Accelerating Spark Workloads in a Mesos Environment with Alluxio

Gene Pang, Software Engineer, Alluxio, Inc.
About Me

Gene Pang
Software Engineer @ Alluxio, Inc.
Alluxio Open Source PMC Member
Ph.D. from AMPLab @ UC Berkeley
Worked at Google before UC Berkeley
Twitter: @unityxx
Github: @gpang
Outline

1. Alluxio Overview
2. Alluxio + Spark + Mesos Use Cases
3. Using Spark with Alluxio on Mesos
4. Deployment with Mesos
5. Demo
Data Ecosystem Yesterday

- One Compute Framework
- Single Storage System
- Co-located
Data Ecosystem Today

• Many Compute Frameworks
• Multiple Storage Systems
• Most not co-located
Data Ecosystem Issues

- Each application manages multiple data sources
- Add/Removing data sources require application changes
- Storage optimizations require application change
- Lower performance due to lack of locality
Data Ecosystem with Alluxio

- Apps only talk to Alluxio
- Simple Add/Remove
- No App Changes
- Memory Performance
Next Gen Analytics with Alluxio

Native File System  Hadoop Compatible File System  Native Key-Value Interface  Fuse Compatible File System

HDFS Interface  Amazon S3 Interface  Swift Interface  GlusterFS Interface

Apps, Data & Storage at Memory Speed

✓ Big Data/IoT  ✓ AI/ML  ✓ Deep Learning  ✓ Cloud Migration  ✓ Multi Platform  ✓ Autonomous
Enabling Next Gen Analytics

1. Unify your Data
2. Architecture Flexibility
3. Improved I/O Performance
Fastest Growing Big Data Open Source Project

- Fastest Growing open-source project in the big data ecosystem
- Running world’s largest production clusters
- 600+ Contributors from 100+ organizations
Big Data Case Study – Qunar.Com

Challenge –
Gain end to end view of business with large volume of data for $5B Travel Site

Queries were slow / not interactive, resulting in operational inefficiency

Solution –
With Alluxio, 300x improvement in performance

Impact –
Increased revenue from immediate response to user behavior
Machine Learning Case Study –

Challenge –
Disparate Data both on-prem and Cloud. Heterogeneous types of data.
Scaling of Exabyte size data. Slow due to disk based approach.

Solution –
Using Alluxio to prevent I/O bottlenecks

Impact –
Orders of magnitude higher performance than before.
Outline

1. Alluxio Overview
2. Alluxio + Spark + Mesos Use Cases
3. Using Spark with Alluxio on Mesos
4. Deployment with Mesos
5. Demo
Sharing Data via Memory

- Two copies of data in memory – double the memory used
- Inter-process Sharing Slowed Down by Network / Disk I/O
Sharing Data via Memory

- Half the memory used
- Inter-process Sharing Happens at Memory Speed
Data Resilience During Crash

Spark Compute

Spark Storage

Mesos

HDFS / Amazon S3

Storage Engine & Execution Engine
Same Process

©2017 Alluxio, Inc. All Rights Reserved
Data Resilience During Crash

- Process Crash Requires Network and/or Disk I/O to Re-read Data
Data Resilience During Crash

- Process Crash Requires Network and/or Disk I/O to Re-read Data
Data Resilience During Crash

Spark Compute

Spark Storage

Alluxio

Storage Engine & Execution Engine
Different process

Mesos

HDFS / Amazon S3
Data Resilience During Crash

Process Crash - Data is Re-read at Memory Speed

Storage Engine & Execution Engine Different process

HDFS / Amazon S3

Mesos

Alluxio

CRASH

block 1
block 3
block 4

block 1
block 2
block 3
block 4
Alluxio Architecture
Applications interact with Alluxio via the Alluxio client

- Native Alluxio Filesystem Client
  - Alluxio specific operations like [un]pin, [un]mount, [un]set TTL
- HDFS-Compatible Filesystem Client
  - No code change necessary
- S3 API
Alluxio Master

Master is responsible for managing metadata
- Filesystem namespace metadata
- Blocks / workers metadata

Primary master writes journal for durable operations
- Secondary masters replay journal entries
Alluxio Worker

Worker is responsible for managing block data
Worker stores block data on various storage media
  • HDD, SSD, Memory
Reads and writes data to underlying storage systems
Outline

1. Alluxio Overview
2. Alluxio + Spark + Mesos Use Cases
3. Using Spark with Alluxio on Mesos
4. Deployment with Mesos
5. Demo
Alluxio on DC/OS

Alluxio brings

A unified view of data across disparate storage systems

High performance & predictable SLA for analytics workloads

DC/OS makes provisioning infrastructure easy

Automates provisioning, management & elastic scaling

Benefits include:

Faster analytics with Spark and other frameworks

Process data from hybrid cloud storage systems (HDFS, S3, etc)
Outline

1. Alluxio Overview
2. Alluxio + Spark + Mesos Use Cases
3. Using Spark with Alluxio on Mesos
4. Deployment with Mesos
5. Demo
Demo Environment

SPARK

Alluxio

MESOS

Amazon Web Services S3
Demo Setup

Alluxio 1.5.0

DC/OS 1.9.4

Spark 2.0.2

Amazon EC2 (m3.xlarge)
Results

Duration (in seconds)

- **Alluxio**: 8x improvement
- **S3**

- **Initial Count**
- **Repeated After Restart**
Conclusion

Easy to use Alluxio with Spark in a Mesos environment

Predictable and improved performance

Easily connect to various storage systems
Thank you!

Gene Pang
Software Engineer
gene@alluxio.com