ALLUXIO

Building High-Performance, Concurrent & Scalable Filesystem Metadata Services

Featuring gRPC, Raft, and RocksDB

Bin Fan @ In-Memory Computing Summit

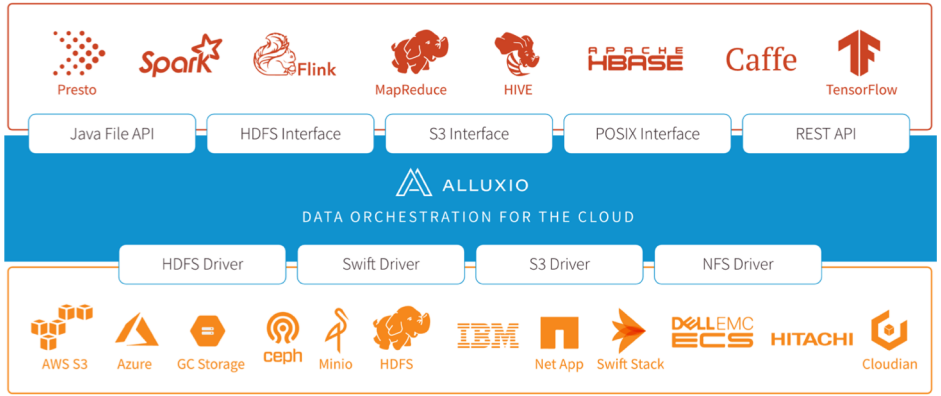
About Me

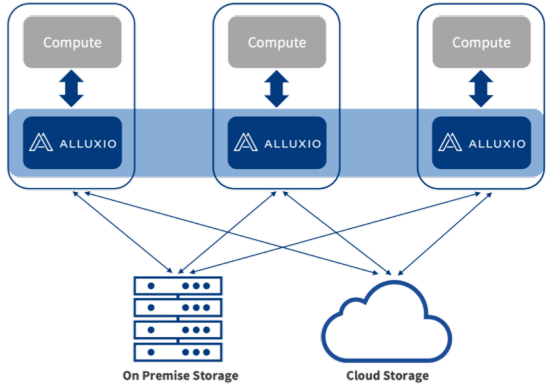
ALLUXIO

Bin Fan

- PhD CS@CMU
- Founding Member, VP of Open Source @ Alluxio
- Email: binfan@alluxio.com

Alluxio Overview



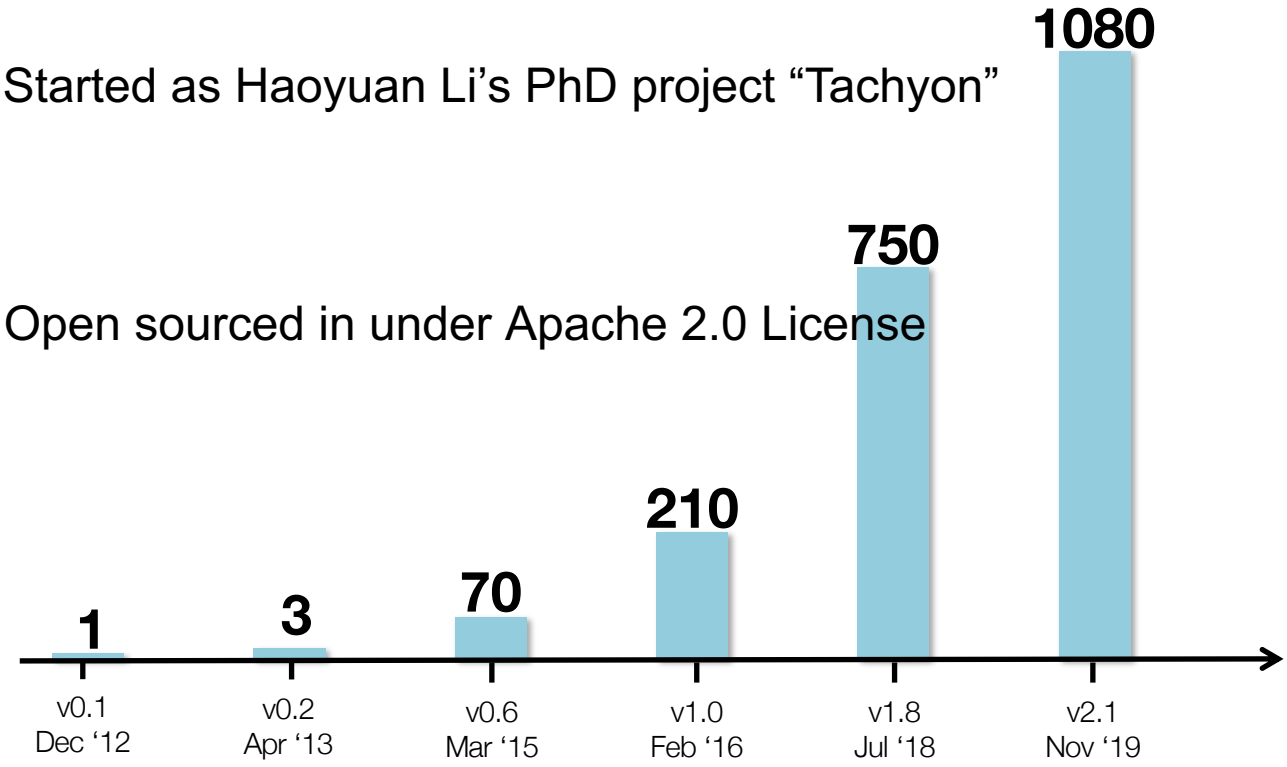


- Open source data orchestration
- Commonly used for data analytics such as OLAP on Hadoop
- Deployed at Huya, Walmart, Tencent, and many others
- Largest deployments of over 1000 nodes

Fast Growing Developer Community

Started as Haoyuan Li's PhD project "Tachyon"

Open sourced in under Apache 2.0 License



Deployed in Hundreds of Companies

Financial Services

DBS
ING
BARCLAYS
WELLS FARGO
PayPal
HUATAI SECURITIES

Retail & Entertainment

Walmart
Shopee
ROBLOX
CAESARS ENTERTAINMENT
NetEase Games
Myntra
JD.COM 京东
苏宁易购 suning.com
唯品会 VIPSHOP.COM

Data & Analytics Services

Aunalytics
LigaData
SAMSUNG SDS
TalkingData
RYTE

Technology

Tencent 腾讯
PERCEPTIN
esri
Tencent Cloud
@WalmartLabs
ESENTIRE
Baidu 百度
ORACLE
Alibaba Cloud
Baidu Cloud
Sogou 搜狗
CUELOGIC
Lucidworks
QINIU 七牛云存储
walkme
MOMO

Consumer

SAMSUNG
nielsen
YAHOO!
链家
Lenovo
bazaarvoice
GUARDANT HEALTH
网易严选

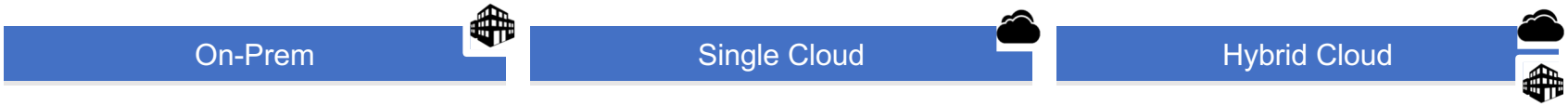
Telco & Media

COMCAST
HUAWEI
虎牙直播 huya.com
swisscom
中国移动 China Mobile
China unicom 中国联通

Travel & Transportation

waze
Ctrip
DiDi
去哪儿? Qunar.Com 聪明你的旅行

Deployed at Scale in Different Environment



- **Huya**: 1300+ nodes
- **Sogou**: 1000+ nodes
- **Momo**: 850 nodes

- **Bazaarvoice: AWS**
- **Ryte: AWS**
- **Walmart Labs: GCP**

- **DBS Bank**
- **ING Bank**
- **Comcast**

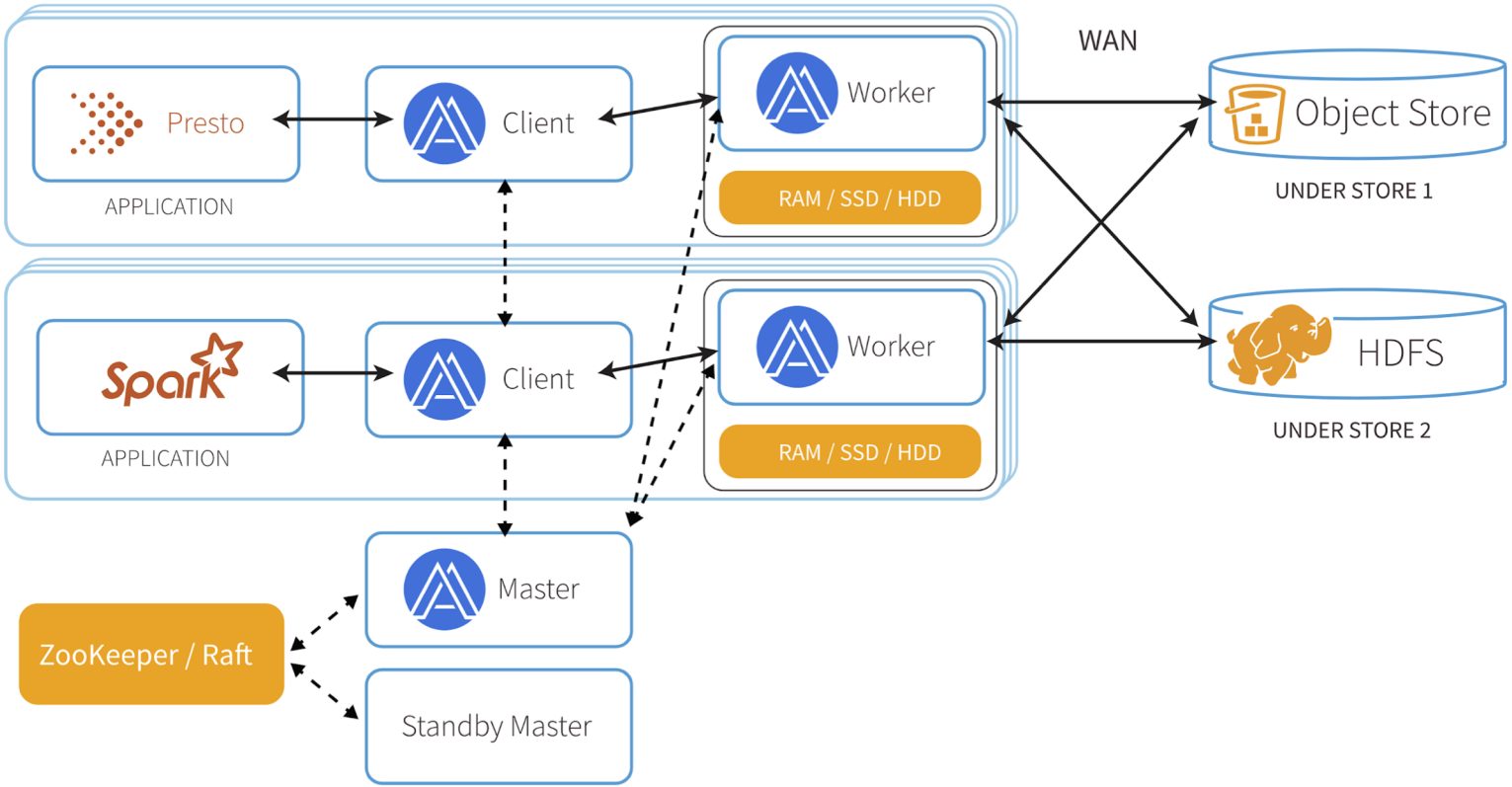
Agenda

1. Architecture
2. Challenges
3. Solutions

ALLUXIO

Architecture

Alluxio Architecture



Alluxio Master

- Responsible for storing and serving **metadata** in Alluxio
- What is Filesystem Metadata
  - Data structure of the Filesystem Tree (namespace)
    - Can include mounts of other file system namespaces
    - The size of the tree can be very large!
  - Data structure to map files to blocks and their locations
    - Very dynamic in Alluxio
  - Who is the primary master
    - One primary + several standby masters

Challenges

Metadata Storage Challenges

- Storing the raw metadata becomes a problem with a large number of files
- On average, each file takes 1KB of on-heap storage
  - 1 billion files would take 1 TB of heap space!
  - A typical JVM runs with < 64GB of heap space
  - GC becomes a big problem when using larger heaps

Metadata Storage Challenges

- Durability for the metadata is important
  - Need to restore state after planned or unplanned restarts or machine loss
- The speed at which the system can recover determines the amount of downtime suffered
  - Restoring a 1TB sized snapshot takes a nontrivial amount of time!

Metadata Serving Challenges

- Common file operations (ie. getStatus, create) need to be fast
  - On heap data structures excel in this case
- Operations need to be optimized for high concurrency
  - Generally many readers and few writers for large-scale analytics

Metadata Serving Challenges

- The metadata service also needs to sustain high load
  - A cluster of 100 machines can easily house over 5k concurrent clients!
- Connection life cycles need to be managed well
  - Connection handshake is expensive
  - Holding an idle connection is also detrimental

Solutions: Combining Different Open-Source Technologies as Building Blocks

ALLUXIO

Solving Scalable Metadata Storage Using RocksDB

ALLUXIO

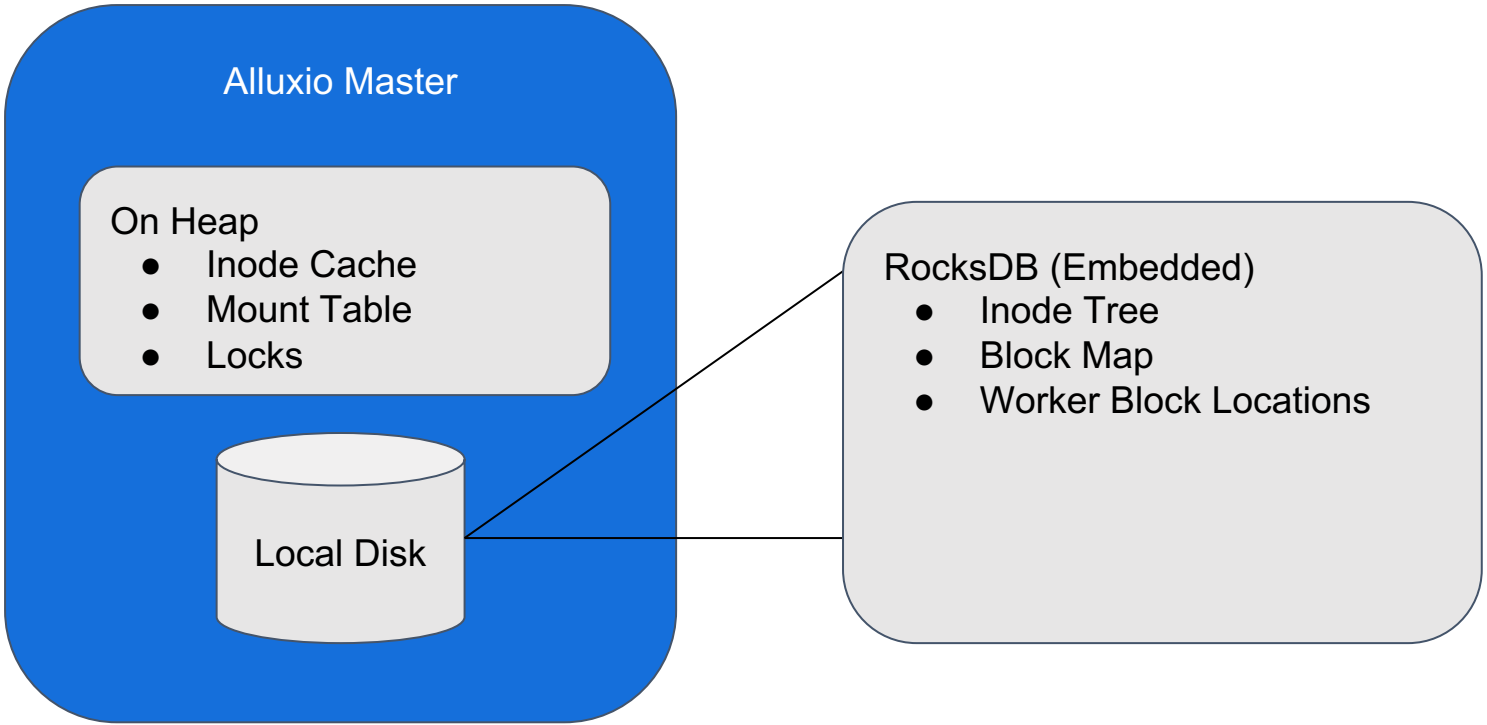
RocksDB

- https://rocksdb.org/
- RocksDB is an embeddable persistent key-value store for fast storage

Tiered Metadata Storage

- Uses an embedded RocksDB to store inode tree
  - Solves the storage heap space problem
- Developed new data structures to optimize for storage in RocksDB
- Internal cache used to mitigate on-disk RocksDB performance
  - Solves the serving latency problem
  - Performance is comparable to previous on-heap implementation
- [In-Progress] Use tiered recovery to incrementally make the namespace available on cold start
  - Solves the recovery problem

Tiered Metadata Storage => 1 Billion Files



Working with RocksDB

- Abstract the metadata storage layer
- Redesign the data structure representation of the Filesystem Tree
  - Each inode is represented by a numerical ID
  - Edge table maps <ID,childname> to <ID of child> **Ex: <1foo, 2>**
  - Inode table maps <ID> to <Metadata blob of inode> **Ex: <2, proto>**
- Two table solution provides good performance for common operations
  - One lookup for listing by using prefix scan
  - Path depth lookups for tree traversal
  - Constant number of inserts for updates/deletes/creates

Example RocksDB Operations

- To create a file, /s3/data/june.txt:
  - Look up **<rootID, s3>** in the **edge table** to get **<s3ID>**
  - Look up **<s3ID, data>** in the **edge table** to get **<dataID>**
  - Look up **<dataID>** in the **inode table** to get **<dataID metadata>**
  - Update **<dataID, dataID metadata>** in the **inode table**
  - Put **<june.txtID, june.txt metadata>** in the **inode table**
  - Put **<dataId, june.txt>** in the **edge table**
- To list children of /:
  - Prefix lookup of **<rootId>** in the **edge table** to get all **<childID>**s
  - Look up each **<childID>** in the **inode table** to get **<child metadata>**

Effects of the Inode Cache

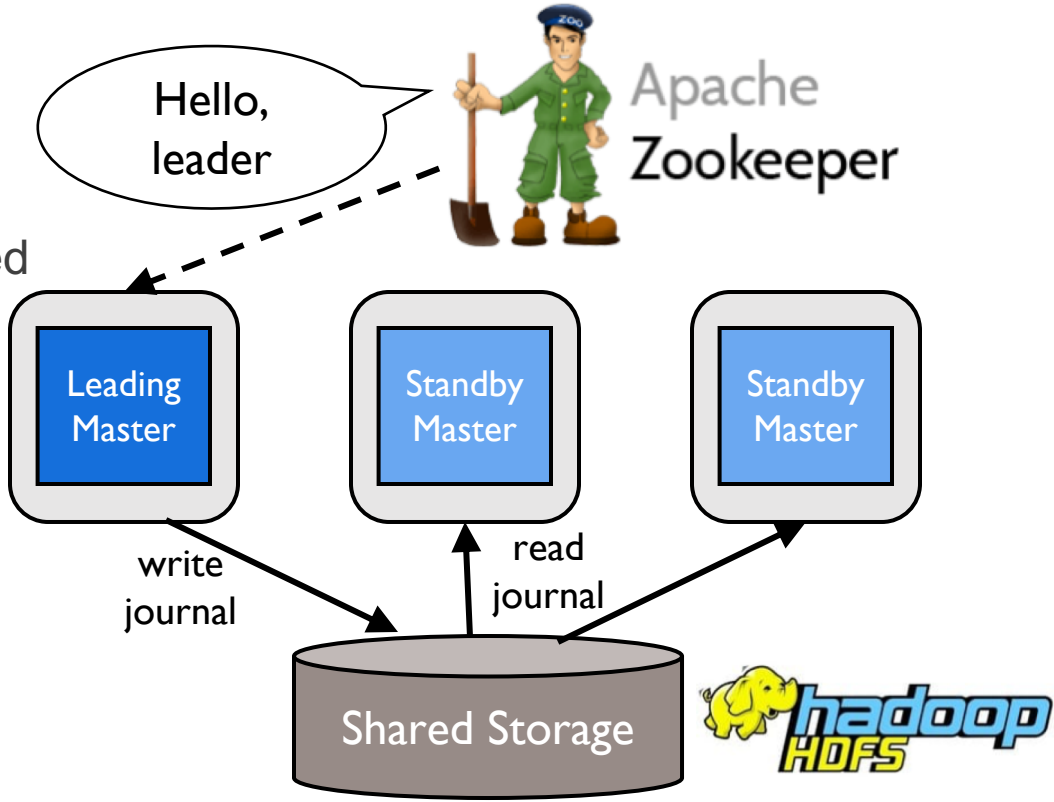
- Generally can store up to 10M inodes
- Caching top levels of the Filesystem Tree greatly speeds up read performance
  - 20-50% performance loss when addressing a filesystem tree that does not mostly fit into memory
- Writes can be buffered in the cache and are asynchronously flushed to RocksDB
- No requirement for durability - that is handled by the journal

Self-Managed Quorum for Leader Election and Journal Fault Tolerance Using Raft

ALLUXIO

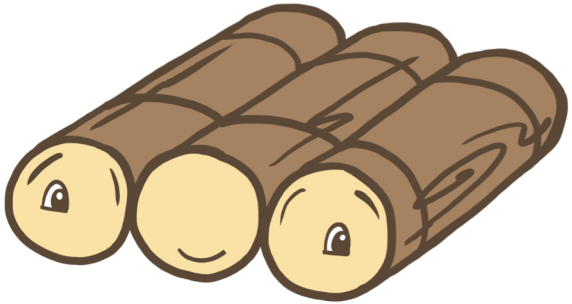
Alluxio 1.x HA Relies on ZK/HDFS

- **Running Alluxio in HA**
  - Zookeeper: Serve and elect the leader master for HA
  - HDFS: Journal Storage shared among masters
- **Problems**
  - Limited choice of journal storage
    - local, streaming writes
  - Hard to debug/recover on service outrage
  - Hard to maintain



RAFT

- https://raft.github.io/
- Raft is a consensus algorithm that is designed to be easy to understand. It's equivalent to Paxos in fault-tolerance and performance.
- Implemented by https://github.com/atomix/copycat

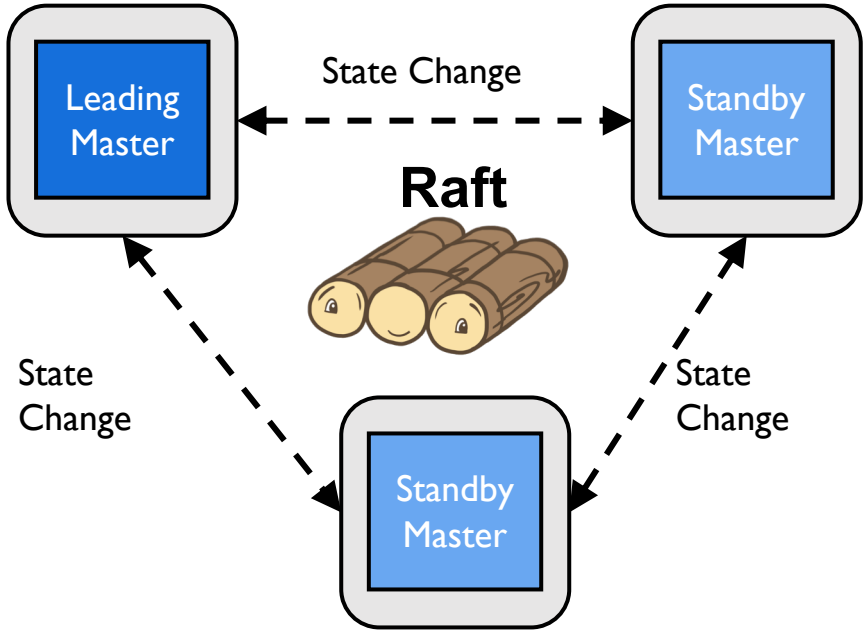


Built-in Fault Tolerance

- Alluxio Masters are run as a quorum for journal fault tolerance
  - Metadata can be recovered, solving the durability problem
  - This was previously done utilizing an external fault tolerance storage
- Alluxio Masters leverage the same quorum to elect a leader
  - Enables hot standbys for rapid recovery in case of single node failure

A New HA Mode with Self-managed Services

- **Consensus achieved internally**
  - Leading masters commits state change
- **Benefits**
  - Local disk for journal
- **Challenges**
  - Performance tuning

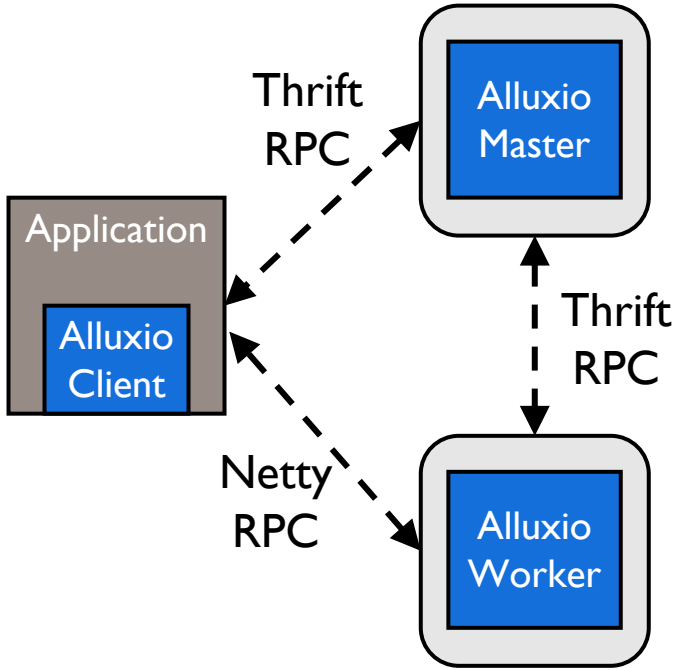


High-Performance and Unified RPC Framework Using gRPC

ALLUXIO

RPC System in Alluxio 1.x

- **Master RPC using Thrift**
  - Filesystem metadata operations
- **Worker RPC using Netty**
  - Data operations
- **Problems**
  - Hard to maintain and extend two systems
  - Thrift is not maintained, no streaming RPC support

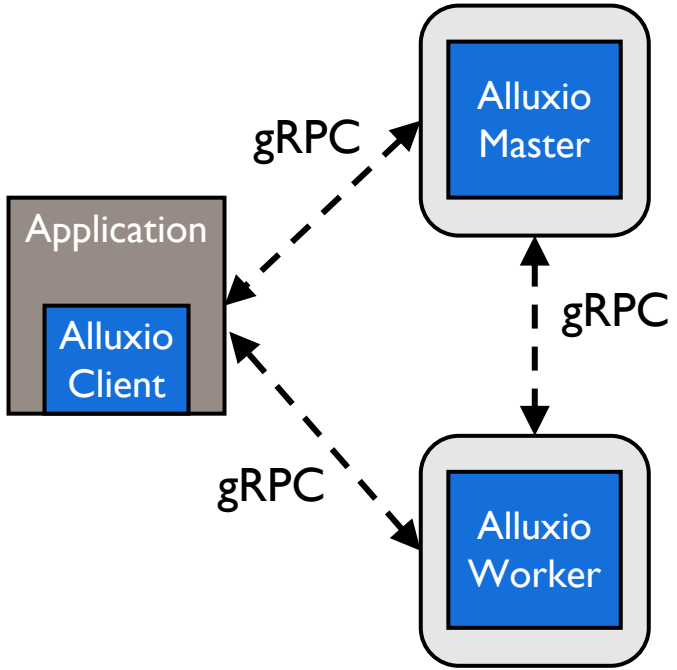


gRPC

- https://grpc.io/
- gRPC is a modern open source high performance RPC framework that can run in any environment
- Works well with Protobuf for serialization

Unified RPC Framework in Alluxio 2.0

- **Unify all RPC interfaces using gRPC**
- **Benefits**
  - Streaming I/O
  - Protobuf everywhere
  - Well maintained & documented
- **Challenges**
  - Performance tuning



gRPC Transport Layer

- Connection multiplexing to reduce the number of connections from # of application threads to # of applications
  - Solves the connection life cycle management problem
- Threading model enables the master to serve concurrent requests at scale
  - Solves the high load problem
- High metadata throughput needs to be matched with efficient IO
  - Consolidated Thrift (Metadata) and Netty (IO)

Check out this blog for more details: https://www.alluxio.com/blog/moving-from-apache-thrift-to-grpc-a-perspective-from-alluxio

Questions?

Alluxio Website - https://www.alluxio.io

Alluxio Community Slack Channel - https://www.alluxio.io/slack

Alluxio Office Hours & Webinars - https://www.alluxio.io/events

ALLUXIO