Volcano: A Kubernetes Native Batch System

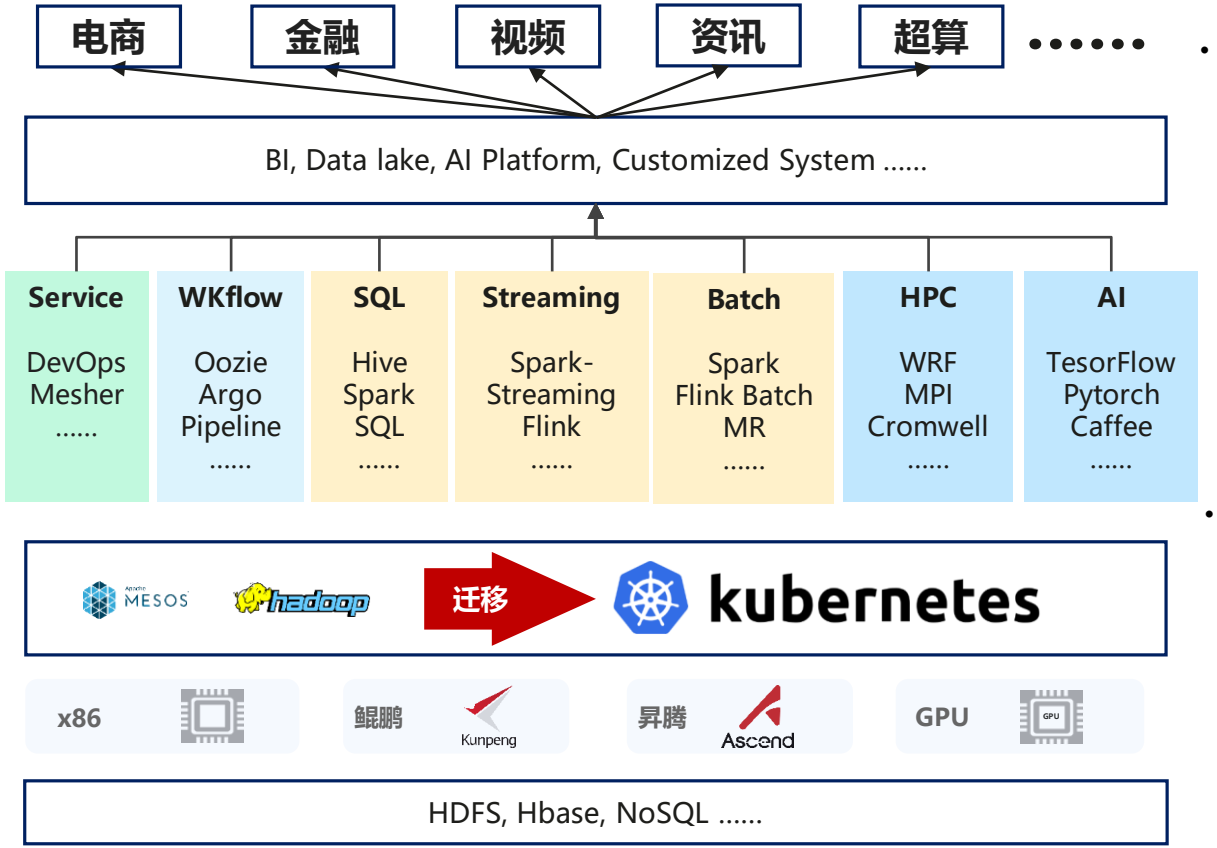
Klaus Ma

@k82cn

github.com/volcano-sh/volcano

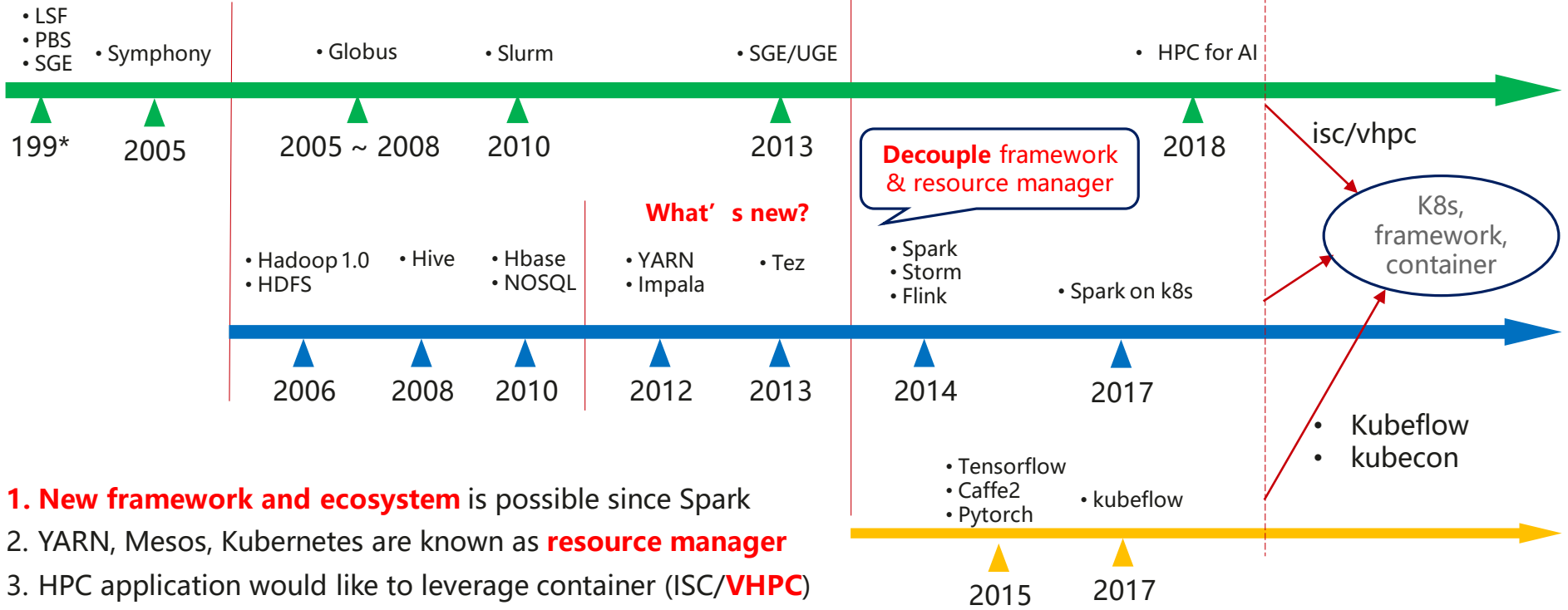
2018 ~ 2020

多领域框架 + 统一资源管理



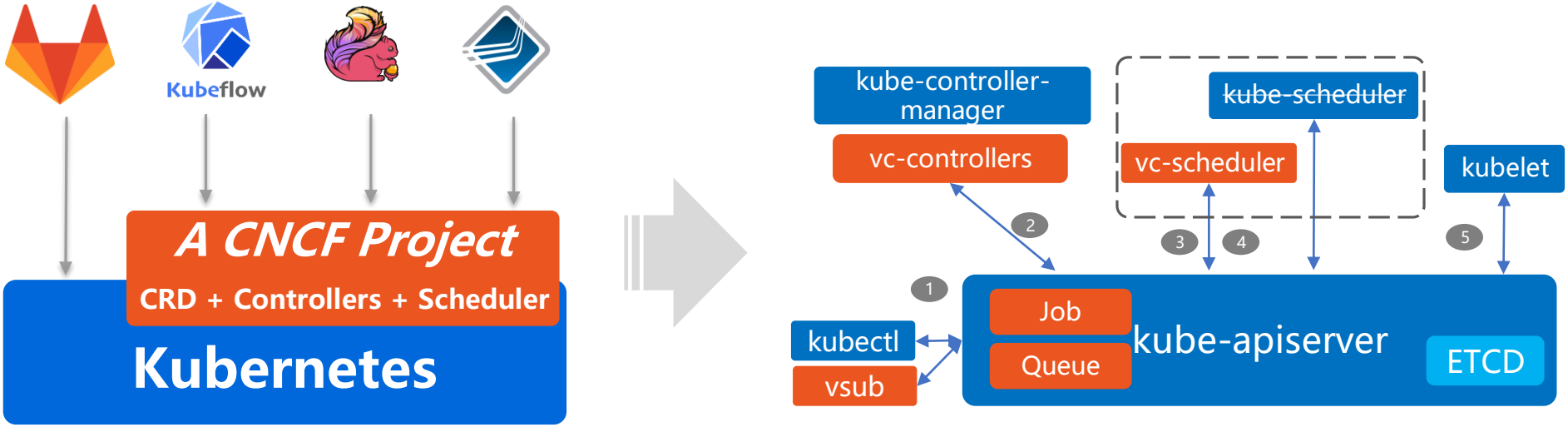
- **业务复杂性驱动统一资源管理**
  - ✓ 随着**业务复杂性**的增加，单一的框架、技术很难支撑业务的发展
  - ✓ 基于不同技术栈的资源池增加了**资源成本，管理、运维成本**
  - ✓ 不同技术栈增加了**技术成本**，e.g. YARN和K8s经验无法互通
- **云原生促进统一资源管理落地**
  - ✓ 云原生(CNCF)提供了**活跃的社区和丰富的生态系统**，支持大数据、AI等多种负载；促使用户向云原生技术迁移
  - ✓ Kubernetes提供了标准的生态、扩展接口，无缝接入**网络、存储、调度**等能力
  - ✓ 各硬件厂商提供对Kubernetes的支持，例如 **GPU，ARM, IB** 等

Kubernetes + 多领域框架



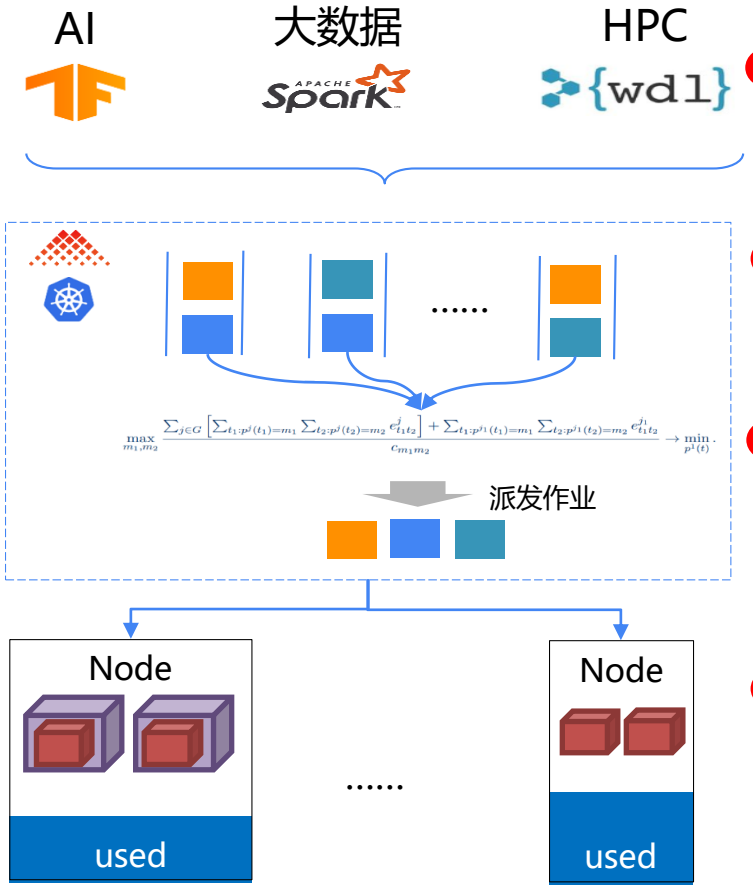
1. **New framework and ecosystem** is possible since Spark
2. YARN, Mesos, Kubernetes are known as **resource manager**
3. HPC application would like to leverage container (ISC/**VHPC**)
4. New bigdata framework, e.g. Spark, start to support kubernetes
5. Most AI application are built on Kubernetes natively

Volcano: A K8S Native Batch System

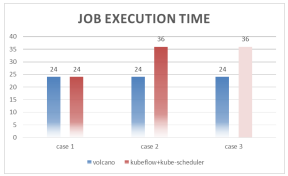


- **挑战 1**: 面向高性能负载的调度策略, e.g. fair-share, gang-scheduling
- **挑战 2**: 支持多种作业生命周期管理, e.g. multiple pod template, error handling
- **挑战 3**: 支持多种异构硬件, e.g. GPU, FPGA
- **挑战 4**: 面向高性能负载的性能优化， e.g. scalability, throughput, network, container runtime

Overall Architecture

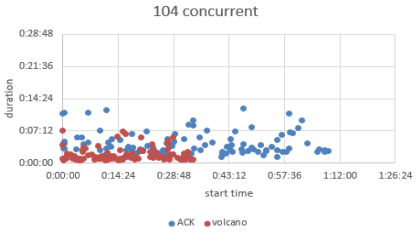


1. 支持多种类型作业混合部署
2. 支持**多队列**用于多租户资源共享，**资源规划**；并**分时复用**资源
3. 支持**多种高级调度策略**，有效提升整集群资源利用率
4. 支持**资源实时监控**，用于高精度资源调度，例如 热点，网络带宽
   - 容器引擎，网络性能优化, e.g. 免加载



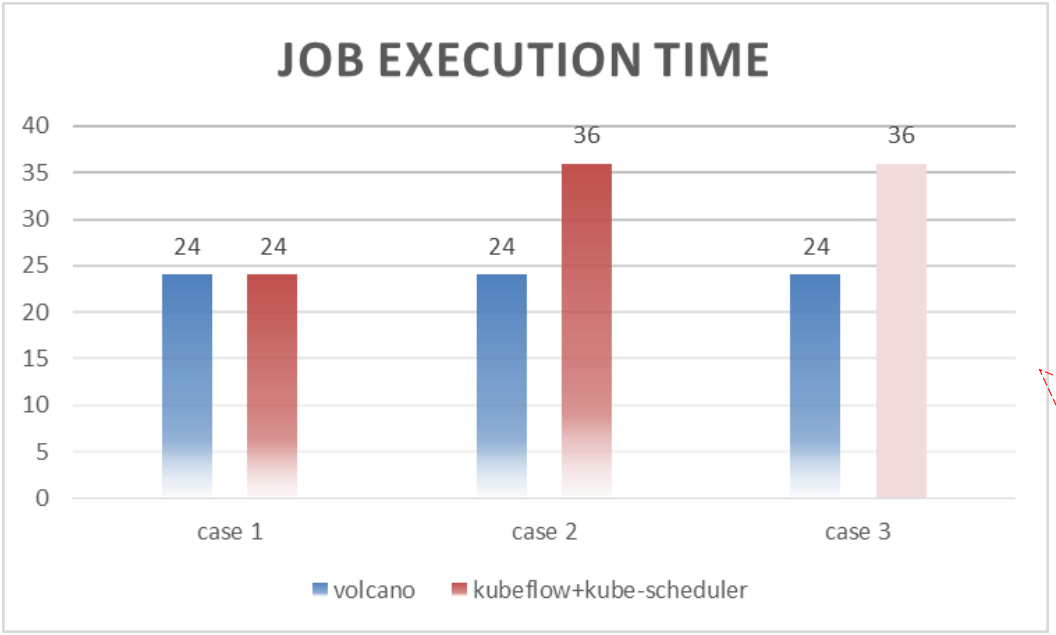
- 相比默认调度器，**AI训练作业性能提升 30%**
- Gang-scheduling同时解决了多作业死锁的情况

- 相比默认调度器，**AI训练作业性能提升 31%**
- 测试结果为 3 个作业的总执行时间; 每个作业包含 2ps + 4workers



- 相比默认调度器，**Spark作业性能提升 51%**
- 测试结果为 104 个 TP-DCS 查询语句

Tensorflow on Volcano (Gang)

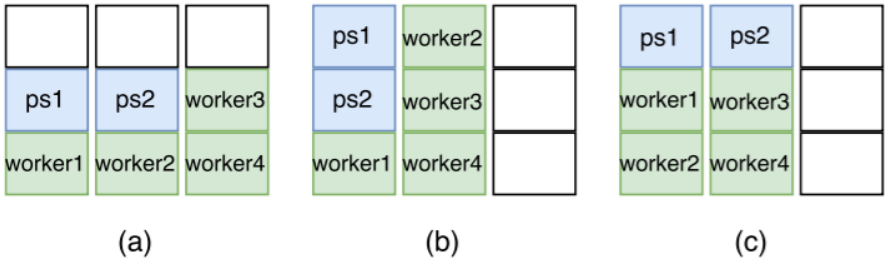


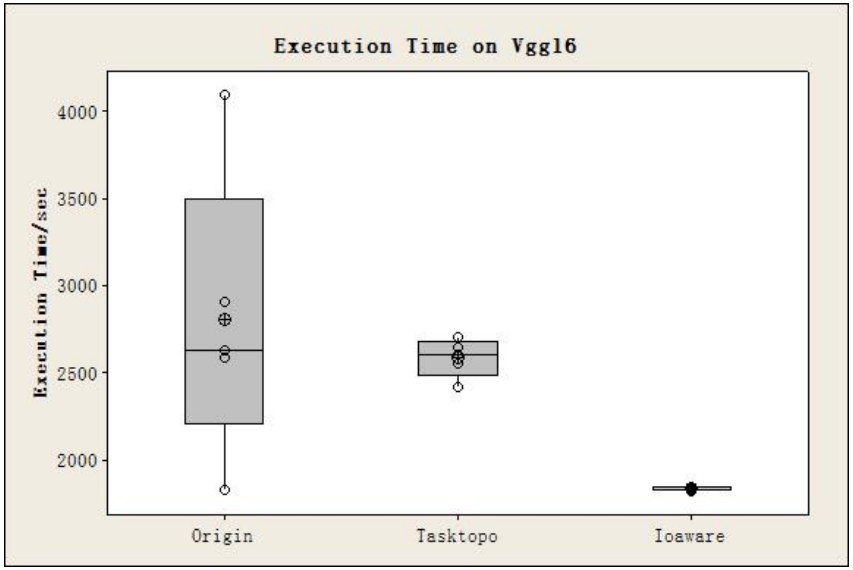
- Case 1: 1 job with 2ps + 4workers
- Case 2: 2 jobs with 2ps + 4workers
- Case 3: 5 jobs with 2ps + 4workers

- 没有足够的资源同时运行两个作业；如果没有 gang-scheduling，其中的一个作业会出现**忙等**！
- 当作业数涨到5后，很大概率出现**死锁**；一般只能完成2个作业

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

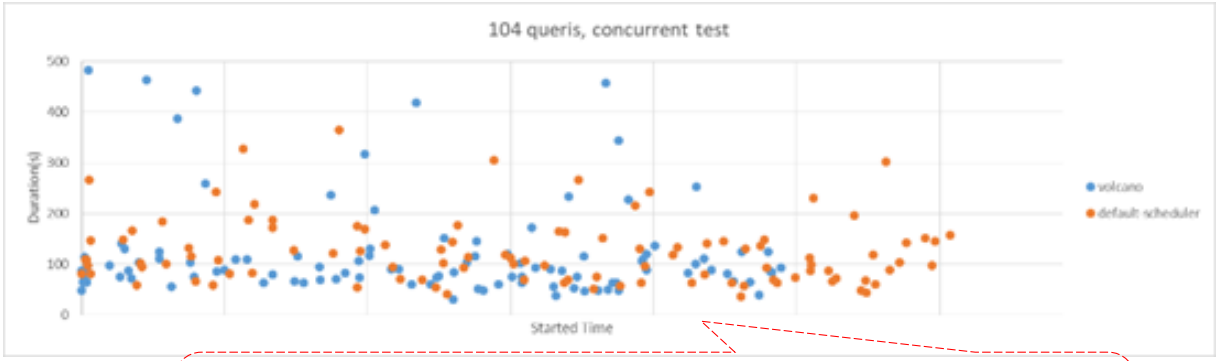
Tensorflow on Volcano (IOAware)





- 3个作业的执行时间总和; 每个作业带2ps + 4workers
- 默认调度器执行时间波动较大
- 执行时间的提高量依据数据在作业中的比例而定
- 减少 Pod Affinity/Anti-Affinity，提高调度器的整体性能

Spark on Volcano



- Spark-sql-perf (TP-DCS, master)
- 104 queries concurrently
- (8cpu, 64G, 1600SSD) * 4nodes
- Kubernetes 1.13
- Driver: 1cpu,4G; Executor: (1cpu,4G)*5

- 如果没有固定的driver节点，**最多同时运行 26 条查询语句**
- 由于Volcano提供了作业级的资源预留，总体性能提高了 **~30%**

- Volcano (min-res): 3.3cpu, 12G
- Kubernetes: 1 node for drivers

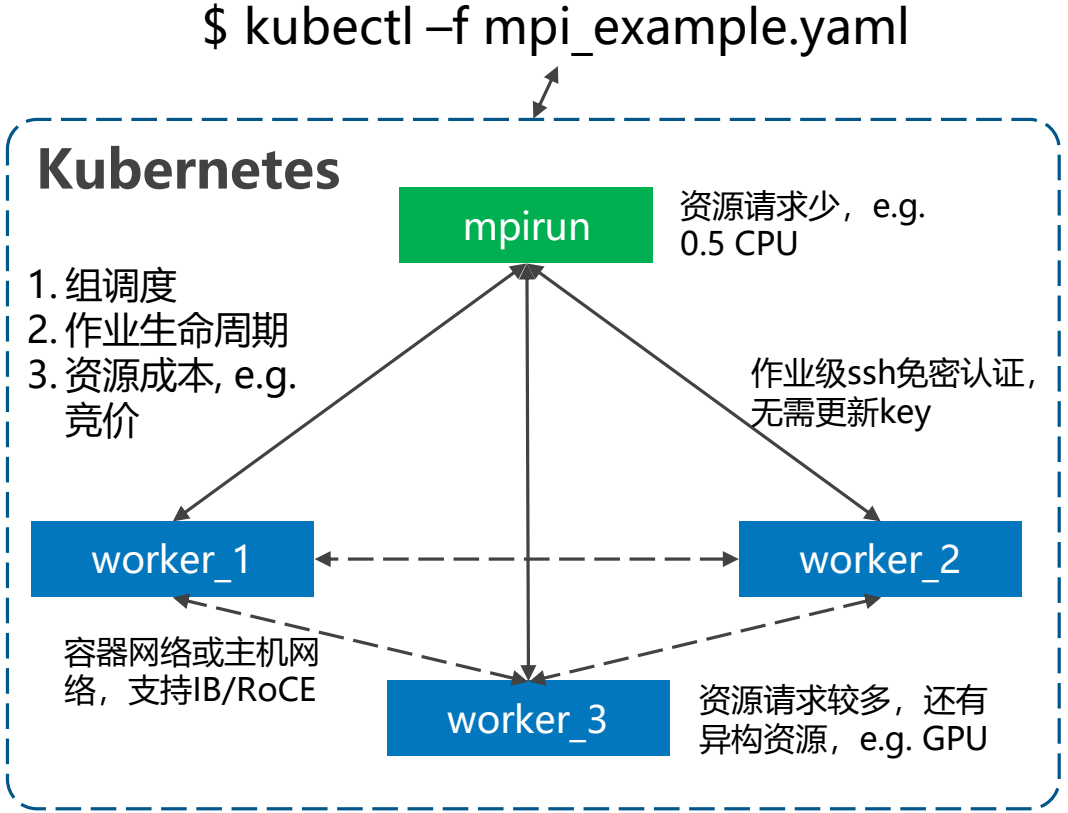
MPI on Volcano

```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
```

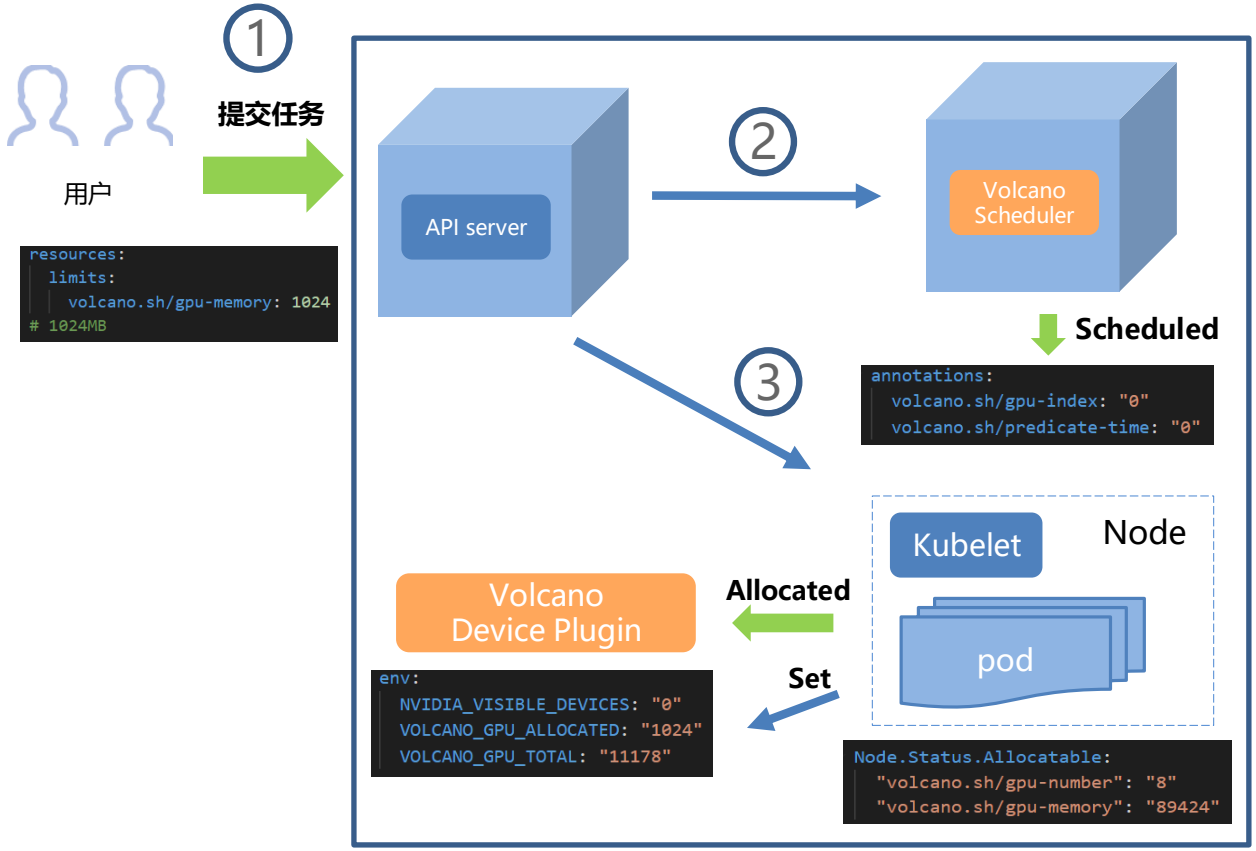
```
Pods:
------------------------------
NAME                                        READY   STATUS      RESTARTS   AGE
lm-mpi-job-mpimaster-0                      0/1     Completed   3          2m
spark-operator-sparkoperator-f78854b64-rh52d   1/1     Running     0          1d


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
```

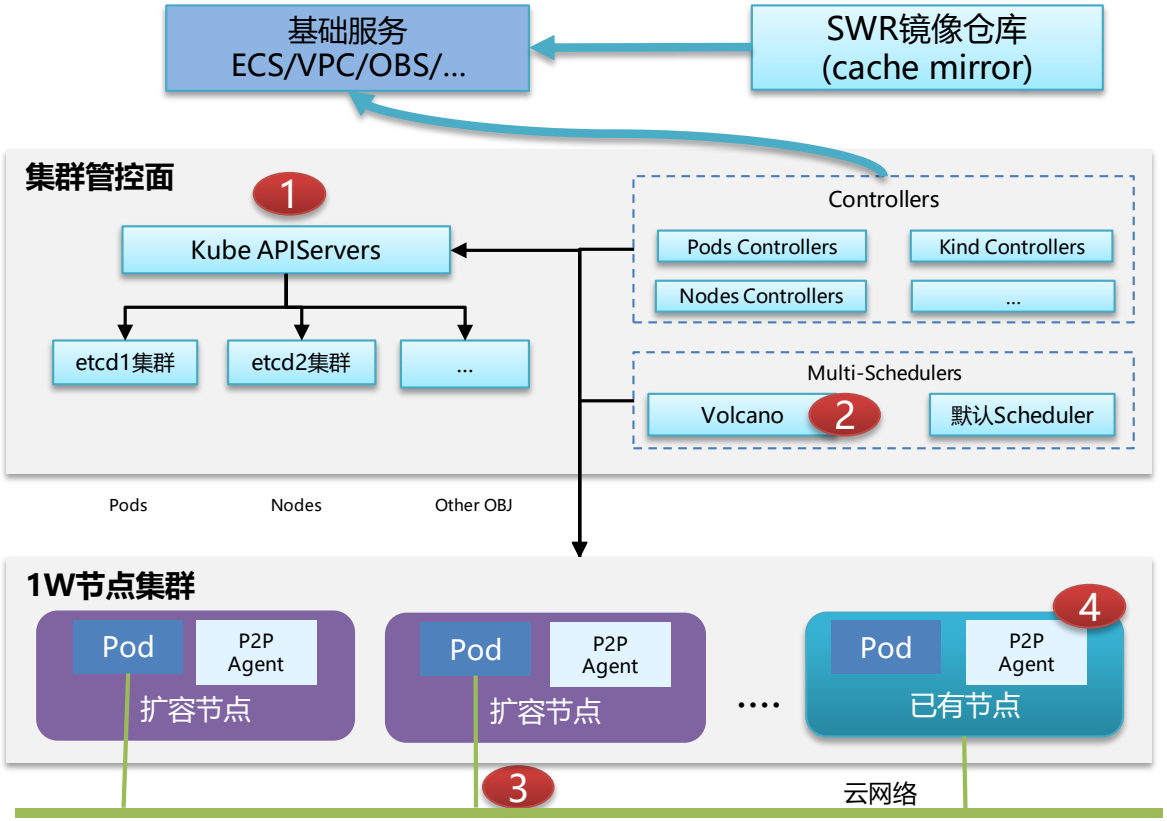


GPU Sharing



- **算力优化：**
  - GPU硬件加速，TensorCore
  - GPU共享
  - 昇腾改造
- **调度算法优化：**
  - Job/Task模型，提供AI类Job统一批量调度
  - 多任务排队，支持多租户/部门共享集群
  - 单Job内多任务集群中最优化亲和性调度、Gang Scheduling等
  - 主流的PS-Worker、Ring AllReduce等分布式训练模型
- **流程优化**
  - 容器镜像
  - CICD流程
  - 日志监控

一万节点，百万容器；2000 Pods/s



① 编排：

- **Etcd 分库分表**，e.g. Event 放到单独库，wal/snapshot 单独挂盘
- 通过一致性哈希分散处理，实现 **controller-manager 多活**
- Kube-apiserver 基于工作负载的弹性扩容

② 调度：

- 通过 **EquivalenceCache，算法剪枝** 等技术提升单调度器的吞吐性能
- 通过共享资源视图实现**调度器多活**，提升调度速率

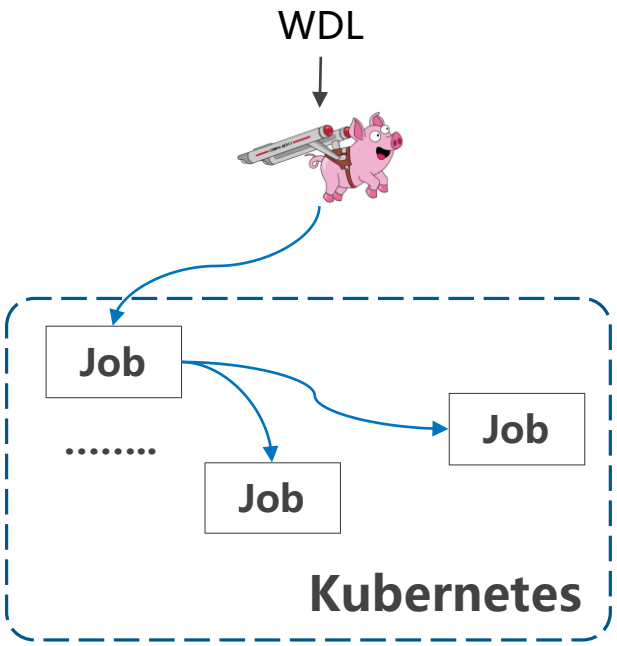
③ 网络：

- 通过trunkport提升单节点容器密度及单集群ENI容量
- 通过 Warm Pool 预申请网口，提升网口发放速度
- 基于eBPF/XDP 支持大规模、高度变化的云原生应用网络，e.g. Service, network policy

④ 引擎：

- containerd 并发 启动优化
- 支持shimv2，提升单节点容器密度
- 镜像下载加速 Lazy loading

Cromwell with Volcano




```
Volcano {
  actor-factory = "cromwell.backend.impl.sfs.config.ConfigBackendLifecycleActorFactory"
  config {
    runtime-attributes = """
    Int runtime_minutes = 600
    Int cpus = 2
    Int requested_memory_mb_per_core = 8000
    String queue = "short"
    """

    submit = """
        vcctl job run -f ${script}
    """
    kill = "vcctl job delete -N ${job_id}"
    check-alive = "vcctl job view -N ${job_id}"
    job-id-regex = "(\\d+)"
  }
}
```

1. Cromwell 社区原生支持Volcano
2. 企业版已经上线 华为云 GCS
3. 通过 cromwell 支持作业依赖
4. Volcano 提供面向作业、数据依赖的调度

https://cromwell.readthedocs.io/en/stable/backends/Volcano/

Volcano CLI

常用命令行：

- vsub
  提交作业
- vcancel
  取消作业
- vsuspend
  挂起作业
- vresume
  重启作业
- vjobs
  查询作业状态及列表
- vqueues
  查询队列状态及列表
- vadmin
  管理员命令行，e.g. 创建队列

开发库 (SDK)：

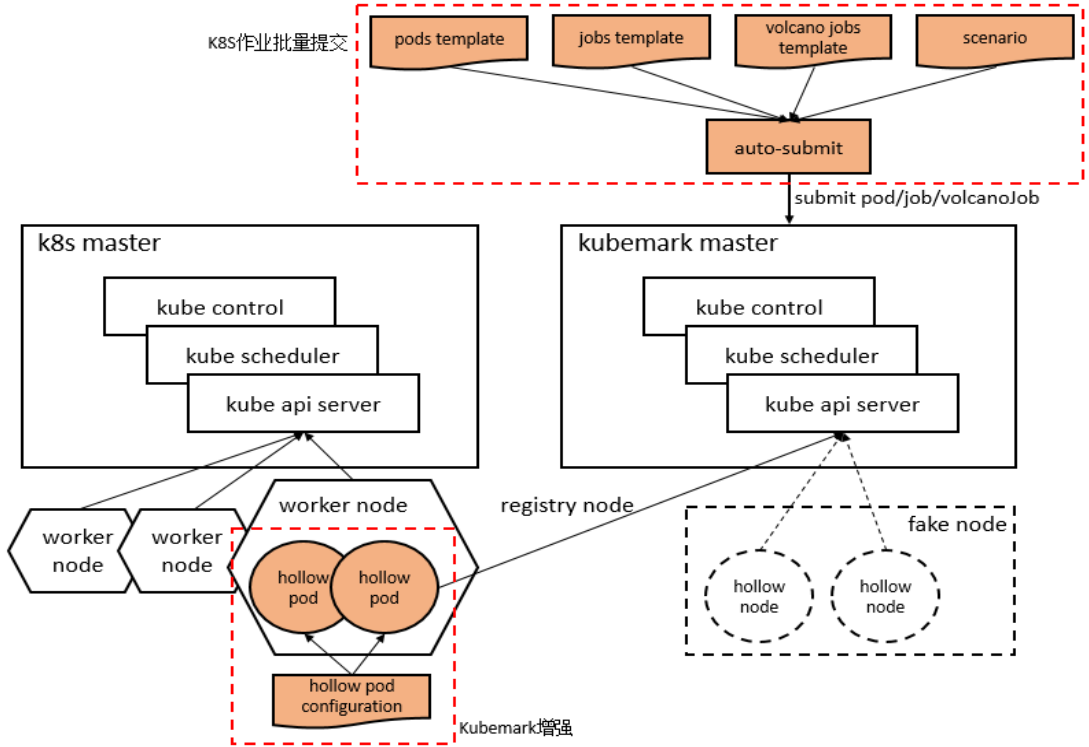
官方支持:

- Go
- Python
- Java
- CSharp
- Javascript
- Haskell

社区支持：

- Node.js
- Rust
- Scala
- ……

KubeSim



简介：

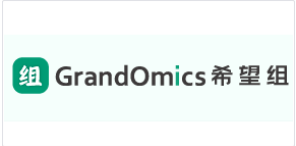
- 集群进行性能测试及调度的描述工具
- 不受资源限制，模拟大规模K8S集群
- 完整的K8S API调用，不会真正创建pod
- 已经支持产品侧大规模专项及调度专项的模拟工作

总体结构：

- Worker cluster：承载kubemark虚拟节点，hollow pod
- Master cluster：管理kubemark虚拟节点，hollow node
- Hollow pod = hollow kubelet + hollow proxy

https://github.com/volcano-sh/kubesim

Adopters & Contributors



https://github.com/volcano-sh/volcano/blob/master/docs/community/adopters.md

Community

- CNCF Sandbox: https://landscape.cncf.io/selected=volcano
- 1360+ GitHub Stars
- ~100 Contributors from Huawei, **AWS**, JD.com, **OpenAI**, Baidu, Tencent, etc.
- **12 formal releases**
- 5 maintainers across 3 independent companies (Huawei: 3, IBM: 1, Tencent 1)
- 14 public adopters (private not included)
- CII best practice: passing
- License: Apache 2 License


Releases

Volcano: A Kubernetes Native Batch System

v0.1:
- IndexedJob
- Multiple Pod template
- Error handling of Pod/Job
- Queue/Job command line
- Delay Pod Creation
- Job plugins

2019.5

v0.2:
- Job Priority
- Queue Capacity
- Fair-share of NS cross Queue
- Binpack algorithm
- Multiple event in job lifecycle policy

2019.9

v0.3:
- Add maxRetry in job controller
- Bug fix of scheduler, e.g. Resource, callback
- Dynamic load scheduler configuration
- Support network policy
- Command line enhancements

2020.1

v0.4:
- Upgrade to go mode
- sign certificate by openssl
- Remove deprecated API versions
- Bug fix

2020.4

v1.0:
- GPU Sharing
- Preempt and reclaim upgrade to Beta
- Job dynamic scale up and down
- Integrate with flink operator
- DAG job based on Argo

2020.7

Thanks!

Further reading:

- https://groups.google.com/forum/#!forum/volcano-sh
- https://slack.cncf.io/ #volcano
- https//github.com/volcano-sh/volcano

Join our community and give us feedback!

Performance Index

名字	英文缩写	描述	计算公式	衡量标准	备注
往返速率	Roundtrip	测试为完成作业，在平台上消耗的时间	作业完成时间 – 作业提交时间	越少越好	作业为空，例如 busybox；并以客户端 收到反馈的时间为准
吞吐率	Throughput	测试系统单位时间内可以接收的任务	单位时间内接收的任务	越多越好	需要保证系统长期稳定的情况下进行测试
作业平均执行时间	AET	测试作业在系统中的平均执行时间	AVG(作业完成时间 – 作业开始执行时间)	越少越好	测试时需要指定测试场景，例如 Terasort, HPL等
作业平均等待时间	AWT	测试作业在系统中的平均等待时间	AVG(作业提交时间 – 作业开始执行时间)	越短越好	根据队列权重进行调整
资源使用率	Utilization	测试集群中资源的实际使用情况	实际使用资源/集群总资源	越接近1越好	
资源分配率	Allocation	测试集群中资源的分配情况	分配资源/集群总资源	越接近1越好	
用户资源分配率	U-Allocation	测试各个用户的资源分配情况	DIV(用户分配到的资源)	用户之间越接近越好	包括单个调度周期和某一时间段的资源分配情况，根据权重调整
资源成本	Cost	完成指定作业集所需要的总成本	SUM(作业执行时间 * 资源成本)	越少越好	资源成本会有不同，比如 竞价实例
用户资源成本	U-Cost	每个用户完成指定作业所需要的成本	DIV(每个用户的作业执行时间 * 资源成本)	• 越少越好 • 用户之间越接近越好	根据用户权重调整

Da (Klaus) Ma

- *Tech Lead of Huawei Container Infra*
- *Founder of Volcano/kube-batch*
- *Tech Lead of **CNCF SIG-Runtime***
- *Tech Lead of **CNCF Research User Group***
- *Emeritus Leads of Kubernetes Machine Learning Working Group*
- *Emeritus Leads of Kubernetes SIG Scheduling*
- *Kubernetes Maintainer*
- *Technical Program Committee of **ISC/VHPC'20/21***

@k82cn