

资源管控

- 文档历史
- 摘要
 - 编写目的
 - 项目背景
 - 任务概述
 - 术语与缩略语
 - 参考资料
- 系统分析设计
 - 核心业务规则
 - 1、当前容量评估脚本流程
 - 资源测算流程改造
 - 2、集群资源评估
 - 容量管控流程
 - 影响因素
 - 配额计算流程
 - 集群资源管控视图
 - 资源管控优化流程
 - 3、中间件资源评估 【待补充】
 - 接口文档
 - 数据库设计
- 非功能性特性设计
 - 监控

文档历史

修订日期	修订内容	修订版本	修订人
22/04/15	集群资源管控方案	v0	yiran.jin

摘要

编写目的

此分析说明书为提供技术设计方案，从模块划分、核心业务规则、可扩展性等方面做出了全面的概括性说明，为后期开发、接入方接入、使用提供指导和帮助。

项目背景

现有的资源预算流程是基于二级产品线维度的年度规划，无法精准的进行项目组维度/集群维度的资源管控。同时，容量预估的时间粒度与资源分配不一致，导致无法清晰的预测到潜在的超额风险。因此，本次任务通过细化配额粒度，制定集群性能目标，定期容量巡检和超额告警进行资源管控。

任务概述

资源管控模块主要依据产品线的年度资源配额，细化项目集群的资源配额，制定年度容量配额与性能目标等功能。本次任务主要包括如下方面：

- 规划项目组维度、集群维度的年度资源配额，制定相关性能目标
- 可视化各项目组的资源使用情况
- 可视化各项目组的性能达标情况
- 支持容量超额告警

术语与缩略语

描述与本文档相关的业务或技术上的术语，如：

缩略语/术语	全称	说明

--	--	--

参考资料

容量评估SOP

系统分析设计

核心业务规则

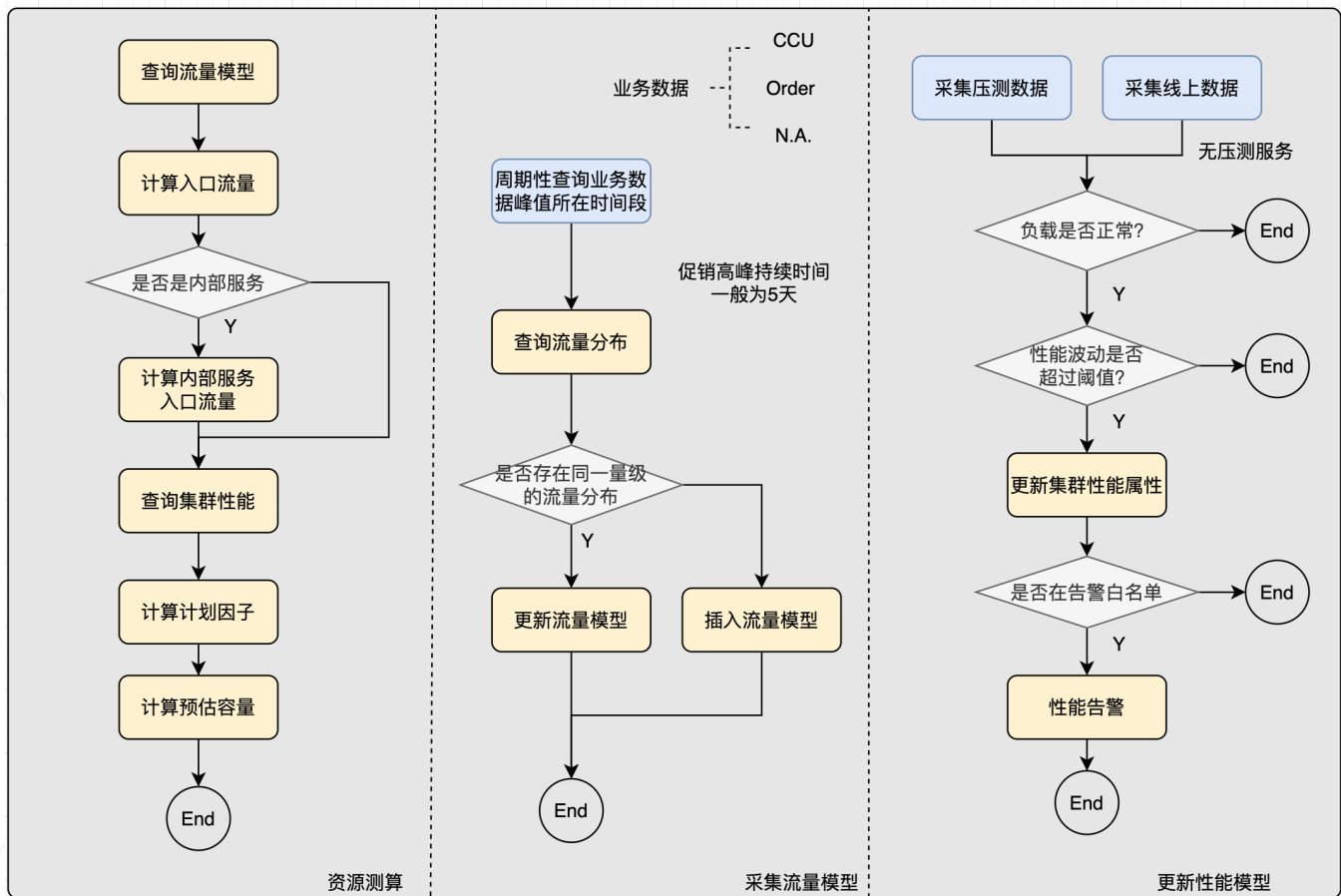
1、当前容量评估脚本流程

当前容量评估脚本流程

	步骤	操作详情		待改进项
确定流量模型	根据CCU数据，确定规模相近的历史促销流量数据源	表格操作		不适用于无接口级上报的服务 不适用于异步任务
	获取对应数据源的接口比例	api/report /static_evaluation/metric /urls/sync	开始时间，结束时间，集群列表 [hard code]	出入参hardcode，半自动化
确定链路模型	预估SLS内部下游服务的	/api//trace_model /calculate_trace_qps TestCalculateTraceQPS	读取指定sheet页面的接口流量数据，	下游链路的产出是指定集群的定制化逻辑
确定性性能模型	计算集群性能（单机QPS）	api/report /static_evaluation/metric /performance/sync TestSyncPerfMetrics	模拟压测的开始结束时间，涉及的地区，参考LIVE数据的时间与集群，替代调整ratio，定时压测任务id，参考模拟压测的id	基于的流程是模拟压测后置动作，会创建压测任务，压测任务的配置是hard code的 只能覆盖到有压测数据的服务，无法评估其余集群 大促评估流程不符合当前使用场景，需要改造
确定容量模型	同步当前实例数至表格中	api/report/static_evaluation/metric/sync		需要兼容多机房的多数据源，支持集群可配
	checkout侧服务	max(各服务预估流量 / 单机qps上限, 2)		order侧服务与异步任务服务的预估不准确
	履约刘翔服务	order系数直接计算		

资源测算流程改造

方案一：



通过改造生成流量模型与性能模型的过程，实现资源测算自动化。

方案二：

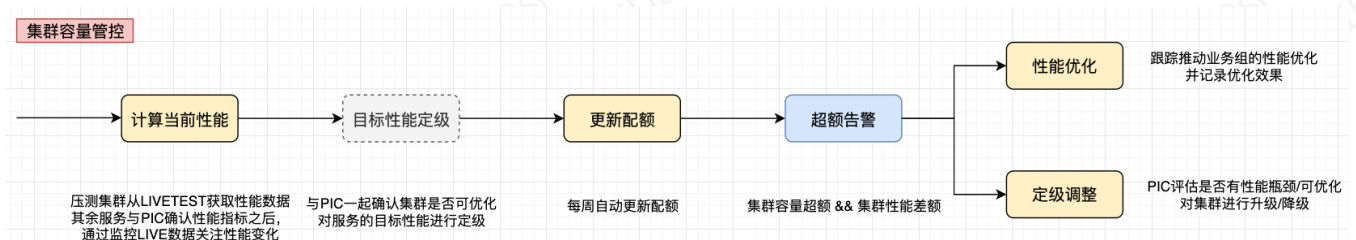
基于机器学习/神经网络的预估

2、集群资源评估

集群容量指标

指标	含义
CPU	集群分配的CPU核数
MEM	集群分配的内存大小(GB)

容量管控流程



计算当前性能：

第一版性能数据取自LIVETEST与业务组评估，之后的性能监控可分为非压测集群与压测集群两类。

- 非压测集群：基于现有性能，对LIVE集群行为进行分析，监控性能变化
- 压测集群：数据源取自自动化压测和模拟压测

目标性能定级 | 优化管控：

基于SLS当前读、写、Admin分离的微服务架构，本方案采取对集群性能定级、追踪优化进度，灵活升降级的方式进行资源管控。

- 不同等级配置分级的优化目标
- 支持追踪与记录优化排期与优化效果
- 对于PIC评估存在性能瓶颈的集群，由AM/M决定是否可降级
 - 降级会导致配额增加，如果是大规模集群，可能会对其余集群的配比产生较大影响，需要谨慎评估
- 对于性能优化明显的集群，自动升级

更新配额：

- 以产品线为维度的资源配额是一定的，然而受各集群的性能变化、新增调用方的接入、系统重构等阶段的影响，项目组每月动态调整细粒度配额
- 对配额变动设置缓冲buffer，eg. 0.2：在0.8-1.2之间的变动，保持原有配额不变
- 对配额增多设置安全红线，eg. 0.5：配额变动超过1.5的集群，需要由PIC分析变动原因并制定优化方案

超额告警：

- 性能下降：定位下降原因，录入优化排期
- 流量激增：推动上游服务，是否可优化调用策略？

影响因素

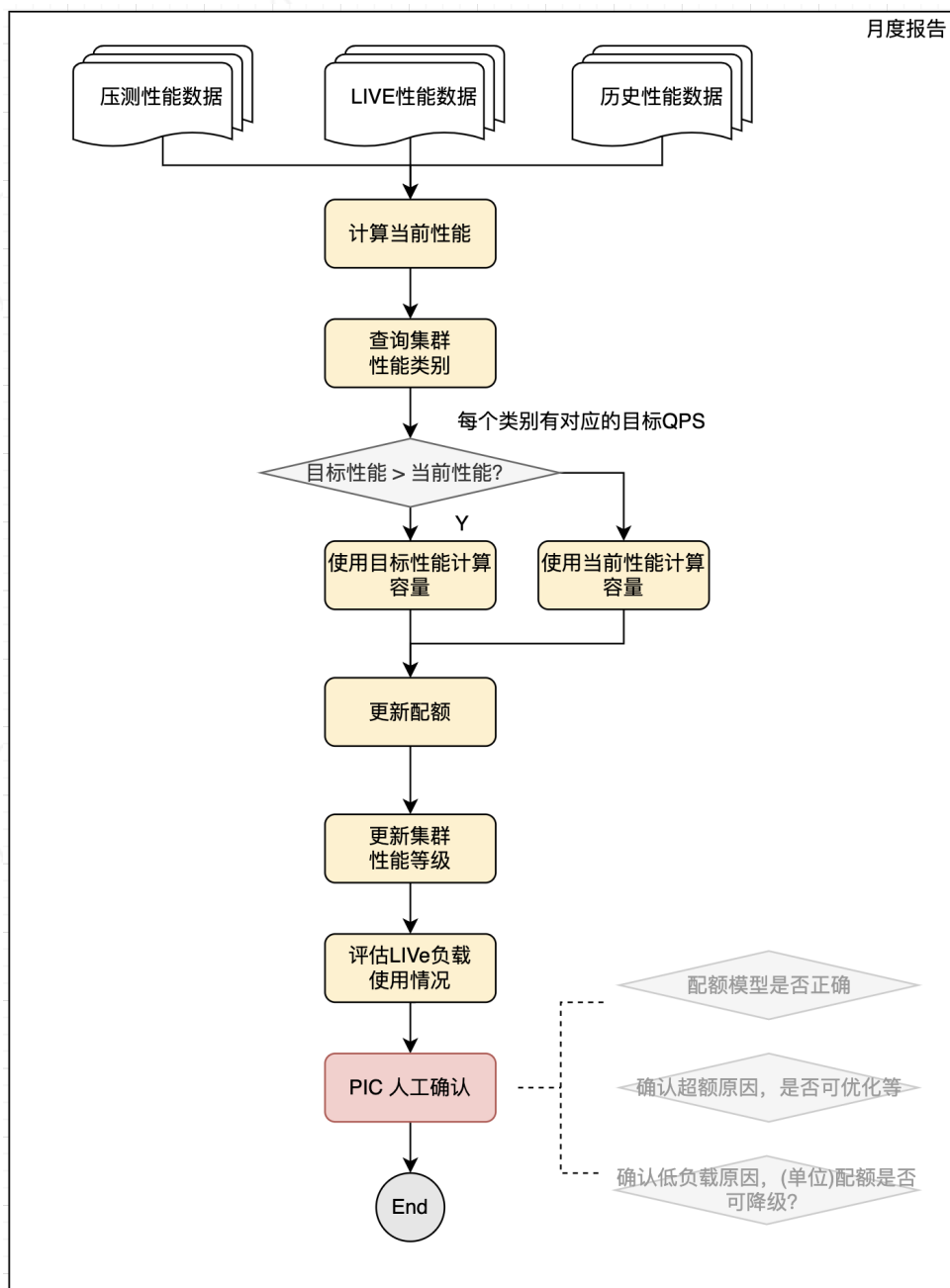
专项	所属项目组	PIC	影响范围	描述	文档/数据连接
下线服务比例	xxx	xxx	xxx	xxx	xxx
迁移灰度比例					
新增服务					

配额计算流程

基于数据可靠性、灵活性等角度出发，配额计算流程可以按照一下几个方案进行计算：

	方案	计算方式	优点	缺点
方案一	依据性能模型的分布	预估流量 * 变化因子 / 当前性能 / 预估容量总量 * 总配额 * buffer	数据贴近服务当前性能	没有对服务进行评审/分类，全部服务需要等比例优化/提升性能
方案二	依据容量模型的分布	当前容量 * 变化因子 / 当前总量 * 总配额 * buffer	容量配比贴近当前分布	容量模型的单一维度的计算，没有考虑对于年底大促的承载能力
方案三	在方法二的基础上，结合服务负载情况优化容量模型	if 负载 < 负载Base : 理论容量 = 容量 * (负载Base/负载) * 相关因子 T = 理论容量 * 变化因子 / 当前总量 * 总配额 * buffer	一定程度上，消除现有容量模型中资源浪费引入的误差	当服务负载较低时，没有经过验证而计算出的负载线性关系时不准确的
方案四	根据服务分类计算	if 当前性能 > 目标性能： 预估流量 * 变化因子 / 当前性能 / 预估总量 * 总配额 * buffer else : 预估流量 * 变化因子 / 目标性能 / 预估总量 * 总配额 * buffer	从业务复杂度出发，对集群进行分类，每一个类别设置不同等级的目标性能，设置不同等级的优化力度，可以更加科学的对系统整体性能进行管控	如何对服务分类，评判标准？

由于容量模型引入的误差较大，指标之间的相关性难以精确评估【样本不充分】，配额计算主要从性能的角度出发，采取下述流程：



由于集群配额与业务逻辑以及迁移进度强相关，需要人工介入确认系统配额是否合理、系统性能类别是否可升级、单位配额是否可降级等决策。

集群资源管控视图

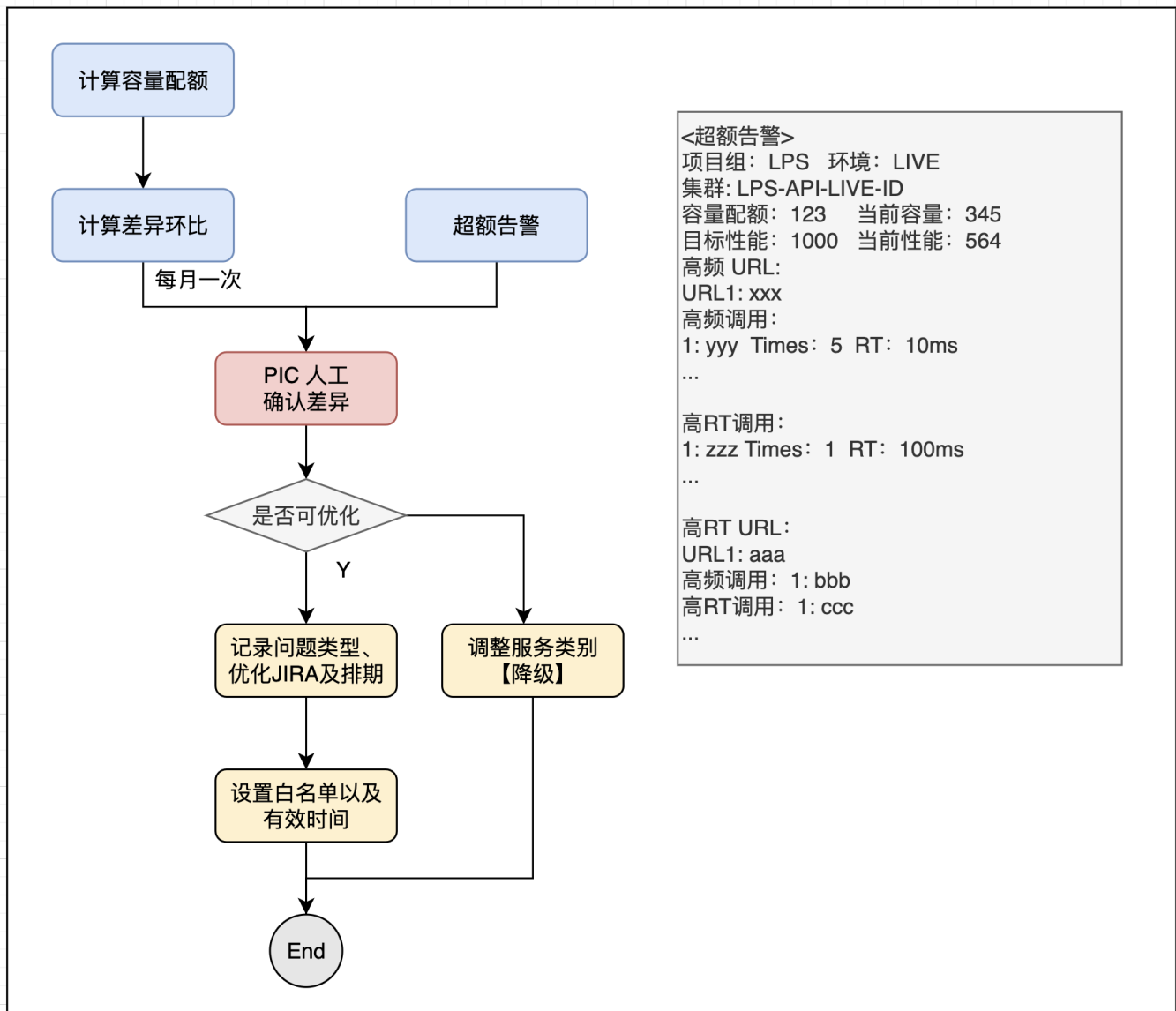
- 支持查看**项目组维度**容量/性能达标百分比
- 支持查看**集群维度**容量/性能达标情况
- 支持查看**资源使用率**
- 支持查看配额详情
- 支持查看预估详情

服务分类/性能定级/确定目标单机性能 [Need updated/ Need checked by PIC]

评估数据来源：RT、Latency

性能等级	基线示例	描述	集群示例
第一档	6000		LCOS-GRPC
第二档	4000		SLS-RATEAPI
第三档	2000		LLS-GRPC
第四档	1500		LPS-API, LFS-GRPC
第五档	1000		LPS-TASK
第六档	200		LCS-TRAFFIC
第七档	70		SLS-LOGISTIC、LCS-API
第八档		异步任务	使用负载与流量倍数进行计算
第九档		Admin 不受大促流量影响	容量不变

资源管控优化流程



后期可以从DMS拉取优化JIRA的状态

配额管控流程

xxx

集群资源管控操作/配置指南

待补充

3、中间件资源评估 【待补充】

接口文档

http://apidoc.i.ssc.shopeemobile.com/project/2255/interface/api/cat_14964

数据库设计

配额表：

```
CREATE TABLE `resource_quota_tab` (  
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `level` tinyint(4) NOT NULL DEFAULT '0' COMMENT '1:biz_line 2:project 3:domain',  
  `biz_id` int(20) unsigned NOT NULL DEFAULT '0',  
  `resource` varchar(64) NOT NULL COMMENT 'az/cache/db/mq',  
  `component` varchar(32) NOT NULL COMMENT 'CPU/MEM...',  
  `quota` int(11) NOT NULL DEFAULT '0',  
  `allocated` int(11) NOT NULL DEFAULT '0',  
  `used` int(11) NOT NULL DEFAULT '0',  
  `status` tinyint(4) unsigned NOT NULL DEFAULT '0' COMMENT '0:alert 1:silent',  
  `effective_time` int(11) unsigned NOT NULL DEFAULT '0',  
  `ctime` int(11) unsigned NOT NULL DEFAULT '0',  
  `mtime` int(11) unsigned NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`),  
  KEY `idx_resource_time` (`resource`, `effective_time`),  
  UNIQUE KEY `uniq_biz_component_time` (`level`, `biz_id`, `resource`, `component`, `effective_time`)  
);
```

历史流量表：

```
CREATE TABLE `historical_biz_scale_tab` (  
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `biz_type` tinyint(4) NOT NULL DEFAULT '0' COMMENT '1: CCU 2: ORDER',  
  `region` varchar(32) NOT NULL,  
  `volume` int(20) unsigned NOT NULL DEFAULT '0',  
  `start_time` int(11) unsigned NOT NULL DEFAULT '0',  
  `end_time` int(11) unsigned NOT NULL DEFAULT '0',  
  `ctime` int(11) unsigned NOT NULL DEFAULT '0',  
  `mtime` int(11) unsigned NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`),  
  KEY `idx_sample_time` (`start_time`),  
  UNIQUE KEY `uniq_biz_component_time` (`biz_type`, `region`, `start_time`)  
);
```

接口比例表：

```
CREATE TABLE `rpc_ratio_tab` (  
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `ratio_version` varchar(128) NOT NULL DEFAULT "",  
  `domain_name` varchar(256) NOT NULL DEFAULT "",  
  `rpc_name` varchar(256) NOT NULL DEFAULT "",  
  `rpc_ratio` decimal(19,6) NOT NULL DEFAULT '0.000000',  
  `sample_num` int(11) unsigned NOT NULL DEFAULT '0',  
  `ctime` int(11) unsigned NOT NULL DEFAULT '0',  
  `mtime` int(11) unsigned NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`),  
  KEY `idx_ratio_version_domain_name` (`ratio_version`,`domain_name`)  
);
```

操作历史表:

```
CREATE TABLE `ops_history_tab` (  
  `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,  
  `ops_type` int(11) unsigned NOT NULL DEFAULT '0',  
  `operator` varchar(128) NOT NULL,  
  `content` mediumtext NOT NULL,  
  `ctime` int(11) unsigned NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`),  
  KEY `operator` (`operator`),  
  UNIQUE KEY `uniq_ops_operator_time` (`ops_type`,`operator`,`ctime`)  
);
```

非功能性特性设计

监控

指标	指标说明	告警项	链接
cur_pod_num	监控是否有机房迁移等操作，可能会导致promql语句失效	cur_pod_num == 0 && past_pod_num != 0	xxx