

# 数据加速让训练如虎添翼

## B站搭建高效AI平台


黎磊, AI Platform Lead, bilibili

倪子凡, Sr. SWE, bilibili

bilibili



黎磊

-  Azure-ed
-  bilibili-ing
- cloud-native, k8s, mlops, serverless, etc.
-   





倪子凡

-  Azure-ed
-  bilibili-ing
- Lovely Puppy, Da Vinci.
-   



bilibili EasyAI

1



Alluxio Meets  
EasyAI

2

Best  
Practice

3

Alluxio Perf

4





-  云原生AI平台
- 支持内部广泛业务
  - ✓ ADs
  - ✓ CV
  - ✓ NLP
  - ✓ Speech
  - ✓ e-commerce
  - ✓ etc.





model dev

model training

model storage

model serving

Volcano

VPA

Hawkeye

Cloud-Native Observability System of bilibili



Alluxio + Fluid

OSS

HDFS



## Master

- 管理全部元数据
- 监控各个Worker状态

## Worker

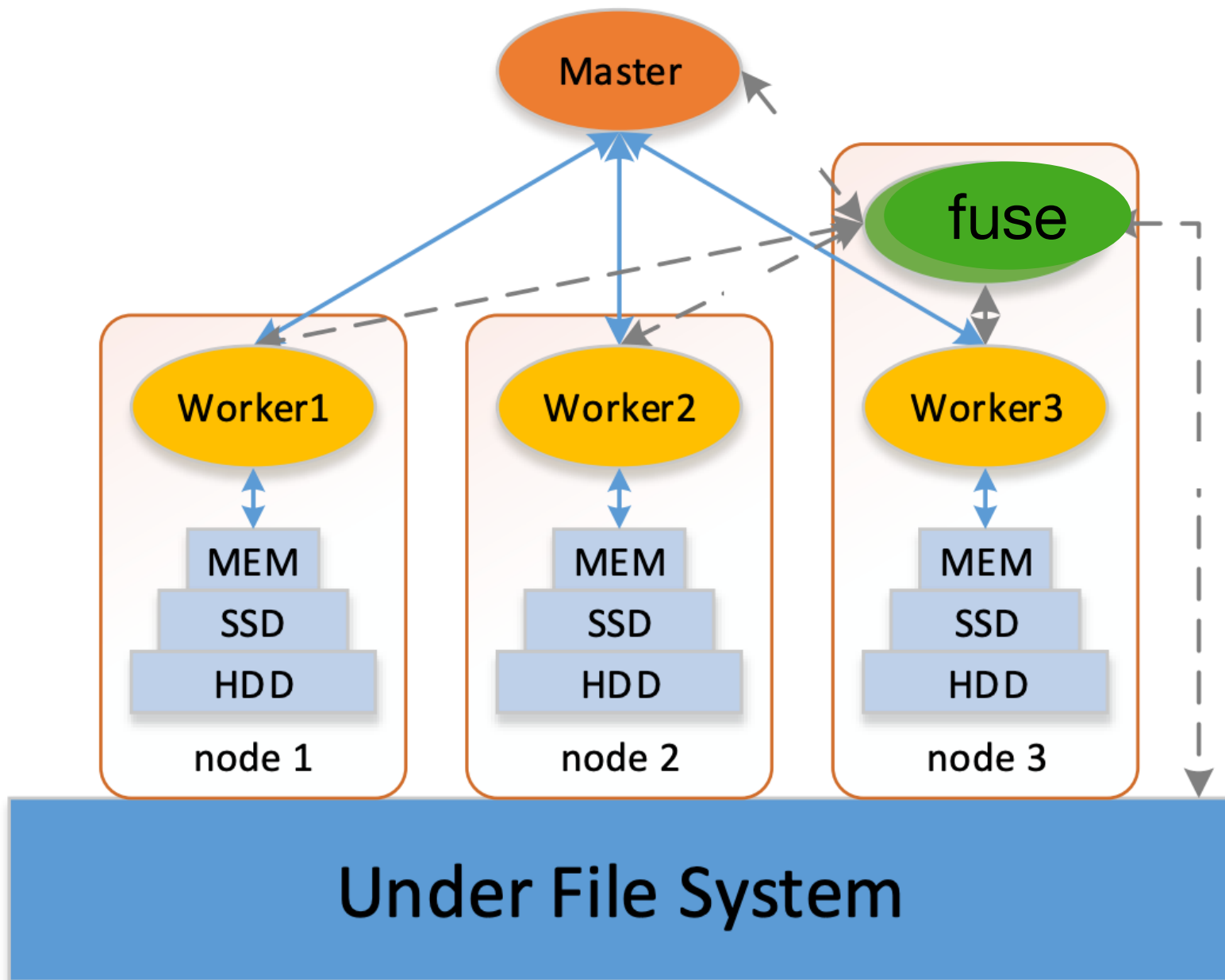
- 管理本地MEM、SSD和HDD

## Fuse

- 提供Posix接口
- 向Master和Worker发送请求

## Under File System

- 实际存储系统

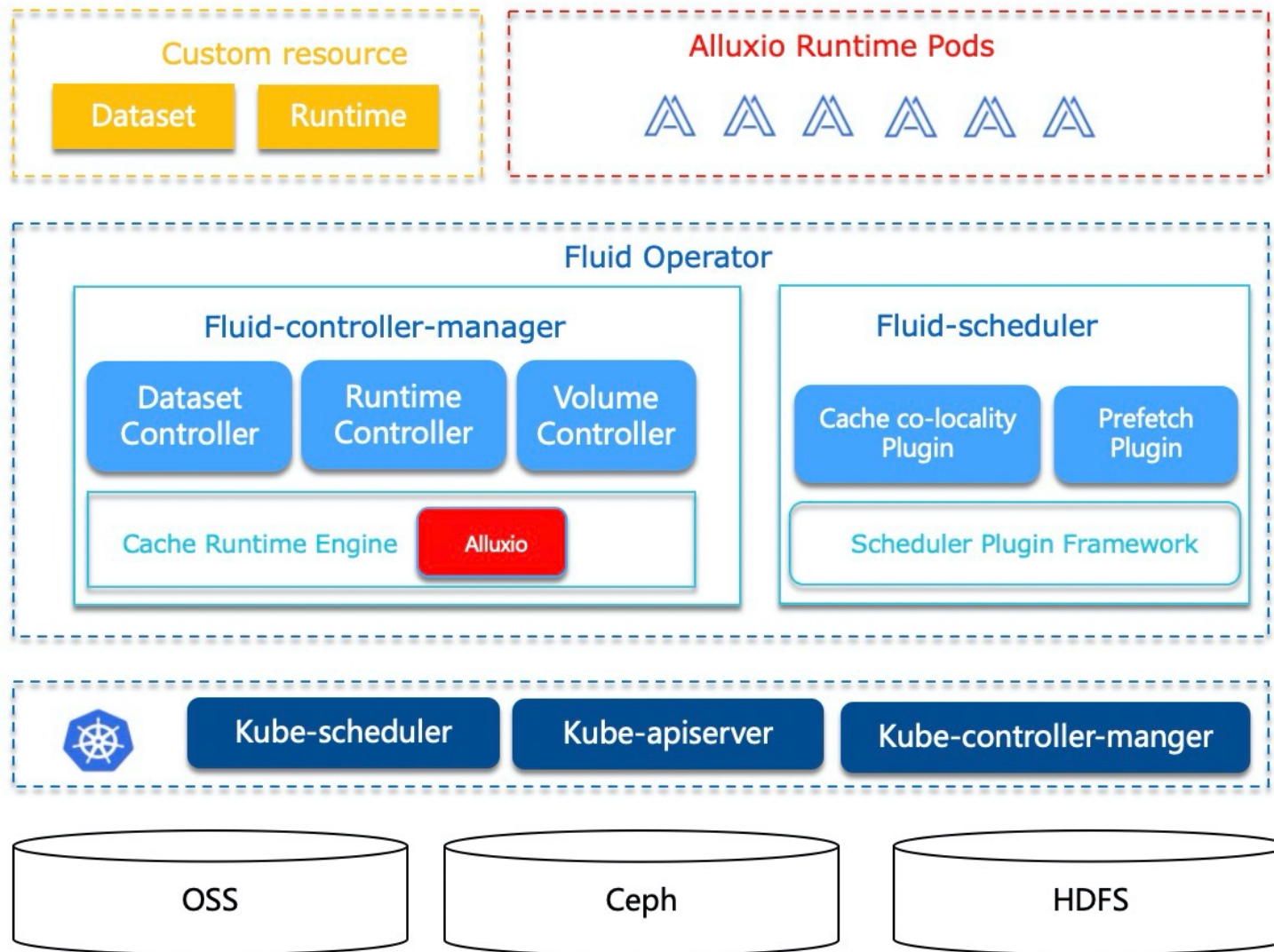




## Fluid

### k8s-native dataset Orchestrator and Accelerator

- Native Support for DataSet Abstraction
- Cloud Data Warming up and Accessing Acceleration
- Co-Orchestration for Data and Application





1 bilibili EasyAI

2 Alluxio Meets  
EasyAI

3 Best  
Practice

4 Alluxio Perf

1  
2



3

4





## Scenario 1 Container Crash

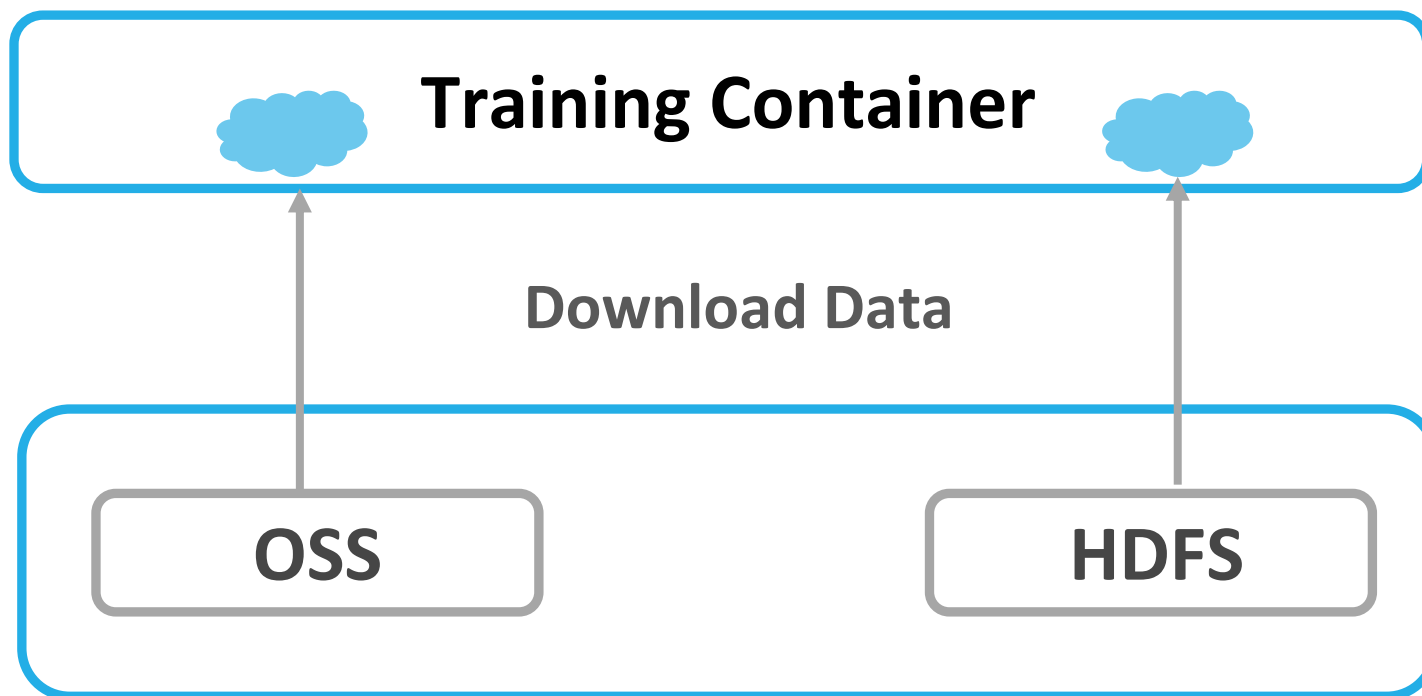
**Training Container**

OSS

HDFS

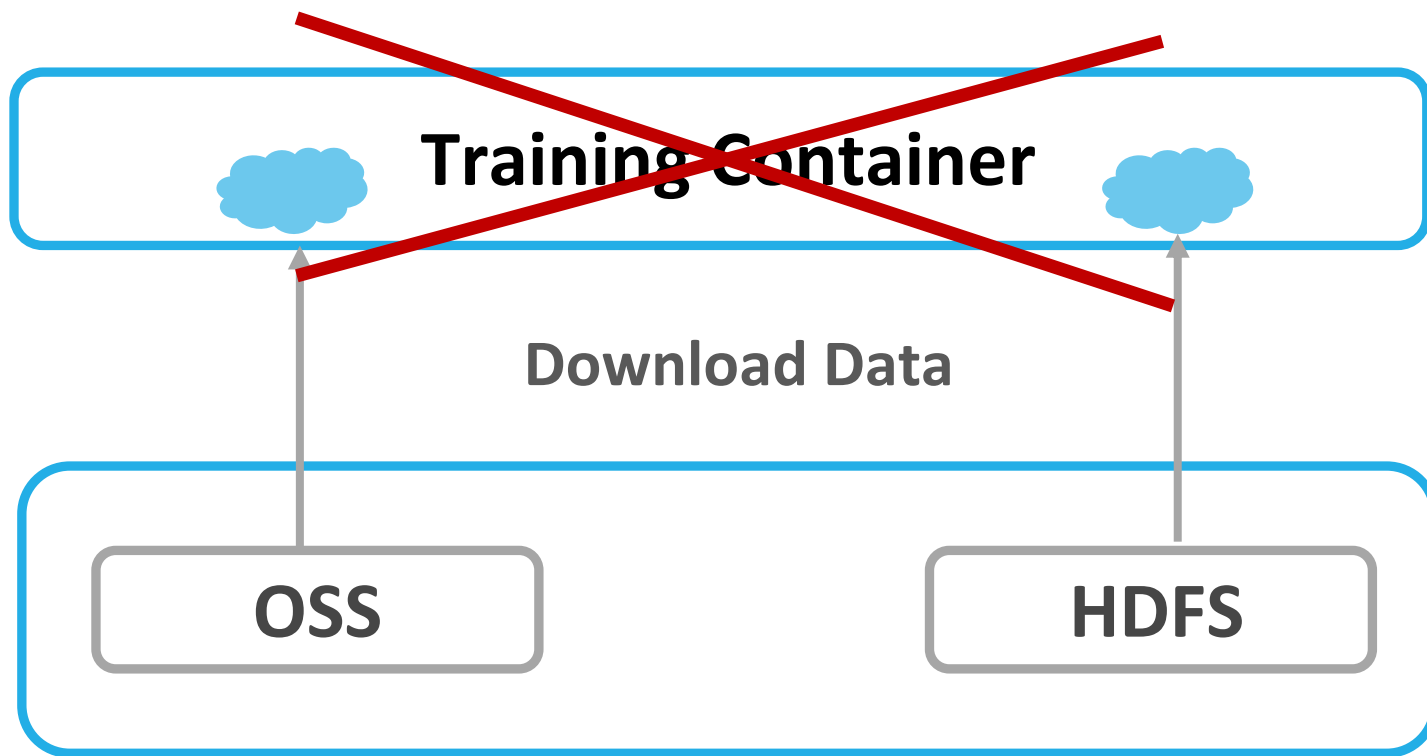


## Scenario 1 Container Crash





## Scenario 1 Container Crash



数据需要重新下载，浪费大量时间和**GPU**算力



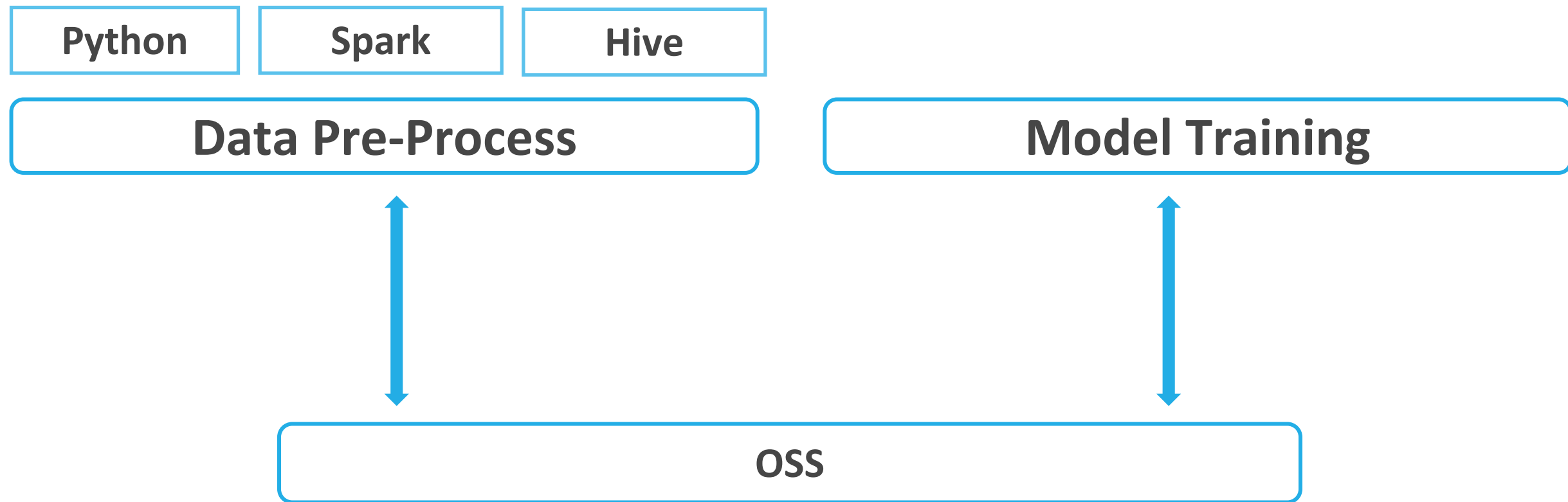
## Scenario 2 数据量超过单台机器限制

数据量超过单台机器容量，用户只能

- 重写数据访问逻辑，以数据流的形式访问数据库数据；
- 不同的数据库，需要写不同的API；
- 需要处理重试、断连逻辑；



## Scenario 3 数据预处理 + 模型训练





急需一个缓存

- 分布式
- 隐藏访问数据的复杂性



## 急需一个缓存

- 分布式
- 隐藏访问数据的复杂性

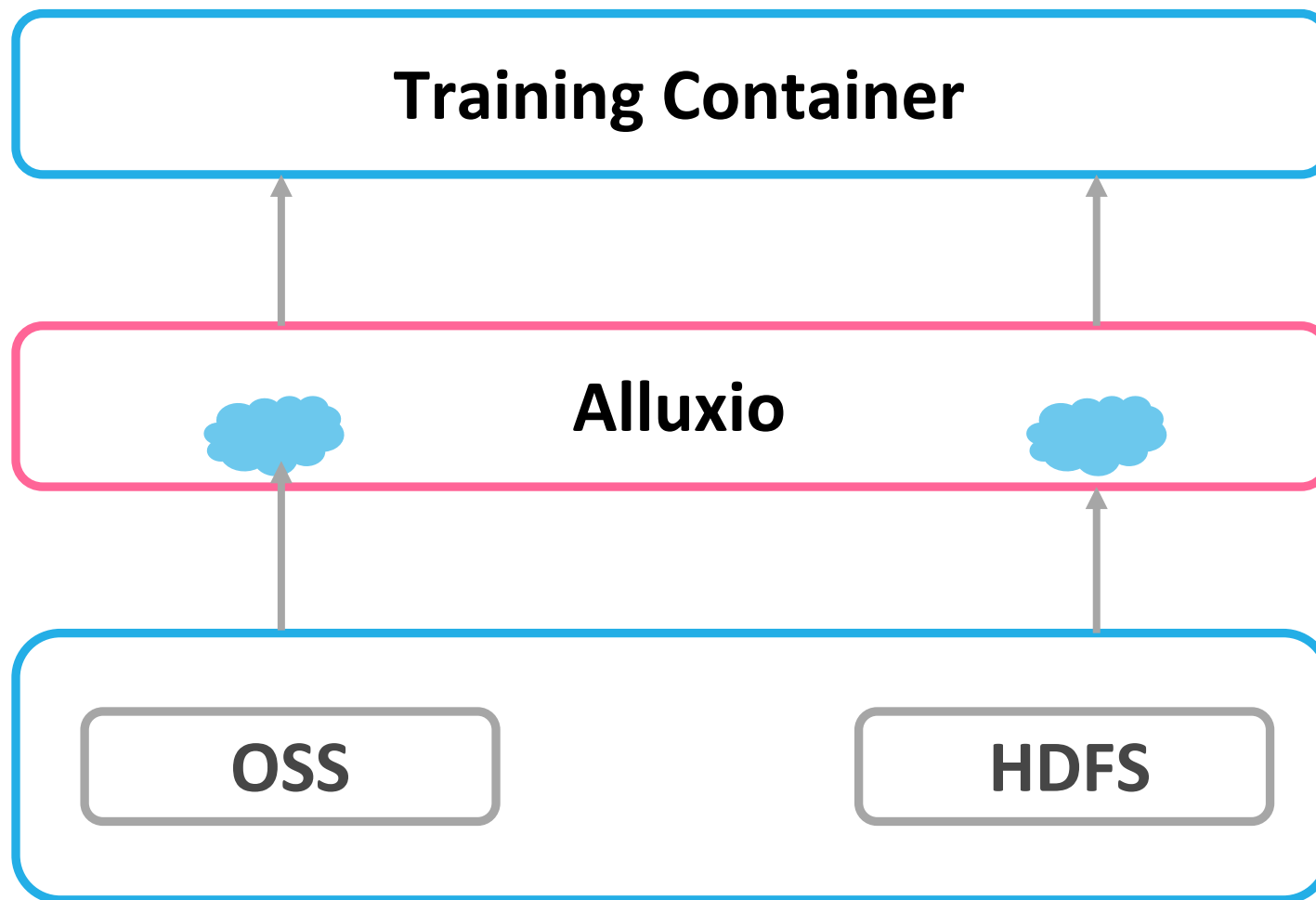
### Alluxio + Fluid

- 分布式缓存海量数据
- Fuse 使得读取数据变得简单(`os.open()`)



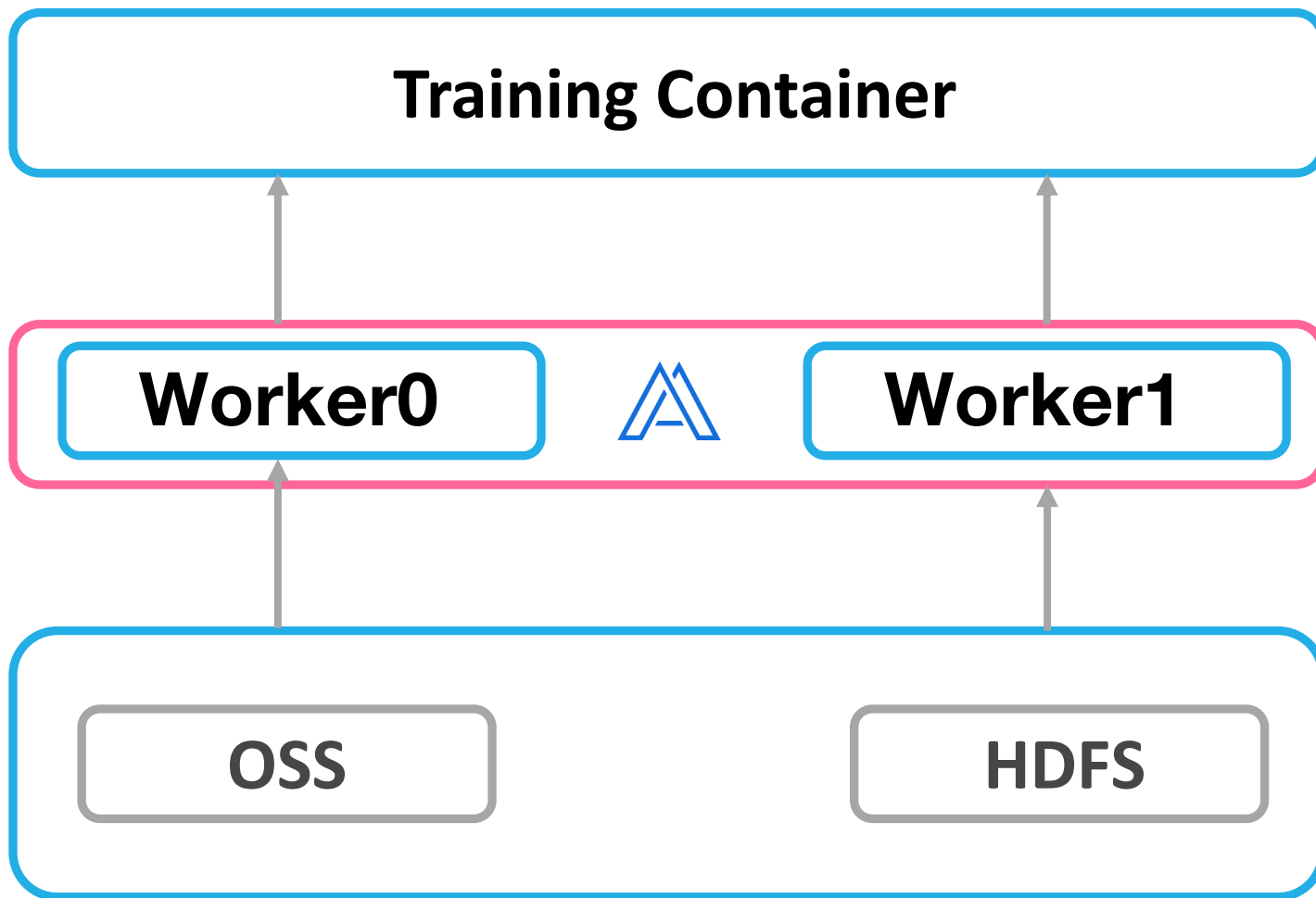


## Scenario 1 Container Crash



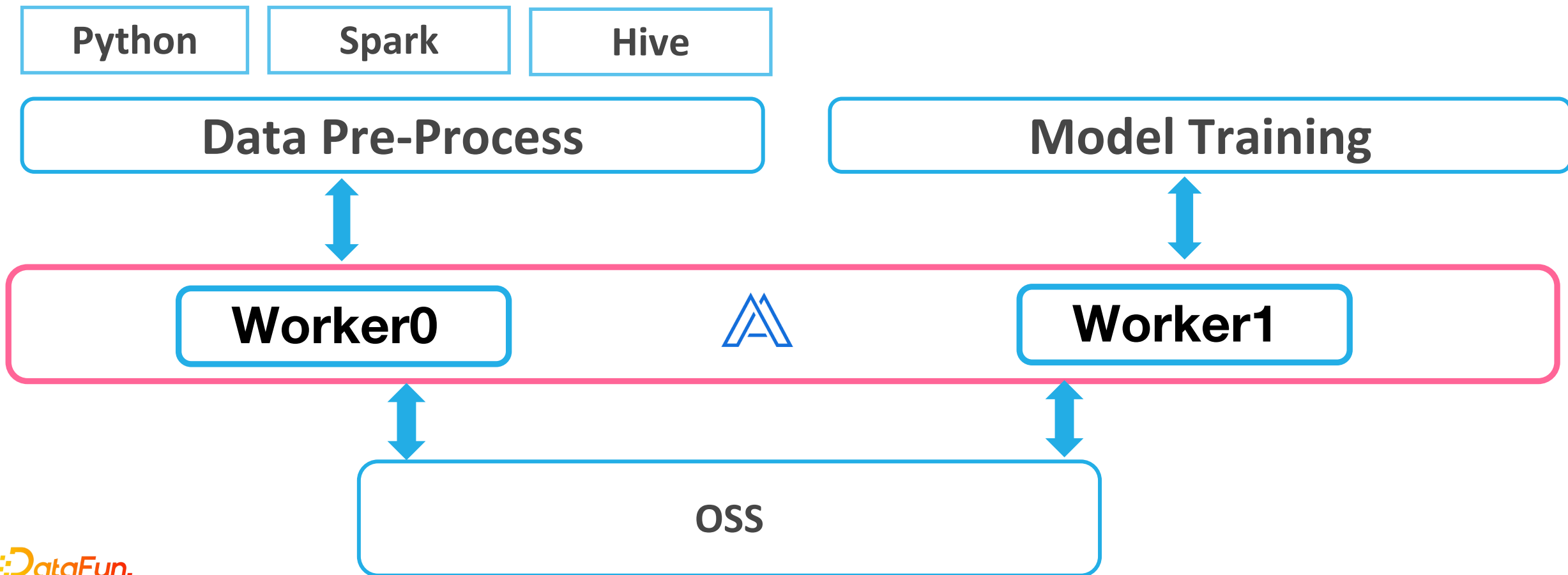


## Scenario 2 数据量超过单台机器限制





## Scenario 3 数据预处理 + 模型训练



bilibili EasyAI

1

Alluxio Meets  
EasyAI

2

Serverless  
fuse

3



Alluxio Perf

4





## 1. Why Serverless

Pod0

GPU0

GPU1



## 1. Why Serverless



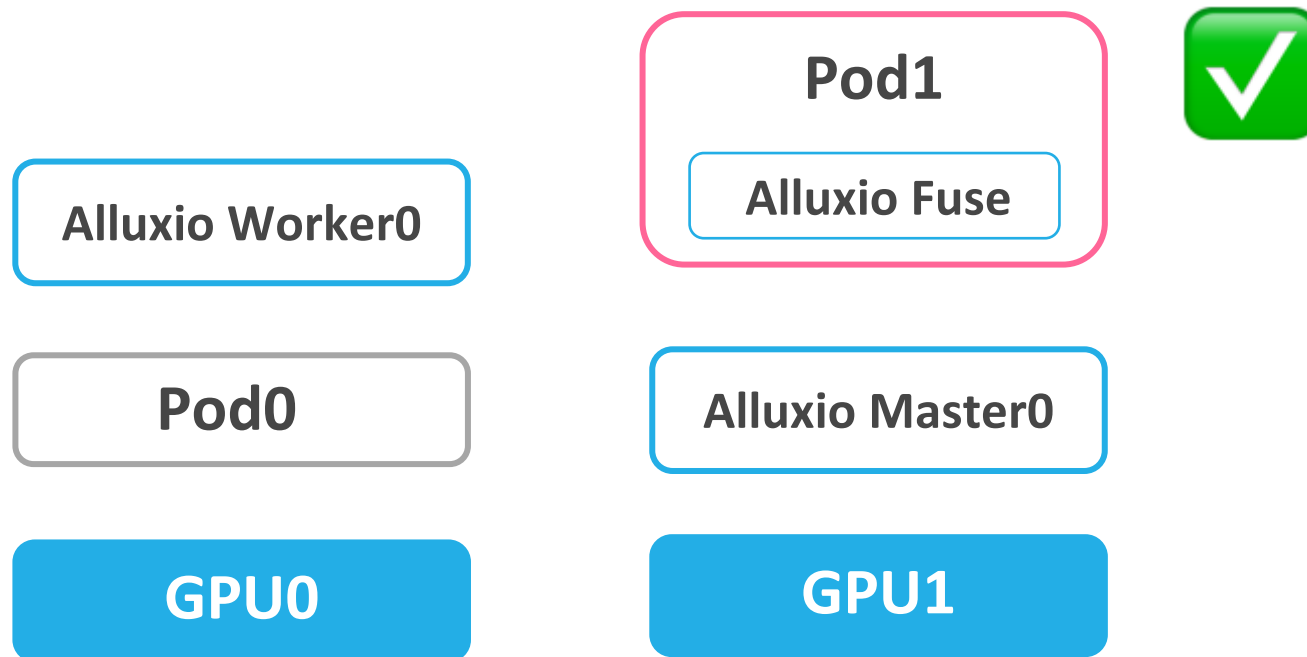


## 1. Why Serverless





## 2. Serverless Solution





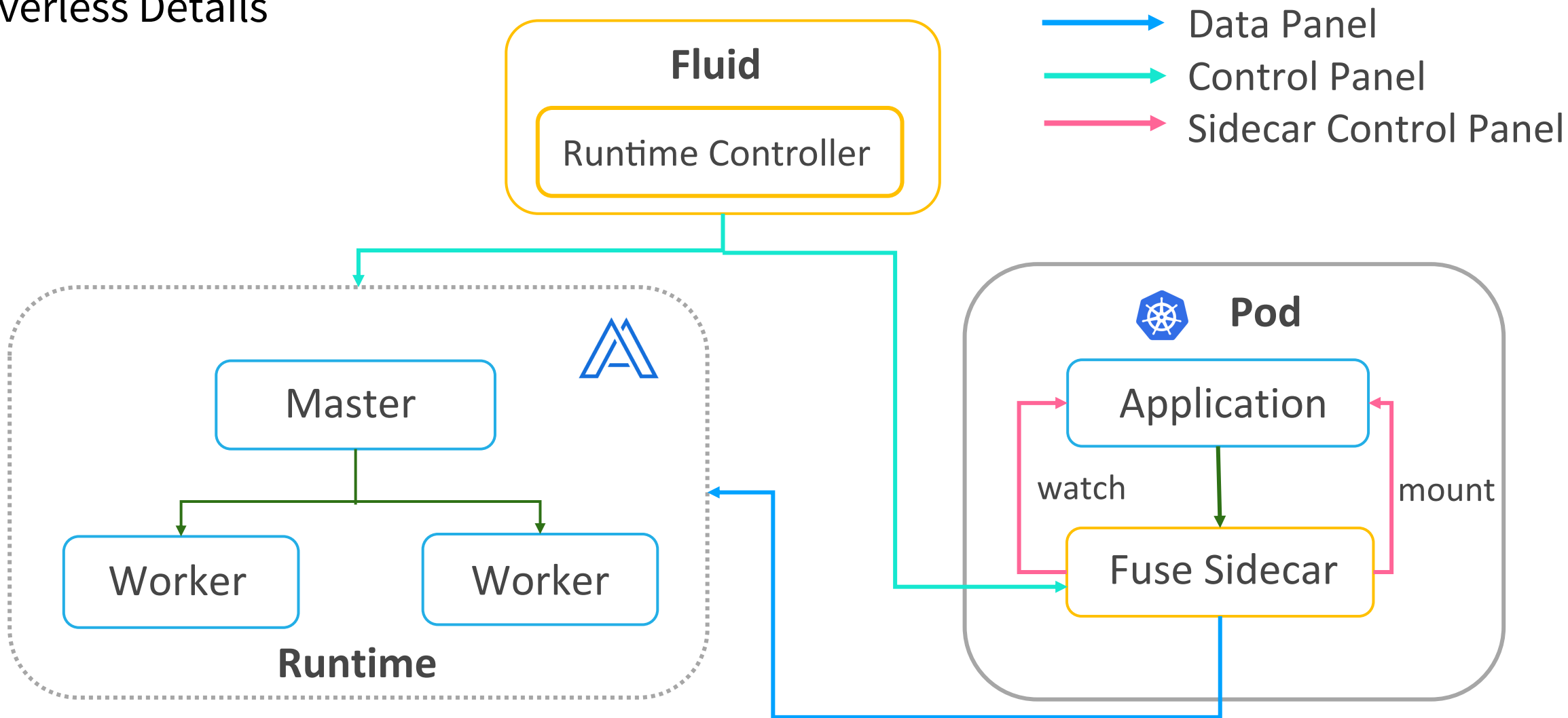


### 3. What is Serverless

- 与serverful相对；
- 按需启动应用；
- 更云原生的方式,与容器更紧密的结合；



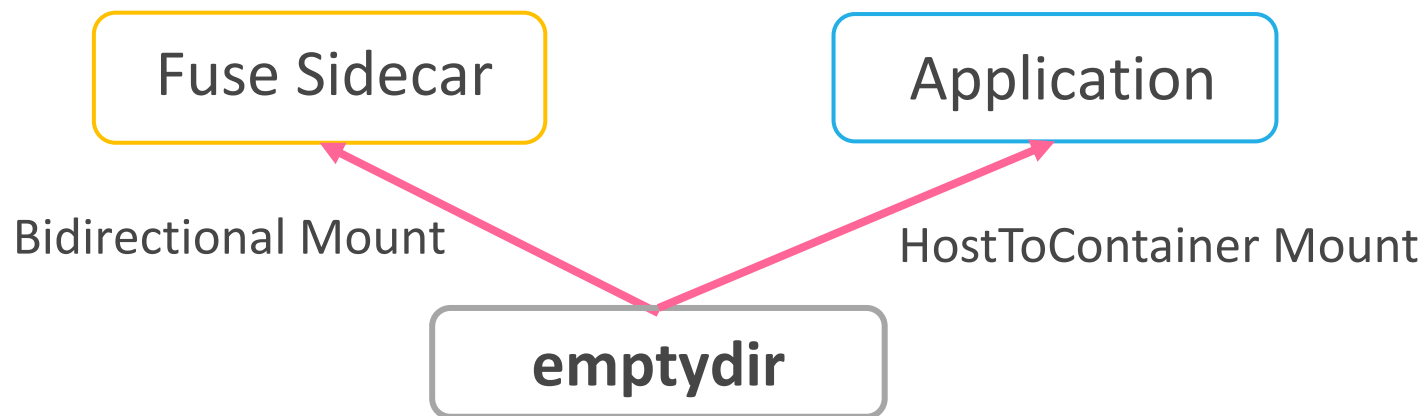
## 4. Serverless Details





## 5. Fuse Sidecar 职责

- 传播fuse信息到application中；
- 保持与application同生命周期；
- 当发现Fuse进程crash时，进行remount；





Problem	Solution
1. Master stop world gc >30s	<ul style="list-style-type: none"><li>• Java: MaxGCPauseMillis 🙌 , ParallelGCThreads 🙌</li><li>• Alluxio Master: Memory request 🙌</li><li>• (Experimental: G1GC, Jdk11)</li></ul>
2. Master availability caused frequent I/O exception thrown	<ul style="list-style-type: none"><li>• alluxio.user.rpc.retry.max.duration 🙌</li><li>• alluxio.user.rpc.retry.base.sleep 🙌</li></ul>
3. Fuse restart	<ul style="list-style-type: none"><li>• Add mechanism of fuse health checking</li></ul>

bilibili EasyAI

1

Alluxio Meets

EasyAI

2

Best

Practice

3

Acceleration  
&  
Cost Saving

4





## Case 1 : Audio Language Recognition Model – acceleration training

Framework	Pytorch
Network	TDNN neural network model
Total Epoches	20
Number of Files	~2.55Millions
File Size	~ 300Ki/file; ~800Gi total
GPU	4 * Nivida V100/16G GPU
Alluxio	2 * 500 Gi worker



## Case 1 : Audio Language Recognition Model – acceleration training

Speed Comparison			
	OSS S3Fuse	Local SSD	OSS Alluxio Cache
Completion Time	242.56h	63.48h	64.17h
Speed-up Ratio	1	3.82	3.78



## Case 2 : Video Portrait Matting – Supporting large scale training

Framework	Pytorch
Network	TDNN neural network model
Total Epoches	50
Number of Files	~20Millions
File Size	~100K/file; ~2Ti in total
GPU	4 * Nvidia V100/32G
Alluxio	4 * 600 Gi worker





## Case 2 : Video Portrait Matting – Supporting large scale training

### W/O Alluxio

- 无法将数据下载到单个节点；
- S3Fuse在海量元数据的情况下表现糟糕；

### With Alluxio

- 每个epoch运行时间稳定在18 hours
- 模型效果有显著提升；



# Q&A

Thanks to

Lu Qiu, Shawn Sun, Jiacheng Liu, Yong Yang@alluxio

Baoyou Chen, Xingui Zeng@bilibili

Yang Che@Alibaba

Lei Li, lilei06@bilibili.com

Zifan Ni, nizifan@bilibili.com