

# 运维自动化“私人定制”

Shared from UPYUN Inc.

**Contributor:** @又拍云存储

**Speaker:** 邵海杨 @海洋之心-悟空

一个真正的强者，不是摆平了多少人，而要看他能帮助多少人



# 运维自动化的keypoint

没有单点故障

系统统一，文件统一...

扩展性好，松耦合

★ 像程序员一样思考运维

直观的业务量化指标

好用的日志分析

争取冗余的资源

# 框架 vs 定制

常见运维工具

自己定制工具

Puppet

Chef

Ansible

Saltstack

bash

sed

awk

定制agent

移植性极佳

学习成本

通用 vs 专用

高大上 vs 短平快

# example-smart

```
awk ' {  
    split($7, a, "/");  
    b[a[3]] += $10  
END {  
    for(i in b) print  
i, b[i]  
}' \  
/var/logs/upyun.access.log \  
g \  
| sort -k2nr \  
| head -30
```

```
===== [REDACTED] (2014_03_23) ] =====  
  
----- Cache Analysis Report -----  
Hot Objects:      = 586914 (71.42%)  
Hot times:        = 103305 (176.01 ms)  
Cache Objects:    = 234880 (28.58%)  
Cache times:      = 122387 (521.06 ms)  
  
New Objects:      = 17230 (2.10%)  
New times:        = 12404.6 (719.94 ms)  
  
----- Cache Object Size -----  
Hot Objs Size:    = 115.78GB (124317450900)  
Cache Objs Size:  = 64.91 GB (69695680072)  
New Objs Size:    = 3.90 GB (4185483363)  
  
----- Response Code Status -----  
code 408          = 2      (0.00%)  
code 200          = 682594 (83.08%)  
code 301          = 1      (0.00%)  
code 403          = 1      (0.00%)  
code 304          = 2405   (0.29%)  
code 404          = 984     (0.12%)  
code 206          = 135586 (16.50%)  
code 416          = 70      (0.01%)
```

# example-II

```
echo "----- Check Request -----"
```

```
tail -n $NUM $NGINX_LOG | grep "HTTP/1.[01]" > $NGINX_LOG.xxx
```

```
echo "----- Response Codes -----"
```

```
awk 'BEGIN{FS="\\"} { \
    split($3,tmp," ");++code[tmp[1]]} \
    END{ \
        for(i in code){\
            printf("%-4s = %-8s\n",i,code[i]);\
        }\
    }' $NGINX_LOG.xxx > /tmp/.curl_codes
```

```
echo "----- Response time -----"
```

```
awk '{ \
    if($NF-3~/hot/) {hot_time+=$NF-5;++time_hot;}\
    else if($NF-3~/cache/) {cache_time+=$NF-5;++time_cache;}\
    else if($NF-3~/local/) {local_time+=$NF-5;++time_local;}\
    else {other_time+=$NF-5;++time_other;}\
}\
END{ \
    printf("hot:(%d)\t%-4s = %-8.2fms\n"cache:(%d)\t%-4s = %-8.2fms\nlocal:(%d)\t%me*1000/time_hot,time_cache,cache_time,cache_time*1000/time_cache,time_local,local_time\
    }' $NGINX_LOG.xxx > /tmp/.curl_time
```

```
code_num=`awk '/^50[024]/{if($3>100) print $3}' /tmp/.curl_codes`
```

```
if [ ! -z $code_num ];then
```



# 一次编译-到处运行

全代码完整打包（二进制/库）

方便统一部署

提供shell非交互式的命令控制

管道流转强大

精简系统平台或镜像虚拟化

简化流程  
转嫁成本  
升级方便

标准化组件（拒绝拖延）

# 程序的本质

- 二进制文件(ELF)
- 库文件(.so)
- 头文件(.h) <- 非必须



• 配置文件



程序

```
[root@OS-NODE1 boot]# whereis bash
bash: /bin/bash /usr/share/man/man1/bash.1.gz
[root@OS-NODE1 boot]# file /bin/bash
/bin/bash: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically
linked (uses shared libs), for GNU/Linux 2.6.18, stripped
[root@OS-NODE1 boot]# ldd /bin/bash
linux-vdso.so.1 => (0x00007fff9c150000)
libtinfo.so.5 => /lib64/libtinfo.so.5 (0x00007f925d2c1000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f925d0bd000)
libc.so.6 => /lib64/libc.so.6 (0x00007f925cd29000)
/lib64/ld-linux-x86-64.so.2 (0x00007f925d4ee000)
```

# 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;    }
```



# example-1

**/etc/init.d/nginx config**

再来十遍也一样！

```
config() {  
> $NGINX_DIR/conf/upstream.confif [ ! -z "$NGINX_UPSTREAM" ];then      xx=$IFS  
for up in $NGINX_UPSTREAM;do      IFS=#;read -r ip weight <<< "$up"  
STRING=$STRING"\tserver $ip:8100 weight=$weight fail_timeout=3s;\n"  
STRING2=$STRING2"\tserver $ip:8200 weight=$weight fail_timeout=3s;\n"      done  
IFS=$xx      echo -en "\n$STRING\n\tconsistent_hash \${uri} 2;\n\t}\n" >>  
$NGINX_DIR/conf/upstream.conf      echo -en "\n$STRING2\n\tconsistent_hash \${uri}  
2;\n\t}\n" >> $NGINX_DIR/conf/upstream.conf} }
```

# example-II

./gen\_ansible.sh CUN-HL-SWI lists\_cdn\_new  
255.255.255.224

# heilongjiang suihua liantong

[CUN-HL-SWI]

# [REDACTED] # VIP

CUN-HL-SWI-163 ansible\_ssh\_host=[REDACTED]

CUN-HL-SWI-164 ansible\_ssh\_host=[REDACTED]

CUN-HL-SWI-165 ansible\_ssh\_host=[REDACTED]

CUN-HL-SWI-166 ansible\_ssh\_host=[REDACTED]

CUN-HL-SWI-167 ansible\_ssh\_host=[REDACTED]

**ansible -i** lists\_cdn\_new CUN-HL-SWI -m shell -a "

**sed -r -i -e**

'/NGINX\_UPSTREAM=/s:.\*: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\"':g' /etc/upyun.cfg" -

f10

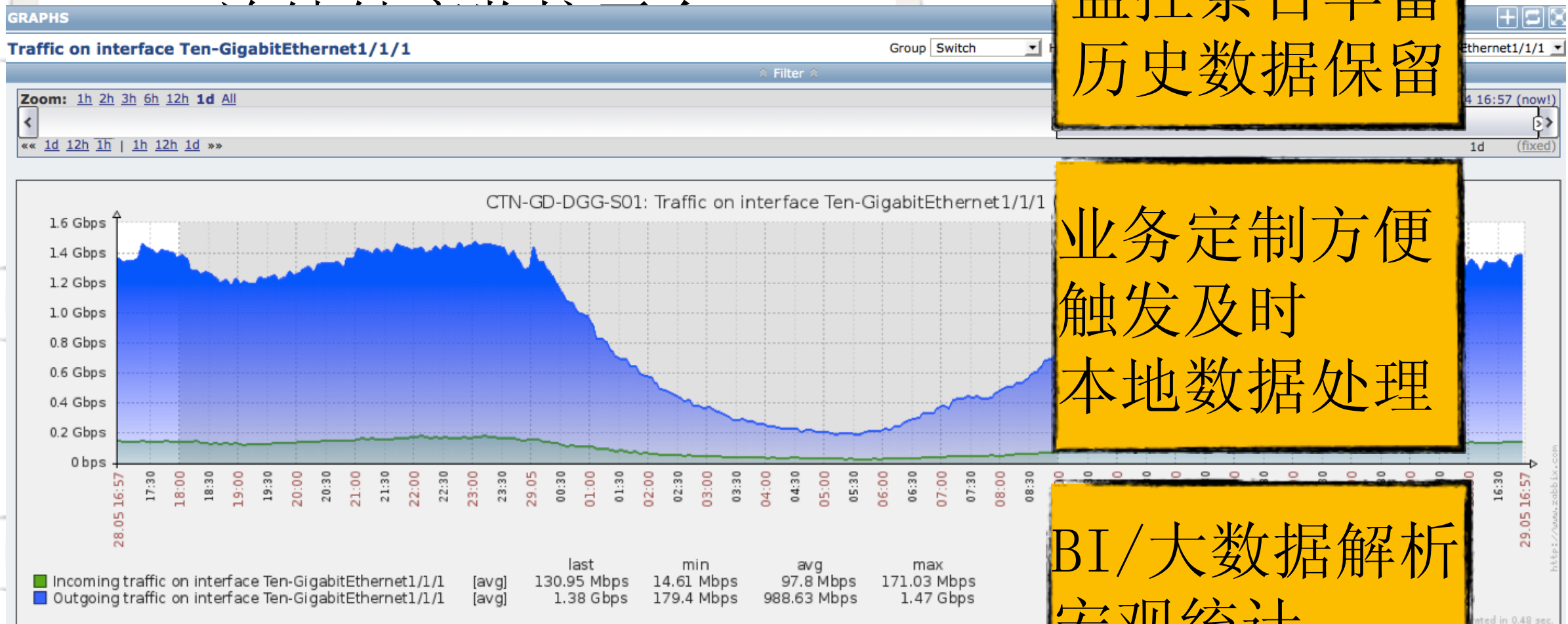
流程化部署，谁用都一样

# 监控常态化

监控条目丰富  
历史数据保留

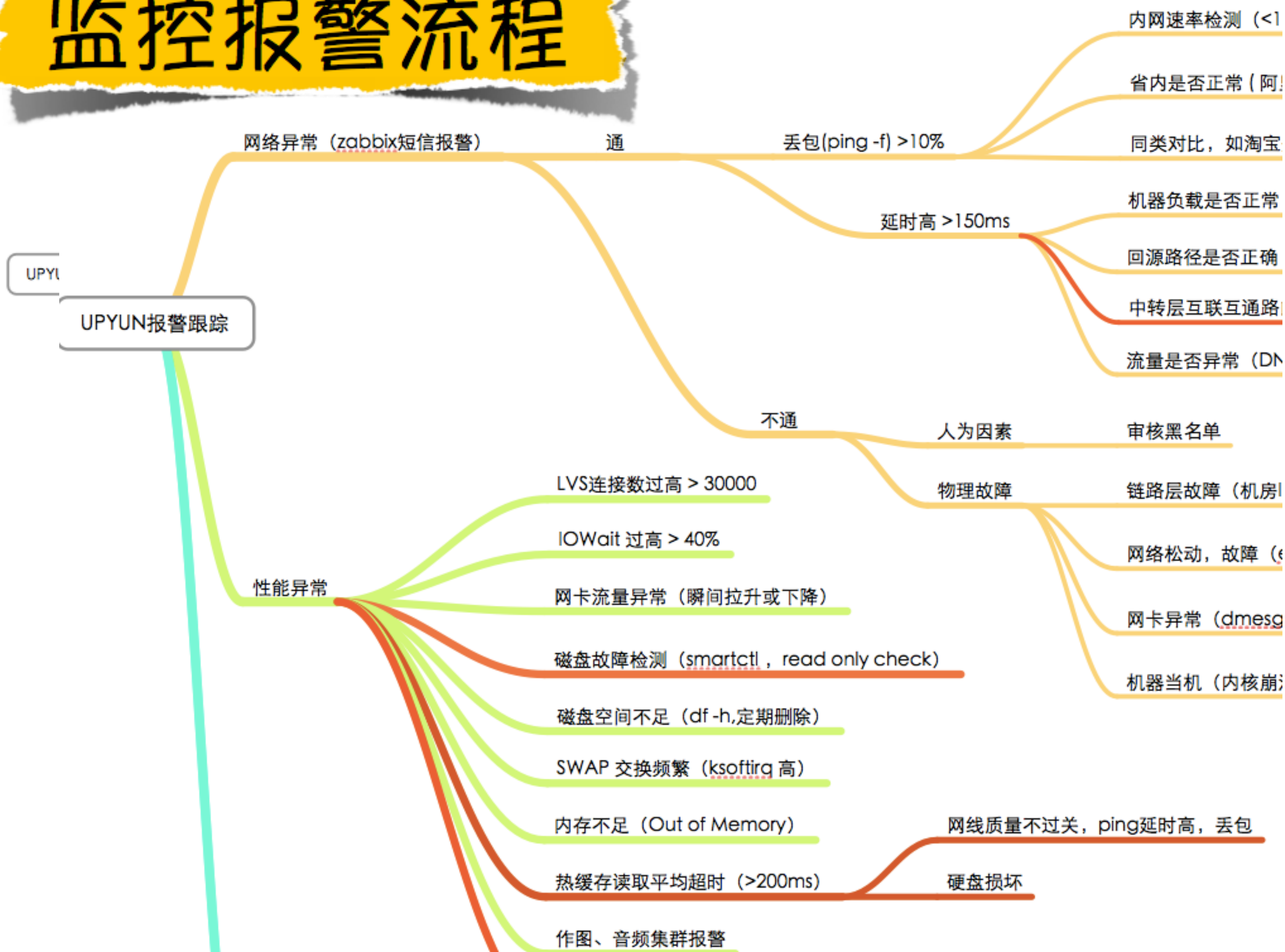
业务定制方便  
触发及时  
本地数据处理

BI/大数据解析  
宏观统计



三位一体， 智能化检测

# 监控报警流程



# example-事件响应

第一人称脚本监控突发事件截图

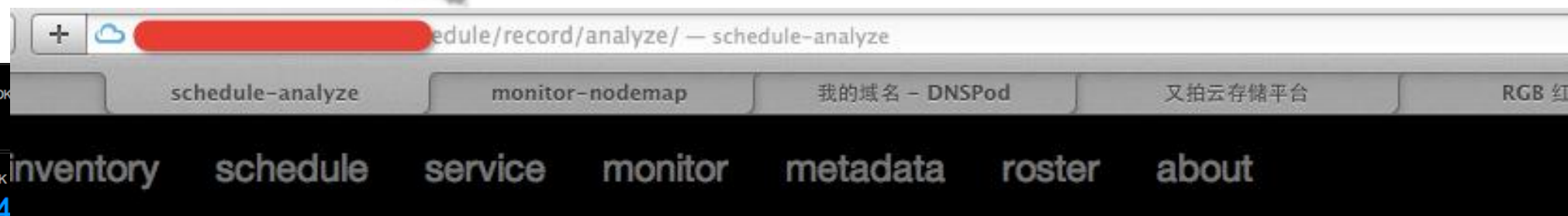
★让机器像人一样有感知

```
check_irq(){  
check_dns(){  
check_log(){  
check_iowait(){  
check_gzip(){  
check_cdn_val(){  
check_cdn_dns(){  
check_network(){  
check_ethx(){  
check_lvs(){  
check_action(){  
check_purge(){  
check_marco(){  
check_message(){  
check_kernel(){  
check_redis(){  
check_sysctl(){  
check_ats(){  
check_baddisk(){  
check_nginx(){  
check_uptime(){  
check_src(){
```



# 性能可视化

FS_READ 6/2	#FS_READ 20/0	FS_READ_OK 5/2
PHP_API 540/486	#PHP_API 63/280	PHP_API_OK 436/4
FTP 184/63	#FTP 36/211	FTP_GET 2/0
#KUZAN-GET-TIME 93/92	#KUZAN-PUT-TIME 18/16	KUZAN-MYSC 37/45
KUZAN-FS-PUT 257/212	KUZAN-FS-PUT-FAIL 7/5	KUZAN-200 417/6
PUT_THUMB 263/230	#PUT_THUMB 13/75	PUT_THUMB 18/44
PUT_AV 2/3	#PUT_AV 2701/4776	PUT_AV_OK 2/4
GM 496/595	#GM 52/53	GM-200 490/5
#A_AVG_TIME 0/0	A_OK 0/0	A_B_OPTION 0/0



SCHEDULE
Record
Analyze
Backup
Immigrate
Returnee




# 结对编程

人员互备，消

redmine+gi

运维平台（封装

 upyun-core / Usopp

Search in this project

FilesCommitsNetworkGraphsIssues 1Merge

develop usopp

Name	Last Update	Last Commit > 71330e418f2 – Merge branch 'feature/tests' into 'develop'
tests	about a month ago	morndust add a configuration file to configure the ftp server and the ftp account
upyun	8 months ago	timebug Usopp v1.2.9
.gitignore	about a month ago	morndust add a configuration file to configure the ftp server and the ftp account
README.md	4 months ago	morndust WorkAround: a little fix
config.py	2 months ago	morndust add a configuration on maximum size of upload file
requirements.txt	8 months ago	timebug [php-api] update requests 2.0.1 fixed Content-Length:0 411 issues
upftpd.py	about a month ago	Monkey Zhang bumped version to v1.3.2
usopp.py	4 months ago	morndust WorkAround: a little fix

README.md

## Usopp

### 基础环境配置

需要 Python 2.6.x 或 2.7.x 环境

```
+ pyftplib==1.2.0
+ pysendfile==2.0.0
+ requests==2.0.1
+ setproctitle==1.1.7
```

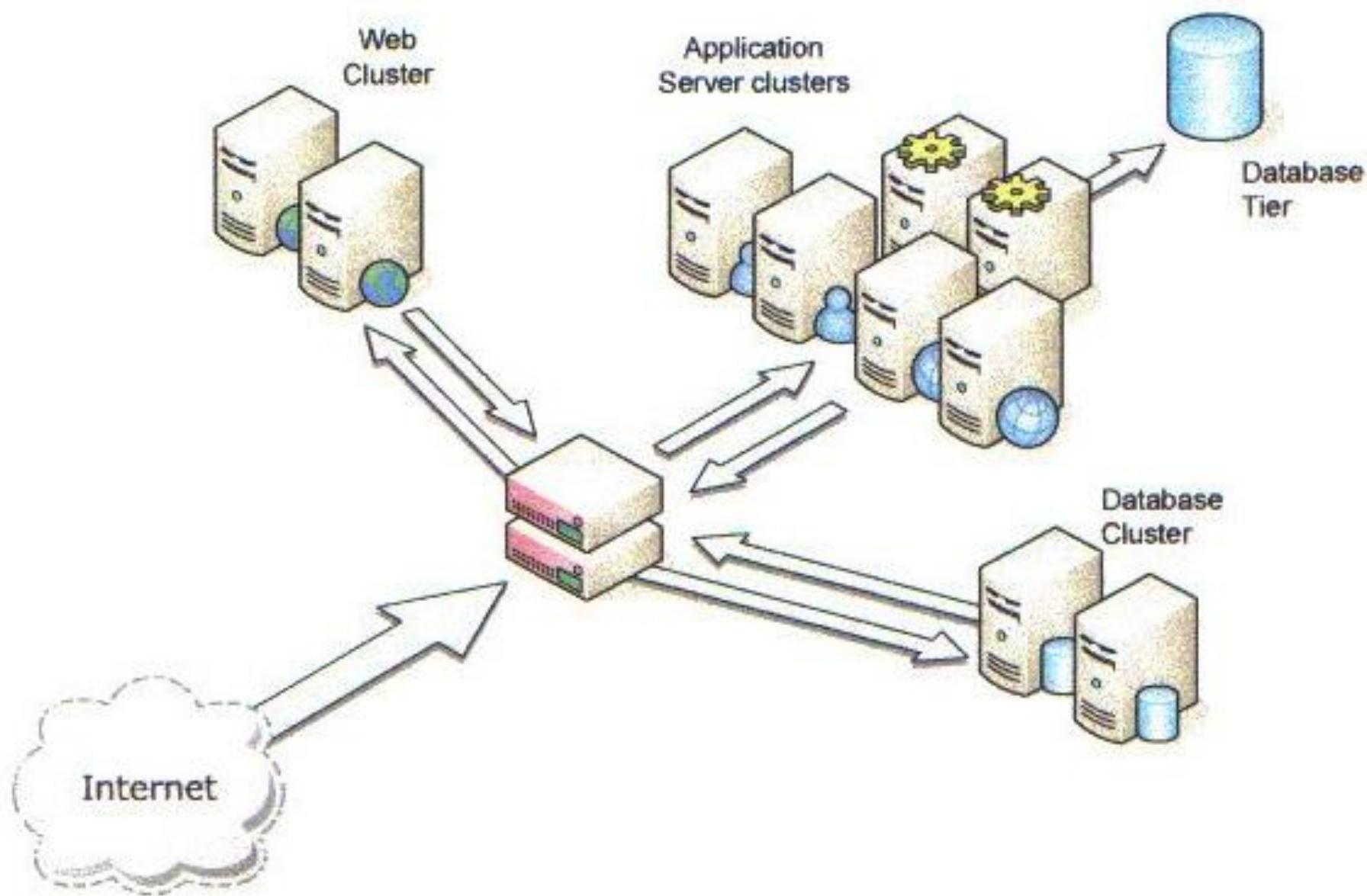
若系统自带 Python 版本过低，为了避免影响到其他 Py 应用，建议安装 pythonbrew & virtualenv 来虚拟化一个新的 Python 环境：

- 1.安装 Pythonbrew

```
# curl -kL http://xrl.us/pythonbrewinstall | bash
```

- 2.编辑 ~/.bashrc

# 运维的魔咒怎么破



★ 让人像架构一样可扩展

# 运维的指导思想

★ 让人像架构一样可扩展

- 与人无关（机器生成）
- 与己无关（互冗互备）
- 与状态无关（无状态可扩展）
- 与数量无关（部署恒定）



# 运维架构设计的关键

- 经济原则（宁花机器1分，不费程序员1秒）
- 扩展原则（设计着眼未来，解耦+负载均衡）
- 生成原则（避免手工hack，程序生成程序）

不断寻找偷懒的方法，聪明工作



# 运维的三大法宝

- 运维自动化 善用脚本使工作自动化，流程化
- 性能可视化 提供连续的健康报表，争取资源
- 监控常态化 及时报警及隔离，触发补救措施

持续集成，频繁发布，犯错趁早，防微杜渐

# 越努力、越幸运

- 方向比努力更重要（南辕北辙）
- 流程比补位更重要（规矩成方圆）
- 方法比拼命更重要（事半功倍）

**Contributor:** @又拍云存储

**Speaker:** 邵海杨 @海洋之心-悟空

一个真正的强者，不是摆平了多少人，而要看他能帮助多少人

因上努力，果上求缘

