Fluid 0.8版本新特性——支持 ARM64 架构处理器以及对 Serverless 场景的深度优化

顾荣，王问骁
南京大学

https://github.com/fluid-cloudnative/fluid

发展背景及挑战

过去十年云计算、大数据、人工智能发展迅猛，将数据密集型任务运行到 K8s 等云原生环境逐渐成为一种趋势

缺乏以应用为中心的数据抽象及其生命周期管理

Computing Resource
(MR slots,Spark executors)

Memory Resource
(cache unit, data buffer)

Static Resource Allocation

Complexity

I/O Bottleneck

Inefficiency

Fluid

什么是 Fluid ？

Fluid 是一个开源的 Kubernetes 原生的分布式数据集编排和加速引擎，主要服务于云原生场景下的数据密集型应用，例如大数据应用、AI应用等。通过定义数据集资源的抽象，实现如下核心功能：

01 **数据集抽象原生支持**

将数据密集型应用所需基础支撑能力功能化，实现数据高效访问并降低多维管理成本

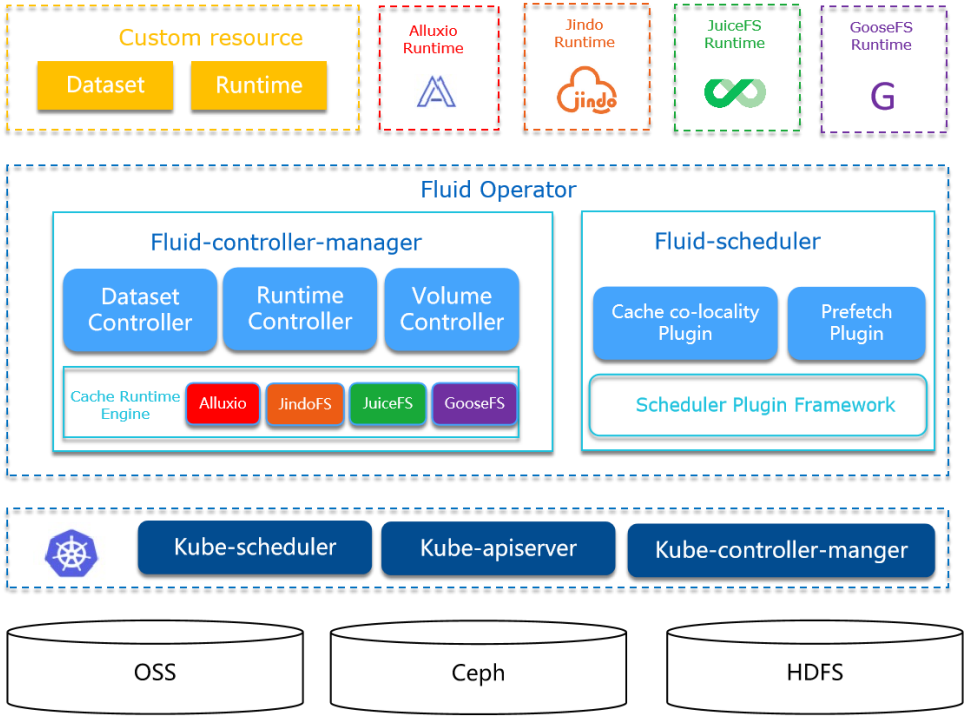
02 **基于容器调度管理的智能数据集编排**

同时考虑应用和数据的特性与位置，实现协同编排，提升性能

03 **异构数据管理和加速**

一次性访问不同来源的底层数据，通过分部署缓存引擎（ Alluxio 等）为云上应用提供数据预热与加速

……

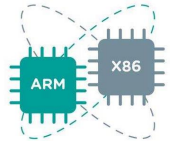


Fluid v0.8 新特性

01

支持 Arm64 架构处理器运行平台

- Fluid 的多架构支持



02

Serverless 场景支持优化的持续探索

- Fluid 的 Serverless 离线化任务支持
- Fluid 支持的 Serverless Kubernetes 平台上线，以及完全的 Serverless 化支持

03

完善对生产环境灵活性、高可用的支持

- Runtime Controller 的动态开启
- 内部组件的高可用部署支持
- Runtime 支持容器网络能力

High Availability

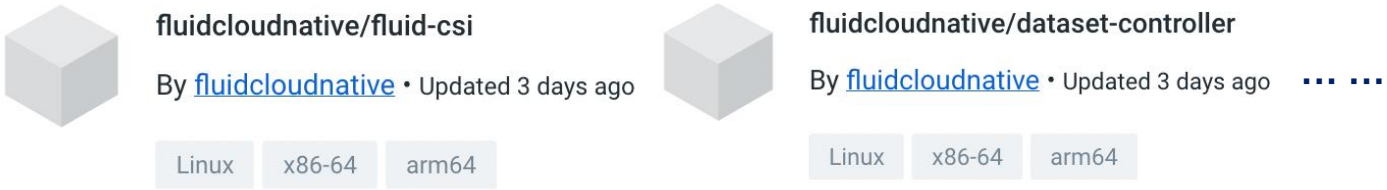
支持 Arm64 架构处理器运行平台

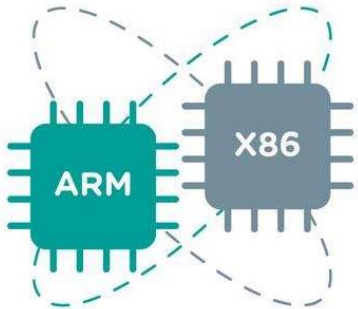
得益于 Arm 架构处理器功耗低、体积小、性价比高的特点，亚马逊云、甲骨文云、Azure和阿里云等巨头纷纷入局 Arm 云主机市场

- **TrendForce 预测，到2025年，基于 Arm 架构的服务器在数据中心的使用率将达到 22%** [1]

这意味着未来将有更多的用户将选择基于 Arm 架构的服务器去运行他们的应用 ……

为了更好的支持 Fluid 多平台的发展，在 0.8 版本中同时提供了基于 Amd64 以及 Arm64 架构的镜像





目前只有 JuiceFS 提供了 Arm 架构的官方镜像，希望未来有更多的分布式缓存引擎支持 Arm 平台！

[1] https://www.trendforce.com/presscenter/news/20220329-11178.html

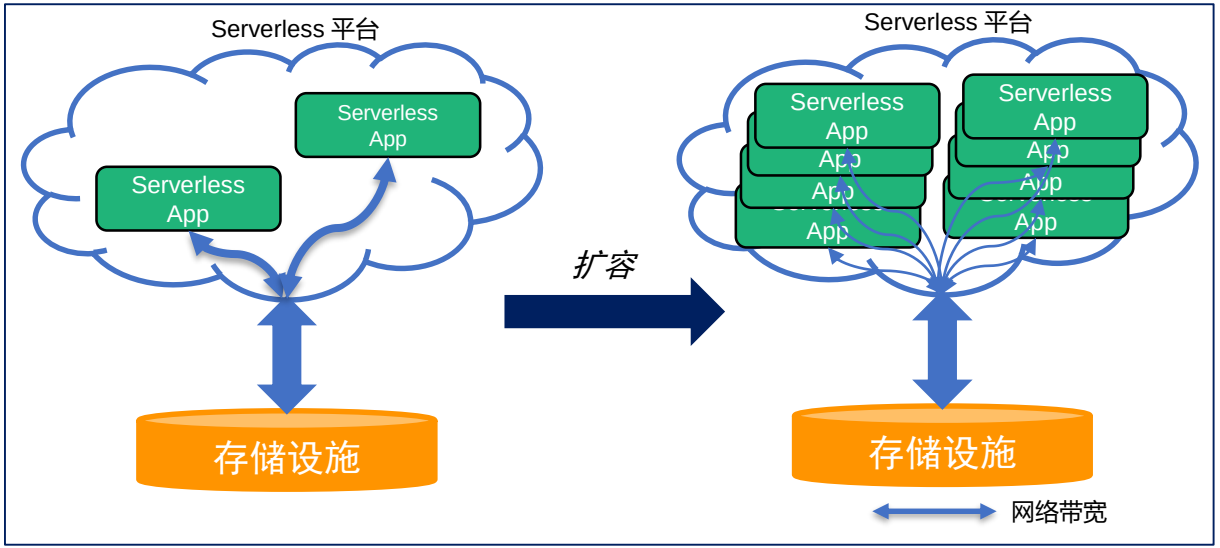
Serverless 场景支持优化的持续探索

Serverless = FaaS + BaaS

Serverless 云平台为用户提供秒级弹性的核心能力，计算资源能够实现秒级，甚至毫秒级弹性扩容，给云存储基础设施带来了巨大的压力和挑战

Serverless 容器化的技术体系对于传统的存储系统提出了新的挑战：

- ***高密访问***
- ***网络延时***
- ***IO 吞吐能力的弹性伸缩***



Serverless 场景支持优化的持续探索

数据访问优化方案

Fluid 与 Serverless 平台相结合，提供了以 ECI（Elastic Container Instance）为例的数据访问优化方案

每个 App Container 都会被控制器以 Sidecar 的方式注入一个 FUSE Container，实现对远程分布式缓存引擎的加速访问。

- 数据平面

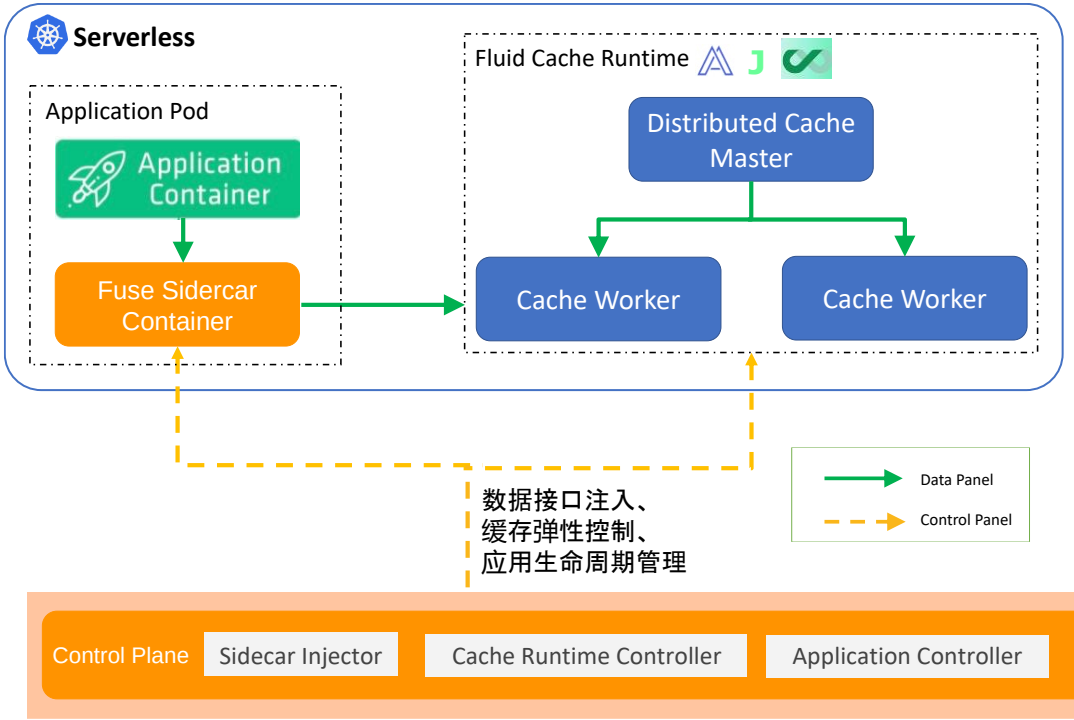
由不同 Runtime 对应的 FUSE Container 组成

- 管理平面

· **Injector**：将 Runtime 实现信息转换为 Sidecar 可以识别的信息，注入到读取数据的应用中

· **Cache Runtime Controller**：控制数据缓存弹性，同时管理数据访问权限

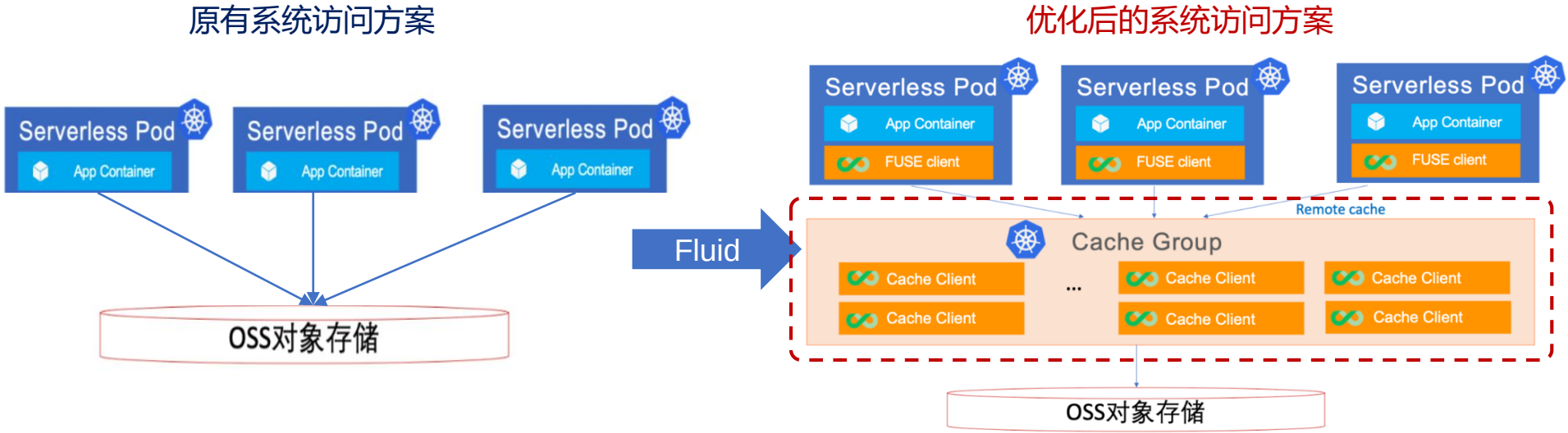
· **Application Controller**：控制 FUSE 容器的生命周期



Serverless 场景支持优化的持续探索

数据访问优化方案

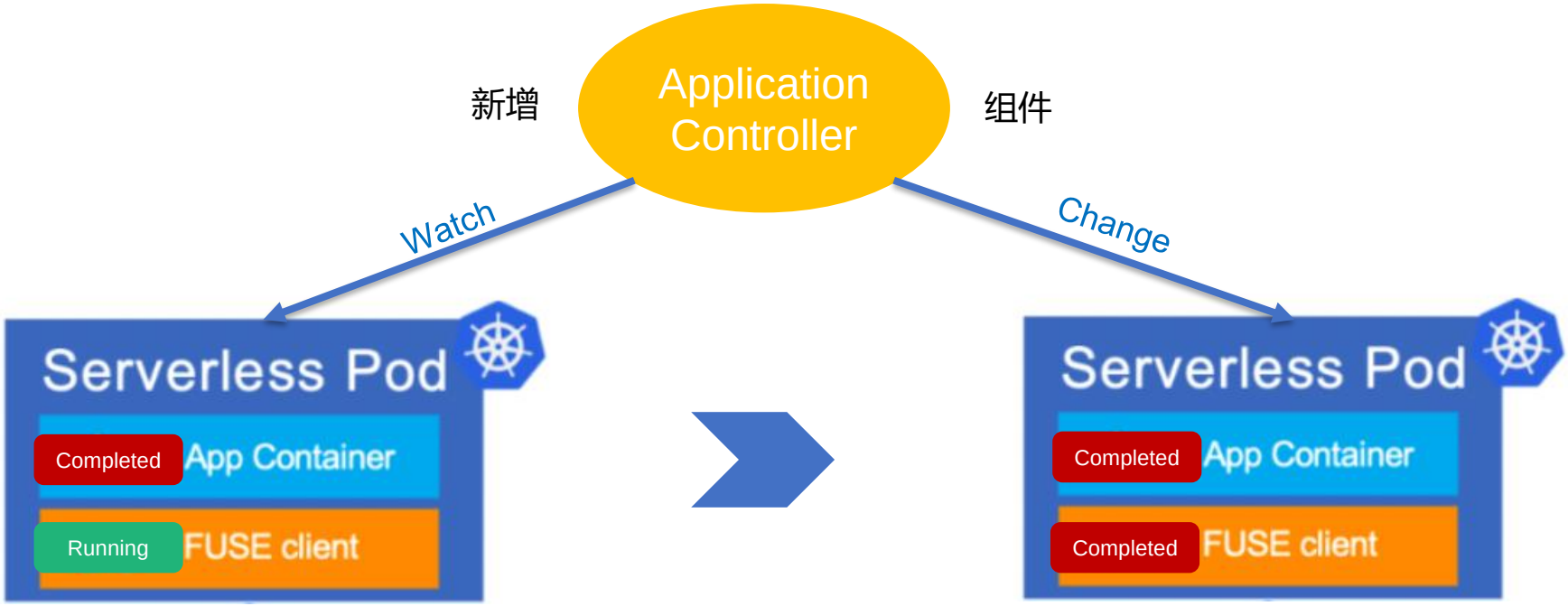
通过 Fluid 快速搭建分布式缓存引擎（如 JuiceFS）的独立集群，缓存集群以 StatefulSet 的形式运行在 ECS 节点中，提供分布式缓存服务，而 FUSE 客户端以 Sidecar 的方式运行在业务 Pod 中，业务 Pod 则运行在 ECI 节点上。



Serverless 场景支持优化的持续探索

Fluid 的 Serverless 离线化任务支持

在离线任务场景中，在业务容器完成任务退出后，用户通常希望注入的 FUSE 容器也随之主动退出



Serverless 场景支持优化的持续探索

Fluid 在 Serverless 下的使用流程

目前 Fluid on ECI 还处于公测阶段（默认没有开启），可以通过申请 *AI 免费体验席位*[1]进行使用

环境准备

Step 1

通过阿里云创建一个 ACK 版 K8s 集群，集群的 Kubernetes 版本不低于1.18。

Step 2

在 ACK 集群中安装 Fluid。

- 通过 helm 的方式安装[2]
- 使用云原生 AI 套件安装

```
$ kubectl create ns fluid-system
$ helm install fluid fluid-<version>.tgz
$ kubectl get po -n fluid-system
NAME                                        READY   STATUS    RESTARTS
AGE
alluxioruntime-controller-64948b68c9-zzsx2  1/1     Running   0
108s
csi-nodeplugin-fluid-2mfcr                  2/2     Running   0
108s
csi-nodeplugin-fluid-l7lv6                  2/2     Running   0
108s
dataset-controller-5465c4bbf9-5ds5p         1/1     Running   0
108s
```

[1] https://survey.aliyun.com/apps/zhiliao/DNgsKoIo1l
[2] https://github.com/fluid-cloudnative/fluid/blob/master/docs/en/userguide/get_started.md

Serverless 场景支持优化的持续探索

Fluid 在 Serverless 下的使用流程

部署分布式缓存引擎（以 JuiceFS 为例）

JuiceFS 采用元数据和数据分开存储的设计，元数据会被存储在元数据服务引擎中（Redis 等），数据会被存储在对象存储中（S3等），元数据服务和对象存储需要用户单独提供。

Step 1

部署 Redis 容器和对应的 Server 为 JuiceFS 提供元数据服务

Step 2

创建 Secret 资源提供元数据服务的访问地址和对象存储服务的参数（ accessKey 和 secretKey ）

```
kubectl create secret generic jfs-secret \
    --from-literal=metaurl=redis://192.168.169.168:6379/1 \
    --from-literal=access-key=<accesskey> \
    --from-literal=secret-key=<secretkey>
```

元数据访问地址

accessKey

secretKey

Serverless 场景支持优化的持续探索

部署分布式缓存引擎（以 JuiceFS 为例）

Step 3

1.创建 Dataset 资源。

```yaml
apiVersion: data.fluid.io/v1alpha1
kind: Dataset
metadata:
  name: jfsdemo
spec:
  mounts:
    - name: minio
      mountPoint: "juicefs:///"
      options:
        bucket: "<bucket-url>"
        storage: "oss"
      encryptOptions:
        - name: metaurl
          valueFrom:
            secretKeyRef:
              name: jfs-secret
              key: metaurl
        - name: access-key
          valueFrom:
            secretKeyRef:
              name: jfs-secret
              key: access-key
        - name: secret-key
          valueFrom:
            secretKeyRef:
              name: jfs-secret
              key: secret-key
```

挂载到 JuiceFS 中的目录

对象存储配置

元数据访问配置

对象存储访问配置

2.创建 Runtime 资源。

JuiceFSRuntime 对象

```yaml
apiVersion: data.fluid.io/v1alpha1
kind: JuiceFSRuntime
metadata:
  name: jfsdemo
spec:
  replicas: 5
  tieredstore:
    levels:
      - mediumtype: MEM
        path: /dev/shm
        quota: 4Gi
        low: "0.1"
```

分布式缓存引擎

副本数

缓存类型和大小

Serverless 场景支持优化的持续探索

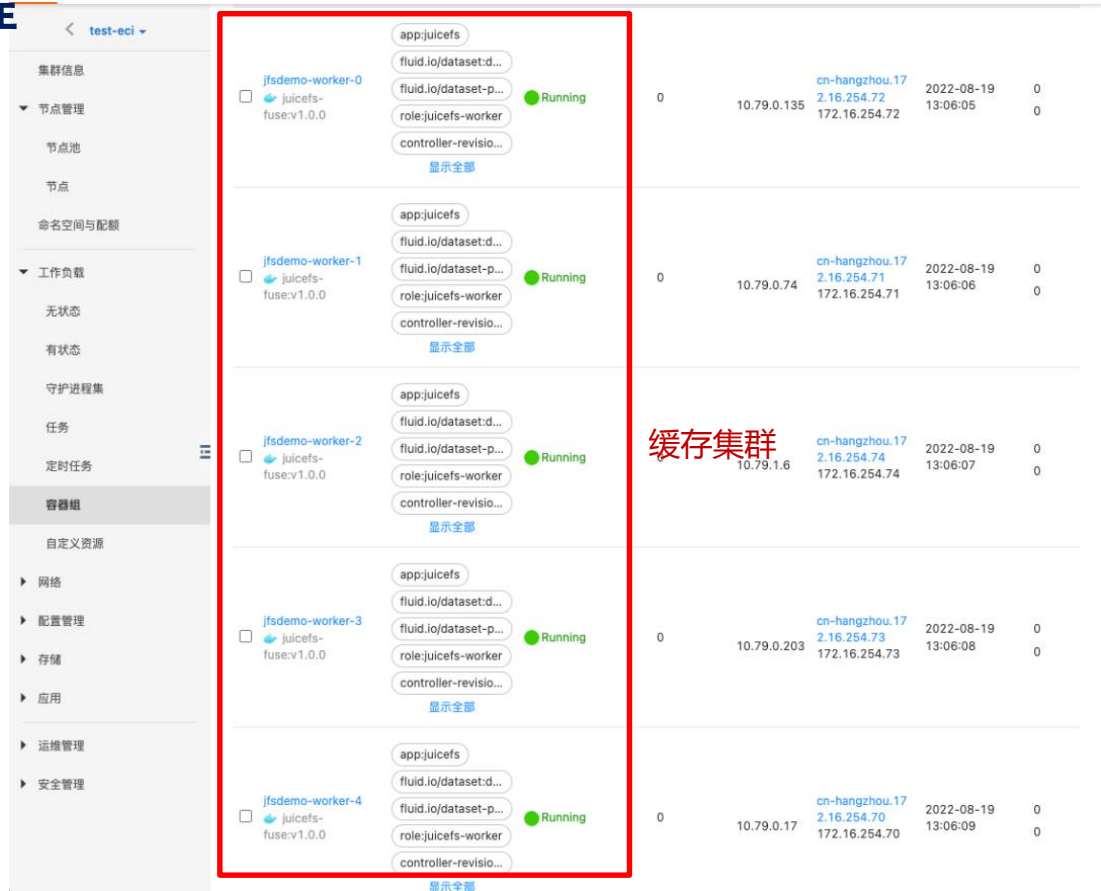
Fluid 在 Serverless 下的使用流程

缓存集群

5 个 Worker 组成了一个 JuiceFS 独立缓存集群，运行在 ECS 节点上，为客户端提供缓存服务。

如果用户有需要也可以创建 Dataload 资源对数据进行预热。

在创建应用 Pod 时只需在数据卷中指定与 Dataset 同名的存储卷即可享受分布式引擎带来的数据加速能力。



Serverless 场景支持优化的持续探索

Fluid 在 Serverless 下的使用流程

创建 Serverless 应用

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: juicefs-app
spec:
  template:
    metadata:
      labels:
        alibabacloud.com/fluid-sidecar-target: eci
        alibabacloud.com/eci: "true
    spec:

      ...

      volumes:
        - name: demo
          persistentVolumeClaim:
            claimName: jfsdemo
```

Serverless 应用

使用阿里云 ECI 能力

挂载前面创建的 Dataset

应用 Pod 创建后运行在虚拟节点上
并且被注入了 FUSE 容器

完善对生产环境灵活性、高可用的支持

1. Runtime Controller 的动态开启

之前开启方式（默认 Alluxio ）

```
helm install --set runtime.juicefs.enabled=true fluid fluid-<version>.tgz
```

Fluid 0.8

动态开启，使用时创建所需的 Runtime 类型

2. 内部组件的高可用部署支持

在新版本中实现了对内部 Webhook 以及 Controllers 组件的高可用支持

```
spec:
  replicas: 1
```

replicas ≧ 1 即开启组件的高可用

完善对生产环境灵活性、高可用的支持

3. Runtime 支持容器网络能力

在一些安全合规的场景下，用户必须要使用容器网络

在容器网络下，同一节点的 Runtime worker 和 FUSE 组件能够实现短路读。

Thank You!