使用Node.J来构建高速和可扩展的后端数据库服务器

摘要——随着应用程序和网站开发的日益增多，开发人员必须使用数据库服务器来高效地存储和传输数据。有各种第三方云数据库提供商（Google，AmazonAWS，Mongolab）可供开发人员开始构建应用程序，而无需担心后端服务（服务器端脚本）。 这种方法对于小规模开发人员来说并不是很有效率和负担得起的。此外，一直有一个关于隐私数据的担忧，因为数据未存储在我们自己的机器中。为了解决开发人员的这个问题，开发人员需要一种方式来构建他们的应用程序，同时托管自己的本地机器作为数据库服务器。也使他们免于繁琐的服务器端编码。Node.Js是主要用于实时应用的服务器端平台，因为它是“事件驱动架构”和“非阻塞I / O”。Node.js已经被发现在I / O操作中快10倍。

索引术语——网站开发，数据库服务器，Node.js，服务器端平台，非阻塞I / O

1、介绍

随着网络的快速发展，应用和网站的发展迅速增长。开发人员总是需要一个数据库服务器来存储应用程序数据。这些数据可能非常大，通常需要随时随地实时访问。目前，如果有后端服务可以与所有平台上的应用程序进行连接，构建应用程序非常方便。但是，开发人员的主要挑战是为其应用挑选/选择/构建灵活的高并发后端。构建后端服务器需要考虑许多因素，如脚本语言，客户端处理机制，数据传输过程等。手机现在远远不止于与之通信的设备。特别是智能手机是有助于使我们的工作和日常生活更容易的产品[13]。

目前，开发人员经常使用以下几种方法将应用程序与后端数据库连接起来：

•第三方云服务器：这种通常称为PaaS（平台即服务）的服务可根据需要为应用开发人员提供各种数据存储计划，并支持不同的平台。

•实现后端服务器：开发人员可以通过使用不同的脚本语言（如PHP，Ruby on Rails等）构建自己的服务器来部署其数据。

2、文献综述

Web服务器最重要的是能够有效地处理多个用户。这与用于编写其脚本的编程语言有很大的关系。因此，与Node.js比较，考虑了像PHP，Python这样的服务器端脚本语言的性能[8]。P. S. Bangare等人提出了一种新的安全加密机制，它是混沌逻辑映射和RC4流密码的组合[10]。

如果您想要某种实时交互，实时结果，Node.js是一个很好的工具。它能够非常快速地向/从Web服务器传送数据。传统上，CPU一次只能做一件事的电脑一直是个大问题。很久以前，它已经被多线程解决了，让我们在单个CPU上有多个“线程”。它一直在它们之间切换，而且相当快，交换机有一大堆开销。为了避免这种开销，node.js通过在单个事件驱动的线程中运行来解决这个问题。而不是在每个请求上创建一个新的线程，每个请求都有一个线程。当一个新的进入，它触发一个运行一些代码的事件。例如，当您连接数据库时，您只需在调用完成后运行回调函数即可，而不是阻止它返回。任何数量的回调都可以响应任何事件，但只有一个回调函数将永远在任何时候执行。程序可能执行的其他任何操作——比如等待来自文件或传入HTTP请求的数据——由Node在幕后并行处理。您的应用程序代码将永远不会在同一时间执行，因为Web服务器最重要的是能够有效地处理多个用户。这与用于编写其脚本的编程语言有很大的关系。因此，与Node.js比较，考虑了像PHP，Python这样的服务器端脚本语言的性能[8]。

重型I / O应用程序从中受益匪浅，而CPU密集型应用程序不会受益。然而，后端数据库服务器的I / O密集度更高，因此通常是有效的权衡。许多中间件任务是I / O绑定的，就像客户端脚本和数据库一样。这些服务器端程序通常不得不等待数据库结果，来自第三方Web服务的反馈或传入的连接请求。Node.js专为这些应用程序而设计。

当对多用户进行比较时，Node.js的性能更好。Node.js的性能比PHP大两倍，比Python大六到七倍。因此使它高度并发和实时[8]。

在考虑Web服务器时，数据库的选择非常重要。应该选择服务器框架。Node.js是用JavaScript环境编写的，由于MongoDB也适用于JavaScript，它可以完美地同步和协同工作。 MongoDB是一个文档数据库。行的概念仍然存在，但列从结构中删除。而不是定义行中应该是什么的列，每一行都是一个文档，而这个文档既定义又保存数据本身。 MongoDB将文档存储为BSON，这是二进制JSON。简而言之，JSON是一种保存数据的JavaScript方法，因此为什么MongoDB非常适合我们以JavaScript为中心的框架。

3、相关工作

最近有很多基于Node.JS的程序的实现。这些系统已经通过将Node与其他工具集成并使应用程序的Web服务器实现[7]。

Stefan Tilkov和Steve Vinoski研究了如何使用基于JavaScript的Node.js来构建高性能程序。他们研究并比较了单线程进程中多线程的性能[7]。

开源社区linnovate开发了一种使用MongoDB，Express和Angular.Js的Node.Js来创建一个称为MEAN技术栈的开发方法来用于全栈开发。这显示了如何将这些所有技术集成到网站和应用程序的服务器上。

4、建议制度

本文旨在为包含Node.js和MongoDB等软件包的应用开发人员实施一个框架，并创建API来轻松地将应用程序与数据库连接起来，而无需担心服务器端编码。此API也可用于为Android，.Net，IOS等各种平台创建驱动程序。

应用程序开发人员需要一个更方便和简单的方法来将其应用程序和网站与数据库服务器连接起来。这个后端服务器应该能够处理多用户请求，并且应该具有高并发性。还应考虑到数据的隐私和安全性。

Node.js目前是JavaScript中的新兴技术。它是一个建立在Chrome的JavaScript运行时的平台，用于轻松构建快速，可扩展的网络应用程序。使用Node的核心功能，并将其与最快和可扩展的数据库（即MongoDB）集成，我们打算为我们的数据库实现一个后台服务器框架，可以帮助您轻松地将我们的前端与MongoDB连接起来。该框架将具有以上优点：

1.它将使开发人员免于编写服务器端脚本，从而使他只集中在前端应用程序。

2.这个框架使用Node.js而不是传统的脚本语言，比如PHP，Ruby on Rails等。

3.这个框架上的所有结构在整个技术栈中只使用一种语言，即JavaScript，因此使其非常适合并同步。

4.由于数据库服务器部署在开发人员系统上，他可以完全控制其隐私和安全性。

Node.js可以与许多数据库（如MySQL，MongoDB等）一起使用，但由于MongoDB与其他数据库的各种优势，我们在项目中使用MongoDB。 MongoDB是日益流行的基于文档的高性能NoSQL数据库。在MongoDB中，数据作为一组键值对存储为文档。它可以在MongoDB中定义多个数据库，其中每个数据库都可以有多个集合，而这些集合只是一组由存储为键值对的数据组成的文档。定义MongoDB文档的数据结构称为BSON（二进制JSON）。 BSON是JSON的二进制表示形式，并且还支持数据类型，如Date，它不支持纯JSON格式。 MongoDB内部将JSON转换为BSON，反之亦然，以获得性能优势，尽管用户可以以JSON格式保存，查询和检索。

MongoDB在包含非常高的写入负载的情况下提供非常好的性能，但数据完整性不是紧迫的问题；一个很好的例子是Craigslist或“纽约时报”等大型繁忙网站的评论部分——顺便说一句，这些网站不是理论上的：这两个都使用MongoDB [4]。我们的主要目的是实现一个集成node.js及其模块（如mongojs，express.js）的单一框架。

MongoDB作为远程数据库的部署在对其性能进行评估后，将其考虑在内。本文旨在实现使用restfull API与常规API（如SOAP或CORBA）的框架。REST是一种架构风格，它使用简单的HTTP调用进行机间通信，而不是像CORBA，COM +，RPC甚至SOAP这样更复杂的选项。使用REST意味着您的呼叫将基于消息，并依赖于HTTP标准来描述这些消息[5]。使用HTTP协议意味着REST是一个简单的请求/响应机制。每个请求返回后续响应[5]。 P. A. Kotwal等人在他们的工作中已经表明了使用App Server及其在研究工作中的作用[14]。

5、详细信息

Node.js是一个软件平台，允许您创建自己的Web服务器，并在其上构建Web应用程序。 Node.js本身不是一个网络服务器，也不是一种语言。它包含一个内置的HTTP服务器库，这意味着您不需要运行单独的Web服务器程序，如Apache或IIS。这最终可以让您更好地控制Web服务器的工作原理，但会增加其在实时环境中的复杂性。以PHP为例，您可以轻松地找到运行Apache的共享服务器webhost，通过FTP发送一些文件，并且您的站点正在运行。这是因为webhost已经为您和其他人配置了Apache。使用Node.js时，在创建应用程序时配置Node.js服务器并非如此。

Express用于管理用户会话，具有MongoDB的可选支持。用户认证通常会使用Node.js，Express，MongoDB和Mongoose，但是有许多第三方模块可以插入到应用程序中，这样您就不用做所有的工作了。可以使用以下代码启用本地端口上的服务器：

var express = require('express')；

app.listen(3000)； //参数是端口号console.log（“端口3000上运行的服务器”）；

//这将启动端口号上的服务器。 300的本地主机

var express = require（'express'）；

app.listen（3000）； //参数是端口号console.log（“端口3000上运行的服务器”）；

Express通过设置Web服务器来收听传入的请求并返回相关的响应来摘取这个难题。此外，它还定义了一个目录结构。这些文件夹中的一个设置为以非阻塞方式提供静态文件，最后一件事是您的应用程序必须等待其他人请求一个CSS文件！您可以直接在Node.js中配置，由Express为您做相关处理工作。

事件和回调的异步编程模型非常适合于必须等待许多事情的服务器，例如传入请求和与其他服务（如MongoDB）的进程间通信。

MongoDB是一个低开销数据库，其中所有实体都是自由形式的BSON——“binary JSON”——文档。这使您可以处理异构数据，并可以轻松处理各种数据格式。由于BSON与JSON兼容，构建REST API很简单——服务器代码可以将请求传递给数据库驱动程序，而无需大量的中间处理。

Node.js和MongoDB在分布式模型中可以通过多台机器进行固有的可扩展和同步化；这种组合对于没有均匀分布负载的应用程序是一个不错的选择。Node.js耦合的异步性与其与MongoDB的兼容性使得它对我们的服务器至关重要，因为它是一个实时运行的框架。我们还必须编写应该在数据库服务器中的所有可能的操作。所有CRUD（创建，读取，更新，删除）到或从数据库。

我们使用Node.js的回调功能构建所有CRUD操作的API。触发时的API导致在数据库中执行其操作并以结果回调函数结束。

插入API的示例如下：

the bellow example of an Insert API:

app.post('/register'， function (req， res) {

console.log("Request for Insertion received with this data")；

console.log(req.body)；

db.users.insert(req.body， function(err，docs){

console.log("Data after insert: ")；

console.log(docs)；

res.json(docs)；

})；

})；

这些API适用于各种HTTP协议。在上面的代码中，我们从客户端获取一个需要插入JSON数据的HTTP请求。该代码通过使用mongodb驱动程序功能来执行，并且使用HTTP响应变量进行响应[6]。然后，此API用于为Android平台构建驱动程序。 API测试也是可行的，S.L.Bangare等人[11] [12]已经提到了相关的工作。

6、结论

本文介绍了使用node.js为Web开发人员以及应用程序开发人员构建高度可扩展和高速后端数据库服务器的框架。它还演示了在提出的项目中使用NoSQL数据库，如MongoDB在其他传统数据库（如MySQL）上的工作。

7、致谢

我们感谢Hodit教授A.P.P. Patil博士，副研究员V.V.Wadhai博士，浦那Sinhgad工程学院院长，为这项研究工作提供支持。另外，我们感谢P. T. Patil博士，VIT教授，Pune和P. Pradeepini博士，K. L.大学教授A.P.他们宝贵的指导。

参考文献

[1] <http://node.js.org/>

[2] <http://php.net/>

[3] <http://python.org/>

[4] <https://www.upguard.com/articles/mysql-vsmongodb>

[5] <http://blog.pluralsight.com/representational-statetransfer-tips>

[6] <https://github.com/mongodb/node-mongodbnative>

[7] S. Tilkov， S. Vinoski， “Node.js: Using JavaScript To Built High-performance Network Programs”， Internet Computing， IEEE， Page(s): 80-83 Volume: 14， Issue: 6， 01 November 2010.

[8] Kai Lei， Yining Ma， Zhi Tan， “Performance Comparison and Evaluation of Web Development Technologies in PHP， Python and Node，js”， Computational Science and Engineering (CSE)， IEEE ， Page(s): 661-668 ，19 December 2014.

[9] Jim R. Wilson， “Node.js the Right Way:Practical Server Side Javascript that Scales”， The Pragmetic express， ISBN-13: 978-1937785734.

[10] P. S. Bangare， S. L. Bangare， “Implementing Separable Reversible Data Hiding In Encrypted Image Using CLM RC4 Method”， International Journal of Engineering Research and Technology (IJERT)， Volume 3， Issue 04， 2014/4/26.

[11] S. L. Banagre， P. S. Bangare， “Automated API Testing Approach”， International Journal of Engineering Science and Technology， Volume 4， Issue 2， Page(s) 673-676， ISSN: 0975-5462， 2012.

[12] S. L. Bangare， P. S. Bangare， “Automated Testing in Development Phase”， International Journal of Engineering Science and Technology， Volume 4， Issue 2， Page(s) 677-680， ISSN: 0975-5462， 2012

[13] P. S. Bangare， S. L. Bangare， “The Campus Navigator: An Android Mobile Application”， International Journal of Advanced Research in Computer and Communication Engineering， Vol. 3， Issue 3， March 2014， Page(s) 5715-5717， ISSN (Online) : 2278.

[14] P. A. Kotwal， S. L. Bangare， “A Location Tracer With Social Networking Services”， International Journal of Engineering and Technology (IJET)， Vol 4 No 1 Feb-Mar 2012， Page(s) 19-23， ISSN: 0975-4024.