ALLUXIO

Best Practices for Using Alluxio with Spark

Gene Pang, Alluxio, Inc.

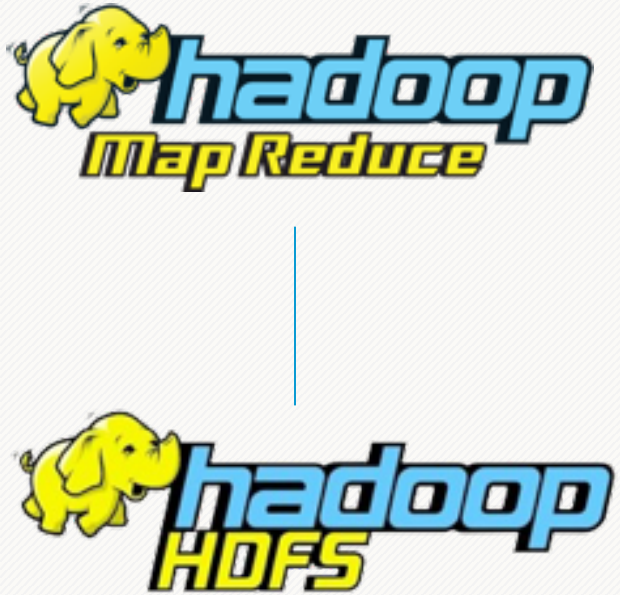
Cheng Chang, Alluxio, Inc.

Spark Summit SF - June 2017

Outline

1. Alluxio Overview
2. Alluxio + Spark Use Cases
3. Using Spark with Alluxio
4. Performance Evaluation
5. Demo

Data Ecosystem Yesterday

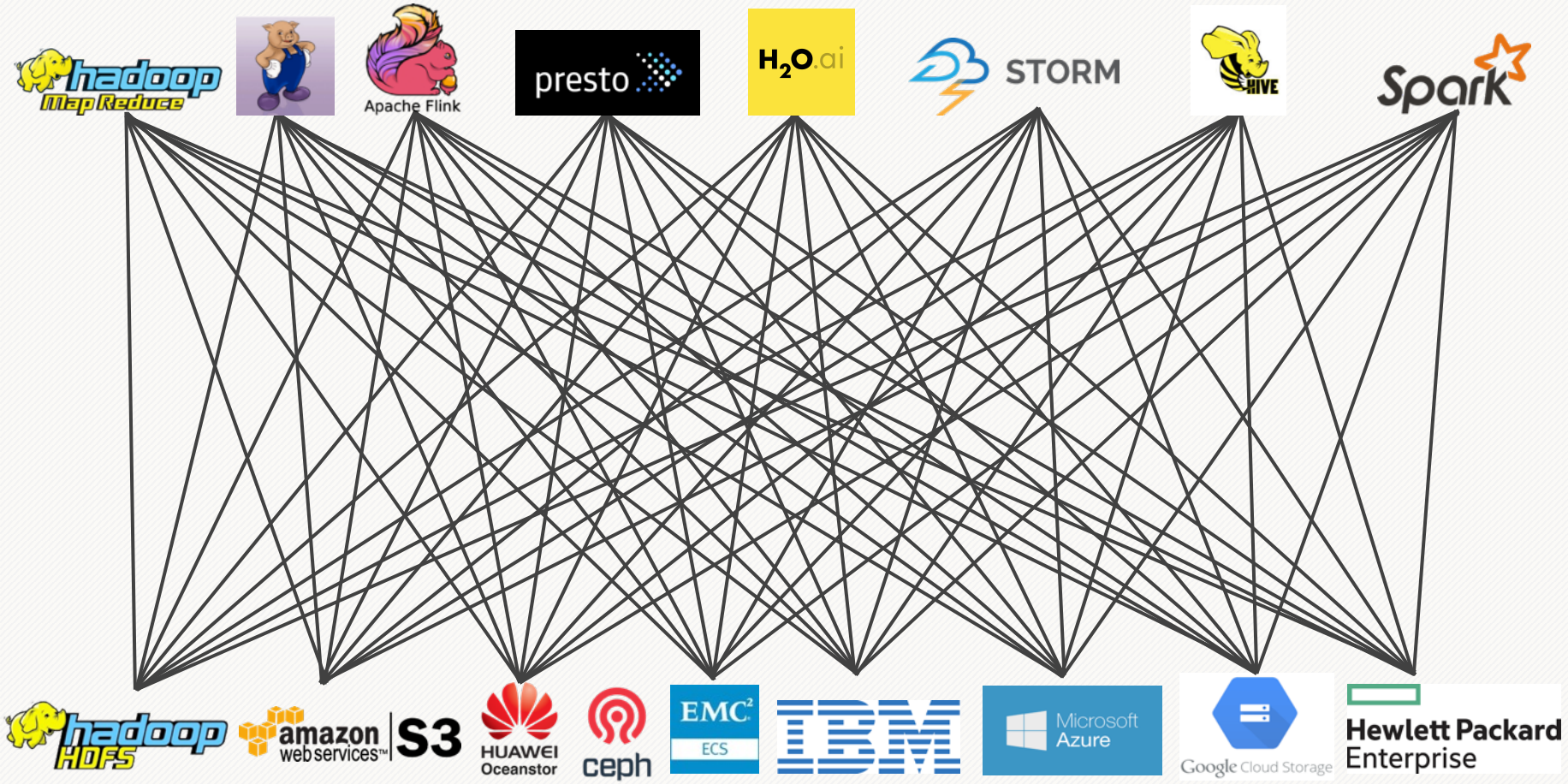


- One Compute Framework
- Single Storage System
- Co-located

Data Ecosystem Today

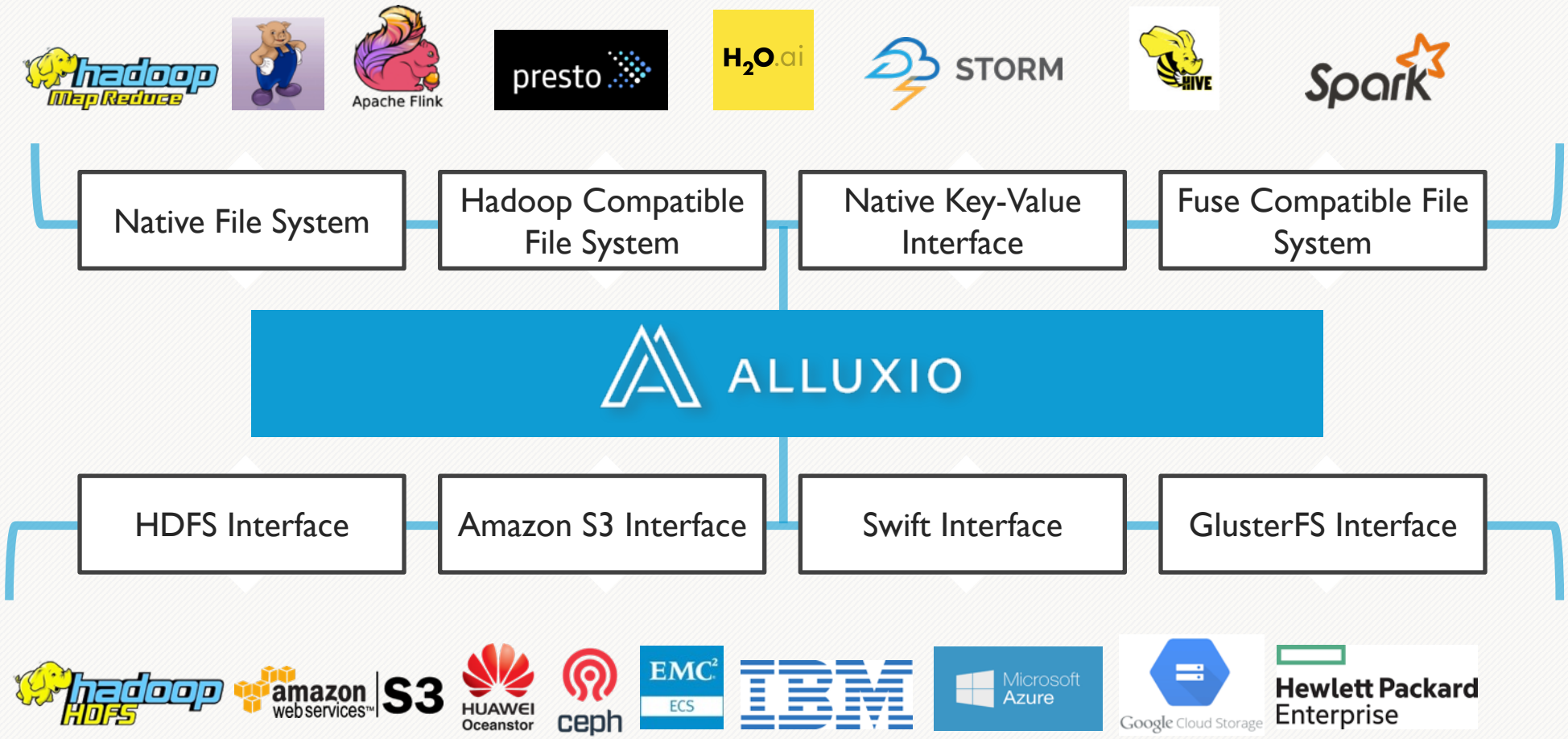
- Many Compute Frameworks
- Multiple Storage Systems
- Most not co-located

Data Ecosystem Issues



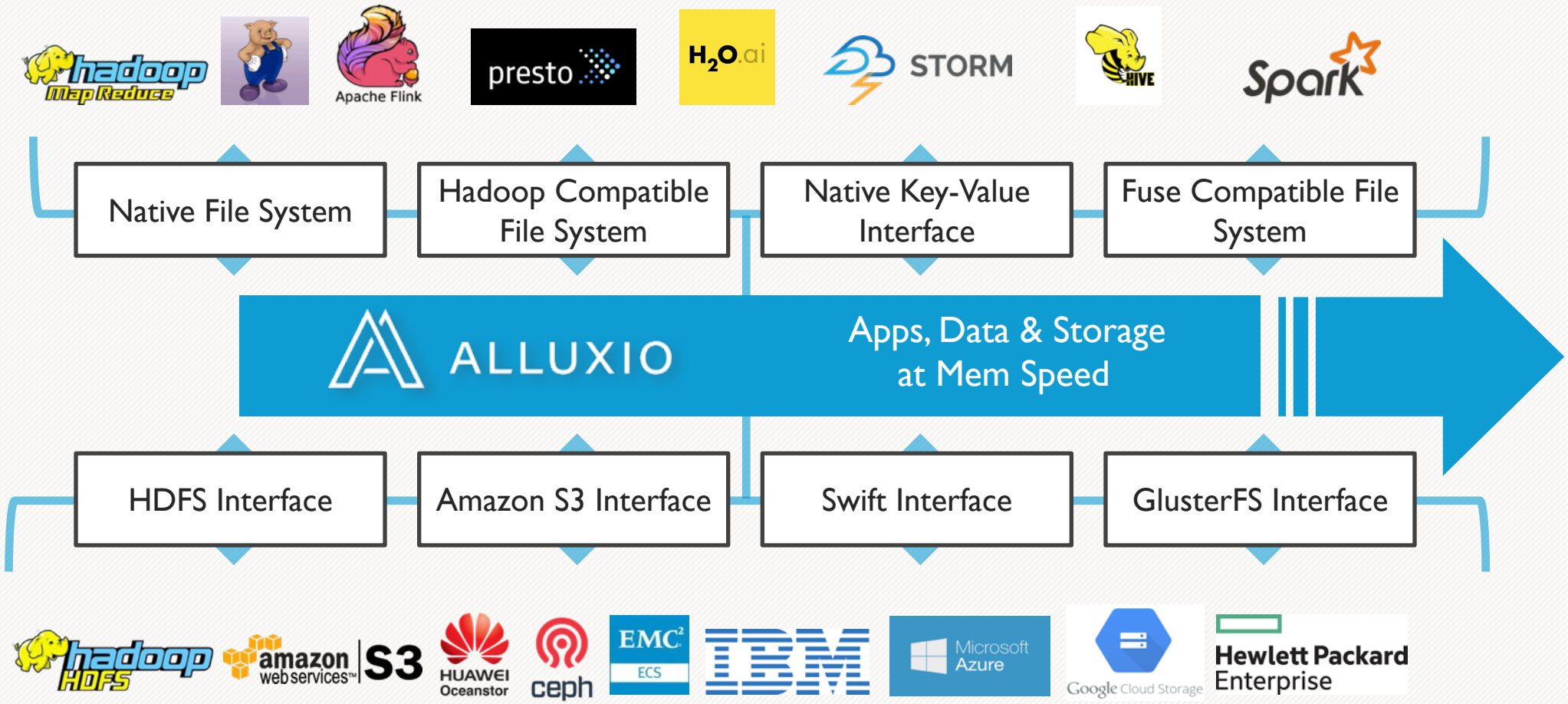
- Each application manage multiple data sources
- Add/Removing data sources require application changes
- Storage optimizations requires application change
- Lower performance due to lack of locality

Data Ecosystem with Alluxio



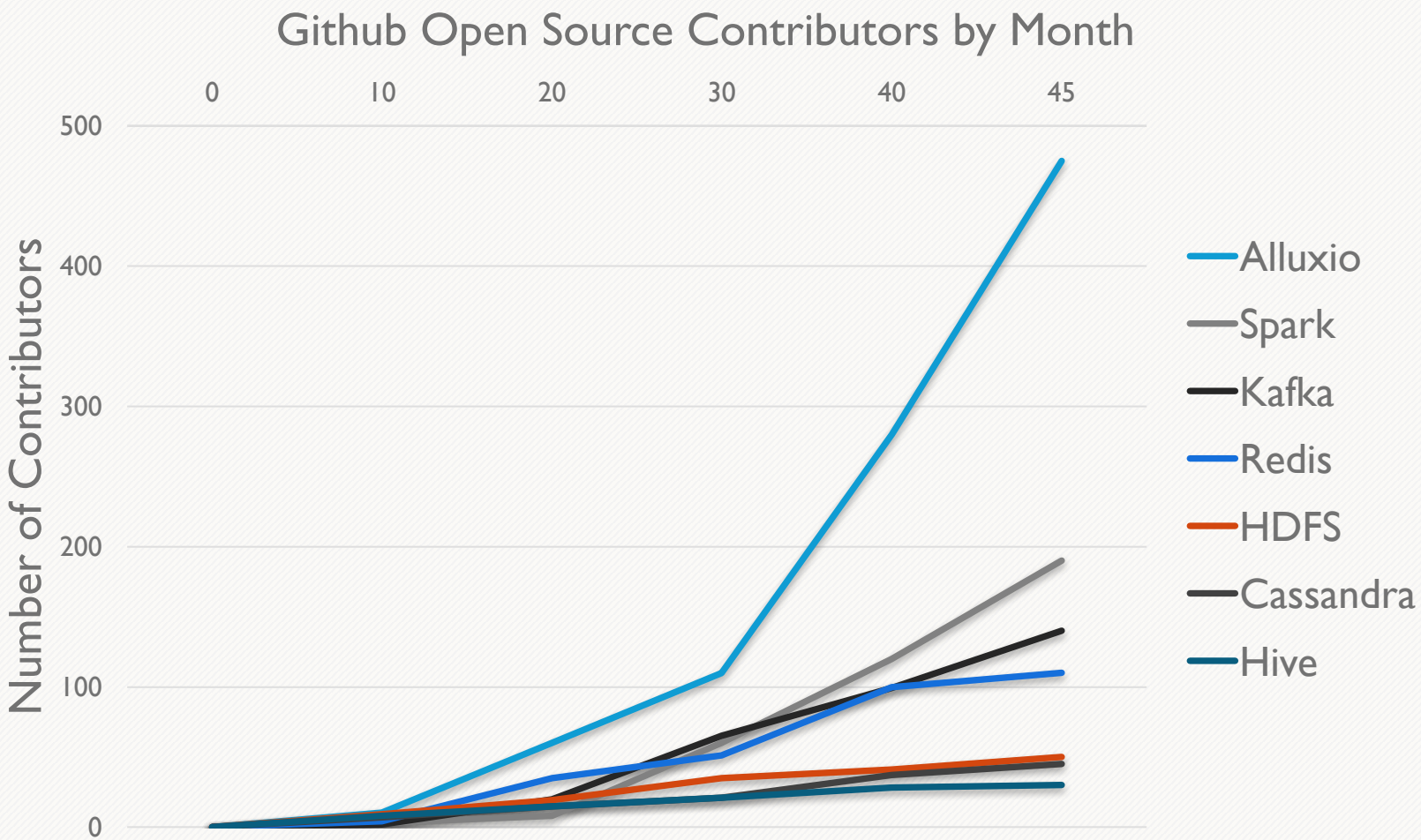
- Apps only talk to Alluxio
- Simple Add/Remove
- No App Changes
- Highest performance in Memory
- No Lock in

Next Gen Analytics with Alluxio



- ✓ Big Data/IoT
- ✓ AI/ML
- ✓ Deep Learning
- ✓ Cloud Migration
- ✓ Multi Platform
- ✓ Autonomous

Fastest Growing Big Data Open Source Projects



Fastest Growing open-source project in the big data ecosystem

Running in large production clusters

500+ Contributors from 100+ organizations

Outline

1. Alluxio Overview
2. Alluxio + Spark Use Cases
3. Using Spark with Alluxio
4. Performance Evaluation
5. Demo

Big Data Case Study – BARCLAYS

SPARK

TERADATA

Challenge –
Gain end to end view of business with large volume of data

Queries were slow / not interactive, resulting in operational inefficiency

SPARK

ALLUXIO

TERADATA

Solution –
ETL Data from Teradata to Alluxio

Impact –
Faster Time to Market – "Now we don't have to work Sundays"

http://bit.ly/2oMx95W

Big Data Case Study – Baidu 百度

SPARK

Baidu File System

Challenge –
Gain end to end view of business with large volume of data

Queries were slow / not interactive, resulting in operational inefficiency

SPARK

ALLUXIO

Baidu File System

Solution –
With Alluxio, data queries are 30X faster

Impact –
Higher operational efficiency

http://bit.ly/2pDHS3O

Big Data Case Study – Qunar.Com

SPARK | FLINK | HDFS | CEPH

Challenge –
Gain end to end view of business with large volume of data for $5B Travel Site

Queries were slow / not interactive, resulting in operational inefficiency

SPARK | FLINK | ALLUXIO | HDFS | CEPH

Solution –
With Alluxio, 300x improvement in performance

Impact –
Increased revenue from immediate response to user behavior
Use case: http://bit.ly/2pDJdrq

Machine Learning Case Study – GUARDANT HEALTH

ALLUXIO

SPARK

HDFS

Challenge –
Disparate Data both on-prem and Cloud. Heterogeneous types of data.

Scaling of Exabyte size data. Slow due to disk based approach.

SPARK

ALLUXIO

MINIO

MESOS

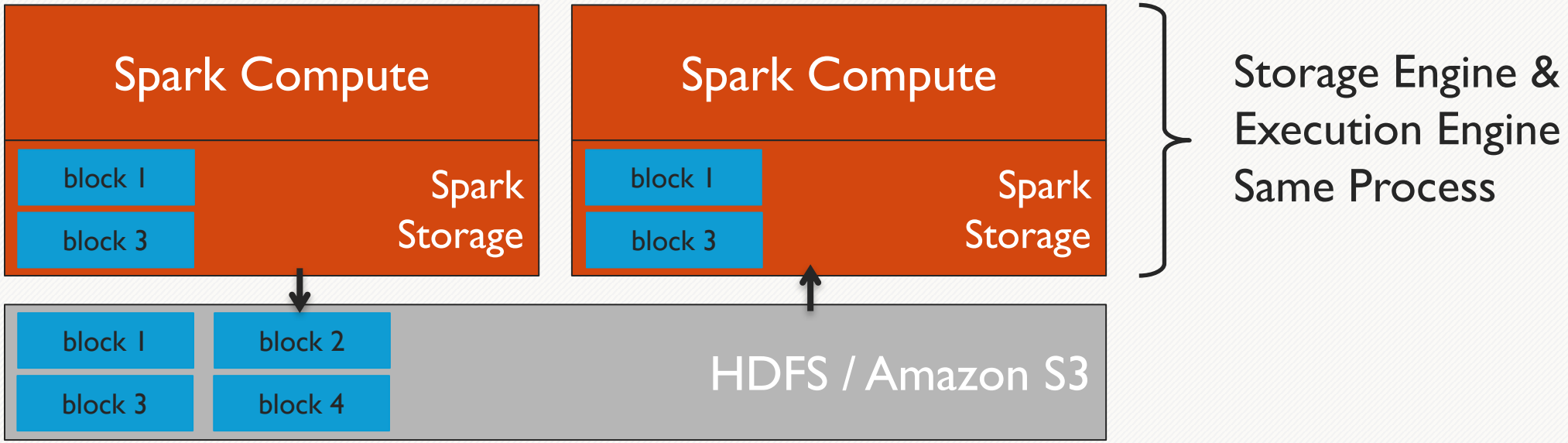
Solution –
Using Alluxio to prevent I/O bottlenecks

Impact –
Orders of magnitude higher performance than before.
http://bit.ly/2p18ds3

Outline

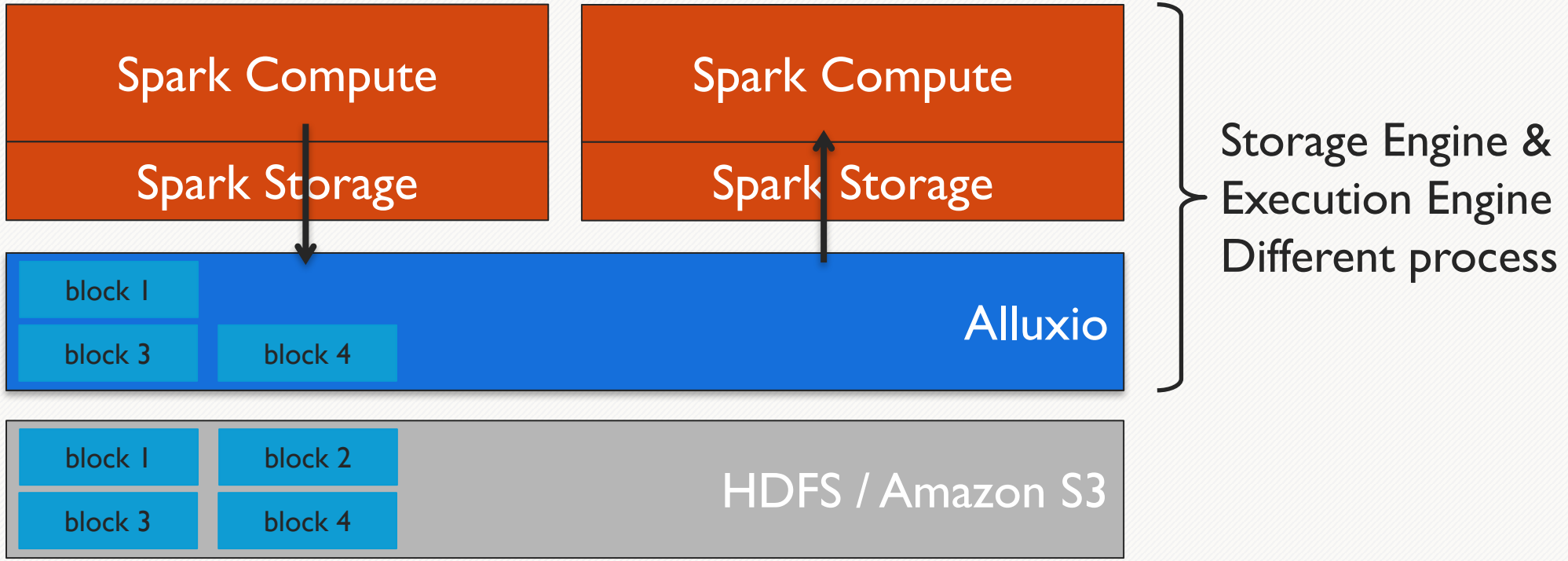
1. Alluxio Overview
2. Alluxio + Spark Use Cases
3. Using Spark with Alluxio
4. Performance Evaluation
5. Demo

Consolidating Memory



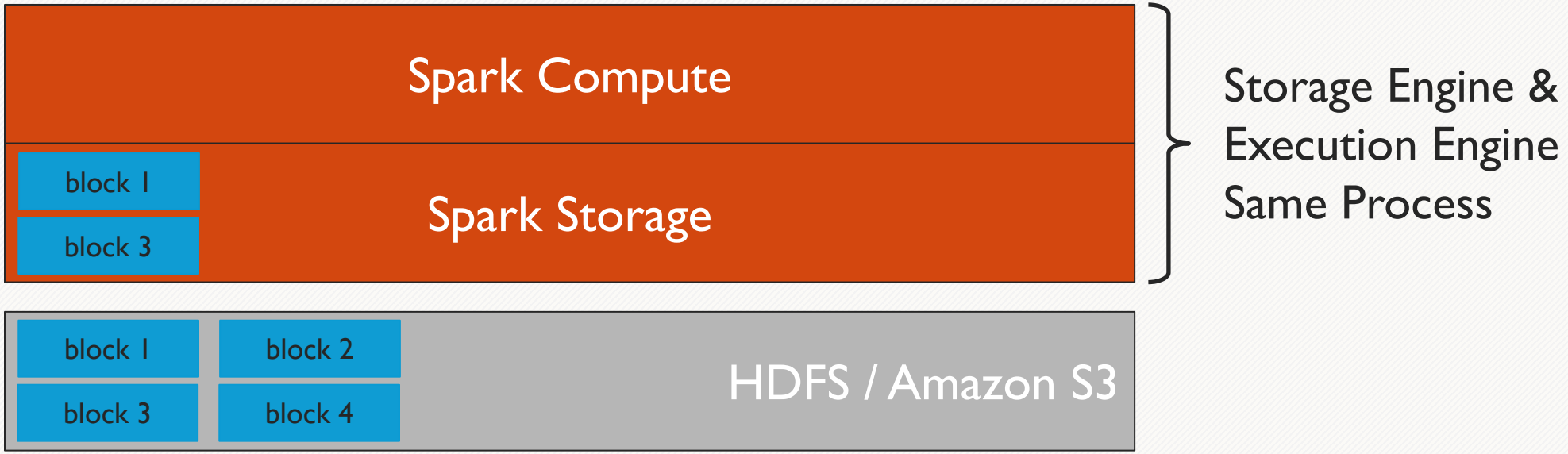
- Two copies of data in memory – double the memory used
- Inter-process Sharing Slowed Down by Network / Disk I/O

Consolidating Memory

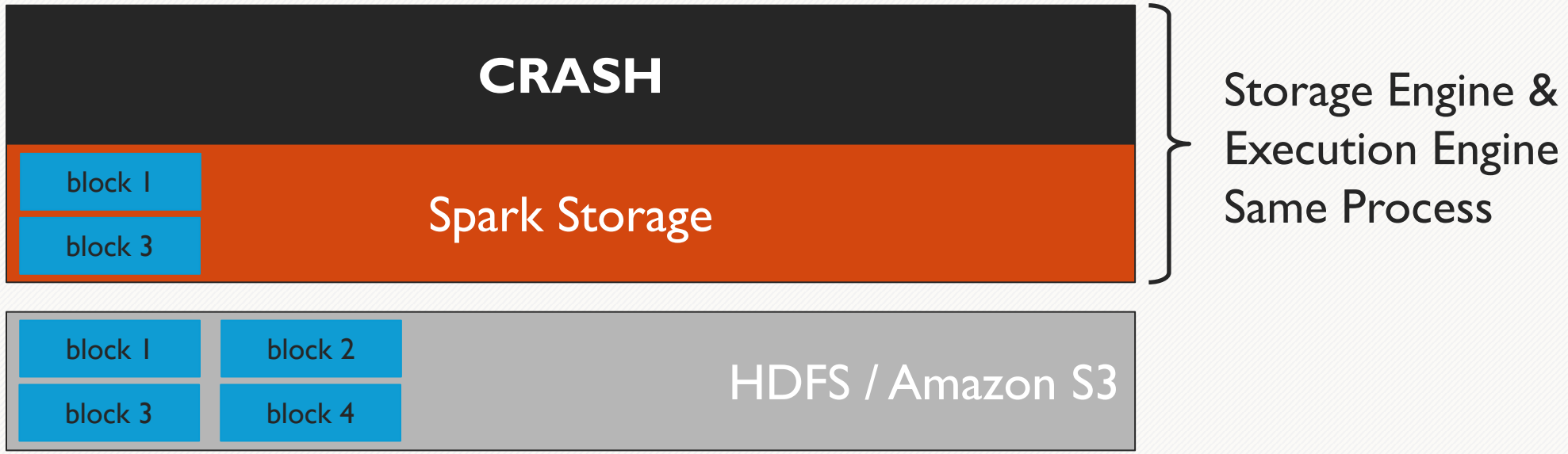


- Half the memory used
- Inter-process Sharing Happens at Memory Speed

Data Resilience During Crash

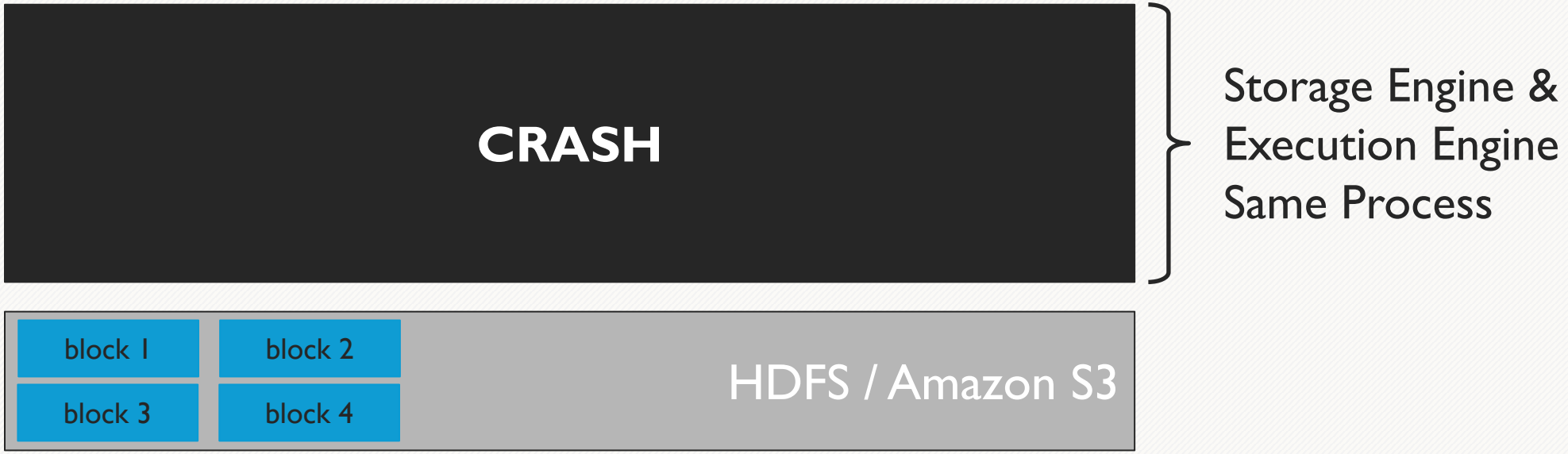


Data Resilience During Crash



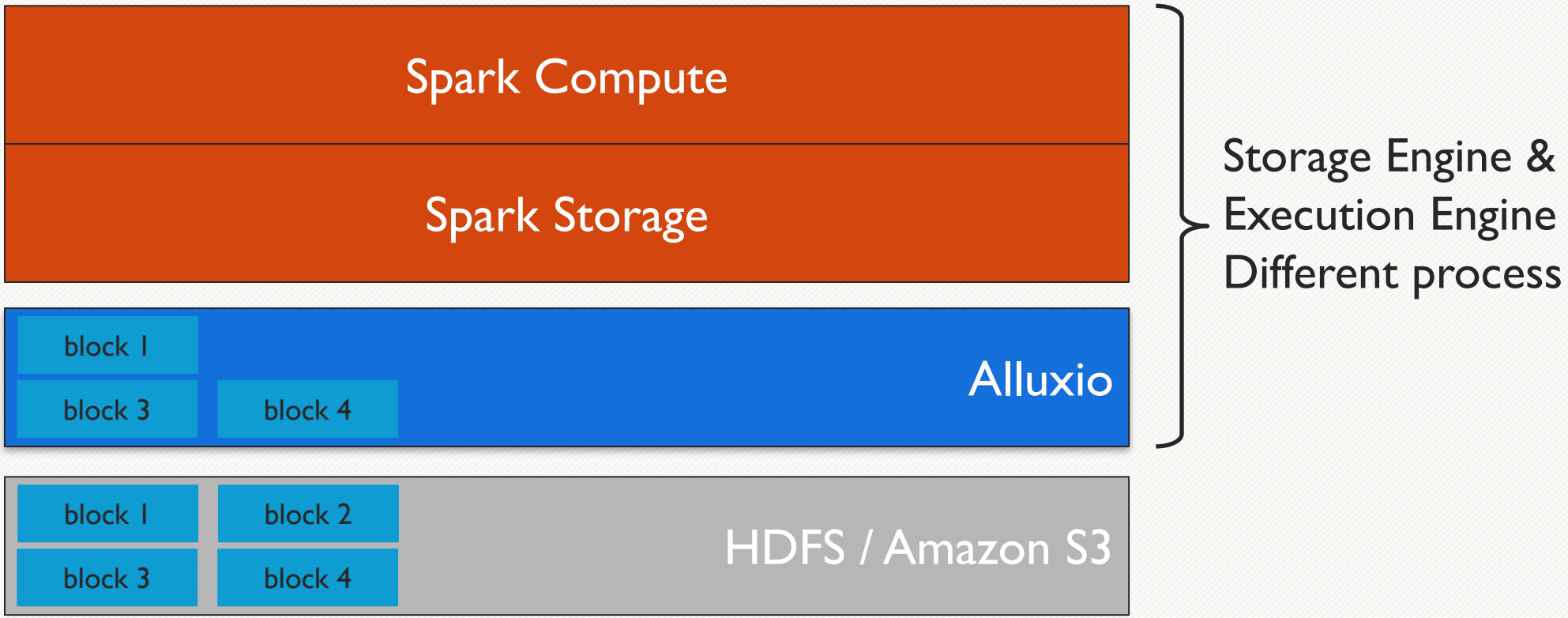
- Process Crash Requires Network and/or Disk I/O to Re-read Data

Data Resilience During Crash

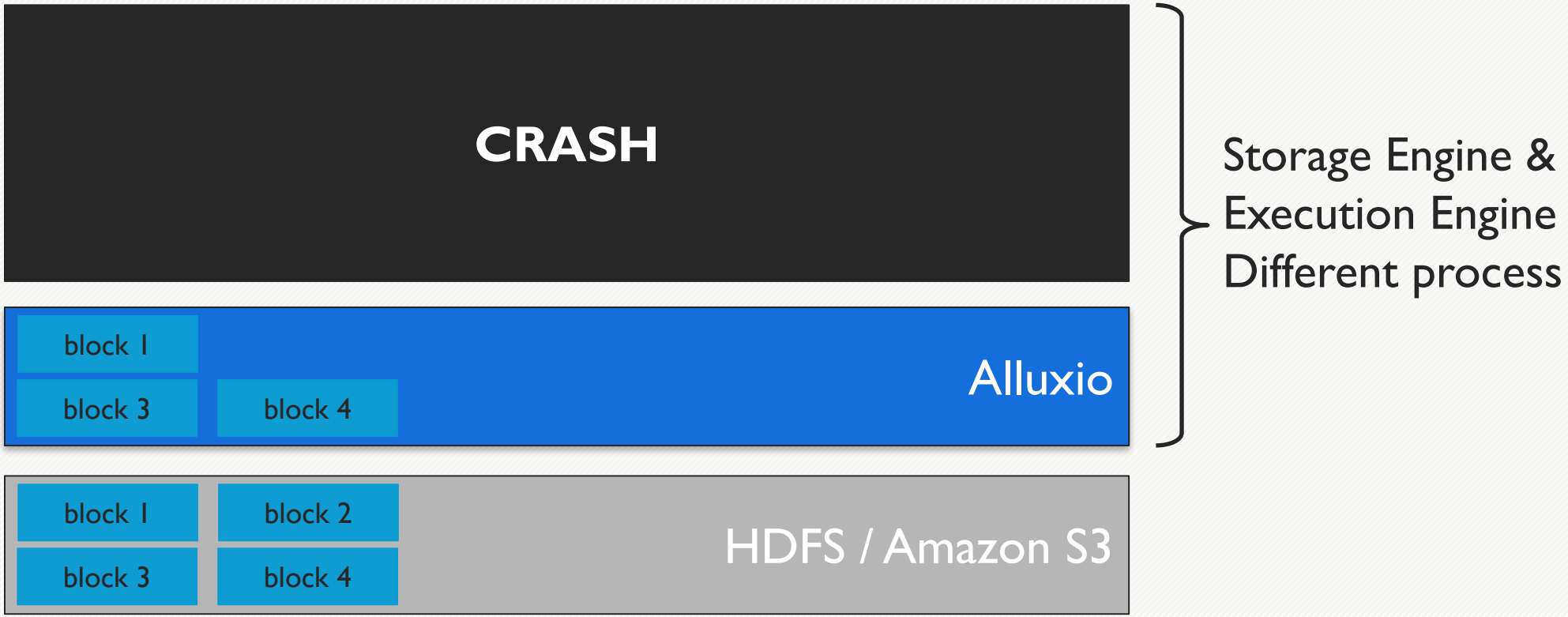


- Process Crash Requires Network and/or Disk I/O to Re-read Data

Data Resilience During Crash



Data Resilience During Crash



- Process Crash – Data is Re-read at Memory Speed

Accessing Alluxio Data From Spark

ALLUXIO

Writing Data	Write to an Alluxio file
Reading Data	Read from an Alluxio file

Code Example for Spark RDDs

ALLUXIO

Writing RDD to Alluxio

```
rdd.saveAsTextFile(alluxioPath)
rdd.saveAsObjectFile(alluxioPath)
```

Reading RDD from Alluxio

```
rdd = sc.textFile(alluxioPath)
rdd = sc.objectFile(alluxioPath)
```

Code Example for Spark DataFrames

ALLUXIO

Writing to Alluxio

```
df.write.parquet(alluxioPath)
```

Reading from Alluxio

```
df = sc.read.parquet(alluxioPath)
```

Outline

1. Alluxio Overview
2. Alluxio + Spark Use Cases
3. Using Spark with Alluxio
4. Performance Evaluation
5. Demo

Experiments

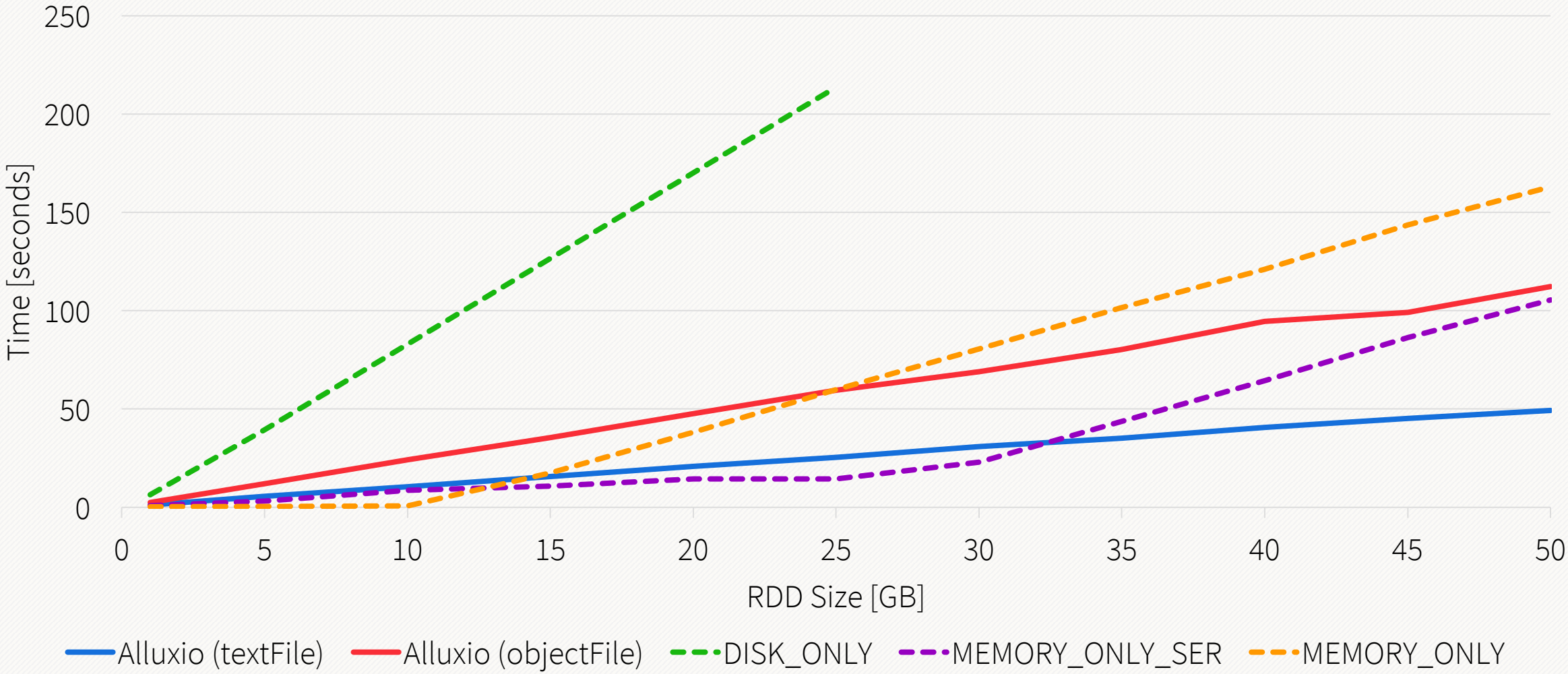
Spark 2.0.0 + Alluxio 1.2.0

Single worker: Amazon r3.2xlarge

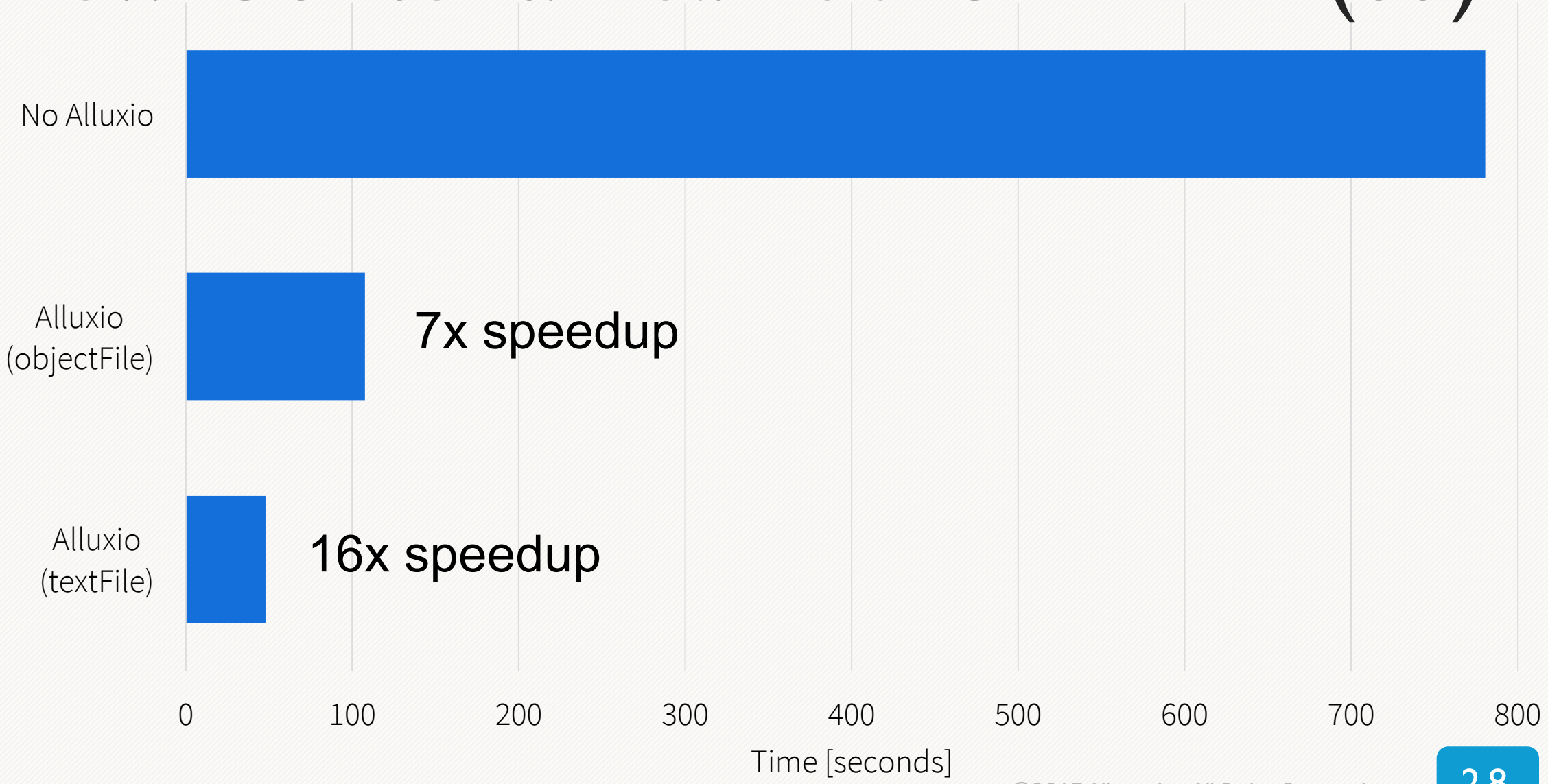
Comparisons:

- Alluxio
- Spark Storage Level: MEMORY_ONLY
- Spark Storage Level: MEMORY_ONLY_SER
- Spark Storage Level: DISK_ONLY

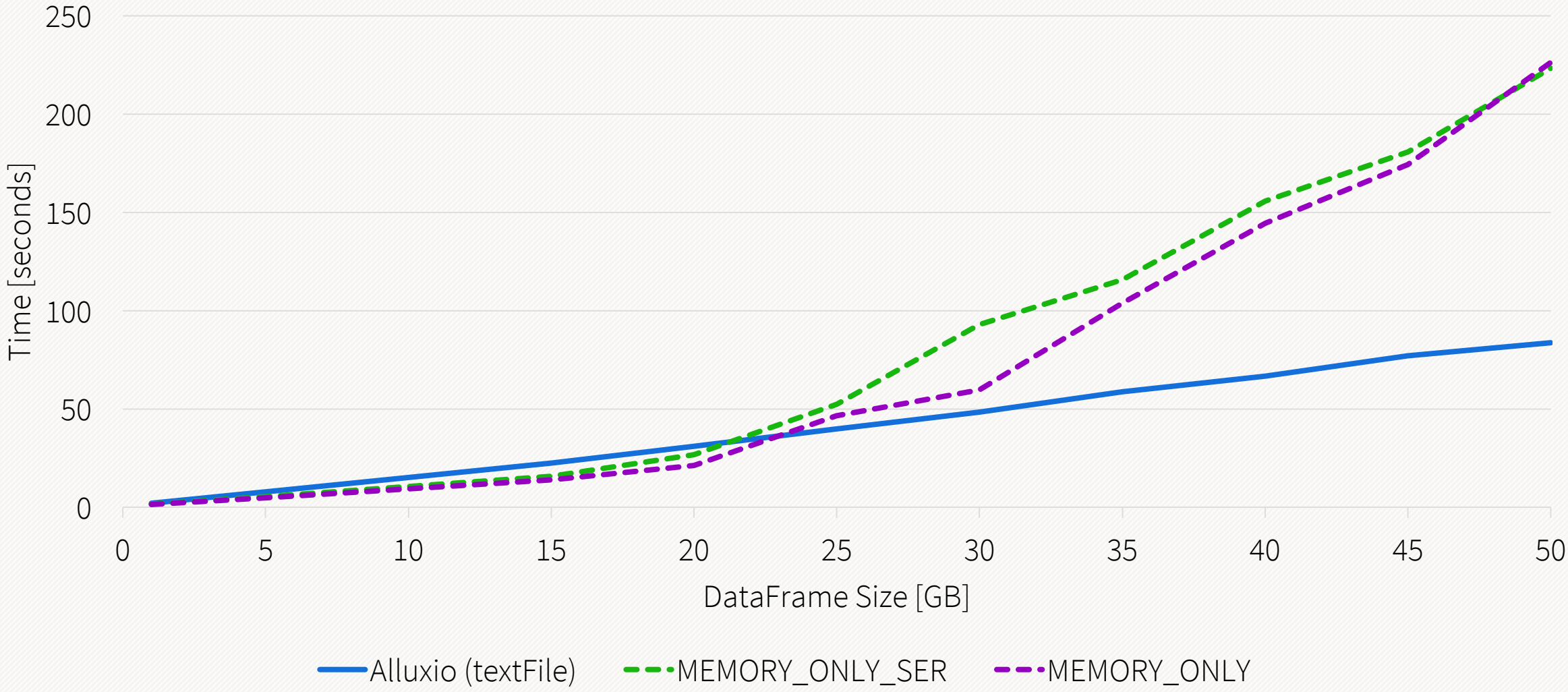
Reading Cached RDD



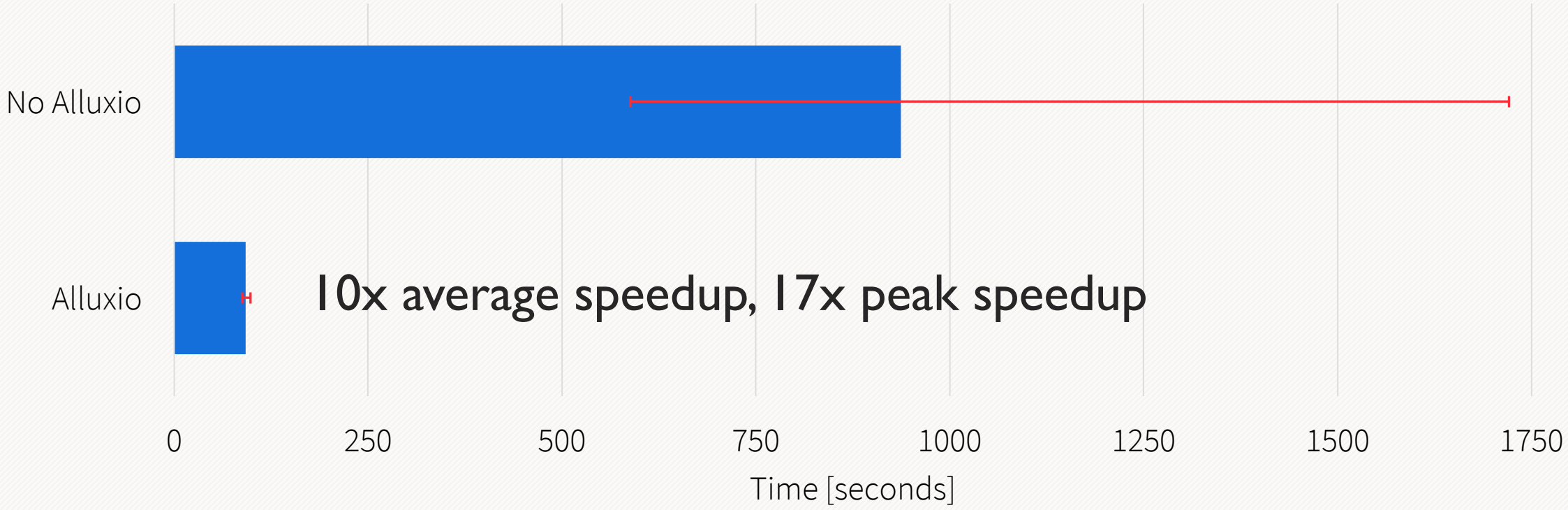
New Context: Read 50 GB RDD (S3)



Reading Cached DataFrame (parquet)



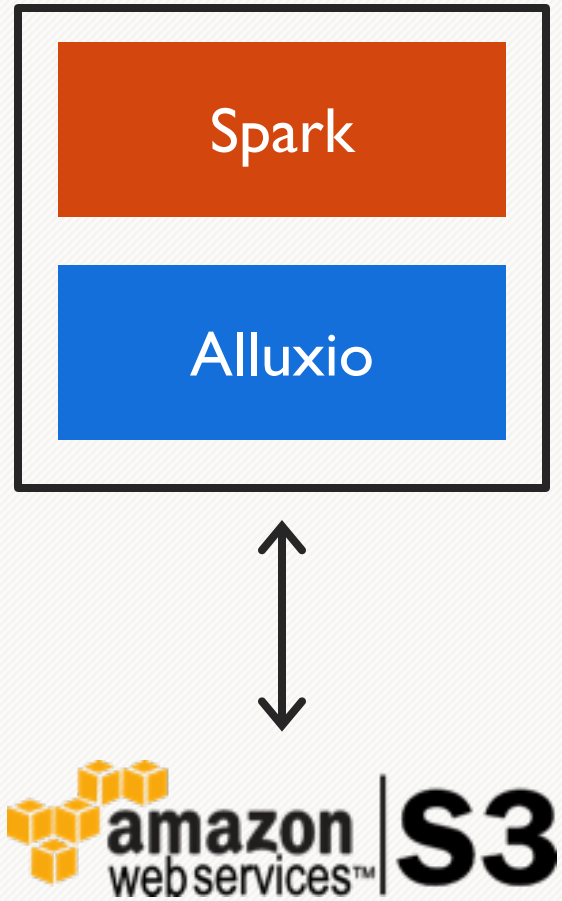
New Context: Read 50 GB DataFrame (S3)



Outline

1. Alluxio Overview
2. Alluxio + Spark Use Cases
3. Using Spark with Alluxio
4. Performance Evaluation
5. Demo

Demo Environment



Conclusion

Easy to use Alluxio with Spark

Predictable and improved performance

Easily connect to various storages

Thank you!

Gene Pang
gene@alluxio.com
Twitter: @unityxx

Cheng Chang
cc@alluxio.com
Twitter: @uronce

Website

www.alluxio.com

E-mail

info@alluxio.com

Social Media

Twitter.com/alluxio

Linkedin.com/alluxio