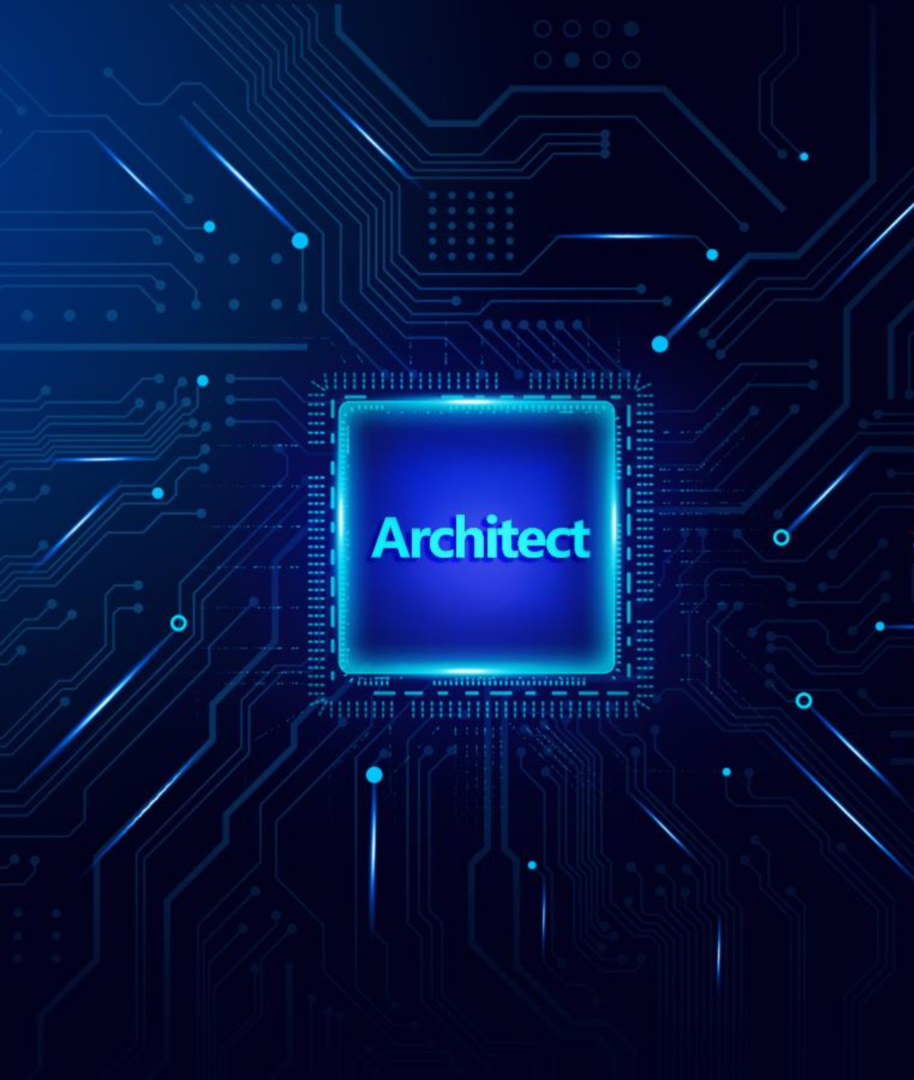


SACC

2022 中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2022

激发架构性能 点亮业务活力

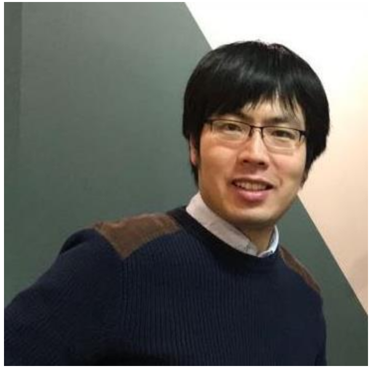
云上会议 网络直播 | 2022年10月27-29日

IT168.com ChinaUnix.net ITPUB

Building a High-performance and Scalable Metadata Service for Distributed File System

Alluxio 创始成员&开源社区副总裁 范斌

About Me



Bin Fan (https://www.linkedin.com/in/bin-fan/)

- Founding Engineer, VP Open Source @ Alluxio
- Alluxio PMC Co-Chair, Presto TSC/committer
- Email: binfan@alluxio.com
- PhD in CS @ Carnegie Mellon University

知 apc999

Alluxio Overview

- Originally a research project (Tachyon) in UC Berkeley AMPLab led by by-then PHD student Haoyuan Li (Alluxio founder CEO)
- Backed by top VCs (e.g., Andreessen Horowitz) with $70M raised in total, Series C ($50M) announced in 2021
- Deployed in production at large scale in Facebook, Uber, Microsoft, Tencent, Tiktok and etc
- More than 1200 Contributors on Github. In 2021, more than 40% commits in Github were contributed by the community users
- The 9th most critical Java-based Open-Source projects on Github by Google/OpenSSF[1]

[1] Google Comes Up With A Metric For Gauging Critical Open-Source Projects

Companies Using Alluxio

TECHNOLOGY

INTERNET

facebook · Baidu 百度 · Alibaba.com · Tencent 腾讯

amazon · JD.COM 京东 · Rakuten · waze

PUBLIC CLOUD PROVIDERS

aws · Google Cloud · Alibaba Cloud · Microsoft Azure · Tencent Cloud · Baidu Cloud

GENERAL

SAMSUNG · Aunalytics rethink data · Microsoft · Adobe

E-COMMERCE

唯品会 VIPSHOP.COM 一家专门做特卖的网站 · 苏宁易购 suning.com · Shopee · Walmart · Myntra

OTHERS

SAMSUNG SAMSUNG SDS · RYTE

TalkingData · CAESARS ENTERTAINMENT

ROBLOX · 链家

yahoo! · NetEase Games

bazaarvoice: · nielsen

GUARDANT HEALTH · Lenovo

eSENTIRE · 网易严选

PERCEPTIN · Ctrip

@WalmartLabs · DiDi

Sogou 搜狗 · esri

Lucidworks · MOMO

CUELOGIC · QINIU 七牛云存储

去哪儿? Qunar.Com · walkme Digital Adoption Platform

ORACLE

FINANCIAL SERVICES

DBS · ING

BARCLAYS · HUATAI SECURITIES

PayPal · WELLS FARGO

TELCO & MEDIA

COMCAST · 虎牙直播 huya.com

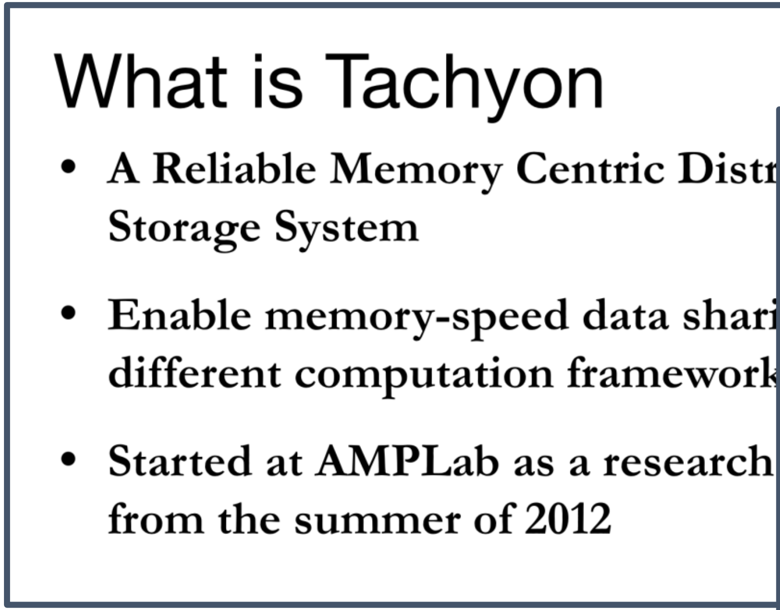
HUAWEI · swisscom

中国移动 China Mobile · China unicom 中国联通

LEARN MORE

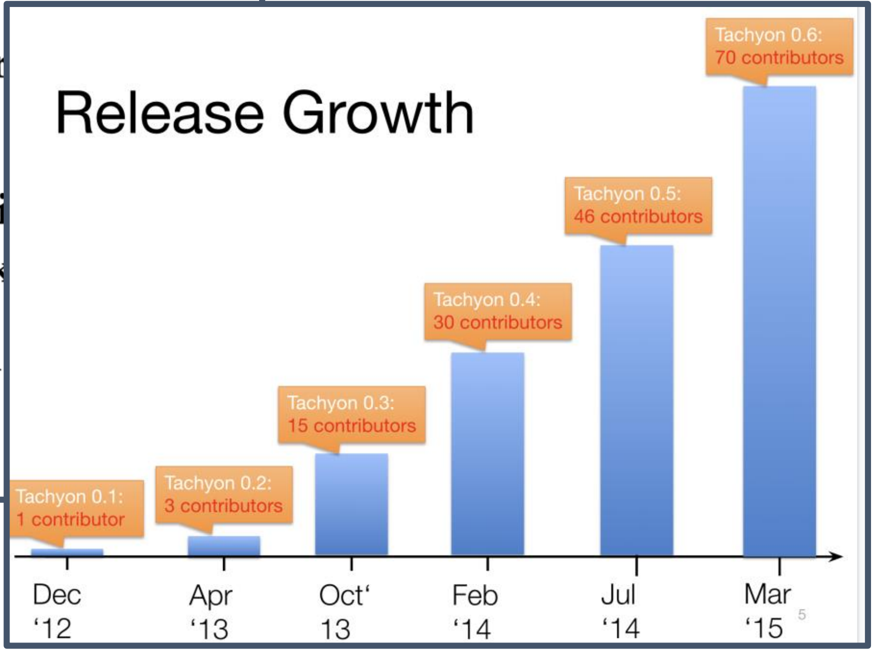
7 Years Ago

Alluxio (Tachyon) in 2015



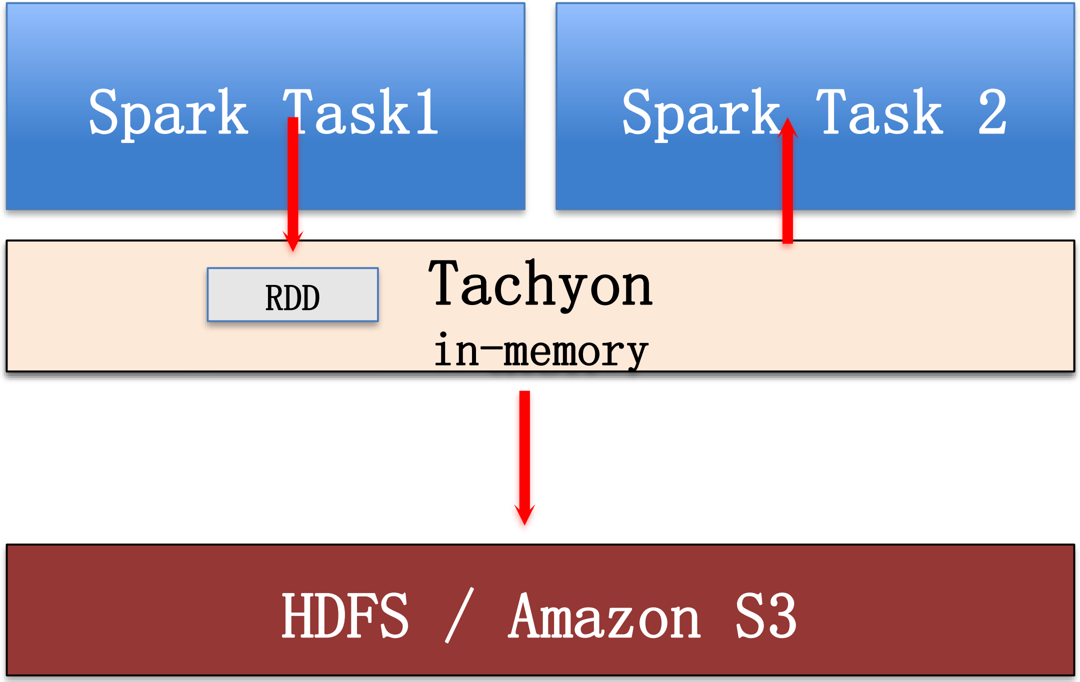
What is Tachyon

- A Reliable Memory Centric Distr
Storage System
- Enable memory-speed data shari
different computation framework
- Started at AMPLab as a research
from the summer of 2012

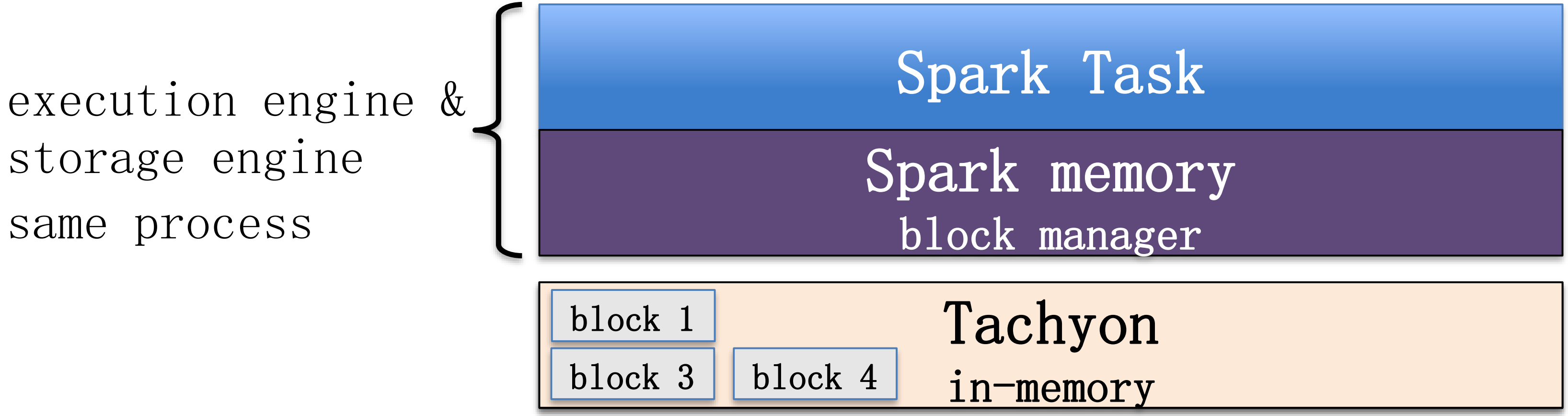


Release Growth

Alluxio(Tachyon) in 2015
Enable Data Sharing Among Spark Jobs



Alluxio(Tachyon) in 2015
Fast Checkpoint for job reliability



Today

What’ s Different

Topology

- On-prem Hadoop → Cloud-native, Multi- or Hybrid-cloud, Multi-datacenter

Computation

- MR/Spark → Spark, Presto, Hive, Tensorflow, Pytorch ....
- More mature frameworks (less frequent OOM etc)

Data access pattern

- Sequential-read (e.g., scanning) on unstructured files → Ad-hoc read into structured/columnar data
- Hundred to thousand of big files → millions of small files

The Evolution from Hadoop to Cloud-native Era

Data Storage

- On-prem & colocated HDFS → S3 !!! and other object stores (possibly across regions like us-east & us-west), and legacy on-prem HDFS in service

Resource/Job Orchestration

- YARN → K8s
    - Lost focus on data locality

Strong Market Demand For Simplification

UNIFICATION OF DATA LAKES

Serve analytics & AI from multiple data locations

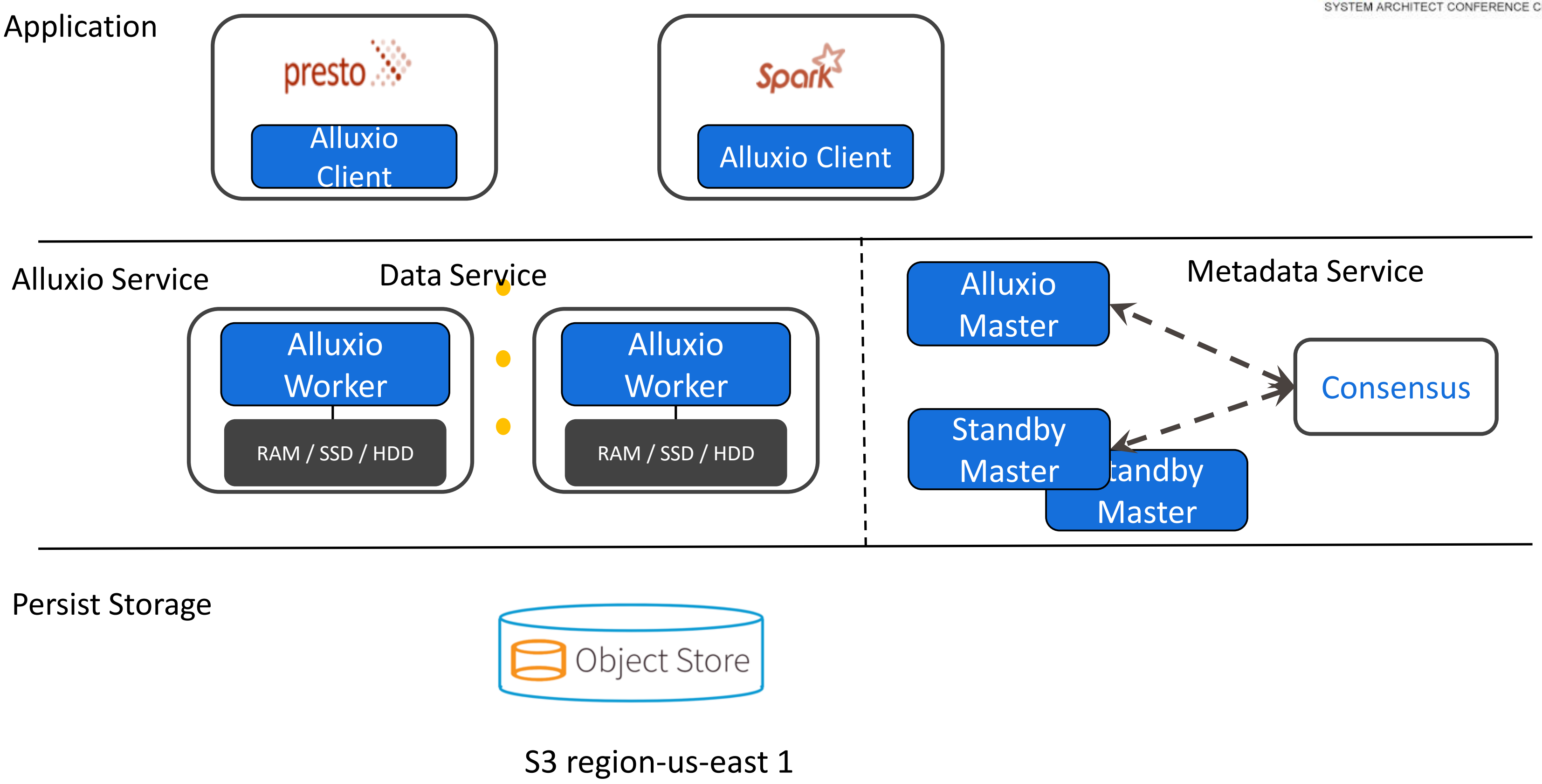
EFFICIENT ACCESS & DATA MANAGEMENT

Acceleration & auto-tiering of remote data sources

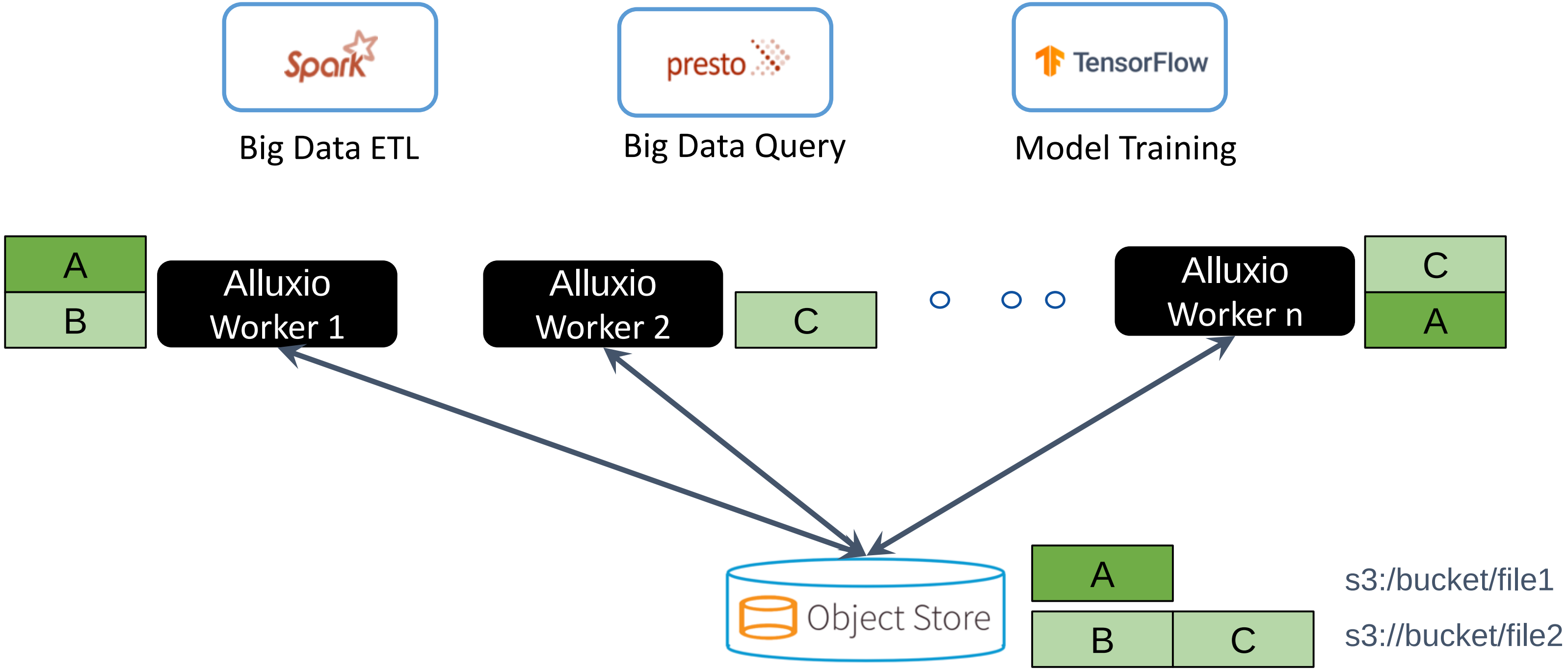
ENVIRONMENT AGNOSTICITY

Agility across regions for private, hybrid or multi-cloud

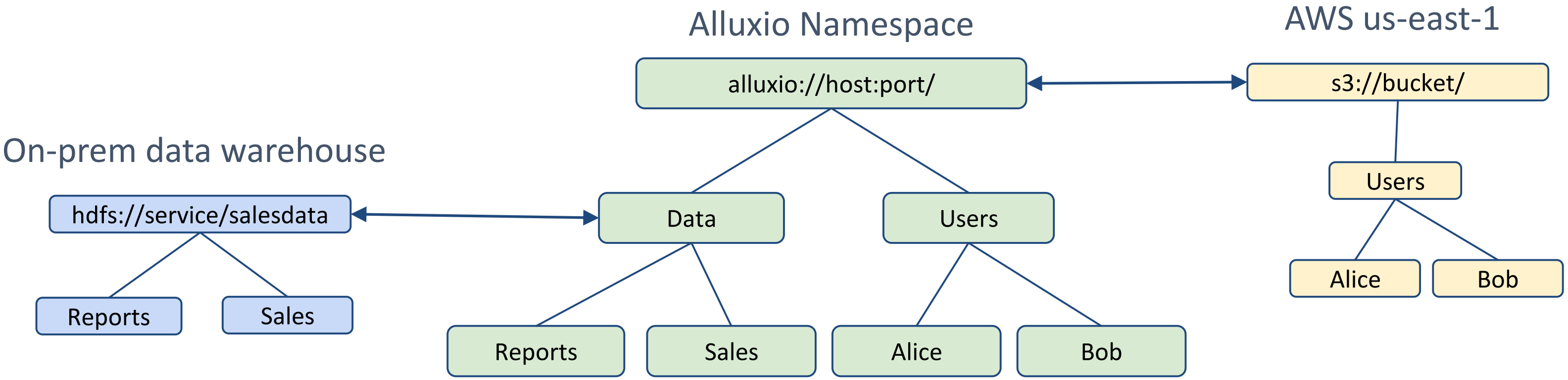
Architecture



Core Feature 1: Distributed Caching



Core Feature 2: Filesystem Namespace Virtualization



- Alluxio can be viewed as a logical file system
  - Multiple different storage service can be mounted into same logical Alluxio namespace
- An Alluxio path is backed by an persistent storage address
  - alluxio://Data/Sales <-> hdfs://service/salesdata/Sales

Challenges to Build Scalable Metadata Services

What is File System Metadata

- Data structure of the Filesystem Hierarchy: Often an Inode tree to represent parent dir, children, permission bits, ower/group, modification time
    - Each node on this inode tree corresponding to one file or directory
    - Commonly seen in all file systems
    - Can include mounts of other file systems in Alluxio and the size of the tree can be very large!
- Sub-file blocks information (block ID -> workers)
    - Index for a distributed system to point to the data server

Factors w.r.t. Design a Scale Metadata Service

- # of Alluxio Servers in a cluster
  - Heartbeat:
    - node -> master
  - Load balancing
    - Workload skew
- # of concurrent clients
- # of files/dirs in this logical file system
- Throughput of metadata RPCs
  - Read ops
  - Write ops
- Speed to fail over to other stand-by masters (avoid Single node of failure)

Single Master Scalability

How to Store File System Metadata

Federating Multiple Storage

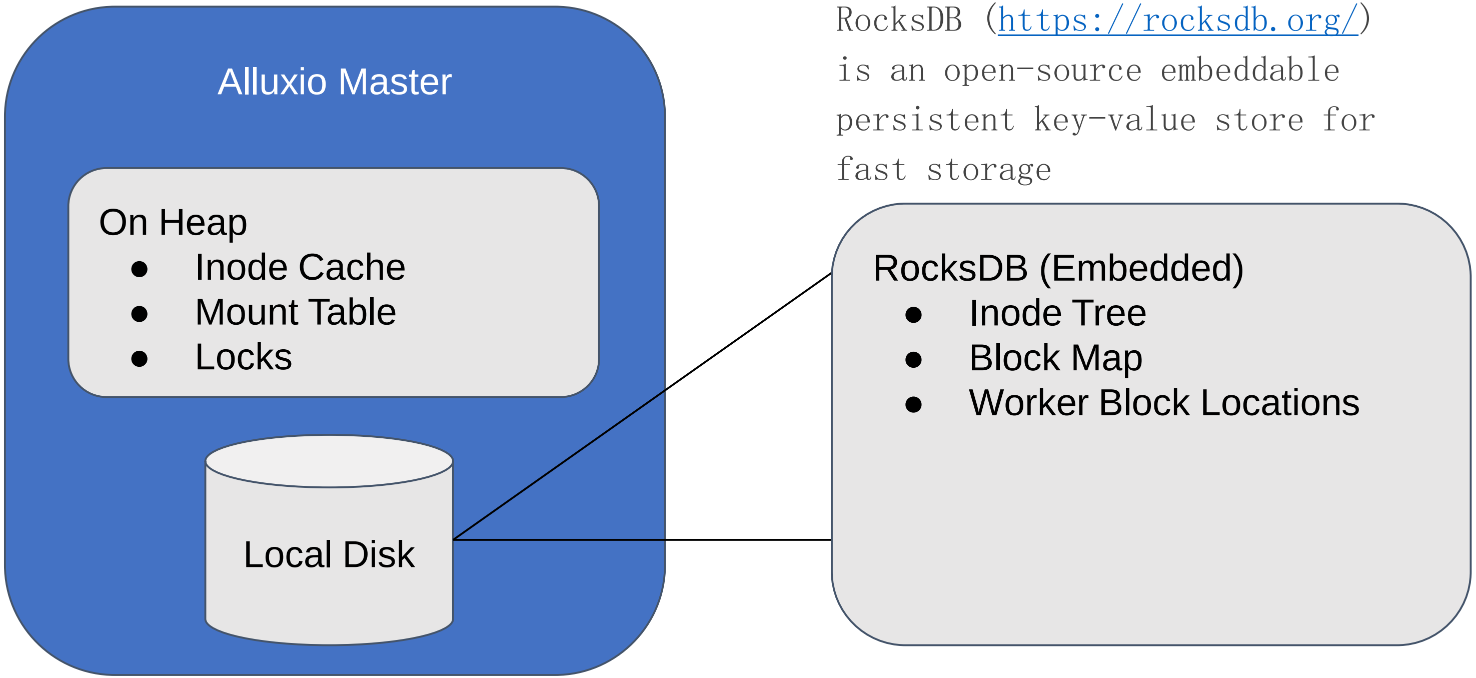
=> We need to handle a “logical file system” multiple times bigger

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

Off-Heap Metadata Storage => 1 Billion Files

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



- Inode Cache
- Mount Table
- Locks

- Inode Tree
- Block Map
- Worker Block Locations

Other 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
- 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

High Availability

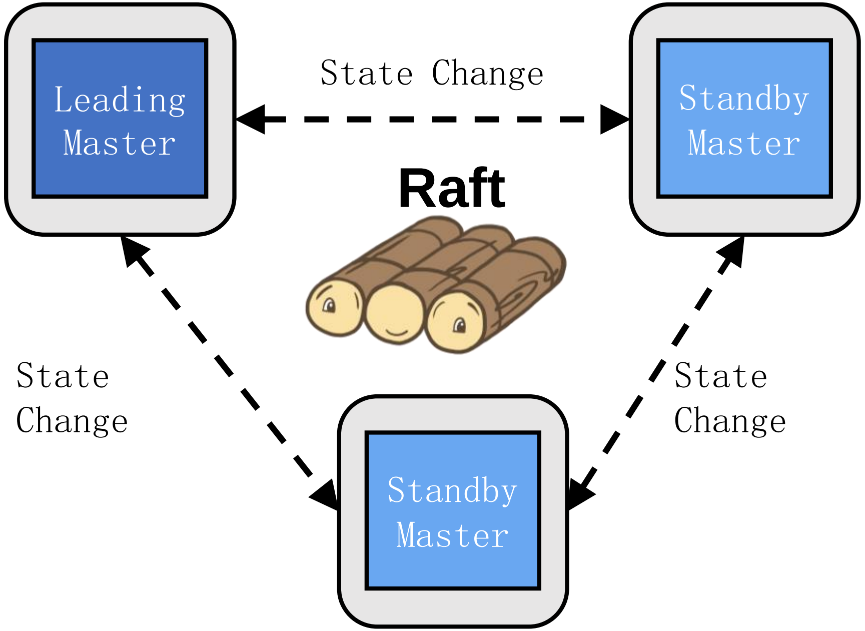
Built-in Fault Tolerance

- Alluxio cluster can recover from restarts, and avoid single-point of failure
  - File system status must be able to be recovered
  - This was previously done utilizing an external fault tolerance storage
- Our approach: Self-Managed Quorum for Leader Election and Journal Fault Tolerance Using Raft
  - Raft is a consensus algorithm that is designed to be easy to understand. It's equivalent to Paxos in fault-tolerance and performance
  - Enables hot standbys for rapid recovery in case of single node failure

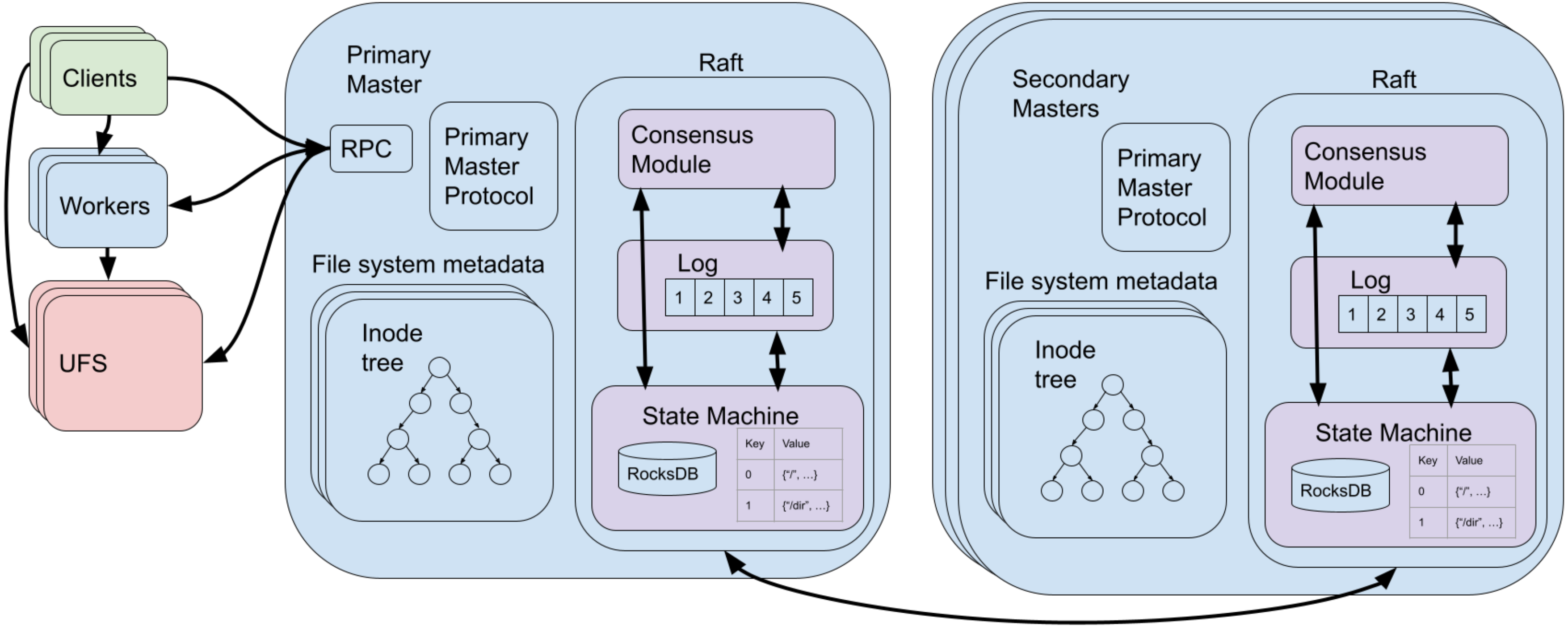
拓展阅读：知乎：漫话分布式系统共识协议: Paxos篇

Built-in Self-Managed Quorum-based Journal

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



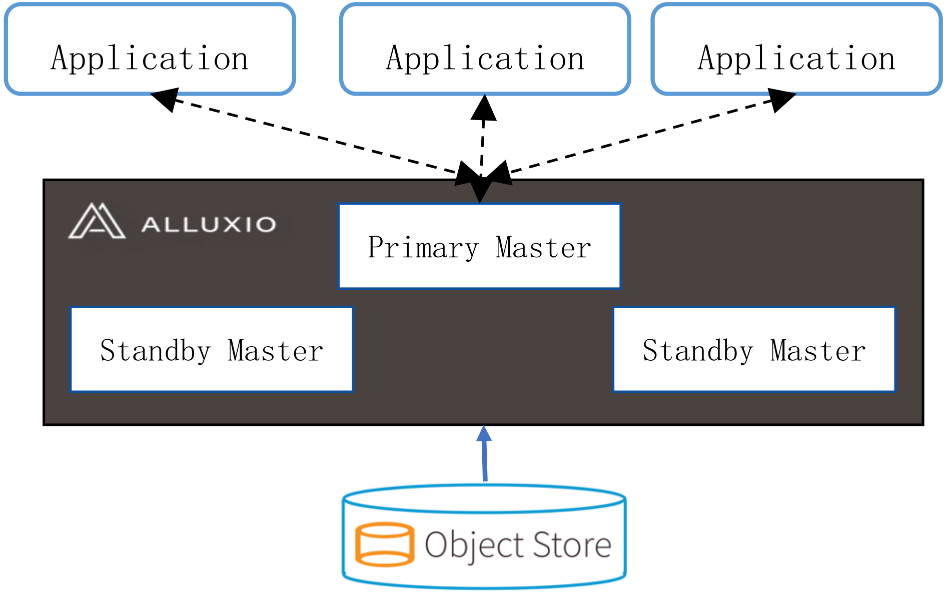
Alluxio + Raft architecture



Consistency

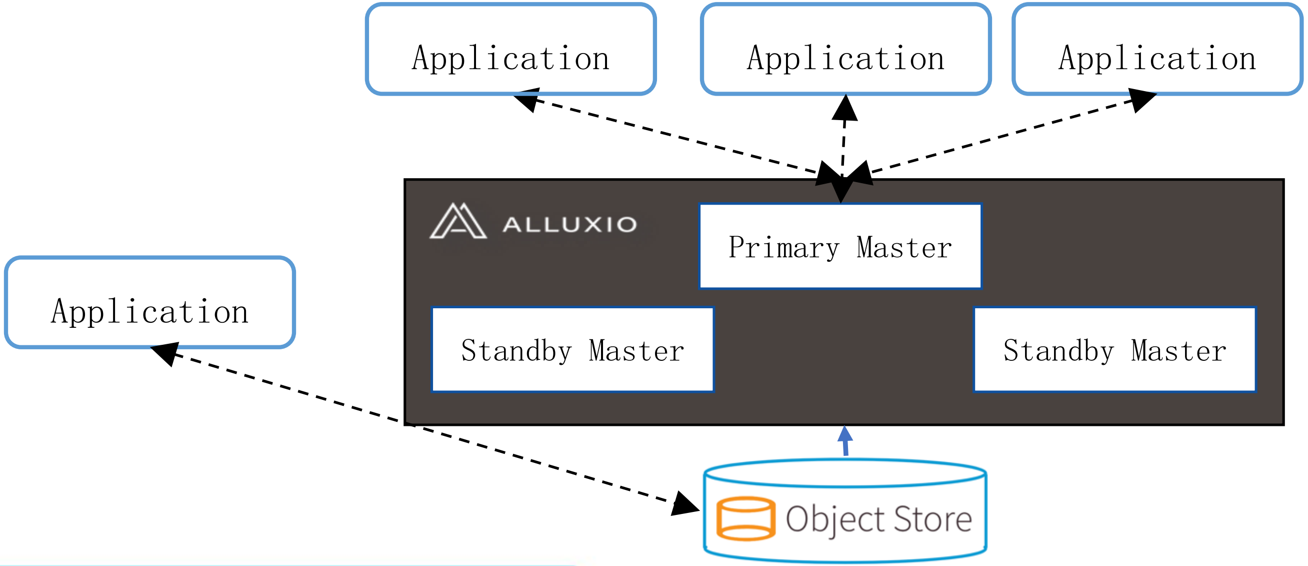
Consider Alluxio File System Alone

- If clients only query and modify Alluxio File System through Alluxio masters, the semantics is strongly consistent



Consider Alluxio File System + UFS

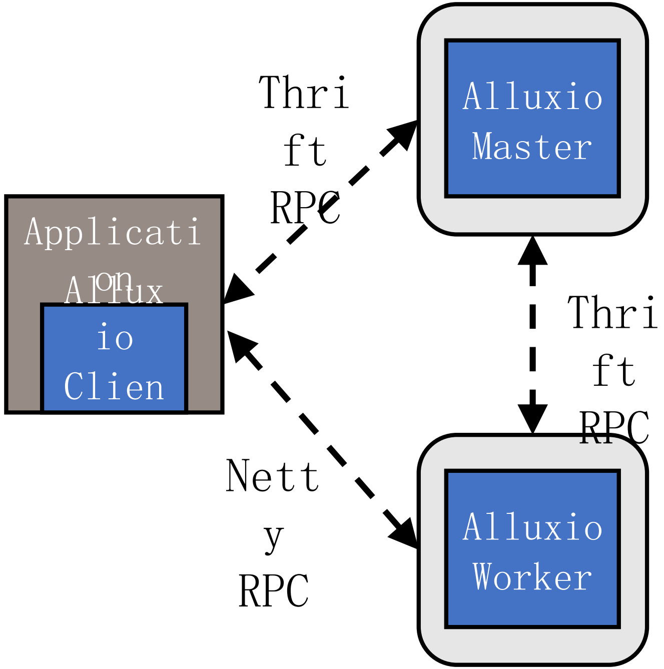
- When clients can modify UFS, Alluxio masters provide synchronization between Alluxio namespace and UFS



Serving Data

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

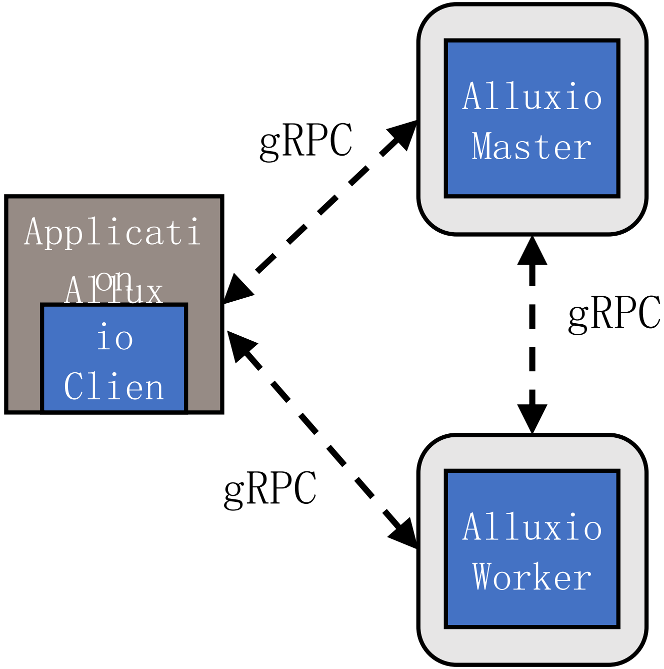


gRPC

- 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

Summary

Summary

- Designing & Implementing a distributed system is hard but also fun
- First you need to well understand the design requirements
- Consistency, Scalability, Reliability – We spent most of our time to fight for
- Do not reinvent the wheel, but also be cautious when introducing new building blocks

Join Slack channel

alluxio.io/slack

官方微信公众号

Slack官方账号

Alluxio小助手

THANKS