## 

## 

## **Volcano**

## **INTRODUCTION:**

Volcano is a cloud native system for high-performance workloads, which has been accepted by [Cloud Native Computing Foundation (CNCF)](https://www.cncf.io/) as its first and only official container batch scheduling project. Volcano supports popular computing frameworks such as [Spark](https://spark.apache.org/), [TensorFlow](https://tensorflow.google.en/), [PyTorch](https://pytorch.org/" \t "_blank), [Flink](https://flink.apache.org/" \t "_blank), [Argo](https://argoproj.github.io/), [MindSpore](https://www.mindspore.cn/en" \t "_blank), and [PaddlePaddle](https://www.paddlepaddle.org.cn/" \t "_blank). Volcano also supports scheduling of computing resources on different architecture, such as x86, Arm, and Kunpeng.

### 

### **Project Summary**

|  |  |
| --- | --- |
| **Website** | https://volcano.sh/en/ |
| **Organization/Foundation Name** | Volcano was accepted to [Cloud Native Computing Foundation](https://www.cncf.io/) (CNCF) on April 9, 2020 |
| **License** | Apache License version 2.0 |
| **Open/Proprietary** | Open Source |
| **Source Path (if open source)** | https://github.com/volcano-sh/volcano |
| **Brief Description** | Volcano is system for running high-performance workloads on Kubernetes. It features powerful batch scheduling capability that Kubernetes cannot provide but is commonly required by many classes of high-performance workloads, including:   * Machine learning/Deep learning * Bioinformatics/Genomics * Other big data applications |

Table 3:Volcano Summary

### **Project Details**

#### **Key Features**

### **Rich scheduling policies**

Volcano supports a variety of scheduling policies:

* Gang scheduling
* Fair-share scheduling
* Queue scheduling
* Preemption scheduling
* Topology-based scheduling
* Reclaim
* Backfill
* Resource reservation

You can also configure plug-ins and actions to use custom scheduling policies.

### **Enhanced job management**

You can use enhanced job features of Volcano for high-performance computing:

* Multi-pod jobs
* Improved error handling
* Indexed jobs

### **Multi-architecture computing**

Volcano can schedule computing resources from multiple architectures:

* x86
* Arm
* Kunpeng
* Ascend
* GPU

### **Faster scheduling**

Compared with existing queue schedulers, Volcano shortens the average scheduling delay through a series of optimizations.

#### **Architecture**

## image**Overall Architecture**

Fig 8: Volcano architecture 1

#### **Application scenarios of Volcano**

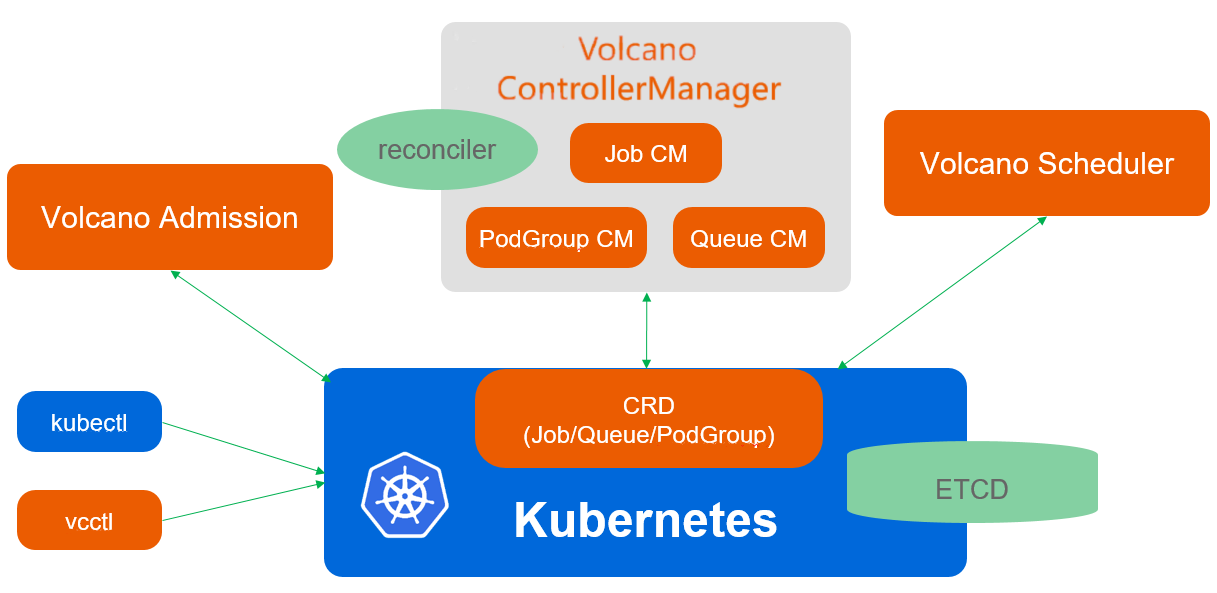
Volcano is designed for high-performance workloads running on Kubernetes. It follows the design and mechanisms of Kubernetes.

Fig 9: Volcano architecture 2

#### **Volcano architecture**

Volcano consists of **scheduler** / **controller manager** / **admission** / **vcctl**:

##### **Scheduler**

Volcano Scheduler schedules jobs to the most suitable node based on actions and plug-ins. Volcano supplements Kubernetes to support multiple scheduling algorithms for jobs.

##### **Controller Manager (CM)**

Volcano CMs manage the lifecycle of Custom Resource Definitions (CRDs). You can use the **Queue CM, PodGroup CM**, and **VCJob CM.**

##### **Admission**

Volcano Admission is responsible for the CRD API validation.

##### **vcctl**

Volcano vcctl is the command line client for Volcano.

## **Ecosystem**

Volcano allows you to use mainstream computing frameworks.

Volcano has been commercially used as the infrastructure scheduling engine by companies and organizations.

### **Kubeflow on Volcano**

Volcano is an enhanced high performance computing task batch processing system built on Kubernetes. As a platform for high performance computing scenarios, it makes up for Kubernetes’ lack of basic capabilities in machine learning, deep learning, HPC, and big data computing scenarios, including gang-schedule scheduling capability, computational task queue management, task-topology, and GPU affinity scheduling. In addition, Volcano has enhanced the batch creation and life cycle management of computing tasks, fair-share, binpack scheduling and other aspects on the basis of the native Kubernetes capability. Volcano has fully solved the problem of distributed training in Kubeflow.

# **Flink on Volcano**

Apache Flink is an open-source streaming framework developed by the Apache Software Foundation. At its core, Apache Flink is a distributed streaming data streaming engine written in Java and Scala. Flink executes any stream data program in data parallelism and pipelining. Flink’s pipelined runtime system can execute both batch and stream programs. In addition, the Flink runtime itself supports the execution of iterative algorithms.

# **MindSpore on Volcano**

MindSpore is a new generation of deep learning framework launched by Huawei. It is derived from the best practices of the whole industry, which best matches the computing power of the rise processor, supports the flexible deployment of the whole scene of terminal, edge and cloud, and creates a brand new AI programming paradigm and lowers the threshold of AI development.

### **Spark on Volcano**

Currently, there are two ways to support the integration of Spark on Kubernetes and volcano. - Spark on Kubernetes native support: maintained by the [Apache Spark community](https://github.com/apache/spark) and Volcano community - Spark Operator support: maintained by the [GoogleCloudPlatform community](https://github.com/GoogleCloudPlatform/spark-on-k8s-operator" \t "_blank) and Volcano community.

Graphical user interface

Description automatically generated**Volcano Contributors**

FIG 10: volcano contributers

# **Volcano Maintainers**

|  |  |  |
| --- | --- | --- |
| **Maintainer** | **GitHub ID** | **Affiliation** |
| Klaus Ma | [k82cn](https://github.com/k82cn) | Huawei |
| Kevin Wang | [kevin-wangzefeng](https://github.com/kevin-wangzefeng) | Huawei |
| Zhonghu Xu | [hzxuzhonghu](https://github.com/hzxuzhonghu) | Huawei |
| Thor-wl | [Thor-wl](https://github.com/Thor-wl) | Hjmicro |
| William-wang | [william-wang](https://github.com/william-wang) | Huawei |
| Liang Tang | [shinytang6](https://github.com/shinytang6) | Baidu |

Table 4: Volvano Maintainers