Shark Update and Upcoming Changes

Reynold Xin
AMPLab, UC Berkeley
May 9, 2013
## Release Versioning & Schedule

<table>
<thead>
<tr>
<th>Shark</th>
<th>Spark</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>Apr 2012</td>
</tr>
</tbody>
</table>
## Release Versioning & Schedule

<table>
<thead>
<tr>
<th></th>
<th>Shark</th>
<th>Spark</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5</td>
<td></td>
<td>Apr 2012</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6</td>
<td></td>
<td>Oct 2012</td>
</tr>
</tbody>
</table>
### Release Versioning & Schedule

<table>
<thead>
<tr>
<th>Shark</th>
<th>Spark</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>Apr 2012</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6</td>
<td>Oct 2012</td>
</tr>
<tr>
<td>0.2.1</td>
<td>0.6.1</td>
<td>Nov 2012</td>
</tr>
</tbody>
</table>
# Release Versioning & Schedule

<table>
<thead>
<tr>
<th>Shark</th>
<th>Spark</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>Apr 2012</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6</td>
<td>Oct 2012</td>
</tr>
<tr>
<td>0.2.1</td>
<td>0.6.1</td>
<td>Nov 2012</td>
</tr>
<tr>
<td>0.3</td>
<td>???</td>
<td>???</td>
</tr>
</tbody>
</table>
Release Versioning & Schedule

1. Synchronize Spark and Shark version numbers

2. Faster release schedule
Release Versioning & Schedule

<table>
<thead>
<tr>
<th>Shark</th>
<th>Spark</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>Apr 2012</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6</td>
<td>Oct 2012</td>
</tr>
<tr>
<td>0.2.1</td>
<td>0.6.1</td>
<td>Nov 2012</td>
</tr>
<tr>
<td>0.3</td>
<td>???</td>
<td>???</td>
</tr>
<tr>
<td>0.7</td>
<td>0.7</td>
<td>May 2013</td>
</tr>
<tr>
<td>0.8</td>
<td>0.8</td>
<td>Summer 2013</td>
</tr>
</tbody>
</table>
Remainder of the talk

1. Tachyon integration

2. Improvements in 0.7

3. Planned improvements in 0.8+
Shark before Tachyon

Storage engine & execution engine
same JVM process

Shark

Spark block manager
(memory)

Block 1
Block 3

HDFS
(disk)

Block 1
Block 2
Block 3
Block 4
Shark before Tachyon

storage engine & execution engine
same JVM process

crashed

Spark block manager
(memory)

HDFS
(disk)
Shark before Tachyon
Loses Cache during Crash

storage engine &
execution engine
same JVM process

block 1  block 2  HDFS
block 3  block 4  (disk)
Shark before Tachyon
Duplicate Memory Blocks

storage engine &
execution engine
same JVM process
(duplicated blocks)
Tachyon
In-memory Data Sharing

execution engine

storage engine (no duplicates)

Shark
Spark cluster 1

Shark
Spark cluster 2

Tachyon
(in memory)

block 1
block 2
block 3

block 1
block 2
block 3
block 4

HDFS
(disk)
Tachyon
Instant Recovery
Tachyon
Instant Recovery

- **Execution Engine**
  - Crashed

- **Storage Engine**
  - Tachyon (in memory)
  - HDFS (disk)
Tachyon
Instant Recovery

- Execution engine (instant recovery)
- Storage engine

Tachyon (in memory)

HDFS (disk)
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)
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 $Id$, $E$
 * @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.