

Threader

**Steam & Magic**

@xuwenhao

Entrepreneur, Evangelist in SAAS and productivity. Ex-Trilogian, Ex-MediaVer, Founder of bothub.ai/, Founder of bukito.cc/, Author of #深入浅出计算机组成原理

Jun. 06, 2020 1 min read

[+ Your Authors](#)[Archive](#)

今天看到 [@haoel](#) 说Google Docs的Post，联想昨天晚上喷团队里的同学把Jira用成了To-Do List，颇有点不吐不快。那就花点时间说说根据过去几年各种公司使用的SAAS/工具的体验。有些工具，真的是用过了就回不去的。

#SAAS #Productivity

基本上，以下这些SAAS/工具是用了之后就回不去的

1. G-Suite全家桶
2. Atlassian全家桶（Confluence+Jira+...）
3. Google Cloud里围绕BigQuery的整套系统
4. Cloudflare
5. NewRelic

#SAAS #Productivity

除了基本的办公协同之外，如果你真的重度使用“数字化系统”来管理公司的业务，你会发现G-Suite真的是太强大了。

第一个就是统一的权限管理，大部分SAAS系统都有Google登陆的支持。公司内部系统，也只支持Google登陆。这样所有同学入职离职，都只需要开一个G-Suite账号就好了，不用入职开账号就一天。

Threader



放在Google Shared Drives里面，意味任何一个同学入职，装个Google Drive客户端，所有他有权限的文档，分门别类就都有了。

第二个是极其强大的Google Form和Google Sheet。公司的业务迭代都是基于数据的，从上到下，都有大量不同粒度的数据和表单要收集。

轻度的OA流程，用Google Form就可以搞定。产品团队搞一个Google Form收集某个业务流程的申请表单。有人申请了，自动邮件提醒，同一个Form的汇总自动变成Google Sheet。

以前各个团队业务数据汇总，用的是极笨的办法，就是各个团队自己做个Excel，然后邮件发送给个产品经理或者业务助理，然后再人肉整合一遍。

而如果一个在线文档大家都能编辑，又有数据安全性的疑虑。

而通过App Script，直接就能生成Google Form。这样，一个月通过Google Form收集一次就好了。

如果不是简单的表单，而是比较多的数据表格，Google Sheet支持从一个Sheet引用另外一个Sheet，直接用IMPORTRANGE函数就可以。

这样就能让各个业务单元维护好自己的Google Sheet，然后汇总完全是自动进行的。

真还是基于业务团队自己搞定，如果研发团队有点资源，直接写个App Script脚本就更方便了。

而且Google Sheet最新支持了到Cell级别的Edit History，也支持到Column级别的权限管理。

这个意味着在更大的团队协作下才是可用的，不用有大量重复的表格。

很多年前大家都提“办公自动化”，这个才叫办公自动化，不是有个Word，可以协同编辑就叫办公自动化的。

也不是干个什么都要再单独做个OA系统。

Threader



在组织和工具层面都已经做好了。他自己只需要专心思考和解决自己的核心业务就好了。把80%的时间花在做事情上，而不是把80%的时间花在协同上。

Google Sheet的另外一个巨大的好处，就是可以直接通过Data Connector连BigQuery，可以引用Data Studio作为Dashboard，也可以作为Data Studio的数据源。

这样，连App Script都不需要，也不用专门做可视化开发，就可以直接让业务团队在Google Sheet里面看到乃至筛选自己需要的数据了。

另外，最近Google Hangout还在便宜的版本里面也支持了直接把Video Conference录下来。

这样，内部的分享培训，仅可以有PPT，还可以直接有视频，特别是如果团队在多地。

对应视频放到Shared Drives里，新同学入职可以直接看视频+PPT，不需要所有事情反复重复讲。

商业版本的G-Suite是无限存储空间的

用G-Suite和GCP是相辅相成的。说实话，GCP在产品丰富度和资料丰富度上，比不上AWS。但是GCP有两个优点，所以其实当时创业的时候就直接选了GCP了。

一个很简单，就是便宜。第二个，就是围绕GCP搭建整个数据系统的成本比较低，也就是GA+Firebase+Stackdriver+BigQuery+DataStudio这一整套。

这其中最重要的就是BigQuery了。BigQuery是目前所有云服务中，唯一一个可以最低从0成本起步，又可以无限扩展的云数据服务，价格还特别便宜。

现在大家的业务都是高度基于数据来进行迭代的，所以创业公司起步就要开始有数据系统，收集数据，分析，可视化。

但是我们创业公司又很穷，上手就RedShift或托管的Hadoop/Spark一个月几百刀用不起，而且还要搭各种Data Pipeline。

如果一开始用单机MySQL，或者日志落地写程序，一方面未来要废弃，另外一方面也要自己维护一套脚本。

Threader



一开始先用GA/Firebase作为数据监测系统，只要做好基础的打点，就有了一套可以分析的数据

Firebase的原始日志，会自动进入BigQuery。你只需要会写SQL就能分析数据，满足GA的图形化界面里面满足不了的数据

#BigQuery

如果分析的数据需要变成自动更新，变成系统报表。你可以通过Google Sheet的Data Connector来做。

如果需要可视化，可以直接用Data Studio，连买Tableau的钱都省了。

然后你自己的各种应用系统的日志，都通过Stackdriver记录，然后配置也落地到BigQuery。

无论是在GCP的console搜索日志，还是基于这些日志出各种报表，告警统计，都是在同一个框架下。

然后，当你发现GA/Firebase的日志不能满足你需求了，你可以把打点，再打一份到你自己的域名下。

然后通过 Cloud Function 写到 BigQuery 里面去。

除此之外，你可能还有很多外部的SAAS数据，比如投放的Facebook/Google的广告，业务系统的MySQL表，等等。

这些，你都可以通过Stitch，也自动同步到BigQuery里面去。如果你的数据量比较大，觉得Stitch太贵，那可以直接用Stitch开源出来的[Singer.io](https://singer.io)自己来做这个同步。

而这样一整套以BigQuery + Data Studio为核心的数据系统，基本上是全免费的，只需要按照使用的Quota来收费。

这个意味着你的业务的数据系统的成本，完全是按照业务系统的体量来的，如果发现没有

Threader



而且整个数据系统的搭建基本没有开发成本。

只需要会写SQL, 然后在图形化的界面上操作配置, 你就得到了一个可以无限扩展的BI系统。

而且可视化的Dashboard是可以直接交互、筛选。还能导入到Google Docs/Sheets/Slides里面拿出去演示。

说完了GCP的BigQuery, 讲讲Atlassian全家桶。目前我们用了 Confluence + Jira + Bitbucket。看过 Opsgenie 和 Service Desk, 不过后两个目前还没舍得花钱用。因为Ops和客户服务端的压力还不算大, 而且国内的客户还是喜欢微信群。

#Confluence #Jira

Confluence和Jira的好处, 其实就是足够好用丰富。你能想到想要有的东西, 都是随手可得。而且各个系统之间是完全打通的。

你可以在一个Confluence的Page里面, 鼠标拖拽一段文字, 然后右键就能立刻建一个Jira的Ticket出来

也可以随时在Jira里面用 /link 快速索引一篇Confluence的文档。

而从文档的系统、编辑, Confluence可以直接嵌入draw.io这样的第三方应用。自己也自带了各种高级组件, 从纯粹展示的Info Panel, 到和Jira Issue集成的Voting Issue, 乃至基于Sprint统计分析的Chart。

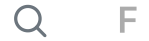
基本上, 围绕项目开发管理能有的需求, 全部一站式解决。

而且Confluence提供了海量的模版, 你要写一个内部的Guide, 直接用How-To Article的模版, 不需要考虑文档结构, 填好内容就好了。

对于很多没有足够多专业训练的同学, 从这些文档模版里都能学会很多事情具体怎么做。

Jira也是一样的, 可以把Sprint和Backlog的界面和流程, 完全配置成你想要的项目管理流程, 并且通过图表追踪每个Backlog大家花的时间, 精力分配, 对于自己思考回顾改进空间相当好用。

Threader



Issue类型都记录成Sub-Task，导致你想要做产品项目的研发效率的分析筛选都很困难。

上面说的，主要还是协同工具，包括整个基于BigQuery的数据体系，也是产研和业务团队都要用的。

而在研发团队内部，除了GCP之外，有两个SAAS我觉得是对大家都有必要，一个是用来做APM监测的NewRelic，另外一个就是CloudFlare。

Newrelic不便宜，但是很好用，关键是对代码没有侵入性。

对于创业公司来说，一开始没有必要从高性能的角度去思考代码怎么写，像我们都是先用Laravel快速实现功能。能用MySQL就用MySQL，但是性能压力什么时候上来，其实你完全不知道不是么。

真的压力上来了，或者遇到性能问题了，有New Relic基本上就是可以快速定位到哪个SQL或者是哪一行代码。

对团队成员的要求也没那么高，你不需要熟悉MySQL也好，应用层也好的各种细的Metrics，90%的情况下，看一下New Relic的Transaction数据，优化一下SQL或者代码，就又能支持很长一段时间了。

而且往后大概率会在架构上做调整，比如把一些写入到MySQL的数据，可能写到BigQuery里，或者特别的热点模块，加一层Cloud Function或者PubSub这样的队列。

而不是在代码层面优化，有了New Relic，你对工程师的系统底层能力的要求变低了，大部分情况下也不需要专门安排压力测试了，而是靠监控来搞定一切

Cloudflare则是快速解决前端的性能问题，20美金，你就有全球无限流量的CDN可以用，而且系统不需要做任何修改，不需要考虑内容是静态还是动态。

包括前端资源的minify，图片自动压缩，动态路由这些问题，都不需要你自己考虑，也不用担心DDOS了，可以说必备

而Argo和Workers在很多应用场景下也很好用。

Threader



业务上，减少去处理各类旁枝末节的细节上。

比如你做一个SAAS，性能问题并不是成功关键，但是你不想要因为性能问题拖累业务，那就用New Relic。

尽可能有明确固定的流程、工具，把一切事物性的工作做了。

让新入职的同学，除了自己核心的工作要思考要研究之外，不要太操心各种杂事。

通过系统和流程，一方面简化工作，一方面可以通过客观的数据评估结果。

这其实是很资本主义的，你希望你的成本和开支沉淀在公司这个组织上，而不是给人教学费上。

最近下一步做的是，进一步细化标准化一切流程性的内容。从创建一个Jira Issue的模版，到Commit消息的格式，乃至给业务团队算绩效，全都基于系统和数据，不需要你发挥聪明才智和主管能动性。

其实云服务发展到今天，很多事情只要做核心工作就好了。

如果你今天要开发一个SAAS，你可以用 Laravel Nova 写你的核心功能，支付之类可以直接用 Paddle，然后各种Dashboard什么的就都有了。

在你做到能够赚钱之前，你不需要考虑性能啊，部署啊，服务器成本啊等等，基本都是0，都有现成工具可以用。

好了，这个Thread也差不多了，欢迎大家有兴趣的可以一起多交流，如果对做海外SAAS，出境电商有兴趣的产研同学也可以来我们办公室聊聊，持续招人中。

You can follow [@xuwenhao](#).

Share this thread

Bookmark



Threader



Tip: mention @threader_app on a Twitter thread with the keyword “compile” to get a link to it.

Since you're here...

... we're asking users like you to make a contribution to support this independent project. In these uncertain times, access to information is vital. Threader gets 1,000,000+ visits a month and our iOS Twitter client was featured as an App of the Day by Apple. Your financial support will help two developers to keep working on this app. Everyone's contribution, big or small, is so valuable. Support Threader by becoming premium or by donating on PayPal. Thank you.