淘宝Tengine

易运维的高性能Nginx服务器

文 / 朱照远，姚伟斌

Tengine的由来

Nginx是近几年脱颖而出的一个非常优秀的Web服务器，它以资源消耗低、并发能力强著称，现在是世界上第三大Web服务器。在淘宝，我们用它来服务静态文件、PHP动态页面，做反向代理和负载均衡等。根据淘宝的实际需求，我们开发了数十个不同用途的模块。但随着使用的增多，它的一些不足和有待改进的地方也逐渐凸显。例如，Nginx不支持动态模块加载，不同的应用往往需要编译不同的RPM包，从而导致运维比较麻烦；Nginx欠缺输入请求体过滤器机制，从而使得开发安全模块比较困难；不支持Syslog的方式发送日志，导致日志管理烦琐等。Nginx缺少的这些功能都不能通过开发第三方模块来实现，因此我们开始对它的核心进行深度定制和开发。另外，我们在Web服务器领域也积累了一些经验和创新性的想法，希望在Nginx优秀的基础上，继续加强它的性能、安全和可运维性。这就是Tengine项目的由来。

Tengine是Nginx的一个超集，它基于Nginx的最新稳定版本，对其核心进行扩展和增强，同时保持对Nginx的100%向后兼容性。使用Nginx为Web服务器的业务可以无缝迁移到Tengine。因为Tengine继承了Nginx的优点，所以相对于Apache这样传统的Web服务器，它性能更高，而资源占用（CPU、内存等）更省。在处理大量并发的请求时，它的表现更出色、稳定。同时，Tengine经受住了淘宝生产线的长时间考验，对于访问繁忙和服务器数目众多的大型网站尤为适合。

基于生产环境的实际需求，我们跟淘宝的运维工程师紧密合作，对Tengine进行开发，因此我们设计出来的模块也更着眼于实用性、可用性和运维性。例如我们开发的动态模块加载功能，就可以免去打包和编译的烦琐工作，让Nginx使用第三方模块像使用Apache一样方便。Tengine的命令行，可以显示编译进去的模块和全部支持的指令。Tengine的Syslog功能，可以支持Syslog、Pipe、File等多种记录方式，相当灵活。结合tsar开发的统计模块，甚至可以统计QPS、响应时间等数据。Tengine的主动式健康检查模块，可以在不改动配置的情况下，感知后端服务器的健康状况，主动屏蔽有问题的服务器。

2011年12月初，Tengine正式开源。项目主页在http://tengine.taobao.org。淘宝之所以开源Tengine，是因为淘宝是开源软件的受惠者，公司一直很支持技术项目的开源以回馈开源社区——通过Tengine的开源，我们希望能帮助和淘宝一样对高性能Web服务器有迫切需要的人或互联网公司，大家一起享受开源带来的技术进步。同时，开源对于Tengine本身的发展也更有利，例如我们可以获得更多用户的意见和建议、Bug反馈甚至是Patch等。现在Tengine在国内和国外有很多用户，各项功能有众多线上系统在使用，因此他们的反馈也让Tengine更加稳定和注重实效。自开源以来，我们每隔一个月左右发布一个新版本，添加和修改一些功能及修复Bug。当Nginx本身升级时，Tengine也会定时合并Nginx的更新。同时我

们也在和Nginx公司合作，将Tengine对于Nginx的改进提交给他们。最近我们翻译了部分Nginx的英文文档，被Nginx官方收录。他们对Tengine的一些功能也表示了浓厚的兴趣，因此Tengine中的部分功能有望在不久的将来出现在标准Nginx中。

Tengine的改进

Tengine目前的一些性能改进如表1所示。

表1　Tengine对Nginx主要改进模块

应用模块	concat，user_agent，footer，slice
upstream模块	upstream_check
框架模块和Web开发	Lua
管理模块	backtrace，sysguard，traffic status
核心补丁或模块	dso，input body filter，syslog，CPU affinity，procs
数据结构	4-heap，trie

计时器优化

Timers（计时器）是网络服务器中一个很重要的基础设置，用来管理读写超时和应用逻辑的超时等。其常见操作有添加超时、删除超时以及查找最小的超时值。Nginx使用Red-black tree（红黑树）作为其计时器的数据结构。红黑树对应于添加、删除和查找最小值的算法复杂度都是O(log n)。在Tengine中，我们将Nginx的计时器数据结构改为了4-heap（四叉最小堆）。四叉堆是二叉堆的变种，比二叉堆有更浅的深度和更好的CPU Cache命中率。最小堆的添加、删除的复杂度和红黑树一样都是O(log n)，但在查找最小值时，它的算法复杂度是O(1)，即只要取出堆顶的第一个元素即可，因此比Nginx的红黑树更适合频繁获取最小值的场景，特别是在处理大量连接时，用最小堆性能提升比较明显。

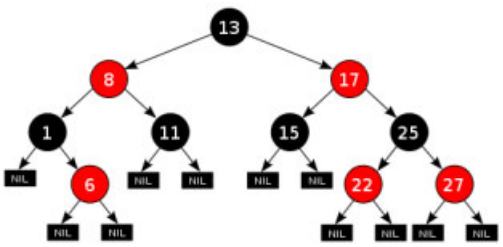

图1　红黑树（图片来自wikipedia）

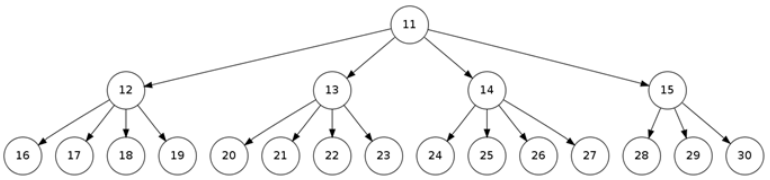

图2　Tengine使用四叉最小堆代替红黑树

浏览器和爬虫的判断优化

判断浏览器的类型是Web服务器的一个常见需求。Nginx中判断浏览器的方法是对关注的浏览器种类在User-Agent头中做暴力查找（strstr）。strstr本身的算法复杂度是O(n^2)，Nginx查找的是多个串，因此其最终算法复杂度是O(n^3)。随着现在移动端的浏览器增多，原有模块的复杂度成指数增长，性能不高。在Tengine中，我们开发了一个全新的user_agent模块，使用了trie（前缀树）来搜索多个可能的浏览器匹配串。它将所有的匹配字符串构造出一个自动机，每次匹配，它的算法复杂度只需要O(n)。因此复杂度不会随着匹配串数量的增加而增加。

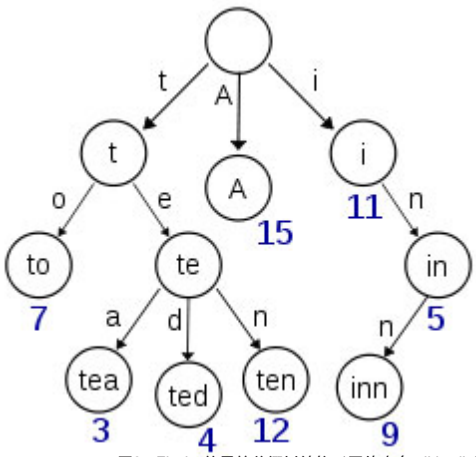

图3　Tingine使用的前缀树结构（图片来自wikipedia）

自动绑定CPU亲缘性

原有的Nginx CPU绑定需要手工操作，在Tengine中我们将Worker进程和CPU进行自动绑定，可以减少因CPU的Cache失效带来的性能损失，从而提高性能。另外，这样也减少了运维配置的工作量。

Tengine对Nginx机制的增强则包含以下几个方面。

Lua模块

基于降低Nginx模块开发难度的初衷，Lua模块（ngx_lua）将Lua嵌入进Nginx核心中，借助

于Lua的协程和Nginx的事件模型实现同步、非阻塞的I/O操作，开发者在Nginx配置文件中可串行同步编写Lua脚本来处理业务逻辑，既可以用它来黏合各种上游（Proxy、Drizzle、Redis、Memcached等）的输出，也可以使用它的Cosocket接口来编写访问上游的客户端。得益于Lua解释器极低的开销和JIT技术（LuaJIT），用户不用编写复杂的C模块就能获得极高的吞吐性能。也可以动态更改逻辑，不用再重新编译Nginx代码，从而带来了极大的灵活性。

Lua模块在初始化时为每个Nginx工作进程创建一个Lua/LuaJIT实例（Lua VM），同一进程处理的所有请求将共享该实例，并且Lua模块将用户Lua代码包装为协程工厂缓存在Nginx内，一个请求到来时协程工厂为它分配一个独立协程来运行业务逻辑。在需要进行阻塞的I/O操作时，Lua模块自动将I/O操作委托给Nginx的事件处理模型，并保存正在运行的协程上下文，返回到Nginx工作进程中处理其他请求，等到I/O操作完成时，又会恢复该协程继续运行。

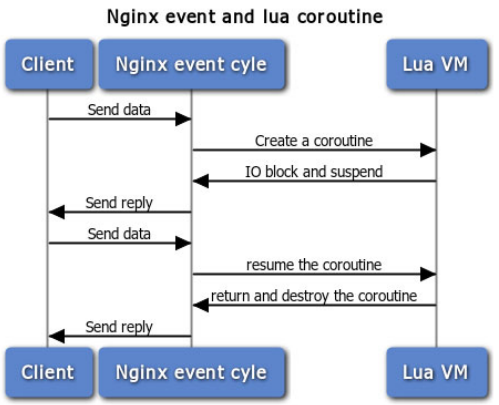

图4 Nginx事件和Lua协程

动态模块支持

Tengine中加入了动态模块功能，对模块实现了动态编译，加入模块不再需要静态编译整个Tengine代码。使用方法类似Apache，在使用时可以当场动态编译想加入的模块，非常方便。

1. 我们提供类似apxs的编译工具，将模块编译成动态链接库。

2. 在Tengine启动时通过动态链接库读入模块的模块结构体，这个结构体包含了模块处理的所有信息。

3. Tengine有内置的模块加载顺序表，也可在配置文件中显式的指定模块的加载顺序，保证模块加载顺序正常。

4. Tengine内部通过两个版本号（Major和Minor）来控制动态链接库（.so文件）的前后兼容性。当Major版本号相同时，较新版本的Tengine兼容较旧版本的.so文件（Tengine的Minor大于.so文件）。只有当Tengine的API发生重大变化时，Major的版本号才发生变化。增加新的API只会增加Minor版本号。

输入体过滤器支持

Nginx没有对请求主体内容的过滤机制，而且在处理较大请求时，可能会缓存到磁盘的临时文件上，因此对输入体的分析和过滤很不方便。Tengine中增加了对于读取用户请求输入体的回调函数，该函数优先于缓存磁盘执行。在收到请求体时会调用这个回调函数，可以方便地对上传的内容进行过滤。而且所有输入体过滤器以链式流程处理。

开启额外进程的机制

Tengine中可以方便地启动进程，这些进程可以独立于原有Nginx工作进程，用来执行某些特殊逻辑（例如非HTTP的应用场景）。该机制在Tengine中增加了一种全新的模块类型，可以开发多个不同用途的进程模块。

对Syslog和管道日志的支持

Syslog功能对于集中式的日志管理非常有用，因此现有大部分的服务器软件都支持Syslog功能。Tengine可以将错误日志和访问日志发送到本地或远程的Syslog服务器。我们完全实现了底层Syslog的协议（使用UDP），解决了Syslog接口阻塞的问题。Tengine也支持通过管道方式将日志写到另一个程序，如Cronolog。此外，在Tengine中，还可以对日志进行抽样，例如只记录1%的日志，从而降低对磁盘I/O的压力，对繁忙的业务颇有用处。

API的增强

Tengine对Nginx的API进行了扩充，如内存操作、HTTP头处理等，以简化模块开发的难度。

目前Tengine比Nginx增加功能模块主要有下面一些。

■ Concat模块

可以组合多个JavaScript和CSS请求变成一个，从而降低下载时间，提高用户体验。该模块对于提高前端的响应时间非常有用。

■ Sysguard模块

在系统的Load或者内存（Swap）使用超过一定阈值或比例时返回等待页面，从而保护服务器。

■ User_agent模块

利用trie结构，扫描浏览器和爬虫的种类，定义$browser和$os变量，比Nginx的Browser模块更加灵活且性能更高。

■ Footer模块

在响应内容后添加一段内容。可用来添加Host信息，对定位大量服务器中出问题的个别机器很有用。

■ Slice模块

用来访问一个文件中的一个片段，可以指定开始和结束的偏移值也可以增加头和尾。

■ Backtrace模块

在遇到异常如崩溃时将调用栈输出到日志以便于问题定位。

■ 主动式Upstream健康检查模块

可以对后端的HTTP、HTTPS、MySQL等类型的服务器定期发起心跳包，维护后端的健康情况，当服务器不可用时，就不再向其发送实际请求。实现了Tengine与后端服务器的高可用。

针对易运维性，我们在几个方面对Tengine进行增强，表2是Nginx与Tengine的对比。

Tengine目前正在做的改进还包括以下方面。

■ 负载均衡的增强如更多负载均衡算法和云的支持等，一致性Hash模块、Session保持模块、后端连接数限制模块、随机负载均衡模块等。

■ Cache功能增强，主要是内存Cache的支持，降低高并发对磁盘I/O带来的影响。

■ 更强的统计模块，可以根据端口或者域名统计流量、连接数等有用的信息。

表2　针对易运维性，Tengine所做的改进

功能	Nginx	Tengine
日志	File	Syslog, Pipe, File，可抽样
编译方式	每次静态编译	核心模块静态编译，功能模块各自编译，动态加载
健康检查方式	被动式检查	主动式心跳检查
状态统计	只显示并发连接数和请求书	可针对域名、端口进行统计，也可对流量进行统计
后端连接数限制	无	可以针对后端服务器进行
连接数限制	命令行可显示版本号和编译	选项还可以显示所以编译进去的模块（-m）、所有支持的指令（-l）、输出所有include 的文件（-d）
CPU 亲缘性	手工绑定	自动绑定
过载保护	无	Sysguard 模块
崩溃输出堆栈	Coredump	Coredump、backtrace 模块

Tengine的社区化发展

目前Tengine主要由淘宝核心系统部维护，其他部门如量子团队、系统保障部等工程师的参与也非常活跃。前淘宝工程师章亦春（agentzh）也给Tengine贡献了大量代码。此外，国内其他互联网公司如搜狗等，也开始参与Tengine的合作开发。

当然，Tengine有着出色表现的最主要原因是我们站在了Nginx这个巨人的肩膀上——正是因为Nginx创立者Igor Sysoev良好的架构设计，优雅的编程风格，对细节的完善处理，让我们受益匪浅，在此我们要对Igor Sysoev致以最高的敬意。

朱照远

花名叔度。淘宝网高级技术专家，任职于核心系统部服务器平台组，Tengine团队成员，负责淘宝Web服务器的开发与Web平台设施的搭建。爱好为开发高性能服务器和Linux内核网络协议栈优化。

姚伟斌

花名文景。淘宝系统工程师，任职于核心系统部服务器平台组，负责淘宝Tengine项目的开发，致力于让Web服务器更好、更快地服务广大用户。