TACHYON

A Reliable Memory-Centric Distributed Storage System

Haoyuan Li

October 16 @ Strata & Hadoop World NYC

Website: tachyon-project.org
Meetup: www.meetup.com/Tachyon

amplab
UC Berkeley

Outline

- Overview
  - Feature 1: Memory Centric Storage Architecture
  - Feature 2: Lineage in Storage
- Open Source
- Roadmap

Outline

- **Overview**
  - Feature 1: Memory Centric Storage Architecture
  - Feature 2: Lineage in Storage
- Open Source
- Roadmap

amplab UC Berkeley Projects

- Design next generation data analytics stack: Berkeley Data Analytics Stack (BDAS)

a cluster manager making it easy to write and deploy distributed applications.

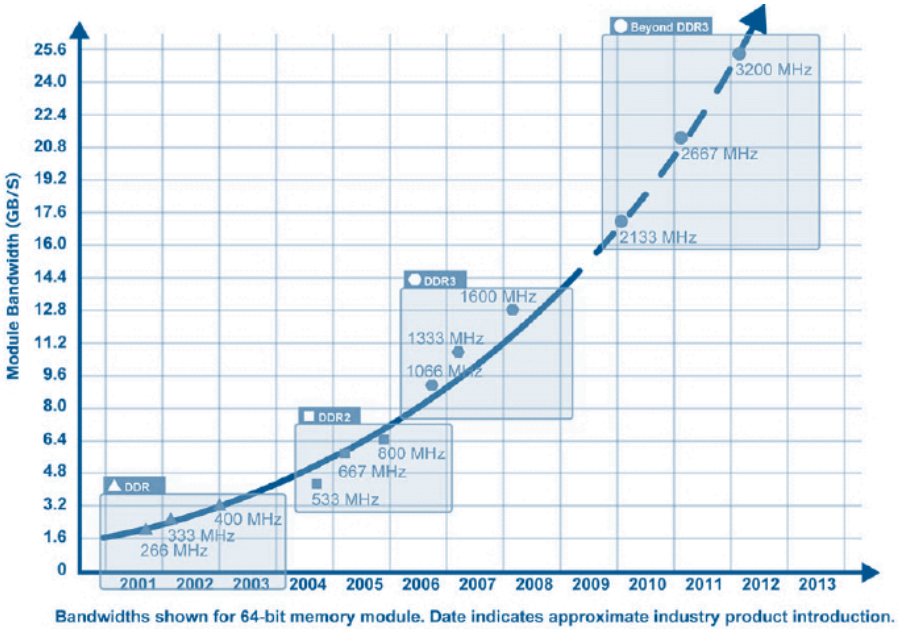
a parallel computing system supporting general and efficient in-memory execution.

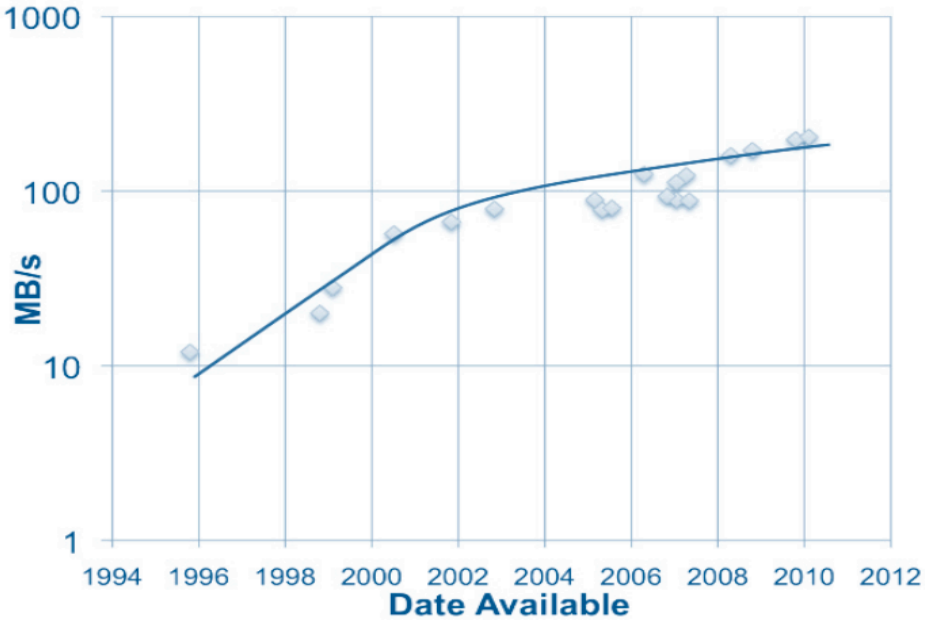
TACHYON a reliable distributed memory-centric storage enabling memory-speed data sharing.

Why Tachyon?

Memory is King

- RAM throughput increasing exponentially
- Disk throughput increasing slowly

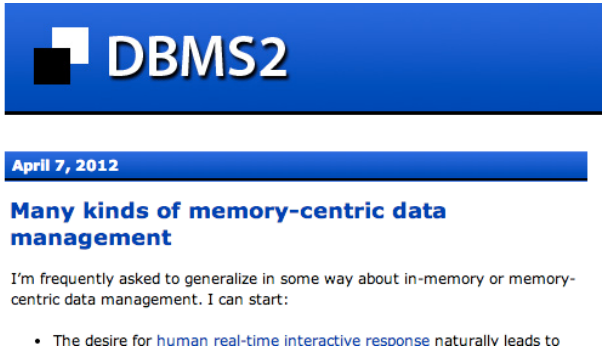




Memory-locality key to interactive response time

Realized by many…

- Frameworks already leverage memory

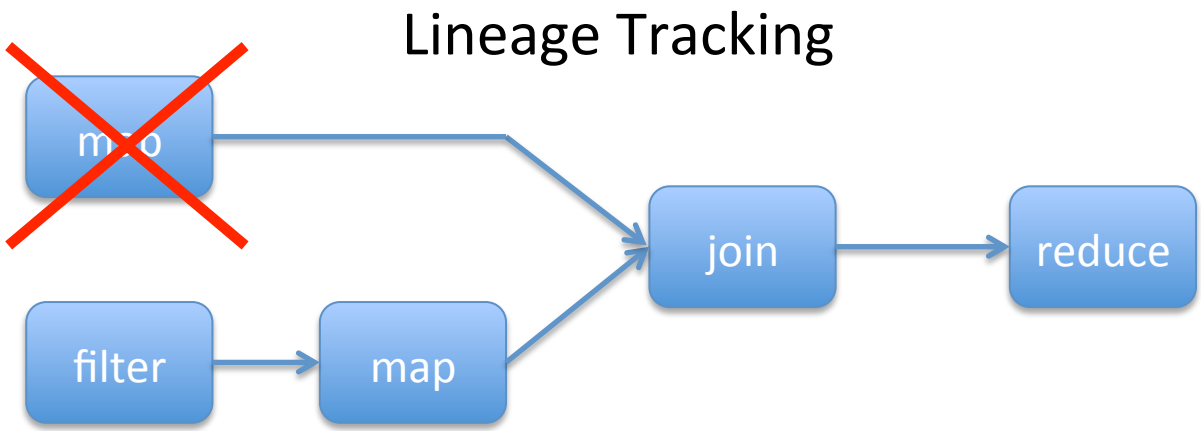


Problem solved?

Missing a Solution for Storage Layer

An Example: Spark

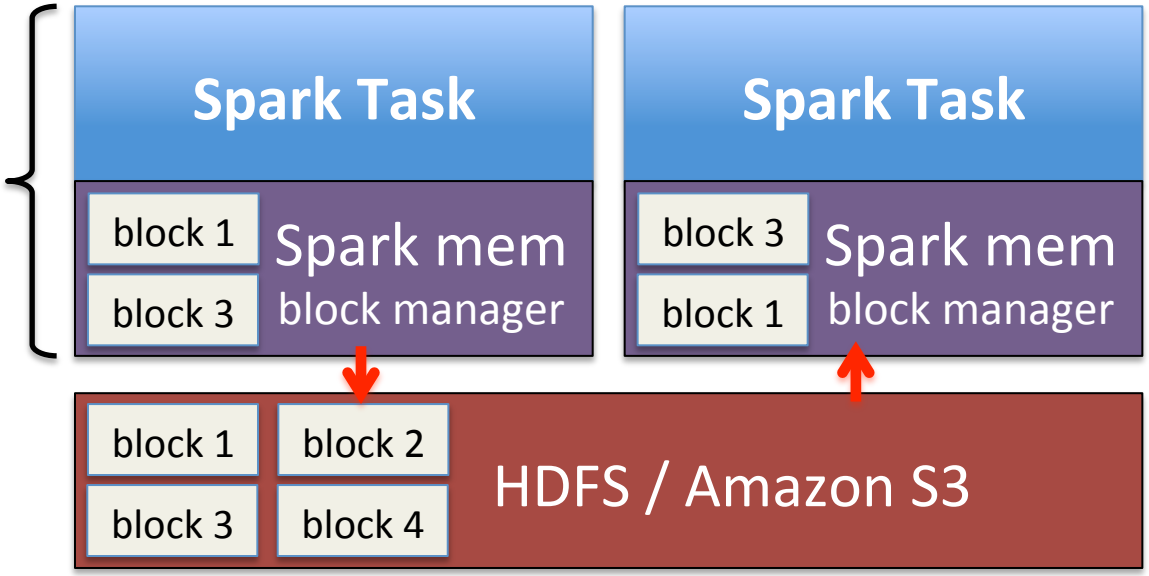
- Fast in-memory data processing framework
  - Keep one in-memory copy inside JVM
  - Track lineage of operations used to derive data
  - Upon failure, use lineage to recompute data



Issue 1

Data Sharing is the bottleneck in analytics pipeline: Slow writes to disk

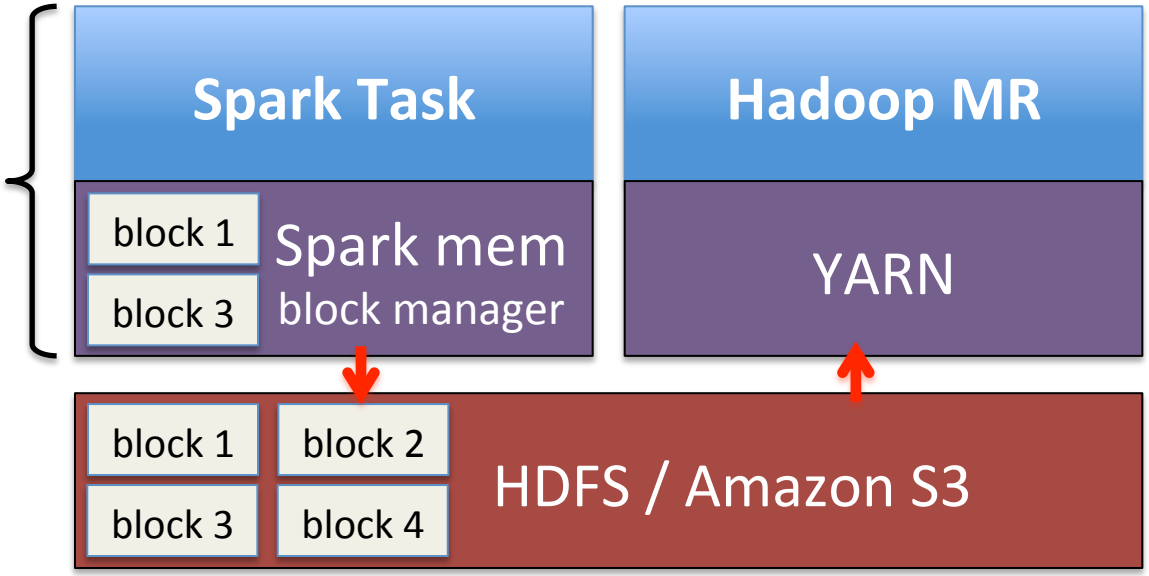
storage engine & execution engine same process (slow writes)



Issue 1

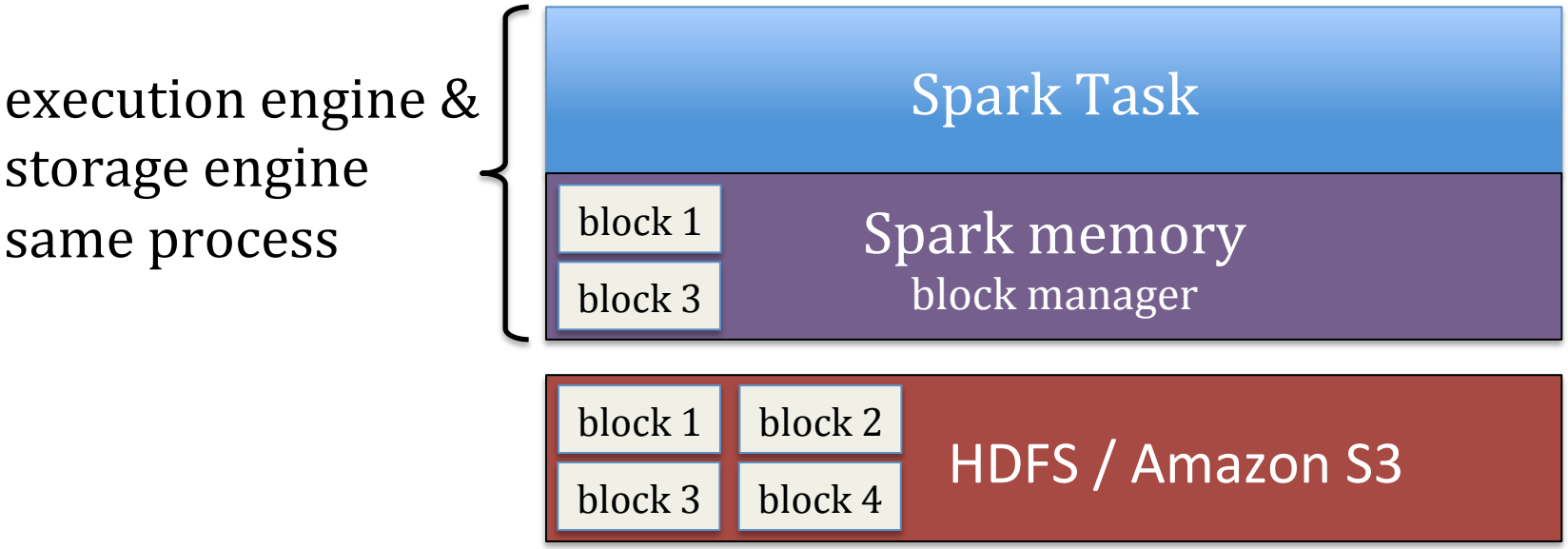
Data Sharing is the bottleneck in analytics pipeline: Slow writes to disk

storage engine & execution engine same process (slow writes)



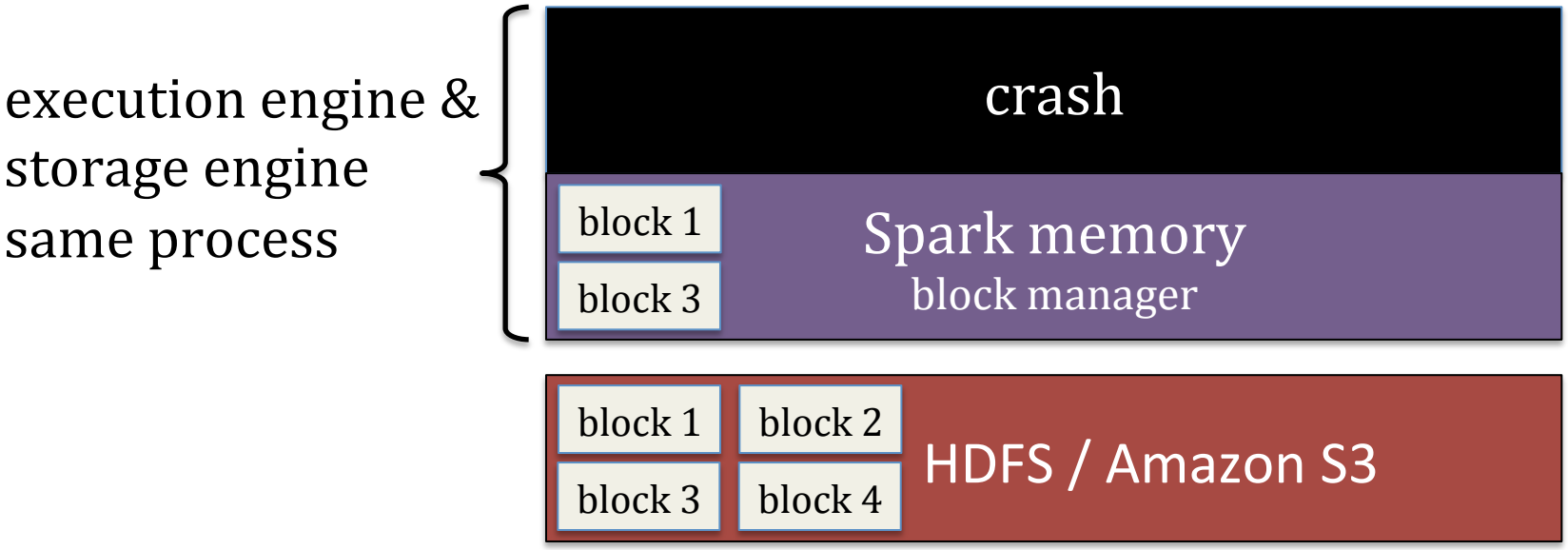
Issue 2

Cache loss when process crashes.



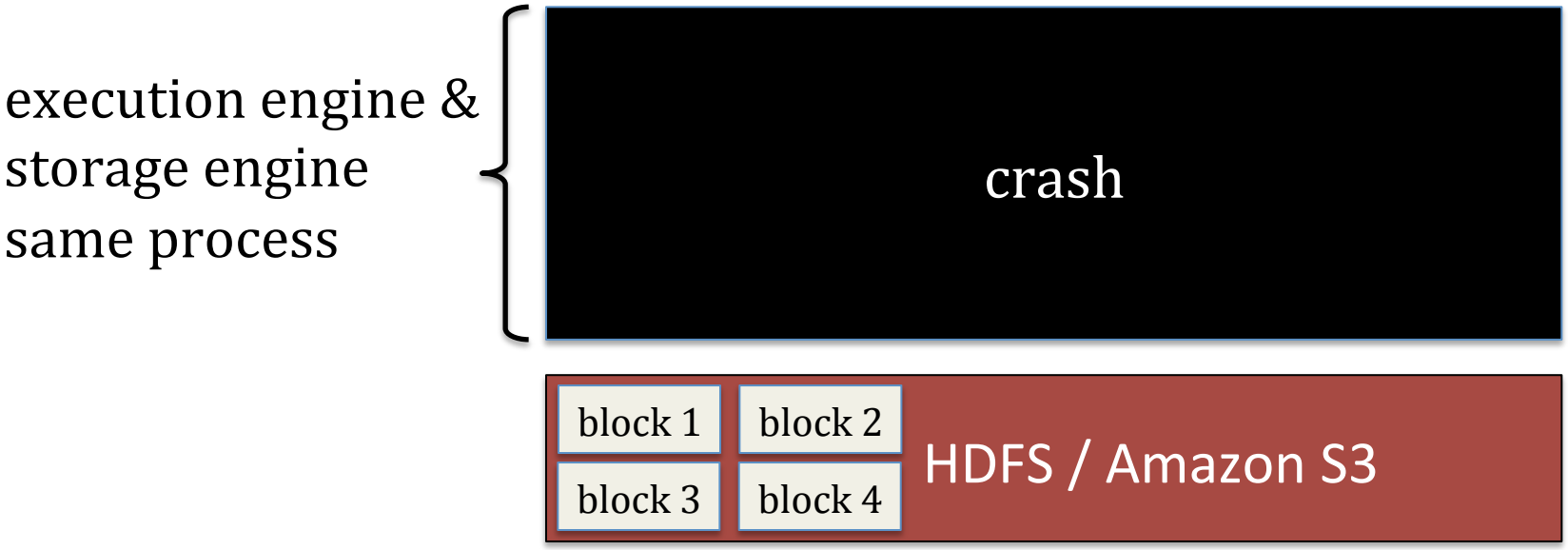
Issue 2

Cache loss when process crashes.



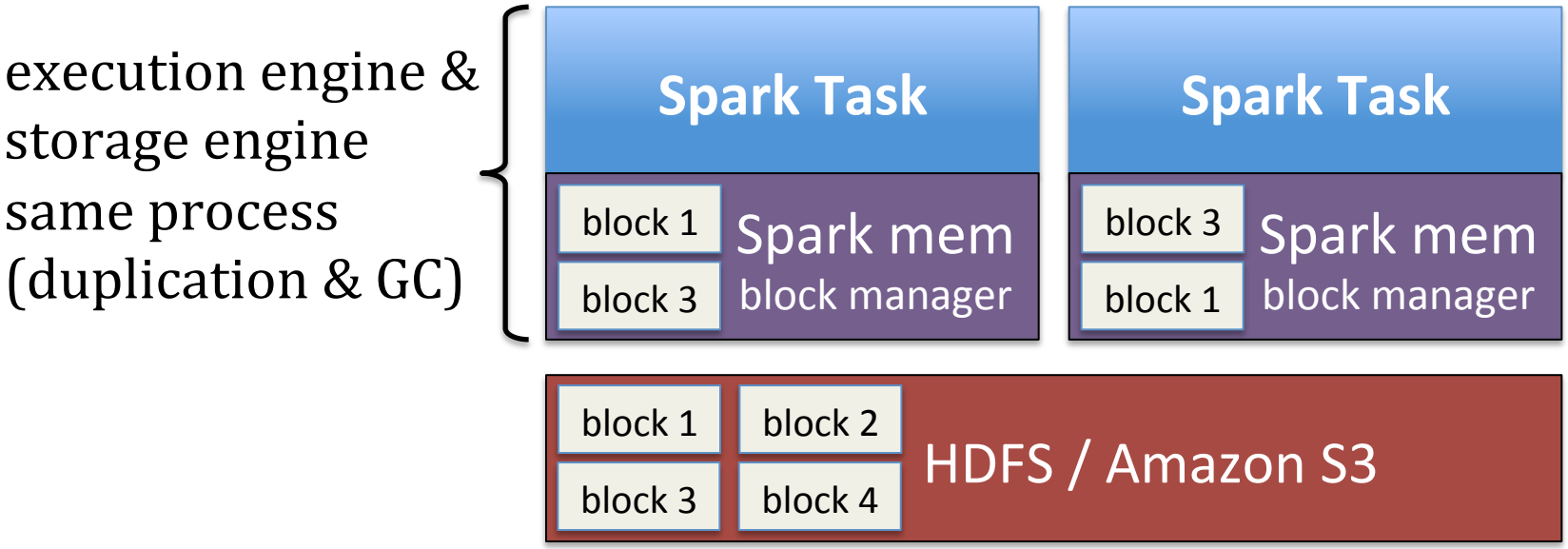
Issue 2

Cache loss when process crashes.



Issue 3

In-memory Data Duplication & Java Garbage Collection



Tachyon

Reliable data sharing at ***memory-speed within and across*** cluster frameworks/jobs

Solution Overview

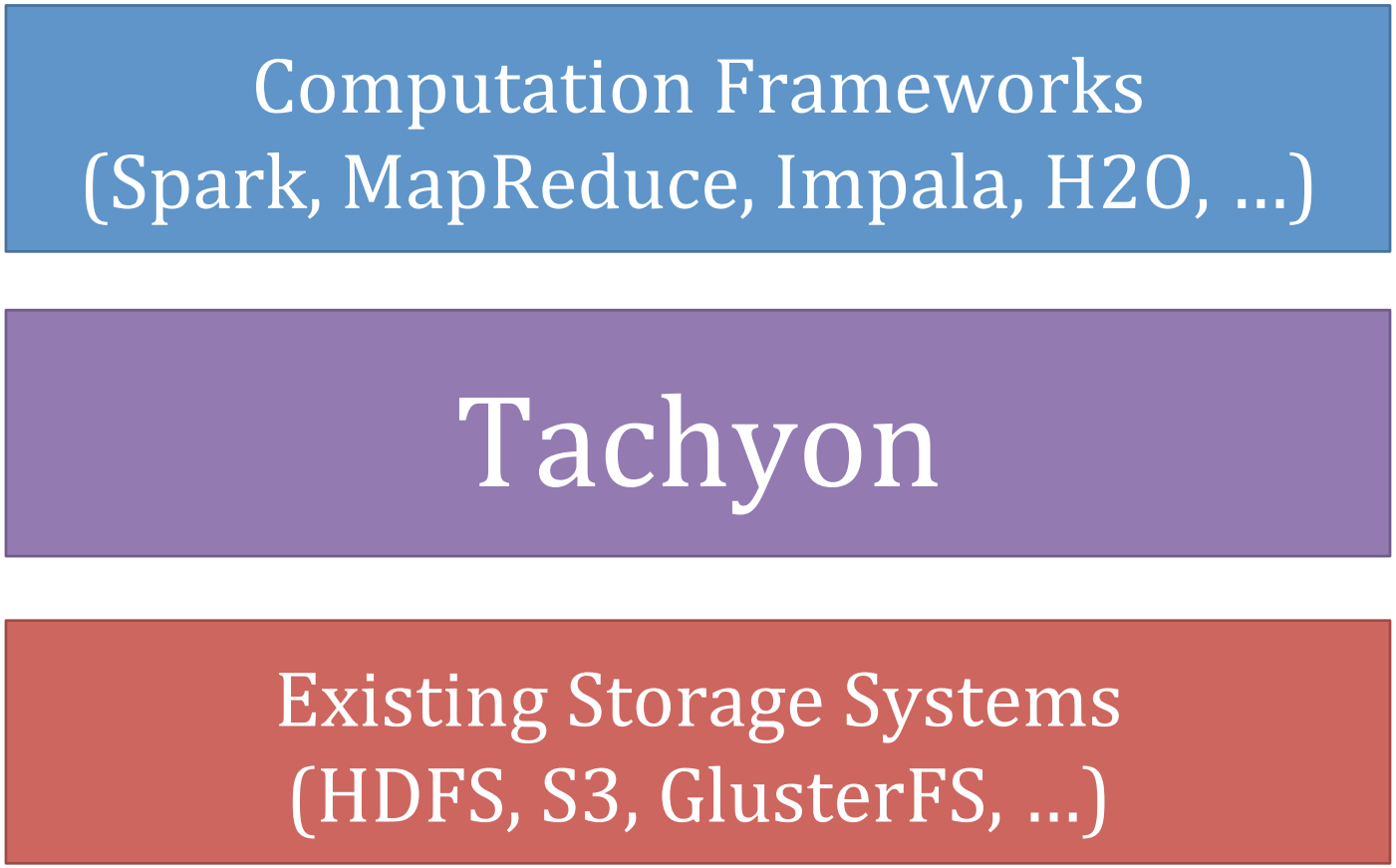
Basic idea

- Feature 1: memory-centric storage architecture
- Feature 2: push lineage down to storage layer

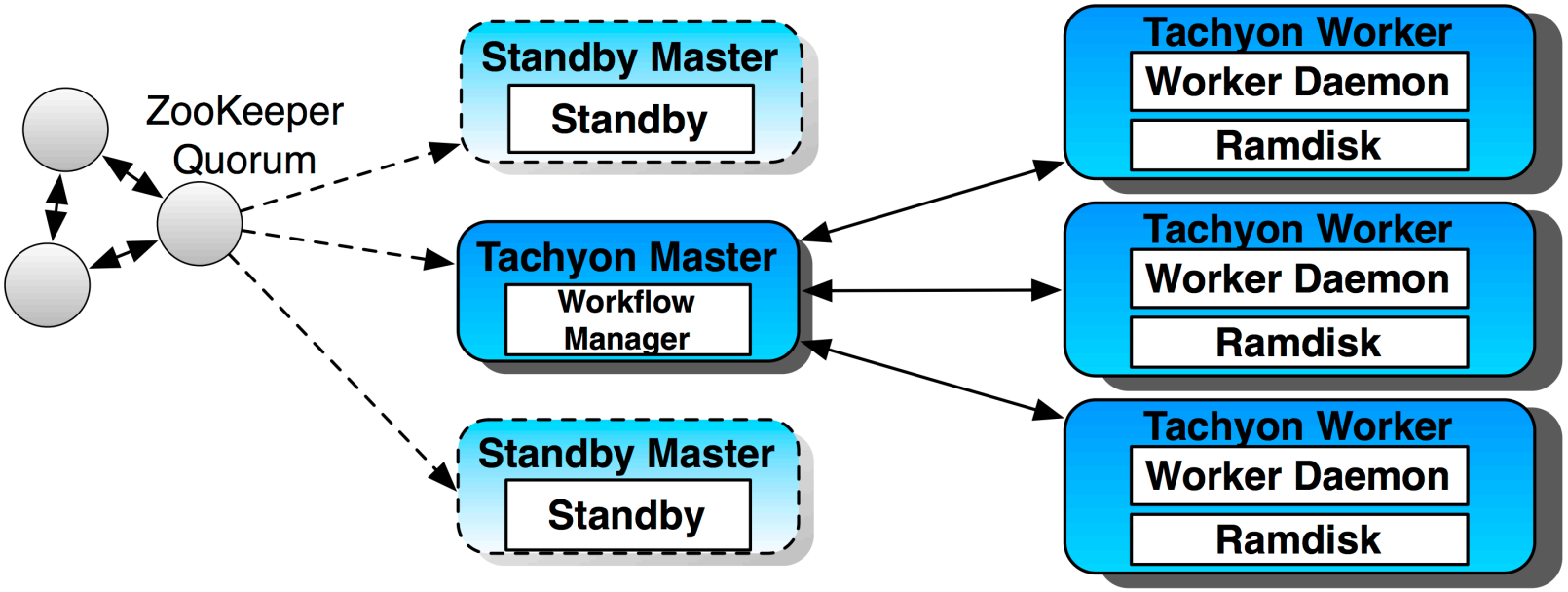
Facts

- One data copy in memory
- Recomputation for fault-tolerance

Stack

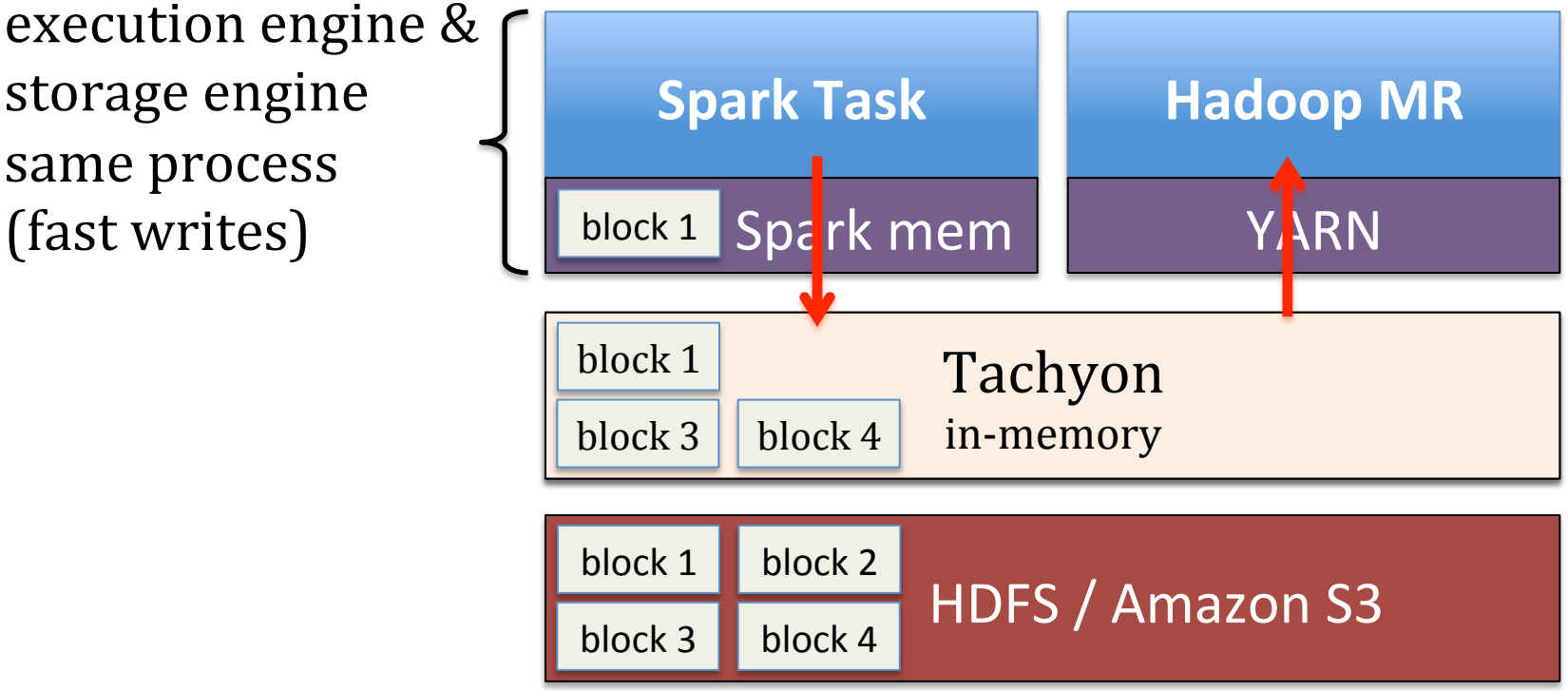

Computation Frameworks
(Spark, MapReduce, Impala, H2O, …)
Tachyon
Existing Storage Systems
(HDFS, S3, GlusterFS, …)

Memory-Centric Storage Architecture



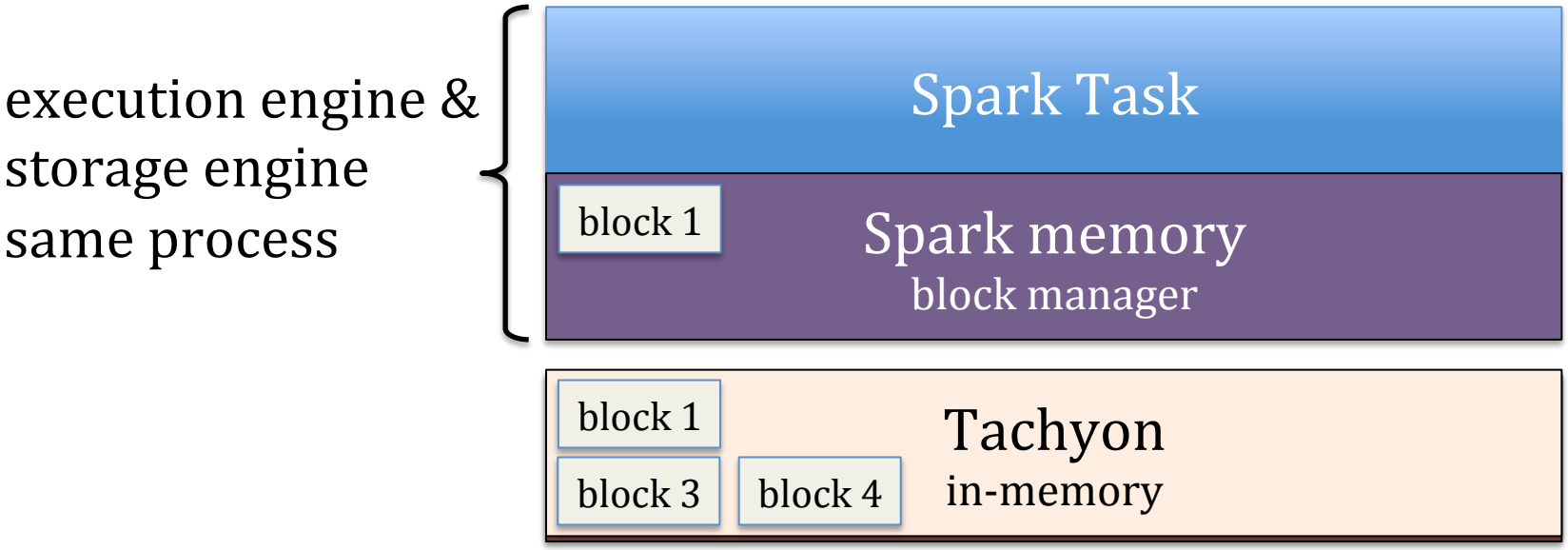
Issue 1 revisited

Memory-speed data sharing among jobs in different frameworks



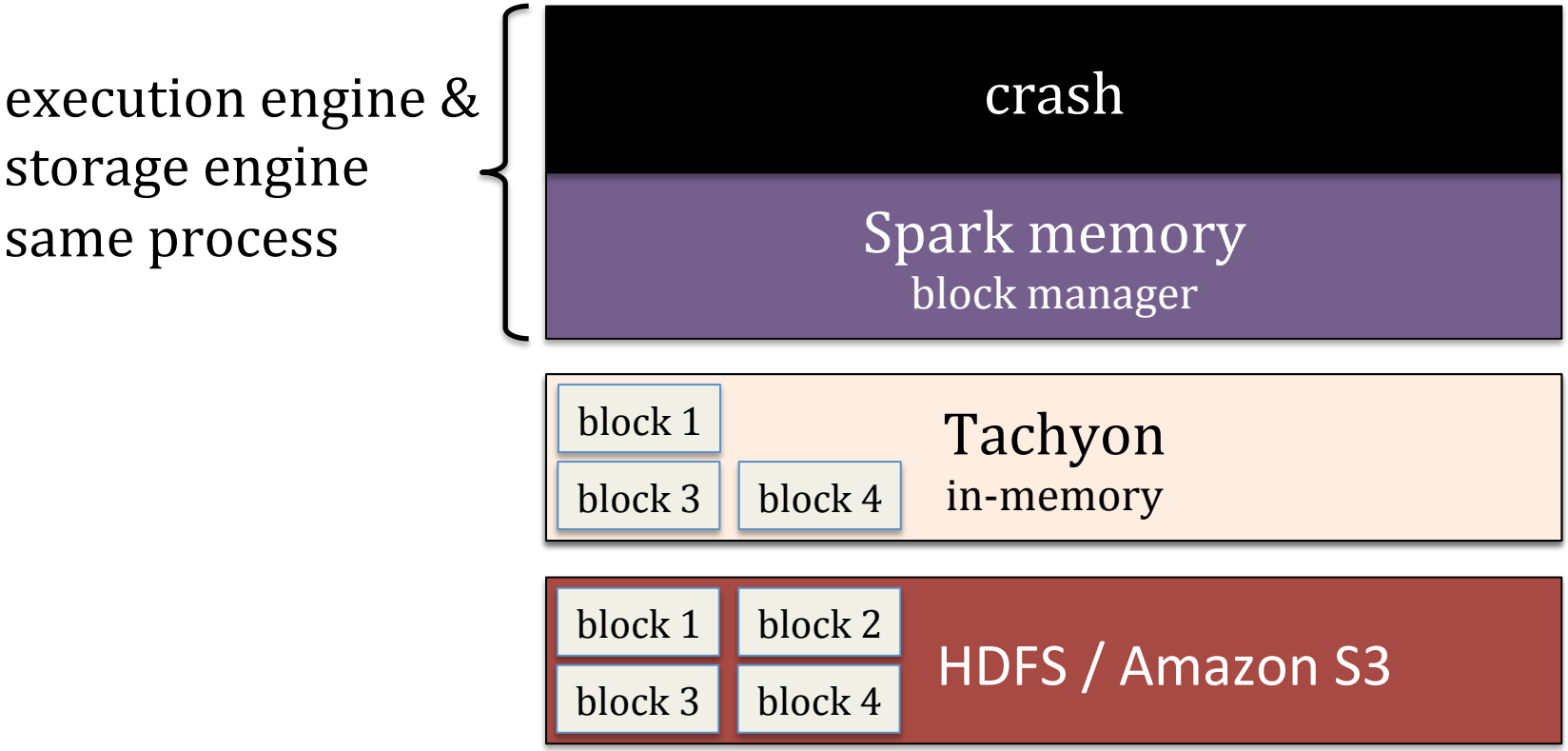
Issue 2 revisited

Keep in-memory data safe, even when a job crashes.



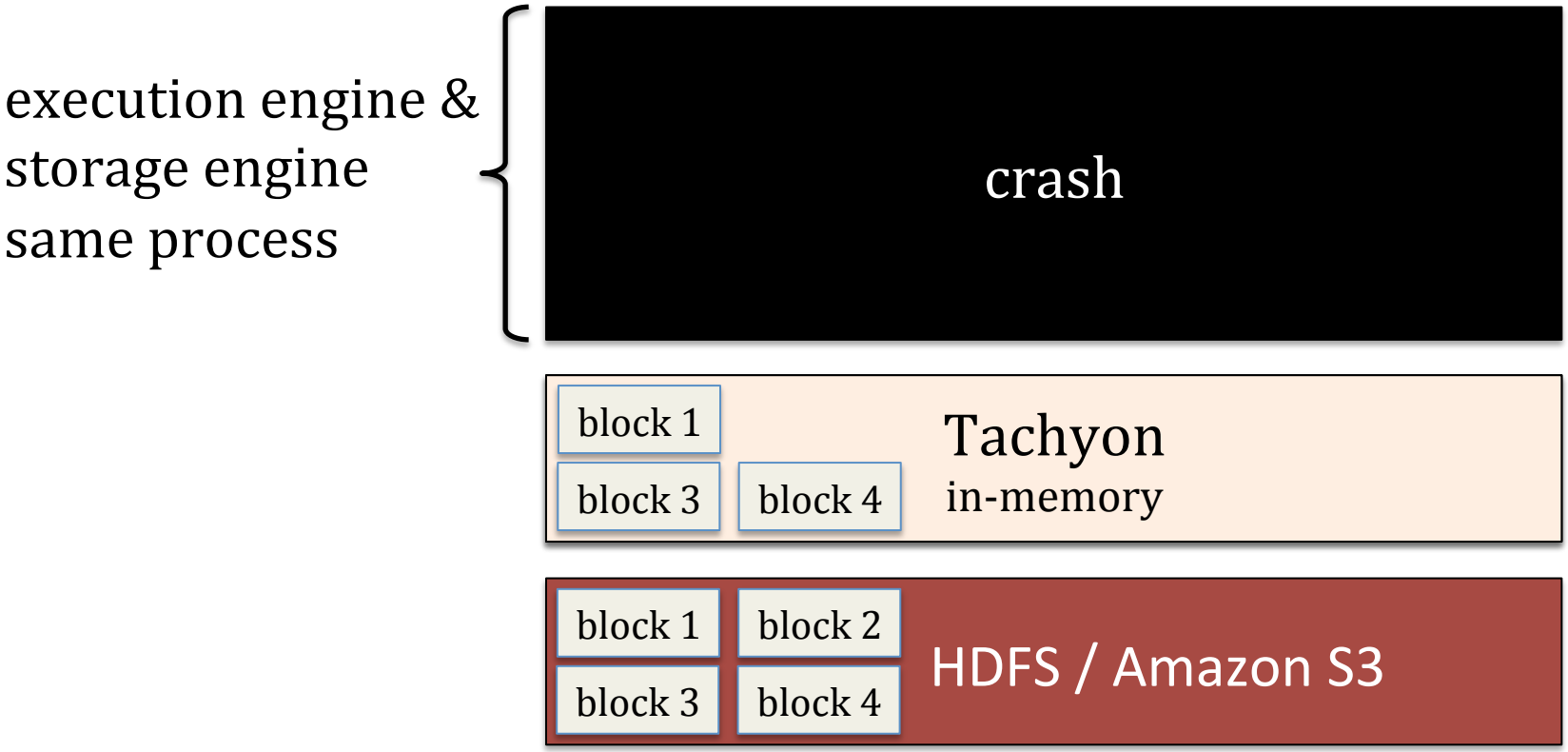
Issue 2 revisited

Keep in-memory data safe, even when a job crashes.



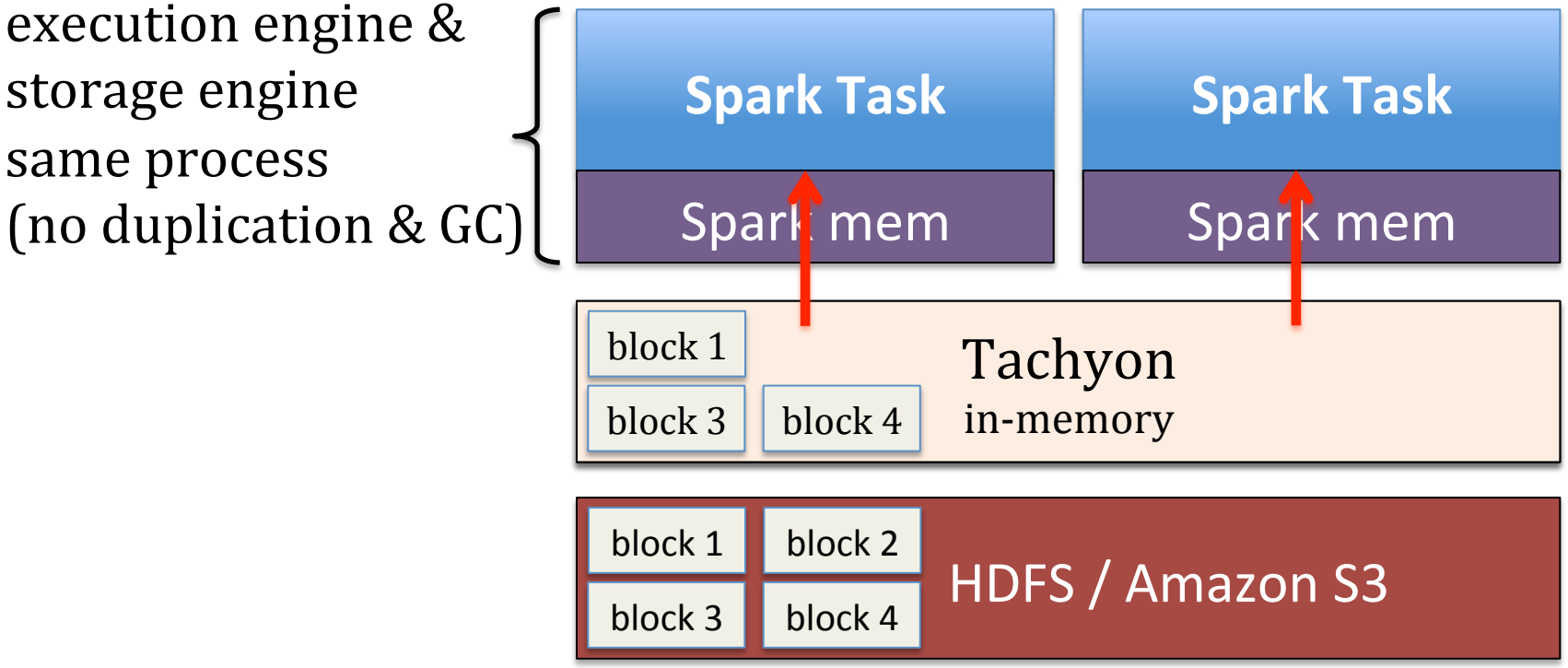
Issue 2 revisited

Keep in-memory data safe, even when a job crashes.

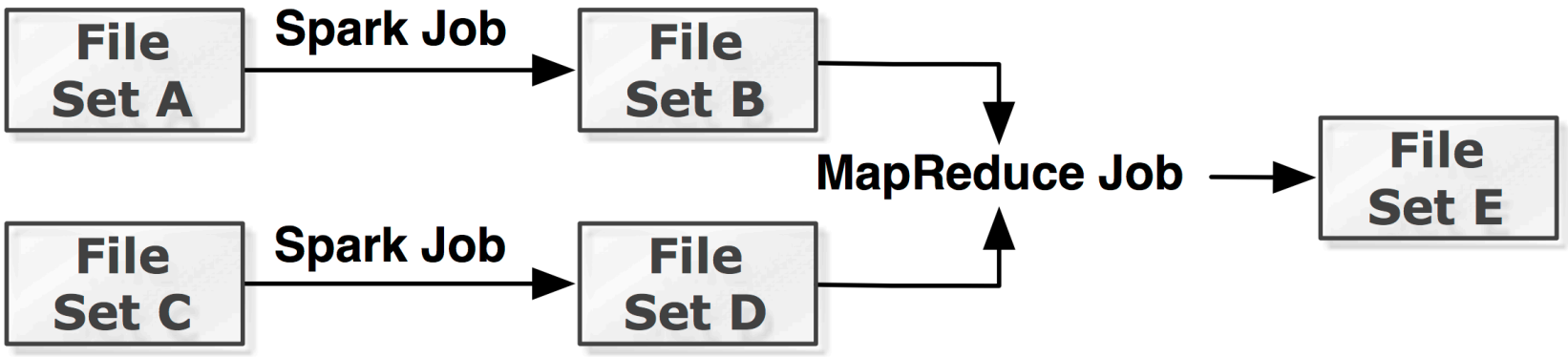


Issue 3 revisited

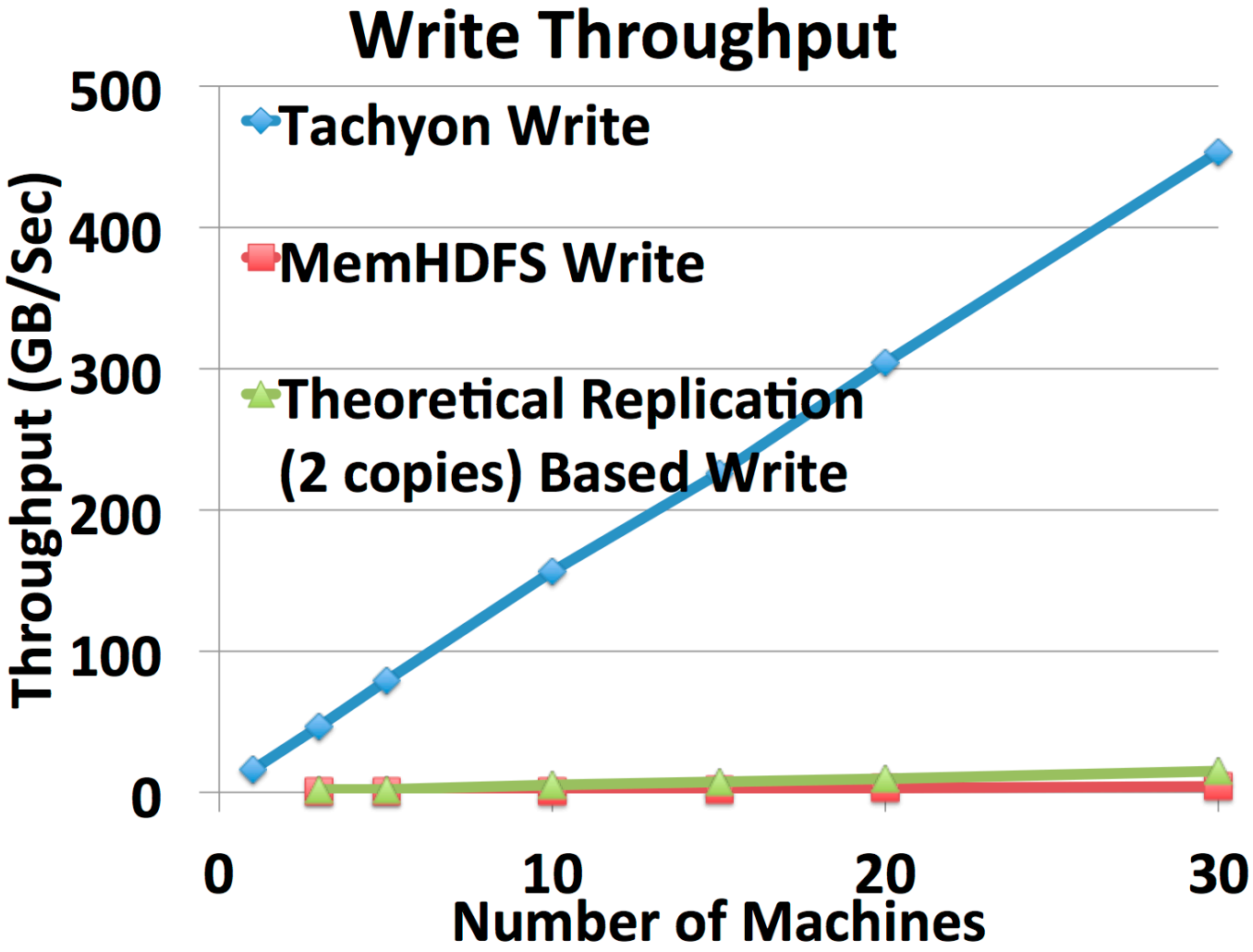
No in-memory data duplication, much less GC



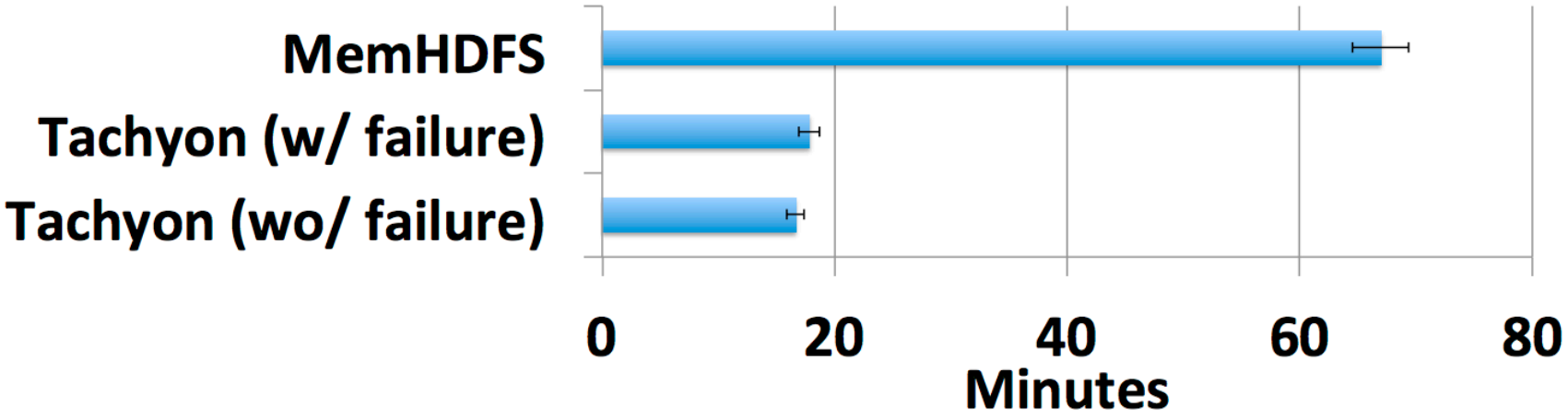
Lineage in Storage (*alpha*)



Comparison with in Memory HDFS



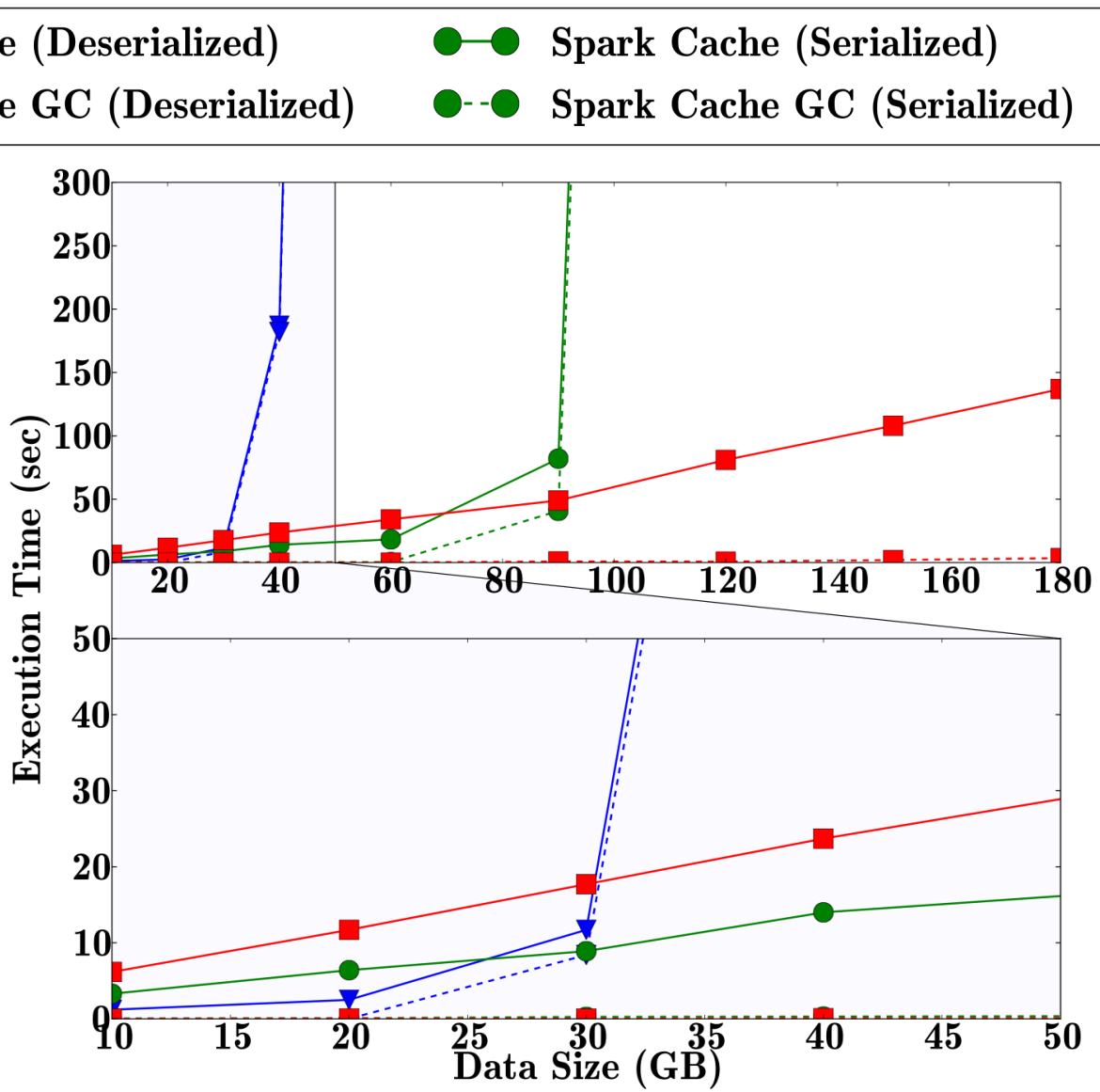
Workflow Improvement



Performance comparison for realistic workflow. The workflow ran 4x faster on Tachyon than on MemHDFS. In case of node failure, applications in Tachyon still finishes 3.8x faster.

Further Improve Spark's Performance

Grep Program



How easy / hard to use Tachyon?

Spark/MapReduce/Shark without Tachyon

- Spark
  - val file = sc.textFile(“hdfs://ip:port/path”)
- Hadoop MapReduce
  - hadoop jar hadoop-examples-1.0.4.jar wordcount hdfs://localhost:19998/input hdfs://localhost:19998/output
- Shark
  - CREATE TABLE orders_cached AS SELECT * FROM orders;

Spark/MapReduce/Shark with Tachyon

- Spark
  - val file = sc.textFile("tachyon://ip:port/path")
- Hadoop MapReduce
  - hadoop jar hadoop-examples-1.0.4.jar wordcount tachyon://localhost:19998/input tachyon://localhost:19998/output
- Shark
  - CREATE TABLE orders_tachyon AS SELECT * FROM orders;

Spark on Tachyon

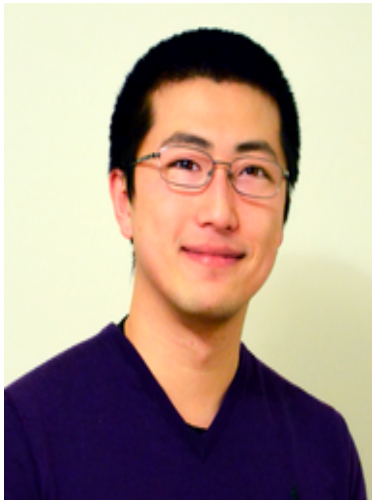
```
./bin/spark-shell
sc.hadoopConfiguration.set("fs.tachyon.impl", "tachyon.hadoop.TFS")

// Load input from Tachyon
val file = sc.textFile("tachyon://localhost:19998/LICENSE")
file.count() ; file.take(10);

// Store RDD OFF_HEAP in Tachyon
import org.apache.spark.storage.StorageLevel;
file.persist(StorageLevel.OFF_HEAP)
file.count(); file.take(10);

// Save output to Tachyon
file.flatMap(line => line.split(" ")).map(s => (s, 1)).reduceByKey((a, b) => a +
b).saveAsTextFile("tachyon://localhost:19998/LICENSE_WC")
```

Outline

- Overview
  - Feature 1: Memory Centric Storage Architecture
  - Feature 2: Lineage in Storage
- **Open Source**
- Roadmap

History

Started at UC Berkeley AMPLab from the summer of 2012





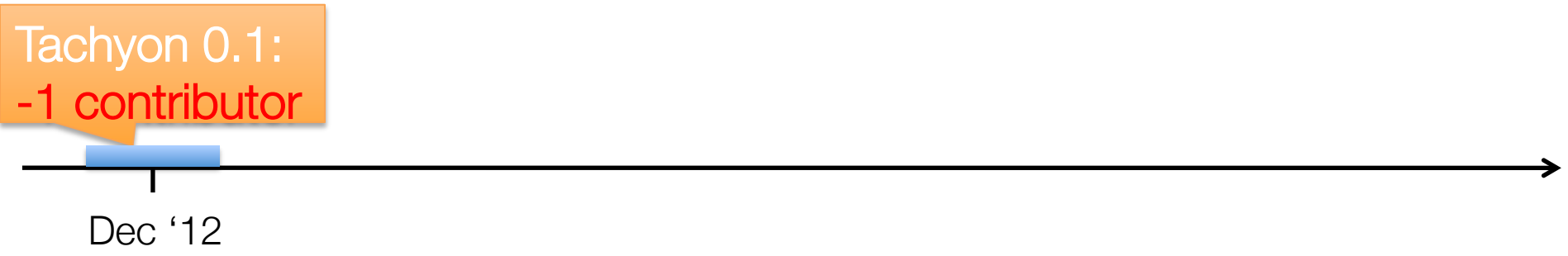
- Reliable, Memory Speed Storage for Cluster Computing Frameworks (UC Berkeley EECS Tech Report)
- Haoyuan Li, Ali Ghodsi, Matei Zaharia, Ion Stoica, Scott Shenker

TACHYON Open Source Status

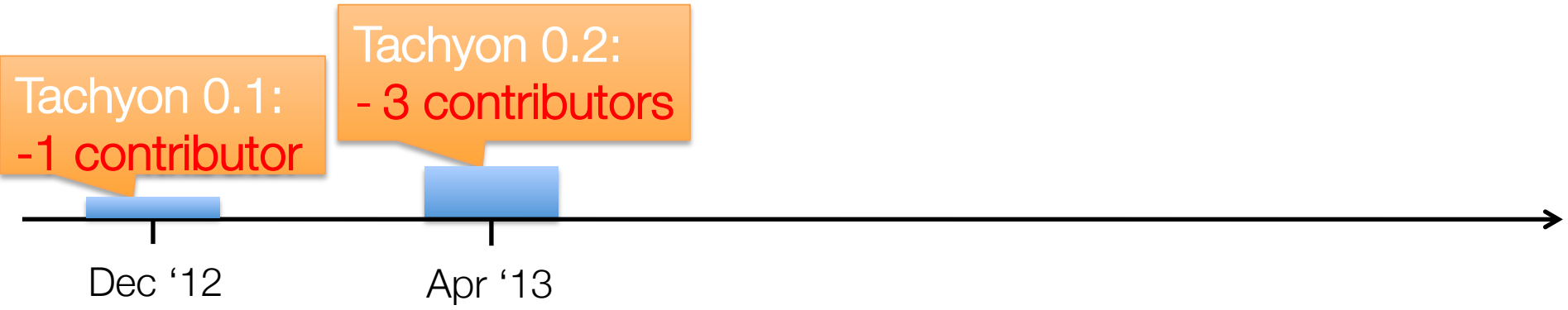
- Apache License 2.0, Version 0.5.0 (July 2014)

- Deployed at tens of companies
- 20+ Companies Contributing
- Spark and MapReduce applications can run without any code change

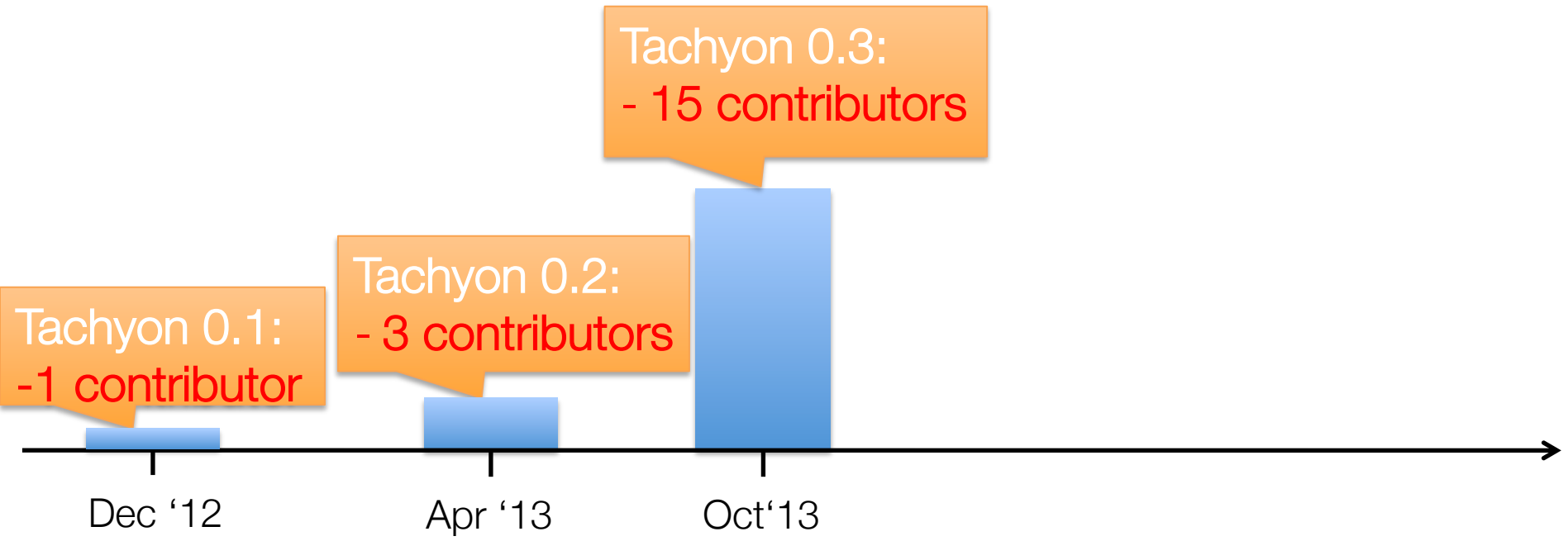
Release Growth



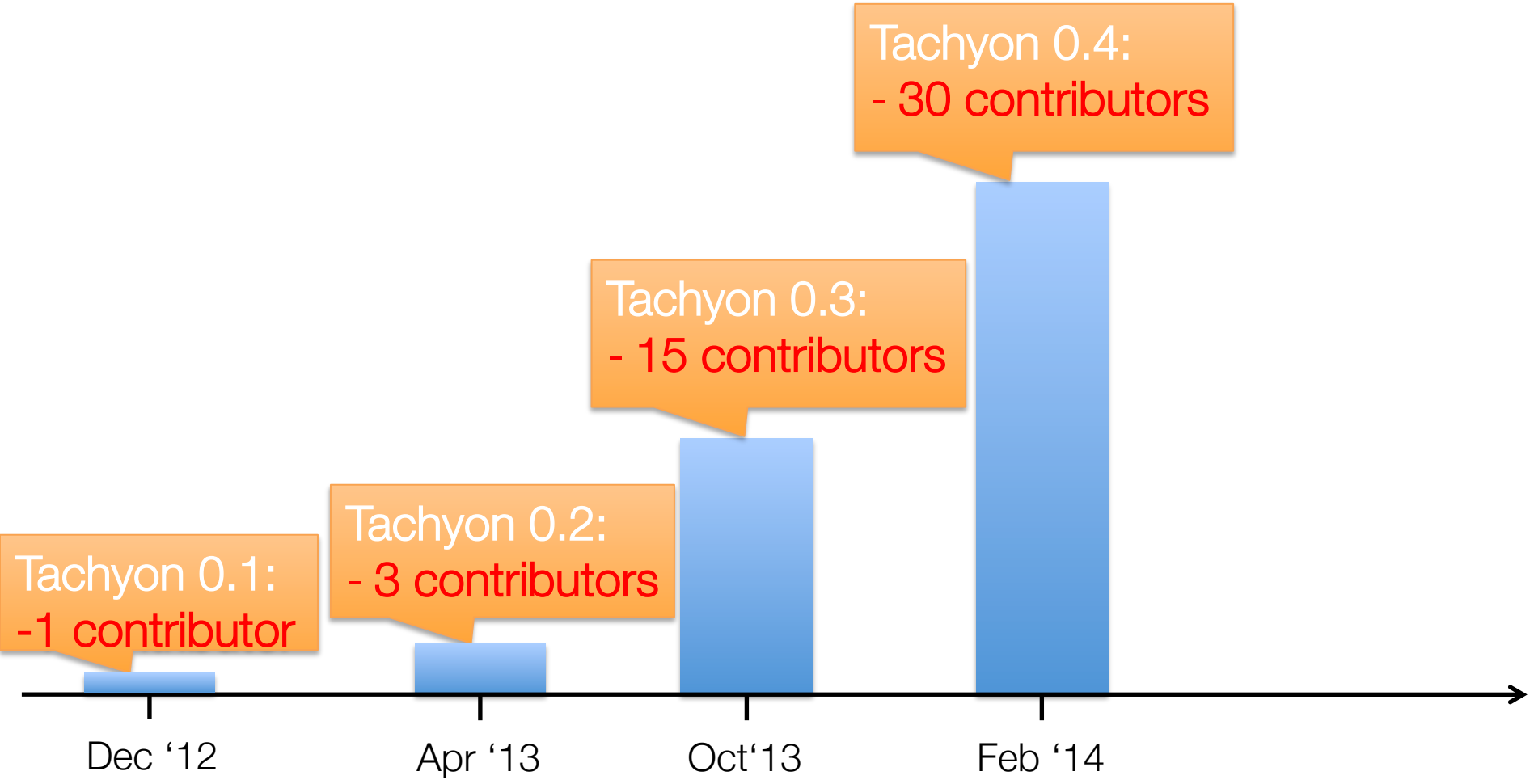
Release Growth



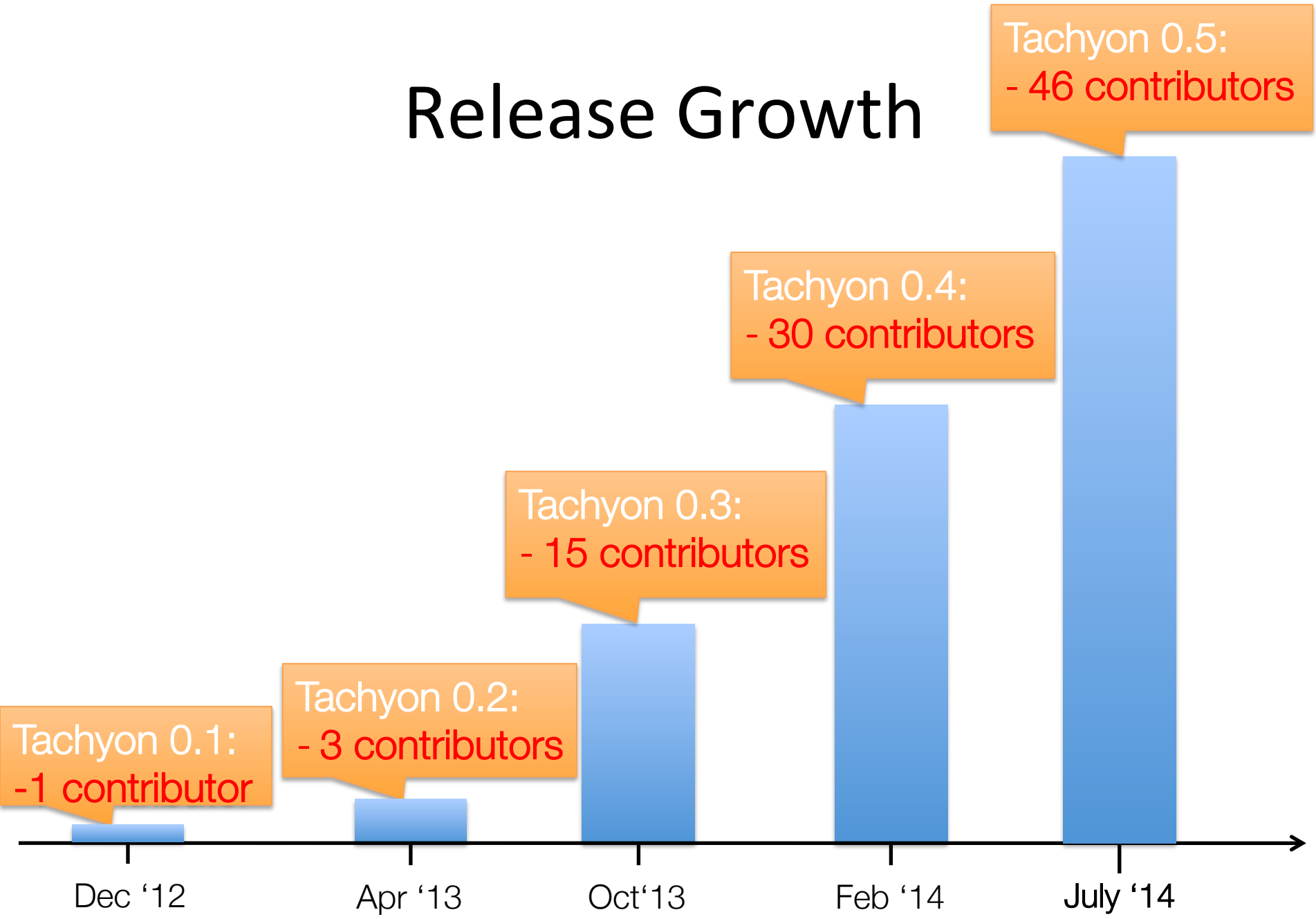
Release Growth



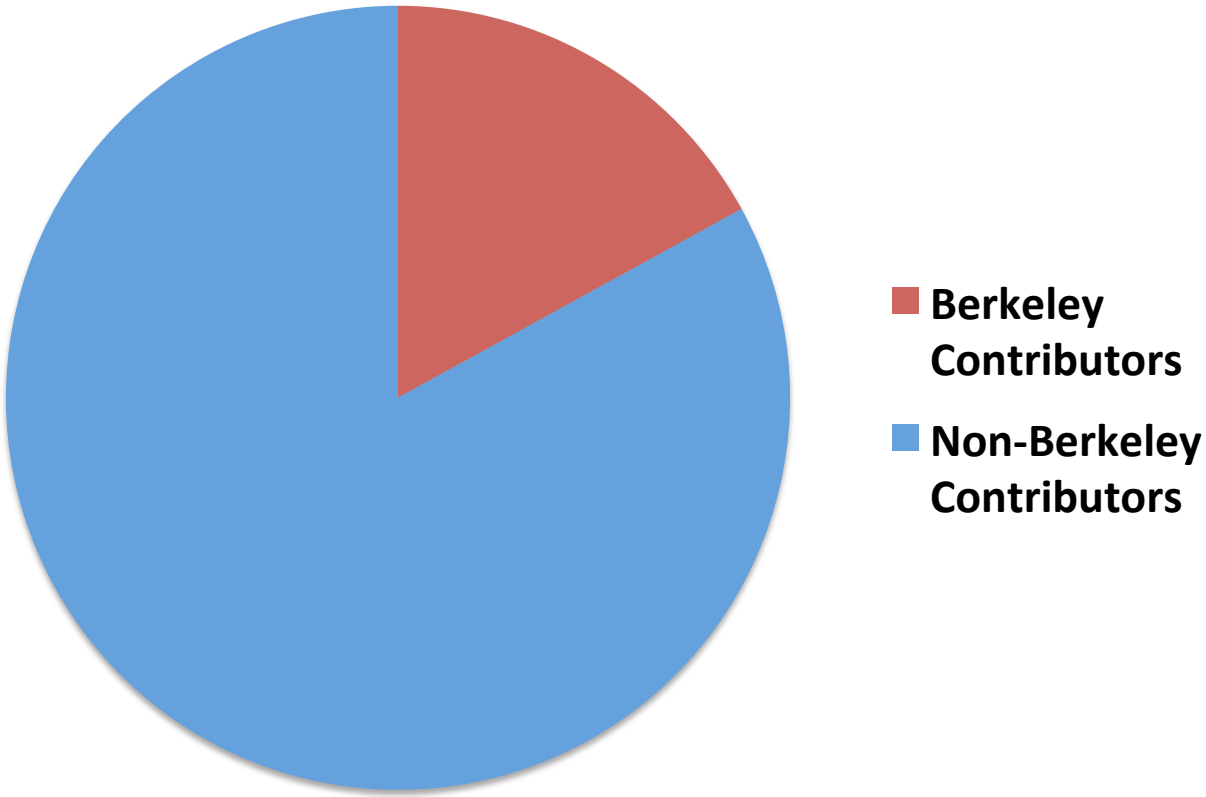
Release Growth



Release Growth



Open Community



Thanks to our Code Contributors!

Aaron Davidson
Achal Soni
Ali Ghodsi
Andrew Ash
Anurag Khandelwal
Aslan Bekirov
Bill Zhao
Brad Childs
Calvin Jia
Chao Chen
Cheng Chang
Cheng Hao
Colin Patrick McCabe
David Capwell
David Zhu
Du Li
Fei Wang
Gerald Zhang
Grace Huang
Haoyuan Li
Henry Saputra
Hobin Yoon
Huamin Chen
Jey Kottalam
Joseph Tang
Juan Zhou
Jun Aoki
Lin Xing
Lukasz Jastrzebski
Manu Goyal
Mark Hamstra
Mingfei Shi
Mubarak Seyed
Nick Lanham
Orcun Simsek
Pengfei Xuan
Qianhao Dong
Qifan Pu
Raymond Liu
Reynold Xin
Robert Metzger
Rong Gu
Sean Zhong
Seonghwan Moon
Shivaram Venkataraman
Srinivas Parayya
Tao Wang
Timothy St. Clair
Thu Kyaw
Vamsi Chitters
Xi Liu
Xiang Zhong
Xiaomin Zhang
Zhao Zhang



Tachyon
is in Fedora 20

Thanks to Redhat!

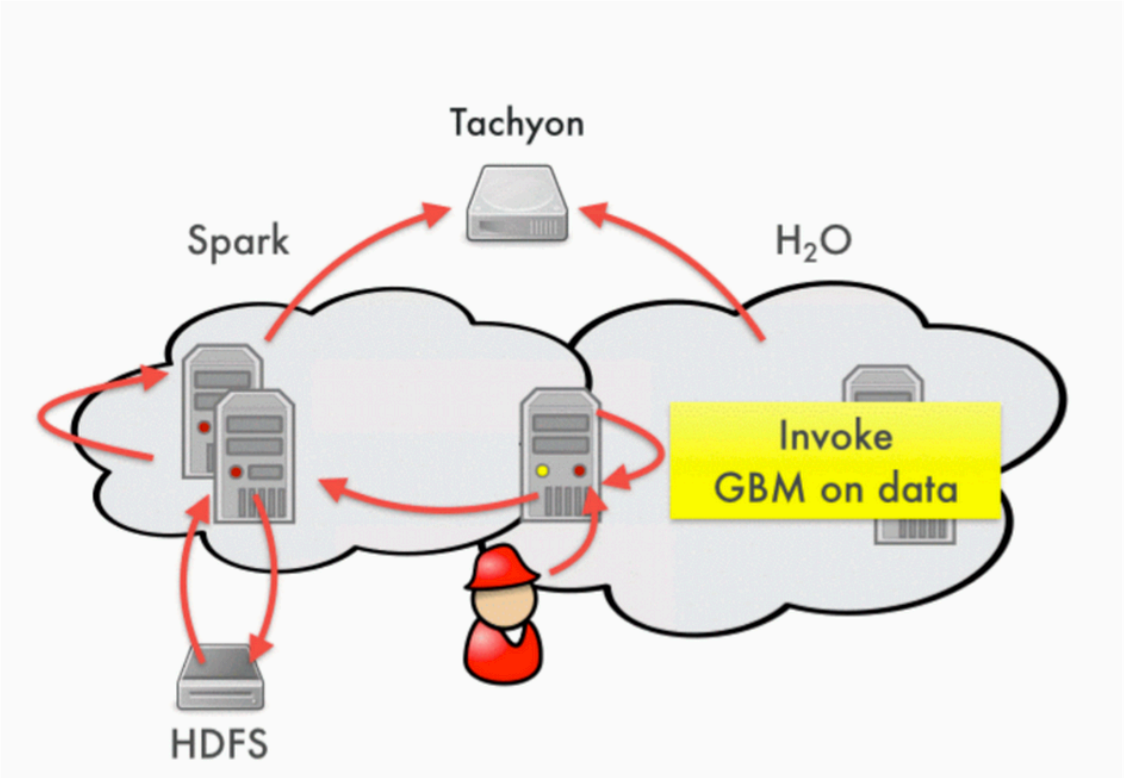
Commercially supported by 

and running in dozens of their customers' clusters

Tachyon is the Default Off-Heap Storage Solution for Spark

Exchange Data Between Spark and H20

Today, data gets parsed and exchanged between Spark and H2O via Tachyon. Users can interactively query big data both via SQL and ML from within the same context.



Believe from Industry

The Future Architecture of a Data Lake: In-memory Data Exchange Platform Using Tachyon and Apache Spark

OCTOBER 14, 2014 | **NEWS** | BY PAUL M. DAVIS

GIGAOM

Pivotal and EMC are betting on Spark cousin Tachyon as in-memory file system

by Derrick Harris OCT. 14, 2014 - 11:47 AM PDT

ZDNet

Pivotal bets on Tachyon as next in-memory file system

database
TRENDS AND APPLICATIONS

Pivotal Expands on Data Lake Vision with Embrace of Project Tachyon

Oct 14, 2014

Reaching wider communities: e.g. GlusterFS

blog.gluster.org/2014/08/glusterfs-and-tachyon/

GLUSTER COMMUNITY

DOCUMENTATION CONTACT ABOUT

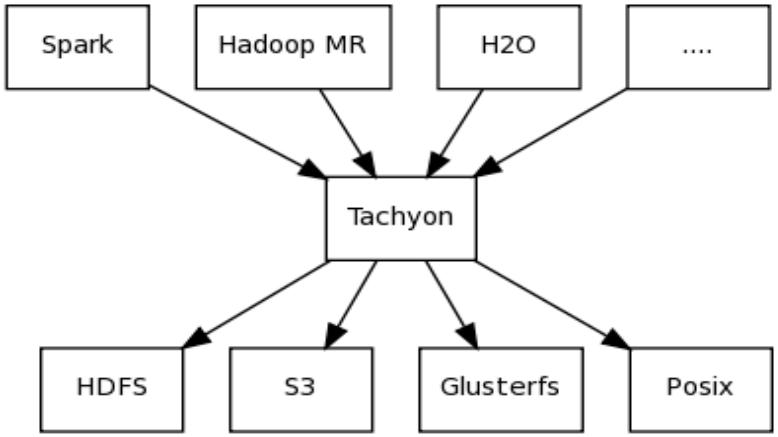
Like 1.1k g+1 Tweet

by huamin on August 7, 2014

← Return to Blog Home

Glusterfs and Tachyon

Tachyon, an in-memory distributed filesystem, is among the most dynamic projects in big data analytics stack. It provides java io like API, support Apache Spark, and vastly improves Spark's performance under large data set.



Under Filesystem Choices (Big Data, Cloud, HPC, Enterprise)

Under Filesystem Choices (Big Data, Cloud, HPC, Enterprise)

Outline

- Overview
  - Feature 1: Memory Centric Storage Architecture
  - Feature 2: Lineage in Storage
- Open Source
- **Roadmap**

Features

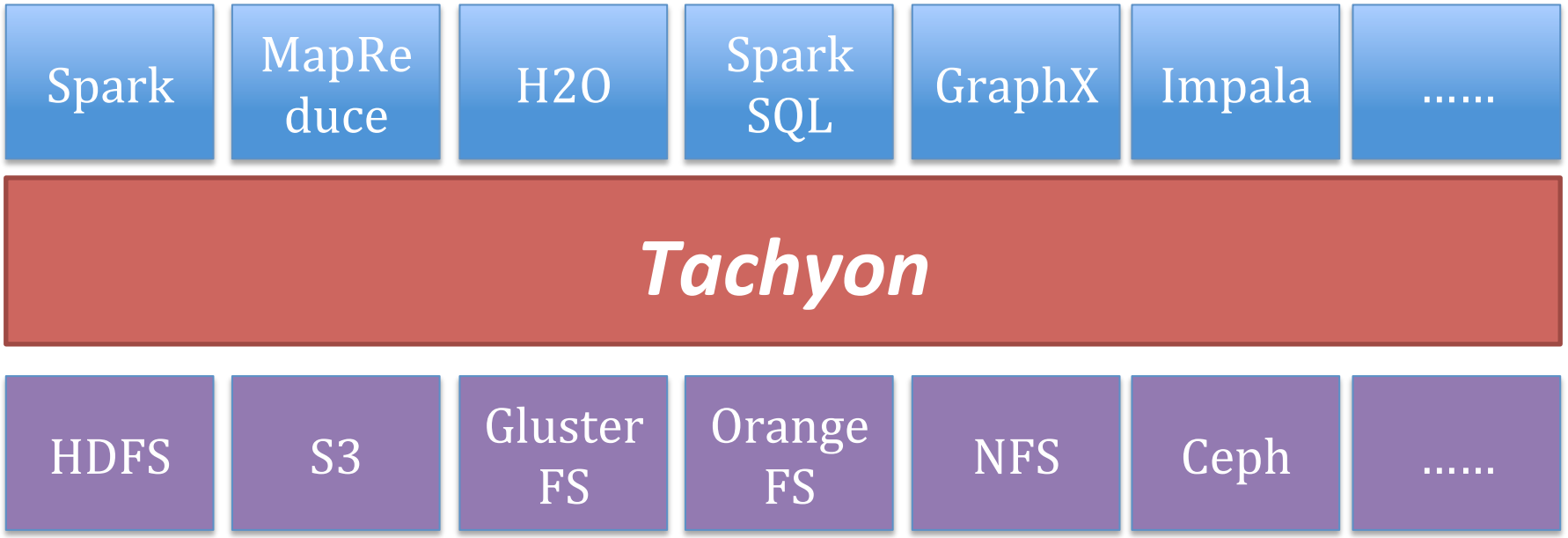
- Memory Centric Storage Architecture
- Lineage in Storage (alpha)
- Hierarchical Local Storage
- Data Serving
- Different hardware
- More…
- Your Requirements?

Short Term Roadmap (0.6 Release)

- Ceph Integration (Ceph Community, Redhat)
- Hierarchical Local Storage (Intel)
- Performance Improvement (Yahoo)
- Multi-tenancy (AMPLab)
- Mesos Integration (Mesos Community, Mesosphere)
- Network Sub-system Improvement (Pivotal)
- ***Many more*** from AMPLab and Industry Contributors

Goal?

Better Assist Other Components



Welcome Collaboration!

Thanks!

Questions?

- *More Information:*
  - *Website: http://tachyon-project.org*
  - *Github: https://github.com/amplab/tachyon*
  - *Meetup: http://www.meetup.com/Tachyon*
- *Email: haoyuan@cs.berkeley.edu*

Release Growth