服务器端 JavaScript 技术分析

张 博 于海洋

（沈阳师范大学 软件学院，辽宁 沈阳 110000）

摘 要：现阶段，随着互联网的深入发展，互联网技术逐渐得到广泛应用，越来越普及。而在构建互联网应用的过程中，最主要的技术支持就是前端技术的支持和服务器的支持。一般来说，Java、PHP 等都是现阶段服务器端的主要实现技术，JavaScript 则成为与之相对应的主流前端技术。笔者详细分析一种新型技术——Node 技术，进而深入分析服务器端 JavaScript 技术。

关键词：JavaScript 技术；Node；服务器

中图分类号：TP393.05 **文献标识码：**A **文章编号：**1003-9767（2018）04-019-02

Server-side JavaScript Technical Analysis

Zhang Bo, Yu Haiyang

(School of Software, Shenyang Normal University, Shenyang Liaoning 110000, China)

Abstract: At present, with the development of the Internet, Internet technology has been widely used, and it is becoming more and more popular. In the process of Building Internet applications, the most important technical support is the support of front-end technology and the support of the server. Generally speaking, Java, PHP and so on are the main implementation technologies of the present server side, and JavaScript is the mainstream front-end technology that corresponds to it. The author analyzes a new technology which is Node technology in detail, and further analyzes the JavaScript technology on the server side.

Key words: JavaScript technology; Node; server

1 JavaScript 技术所存在的问题

近年来，富客户端的概念在计算机业内开始逐渐流行起来，在相关技术的支持下，许多基于 JavaScript 的互联网大型应用出现。虽然 JavaScript 技术在近几年来发展迅猛，但是不可否认的是，JavaScript 技术自身的局限性是其发展的重要阻碍。例如，最重要的一点就是，JavaScript 技术十分依赖浏览器，这也就决定了其不能够在一切环境中运行。另外，JavaScript 技术很有可能由于网络因素的影响，导致代码出现不稳定的状况，从而引起整个应用的混乱。虽然随着计算机技术的不断深入发展，已出现了解决这些问题行之有效的方法，但是这些方法的效率普遍较低，并且需要花费庞大的人力资源，在实际中缺乏可行性。

2 Node 综述

Node 就是针对于 JavaScript 技术的局限性而研发出来的一种新型技术手段。Node 的主要作用就是其能够将前端和服务器端巧妙结合起来，这样就能够将 JavaScript 部署到服务器的前端，从而在服务器端建设一个高速的 JavaScript。通过这种手段，就能够以一种简洁的方式弥补 JavaScript 技术存在的一系列不足，使其能够更好发展。

如果想深入了解 Node，开发人员就需要首先了解 V8 引擎，V8 引擎是一种开源 JavaScript 引擎，是由丹麦 iGoogle 所开发的。V8 引擎最大的优势，就是其能够在执行 JavaScript 之前，将整个代码编译成为机器码，这样就避免了字节码或直译带来的麻烦，极大提升了运行效率[1]。另外，V8 引擎性能良好，其中最重要的一点就是其通过内联缓存的方法，提升了整体性能。正是因为这些优化和工作方式的改进，JavaScript 程序与 V8 引擎的速度可以媲美二进制编译。

了解了 V8 引擎，就可以详细阐释 Node。简单来说，Node 是 JavaScript 的一个运行环境。Node 在 V8 引擎原有的基础上对其进行了二次封装，并进一步优化了其特殊用例，这样就为 JavaScript 提供了一个替代的 API，使得 V8 引擎能

作者简介：张博（1994-），男，吉林辽源人，本科。研究方向：JavaScript 技术。

够在非浏览器的环境下运行 [1]。

3 Node 使用的意义

Node 就是为了优化 JavaScript 而存在的。在开发 Node 过程中所提出的首要目标就是，通过 Node 对于 JavaScript 的强大辅助功能，为其提供一种简单的构建可伸缩网络程序的方法。提到这里，开发人员就有必要了解现阶段服务器端程序所存在的问题，如果在 PHP 和 Java 这些服务器语言当中，其所有的连接都可以在服务器中形成一个新的线程，开发人员再假设每一个线程所需的配套内存为 2GB，那么在一个 8 GB RAM 的服务器当中，最大连接数就是 4 000 个，但是这个数字仅仅是理论数字，在现实中想要达到十分困难。因此，开发人员能够看出，现阶段服务器端程序所存在的最大问题就是，伴随着网站访问人数的持续增加，并发数的需求也逐渐上涨，因此，在网站运行过程中就需要不断增加服务器，这样就会大大增加成本的耗费。同时，如果一味增加服务器，随之而来的就是用户可能与不同的服务器进行交互。并发连接数限制的问题，现在已成为限制 Web 应用程序发展的最重要因素。

而通过使用 Node，就可以有效解决上述问题。Node 可以改变连接到服务器当中的方式，从而使每个连接都能够形成一个进程，与众不同的是，这些生成的进程不需要分配配套的内存块即可正常开始工作 [2]。另外，Node 不支持锁机制，对于 I/O 调用也没有任何阻碍。因此，Node 就得到了十分成功的改进，将原本作为阻碍的最大并发连接数成功转移到了单个系统的流量当中，极大提升了整个服务器端程序的工作效率，节约了成本。对于整个服务器端 JavaScript 技术的发展而言可谓大有裨益。

4 Node 的基本工作原理

Node 的基本工作原理主要分为两个方面，即事件驱动、异步编程。在 Java 中，回调函数是所有计算机工作者所经常用到的一种函数类型，如果 Socket 对象满足了既定状态，在程序中所注册的回调函数就会随之执行，同时，计算机开发人员在应用 Node 的过程当中可以依照自己的需求来定义回调函数。并且，这些被定义的回调函数也是按照异步的方式来执行的，换句话说，这些回调函数看似是在程序当中按顺序注册完成的，但是在触发的过程中，则与顺序完全没有关联，触发的唯一要素就是触发条件，只要满足触发条件，回调函数就可以运行 [2]。另外，Node 还采用了一种特殊的“非阻塞”数据库来支持事件的循环，这个数据库的最主要作用就是提供一个接口，使得文件系统与数据库之间能够与服务器端程序对接。这样一来，向文件系统发出请求指令时，就不必再等待硬盘就绪，一旦硬盘准备就绪，非阻塞接口就会直接通知 Node，程序就会直接处理请求。这样改善了系统资源无意义浪费的情况，在等待的过程中，服务器可以处理其他的任务。而异步编程就是一种非阻塞式的编程，并发数的问题是服务器端程序开发过程中所面临的最大问题，如果使用传统的阻塞式函数编程，就会引起资源的严重浪费，同时，还会产生不必要的延迟，通过异步编程的应用，可以全方位提升资源的利用效率，Node 单线程、单进程的运行模式，与 JavaScript 的运行方式能够完美契合，这极大提升了 JavaScript 性能。

5 Node 的不足及相应的调整方案

虽然 Node 对于服务器端 JavaScript 技术有着十分明显的提升效果，但是 Node 本身并不是完美的，还有一些不足之处。例如，Node 的单线程单进程模式虽然可以契合 JavaScript 的运行方式，使二者不会产生根本性质的冲突，提升了稳定性，但是这种单线程单进程的运行模式易导致整个请求队列崩溃，因为所有的请求在单线程单进程的运行模式下都是集中在一个高速队列中的，一旦其中的一个请求出现了问题，那么势必会导致整个队列受到影响。另外，Node 虽然将 RAM 性能中的一些弱点成功转移到了系统的运算能力方面，但这也导致了 CPU 的负荷加大，并且在单线程单进程的模式中，多核处理器无法全面使用，这也是 Node 面临的一个重大的问题。

针对这些问题，Node 也给出了相应的处理意见，开发人员可以通过利用软件实施负载均衡的方式来解决，这样就能够根据实际情况将请求发送到多个进程当中。另外，开发人员使用操作系统的内核来进行负载均衡，也能够避免上述问题出现。

6 结 语

Node 是一种新型技术，它能够从根本上优化服务器端 JavaScript 技术，但是这种技术目前尚不完美，还存在一定的缺陷，但是其实用价值是完全值得肯定的。相关的计算机工作人员可以根据实际情况合理应用该项技术，从而为服务器端 JavaScript 技术的运行保驾护航。

参考文献

[1] 高原 . 服务器端 javascript 技术研究 [J]. 信息与电脑 (理论版),2012(1):78.

[2] 骆文亮 .Node.js 服务器技术初探 [J]. 无线互联科技 ,2014(3):227.