专访阿里陶辉：大规模分布式系统、高性能服务器设计经验分享

发表于2014-06-27 16:25| 30655次阅读| 来源CSDN| 58 条评论| 作者魏伟

[云计算](http://www.csdn.net/tag/%E4%BA%91%E8%AE%A1%E7%AE%97/news)[Nginx](http://www.csdn.net/tag/nginx/news)[开源](http://www.csdn.net/tag/%E5%BC%80%E6%BA%90/news)

**摘要：**先后就职于在国内知名的互联网公司，目前在阿里云弹性计算部门做架构设计与核心模块代码的编写，主要负责云服务器管理系统和存储系统的优化。陶辉就大规模分布式系统、高性能服务器设计分享了自己的看法。

关注[陶辉](http://blog.csdn.net/russell_tao)很长时间，初次对陶辉的了解还是在我们CSDN的博客上，从2007年开始写博客，一直到现在，如果不是对技术的追求和热爱，以及热爱分享的精神，我想不是很多人能坚持下来，拥有多年大型互联网公司的从业经验，对linux下的高性能服务器开发、大规模分布式系统的设计有着丰富经验，对企业的Nginx\开发Nginx模块也有着独到的理解。

**免费订阅“CSDN云计算”微信公众号，实时掌握第一手云中消息！**

CSDN作为国内最专业的云计算服务平台，提供云计算、大数据、虚拟化、数据中心、OpenStack、CloudStack、Hadoop、Spark、机器学习、智能算法等相关云计算观点，云计算技术，云计算平台，云计算实践，云计算产业资讯等服务。

Nginx是一个高性能的 HTTP 和反向代理服务器， Nginx 是由俄罗斯人 Igor Sysoev 为 Rambler.ru 开发的，其将源代码以类BSD许可证的形式发布，因它的稳定性、丰富的功能集、示例配置文件和低系统资源的消耗而闻名。在谈到Nginx未来会取代Apache时，陶辉表示云与端的时代，端越来越多，云的性能就会越来越重要，服务器资源的效率在企业成本上将占据越发重要的地位，从这个角度来说，Apache的市场份额被Nginx取代的趋势是不会变的。这次，笔者有幸联系到陶辉，他就大型软件的开发，云计算等分享了自己的经验心得。

**陶辉**

**毕业于西安交通大学计算机科学与技术专业，先后工作于华为、腾讯、思科、阿里巴巴等公司。**

**对linux下的高性能服务器开发、大规模分布式系统的设计有着丰富经验。著有《深入理解Nginx：模块开发与架构解析》一书。**

**CSDN：给大家介绍一下您及目前从事的工作？**

**陶辉：**我目前在阿里云 弹性计算部门做架构设计与核心模块代码的编写，主要负责云服务器管理系统和存储系统的优化。大家在阿里云上购买的ECS服务器就是弹性计算部门的产品。

**CSDN：是什么缘由促使您写了一本关于Nginx的书籍？**

**陶辉：**大约2010年在思科工作时需要开发一个云文档系统，当时选用了Nginx作为web容器开发一个有较高性能诉求的文件上传下载服务。

可是发现很难从网上找到系统的资料可以指导Nginx模块的开发，对于Nginx的设计思路也是一头雾水，只好看它的源代码来倒推作者的初衷与模块设计思路。那个过程对于已有多年服务器开发经验的我来说也是痛苦的，于是就开始陆续写了一些Nginx模块开发的文章分享到CSDN上，希望能够帮助其他开发者快速解决问题。这时机械工业出版社的编辑lisa看到这些文章找到了我，希望我能写一本系统的、从开发者角度介绍Nginx的书。这本书诞生的缘由就在这里。

**CSDN：目前国内知名的互联网公司很多都在使用Nginx，您觉得企业在使用Nginx\开发Nginx模块的过程中需要注意什么？**

**陶辉：**仅当需要并发处理万级别或以上的TCP连接时，才应当考虑Nginx。

当官方Nginx无法满足项目需求，在开发你的个性化模块之前，先看一看大量的Nginx第三方模块里，有没有能够解决问题的Nginx模块，不要重复开发轮子，尤其Nginx轮子的开发难度还不低。Nginx.conf里可以玩的花样很多（这由每一个Nginx模块决定，如ngx\_lua这样的模块还可以在里面插入lua语法），或许一段几十行的配置就能完成复杂的功能。

如果确实没有满足需求的模块，那么，再看看能不能通过类似subrequest这样的机制将问题分解为多个子问题，其中多数子问题可以由现成的模块完成，或者通过proxy机制来与其他现成的组件通过tcp协议交互完成。组合这些子问题来构成解是个好习惯。

若真有必要编写Nginx模块，先要确保它只是解决一个非常纯粹、简单的子问题，不要耦合太多的需求。Nginx进程里是拒绝任何阻塞操作的，这是因为模块都运行在IO核心处理线程中的。任何一个边缘化的模块都可能因为自己小小的阻塞调用毁掉Nginx的高性能。所以，慎重的考虑模块中的每一个调用，确认它们不会导致进程进入sleep状态，确认它们不会在那里空转占用系统资源。好好使用Nginx定时器事件、共享内存，往往能解掉上述问题。

写模块时，使用好Nginx的变量机制，让自己的模块插上http框架的翅膀，根据框架解析出的变量来做灵活的处理。甚至，提供一些新的变量作为底层模块而给上层模块使用。

使用好nginx.conf，通过灵活的配置来提供丰富的功能。

debug级别的error.log日志非常详尽，仅有它就可以定位出很多你的模块bug，别忘了使用它必须在编译时加入--with-debug。

最后，如果模块可能对其他人有帮助，那么，分享它吧。

**CSDN：对于正在学习Nginx的同学有什么建议？对开源软件的学习有没有什么分享的？**

**陶辉：**其实《深入理解Nginx》这本书的目录，就是我推荐的学习路径。

首先，从最外面看Nginx是什么样的，了解进程模型、配置文件语法、基本功能等。

其次，从尝试编写最简单的http模块入手，渐渐地使用到Nginx Http框架的一些高级特性，了解Nginx的内存池、各数据结构的用法等；

再次，系统的了解Nginx框架，包括它如何启动、如何停止、如何升级、如何重载配置，多进程间如何负载均衡，http连接的建立、URL与包头的收取、解析、选用哪些http模块处理请求、如何向客户端回响应等。

这样学习Nginx，大家就可以清晰得了解异步事件框架，理解松耦合设计与web请求的处理方式。

学习其他开源软件也可以仿照这一过程。

**CSDN：我知道你在思科、腾讯等企业工作过，关于大规模分布式系统、高性能服务器设计上有没有什么经验、心得和大家分享一下？**

**陶辉：**使用高级语言、中间件来开发较大型软件时，一定先有一个评估标准：这样玩性能不是问题。这里的潜台词是把性能当做了基础货币。例如，使用python代替C进行程序开发所牺牲的性能C1，与其带来的其他好处C2相比，必须C1<C2。所以，性能其实是一个永恒的话题。下面零散的谈谈我对性能的理解。

单组件的性能提升上，算法最重要。特别是越前沿的技术、场景，通用算法的功效距离期望值就越远，开发者这时要能够正确的分析不同的算法在各种情况下的运行时间，基于你的数据特性设计个性化的算法以提升性能，PS，这里终于可以用到大学里学过的如概念论这样的数学技巧了。同时，细致的梳理业务，能够并行处理的绝对不要串行化，谨慎加锁，提高吞吐量。

对于组件依赖的其他系统，也需要深刻理解如何使用它，才能最大化硬件效率。例如操作系统，如果组件使用多线程去抢占有限的CPU资源，就必须评估进程间切换的代价，这往往是性能大杀器；了解不同的设备间的访问速度（如SSD硬盘、内存等），将快速设备放在慢速设备前作为缓存；使用TCP作为通讯协议时，既要了解理论也要了解实现，包括演变过程，在实践中才能高效的使用、改进它；减少内存等资源的频繁使用，考虑内存池及如何避免大块内存拷贝；提高缓存的命中率，如coding时应当考虑变量是否经常落到CPU CACHE中，及代码分支预测的命中率等等。

其实性能“高”也是相对的，需要从开发效率和运行效率上权衡。协程是一个很好的方向，通过创建协程栈来伪造线程开发环境提升开发效率，通过改变底层阻塞API的实现来提升吞吐量、运行效率。例如linux上的ucontext、nginx的lua模块，这里最大的问题还是阻塞API的协程化改造。

团队的技术积累、业务特点都对开发效率有不同的要求，架构上的scalability也是重要约束，能够通过水平扩展线性地提升性能时，就可以通过牺牲单组件性能来提升开发效率了。

分布式系统的ACP是一个权衡问题，适当的牺牲一致性是常见解决方案。scalability是一个重要属性，而这个属性会带来请求串行化的场景，常用zookeeper这类系统来提供可靠的锁服务。有了scalability常会导致系统引入缓存服务：组件的主存不可缓存了。缓存也有很多种成熟的解决方案，如memcached、redis等。

开发大型系统时组件间的高内聚松耦合很重要，否则代码很快就难以维护，有一种解耦方式比较受青睐：使用如rabbitmq等服务来提供异步消息订阅通知机制。

多个会落地的数据服务可能会引入“事务”，而分布式事务解决起来是比较头疼的，paxos两段式提交常常是首选，事务的回滚、清理、残留未完成事务的回滚等都是需要考虑的事项，可以借鉴关系数据库的undo、redo等事务解决方式。

**CSDN：能否谈谈Nginx目前还有哪些不足？还有哪些地方有待完善？**

**陶辉：**我们对Nginx的期望一直在提高：早期只把它用做静态web与反向代理，渐渐地希望它能够处理动态请求。这样，在Nginx进程内部增加功能就越来越重要。

怎样增加动态请求的处理呢？最方便的是使用一些抽象了常用动态功能的模块，这些模块以nginx.conf中的配置来定义web请求的动态处理流程。然而，很多时候这些模块模块只能处理大众化的需求，这样程序员们只好挽起袖子直接编写C代码了。

但是，nginx模块的开发门槛还挺高，需要开发者对于服务器的非阻塞调用、事件模型有较深的理解，而如果请求处理时需要有全局化视角时，麻烦的多进程通信又来了，开发者不能使用简单的堆分配对象，而要使用nginx\_slab管理内存。

因此，除了期待更多的开发者贡献出多样的抽象模块，目前nginx最应该完善的应当是二次开发的易用性--能够更方便、快速的开发出高性能的nginx模块。例如，nginx的框架可以考虑支持多线程模型，可以考虑支持ucontext协程方式，使开发时不用考虑API的异步回调，不用考虑锁的满足条件。

**CSDN：Nginx市场份额一直稳步提升，您觉得Nginx未来会取代Apache吗？**

**陶辉：**在云与端的时代，端越来越多，云的性能就会越来越重要；而互联网思维本就不高富帅，同时会服务所有草根用户，而用户体验也需要提升，所以，服务器资源的效率在企业成本上将占据越发重要的地位。从这个角度来说，Apache的市场份额被Nginx取代的趋势是不会变的。

**CSDN：能否谈谈对目前国内云计算市场有什么看法？有哪些趋势值得去关注？**

**陶辉：**云市场发展开始加速，云服务提供商将开始在国计民生中扮演愈发重要的角色，社会基础服务将会进入公有云中，从而对云服务的可靠性安全性有了非常高的要求。比如，早期公有云一个技术人员眼中的小bug，这时就很可能会对社会生活造成严重影响。如阿里云这样的主流公有云服务提供商必须承担起社会使命，犹如水、电、空气一样不能中断、不能出错地提供服务。服务质量越来越重要。

另一方面，由于云服务提供商通过规模效应可以提供更廉价的服务，所以企业、个人都在将自己的服务上云，这又在推动着主流云商必须思考如何以更低的成本提供服务。所以，云商必须深入到基础设施中，把原先的通用性设备改造成适合云的专有设备，以此提高效率；必须提升原先不适合为云服务的管理系统，以满足不间断服务的要求。例如，云商将需要自己运维网络，需要与硬件厂商合作，设计适合特定场景的网卡、CPU、内存等，软硬结合着在底层增加效率，降低企业成本。

**本文为CSDN原创文章，未经允许不得转载，如需转载请联系market#csdn.net(#换成@)**