

Fluid 1.0版发布，打通云原生高效数据使用的“最后一公里”

得益于云原生技术在资源成本集约、部署运维便捷、算力弹性灵活方面的优势，越来越多企业和开发者将数据密集型应用，特别是AI和大数据领域应用，运行于云原生环境中。

然而，云原生计算与存储分离架构虽然带来了资源经济性与扩容灵活性方面的优势，但也引入了数据访问延迟高、带宽开销大等方面的使用问题。

在数据访问接口层面，Kubernetes只提供了传统数据访问接入层面的接口，即异构存储服务接入和管理标准接口（CSI，Container Storage Interface），对应用如何在容器集群中高效使用和灵活管理数据并没有定义。然而，这是很多数据密集型应用依赖这样的高层数据访问和管理接口

例如：在运行AI模型训练任务时，数据科学家需要能够管理数据集版本、控制访问权限、数据集预处理、动态数据源更新、加速异构数据读取等。但是，在Fluid开源项目诞生之前，Kubernetes开源生态社区中还没有这样的标准方案，这是云原生环境拥抱大数据与AI应用的一块重要拼图。

为了解决这些挑战，Fluid开源项目对“计算任务使用数据的过程”进行抽象，提出了云原生弹性数据抽象概念（如：**DataSet**），并作为“一等公民”在Kubernetes中实现。

围绕弹性数据集Dataset，我们创建了云原生数据编排与加速系统Fluid，来实现Dataset管理（CRUD操作）、权限控制和访问加速等能力。在2021年4月份进入云原生计算基金会（CNCF）之后，经过36个月的不断研发迭代和生产环境验证，现在正式发布了其成熟稳定的v1.0大版本。

Website: <https://fluid-cloudnative.github.io/>

GitHub: <https://github.com/fluid-cloudnative/fluid>

开源驱动发展，生产环境验证



Fluid技术源自校企科研合作，项目正式开源后，吸引了来自不同行业，不同规模的社区用户将Fluid应用到更广泛的场景中：**AIGC**，大模型，大数据，混合云，云上开发机管理，自动驾驶数据仿真等。Fluid在支撑云上真实应用中不断迭代改进，并应用到生产环境中，系统的稳定性，性能和规模也变得成熟。

据公有云和私有云环境的统计，目前已有上千个Kubernetes集群在持续使用Fluid，其中在用户机器学习平台可以支持最大上万节点规模。每天在云原生环境中创建Fluid的用户主要来自互联网、科技、金融、电信、教育、自动驾驶与机器人、智能制造等领域。

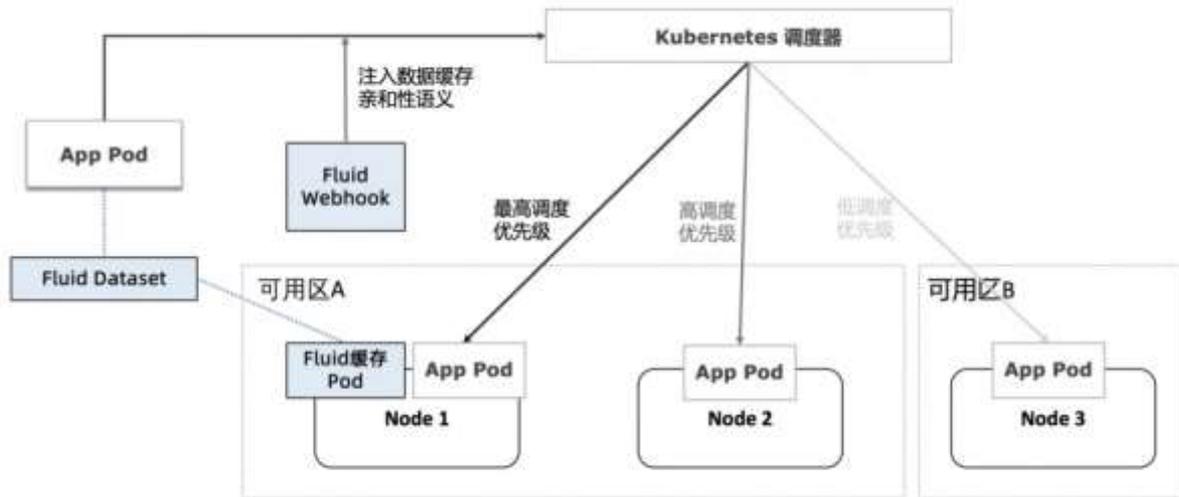
Fluid开源社区用户包括有小米、阿里巴巴集团、阿里云PAI机器学习平台、中国电信、微博、B站、360、乾象、作业帮、赢彻、虎牙、Oppo、云知声、云刻行、深势科技等。更多使用信息请查看注册用户列表。

部分用户也在Fluid开源社区分享了他们在不同场景的实践：

- [小米机器学习平台：基于 Fluid 的高效 Serverless 混合云容器 AI 平台](#)
- [从资源弹性到数据弹性，乾象如何将云上量化研究效率提升 40%？](#)
- [深势科技基于 Serverless 容器为科研人员打造高效的开发平台](#)
- [阿里集团基于Fluid+JindoCache加速大模型训练的实践](#)
- [作业帮检索服务基于 Fluid 的计算存储分离实践](#)

1. 可灵活配置的多级数据亲和性调度机制

Fluid的多级数据亲和性调度能力允许用户根据数据集缓存的位置信息对任务进行调度，而无需深入了解底层数据缓存的具体排布。Fluid通过以下方式实现这一调度策略：





2. 增加自定义数据操作DataProcess和丰富触发策略

- Fluid负责在Kubernetes中编排数据和使用数据的计算任务，不仅包括上文提到的空间上的编排，也包括时间上的编排。

空间上的编排意味着计算任务会优先调度到有缓存数据和临近缓存的节点上，这样能够提升数据密集型应用的性能。

时间上的编排则允许同时提交数据操作和任务，但在任务执行之前，要进行一些数据迁移和预热操作，以确保任务在无人值守的情况下顺利运行，提升工程效率。

- 为此，在最新版本的 Fluid里提供一种新的数据操作类型DataProcess，为数据科学家提供自定义数据处理逻辑的抽象；并且在此基础针对所有的Fluid数据操作提供不同的触发机制，包括：`once`, `onEvent`, `Cron`

2. 增加自定义数据操作DataProcess和丰富触发策略

以下例子为每2分钟运行一次数据预热:

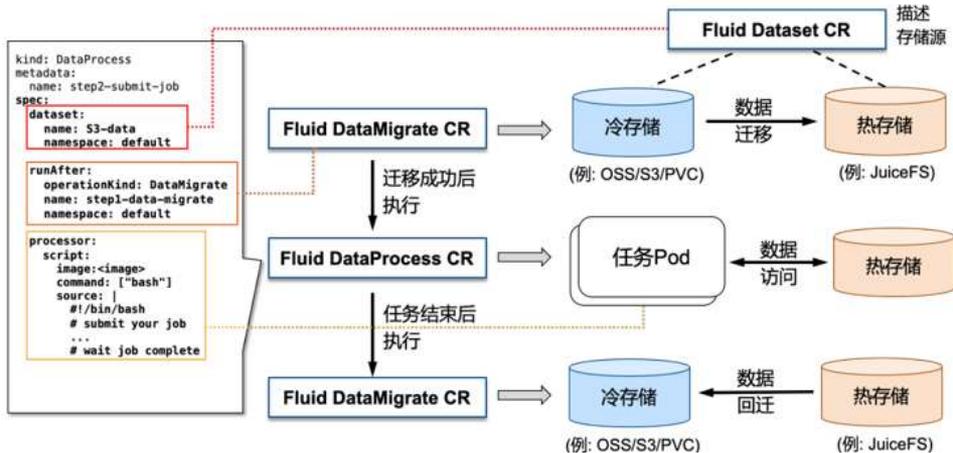
```
1  apiVersion: data.fluid.io/v1alpha1
2  kind: DataLoad
3  metadata:
4    name: cron-dataload
5  spec:
6    dataset:
7      name: demo
8      namespace: default
9    policy: Cron
10   schedule: "*/2 * * * *" # Run every 2 min
```

Fluid 1.0 新增核心功能速览

3. 数据流 DataFlow

进一步地，Fluid提供DataFlow数据流功能，允许用户通过Fluid提供的API定义自动化的数据处理流程：

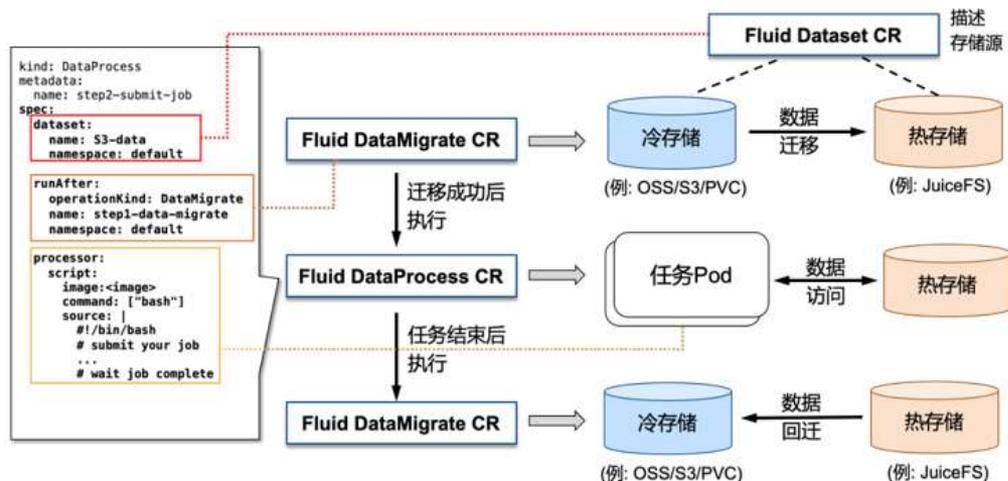
DataFlow支持Fluid的全部数据操作，包括缓存预热（DataLoad）、数据迁移（DataMigrate）、数据备份（DataBackup）等面向运维侧人员的自动化数据操作和面向数据科学家的数据处理(DataProcess)相结合，实现简单的数据操作。



3. 数据流DataFlow

以该图为例，整个的顺序是

- 将需要消费的数据从云上低速存储（例如：OSS、HDFS）迁移到高速存储（例如：JuiceFS，GPFS）
- 启动AI模型训练
- AI模型训练完成后将数据迁移回低速存储



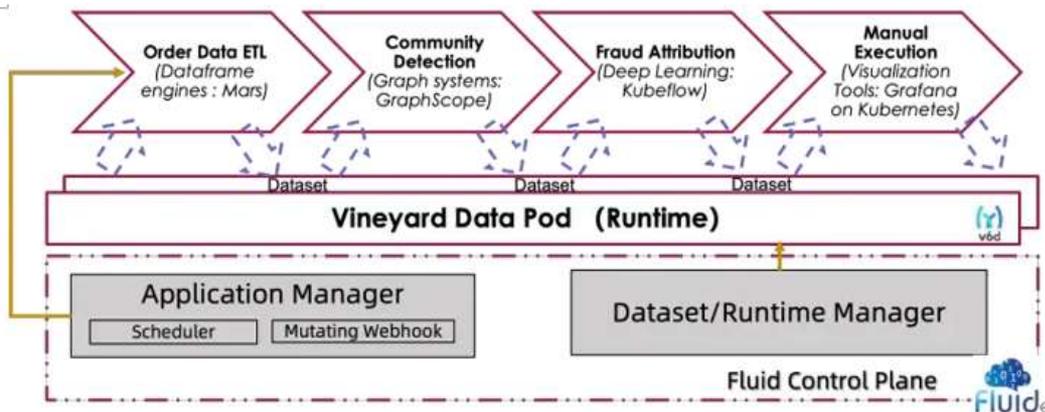
4. 通过Python SDK使用Fluid

在实践中，我们发现数据科学家更倾向于应该用代码（Python）而不是 YAML 来定义。为此Fluid提供更高层次的Python 接口来简化数据集的自动化操作和数据流的编写，下面是上面流程的Python实现：

```
1 flow = dataset.migrate(path="/data/", \  
2     migrate_direction=constants.DATA_MIGRATE_DIRECTION_FROM) \  
3     .load("/data/1.txt") \  
4     .process(processor=create_processor(train)) \  
5     .migrate(path="/data/", \  
6     migrate_direction=constants.DATA_MIGRATE_DIRECTION_TO) \  
7 \  
8 run = flow.run()
```

5.新增Vineyard对象缓存引擎支持

Fluid支持以插件的方式接入分布式缓存，目前已经支持Alluxio, JindoFS, JuiceFS等针对文件系统的分布式缓存引擎。在Fluid 1.0版本接入了分布式内存数据管理引擎Vineyard，结合了Vineyard的高效数据共享机制和Fluid的数据任务编排能力，为数据科学家提供了以Python作为操作接口的方式，让他们能以熟悉的方式高效地进行Kubernetes上的中间数据管理。





6.更多其他更新

在生产环境使用开源软件，稳定，规模和安全一直是重中之重。因此这也是Fluid持续关注和加强的领域。

- 服务于大规模Kubernetes场景
- FUSE挂载点自动恢复增强的生产可用
- 收敛Fluid组件的安全权限

帮助AI/大数据的用户在Kubernetes中更加高效、灵活、经济、安全地使用数据，这是Fluid开源项目的目标和愿景：

在1.0版本，Fluid进一步打破了数据和计算的壁垒，实现了用户可以从异构Kubernetes环境（包括runC和KataContainer）灵活使用异构数据源（对象存储，传统分布式存储，可编程的内存对象），同时通过Alluxio，JuiceFS，JindoFS，Vineyard等多种分布式缓存引擎和数据亲和性调度提升应用访问数据的效率。

在未来版本中，Fluid会继续与Kubernetes云原生生态紧密结合，同时更加关注数据科学家的效率和体验，我们计划专注解决以下问题：

- 1.针对大模型推理场景的优化，面向多种场景提升大模型加载效率。
- 2.与Kubernetes调度器相结合，自适应地实现根据Kubernetes调度器的调度结果，选择合适的数据访问方式（CSI模式和sidecar模式的自动识别）。
- 3.除了面向运行环境，支持数据科学家在使用开发环境中更灵活地使用Fluid，例如：解决数据源变更带来的容器重新启动，易造成临时数据丢失方面的问题。

我们感谢所有为Fluid 1.0 版本发布付出努力的开源项目贡献者！详细贡献内容和贡献者信息请参见Fluid开源项目1.0版本release note:

<https://github.com/fluid-cloudnative/fluid/releases/tag/v1.0.0>

我们感谢Fluid开源社区用户给予的验证反馈和合作支持！Fluid开源社区用户公开信息登记列表请参见：

<https://github.com/fluid-cloudnative/fluid/blob/master/ADOPTERS.md>



Thank You!