小米机器学习平台：基于 Fluid 的高效 Serverless 混合云容器 AI 平台

转载自InfoQ Fluid开源项目 2023-09-12 13:23 发表于江苏

作者：小米机器学习平台 何逸凡、刘国明

策划：褚杏娟

业务背景

小米机器学习平台（以下简称 CloudML），是小米针对机器学习进行全流程优化的高性能、分布式云服务。开发者可以在云端使用 GPU 训练模型，秒级启动分布式训练任务，兼容 TensorFlow、PyTorch、PaddlePaddle、DeepSpeed 等深度学习框架，可以一键部署训练好的模型，或者创建基于 GPU 的开发环境，提供模型开发、训练、调优、测试、部署和预测等一站式解决方案。CloudML 还开放了 API、SDK、命令行和 Web 控制台等多种访问方式，支持灵活的秒级计费，方便人工智能专家使用。但是近年来由于需求规模的不断扩大，越来越多地面临算力资源不足的挑战。

技术选型: 混合云下的 Serverless 容器

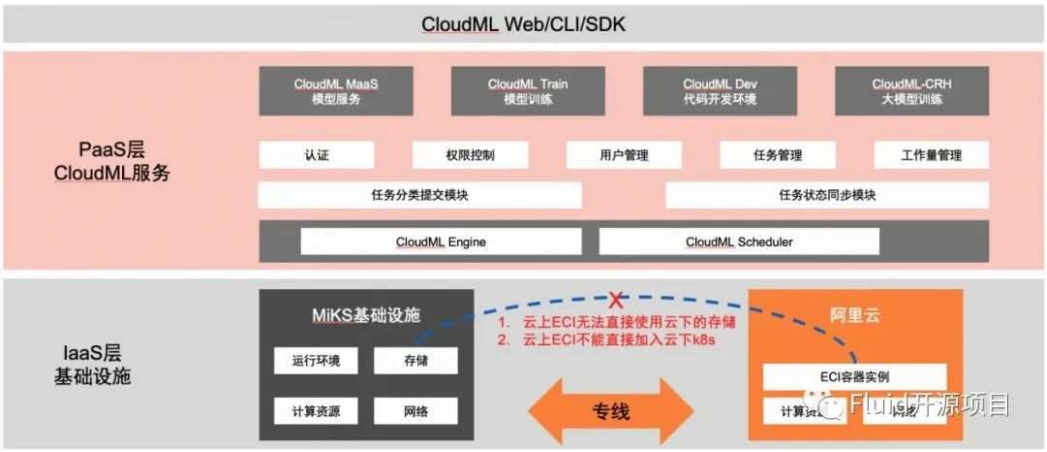
CloudML 所依托的自建算力集群由于资源池容量、资源弹性能力相对有限，导致在业务低谷时存在高昂的资源闲置成本，业务高峰时存在资源紧张的问题。在此背景下，我们考虑引入公共云上的算力资源作为自建资源池的补充，但是考虑到业务自身的演进和数据安全可控，CloudML 采用了混合云架构。资源调度策略是以自建集群的资源为主，按照业务对资源的需求弹性使用公有云资源。具体实现方法是在资源需求高峰时使用公有云资源，在低谷时释放掉公有云资源。这样一方面能够满足负载高峰时业务的算力需求，另一方面也兼顾了数据安全、成本等多方面的要求。

混合云场景下我们优先采用了Serverless容器进行落地，这主要其具有两个重要的优势：在操作方面，能够屏蔽混合云的基础设施差异性，降低运维复杂度。在成本方面，这类型容器能够实现快速按需扩容和销毁，按资源实际有效使用时长付费，最大化的降低成本。

混合云下使用 Serverless 容器的技术挑战

迁移到基于 Serverless 容器架构的混合云之后，我们获得了 Serverless 容器带来的敏捷、安全、弹性、低成本等优势，然而我们也遇到了几个重要的技术挑战：

1. **目前公有云平台不支持对 Serverless 容器平台中的存储类型进行定制扩展**：公有云只支持云厂商自带的公有云存储类型（如 NAS、对象存储等），无法直接适配公司内部自研的分布式文件存储产品（以下简称 StarFS）。
2. **缺乏可信透明的数据接入方式**：如何在 Serverless 容器的黑盒系统使用过程中规避数据泄露，如何确保数据在存储、传输、访问过程中安全可靠。
3. **CloudML 平台用户不应感知基础设施层面的差异**：切换到混合云后，用户提交的计算任务会运行在公有云或自建集群中。当用户任务在公有云和自建集群之间进行迁移时，用户使用体验需要与自建集群上使用体验保持一致。例如混合云场景中，最好沿用原来在自建集群中已有的 PVC、PV 资源而不需要做过多的变更。



解决方案：ACK Fluid 打通混合云下数据访问

经过调研选择之后，我们发现基于Fluid架构的方案几乎可以完美解决上述难题，Fluid 是云原生计算基金会旗下一个面向数据密集型应用的高效支撑的开源项目。

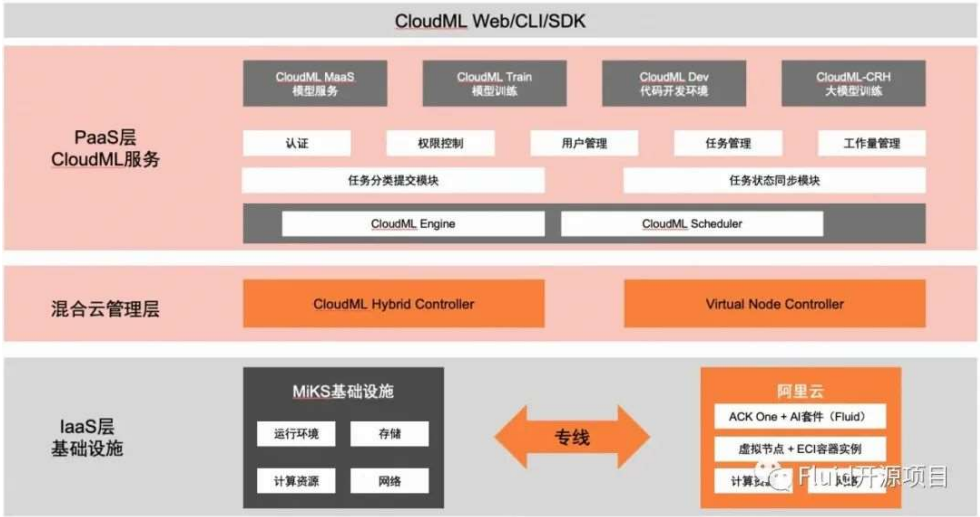
1. **简单可扩展**：Fluid 最新的版本中 ThinRuntime 开发简单，无需具备复杂的 Kubernetes 开发知识。基于这套方案，只需要了解 Dockerfile 构建就可以完成，一般开发工作 2-3 小时左右，从而显著降低了接入 StarFS 的工作成本。
2. **安全可控的数据访问**：Fluid 中 ThinRuntime 以容器化的方式支持我们以自定义方式实现数据访问。整个数据访问过程可控，无需向公共云平台提供实现细节。
3. **无侵入的客户端部署**：Fluid 同时支持 CSI 和 FUSE Sidecar 两种客户端部署模式，在 Serverless Container 的模式下 FUSE Sidecar 是一种对公共云平台无侵入式接入的选择。
4. **无缝接入开源 K8s 体系、兼具可观测性和可控制性**：第三方存储客户端只需要实现自身的容器化，就可以无缝接入 Kubernetes 体系，并获得可观测性和计算资源可控制性。

但是，我们在实践中发现：

1. 开源 Fluid 中的 FUSE Sidecar 需要依赖 privileged 权限，这在多数的 Serverless Container 平台下都是不支持的；
2. Fluid 的抽象层相关资源对象生命期的管理比较复杂。比如需要创建和管理 Fluid Dataset 资源的生命周期。导致使用人员有一定的学习成本，业务也有相应改造成本。

这时我们发现，阿里云 ACK 云原生 AI 套件中提供的 ack-fluid 存储系统接入方案可以很好的解决这个问题：

1. ack-fluid 基于开源 Fluid 标准对于 ThinRuntime 提供了完整的支持，只要满足开源要求就可以适配 ack-fluid。StarFS 接入只需在开源 Fluid 下即可完成调试，同时借助 ACK One 注册集群模式可获得阿里云商业版 Fluid 全部功能。
2. ack-fluid 与阿里云的 ECI 做了无缝支持，无需开启 privileged 权限，就可以满足云上弹性容器实例 ECI 访问云下自建存储系统的需求。
3. ack-fluid 提供对于 StarFS 自建 pvc 的丝滑兼容，无需了解 Fluid 的使用方式，只需要 pvc 中添加特定 label 即可，满足了 CloudML 用户无需感知基础设施层面的差异的需求。而在开源 Fluid 中这个工作就非常复杂，需要手动创建和管理 Dataset 和 ThinRuntime 的生命周期。



最终通过 Fluid 提供的通用标准协议，我们只需要按照 Fluid 中的 ThinRuntime 开发规范对接 StarFS 的存储协议，就可以实现云上弹性计算资源 ECI 访问云下自建存储系统的能力。ack-fluid 可以将 Serverless Pod 中数据访问从 CSI 协议转换成 sidecar 协议，同时也可以通过 ThinRuntime 的资源设置容器中的数据访问的资源分配情况。

操作步骤

1. 部署 fluid 管理组件 dataset-controller、fluid-webhook、fluid-pvc-wrapper、thinruntime-controller, 安装 ack-fluid , 安装过程可以参考文档：https://help.aliyun.com/document_detail/208336.html

2. 配置 Starfs 相关的 k8s 原生的 pv、pvc;

3. 自动创建 Fluid Dataset 和 ThinRuntime;

执行以下命令在 PVC 上打上特殊标签，触发 Fluid 使用上述创建的名为 ossfs-profile 的 ThinRuntimeProfile 作为模版，自动创建 Dataset 和 ThinRuntime。

```
$ kubectl label pvc starfs-pvc fluid.io/runtime-profile=starfs-profile
```

4. 创建 ThinRuntimeProfile，通过其注入 fuse-client 启动方式：

```
apiVersion: data.fluid.io/v1alpha1

kind: ThinRuntimeProfile

metadata:

  name: ossfs-profile

spec:

  fileSystemType: starfs

  fuse:

    command:

    - sh

    - -c

    - python3 /fluid_config_init.py && chmod u+x /mount-starfs.sh && /mount-s

    image: micr.cloud.cn/cloudml/starfs-eci-mount

```

```
    imagePullPolicy: Always

    imageTag: v1.4.1

    resources:

      requests:

        cpu: 500m

        memory: 1Gi
```

结果收益

混合云场景下 Serverless 容器方案完美落地，很好地满足了我们简单、安全、弹性、低成本等诉求，小米 CloudML 深度学习平台可以稳定高效地响应业务需求。

尤其值得一提的是，通过引入阿里云容器服务 ACK Fluid 很好地解决了相关技术难点：

- 首先，对于自有定制化存储 StarFS 的访问提供了很好的扩展支持，并且得益于 Fluid 提供的数据集可观测性功能，我们能够获取云上工作负载的数据访问特性，从而支持数据热加载和资源分配调优。
- 其次，方案接入简单、管理便捷。我们自行完成 StarFS 与 Kubernetes 环境的对接工作（编写自定义的 CSI 插件或编写 Runtime Controller 与 Fluid 对接），整个 thinRuntime 开发简单，无需我们具备复杂的 Kubernetes 定制开发知识。基于这套方案，我们只需要了解 Dockerfile 构建就可以完成，开发工作 2-3 小时左右，显著降低了使用 ECI 接入 StarFS 的工作成本。

未来展望

未来，小米机器学习平台计划进一步拓展不同的应用场景和优化用户体验，提升训练任务性能及资源效率。具体来说：

1. **降低专线网络成本**：我们目前已经在自建集群接入了公上弹性计算资源，间的也打通了公上算力与自建数据集通道。但是实际成本和性能还存在优化空间。因此，下一步需要持续优化这项工作，比如考虑使用分布式缓存运行时来降低专线网络成本。

2. **协同任务调度与数据缓存弹性**：使得我们的业务系统有能力识别一段时间内使用相同数据集的任务并发量，并在任务排队过程中执行数据预热与弹性扩缩容。当数据缓存或访问吞吐达到一定条件时，将排队任务从等待转换为可用状态，从而加速计算对数据的访问。

同时，由于 ack-fluid 基于开源 Fluid 构建，拥有灵活性、相对透明及有开源社区支持的好处。感谢 Fluid 社区以及阿里云车漾、徐之浩和南京大学的顾荣老师的帮助。未来我们计划一起共建 Fluid 社区，助力 Fluid 开源技术的发展。

本文转载自InfoQ，原文链接请点击文末**阅读原文**。

阅读原文