

# Best Practices for Using Alluxio with Spark

*Gene Pang, Alluxio, Inc.*

*Spark Summit EU - October 2017*

# • About Me

- Gene Pang
- Software engineer @ Alluxio, Inc.
- Alluxio open source PMC member
- Ph.D. from AMPLab @ UC Berkeley
- Worked at Google before UC Berkeley
- Twitter: @unityxx
- Github: @gpang



# Outline

- 1 Alluxio Overview
- 2 Alluxio + Spark Use Cases
- 3 Alluxio Architecture
- 4 Using Spark with Alluxio
- 5 Experiments

# • Data Ecosystem Yesterday



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

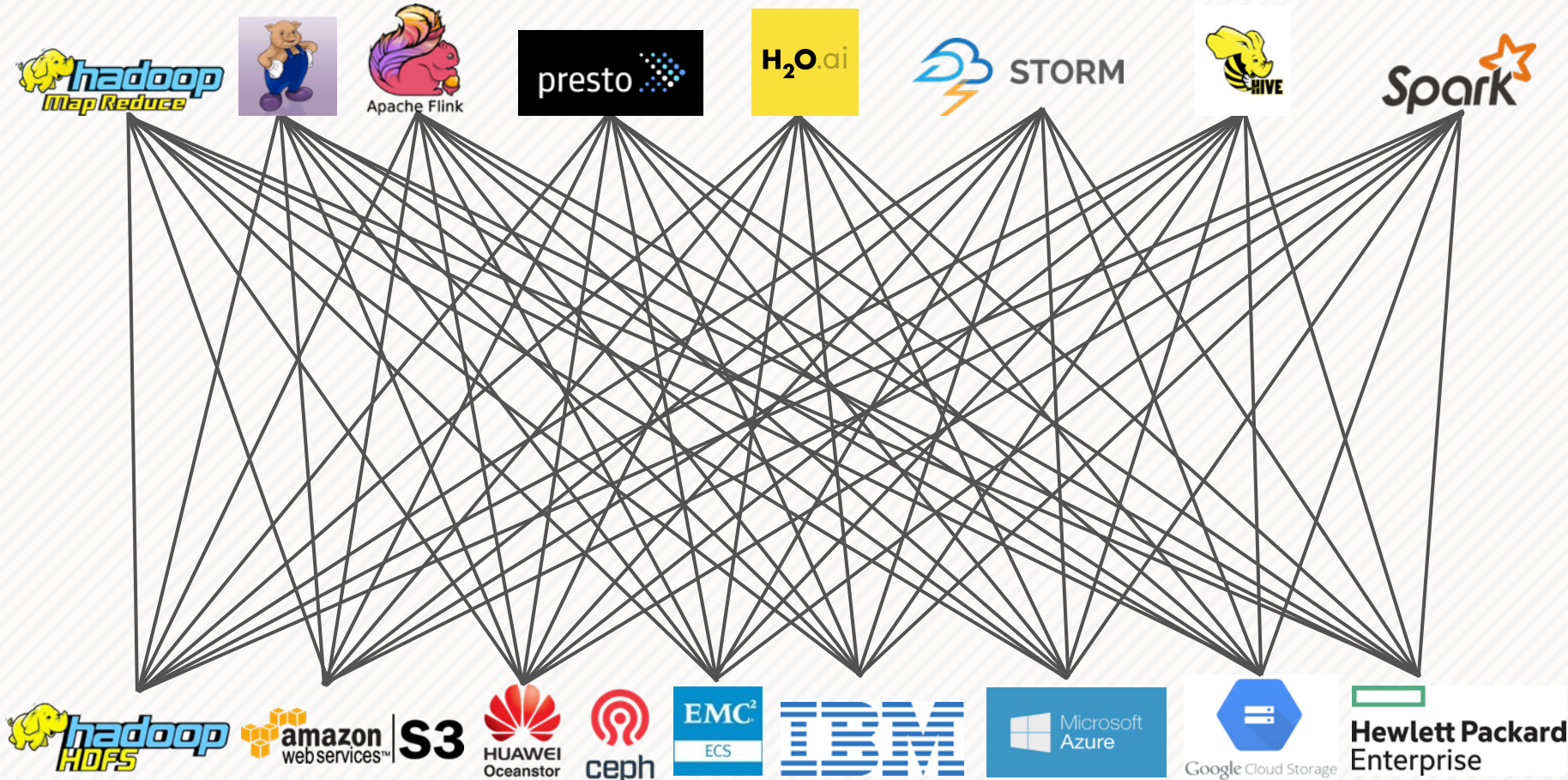
# • Data Ecosystem Today



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

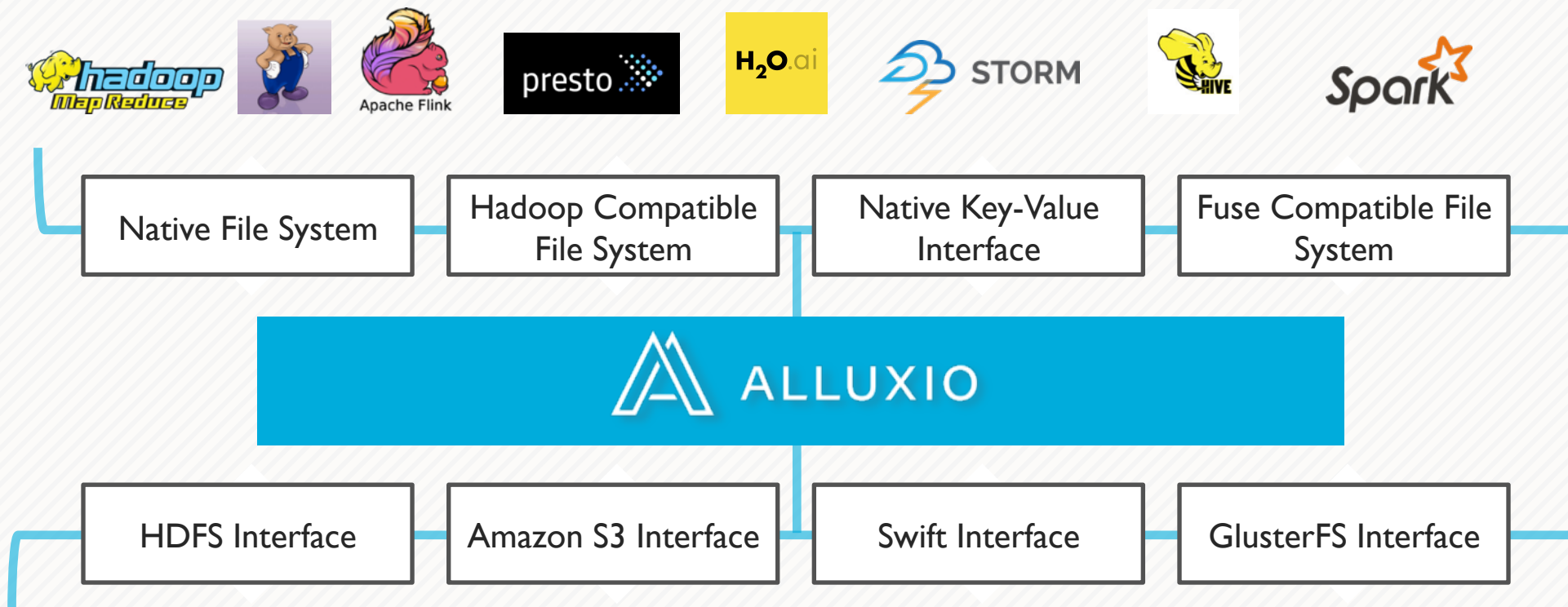


# Data Ecosystem Issues



- 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

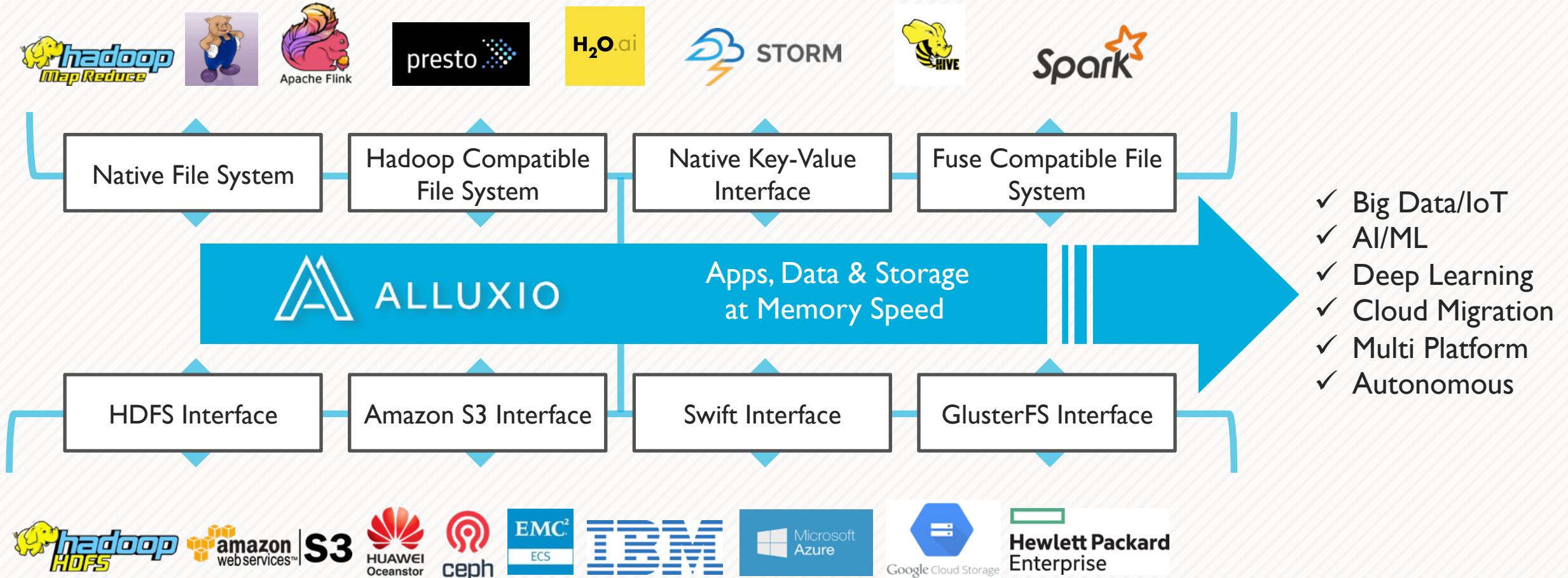


- Apps only talk to Alluxio
- Simple Add/Remove
- No App Changes
- Memory Performance





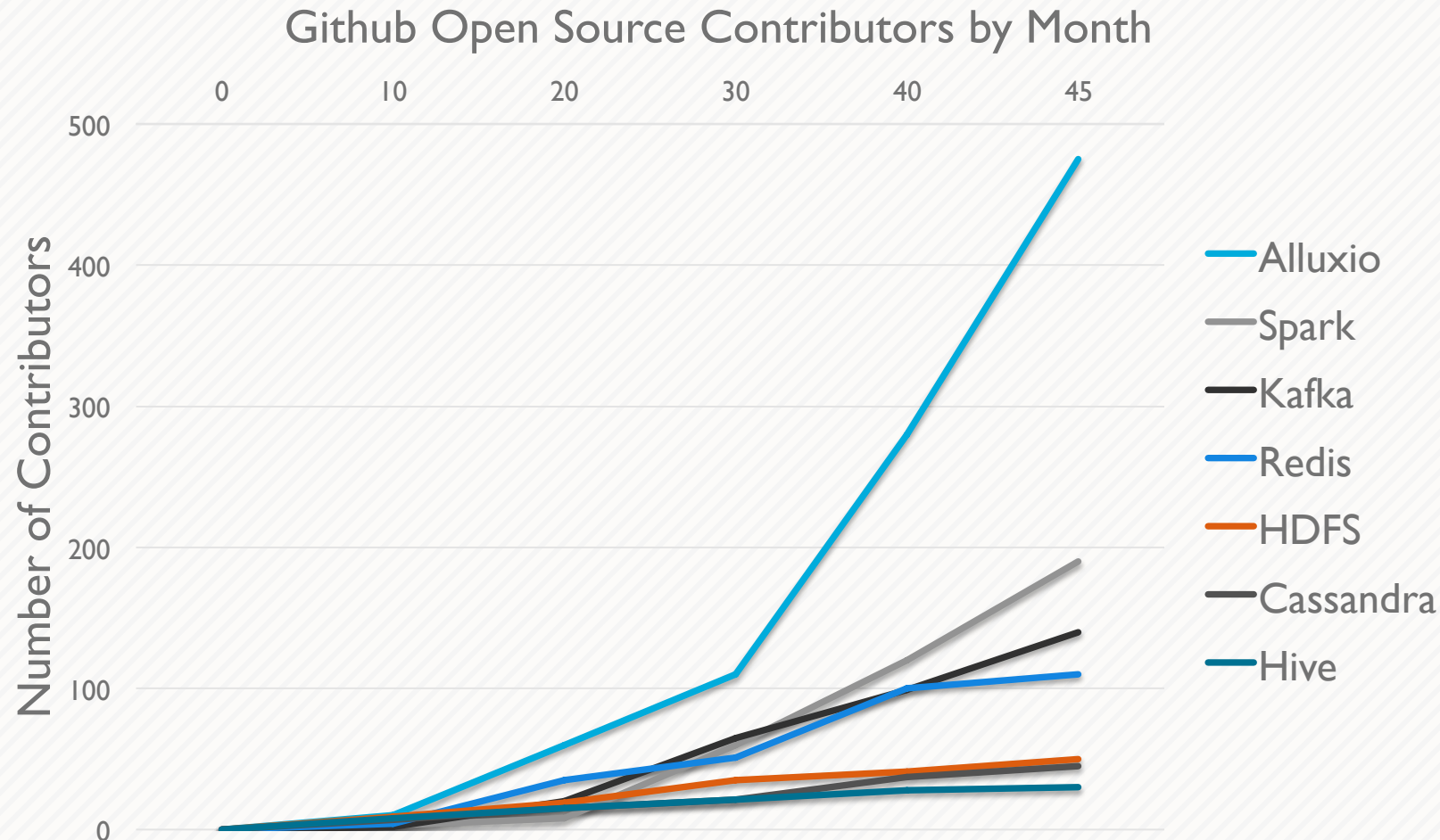
# Next Gen Analytics with Alluxio







# Fastest Growing Big Data Open Source Projects



Fastest Growing open-source project in the big data ecosystem

Running in large production clusters

600+ Contributors from 100+ organizations



# Outline

- 1 Alluxio Overview
- 2 Alluxio + Spark Use Cases
- 3 Alluxio Architecture
- 4 Using Spark with Alluxio
- 5 Experiments

# 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 – 去哪儿? 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



HDFS

CEPH

**Solution –**

With Alluxio, 300x improvement in performance

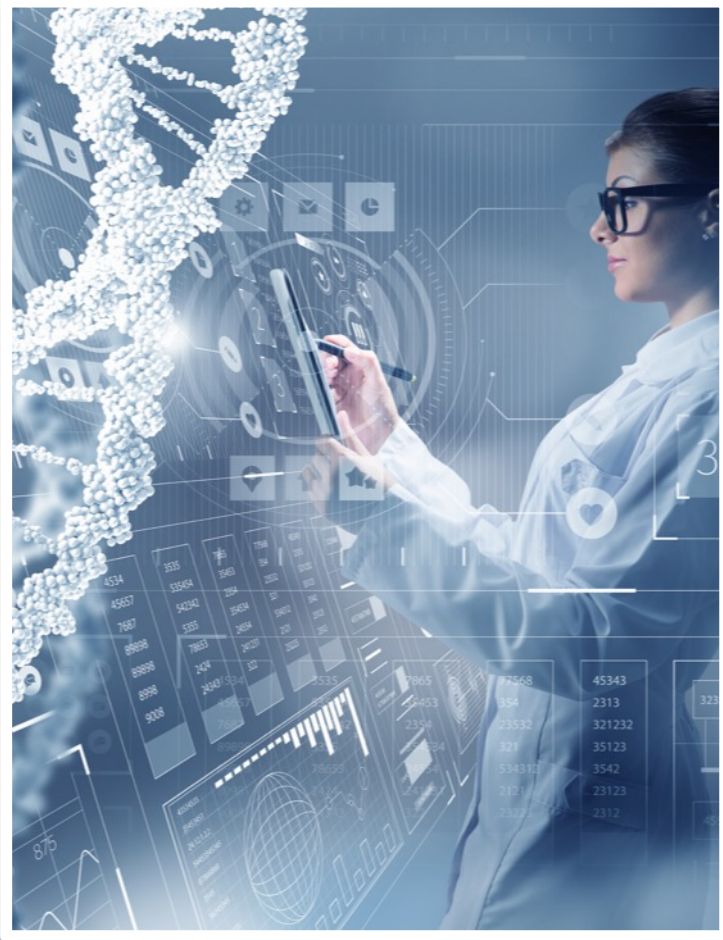
**Impact –**

Increased revenue from immediate response to user behavior

Use case: <http://bit.ly/2pDJdrq>



# Machine Learning Case Study –



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



MINIO

MESOS

## Solution –

Using Alluxio to prevent I/O bottlenecks

## Impact –

Orders of magnitude higher performance than before.  
<http://bit.ly/2p18ds3>

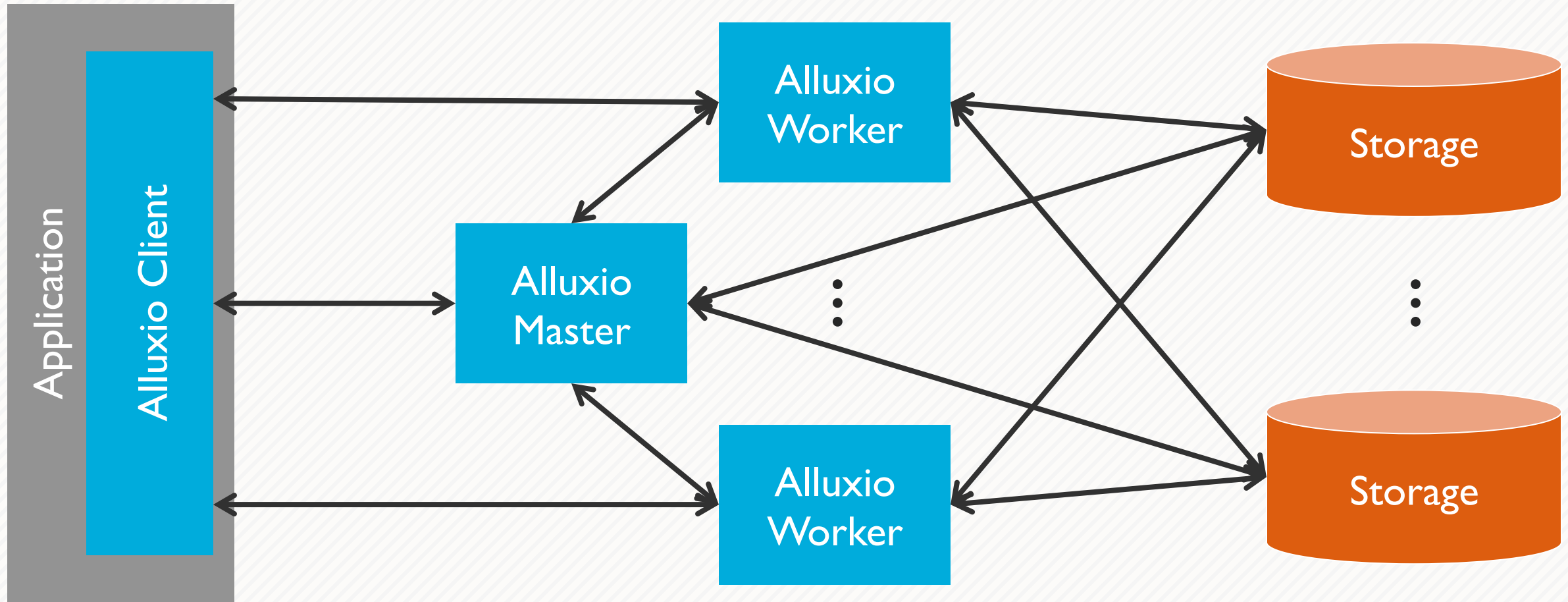


# Outline

- 1 Alluxio Overview
- 2 Alluxio + Spark Use Cases
- 3 Alluxio Architecture
- 4 Using Spark with Alluxio
- 5 Experiments



# Alluxio Architecture



# • Alluxio Client

Applications interact with Alluxio via the Alluxio client

- Java Native Alluxio Filesystem Client
  - Alluxio specific operations like [un]pin, [un]mount, [un]set TTL
- HDFS-Compatible Filesystem Client
  - No code change necessary
- S3 API

# • Alluxio Master

Master is responsible for managing metadata

- Filesystem namespace metadata
- Blocks / workers metadata

Primary master writes journal for durable operations

- Secondary masters replay journal entries

# • Alluxio Worker

Worker is responsible for managing block data

Worker stores block data on various storage media

- HDD, SSD, Memory

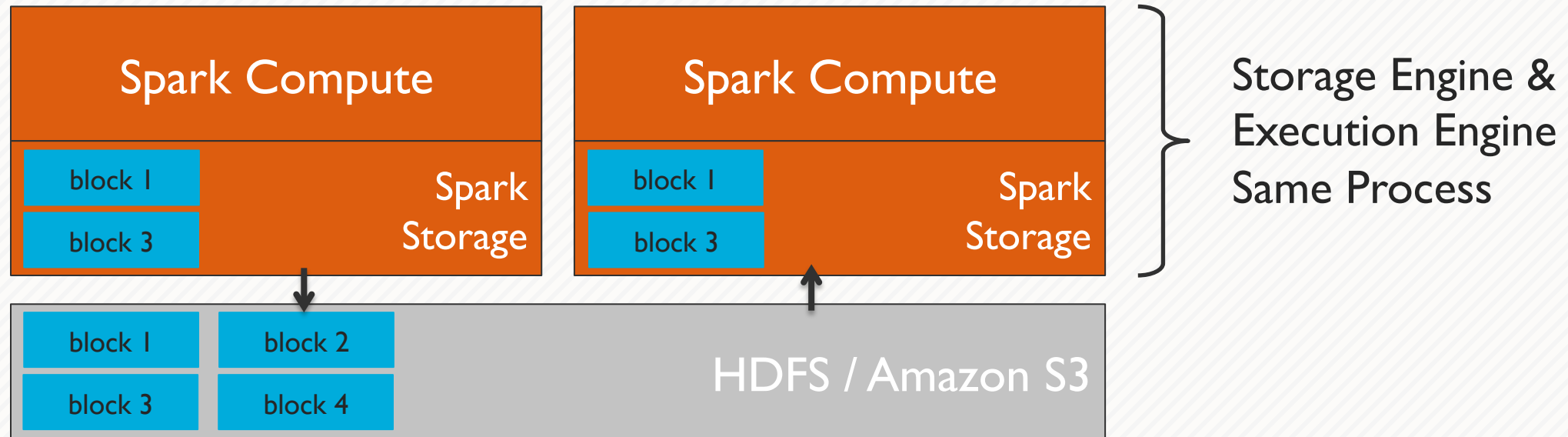
Reads and writes data to underlying storage systems



# Outline

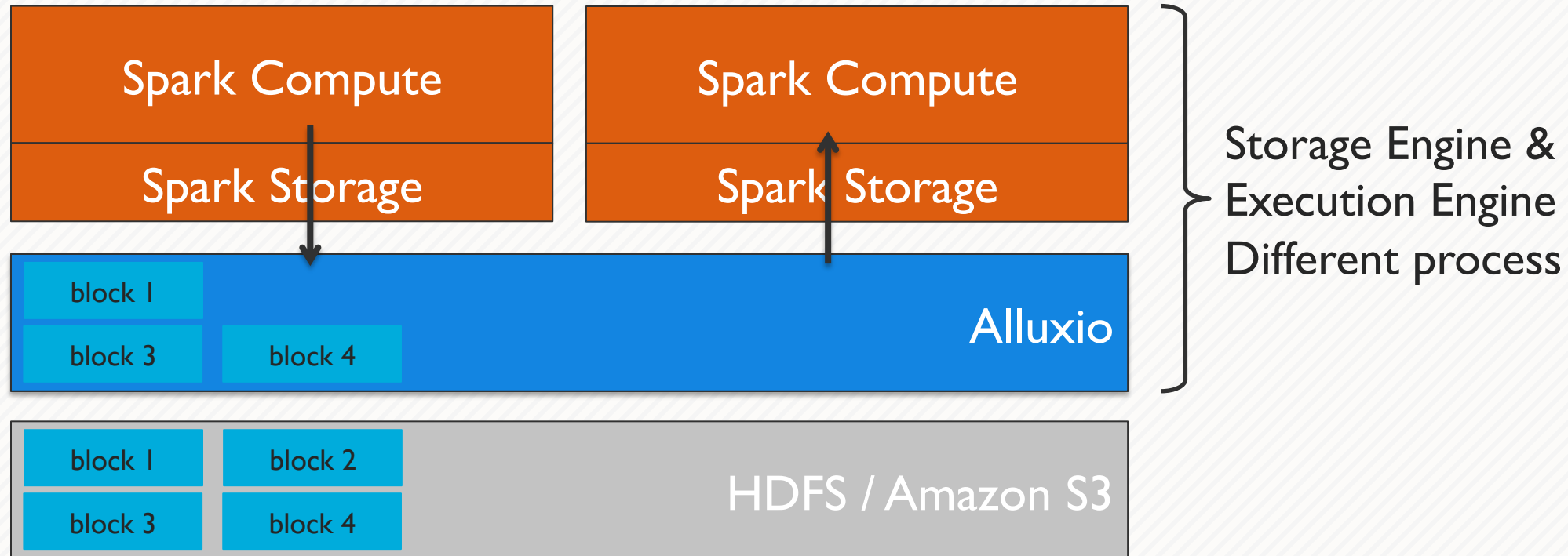
- 1 Alluxio Overview
- 2 Alluxio + Spark Use Cases
- 3 Alluxio Architecture
- 4 Using Spark with Alluxio
- 5 Experiments

# Sharing Data via Memory



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

# Sharing Data via Memory



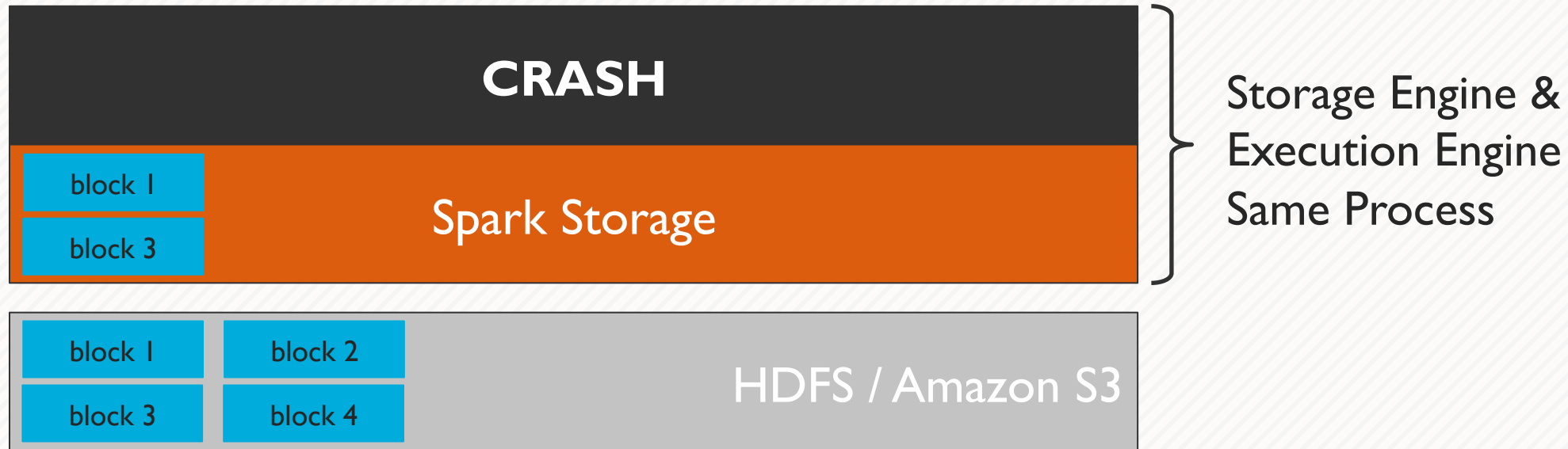
- Half the memory used
- Sharing Data at Memory Speed



# • Data Resilience During Crash

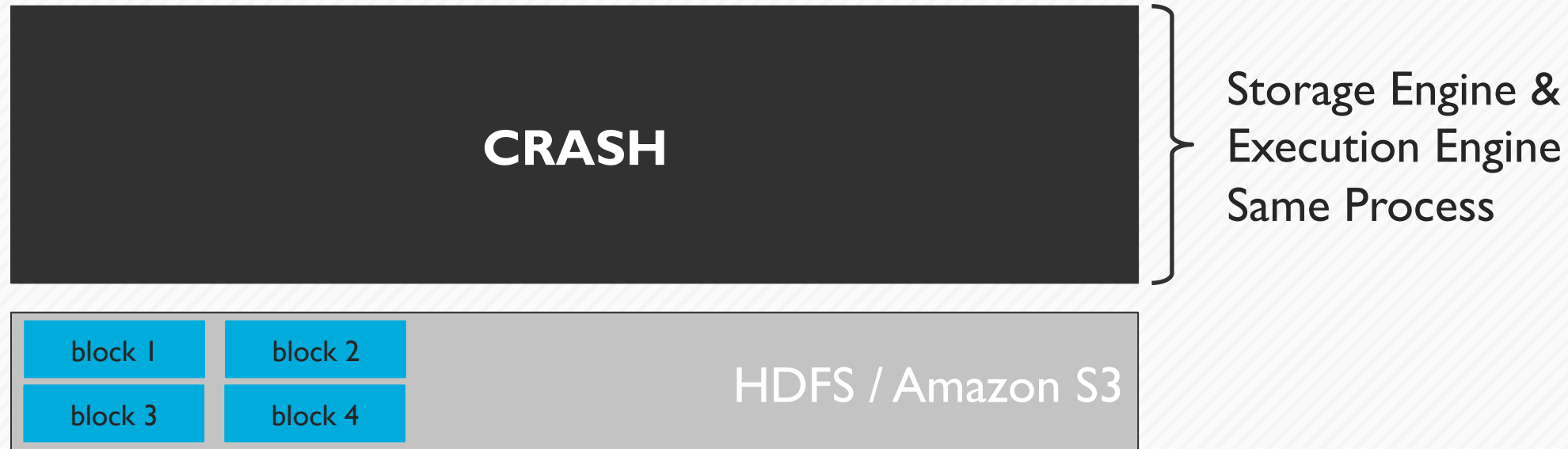


# • Data Resilience During Crash



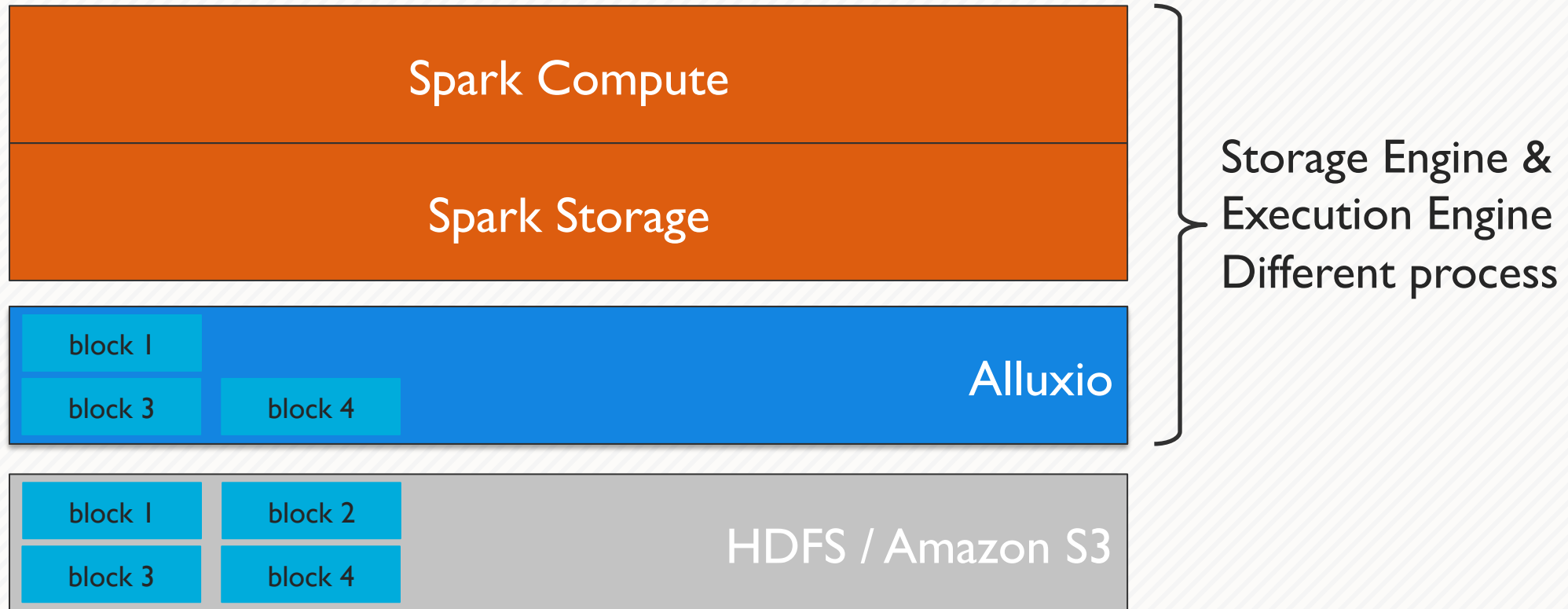
- Process Crash Requires Network and/or Disk I/O to Re-read Data

# • Data Resilience During Crash

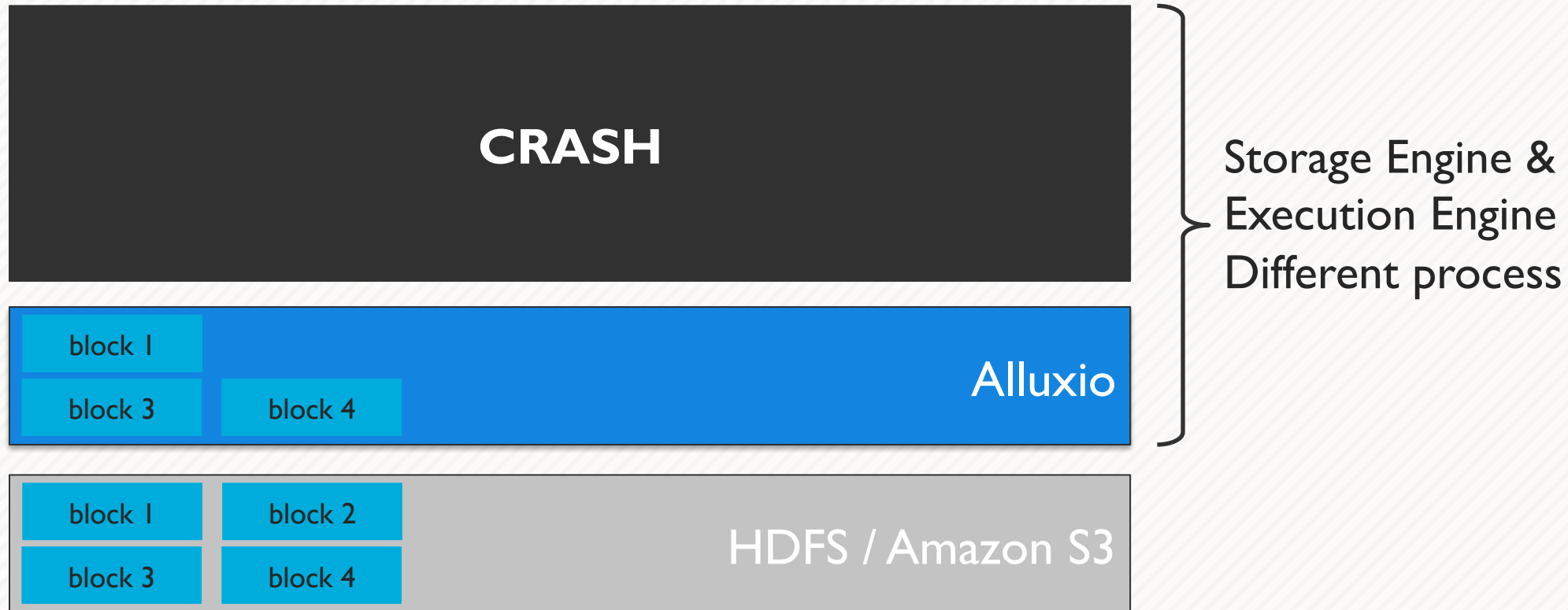


- Process Crash Requires Network and/or Disk I/O to Re-read Data

# • Data Resilience During Crash



# • Data Resilience During Crash



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

# • Accessing Alluxio Data From Spark

Writing Data

Write to an Alluxio file

Reading Data

Read from an Alluxio file

# • Code Example for Spark RDDs

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

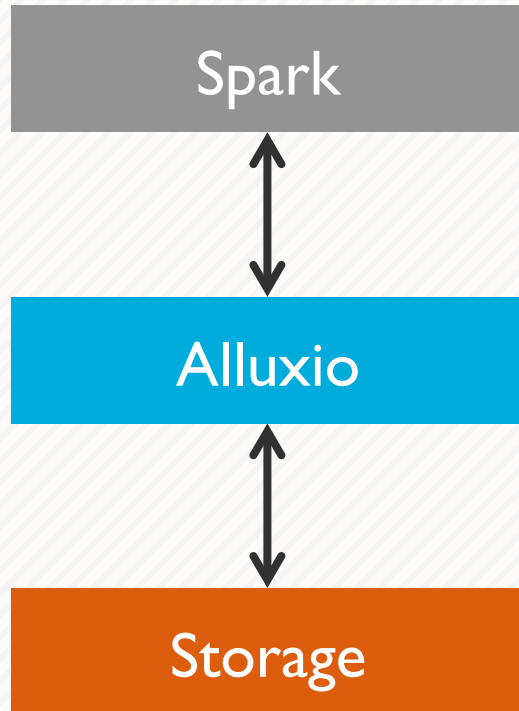
Writing to Alluxio

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

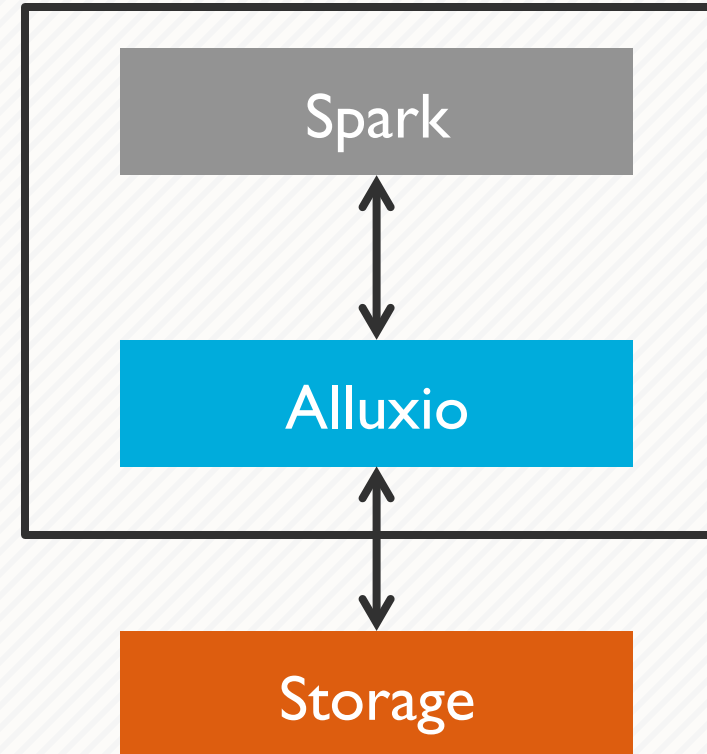
Reading from Alluxio

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

# Deploying Alluxio with Spark



Deploy Alluxio between  
Spark and Storage



Colocate Alluxio Workers with Spark  
for optimal I/O performance



# Outline

- 1 Alluxio Overview
- 2 Alluxio + Spark Use Cases
- 3 Alluxio Architecture
- 4 Using Spark with Alluxio
- 5 Experiments

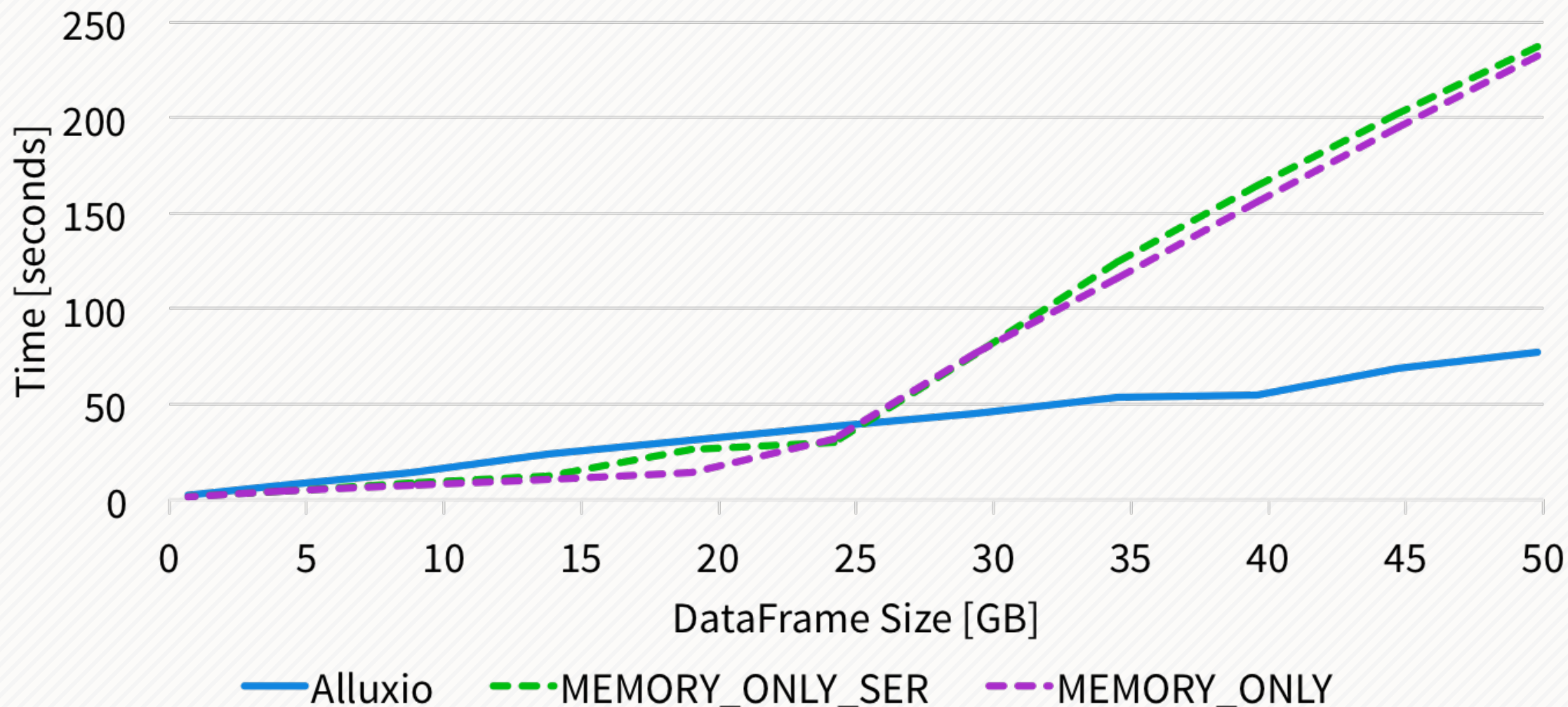
# • Experiments

Spark 2.2.0 + Alluxio 1.6.0

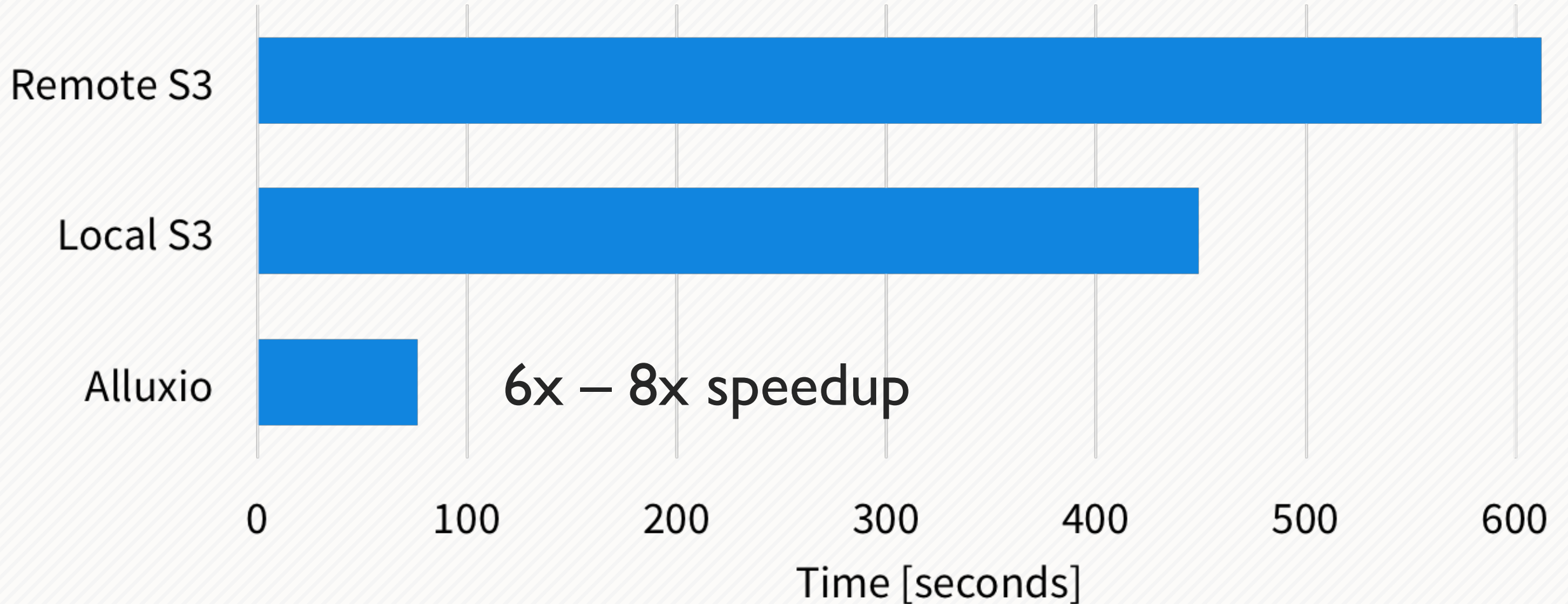
Single worker: Amazon r3.2xlarge

Compare reading cached parquet files

# Reading Cached DataFrame (parquet)



# New Context: 50 GB DataFrame (S3)



# • Conclusion

Easy to use Alluxio with Spark

Alluxio enables improved I/O performance

Easily interact with various storage systems with Alluxio



# Thank you!

Gene Pang  
gene@alluxio.com  
Twitter: @unityxx



## Website

[www.alluxio.com](http://www.alluxio.com)



## E-mail

[info@alluxio.com](mailto:info@alluxio.com)



## Social Media

○ [Twitter.com/alluxio](https://twitter.com/alluxio)

\* [Linkedin.com/alluxio](https://www.linkedin.com/company/alluxio)