The state of Spark in the cloud

Nicolas Poggi

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

SPARK SUMMIT EUROPE 2017

Oct 2017

aloja

Outline

1. Intro
    1. PaaS Cloud
    2. BigBench
2. Part I – Scalability
    1. Hive vs. Spark
3. Part II – Additional experiments
    1. Versions, Concurrency, 10TB
4. Summary

Motivation

- 2016 **SQL-on-Hadoop paper** and presentations
  - Focused on Hive, due to **SparkSQL** not being ready to use in PaaS
  - Used benchmark (TPC-H)

- Early 2017, **BigBench** testing Hive and Spark and TPC-TC paper
  - New code available in May for MLlib2 compatibility

- **Goals:**

Evaluate the current out-of-the-box experience of Spark v2 in PaaS cloud

  - Using Hive as baseline

Platform-as-a-Service Spark

aloja

- Simplified management
- Cloud-based managed Hadoop services
  - Ready to use Spark, Hive, …
- Deploys in minutes, on-demand, elastic
- *Pay-as-you-go* pricing model
- Decoupled compute and storage
- Optimized for general purpose
  - Fined tuned to the cloud provider architecture

Surveyed PaaS services

- Amazon Elastic Map Reduce (EMR)
  - **Released:** Apr 2009
  - **OS:** Amazon Linux AMI (RHEL-like)
  - **Spark 2.1.0 and Hive 2.1 (Tez)**
  - **VM: M4.2xlarge (32GB RAM)**

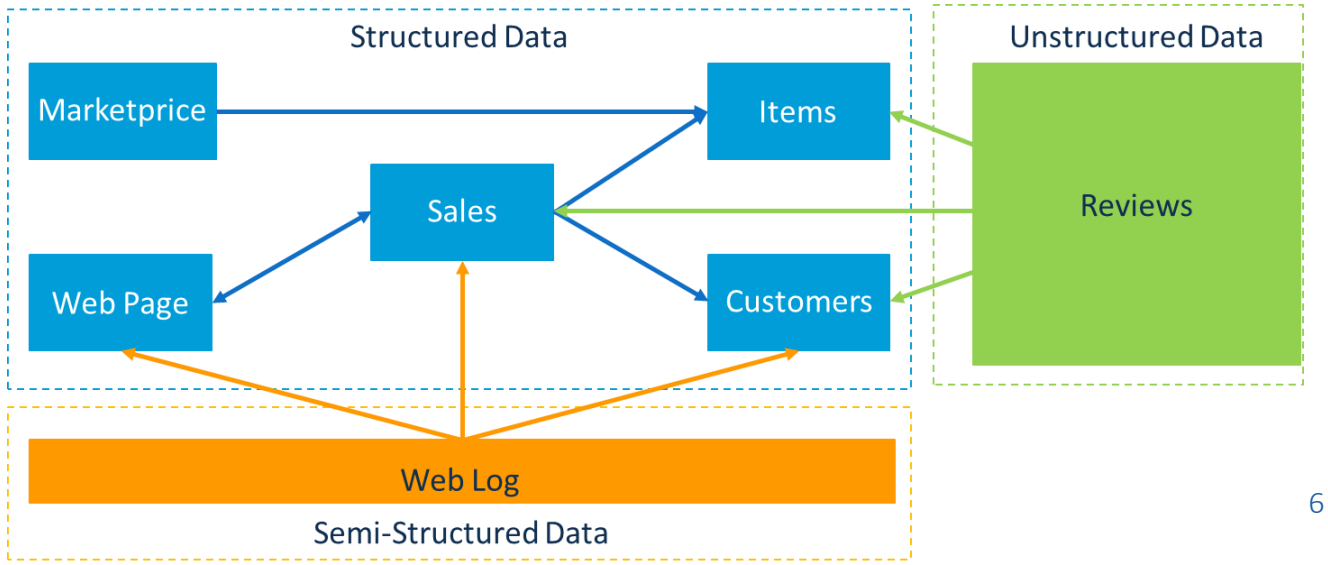
- Google Cloud DataProc (GCD)
  - **Released:** Feb 2016
  - **OS:** Debian 8.4
  - **Spark 2.1.0 (preview),** Hive 2.1 (M/R)
  - **VM: n1-standard-8 (30GB RAM)**

- Azure HDInsight (HDI)
  - **Released:** Oct 2013
  - **OS:** Ubuntu 16.04 (HDP-based)
  - **Spark 2.1.0 and 1.6.3, Hive 1.2 (Tez)**
  - **VM: D4v2 (28GB RAM)**

- Target deployment **128-cores**:
  - **16** data nodes with 8-cores each
  - Master node with 16-cores
  - Decoupled storage only
    - EBS, WASB, GCS

What is **BigBench** (TPCx-BB)

- End-to-end application level benchmark specification
  - result of many years of collaboration of industry and academia
- Covers most Big Data Analytical properties (3Vs)
- 30 business use cases for a retailer company
  - Merchandizing,
  - pricing,
  - customers …
- Defines data scale factors
  - 1GB to PBs

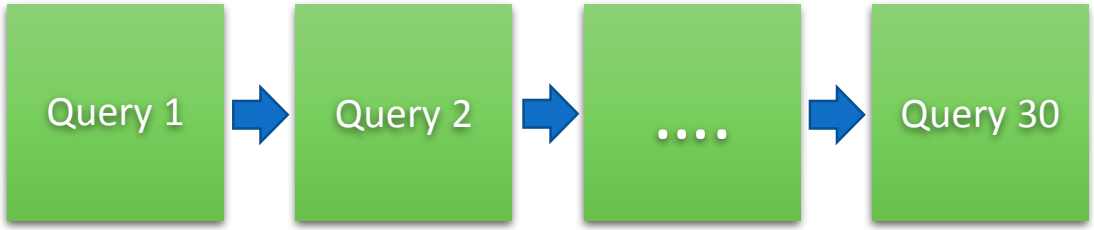


```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.1.0
      /_/
```

aloja

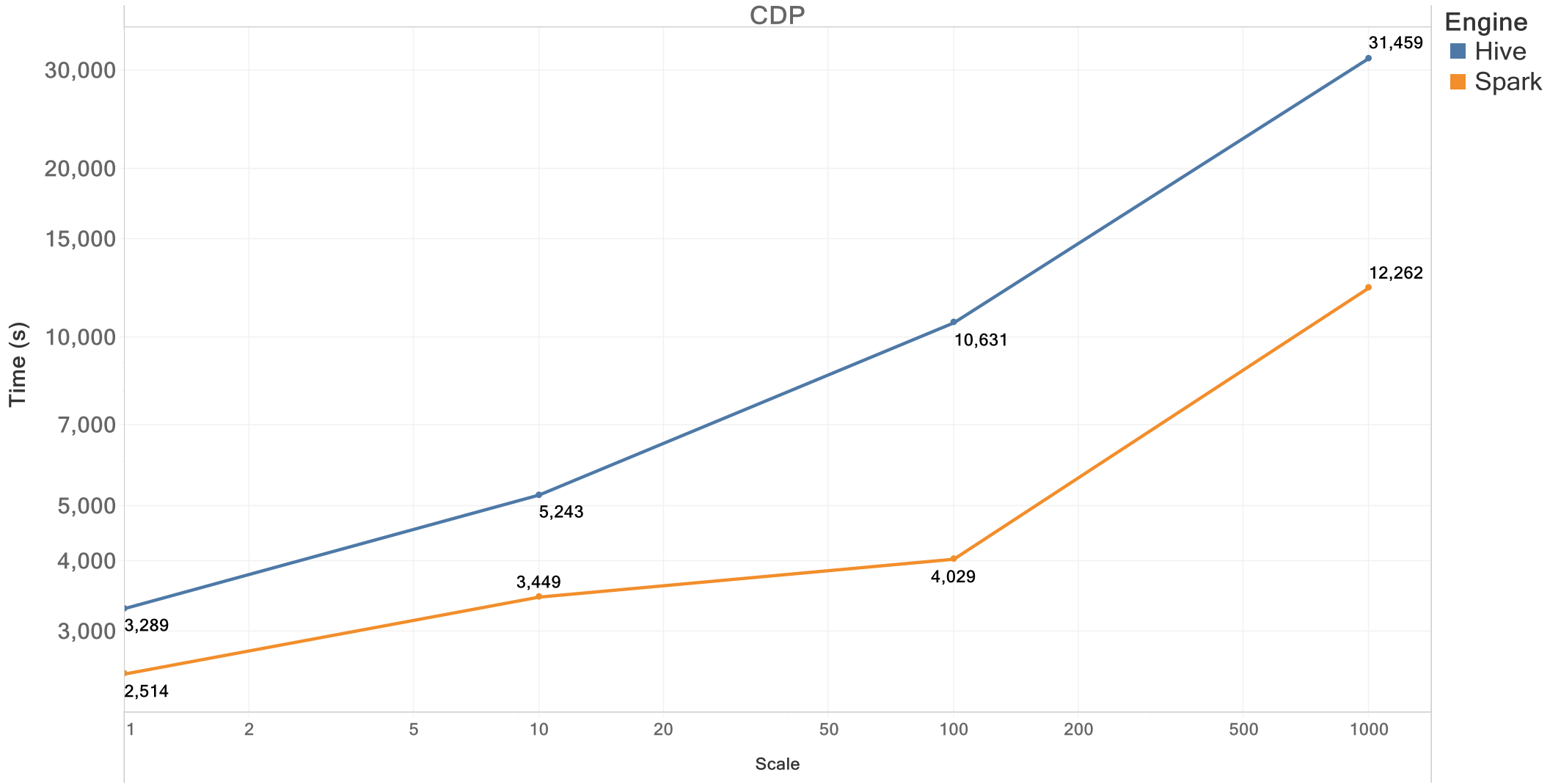
Sequential Hive vs Spark 2.1

Queries 1-30 on Spark 2.1 (power runs)

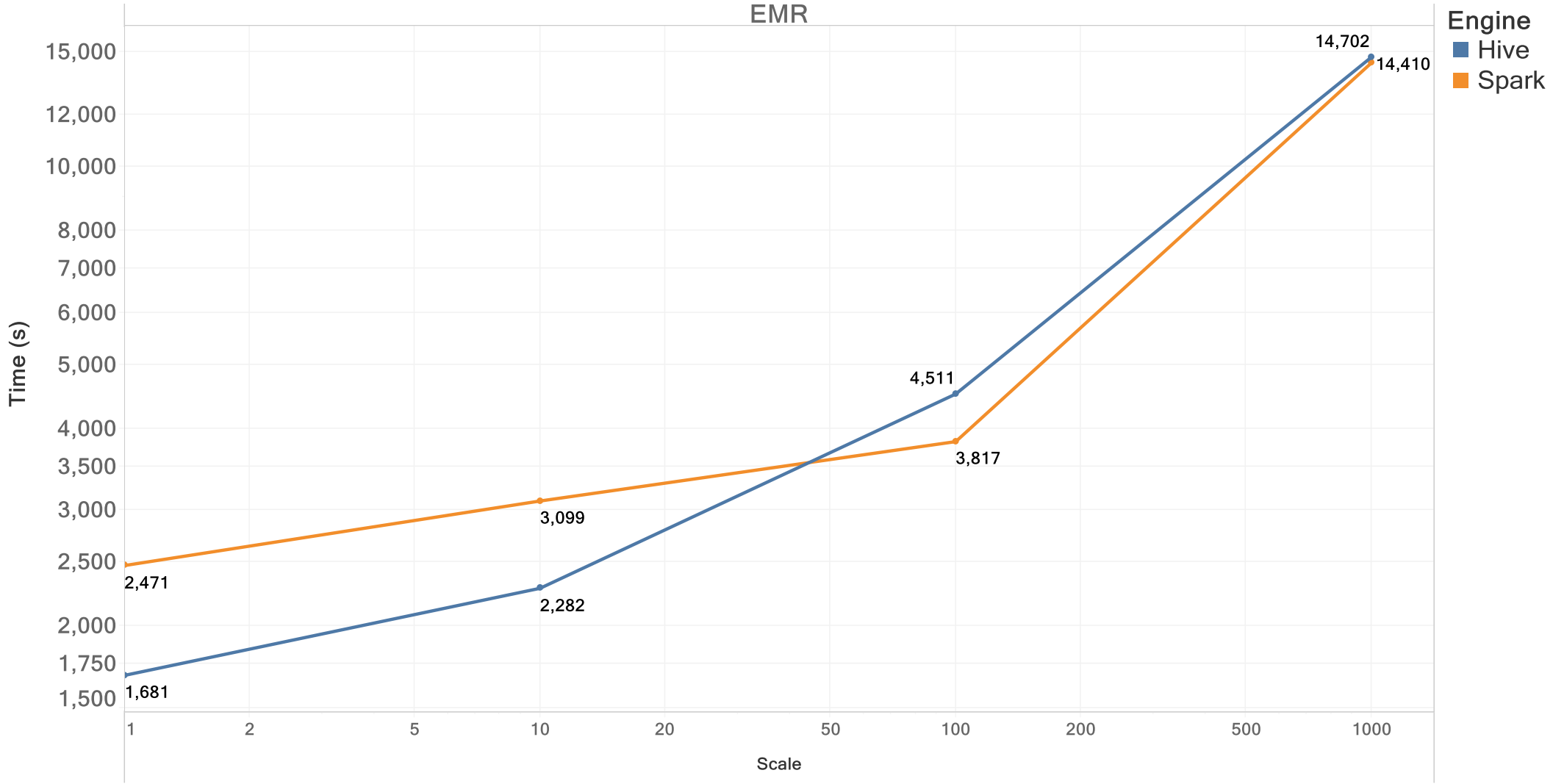


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

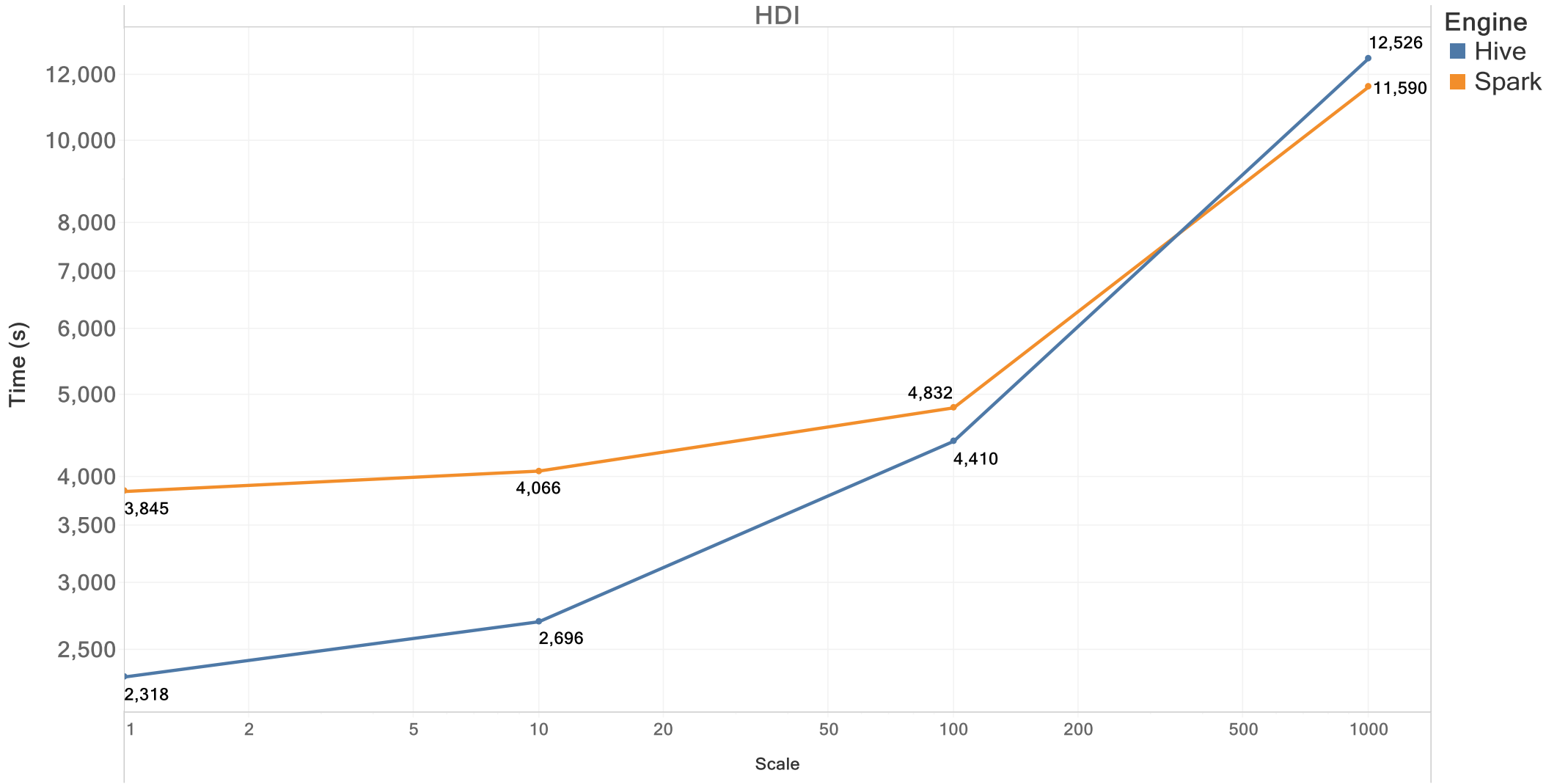
BB 1GB-1TB Scalability Dataproc: Hive 2.1 (M/R) vs Spark 2.1



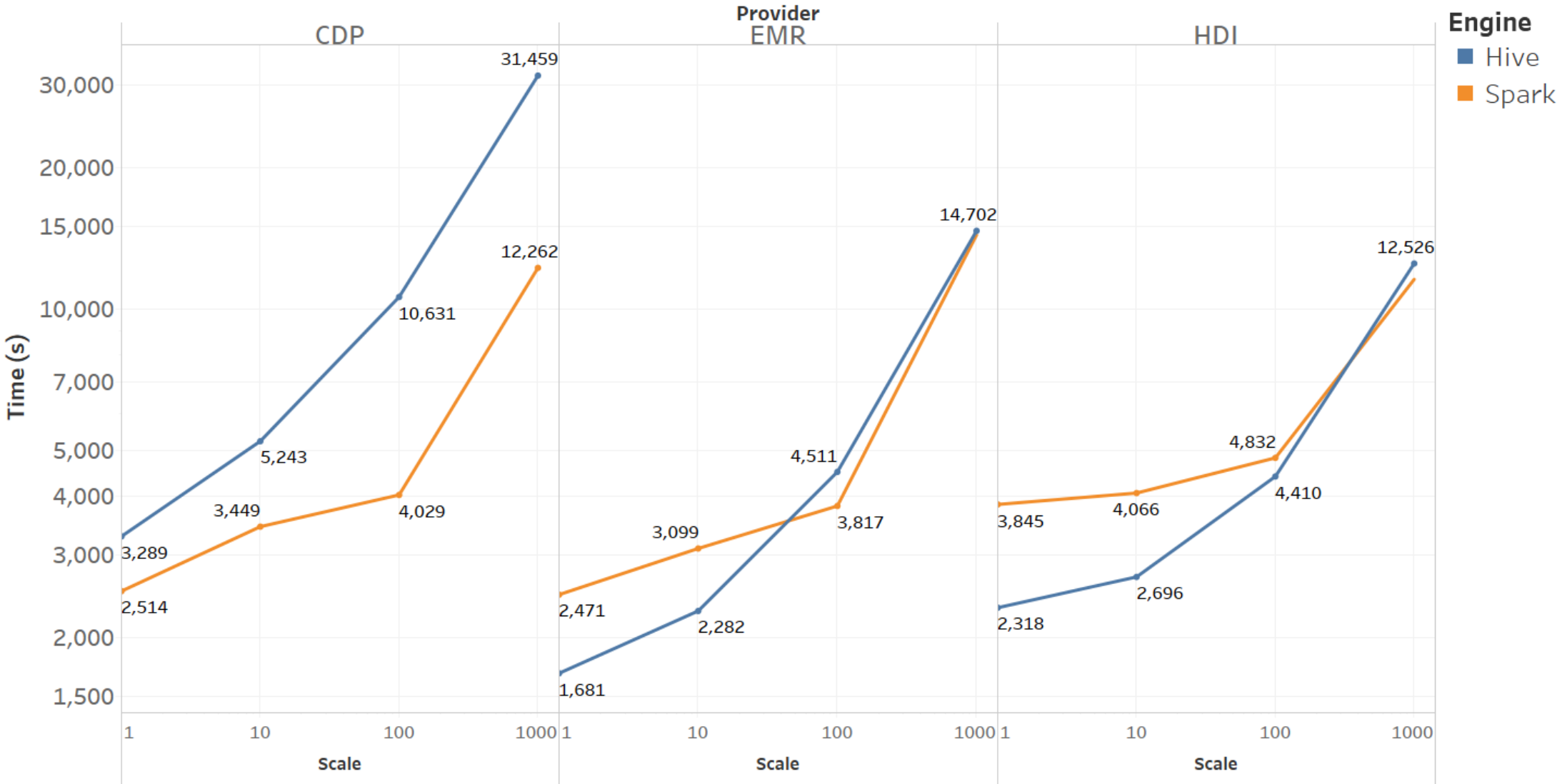
BB 1GB-1TB Scalability EMR: Hive 2.1 (Tez) vs Spark 2.1



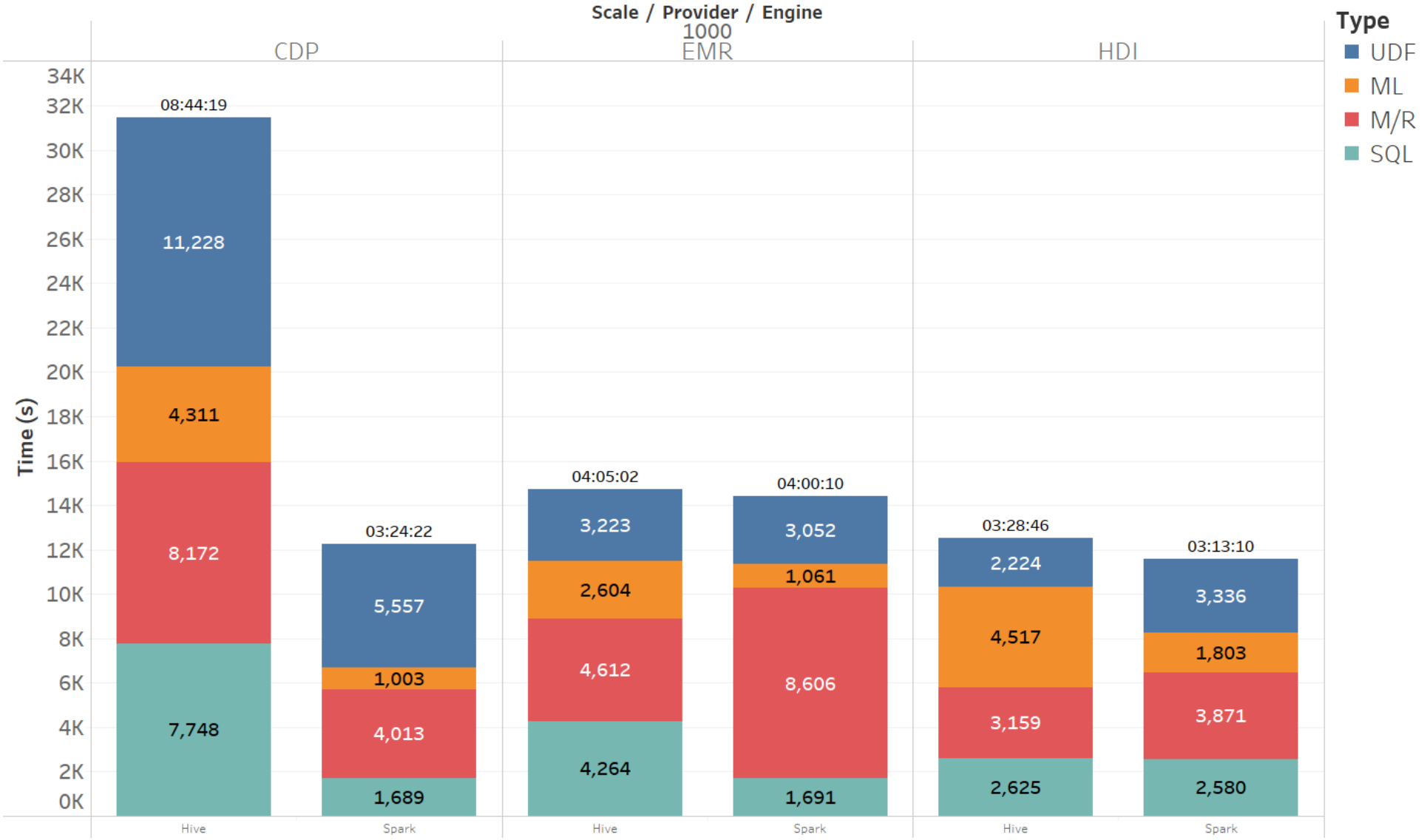
BB 1GB-1TB Scalability HDI: Hive 1.2 (Tez) vs Spark 2.1



BB 1GB-1TB Scalability: Hive vs Spark 2.1
All providers

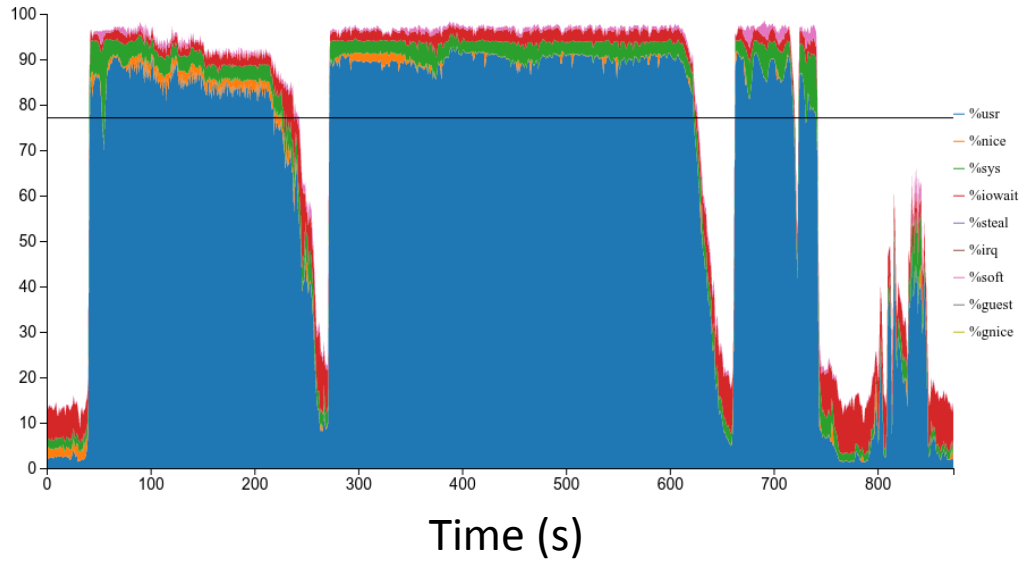


BB 1TB Power runs : Hive vs Spark 2.1 (ALL)



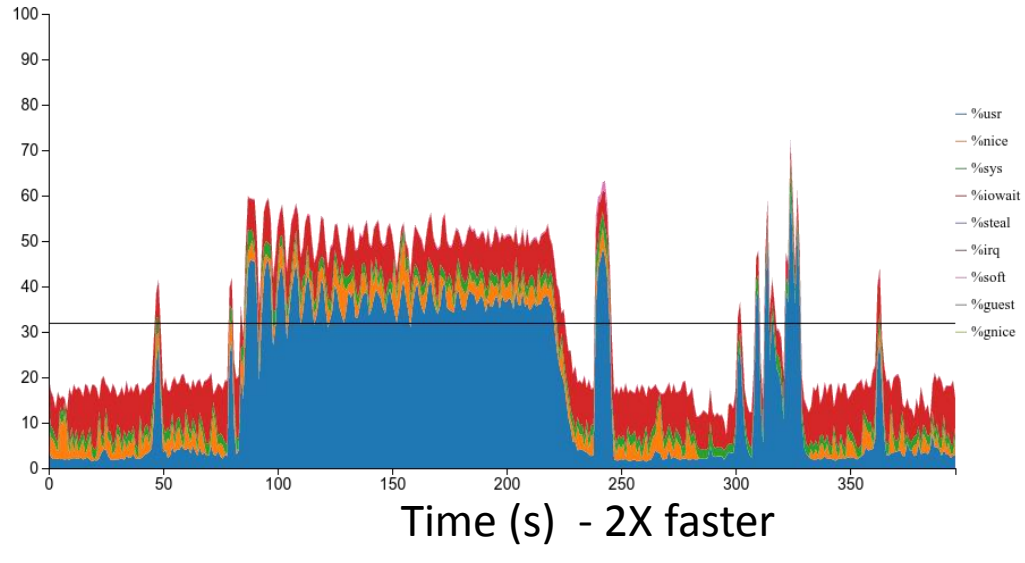
CPU % Q5 (ML) in Hive and Spark (HDI)

- Hive (MLlib2)



Time (s)

- Spark (MLlib2)

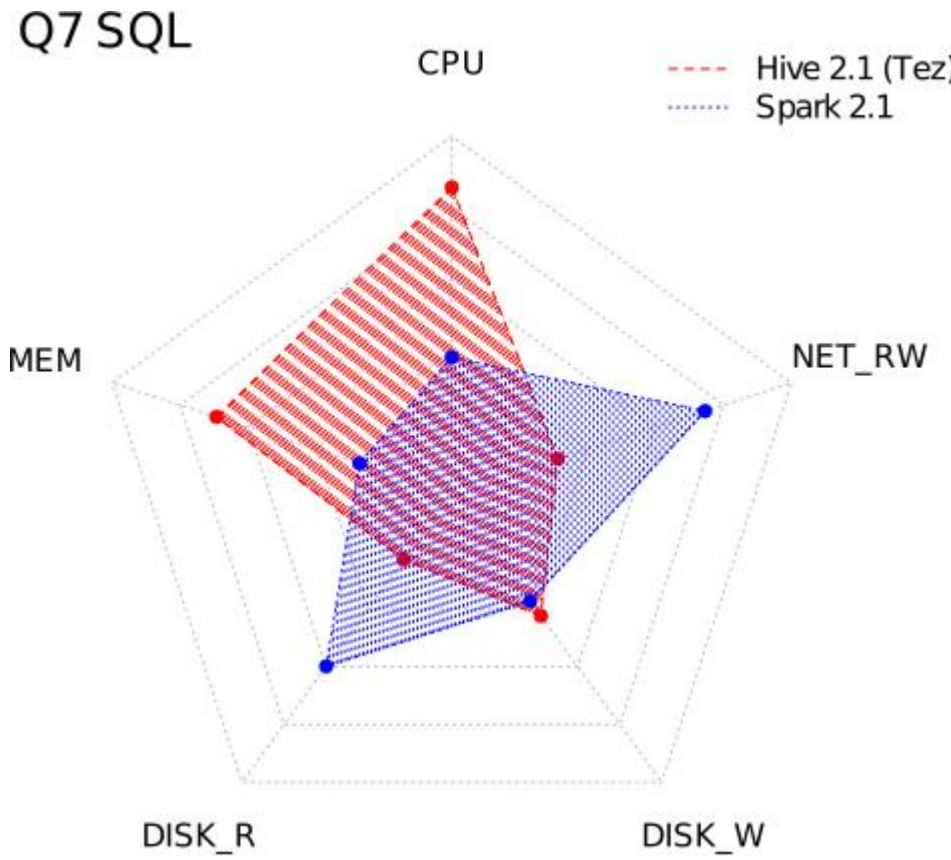


Time (s) - 2X faster

%usr %nice %sys %iowait %soft

Radar charts – query characterization

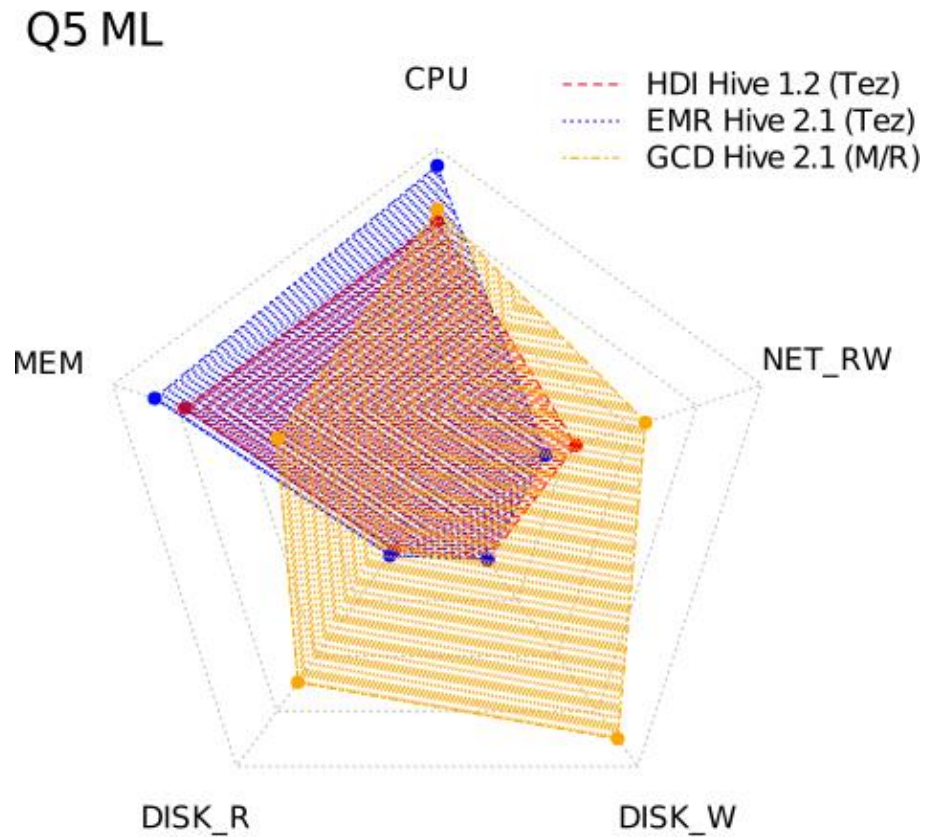
- Useful for **displaying multivariate** data (5 resources)
- Quickly identify similarities and differences.
- From example
  - Hive and Spark
  - Only Disk Write is similar
  - Hive consumes more MEM and CPU
  - Spark read more from disk (DISK_R)
    - And moderately more network



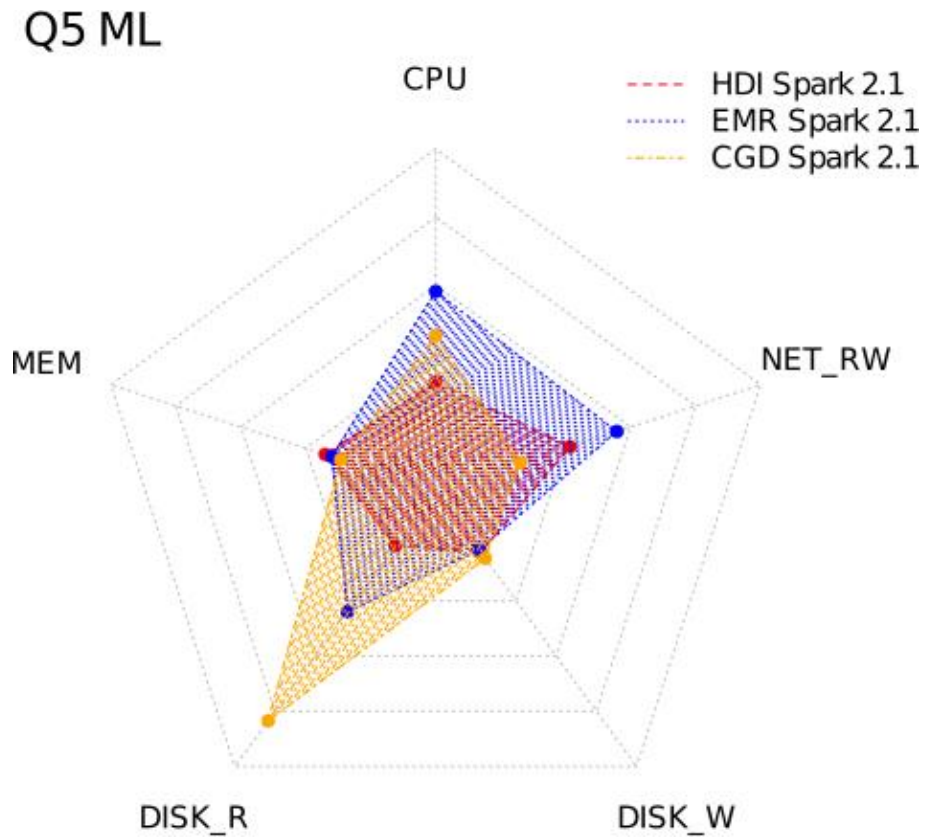
Sample radar chart for Q7 in EMR at 1TB

BB 1TB Query 5 (ML) providers comparison

Hive (MLlib2)



Spark (MLlib2)



Other comparisons:

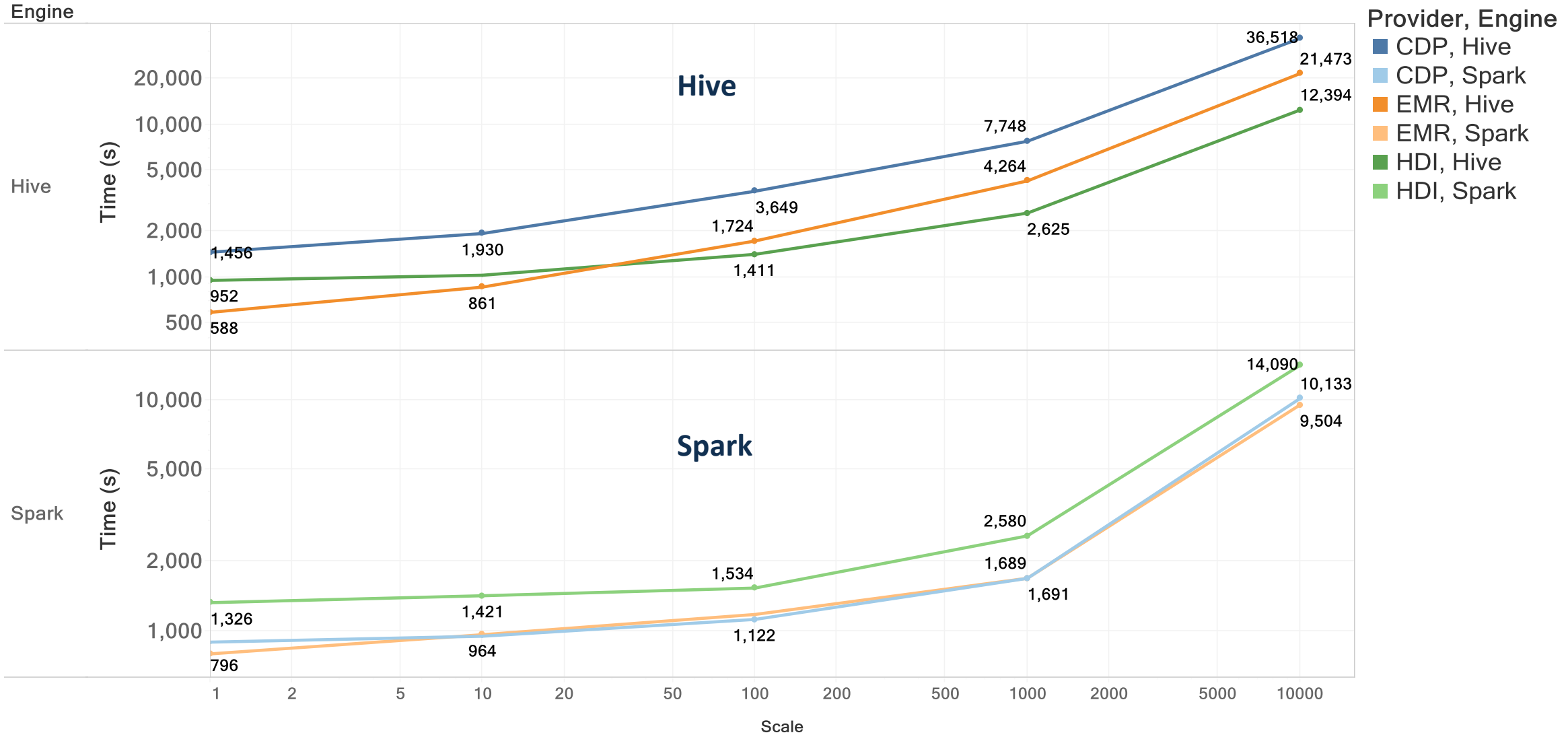
10TB SQL-Only

2.0.2 vs 2.1.0

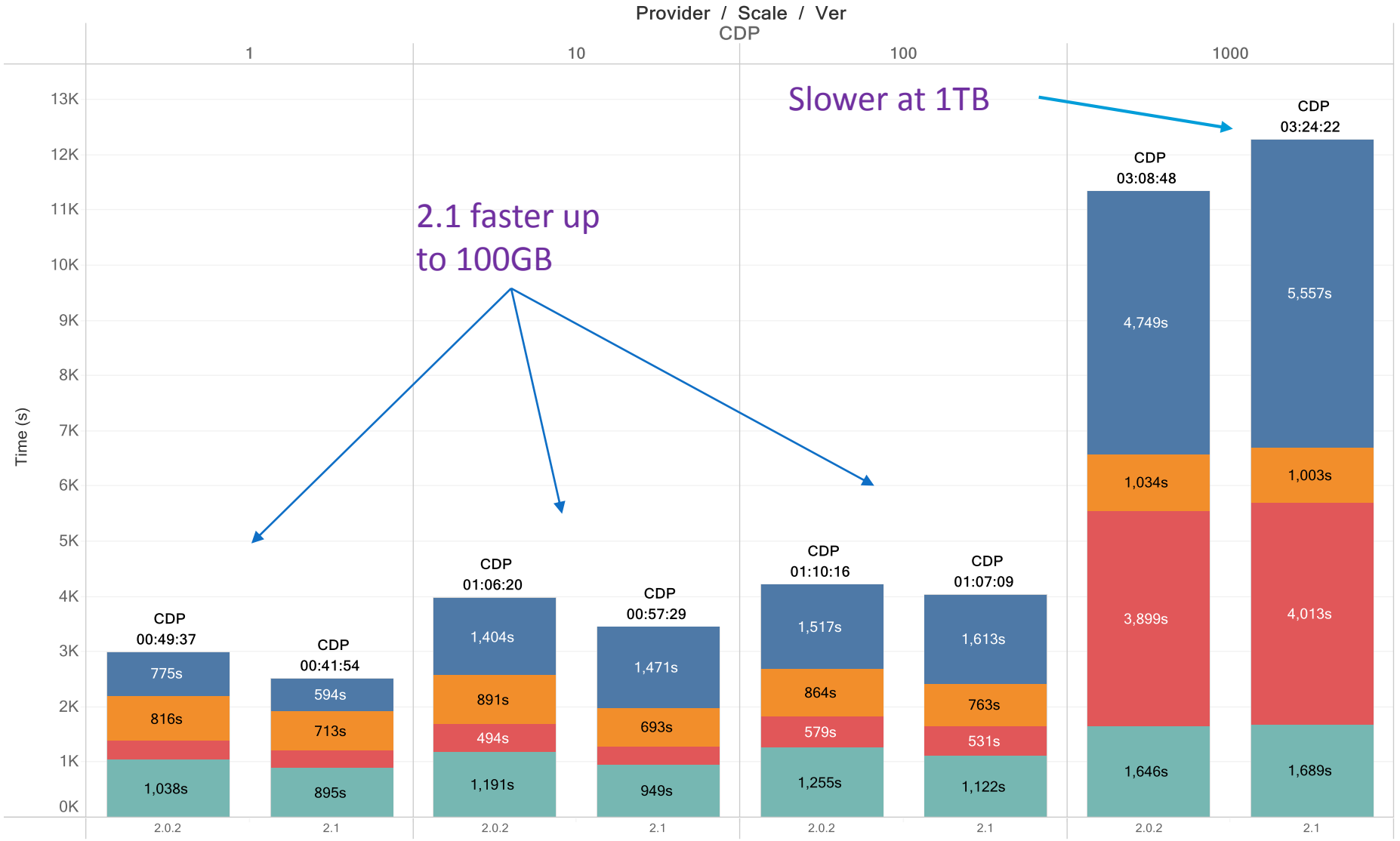
1.6.3 vs 2.1.0

MLlib v1 vs v2

BB 1GB-10TB Scalability SQL-only queries



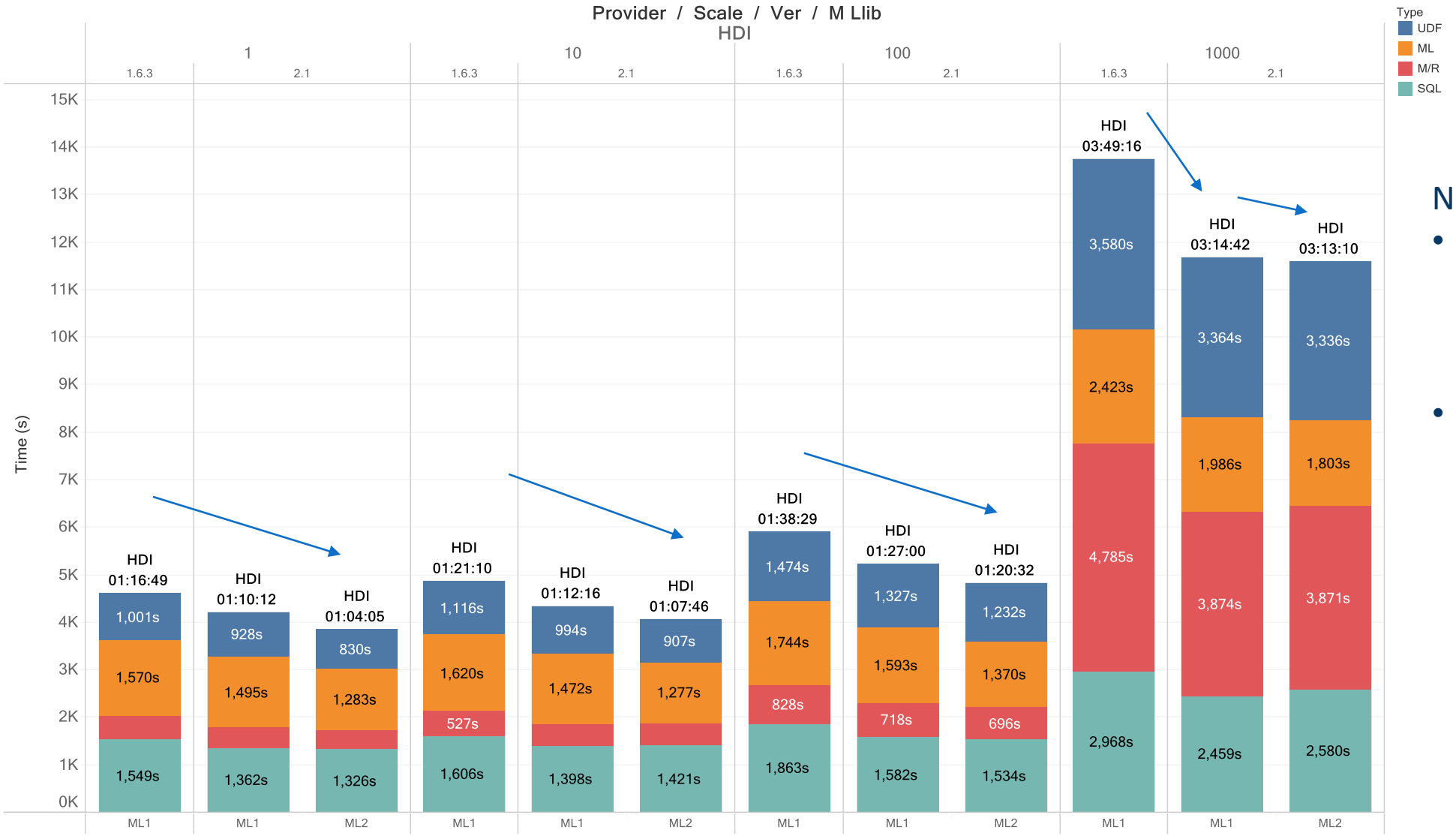
BigBench 1GB-1TB: Spark **2.0.2** vs **2.1.0** (CDP)



Notes:
Spark 2.1 a bit faster at small scales, slower at 100 GB and 1 TB on the UDF/NLP queries

BigBench 1GB-1TB: Spark 1.6.3 vs 2.1.0

MLlib 1 vs 2.1 MLlib 2(HDI)

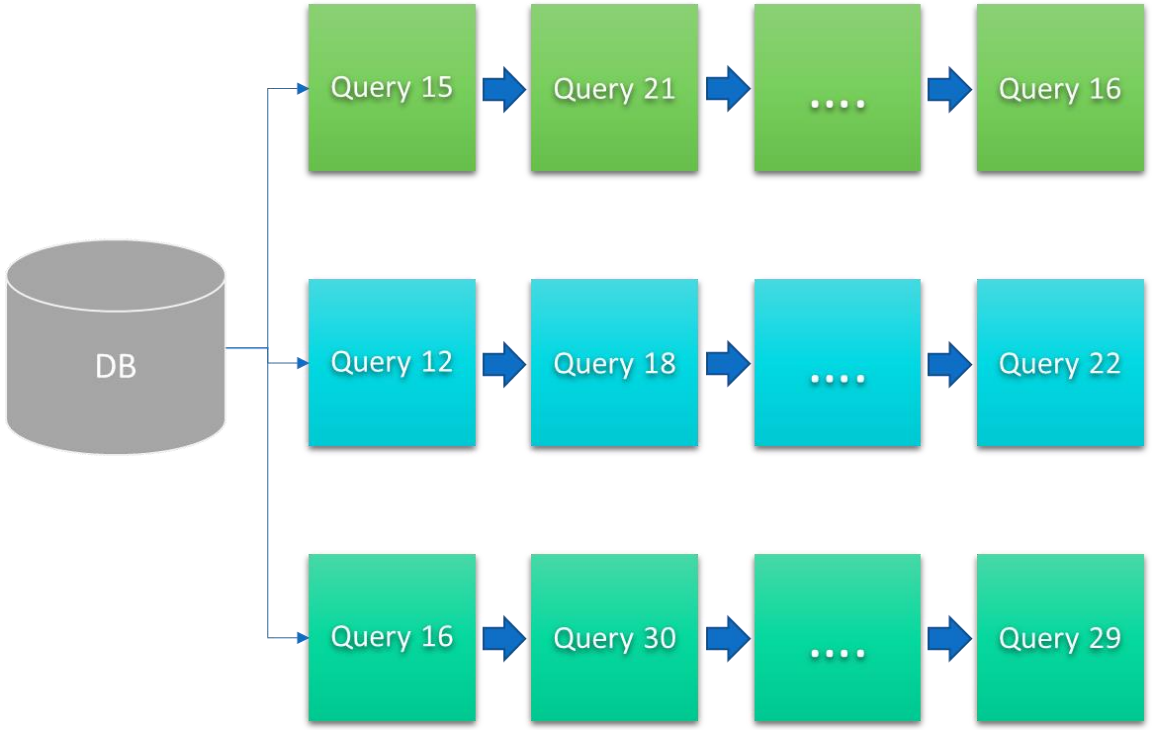


Notes:

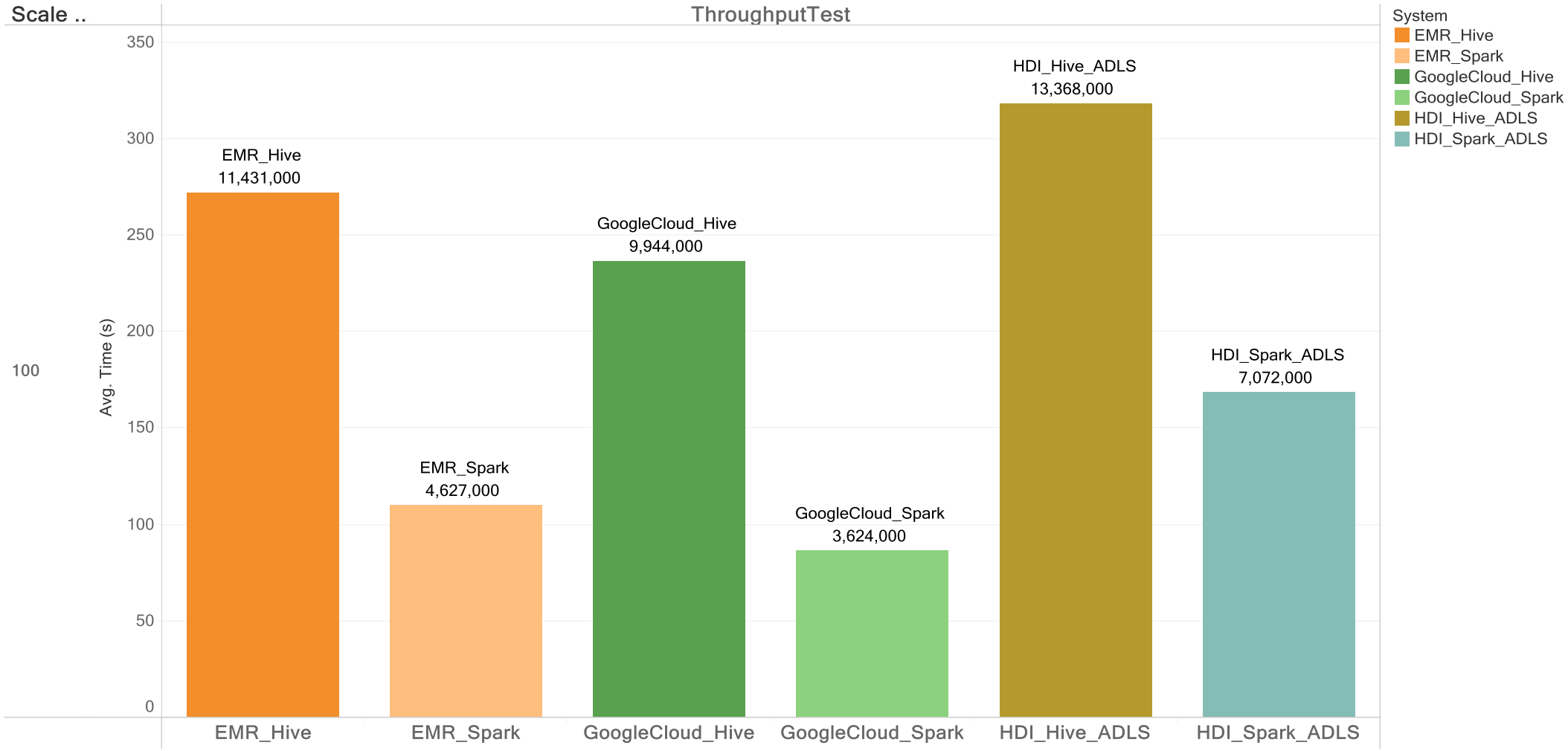
- Spark 2.1 is always faster than 1.6.3 in HDI
- MLlilb 2 using dataframes over RDDs is only slightly faster than V1.

Concurrency runs (throughput)

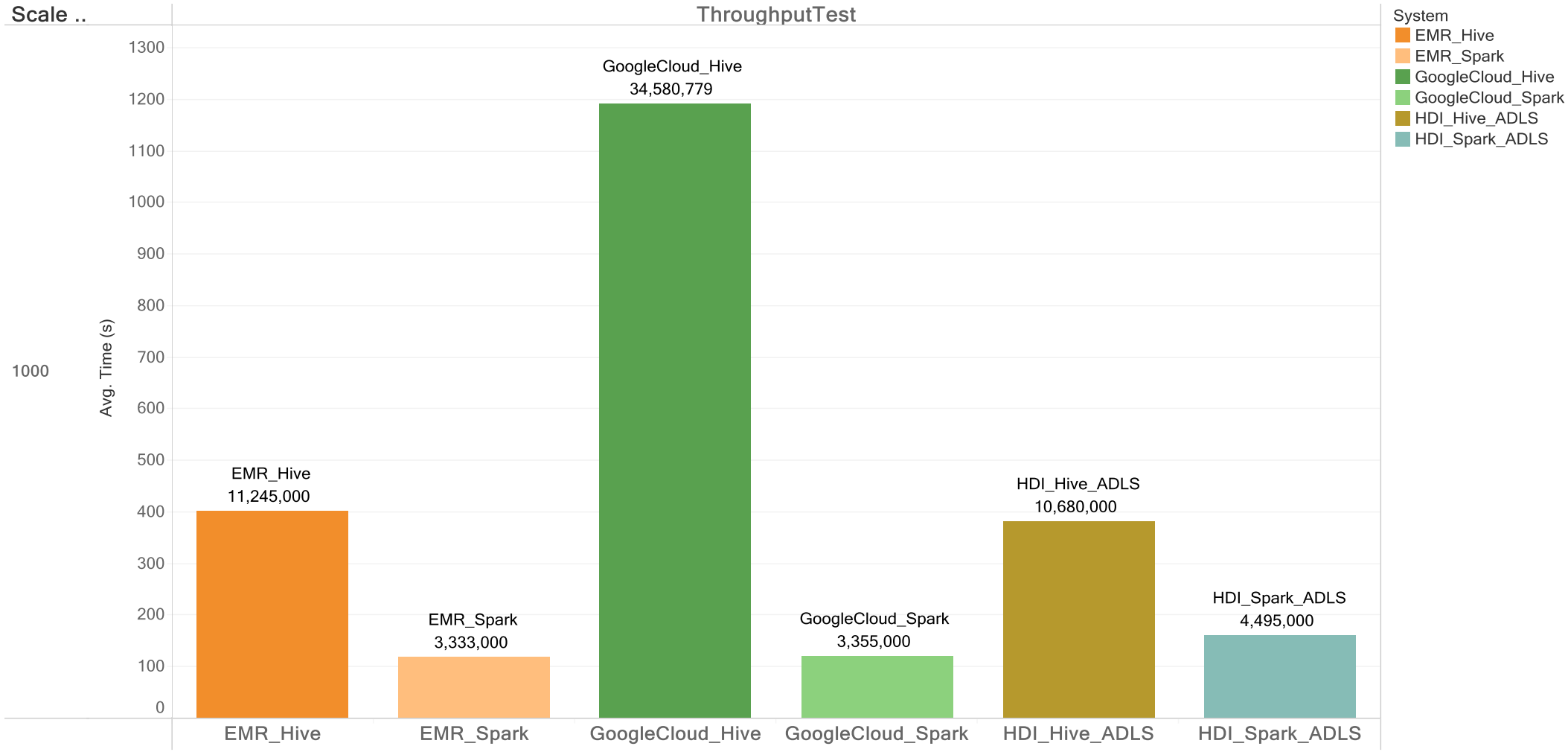
SQL-only: 100GB – 1TB 512-core cluster



BB Throughput at 100GB 8 streams SQL-only (512-cores)



BB Throughput at 1TB 4 streams SQL-only (512-cores)



Conclusions

- All providers have up to date (2.1.0) and well tuned versions of Spark
  - They could run BigBench up to 1TB on medium-sized cluster,
    - [Almost] Out-of-the box
- Performance similar among providers for similar cluster types and disk configs
  - Difference according to scale (and pricing)
- Spark 2.1.0 is faster than previous versions
  - Also MLlib 2 with dataframes
  - But improvements within the 30% range
- Hive (+Tez + MLlib) are still slightly faster than Spark at lower scales for sequential
- But Spark significantly faster at high data scales and concurrency
- BigBench has been useful to stress a cluster with different workloads
  - Highlights config problems fast and stresses scale limits
  - Helpful for tuning the clusters

Resources and references

BigBench and ALOJA

- **BigBench Spark 2 branch** *(thanks Christoph and Michael from bankmark.de)*:
  - https://github.com/carabolic/Big-Data-Benchmark-for-Big-Bench/tree/spark2
- Original BigBench Implementation repository
  - https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench
- ALOJA benchmarking platform
  - https://github.com/Aloja/aloja
  - ALOJA fork of BigBench (adds support for HDI and fixes spark)
    - https://github.com/Aloja/Big-Data-Benchmark-for-Big-Bench

Papers and slides

- https://www.slideshare.net/ni_po
- Characterizing TPCx-BB Queries, Hive, and Spark in Multi-Cloud Environments – N. Poggi et. Al
  - TPC-TC 2017
- The State of SQL-on-Hadoop in the Cloud – N. Poggi et. al.
  - IEEE Big Data 2016
  - https://doi.org/10.1109/BigData.2016.7840751

The state of  in the cloud

Thanks, questions?

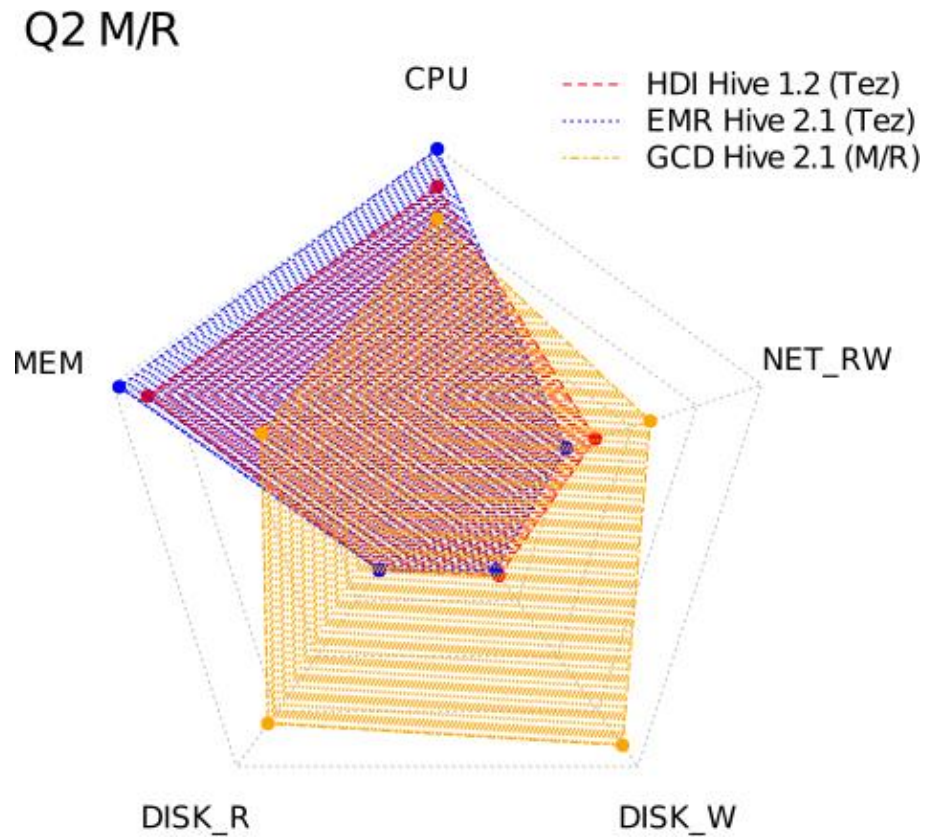
Follow up / feedback : Npoggi@ac.upc.edu

Twitter: ni_po

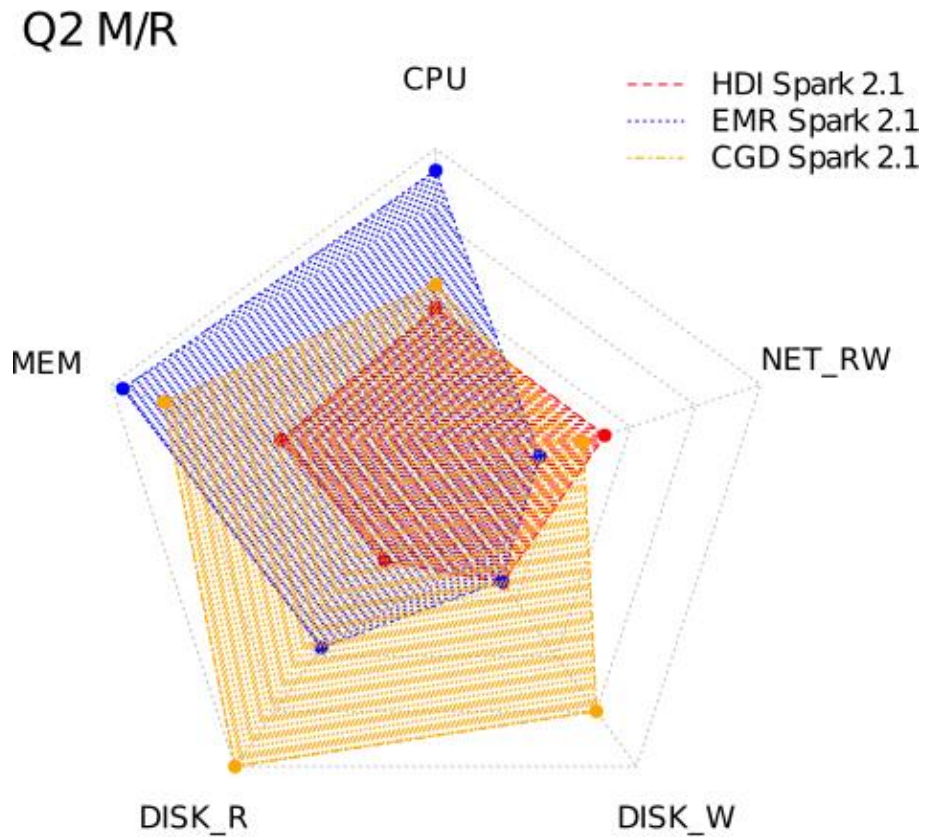
Extra slides

BB 1TB Query 2 (M/R) providers comparison

Hive



Spark



Spark config

	EMR	CDP	HDI
Java version	OpenJDK 1.8.0_121	OpenJDK 1.8.0_121	OpenJDK 1.8.0_131
Spark version	2.1.0	2.1	2.1.0.2.6.0.2-76
Driver memory	5G	5G	5G
Executor memory	5G	**10G**	4G
Executor cores	4	4	**3**
Executor instances	**Dynamic**	**Dynamic**	20
dynamicAllocation enabled	TRUE	TRUE	FALSE
Executor memoryOverhead	Default (384MB)	1,117 MB	Default (384MB)