A Reliable Memory-Centric Distributed Storage System

Haoyuan Li, Tachyon Nexus
haoyuan@tachyonnexus.com
September 22, 2015 @ SDC 2015

TACHYON
NEXUS

- Team consists of Tachyon creators, top contributors, people from UC Berkeley, Google, CMU, VMware, Stanford, Facebook, etc.

- $7.5 million Series A from Andreessen Horowitz

- Committed to Tachyon Open Source

TACHYON NEXUS

WE'RE HIRING!

Outline

- Overview
  - Motivation
  - Tachyon Architecture
  - Using Tachyon
- Open Source
  - Status
  - Production Use Cases
- Roadmap

Outline

- **Overview**
  - Motivation
  - Tachyon Architecture
  - Using Tachyon
- Open Source
  - Status
  - Production Use Cases
- Roadmap

TACHYON NEXUS

Tachyon: Born in UC Berkeley AMPLab

Apache MESOS

Cluster manager

Spark

Parallel computation framework

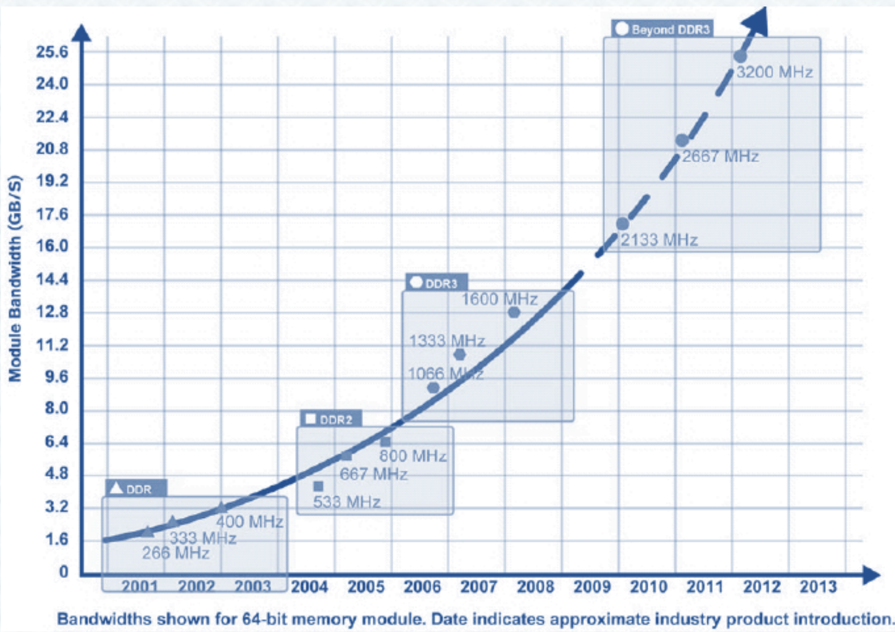
TACHYON

Reliable, distributed memory-centric storage system

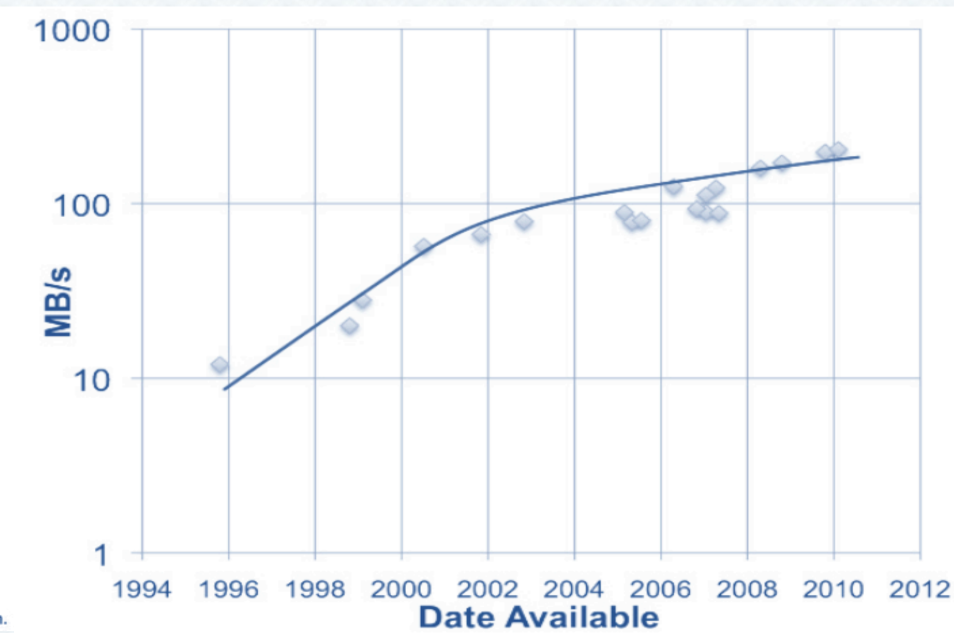
Why Tachyon?

Memory is **Fast**

- RAM throughput increasing **exponentially**

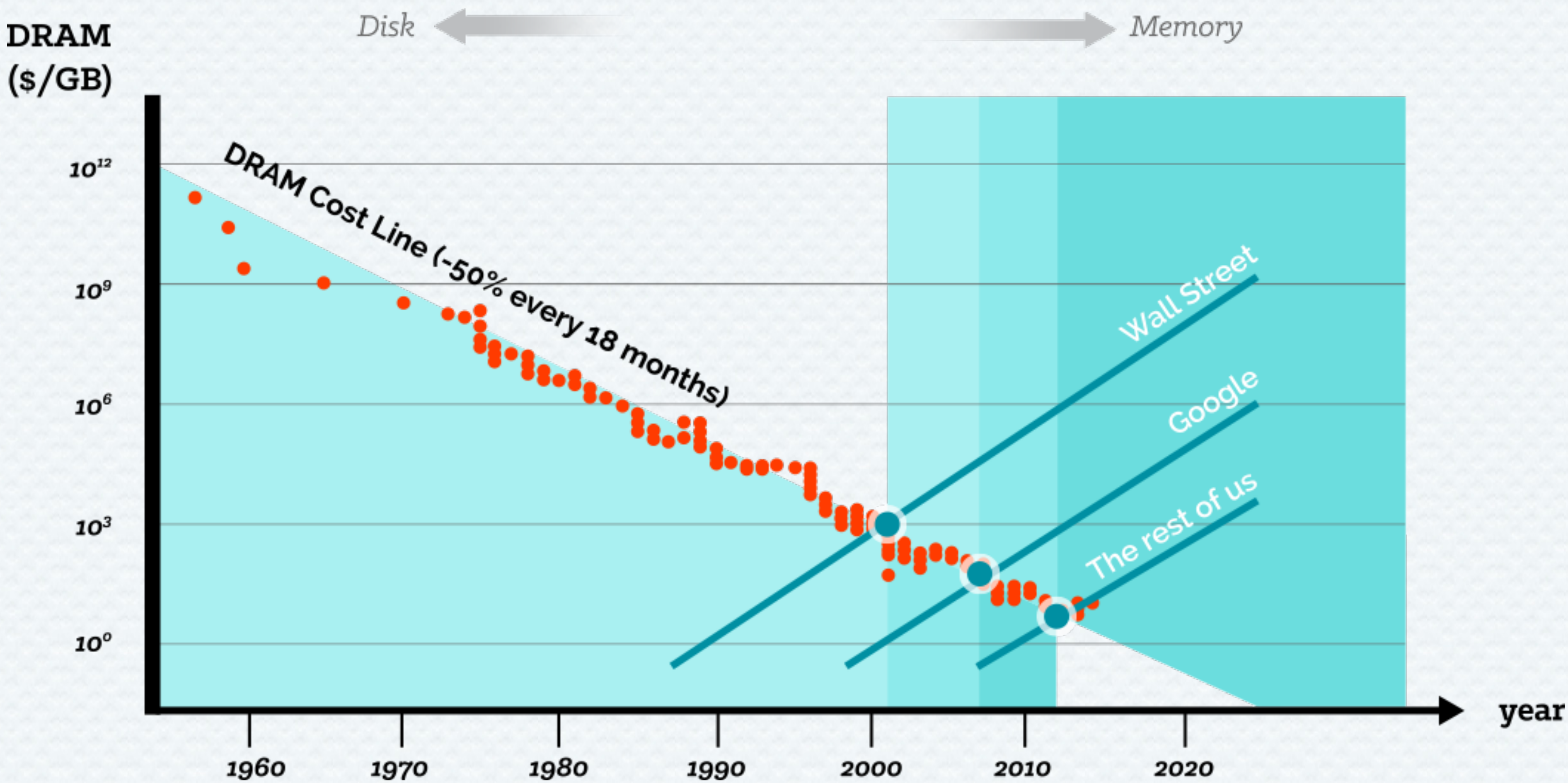


- Disk throughput increasing **slowly**



Memory-locality key to interactive response times

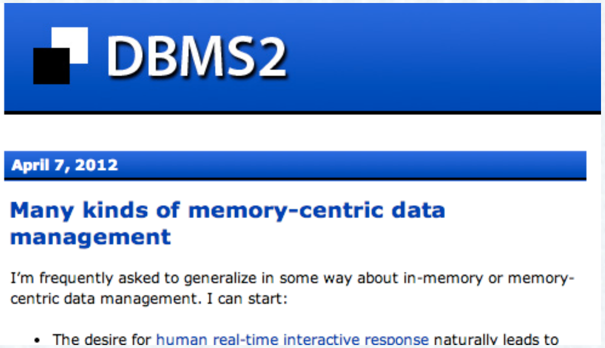
Memory is Cheaper



source: jcmit.com

TACHYON NEXUS

Realized by many…


DBMS2

April 7, 2012

Many kinds of memory-centric data management

I'm frequently asked to generalize in some way about in-memory or memory-centric data management. I can start:

- The desire for human real-time interactive response naturally leads to

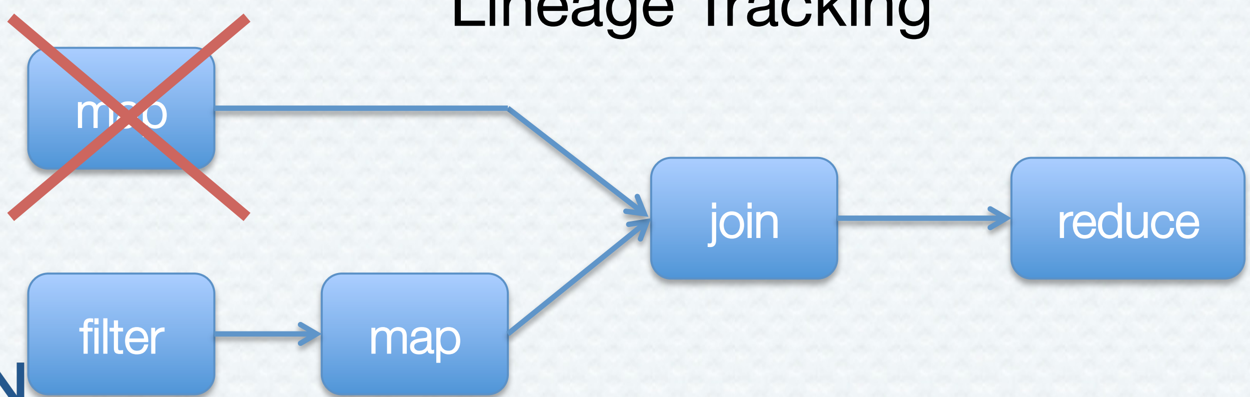
Is the Problem Solved?

Missing a Solution for the 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

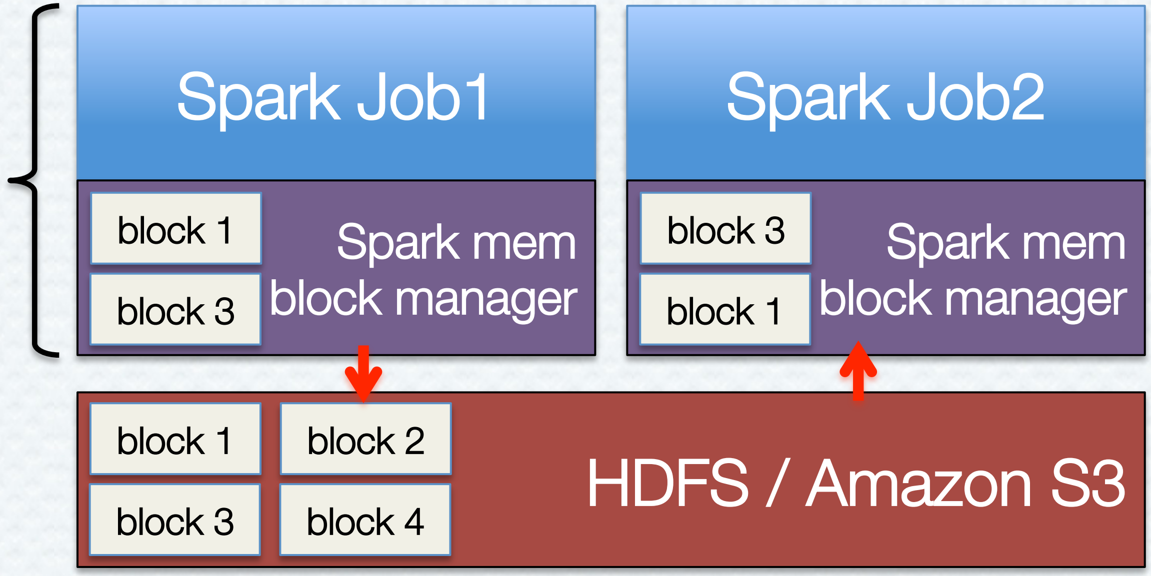
Lineage Tracking



Issue 1

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

storage engine &
execution engine
same process
(slow writes)

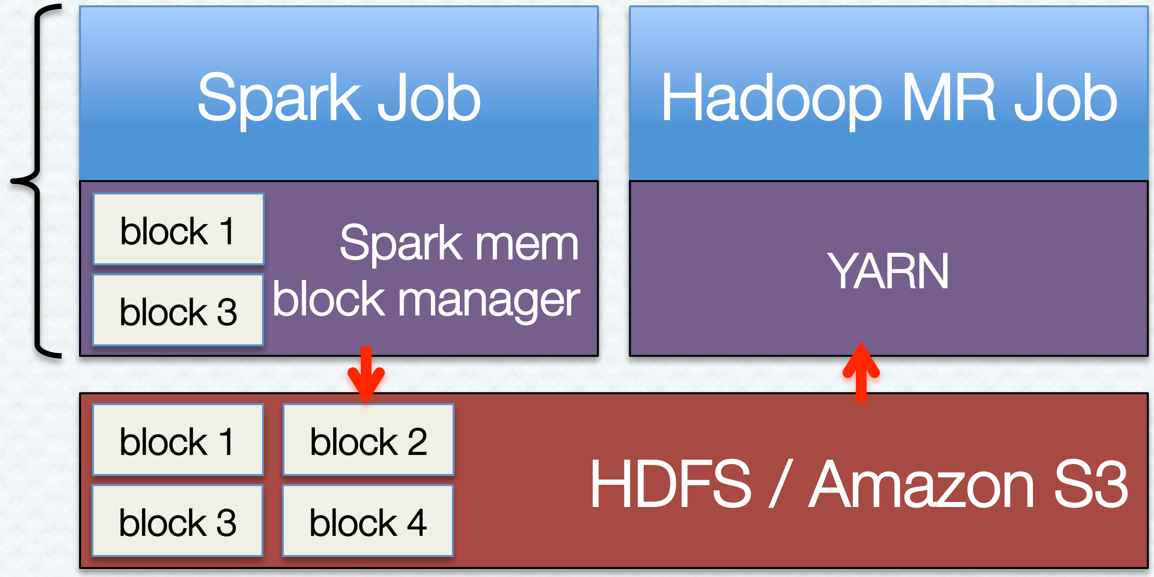


TACHYON
NEXUS

Issue 1

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

storage engine & execution engine same process (slow writes)

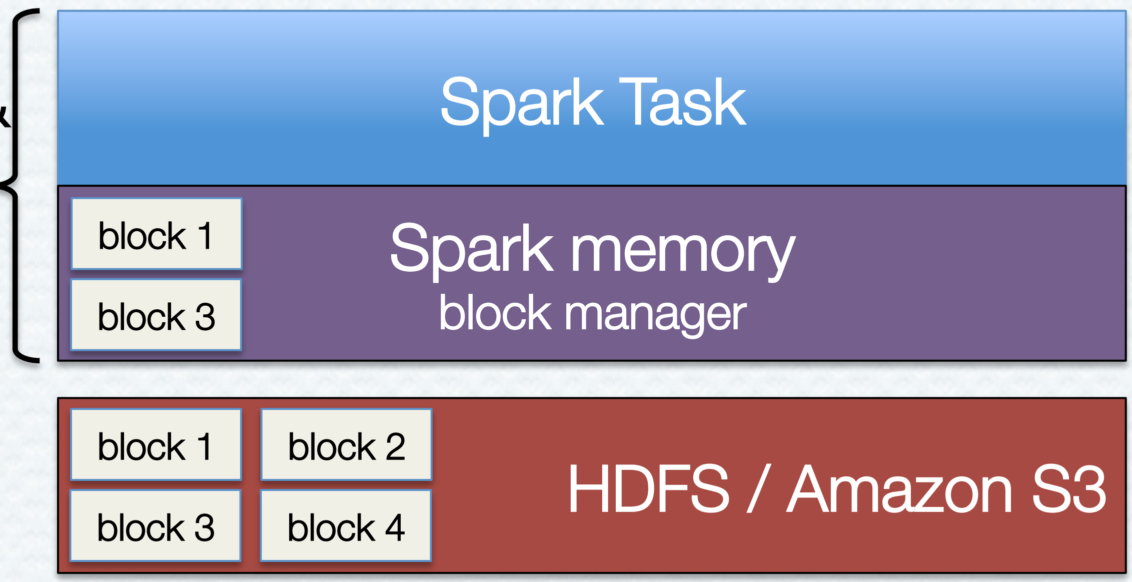


TACHYON NEXUS

Issue 2

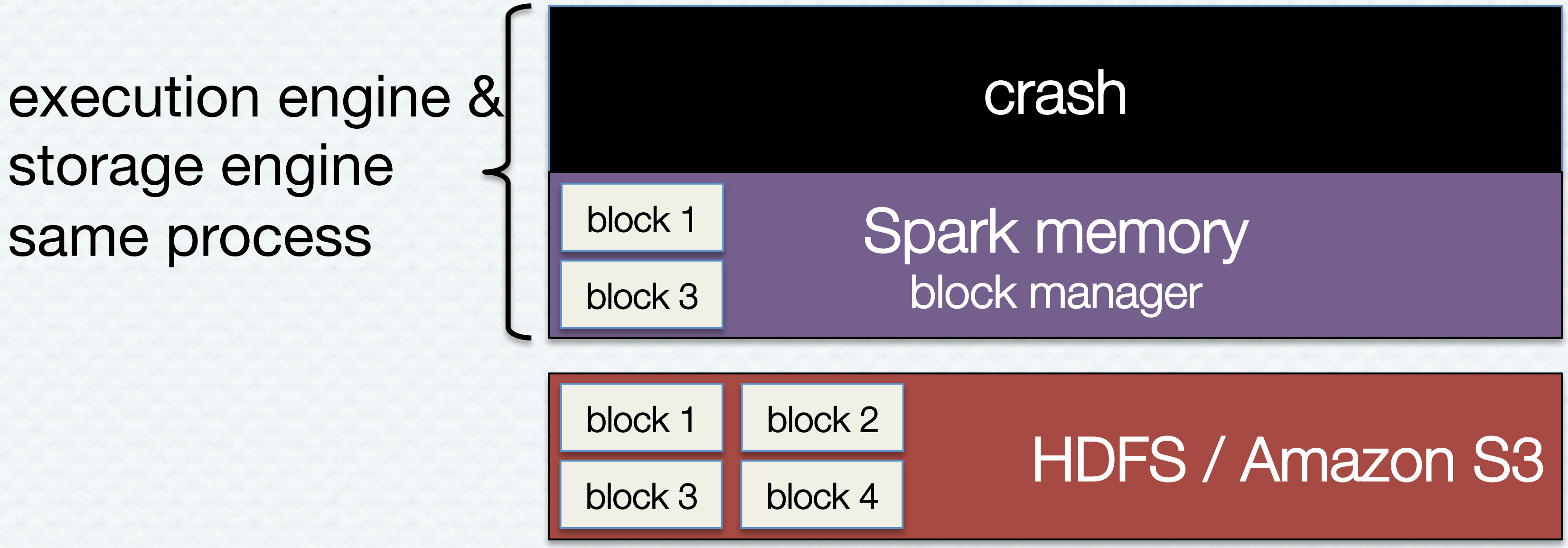
Cache loss when process crashes

execution engine &
storage engine
same process



Issue 2

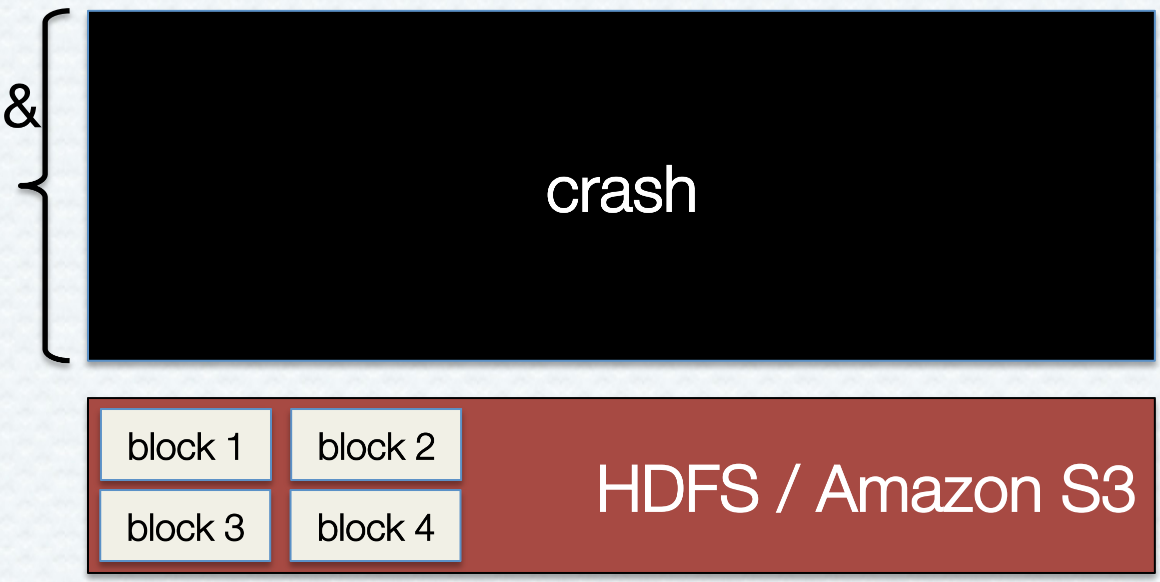
Cache loss when process crashes



Issue 2

Cache loss when process crashes

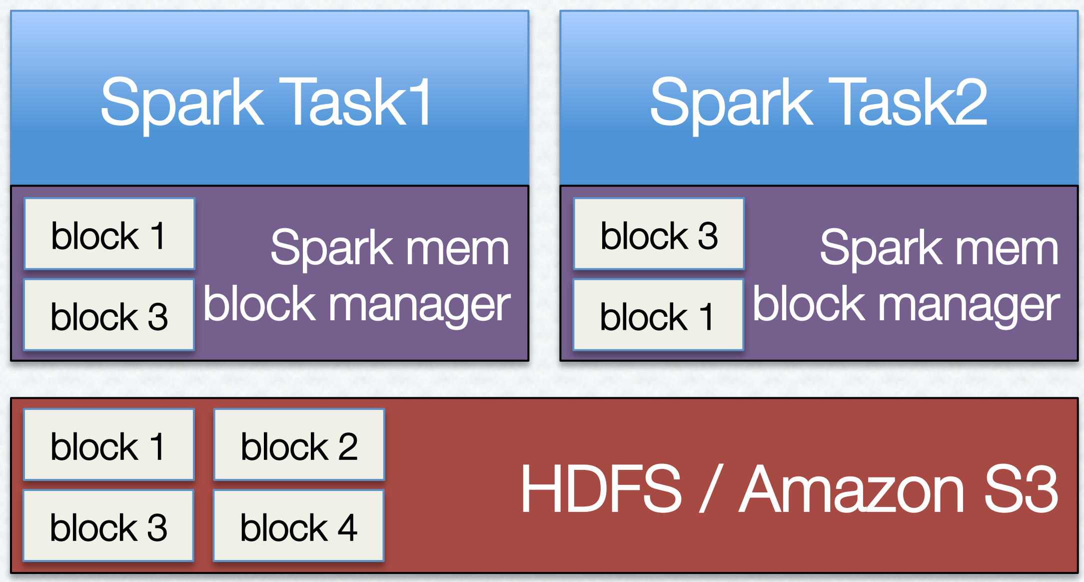
execution engine &
storage engine
same process



Issue 3

In-memory Data Duplication & Java Garbage Collection

execution engine & storage engine same process (duplication & GC)



Tachyon

Reliable data sharing at

memory-speed within and across

cluster frameworks/jobs

Technical Overview

Ideas

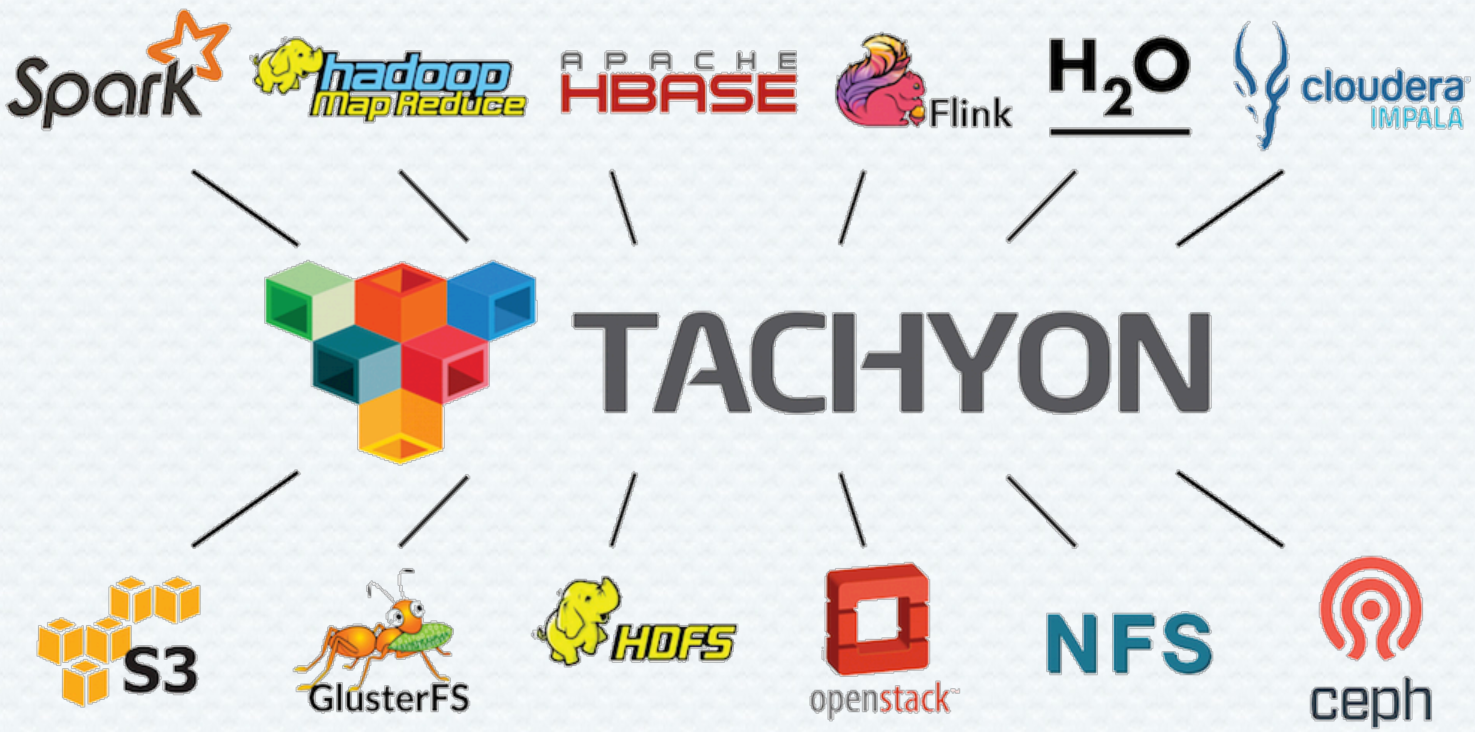
- A **memory-centric** storage architecture
- Push **lineage** down to storage layer
- Manage **tiered** storage

Facts

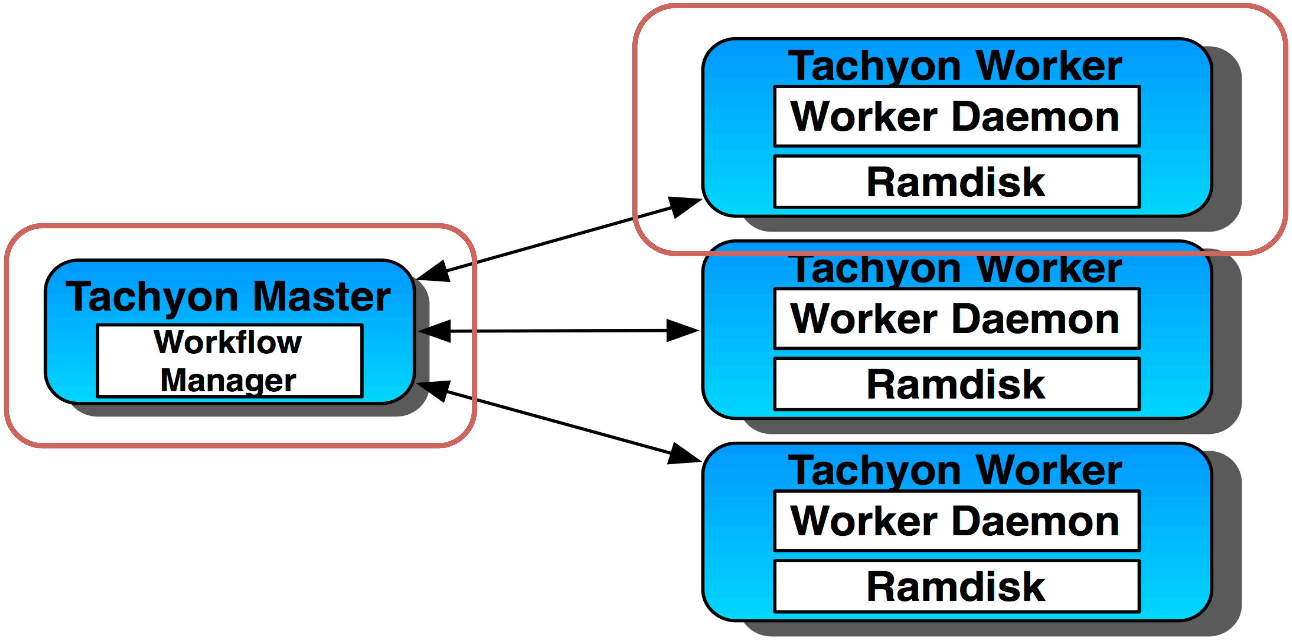
- One data copy in memory
- Re-computation for fault-tolerance

TACHYON
NEXUS

Eco-System

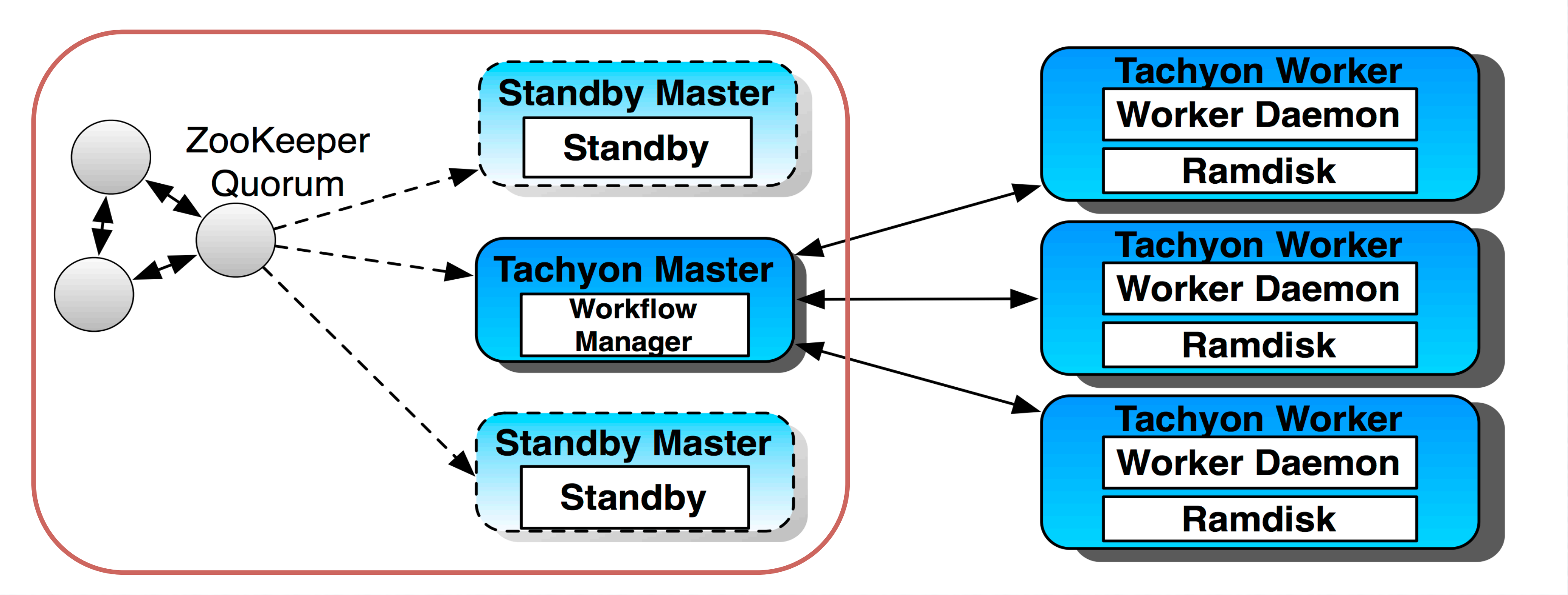

Spark
hadoop Map Reduce
APACHE HBASE
Flink
H2O
cloudera IMPALA
TACHYON
S3
GlusterFS
HDFS
openstack
NFS
ceph

Tachyon Memory-Centric Architecture



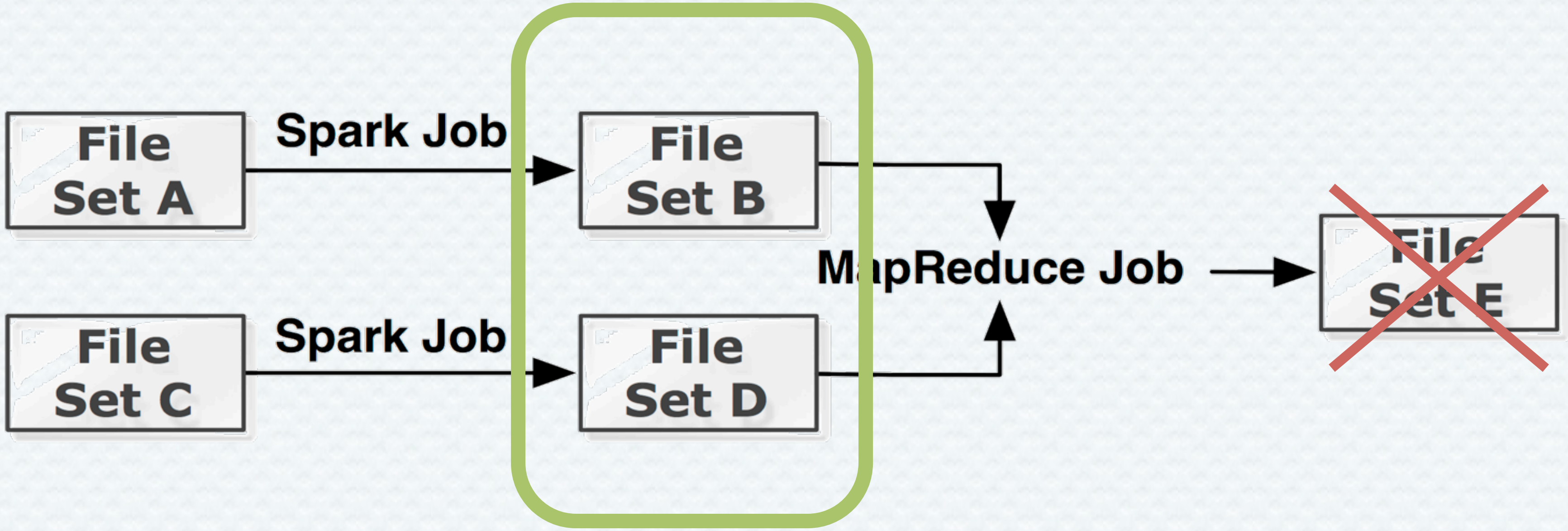
TACHYON
N E X U S

Tachyon Memory-Centric Architecture



TACHYON
N E X U S

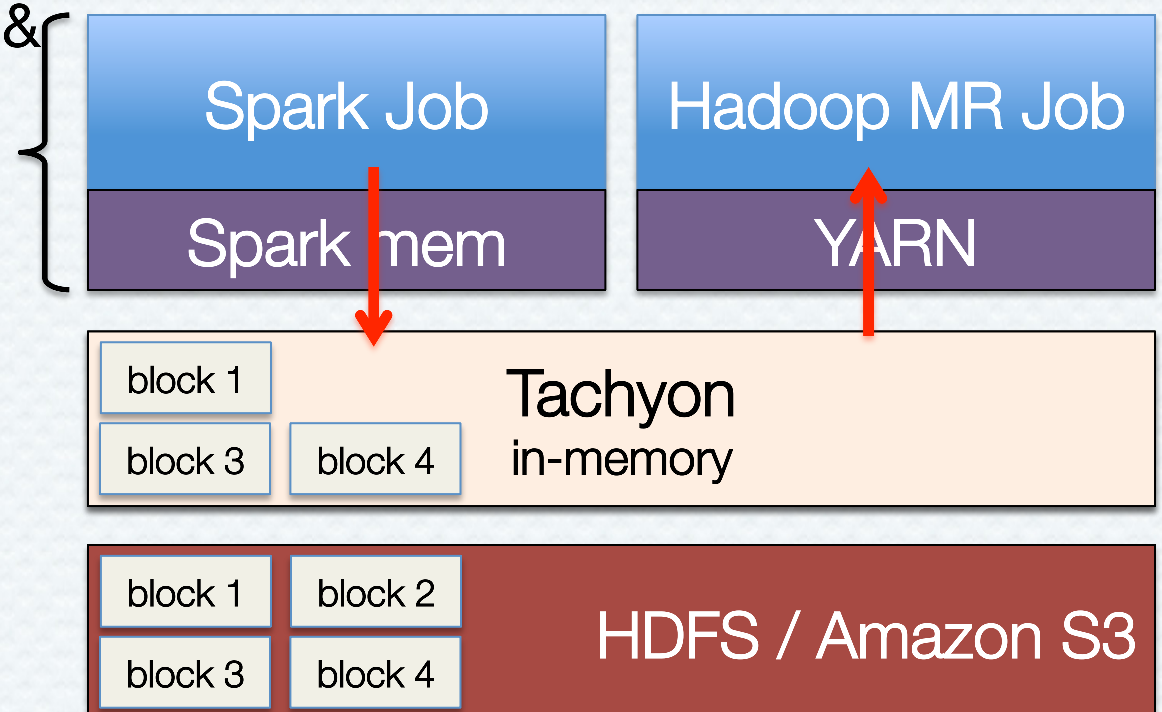
Lineage in Tachyon



Issue 1 revisited

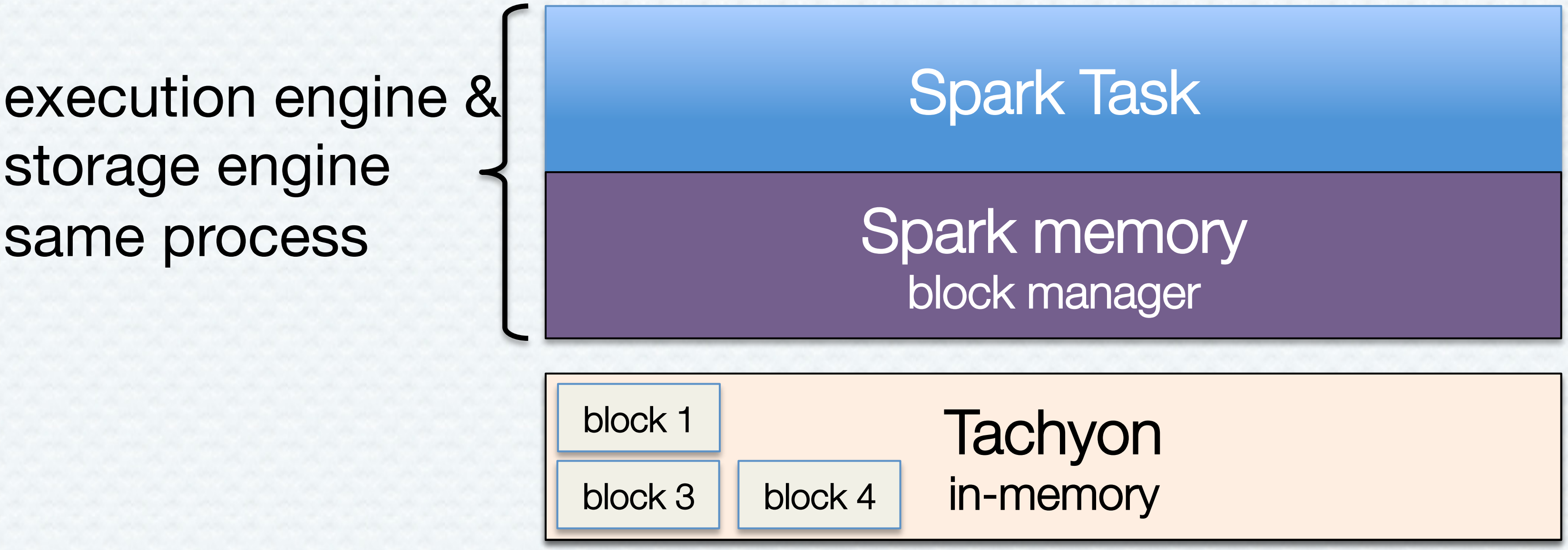
Memory-speed data sharing among jobs in different frameworks

execution engine & storage engine same process (fast writes)



Issue 2 revisited

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

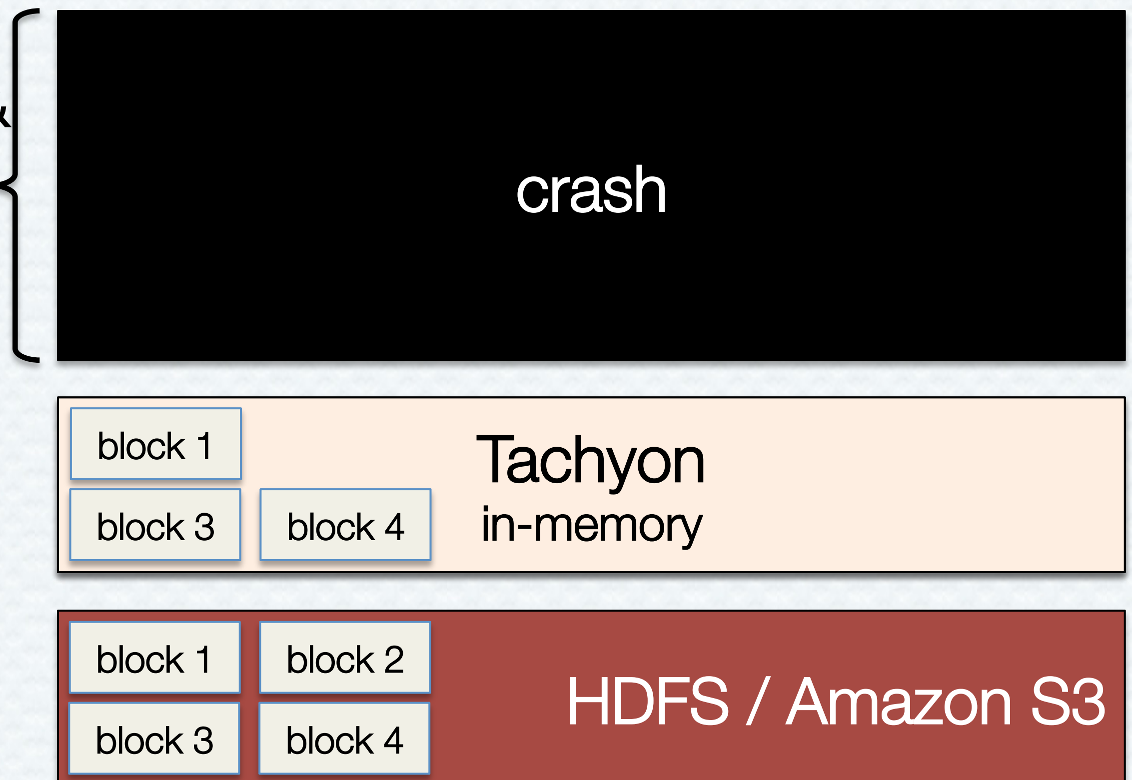


TACHYON
N E X U S

Issue 2 revisited

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

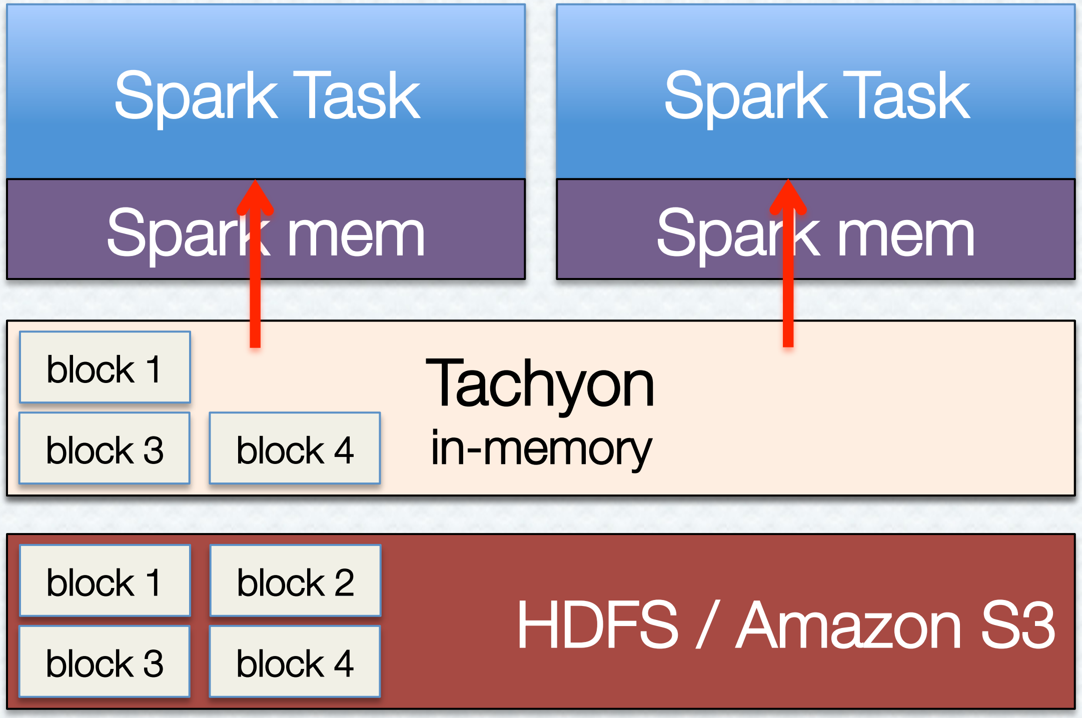
execution engine &
storage engine
same process



Issue 3 revisited

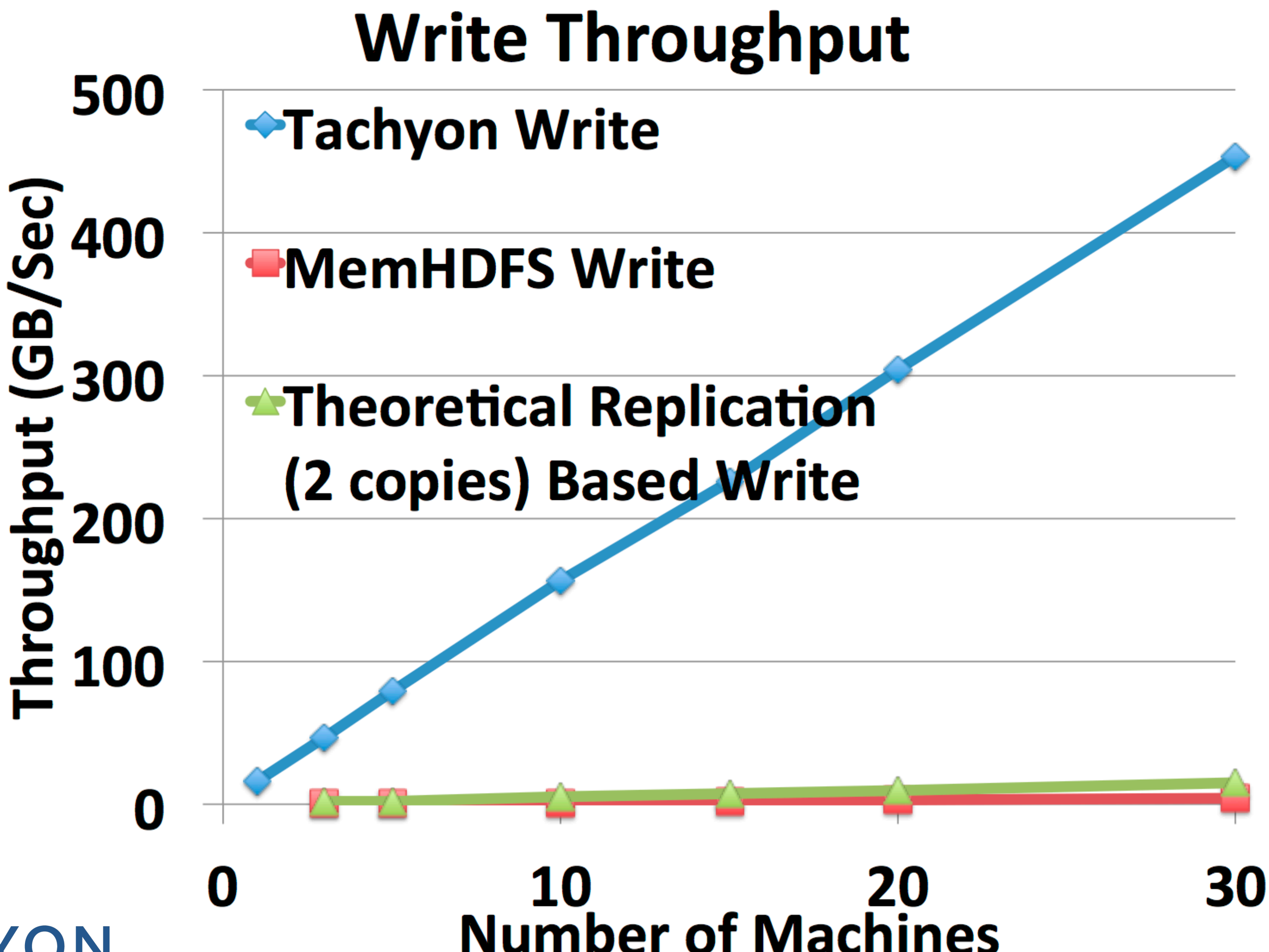
No in-memory data duplication, much less GC

execution engine & storage engine same process (no duplication & GC)



TACHYON NEXUS

Comparison with In-Memory HDFS



TACHYON NEXUS

Outline

- Overview
  - Motivation
  - Tachyon Architecture
  - Using Tachyon
- **Open Source**
  - Status
  - Production Use Cases
- Roadmap

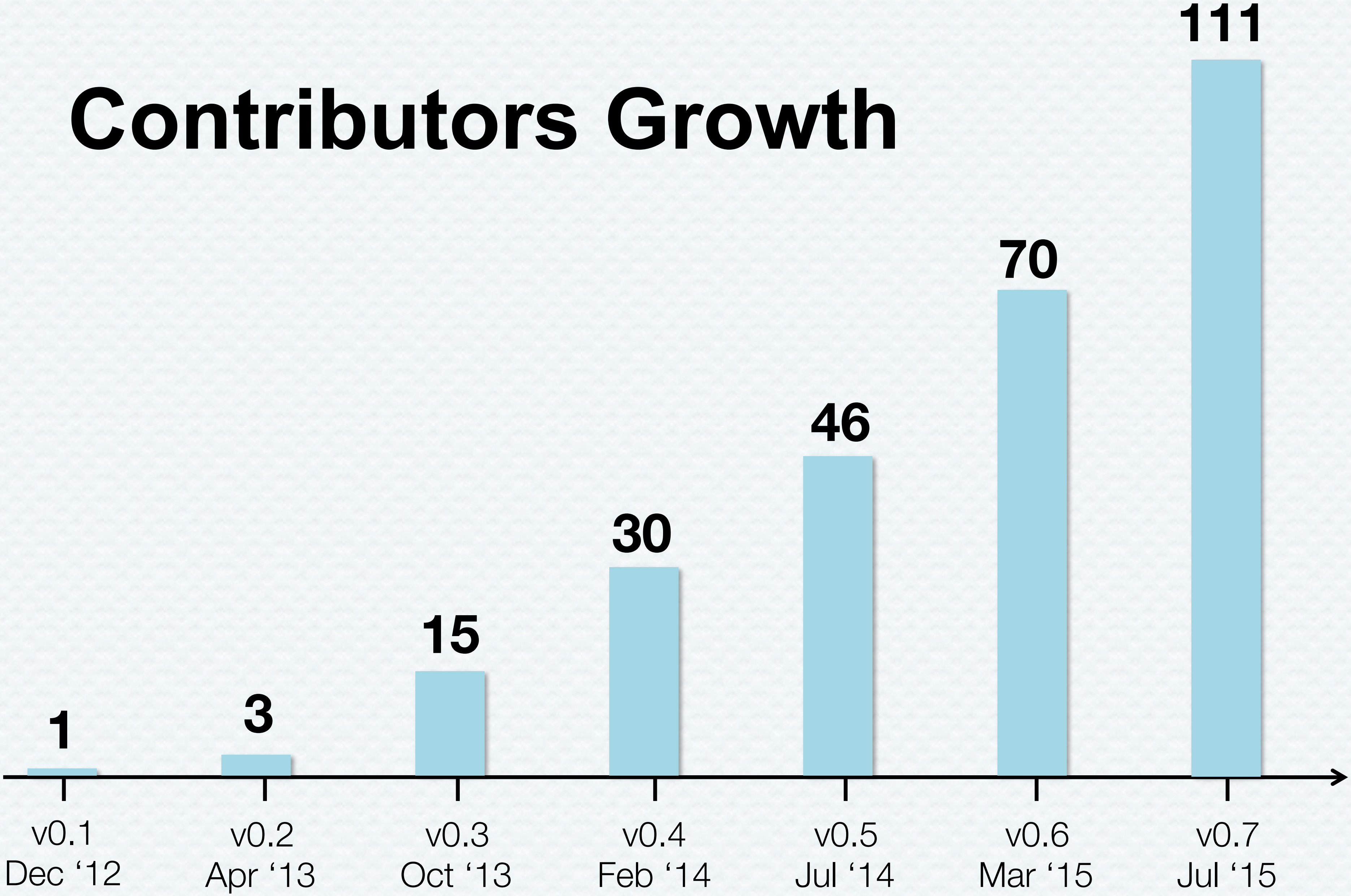
TACHYON NEXUS

Open Source Status

- Started at UC Berkeley AMPLab in Summer 2012
- Apache License 2.0, Version 0.7.1 (August 2015)

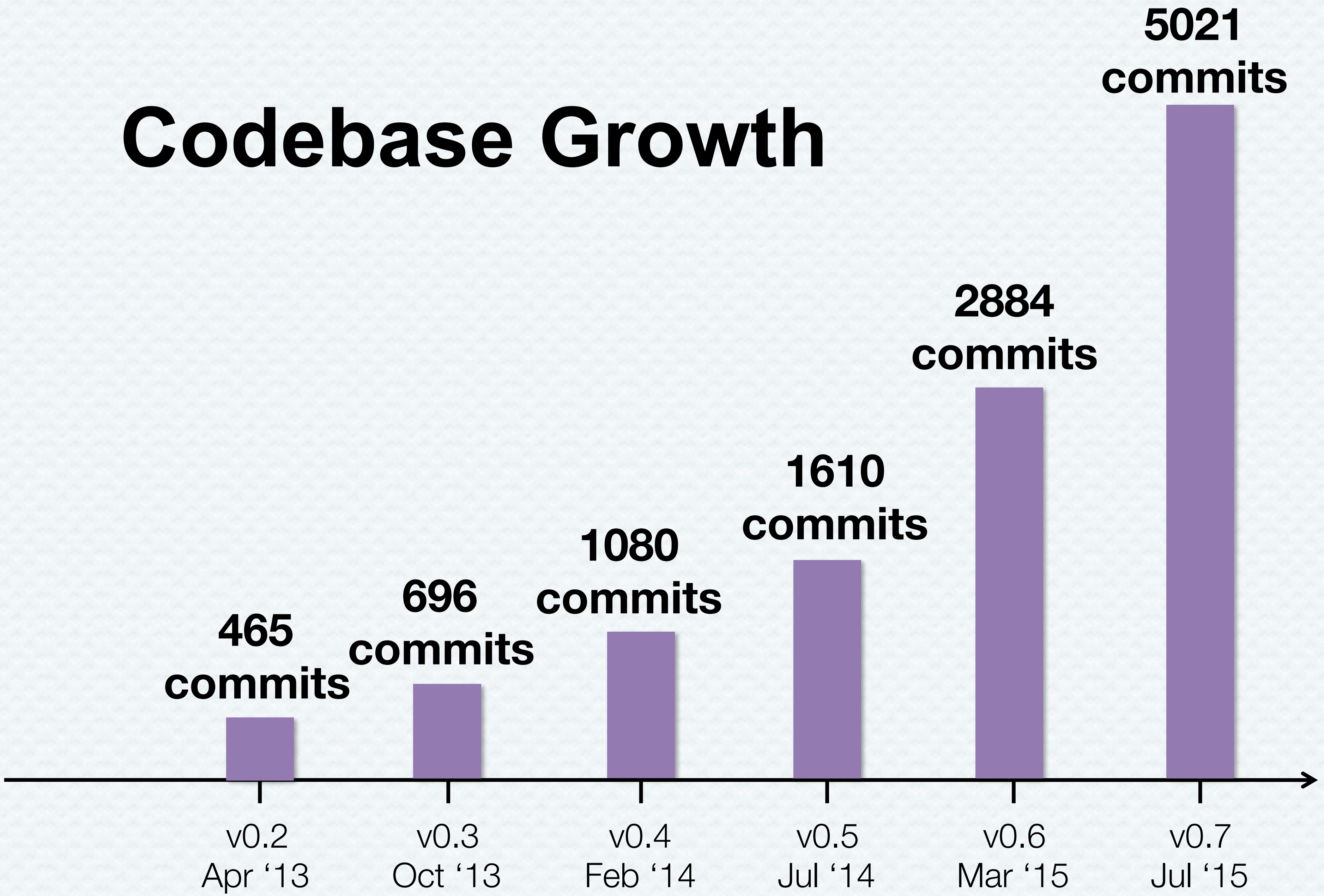
- Deployed at > 50 companies (July 2014)
- 30+ Companies Contributing

Contributors Growth



TACHYON
NEXUS

Codebase Growth



TACHYON NEXUS

Thanks to Our Contributors!

YAHOO!

intel

IBM

Bai du 百度

CLEMSON UNIVERSITY SOUTH CAROLINA 1889

HUAWEI

TACHYON NEXUS

NOKIA

databricks

CRAY THE SUPERCOMPUTER COMPANY

redhat

Berkeley UNIVERSITY OF CALIFORNIA

NANJING UNIVERSITY 1902

Palantir Technologies

Pivotal

Atigeo

Mellanox TECHNOLOGIES

cloudera

TSINGHUA UNIVERSITY 1911

DATA ClearStory Now You See It

Reported Tachyon Usage

8.17.2015

Tachyon for ultra-fast Big Data processing

Editor's note: This article is by cloud analytics infrastructure expert Gil Vernik, IBM Research-Haifa.

Today's massive growth in data sets means that storage is increasingly becoming a critical bottleneck for system workloads. My storage team in Haifa, Israel wants to analyze and understand these massive volumes of data, and we need to store them somewhere reliable. Although disk space is an option, it's too slow to carry out fast Big Data processing. In-memory computing, which keeps the data in a server's RAM for fast access and processing, offers a good solution for processing Big Data workloads – but it's limited and expensive.

Enter Tachyon, a memory-centric distributed storage system that offers processing at memory-speed and reliable storage. Its software works with servers in clusters so there's plenty of room for storage, and a unique proprietary feature eliminates the need for replication to ensure fault tolerance. Now, we've connected Tachyon to Swift so it can work effortlessly with Swift and SoftLayer. The result? Tachyon is even more flexibile and efficient.

TACHYON NEXUS

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

Use Case: Baidu

- Framework: SparkSQL
- Under Storage: Baidu's File System
- Storage Media: MEM + HDD
- 100+ nodes deployment
- 1PB+ managed space
- 30x Performance Improvement

Use Case: a SAAS Company

- Framework: Impala
- Under Storage: S3
- Storage Media: MEM + SSD
- 15x Performance Improvement

TACHYON NEXUS

Use Case: an Oil Company

- Framework: Spark
- Under Storage: GlusterFS
- Storage Media: MEM only
- Analyzing data in traditional storage

Use Case: a SAAS Company

- Framework: Spark
- Under Storage: S3
- Storage Media: SSD only
- Elastic Tachyon deployment

TACHYON NEXUS

Outline

- Overview
  - Motivation
  - Tachyon Architecture
  - Using Tachyon
- Open Source
  - Status
  - Production Use Cases
- **Roadmap**

TACHYON NEXUS

New Features

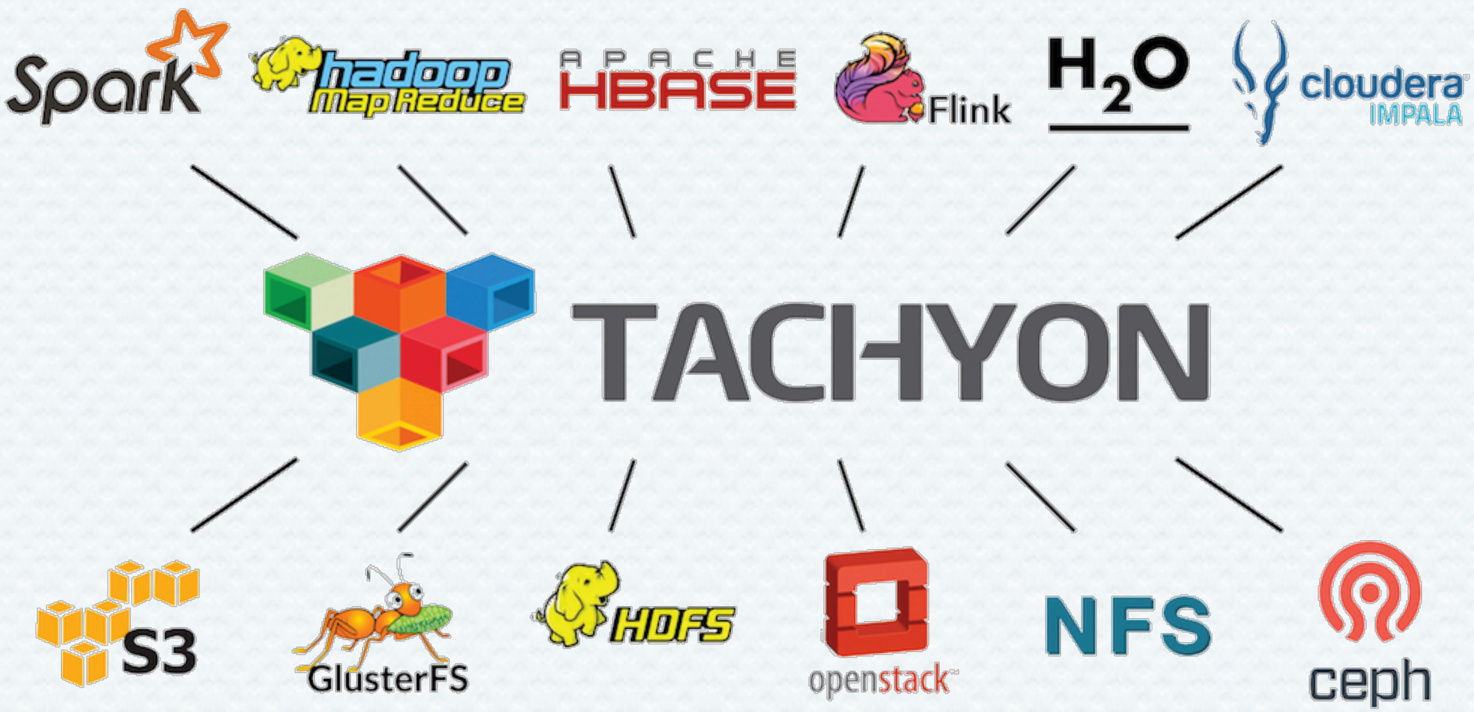
- Lineage in Storage (alpha)
- Tiered Storage (alpha)

New Features

- Lineage in Storage (alpha)
- Tiered Storage (alpha)
- Data Serving
- Support for New Hardware
- …
- Your New Feature!

Tachyon’s Goal?

Distributed Memory-Centric Storage: Better Assist Other Components



Welcome Collaboration!

[JIRA New Contributor Tasks]

TACHYON

- Website: http://tachyon-project.org
- Github: https://github.com/amplab/tachyon
- Meetup: http://www.meetup.com/Tachyon
- News Letter Subscription: http://goo.gl/mwB2sX
- Email: haoyuan@tachyonnexus.com