美团 meituan.com
新美大技术团队

报警分析系统解密

美团大众点评

汇报人：龙雪刚

目录
Contents

Part 1

值班现状

24h待命，及时响应，快速定位，重复劳动

现状 & 痛点

24h 待命

及时 响应

快速 定位

重复 劳动

24h待命

全天任何时间段保障线上服务，即使下雪的凌辰

及时响应

报警触发时，快速响应，1分钟内开始处理

快速定位

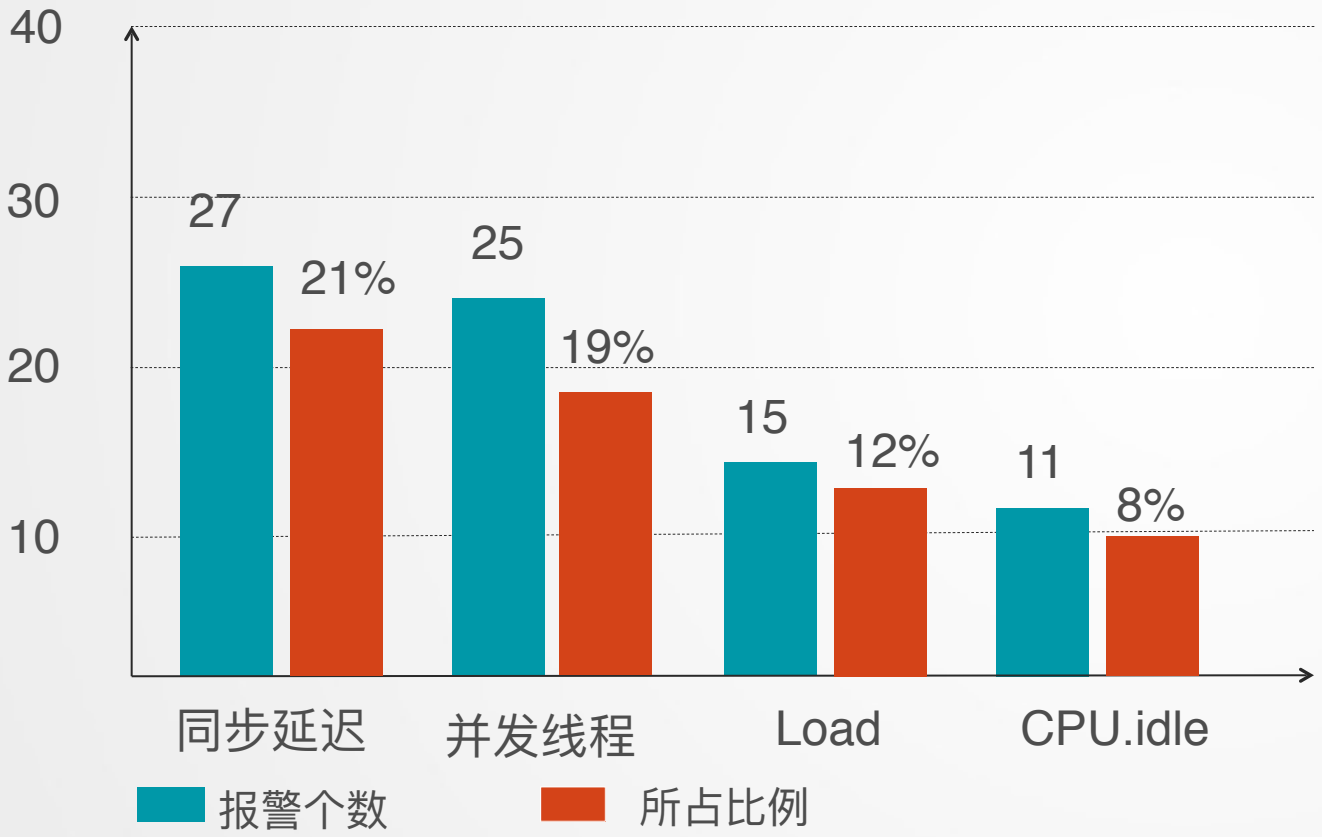
快速准确定位原因，缩短故障影响时长

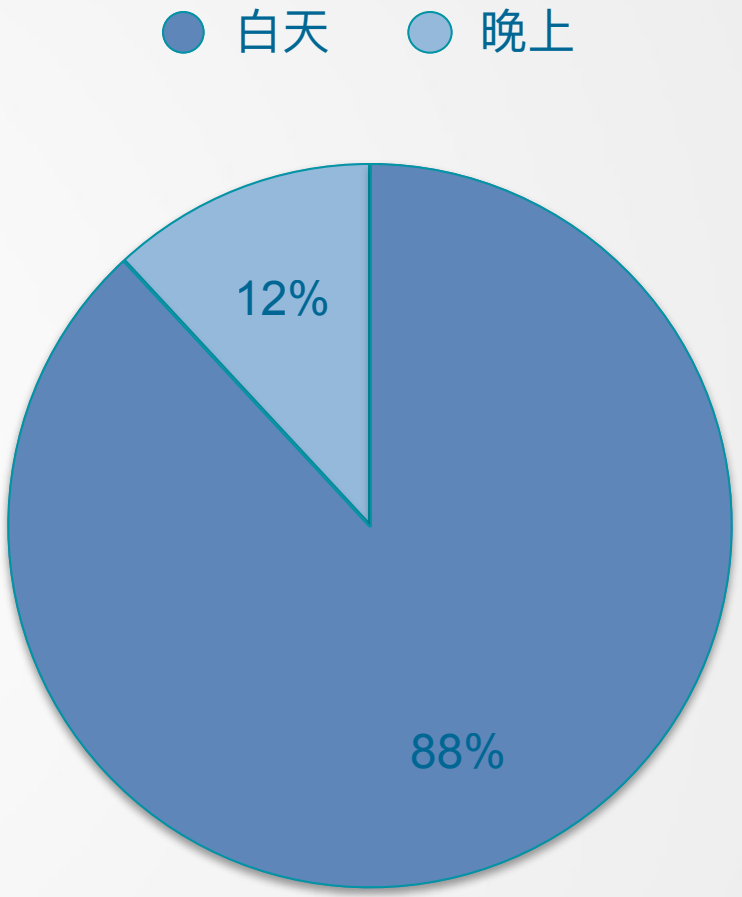
重复劳动

报警类型有限，原因总是那么几种，时间长了，完全是重复劳动

现状 & 痛点

主要报警类型次数/时间段





基本思路

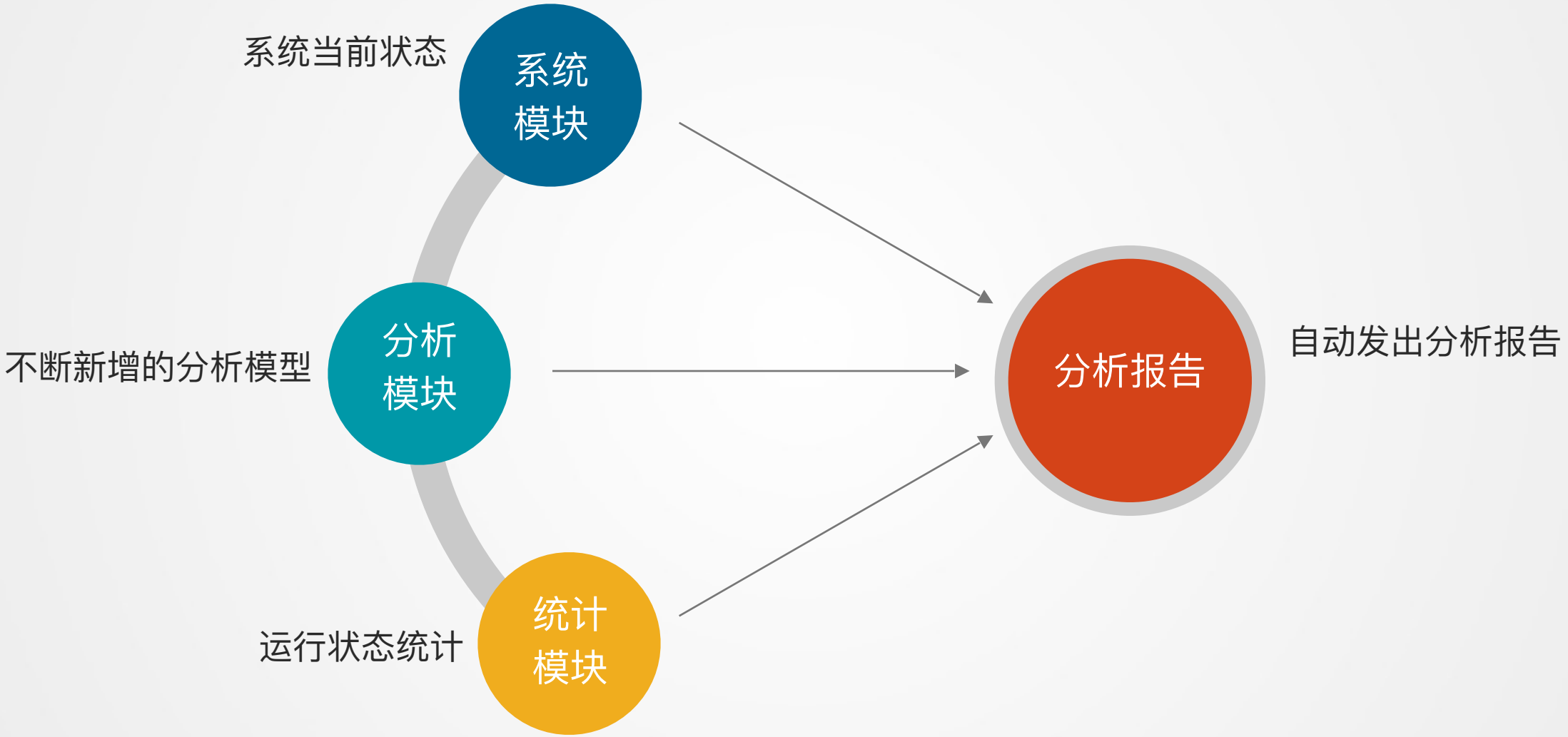
- 让报警尽快恢复。
- 找到原因，并解决它。
- 只做DBA登陆机器的前5分钟干的事。

Part 2

系统架构

系统模块，分析模块，统计模块

组织架构



系统模块

元信息

- 集群信息
- 硬件信息

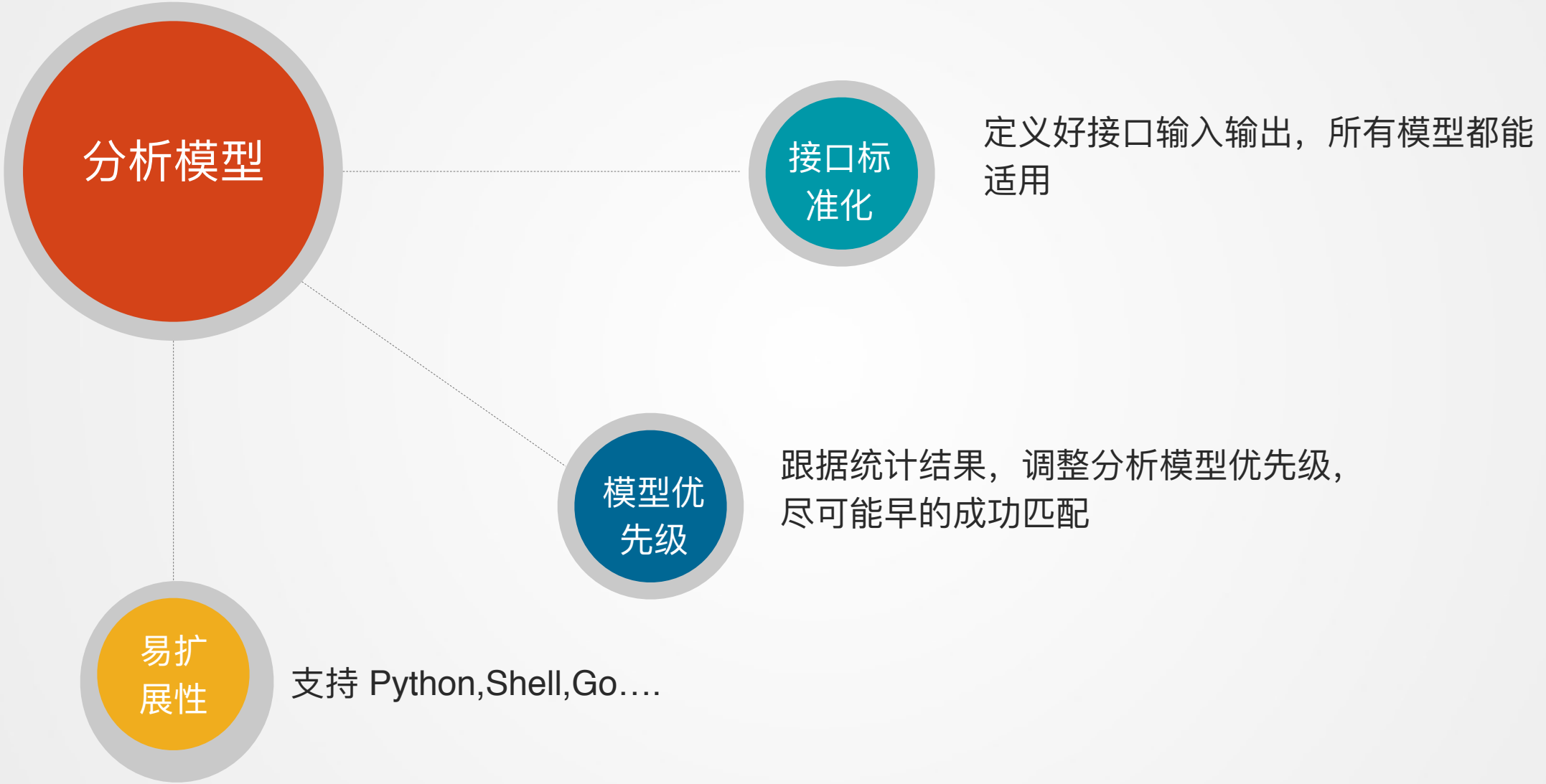
系统日志

- Kern日志
- Megacli日志

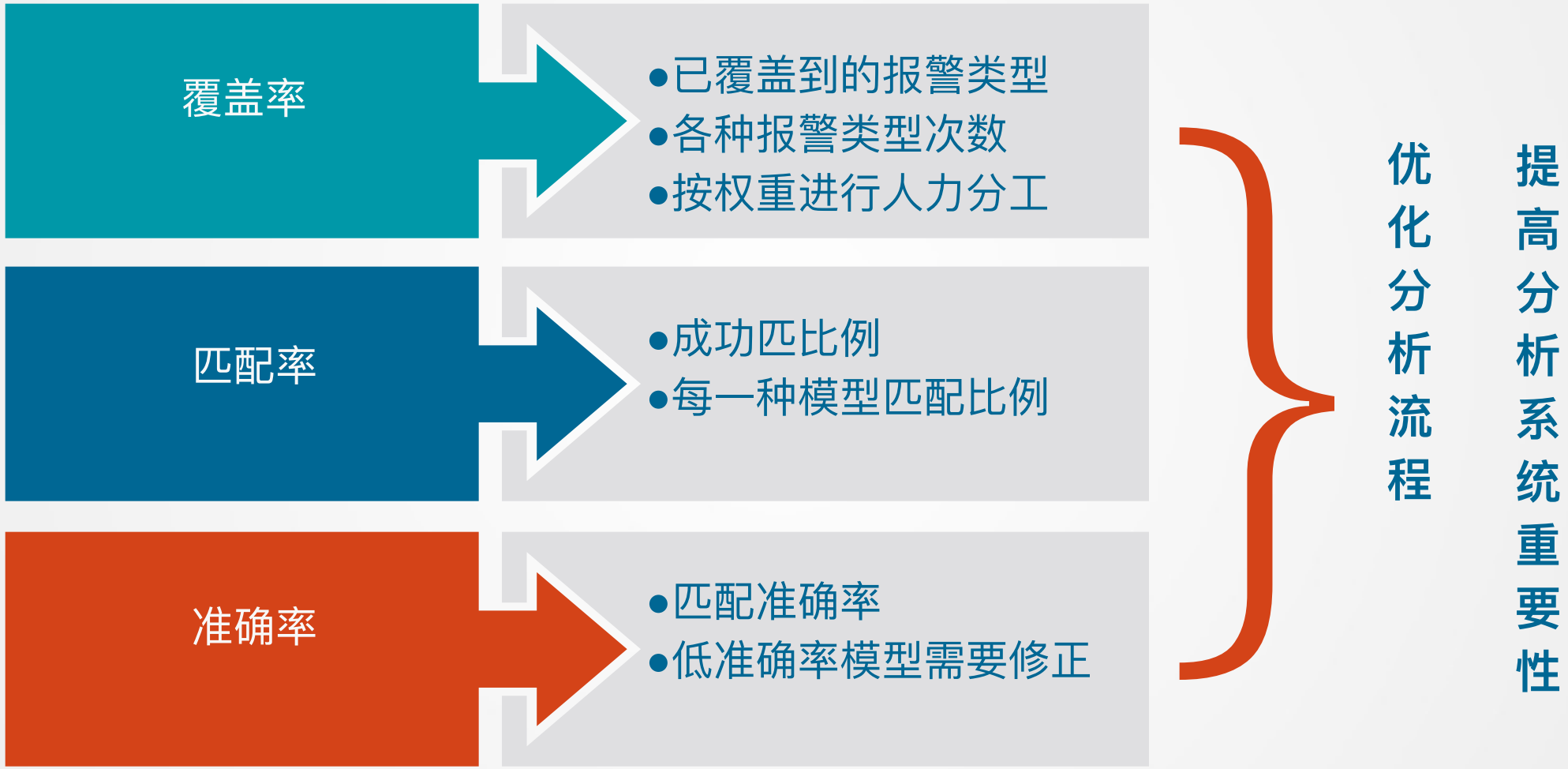
负载

- 机器负载
- MySQL负载

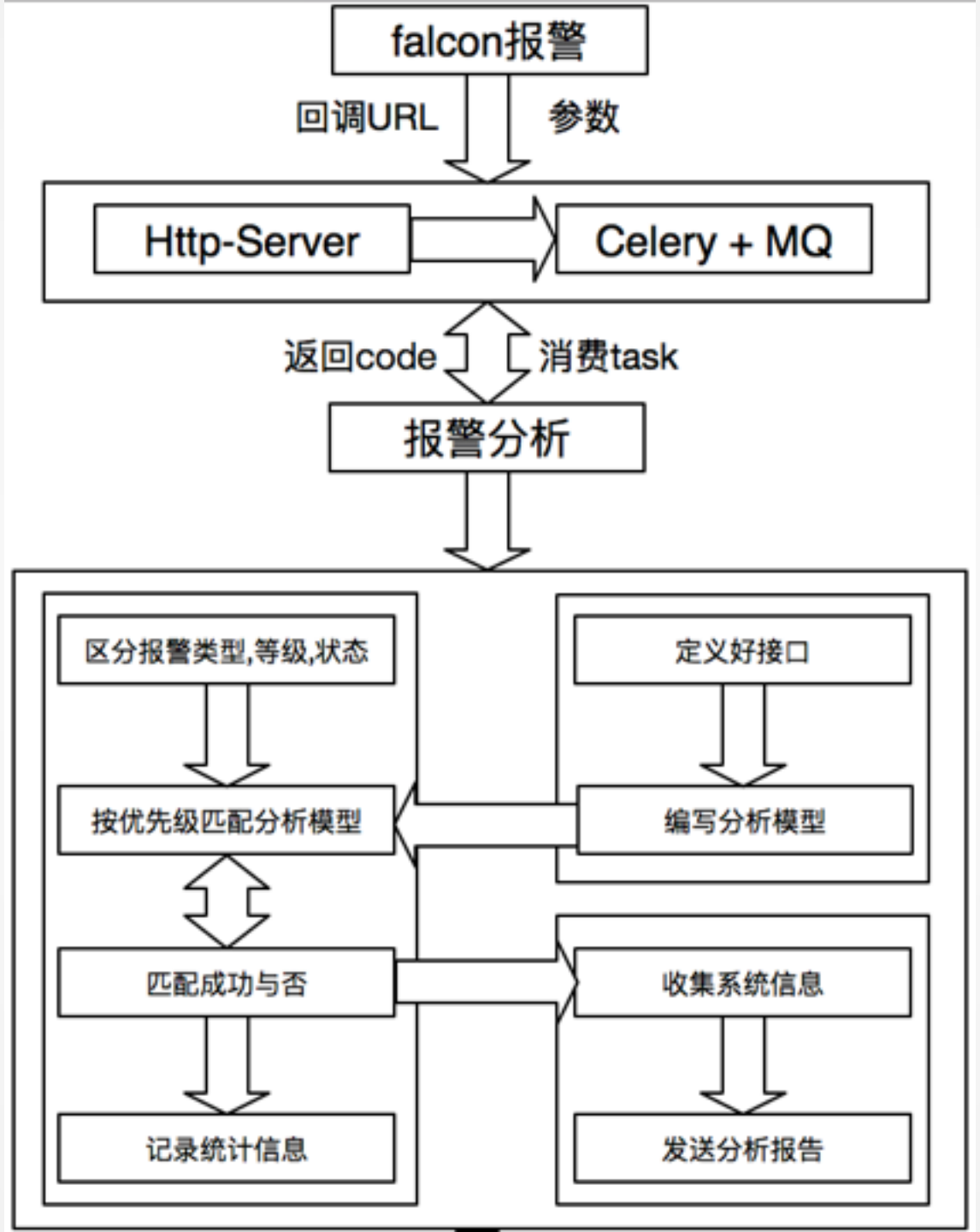
分析模块


分析模型
接口标准化
定义好接口输入输出，所有模型都能适用
模型优先级
跟据统计结果，调整分析模型优先级，尽可能早的成功匹配
易扩展性
支持 Python,Shell,Go….

统计模块


覆盖率
匹配率
准确率
•已覆盖到的报警类型
•各种报警类型次数
•按权重进行人力分工
•成功匹比例
•每一种模型匹配比例
•匹配准确率
•低准确率模型需要修正
优化分析流程
提高分析系统重要性

分析流程图


falcon报警
回调URL
参数
Http-Server
Celery + MQ
返回code
消费task
报警分析
区分报警类型,等级,状态
定义好接口
按优先级匹配分析模型
编写分析模型
匹配成功与否
收集系统信息
记录统计信息
发送分析报告

Part 3

运行现状

优势，覆盖率，匹配率

系统优势

系统运行之前

每一个报警都需要及时登陆到线上快速分析原因，告知相关研发人员

- 重复劳动
- 5min定位原因
- 人工通知RD

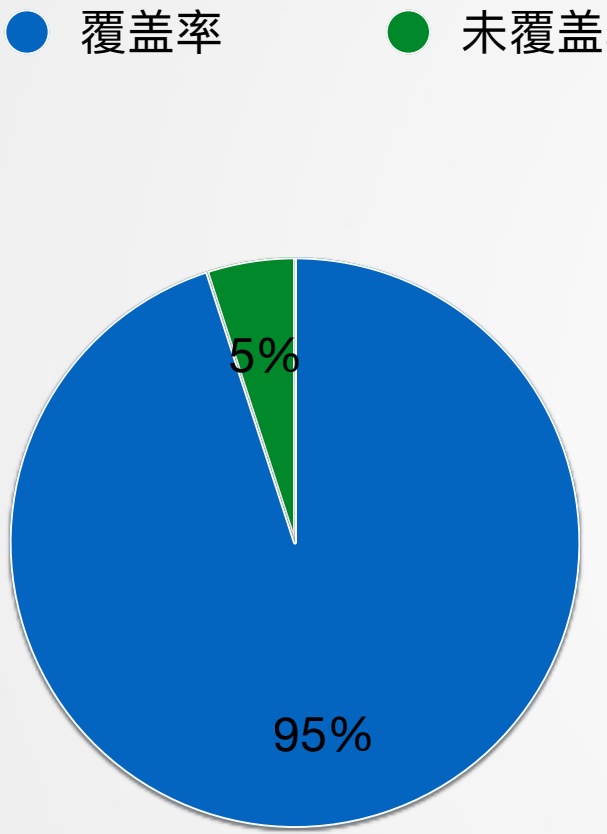
VS

系统运行之后

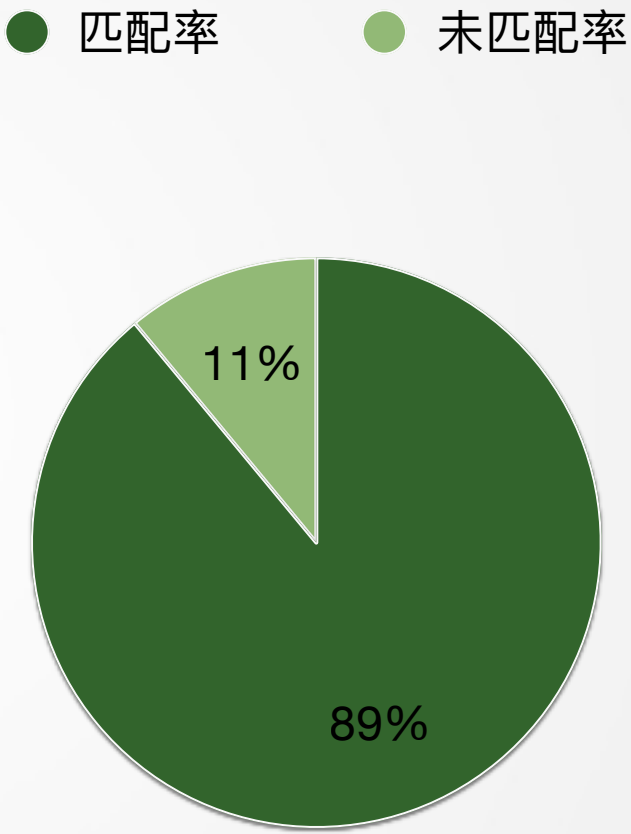
登陆系统之前，报警原因就已经录入系统，自动通知研发人员

- 程序来干
- 30s定位原因
- 通过配置可以自动通知RD

运行现状



&



Part 4

举个栗子

同步延迟分析过程

同步延迟分析

判断报警类型

```
if str(argu_module.status).upper() == "OK" or argu_module.priority >= ALARM_LEVEL:
    log.result_print(0,"SUCCESS","报警恢复,或者报警等级不处理。主机名:%s, metric: %s,
elif argu_module.step > 1:
    log.result_print(0,"SUCCESS","不是第一次报警,已经在处理")
else:
    # 开始分析之前，先向task表， detail表保存一条记录
```

匹配分析模型

```
for analyze_module in analyze_lists:
    analyze_status = 1 # 为了区分是匹配失败，还是没有对应的分析模型
    analyze_cmd = "cd %s; ./%s -e %s -h %s" % (SCRIPTS_DIR,analyze_module.analyze_s
    log.process_print("processing","尝试匹配分析模型,分析模型名:%s " % analyze_modu
    (script_status, analyze_output_argument) = commands.getstatusoutput(analyze_cmd
    if script_status ==0:
        # 脚本执行没有报错
```

获取系统状态

```
if metric in LOGS_DICT:
    analyze_log_types = LOGS_DICT[metric]
    _do_mtadba = 0
    if "syslog" in analyze_log_types and "mtdba" in analyze_log_types:
        _do_mtadba = 1
    #获取系统状态
    analyze_cmd = "%s/aly_background.sh -e %s -h %s -d %d" %(SCRIPTS_DIR,eventid, endpoint,_do_mtadba)
```

发送报告

```
# 将分析结果保存
db.update_detail_emailcomment(analyze_html,xm_url,argu_module.eventid)
log.process_print("processing","将分析结果保存到detail表中",argu_module)
# 更新task表update_time,记录整个过程花费的时间。
db.dml("update task set update_time = '%s' where eventid = '%s'" % (get_now(), argu_module.eventid))

# xm 发送 url，邮件直接发送结果
sendMail(analyze_html,argu_module)
sendXM(xm_url,argu_module)
```

报警机器:[redacted] 报警时间:2016-06-28 16:08:00 报警类型:Seconds_Behind_Master 报警事件: 1467101280_s_6021_2111_42a97ef414bfcdf143b536cbf30f87b6

心有用

机器硬件信息:物理机,SAS

服务组screen地址: http://d.falcon.sankuai.com/screen/7758?start=-3600

今日iolog :

```
FireWare 错误日志：无
S.M.A.R.T 错误日志：无
Media Error Count：无
Megacli LDInfo Optimal:无
Megacli FwTermLog :

    06/28/16  2:05:44: BQ27501FwVersion : 1.23
    06/28/16  3:40:01: BQ27501FwVersion : 1.23
```

今日kernlog: 无

详细分析如下:

```
执行的SQL：delete from [redacted] where crashTimeStamp<'2016-06-20 15:45:57' limit 1000000
 它对应的binlog文件:mysql-bin.002551,GTID:af021e97-e617-11e5-a9eb-ecf4bbd62070:248163661,在主库执行的时长为142s,它开始执行的时间为20160628 16:03:59, 完成时间为:20160628 16:06:21
```

当前mysql 状态:

	QPS TPS					Hit%		threads				bytes	
time	ins	upd	del	sel	lud	lor	hit	run	con	cre	cac	recv	send
16:08:53	0	0	0	0	0	0	100.00	0	0	0	0	0	0
16:08:54	84	3	0	218	87	783052	100.00	1	100	0	84	291k	295k
16:08:55	64	3	0	147	67	971149	100.00	1	100	0	84	309k	206k
16:08:56	61	2	0	172	63	1048594	100.00	1	100	0	84	255k	232k
16:08:57	49	1	0	161	50	776171	100.00	1	100	0	84	116k	236k
16:08:58	65	2	0	155	67	944056	100.00	1	100	0	84	258k	263k

当前innodb 状态:

	innodb bp pages status				innodb data status				innodb log		his log(byte) read query					
time	data	free	dirty	flush	reads	writes	read	written	fsyncs	written	list	uflush	uckpt	view	inside	que
16:08:59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16:09:00	7052567	5478	102582	912	0	1009	0	40.6m	6	12.1m	12224	5.3m	815.0m	0	0	0
16:09:01	7052570	5664	102618	832	0	915	0	37.6m	7	11.6m	12283	3.0m	826.9m	0	0	0
16:09:02	7052577	5829	102097	1427	1	1538	16k	56.5m	10	11.9m	12381	5.6m	839.7m	0	0	0
16:09:03	7052584	6026	102155	1161	1	1271	16k	50.0m	9	13.7m	12439	5.9m	802.9m	0	0	0
16:09:04	7052588	6226	101893	1073	2	1157	32k	46.6m	7	13.3m	12505	6.1m	815.6m	0	0	0

当前机器状态:

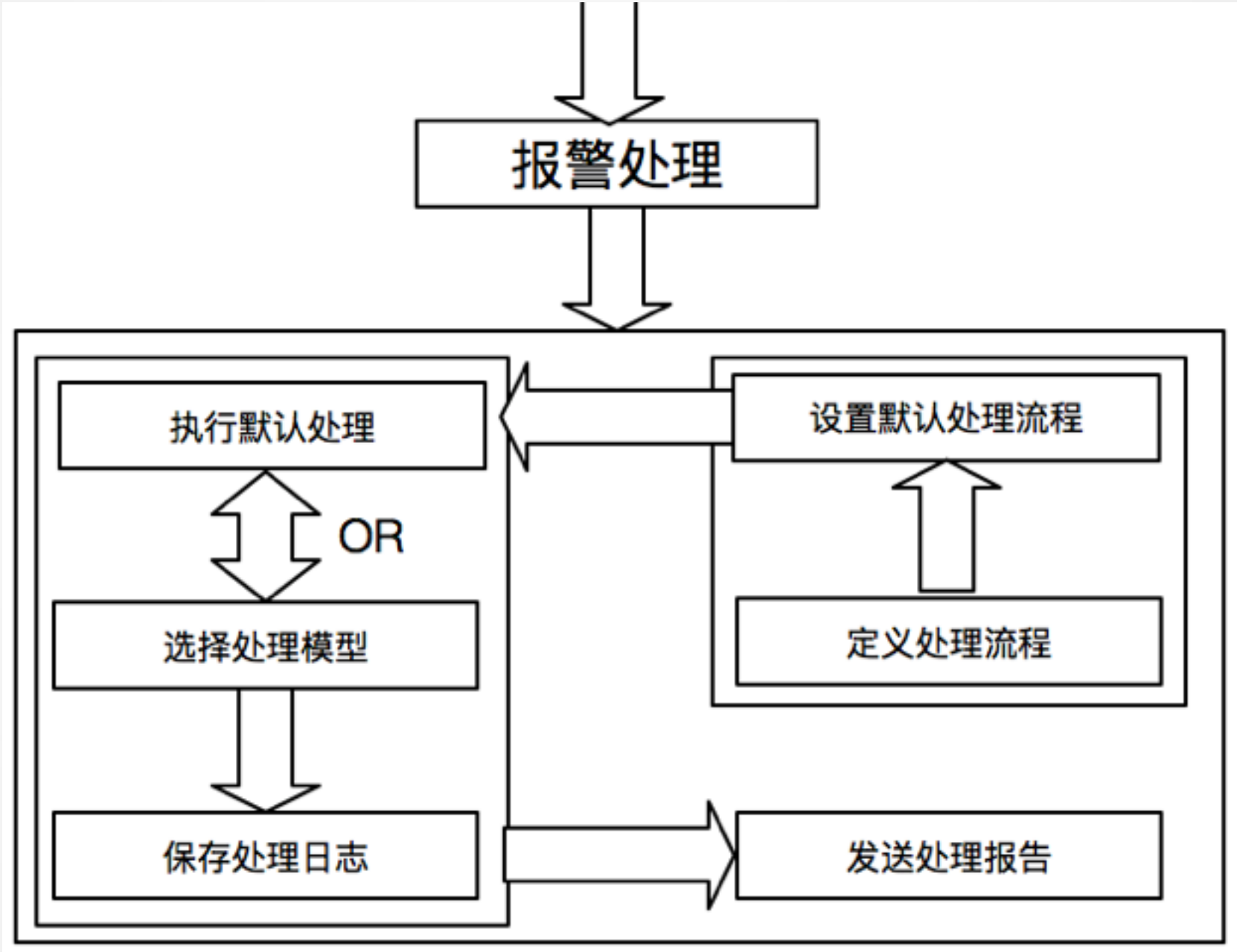
	load avg			cpu usage				swap		mem usage			io usage							
time	1m	5m	15m	usr	sys	idl	iow	si	so	cached	free	used	r/s	w/s	rkB/s	wkB/s	queue	await	svctm	%util
16:09:06	1.25	0.56	0.24	0	0	99	0	0	0	7150	8622	120391	0.5	36.6	4.2	1113.6	0.0	0.5	0.2	0.6
16:09:07	1.25	0.56	0.24	4	0	95	0	0	0	7151	8615	120398	8.9	1343.0	142.5	47113.2	0.3	0.2	0.0	4.2

Part 5

TODO

根据分析结果自动处理

处理流程图



梦想

持之以恒

永不放弃

谢谢大家!

汇报人：龙雪刚