

Fine-tuning LLM with Argo Workflows : A Kubernetes-native Approach

Shuangkun Tian

Argo Maintainer, Alibaba Cloud Software Engineer



Content

- 01 The Challenge of Fine-tuning**
- 02 Why Argo Workflows for Fine-tuning**
- 03 Building a TCM Assistant on DeepSeek**
- 04 Benefits and Future**

Part 01

The Challenge of Fine-tuning

AI



What is Fine-tuning ?

- Adapts pre-trained models to specific tasks/domains via targeted training.



Base Model: Deepseek R1,
GPT-3、BERT



Fine-tuned Model: DeepSeek-
Finance、Claude、SciBERT

Challenge of Fine-tuning

- Substantial Computational Resources
 - 🖥️ Various heterogeneous devices : CPU、 GPU、 DPU
 - 🔥 Computational Cost : Single training run over \$10k+
- Complex Workflows
 - 🔄 Muti-stages : Preprocessing → Training → Evaluation ...
 - ⚙️ Larger Tasks (1w+) 、 Muti-workflows (1000+)

Manual workflows = High cost + Low reliability

Part 02

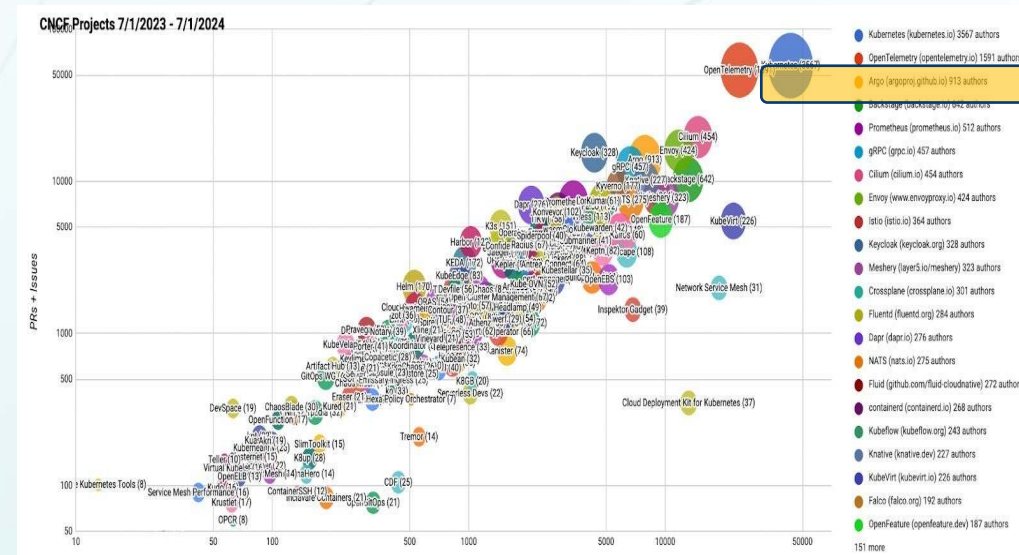
Why Argo Workflows ?

AI

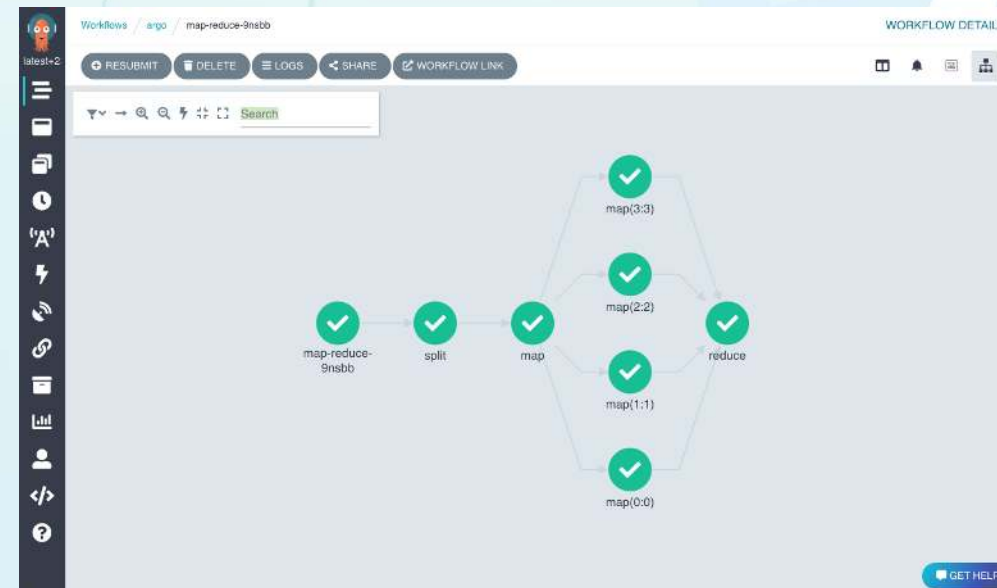


What is Argo Workflows ?

- Argo Project
 - Argo Workflows
 - Argo CD
 - Argo Events、
 - Argo Rollout
 - Third Active Community in CNCF



- Most Popular Workflow Engine
 - Machine Learning pipelines
 - Data and batch processing
 - Infrastructure automation
 - CI/CD



Argo Workflows for ML Pipelines



Kubeflow Pipelines



mlflow™



Numa flow

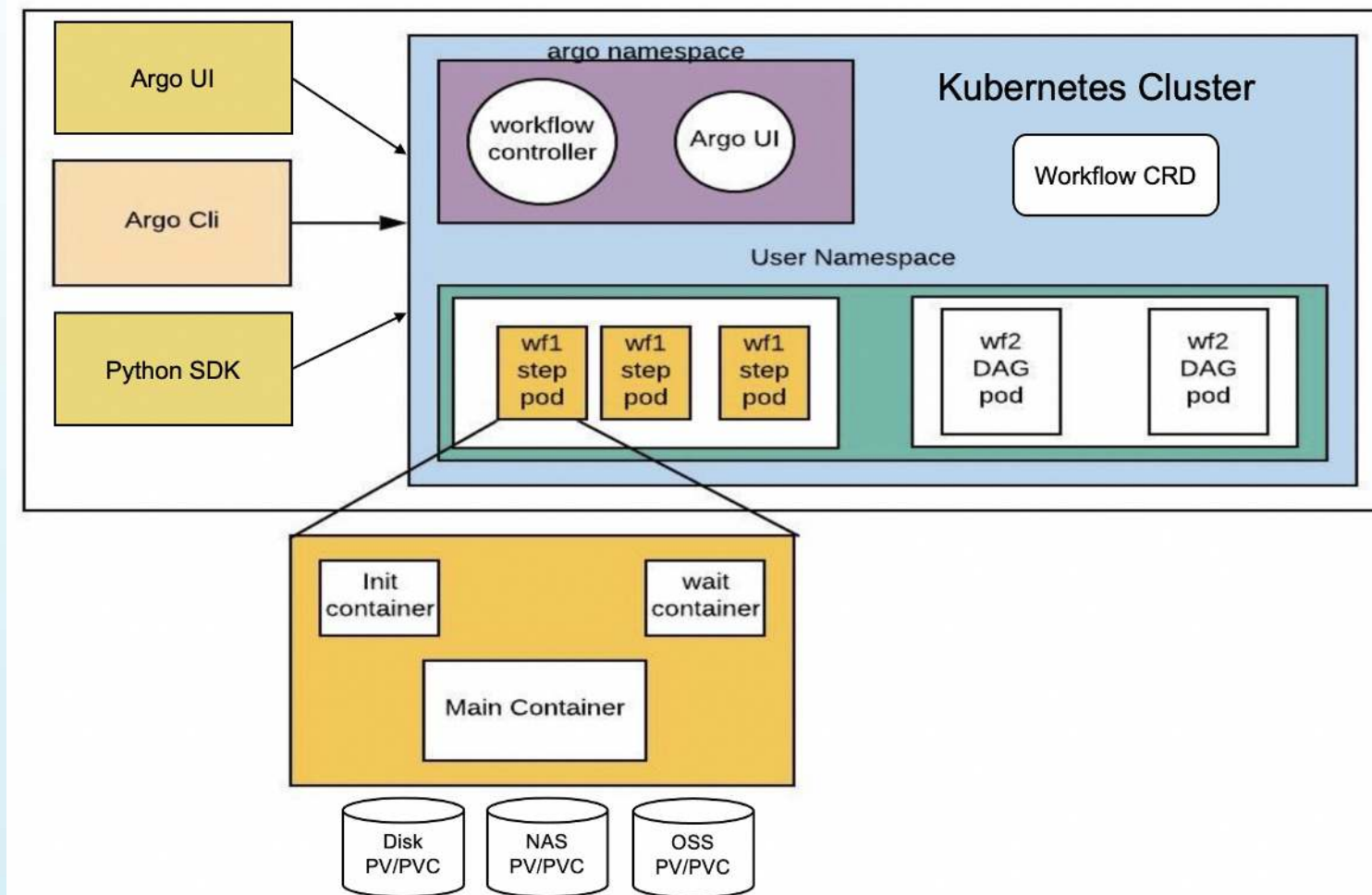


Argo Workflows

- Over 8K companies use Argo or those ML tools based on Argo
- Argo Workflows has been a core component in the orchestration of AI/ML workloads on Kubernetes.

Why Argo Workflows for AI/Fine-tuning

- Kubernetes-native
- Scalability
- Reproducibility
- Fault tolerance
- Visibility
- Ease of Use
- YAML/Python
- ...



Part 03

Building a Traditional Chinese Medicine Assistant Based on DeepSeek



Workflow Defintion

```
apiVersion: argoproj.io/v1alpha1
kind: Workflow
metadata:
  name: fine-tuning
spec:
  entrypoint: start
  templates:
  - name: start
    steps:
    - name: training
      template: fine-tune-template
    - name: evaluation
      template: evaluation-template
  - name: fine-tune-template
    container:
      image: argo-demo/fine-tuning:beta
      command: [python]
      args: ["fine-tune.py", "{{model.name}}"]
      resources:
        requests:
          cpu: 2
          memory: 4Gi
          nvidia.com/gpu: 1
  - name: evaluation-template
    container: ...
```

Dependency Definition:

Serial dependencites / Complex DAGs

Template:

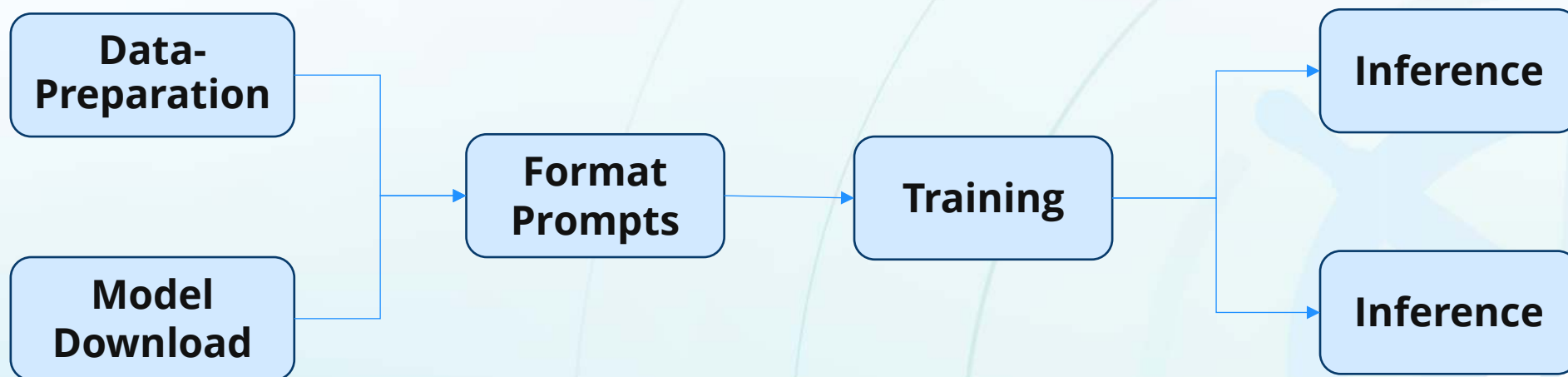
Image, Arguments,
Command, and Resouces

Fine-tuning LLM

- Dataset Prepare
 - HuggingFace Hub (e.g., `datasets.load_dataset("glue", "marc")`)
 - Raw Data → Clean → Tokenize
- Base Model
 - DeepSeek-R1、DeepSeek-R1-Distill
 - 4-bit Quantized Base Model
- Training
 - LoRA Adapters
 - Full Fine-tuning
- Evaluation
 - Human evaluation

AI

Fine-tuning Chinese Medicine Assistant on Deepseek



- Traditional-Chinese-Medicine-Dataset-SFT
- DeepSeek-R1-Distill-Qwen-7B
- 4bit

- Tokenization
- Prompt_Style
- Traditional Chinese Medicine Expert

- LoRA
- GPU
- Train Parameters

- Base Model
- Fine tuned Model
- How to cure persistent cough

Demo on ACK Argo Workflows in Alibaba Cloud



- <https://github.com/AliyunContainerService/argo-workflow-examples/tree/main/fine-tune-with-argo>

The screenshot displays the Alibaba Cloud ACK console interface. The left sidebar shows the navigation menu with options like '注册集群', '边缘集群', 'ACK集群', and '工作流集群'. The main content area is titled 'signor' and shows the cluster's status as '运行中' (Running). Below this, there are tabs for '基本信息' (Basic Information) and '连接信息' (Connection Information). The '基本信息' tab is active, showing details such as the cluster ID, region, and various settings like '删除保护' (Delete Protection) and '事件驱动' (Event-driven). A section titled '关联云资源' (Associated Cloud Resources) lists resources like VPC, Security Group, SLB, and VSwitch. At the bottom, there are sections for '常用操作' (Common Operations) including '工作流控制台 (Argo)' and '日志服务SLS', both of which are enabled.

分布式容器平台

signor

运行中 创建工作流集群 删除工作流集群

分布式工作流Argo，是无服务器Serverless工作流引擎。基于Kubernetes集群构建，托管了开源Argo Workflows，采用无服务器模式，使用阿里云弹性容器实例ECI运行工作流；通过优化Kubernetes集群参数，实现大规模工作流的高效弹性调度；同时，配合抢占式ECI实例，优化成本。

基础信息 连接信息

基本信息

集群ID	c52eb35c269254513b0c07e37e024b9c5	状态	运行中
地域	亚太东南1（新加坡）	创建时间	2025-01-17 17:33:51
删除保护	未开启	API Server 公网链接端点	https://8.222.191.62:6443
事件驱动	未开启	API Server 内网链接端点	https://192.168.0.103:6443 设置访问控制
集群监控	未开启		

关联云资源

虚拟专有网络VPC	vpc-t4njznwdspedo10dek823
安全组	sg-t4nexu9qmkgkux2bhcy3
APIServer负载均衡 (SLB)	lb-t4nnj16uwc2qj908e9tr2
虚拟交换机	vsw-t4ngxdto3nvzy88w7r4t7 添加

常用操作

工作流控制台 (Argo) 开启

可前往访问Argo UI管理Argo Workflows。

→ 设置访问控制

日志服务SLS 开启

工作流集群集成了阿里云日志服务SLS，收集工作流运行过程中Pod产生的日志，上报到您账号下...

→

Part 04

Benefits and Future

AI



Benefits by this Approach

- Save Cost
 - Use CPU to handle time-consuming tasks such as downloads
 - GPU only for train
- Improve efficiency
 - Automation Task Orchestration
 - Ease to Scale
- Reproducibility
 - Traceable
 - Version Control

AI

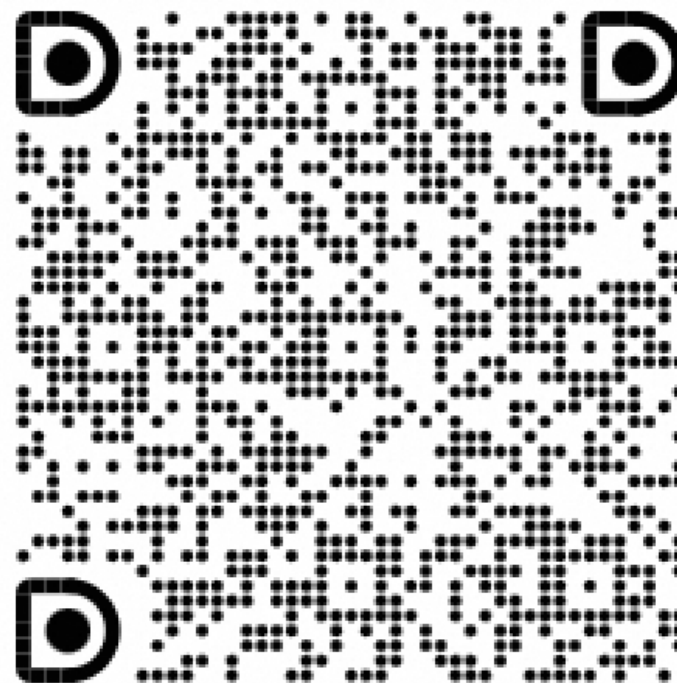
Future

- CI/CD System
 - Improve R&D efficiency
- Argo Events
 - Event-Driven Workflow
- Integrate Spark、Ray、Pytorch
 - Unify platform

<https://github.com/argoproj/argo-workflows>



ACK One & Argo Customer Exchange Group



Thanks.

