Shark Update and Upcoming Changes

Reynold Xin

AMPLab, UC Berkeley

May 9, 2013

Release Versioning & Schedule

Shark	Spark	Time
0.1	0.5	Apr 2012

Release Versioning & Schedule

Shark	Spark	Time
0.1	0.5	Apr 2012
0.2	0.6	Oct 2012

Release Versioning & Schedule

Shark	Spark	Time
0.1	0.5	Apr 2012
0.2	0.6	Oct 2012
0.2.1	0.6.1	Nov 2012

Release Versioning & Schedule

Shark	Spark	Time
0.1	0.5	Apr 2012
0.2	0.6	Oct 2012
0.2.1	0.6.1	Nov 2012
0.3	???	???

Release Versioning & Schedule

1. Synchronize Spark and Shark version numbers

2. Faster release schedule

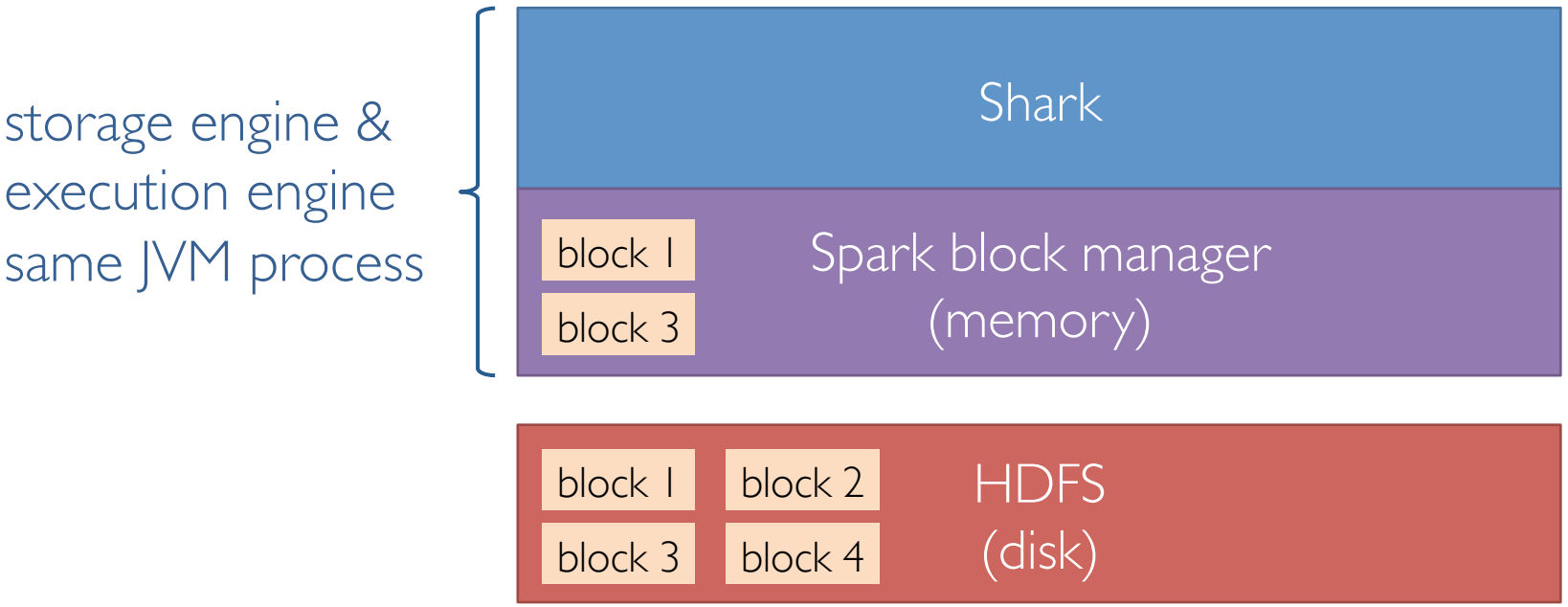
Release Versioning & Schedule

Shark	Spark	Time
0.1	0.5	Apr 2012
0.2	0.6	Oct 2012
0.2.1	0.6.1	Nov 2012
~~0.3~~	???	???
0.7	0.7	May 2013
0.8	0.8	Summer 2013

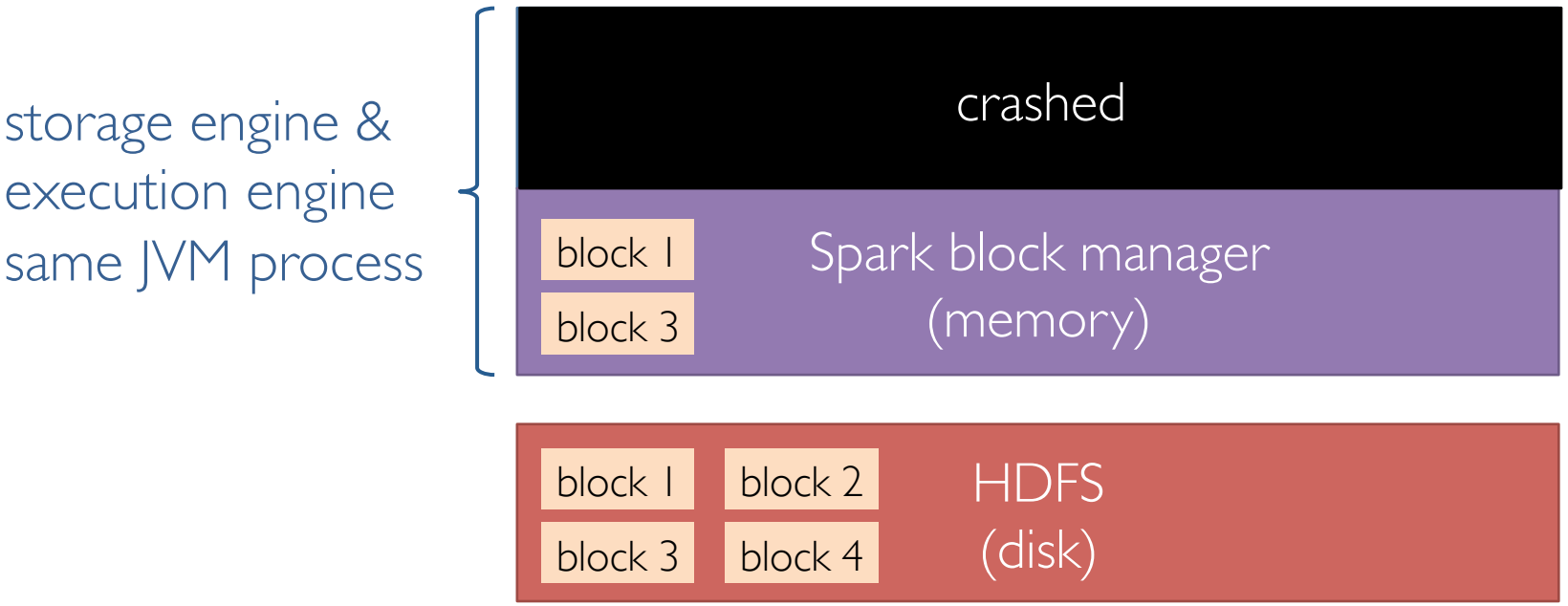
Remainder of the talk

1. Tachyon integration
2. Improvements in 0.7
3. Planned improvements in 0.8+

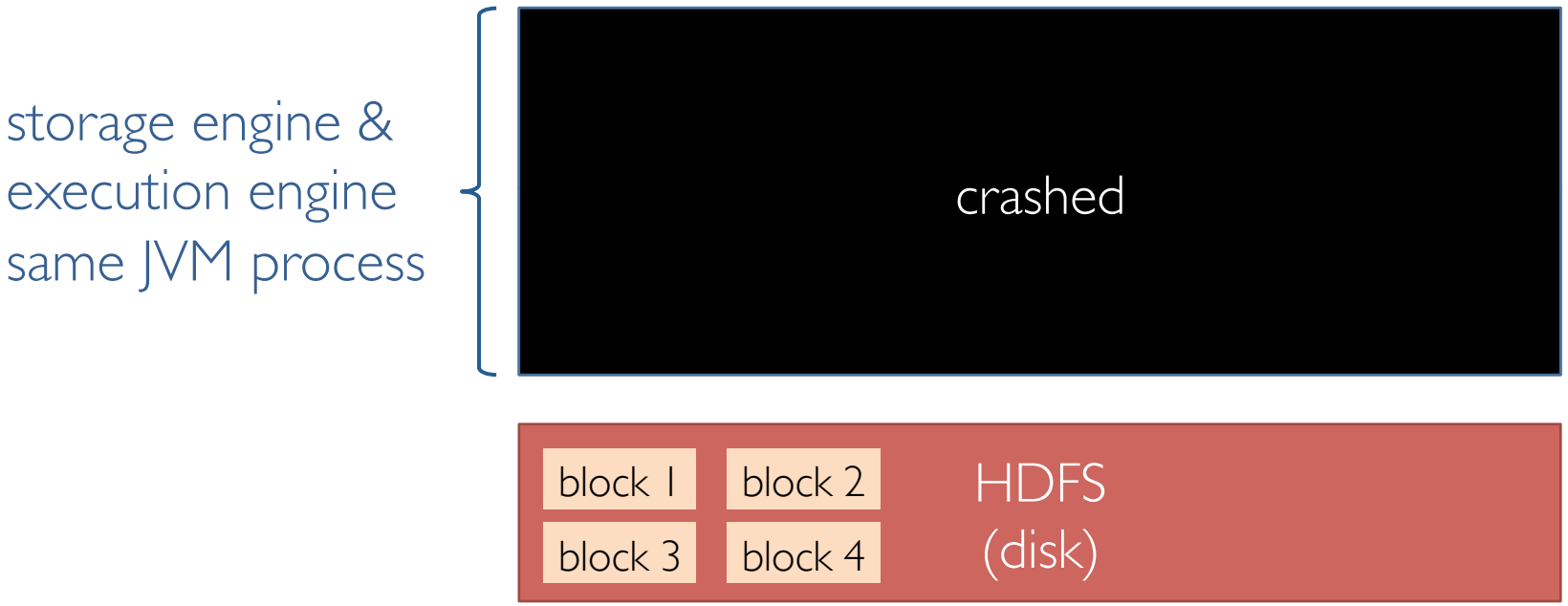
Shark before Tachyon



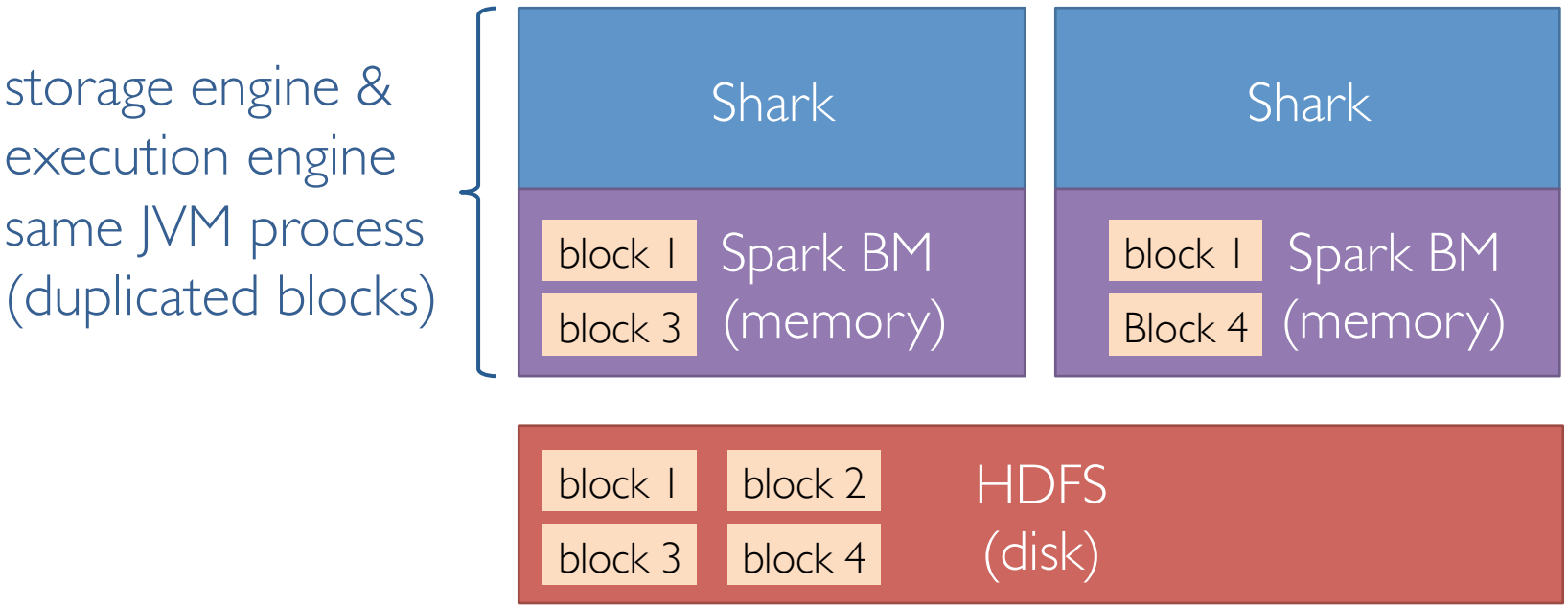
Shark before Tachyon



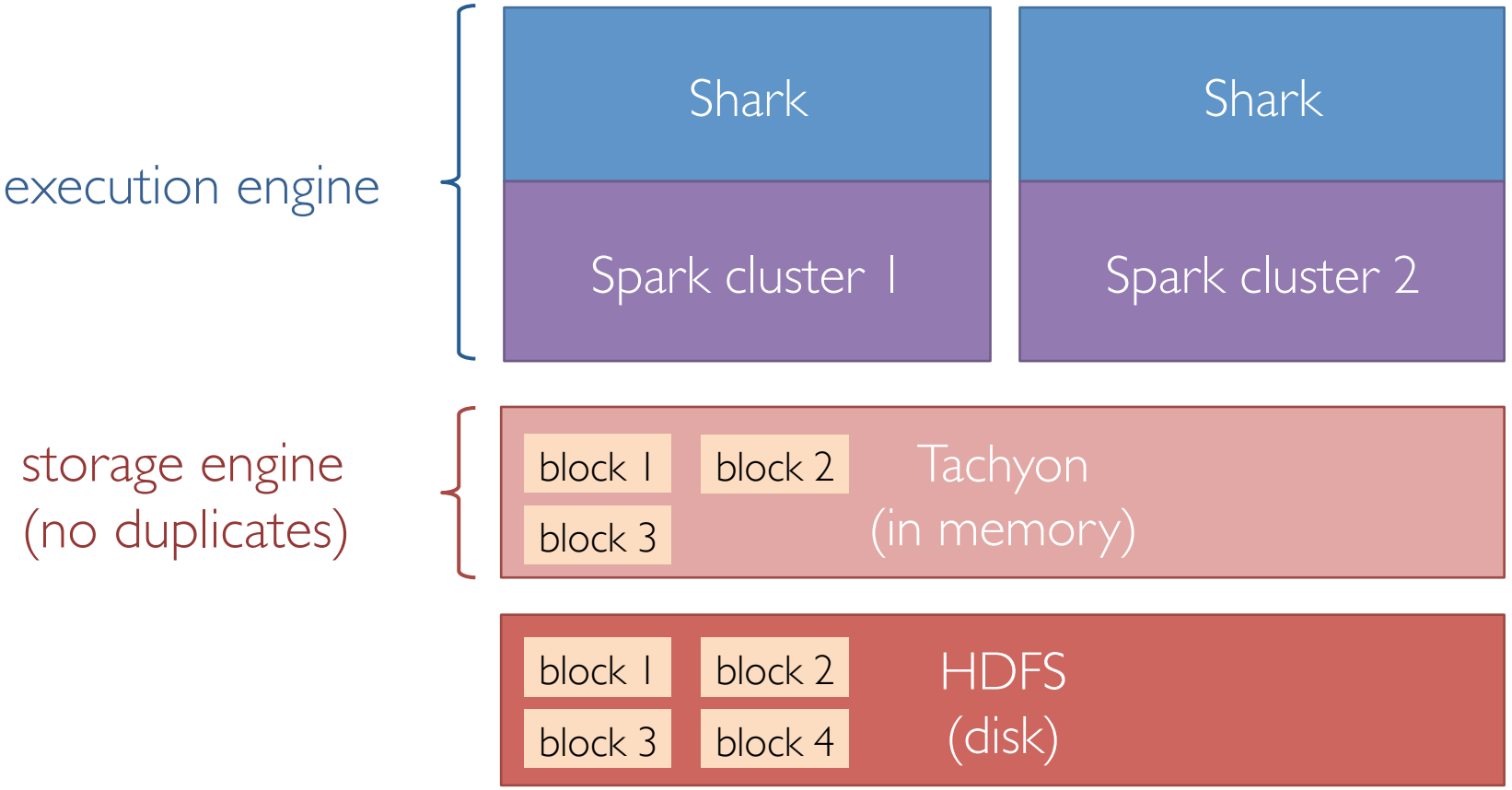
Shark before Tachyon Loses Cache during Crash



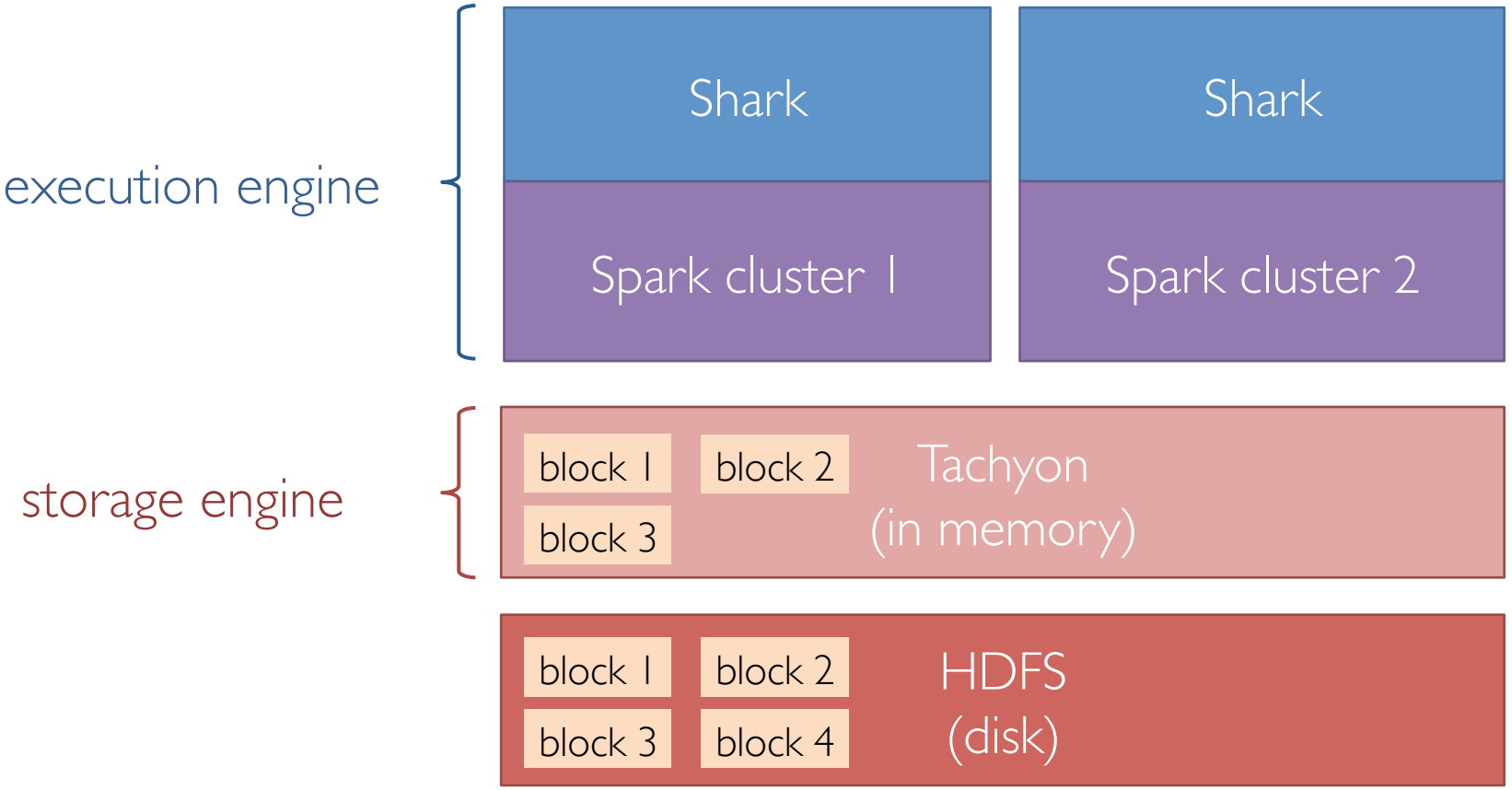
Shark before Tachyon
Duplicate Memory Blocks



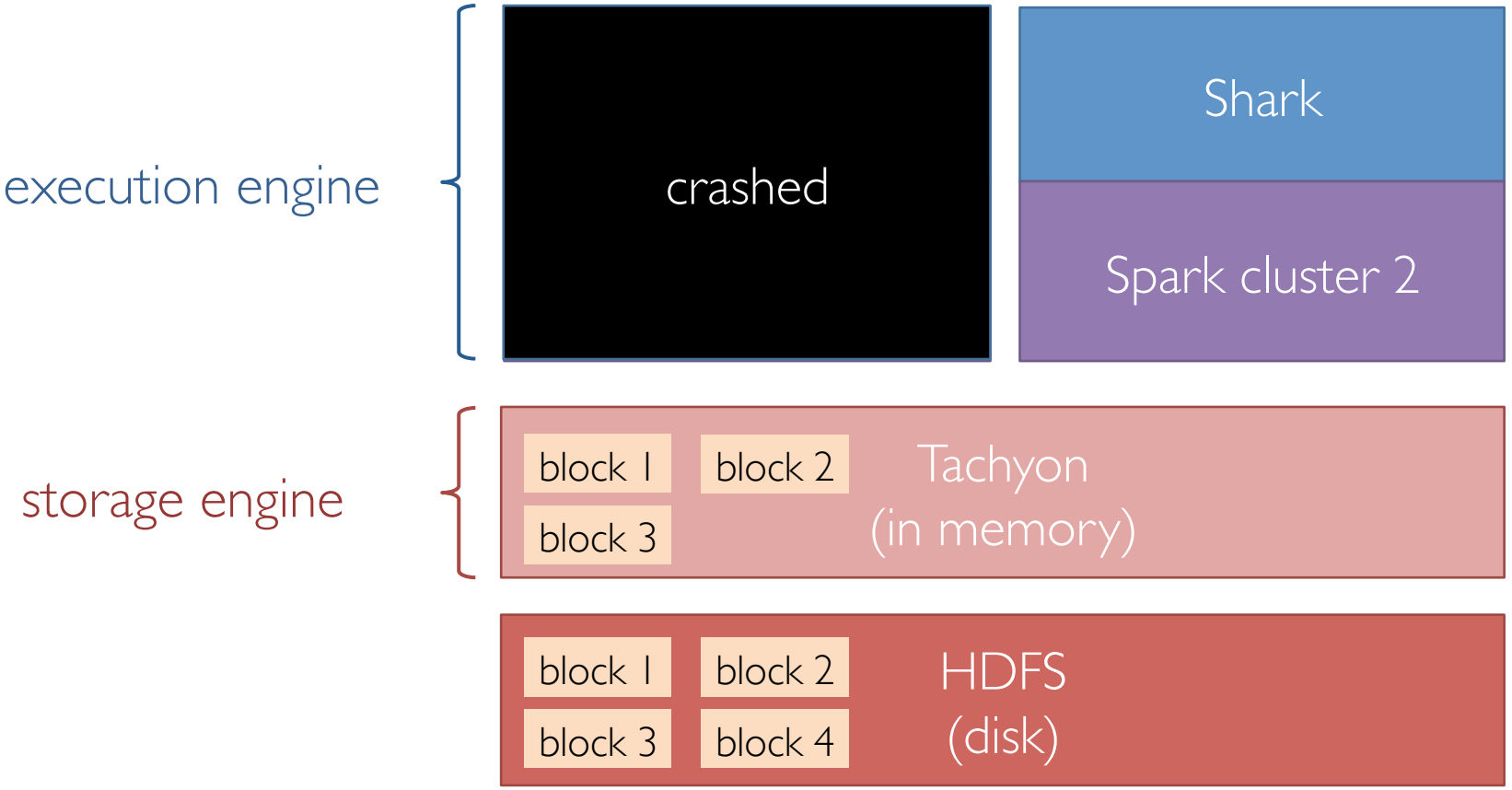
Tachyon
In-memory Data Sharing



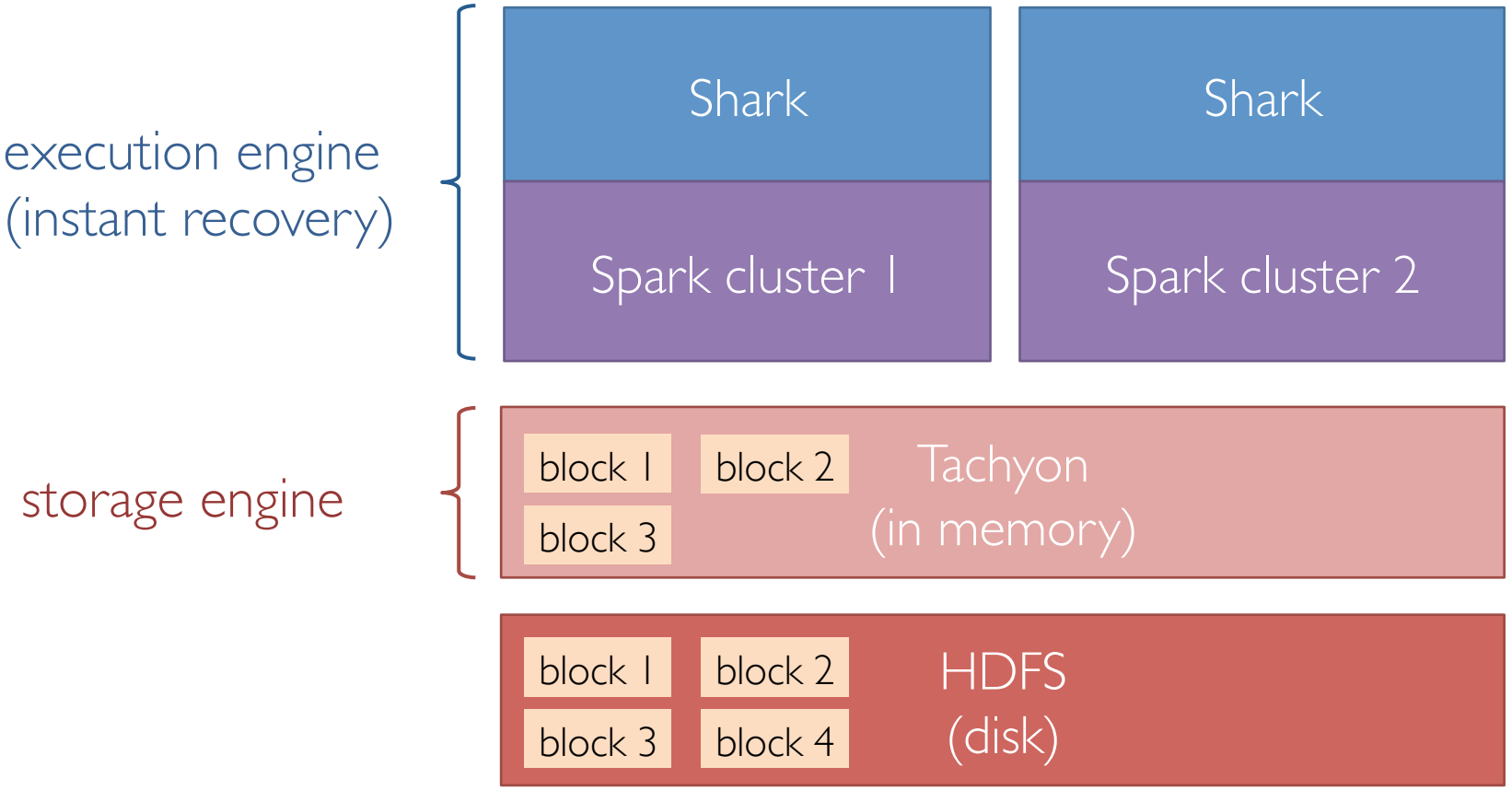
Tachyon
Instant Recovery



Tachyon
Instant Recovery



Tachyon
Instant Recovery



Shark with Tachyon

```
CREATE TABLE data TBLPROPERTIES(“shark.cache” = “tachyon”)
AS SELECT a, b, c from data_on_disk WHERE month=“May”
```

1. In-memory data sharing across multiple Shark instances (i.e. stronger isolation)
2. Instant recovery of in-memory tables
3. Reduced heap size => faster GC

Isn't it slow for JVM programs to deserialize off-heap data?

Efficient Tachyon Integration

Tachyon provides a column-based API: Shark table columns are stored as files in Tachyon (RAMFS)

Java NIO memory-mapped files (no memory copy)

"Unsafe" for DirectByteBuffer reads (C style memory reads)

```
/*
 * %W% %E%
 *
 * Copyright (c) 2006,2010 Oracle and/or its affiliates. All rights reserved.
 * ORACLE PROPRIETARY/CONFIDENTIAL. Use is subject to license terms.
 */

package sun.misc;

import java.security.*;
import java.lang.reflect.*;


/**
 * A collection of methods for performing low-level, unsafe operations.
 * Although the class and all methods are public, use of this class is
 * limited because only trusted code can obtain instances of it.
 *
 * @author John R. Rose
 * @version  %I%, %E%
 * @see #getUnsafe
 */

public final class Unsafe {

    private static native void registerNatives();
    static {
        registerNatives();
    }

    private Unsafe() {}

    private static final Unsafe theUnsafe = new Unsafe();

    /**
     * Provides the caller with the capability of performing unsafe
     * operations.
```

Other Improvements in 0.7

Enhanced EC2/S3/EMR Support

- CLI can directly execute queries defined in a S3 file (`bin/shark -f s3://…`)
- Picks up AWS credentials from environmental variables automatically

New Data Types: timestamp, binary

Avro SerDes

Maven / Debian package (ClearStory)

Other Improvements in 0.7

Improved sql2rdd API (ClearStory & AMP)

Improved LIMIT 0 handling

- » Avoid launching any tasks if LIMIT 0
- » Some BI tools use LIMIT 0 to test whether a table exists

Improved map join implementation (Yahoo!)

Inserting data into in-memory tables

Bug fixes (ClearStory)

Improvements (0.8+)

Fair scheduler for Shark server (Intel)

Improved shuffle on 16+ cores (Intel)

Performance improvements for high cardinality joins and aggregations (AMP)

Expression byte code generation (Yahoo! & Intel)

Remove cached tables/partitions (Yahoo! & AMP)

In-memory data compression

Thanks!

We are looking for
future meetup locations.