Volcano: Bring Batch Capability Into Kubernetes

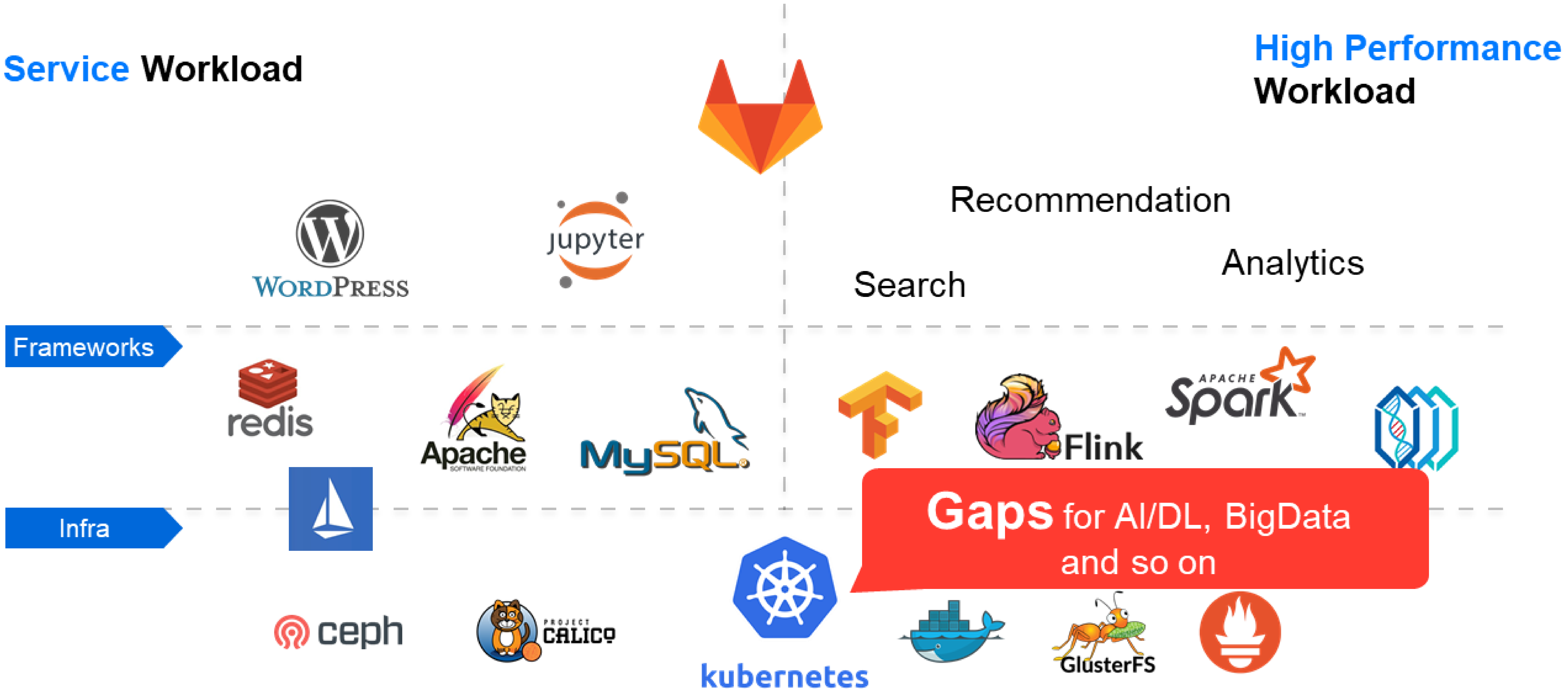
Da Ma (@k82cn)

Huawei Expert

About The SPEAKER

Da Ma Software Architect

- Kubernetes SIG-Scheduling co-Leader
- Volcano & kube-batch creator
- Expert at Huawei (now)
- Ex-IBM Spectrum CE/L3 Team/Tech Lead
- Jilin University master’ s degree, majoring in grid computing and distributed system



Gaps for AI/DL, BigData and so on

Gaps

Job/Queue Management

- Queue status/configuration
- Hierarchical queue
- Job with multiple pod template
- Lifecycle management of Job, e.g. restart, suspend/resume
- Error Handling, e.g. restart job if pod failed (MPI, TFJob)
- Indexed Job
- Task dependency, e.g. Spark (executor/driver)
- Delay Pod Creation
- …

Runtime

- Singularity
- …

Scheduler

- Coscheduling
- Faire-share
- Queue
- Preemption/Reclaim
- Reserve/Backfill
- Topology (network, accelerator)
- …

Others

- Throughput
- Round-trip
- Data locality (Data Aware Scheduling)
- …

Volcano: A Kubernetes Native Batch System

Website: https://volcano.sh

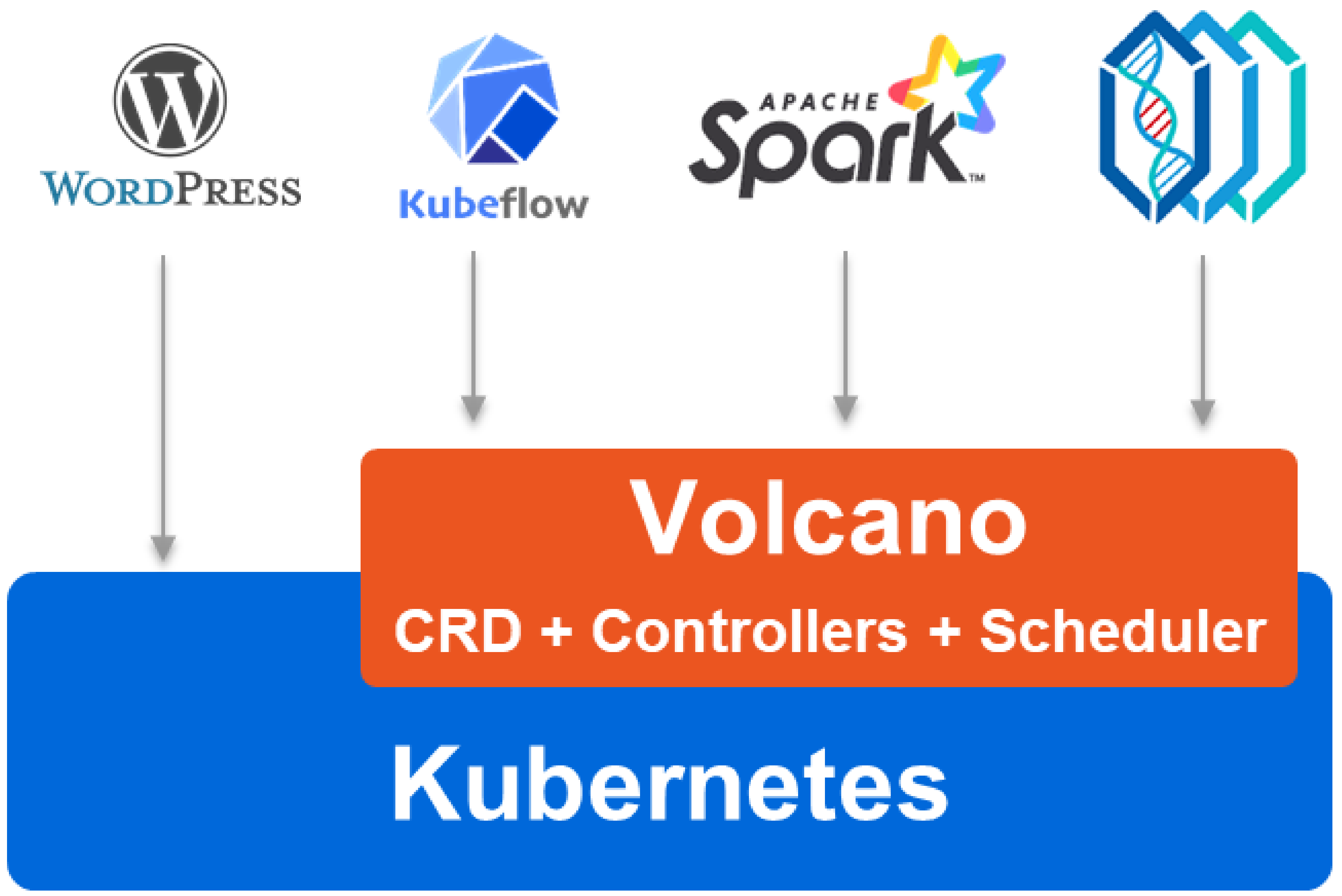
Github: http://github.com/volcano-sh/volcano

Twitter: https://twitter.com/volcano_sh

Slack: http://volcano-sh.slack.com

Email: volcano-sh@googlegroups.com

Overall Architecture



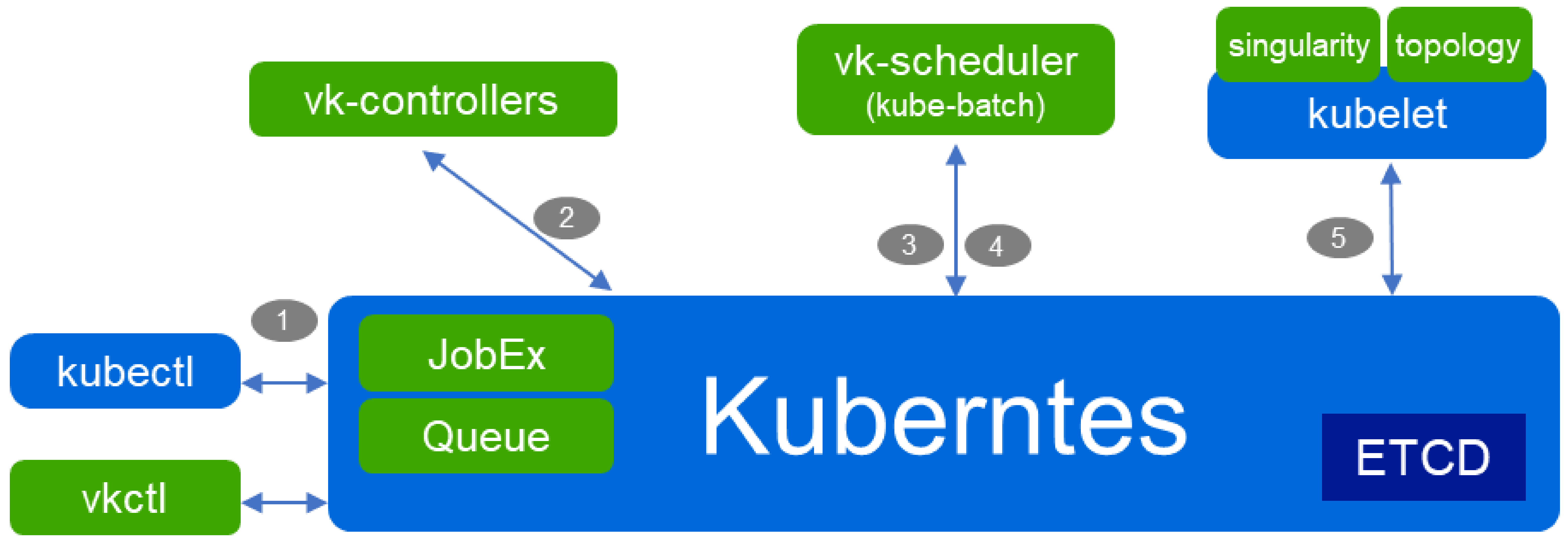
Domain frameworks:

- Deployment/Installation of framework in k8s
- Map framework's terms/concepts into common concept, e.g. Job, Queue
- Enable related features for frameworks, e.g. gang-scheduling for TensorFlow training

Common Service for high performance workload:

- Batch scheduling, e.g. fair-share, gang-scheduling
- Enhanced job management, e.g. multiple pod template, error handling
- Accelerator, e.g. GPU, FPGA
- kubectl plugins, e.g. show Job/Queue information

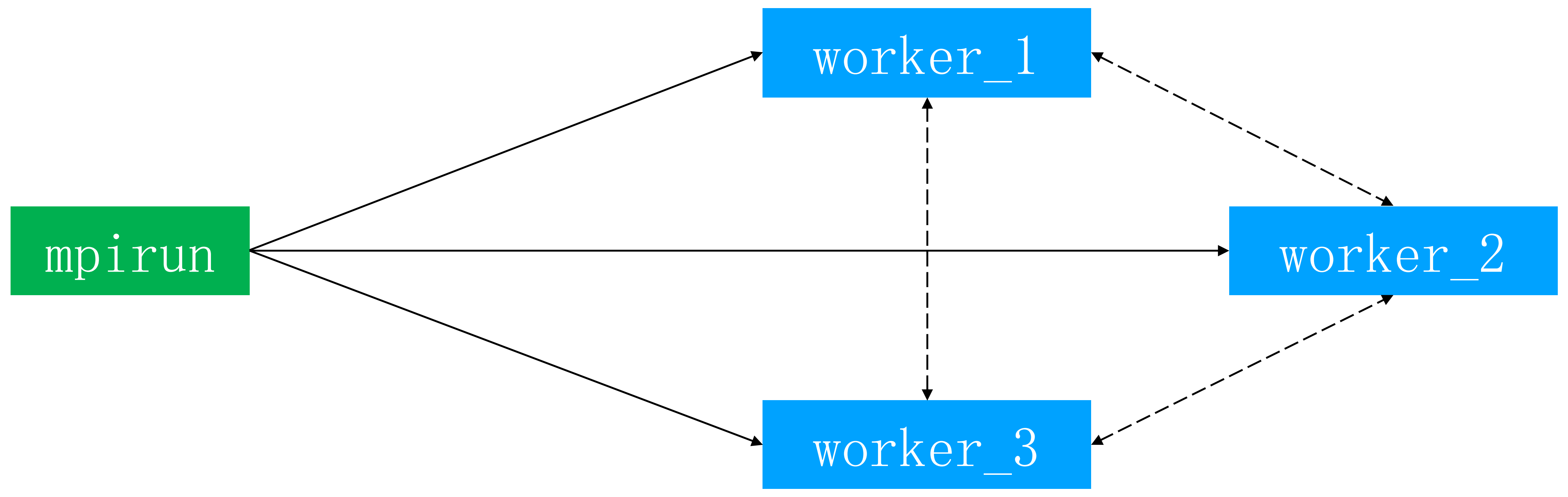
Overall Architecture

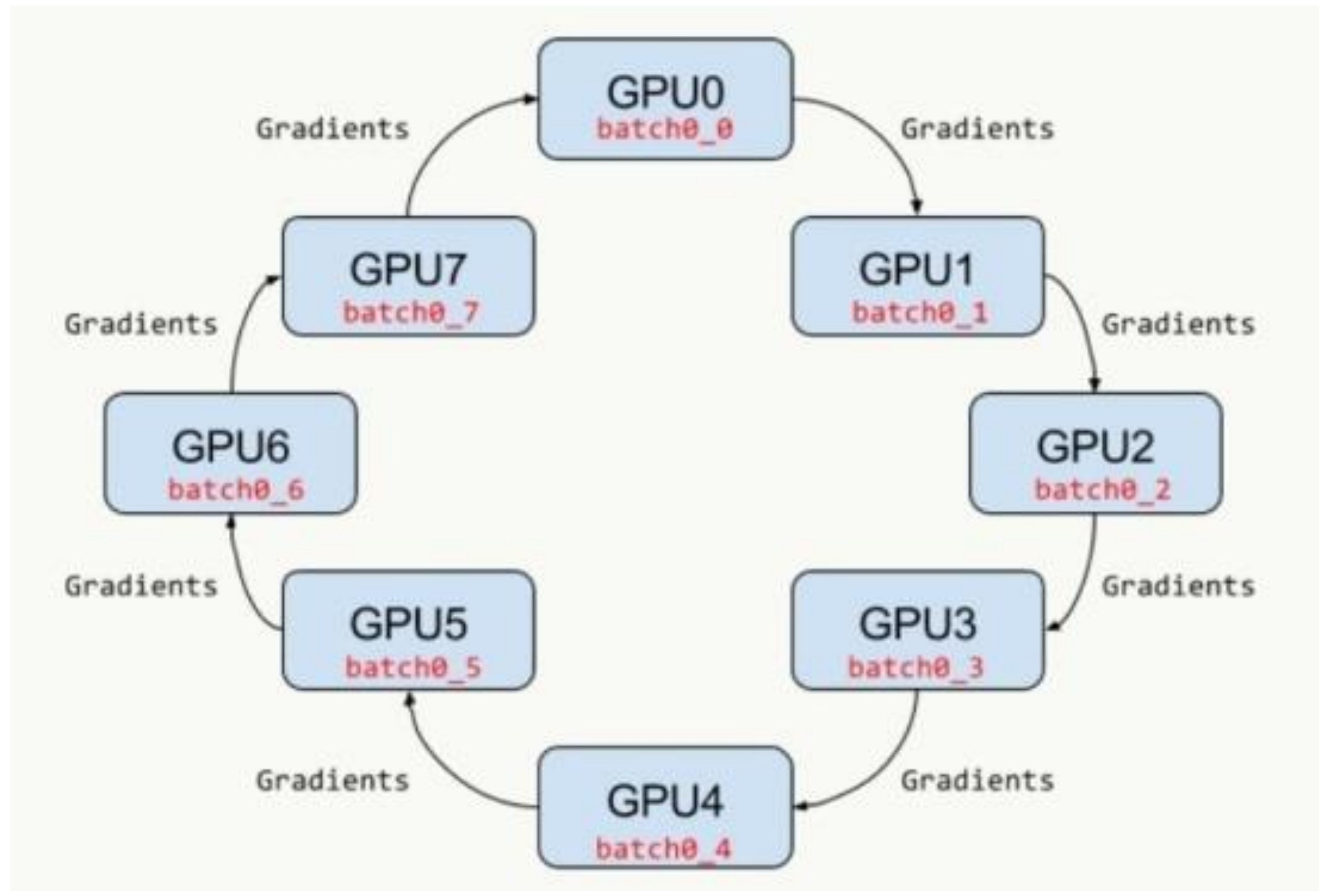


- The policy in **vk-scheduler** is pluggable, e.g. DRF, Priority, Gang
- **vk-controllers** includes **JobExController**, **QueueController**
- **Volcano handles high performance workload**

- Kubectl creates a ***JobEx*** object in apiserver if all admission passed
- ***JobExController*** *create Pods based on its replicas and templates*
- **vk-scheduler** get the notification of Pod from apiserver
- **vk-scheduler** chooses one host for the Pod of ***JobEx*** based on its policy
- kubelet gets the notification of Pod from apiserver; and then start the container

Scenarios: MPI

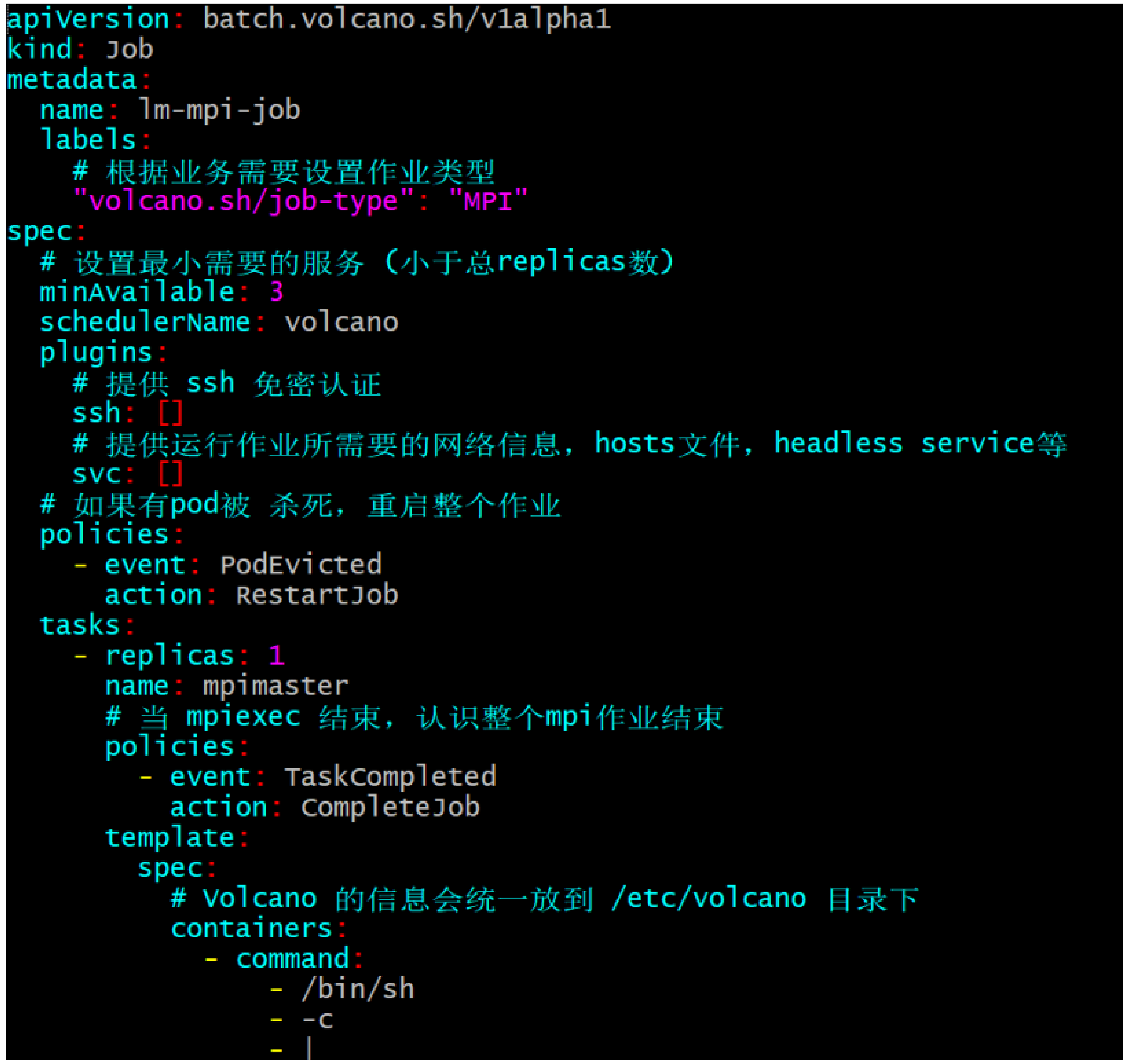




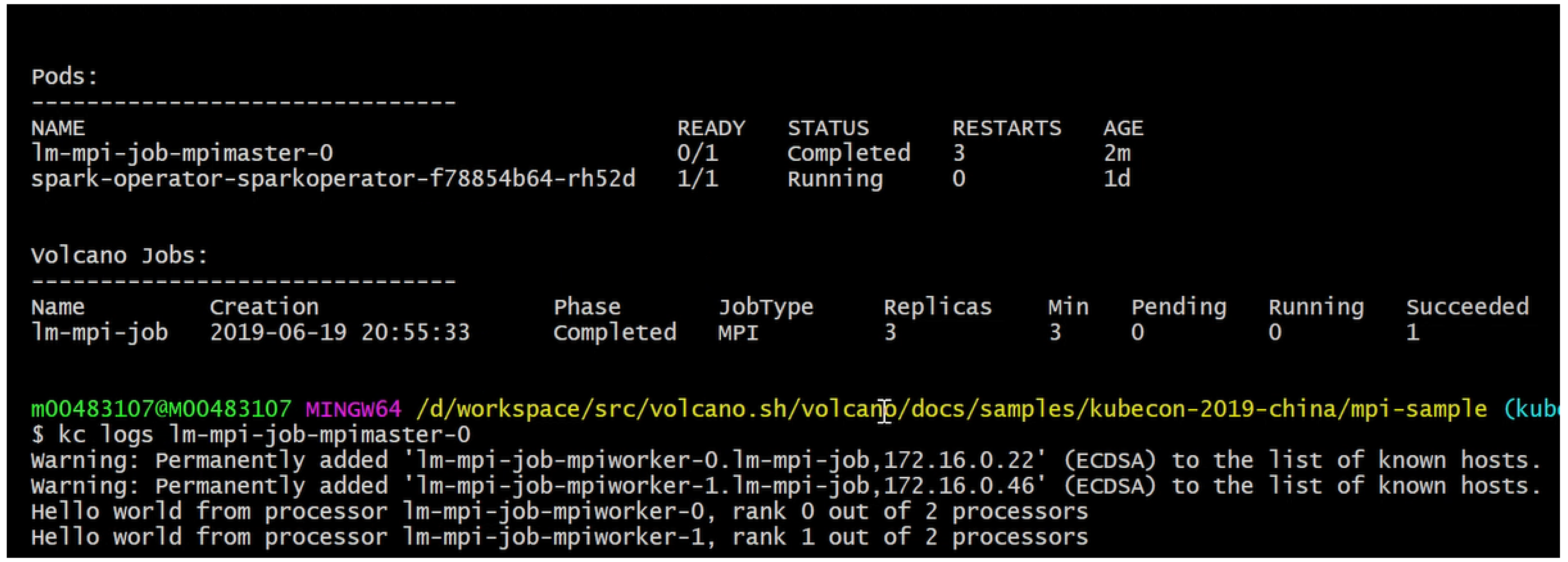
- Multiple Pod Template
- Lifecycle Policy
- Gang-scheduling

- ssh or kubectl
- Complete job when mpirun completed
- Headless service

Scenarios: MPI



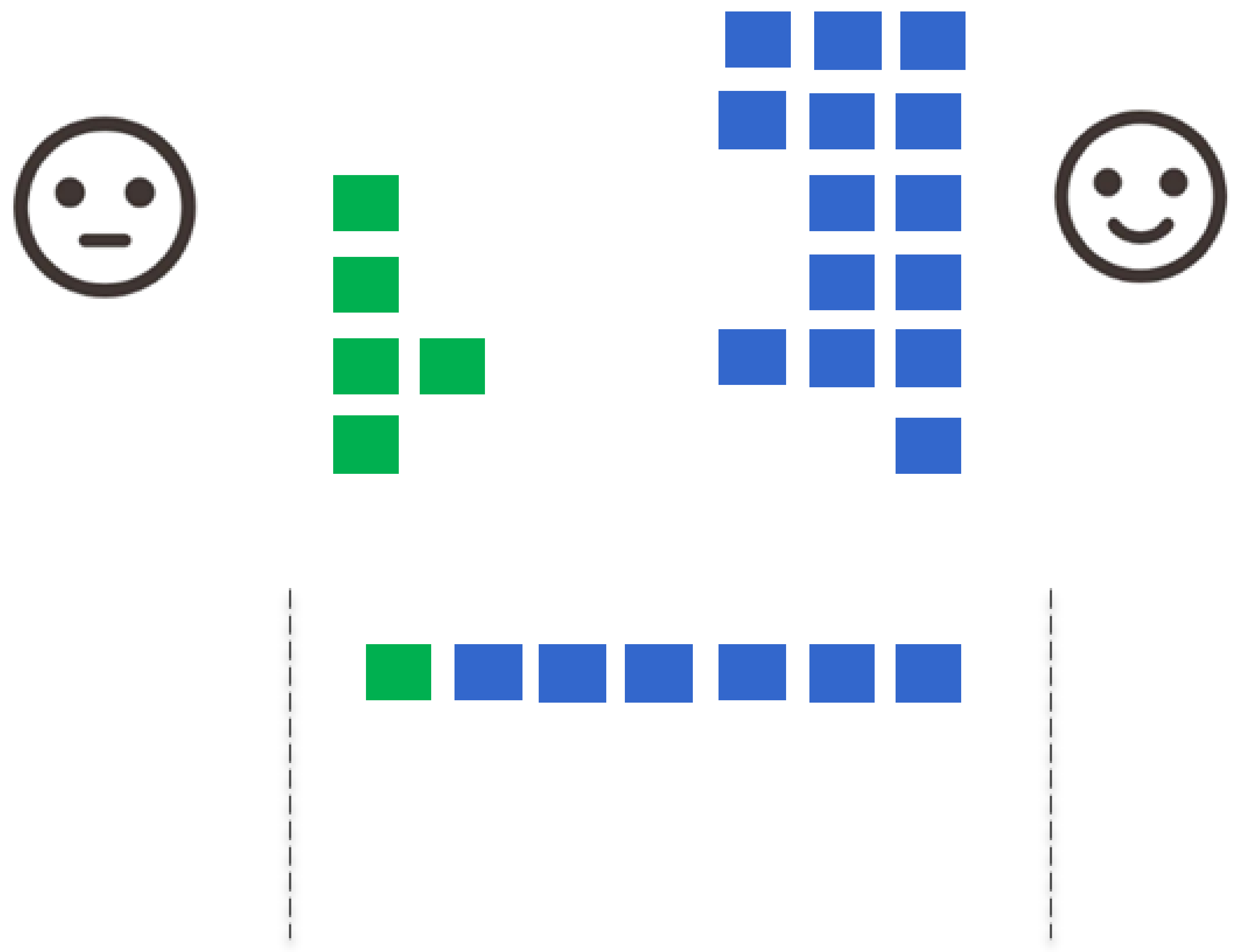
```yaml
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: lm-mpi-job
  labels:
    # 根据业务需要设置作业类型
    "volcano.sh/job-type": "MPI"
spec:
  # 设置最小需要的服务 (小于总replicas数)
  minAvailable: 3
  schedulerName: volcano
  plugins:
    # 提供 ssh 免密认证
    ssh: []
    # 提供运行作业所需要的网络信息, hosts文件, headless service等
    svc: []
  # 如果有pod被 杀死, 重启整个作业
  policies:
    - event: PodEvicted
      action: RestartJob
  tasks:
    - replicas: 1
      name: mpimaster
      # 当 mpiexec 结束, 认识整个mpi作业结束
      policies:
        - event: TaskCompleted
          action: CompleteJob
      template:
        spec:
          # Volcano 的信息会统一放到 /etc/volcano 目录下
          containers:
            - command:
                - /bin/sh
                - -c
                - |
```



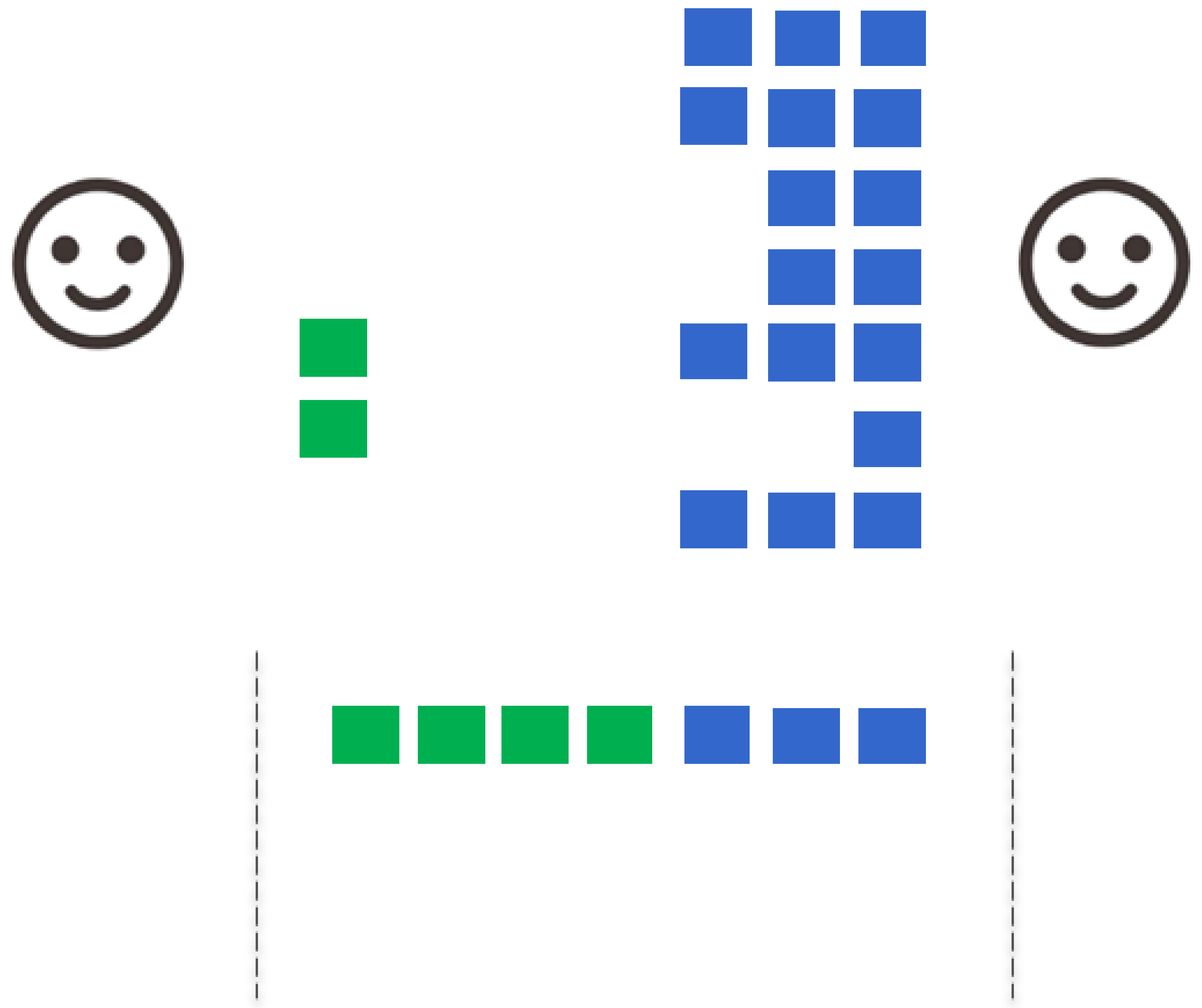
```
Pods:
--------------------------------
NAME                                       READY   STATUS      RESTARTS   AGE
lm-mpi-job-mpimaster-0                     0/1     Completed   3          2m
spark-operator-sparkoperator-f78854b64-rh52d   1/1     Running     0          1d


Volcano Jobs:
--------------------------------
Name         Creation              Phase       JobType     Replicas    Min   Pending    Running    Succeeded
lm-mpi-job   2019-06-19 20:55:33   Completed   MPI         3           3     0          0          1

m00483107@M00483107 MINGW64 /d/workspace/src/volcano.sh/volcano/docs/samples/kubecon-2019-china/mpi-sample (kub
$ kc logs lm-mpi-job-mpimaster-0
Warning: Permanently added 'lm-mpi-job-mpiworker-0.lm-mpi-job,172.16.0.22' (ECDSA) to the list of known hosts.
Warning: Permanently added 'lm-mpi-job-mpiworker-1.lm-mpi-job,172.16.0.46' (ECDSA) to the list of known hosts.
Hello world from processor lm-mpi-job-mpiworker-0, rank 0 out of 2 processors
Hello world from processor lm-mpi-job-mpiworker-1, rank 1 out of 2 processors
```

- The workers are deleted by job controller because of sshd
- The pod of mpiexec/mpirun will not be deleted for output
- The pod of mpiexec/mpirun may restart few times because of network setup delay

Scenarios: Faire Share

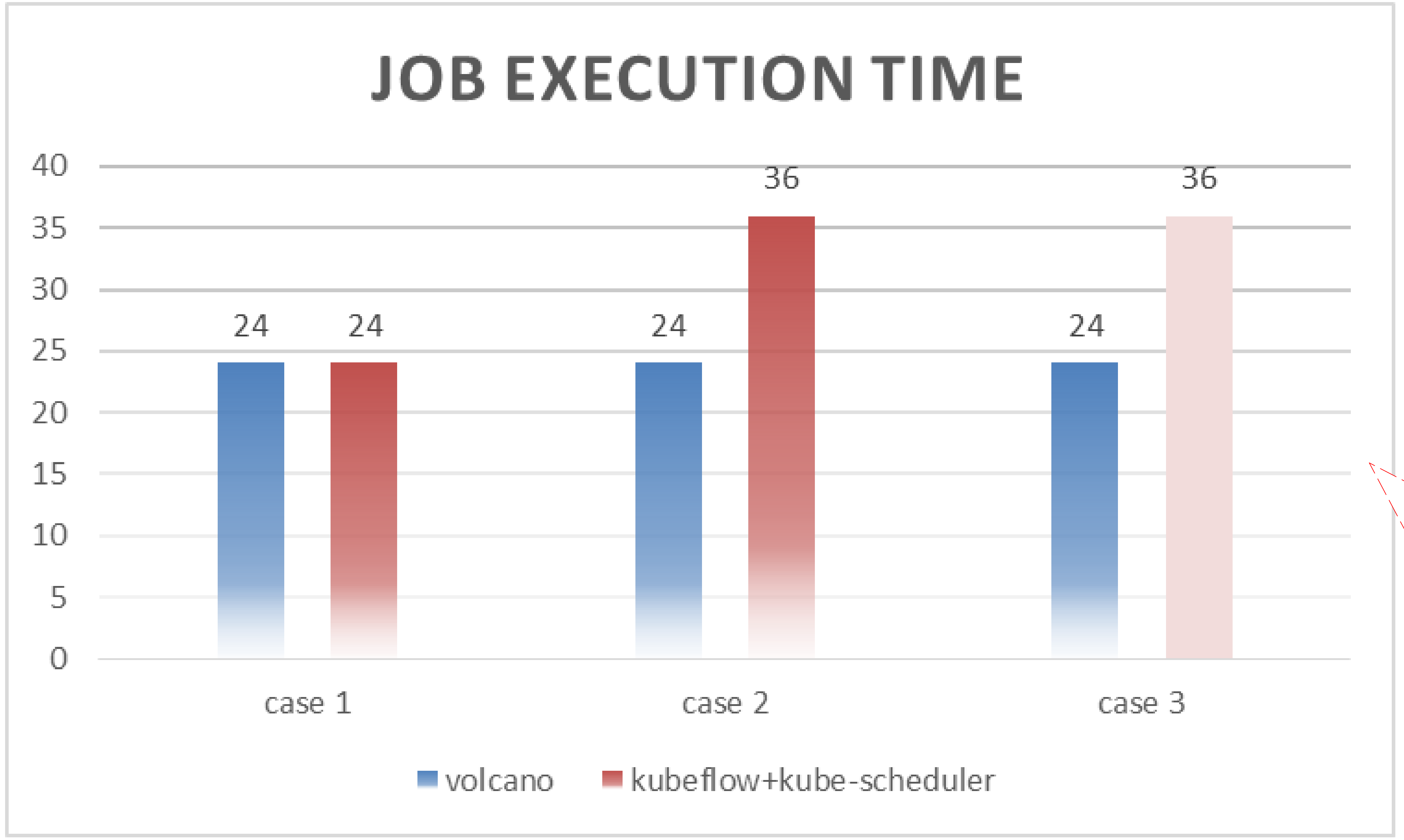


The more workload, the more resources???



Share resources by weight !!!

Gang-scheduling: Job Execution Time

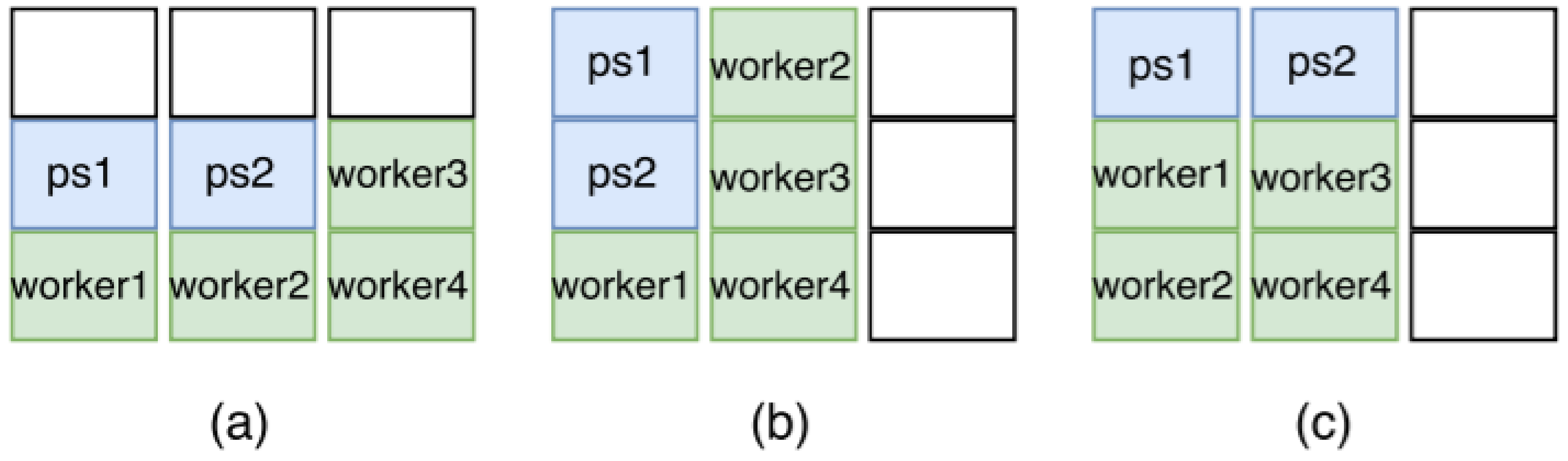


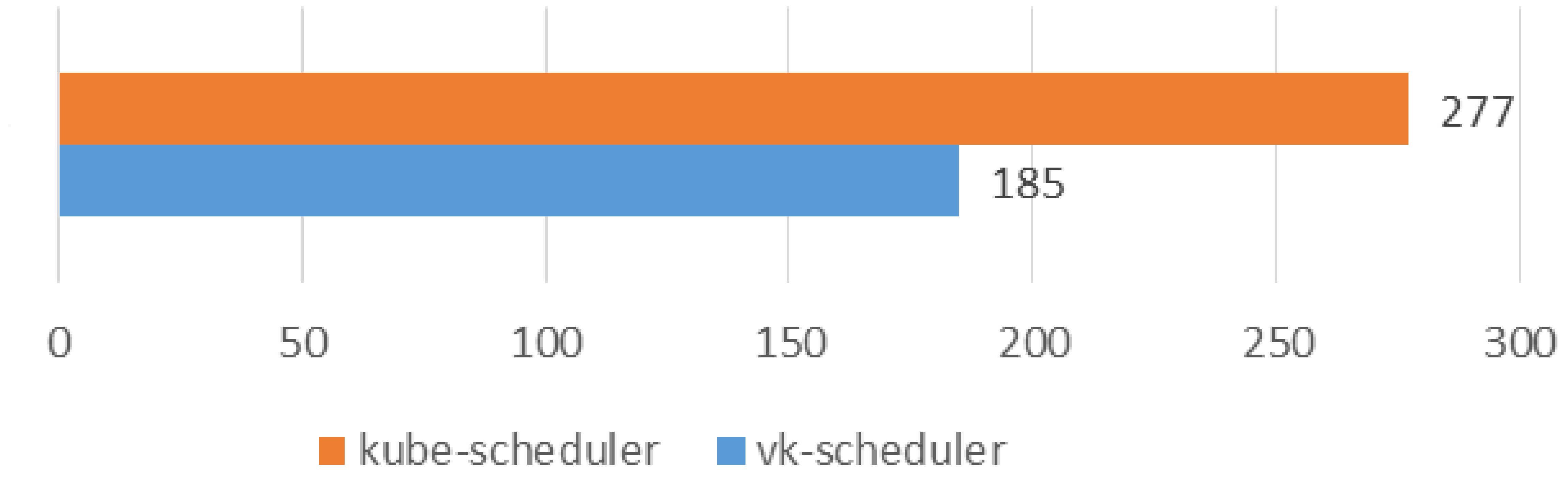
- Case 1: 1 job with 2ps + 4workers
- **Case 2: 2 jobs with 2ps + 4workers**
- **Case 3: 5 jobs with 2ps + 4workers**

- No enough resource for 2 Jobs to run concurrently; one of them **wasting** resources without Gang-Scheduling !
- 2 of 5 jobs was finished because of deadlock (+20 hours)

http://status.openlabtesting.org/builds?project=theopenlab%2Fvolcano

Task-Topology + Binpack





- The execution time of 3 jobs in total; 2ps + 4workers for each job
- The execution time is unstable when tested by default scheduler
- The improvement dependent on data exchanges between pods
- Task-topology within a Job also improved scheduler's performance
- **Open Source** at volcano-sh/volcano#272

Reference: "Optimus: An Efficient Dynamic Resource Scheduler for Deep Learning Clusters"

Integrations

Framework	Status	API
MPI	Done	Volcano Job
Horovod	Done	Volcano Job
Kubeflow/tf-operator	Done	PodGroup
Kubeflow/arena	Done	Volcano Job
Spark-Operator	On-going	PodGroup
Cromwell	On-going	Volcano Job
PaddlePaddle	On-going	Volcano Job
…	On-going	Volcano Job / PodGroup

Pipeline

- GPU Share/Topology
- Job Management
- Queue Management
- Hierarchical Queue
- Preemption/Reclaim
- ……

Call for Contribution