

从零开始，打造千万级用户社区架构

金伟梅

辣妈帮的简短介绍

- 2013年5月，160W，活跃用户30W，日请求数5700W。
- 2014年5月，1800W，活跃用户100W，日请求数1.4亿。
- 2015年1月，5400W，活跃用户420W，日请求数2.3亿。

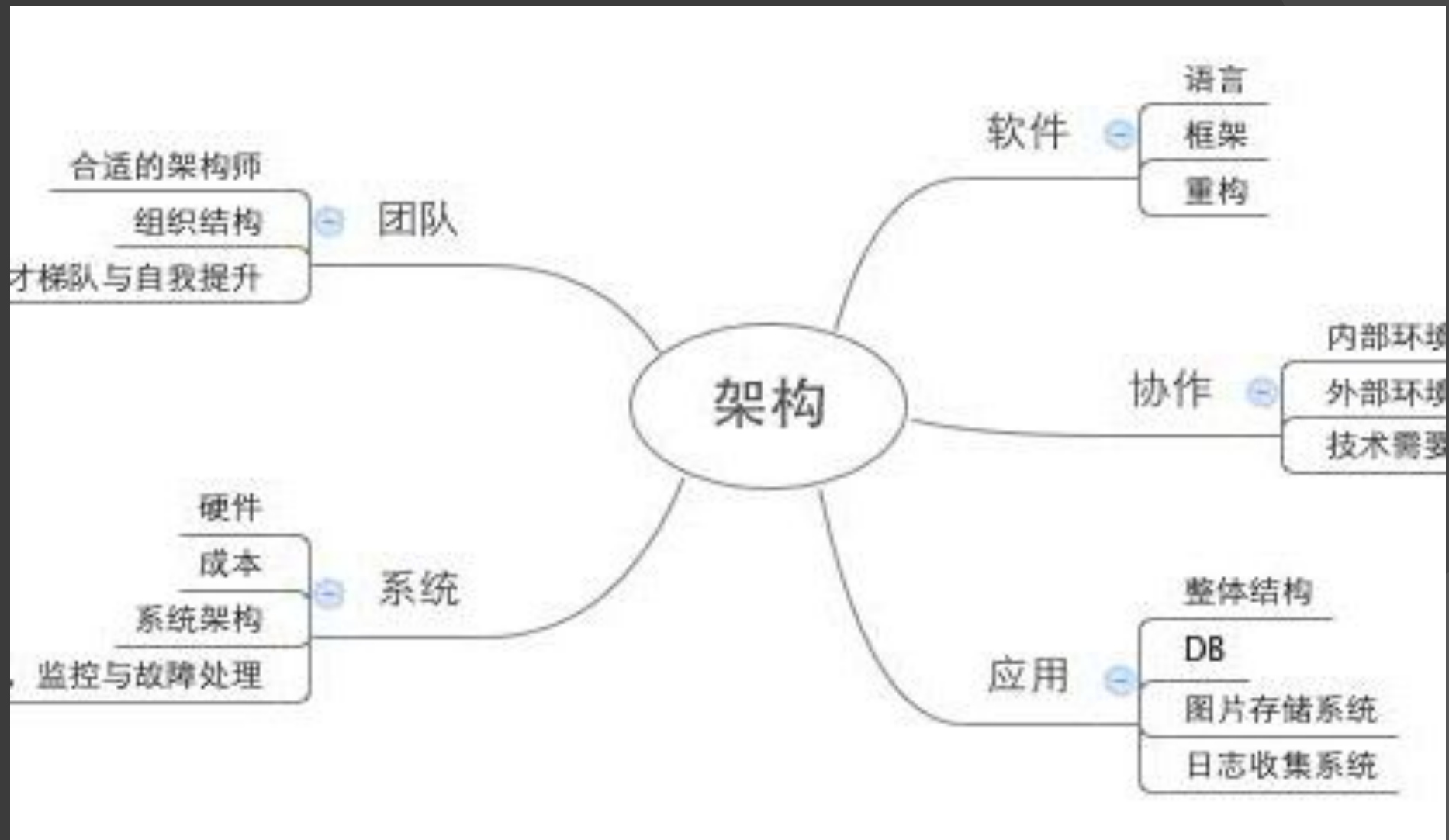
一切不是从0开始，而是从负数开始的

- 对于初期的创业团队而言，最大的成本是人力成本和时间成本。
- 人力成本高昂，就会选择缺乏经验的程序员，省掉测试人员，整个团队也会缺乏技术架构的意识和经验。
- 产出的大部分项目，从立项到产品上线，一般只有2个月左右的开发时间上线，无法保证产品质量。
- 很大一部分项目会选择二次开发和外包，这会留下巨大的隐患，但更坏的结果是，很多独立开发的项目，会比二次开发更差。
- 一旦产品验证成功，功能将会不断快速迭代，先天不足的情况会越来越糟糕。
- 伴随推广的持续投入，用户疯狂增长，之前在技术研发环节欠下的债会滚雪球增加，最终开始系统响应变慢，出现频频崩溃。
- 这个时候，创业团队内部会陷入互相指责的困境，因为技术上的短板，巨大的压力，可能导致程序员扛不住压力出走，让一切雪上加霜，这个时候，大家都想到了一个词，重构。

重构是推倒重来？还是改造？

- 在不断提到重构和架构这两个词汇的时候，意味着业务在快速扩大，产品功能需要更加快速迭代，同时性能严重影响了用户体验。
- 通过事实，团队终于认识到了技术拥有一个良好架构的重要性。
- 是推倒一切重来，还是改造？
- 但是，时间不等人，每天的用户数还在以万计的递增。
- 不能否定之前团队的努力，不能推倒一切重来。
- 重构应该是从局部改造开始，完成全局重构。
- 改造比从新做一个东西难很多。

架构从来不是单一的技术问题



架构之团队：合适的架构师或Leader

- 人是一个良好技术架构的基础，合适的架构师或者具备架构经验的工程师，是一个团队的基础。
- 低调，务实，勇于承担，敢于否决。
- 具备各个基础岗位的实际工作经验，各种中小公司的社会工作经验，跳槽频繁可以是优点。
- 情商高，保持谦逊的心。
- 工作的目的是为了降低整个技术团队的工作难度，而不是相反。
- 少空谈，多做事，以身作则，团队最需要的是给他们在当前阶段带来切实帮助的人，只指挥不动手很难获得团队认可。
- 快速响应，保持手机7*24小时畅通。
- 如果BOSS有喜欢陪着加班，或者在处理突发故障喜欢站在旁边指挥的习惯，请他回家或者回办公室。
- 发自内心的喜欢这个产品，利用尽可能多的时间体验产品。

架构之团队：组织结构

- 技术是双重管理，技术管理和行政管理前期应该集中，保证团队执行能快速推进。
- 注意团队中的新老交替与融合，不要轻易否定之前团队的努力。
- 顶住外部压力，安排专人对接运营部门的需求，将其他程序员隔离在相对安静的环境。
- 研发过程中，强调项目管理意识和测试意识，不失敏捷。

架构之团队：人才梯队与自我培养

- 完成一个技术架构或重构的实践，一人之力不行，需要整个团队支撑，整个架构的后续维护，必须有人自始至终跟上。
- 别人能做80分的事情，你不要做，放手让他做，最后一起去补全那20分的事情。
- 整个团队务必在短时间有较大的能力提升，包括技术和其他方面，才让团队成员觉得累有所值，拒绝外界薪资猛增的诱惑，也让团队的后续发展打牢基础。
- 麻雀虽小五脏俱全，分工要明确。
- 将程序员变成主程，将主程变成leader，才不负光阴。

辣妈帮的技术团队发展

- 至2013年初：3人。
- 2014年初：7人。
- 2014年下半年：30人。
- 现在：70人，目标80-90人。
- 技术团队的扩充总是滞后于需求的发展，如果做到快速扩充，帮助成员转换角色，与应对未知的明天。

架构之软件：语言

- PHP
- 简单，门槛低，能快速上手，快速发布。
- 试验，从0培养一个PHPer，只需要3个月。
- 语言只是一种工具，程序员应该将它当成工具，而不是信仰。
- 什么都能做的语言，那就是什么都做不好的语言。
- 多语言的开发和维护对于创业团队来说，是致命的。

架构之软件： 框架

- 怎么选择一个好的开发框架？
- 具备普适性，强的代码规范，便捷快速开发，追求性能与开发效率的平衡。
- 我们自己研发的框架，之所以重复造轮子的意义。
- 提供强的编码规范约束。
- 组件和模块复用。
- 一个入口，适应各个开发场景。
- 是系统分布式集群，和团队分工协作的基础。

架构之软件：重构

- 重构不是一锤子的买卖，而是一个一直持续的过程。
- 推倒重来的想法，需要控制在局部，进而全局改造。
- 让团队成员具有重构的意识，每周一次代码质量审核。

架构之系统：硬件

- 硬件的采购配置需要统一，每一个应用场景尽量采取统一的硬件配置，考虑与业务需求，2-3年的使用预期。
- DB, CACHE, WEB, FS
- 考虑冗余和备用机器。

架构之系统：成本

- 硬件和网络带宽是技术团队除了人力开销外，最大的成本。
- 尽量从各方面节省成本。
- 理解什么叫战略浪费，冗余和备用机器，及多机房备份，线路备份，都是有必要的。

架构之系统：系统架构

- 系统架构主要考虑几点，性能，扩展性，规范性。
- 建立多种测试环境，本地环境，alpha，beta，stable。
- 让整个集群在程序员眼中透明不可见。

架构之系统：数据，监控与故障处理

- 建立完整的日志收集处理系统。
- 监控建立在日志系统之上
- 故障发生是绝对的，第一时间响应和善后才是需要考虑的。
- 热备与冗余机器在各个应用环境下的选择。

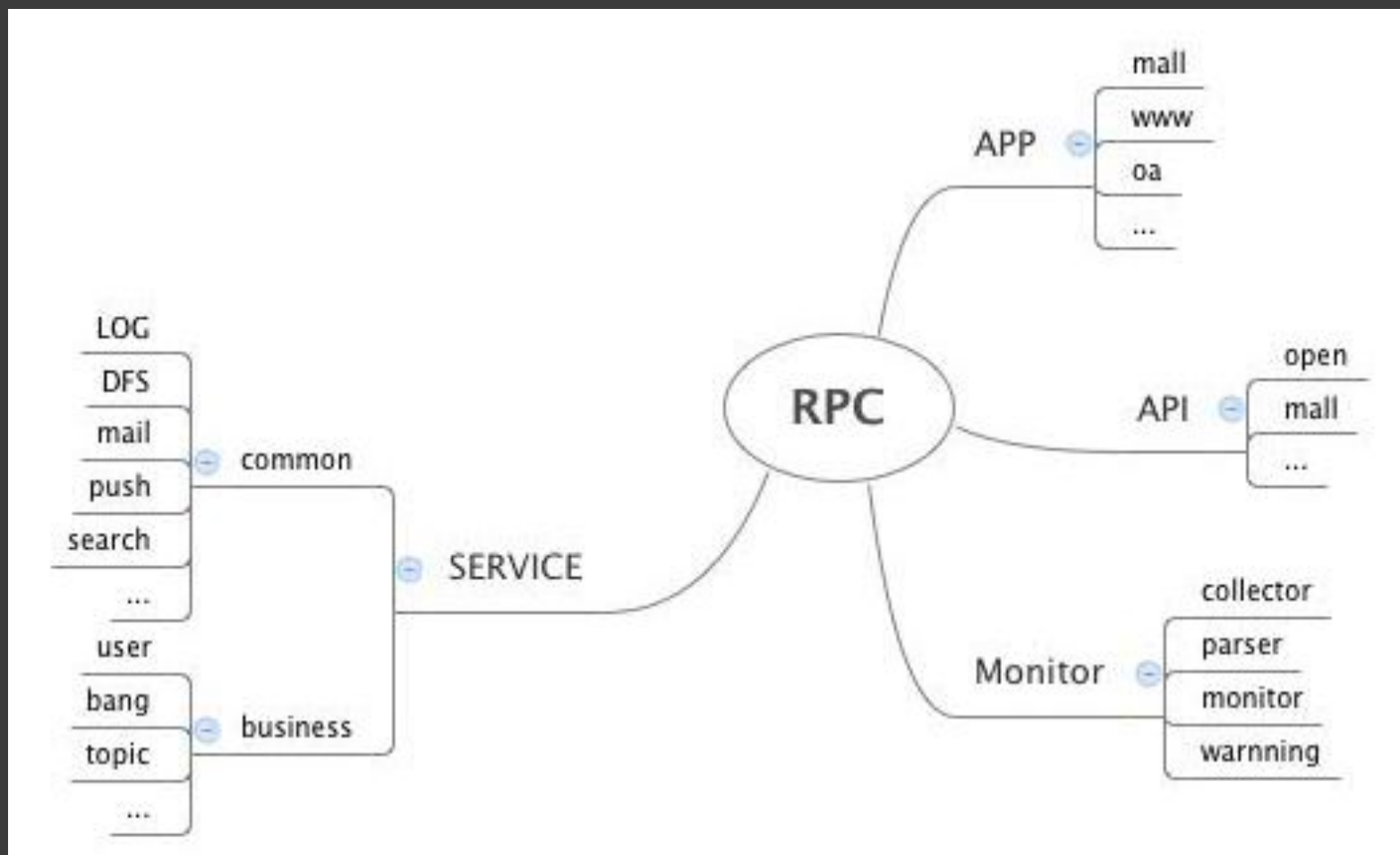
架构之协作：内部环境

- 内部需要充分的技术交流，保证技术架构的意图能够得到理解和执行。
- 先定规则，规则不好可以改进，在改进之前，必须无条件遵守。
- 每周四下午，每个程序员审核一周来所写代码，Leader审核每个成员所写代码，及时纠正。

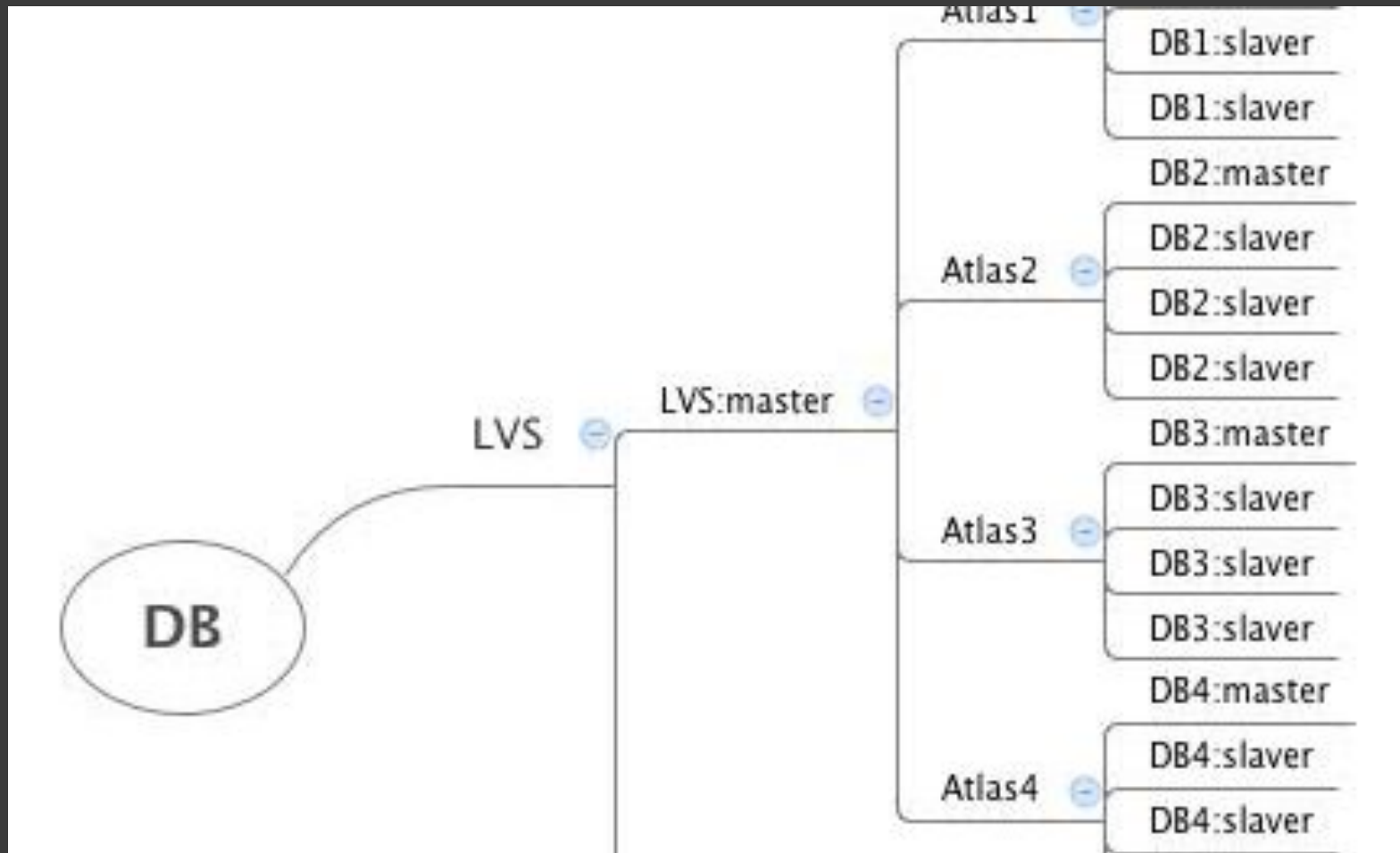
架构之协作：外部环境

- 技术团队支撑所有部门的技术活动，是巨大的压力，也是资本。
- 和而不同，顺势而为。
- 技术架构不能脱离真正的实际情况。

架构之应用：整体架构



架构之应用：DB

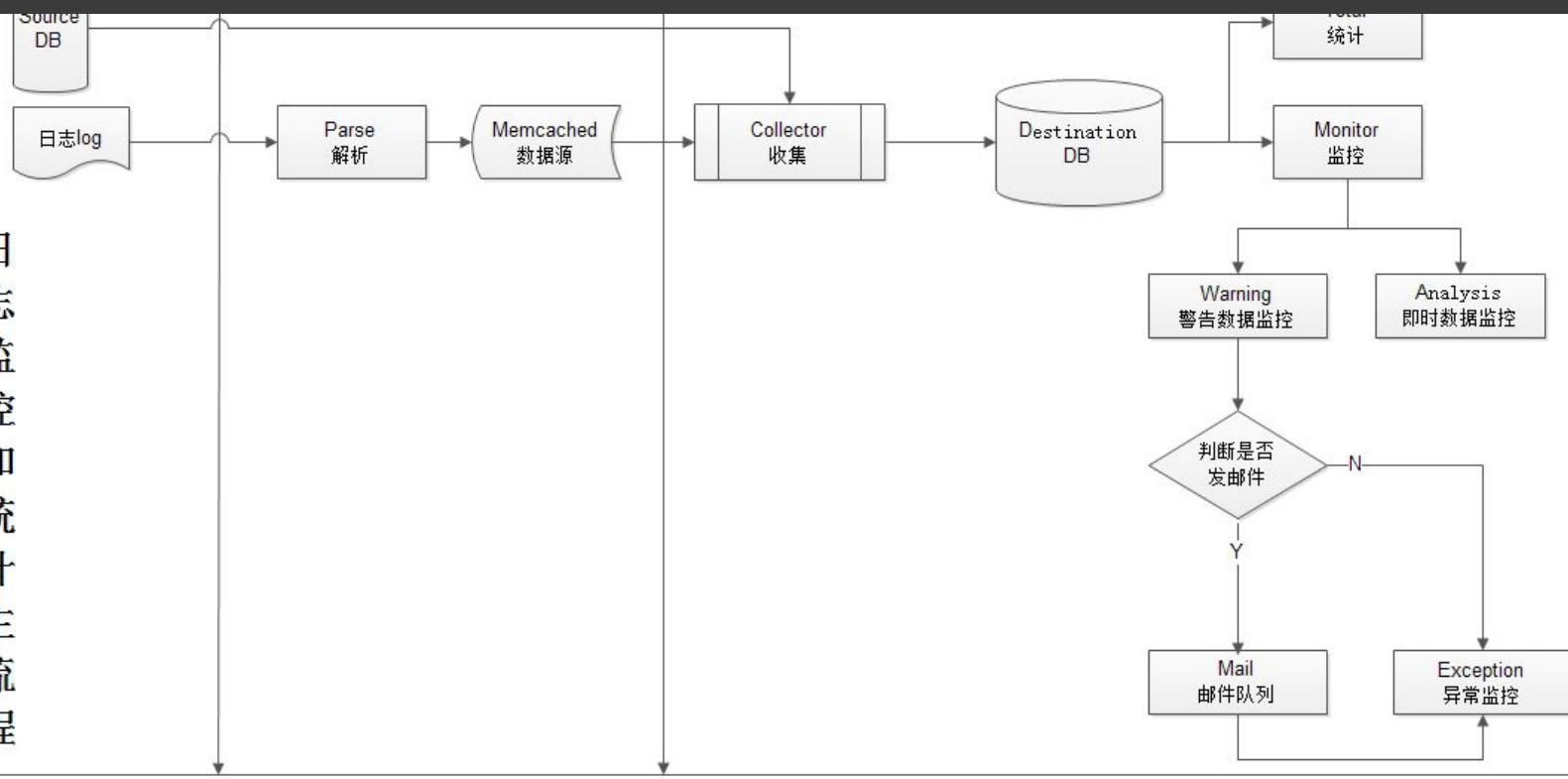


架构之应用：图片存储系统

- 底层存储是fastdfs，多地同步。
- Nginx+lua(openresty),处理上传，图片裁剪压缩，缓存策略等，

架构之应用：日志收集系统

日志监控和统计主流程



数据收集流程



总结：化繁为简

- 一个服务器品牌(IBM+DELL各配置)->DELL (WEB,DB,CACHE,FS)
- 一个操作系统, (CENTOS各版本, Arch LINUX,ubuntu->CentOS 64位)
- 一个程序语言, PHP (WEB+CMD+Service+RPC)
- 一个图片存储系统, (磁盘+NFS+rsync) ->Nginx+lua (openresty) +fastdfs+CDN
- 一个NOSQL, Redis = (Memcache+mongoDB+Queue+Coaster+LIST)
- 一个DB, MYSQL
- 一个语言编码规范。
- 一个框架(开源系统+CI+原生PHP+thinkphp)->tiny。
- 一个系统架构。

一些比较好玩的开源技术

- DB组件, Atlas, Mycat
- Cache, ssdb
- WEB, openresty(图片存储, SQL代理, 实时聊天)
- Swoole RPC通讯。
- 对于开源软件, 没有跟踪2年左右, 不要轻易尝试。

感谢！

- 谢谢大家，谢谢活动的组织者。
- 顺便招人（10-25K招募PHPer，数据分析）。