架构

介绍开发步骤前，先了解下整个 TOPARGUS 系统的架构。

- agent: 报警采集端，分布于每一台机器上，上报监控信息给 proxy
- proxy: 报警上报中心，负责接收各个 agent 的告警信息，写入 redis 缓存
- consumer: 拉取 redis 缓存，根据 type 进行并行消费，处理整合数据，存入 db
- db: consumer 模块处理完成之后的数据持久化，也同时为 dash server 提供数据源
- dash server(dash): 服务于前端查询，从 db 中查询数据提供给前端

其中涉及到需要二次开发的模块主要是 agent 以及 consumer。

agent 子模块开发

agent 是监控报警的采集代理，部署于每一台机器上，通过分析日志的方式或者定时执行的方式采集数据，进行上报。

> https://github.com/smaugx/top-argus-agent.git

这里有两种方式提供数据，第一种是抓取日志关键字的方式，第二种是定时执行脚本采集的方式。这里我们暂时只讨论定时执行脚本的方式。

模块开发

定时脚本

工作原理是主进程会定时 x 分钟执行 top-argus-agent/agent/cron 目录下的脚本，接管 stdout，获取每个脚本的标准输出，然后包装该输出作为上报内容。

其中**脚本的输出格式为 json string，必要的字段是 type**:

```
{
  "key3": "value3",
  "key2": "value2",
  "key1": "value1",
  "type": "demo"
}
```

具体开发方式可以参考脚本 agent/cron/demo_cron.py

搜索日志关键字

暂时不介绍这种方式（精力有限，文档还没好）

agent 启动方式

agent 主进程负责执行采集任务的管理，其中多个子线程负责具体的采集业务。其中主要包括三大类：

- xtop.log 的监控，过滤关键字，组装数据
- cron/ 目录下的脚本定时执行，获取脚本的标准输出，组装数据
- 消费上报，从任务队列里获取上报内容，上报给 proxy

主进程启动方式

```
cd top-argus-agent
source  vvlinux/bin/activate
cd agent
nohup python argus_agent.py -a 127.0.0.1:9090 -f ./xtop.log > /dev/null & 2>&1
```

其中参数说明如下：

```
(vvlinux) [topuser@topchain-rec-nyc1-2019110703-1-x agent]$ python argus_agent.py
 -h
usage: argus_agent.py [-h] [-a ALARM] [-f FILE]

TOP-Argus Agent，拉取远程配置，报警采集并上报

optional arguments:
  -h, --help            show this help message and exit
  -a ALARM, --alarm ALARM
                        alarm proxy host, agent pull config and push alarm to
                        this proxy host, eg: 127.0.0.1:9090
  -f FILE, --file FILE  log file for agent to watch, eg: ./xtop.log
```

注意启动 agent 之前，确保 proxy 是启动正常的，不然 agent 可能会采用自己本地的默认配置启动（当然也没什么大问题）。

consumer 子模块开发

项目地址： top-argus

工作原理是从 redis 拉取数据，每个子模块（子进程）绑定到某一个 type 或者多个 type，进行并行消费。这里需要注意的是绑定 type 需要慎重考虑，为了并行消费，最好不要出现进程间通信的情况。

子模块

我们假定该子模块关心某一个 type 或者多个 type 的报警信息，那么实现一个 consumer。

具体可参考：consumer/demo_consumer.py。 核心在于实现成员函数:

```
    # focus on packet_info(drop_rate,hop_num,timing)
    def demo_alarm(self, content):
        slog.info('demo_alarm begin:{0}'.format(json.dumps(content)))
        # TODO(user) do something statictis or calculate

        # dump result to db
        self.dump_db()
        return True
```

这里要对上报的数据进行处理，汇总，统计，分析等，然后把处理完成的数据存储到 db 中。

关于 db 中表结构的设计文档暂时待定。（具体咨询 smaug)

consumer 主进程启动

```
(vvlinux) [root@topchain-archive-nyc1-2019121907-2 consumer]# python main_consumer
.py   -h
usage: main_consumer.py [-h] [-t TYPE] [-e ENV]

TOP-Argus consumer，多进程方式启动一个消费者，绑定到某个类型的报警内容，进行消费

optional arguments:
  -h, --help            show this help message and exit
  -t TYPE, --type TYPE  bind with alarm_type,eg: packet, networksize...
  -e ENV, --env ENV     env, eg: test,docker
```