Big Trends in Big Data

Reynold Xin
rxin@databricks.com
2018-03-12, SFU Big Data Class

Top 5 car companies, by market cap

Toyota	$200 B	1937
Volkswagen	$100 B	1937
Mercedes-Benz	$95 B	1926
BMW	$73 B	1916
Honda	$62 B	1948

Rough approximation, as of Feb 4, 2018

Top 5 car companies, by market cap

Toyota	$200 B	1937
Volkswagen	$100 B	1937
Mercedes-Benz	$95 B	1926
BMW	$73 B	1916
Honda	$62 B	1948
Tesla	$58 B	2003

Rough approximation, as of Feb 4, 2018

“Paradigm Shifts”: Opportunities for New Entrants

Paradigm shifts require major architectural changes or business model changes.

Electric cars (Tesla, BYD), PC (Apple, Intel, Microsoft), Internet (Google), SaaS (Salesforce), relational databases (Oracle).

Relational Databases

Designed for **tabular** data

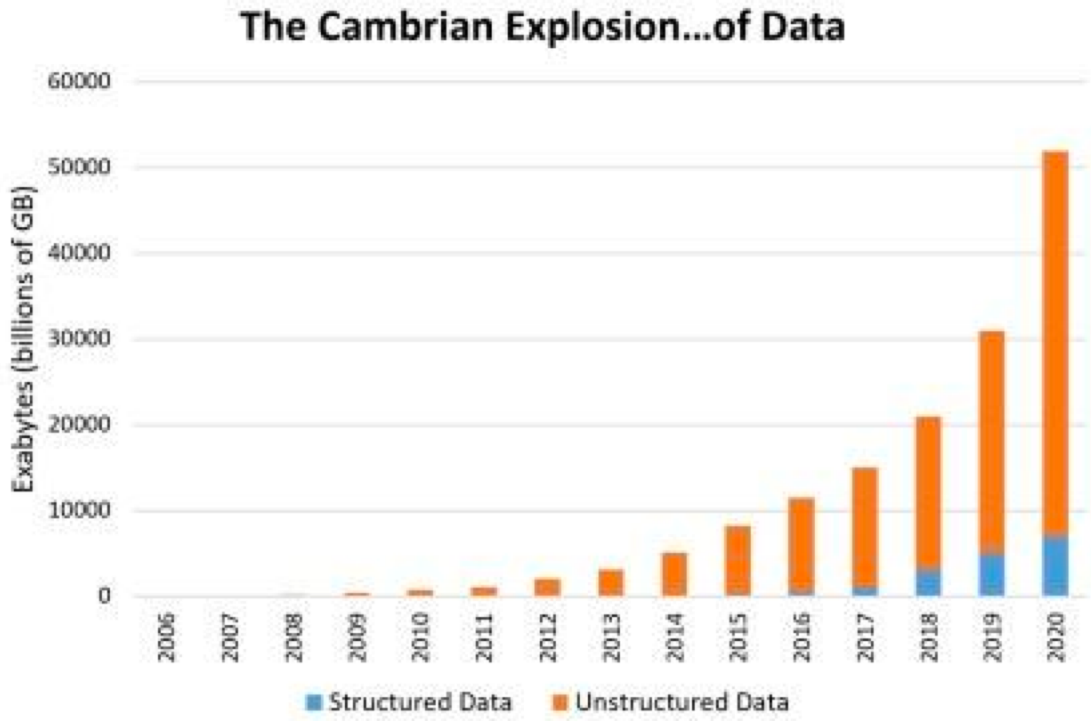
High level declarative interface: “what” rather than “how”.

- Enabled tremendous innovation: indexes, column stores, compression.
- Enabled higher level business applications.

$30B (annual rev) industry

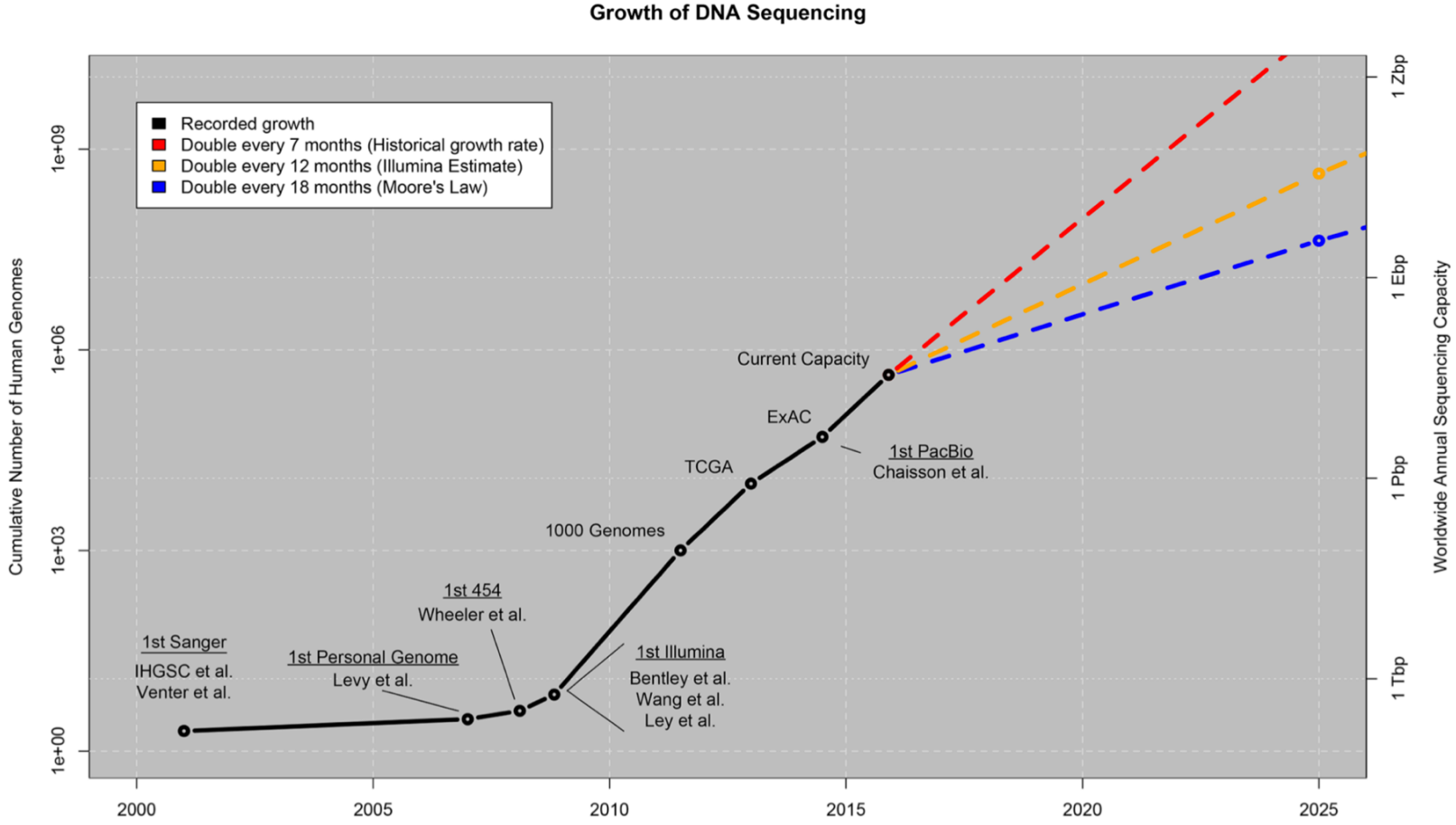
- Oracle, SQL Server, DB2, Teradata, …

Big Data



databricks

Source: https://www.eetimes.com/author.asp?section_id=36&doc_id=1330462&image_number=3


Growth of DNA Sequencing
Recorded growth
Double every 7 months (Historical growth rate)
Double every 12 months (Illumina Estimate)
Double every 18 months (Moore's Law)
Cumulative Number of Human Genomes
1e+00
1e+03
1e+06
1e+09
Worldwide Annual Sequencing Capacity
1 Tbp
1 Pbp
1 Ebp
1 Zbp
2000
2005
2010
2015
2020
2025
1st Sanger
IHGSC et al.
Venter et al.
1st Personal Genome
Levy et al.
1st 454
Wheeler et al.
1st Illumina
Bentley et al.
Wang et al.
Ley et al.
1000 Genomes
TCGA
ExAC
1st PacBio
Chaisson et al.
Current Capacity

databricks

Big Data Paradigm Shift

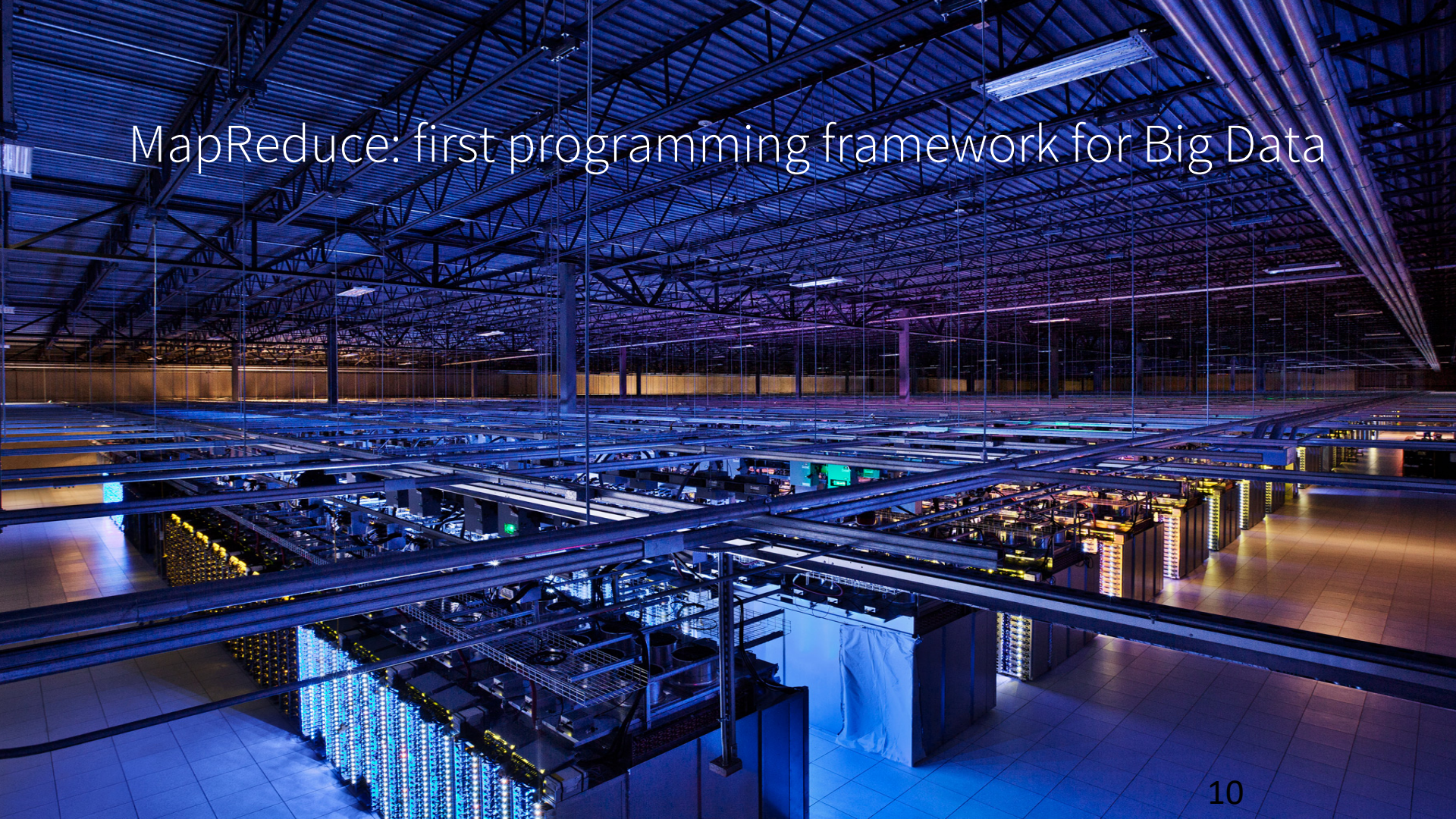
"Big data" is high-**volume**, -**velocity** and -**variety** information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.

Volume: data size

Velocity: rate of data coming in

Variety: data sources, formats, workloads

MapReduce: first programming framework for Big Data



MapReduce: A major step backwards

By David DeWitt on January 17, 2008 4:20 PM | Permalink | Comments (44) | TrackBacks (1)
[Note: Although the system attributes this post to a single author, it was written by David J. DeWitt and Michael Stonebraker]

On January 8, a Database Column reader asked for our views on new distributed database research efforts, and we'll begin here v
to discuss it, since the recent trade press has been filled with news of the revolution of so-called "cloud computing." This paradigı
processors working in parallel to solve a computing problem. In effect, this suggests constructing a data center by lining up a largı
much smaller number of high-end servers.

For example, IBM and Google have announced plans to make a 1,000 processor cluster available to a few select universities to te
software tool called MapReduce [1]. Berkeley has gone so far as to plan on teaching their freshman how to program using the Ma

As both educators and researchers, we are amazed at the hype that the MapReduce proponents have spread about how it represen
data-intensive applications. MapReduce may be a good idea for writing certain types of general-purpose computations, but to the

1. A giant step backward in the programming paradigm for large-scale data intensive applications
2. A sub-optimal implementation, in that it uses brute force instead of indexing
3. Not novel at all -- it represents a specific implementation of well known techniques developed nearly 25 years ago
4. Missing most of the features that are routinely included in current DBMS

“This is an exciting time. The bar for open source software is at historical low.”

- Michael J. Carey @ UC Berkeley, 2012
(my academic grand-advisor)

A slide from 2013 …

Spark

Fast and expressive cluster computing system interoperable with Apache Hadoop

Improves efficiency through:
- » In-memory computing primitives
- » General computation graphs

→ Up to 100× faster (2-10× on disk)

Improves usability through:
- » Rich APIs in Scala, Java, Python
- » Interactive shell

→ Often 5× less code

What led to Spark's success?

Diverse team background

- systems, networking, databases, machine learning, Python, R

Took the best ideas out of MapReduce + Databases + Data Science

- Scalability and flexibility of MapReduce
- Higher level APIs (SQL, DataFrame, ML Pipeline)

Apache Spark
Computer software

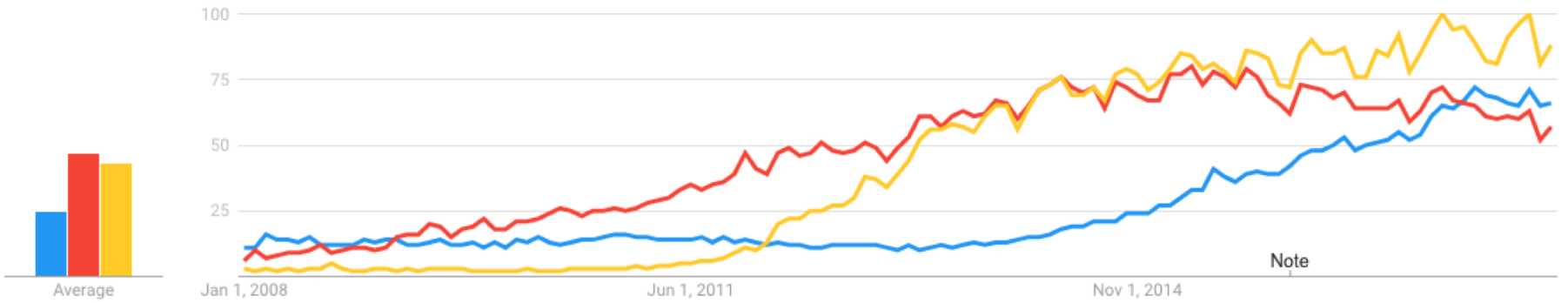
Apache Hadoop
Software

big data
Topic

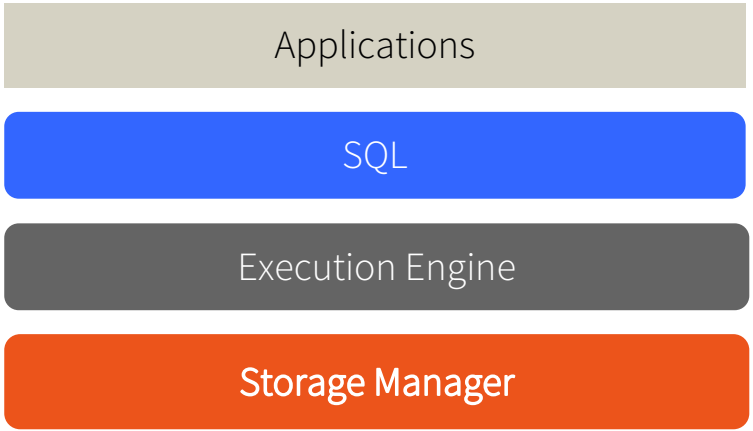
+ Add comparison

Worldwide | 1/1/08 - 1/1/18 | All categories | Web Search

Interest over time

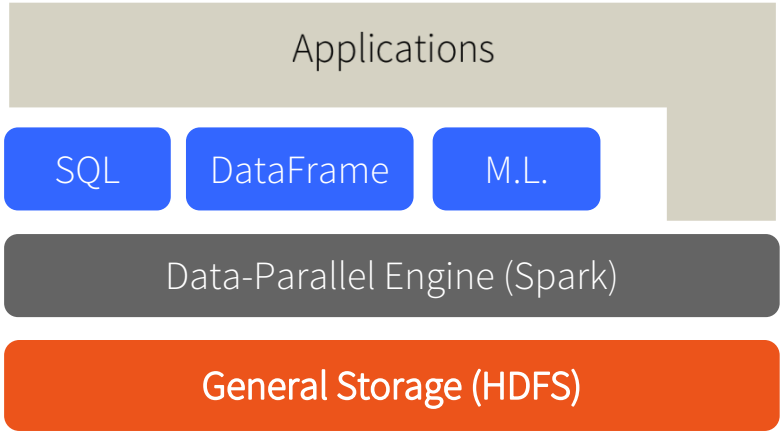


Relational Databases



One way (SQL) in/out and
data must be structured

Big Data Stack



Not only SQL
Structured, semi-structured, unstructured data

Big Data + Cloud + AI

Cloud Computing Democratizes …

Scale of compute & storage

Best-in-class infrastructure

Distribution and global availability

Storing and processing terabytes of data no longer the privilege of 1% tech companies.

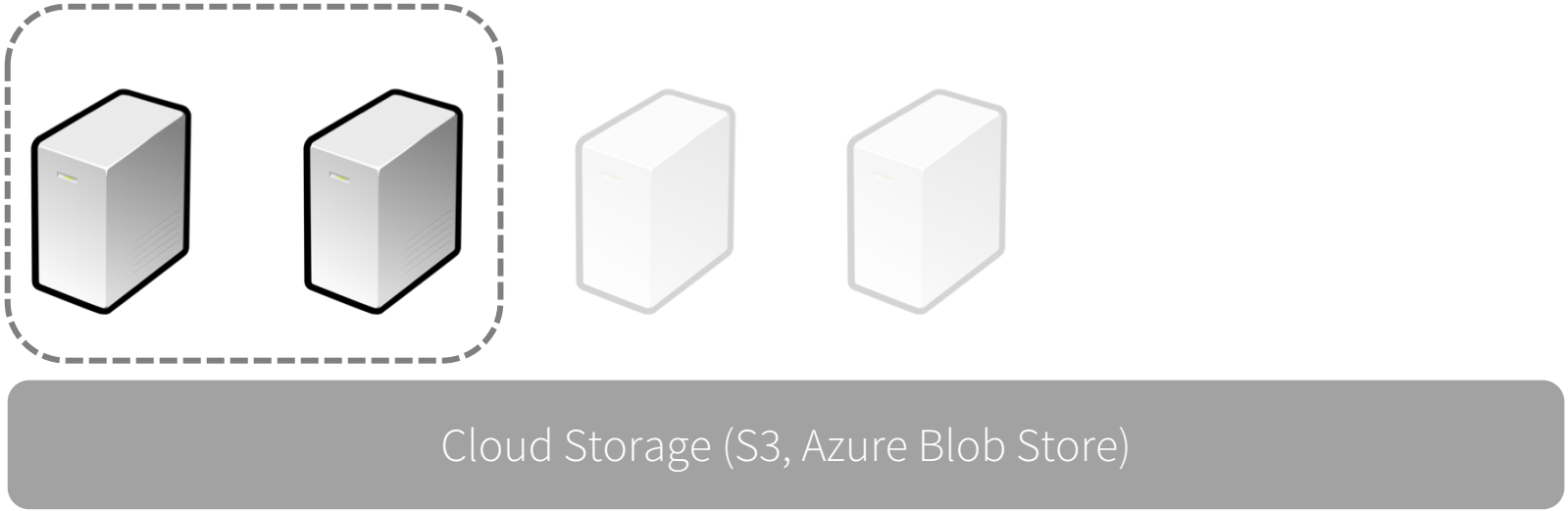
Cloud Computing

Two major architectural changes: Decoupled storage + Elasticity

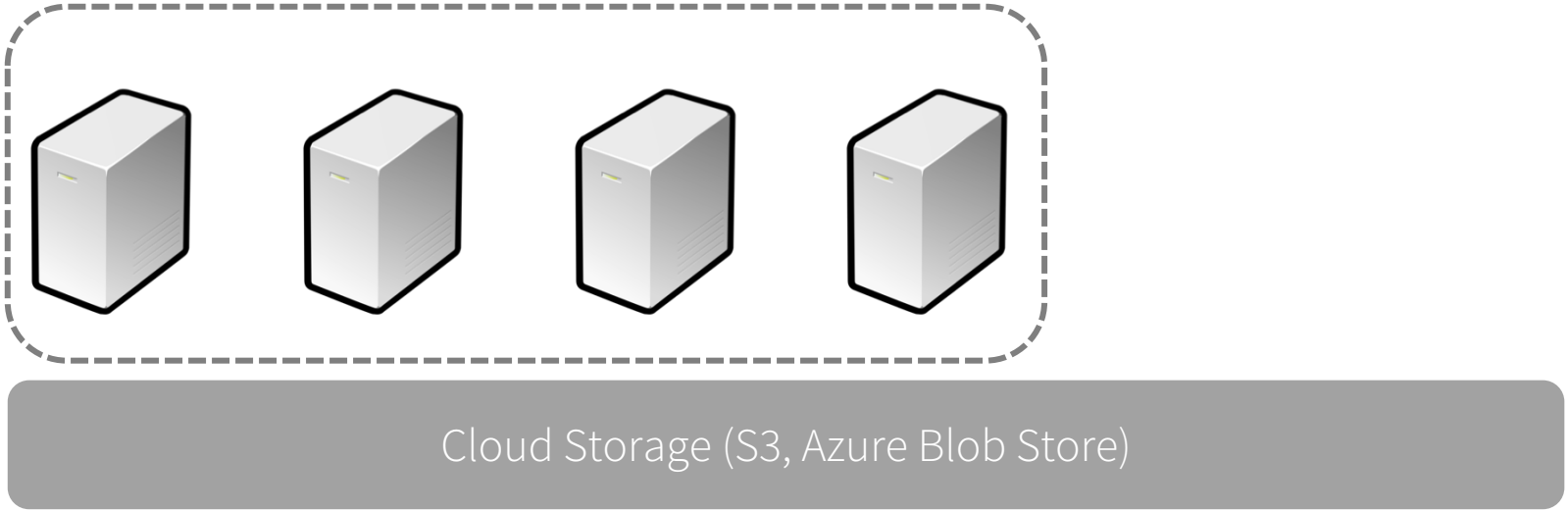
Invalidates old assumptions: fixed resources + collocated storage & compute

Scaling in the Cloud

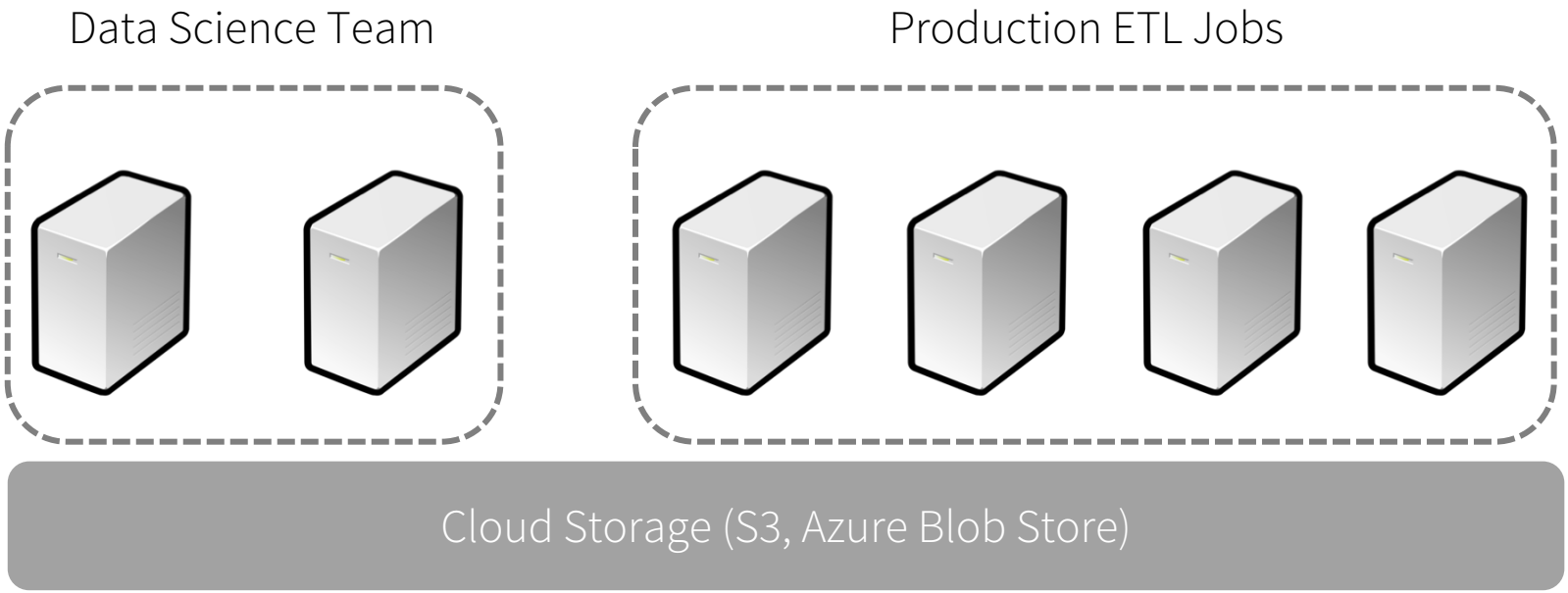
Scaling Databricks Resources



Scaling Databricks Resources



Scaling Databricks Resources



databricks

Done in minutes!

Scaling Redshift (a “cloud” database)

Step 1. Freeze the cluster (read only, all sessions disconnected)

Step 2. Launches an entire new cluster

Step 3. Copies all data from old cluster to the new cluster

Step 4. Switches connection end point over to new cluster

Re: How long should it take to resize a cluster?

Reply

Posted by: gabriel_burt

Posted on: Mar 21, 2013 2:59 PM

in response to: gabri[illegible]

It's still going, 18
supposed to be al

Found out you ca

```
pip instal
aws redshi
```

Amazon Redshift Stuck at 99% during resize operation

4

I've been waiting several hours for my redshift cluster complete a resize. It has been stuck at 99% complete with 0 time and 0 data remaining for 2 hours now. A quick web and forum search shows that this is fairly common. All the threads have no details about resolution, other than that an AWS rep PMs the user.

This is a cross-post from the AWS discussion forums.

a

sh

2

I contacted AWS support in person (at the SF Market street loft).

It turns out that copying the last 1% of the data in a resize can take 80 to 90% of the time. In this case, the first 99% of the data was copied in 1 hour. The rest of the resize took 10 more hours. Unfortunately, neither the documentation nor the progress bar and time remaining indicators reflect this.

Your milage may vary.

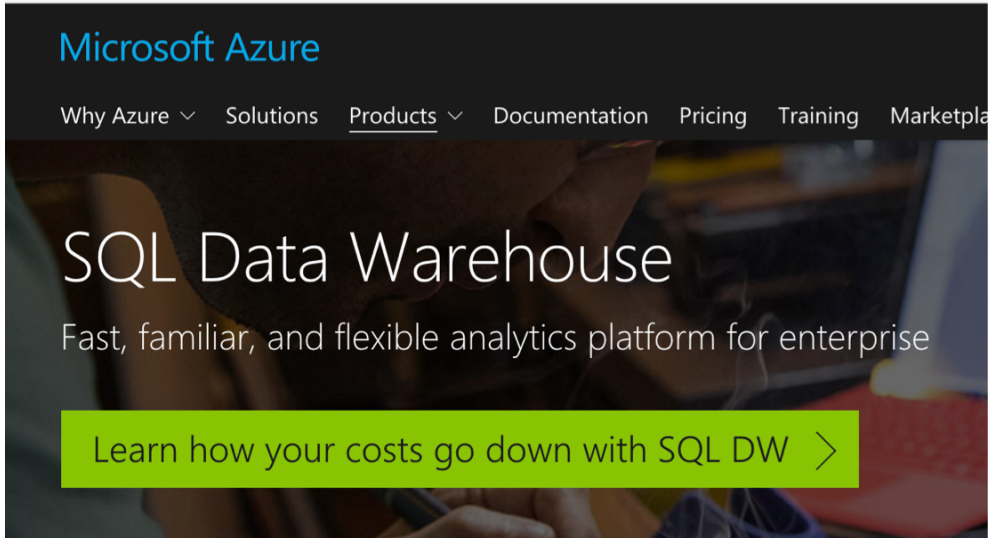
share improve this answer

answered Aug 5 '15 at 18:02

DaveA

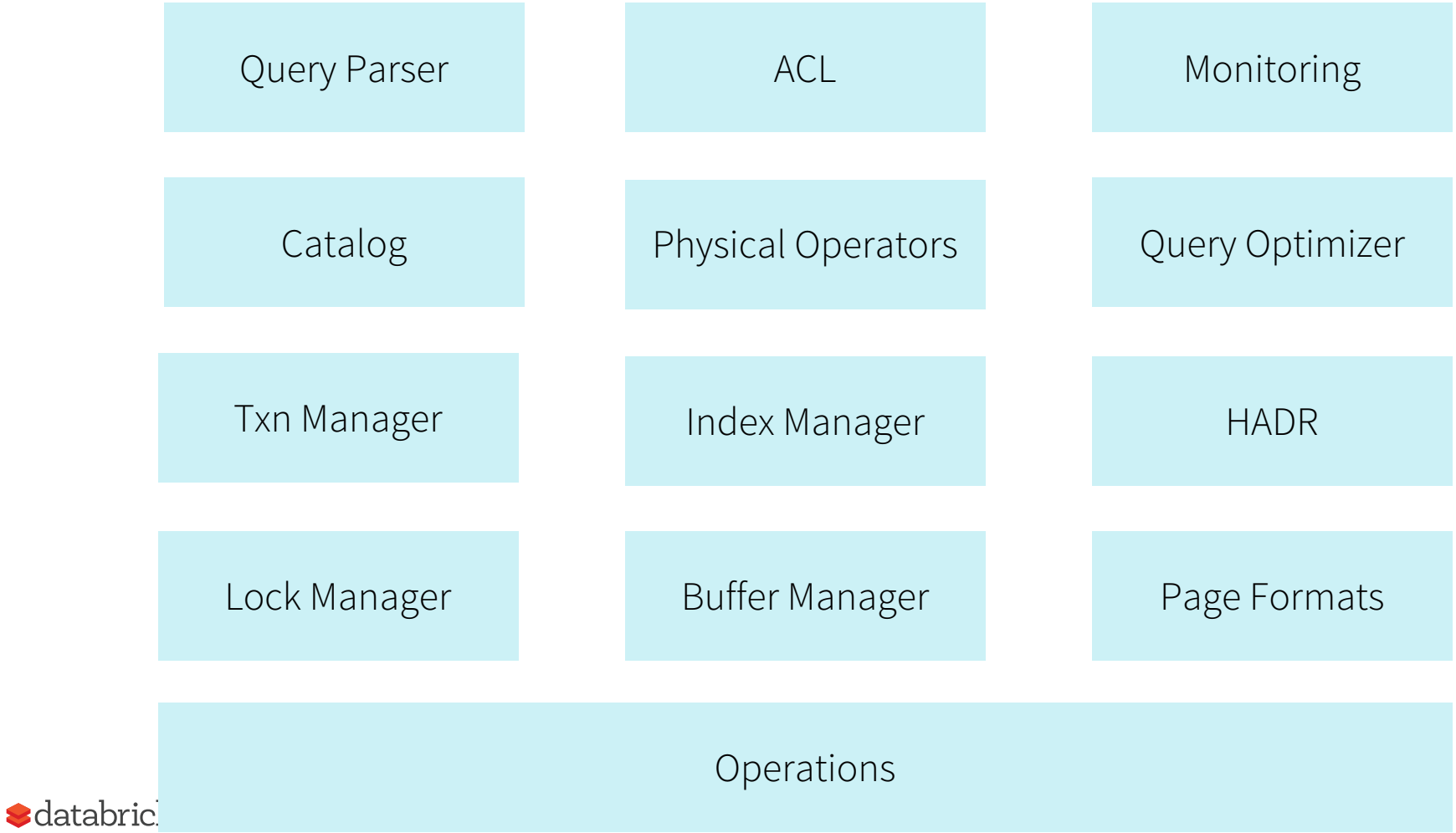
532 ● 1 ● 6 ● 15

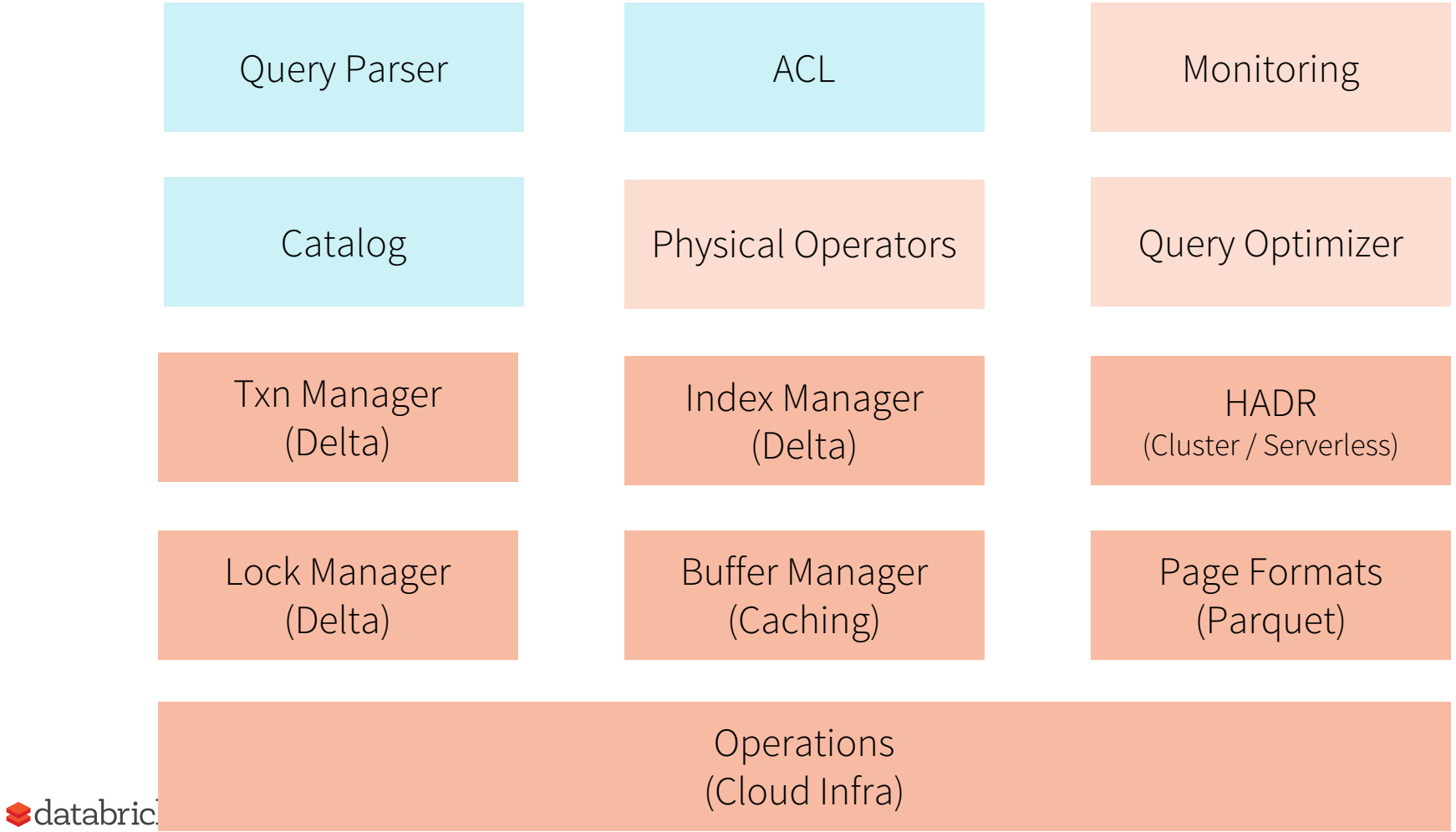
databricks



Concurrency limits

DWU	Max concurrent queries
DW100	4
DW200	8
DW300	12
DW400	16
DW500	20
DW600	24
DW1000	32
DW1200	32
DW1500	32
DW2000	32
DW3000	32
DW6000	32


Query Parser
ACL
Monitoring
Catalog
Physical Operators
Query Optimizer
Txn Manager
Index Manager
HADR
Lock Manager
Buffer Manager
Page Formats
Operations


Query Parser
ACL
Monitoring
Catalog
Physical Operators
Query Optimizer
Txn Manager
(Delta)
Index Manager
(Delta)
HADR
(Cluster / Serverless)
Lock Manager
(Delta)
Buffer Manager
(Caching)
Page Formats
(Parquet)
Operations
(Cloud Infra)

Redefining Data Systems

Resource isolation and scaling (and decoupled storage)

Beyond SQL

Automated cloud infrastructure

Conclusion

From a technical point of view, this is the **most exciting time of all** to be working on data.

Traditional data system architectures are outdated. The industry is redefining how data systems are built.

Thank you!

rxin@databricks.com