BDTC 2017 中国大数据技术大会
Big Data Technology Conference 2017

腾讯docker云平台GaiaStack

罗韩梅

目录
CONTENT

GAIASTACK

关于GaiaStack Docker私有云

GaiaStack是腾讯基于Kubernetes打造的Docker私有云解决方案，腾讯内部所有BG都有产品在GaiaStack上运行，包括IEG、TEG、OMG、WXG、SNG、CDG的信鸽、MTA、游戏云、EasyCount、广点通、深度学习平台等众多产品和服务。

一个通用的资源管理和调度平台，作为集群操作系统服务于上层各类应用。
（as a **Cluster Operating System**）

- ✓ 将一个数据中心的硬件资源逻辑上整合成一台服务器
- ✓ 为云应用软件提供统一、标准的接口
- ✓ 管理海量的任务以及资源调配

GaiaStack
提供了从构建至交付到运行的一整套的解决方案


容器服务

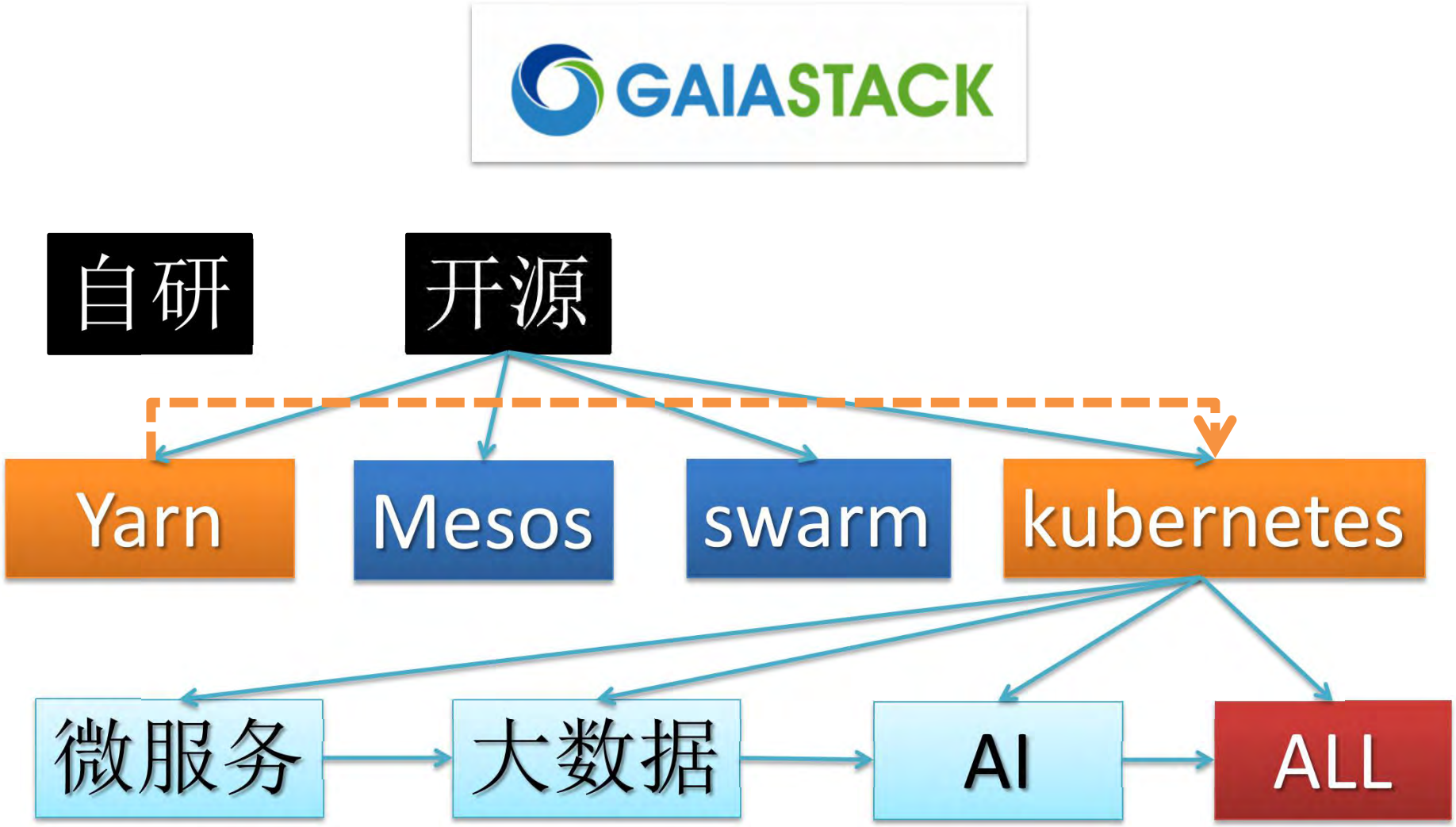

持续集成

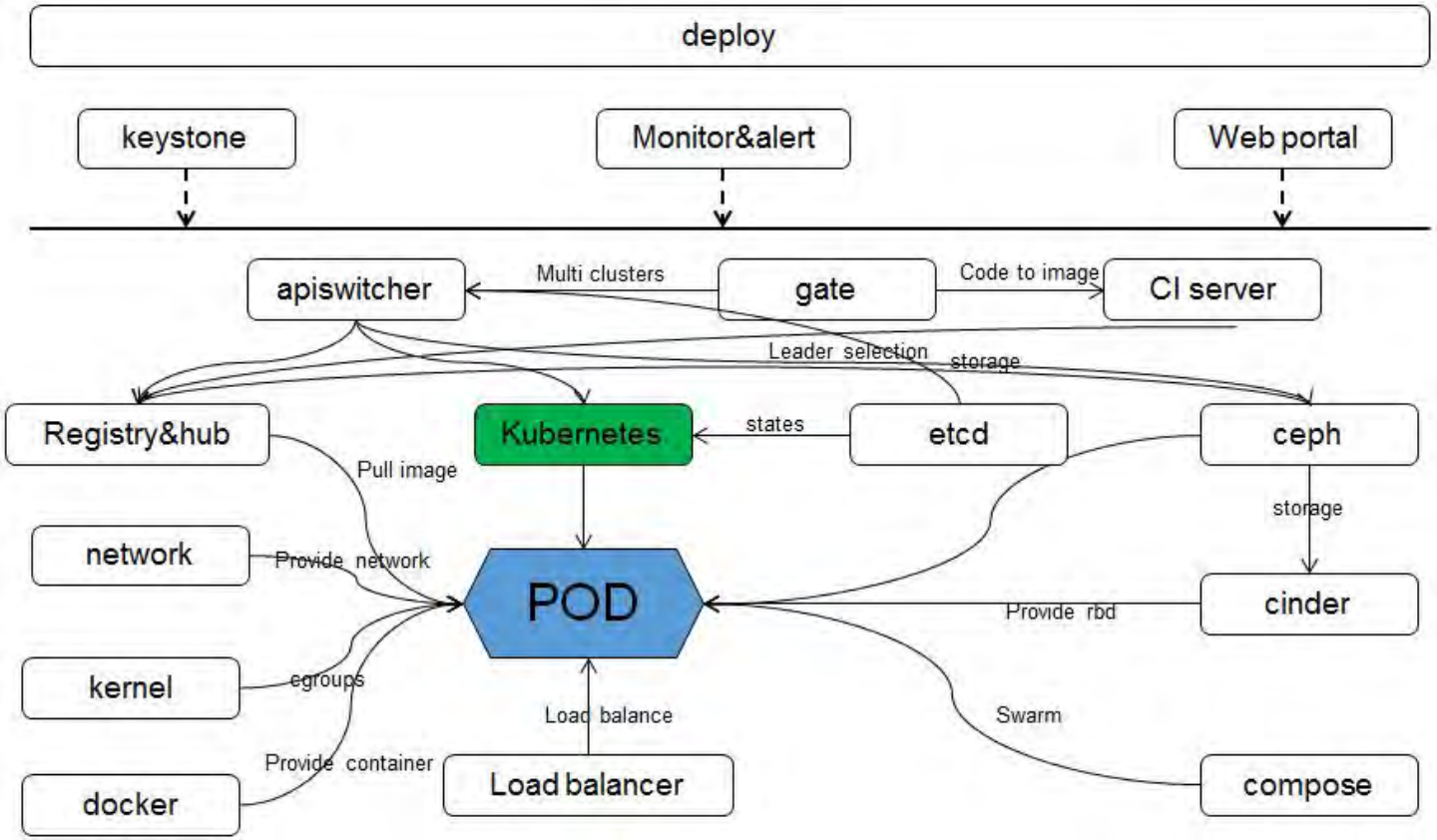

镜像仓库


资源编排

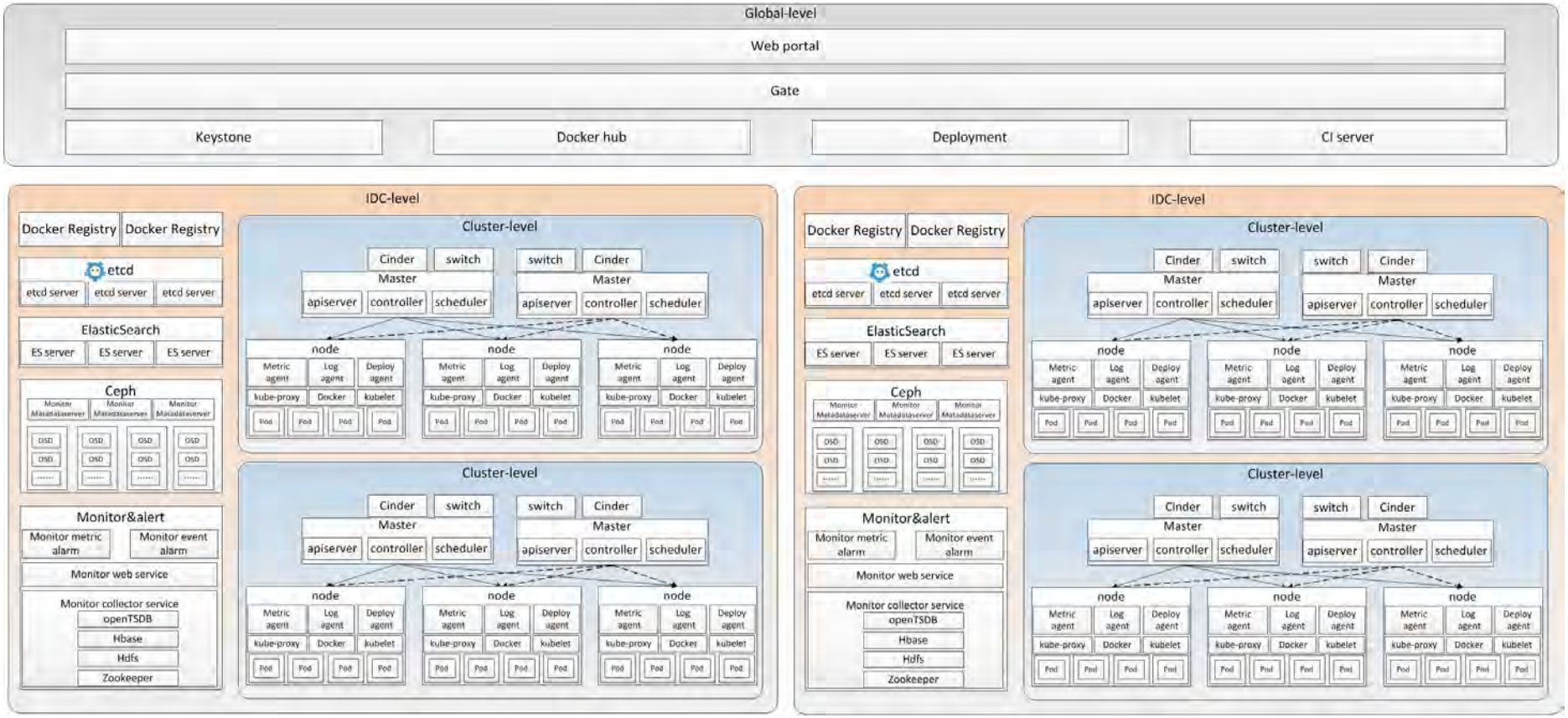

私有集群


部署与管理


GAIASTACK
自研
开源
Yarn
Mesos
swarm
kubernetes
微服务
大数据
AI
ALL


deploy
keystone
Monitor&alert
Web portal
apiswitcher
Multi clusters
gate
Code to image
CI server
Leader selection
storage
Registry&hub
Kubernetes
states
etcd
ceph
Pull image
storage
network
Provide network
POD
cinder
Provide rbd
kernel
egroups
Load balance
Swarm
Provide container
Load balancer
docker
compose

GaiaStack架构及特性



keystone

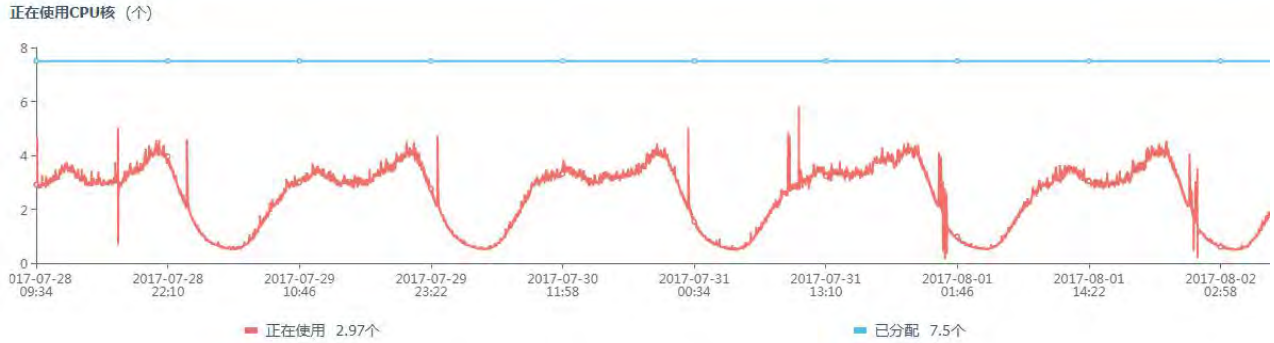
cinder

套件

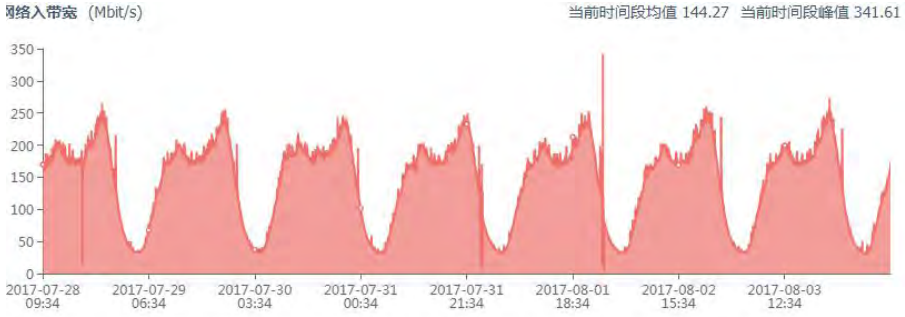
监控

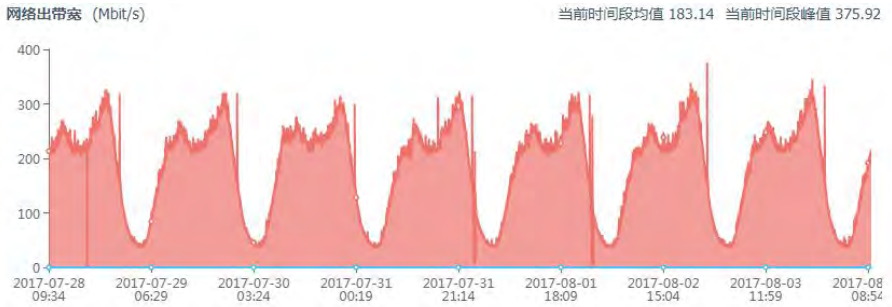
GaiaStack架构及特性

Online + offline



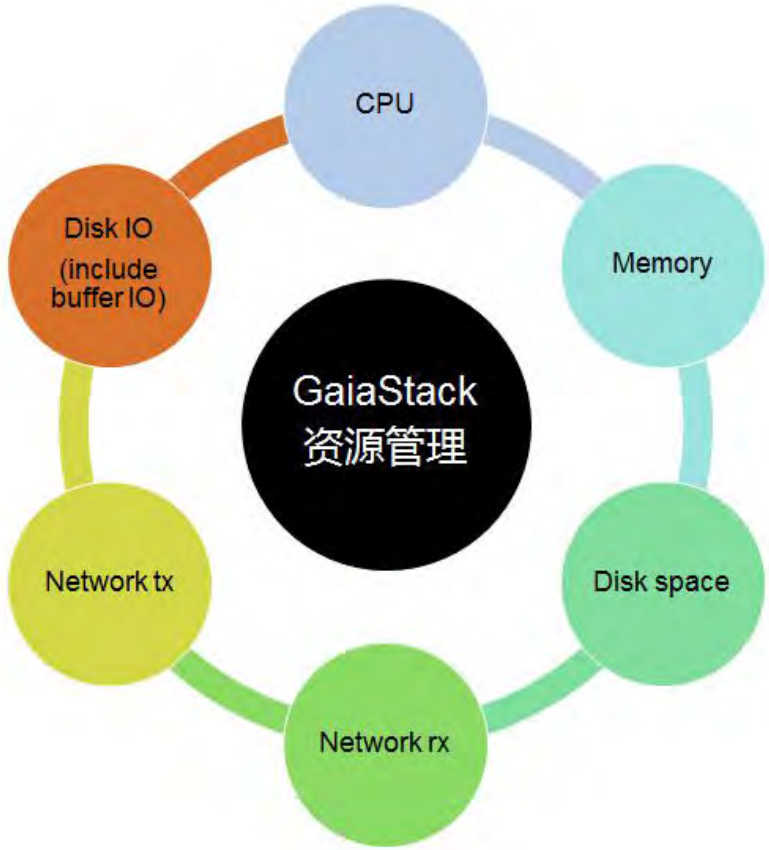
- 在线业务通常有以天为周期的资源特征
- 但是每个小时/每半小时资源都不同，甚至波动较大





kernel　docker　ceph　Registry　kubernetes

kernel

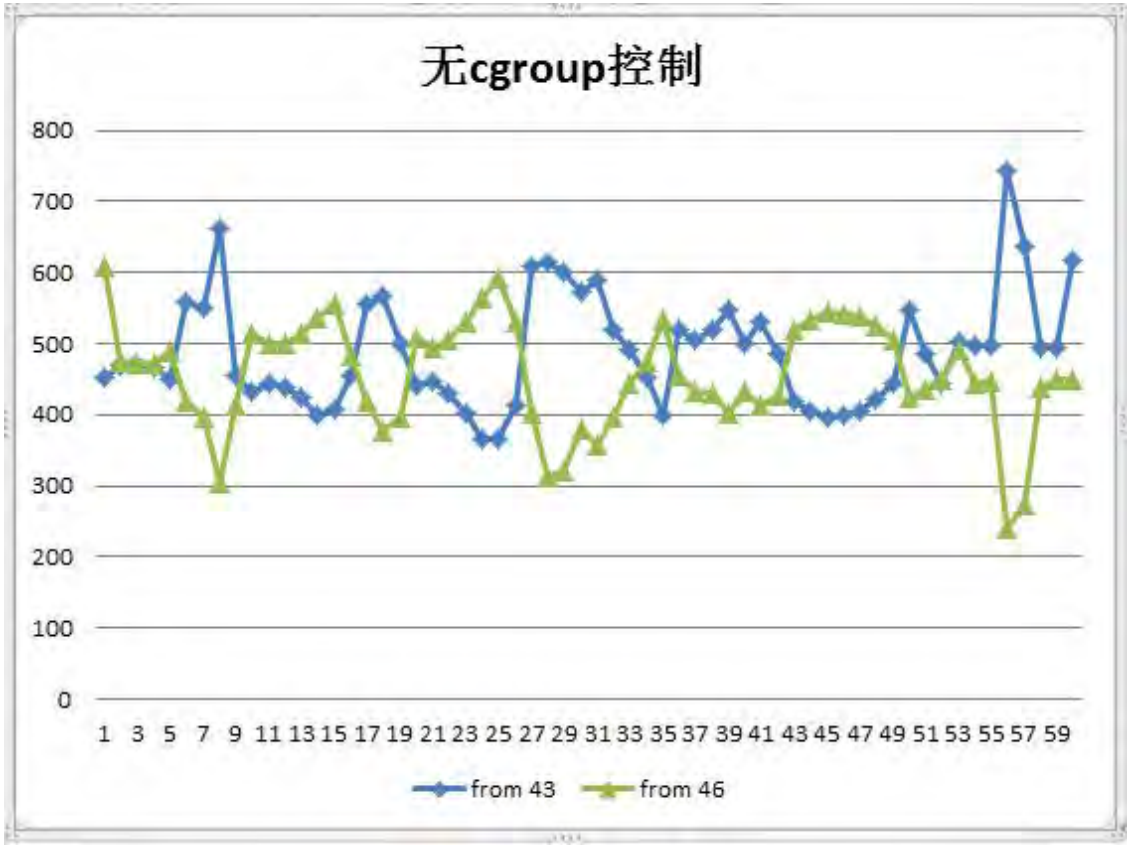


增加资源维度

更多的资源管理纬度
弹性的CPU控制
弹性的内存控制
弹性的磁盘容量控制
弹性的网络出带宽控制
弹性的网络入带宽控制
弹性的Disk IO控制
Buffer IO控制

Kernel： network IO

下图是两个进程都拼命争抢网络带宽时的效果。两个进程的带宽和时延都得不到任何程度的保证。



设计目标

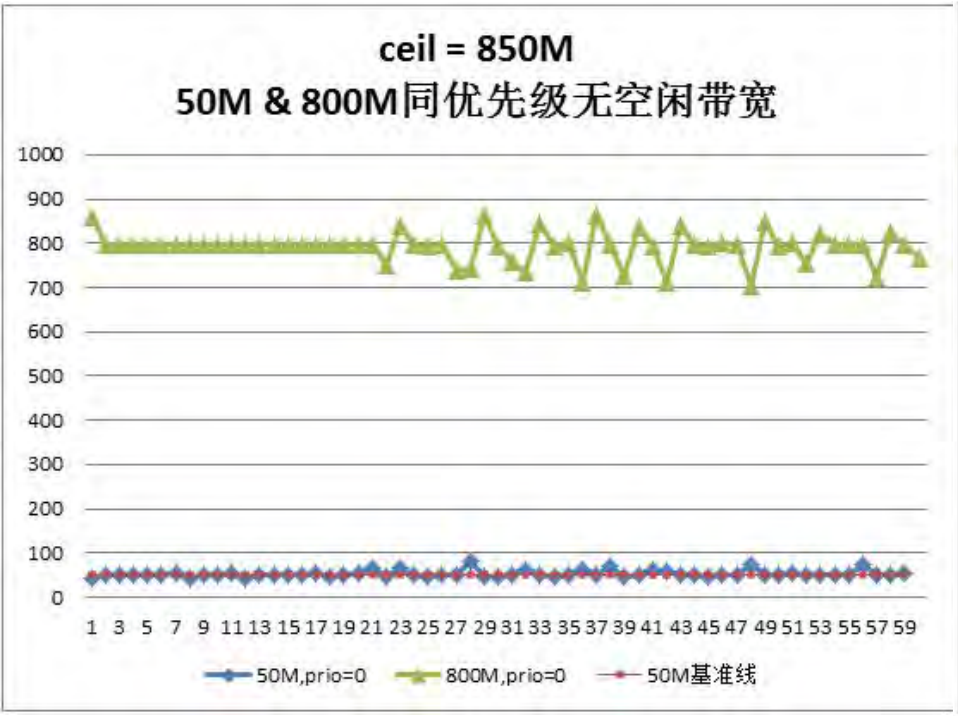
■在某个cgroup网络繁忙时，能保证其设定配额不会被其他cgroup挤占

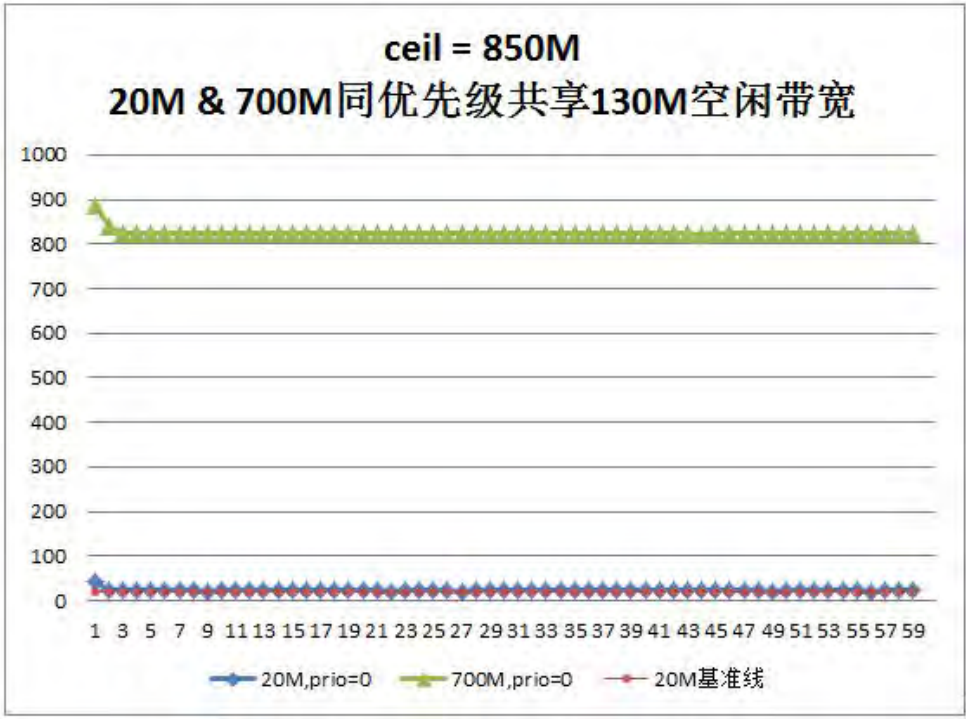
■在某个cgroup没有用满其配额时，其他cgroup可以自动使用其空闲的部分带宽

■在多个cgroup分享其他cgroup的空闲带宽时，优先级高的优先； 优先级相同时， 配额大的占用多，配额小的占用少

■尽量减少为了流控而主动丢包

Kernel： network IO



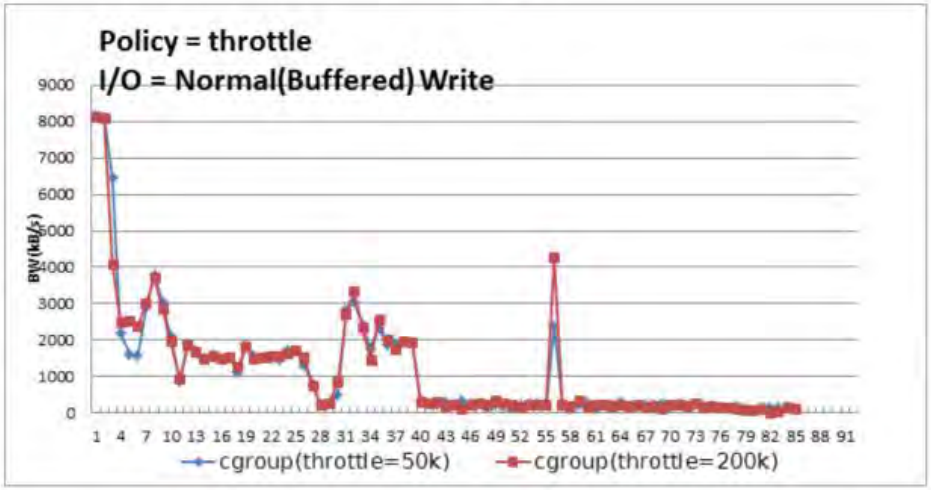


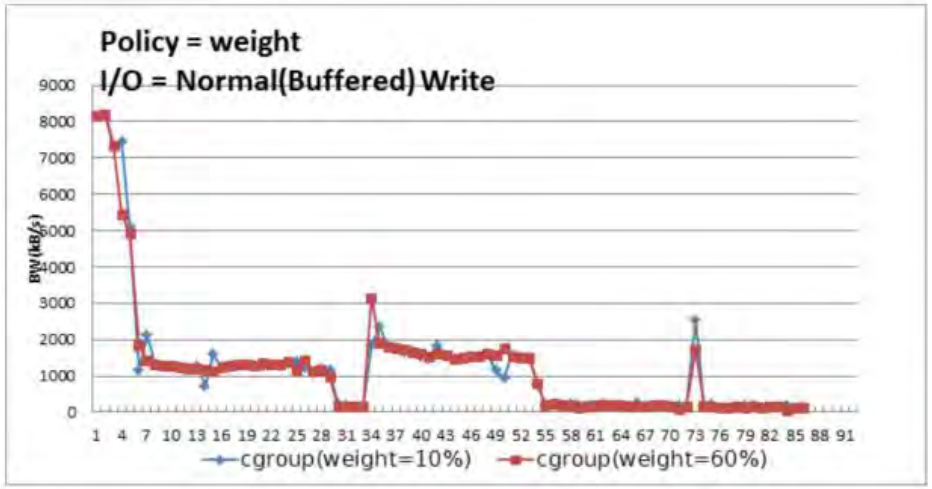
- 队列： 不增加队列， 对每个报文直接在正常代码路径上进行决策
- Cgroup区分(标记)： 在正常处理流程中，报文查找到目标socket结构之后，根据socket的owner process来确定cgroup
- 报文决策： 令牌桶 + 共享令牌池 + 显式借令牌
- 限速方式： ECN标记 + TCP滑窗 + 丢包

Kernel： Disk IO

对buffer io失控。cgroup通过识别pid，控制磁盘io。但在buffer io中，失去了原有的pid信息，导致不可控。

Normal Write





throttle=50k	
最小	90.0
最大	8185.0
平均	1154.1
方差	2368974.4

throttle=200k	
最小	17.0
最大	8155.0
平均	1171.8
方差	2282716.1

期望比例	0.2500
实际比例	0.9849

weight=10%	
最小	125.0
最大	8206.0
平均	1264.1
方差	2718006.0

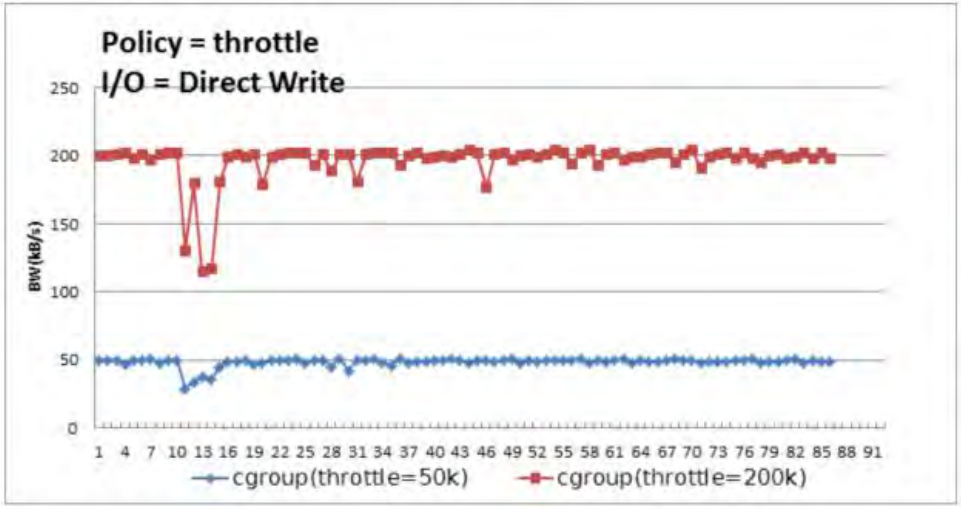
weight=60%	
最小	37.0
最大	8206.0
平均	1244.0
方差	2483813.4

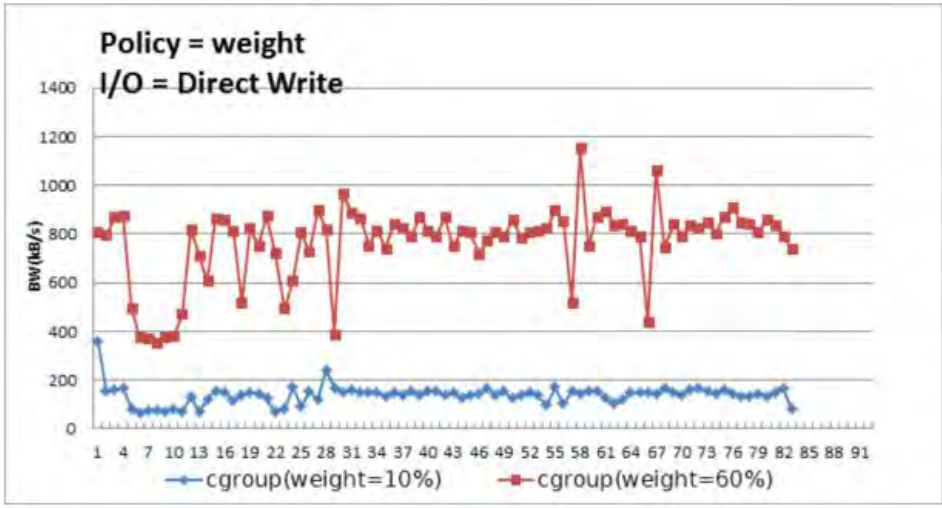
期望比例	0.1667
实际比例	1.0162

Kernel： Disk IO

- 在io weight方面（cfq机制），是通过时间片进行分割的，而不是通过iops或者bps进行衡量。用户观察到的传输数据波动比较大。
- cgroup对io的控制目前是hard模式(throttle)，即给某个进程限速后，该进程永远不会超过该速率

Direct Write

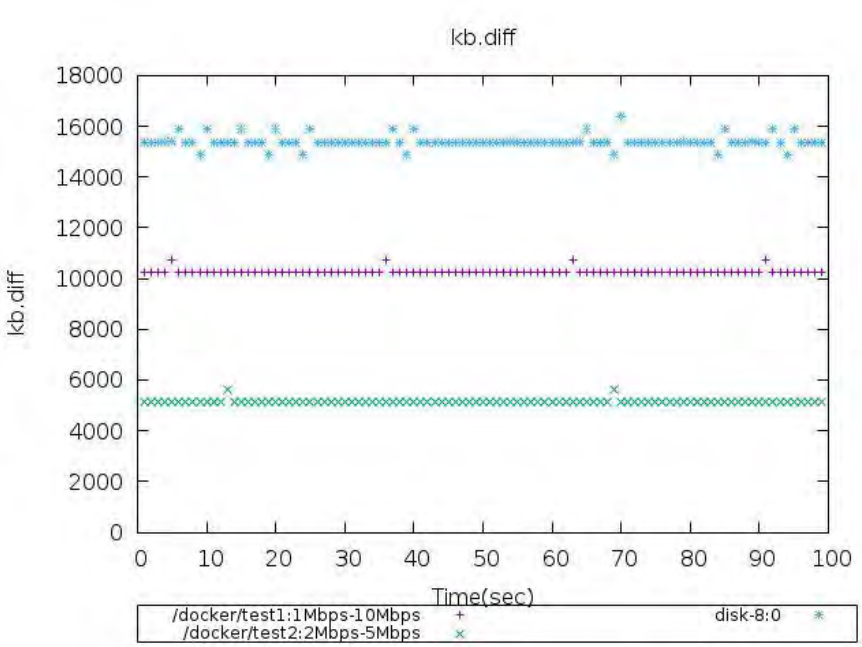




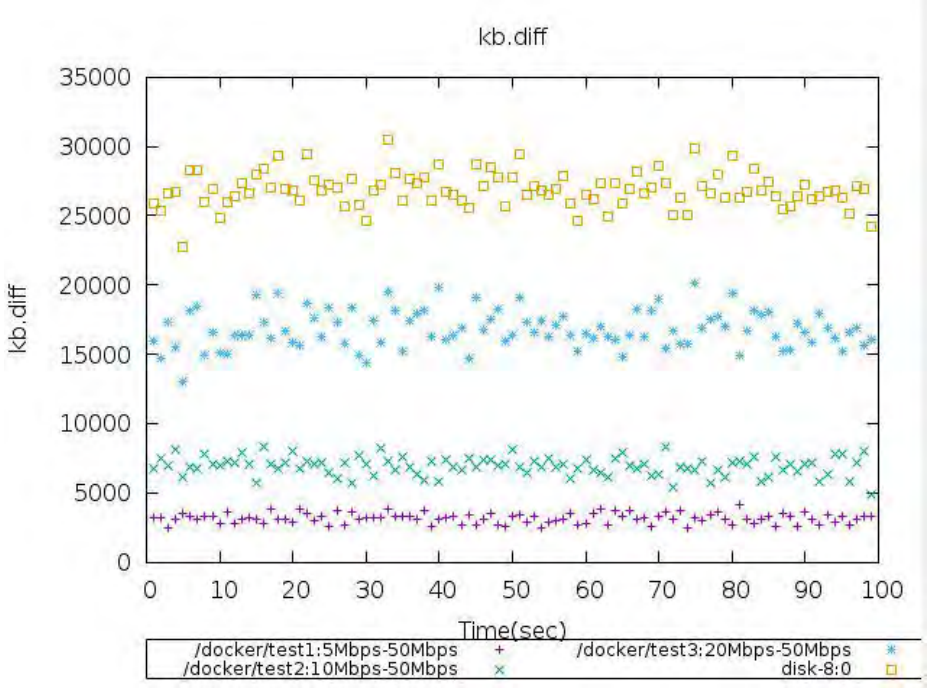
throttle=50k			throttle=200k			期望比例	0.2500
最小	29.0		最小	115.0		实际比例	0.2482
最大	51.0		最大	204.0			
平均	48.6		平均	195.9			
方差	12.7		方差	234.8			

weight=10%			weight=60%			期望比例	0.1667
最小	67.0		最小	353.0		实际比例	0.1808
最大	358.0		最大	1154.0			
平均	138.5		平均	766.2			
方差	1536.4		方差	24653.2			

Kernel： Disk IO



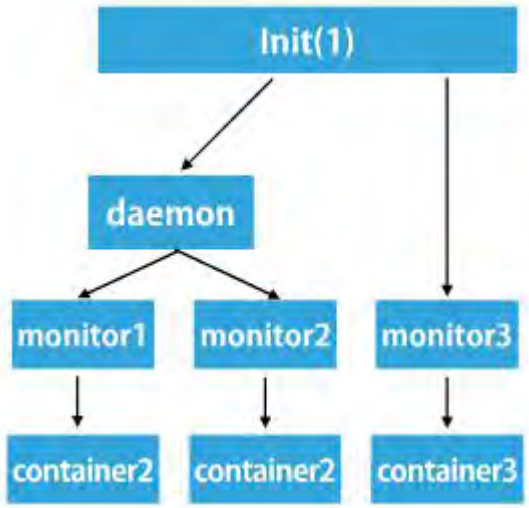
Buffer write测试



Direct write测试

Docker

- Bug fix
- Docker热升级
- 网络插件
- 弹性内存控制
- RBD插件



- Docker daemon停止时主动杀掉所有container，主要受限于
  - 用户进程是daemon的子进程
  - IO流经过daemon缓存
- 原来的两层进程父子关系变为三层，monitor由goroutine改为进程，由它等待container运行结束
- Docker重启时，monitor孤儿进程托管给init进程，container不受任何影响
- Docker重启后恢复所有Container状态

ceph

离线数据存储

cephFS

- 多MDS
- Cephfs大目录处理优化
  - 优化mds对大量文件的目录处理速度，速度提高6~10倍
  - 提高了mds主备切换的速度
- cephfs内核模块的bug fix
  - 稳定性改进
  - 支持quota
  - 支持Jewel, 支持keyring挂载权限

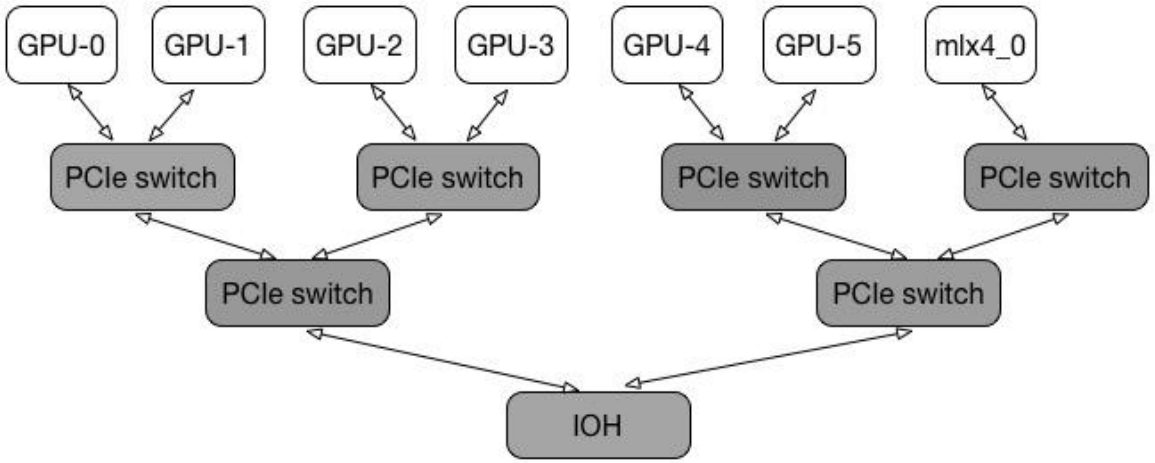
云硬盘

cephRBD

- 普通云硬盘
- 内置云硬盘
  - 临时云硬盘
  - 可迁移云硬盘

kubernetes

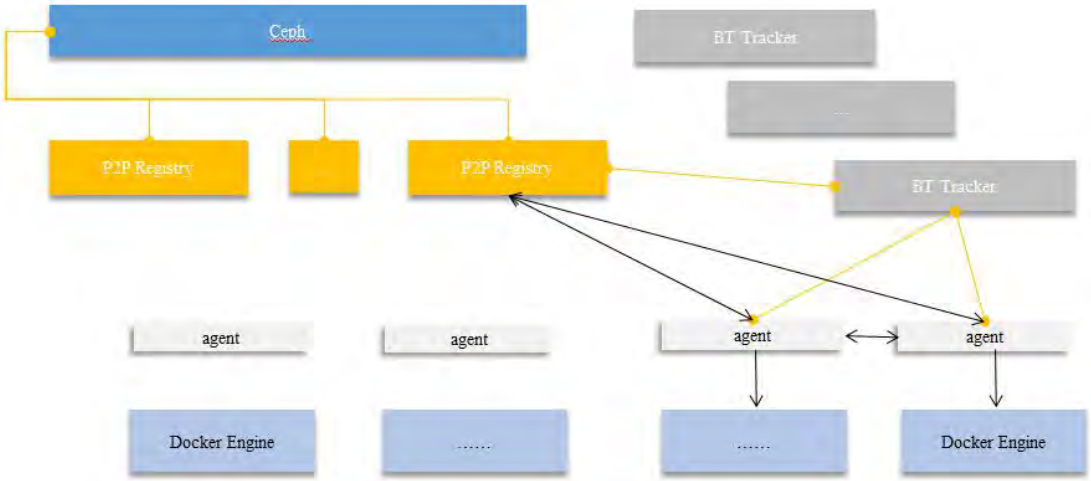
- Quota管理
  - 在线
  - 离线
- APP
  - 引入Tapp
- 网络模式
  - Overlay
  - Floating IP
  - NAT
  - Host
- 磁盘管理
  - Log、data
  - 云盘
- GPU应用



registry

GaiaStack有大规模部署应用的场景（如机器学习作业），需要对registry做相应优化

- 镜像仓库优化（迁移至registry2.0，使用ceph作为后端存储)
- registry服务高可用与负载均衡
- 多集群镜像的同步
- 基于P2P的大规模镜像分发



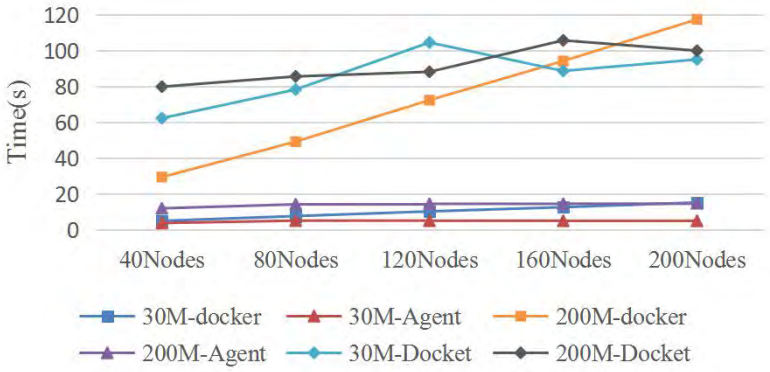
主要设计思想：

- 在镜像下载过程中，引入BT协议
- 在Blob上传时，对Blob生成种子
- 在下载镜像的Blob时，先下载种子，再通过种子文件下载数据

主要有三个组件：

- **P2P Registry**：镜像仓库, 种子生成
- **Agent**：P2P下载任务的主要功能组件
- **Tracker**：P2P下载资源查询定位组件

registry



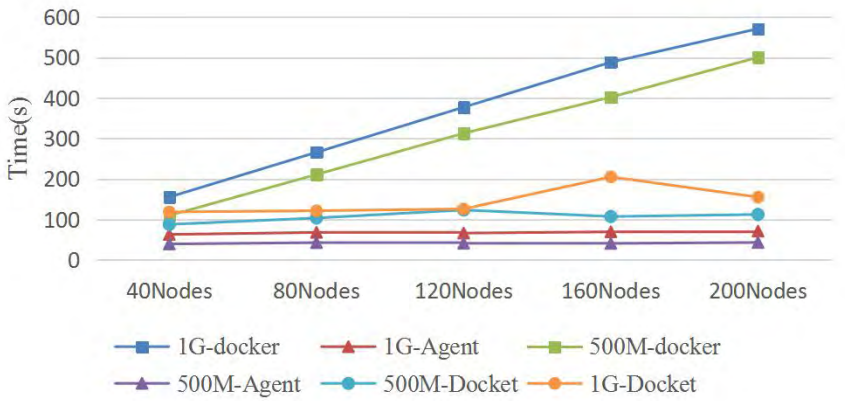


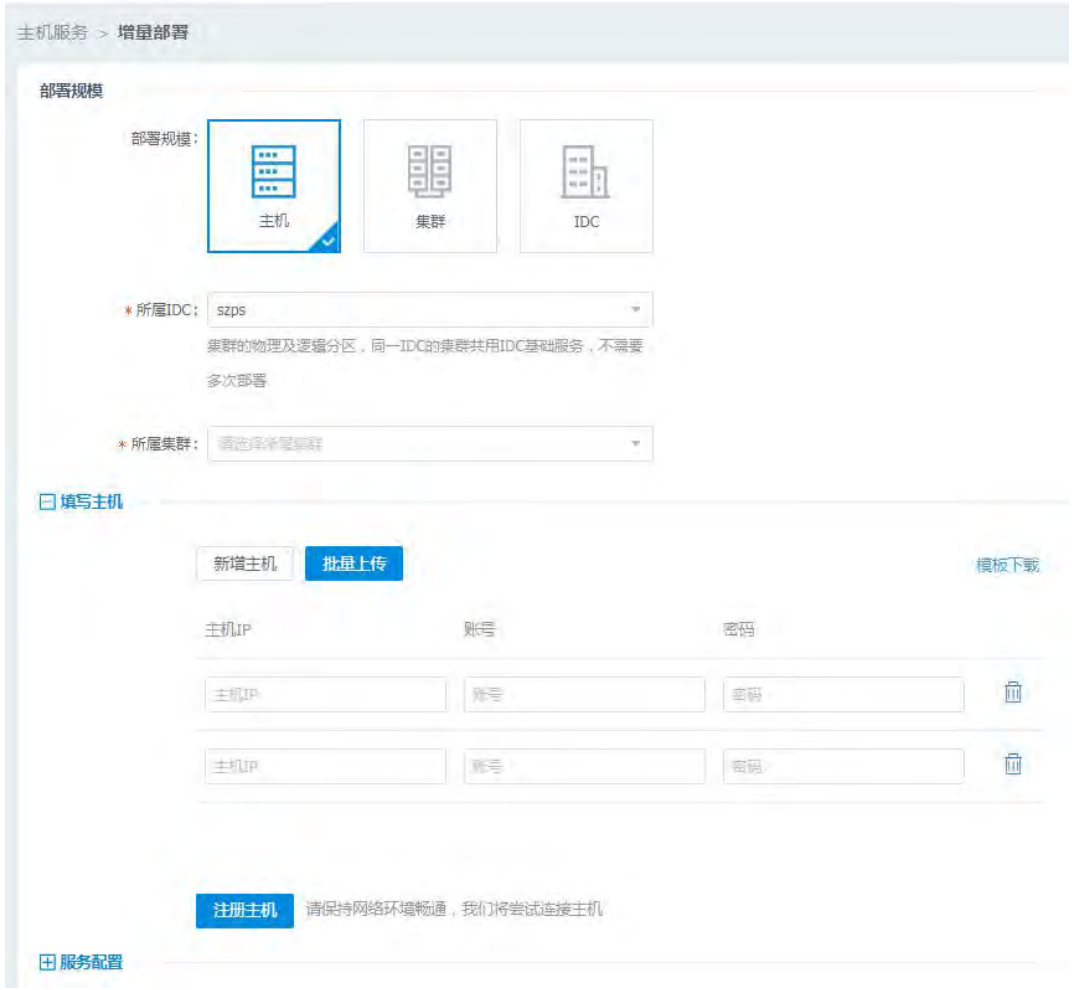
图5 P2Ppull与docker原生pull在不同并发量下的distribution时间对比，在腾讯云的200台物理机上进行了实验

镜像选择：
registry（30M）,centos（200M），hadoop（500M），tensorflow（1G）
Distribution time = Average(pull(image))

从图5、6可以看出，引入P2P后，pull镜像的速度显著提升，而且使得registry的流量大幅度减少（蓝色是如果不用P2Pregistry应该承担的流量）。

图6 在整个实验中，使用P2P下载时，registry(绿色)与agent(蓝色)出流量占比

开放的GaiaStack



Thanks

BDTC 2017 中国大数据技术大会
Big Data Technology Conference 2017