upyun

upyun | IT大咖说 不止于技术

! DevOps ?

upyun

upyun | IT大咖说 不止于技术

- DevOps的八荣八耻
- Heroku PaaS的12要素宣言
- OpenStack IaaS的设计指导思想

upyun

1 以可配置为荣 以硬编码为耻

upyun | IT大咖说 不止于技术

程序生成程序

提炼应用参数

/etc/upyun.cfg

```
NGINX_UPSTREAM="
192.168.251.163#1
192.168.251.164#2
192.168.251.165#3
192.168.251.166#4
192.168.251.167#5"
```

/etc/init.d/nginx config

动态生成配置文件

/opt/nginx/conf/upstream.conf

```
upstream  hot.yupoo.com {
        server 192.168.251.163:8100 weight=1;
        server 192.168.251.164:8100 weight=2;
        server 192.168.251.165:8100 weight=3;
        server 192.168.251.166:8100 weight=4;
        server 192.168.251.167:8100 weight=5;

        consistent_hash     $uri        2;
        }
```

- 各种开关变量,可调参数,上下游关系

upyun

1 以可配置为荣 以硬编码为耻

upyun | IT大咖说 不止于技术

- 本地配置,程序生成 (txt/ini/cfg)
- 集中配置, 动态生成(Yaml/Json)
- 环境变量(代码无侵入&语言无关性)
- 服务自动发现,自动注册(etcd/consul)

经济原则 (宁花机器一分,不费程序员一秒)

upyun

2 以互备为荣 以单点为耻

upyun | IT大咖说 不止于技术

- LVS + Keepalived + VRRP
- OSPF + Zebra + Quagga
- Nginx + Haproxy / LVS
- Mysql master/slave/Galen MMM
- rabbitmq/redis/kafka
- consul / etcd / doozer / zookeeper
- hadoop/elk

墨菲定律: “凡事只要有可能出错，那就一定会出错。”

upyun

2 以互备为荣 以单点为耻

- 无状态/负载均衡(F5,LVS,Haproxy...)
- 无共享/消息队列(Redis,Kafka...)
- 松耦合/异步处理(Gearman,Celery...)
- 分布式/集群协作(Hadoop/Ceph...)

upyun

3 以随时重启为荣
以不能迁移为耻

upyun | IT大咖说 不止于技术

- Pet -> Cow 故障是常态
- OpenStack 虚拟机编排
- LXC/RKT Docker导入导出

墨菲定律: “凡事只要有可能出错，那就一定会出错。”

upyun

4 以整体交付为荣 以部份交付为耻

upyun | IT大咖说 不止于技术

- OpenStack - 云计算, 云网络, 云存储
  统一调度分配资源
  任何资源都是可度量
  在线使用,回收,备份,重载
- Docker - Build, Ship, and Run
  统一配置,统一入口
  开发,测试,运维三位一体
  随时随地调用

运维/研发,从OM->OB,类服务商PaaS

upyun

5 以无状态为荣
以有状态为耻

upyun | IT大咖说 不止于技术

- 资源竞争
- 锁竞争
- 横向扩展差
- 不能重启/互备

扩展原则 (设计着眼未来，解耦+负载均衡)

upyun

5 以无状态为荣
以有状态为耻

upyun | IT大咖说 不止于技术

- 可靠中间件(生产级/工业级/分布式)
- 公共资源池 (云XX/容器云/弹性云)
- 能够被计算(ceph/paxos, etcd/raft)

upyun

6 以标准化为荣 以特殊化为耻

upyun | IT大咖说 不止于技术

- 统一输入输出(git/ansible)
- 统一流程控制(yaml/playbook)
- 抽象底层设计 -> 复用组件 (函数库)
- PC , 容器服务, server-less = 微服务

香农-信息熵理论:

变量的不确定性越大，熵也就越大，把它搞清楚所需要的信息量也就越大。
所以一个孤立的系统，始终会趋向于越来越乱（无序化）的方向发展。

upyun

7 以自动化工具为荣 以手动 + 人肉为耻

upyun | IT大咖说 不止于技术

- bash/sed/awk/rsync/fabric(业务模型标准化)
- puppet/ansible playbook
- tgz/rpm/pkg 打包部署
- 集成测试/自动测试/发布
- ELK 集中日志分析/大数据分析
- 不要图形,不要交互,不要终端

生成原则 (避免手工hack,程序生成程序)

upyun

8 以无人值守为荣 以人工介入为耻

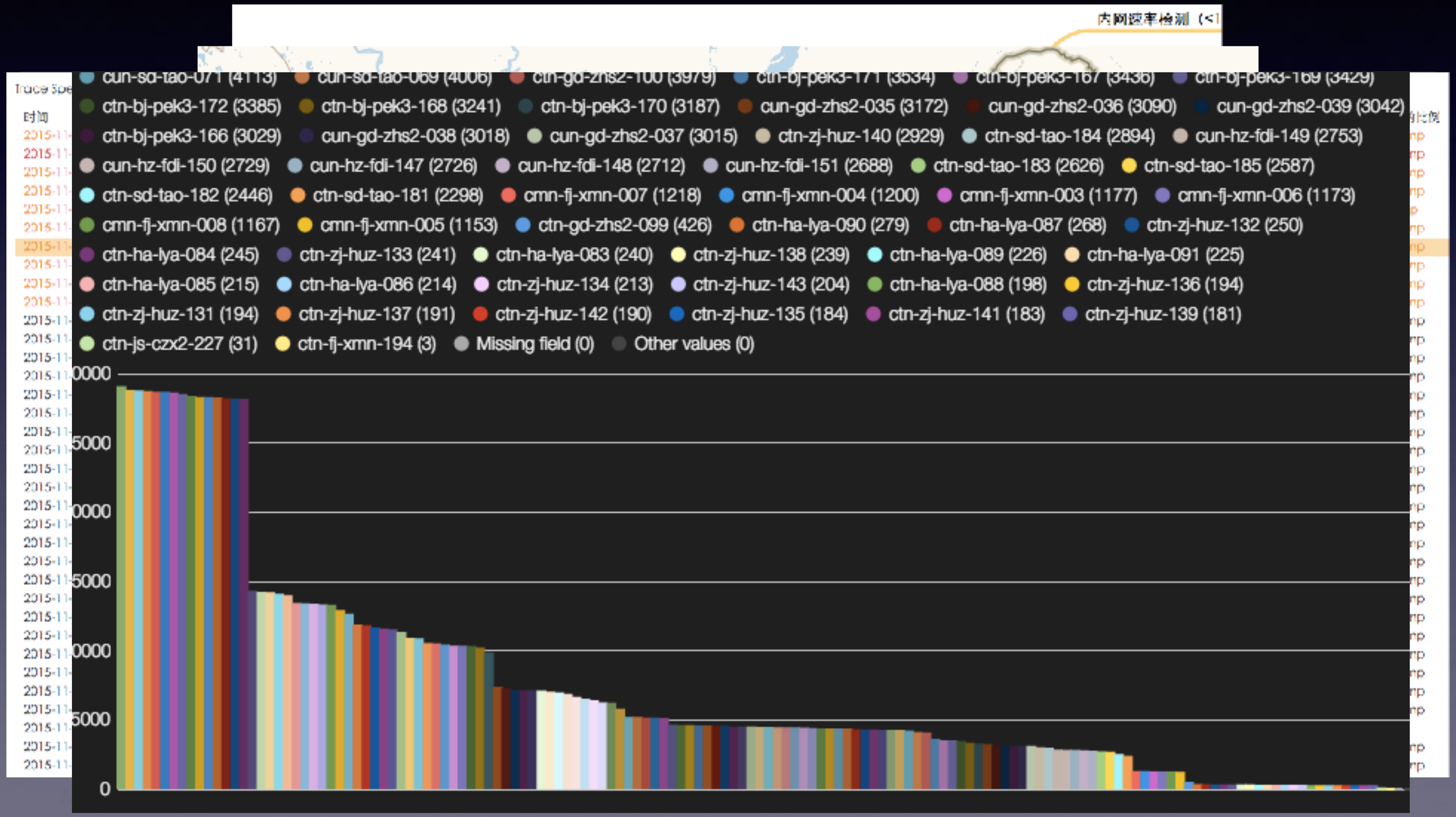
upyun | IT大咖说 不止于技术

- 运维自动化(标准化组件,标准化流程)
- 监控常态化(同构大数据分析/处理)
- 性能可视化(数据的有效展示)

节省时间和精力　　从容应付业务增长　　保持稳定的执行效率

upyun

upyun | IT大咖说 不止于技术



upyun

upyun | IT大咖说 不止于技术

• Heroku PaaS的12要素宣言

I. 基准代码 - 一份基准代码，多处部署(标准化,复用组件,函数库)
II. 依赖 - 显式声明依赖关系(nodejs/go/maven)
III. 配置 - 在环境中存储配置
IV. 后端服务 - 把后端服务当作附加资源(url/服务发现)
V. 构建，发布，运行 - 严格分离构建和运行 (统一性,标准化,模板/容器)
VI. 进程 - 以一个或多个进程, 必须无状态且无共享 (任何需要持久化的数据都要存储
端服务内)
VII. 端口绑定 - 通过端口绑定提供服务(Restful/http)
VIII. 并发 - 通过进程模型进行扩展 (方便扩展/分布式/弹性)
IX. 易处理 - 快速启动和优雅终止可最大化健壮性(雅虎准则)
X. 开发环境与线上环境等价 - 尽可能的保持开发、预发布、线上环境相同
XI. 日志 - 把日志当作事件流(rsysylog/heka/logstash流式收集/过滤/处理/展示)
XII. 管理进程 - 后台管理任务当作一次性进程运行(server-less,微服务)

upyun

upyun IT大咖说 不止于技术

• No.1 OpenStack IaaS的设计指导思想

- 可扩展性和伸缩性是我们的主要目标;
- 任何影响到可扩展性和伸缩性的功能都必须是可选的;
- 所有的环节必须是异步的,如果不能异步,参考第二条原则;
- 所有的基础组件必须能横向扩展;
- 始终使用无共享的架构,如果不能实现,参考第二条原则;
- 所有的组件部署形式都是支持分布式的;
- 接受最终一致性,并在适合的条件下使用;
- 测试一切;

upyun

upyun | IT大咖说 不止于技术

DevOps的本质

- 弹性(横向扩展,负载均衡,资源复用,编排回滚...)
- 无关性 (硬件,系统,网络,抚平差异,皆面向服务)
- 不可变基础设施(虚拟化,容器化,SDN,随取随用)