

# Docker 架构私有云的机遇和挑战

- 王振威

# Geekbang>

## 极客邦科技

整合全球最优质学习资源, 帮助技术人 and 企业成长  
Growing Technicians, Growing Companies

**InfoQ**  
new

专注中高端技术人员的  
技术媒体



**EGO** EXTRA GEEKS' ORGANIZATION  
NETWORKS

高端技术人员  
学习型社交网络



**StuQ**  
new

实践驱动的  
IT 职业学习和服务平台



**GiT** GEEKBANG  
INTERNATIONAL  
TRAINING  
极客邦培训

一线专家驱动的  
企业培训服务



旧金山 伦敦 北京 圣保罗 东京 纽约 上海  
San Francisco London Beijing Sao Paulo Tokyo New York Shanghai

# QCon

## 全球软件开发大会

2016年4月21-23日 | 北京·国际会议中心

主办方 **Geekbang**  **InfoQ**  
极客邦科技

**7折** 优惠 (截至12月27日)  
现在报名, 节省2040元/张, 团购享受更多优惠

[www.qconbeijing.com](http://www.qconbeijing.com)



扫描获取更多大会信息

# 内容梗概

- 关于 Docker
- 为什么变迁
- 架构变迁三步走
- Docker 的问题



# Docker ? 私有云 ?

- Docker : 一门新兴的容器技术
- 私有云 : 企业内部云服务平台

# Docker 为什么适合？

- 构建快：应用+运行环境 = 镜像
- 启动快：容器相比于虚机，更轻量级
- 迁移快：应用以容器的方式标准化交付，标准化运行

# 看下我们的架构图







# 事出有因

- 混乱的环境：Java, Golang, Ruby
- 混乱的配置：Upstart, authorized\_keys, dependency，各种脚本
- 混乱的监控：ErrorReporter，Message
- 混乱的资源：计算资源与预估不匹配



# 有因必有果

- 环境不匹配导致，测试跟生产不一致
- 配置混乱导致事故频发
- 监控不统一导致运维难上加难
- 资源效率低导致成本很高却达不到相应目标



# DevOps 变迁原则

- 即面向未来，又不过于激进
- 即追求稳定，又不过与保守

# 我们团队的做法

- 技术选型

OS	Windows/Ubuntu/CentOS/Redhat/ <i>Ubuntu</i>
Container	Rocket/RunC/ <i>Docker</i>
Service Discovery	Consul/ <i>Etcd</i>
Config	JSON/INI/ <i>YAML</i>
Container Management	K8s/Mesos/Swarm/Compose/ <i>None</i>

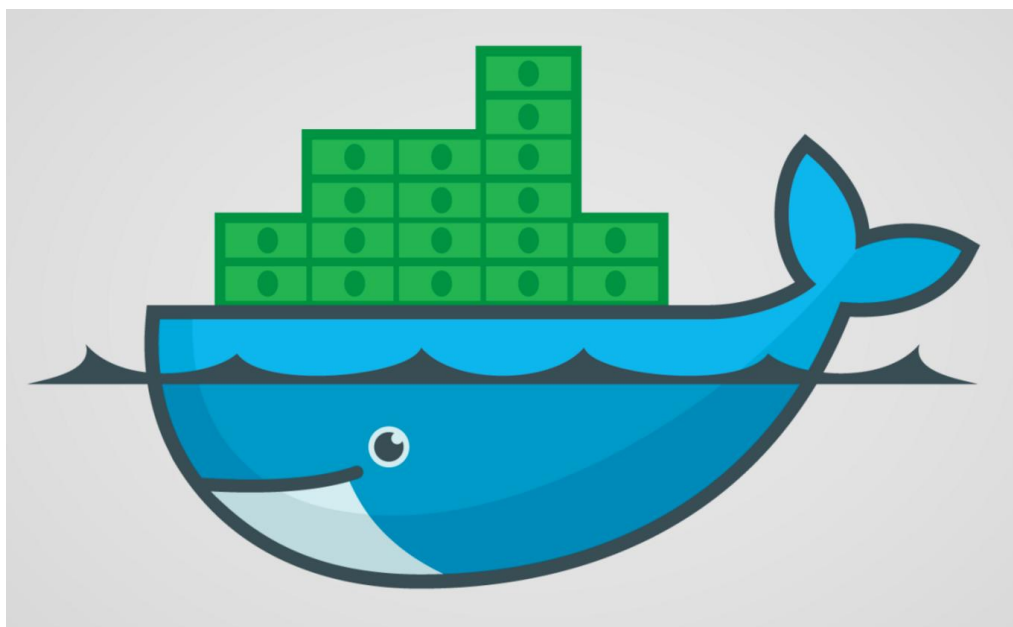


# 架构变迁三步走遵循要点

- 平滑演进，向后兼容
- 微服务，无状态化
- 多实例，硬件分离

# 第一步：Dockerize

- 无状态化应用
- 构建脚本和 Dockerfile
- 装入容器





# 最简单的 Dockerfile

```
# Base
```

```
FROM java:jdk-7
```

```
COPY ./src/target/app-1.0.jar /app/
```

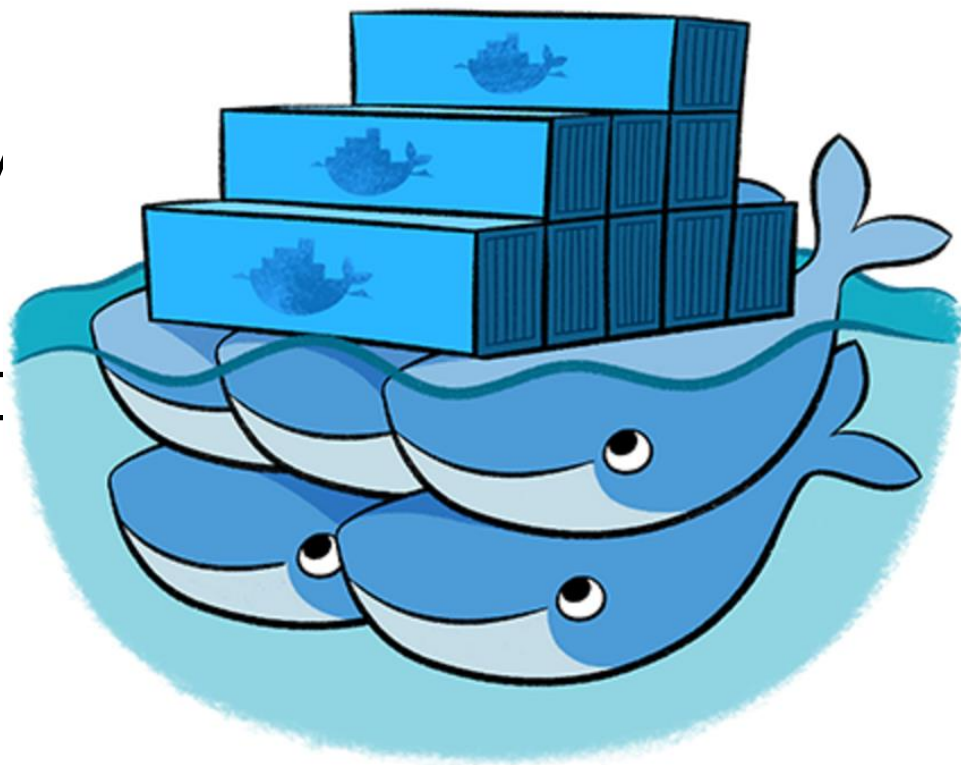
```
# ENTRYPOINT
```

```
WORKDIR /app
```

```
CMD [ "java", "-Dfile.encoding=UTF-8", "-jar", "./app-1.0.jar"  
]
```

# 第二步：管理你的容器

- (更直接) docker run / start , stop / restart
- (更灵活) docker remote API
- (更强大) 编排系统



# 管理工具的选择

- conf 文件配合 docker remote API
- 根据实际情况，选择 docker 的一些特性，例如文件系统，网络模式，资源限定等
- 编写便捷的操作工具 cli / web

# 配置示例

```
142 - name: repo-manager-repo1
143   image: repo-manager-repo1:latest
144   run_on_host: host-17
145   port: ""
146   envs:
147     - key: manager_port
148       | value: 8888
149     - key: http_port
150       | value: 8889
151     - key: max_conn
152       | value: 128
153   volumes:
154     - container_path: "/data/git"
155       | host_path: "/data/git"
156       | read_only: false
157   ulimits: []
158   log_config: null
159   entrypoint: ""
160   cmd: ""
161   network_mode: "host"
```

# 更新命令操作示例

Find running container: [/repo-manager-worker4\_1446547185]

Version: 9206804

Created: Nov 3, 2015 at 6:37:28pm

ImageID: ca79ee06f6d066c2ca0c84d497fe3f65ca9d123d66c0324f49179b08cdad239c

ContainerID: 0b322b6e26ef95762f880656abf9155948def39861209bc7aa2ee4b0e526e099

The tags in the registry

Version: 9206804      IMAGE ID: ca79ee06f6d066c2ca0c84d497fe3f65ca9d123d66c0324f49179b08cdad239c

CREATED: Nov 3, 2015 at 6:37:28pm

Version: latest      IMAGE ID: ca79ee06f6d066c2ca0c84d497fe3f65ca9d123d66c0324f49179b08cdad239c

CREATED: Nov 3, 2015 at 6:37:28pm

Version: 2a4fd3d      IMAGE ID: 2ab6954a23768a026b9c8fb3b9db0b9a80c77093dfb4ffbc40cc9dc5c044baf5

CREATED: Sep 15, 2015 at 4:32:54pm

Version: 7e6cc47      IMAGE ID: 217b7ca6ff9faf4af9b36fa5a41ddf5606d73655f6e87d9272fa9e8945bb63c2

CREATED: Sep 14, 2015 at 3:59:16pm

Version: a533163      IMAGE ID: ecac0d9a314c5f34eac4b60134867f79f81f71808102591c3f640b654cb7340f

CREATED: Aug 26, 2015 at 3:08:35pm

Version: 2e34403      IMAGE ID: beb5c5db2d8f025ea57220936f787b8ac2cb5ccba8c3ee250c298ae263b78e14

CREATED: Aug 20, 2015 at 5:43:50pm

Version: 58ad9a5      IMAGE ID: 8da495c2613f8308937fffec7ace9264424124a0d589ef611b9bca106cdc3416

CREATED: Aug 20, 2015 at 3:04:27pm

Version: c0d440d      IMAGE ID: 06e307ae8349fe71a336fb7dbc36c730d5a34447e8fa806925cad49689332eba

CREATED: Aug 19, 2015 at 3:57:14pm

Version: 5e194bb      IMAGE ID: a5d1bd1de8c209793daaf7f66c2cd81cadb1406f06a24ed219888c69cad5acd1



# DockerUI 界面

## DockerUI

Home

Containers

Images

Settings

Container: 9c8a34d00df172b317647d25529d3ba48560ea46c53247f7aa9214cb62d0537f

Start

Stop

Created: 2013-06-08T10:49:43.968798899-09:00

Path: /bin/sh

Args: ["-c", "/usr/local/bin/sentry --config=/sentry.conf.py start"]

SysInitpath: /usr/local/bin/docker

Image: [5886995bfd1827c82172e0b18642b1b8b3a27dfe7d49e3fde9ad81aa05b530ce](#)

Running: true

Remove Container

Michael Crosby

# 第三步：釜底抽薪

- 即使用 dockerize 的容器逐步替换系统中所有基础
- 包括，监控系统，负载均衡系统，服务发现，日志中心，消息中心等等基础业务组件
- 使计算存储分离，多实例，高可用，等这些概念有机结合

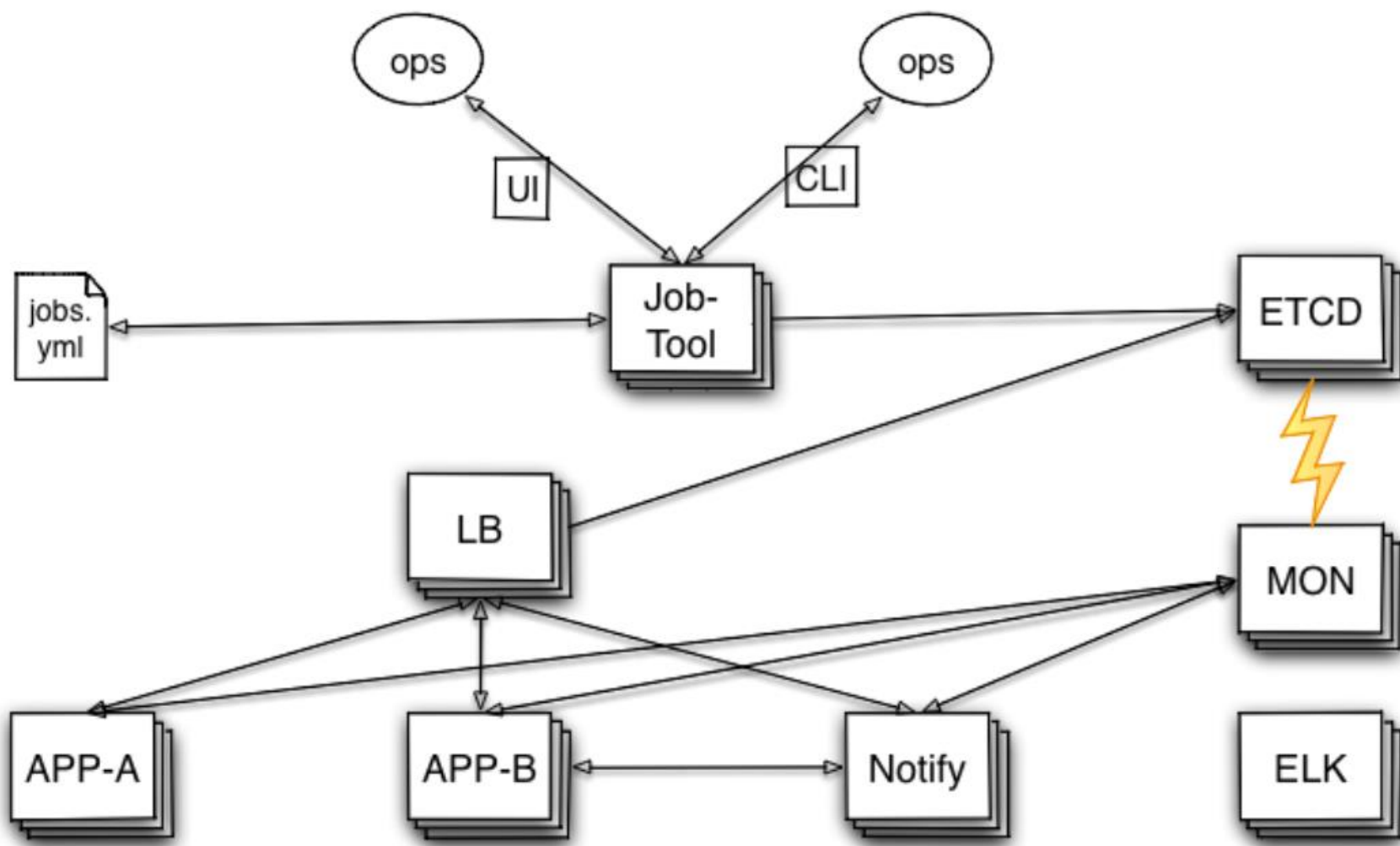




# 替换监控组件示例

CONTAINER ID	IMAGE	STATUS	PORTS	NAMES	COMMAND
8651b79572dd al 3 weeks ago	prom/prometheus:latest	Up 11 days	0.0.0.0:80->9090/tcp	prometheus	"/bin/prometheus -
6928d66e3c26 6 weeks ago	docker-registry.coding.local/alertsender:3e946f2-dirty	Up 6 weeks		alertsender_1444897710	"./alertsender"
865ed64cdb9d 6 weeks ago	docker-registry.coding.local/prom_node:0.11.0-6-g704e8f7	Up 4 weeks		node_mon_1444873993	"./node_exporter"
a95f0f1d75ed m/ 6 weeks ago	docker-registry.coding.local/cadvisor:latest	Up 6 weeks	0.0.0.0:8080->8080/tcp	cadvisor	"/go/src/github.co
dd1179534289 ig 6 weeks ago	docker-registry.coding.local/alertmanager:1.0	Up 6 weeks		alertmanager_1444613411	"/bin/go-run -conf
0fc7767f0a83 pu 6 months ago	prom/pushgateway:latest	Up 8 weeks		prom_pushgateway_1431145045	"/pushgateway/bin/

# 形成如下架构

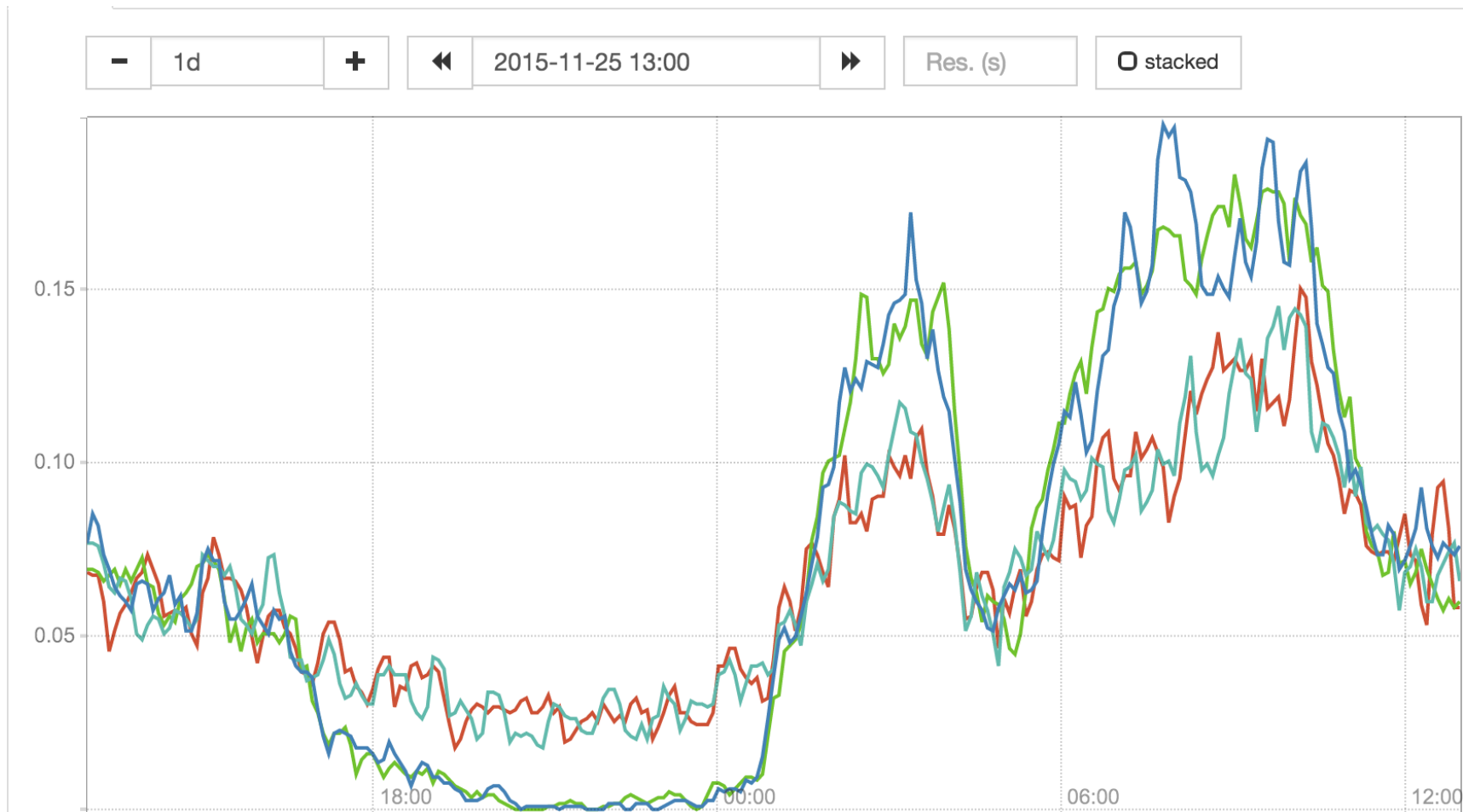


# 一些细节

- LB 系统：Nginx / HAProxy / confd / Etcd
- 监控系统：Prometheus / cAdvisor / Http Metrics
- Docker Registry V1
- Docker 网络：Host
- Docker 日志：Mount 宿主机



# Prometheus报表示例



# 架构的发展方向

- Job-Tool 进化成 Job DashBoard ，集成监控 (cAdvisor) ，日志(ELK)等功能
- 利用监控系统的硬件指标，根据业务用量实现自动扩容，缩容
- 分析各个业务对硬件资源的使用量和高低峰，设计混布实现提升硬件使用率
- docker image 的构建和管理
- 动态调整 container 的资源限制



# Docker 的问题

- Docker Image: 食之无味，弃之可惜
- Docker Daemon: 这货管得事太多了，还相当危险
- Docker Net: 容器就是容器，不是虚拟机
- Docker Logs: syslog 和 jsonlog 都不尽如人意



# Docker 的坑

- Docker 1.9.1 版本以下，容器标准输出输出大量数据，会导致内存泄露，从而导致 Docker Daemon crash
- Docker Daemon 在频繁创建删除容器（每天几十万个）会出现性能严重下降等问题，只能重启 Docker Daemon





# 标准输出问题

- 必要条件一：输出数据量大
- 必要条件二：输出数据快
- 必要条件三：输出被 Attach



# 标准输出问题

- 重现方式一： `docker run ubuntu yes "something long"`
- 重现方式二： `docker run -i ubuntu dd if=/dev/zero of=/proc/self/fd/1 bs=1M count=1000`
- Issue:  
<https://github.com/docker/docker/issues/14460>
- Fix By: <https://github.com/docker/docker/pull/17877>

# 并发性能问题

- 测试环境比较复杂，还在进一步研究中，欢迎各位共同研究



# Q&A