Fluid：支持跨 Namespace 的数据集共享

汇报人：顾荣

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

2022 12 月 8 日

背景

业务背景基本介绍

Fluid 以 Kubernetes 原生的 Namespace 作为划分，对云上数据集（Dataset）进行管理。Fluid以用户定义的数据集为最小管理单元，负责维护数据集的数据缓存生命周期。用户可在容器内挂载 PVC 进行数据访问，并获得经数据缓存加速后更高的数据访问效率。

某些业务场景下，用户会在多个不同的 Namespace 中创建数据密集型作业，且这些作业将访问相同的数据集。例如，多个数据科学家共享相同的数据集，各数据科学家拥有自己独立的 Namespace 提交作业。如果对于每个 Namespace 重新部署缓存系统并进行缓存预热，那么就会造成数据冗余和作业启动延迟问题。

期望解决的问题

当前 Fluid 对数据缓存的管理仅支持同 Namespace 下的数据访问（即应用容器必须与所需的Dataset位于同一 Namespace），缺少对跨 Namespace 数据访问功能的支持。

P0 作为数据科学家，希望复用集群中已有的属于其他 Namespace 的数据缓存，无需重新部署缓存系统即可通过已有的缓存系统访问数据。

关键设计

01 核心设计思路

设计实现轻量化的 Dataset 和 RefController

- Dataset 增加对集群现有 Dataset 的引用功能，其行为与被引用 Dataset 保持一致；
- RefController 只处理引用类型的 Dataset，仅将数据访问过程所需的系统配置从原 Namespace 镜像复制到轻量化 Runtime 所在 Namespace，以及创建 PV / PVC；
- 具备实际缓存引擎的 Runtime 不能与声明引用的Dataset绑定；
- 不支持 Dataset 递归引用；
- 先删引用数据集，再删实际缓存数据集；

关键设计

02 设计原则

设计实现轻量化的 Dataset 和 RefController

- 以挂载 PVC 资源作为用户执行数据访问的交互 API，用户体验保持一致；
- Fluid 现有的 Dataset-Runtime 不再具备一对一的映射关系；
- 以 Dataset 为中心的数据集管理视图保持一致

关键设计

03 架构设计

Serverful 场景下，基于 Fuse + CSI 插件的跨 Namespace 数据访问架构设计

•Dataset B在CR Spec中需要指定引用位于ns-a Namespace中的Dataset A；

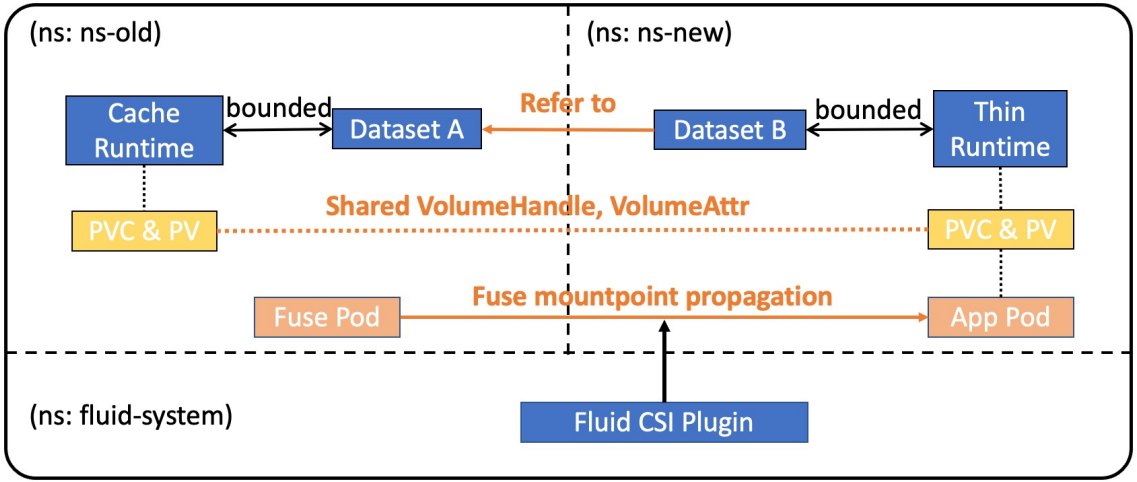
•Dataset B需要与轻量化的Runtime对象(ThinRuntime) 绑定；

•ThinRuntime Controller为新增组件，其功能包括：

1，对ThinRuntime进行声明周期管理

2，在ThinRuntime所在Namespace创建PV & PVC，PV中字段与引用的Cache Runtime对应PV保持一致

3，引用的CacheRuntime系统配置信息（ConfigMaps）复制到ThinRuntime所在Namespace



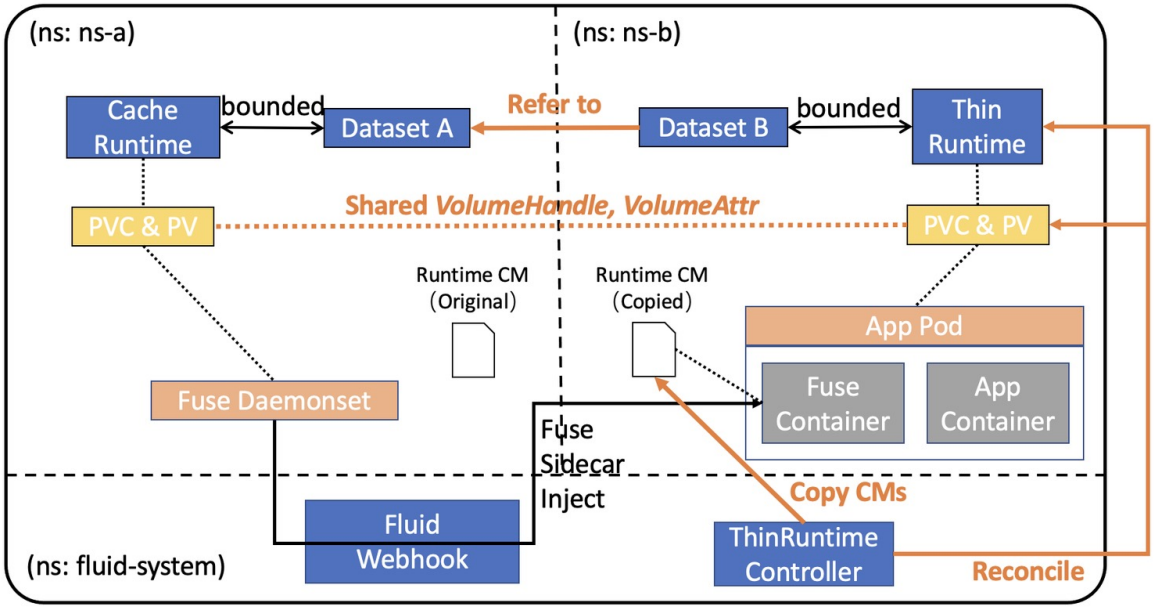
关键设计

03 架构设计

Serverless 场景下，基于 Fuse Sidecar 的跨 Namespace 数据访问架构设计

相较于 Serverful 场景：

在将引用的 CacheRuntime 系统配置信息（ConfigMaps）复制到 ThinRuntime 所在 Namespace，同时将原有的 Fuse Pod 以 sidecar 的方式注入 App Pod 内



详细设计

跨Namespace数据访问API设计和用户交互流程

Dataset API 变更

1.Spec 中新增字段 DatasetRef，用于保存所有的 Virutal Dataset

2.Spec 中 mounts 的 mountPoint 字段，新增协议前缀 dataset://

```
apiVersion: data.fluid.io/v1alpha1
kind: Dataset
metadata:
  name: virtual-dataset
spec:
  mounts:
  - mountPoint: dataset://<namespace>/<dataset_name>
    name: physical-dataset
```

详细设计

跨Namespace数据访问API设计和用户交互流程

Runtime Controller 逻辑变更

1.不支持绑定 refdataset

DatasetController变更

1.保持 Virtual Dataset 和 Physical Dataset 的 Status 字段的一致性；

当 Physical Dataset 状态变更时，应该及时更新 Virtual Dataset的状态；

2.删除 Physical Dataset 时，判断是否有 Virtual Dataset 引用，若有则无法删除；

3.只处理不包含引用的 Dataset

mount字段不包含 dataset:// 类型

详细设计

跨Namespace数据访问API设计和用户交互流程

DatasetRefController 运行流程

1.不支持 Virtual Dataset 的递归引用

判断其引用的Dataset是否是个Virtual Dataset（mount是否为dataset://）

2.不支持 Virtual Dataset 和其它形式的 mount 同时存在

mountPoint 包含 dataset:// 和其它形式

3.将 Virtual Dataset 添加到 Physical Dataset 的 DatasetRef 字段中，删除时去除字段

获取 Physical Dataset 的 DatasetRef 字段，如果不包含，则添加进去

4.跨 Namespace 数据访问所需的 PV/PVC 由 RefController 的 Reconcile 逻辑完成

详细设计

跨Namespace数据访问API设计和用户交互流程

Webhook 变更

1.获取 RuntimeInfo 的逻辑

如果 PVC 指的是 Virtual Dataset，则获取 Physical Dataset 的 Runtime 信息

2.(func injectCheckMountReadyScript) ConfigMap "check-fluid-mount-ready" 的生成

namespace 当前绑定的是 runtime 的 namespace，需要改成 virtual dataset 的namespace

3.（func GetTemplateToInjectForFuse) 获取 runtime 的 fuse daemonset 和 dataset 信息

dataset 信息根据 pvc的name/namespace 获取而不是根据 runtime 信息获取

4.获取 PVC 的 mount 属性参数

根据 pvc 的 name/namespace 获取 pvc 的属性

5. ConfigMap "${name}-${mount_type}-check-mount" 的创建（）

namespace 应该是 pvc(virtual dataset)的 namespace；owner 的 dataset 应该是 pvc 对应的 dataset，而不是 runtime 对应的 dataset。

Thank You!