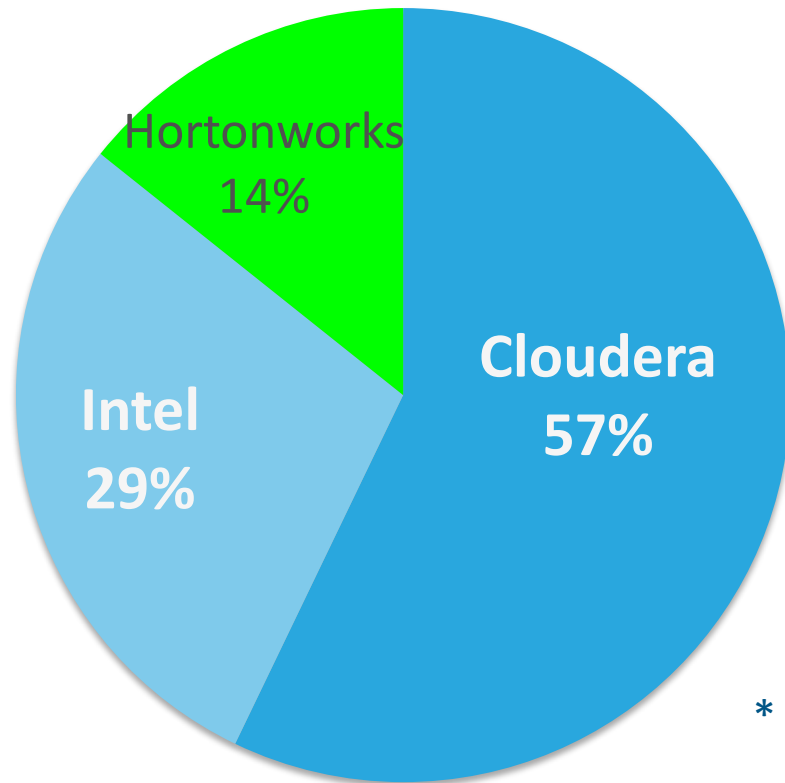
cloudera

Spark at Cloudera

- Cloudera was the **first** Hadoop vendor to ship and support Spark
- Spark is a **fully integrated** part of Cloudera's platform
 - Shared data, metadata, resource management, administration, security, and governance
 - Complements specialized analytic tools for comprehensive big data platform
- Cloudera is the **first** Hadoop vendor to offer Spark training
 - Trained more customers than any other vendor
 - Most popular training course
- Cloudera has **5x the engineering resources** of the next competitor
 - Most committers on staff and most changes contributed
 - Well-trained staff across the globe with expertise implementing a broad range of Spark use cases

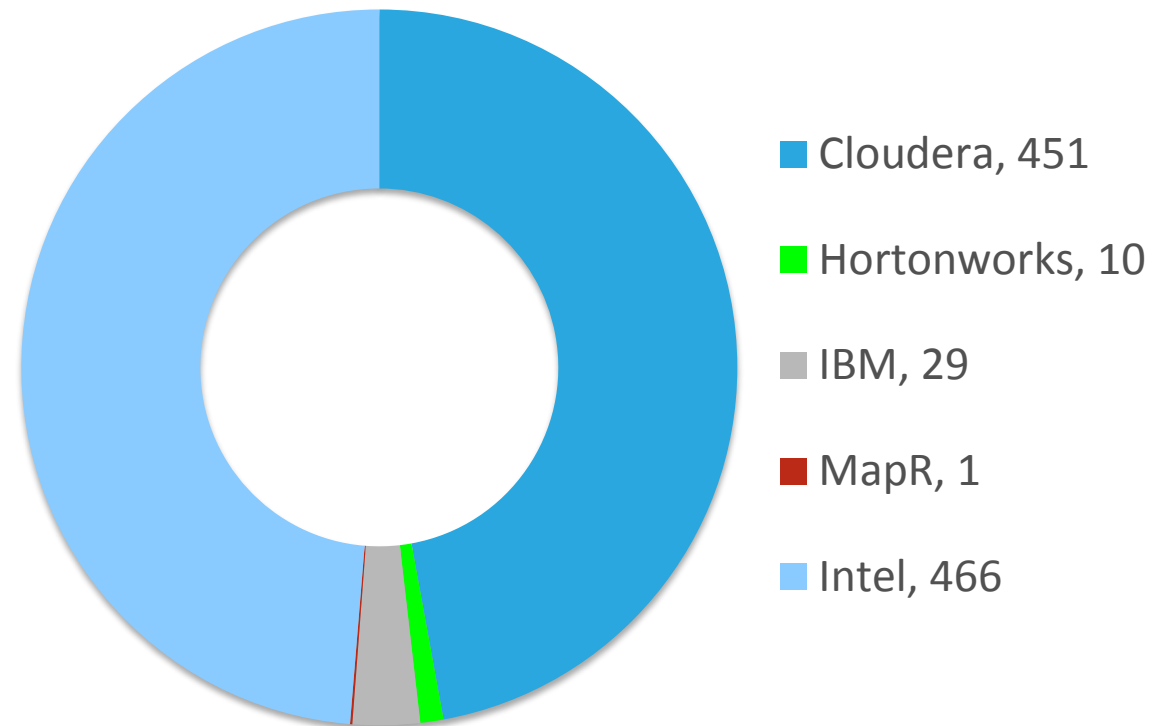
Cloudera's Engineering Commitment to Spark

Spark Committers by Hadoop Distribution*



* IBM and MapR have 0 committers

Spark Patches by Hadoop Distribution



Cloudera Customers

- More customers running Spark than all other vendors combined
 - Over **150** customers
 - Spark clusters as large as **800** nodes
 - Diverse range of use cases across multiple industries
 - Search personalization
 - Genomics research
 - Insurance modeling
 - Advertising optimization
 - Predictive modeling of disease conditions

Cloudera Customer Use Cases

Core Spark



Financial
Services

- Portfolio Risk Analysis
- ETL Pipeline Speed-Up
- 20+ years of stock data



Health

- Identify disease-causing genes in the full human genome
- Calculate Jaccard scores on health care data sets



ERP

- Optical Character Recognition and Bill Classification



Data
Services

- Trend analysis
- Document classification (LDA)
- Fraud analytics

cloudera

Spark Streaming



Financial
Services

- Online Fraud Detection



Health

- Incident Prediction for Sepsis



Retail

- Online Recommendation Systems
- Real-Time Inventory Management



Ad Tech

- Real-Time Ad Performance Analysis

Spark will replace MapReduce
as the standard execution engine for Hadoop

Community Initiative: Spark Supersedes MapReduce

Cloudera is driving community development to port components to Spark:

Stage 1

- Crunch on Spark
- Search on Spark

Stage 2

- Hive on Spark (beta)
- Spark on HBase (beta)

Stage 3

- Pig on Spark (alpha)
- Sqoop on Spark

Uniting Spark and Hadoop

The One Platform Initiative Investment Areas

Management

Leverage Hadoop-native resource management.

Security

Full support for Hadoop security and beyond.

Scale

Enable 10k-node clusters.

Streaming

Support for 80% of common stream processing workloads.

Management

Leverage Hadoop-native resource management

Spark-on-YARN

- Drove Spark-On-YARN Integration
- Improve Spark-On-YARN Integration for better multi-tenancy, performance and ease of use

Automation

- Dynamic Resource Allocation based on needs of job
- Smart auto-selection and tuning of job parameters (when data volumes change)

Metrics

- Improved metrics for debugging and monitoring
- Improve metrics for visibility into resource utilization
- Revamp WebUI for better debugging and monitoring (especially at high concurrency)

Accessibility

- SparkSQL & Hive integration improvements
- Easy Python dependency management for PySpark

Security

Full support for Hadoop security and beyond

Perimeter

- Kerberos Integration

Access

- HDFS Sync (Apache Sentry)
- Enable column- and view-level security

Visibility

- Audit/Lineage via Cloudera Navigator
- Full Spark PCI compliance

Data Protection

- Integration with Intel's Advanced Encryption libraries

Scale

Enable 10K-Node Clusters

Fault-Tolerance

- Revamp Scheduler handling of node failure
- Dynamic resource utilization & prioritization

Stability

- Sort Based Shuffle Improvements for improved stability at scale
- Stress test at scale with mixed multi-tenant workloads
- Scale Spark History Server for 1000s of jobs

Performance

- Task scheduling based on HDFS data locality & HDFS caching (reduce data movement and enable more jobs)
- Integrate with HDFS Discardable Distributed Memory (reduce memory pressure)
- Scheduler improvements for performance at scale

Streaming

Support for 80% of common stream processing workloads

Zero Data Loss

- Delivered full data resiliency

Ingest

- Delivered Flume integration and drove Kafka integration

Management

- Streaming application management via Cloudera Manager for zero downtime

Performance

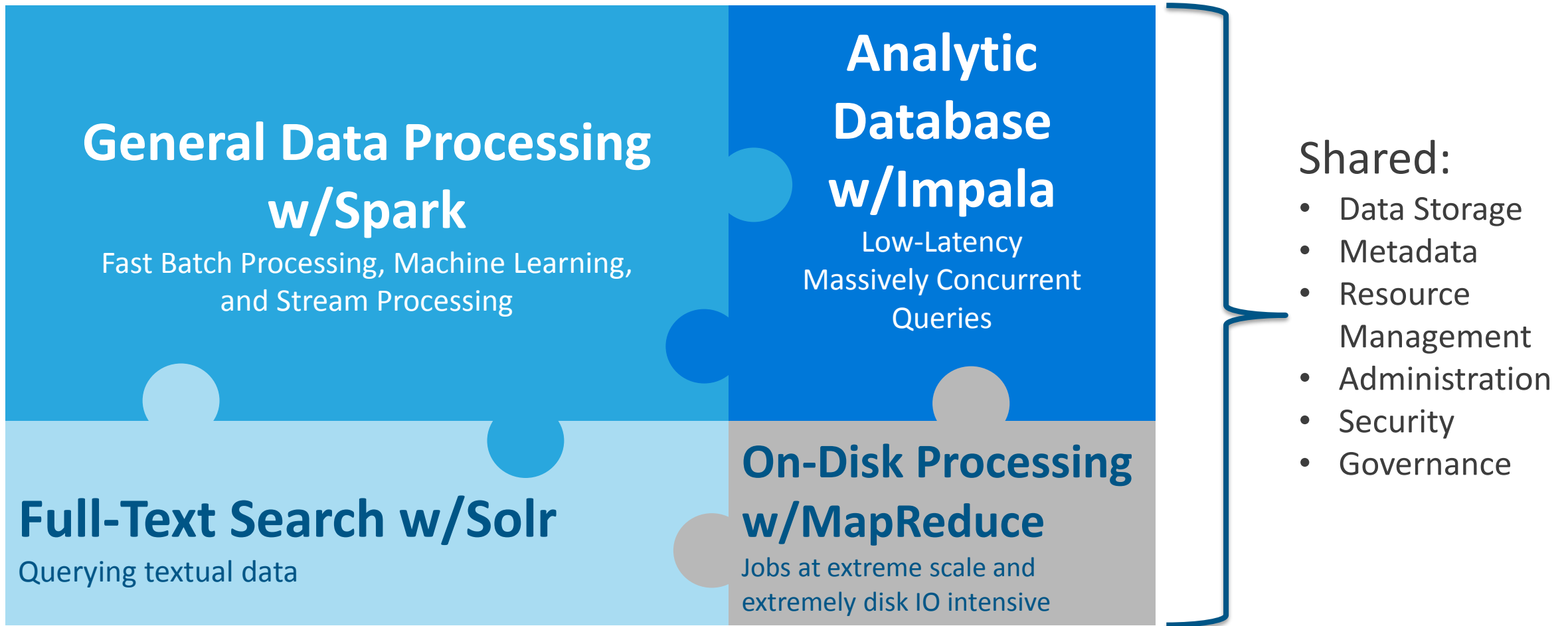
- Improved State Management to enable maintaining a high volume of state information

Accessibility

- SQL interfaces and API extensions for Streaming jobs

The Future of Data Processing on Hadoop

Spark complemented by specialized fit-for-purpose engines



Spark Resources

- Learn Spark
 - O'Reilly [Advanced Analytics with Spark](#) eBook (written by Clouderans)
 - [Cloudera Developer Blog](#)
 - cloudera.com/spark
- Get Trained
 - [Cloudera Spark Training](#)
- Try it Out
 - [Cloudera Live Spark Tutorial](#)



cloudera

Thank You!

gpymm@cloudera.com