

The logo for SDCC 2016, featuring the text "SDCC 2016" in a bold, blue, sans-serif font. The background of the slide includes a light blue grid pattern, a large decorative graphic of blue dots on the left side, and several circular icons: an hourglass, a power button, a recycling symbol, a gear, a laptop, and a starburst.

中国软件开发者大会

SOFTWARE DEVELOPER CONFERENCE CHINA

## Docker应用

# Agenda

1. 基于容器的架构
2. Mesos相关技术使用
3. 遇到的一些问题

# 1. 基于容器的架构

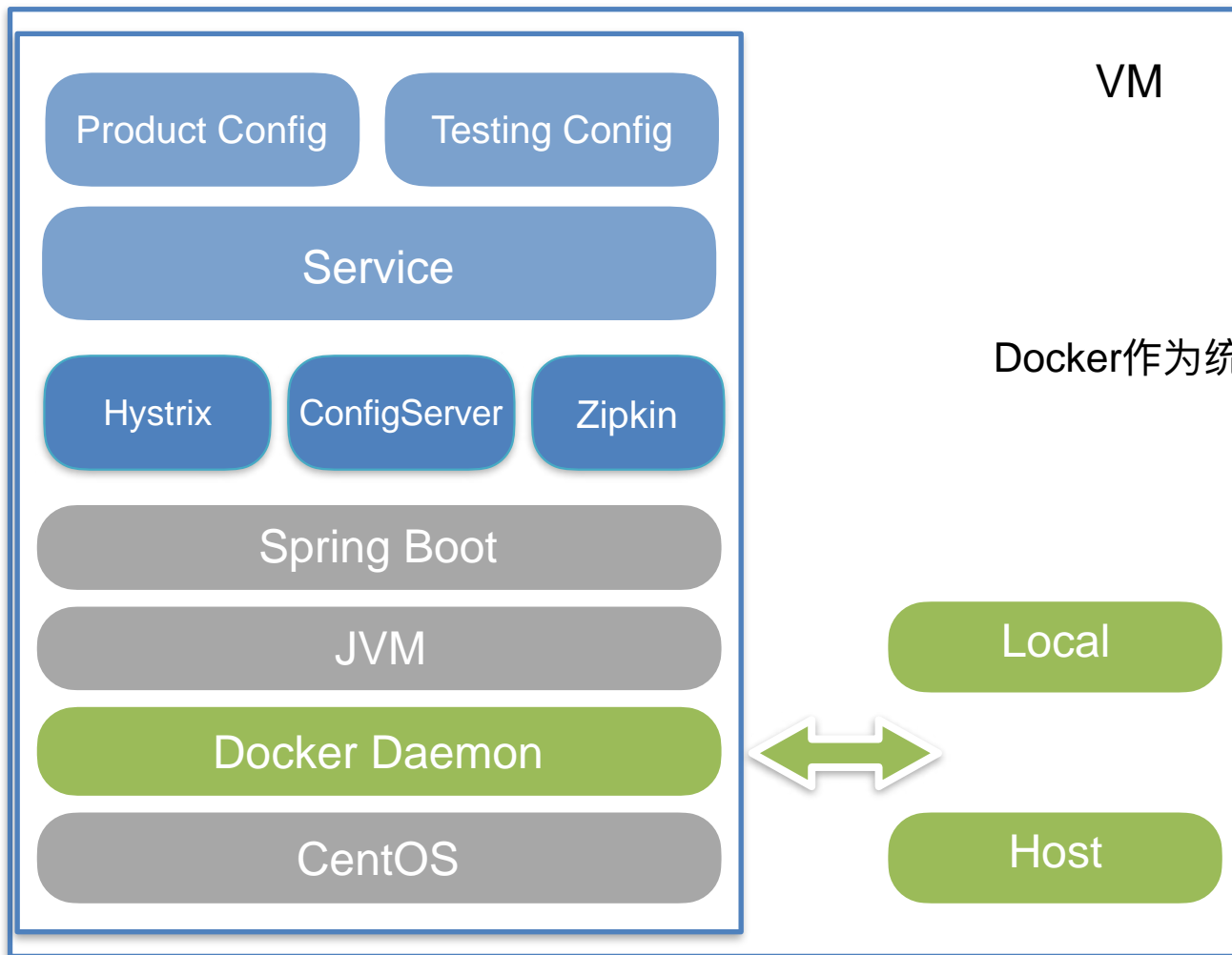


# 必备的组件

使用容器之后，基于VM上一些流程需要重新改造，才能提供基本的服务。我们的系统现有的基本组件如下：

- 容器化->包管理->CI/CD
- 配置管理
- 服务发现
- 日志收集
- 监控，报警
- 升级回滚策略
- 服务域名解析
- 负载均衡

# 单机模式



# Docker节点

App1

App2

AppN

Application Layer

LogStash

Zipkin  
Collector

Cadvisor

Node  
Explorer

Service Layer 监控、日志、性能

Docker Node

Storage

Dashboard

WebManager

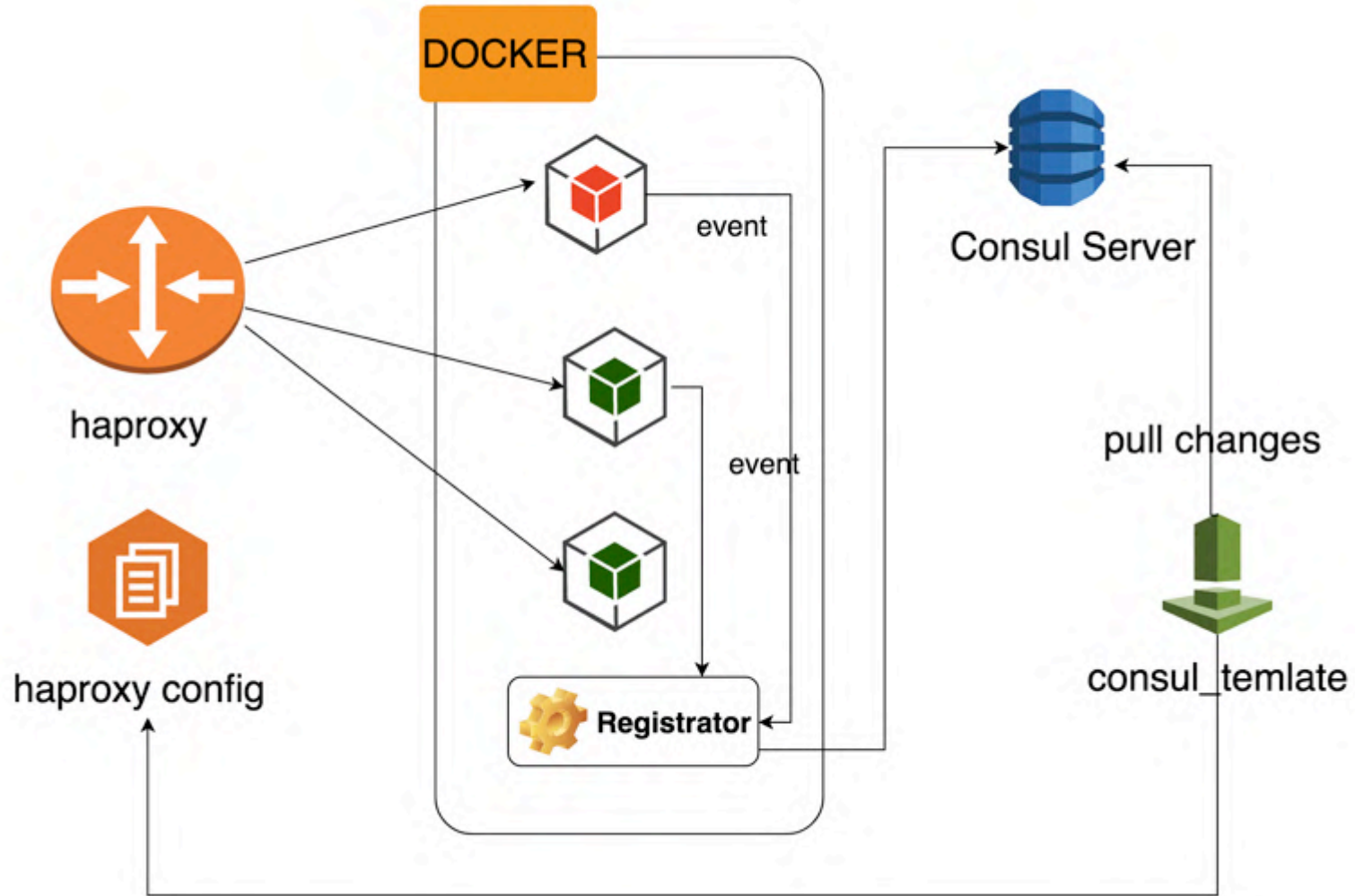
# 资源利用率

	VM	Mesos已分配	Mesos实际使用
CPU	< 5%	70%	30%
Mem	< 30%	80%	70%

- 之前的很多非核心服务，为避免单点，至少部署两个节点，但导致利用率很低
- 实际使用和分配的比例之间还有空间。Mesos提供的over-subscription可以进一步提高利用率



# 服务发现



# 日志收集策略

## 1. 异常日志

- 高优先级，第一时间处理

Email logger

## 2. 结构化的Info/Error日志

- 低延时，统计报警，实时分析

Logstash -> ES

## 3. 非结构化的Debug日志

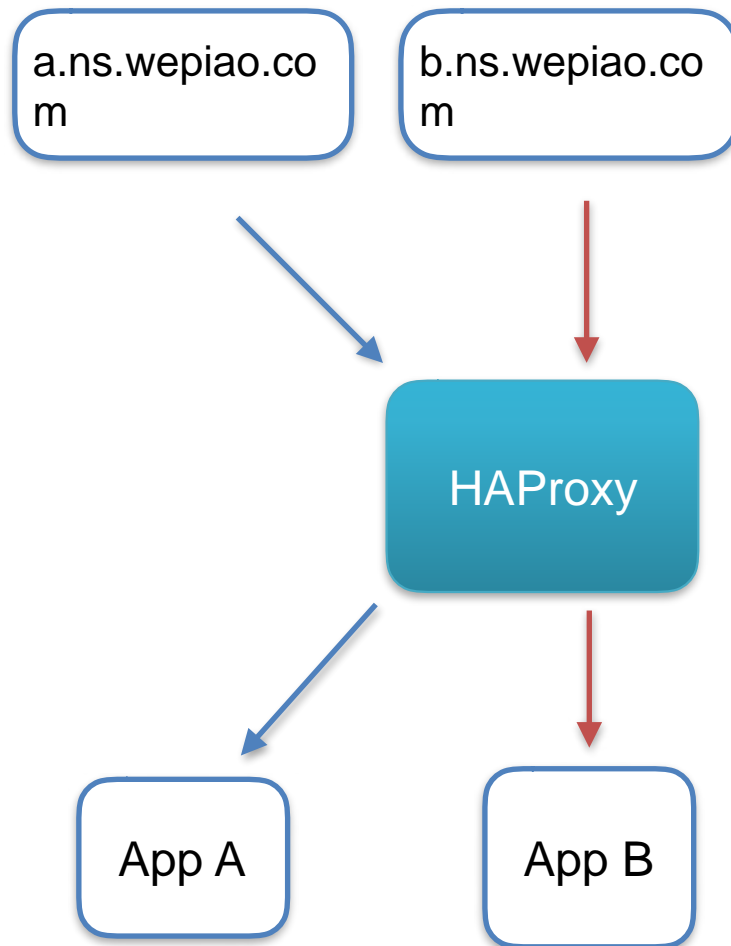
- 调试，备查，审计
- 轮转和集中存放

logstash -> archive

容器中的动态服务，不再提供ssh登陆的方式来查看日志

# 泛域名

- 泛域名解析到 \*.ns.wepiao.com
- 为测试环境，持续集成提供无限制的动态域名服务。不同工程和分支都有特定的域名对应
- 容器启动/停止事件会自动触发修改解析规则



# 微服务

- 拆大为小；各服务之间耦合小。每个服务的规模小到可控制的粒度，例如花几周可以重写某个服务
- 一定要有超时，熔断，隔离：hystrix
- 追踪请求依赖和耗时：requestId/zipkin

# 容器 vs 微服务 vs DevOps

- 容器本身只是一种技术，为上层建筑提供便利
- 微服务是软件架构上的新潮流
- 而DevOps更多关注的软件工程中的的人和流程的变化

# 一致性

- 核心业务：强一致mysql
- 服务之间有关联的数据：事后检查，业务监控，再恢复修补
- 幂等操作，调用者重试不会产生副作用

## 2.Mesos相关技术使用

# Mesos: 两层调度框架

Mesos Frameworks Agents Offers

Master 9384376f-b8d4-45a5-b3de-9c4f16a04a3d

Cluster: mesos\_prod\_cluster  
Server: 10.3.104.61:5050  
Version: 1.0.1  
Built: 2 months ago by centos  
Started: 4 weeks ago  
Elected: 4 weeks ago

LOG

Agents

Activated	7
Deactivated	0

Tasks

Staging	0
Starting	0
Running	21
Killing	0
Finished	33,987
Killed	52
Failed	3,101
Lost	3
Orphan	1

Resources

	CPUs	GPUs	Mem	Disk
Total	44	0	50.4 GB	101.9 GB
Used	17	0	15.3 GB	128 MB
Offered	0	0	0 B	0 B
Idle	27	0	35.1 GB	101.8 GB

### Active Tasks

ID	Name	State
movie-backend_hystrix_bis-gewara-turbine.09ee804e-a4b5-11e6-aecb-02424ef37bce	bis-gewara-turbine.hystrix.movie-backend	RUNNING
movie-backend_bis-gewara.0c50f6cd-a4b4-11e6-aecb-02424ef37bce	bis-gewara.movie-backend	RUNNING
movie-backend_bis-gewara.e515ef2c-a4b3-11e6-aecb-02424ef37bce	bis-gewara.movie-backend	RUNNING
movie-backend_hystrix_dashboard.7bce	dashboard	RUNNING
movie-backend_wp-pay.06a17e57-	pay.movie	RUNNING
od_sisyphus_worker.dd438425-a27d-11e6-aecb-02424ef37bce	worker.sisyphus.od	RUNNING
od_sisyphus_web.c80a2aa4-a27d-11e6-aecb-02424ef37bce	web.sisyphus.od	RUNNING
od_virtualbis_landlady.8f98e236-a260-11e6-aecb-02424ef37bce	landlady.virtualbis.od	RUNNING
od_virtualbis_grocer.6bd3bd00-a25f-11e6-aecb-02424ef37bce	grocer.virtualbis.od	RUNNING
movie-backend_bis-gateway.bd68ceeb-a1af-11e6-aecb-02424ef37bce	bis-gateway.movie-backend	RUNNING
movie-backend_print-ticket.b3b067-	ticket	RUNNING
od_unsale-redis-consumer.0c9408c-	sale-redis	RUNNING
marathon-lb.2b5f50c3-9f47-11e6-aecb-02424ef37bce	marathon-lb	RUNNING
marathon-lb.2b5f50c4-9f47-11e6-		RUNNING
marathon-lb.2b5f77d5-9f47-11e6-		RUNNING
chronos.55b497dc-944d-11e6-802d-0242aff04cd	chronos	RUNNING
chronos.5b7087ae-944e-11e6-802d-0242aff04cd	chronos	RUNNING
chronos.23b425c9-944d-11e6-802d-0242aff04cd	chronos	RUNNING

### Completed Tasks

ID	Name	State	Started	Stopp
od_virtualbis_landlady.8f98e236-a260-11e6-aecb-02424ef37bce	landlady.virtualbis.od	FAILED		3 day

Layer2: Frameworks

Layer1: Allocator

Resources

Marathon

Chronos

Jenki

Mesos

C

Memory

Disk

Network

VM



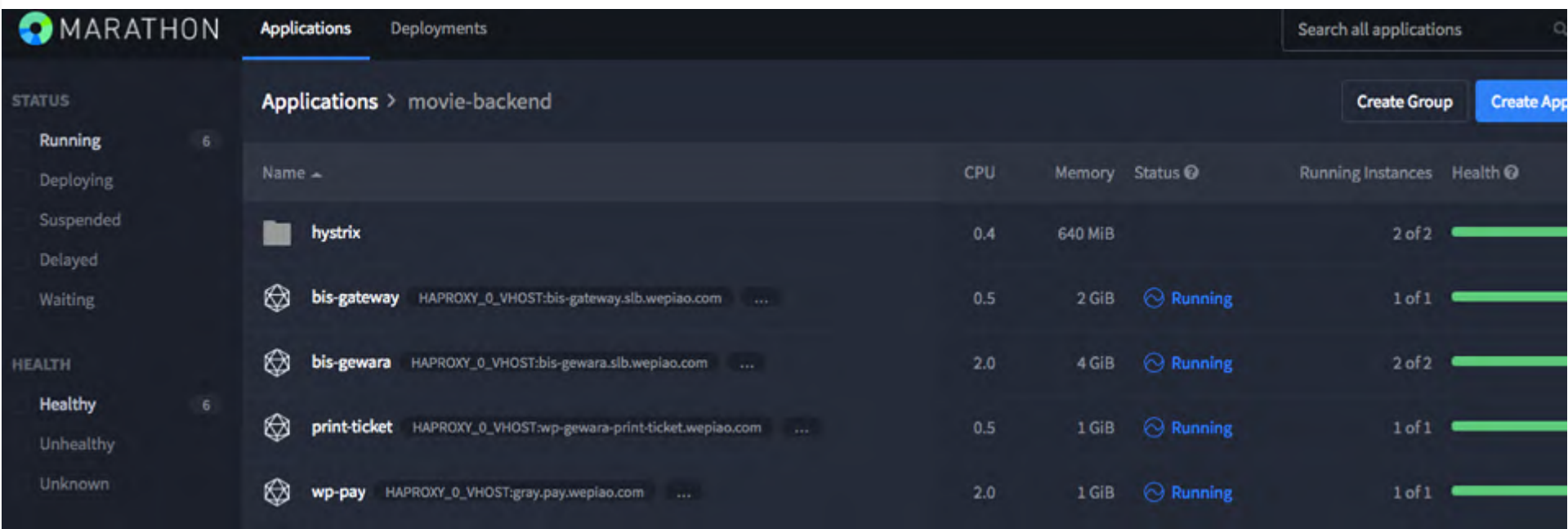
# Marathon

## 优点

- 运行long-running services
- 功能强大：蓝绿部署，健康检查，动态扩缩容，分组管理，服务发现，滚动升级等

## 不足

- 业务隔离，分组的功能欠缺



The screenshot displays the Marathon web interface. The top navigation bar includes 'MARATHON', 'Applications', and 'Deployments'. A search bar on the right is labeled 'Search all applications'. The main content area is titled 'Applications > movie-backend' and features a table of running applications. The table columns are Name, CPU, Memory, Status, Running Instances, and Health. The applications listed are hystrix, bis-gateway, bis-gewara, print-ticket, and wp-pay, all in a 'Running' state with green health bars.

Name	CPU	Memory	Status	Running Instances	Health
hystrix	0.4	640 MiB		2 of 2	<span style="color: green;">█</span>
bis-gateway	0.5	2 GiB	Running	1 of 1	<span style="color: green;">█</span>
bis-gewara	2.0	4 GiB	Running	2 of 2	<span style="color: green;">█</span>
print-ticket	0.5	1 GiB	Running	1 of 1	<span style="color: green;">█</span>
wp-pay	2.0	1 GiB	Running	1 of 1	<span style="color: green;">█</span>

# Chronos vs. Cron

## 优点

- 分布式环境工作，定时任务不再有单点问题
- 简单易用，稳定
- API功能完整

## 不足

- 界面太简单，甚至有bug
- 对容器环境的任务只能通过API操作
- 控制时间的方式单一
- 缺少分组管理和权限管理
- 自己包装前台页面，来调用API完整定制化的需求

The screenshot displays the Chronos web interface. The top header shows 'CHRONOS ...' and 'TOTAL JOBS 12 FAILED JOBS 0'. Below this is a search bar and buttons for 'Graph' and 'New Job'. The main content is a table of jobs with columns for NAME, LAST, and STATE. The job 'update-static-seat-redis' is highlighted. To the right, a detailed view of this job is shown, including its name, description, command, owner information, last success and error times, and a schedule.

NAME	LAST	STATE
firephoenix_snacks_docker	fresh	idle
redis-unsale-check	SUCCESS	idle
cinema-match-audit	SUCCESS	idle
update-static-seat-redis	SUCCESS	idle
massterma_makeplan	SUCCESS	idle
update-alive-attendance	SUCCESS	idle
update-past-attendance	SUCCESS	idle
prod-snacks-firephoenix	SUCCESS	idle
pre-snacks-firephoenix	SUCCESS	idle
test-snacks-firephoenix	SUCCESS	idle

**Job Details: update-static-seat-redis**

DESCRIPTION: 定时更新影厅静态座位Redis

COMMAND: [REDACTED]

OWNER(S): [REDACTED]

OWNER NAME: [REDACTED]

LAST SUCCESS: 2016-11-07T06:00:57.122Z (57)

LAST ERROR: 2016-11-02T14:14:33.138Z (1)

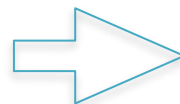
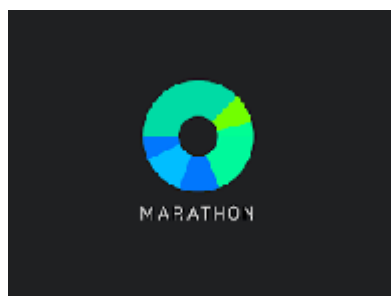
SCHEDULE: R/2016-11-07T06:00:00.000Z/PT2h

JOB RUNTIME (PERCENTILES):  
50th: 57.12 seconds    75th: 57.12 seconds    90th: 57.12 seconds

# Jenkins on Mesos

利用mesos集群的资源，而不再需要单独维护jenkins集群

1. 使用Marathon来维护Jenkins Master的稳定
2. Jenkins作为Mesos上的一个Framework来获得集群的资源
3. 最后通过Docker container把任务运行在Mesos Slave上



# 遇到的一些的问题

- 容器社区“百花齐放”有好处也有坏处
  - 三分天下，mesos/k8s/swarm，都需要投入精力
  - 各种框架，方案大同小异，选型，评估，试错的成本都不低
- 核心服务应用较少，被验证的程度尚浅
- 同时维护两套流程：VM vs. 容器
- 疑难杂症（重启，重装，升级，有不少问题我们找不到问题）
- 需要开发可视化的管理界面，而不是命令行
- 多租户：不同生产服务之间互不影响。mesos 的role/weight/reservation等概念太底层。需要更高层的管理工具
- 分离的集群：kafka, es, hadoop, spark等。每个集群都需要单独维护。有状态的服务合并到集群还不是很成熟
- HAProxy默认配置转发部分中文url出400错误。需要开启兼容一些上太不合法的url
- 蓝绿部署的过程中，CDN回源，导致js/css和html指向了两个不兼容的版本
- 容器分配的内存不足，会被突然杀死



**SDCC 2016**

**中国软件开发者大会**

SOFTWARE DEVELOPER CONFERENCE CHINA

谢谢!