# SSP综合设计文档

摘要：

主要讲一下Adx的技术、业务实现。

1. 主要功能

接收到SSP发来的请求，对请求做些处理，然后并发的请求各家DSP，超时时限为100ms，在时限之内响应并出价的DSP，将被放到一起进行竞价，Adx将胜出者的广告返回给SSP（同时也会对SSP进行出价），到SSP方参与竞价。

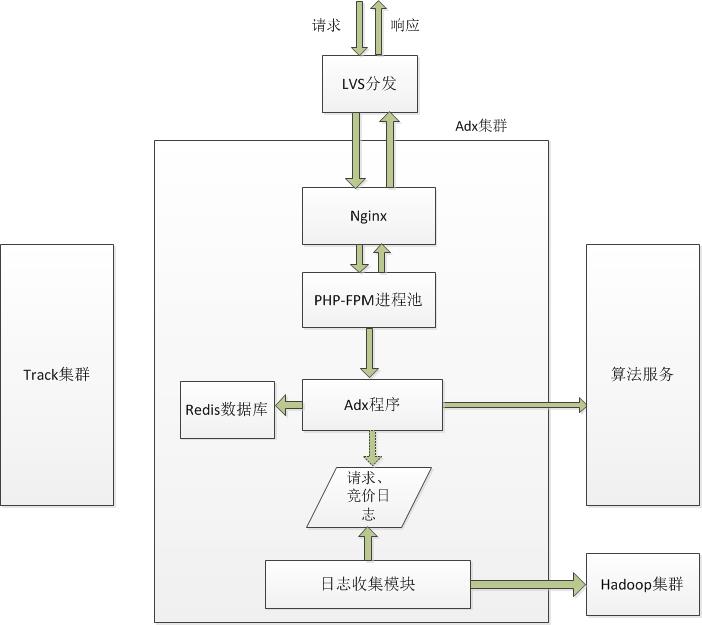
1. 技术架构
2. 接收和处理请求：Nginx+PHP-FPM，Nginx连PHP使用域套接字；开启Keep-Alive长连接。
3. 并发请求各家DSP：libcurl + multicurl，cURL 版本 >= libcurl/7.21.0 版本，毫秒级超时才生效。设置超时时间，可以提升并发数。

缺点：因PHP进程模型的原因，使用短链接请求DSP，导致系统临时端口占用过多，应该改成长连接keep-alive方式

1. 数据缓存：本机Redis，采用域套接字访问。
2. 请求、竞价日志：写本地文件（没有异步，本地写文件速度很快）。
3. 竞价策略：使用网络访问算法服务（内网）。

说明：

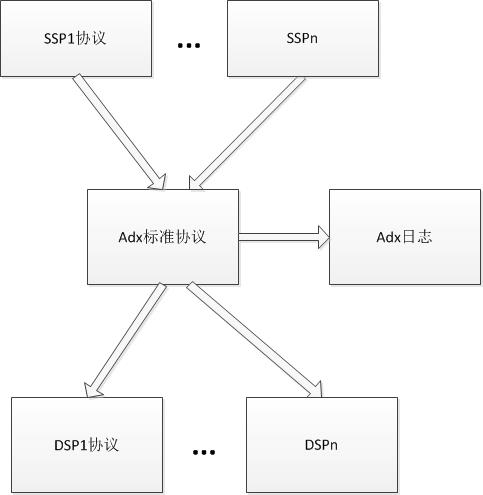
当前的架构（10台CentOS机器）支撑了我们5个亿的日流量，还没有到达瓶颈。控制好DSP的超时时间，是提高并发处理请求数的有效方法。



1. 业务架构
2. 内部交互使用统一的标准协议。
3. 将SSP的协议转换成Adx的标准协议。
4. 记录SSP请求日志。
5. 将Adx的标准协议转换成DSP的协议。
6. 对参与竞价的多家DSP，进行竞价处理，产生胜出者。
7. 记录DSP竞价日志（DSP在Adx中参与竞价）。
8. 记录二次竞价日志（Adx在SSP方参与竞价）。
9. 通知DSP竞价成功。
10. Track集群记录展示、点击日志。
11. SSP和DSP应该是可配置的，以类似插件的方式加载。

说明：

将各方SSP的协议转成Adx的标准协议，有助于统一处理，去算法竞价、记录请求、竞价日志。Adx内部使用标准协议，可以将核心业务抽离出来，做成独立模块，再去接第三方SSP、DSP时，主要的工作只需做协议转换即可，开发工作量就变得很少。



1. 数据回收
2. 跟Adx相关的日志（请求、竞价日志），Adx在Bid的过程中直接写入本机磁盘，然后通过本机上的日志任务，将日志压缩分割，再上传到Hadoop集群进行处理。
3. 用户行为日志（展示、点击日志），则发往另外的Track集群，有同样的日志处理服务。