WISCONSIN

UNIVERSITY OF WISCONSIN–MADISON

CS 839: Design the Next-Generation Database

Lecture 21: Cloud Data Warehousing

Xiangyao Yu

4/7/2020

Announcements

Course project

- Submission deadline: **Apr. 23**
- Peer review: **Apr. 23 – Apr. 30**
- Presentation: **Apr. 28 & 30**
- Camera ready deadline: **May 4**

VLDB format: https://vldb2020.org/formatting-guidelines.html

The final report should be at least 4 pages (excluding references) and up to 12 pages

More details will be announced soon

Discussion Highlights

Cloud storage vs. SmartSSD

- SmartSSD for OLAP while Aurora for OLTP?
- SmartSSD on the read path; Aurora on the write path
- Computation in cloud storage more powerful than SmartSSD
- SmartSSD serves one node while Cloud storage serves multiple nodes
- Cloud storage has higher latency

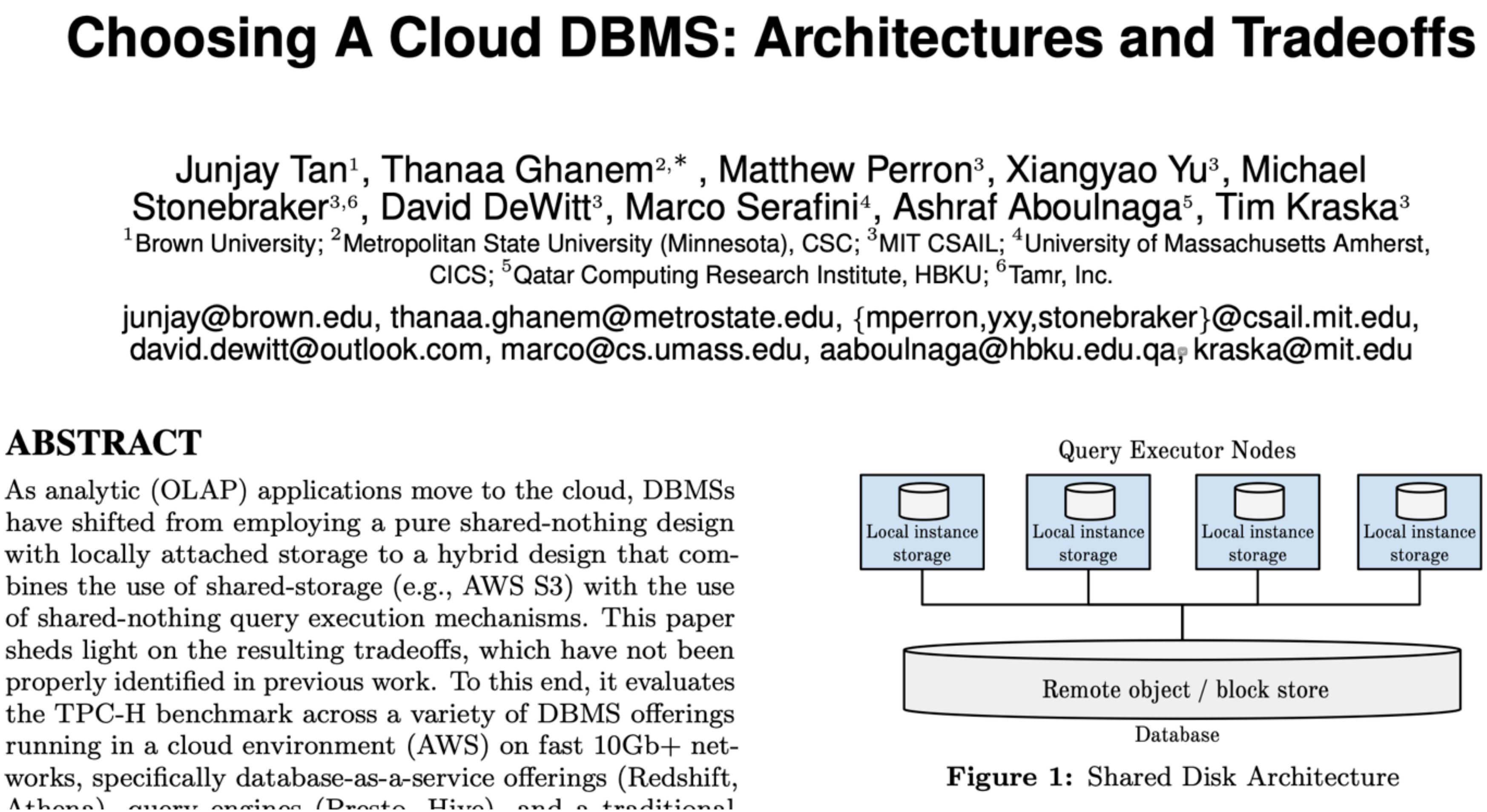
Challenges of multi-master

- Uniqueness of LSN and ordering guarantees
- Concurrency control (locking, leader election)
- Commit protocol (2PC, or 1PC as what Aurora uses)
- Network overhead

Other applications benefit from cloud storage

- Serverless application
- Publish-subscribe system like Apache Kafka
- Graph/Document Store
- Machine learning
- Big data analytics

Today's Paper



Choosing A Cloud DBMS: Architectures and Tradeoffs

Junjay Tan[1], Thanaa Ghanem[2,*] , Matthew Perron[3], Xiangyao Yu[3], Michael Stonebraker[3,6], David DeWitt[3], Marco Serafini[4], Ashraf Aboulnaga[5], Tim Kraska[3]
[1]Brown University; [2]Metropolitan State University (Minnesota), CSC; [3]MIT CSAIL; [4]University of Massachusetts Amherst, CICS; [5]Qatar Computing Research Institute, HBKU; [6]Tamr, Inc.
junjay@brown.edu, thanaa.ghanem@metrostate.edu, {mperron,yxy,stonebraker}@csail.mit.edu, david.dewitt@outlook.com, marco@cs.umass.edu, aaboulnaga@hbku.edu.qa, kraska@mit.edu

ABSTRACT

As analytic (OLAP) applications move to the cloud, DBMSs have shifted from employing a pure shared-nothing design with locally attached storage to a hybrid design that combines the use of shared-storage (e.g., AWS S3) with the use of shared-nothing query execution mechanisms. This paper sheds light on the resulting tradeoffs, which have not been properly identified in previous work. To this end, it evaluates the TPC-H benchmark across a variety of DBMS offerings running in a cloud environment (AWS) on fast 10Gb+ networks, specifically database-as-a-service offerings (Redshift, Athena), query engines (Presto, Hive), and a traditional

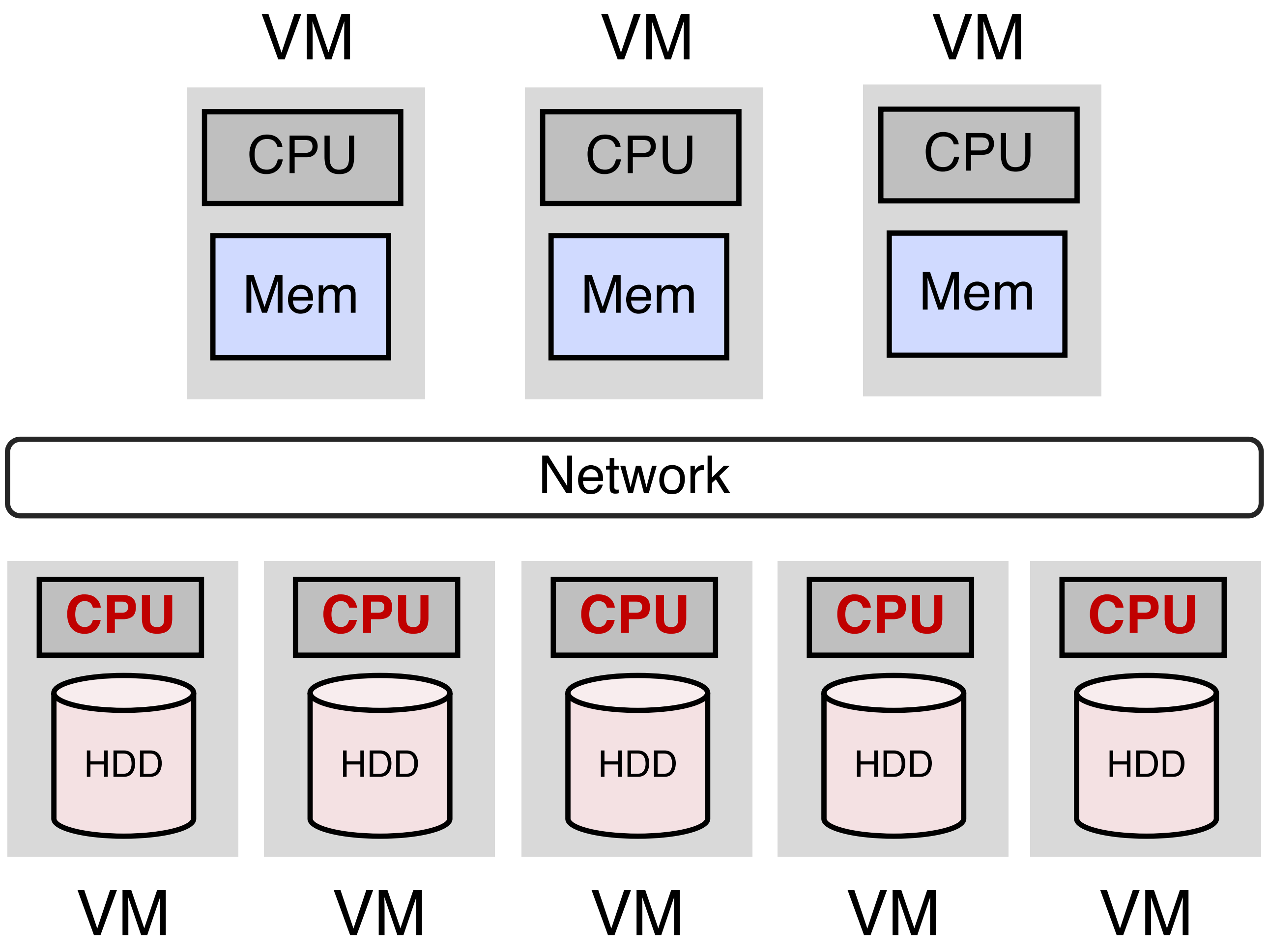
Figure 1: Shared Disk Architecture

VLDB 2019

A paper that costs

- Four students/postdocs
- 1.5 years and
- 30,000+ dollars

Cloud Storage Disaggregation



Storage disaggregation
Smartness in storage

For OLTP

- Aurora: Push logging to storage

For OLAP

- Data in shared storage costs less

Cloud Computing – VM Instances

General purpose

- A1, T3, T3a, T2, M6g, M5, M5a, M5n, M4

Compute optimized

- C5, C5n, C4

Memory optimized

- R5, R5a, R5n, R4, X1e, X1, High memory, z1d

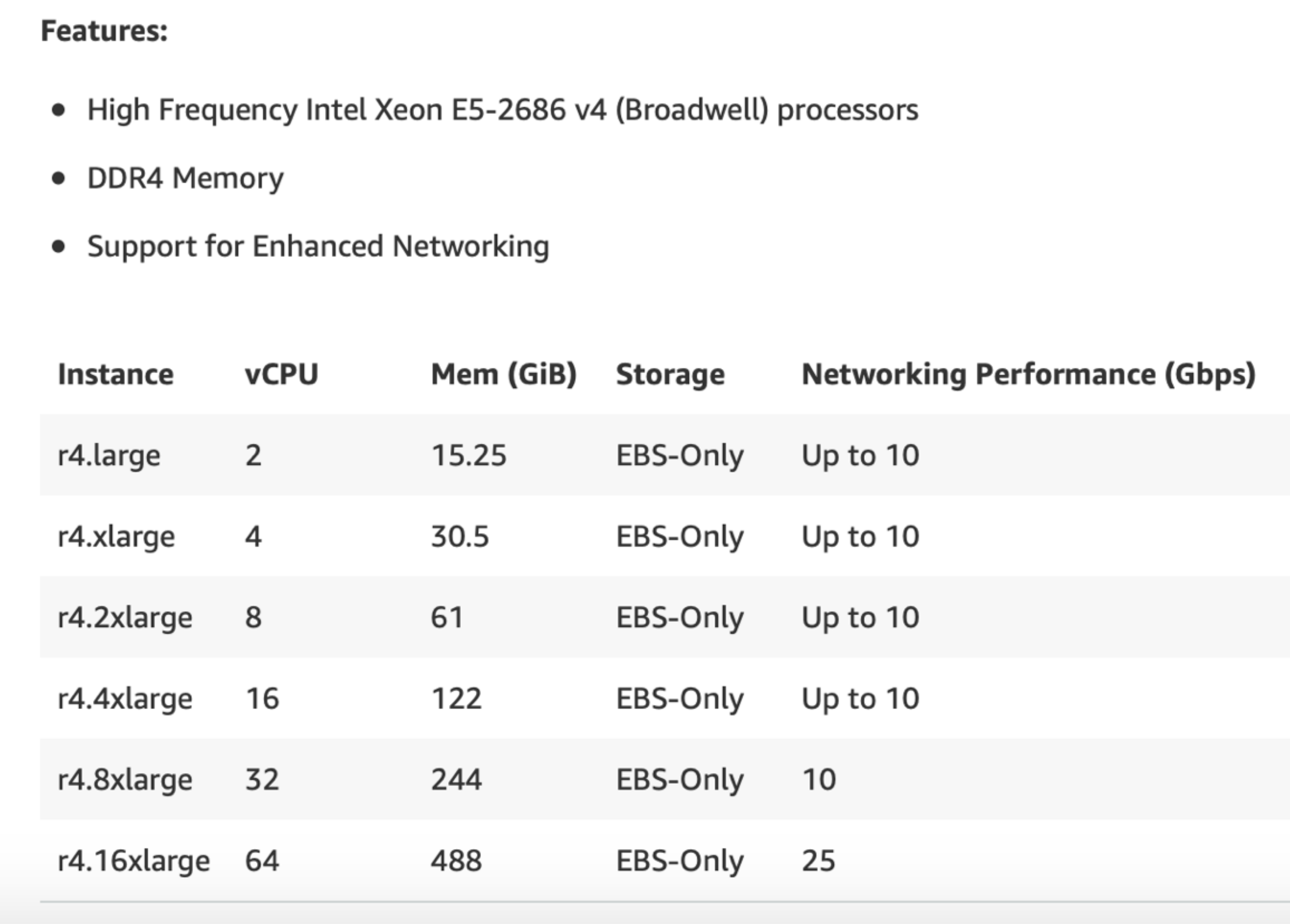
Accelerated computing

- P3, P2, Inf1, G4, G3, F1

Storage optimized

- I3, I3en, D2, H1

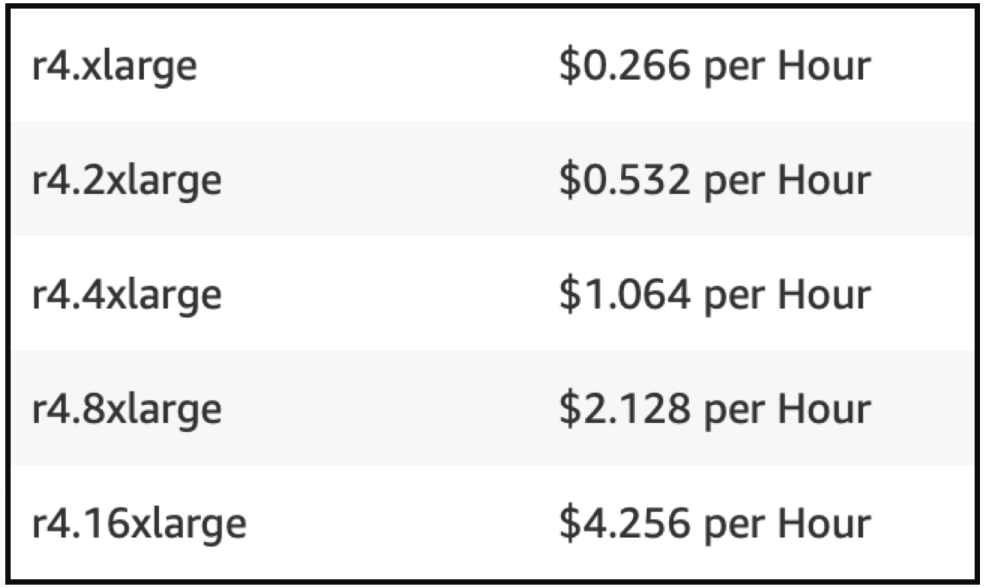
Cloud Computing – R4 Instances



Features:

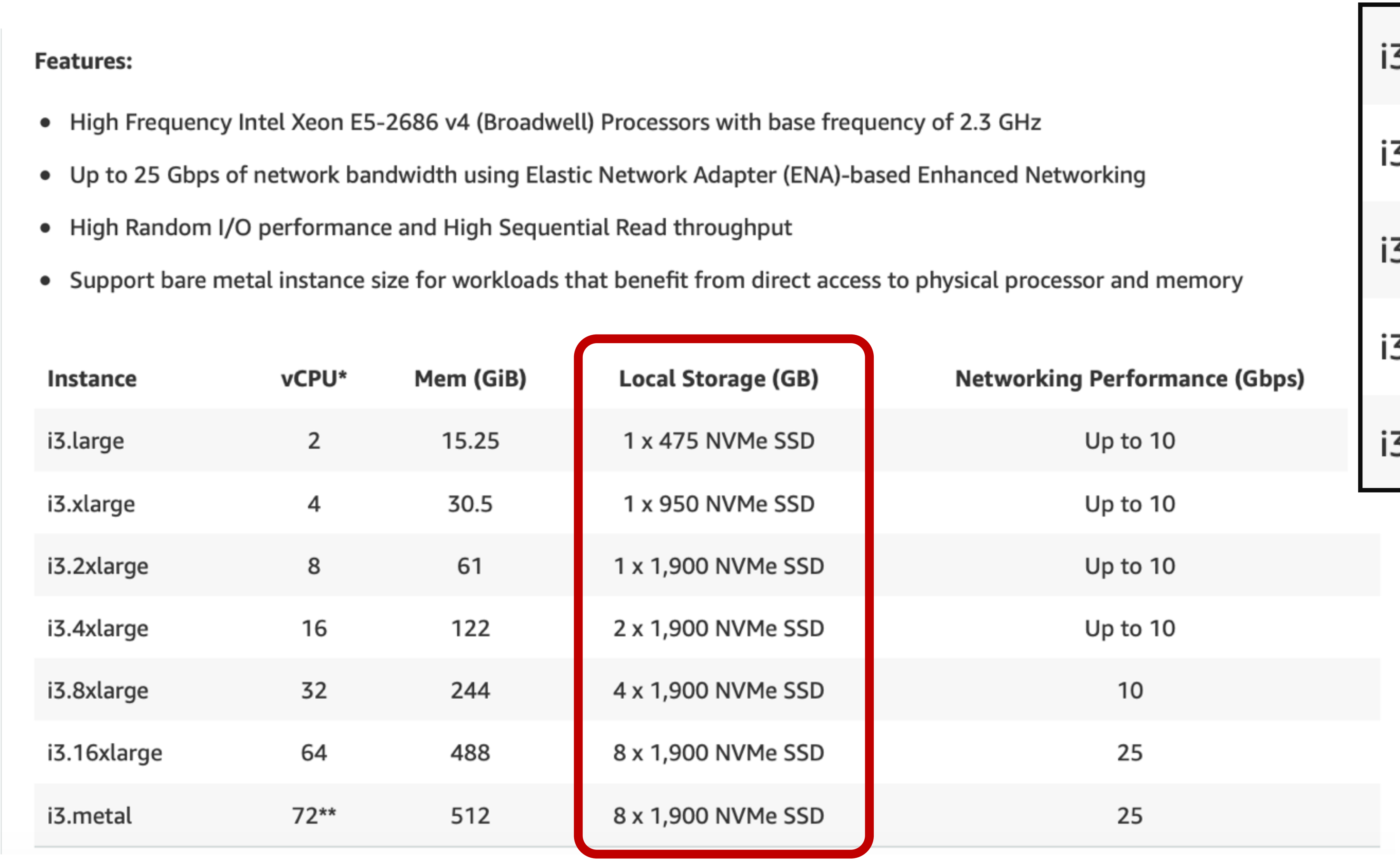
- High Frequency Intel Xeon E5-2686 v4 (Broadwell) processors
- DDR4 Memory
- Support for Enhanced Networking

Instance	vCPU	Mem (GiB)	Storage	Networking Performance (Gbps)
r4.large	2	15.25	EBS-Only	Up to 10
r4.xlarge	4	30.5	EBS-Only	Up to 10
r4.2xlarge	8	61	EBS-Only	Up to 10
r4.4xlarge	16	122	EBS-Only	Up to 10
r4.8xlarge	32	244	EBS-Only	10
r4.16xlarge	64	488	EBS-Only	25



r4.xlarge	$0.266 per Hour
r4.2xlarge	$0.532 per Hour
r4.4xlarge	$1.064 per Hour
r4.8xlarge	$2.128 per Hour
r4.16xlarge	$4.256 per Hour

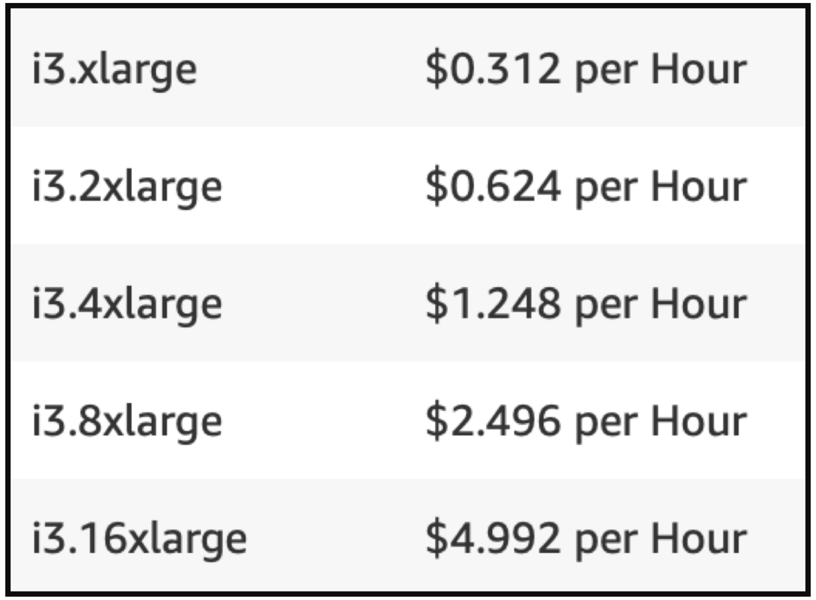
Cloud Computing – I3 Instances



Features:

- High Frequency Intel Xeon E5-2686 v4 (Broadwell) Processors with base frequency of 2.3 GHz
- Up to 25 Gbps of network bandwidth using Elastic Network Adapter (ENA)-based Enhanced Networking
- High Random I/O performance and High Sequential Read throughput
- Support bare metal instance size for workloads that benefit from direct access to physical processor and memory

Instance	vCPU*	Mem (GiB)	Local Storage (GB)	Networking Performance (Gbps)
i3.large	2	15.25	1 x 475 NVMe SSD	Up to 10
i3.xlarge	4	30.5	1 x 950 NVMe SSD	Up to 10
i3.2xlarge	8	61	1 x 1,900 NVMe SSD	Up to 10
i3.4xlarge	16	122	2 x 1,900 NVMe SSD	Up to 10
i3.8xlarge	32	244	4 x 1,900 NVMe SSD	10
i3.16xlarge	64	488	8 x 1,900 NVMe SSD	25
i3.metal	72**	512	8 x 1,900 NVMe SSD	25



i3.xlarge	$0.312 per Hour
i3.2xlarge	$0.624 per Hour
i3.4xlarge	$1.248 per Hour
i3.8xlarge	$2.496 per Hour
i3.16xlarge	$4.992 per Hour

Up to 7.5 GB/s storage bandwidth

Systems Tested

Database-as-a-Service (DBaaS)

- Redshift
- Redshift Spectrum
- Athena

Query engines

- Presto
- Apache Hive

Cloud agnostic OLTP DBMS

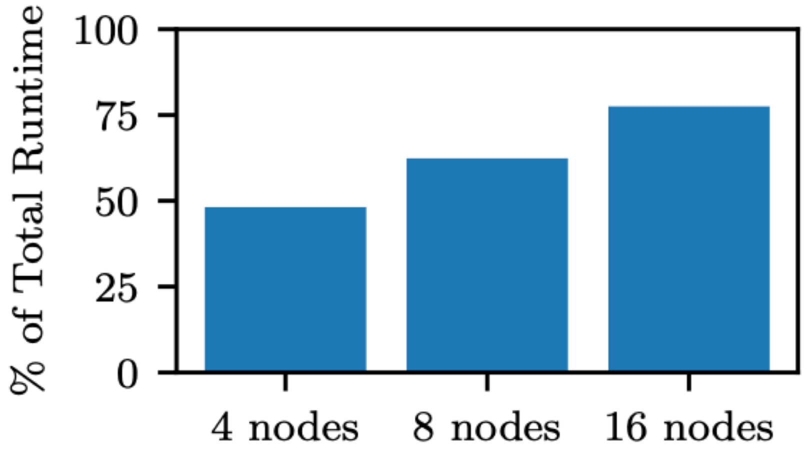
- Vertica

Redshift

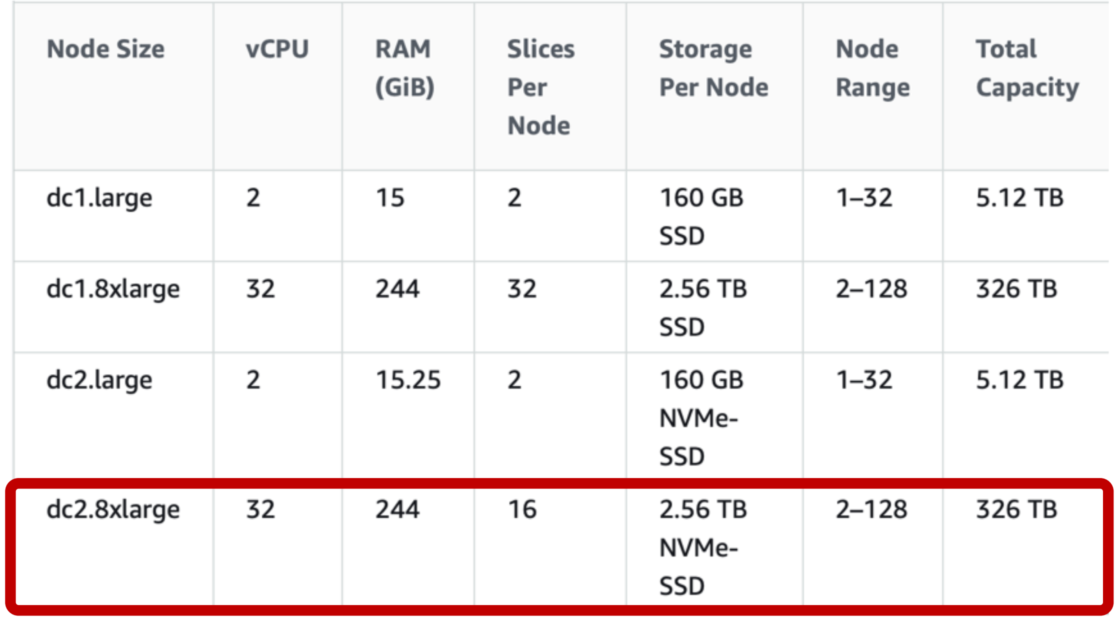
Highly-optimized shared-nothing architecture

Query compilation

Limited instance types



Redshift Query Compilation Time
(% of total runtime)

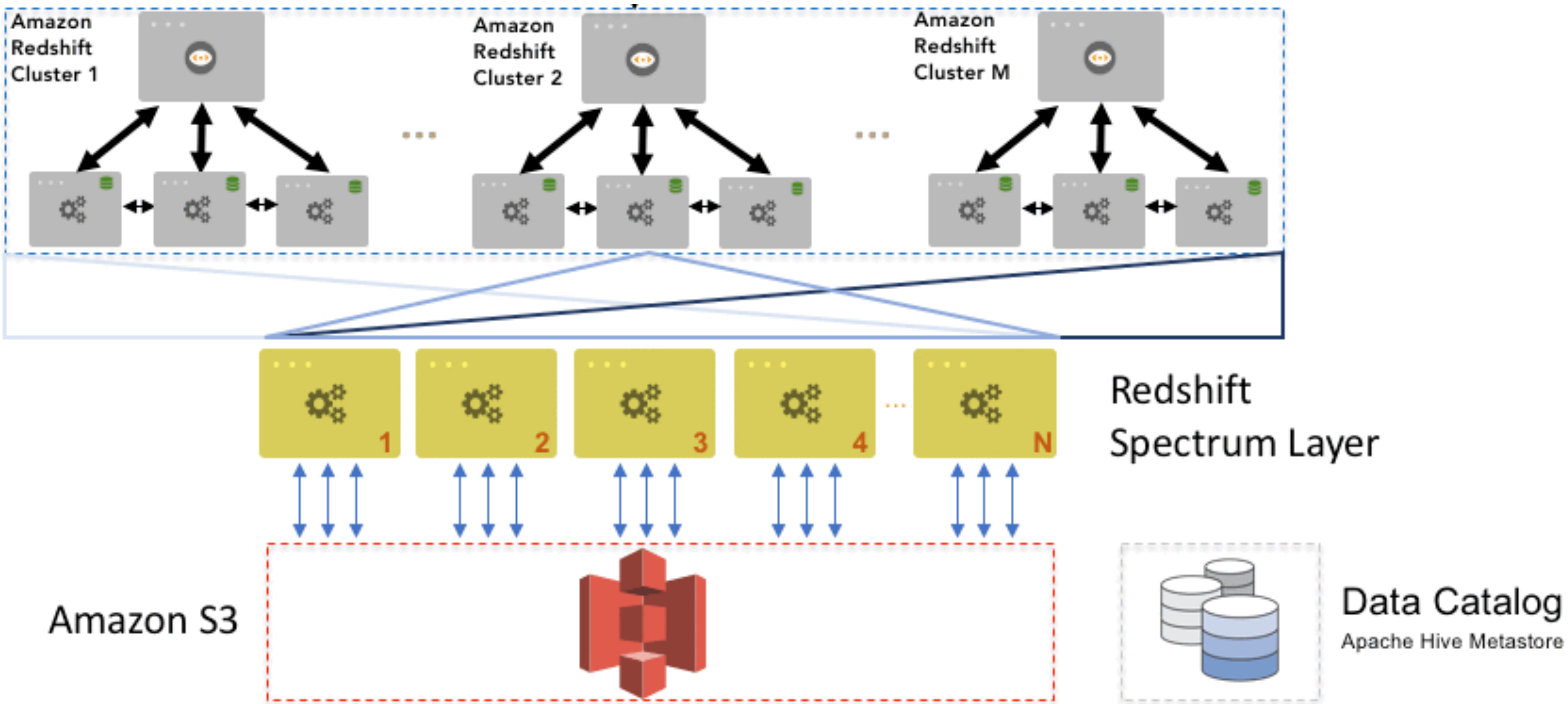


Node Size	vCPU	RAM (GiB)	Slices Per Node	Storage Per Node	Node Range	Total Capacity
dc1.large	2	15	2	160 GB SSD	1–32	5.12 TB
dc1.8xlarge	32	244	32	2.56 TB SSD	2–128	326 TB
dc2.large	2	15.25	2	160 GB NVMe-SSD	1–32	5.12 TB
dc2.8xlarge	32	244	16	2.56 TB NVMe-SSD	2–128	326 TB

i3.8xlarge: $2.496 per hour
dc2.8xlarge: $4.8 per hour

= i3 instance

Redshift Spectrum

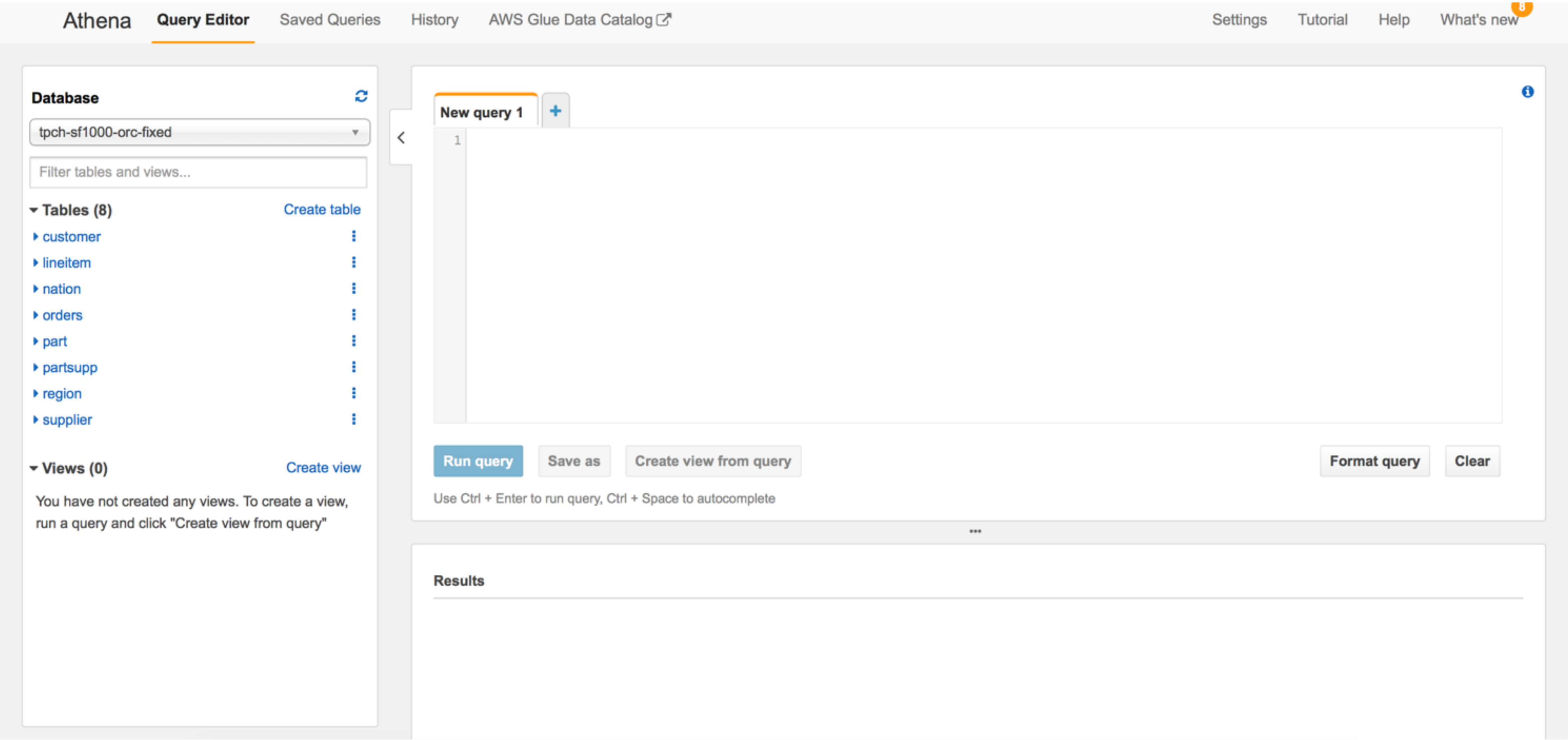


Spectrum layer

- Independent scaling, shared across redshift clusters
- Computation pushdown (e.g., predicate filtering, aggregation)

Cost = Redshift cluster cost + $5 per TB scanned from S3

Athena (Serverless)



Database: tpch-sf1000-orc-fixed

Tables (8)

- customer
- lineitem
- nation
- orders
- part
- partsupp
- region
- supplier

Views (0)

You have not created any views. To create a view, run a query and click "Create view from query"

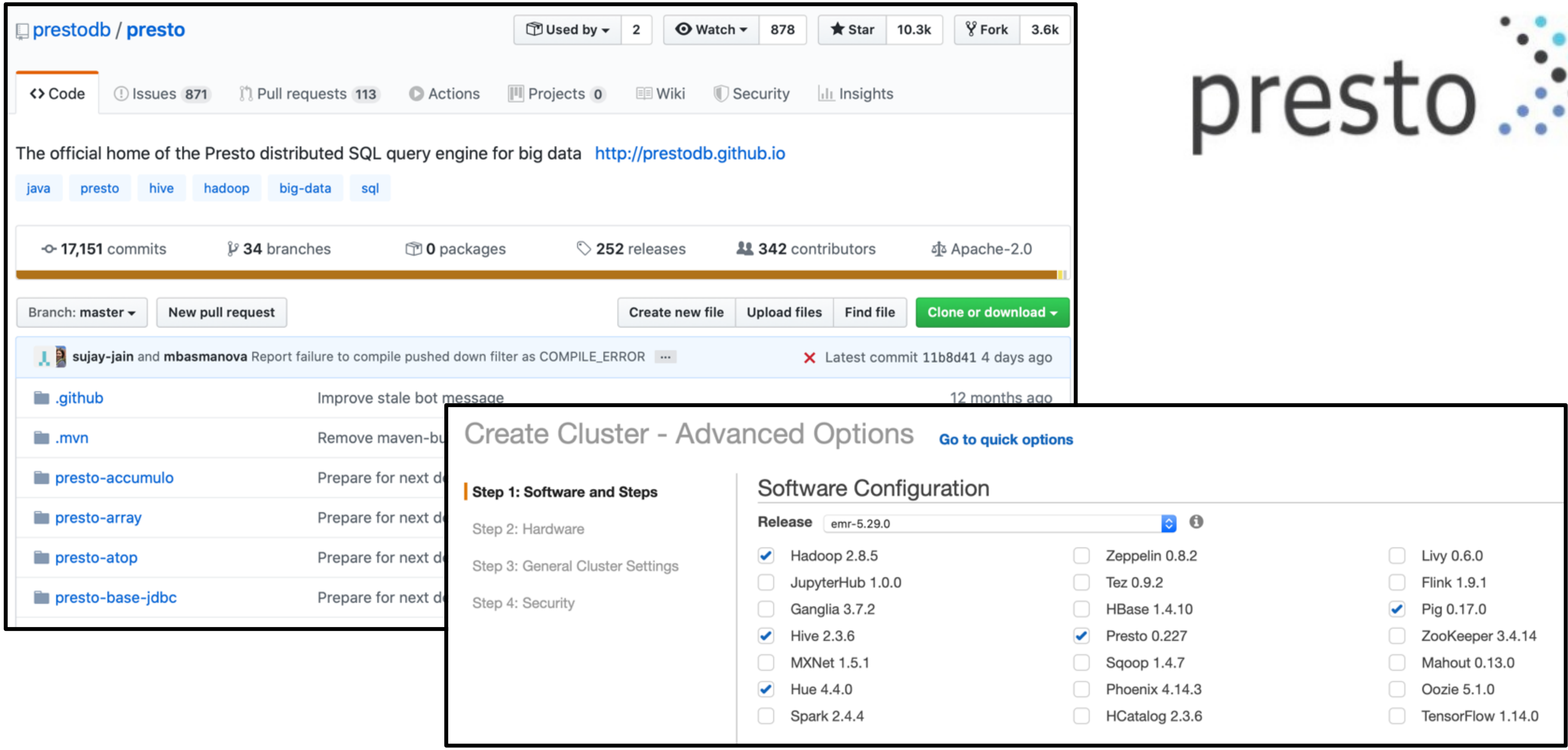
Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Automatically adjusts the type and number of nodes

Charges by the amount of S3 data scanned ($5 per TB scanned)

Presto

Originally developed at Facebook, open sourced in 2013



prestodb / presto

The official home of the Presto distributed SQL query engine for big data http://prestodb.github.io

java presto hive hadoop big-data sql

17,151 commits · 34 branches · 0 packages · 252 releases · 342 contributors · Apache-2.0

sujay-jain and mbasmanova Report failure to compile pushed down filter as COMPILE_ERROR … ✕ Latest commit 11b8d41 4 days ago

.github	Improve stale bot message	12 months ago
.mvn	Remove maven-bu	
presto-accumulo	Prepare for next d	
presto-array	Prepare for next d	
presto-atop	Prepare for next d	
presto-base-jdbc	Prepare for next d	

Create Cluster - Advanced Options Go to quick options

Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

Software Configuration

Release emr-5.29.0

☑ Hadoop 2.8.5
☐ JupyterHub 1.0.0
☐ Ganglia 3.7.2
☑ Hive 2.3.6
☐ MXNet 1.5.1
☑ Hue 4.4.0
☐ Spark 2.4.4
☐ Zeppelin 0.8.2
☐ Tez 0.9.2
☐ HBase 1.4.10
☑ Presto 0.227
☐ Sqoop 1.4.7
☐ Phoenix 4.14.3
☐ HCatalog 2.3.6
☐ Livy 0.6.0
☐ Flink 1.9.1
☑ Pig 0.17.0
☐ ZooKeeper 3.4.14
☐ Mahout 0.13.0
☐ Oozie 5.1.0
☐ TensorFlow 1.14.0

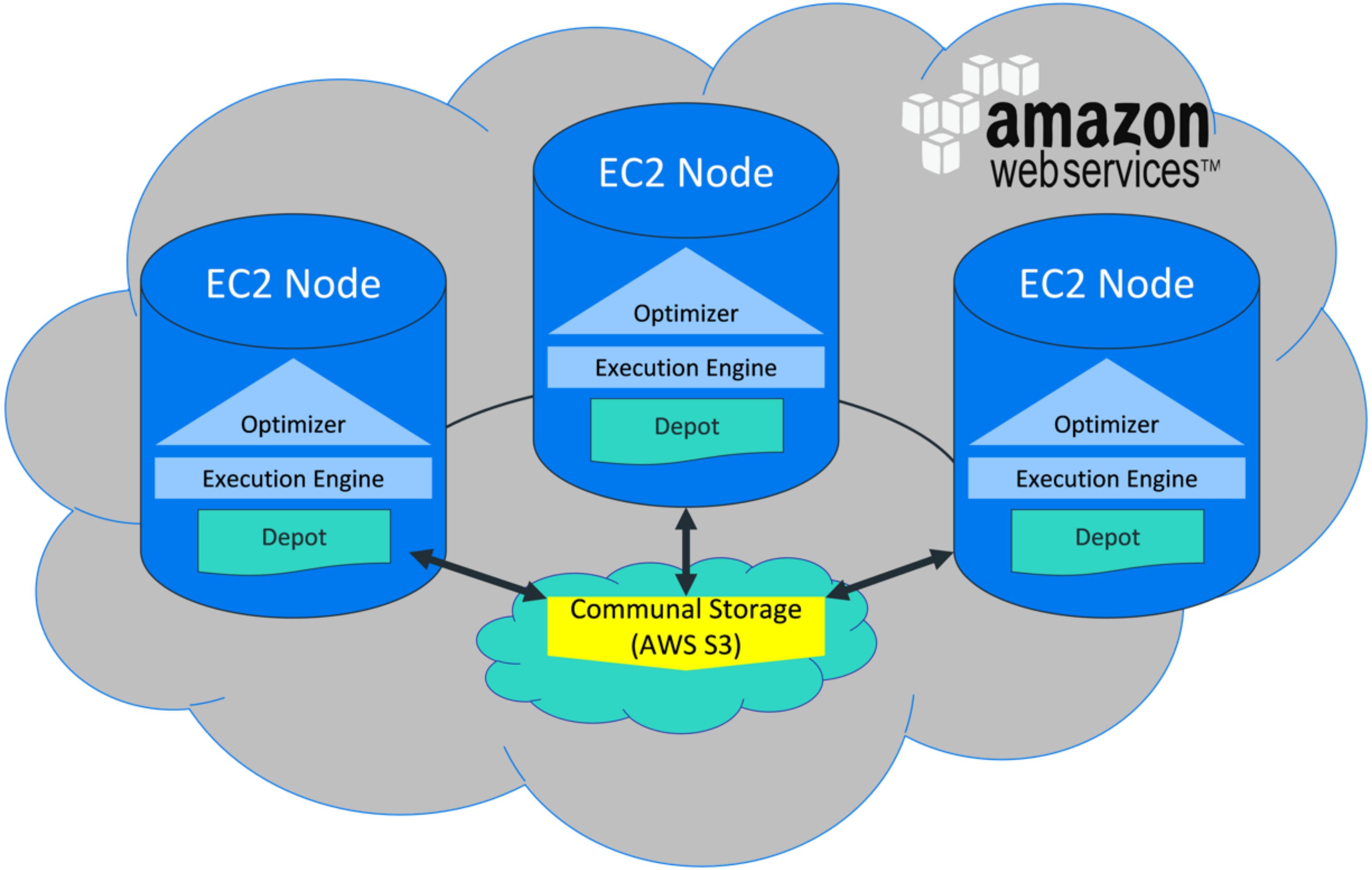
Apache Hive

Originally built on top of Hadoop but now bypass the Hadoop execution engine

Storage: HDFS (on EBS or instance store), S3

Run through Hortonworks data platform

Vertica



Vertica with Eon mode can store data in the cloud

Summary

	DBMS	Compute node	Storage
DB as a Service (DBaaS)	Redshift	dc2.8xlarge	InS
	Redshift Spectrum	dc2.8xlarge	InS/S3
	Athena	Unknown	S3
Query engine	Presto	r4.8xlarge	S3
	Hive	r4.8xlarge	S3/HDFS
Cloud agnostic OLAP	Vertica	r4.8xlarge	EBS/InS/S3

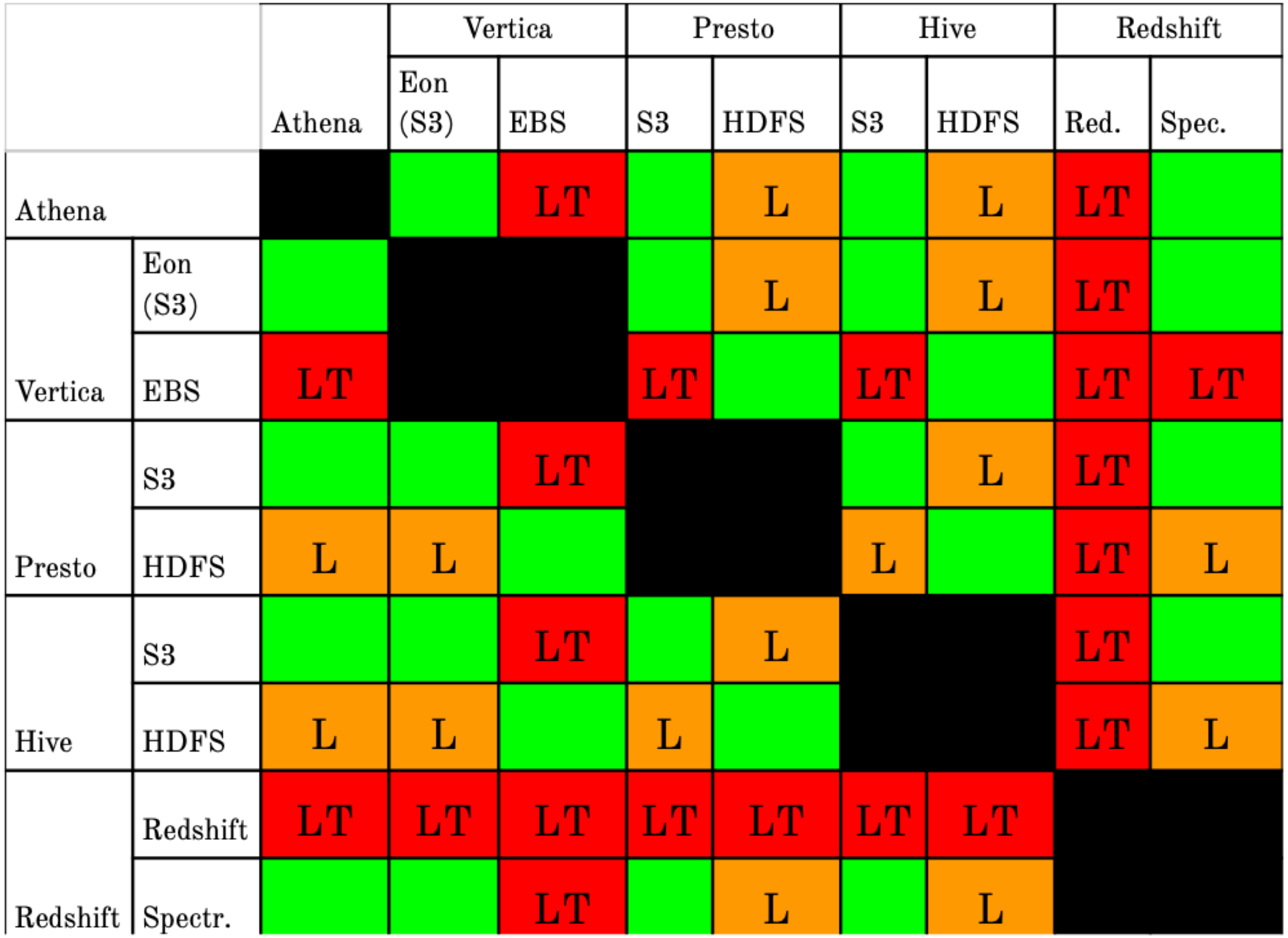
Systems We Did Not Test

Apache Drill

Apache Spark SQL

Snowflake

Data Compatibility among Systems



			Vertica		Presto		Hive		Redshift	
		Athena	Eon (S3)	EBS	S3	HDFS	S3	HDFS	Red.	Spec.
Athena		⬛	🟩	🟥 LT	🟩	🟧 L	🟩	🟧 L	🟥 LT	🟩
Vertica	Eon (S3)	🟩	⬛	⬛	🟩	🟧 L	🟩	🟧 L	🟥 LT	🟩
	EBS	🟥 LT	⬛	⬛	🟥 LT	🟩	🟥 LT	🟩	🟥 LT	🟥 LT
Presto	S3	🟩	🟩	🟥 LT	⬛	⬛	🟩	🟧 L	🟥 LT	🟩
	HDFS	🟧 L	🟧 L	🟩	⬛	⬛	🟧 L	🟩	🟥 LT	🟧 L
Hive	S3	🟩	🟩	🟥 LT	🟩	🟧 L	⬛	⬛	🟥 LT	🟩
	HDFS	🟧 L	🟧 L	🟩	🟧 L	🟩	⬛	⬛	🟥 LT	🟧 L
Redshift	Redshift	🟥 LT	🟥 LT	🟥 LT	🟥 LT	🟥 LT	🟥 LT	🟥 LT	⬛	⬛
	Spectr.	🟩	🟩	🟥 LT	🟩	🟧 L	🟩	🟧 L	⬛	⬛

Redshift not compatible with other systems

S3-based systems are all compatible

Complexity of Fair Comparison

System setup

Query optimization

Data format

Data types

Data partitioning

Query restriction

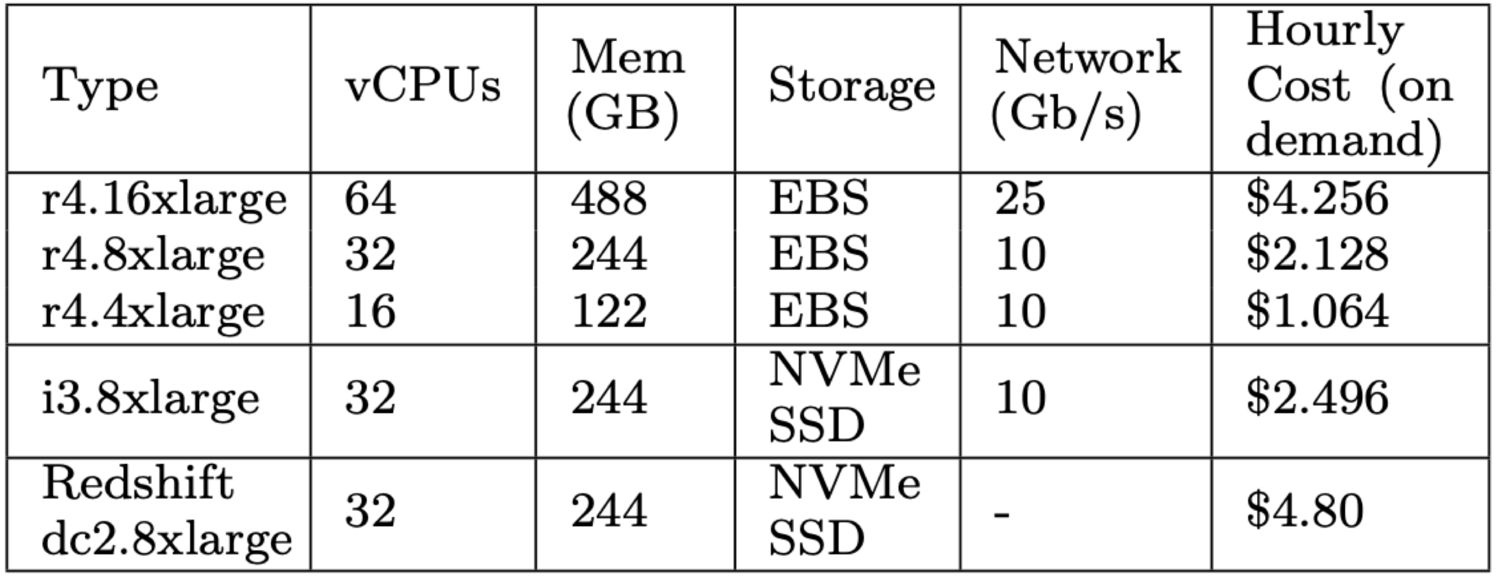
Caching

Spilling to disk

etc.

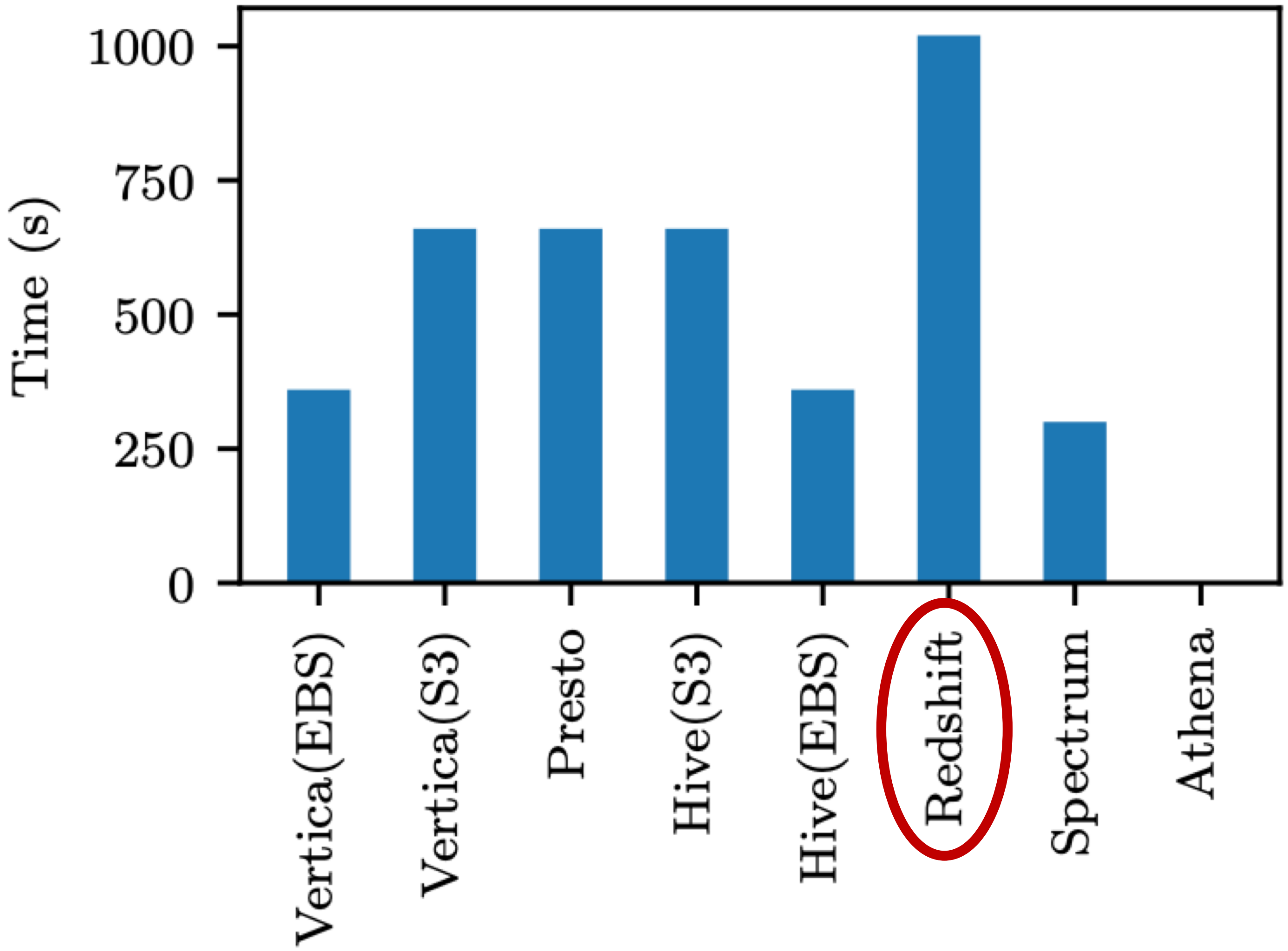
System Settings

- Single User (No parallel queries)
- Only SSD storage
- Fast networks (10 Gb/s+)
  - Base cluster is 4 nodes, r4.8xlarge (32 vCPU, 244GB RAM)
- TPC-H 1000SF (1 TB raw)



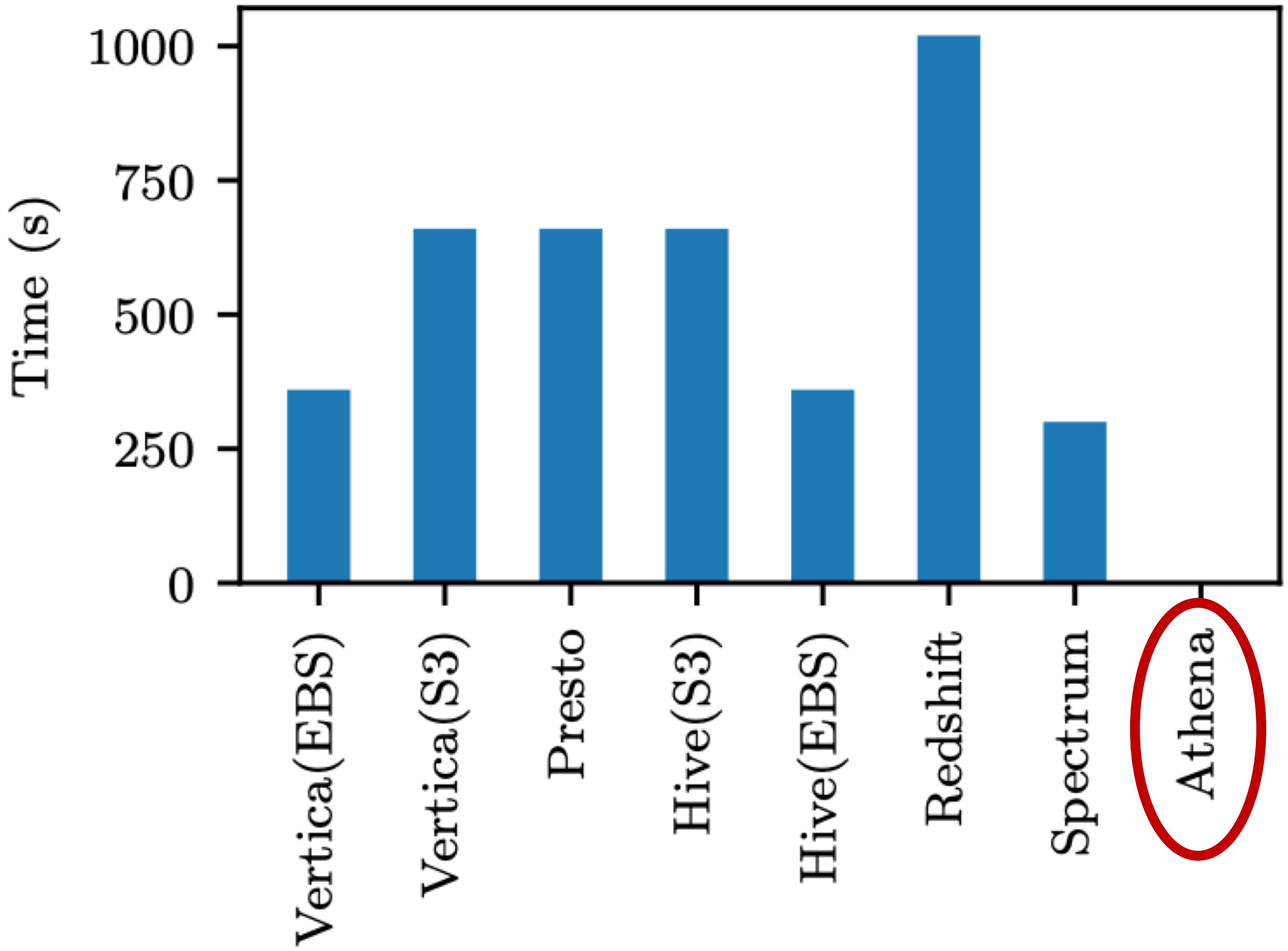
Type	vCPUs	Mem (GB)	Storage	Network (Gb/s)	Hourly Cost (on demand)
r4.16xlarge	64	488	EBS	25	$4.256
r4.8xlarge	32	244	EBS	10	$2.128
r4.4xlarge	16	122	EBS	10	$1.064
i3.8xlarge	32	244	NVMe SSD	10	$2.496
Redshift dc2.8xlarge	32	244	NVMe SSD	-	$4.80

Experiments – Time to First Insight



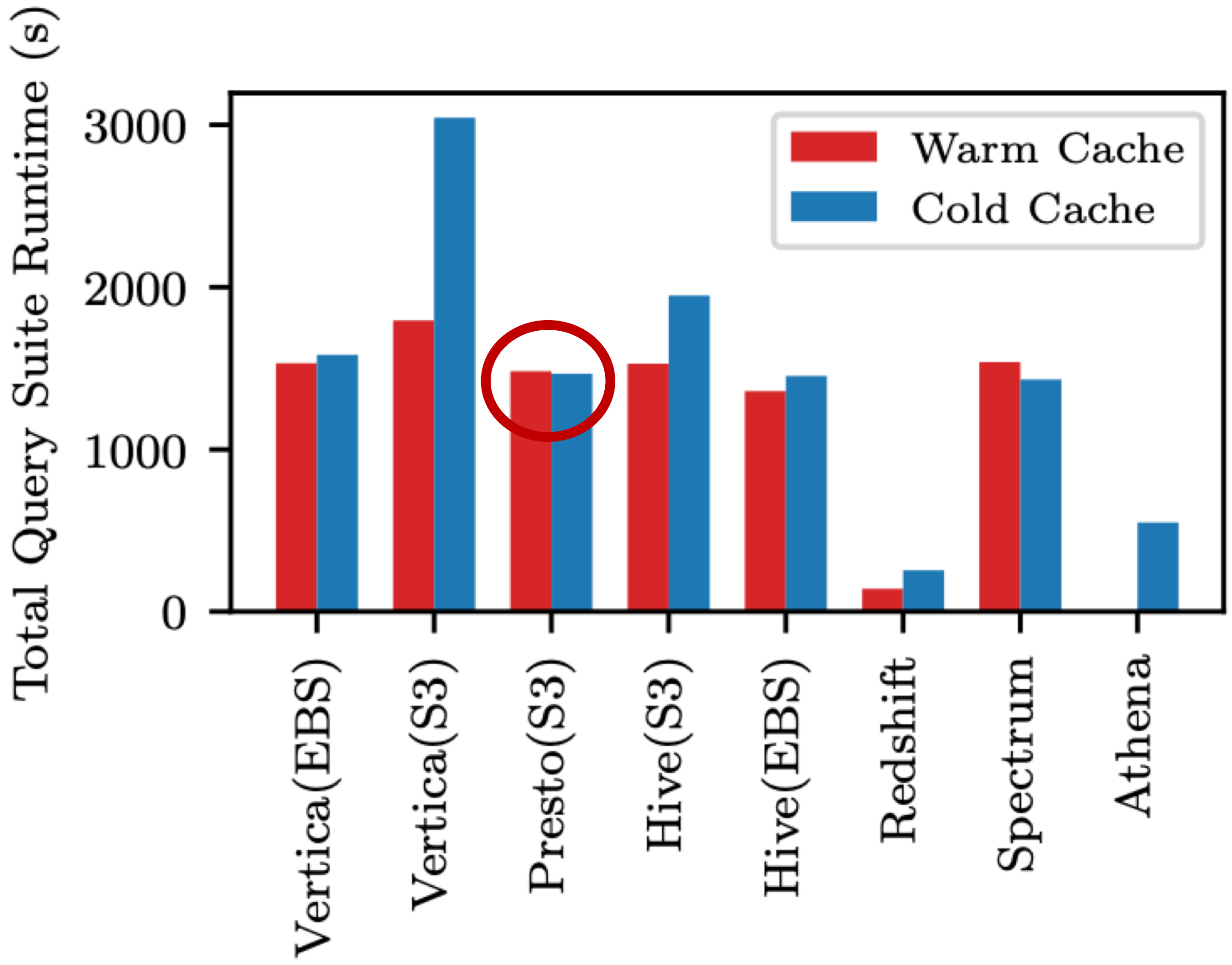
Redshift loads data from S3 to instance store

Experiments – Time to First Insight



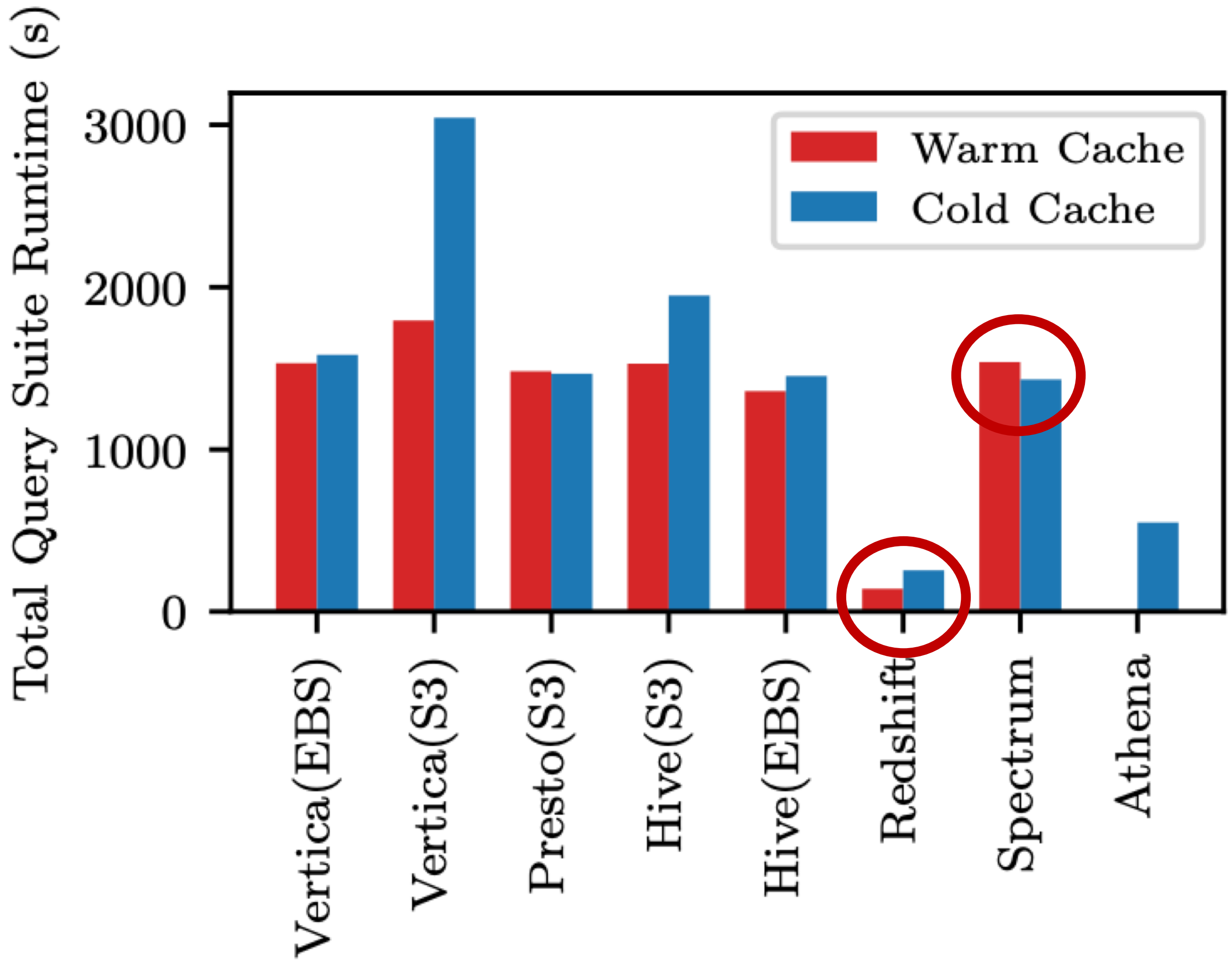
Athena is serverless and requires no initialization

Experiments – Caching Benefits



Some systems (like Presto) do not cache

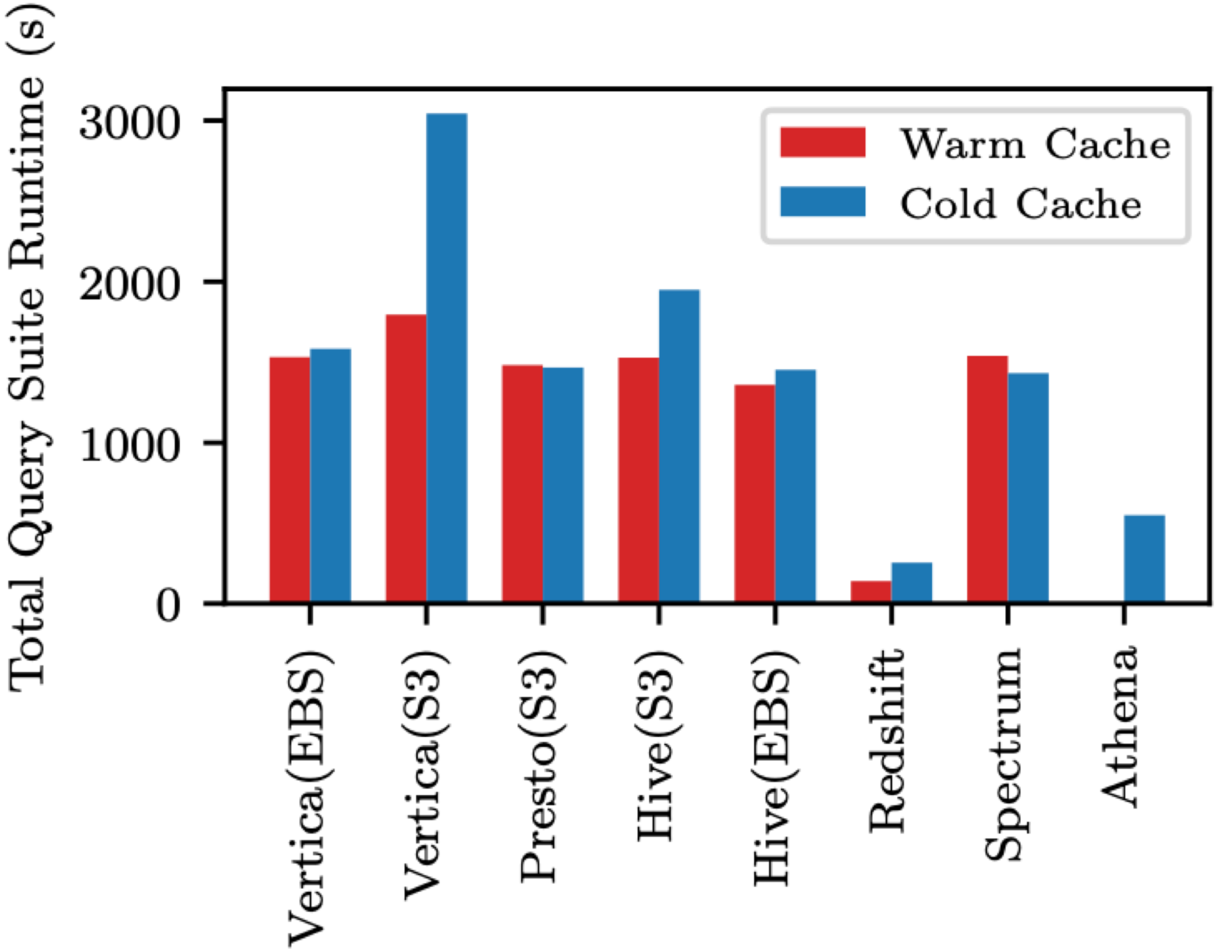
Experiments – Caching Benefits



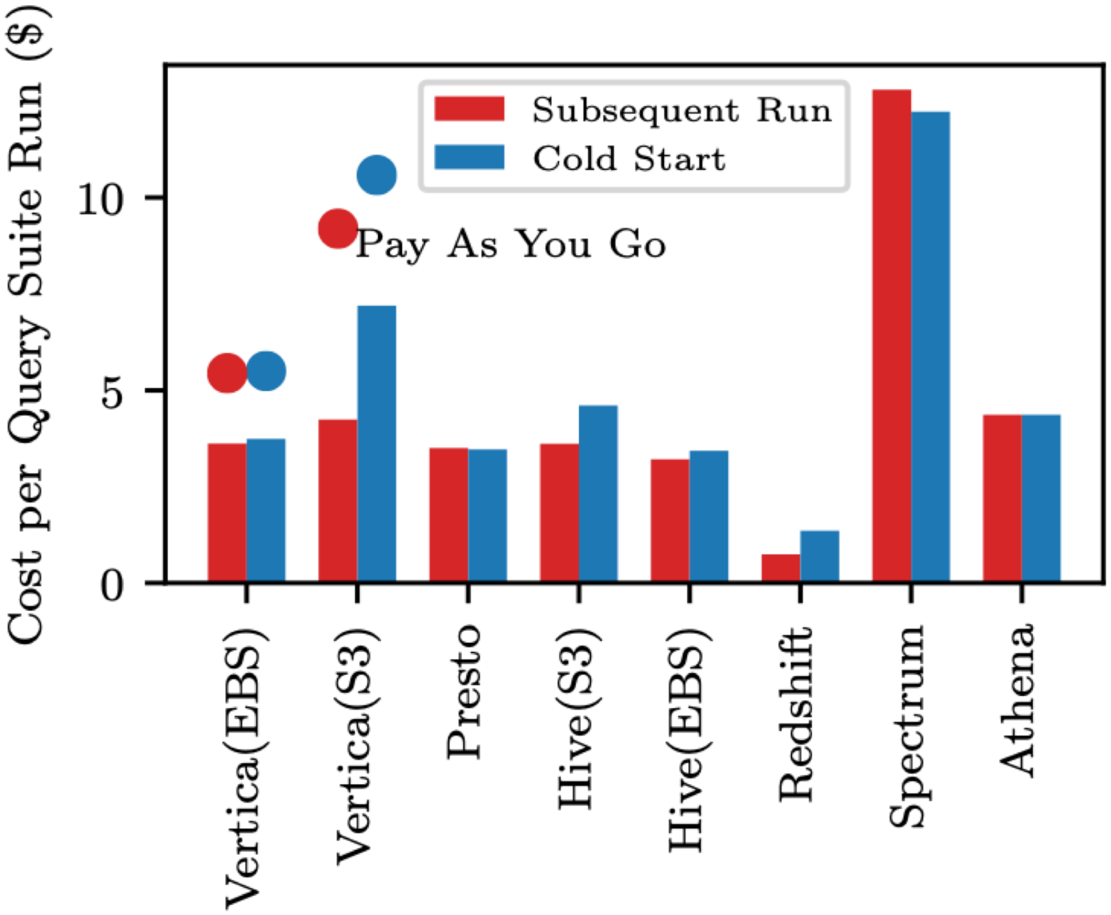
Redshift can be considered as caching in Spectrum

The caching decision is static

Experiments – Query Cost

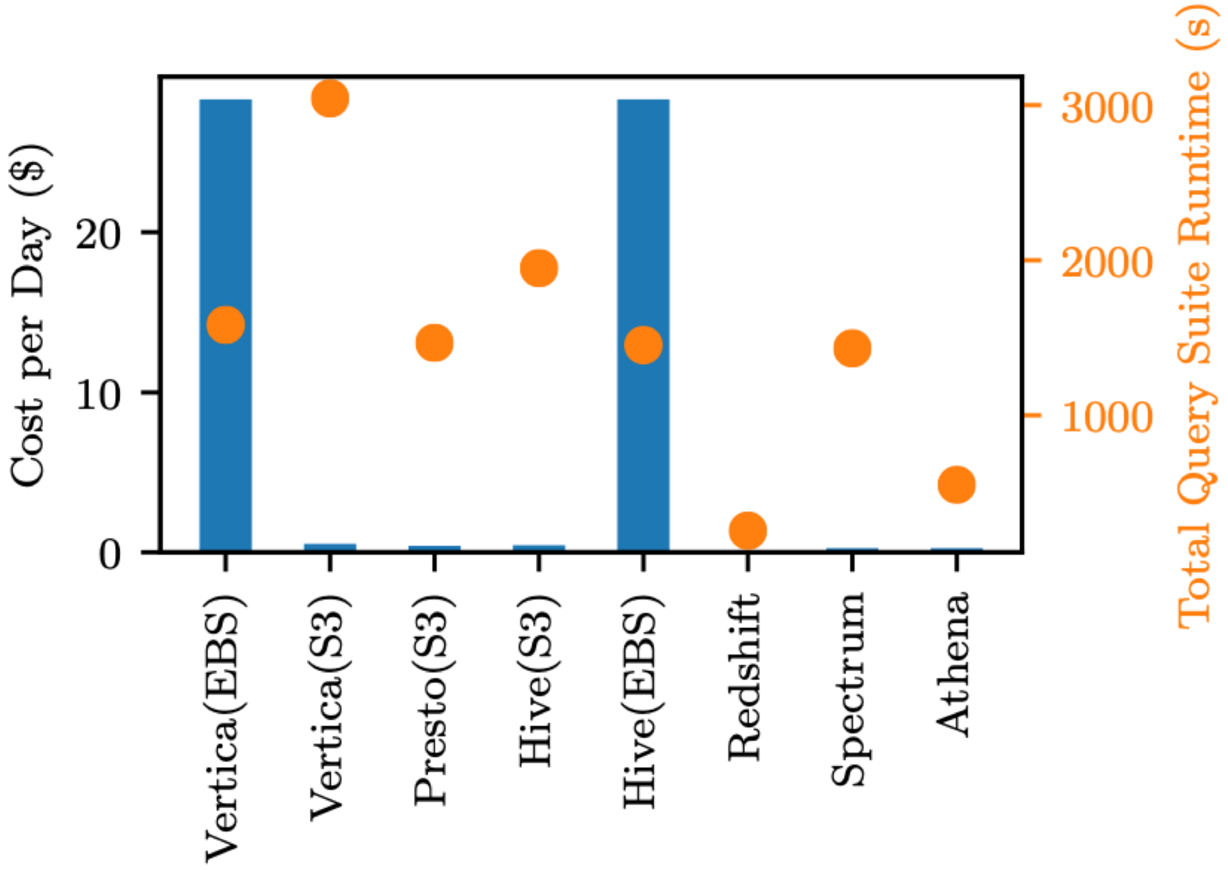


Runtime

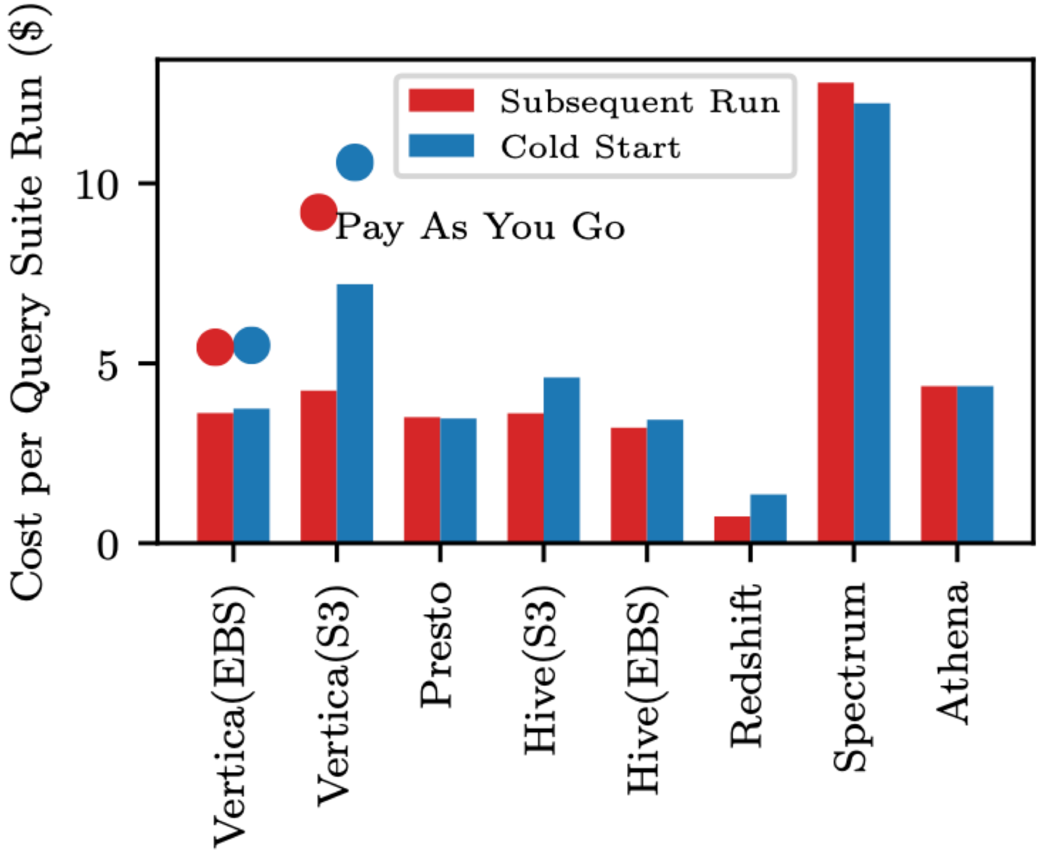


Cost

Experiments – Storage Cost

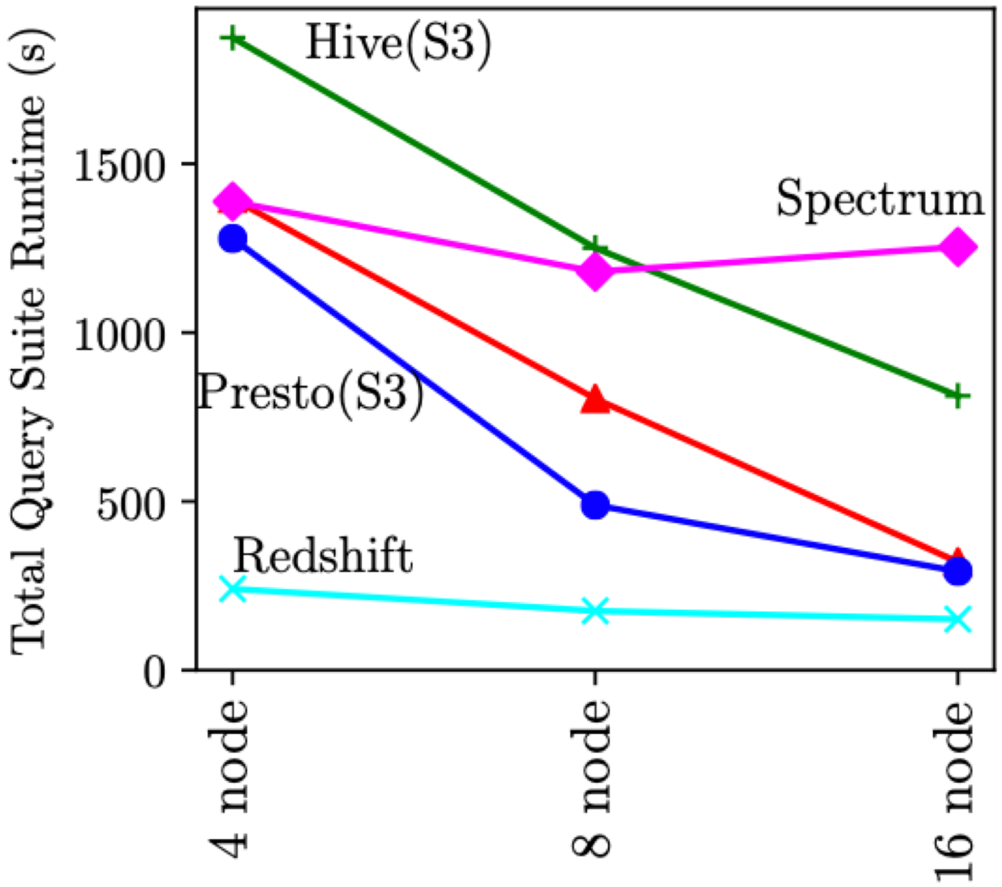


Storage cost

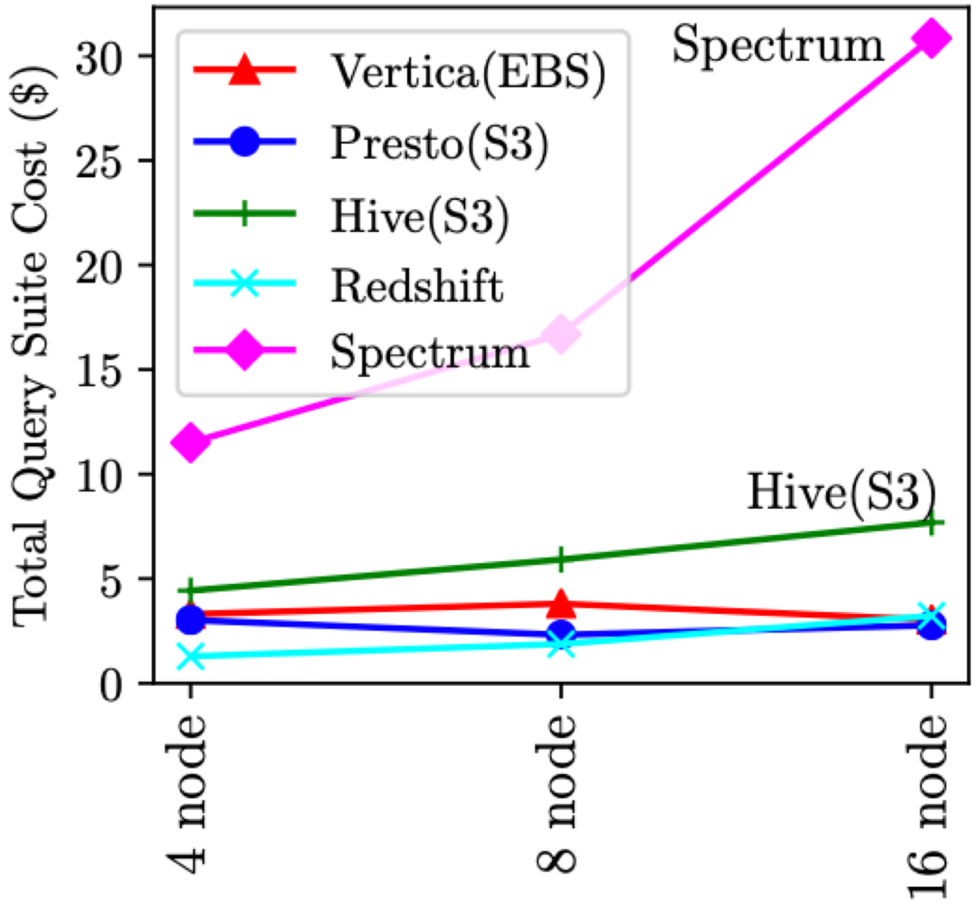


Query cost

Experiments – Scaling Out

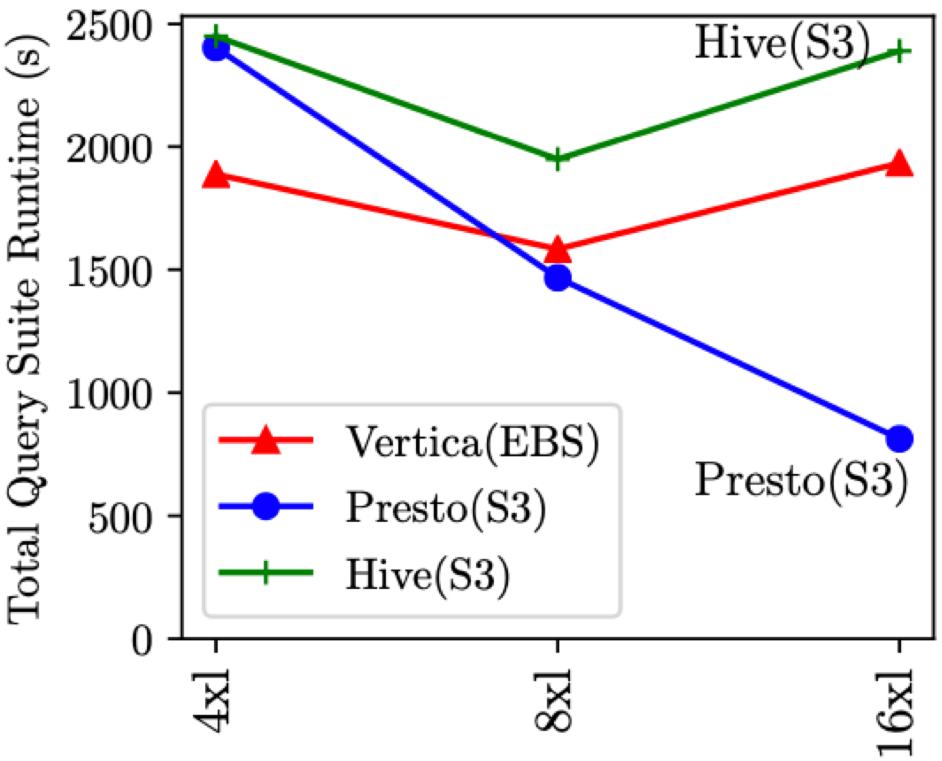


(c) 8xl Runtime (15 queries)

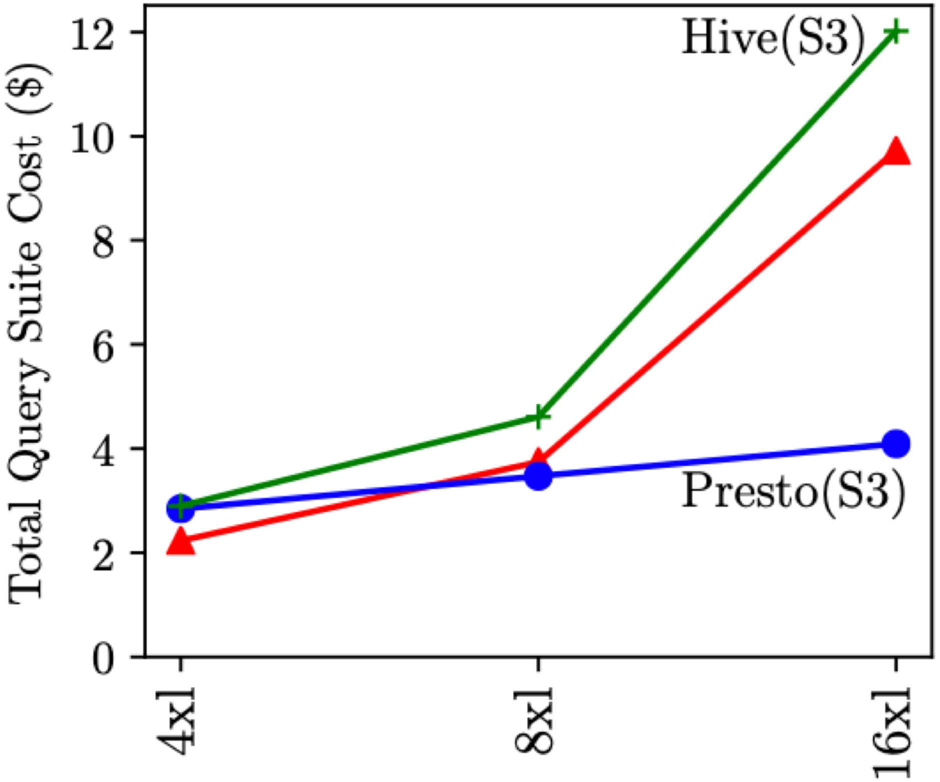


(d) 8xl Cost (15 queries)

Experiments – Scaling Up



(a) Runtime



(b) Cost

Cloud Data Warehousing – Q/A

/dev/null?

Why TPC-H?

Ease of adopting

A similar study on OLTP?

SmartSSD integrated into the stack?

Horizontal vs. vertical scaling?

ORC (Optimized Row Columnar)? Parquet?

Why data scan pricing model?

Comparison of Google, Azure, AWS?

Group Discussion

No system behaves strictly better than all the others in terms of performance and cost. What kind of design may combine the benefits of these existing systems?

Serverless databases expose a higher-level of abstraction to users, which gives cloud service providers more room for performance optimizations. What optimization opportunities can you see?

How can cloud databases benefit from the new hardware devices that we have discussed in this course?