Amazon Web Services File System
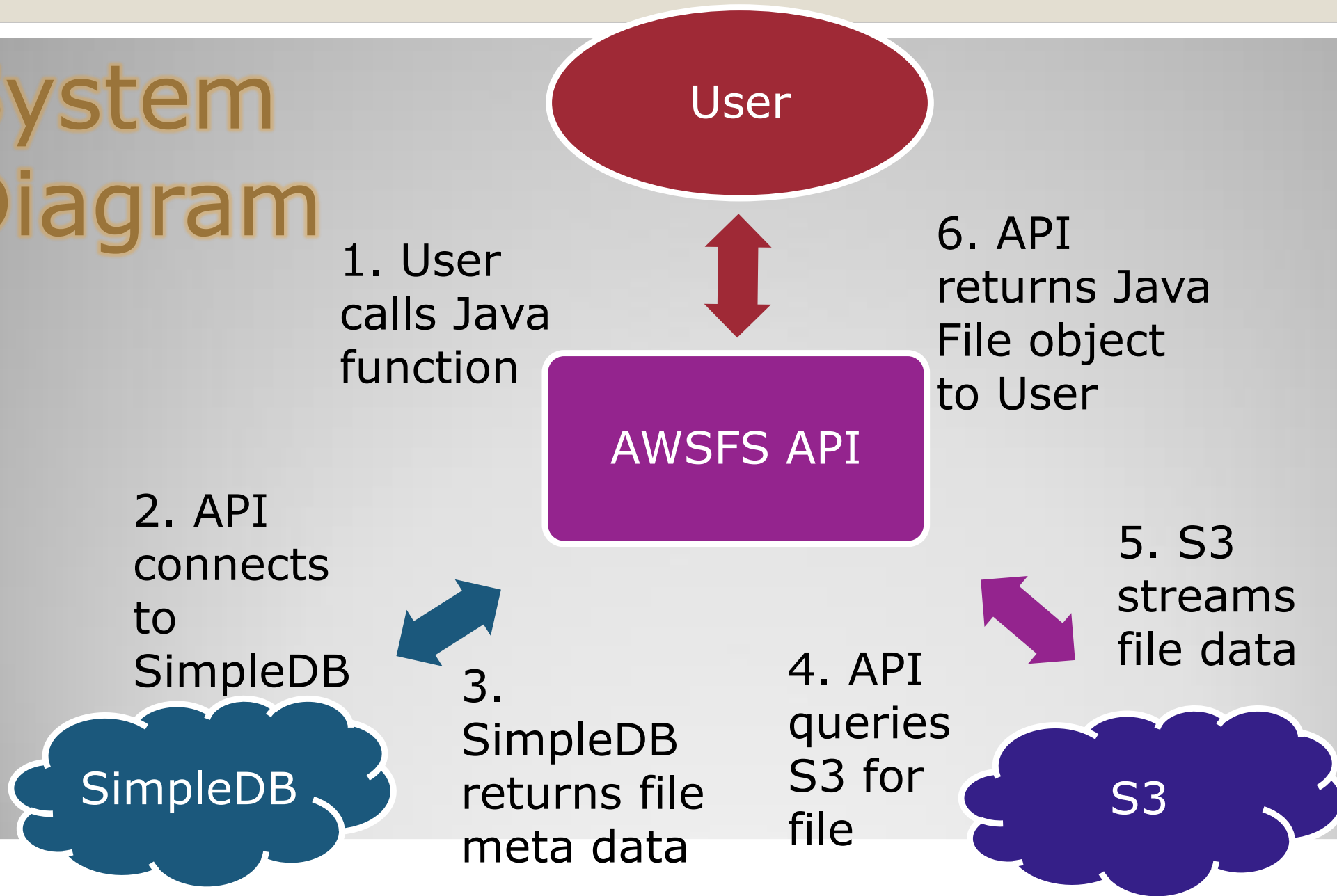
By: Andrew Caron and Eric Antunes

Objectives

- Cloud-based
  - Allow multiple users to simultaneously connect, store data, modify data, and retrieve data
  - Easy to use Java API for system-agnostic access

- Dynamic and Scalable
  - Ability to handle numerous large data requests
  - Full ability to create full directory trees of near limitless size

- Full scale of file system operations
  - Reading, writing, opening, closing and similar file commands
  - Directory trees, links, renaming, moving, copying

- Amazon Web Services S3
  - Provides flat storage space to host files
  - Allows uploading and downloading

- Amazon Web Services SimpleDB
  - Highly available and flexible data store
  - Non-relation data
  - Easy store and query API

Cloud Services Used

System Diagram

User

1. User calls Java function

6. API returns Java File object to User

AWSFS API

2. API connects to SimpleDB

5. S3 streams file data

3. SimpleDB returns file meta data

4. API queries S3 for file

SimpleDB

S3

```xml
<?xml version="1.0"?>
<GetAttributesResponse xmlns="http://sdb.amazonaws.com/doc/2009-04-15/">
  <GetAttributesResult>
    <Attribute>
      <Name>date created</Name>
      <Value>6-9-2013</Value>
    </Attribute>
    <Attribute>
      <Name>name</Name>
      <Value>test.txt</Value>
    </Attribute>
    <Attribute>
      <Name>parent id</Name>
      <Value>0</Value>
    </Attribute>
    <Attribute>
      <Name>type</Name>
      <Value>file</Value>
    </Attribute>
    + <Attribute>
  </GetAttributesResult>
  <ResponseMetadata>
    <RequestId>e24bc752-7375-061a-6c7a-a83834ab4a0b</RequestId>
    <BoxUsage>0.0000093702</BoxUsage>
  </ResponseMetadata>
</GetAttributesResponse>
```

Upload   **Create Folder**   **Actions** ▾   **None**   **Properties**   **Transfers**   ⟳

All Buckets / **awsfs-demo**

	Name	Storage Class	Size	Last Modified
☐ ♪	08 All the Small Things.m4p	Standard	3 MB	Thu Dec 04 23:58:57 GMT-500 2014
☐ 🗋	08_Interactive_Datapath_Example.pptx	Standard	424.7 KB	Thu Dec 04 23:59:00 GMT-500 2014
☐ 🖼	2.jpg	Standard	188.7 KB	Thu Dec 04 23:58:56 GMT-500 2014
☐ 🗜	61340e85.zip	Standard	1.4 KB	Thu Dec 04 23:59:01 GMT-500 2014
☐ 🗎	Assignment 0.docx	Standard	65.7 KB	Thu Dec 04 23:57:20 GMT-500 2014
☐ 🗋	compare.c	Standard	4 KB	Thu Dec 04 23:57:21 GMT-500 2014
☐ 🖼	otter.jpg	Standard	11.2 KB	Thu Dec 04 23:57:20 GMT-500 2014
☐ 🗎	report.pdf	Standard	95.7 KB	Thu Dec 04 23:57:22 GMT-500 2014
☐ 🗎	test.txt	Standard	61 bytes	Thu Dec 04 23:57:20 GMT-500 2014
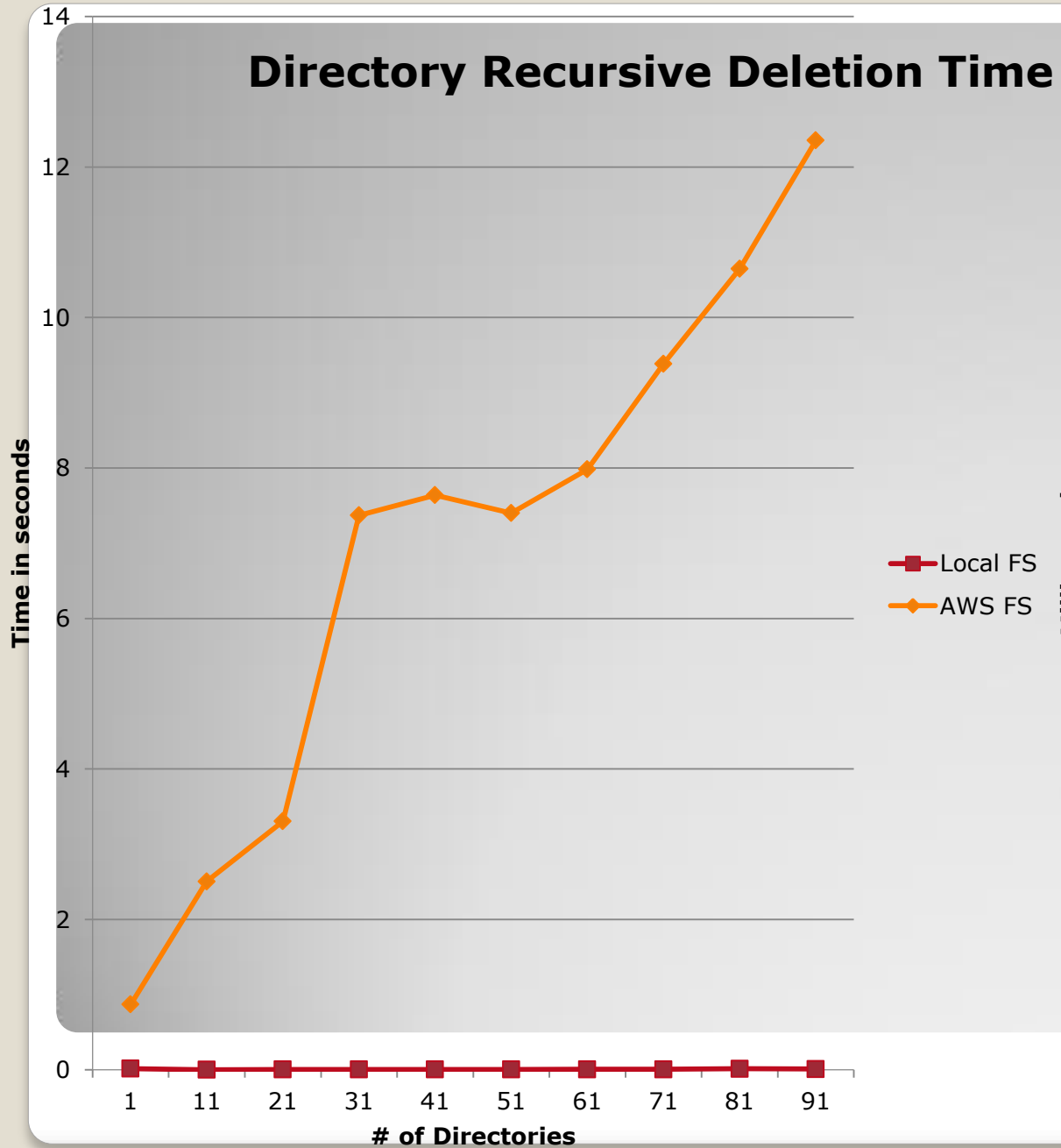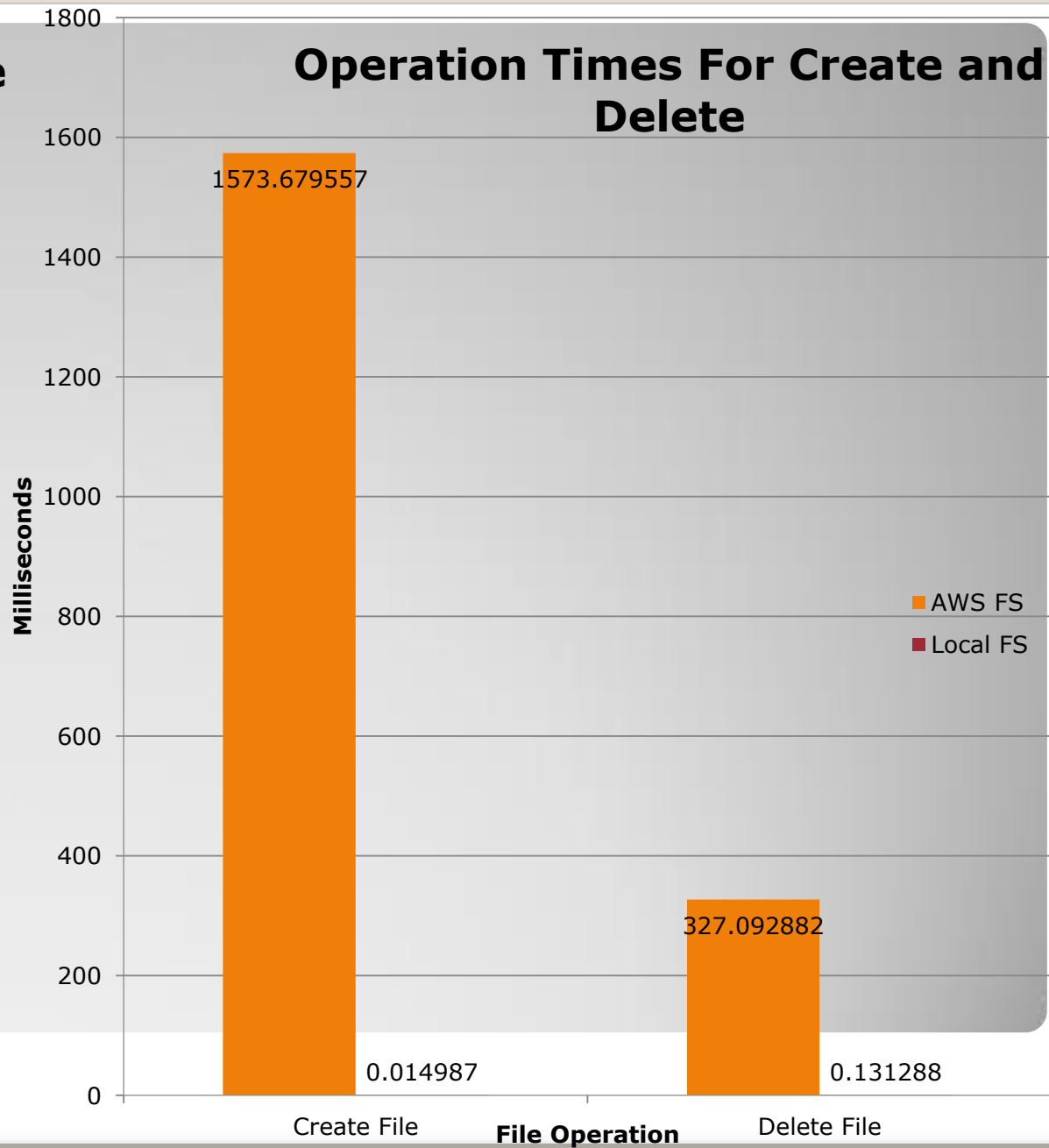
Challenges

- Reduce need for multiple SimpleDB queries for a single file
- Ability to perform multiple reads/writes on a file in quick succession
- Keep data (especially meta data) consistent between client, S3, and SimpleDB
- Unique identifiers for files separate from name or path for S3 storage and file operations

Results

- Dynamic file system that is highly available and partition tolerant
- Accessible from any system supporting Java
- Simple yet extensive API to cover all needs of users for a file system

Questions?