LLAP: Sub-Second Analytical Queries in Hive

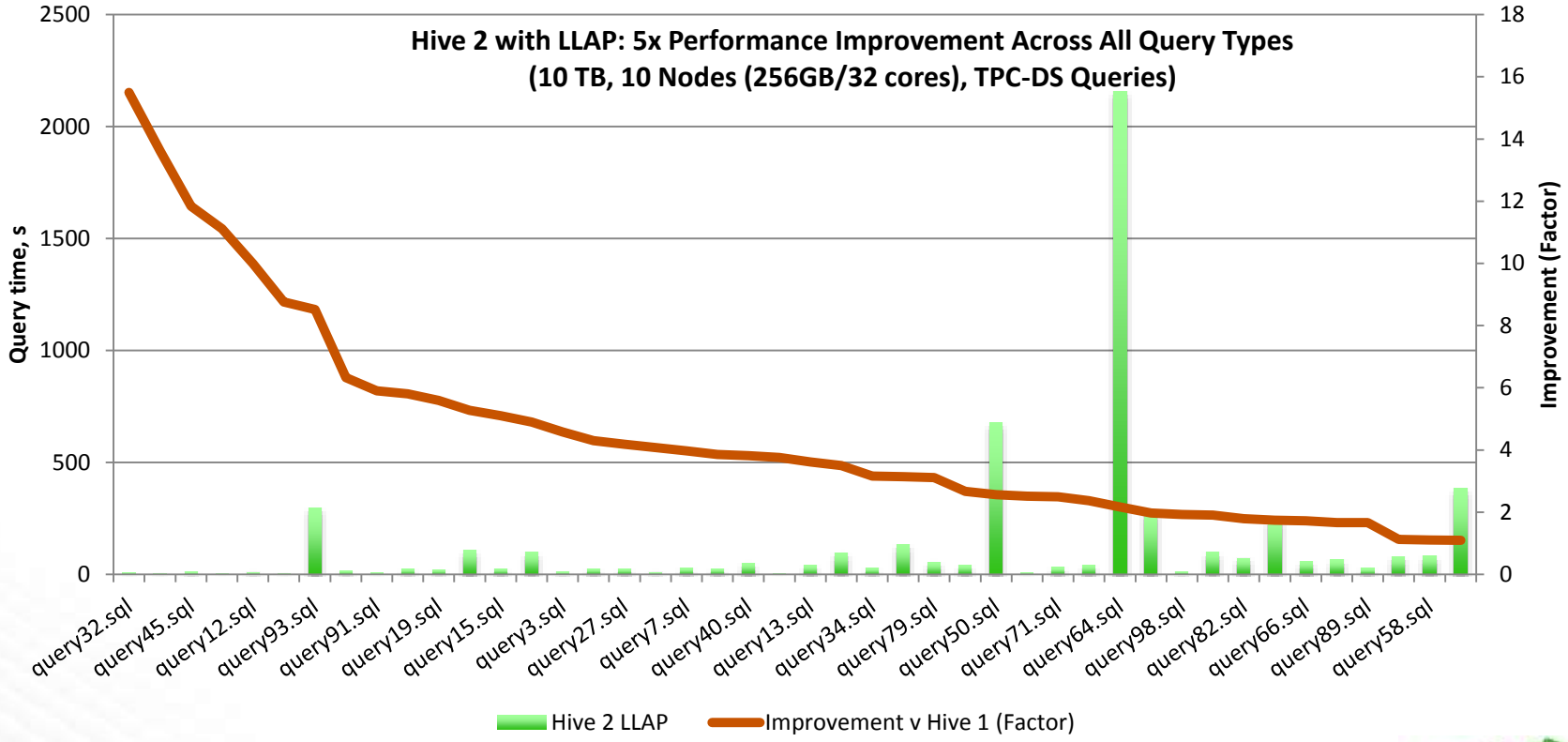
Gunther Hagleitner

```
hive (tpcds_bin_partitioned_orc_1000)> set hive.llap.execution.mode;
hive.llap.execution.mode=all
hive (tpcds_bin_partitioned_orc_1000)> set hive.llap.execution.mode
```

```
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 2 .......... container     SUCCEEDED     31         31        0        0       0       0
Map 5 .......... container     SUCCEEDED      7          7        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 9.47 s
----------------------------------------------------------------------------------------------
Status: DAG finished successfully in 9.47 seconds
```

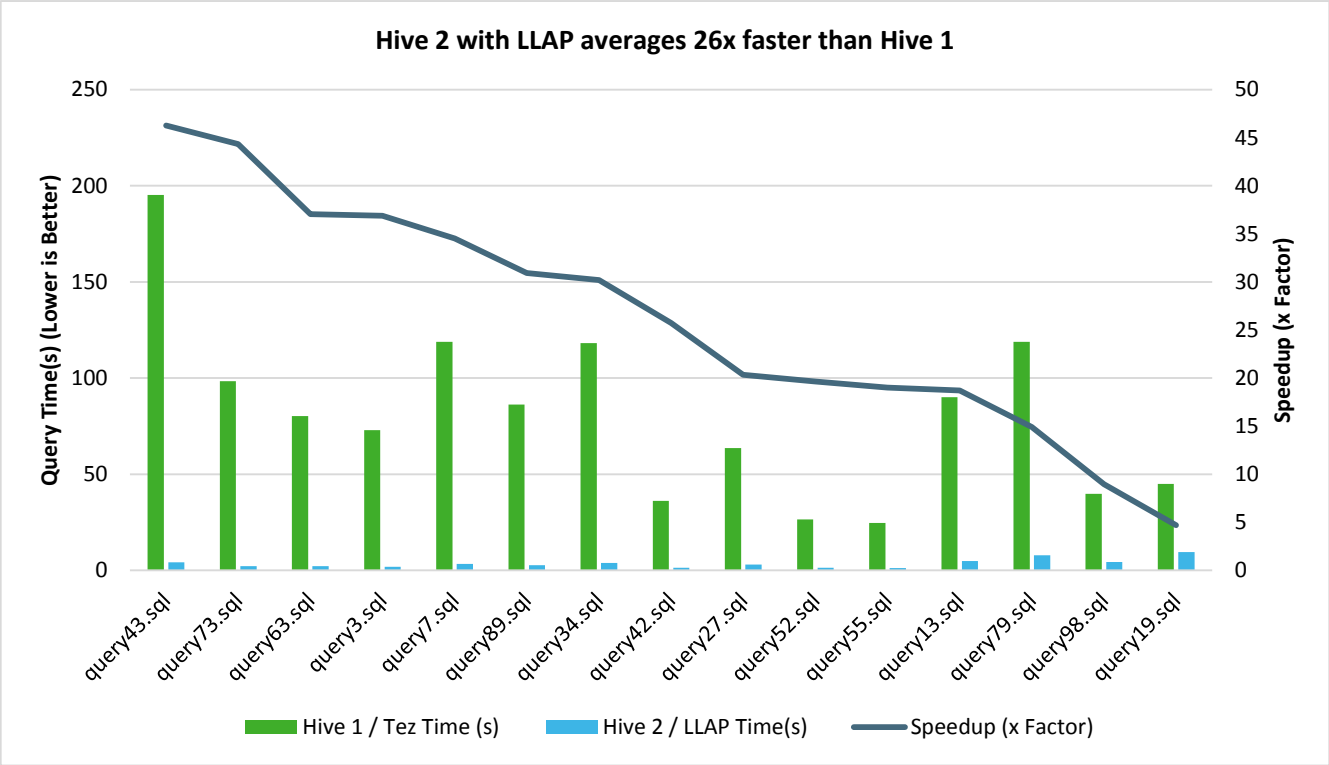
```
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 ..........        llap     SUCCEEDED      1          1        0        0       0       0
Map 2 ..........        llap     SUCCEEDED     31         31        0        0       0       0
Map 5 ..........        llap     SUCCEEDED      7          7        0        0       0       0
Reducer 3 ......        llap     SUCCEEDED      1          1        0        0       0       0
Reducer 4 ......        llap     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 05/05  [==========================>>] 100%  ELAPSED TIME: 0.89 s
----------------------------------------------------------------------------------------------
Status: DAG finished successfully in 0.89 seconds
```

Hive 2 with LLAP: 5x Performance Boost at 10 TB Scale



© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hive 2 with LLAP: 25+x Performance Boost: Interactive / 1TB Scale

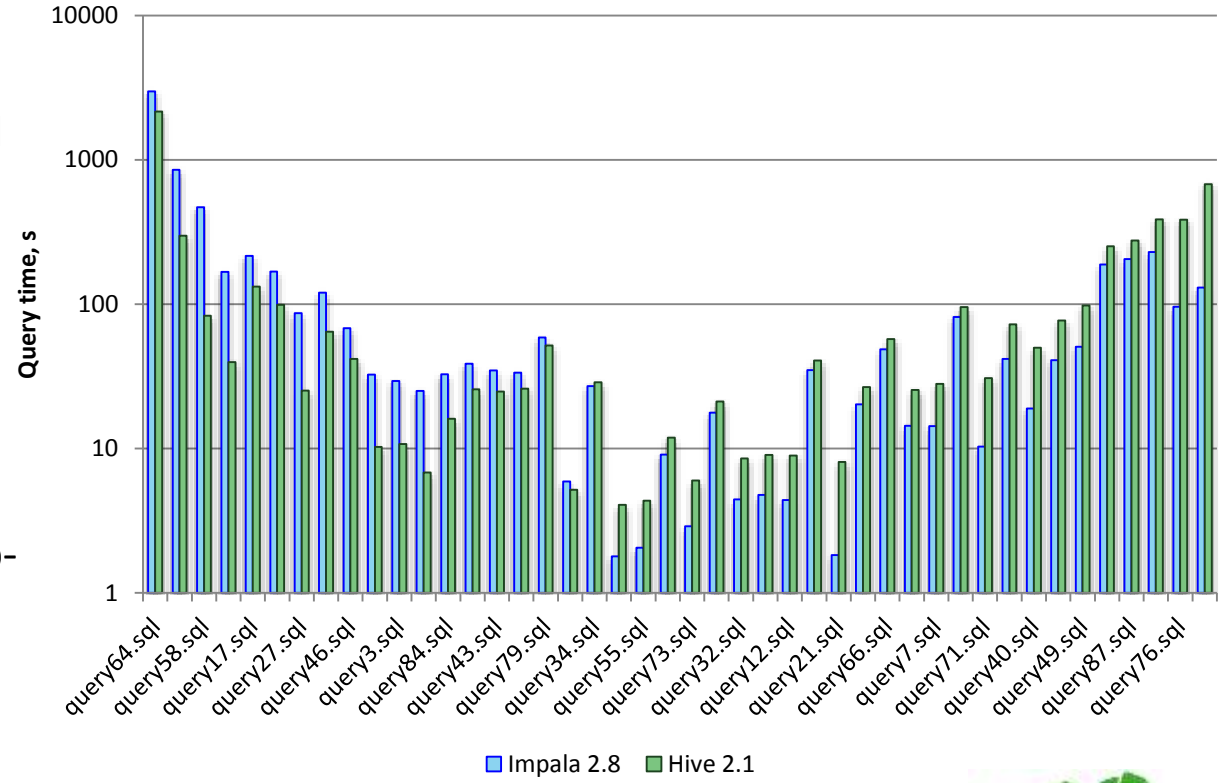


© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Apache Hive vs. Apache Impala at 10TB

Highlights

- 10TB scale on 10 identical nodes.
- Hive and Impala showed similar times on most smaller queries.
- Hive scaled better, completing larger queries quicker with lower end-to-end time for all queries

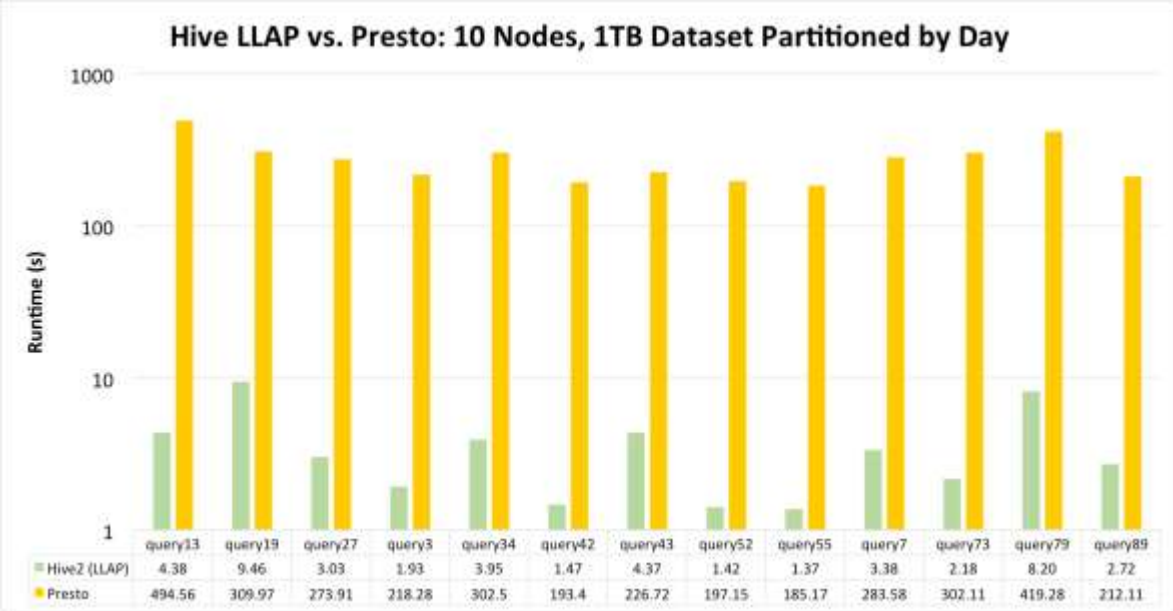


© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Apache Hive vs. Presto on a partitioned 1TB dataset.

Highlights

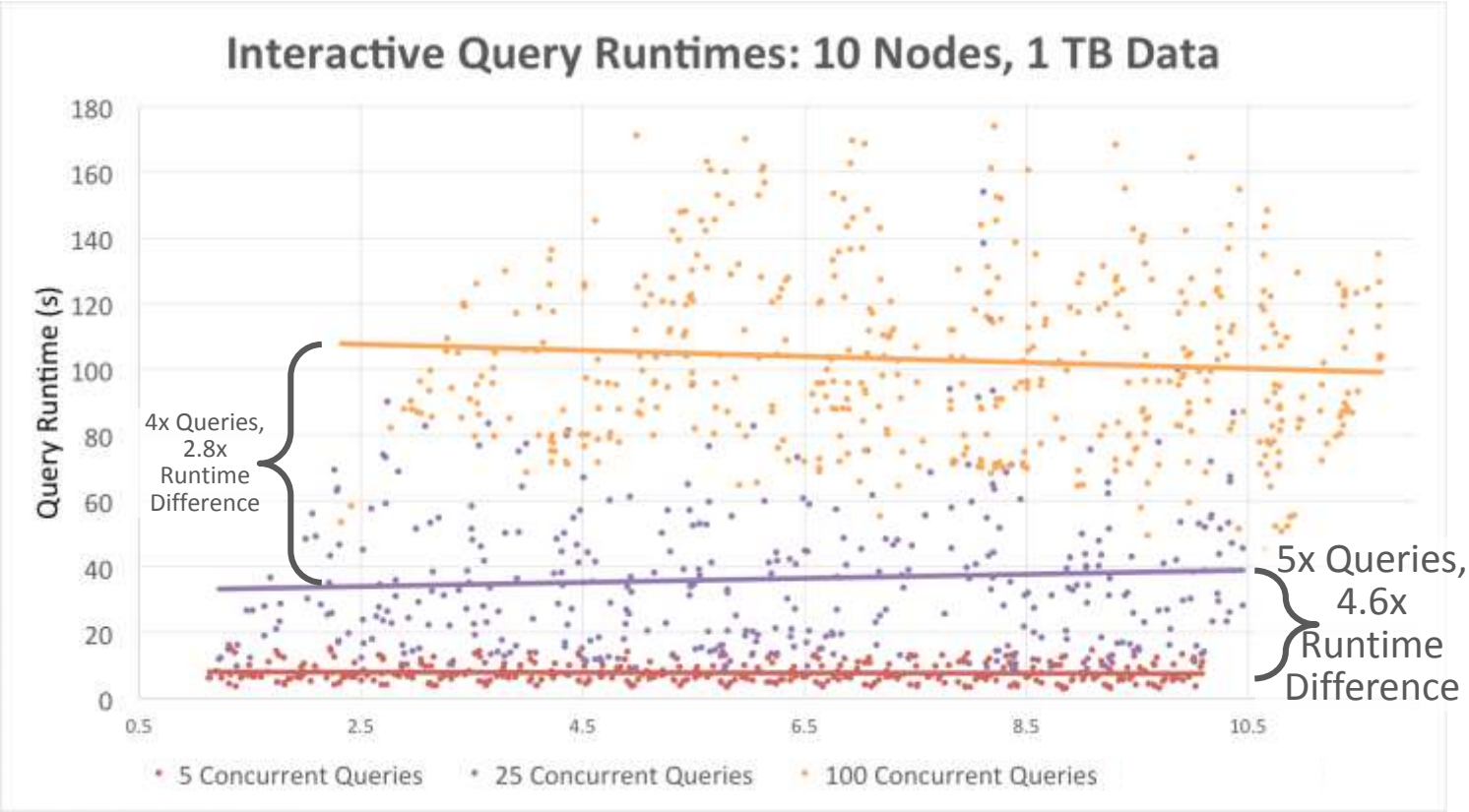
- Presto lacks basic performance optimizations like dynamic partition pruning.
- On a real dataset / workload Presto perform poorly without full re-writes.
- Example: Query 55 without re-writes = 185.17s, with re-writes = 16s. LLAP = 1.37s.



	query13	query19	query27	query3	query34	query42	query43	query52	query55	query7	query73	query79	query89
Hive2 (LLAP)	4.38	9.46	3.03	1.93	3.95	1.47	4.37	1.42	1.37	3.38	2.18	8.20	2.72
Presto	494.56	309.97	273.91	218.28	302.5	193.4	226.72	197.15	185.17	283.58	302.11	419.28	212.11

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hive LLAP: Stable Performance under High Concurrency

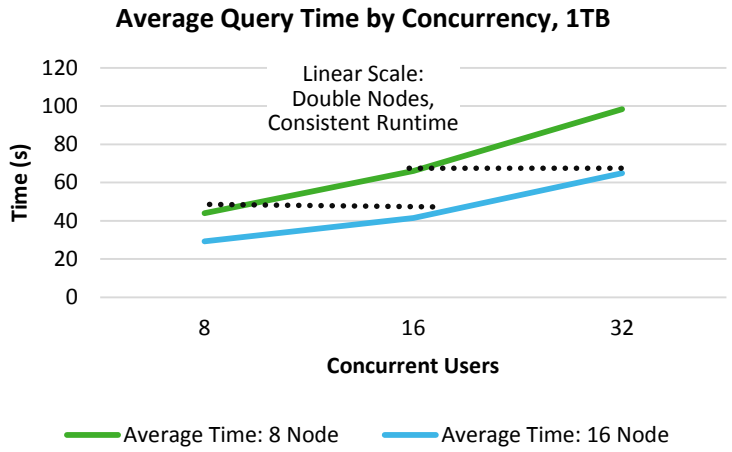


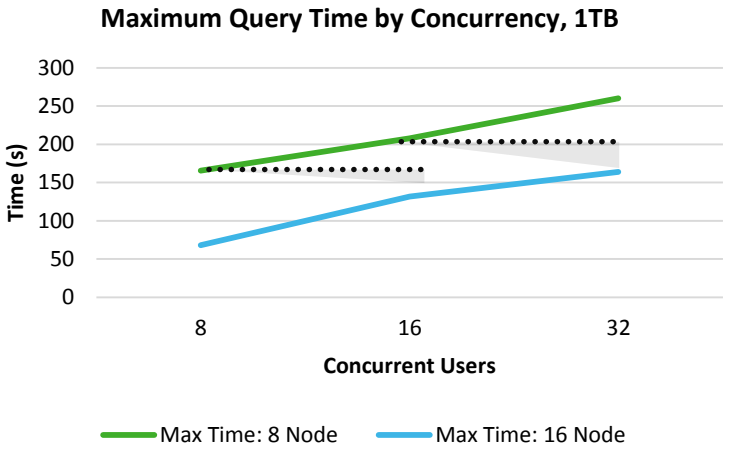
Mark	Concurrent Queries	Average Runtime
	5	7.76s
	25	36.24s
	100	102.89s

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

HORTONWORKS

Hive with LLAP: Linear Scale

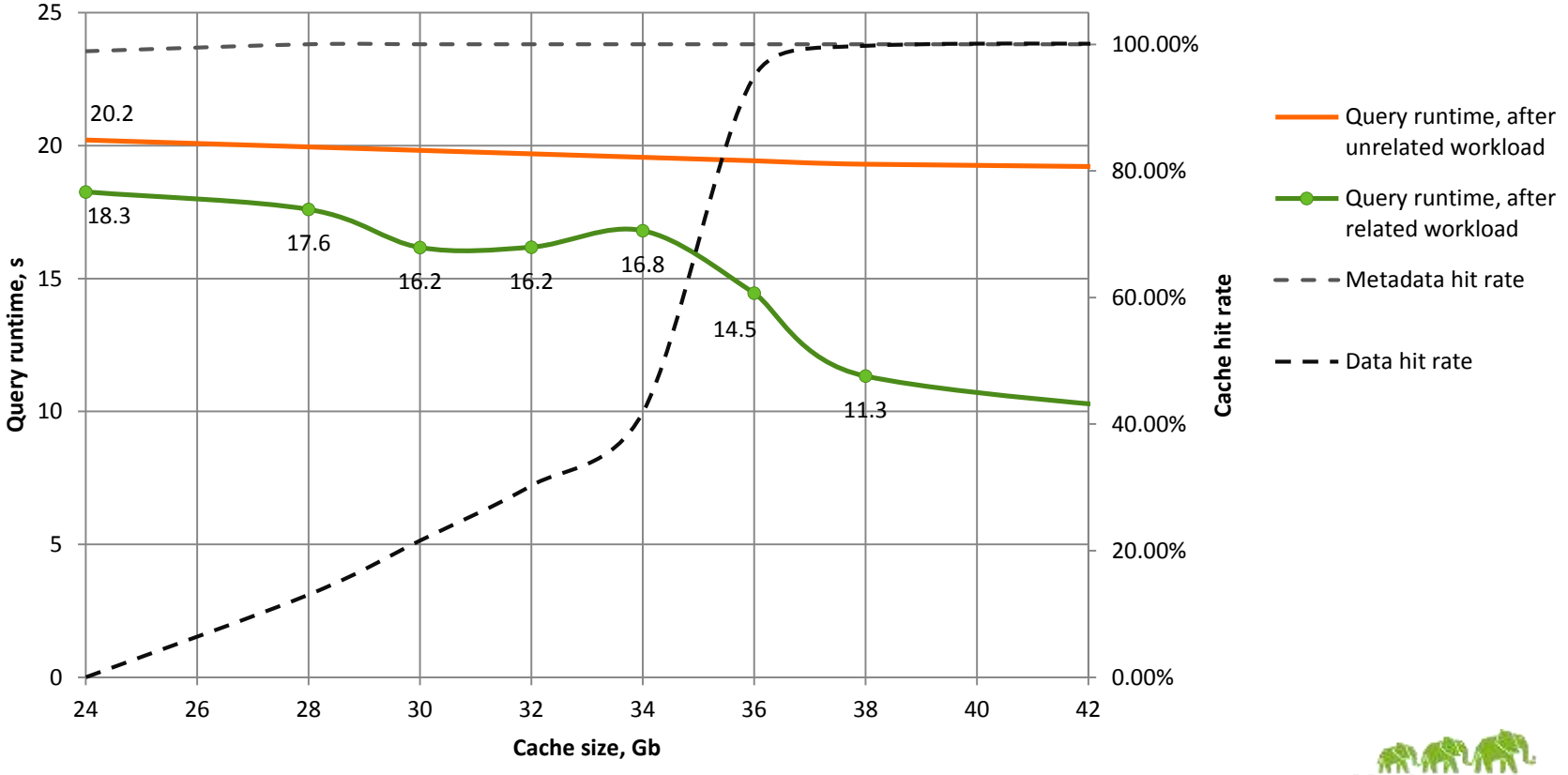




Average and Maximum query times
Mixed SQL workload of simple and complex queries
1TB Dataset Size

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Performance – cache on HDFS, 1Tb scale

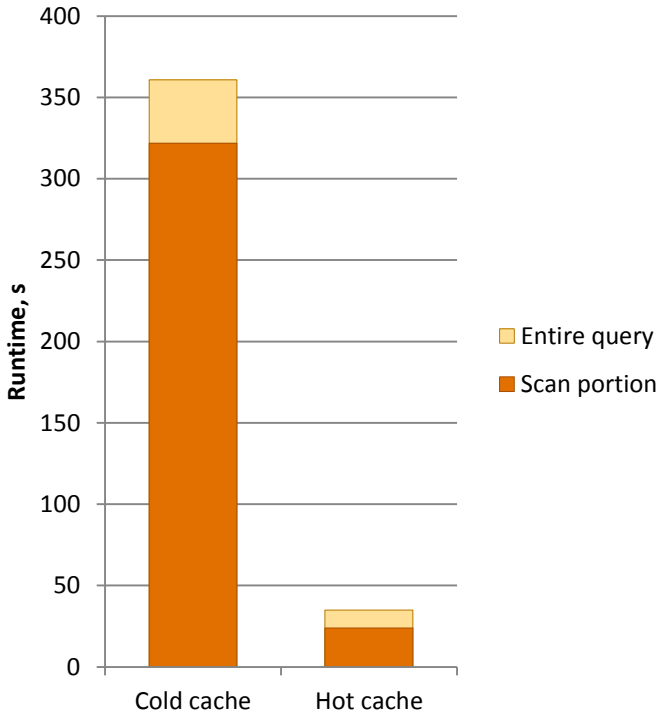


© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks

Performance – cache on cloud storage

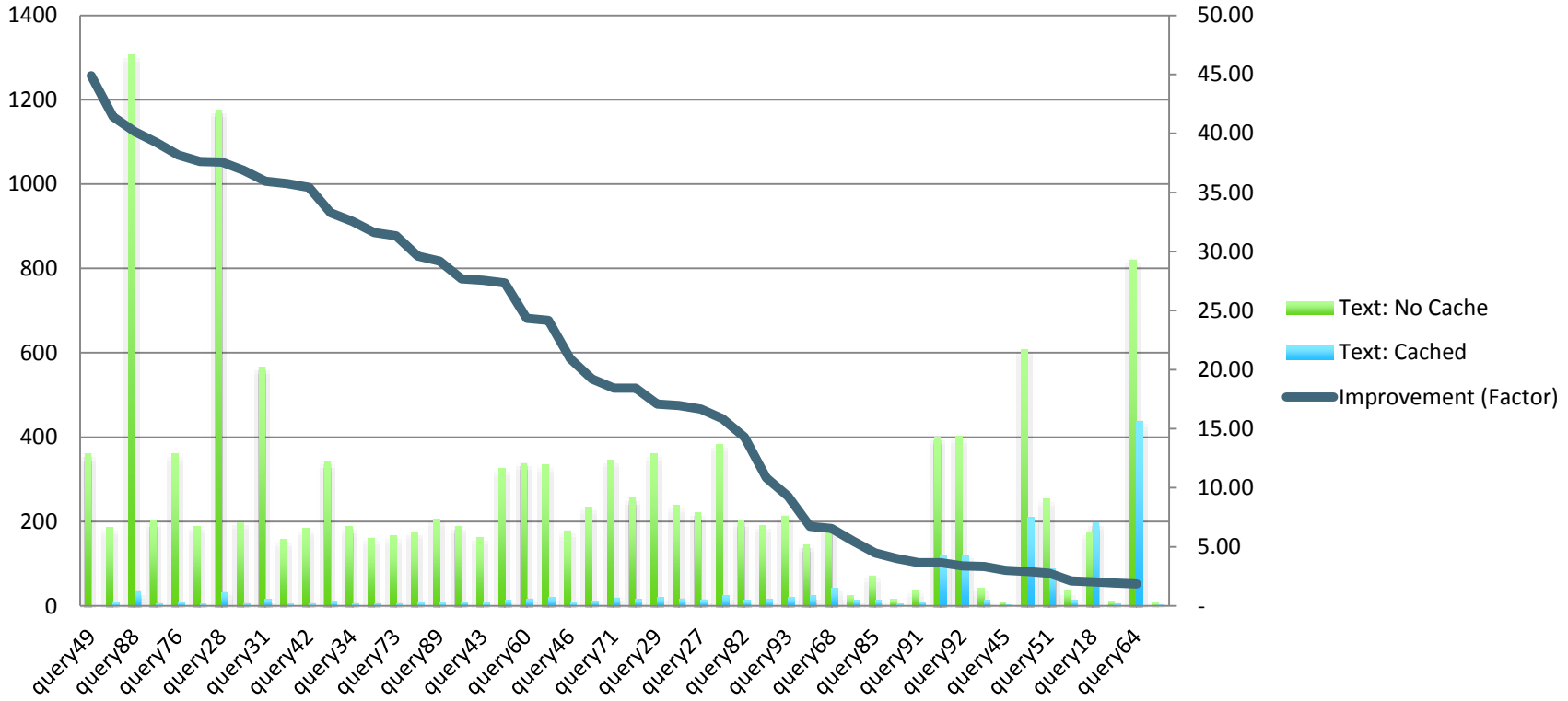
- Cloud storage, slower than on-prem HDFS
- Large scan on 1Tb scale
- Local HD/SDD cache
  - Massive improvement on slow storage with little memory cost



© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks

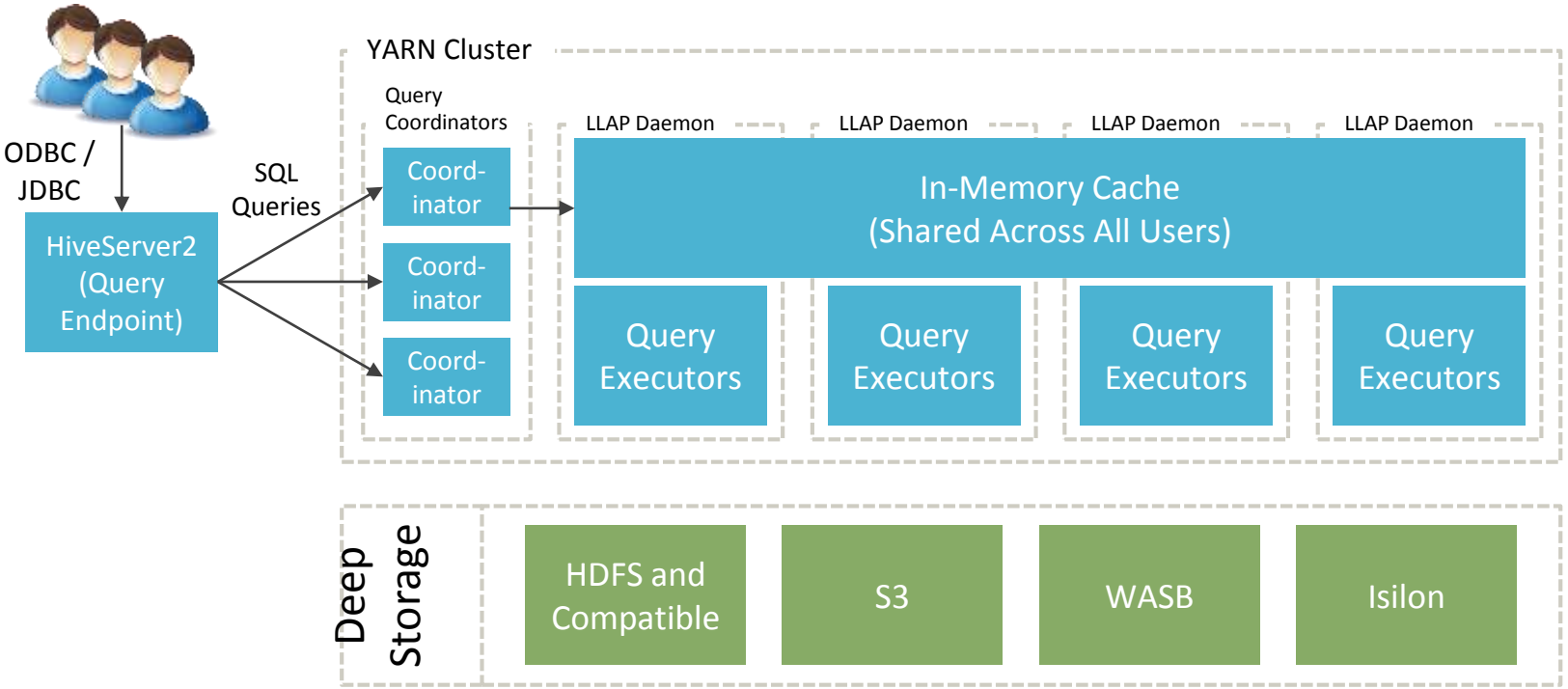
Hive with LLAP on text data (CSV, logs, etc)



© Hortonworks Inc. 2011 – 2017. All Rights Reserved

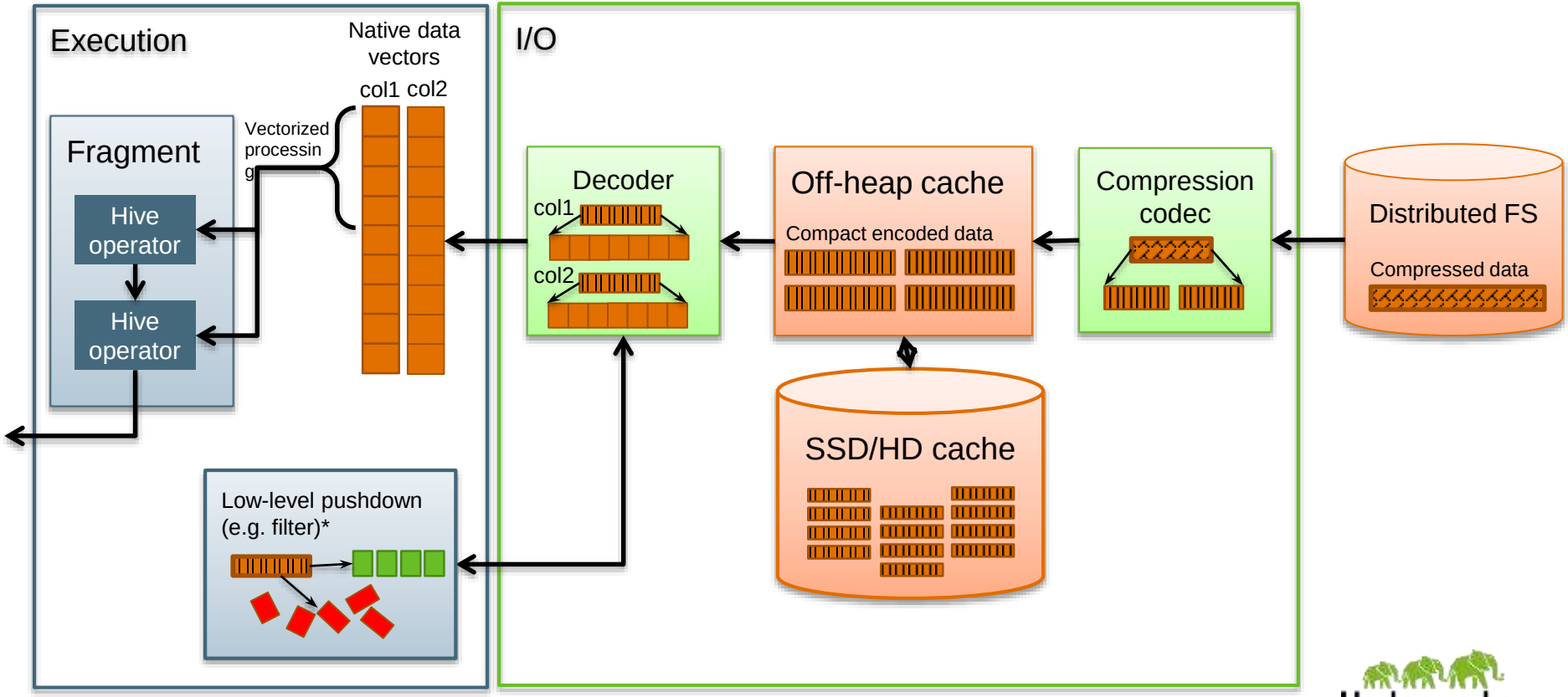
Hortonworks

Hive with LLAP: Architecture Overview



© Hortonworks Inc. 2011 – 2017. All Rights Reserved

LLAP: In-memory processing

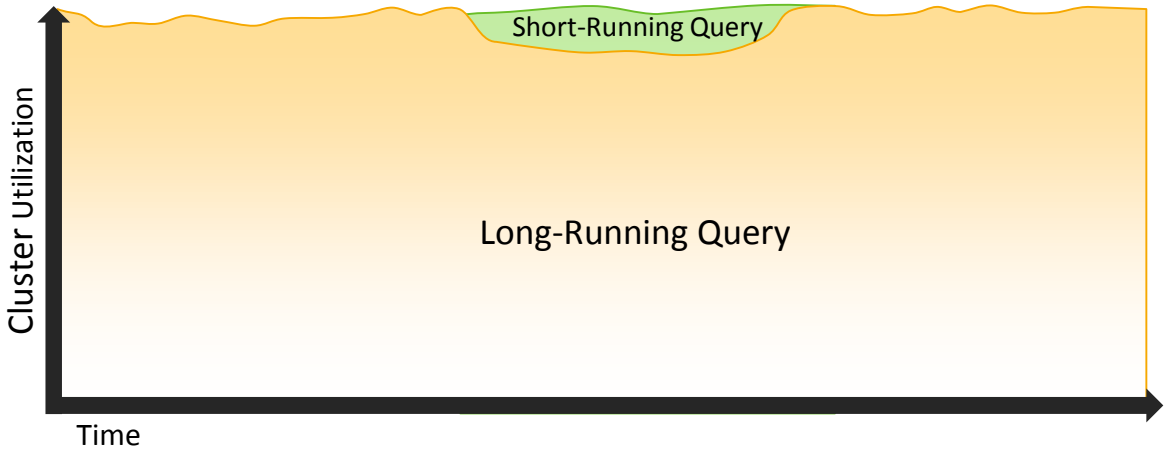


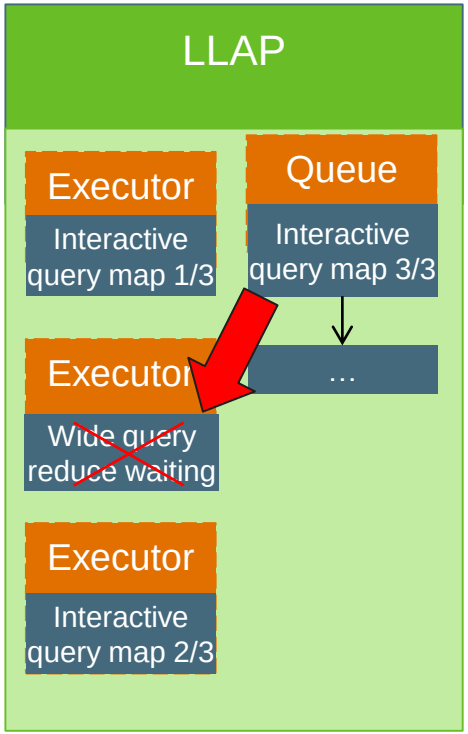
© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks

Parallel queries – priorities, preemption

- Lower-priority fragments can be preempted
  - For example, a fragment can start running before its inputs are ready, for better pipelining; such fragments may be preempted
- LLAP work queue examines the DAG parameters to give preference to interactive (BI) queries





© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks

Sidebar: Relational data node

- LLAP can provide a "relational datanode" view of the data
- Standard interface to read tables directly from LLAP nodes
  - Supports column-level security
  - LLAP handles ACID, schema-evolution
- Optimized for fast access/scans
  - Read horizontal table partitions in parallel on compute nodes
  - Push projections, filters, aggregates into LLAP
  - Utilizes cache and LLAP IO subsystem

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Example - SparkSQL integration

- Allowing direct access to Hive table data from Spark breaks security model for warehouses secured using Ranger or SQL Standard Authorization
- Other features may not work correctly unless support added in Spark
  - Row/Cell level security (when implemented), ACID, Schema Evolution, …

- SparkSQL has interfaces for external sources; they can be implemented to access Hive data via HS2/LLAP
- SparkSQL Catalyst optimizer hooks can be used to push processing into LLAP (e.g. filters on the tables)
  - Some optimizations, like dynamic partition pruning, can be used by Hive even if SparkSQL doesn't support them; if execution pushdown is advanced enough

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

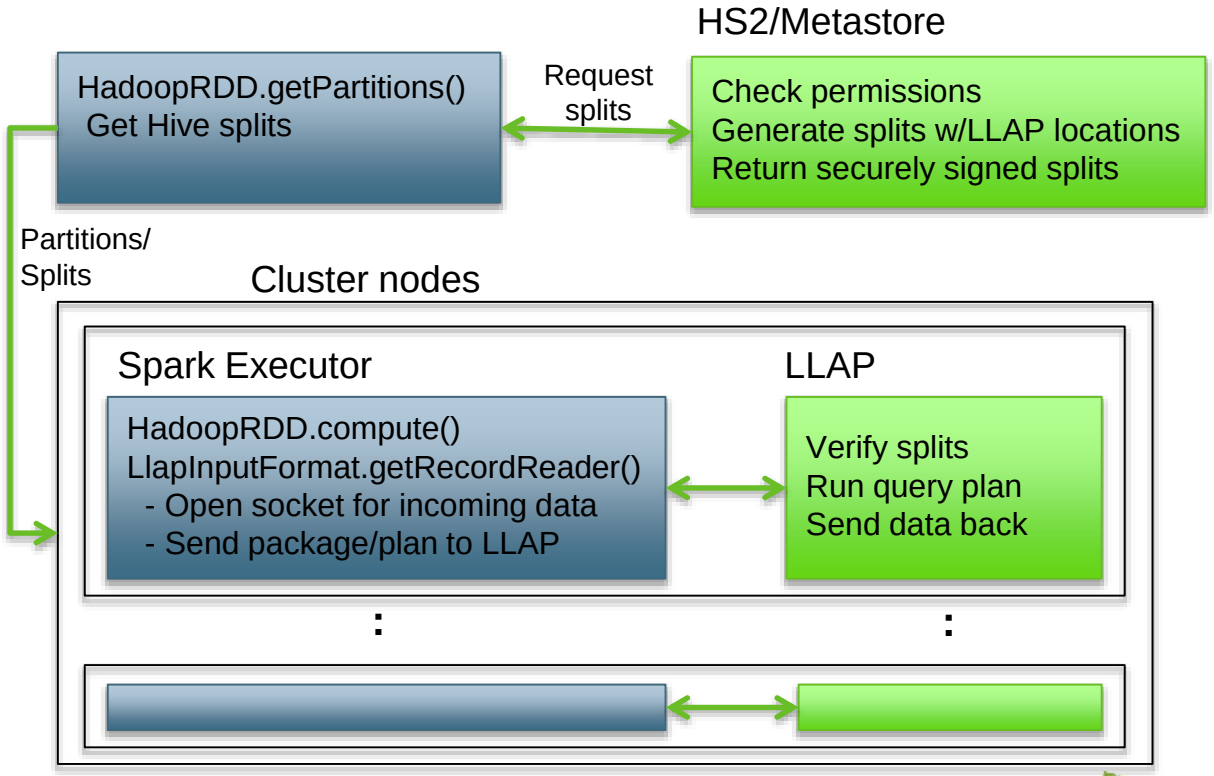
Hortonworks

Example - SparkSQL integration – execution flow

```
var llapContext =
LlapContext.newInstance(
sparkContext, jdbcUrl)

var df: DataFrame =
llapContext.sql("select *
from tpch_text_5.region")
```

DataFrame for Hive/LLAP data



© Hortonworks Inc. 2011 – 2017. All Rights Reserved

LLAP Availability

- First version shipped in Apache Hive 2.0
  - Hive 2.0.1 contains important bug fixes to make it production ready
  - Hive 2.1 has new features and further perf improvements

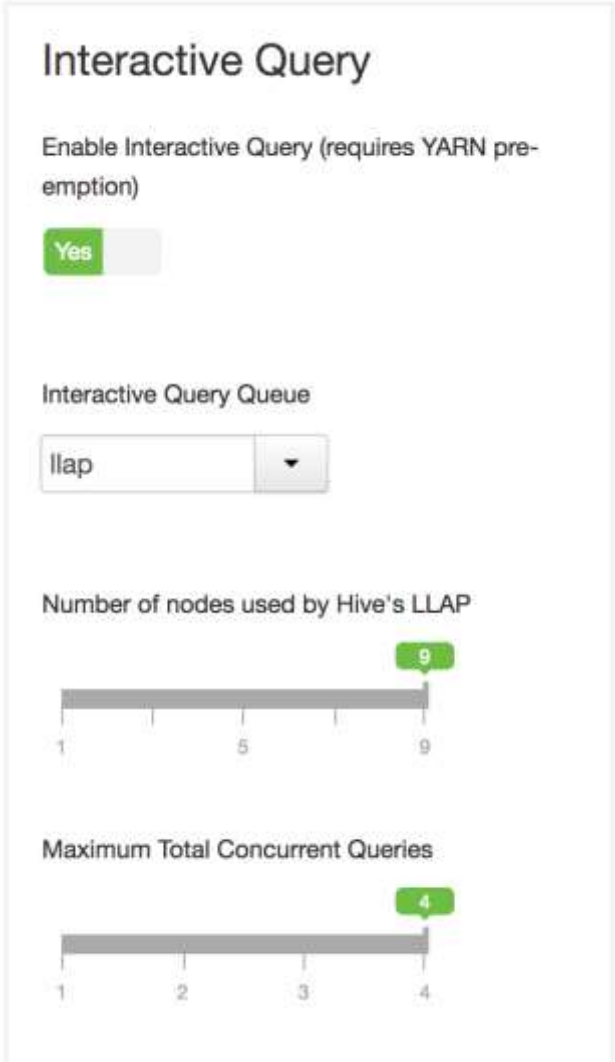
- HDP 2.5 includes beta version of LLAP
- HDP 2.6 includes GA version of Apache Hive 2.1 w/ LLAP

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks

Setup: Ambari

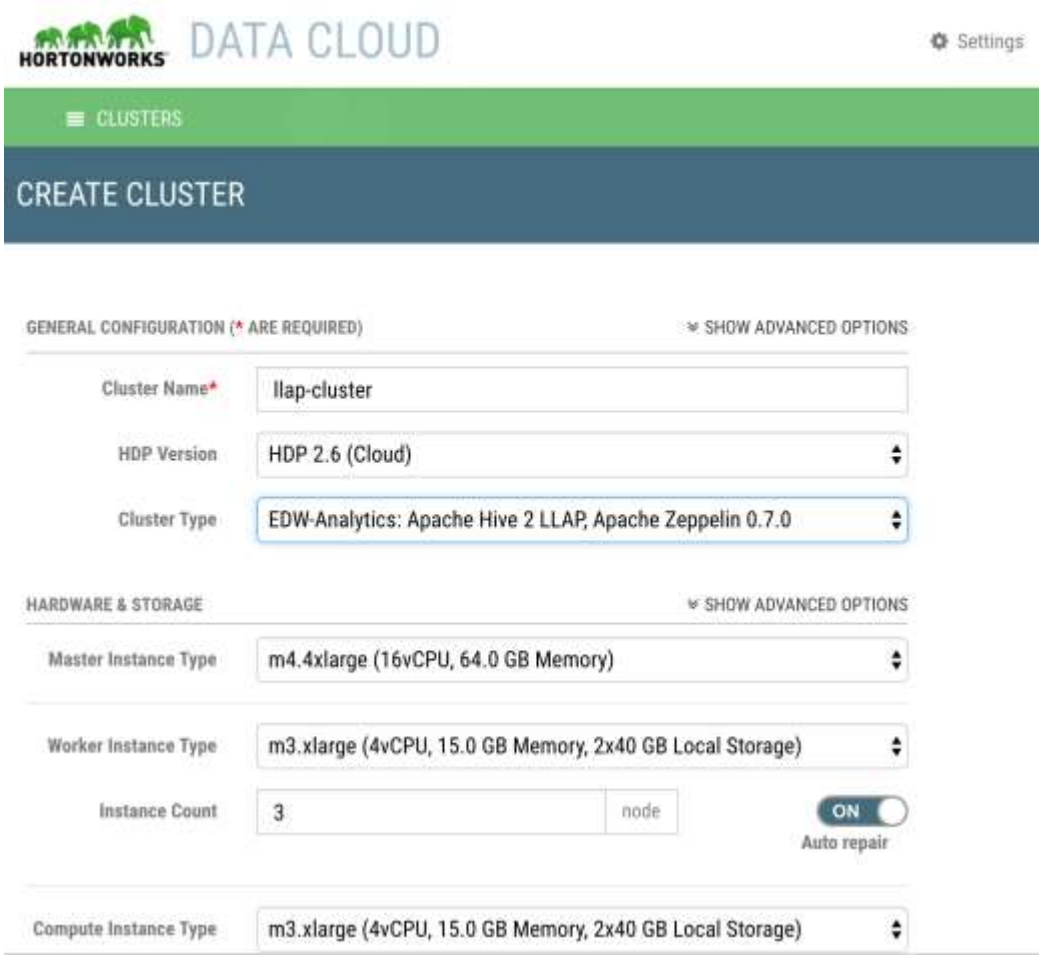
- Simple setup
- Integrated into Hive configuration page
- Launches HS2 instance and LLAP cluster
- Handles configuration, slider, security, alerts and monitoring, …
- Requires Ambari 2.5.x
- Requires HDP 2.6.x stack (includes Apache Hive 2.1 w/ LLAP)



© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Setup: Cloud

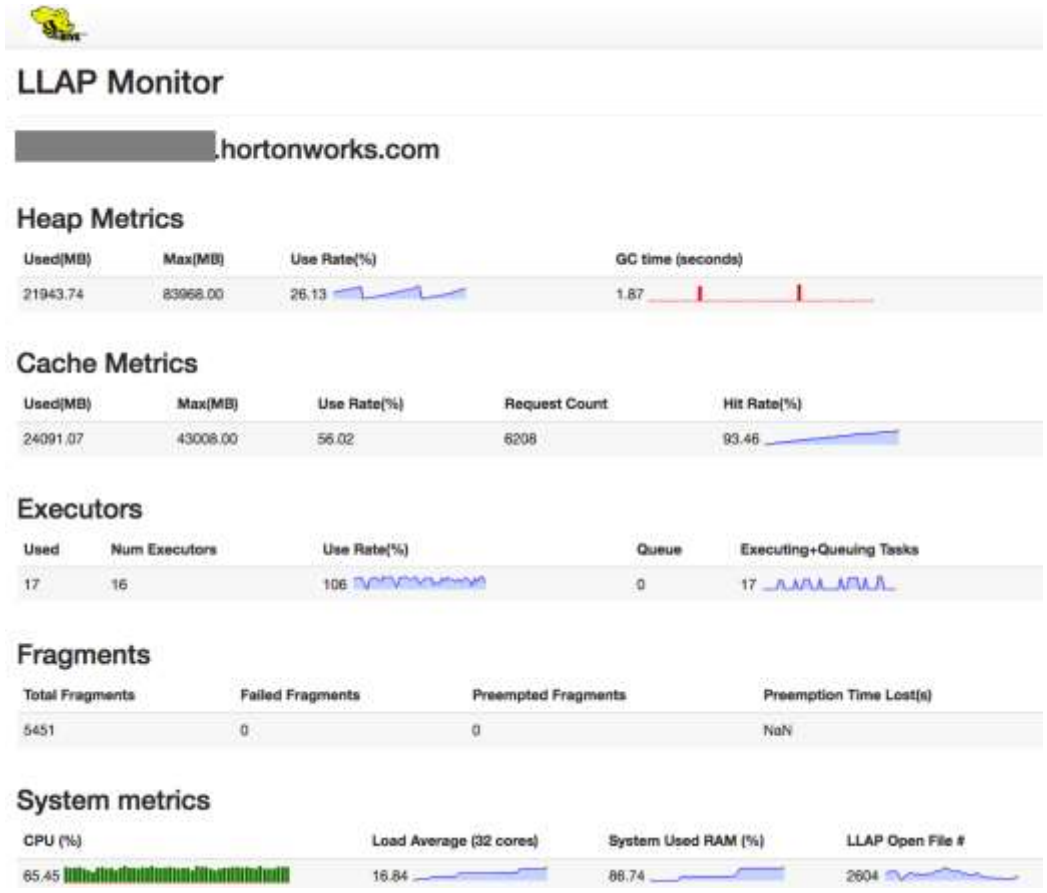
- **Spin up LLAP clusters in minutes**
- **Exposes HS2 endpoint**
  - JDBC/ODBC
  - Zeppelin/Hive view also options
- **RDS + S3 friendly**
  - Use HD cache for fast S3 access
  - RDS for shared metastore instance
- **HDP 2.6 available now!**



© Hortonworks Inc. 2011 – 2017. All Rights Reserved
Hortonworks

Monitoring

- Simple UI for monitoring
- JMX endpoint with more data
- Logs and jstack endpoints

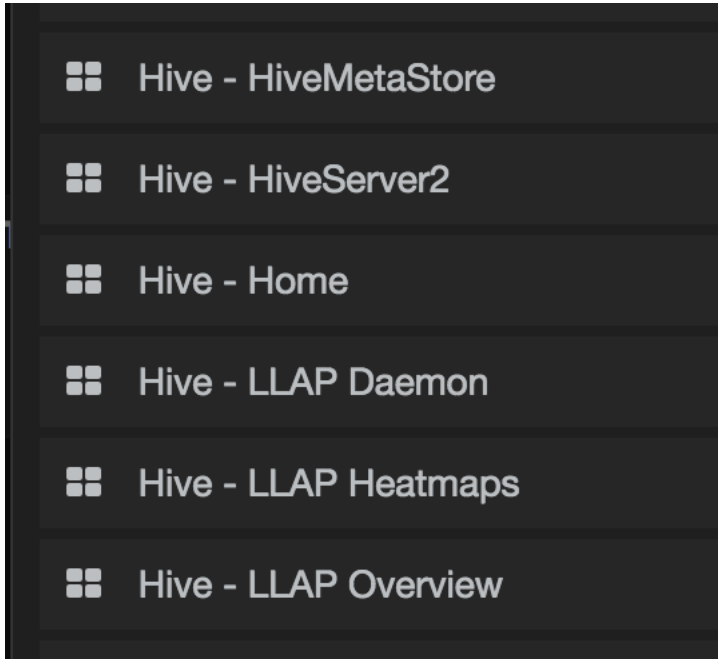


© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks

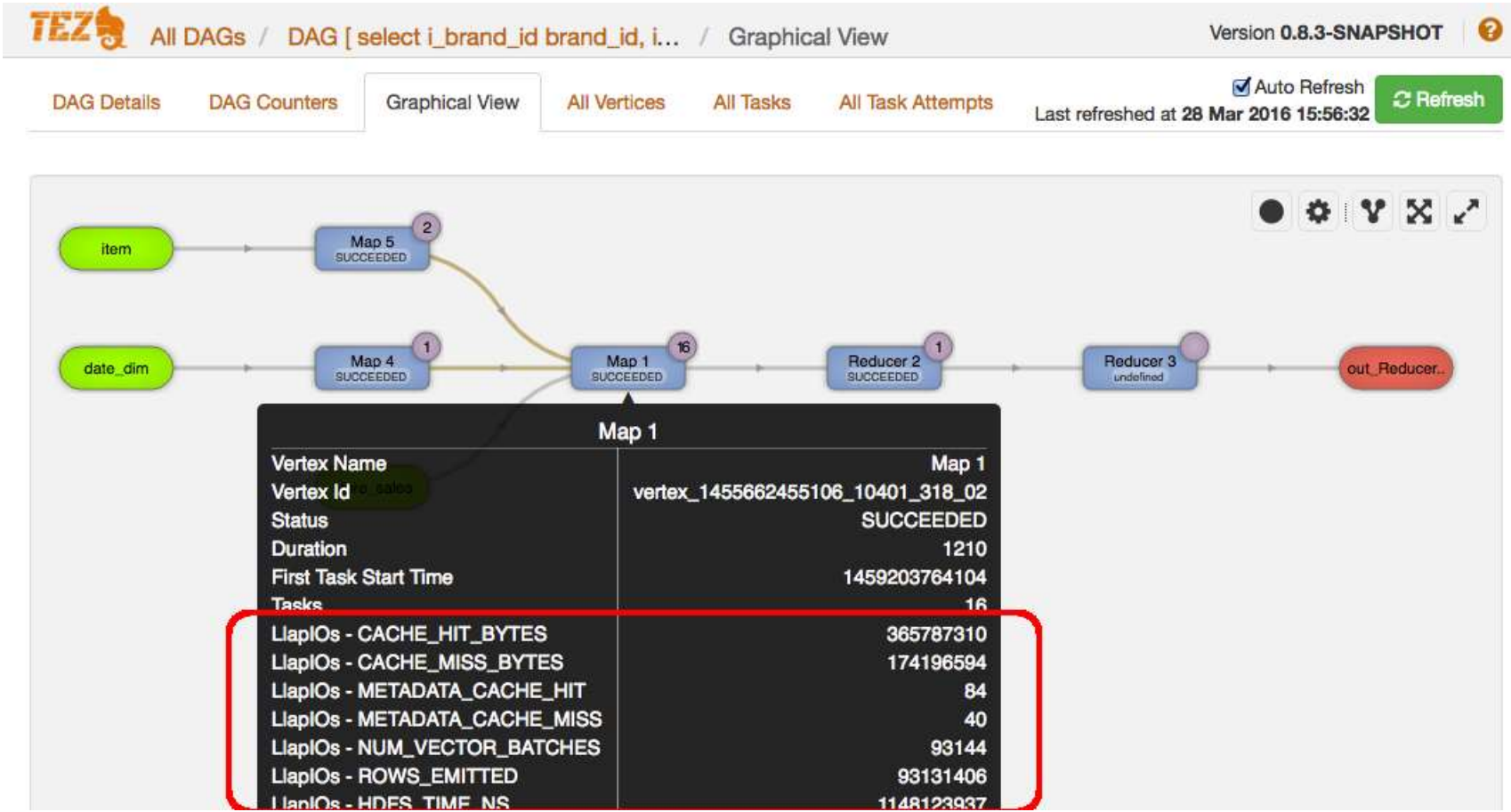
Monitoring

- HDP ships with number of default Grafana dashboards for LLAP
- Show wealth of metrics for cache, compute and scheduling
- Helps to identify bottlenecks and issues
- Helps with capacity planning
- Easy to extend or integrate with other systems



© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Watching queries – Tez UI integration



Future work

- **Fine-grained workload management**
  - Reservations & priorities
- **Configurable guardrails**
  - Maximum permissible scan range, runtime, cost
  - Integrated with CBO's cost model
- **Admin tools**
  - Identify hotspots & data layout issues
  - Easy reporting on health, performance and utilization

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks

Summary

- LLAP is a new execution substrate for fast, concurrent analytical workloads, harnessing Hive vectorized SQL engine and efficient in-memory caching layer
- Provides secure relational view of the data through a simple API
- Available in Hive 2, integrated with ecosystem (Ambari, Cloud, Grafana, jmx, Hive & Tez views)

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Questions?

Interested? Stop by the Hortonworks booth to learn more

© Hortonworks Inc. 2011 – 2017. All Rights Reserved

Hortonworks