DTCC

2017第八届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2017

阿里巴巴数据库容器化资源调度与实践

炎烈 @ 阿里巴巴

个人简介

- 2008 - 2016曾在云壤、百度、360、豌豆荚从事运维与运维开发工作。
- 2016年2月加入阿里巴巴数据库团队。
- 2016年7月开始负责阿里巴巴数据库资源交付与调度
- 2016年阿里巴巴双11大促实现数据库业务容器化。

目录

- 需求背景
- 设计目标
- 调度系统
- 生产实践

目录

- ~~需求背景~~
- 设计目标
- 调度系统
- 生产实践

容器为调度提供可能

目录

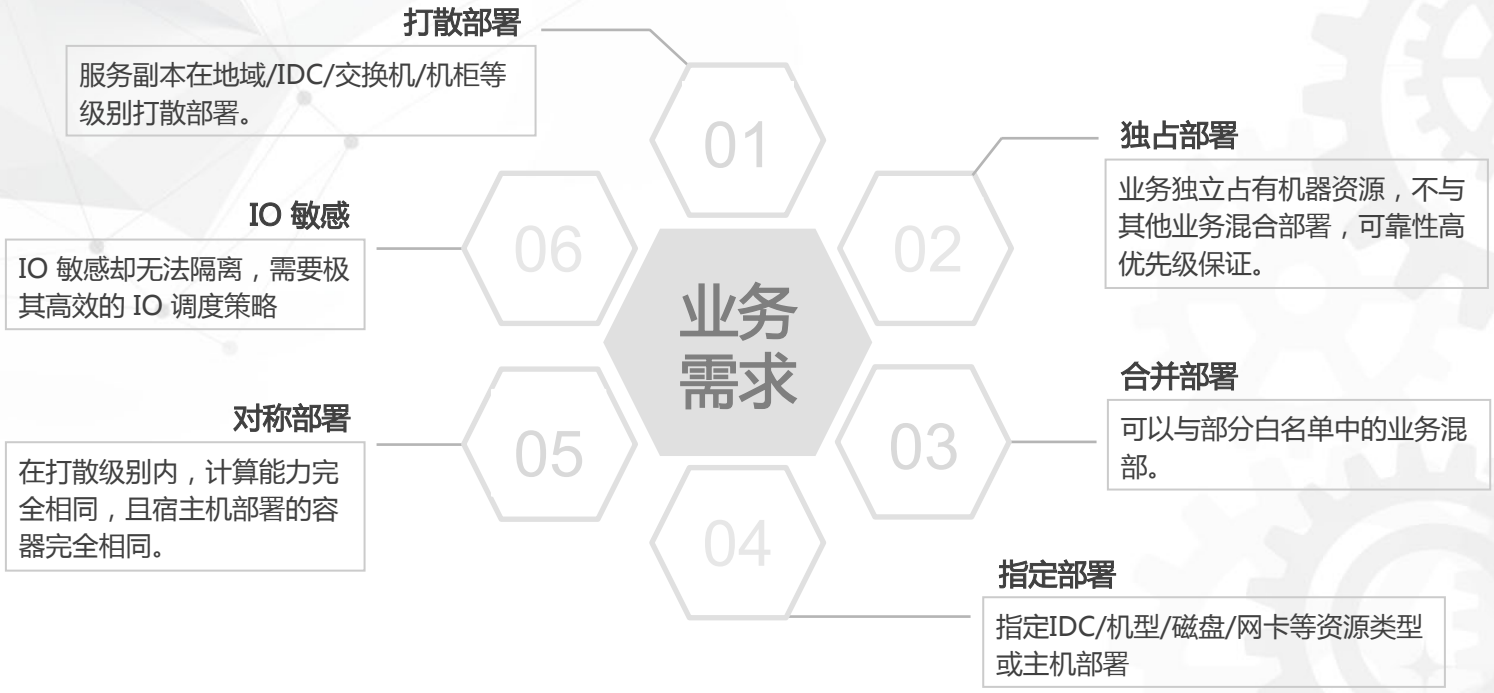
- ~~需求背景~~
- 设计目标
- 调度系统
- 生产实践

设计目标

- 满足业务需求
- 高效资源交付

业务的需求是什么？
我们的目标是什么？

满足业务需求


业务需求
01
打散部署
服务副本在地域/IDC/交换机/机柜等级别打散部署。
02
独占部署
业务独立占有机器资源，不与其他业务混合部署，可靠性高优先级保证。
03
合并部署
可以与部分白名单中的业务混部。
04
指定部署
指定IDC/机型/磁盘/网卡等资源类型或主机部署
05
对称部署
在打散级别内，计算能力完全相同，且宿主机部署的容器完全相同。
06
IO 敏感
IO 敏感却无法隔离，需要极其高效的 IO 调度策略

高效资源交付

01

屏蔽底层资源

- 公有云
- 物理机

02

提升资源利用率

- 密度
- CPU

03

减少资源碎片

- 内存
- 磁盘

04

提升运维效率

- 有状态 -> 无状态

目录

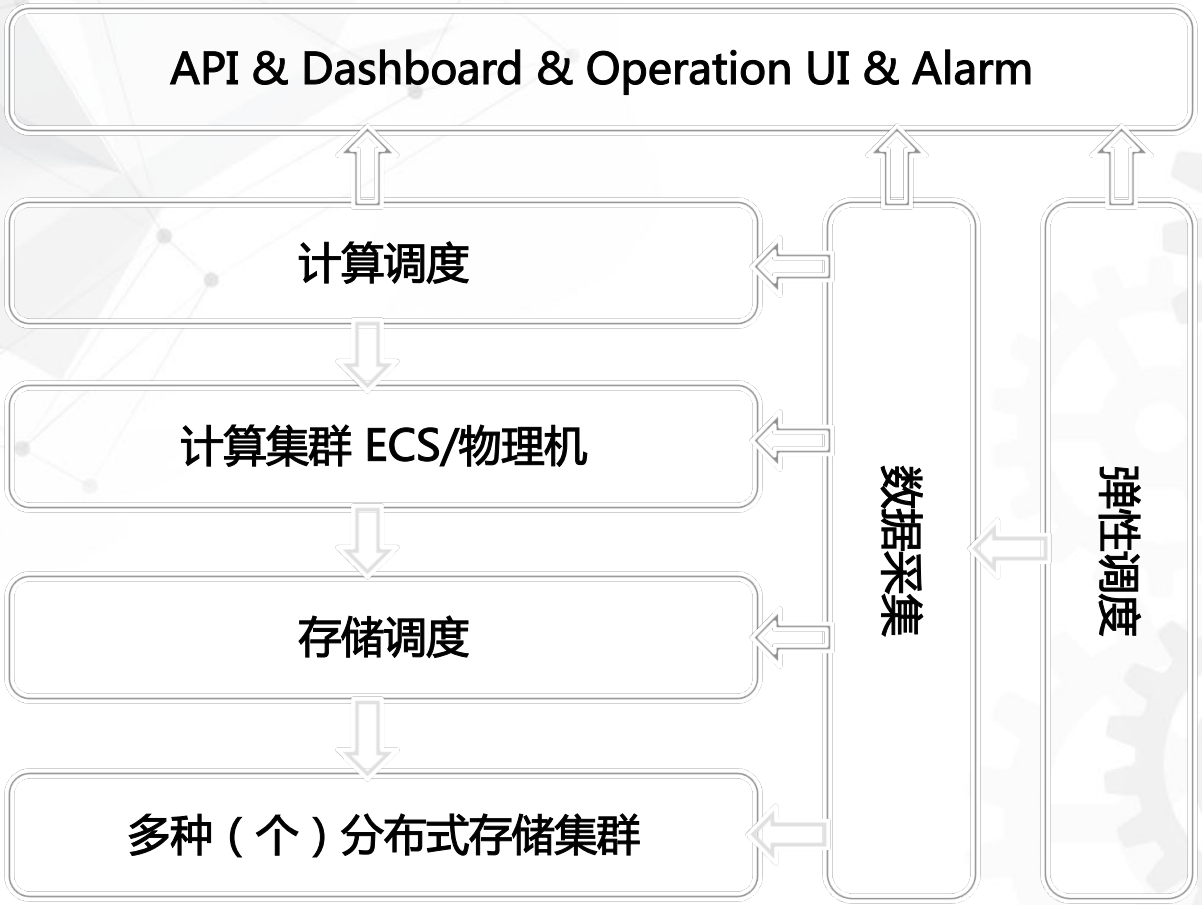
- ~~需求背景~~
- 设计目标
- 调度系统
- 生产实践

调度系统

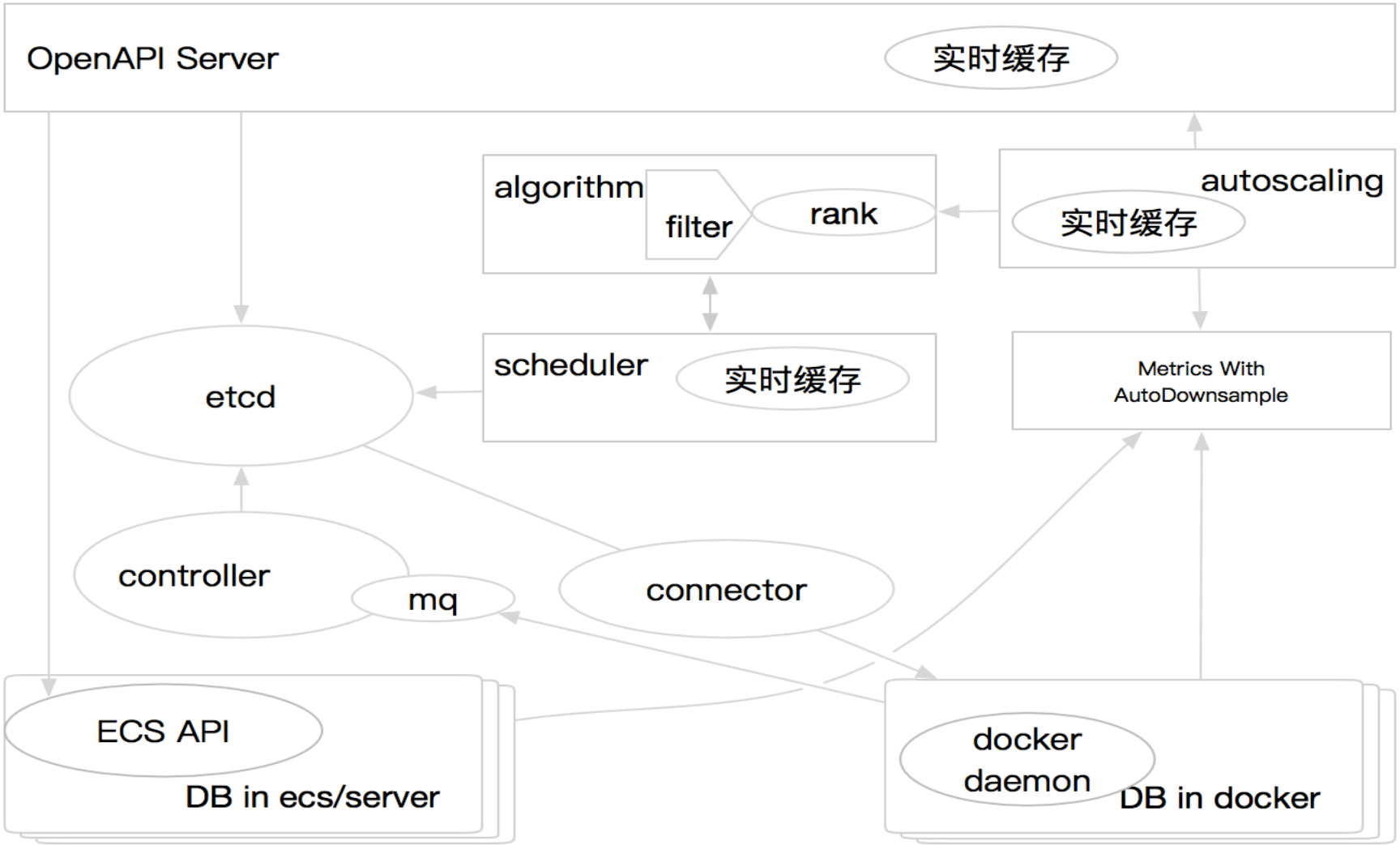
- 系统架构
- 调度架构
- 快速调度
- 支持混部
- 存储分离
- 弹性调度

系统长什么样子？
如何做最快调度？
如何解决碎片化？
如何更节省成本？

系统架构

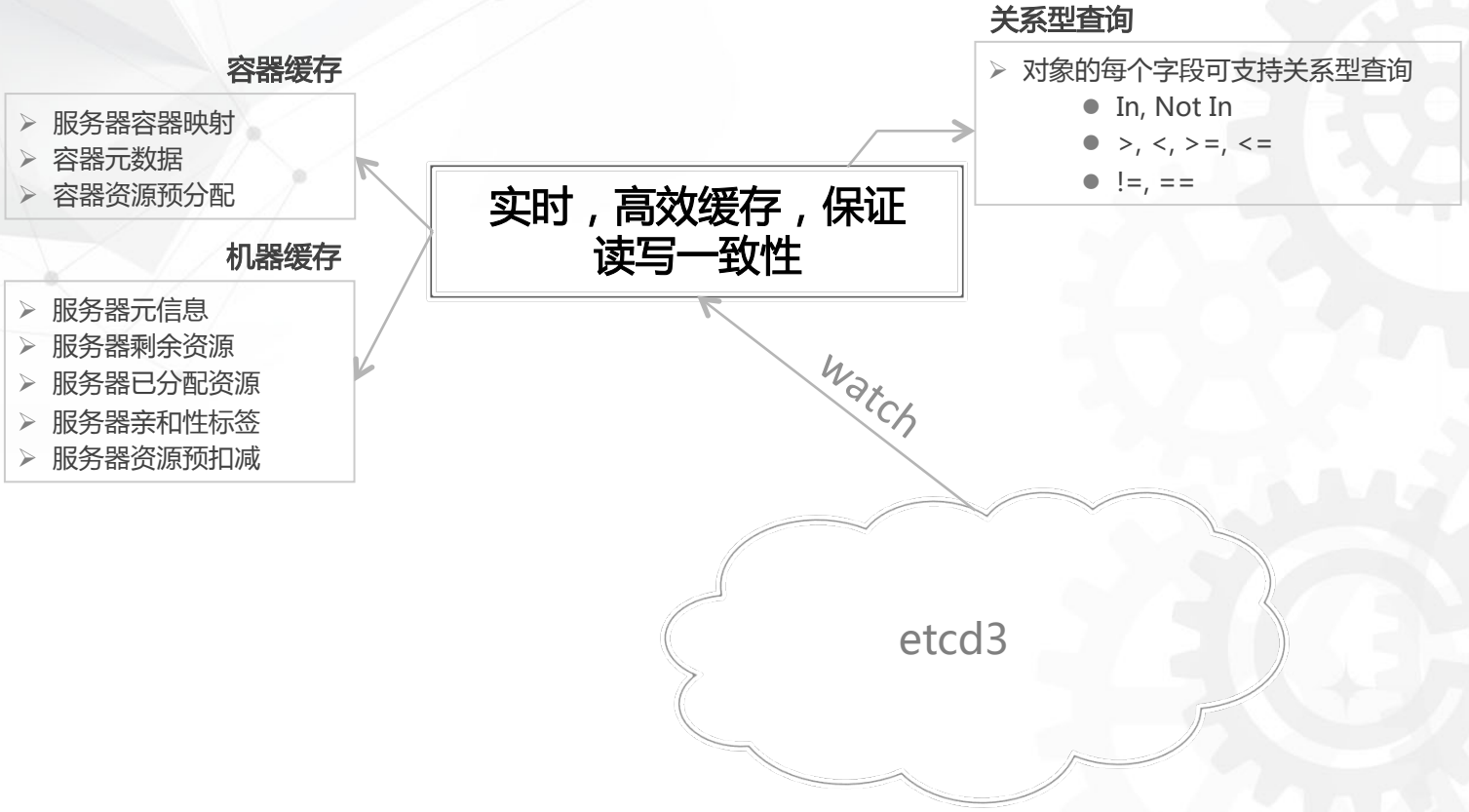

API & Dashboard & Operation UI & Alarm
计算调度
计算集群 ECS/物理机
存储调度
多种（个）分布式存储集群
数据采集
弹性调度

调度架构


OpenAPI Server
实时缓存
algorithm
filter
rank
autoscaling
实时缓存
scheduler
实时缓存
etcd
Metrics With
AutoDownsample
controller
mq
connector
ECS API
DB in ecs/server
docker
daemon
DB in docker

快速调度

- 万台服务器毫秒级



支持混部

- 隔离方案
  - ✓Cgroup Parent
  - ✓IO Device
- 调度方式
  - ✓独立调度
  - ✓不同售卖比

存储分离

Volume Plugin

```
{
  "Name": "plugin-example",
  "Addr": "https://example.com/docker/plugin",
  "TLSConfig": {
    "InsecureSkipVerify": false,
    "CAFile": "/usr/shared/docker/certs/example-ca.pem",
    "CertFile": "/usr/shared/docker/certs/example-cert.pem",
    "KeyFile": "/usr/shared/docker/certs/example-key.pem"
  }
}
```

✓意义

- 容器与外部存储整合
- 超过容器生命周期

✓种类

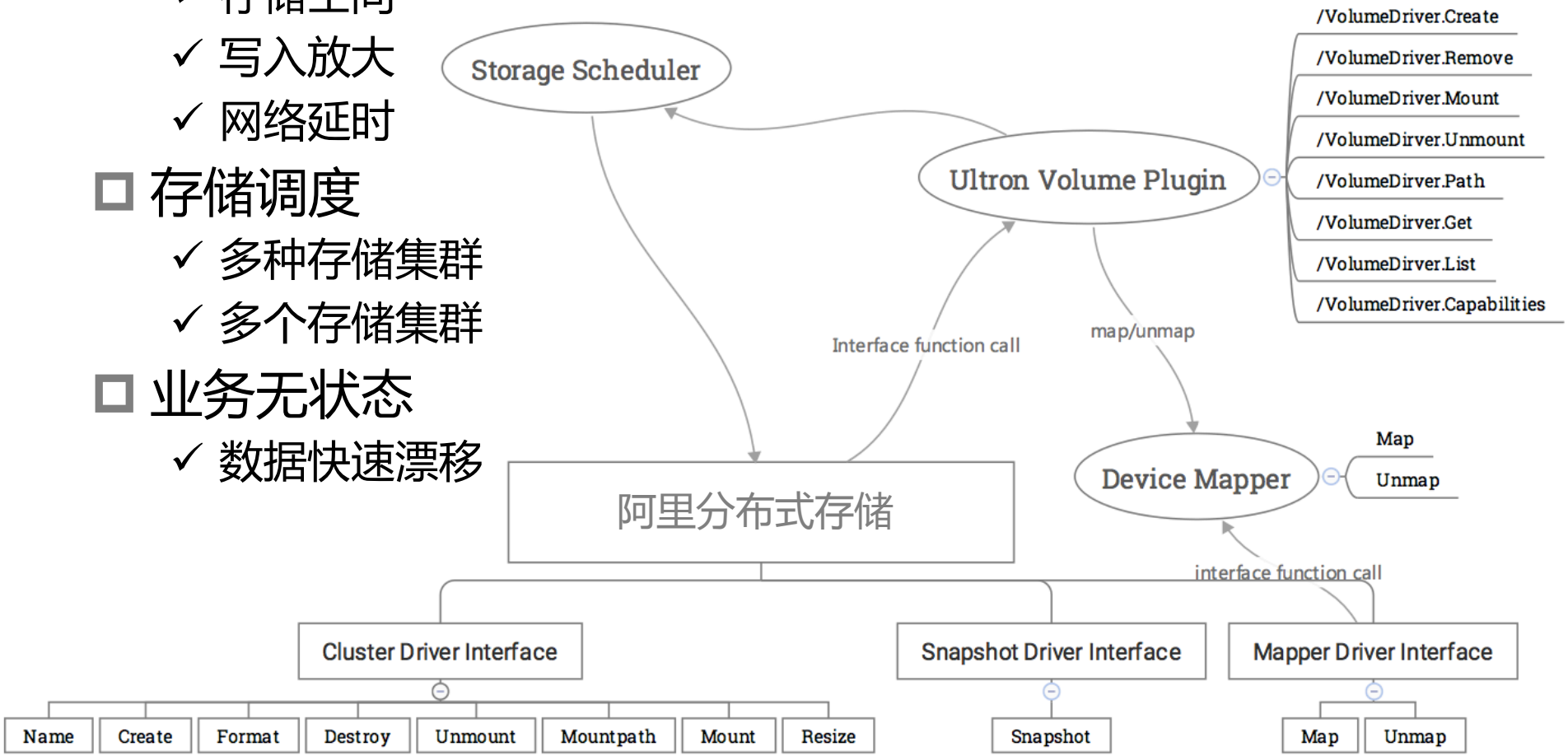
- .sock files are UNIX domain sockets
- .spec files are text files containing a URL, such as unix:///other.sock or tcp://localhost:8080.
- .json files are text files containing a full json specification for the plugin.

✓部署

- .sock /run/docker/plugins
- .spec, .json /etc/docker/plugins or /usr/lib/docker/plugins

存储分离

- 可行性
  - ✓ 存储空间
  - ✓ 写入放大
  - ✓ 网络延时
- 存储调度
  - ✓ 多种存储集群
  - ✓ 多个存储集群
- 业务无状态
  - ✓ 数据快速漂移



弹性调度

- 精确垂直与水平伸缩
  - ✓ 最小成本迁移
  - ✓ 精确扩容所用
- 基于历史数据的最小成本迁移以及透明伸缩。
  - ✓ 最小成本迁移
  - ✓ 透明扩容缩容

目录

- ~~需求背景~~
- 设计目标
- 调度系统
- 生产实践

生产实践

- 控制风险
- 运维工具
- 业务混部

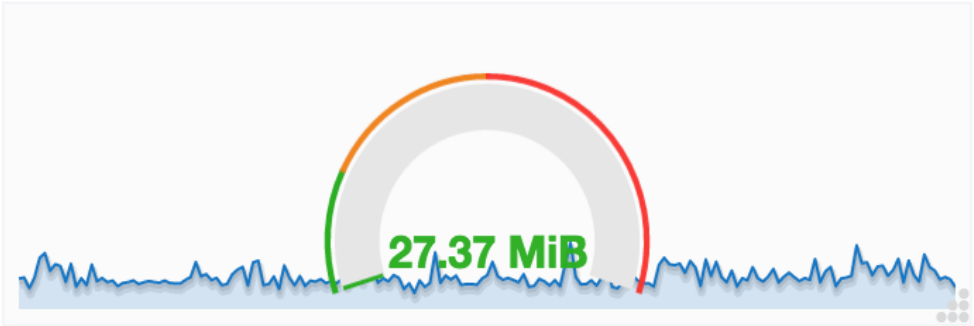
OOM频繁怎么办？
出现问题如何运维？
业务混部的现状如何？

控制风险

- 解除隔离
  - ✓ 10秒
  - ✓ 数万容器
  - ✓ 去除限制
- 内存超卖
  - ✓ 防止OOM，举例：
    - 内存 Limit 与数据库内存设置（x），物理机容器不OOM时 n = sum(x)/capacity (n 即为可分配为容器内存百分比，这里假设 n 为80%，小规格容器不OOM常量为1.2)，则：
      - 全部容器 sum(x) = capacity * 80%
      - 容器 limit = min(x + capacity * 20%, capacity, x*1.2)

运维工具

- 监管控平台
  - ✓ 数据展示
  - ✓ 异常报警
  - ✓ 为弹性提供数据
- 黑屏诊断工具
  - ✓ 命令行工具



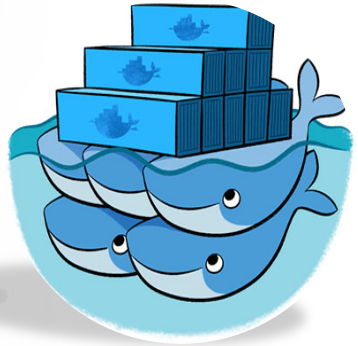
info		cpu				mem	sda				sdb				net	
Time	Hostname	usr	sys	idl	wai	usage	r/s	w/s	rb/s	wb/s	r/s	w/s	rb/s	wb/s	bytin	bytout
2017-05-09T13:29:16+08:00	db251138224[redacted]	01.57	01.00	97.37	00.06	44.06	0	0	0Mi	0Mi	15	255	0Mi	1Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009817	00.09	00.06	97.42	00.03	31.68	0	0	0Mi	0Mi	0	5	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009808	00.09	00.09	97.38	00.03	24.45	0	0	0Mi	0Mi	0	6	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009809	00.19	00.12	97.41	00.03	40.28	0	0	0Mi	0Mi	0	6	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009808	00.09	00.03	97.41	00.06	24.37	0	0	0Mi	0Mi	1	6	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009816	00.09	00.09	97.44	00.03	77.46	0	0	0Mi	0Mi	4	16	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009816	00.09	00.09	97.35	00.06	78.42	0	0	0Mi	0Mi	7	14	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009816	00.09	00.06	97.41	00.06	79.09	0	0	0Mi	0Mi	1	9	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009813	00.06	00.06	97.38	00.03	28.45	0	0	0Mi	0Mi	1	5	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009813	00.47	00.16	97.41	00.03	27.34	0	0	0Mi	0Mi	0	6	0Mi	0Mi	0Mi	0Mi
2017-05-09T13:29:16+08:00	docker01125009809	00.06	00.12	97.41	00.03	41.52	0	0	0Mi	0Mi	1	5	0Mi	0Mi	0Mi	0Mi

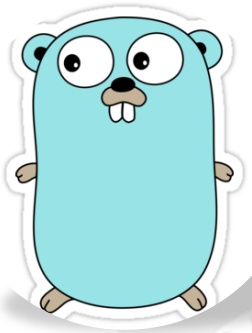
业务混部

- 混部现状
  - ✓在线混部
  - ✓在离线混部
  - ✓离在线混部

站在巨人肩膀上

- 技术选型
  - ✓ docker
  - ✓ golang
  - ✓ etcd3
  - ✓ gnatsd
  - ✓ prometheus
- 参考系统
  - ✓ kubernetes
  - ✓ Swarm





NATS

相关链接

- https://www.docker.com
- https://www.golang.org
- https://github.com/coreos/etcd
- https://github.com/nats-io/gnatsd
- https://prometheus.io
- https://kubernetes.io

Q & A

We are hiring: 技术专家、高级技术专家

- 资源调度开发
  - ✓ golang/k8s/mesos/docker/swarm
  - ✓ etcd/分布式存储/raft/paxos
- 数据库架构师
  - ✓ 业务解决方案
  - ✓ 业务架构能力

邮箱：guoan.qga@alibaba-inc.com

THANKS

SequeMedia 盛拓传媒 IT168.com ITPUB ChinaUnix