QCon

从“救火”走向“防火”

——商业平台业务运维实践

房秀丽

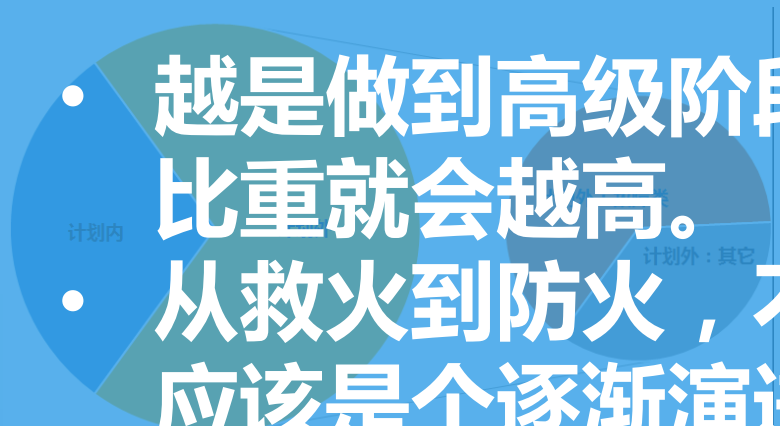
2015-04-24

救火

防火

时间都去哪儿了

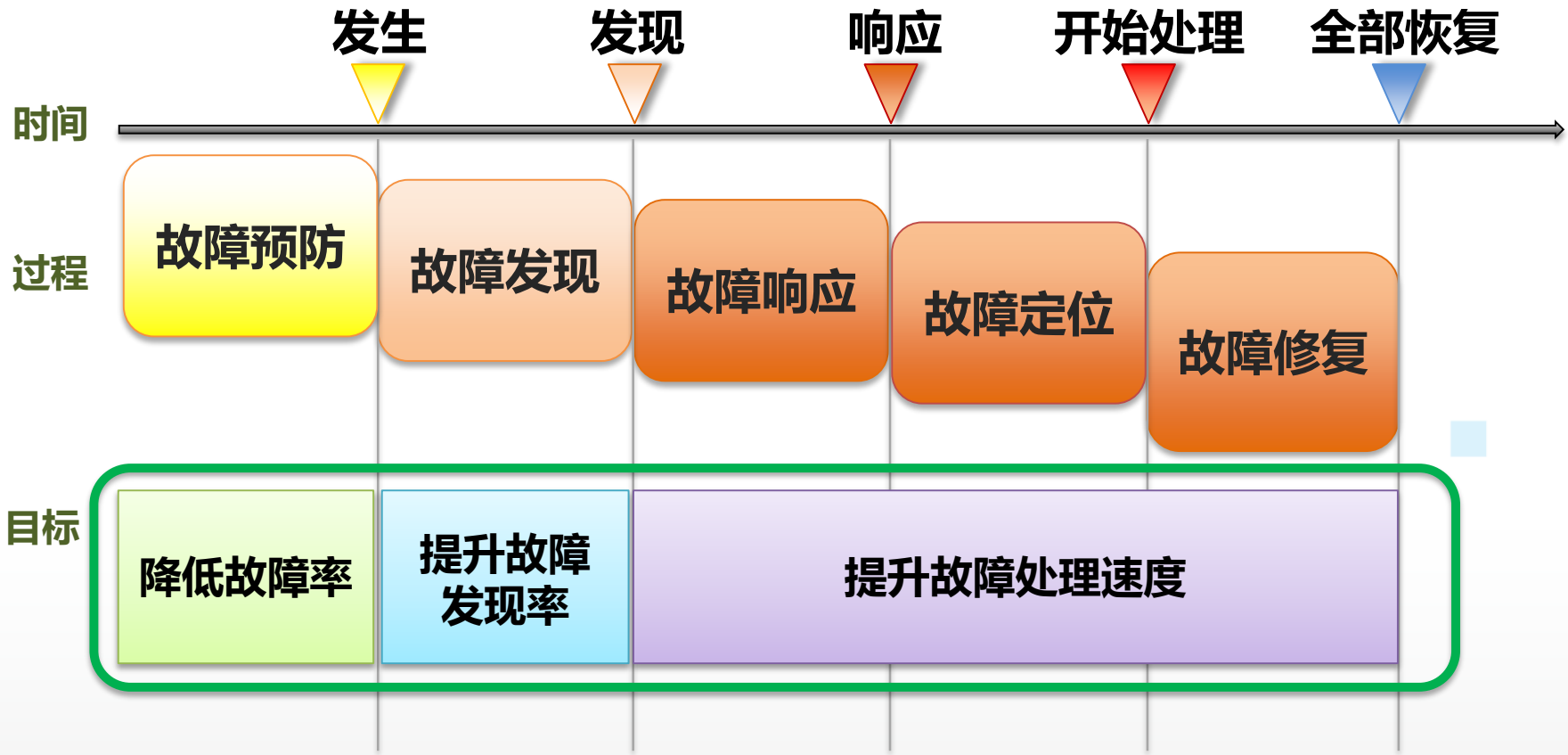
救火



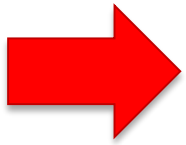
防火

- 越是做到高级阶段，防火的工作所占的比重就会越高。
- 从救火到防火，不是一蹴而就的事情，应该是个逐渐演进的过程

思考点


发生
发现
响应
开始处理
全部恢复
时间
过程
故障预防
故障发现
故障响应
故障定位
故障修复
目标
降低故障率
提升故障发现率
提升故障处理速度

Agenda



1. **如何提升故障处理速度**
2. 如何提升故障发现率
3. 如何降低故障率

目标与措施

- 职责与分工
- 7*24值班规范
- 故障响应规范
- 应急处理预案
- 故障预演

- 故障运维手册
- 分析数据及图表
- 报警内容

- 数据修复工具
- 流量切换工具
- 服务降级工具
- 回滚工具
- 数据提取工具
- 任务修复工具
- ……

这些都做了，还有提升的空间吗？

主要受哪些影响因素

影响因素	应对方法
1、新手	**优化报警内容**，使报警内容变得可依赖。报警内容除了报告问题，还将故障的判断和处理方法附在其中。
2、故障处理步骤繁杂	利用数据任务调度管理系统，对**数据任务进行统一管理**

优化报警内容

B <[redacted]@sogou-inc.com>

d[redacted]全量数据生成时间超长或未完成。

收件人 Zhang[redacted]事业部); Ga[redacted]业部); Li[redacted]事业部); ⊞ biz[redacted]se; L[redacted]技术部); Liu[redacted]平台研

该邮件的重要性为: 高。

dump[redacted]全量数据生成时间超长或未完成，文件：/[redacted]/pope/2015041408.data.done 未按时生成。
服务器：10.13[redacted].58

报警处理方法：
1，登录 10.13[redacted].58 执行命令：ll /[redacted]/pope/2015041408.*.done，确认以下 4 个 done 文件是否全部生成
2015041408.01.done
2015041408.02.done
2015041408.03.done
2015041408.04.done

done 文件序号（01-04）分别对应机器的查询方式：运行命令（cat [redacted] | grep dumper0）
2，确认缺少 done 文件对应的 IP 后，登录对应服务器，按如下方法检查
3、检查进程是否存在，命令(ps -ef|grep start_d[redacted]ri.sh)，（注意启动时间是否为 2015041408 这一小时）
4、检查 done 文件是否生成，命令：（ll /[redacted]/dump/dump.2015041408.done）
5、根据 3,4 步结果判断：
a、如果有进程，说明程序未运行完毕，等待程序运行完毕即可。
b、如果没有进程，但 done 文件已有，说明已制作完毕。
c、如果没有进程，也没有 done 文件，说明程序异常。联系运维负责人处理。

减少对运维人员经验的依赖，使得新人和值班人员都可以快速处理

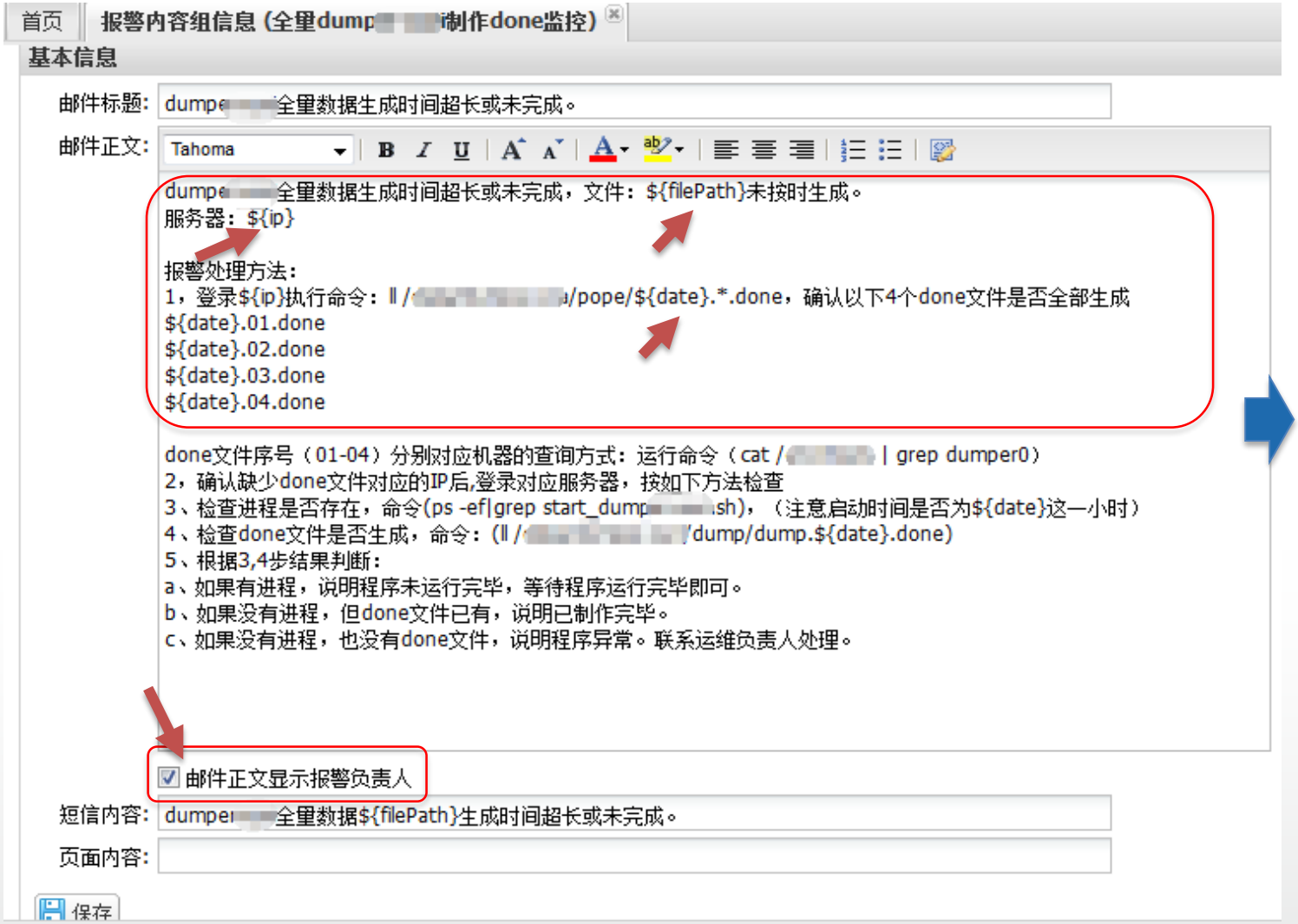
报警负责人：

姓名	邮件	手机号	分机号	负责人顺序
高[redacted]	gac[redacted]@sogou-inc.com	18[redacted]0	[redacted]7	1
李[redacted]	lih[redacted]8@sogou-inc.com	1[redacted]1		2

该报警对应的运维专员联系方式

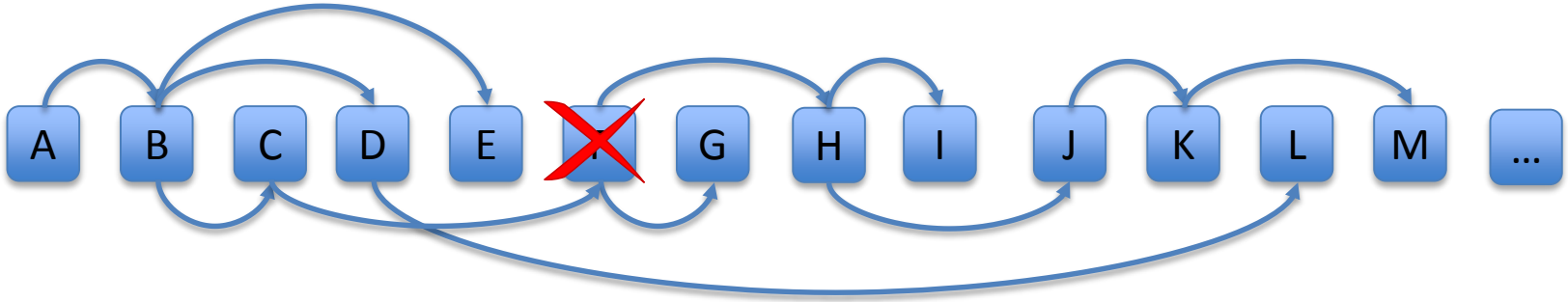
该报警对应的开发人员联系方式

灵活配置报警内容



降低维护成本，
提升工作效率

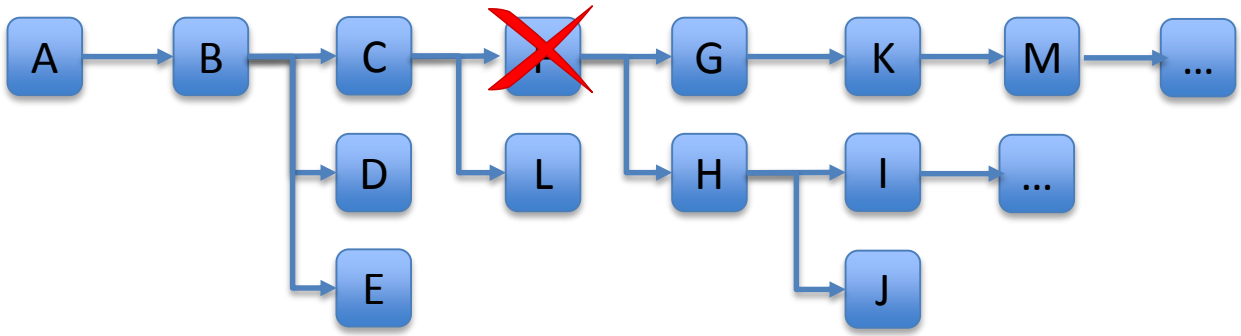
繁杂数据故障处理



存在的问题：

- 不能快速识别哪些任务失败了，影响了谁
- 一个任务失败会导致多个任务失败，每个任务的任务都会发一个报警
- 处理多个任务失败时，需要人工确认修复顺序，还要等待每个任务执行完成后再人工执行下一个
- 如果所需数据源存在短暂延迟到位，会导致任务执行失败并报警，有时会对运维人员产生干扰

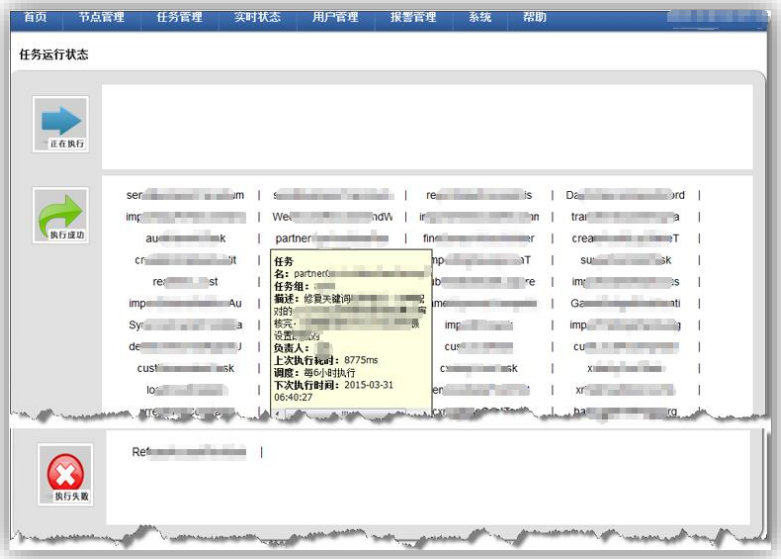
繁杂数据故障处理

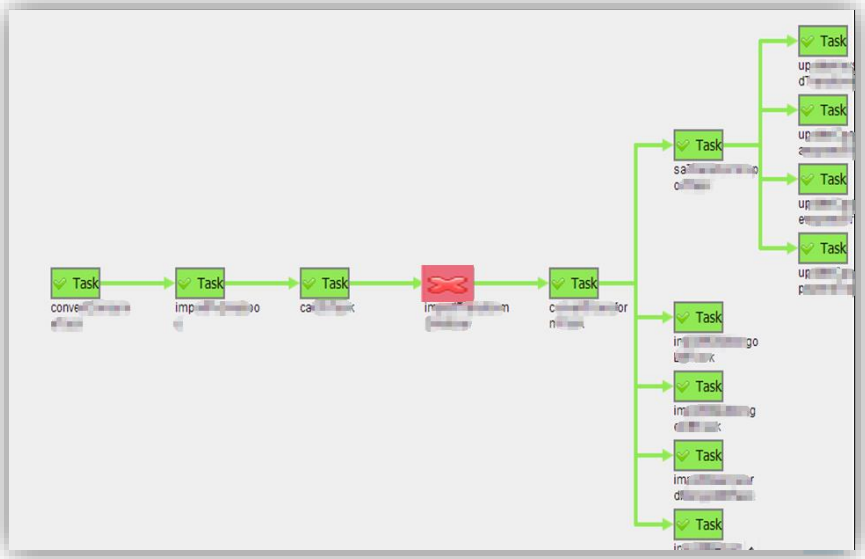


数据任务调度管理系统：

- 实时自动的可视化数据任务关系图
- 能快速识别哪些执行失败及影响范围
- 父节点任务失败后报警，子任务节点不再执行（也不会报警）
- 恢复关键路径节点任务时，只需选择带依赖执行，后续子节点任务会自动执行，无需人工干预，等待
- 支持每个任务自定义重试次数和间隔，如果任务所需数据源存在延迟提供现象，不会马上报警，直到达到最大重试次数为止，降低对运维人员的干扰

繁杂数据故障处理

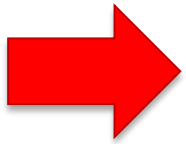

首页
节点管理
任务管理
实时状态
用户管理
报警管理
系统
帮助
任务运行状态
正在执行
执行成功
执行失败


Task


首页
节点管理
任务管理
实时状态
用户管理
报警管理
系统
帮助
任务信息查看（任务名称：
Task 任务组名：
查看任务
参数管理
依赖管理
调度管理
查看日志
立即运行
修改任务
新建参数
新建触发器
删除任务
重新调度
取消调度
重置状态
导出任务
任务名称：
任务组：
任务BeanId：
描述：
负责人：
类型：其他类型
任务当前状态：上次执行成功
重要级别：B2
是否邮件报警：是
是否短信报警：是
是否节假日短信报警：否
预期耗时（秒）：3600
执行间隔：每小时执行
是否404任务：否
重试次数：0
重试间隔（毫秒）：30000
修改时间：2014-8-11 16:09:03
运行节点：
运行类型：Java
可执行代码路径：
任务执行命令：
失败的影响：
失败后的处理方法：
报警用户组：
默认任务参数：
参数键
参数值
executedate
t(#today)
importdate
t(#yesterday)


任务日志查看（任务名称：
Task 任务组名：
查看任务
参数管理
依赖管理
调度管理
查看日志
立即运行
日志类型
请选择
查询
联合查询
每页显示 20 条记录 共4,341个 首页 上一页 下一页 末页 1/218页 跳到 1 GO
任务名称
任务组名
开始时间
结束时间
总耗时(毫秒)
执行情况
执行状态
执行方式
操作人
15-04-10 09:56:37
15-04-10 09:56:46
8339
执行成功
系统依赖调度
SYSTEM
15-04-10 09:56:37
0
正在执行
15-04-10 08:56:38
15-04-10 08:56:46
8406
15-04-10 08:56:38
0
15-04-10 07:56:38
15-04-10 07:56:46
8460
15-04-10 07:56:38
0
15-04-10 06:56:38
15-04-10 07:01:26
288007
15-04-10 06:56:38
0
15-04-10 05:56:38
15-04-10 05:56:46
8277
15-04-10 05:56:38
0
15-04-10 04:56:38
15-04-10 04:56:46
8439

Agenda



1. 如何提升故障处理速度
2. **如何提升故障发现率**
3. 如何降低故障率

完善监控指标

基础资源类

- 网络连通性
- 机器存活
- 远程可达
- 丢包检测
- 容量监控
- 磁盘故障
- 磁盘坏道
- 内存条检测

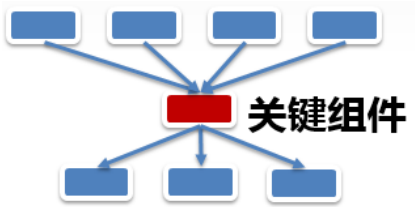
常规业务类

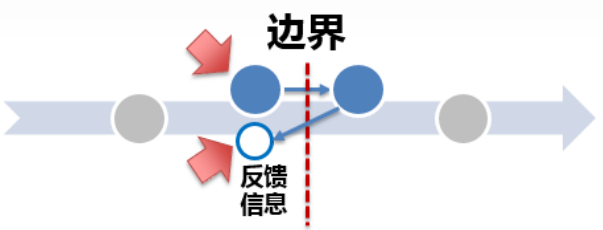
- 端口
- 进程
- Curl
- 工作日志

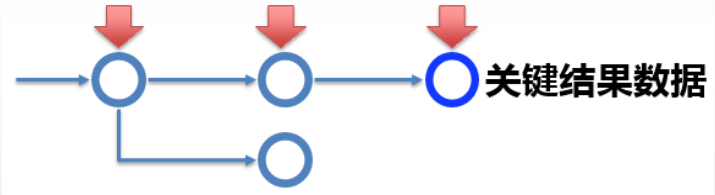
自定义业务类

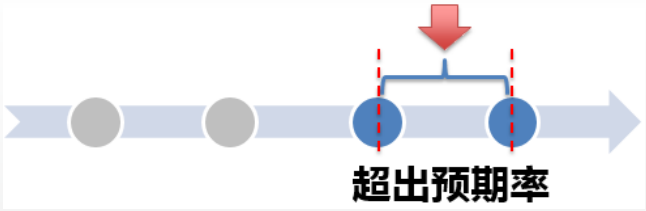
- 任务始止类
- 关键组件监控
- 一致性类
- 跨界类监控
- 数据流监控
- 超时/延时类
- 失败率/成功率
- ……

自定义业务类监控





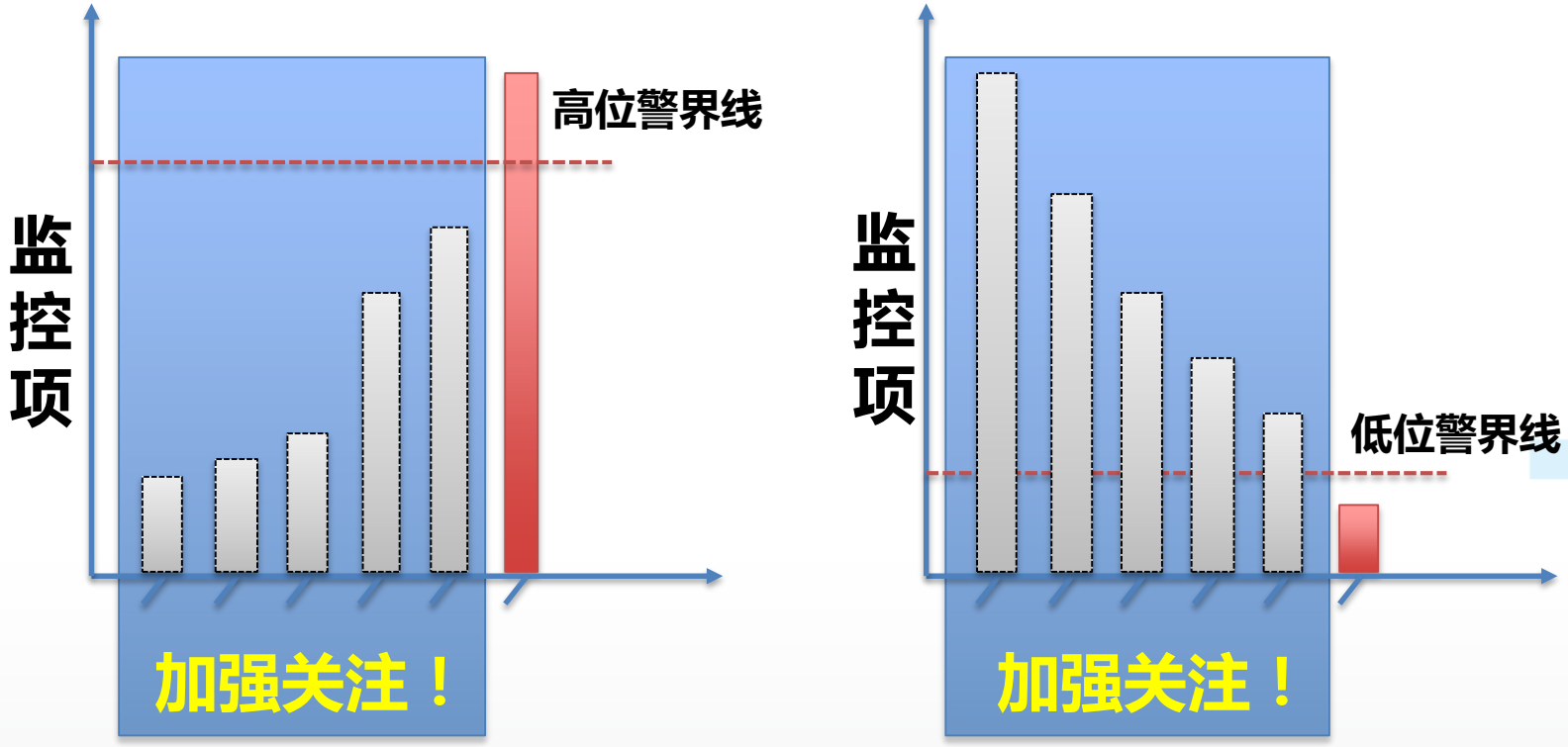




……

……

业务系统健康度趋势



全面

导致短信报警过多

产生麻木心理

降噪

取得的效果	•监控的系统：300+ •监控的实例：20000+ •运维人员短信接收：人均每天6条
采用的手段	•报警策略 •报警分层 •精准下发

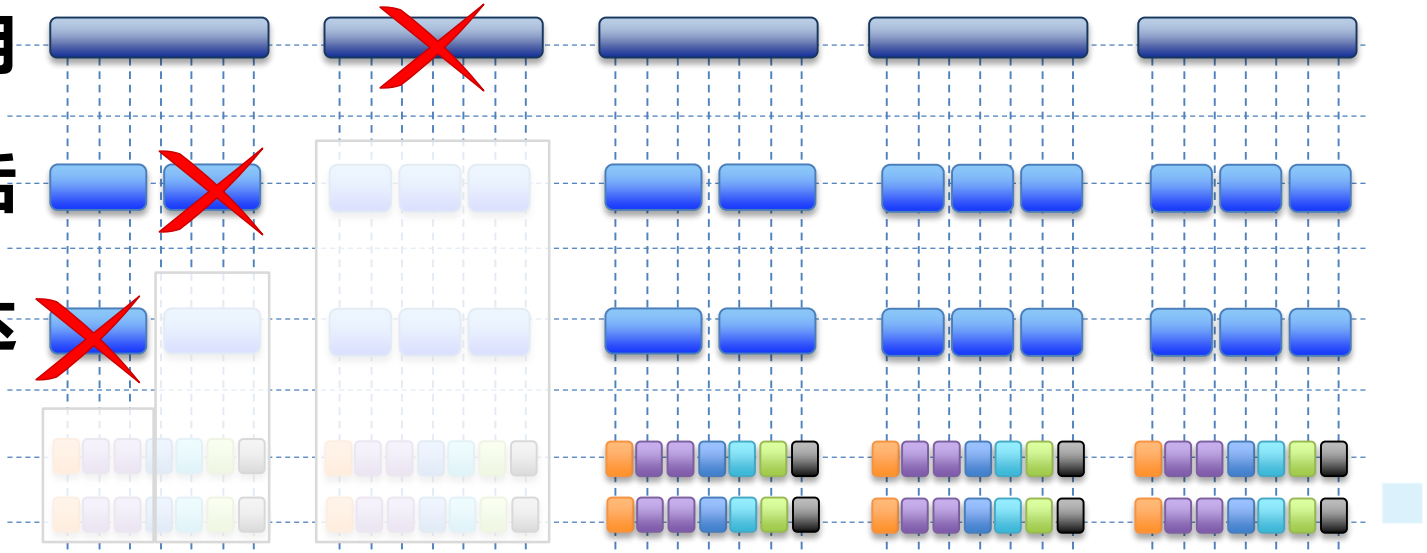
报警分层

网段可用

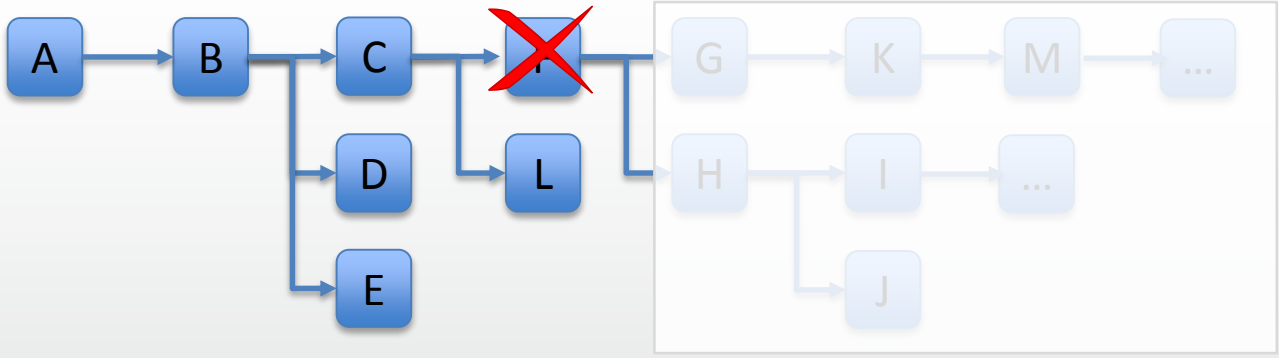
机器存活

远程可达

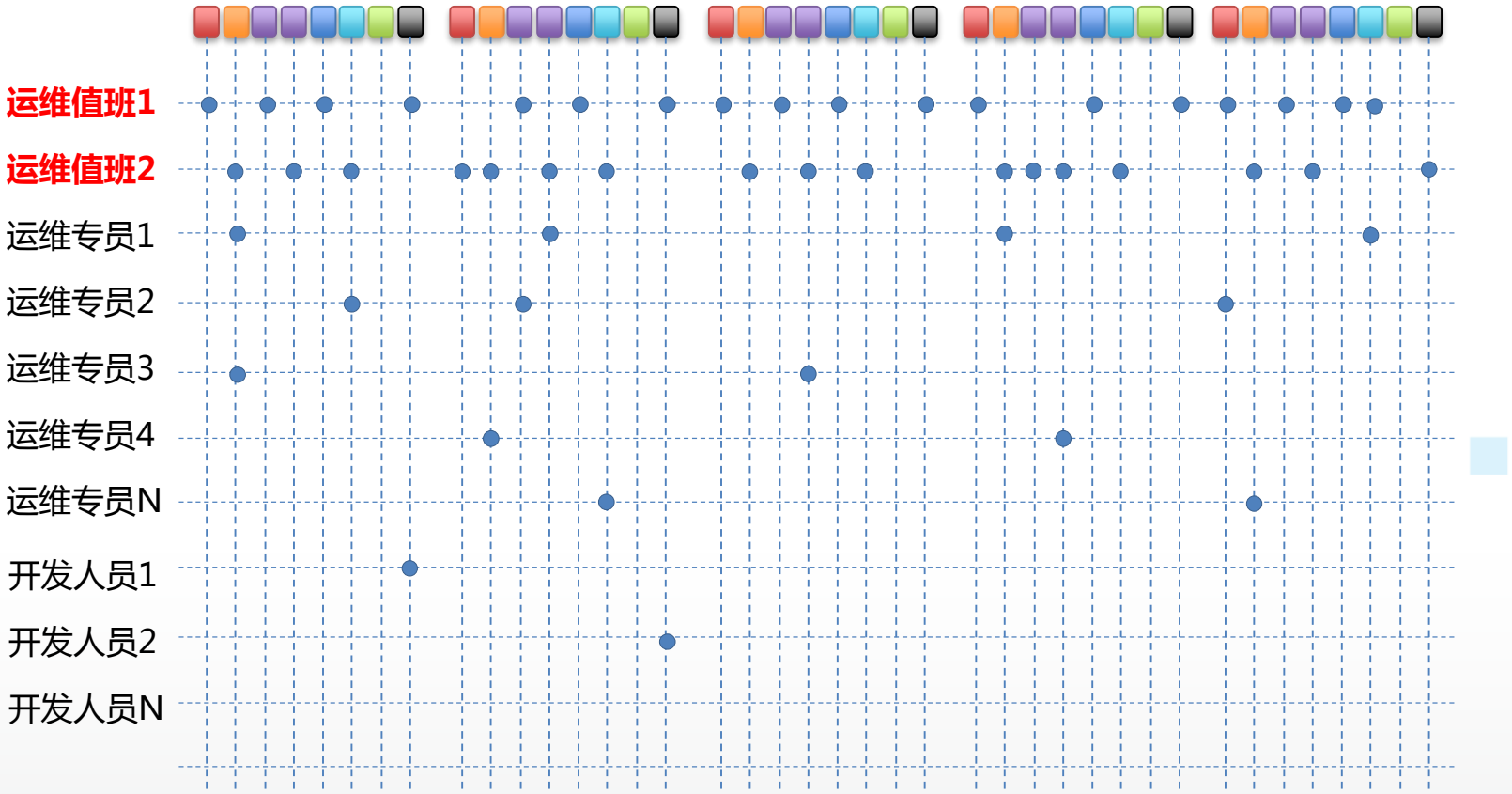
业务类



数据依赖任务



报警精准下发


运维值班1
运维值班2
运维专员1
运维专员2
运维专员3
运维专员4
运维专员N
开发人员1
开发人员2
开发人员N

报警点管理

详细信息 | 查看 |

排序	ID	名称	描述	邮件报警	短信报警	页...	故障等级	自动收集...	首次报警	报警间隔
1	147	curl报警	curl报警	是	是	否	二级	是	2	5

* 邮件报警: (•) 是 ( ) 否

* 短信报警: (•) 是 ( ) 否

* 短信报警方式: 工作日+节假日

短信报警时间段: [ ] - [ ] (报警时间段，如果全天随时报警，请置为空。时间点可手工输入)

* 凌晨短信延时报警: ( ) 是 (•) 否

* 页面报警: ( ) 是 (•) 否

* 报警内容: curl_domain报警

* 报警接收人:

值班: [✓] 应用值班 [ ] 系统值班

运维: [ ] 运维专员 [✓] 运维组长 [✓] 运维组

开发: [ ] 开发专员 [ ] 开发组长 [✓] 开发组

测试: [ ] 测试专员 [ ] 测试组长 [ ] 测试组

系统: [ ] 系统组

其他: [ ]

* 报警负责人:

第一负责人: 运维专员

第二负责人: 运维组长

其他: [ ]

* 报警级别: Major

* 故障自动收集: (•) 是 ( ) 否

* 首次报警: 2 次 (故障累计几次后，进行第一次报警)

* 报警间隔: 5 次 (上一次报警后，故障累计几次再报警)

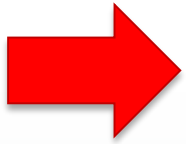
当前位置：常用工具/运维工具/值班信息

值班信息 | 1组排班表 | 2组排班表 | dba组排班表 | 系统组排班表

状态	姓名	开始时间	结束时间	E-mail	手机	分机
已结束	[illegible]	2015-04-08 18:00:00 [周三]	2015-04-09 18:00:00	ga[illegible]@sogou-inc.com	1[illegible]	[illegible]
当前值班	[illegible]	2015-04-09 18:00:00 [周四]	2015-04-10 18:00:00	zhan[illegible]@sogou-inc.com	1[illegible]	[illegible]6
未开始	[illegible]	2015-04-10 18:00:00 [周五]	2015-04-11 18:00:00	wang[illegible]@sogou-inc.com	1[illegible]	5[illegible]6
未开始	[illegible]	2015-04-11 18:00:00 [周六]	2015-04-12 18:00:00	liu[illegible]@sogou-inc.com	1[illegible]	9[illegible]5
未开始	[illegible]	2015-04-12 18:00:00 [周日]	2015-04-13 18:00:00	ga[illegible]@sogou-inc.com	1[illegible]	5[illegible]7
未开始	[illegible]	2015-04-13 18:00:00 [周一]	2015-04-14 18:00:00	zhan[illegible]@sogou-inc.com	1[illegible]	5[illegible]6
未开始	[illegible]	2015-04-14 18:00:00 [周二]	2015-04-15 18:00:00	wang[illegible]@sogou-inc.com	13[illegible]	5[illegible]6
未开始	[illegible]	2015-04-15 18:00:00 [周三]	2015-04-16 18:00:00	liuj[illegible]@sogou-inc.com	1[illegible]	9[illegible]5
未开始	[illegible]	2015-04-16 18:00:00 [周四]	2015-04-17 18:00:00	g[illegible]@sogou-inc.com	187[illegible]	5[illegible]7
未开始	[illegible]	2015-04-17 18:00:00 [周五]	2015-04-18 18:00:00	zhang[illegible]@sogou-inc.com	17[illegible]	5[illegible]6
未开始	[illegible]	2015-04-18 18:00:00 [周六]	2015-04-19 18:00:00	wang[illegible]@sogou-inc.com	137[illegible]18	5[illegible]6
未开始	[illegible]	2015-04-19 18:00:00 [周日]	2015-04-20 18:00:00	liu[illegible]@sogou-inc.com	13[illegible]	9[illegible]5
未开始	[illegible]	2015-04-20 18:00:00 [周一]	2015-04-21 18:00:00	gao[illegible]@sogou-inc.com	1[illegible]	5[illegible]7
未开始	[illegible]	2015-04-21 18:00:00 [周二]	2015-04-22 18:00:00	zha[illegible]@sogou-inc.com	1[illegible]	5[illegible]6
未开始	[illegible]	2015-04-22 18:00:00 [周三]	2015-04-23 18:00:00	[illegible]@sogou-inc.com	13[illegible]	5[illegible]6

Agenda

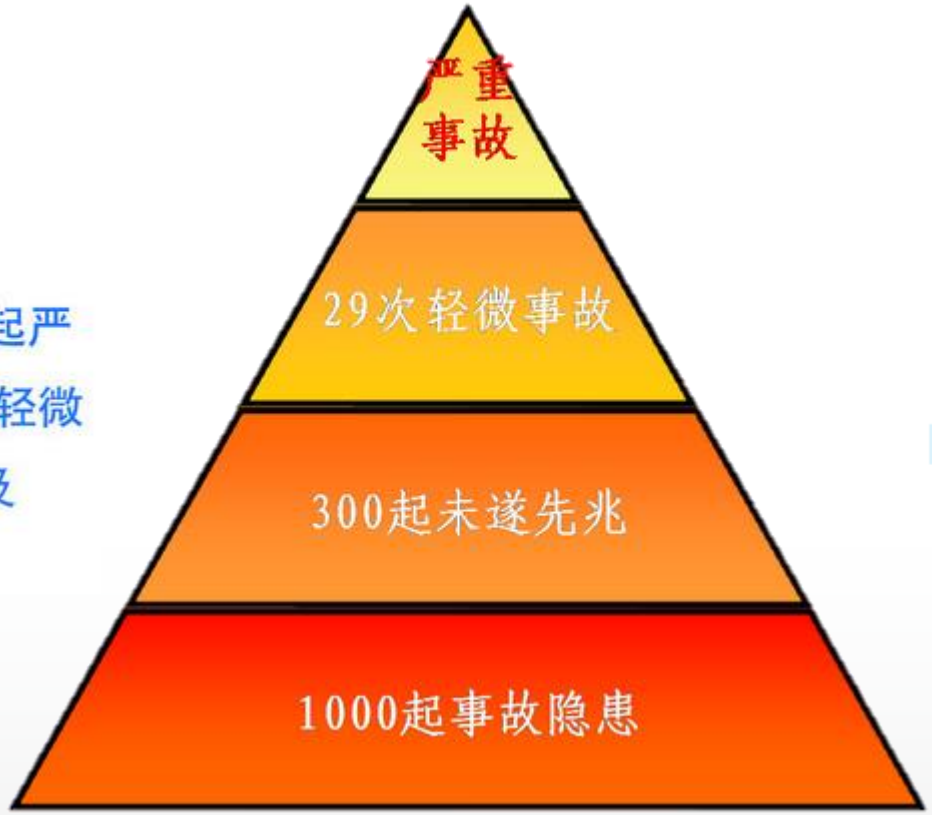
1. 如何提升故障处理速度
2. 如何提升故障发现率
3. **如何降低故障率**



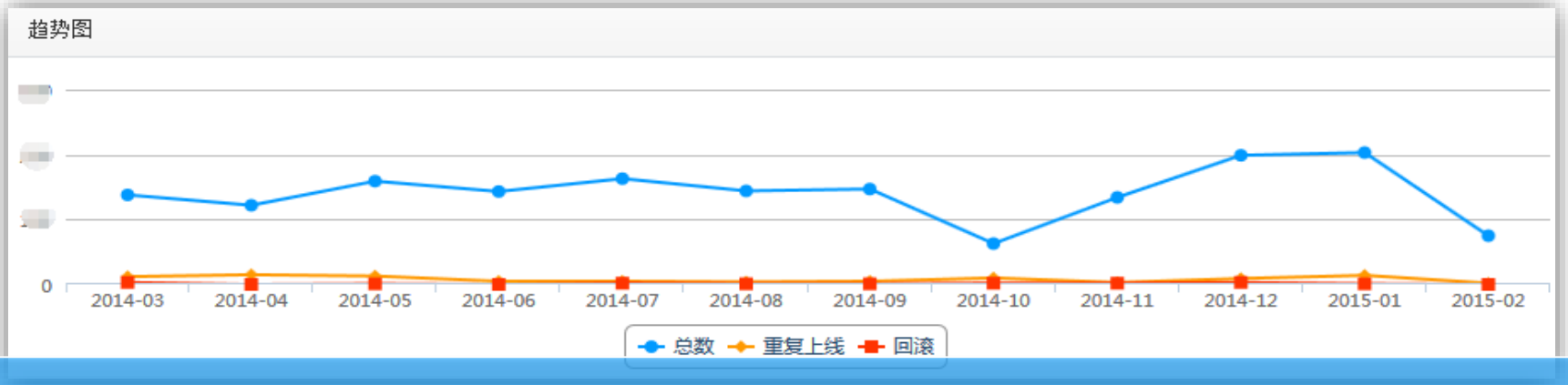
它山之石，可以攻玉

海恩法则

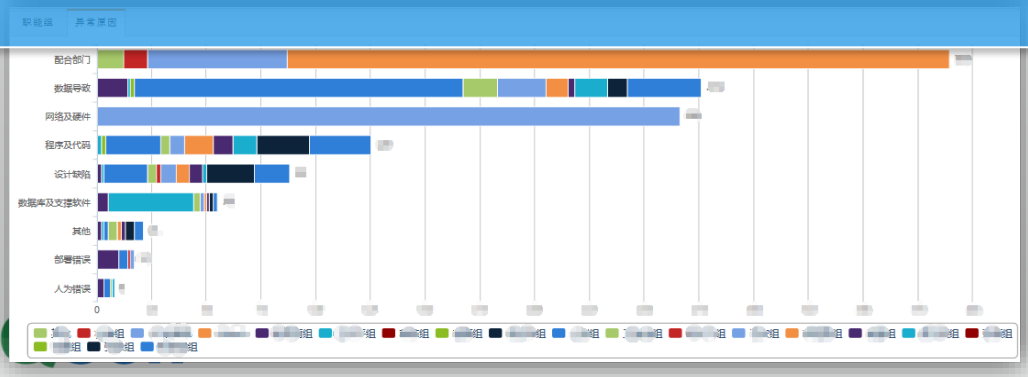
“海恩法则”认为，每一起严重事故的背后，必然有 29 次轻微事故和 300 次未遂先兆，以及 1000 个事故隐患。

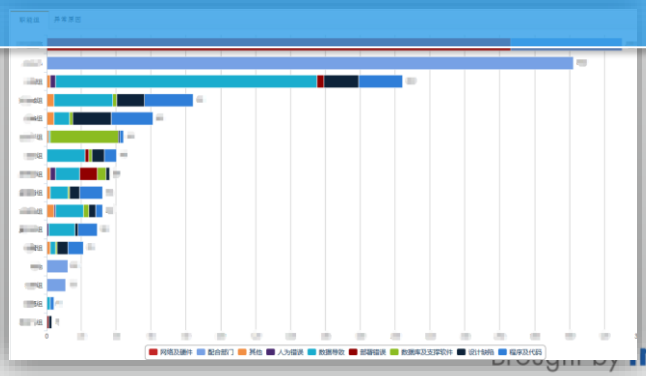


数据说话



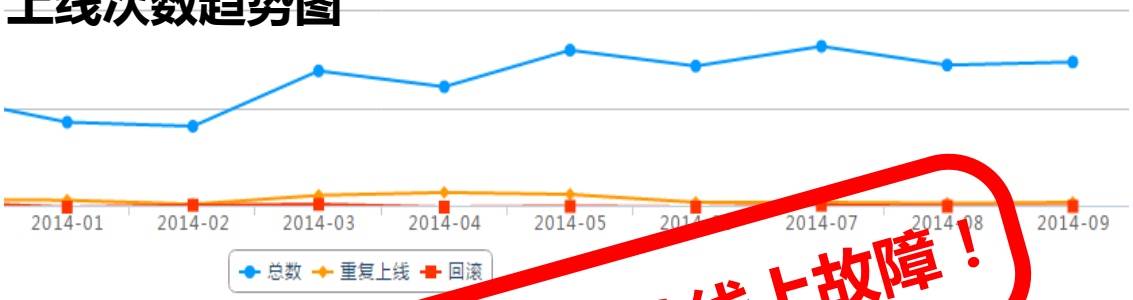
有针对性的推动改进！
避免重复性故障！



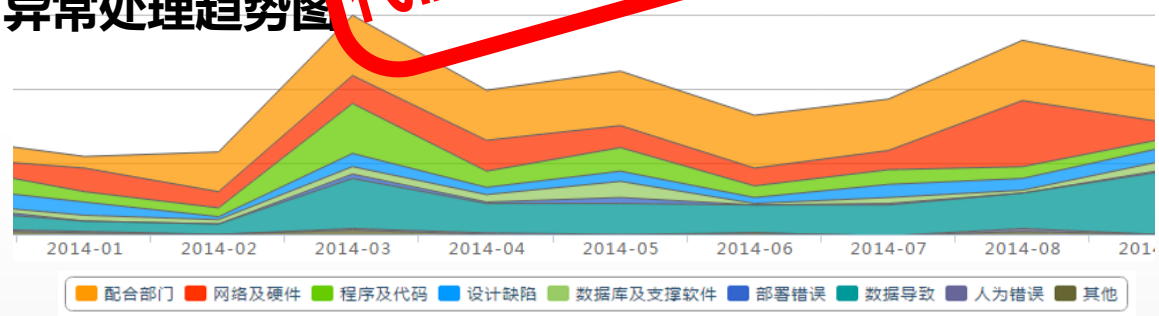


代码发布是故障的导火索

上线次数趋势图



异常处理趋势图



代码发布导致线上故障！

推动开发人员有针对性解决

- 程序Bug
- 设计缺陷

- 代码配置
- 部署问题

变更操作是故障的导火索

常见运维变更

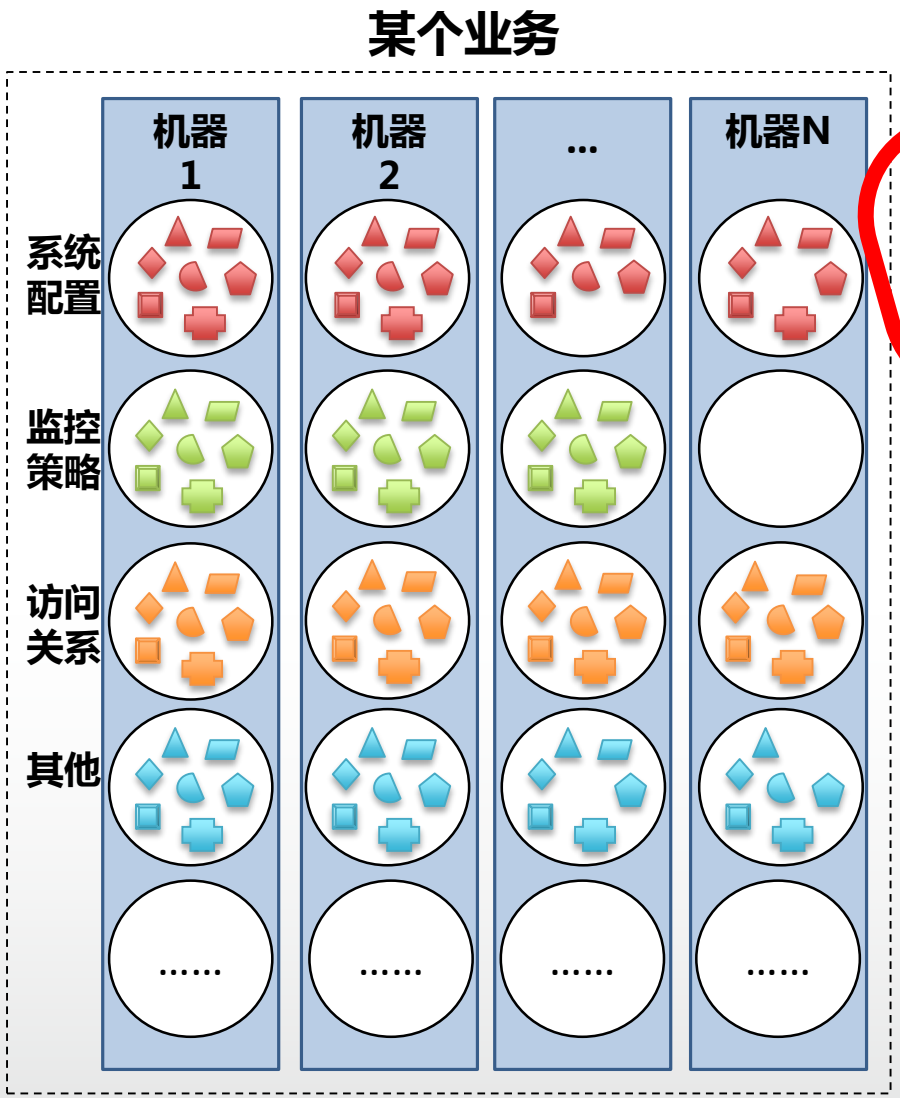
- 业务模块新增机器
- 机房迁移，变更IP
- 机器故障更换
- 下线业务模块
- ……

涉及哪方面的操作

- 维护系统配置
- 部署相关监控
- 维护应用环境及配置
- 维护访问控制关系
- 更新代码配置（研发人员）
- ……

涉及大量的IP、访问关系等信息，操作繁杂！

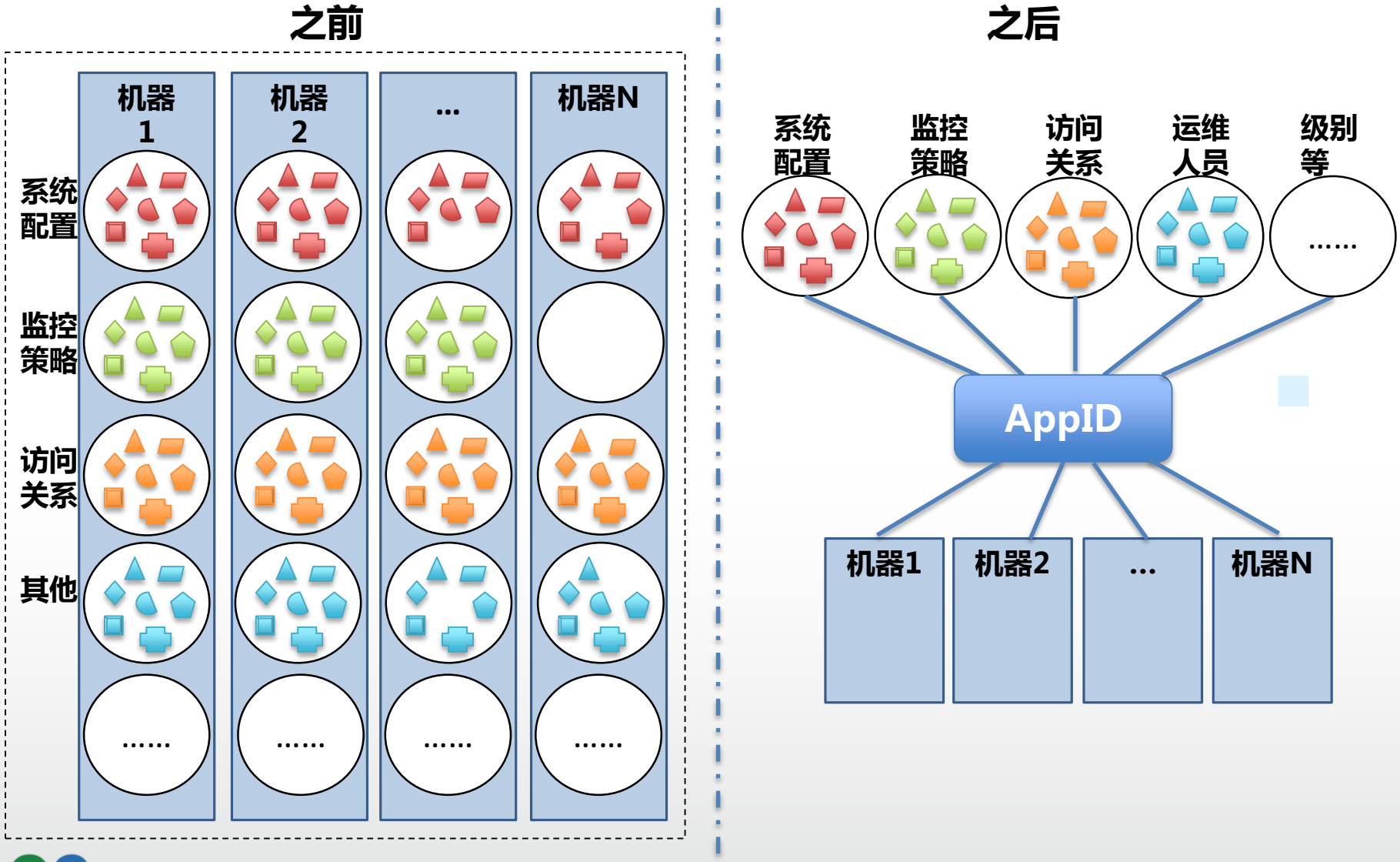
操作繁杂，易出错



降低操作复杂度，避免人为操作问题！

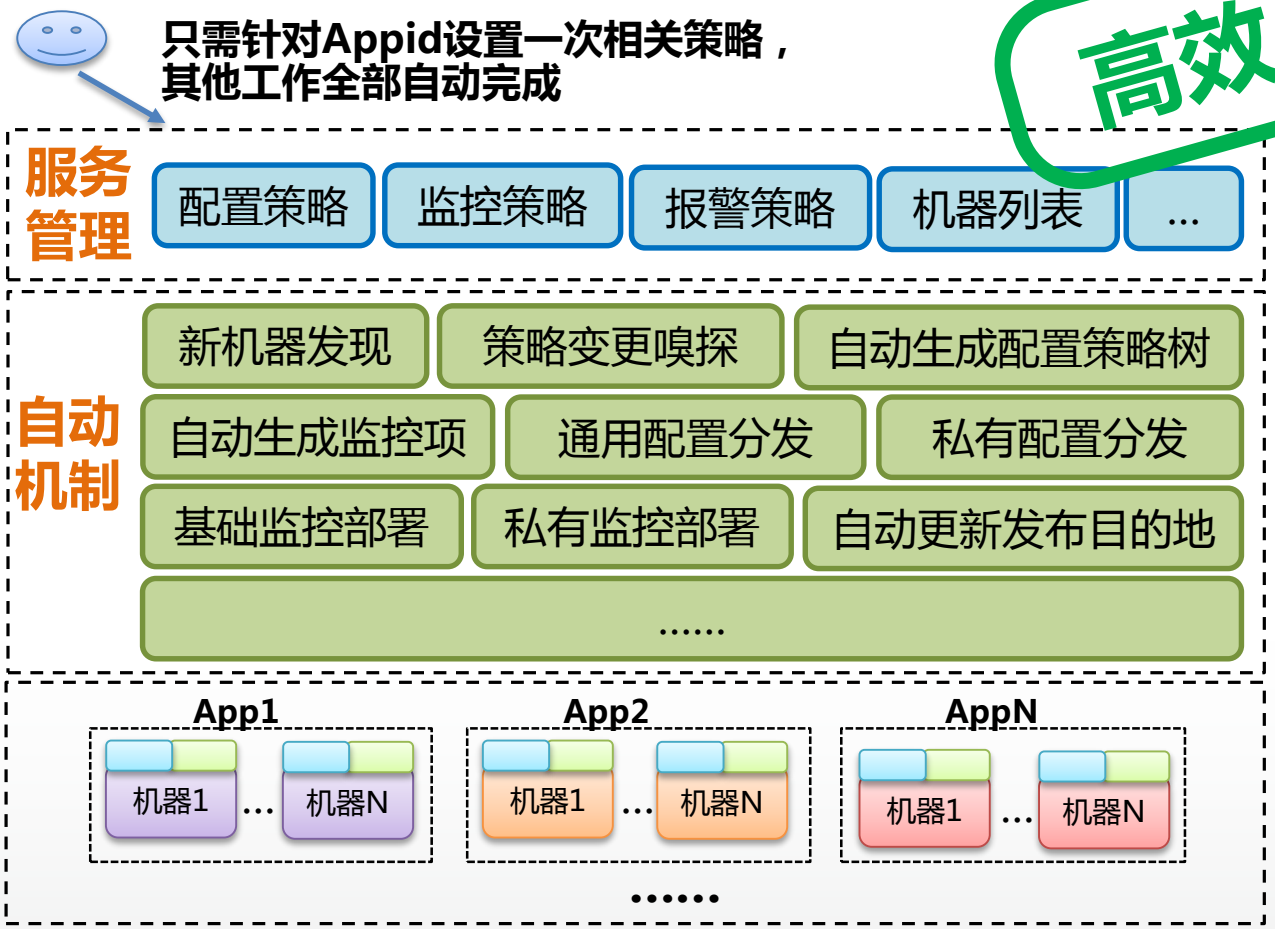
- 抽象运维对象
- 减少人工干预
- 优化技术架构

抽象运维对象


之前
机器1
机器2
...
机器N
系统配置
监控策略
访问关系
其他
.....
之后
系统配置
监控策略
访问关系
运维人员
级别等
.....
AppID
机器1
机器2
...
机器N

减少人工干预

高效，无遗漏！

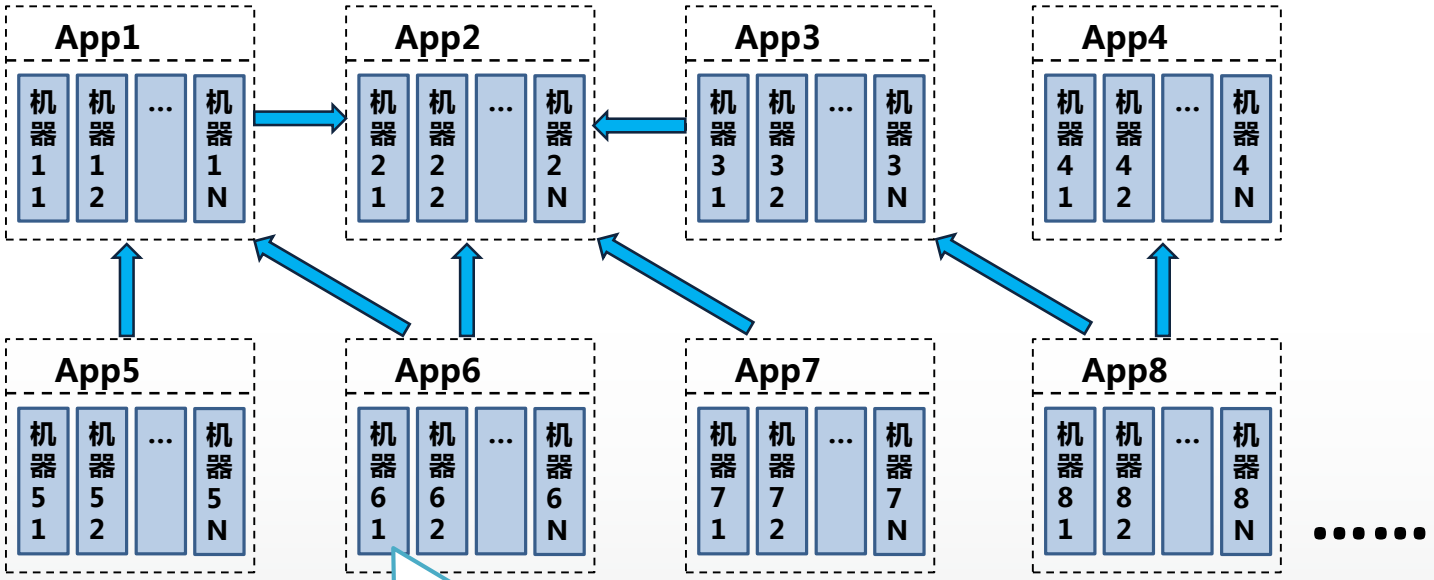


机器规模越大效果越明显

1. 当某App中有新机器到位时，会对该机器自动部署相关的系统配置、监控项，同时自动更新对应App的代码发布地址列表
2. 当某个App有系统配置更新、监控策略更新时，会下发到所有相关的机器

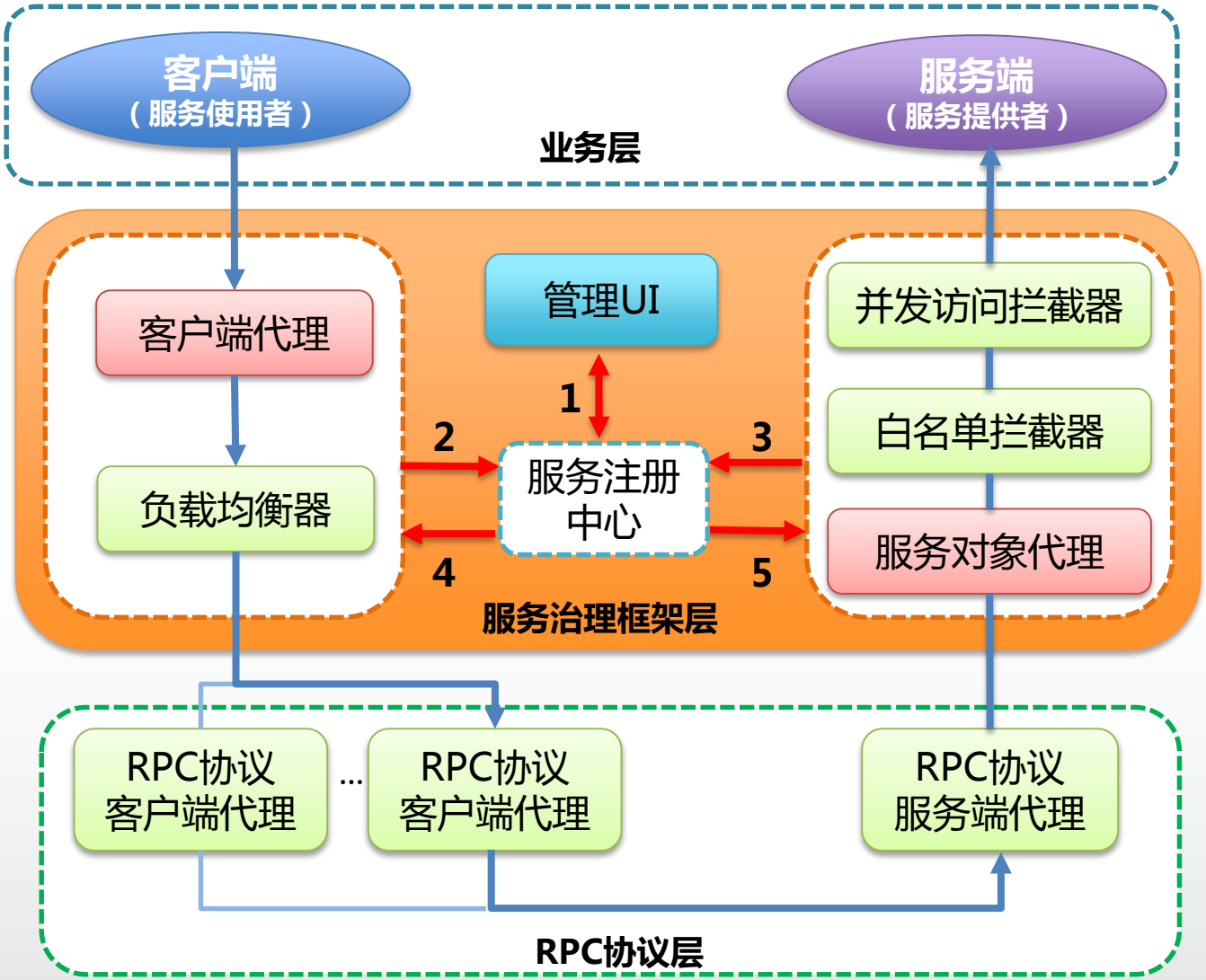
应用间复杂且不透明的访问关系

服务端要确认被哪些客户端访问，才能保证正确的开通访问白名单：
机器+接口+方法
机器11、机器12……机器1N
机器31、机器32……机器3N
机器61、机器62……机器6N



客户端需要访问服务端：
机器11、机器12……机器1N
机器21、机器22……机器2N

规范化后的开发架构



1. 人工在管理界面中进行服务注册与维护，管理服务端及接口，以及客户端对哪些接口访问等
2. 客户端启动时获取服务端机器及URL列表
3. 服务端启动时获取访问控制列表
4. 服务端及接口发生变更时自动推送给客户端，更新调用服务器的机器及URL列表
5. 客户端及接口发生变化时自动推送给服务端更新访问控制列表

小范围试点，效果显著，推广使用

访问控制、自动路由、负载均衡

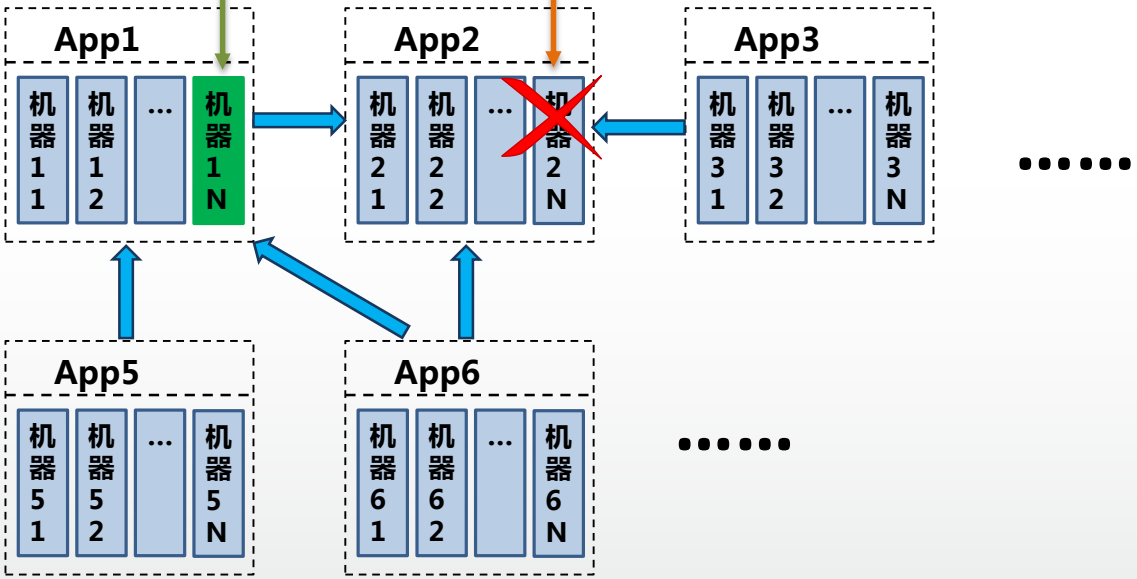
如果App1有新机器到位：

- 框架会自动通知App5、App6的所有机器，App1已有新机器为他们提供服务，各机器自动将请求均衡落到App1的各台机器上
- 同时，自动通知App2的所有机器，有新机器需要纳入到他们的访问白名单

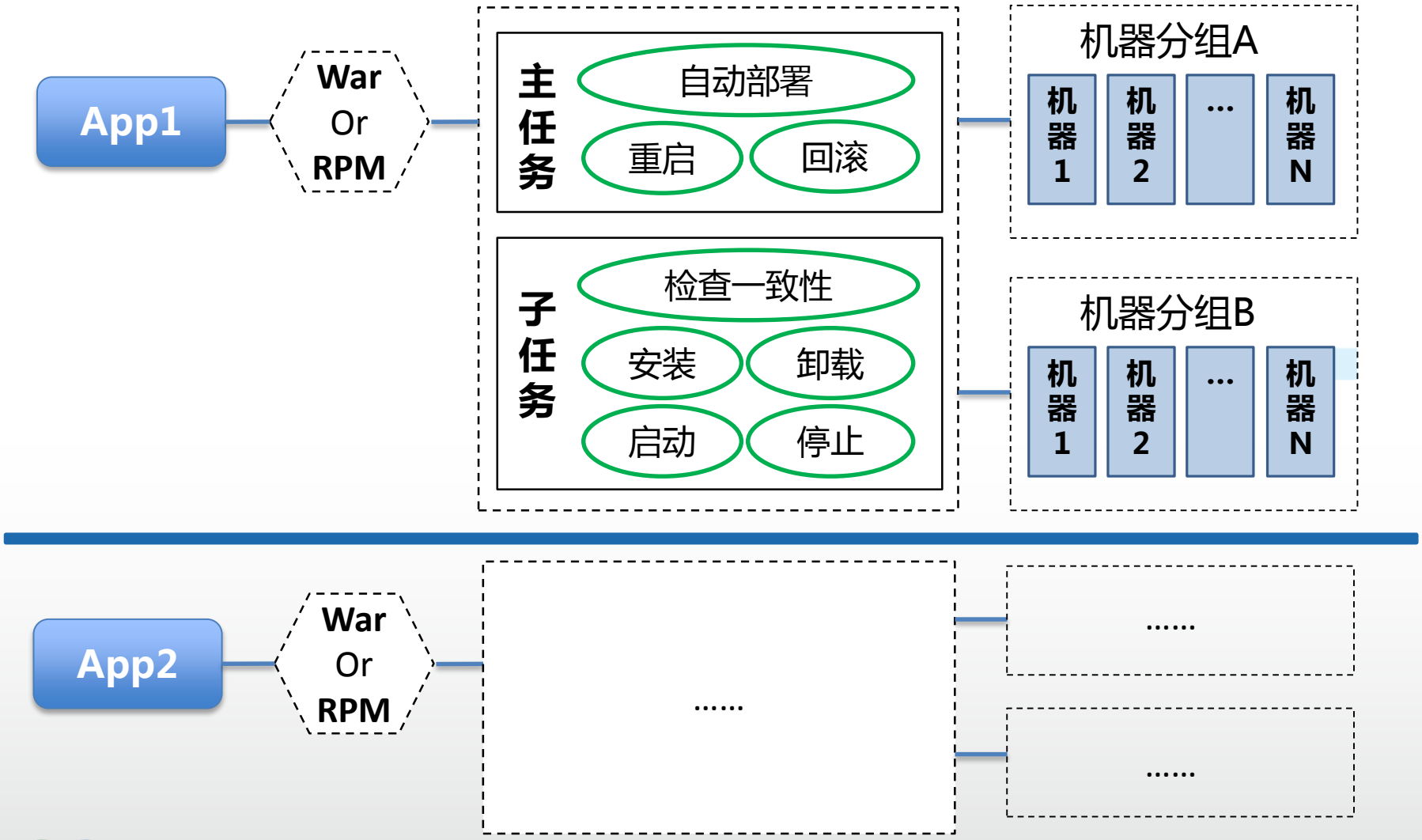
自动容错、负载均衡

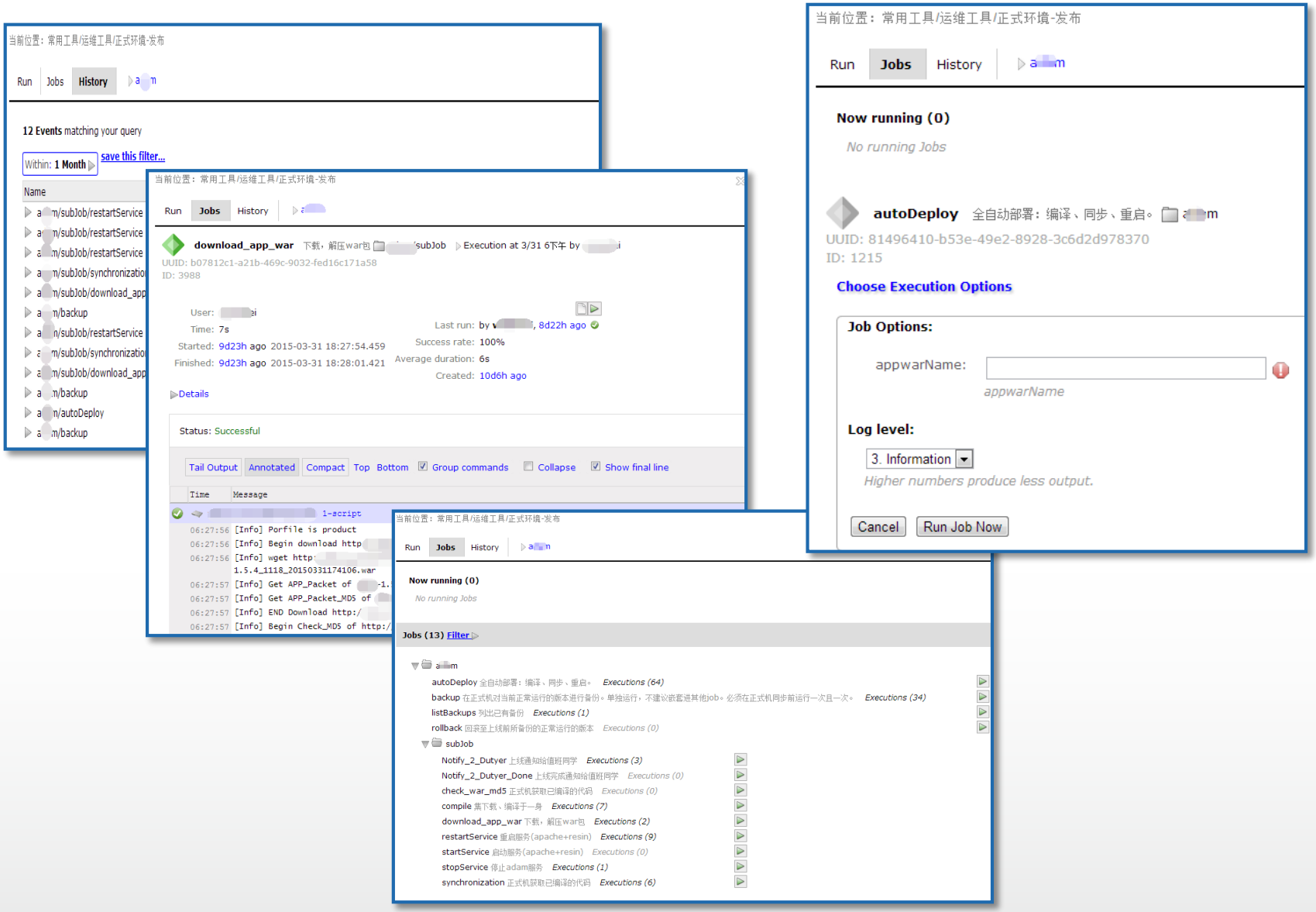
如果App2有机器宕机：

- 框架会自动通知访问它的App1、App3、App6的所有机器，App2有机器故障已不能提供服务
- 各机器的访问请求不再向故障机器发送，并自动将请求均衡落到App2存活的机器上

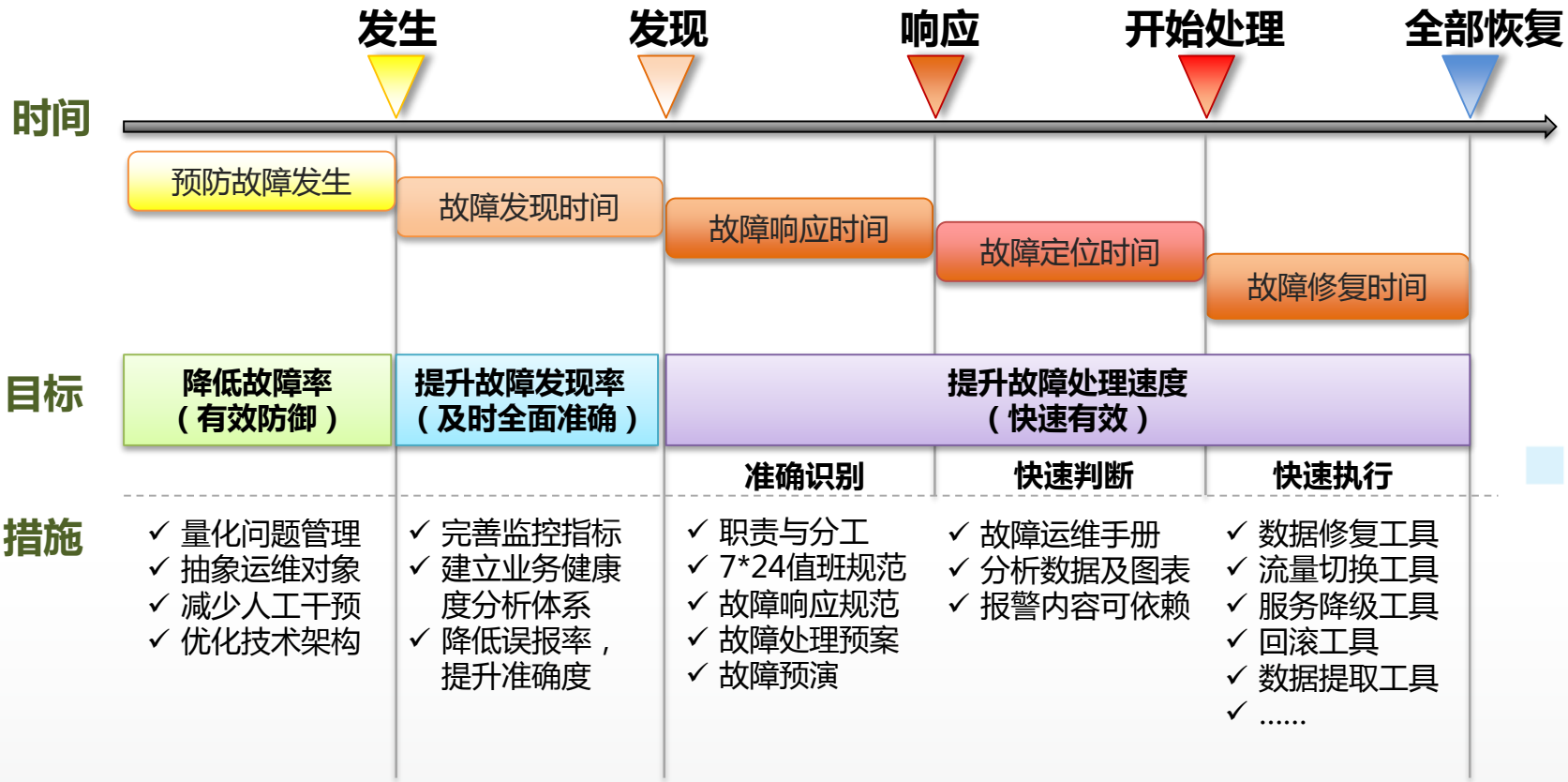


标准化发布方式


App1
War
Or
RPM
主任务
自动部署
重启
回滚
子任务
检查一致性
安装
卸载
启动
停止
机器分组A
机器1
机器2
...
机器N
机器分组B
机器1
机器2
...
机器N
App2
War
Or
RPM
.....
.....
.....


当前位置：常用工具/运维工具/正式环境-发布
Run
Jobs
History
12 Events matching your query
Within: 1 Month
save this filter...
Name
download_app_war 下载，解压war包
Execution at 3/31 6下午 by
UUID: b07812c1-a21b-469c-9032-fed16c171a58
ID: 3988
User:
Time: 7s
Started: 9d23h ago 2015-03-31 18:27:54.459
Finished: 9d23h ago 2015-03-31 18:28:01.421
Last run: by , 8d22h ago
Success rate: 100%
Average duration: 6s
Created: 10d6h ago
Details
Status: Successful
Tail Output
Annotated
Compact
Top
Bottom
Group commands
Collapse
Show final line
Time
Message
Now running (0)
No running Jobs
Jobs (13) Filter
autoDeploy 全自动部署：编译、同步、重启。 Executions (64)
backup 在正式机对当前正常运行的版本进行备份。单独运行，不建议嵌套进其他job。必须在正式机同步前运行一次且一次。 Executions (34)
listBackups 列出已有备份 Executions (1)
rollback 回滚至上线前所备份的正常运行的版本 Executions (0)
subJob
Notify_2_Dutyer 上线通知给值班同学 Executions (3)
Notify_2_Dutyer_Done 上线完成通知给值班同学 Executions (0)
check_war_md5 正式机获取已编译的代码 Executions (0)
compile 集下载、编译于一身 Executions (7)
download_app_war 下载，解压war包 Executions (2)
restartService 重启服务(apache+resin) Executions (9)
startService 启动服务(apache+resin) Executions (0)
stopService 停止adam服务 Executions (1)
synchronization 正式机获取已编译的代码 Executions (6)
autoDeploy 全自动部署：编译、同步、重启。
UUID: 81496410-b53e-49e2-8928-3c6d2d978370
ID: 1215
Choose Execution Options
Job Options:
appwarName:
appwarName
Log level:
3. Information
Higher numbers produce less output.
Cancel
Run Job Now

从救火走向防火


发生
发现
响应
开始处理
全部恢复
时间
预防故障发生
故障发现时间
故障响应时间
故障定位时间
故障修复时间
目标
降低故障率
（有效防御）
提升故障发现率
（及时全面准确）
提升故障处理速度
（快速有效）
准确识别
快速判断
快速执行
措施
✓ 量化问题管理
✓ 抽象运维对象
✓ 减少人工干预
✓ 优化技术架构
✓ 完善监控指标
✓ 建立业务健康度分析体系
✓ 降低误报率，提升准确度
✓ 职责与分工
✓ 7*24值班规范
✓ 故障响应规范
✓ 故障处理预案
✓ 故障预演
✓ 故障运维手册
✓ 分析数据及图表
✓ 报警内容可依赖
✓ 数据修复工具
✓ 流量切换工具
✓ 服务降级工具
✓ 回滚工具
✓ 数据提取工具
✓ .....

后续努力方向

- 更智能
  - 智能监控
  - 智能故障修复
  - 智能健康度评估
- 更轻量
  - 简单易控
  - 随时随地

欢迎关注

搜狗微信号: **sogou-inc**

我的微信号: **fxl_wx**

www.infoq.com/cn

InfoQueue

@InfoQ

infoqchina

软件
正在改变世界！