方楠的行业版数据 TL 工作指北

方楠的行业版数据 TL 工作指北

1、计划节奏

如期望承担数据 TL 任务，建议一共分 3 周逐步过渡

- 第一周：阅读 gitlab-大数据 Document，尤其是文档运维工作，接手所有代码点赞，处理一次"失联节点故障"，布置禅道任务。
- 第二周：moonbrook 代码由被交接人合并，负责处理日常五事务
- 第三周：负责所有代码合并，完成一次爬虫技术方案设计和落地

（p.s.再提醒一下阅读 document 的重要性哈，运维工作文档说的非常全面，但是不够具体，此文档就是运维工作文档的延伸和实践解答）

2、日常运维五事务

事务 1:爬虫服务器维稳运行

保证爬虫服务器正常运行，一般不正常体现在以下几个现象

- 钉钉监控日志报节点失联，这会**导致服务完全不可用**。
- 提升爬虫速度的时候，速度不及预期。如设置提升 2 倍速度，实际提升 1.2 倍。这时候集群虽然会正常运行，但**大概率集群中的其他爬虫请求速度不及预期，整体新鲜度会在数据周会复盘时发现整体下降**。组员的动作会拆东墙补西墙，甚至说导致找其他分组部署，浪费大量工时却没有整体性解决，都是内耗。
- 爬虫运行一段时间后，日志完全不滚动又不运行，我们俗称“假死”。这个大概率是 venom 一个积存已久的 bug：队列任务 data_valley 统计与实际异常导致。可能实际任务数是 0 了，但是 data_valley 的统计却是 1000（silk 设置的上限值），由于 silk 是根据 datavalley 的统计下发任务，因此不会再下发了，**手工点击 venom 系统的 reset inqueue count 即可解决**。(20240918 当 mongoDB 和 redis 单一集群未设置成同一实例时会出现)
- 日志中大量报错 599 或者 No Proxy，因为我们强制要求使用代理采集，**代理异常等于服务不可用**。特别的，如果有少部分（<10%）599 属于正常情况，短效代理会有过期的情况产生。
- 日志中大量报破解接口异常错误，一般来说是**破解源改版**，还有可能因为智海的破解服务尚未做到责任到线（20240905），甚至难以定位调用者 IP，因此当**智海的破解服务资源到瓶颈时，破解成功率一般会以指数状下滑（瞬间从 90% 到 10% 甚至 0%,我称之为死亡螺旋），而非线性下滑**。

- 日志中大量报【拿不到帐号】或者请求失败，大概率是登陆改版或者 carnage 服务对帐号的登陆/更新速度已经跟不上爬虫的采集速度了，**要增加 carnage 线程实例数，这个往往会被忽略**，因为组员仅能修改 venom 的采集速度和 carnage 的代码，但是 carnage 实例个数是无法修改的。

对于以上问题，TL 对应的高效处理方法是

- 节点失联的技术表象原因是
  - **85% 的问题是因为 mongoDB 内存爆炸而挂掉了**
  - 10% 的问题是因为 redis 内存爆炸而挂掉了
  - 5% 的问题是因为磁盘爆满而挂掉了
  - 对于阿里云服务器，服务器较为健壮，一般为 mongoDB 进程被系统自动 kill 掉，对于火山云服务器，服务器较为脆弱，一般会反馈为服务器无法在 jumpserver 连通，需要找到大佬重启。
- 针对一个失联节点，一般按以下步骤依次进行修复
  - 第一步：质疑集群 venom 的各实例的比例协调不合理。经过我最佳实践，SchedulerMain 应为每个节点 3 个进程、Processor 为 8 个进程、ResultHander 为 8 个进程，剩余 fetcher/carnage/mongoDB/redis 在集群稳定运行(队列不为 0)时，应吃满大约 80% 的内存为佳。实际的现有服务器中，有大量 Processor/ResultHander 在 15+ 的情况出现**(20240910 已经全部修复)**，根据我的反复测试，总体效率较低。除非特别典型的队列任务堆积，不建议修改我的推荐进程数。

(集群查询情况范例：http://120.76.73.216:4096/group/vg_qcs_mft_hd)

- 第二步：排查 mongoDB 问题，这往往是爬虫开发人员有以下问题是导致内存爆炸：
  - 被客诉或者产品施压，**近期有 30% 以上的提速的爬虫，尤其是还要使用 carnage 的概率更高**。因为更高的提速会使得登陆态频繁掉线，引起**集群资源产生死亡螺旋**。
  - 查询 mongoDB 内部数据有大量失败任务堵塞，短期的堵塞往往不致命，可能是爬虫改版或者失效超过一周导致，且失败任务未 Drop 抛异常所致，一般**堵塞任务常超过 20w+**。
  - 任务优先级设置不合理，裂变出的任务优先级比原任务相同或更低。平常还好，一旦名单压力过大时就会快速膨胀。所以可能**上线初期没有问题，后期突然出现频繁失联的问题，该问题比较隐晦难以排查**。
  - 爬虫裂变任务无法收敛，这一般是无限重试/回调循环等导致。常发生于**调用智海服务的完全的新爬虫上线时**，因为调用智海服务后爬虫的整体链路往往大于 10 个 CallBack，很容易循环回调，合代码时要特别注意。
- 第三步：排查 Redis 问题，Redis 问题一般比较简单：
  - 有爬虫开发人员违规一次性手工往 redis 队列推送大量的名单导致，现在大家都很注意了，很少有这样的问题发生了
  - Redis 部署所在的服务器 venom 进程开太多挤占了。因为 redis 的内存跟**存储总量影响很大，往往在压力小的时候看不出来，不好排查，比较容易死亡螺旋**。这边不推荐购买 redis 实例代替本地部署，因为 venom 的 redis 调用常为一进一出，存储总量不大但流量非常大。火山采买的 redis 流量有限制，不能满足 venom 这种吞吐量的需求。
- 第四步：排查磁盘问题，磁盘问题一般比较简单，但也有一定技巧：

- 大概率是 log 太多的问题。首先可以修改 supervisor 日志保存个数。
- 如果最近新增爬虫不多，那么可能是大量的报错产生日志大小突变，往往是瑞阳或者智海的服务出现了问题。**我们的组员负责的模块很多，饱和度很高，往往不能及时的发现问题**。可以看下就近的爬虫日志，这样就可以在数据周会之前发现问题。
- 还有一种可能就是 venom 的解析上传出现了问题，如果解析上传出现了问题，不能轻易的删除爬虫数据（这是数据不是日志，会丢数据的，要警觉），要跟瑞阳一起解决上传的问题。数据上传是异步的 crontab 里的配置，如果 root 里没有请查看 spider 的权限。有些在 root 有些在 spider，跟部署者的习惯有关（如果想学习如何完整的部署一个 venom+carnage 主从服务，可以结合我的【Venom 部署指北】，请教李阳解决，框架是 python2 已经很旧了，且小细节真的特别的多，一定要多沟通）

事务 2：代码合并与点赞

- Moonbrook：
  - 统计系统是最多的合并需求，也是组员代码量的主要来源。往往不用细看，由于 moonbrook 不可以本地调试，一般来说都需要 2-3 次提交才能正确跑上。合并代码后，组员可以自行启动、查看运行进展、下载 5Mb 或者 50Mb 以内的数据，对于大于的情况（99% 的导出需求都大于），请查阅 Moonbrook 数据导出章节
  - 我们要尽可能鼓励使用 Hue 和 data_valley 内部库点查来进行查询。一是速度大约是 MoonBrook 的 100 倍，另一个是 50% 的产品经理自己就可以做到查询，节约研发组员的时间。
- venom_script：
  - 这是爬虫代码的合并。主要是需要组员（尤其是新同学）给出一个正确采集的 demo 和截图。另外就是简单查看以下内容：
    - 是否正确的重试
    - 是否正确的使用了优先级
    - 关键处（**尤其是智海破解接口调用处**）是否有日志?日志是否过多?
    - meta 是否存了过大的对象（集群调度器很容易崩）
- tungee_parser：
  - 这是解析代码的合并。主要是需要组员正确的完成解析的开发，对于首次的开发源，可能还需要正确的撰写校验库。
    - 对于**非必填字段是否无意漏采**（非常常见，不影响上线，很难出工单，但影响产品力），理论上解析结果都需要产品验收，直接把解析的 json 发给产品即可。
    - 如果这是一个采集量极大的爬虫，初中级组员未必能够正确的理解解析开销，对于大的 dom 对象的反复加载，正则解析器的反复实例化可能要注意解析性能开销。爬虫量较小的可以忽略
- silk
  - 一个季度只合过 1 次，很少合并相关代码，我没什么经验。
- pytrans

- 这个代码一般 TL 只有点赞的权限并没有合并的权限，但是一般设计内部库表结构的增加和修改会涉及。**建表和修改表一定要跟大佬对好方案再做，再提 MR，不然 90% 会被打回合并**（可能是我菜？）。

事务 3：Moonbrook 数据导出

- 数据安全是公司的重点，即是老生常提，又有前车之鉴。
  - 首先本文不提供任何 Moonbrook 的导出提效建议，因为 100% 我觉得都涉及违规，但是这个导出的确是很繁重的工作，那就各位 TL 八仙过海各显神通吧～
  - 对于 50MB 以下的文件，组员会发授权链接，点击授权链接组员可自行下载
  - 对于可以直接导入名单服务器的，建议是 TL 直接在 **jumpserver–文件管理器上拖动。不要下载到本地**
  - 对于超大名单文件(>1GB)的，建议是让大佬操作内部 scp 传输，速度快可达到 100MB/s，不要使用拖动的办法。

事务 4：silk 配置修改和爬虫责任人/条线修改

- 这两个都是 data_valley 的功能，但是只有 TL 有权限，需要瑞阳添加 data_valley 权限。
  - silk 配置直接按组员发过来的填写即可，新增的话会报一个错误（落下一个红叉），这个错误是插入 mongoDB 成功后插入 ES 失败的错误，**不用管刷新就可以了**，是个老 bug 了
  - 修改爬虫责任人只起到标记作用，但是**修改所在模块（比如从联系方式–>制造业）是指定了解析和入库的服务器的，一定要正确修改，否则会堵塞别的组的解析或者入库队列**

事务 5：禅道相关

- 略

3、爬虫技术方案设计和落地指北（周会在这里）

这里都说一些（行业版）我经历的比较务实的实际情况啦，就不再重提什么规范了

与产品沟通阶段

- 确认负责人的组员是谁，确定现在工作与组员现有工作的优先级。如果是与 PD 沟通往往比较效率，如果是 PD 手下的产品经理，还是要 PD 拿决定。
- 与产品经理或 PD 确认以下工作：

- 数据在哪个版本的哪里上线？（**主要是为了确定其用途和重要性，佐证下面的所有事情**）
- 源列表
- 采集入口和字段
- 质控方法（我的习惯要先设计质控指标，**指标为了决定爬虫采集策略**），其中行业版往往会确定一个基础数据要达到的量级底线
- 采集策略（分别确定获新爬虫和更新爬虫，获新要有策略，更新要有新鲜度指标。根据网站**调研的更新速度**或者**新鲜度指标**预估总请求量压力，目的是评估**哪个采集集群适合放该爬虫**）
- 确认是否有字段关联映射工作（需求是应该做 解析映射/定时映射/实时映射/大数据映射，这里有一个坑，**产品总是会低估映射这件事的难度从而导致延期**，映射平均要 3–8 轮修改才能达到理想的映射准确率，我认为 TL 有义务给产品规划一种比较高效的验证方法从而快速迭代）
- 破解（TL 告诉产品，这件事不需破解/已有破解接口复用/需要破解/需要智海破解）
- 帐号（产品告诉 TL，是否需要帐号？帐号来源？帐号风控？**帐号风控会导致封号产品愿意接受成本**吗？决定是激进的测试策略还是缓慢的测试策略）
- 达到上线最基本的量级底线的需要的粗运行时间（**目的是粗略倒推上线时间，不要太离谱的近，否则就需要砍需求**）

任务执行阶段

- 与 PD 沟通好后，就需要安排任务：

- 开发时间安排（组员先自己估，组员自己提主要风险和主要工作瓶颈，主要矛盾是合理的我就接受）爬虫开发模糊性很大，不建议时间安排过于紧张，要给一定的思考时间，因为估时压的很紧，质量“较差”的爬虫也可以正常运行，但是会频繁“改版”，如果时间充裕，设计较好的兼容性，思考一些网站的特征，就不易“改版”，常见的爬虫开发估时框架：
  - 简单 Venom 2 工时，耗时工作是：分析网页结构是否有多种页面
  - 简单 Parser 2 工时，耗时工作是：写测试用例
  - 破解接口，**现在破解有全部归给智海和数据 TL 的“不良趋势”**。一般先不安排时间，由 TL 负责给出基本技术方向和估时，甚至代为完成，经过 TL 简单估计后，一般不超过 2 小时，或者做不了需要智海帮忙。耗时工作是：JS 逆向
  - 新映射开发 8–16 工时，一般是按 8 工时安排给组员，但**实际 TL 心里应该有 16+ 工时的计划和打算**
  - 统计和周报脚本的开发 1 工时，**这个还是蛮重要的，不说组员一般不会做**。
  - 对于有帐号的，一般是按 8 工时安排给组员，耗时工作主要是**风控策略探索**和 **carnage 开发**。**TL 需要想尽办法收集采集情报**（往往组员不会做这个事，比如各种问其他组，查询相关采集的网上资料，咨询同行等），最好在冷启动时就给出一个接近的方案，可以大大降低组员的开发时间。**实际 TL 心里应该有 16+ 工时，甚至无法实现的计划和打算，我一般会找一个供应商作为备选方案，供应商本身沟通的情报就极具价值，另外也可以兜底**
- 任务跟进
  - 除了公司常用的数据周会的形式，而我常用**查岗**的形式（每天 2 次沟通现有的困难，就面向过程，不会问目标和成果）。组员往往只会在周报上汇报目标和成果，而实际困难可能有更优的解法，主要是解决**方向和效率问题**。如果真的在解决一个值得解决的问题已经是最优路径了，那我的工作就是就尽量保证该组员不被打扰
- 数据周会
  - 我有一个目标，就是降低数据周会的时间，提高数据周会的效率。这个大家仁者见仁，智者见智。我只是分享除了**大家都知道的目标导向**外，我的一些会议上做法：
    - **不写 PPT，只写 EXCEL**。因为就展示形态来言，两者区别不大。但是 EXCEL 更容易让程序员实现统计自动化。同理，能用超链就不要复制粘贴。
    - 周会的每一个页展示的数据要尽可能的多，因为每个人角度不同，关心的事情不一样。讨论主要的，但是各自看各自想看的。
    - 我提议当场估时，而不要事后估时，这个仁者见仁。
    - 本次周会只讨论上周的工作和本周要干的事情，“短视”一些，不发散，不超过 7 天以后的未来，不搞“脑爆”，不聊闲话。
  - 总之：会议上会聊什么，聊的内容，聊的结果，基于日常高频有效的管理工作，应都在 100%TL 的意料之中。数据周会只是一个“必要形式”，只有三个功能，解答产品疑惑、解答领导疑惑、让组员有一点仪式感和成就感

维稳阶段

- 产品的数据指标不全等于技术维稳指标。产品的数据指标有的时候获取成本较高（比如完整度需要人工抽验）。我认为**数据维稳指标应该是简单易得的**（入库量、更新数、新鲜度、爬虫状态等）。**数据维稳指标要高频过，而产品的数据指标主要根据产品方向和客诉有需要的过**。如果不拆开，产品数据指标由于成本较高，就很难每周兼顾，就会导致数据模块失控
- 产品处理工单时经常缺失全局感。TL 要时刻定位“根本问题”，并且及时打断**产品高频越过 TL 的直接指挥**。否则组员的时间成本管理很快失控，无法拿到想要的目标。
- 要跟质控搞好关系.....不仅是 TL 还是组员。应该常找质控而不是让质控常找 TL。由于菊姐他们经常问题反馈而得不到解决，常常是一种比较消极的心灰意冷的状态。**主动询问质控我关心的数据维度的这个的动作**不仅会让质控感觉到重视，工作有意义价值，并且还能得到免费的质控劳动力，质控也能快速的检索问题，一举两得。
- 【仅限于我，不要学习！】当**仅限维稳问题**出现时，先寻找偷鸡办法，再常规解决：
    - 是否可以拿竞品数据入库?
    - 是否可以采集竞品名单代替全量更新?
    - 是否可以申请服务器资源?
    - 是否可以本地运行脚本或者爬虫，越过繁琐的上线流程?
    - 是否其他组已经做过或有成熟案例可以抄？甚至跨跃拓客甚至公司的范畴。
    - 是否可以申请平台架构的同学介入帮忙?
    - 是否可以采购?