



Is Kubernetes ready for statefulset workloads?





Kelsey Hightower ✓

@kelseyhightower

Following



Kubernetes has made huge improvements in the ability to run stateful workloads including databases and message queues, but I still prefer not to run them on Kubernetes.

6:04 AM - 13 Feb 2018

296 Retweets 663 Likes



- Kubernetes为构建有状态的应用提供了哪些资源？
- 基于这些资源到底能不能将有状态应用（如数据库）的运行到 kubernetes？

Contents



- What kubernetes offer?
- How to build statefulset workloads like database?
- The problems we are facing and thinking

What kubernetes offer?



Persistent Volume

- PV
 - 持久化
 - VolumePlugin
- PVC
 - 解耦存储细节
- StorageClass
 - 定义不同规格的存储池
- provisioner
 - 扩展存储类型

```
apiVersion: v1
kind: Pod
metadata:
  name: mysql
spec:
  volumes:
    - name: mysql-data
      gcePersistentDisk:
        pdName: mysql
        fsType: nfs4
  containers:
    - image: mysql
      name: mysql
      volumeMounts:
        - name: mysql-data
          mountPath: /var/lib/mysql
```

```
apiVersion: v1
kind: Pod
metadata:
  name: mysql
spec:
  volumes:
    - name: mysql-data
      persistentVolumeClaim:
        claimName: mysql-pvc
  containers:
    - image: mysql
      name: mysql
      volumeMounts:
        - name: mysql-data
          mountPath: /var/lib/mysql
```

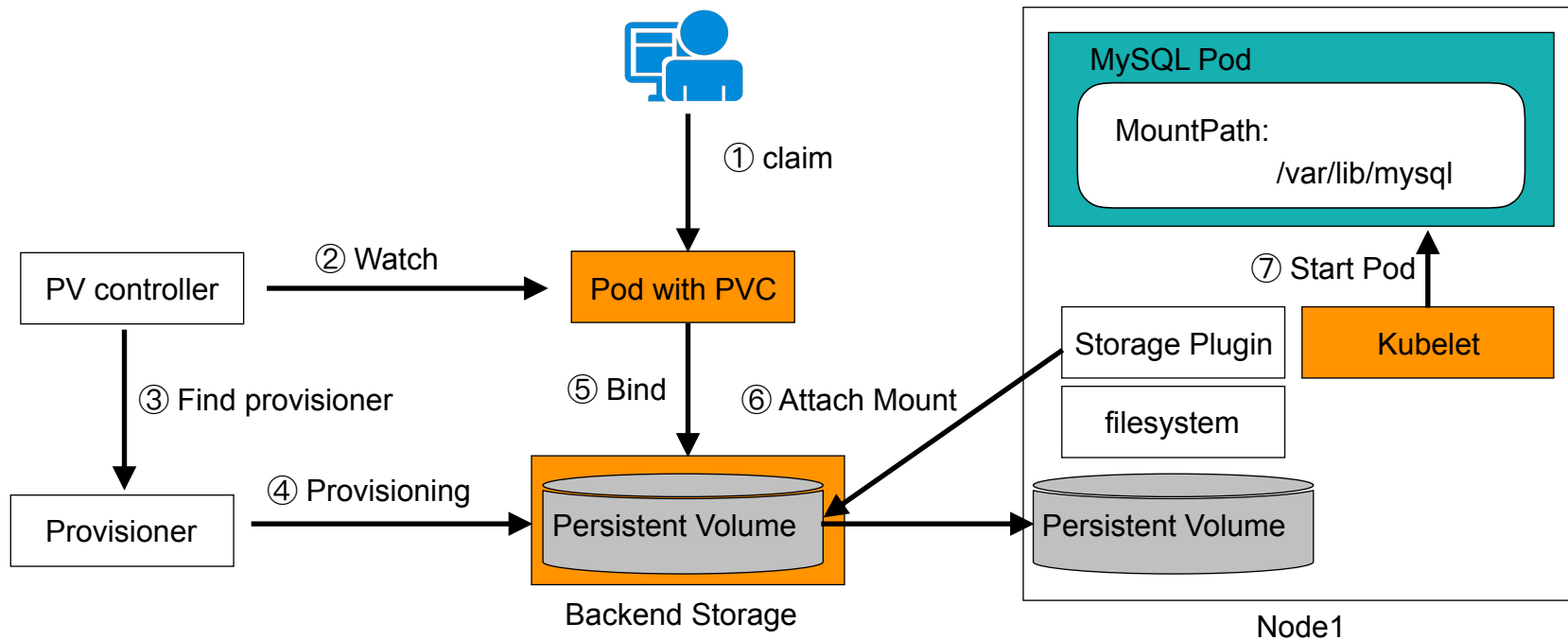
```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: fast
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd
  zone: europe-west1-b
```

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: slow
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-hdd
  zone: europe-west1-a
```

What kubernetes offer?



Persistent Volume 挂载过程



What kubernetes offer?



Statefulset

- Ordinal
- Stable Network ID
 - $\$(statefulset\ name)-\$(ordinal)$
- Storage
 - Pod with a single PersistentVolume
- Created sequentially, in order from $\{0..N-1\}$
- Terminated in reverse order, from $\{N-1..0\}$.

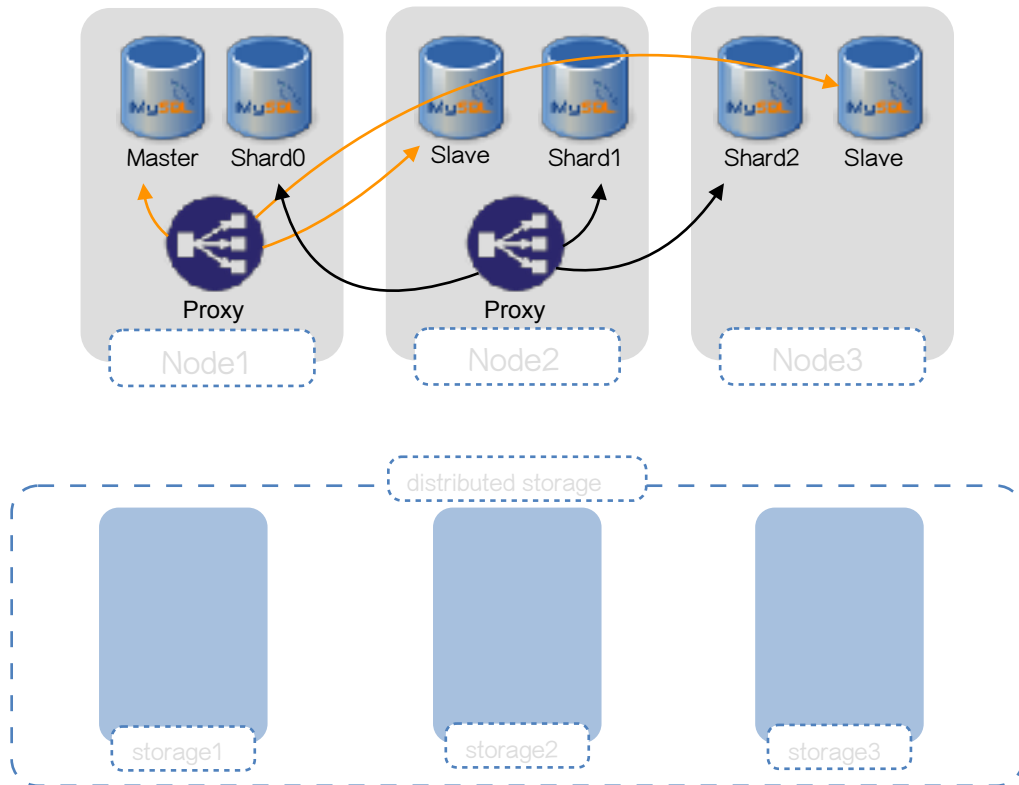
```
apiVersion: apps/v1beta1
kind: StatefulSet
...
spec:
  replicas: 3
  template:
    metadata:
    ...
    spec:
      containers:
      - name: mysql
        image: mysql
        volumeMounts:
          - name: data
            mountPath: /var/lib/mysql
      volumeClaimTemplates:
      - metadata:
          name: data
        spec:
          resources:
            requests:
              storage: 1Gi
          accessModes:
            - ReadWriteOnce
```

Build database workload



- 计算存储分离架构
 - 计算: worker node
 - 存储: 分布式存储
- kubernetes
 - statefulset提供数据库集群
 - pvc提供持久存储
 - service提供资源访问

整体复杂度巨大



Build database workload



- 计算存储分离架构
 - 计算: worker node
 - 存储: 分布式存储
- kubernetes
 - statefulset提供数据库集群
 - pvc提供持久存储
 - service提供资源访问
- CRD: 抽象资源对象MySQL+Proxy
 - Read/write split
 - sharding

整体复杂度巨大

awesome-kubernetes-extensions

5 resources tracking a number of Kubernetes extensions.

Please send a pull request if you are using Kubernetes Third-Party Resources, Custom Resource Definitions, or the API Server Aggregator and we will add yours to the list.

- Rodd Operator: <https://github.com/looklook/brea/master/demos/kubernetes>
- Elasticsearch Operator: <http://github.com/ucm-inc/enterprise-elasticsearch-operator>
- Dood Operator: <https://consoa.com/blog/introducing-the-etcd-operator.html>
- Funnelview Operator: <https://www.funnelview.com/funnelview-operator.html>
- CoreOS Teletonic: <https://coreos.com/teletonic>
- Kelsey: lightewer's kube-cert-manager: <https://github.com/kelseyhighower/kube-cert-manager>
- Operational Rigging: <https://github.com/gravitational/rigging>
- PalmStone-Demos kube-cert-manager: <https://github.com/PalmStone-Demos/kube-cert-manager>
- LikedNumbers: <https://lixiarc.io/blog/8.8.8/skward/index.html#kubernetes>
- Flannel Networking: <https://github.com/coreos/flannel>
- Project Calico Networking: <https://github.com/projectcalico/calico-gp>
- Das-018C Identity Provider: <https://github.com/das018c>
- Kikahonets Remote Manager: <https://github.com/1pmw/remote-kubernetes-cluster-manager>
- Giant Swarm's Kubectonl System: <https://blog.kubernetes.io/2017/09/19/how-we-run-kubernetes-in-kubernetes-kubectonl.html>
- Digital Ocean's Internal CA: <https://github.com/digitalocean/terraform-provider-kubernetes>
- Jorjanz operator - <https://github.com/jorjanz/operator>
- HAProxy based Ingress Controller - <https://github.com/appscode/voyager>
- Mirantis App Controller - <https://github.com/mirantis/mirantis-app-controller>
- kubernetes: <https://github.com/skylion/kubeflow>
- Kubernetes network stack: <https://github.com/mperio/kubernetes>
- SWHU House Hostgret Operator (I PH): create/destroy databases in the cluster or in the cloud.
- SWHU House IIR Operator (I PH): whitelisting ip's and management of these ip's/tables
- SBF OpenStack operator creates various resources in OpenStack (virtually, Keycloak, Openstack, projects, users, groups and roles) will expand to Swift accounts, Designate connections later: <https://github.com/sbf/kubernetes-operator/tree/master/openstack-operator>
- KubeVirt - run virtual machines on Kubernetes: <https://github.com/kubevirt/kubevirt>
- [@mhu](#) manage the physical resources that like currently Amazon, Elasticache, InstanceProfile, Network, EBS, IAM, etc.

Build database workload



- 计算存储分离架构
 - 计算: worker node
 - 存储: 分布式存储
- kuberentes
 - statefulset提供数据库集群
 - pvc提供持久存储
 - service提供资源访问
- CRD: 抽象资源对象MySQL+Proxy
 - Read/write split
 - sharding

整体复杂度巨大

Projects

We host and nurture components of cloud-native software stacks, including Kubernetes, Prometheus and Envoy. Kubernetes and other CNCF projects are some of the highest velocity projects in the history of open source. We are regularly adding new projects to better support a full stack cloud-native environment.



Vitess



Vitess is a database clustering system for horizontal scaling of MySQL through generalized sharding. By encapsulating shard routing logic, Vitess allows application code and database queries to remain agnostic to the distribution of data onto multiple shards. With Vitess, you can even split and merge shards as your needs grow with an atomic cutover step that takes only a few seconds. Vitess has been a core component of YouTube's database infrastructure since 2011, and has grown to encompass tens of thousands of MySQL nodes. It's architected to run as effectively in a public or private cloud architecture as it does on dedicated hardware. It combines and extends many important MySQL features with the scalability of a NoSQL database.

The problems



IT大咖说
知识共享平台



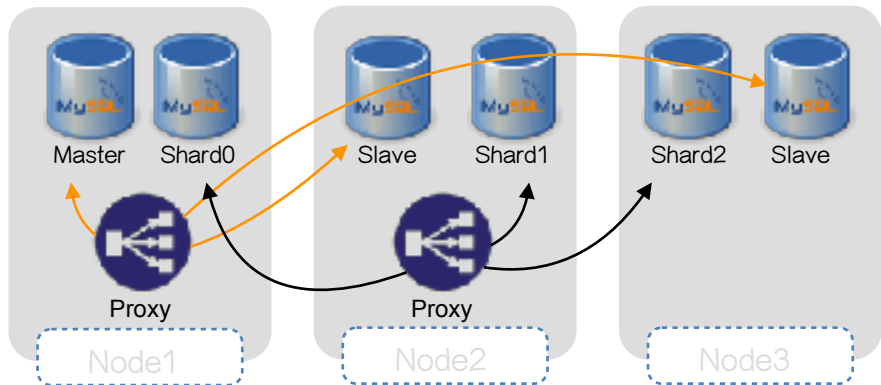
The problems



Performance

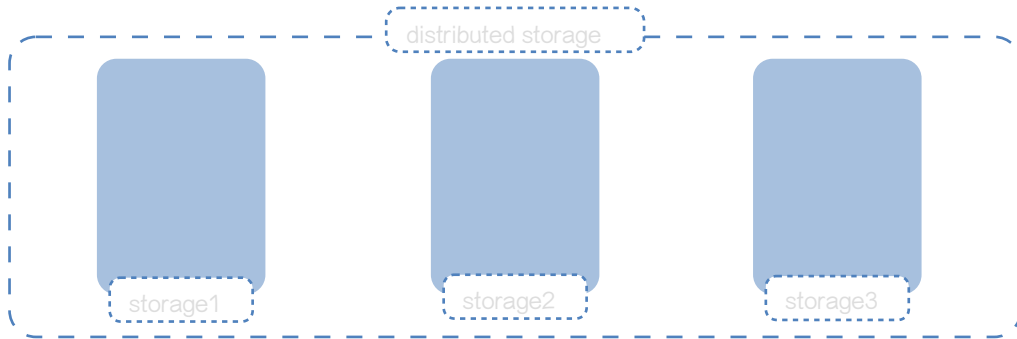
计算存储分离

- 提高了数据库实例的部署密度和计算资源利用率
- 架构清晰
- 扩展方便



问题

- IO路径更长，网络开销明显
- 无法针对Pod进行限流
- 数据库Latency Sensitive 型应用，网络延时会极大影响数据库能力(QPS,TPS)



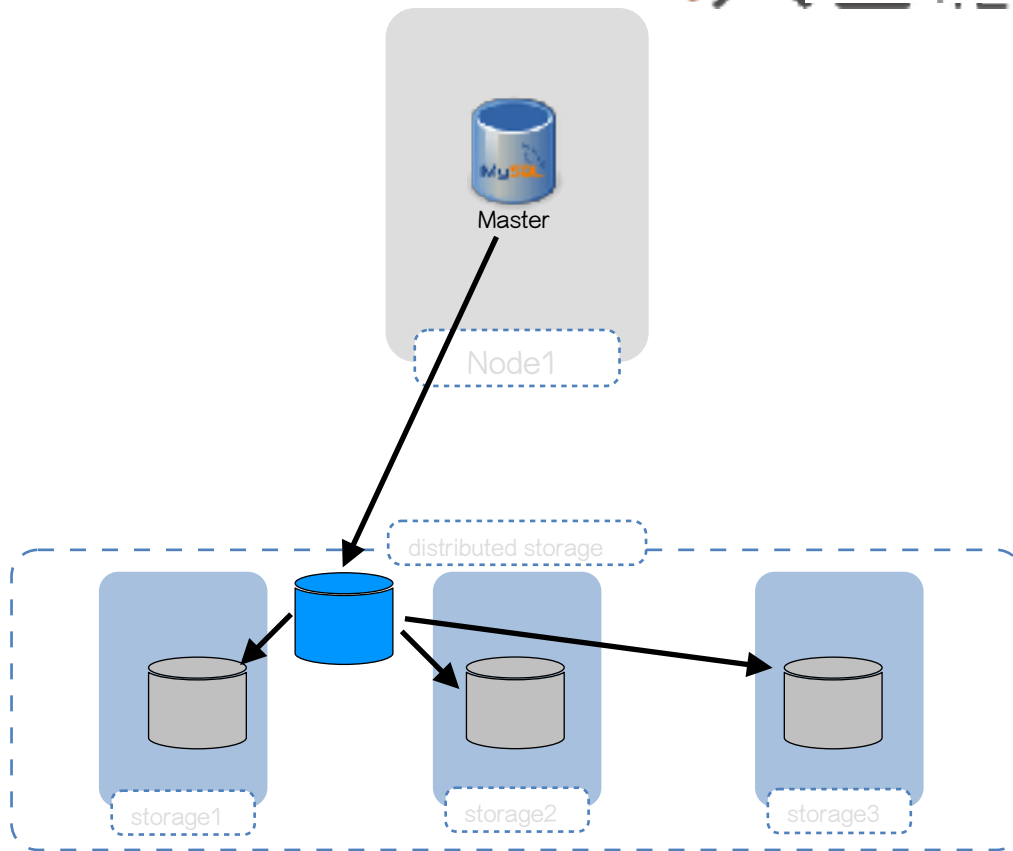
The problems



Performance

结合MySQL数据库特性的性能优化

- 减少IO发生次数
- 数据库层关闭DoubleWrite
- 文件系统层，支持 Atomic Write

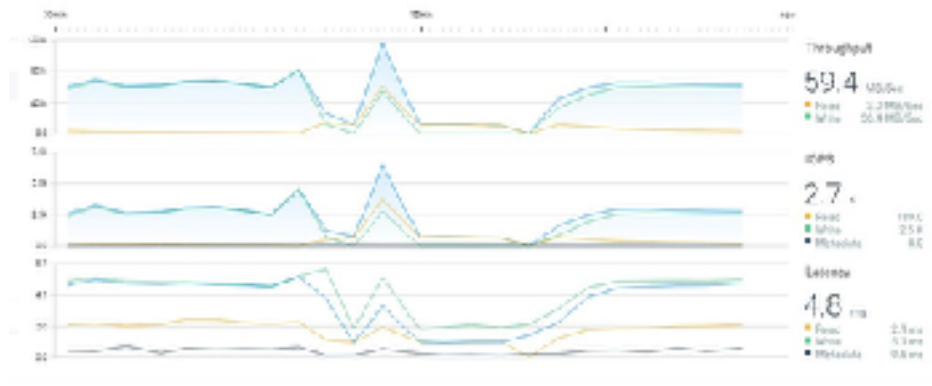
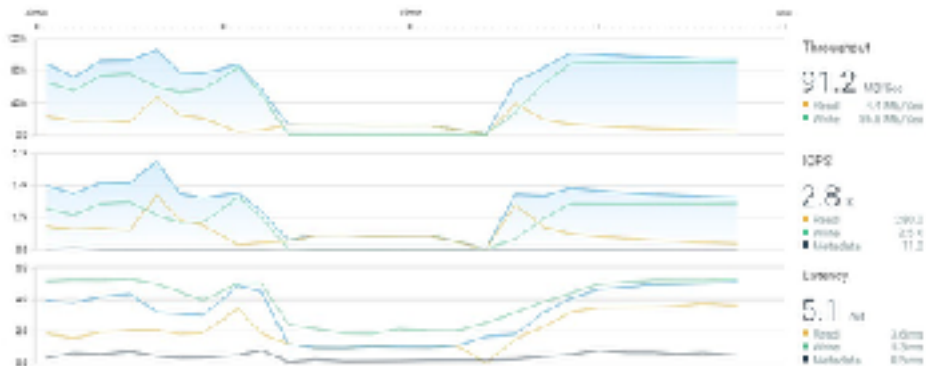


The problems



IT大咖说
知识共享平台

Performance



Double write on

Double write off

- Sysbench指标
 - TPS ↑28.0892%, QPS ↑28.0893%, RST ↓169.2033%
- 分布式文件系统指标
 - IOPS 提升22.3%
 - Latency 下降 39%
 - 在IOPS 提升22.3%的情况下, Throughput 仅多消耗 3.6%

The problems



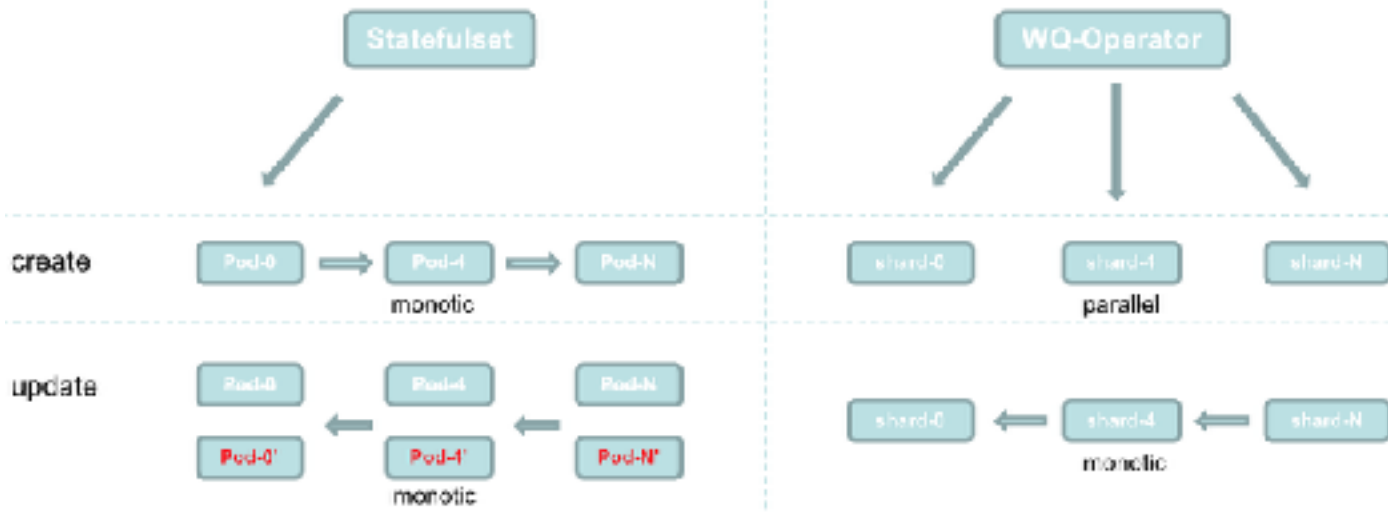
RollingUpdate

集群资源更新

- CPU、memory
- db/proxy config

异常处理

- node reboot
- instance restart
- human mistakes



CRD并没有提供RollingUpdate的机制

ControllerRevision

Summary



- Kubernetes
 - pv/pvc
 - statefulset
 - service
- Build database
- The problems
 - Performance
 - RollingUpdate

Thanks & QA