**监控系统接入说明**

V 0.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **作 者** | 成桂锋 | | **创建日期** | 2018-05-02 |
| **修改日期** | **修订版本** | **修改说明** | | **修改人** |
| 2018-05-02 | 0.1 | 创建 | | 成桂锋 |
|  |  |  | |  |
|  |  |  | |  |
|  |  |  | |  |

目 录

[1. 基础监控 2](#_Toc513040559)

[接入前准备 2](#_Toc513040560)

[命令行调用方式 2](#_Toc513040561)

[api说明 3](#_Toc513040562)

[2. 多维监控 4](#_Toc513040563)

[新增监控项 4](#_Toc513040564)

[上报协议 5](#_Toc513040565)

[上报方式 6](#_Toc513040566)

[命令行调用方式 6](#_Toc513040567)

[api 6](#_Toc513040568)

[进程自己打日志方式（文件名为\*.log：日志名字自定） 7](#_Toc513040569)

[3. 告警配置 7](#_Toc513040570)

[4. 告警屏蔽 9](#_Toc513040571)

# 基础监控

## 适用场景

1. 简单数值型曲线展示
2. 简单数字型/字符型告警

## 接入前准备

1. 到基础监控web页面申请监控ID（fid），需要发送告警的还需要建立告警策略获取策略ID（sid）
2. 接入的机器需要安装op\_agent\_d，co\_agent：目前基本都已经自动安装

## 命令行调用方式

指令位置：/data/services/op\_agent\_d/tools

1. 发送告警：
2. send\_alarm fid sid alarm\_msg

fid、sid都不能为0，要和页面申请的一致

1. send\_alarm fid 0 proc\_name alarm\_msg

sid为0时，会根据proc\_ame自动查找对应监控项下进程名匹配的、权重最高的策略来发送告警

1. send\_alarm fid sid proc\_name alarm\_msg

fid、sid要和告警策略匹配，proc\_name可以为任意

1. send\_alarm "fid=fid\_value&sid=sid\_value&msg=alarm\_msg&appname=proc\_name&pid=pid\_value&port=port\_value"

fid/sid/msg不可缺失，其他为可选参数：

1. 可添加pid、port参数：必须为int32范围内整数
2. 可添加ip参数：默认告警显示时使用发出告警的机器ip，当提供此参数时，用此参数代替
3. 当sid为0时，appname必须存在，且应该匹配策略配置的进程名

注：proc\_name、pid、port可以理解为一个变量名称，未必要对应真正的进程名、pid、port，可以灵活使用，只要类型一致即可

例子:

1. send\_alarm 123 456 test
2. send\_alarm 123 0 alarm test
3. send\_alarm 123 456 alarm test
4. send\_alarm "fid=123&sid=456&msg=test&appname=alarm"
5. send\_alarm "fid=123&msg=test&appname=alarm"
6. send\_alarm "fid=123&msg=test&appname=alarm&info=extra"
7. 发送监控数据
8. send\_value fid value

发送单一监控项

1. send\_value "fid=fid\_value&value=value\_value"

send\_value "fid=fid\_value1&value=value\_value1|fid=fid\_value2&value=value\_value2"

send\_value "fid=fid\_value&value=value\_value&appname=proc\_name&pid=pid\_value&port=port\_value"

可同时发送多个监控项，并可添加额外参数：

1. fid/value为必选参数
2. appname/pid/port/msg/ip为可选参数：msg为额外的告警信息，ip可替代默认的本机ip
3. 当appname存在时，必须和配置的众多策略中的某个进程名一致

例子:

1. send\_value 123 456
2. send\_value "fid=123&value=456&pid=789&appname=test"
3. send\_value "fid=123&value=456&pid=789&appname=test&port=123"
4. send\_value "fid=123&value=456&pid=789&appname=test&info=test"
5. send\_value "fid=123&value=456|fid=321&value=789"
6. 抑制说明

抑制次数计数所用的key为：fid\_sid\_appname\_pid\_port，msg是不包含在计数的key内的，同样的组合会根据抑制策略进行抑制。如果需要每条都告警，可以生成随机的pid、port

## api说明

1. 代码地址：git a/imserver/server\_common/server-lib/

* op\_agent\_client.cpp
* op\_agent\_client.h
* op\_agent\_protocol.h

1. Include头文件：op\_agent\_client.h
2. 声名agent实例：opdev::AgentClient g\_agent\_client;
3. 初始化socket：g\_agent\_client.init\_socket()， 请检查返回值
4. 调用函数发送告警

bool send\_alarm(const char \*data); // 可自行构建json字符串，格式及必须的域请参考alarm\_to\_json函数实现

bool send\_alarm(int fid, int sid, const char \*msg); // 参考命令行用法

bool send\_alarm(int fid, int sid, const char \*appname, const char \*msg); //参考命令行用法

bool send\_alarm(int fid, int sid, const char \*appname, const char \*msg, const char \*info); // info提供额外信息，实际也可以放到msg里面

1. 调用函数发送数据

bool send\_value(const char \*data); // 可自行构建json字符串，格式及必须的域请参考value\_to\_json函数实现

bool send\_value(int fid, double value); // 参考命令行用法

bool send\_value(int fid, double value, const char \*appname); // 参考命令行用法

bool send\_value(int fid, double value, const char \*appname, int port); // 参考命令行用法

bool send\_value(int fid, double value, const char \*appname, int port, const char \*info); // info提供额外信息，告警的时候会加到告警信息

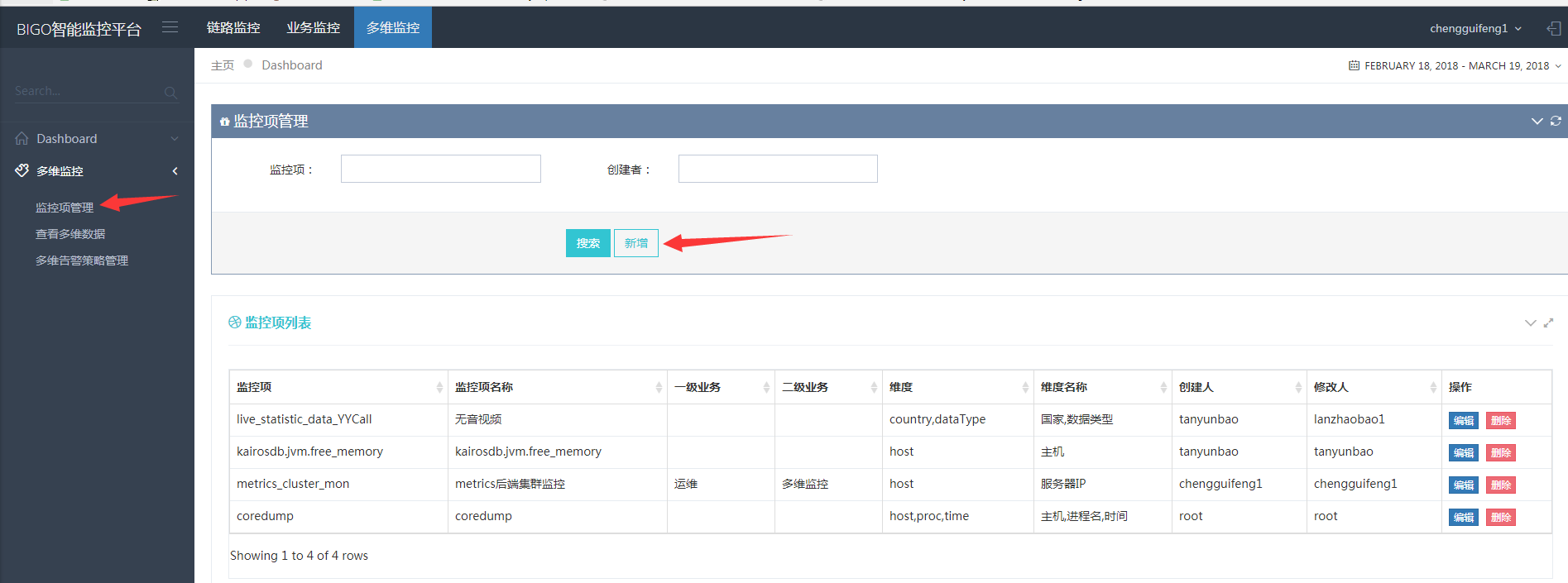
# 多维监控

## 适用场景

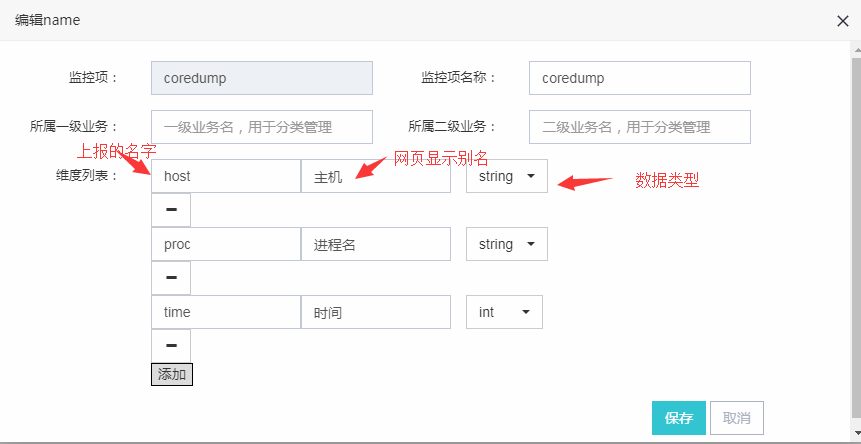
复杂的数值型

## 新增监控项

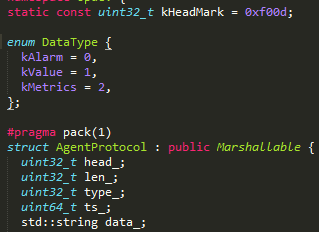
1. 到web页面新增监控项信息：<http://mon.sysop.bigo.sg/metrics_query.html>，确保名字唯一



2. 填写监控项对应维度信息



## 上报协议



|  |  |  |
| --- | --- | --- |
| head\_ | uint32\_t | 值应为0xf00d |
| len\_ | uint32\_t | 含头部在内的长度 |
| type\_ | uint32\_t | metrics对应为2 |
| ts\_ | uint64\_t | 时间，单位为秒 |
| data\_ | string | Json格式字符串：  name：必须，申请的监控项名字  value：必须，long或者double类型数字  timestamp：必须，上报时间，单位为毫秒  type：可选  tags：必须，各种维度信息，维度信息对应值不能为空  注意：   1. Tags的key、value不要有引号、双引号等特殊字符 2. Tags内不要含有太多可变值，这样会令维度组合无限放大，严重影响系统性能 3. 另外每条数据是以name、timestamp、tags为key入库，同一个key的值会被最新的值覆盖，也就是只能支持毫秒级不同的数据   例子：  {  **"name"**: "cpu.avg",  **"timestamp"**: 1359788400000,  **"type"**: "long",  **"value"**: 123,  **"tags"**:{**"host"**:"1.1.1.1", **"type"**:"1min",**"core"**:1}  } |

## 上报方式

注：因为不少人使用多维进行字符串告警，但是维度组合越多，查询越慢，字符串不断变化会令tag组合暴增，查询很慢。因此，如果tag名字为alarmMsg，将不会入库，但告警可以配置显示。

### 命令行调用方式

指令位置：/data/services/op\_agent\_d/tools，timestamp默认为当前时间

1. send\_metrics name value tag\_1=value\_1 tag\_2=value\_2
2. send\_metrics "name|value|tag\_1=value\_1&tag\_2=value\_2"

例子：

1. send\_metrics cpu.avg 1.23 host=192.168.1.1 cpu=0 type=1\_min
2. send\_metrics "cpu.avg|1.23|host=192.168.1.1&cpu=0&type=1\_min"

### 原始api（不推荐使用）

1. 代码地址：git a/imserver/server\_common/server-lib/

* op\_agent\_client.cpp
* op\_agent\_client.h
* op\_agent\_protocol.h

1. Include头文件：op\_agent\_client.h"
2. 声名agent实例：opdev::AgentClient g\_agent\_client;
3. 初始化socket：g\_agent\_client.init\_socket()， 请检查返回值
4. 调用各种函数发送数据
5. bool send\_metrics(const char \*data)

data：符合格式规定的json字符串（参看上报协议data\_说明）

例子：

struct timeval tv;

gettimeofday(&tv, NULL);

long int ms = tv.tv\_sec \* 1000 + tv.tv\_usec / 1000;

ss << "{";

ss << "\"name\":\"" << "cpu.avg";

ss << "\",\"value\":" << 123.123;

ss << ",\"timestamp\":" << ms;

ss << ",\"tags\":{";

ss << "\"" << "host" << "\":\"" << "3.3.3.3" << "\"";

ss << ",\"" << "num" << "\":\"" << "3" << "\"";

ss << ",\"" << "slot" << "\":\"" << "1" << "\"";

ss << "}}";

send\_metrics(ss.str().c\_str());

1. bool send\_metrics(const char \*name, double value, int tags\_num, ...)

例子：

send\_metrics("cpu.max", 10, 2, "host", "1.1.1.1", "num", "0"); // 2为tag的数目，后面是key、value、key、value。使用此函数要小心，tag个数不能出错，否则会coredump；另外，此函数tag类型只能是string类型，不过上层会根据配置类型0做转换

### 进阶api（暂时只封装了发送某些执行metrics函数，可以参照相关函数封装自己的上报函数）

1. 代码地址：git a/imserver/server\_common/server-lib/

* stat.cpp
* stat.h

1. Include头文件：stat.h
2. 初始化agent：

* Stat::getInstance()->init(\_\_serverConfig.getName())：metrics的tag默认包含ip，appname需要提供给init
* Stat::getInstance()->init(\_\_serverConfig.getName(), \_\_serverConfig.getGroupId())：metrics的tag默认包含ip，appname、groupid需要提供给init

1. 封装函数说明：
   1. stat(const std::string& item, const double value, const std::map<std::string, variantType>& tags)：可以使用此函数封装自己的metrics api，tags值支持int、string类型，请参照stat.cpp封装好的函数封装自己的上报函数

* item：metrics监控项名字
* value：监控值
* tags：参考上报协议说明
  1. statQueueMonitor(const uint64\_t queueSize)：对应QueueMonitor.CPP中的QueueMonitor::TimerCallback()打印的queueSize
  2. statQueueMonitor(const uint64\_t queueSize, const uint64\_t dealPacket, const uint64\_t dropPacket, const int activeThreads, const int procSize)：对应QueueMonitorCodel.CPP中的QueueMonitor::TimerCallback()打印的各个变量
  3. statCallGWFail(const std::string& service, const std::string& interface)：访问gateway发生错误时调用
  4. statDbErr()：访问Db发生错误时调用
  5. statLinkMsg(const uint32\_t uri, const uint32\_t totalNum, const uint32\_t timeoutNum, const double timeoutRate, const uint32\_t timeoutUidNum, const uint32\_t rtt)：对应LinkMsgStat.CPP中的CLinkMsgStat::printStat()打印的各个变量

## 进程自己打日志方式（文件名为\*.log：日志名字自定）

注意：此接入方式有丢失数据风险，特别是数据量大的日志。因为此目录的日志会由系统进行切割压缩，在新增加了日志，而发送进程又未读取，并且系统进行压缩的时候，就可能会有部分数据丢失。

1. 日志路径：/data/yy/log/bims\_metrics
2. 数据格式：参看上报协议data\_说明

例子：

{"name":"metrics\_cluster\_mon","tags":{"host":"103.211.231.84"},"timestamp":1521443402000,"value":1}

### 告警配置

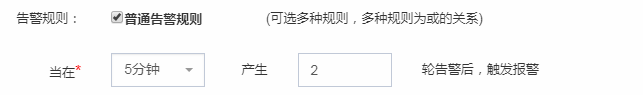
1. 告警策略配置网址：<http://mon.sysop.bigo.sg/metrics_strategy.html>
2. 新增策略





选项说明：

1. 监控项名称：必选。选择要增加策略的监控项。选择后，下方会出现各个维度的选项，可以根据需要选取，也可以不设置任何维度值
2. VALUE：可选，告警阈值，不填则不会将值作为判定告警条件
3. 抑制规则：必选
   1. 普通告警抑制



5分钟2次：是指5分钟内发生两次告警就告警，5分钟内不再告警

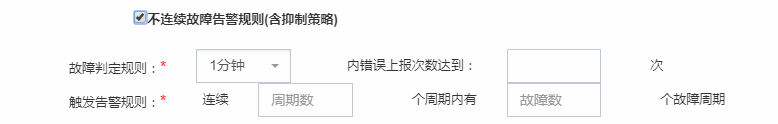
* 1. 连续故障告警



故障：1分钟内发生3次告警才形成1个故障。起点将会选取离第一个告警最近的2分钟倍数处开始按每2分钟划分单元格

触发告警：连续出现2个故障才告警

* 1. 不连续故障告警



故障：意义与上一规则相同

触发告警：是指在5个周期（每个周期1分钟）内，发生3个故障才告警，故障可以是不连续的

1. 所属业务：如果你不需要发送微会群，可以用默认值，如果你的微会群属于特定业务，需要选取对应的业务，否则可能发送失败
2. 通知模式：
   1. 请谨慎勾选短信选项，因为会产生费用，特别是告警频率高，人员多的策略
   2. 默认将会发送到联系人的个人微会
3. 告警联系人：必选。这项是web前端直接从微会个人信息数据库同步信息的，只有微会名字和中文名字一致才能同步到手机、微会号信息，微会名字不能有多余信息，比如名字“张三”，如果微会名称是“bigo\_张三”，这样不能同步到信息
4. 通知方案：
   1. 勾选接入一线值班会将告警发送到值班群
   2. 微会群：需要发送的微会群号，注意，这个群号是alarmType，不是微会显示的群号