**NodeJS的特点概述：**

　　NodeJS宣称其目标是“旨在提供一种简单的构建可伸缩网络程序的方法”，那么它的出现是为了解决什么问题呢，它有什么优缺点以及它适用于什么场景呢？

　　本文就个人使用经验对这些问题进行探讨。

**一. NodeJS的特点**

　　我们先来看看NodeJS官网上的介绍：

　　Node.js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

　　其特点为：

　　1. 它是一个Javascript运行环境

　　2. 依赖于Chrome V8引擎进行代码解释

　　3. 事件驱动

　　4. 非阻塞I/O

　　5. 轻量、可伸缩，适于实时数据交互应用

　　6. 单进程，单线程

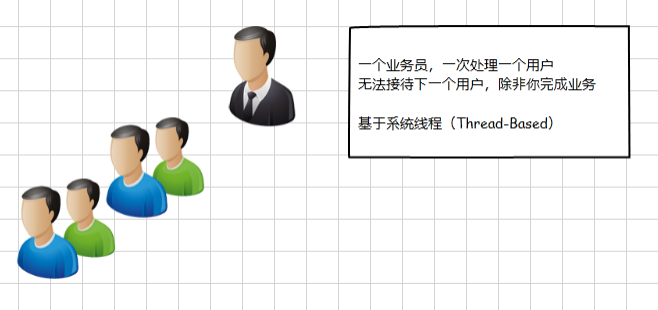
**二. NodeJS带来的对系统瓶颈的解决方案**

　　它的出现确实能为我们解决现实当中系统瓶颈提供了新的思路和方案，下面我们看看它能解决什么问题

**1. 并发连接**

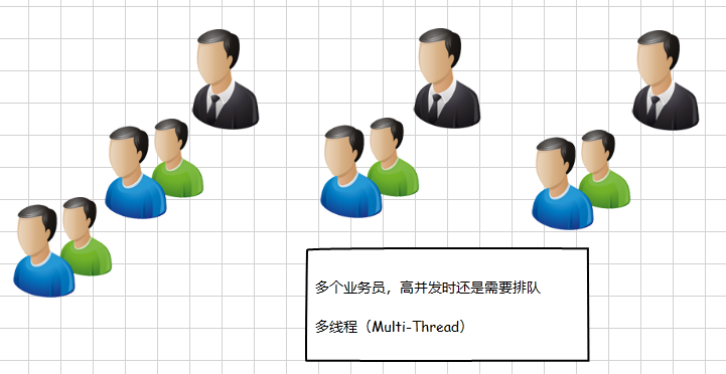
　　举个例子，想象一个场景，我们在银行排队办理业务，我们看看下面两个模型

**（1）系统线程模型：**



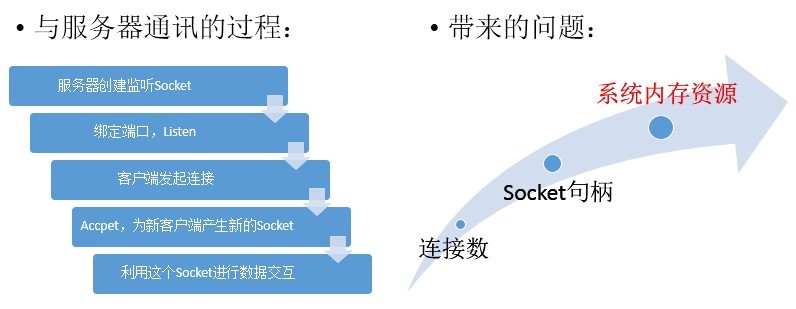
　　这种模型的问题显而易见，服务端只有一个线程，并发请求（用户）到达只能处理一个，其余的要先等待，这就是阻塞，正在享受服务的请求阻塞后面的请求了

**（2）多线程、线程池模型：**



　　这个模型已经比上一个有所进步，它调节服务端线程的数量来提高对并发请求的接收和响应，但并发量高的时候，请求仍然需要等待，它有个更严重的问题：

　　回到代码层面上来讲，我们看看客户端请求与服务端通讯的过程：



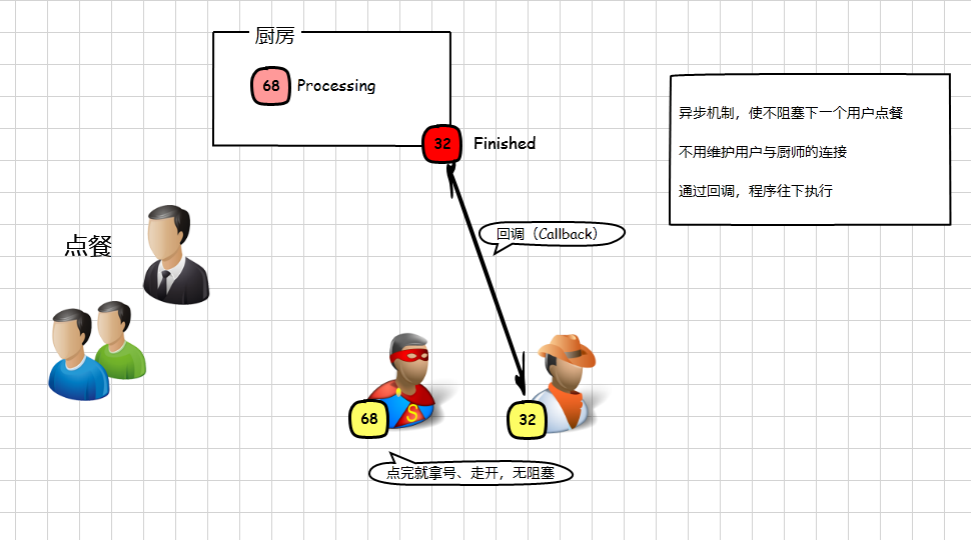
　　服务端与客户端每建立一个连接，都要为这个连接分配一套配套的资源，主要体现为系统内存资源，以PHP为例，维护一个连接可能需要20M的内存

　　这就是为什么一般并发量一大，就需要多开服务器

　　那么NodeJS是怎么解决这个问题的呢？

　　我们来看另外一个模型，想象一下我们在快餐店点餐吃饭的场景

**（3）异步、事件驱动模型**



　　我们同样是要发起请求，等待服务器端响应；但是与银行例子不同的是，这次我们点完餐后拿到了一个号码，

　　拿到号码，我们往往会在位置上等待，而在我们后面的请求会继续得到处理，同样是拿了一个号码然后到一旁等待，接待员能一直进行处理。

　　等到饭菜做号了，会喊号码，我们拿到了自己的饭菜，进行后续的处理（吃饭）

　　这个喊号码的动作在NodeJS中叫做回调（Callback），能在事件（烧菜，I/O）处理完成后继续执行后面的逻辑（吃饭），

　　这体现了NodeJS的显著特点，异步机制、事件驱动

　　整个过程没有阻塞新用户的连接（点餐），也不需要维护已经点餐的用户与厨师的连接

　　基于这样的机制，理论上陆续有用户请求连接，NodeJS都可以进行响应，因此NodeJS能支持比Java、PHP程序更高的并发量

　　虽然维护事件队列也需要成本，再由于NodeJS是单线程，事件队列越长，得到响应的时间就越长，并发量上去还是会力不从心

**总结一下NodeJS是怎么解决并发连接这个问题的：**

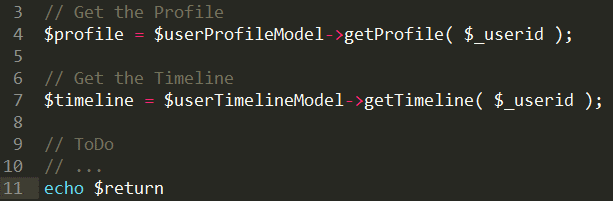
**更改连接到服务器的方式，每个连接发射（emit）一个在NodeJS引擎进程中运行的事件（Event），放进事件队列当中，**

**而不是为每个连接生成一个新的OS线程（并为其分配一些配套内存）**

**2. I/O阻塞**

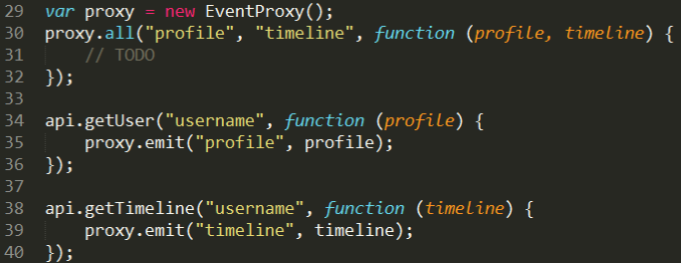
　　NodeJS解决的另外一个问题是I/O阻塞，看看这样的业务场景：需要从多个数据源拉取数据，然后进行处理

　　（1）串行获取数据，这是我们一般的解决方案，以PHP为例



　　假如获取profile和timeline操作各需要1S，那么串行获取就需要2S

　　（2）NodeJS非阻塞I/O，发射/监听事件来控制执行过程



　　NodeJS遇到I/O事件会创建一个线程去执行，然后主线程会继续往下执行的，

　　因此，拿profile的动作触发一个I/O事件，马上就会执行拿timeline的动作，

　　两个动作并行执行，假如各需要1S，那么总的时间也就是1S

　　它们的I/O操作执行完成后，发射一个事件，profile和timeline，

　　事件代理接收后继续往下执行后面的逻辑，这就是NodeJS非阻塞I/O的特点

**总结一下：**

**Java、PHP也有办法实现并行请求（子线程），但NodeJS通过回调函数（Callback）和异步机制会做得很自然**

**三. NodeJS的优缺点**

**优点：**

　　1. 高并发（最重要的优点）

　　2. 适合I/O密集型应用

**缺点：**

　　1. 不适合CPU密集型应用；CPU密集型应用给Node带来的挑战主要是：由于JavaScript单线程的原因，如果有长时间运行的计算（比如大循环），将会导致CPU时间片不能释放，使得后续I/O无法发起；

　　　　解决方案：分解大型运算任务为多个小任务，使得运算能够适时释放，不阻塞I/O调用的发起；

　　2. 只支持单核CPU，不能充分利用CPU

　　3. 可靠性低，一旦代码某个环节崩溃，整个系统都崩溃

　　　　原因：单进程，单线程

　　　　解决方案：（1）Nnigx反向代理，负载均衡，开多个进程，绑定多个端口；

　　　　　　　　　（2）开多个进程监听同一个端口，使用cluster模块；

　　4. 开源组件库质量参差不齐，更新快，向下不兼容

　　5. Debug不方便，错误没有stack trace

**四. 适合NodeJS的场景**

　　1. RESTful API

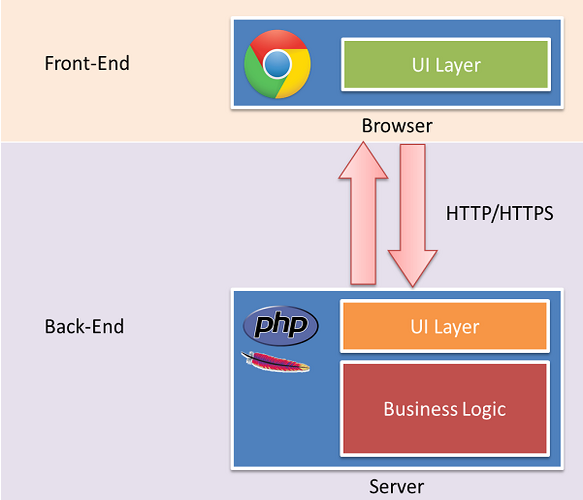
　　这是NodeJS最理想的应用场景，可以处理数万条连接，本身没有太多的逻辑，只需要请求API，组织数据进行返回即可。

　　它本质上只是从某个数据库中查找一些值并将它们组成一个响应。

　　由于响应是少量文本，入站请求也是少量的文本，因此流量不高，一台机器甚至也可以处理最繁忙的公司的API需求。

　　2. 统一Web应用的UI层

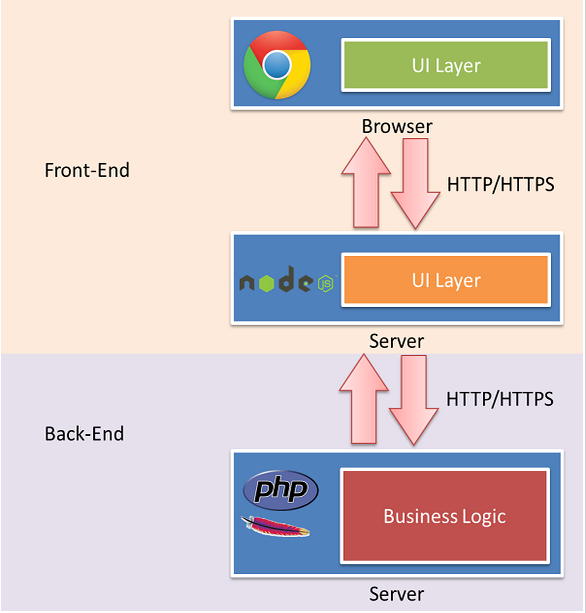
　　目前MVC的架构，在某种意义上来说，Web开发有两个UI层，一个是在浏览器里面我们最终看到的，另一个在server端，负责生成和拼接页面。



　　不讨论这种架构是好是坏，但是有另外一种实践，面向服务的架构，更好的做前后端的依赖分离。

　　如果所有的关键业务逻辑都封装成REST调用，就意味着在上层只需要考虑如何用这些REST接口构建具体的应用。

　　那些后端程序员们根本不操心具体数据是如何从一个页面传递到另一个页面的，他们也不用管用户数据更新是通过Ajax异步获取的还是通过刷新页面



　　3. 大量Ajax请求的应用

　　例如个性化应用，每个用户看到的页面都不一样，缓存失效，需要在页面加载的时候发起Ajax请求，

　　NodeJS能响应大量的并发请求

　　总而言之，NodeJS适合运用在高并发、I/O密集、少量业务逻辑的场景

**五. 结尾**

　　其实NodeJS能实现几乎一切的应用

　　我们考虑的点只是适不适合用它来做

**六. 参考文献**

　　[1] [Node.js能构建支持并发和高负载的大型应用吗?](http://www.dewen.org/q/6209/Node.js%E8%83%BD%E6%9E%84%E5%BB%BA%E6%94%AF%E6%8C%81%E5%B9%B6%E5%8F%91%E5%92%8C%E9%AB%98%E8%B4%9F%E8%BD%BD%E7%9A%84%E5%A4%A7%E5%9E%8B%E5%BA%94%E7%94%A8%E5%90%97%3F)

　　[2] [理解Node.js事件驱动编程](http://www.cnblogs.com/lua5/archive/2011/02/01/1948760.html)

　　[3] [是否不擅长CPU密集型业务](http://book.51cto.com/art/201311/417116.htm)

　　[4] [使用 Node.js 的优势和劣势都有哪些？有大公司用吗？](http://www.zhihu.com/question/19653241)

　　[5] [Node.js给前端带来了什么](http://w3ctech.com/p/1623)

使用 Node.js 的优势和劣势都有哪些？

有大公司用吗？

[3 条评论](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241)

按投票排序 [按时间排序](http://www.zhihu.com/question/19653241?sort=created)

37 个回答

[什么是答案总结？](http://www.zhihu.com/question/20014415) 答案总结

赞同 86反对，不会显示你的姓名

[收起](javascript:;)

匿名用户

知乎用户、[钟汉津](http://www.zhihu.com/people/zhong-han-jin)、[eamin hu](http://www.zhihu.com/people/eamin-hu) [等人赞同](javascript:;)

我们在用 Node.js 处理知乎主站的 web 实时推送。你现在看到的 Feed 、消息的实时更新，背后就是几个 node 进程扛起来的。  
  
优点：  
  
1. 处理高并发场景性能更高  
  
在用 socket.io 之前，推送服务是用 ajax polling 做的。我们用 Tornado 和 Node.js 做过两个版本的推送服务。在当时的测试环境下，Node.js 的 CPU 时间是 Tornado 的三分之一，内存使用是 Tornado 的一半，代码行数只有 Tornado 的三分之一（Node.js 版是用 coffee 写的）。后来我们使用了 socket.io，CPU 开销进一步降低。  
  
2. 函数式编程非常适合写异步回调链  
  
用 Node.js 配合 CoffeeScript 写异步操作链非常便利，相比之下 Tornado 无论是写命名函数的回调，还是 yield 一个 Task 都没那么自然。  
  
缺点：  
  
1. 大量匿名函数使异常栈变得不好看。  
2. 无法以 request 为单位 catch 异常，必须确保不要在不 catch 异常的第三方库的回调里的抛异常，这在一个异步操作链条里是一件比较麻烦的事。解决方法之一是对那些不 catch 异常的第三方库做一些封装，把所有的异常变成事件，改成 on('error') 形式的 API。

[编辑于 2014-03-11](http://www.zhihu.com/question/19653241/answer/23306641" \t "_blank) [17 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 170反对，不会显示你的姓名

[http://pic1.zhimg.com/4cdb1b660_s.jpgFengqiAsia](http://www.zhihu.com/people/fengqiasia)，香港联科集团旗下公有云

[收起](javascript:;)

[总有刁民想害寡人](http://www.zhihu.com/people/tu-po-zhe-pian-lan-tian)、[Vincent Weiss](http://www.zhihu.com/people/vincent-weiss)、[Anderson Aranya](http://www.zhihu.com/people/anderson-aranya) [等人赞同](javascript:;)

先回答第二个问题吧。Node.js其实有很多大公司都在用的，比如eBay, Microsoft, 你可以去Node.js官网看看：[node.js](http://nodejs.org/)  
要想知道更详细的列表，可以看这里：<https://github.com/joyent/node/wiki/Projects,-Applications,-and-Companies-Using-Node>  
  
再稍微花点时间，搜集一些网上资料回答第一个问题：优势和劣势。  
要讲清楚这个问题，先讲讲整个Web应用程序架构（包括流量、处理器速度和内存速度）中的瓶颈。瓶颈在于服务器能够处理的并发连接的最大数量。Node.js解决这个问题的方法是：更改连接到服务器的方式。每个连接发射一个在Node.js引擎的进程中运行的事件，而不是为每个连接生成一个新的OS线程（并为其分配一些配套内存）。Node.js不会死锁，因为它根本不允许使用锁，它不会直接阻塞 I/O 调用。Node.js还宣称，运行它的服务器能支持数万个并发连接。  
  
Node本身运行V8 JavaScript。V8 JavaScript引擎是Google用于其Chrome浏览器的底层JavaScript引擎。Google使用V8创建了一个用C++编写的超快解释器，该解释器拥有另一个独特特征：您可以下载该引擎并将其嵌入任何应用程序。V8 JavaScript引擎并不仅限于在一个浏览器中运行。因此，Node.js实际上会使用Google编写的V8 JavaScript引擎，并将其重建为可在服务器上使用。  
  
Node.js优点：  
1、采用事件驱动、异步编程，为网络服务而设计。其实Javascript的匿名函数和闭包特性非常适合事件驱动、异步编程。而且JavaScript也简单易学，很多前端设计人员可以很快上手做后端设计。  
2、Node.js非阻塞模式的IO处理给Node.js带来在相对低系统资源耗用下的高性能与出众的负载能力，非常适合用作依赖其它IO资源的中间层服务。3、Node.js轻量高效，可以认为是数据密集型分布式部署环境下的实时应用系统的完美解决方案。Node非常适合如下情况：在响应客户端之前，您预计可能有很高的流量，但所需的服务器端逻辑和处理不一定很多。  
  
Node.js缺点：  
1、可靠性低  
2、单进程，单线程，只支持单核CPU，不能充分的利用多核CPU服务器。一旦这个进程崩掉，那么整个web服务就崩掉了。  
  
不过以上缺点可以可以通过代码的健壮性来弥补。目前Node.js的网络服务器有以下几种支持多进程的方式：  
  
#1 开启多个进程，每个进程绑定不同的端口，用反向代理服务器如 Nginx 做负载均衡，好处是我们可以借助强大的 Nginx 做一些过滤检查之类的操作，同时能够实现比较好的均衡策略，但坏处也是显而易见——我们引入了一个间接层。  
#2 多进程绑定在同一个端口侦听。在Node.js中，提供了进程间发送“文件句柄” 的功能，这个功能实在是太有用了（貌似是yahoo 的工程师提交的一个patch） ，不明真相的群众可以看这里： [Unix socket magic](http://www.lst.de/~okir/blackhats/node121.html)  
#3 一个进程负责监听、接收连接，然后把接收到的连接平均发送到子进程中去处理。  
  
在Node.js v0.5.10+ 中，内置了cluster 库，官方宣称直接支持多进程运行方式。Node.js 官方为了让API 接口傻瓜化，用了一些比较tricky的方法，代码也比较绕。这种多进程的方式，不可避免的要牵涉到进程通信、进程管理之类的东西。  
此外，有两个Node.js的module：multi-node 和 cluster ，采用的策略和以上介绍的类似，但使用这些module往往有一些缺点：  
#1 更新不及时  
#2 复杂庞大，往往绑定了很多其他的功能，用户往往被绑架  
#3 遇到问题难以解决  
  
  
Node表现出众的典型示例包括：  
  
1、RESTful API  
提供RESTful API的Web服务接收几个参数，解析它们，组合一个响应，并返回一个响应（通常是较少的文本）给用户。这是适合Node的理想情况，因为您可以构建它来处理数万条连接。它仍然不需要大量逻辑；它本质上只是从某个数据库中查找一些值并将它们组成一个响应。由于响应是少量文本，入站请求也是少量的文本，因此流量不高，一台机器甚至也可以处理最繁忙的公司的API需求。  
  
2、Twitter队列  
想像一下像Twitter这样的公司，它必须接收tweets并将其写入数据库。实际上，每秒几乎有数千条tweet达到，数据库不可能及时处理高峰时段所需的写入数量。Node成为这个问题的解决方案的重要一环。如您所见，Node能处理数万条入站tweet。它能快速而又轻松地将它们写入一个内存排队机制（例如memcached），另一个单独进程可以从那里将它们写入数据库。Node在这里的角色是迅速收集tweet，并将这个信息传递给另一个负责写入的进程。想象一下另一种设计（常规PHP服务器会自己尝试处理对数据库本身的写入）：每个tweet都会在写入数据库时导致一个短暂的延迟，因为数据库调用正在阻塞通道。由于数据库延迟，一台这样设计的机器每秒可能只能处理2000条入站tweet。每秒处理100万条tweet则需要500个服务器。相反，Node能处理每个连接而不会阻塞通道，从而能够捕获尽可能多的tweets。一个能处理50000条tweet的Node机器仅需20台服务器即可。  
  
3、电子游戏统计数据  
如果您在线玩过《使命召唤》这款游戏，当您查看游戏统计数据时，就会立即意识到一个问题：要生成那种级别的统计数据，必须跟踪海量信息。这样，如果有数百万玩家同时在线玩游戏，而且他们处于游戏中的不同位置，那么很快就会生成海量信息。Node是这种场景的一种很好的解决方案，因为它能采集游戏生成的数据，对数据进行最少的合并，然后对数据进行排队，以便将它们写入数据库。使用整个服务器来跟踪玩家在游戏中发射了多少子弹看起来很愚蠢，如果您使用Apache这样的服务器，可能会有一些有用的限制；但相反，如果您专门使用一个服务器来跟踪一个游戏的所有统计数据，就像使用运行Node的服务器所做的那样，那看起来似乎是一种明智之举。  
  
  
总的来说，Node.js的应用场景  
  
1) 适合  
  
JSON APIs——构建一个Rest/JSON API服务，Node.js可以充分发挥其非阻塞IO模型以及JavaScript对JSON的功能支持(如JSON.stringfy函数)  
单页面、多Ajax请求应用——如Gmail，前端有大量的异步请求，需要服务后端有极高的响应速度  
基于Node.js开发Unix命令行工具——Node.js可以大量生产子进程，并以流的方式输出，这使得它非常适合做Unix命令行工具  
流式数据——传统的Web应用，通常会将HTTP请求和响应看成是原子事件。而Node.js会充分利用流式数据这个特点，构建非常酷的应用。如实时文件上传系统transloadit  
准实时应用系统——如聊天系统、微博系统，但Javascript是有垃圾回收机制的，这就意味着，系统的响应时间是不平滑的(GC垃圾回收会导致系统这一时刻停止工作)。如果想要构建硬实时应用系统，Erlang是个不错的选择  
  
  
2) 不适合  
  
CPU使用率较重、IO使用率较轻的应用——如视频编码、人工智能等，Node.js的优势无法发挥  
简单Web应用——此类应用的特点是，流量低、物理架构简单，Node.js无法提供像Ruby的Rails或者Python的Django这样强大的框架  
NoSQL + Node.js——如果仅仅是为了追求时髦，且自己对这两门技术还未深入理解的情况下，不要冒险将业务系统搭建在这两个漂亮的名词上，建议使用MySQL之类的传统数据库  
  
如果系统可以匹配Node.js的适用场景，那么是时候采取具体的措施来说服老板了。  
  
  
说服自己老板采用Node.js的方式  
  
构建一个简单的原型——花一周时间构建系统某一部分的原型是非常值得的，同时也很容易和老板在某一点达成一致，等到系统真的在某一部分应用了Node.js，就是打开局面的时候  
寻找开发者——首先JavaScript语言的普及度很高，一般公司都不乏Web前端工程师，而此类工程师的学习门槛也非常低。这就意味着Node.js很容易招人，或者公司就隐藏了一些高手  
强大的社区支持——Node.js社区非常活跃，吸引很多优秀的工程师，这就意味着公司可以很容易从社区得到免费或者付费的支持  
系统性能考虑——JavaScript引擎Google V8，加之原生异步IO模型，使得Node.js在性能的表现非常出色，处理数以千计的并发请求非常轻松  
专业公司的支持——使用开源技术的最大问题是，原作者不承诺对其产品进行技术支持或者质量保证。现在Node.js已经得到Joyent公司的赞助，这就保证了未来Node.js的发展是可持续性的  
  
  
  
  
参考文档：  
1. [浅谈Node.js的工作原理及优缺点](http://www.52maomao.info/node-js-working-principle-and-advantages-and-disadvantages.html)  
2. [Node.js优缺点](http://blog.csdn.net/kaosini/article/details/8089597)  
3. [NodeJs 多核多进程并行框架实作](http://cnodejs.org/topic/4f16442ccae1f4aa27001081)  
4. [Flexi传授如何说服自己的老板采用Node.js](http://www.infoq.com/cn/news/2012/05/suggest-boss-nodejs)  
  
风起亚洲云：香港联科集团旗下 基于Joyent技术的风起亚洲云充分发挥了Node.js的优势,从而降低了云平台的CPU功耗，增加容量和扩展性。如果您有兴趣，请点击：[風起亞洲](http://www.fengqi.asia/cn/index.html) [显示全部](http://www.zhihu.com/question/19653241/answer/15993549)

[发布于 2013-01-21](http://www.zhihu.com/question/19653241/answer/15993549" \t "_blank) [3 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 42反对，不会显示你的姓名

[http://pic1.zhimg.com/da8e974dc_s.jpg](javascript:void(0);)知乎用户，上知乎，求欢乐

[收起](javascript:;)

[金枪鱼](http://www.zhihu.com/people/zhong-jiu-80)、知乎用户、[徐宁](http://www.zhihu.com/people/xu-zhu-82-67) [等人赞同](javascript:;)

Node.js 的优势是「高并发」，所以很适合用来做 IO 调度，但不太适合用来做复杂计算。  
  
Yahoo 在使用 Node.js，并且已经有了自己的 Node.js 云服务，使用起来类似于 Heroku 或 Google AppEngine——上传应用的源代码，云端自动处理好依赖关系，然后分配适当的资源来运行。Yahoo 使用的方式也很明确，后端真正复杂的逻辑不会用 Node.js 来写，而是 C++ 写好的 JSON API 接口。当来自 Internet 的请求来到 Node.js 时，Node.js 处理部分业务逻辑，但不做复杂计算，将业务逻辑分解为后端 JSON API 调用能解决的问题，然后发起调用。Node.js 的优势在于这些 JSON API 调用都是异步的，等待返回期间不占用任何资源，所以 Node.js 作为前端服务器能够承载更高的并发度。  
  
如果个性化服务是一种趋势，那么 Node.js 高并发的优势就比较明显。个性化服务使得内容数据不能在前端服务器做缓存，因为每个人看到的都不一样。这时候前端服务器就必须处理每一个请求，但后端数据计算则可以并行地做，让用户感觉不到明显延时。

[发布于 2012-08-02](http://www.zhihu.com/question/19653241/answer/15007183" \t "_blank) [5 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 305反对，不会显示你的姓名

[http://pic2.zhimg.com/c5ef55785_s.jpg宓俊](http://www.zhihu.com/people/roquin)，用Node.js开发了 Am I Hacked.

[收起](javascript:;)

[Anderson Aranya](http://www.zhihu.com/people/anderson-aranya)、知乎用户、[Jaytalent](http://www.zhihu.com/people/jaytalent) [等人赞同](javascript:;)

我用 Node.js 开发了 Am I Hacked，算是有一点用 Node.js 支持大流量的经验。先列一些数字

服务器是 Linode 512，也就是 Linode 上最低端的 VPS ，只有 512MB 的内存。

数据库，Node.js 程序和静态文件都放在同一台服务器上。

大部分查询耗时 20-100ms 。少数查询由于缓存 miss 较多，耗时会高达500ms。

最高日PV超过了一百万，Google Analytic 上显示的同时在线人数最高达2000。

平均每秒能完成20-30次查询，瓶颈在磁盘IO，CPU几乎无压力。

虽然压力如此之大，首页几乎都能在一秒内打开，查询也会在3秒内返回。

Node.js 程序占用内存 90MB-110MB，剩余内存都被磁盘缓存占据。

以我的了解，Python 和 Ruby 上的非 Event Driven 的 Framework 根本不可能达到这样的性能。  
  
然后说说 Node.js 的其他优点

Node.js 的架构与 Django, Rails 等传统的 Framework 不同，不需要放在 Nginx / Apache 后，利用 WSGI， CGI 之类的接口一板一眼的 [接受Request] -> [运行程序逻辑] -> [生成并返回Response]。这是一个巨大的变化，之前一些无法想象的功能都有可能实现了。比如 <https://github.com/Miserlou/DirtyShare> 可以用浏览器实现 P2P 的文件传输。正因为 Node.js 可以更精细的控制 Request 和 Response 的时间和内容，websocket 似乎天生就是为 Node.js 而生的，而配合 <http://socket.io> 这个神奇的库之后，在 realtime webapp 这个领域，Node.js 已经没有对手了。

Node.js 的包管理器 npm 设计得比 python 和 ruby 好很多。有很多的 module 开发者。

当然也有一些缺点

Debug 很困难。没有 stack trace，出了问题很难查找问题的原因。

如果设计不好，很容易让代码充满 callback 。实在受不了的可以考虑一下 <https://github.com/laverdet/node-fibers/> 这个项目。不过 Node.js 的核心团队并不推荐使用。

有没有大公司使用？

LinkedIn Mobile 的 服务器端完全是用 Node.js 写的。

Yahoo 有一部分新项目使用了 Node.js。

阿里巴巴内部也有一些新项目用到了 Node.js。

[发布于 2012-07-28](http://www.zhihu.com/question/19653241/answer/14969381" \t "_blank) [46 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 18反对，不会显示你的姓名

[http://pic3.zhimg.com/ccf676352_s.jpg徐海峰](http://www.zhihu.com/people/why520crazy)，Node.js，AngularJS，C#爱好者

[收起](javascript:;)

[对比强烈的对比](http://www.zhihu.com/people/cipher-chen-89)、知乎用户、知乎用户 [等人赞同](javascript:;)

优点：  
1. 异步事件驱动，单进程线程，占用服务器资源少，高并发支持好，虽然单进程，但可以通过官方的 cluster 模块开启多个实例充分利用多核CPU的优势. 节约了服务器的资源，同时又能达到理想的状态  
2. 入门简单，非常适合做 单页程序 + RESTfull API，Worktile就是采用Angular JS + Node.js实现的SPA，基本上完美配合。  
3. 我觉得Node.js也非常适合动态网页web开发，虽然官方没有提供像Apache 和 Tomcat 这样的网页服务器和JSP这样的动态创建网页的技术，但是有很多优秀的第三方模块可以使用， 用 express mvc框架 加上你喜欢的模板引擎（ejs，jade，dot ），感觉还是很棒的  
  
缺点：  
1. 异步编程的缺点往往就是到处callback，代码不优雅，但是可以通过一些第三方的同步模式的模块进行弥补  
2. 不适合做企业级应用开发，特别是复杂业务逻辑的，代码不好维护，事务支持不是很好。  
  
有人说debug成本高，这要取决于你使用什么开发工具了，推荐webstorm，可以对Node.js程序进行单步调试  
  
另外，Node.js也可以独立做操作复杂的后台服务，所以想怎么玩就怎么玩，一切代码尽在你的手中，我也是今年刚接触Node.js，入门快 简单 带给我的是惊喜。每种开发语言都有他的优势和劣势，作为码农，强烈建议没事学学Node.js，我公司的产品 [Worktile 让工作更简单](https://worktile.com/) 就是采用Node.js 开发的，性能和稳定性都还不错

[编辑于 2013-12-25](http://www.zhihu.com/question/19653241/answer/19481226" \t "_blank) [8 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 30反对，不会显示你的姓名

[http://pic2.zhimg.com/50c8e96c1_s.jpg赵东炜](http://www.zhihu.com/people/jackyz)，创业者, Coder ...

[收起](javascript:;)

知乎用户、[Lanyd Li](http://www.zhihu.com/people/lanyd-li)、[廖凯恩](http://www.zhihu.com/people/liao-kai-en) [等人赞同](javascript:;)

我们正在用 node.js 开发系统。系统架构和肉饼同学描述的一模一样：  
前端 javascript ／ app＋JSONAPI＋node后端。  
  
到目前为止的感觉是：  
  
好的部分  
  
1，统一语言。整个 team 里除了我，都是刚毕业（当然有个学习曲线了，但过了就好了），现在所有人前后端通吃，有问题谁都能从前端一直追到数据库。  
2，统一模型。如果你已经习惯异步和回调，那么配合 redis/mq 之类的设施，思考起来会很一致。  
3，社区活跃。各种包，几乎所有能想得到的需求，都有人做了包出来 npm 一装就好。  
  
不好的部分  
  
1，有的包成熟度不高，有时需要 debug 包的内部问题，已经碰到好几回，好在都是开源的，虽说费点劲，但大多都能解决。  
2，容易写出糟糕的代码，callback 的执行流程有时并不是很符合直觉，需要定期 review 和重构来加以避免。  
  
另外，澄清一下  
  
单线程，tcp，这些是常规误解，在 node 0.8 以上版本已经都不是问题。此外，我个人并不认为 node 不适合做 web 开发，虽然现在架构的趋势是比较偏向于 thin json api，但 express 框架确实也能很好地（而且很漂亮）解决上述诸如：路由、缓存、中间件、cookie、session、template 之类的 web 典型问题。  
  
至于是否面向对象，我觉得和代码的组织没有关系。node 提供了完备的 pacakge 机制。代码组织不是问题。

[发布于 2012-08-02](http://www.zhihu.com/question/19653241/answer/15004577" \t "_blank) [8 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 5反对，不会显示你的姓名

[http://pic3.zhimg.com/dd97d253e_s.jpg苏千](http://www.zhihu.com/people/imk2)，生命是一场幻觉

[收起](javascript:;)

[wei teng](http://www.zhihu.com/people/wei-teng-73-22)、[十斗](http://www.zhihu.com/people/tencircle)、[宁克凡](http://www.zhihu.com/people/zhu-ke-fan) [等人赞同](javascript:;)

都看别人说这么多了，还不如开启自己人生的第一个node进程

[发布于 2014-03-12](http://www.zhihu.com/question/19653241/answer/23356763" \t "_blank) [添加评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 2反对，不会显示你的姓名

[http://pic2.zhimg.com/35faadbf0e4695d0781d136c1ab53419_s.jpgYorkie](http://www.zhihu.com/people/yorkie)，<https://github.com/yorkie>

[收起](javascript:;)

知乎用户、[皇甫冰](http://www.zhihu.com/people/bfwzwzgl) 赞同

看了楼上一些回答，说什么javascript是动态语言，不好debug之类的，而且对比来对比去，都是web server上的对比。  
1. NodeJS作为web server是一个相当出色的IO服务器，而Nodejs性能之所以如此高是依赖于joyent自己搞的libuv，而开发效率之所以高是因为v8，所以如果你觉得js不好调试，大可以直接使用libuv，然后各种断点，当然这是在你不会使用nodejs自带的debug模块的前提下  
2. NodeJS真的就只能做做web server，其他免谈。问题来源主要是v8与内存问题，之前做过一个类似爬虫的程序，需要支持万级别用户，用NodeJS写的话，效率没有问题，但内存问题严重，500个任务就需要消耗差不多700-800M空间，后台用libuv+c重写，业务逻辑类似，已经可以单机支持4w-5w的用户量了。因此如果是对内存有非常严格的要求，就建议不要使用node了

[发布于 2014-01-04](http://www.zhihu.com/question/19653241/answer/21375381" \t "_blank) [添加评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 5反对，不会显示你的姓名

[http://pic1.zhimg.com/c048571e0_s.jpg包一磊](http://www.zhihu.com/people/bao-yilei-32)，Dorado7主要作者

[收起](javascript:;)

[Wang Namelos](http://www.zhihu.com/people/wang-namelos)、[Liu Liang](http://www.zhihu.com/people/liu-liang-79-49)、[风吹雪](http://www.zhihu.com/people/feng-chui-xue-91) [等人赞同](javascript:;)

Shelton在BigBang 3x13是这样说的——Windows 7 is much more user-friendly than Windows Vista… I don't like that!  
如果你们公司里有很多小白，那还是看看.Net或Java吧。  
如果你们公司里有很多大牛，千万别提Node.js，否则你拦都拦不住。

[编辑于 2014-03-10](http://www.zhihu.com/question/19653241/answer/23283626" \t "_blank) [2 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 1反对，不会显示你的姓名

[http://pic1.zhimg.com/6a07059a8_s.jpg叔度](http://www.zhihu.com/people/shudu)，系统开发工程师

[收起](javascript:;)

知乎用户 赞同

好处是前端同学也可以写后端程序了（JavaScript）。  
劣势是异步回调机制变成不是很容易。

[发布于 2012-07-27](http://www.zhihu.com/question/19653241/answer/14965823" \t "_blank) [添加评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 16反对，不会显示你的姓名

[http://pic4.zhimg.com/490b338cb_s.jpgcheng](http://www.zhihu.com/people/chengxu)，软件工程师

[收起](javascript:;)

[徐宁](http://www.zhihu.com/people/xu-zhu-82-67)、[Wang Namelos](http://www.zhihu.com/people/wang-namelos)、知乎用户 [等人赞同](javascript:;)

我觉得nodejs目前还不足以取代lamp,lnmp架构，但是很多公司已经尝试使用nodejs做些实时的并发后端。盲目将nodejs来做web框架使用，并不一定可以发挥其优势，而且现在大多数公司不会贸然将新兴的技术用在核心的业务上，比如说web框架。但在其真正有优势的领域，有技术前瞻性的技术领头人，还是会合理采用。  
我所知道的使用nodejs做生产级别应用的，包括搜狐，搜狐小纸条后端就是用nodejs来处理消息的推送的，之前跟其负责人有过交流。  
并不像很多人所言，nodejs会首先在web前段业务层上取代PHP等语言，相反，我认为，其最有可能会首先在服务器端取代原有的很多mq，server做某些工作，而这也应该是正确的方向。只是用nodejs来做框架开发用，有点偏颇，比如说用nodejs来开发某个社区nodeclub等。那只是为用nodejs而用nodejs。  
nodejs的比较对象应该是tornado这些产品，而不是php 。。。

[编辑于 2012-08-04](http://www.zhihu.com/question/19653241/answer/15018653" \t "_blank) [3 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 4反对，不会显示你的姓名

[http://pic2.zhimg.com/f67290c19_s.jpgKenny Lang](http://www.zhihu.com/people/kenny-lang)，热血软件工程师 / Googler

[收起](javascript:;)

[无名](http://www.zhihu.com/people/wu-ming-21-77-42)、知乎用户、[codedump](http://www.zhihu.com/people/codedump) [等人赞同](javascript:;)

讨论NodeJs的优劣，就必须要说道javascript的优劣。  
javascript的严重劣势：  
1.debug成本很高。  
2.出现bug时behavior不稳定。  
3. js本身存在bug。  
4. 代码不易读。  
  
但javascript的最大优势：  
1.快。  
2.占用资源少。  
3.易上手。  
  
因而，NodeJs继承了js的有点的缺点。但是，由于这个framework比较火，所有在陆陆续续出现了好多比较好用的add-on package。安装这些package也极为简单。由于这些add-on，NodeJs就有了很多新优点：  
  
1. 和mangoDB等NoSql的数据库有很方便易用的接口(interface). （个人认为non-sql的database是未来）  
2. 和facebook, twitter, weibo, wechat等其他社交网络登陆认证系统有方便易用的借口——passport。等等  
3. 前后端、数据库统一。  
  
有人说，在未来10年内NodeJs将变成server的主导framework。但10年后人们将会为这个based on JavaScript的framework付出惨痛代价。

[编辑于 2013-06-12](http://www.zhihu.com/question/19653241/answer/17471199" \t "_blank) [6 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 1反对，不会显示你的姓名

[http://pic2.zhimg.com/515673c4aee91fd04491030e83e76ebd_s.jpg谢梦非](http://www.zhihu.com/people/xiemengfei)，INTP

[收起](javascript:;)

[欧阳婕](http://www.zhihu.com/people/ou-yang-jie-90) 赞同

优势：  
1.JS语法优势  
2.异步，性能好  
  
劣势：  
单线程，难以充分利用多核资源

[发布于 2013-12-10](http://www.zhihu.com/question/19653241/answer/20706999" \t "_blank) [1 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 14反对，不会显示你的姓名

[http://pic1.zhimg.com/da8e974dc_s.jpg张春亮](http://www.zhihu.com/people/qi-long-96)

[收起](javascript:;)

[dc lin](http://www.zhihu.com/people/dc-lin)、[赵谜](http://www.zhihu.com/people/zhao-mi-49)、知乎用户 [等人赞同](javascript:;)

有大公司用吗？ 用的不要太多， Groupon, Taobao, 最近Paypal完成转到node.js上面来了  
  
转： [PayPal为什么从Java迁移到Node.js，性能提高一倍，文件代码减少44%](http://ourjs.com/detail/52a914f0127c763203000008)

大家都知道PayPal是另一家迁移到Node.js平台的大型公司，Jeff Harrell的这篇博文 [Node.js at PayPal](https://www.paypal-engineering.com/2013/11/22/node-js-at-paypal/) 解释了为什么从Java迁移出来的原因： 开发效率提高一倍（2个人用更少的时间干了5个人的活）， 性能提高一倍， 代码量减少33%, 文件减少40%：

（小编： 个人认为深层次原因是Java正在越来越走向封闭，而且变得越来越复杂而且oracle正在对Java收费，参见： [Oracle计划发布收费版JVM](http://www.infoq.com/cn/news/2010/11/oracle-to-release-two-jvms) ， 这促使了越来越多的公司加入了去Java化的队伍）

外面有很多人说PayPal正在迁移到node.js平台。我很高兴地在这里宣布，传言是真的，我们正在从Java迁移至node.js

由于历史原因，我们的工程师一直分为两拨人，一拨在浏览器上写代码（HTML,CSS,JavaScript）；另一拨用Java写应用层的代码。想象一下，一个写HTML的不得不去叫一个写Java将A/B两个页面链接到一起吗？我们正在这样干，我们称这样的人为全端工程师，那些即可以设计精美界面和服务器后台的那些人。现在前后端已经没有界限了，这曾经是阻碍PayPal发现的一个很大的瓶颈。

Node.js帮助我们将前、后端合二为一，现在我们一个全端团队即可解决用户的所有问题。

早期采纳

像其他人一样，我们刚开始使用node.js做了一些demo用的原型程序。跟很多人一样，她表现出来的超高性能，让我们最终决定把她放到线上去。

我们最初使用express来路由请求，nconf用来配置，grunt用来创建tasks。Express非常普及，但是我们发现Express 在多个团队协作时表现出的可伸缩性不足，它并不适合所有场合。Expres非常灵活，但在大型团队开发上的可扩展性不佳。最终我们的队员基于原生的 node.js，并创建了Karken.js；她并不是一个框架，更像是一个规范，但相对于express，她更适合大型团队的扩展。我们希望我们的工程师专注他们的应用，而不是专注他们的运行环境。

我们已经在内部使用kraken.js好几个月了（我们马上会把他开源的！）我们的工程师非常渴望这个内部框架能尽快上线。

（小编：预测karken.js即将是，另一个超火的后端框架，火热程度参考twitter的bootstrap）

将node.js布署到线上

我们第一个采用nodejs的产品不是一个小的应用；是我们的浏览量最多的用户首页。我们希望步子迈得大一点，但是我们清楚知道其中的风险，所以我们同时还并行地运行了一个Java的程序。我们在开发和扩展Java方面非常有经验。所以一旦node.js应用出问题了，我们可以立即切回Java。不过，同时我们也发现了一些非常有意思的数据。

开发

从1月份开始，我们花了几个月的时间来搭建node.js的基础设施。比如：sessions（会话），centralized logging(集中日志)，keystores(存储)。在这期间我们有5位Java工程师在开发Java。在开发了两个月后，两位工程师开始开发 node.js应用。在6月初两个团队的开发进度已经一样了，两者的功能完全一样。开发node.js应用的那个小团队，尽管推迟了两个月，但是很快赶上了。这里我们对这些相同功能做的一些单元测试得出的结果：

Node.js的是：

更少的人开发的node.js应用比Java的快一倍；  
节省了33%的代码量；  
少了40的文件；

（小编，这里作者的意思并不是Java程序员的素质没有node.js的好，Java语言的特点决定她需要更多的人，更多的时间，更多的代码去完成在node.js下的同样的工作，并且吃力不讨好。参考：[他们为什么说面向对象有问题,探讨面向对象的一些缺陷](http://ourjs.com/detail/528b2d6ab5cbfd990b000002) ；性能测评：[Node.JS比Java EE快20%](http://ourjs.com/detail/52726d92031db63703000001) ）

这是一个非常鼓舞人的证据，我们似乎应该更快地迁移到JavaScript平台上去。我们立即做了一个决定，暂停Java应用的开发，全心全意开发 JavaScript应用。这对开发Java项目的工程师来说是个好消息，他们已经消除了对node.js的疑虑，非常高兴地投入到了并行的 node.js开发上来，这样我们的开发效率提高了两倍。

性能

性能是一个非常有意思和具有争议性的话题。在我们这，我们有两个平台实现完全一样功能的程序；一个是使用基于Spring的内部Java框架；另一个是基于kraken.js，express，dust.js和其他开源框架。 这些程序包含三个API，每个API来响应2到5个请求，由Dust来模拟获取数据和显示页面。

我们用线上的环境去测试这两个应用，并收集了完成响应的时间和请求数。

node.js vs Java 性能对比

在这张图上你可以看到node.js应用的优势：每秒请求数量是Java的两倍。不过更有意思的是我们仅使用了单核的node去跟5核的Java来对比，我们非常希望将来继承扩大node.js的优势。  
渲染相同的页面，node.js节省了35%的时间。即每个页面节约了200豪秒，用户可以清楚地感觉到这样的区别。

未来

我们将继续使用node.js来构建我们的Web应用。像我们正在开发的那些门户，和已经上线的用户概览页面。还有一打正在进入Beta测试的那些工程，我们会继续分享我们在上线过程中的经验，数据。对于PayPal来说这是一个另人激动的时刻。

[显示全部](http://www.zhihu.com/question/19653241/answer/21022363)

[编辑于 2014-08-14](http://www.zhihu.com/question/19653241/answer/21022363" \t "_blank) [2 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 3反对，不会显示你的姓名

[http://pic1.zhimg.com/a73376cb4_s.jpg谈熠](http://www.zhihu.com/people/halfmile)

[收起](javascript:;)

[十斗](http://www.zhihu.com/people/tencircle)、知乎用户、知乎用户 赞同

优势：互联网上 web 应用正在急速萎缩，而不需要web/html这张皮的应用正在疯长。node 和 erlang 是两个最适合这个潮流变迁的语言。而对人脑和肉眼来说，javascript 更加友善。  
  
劣势：太新了，API还在成熟过程中；不少开发者对javascript持有本能的抵触情绪。  
  
此外，  
  
1. 对于上面几位所说的：所谓 “不稳定，单线程” 劣势，我想说 没仔细深入的玩过之前，先不要妄下定论。V8是恐怕是这个星球上能被最广泛使用的最靠谱的虚拟机。  
  
2. 对于LZ，互联网上有哪个"大"公司能够10年一成不变，屹立不倒吗？

[发布于 2013-04-01](http://www.zhihu.com/question/19653241/answer/16554740" \t "_blank) [2 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 6反对，不会显示你的姓名

[http://pic1.zhimg.com/da8e974dc_s.jpg陈嘉豪](http://www.zhihu.com/people/chen-jia-hao)，软件工程师

[收起](javascript:;)

[阿咩](http://www.zhihu.com/people/a-mie-49)、[李强](http://www.zhihu.com/people/li-qiang-87-82)、[chanble](http://www.zhihu.com/people/chanble) [等人赞同](javascript:;)

优势：  
先简单说一下，node.js官网解释是Evented I/O for  
V8 JavaScript。Evented IO实际上单线程epoll/kqueue模型，简单来说就是比select和poll等效率更高的事件驱动模型；而单线程主要是为了减少locking的开销。这两点使得node.js能够同时处理大量的高并发链接（这种模型与nginx，redis以及haproxy的结构也比较类似）。  
至于V8，则是相当出色的Javascript引擎，好处是熟悉的人多，人力成本低，开发效率高，同时不失语言的运行效率。  
  
劣势：  
由于是单线程，node.js在多核/多CPU的架构上无法充分利用（目前也存在一些解决方案）。同时，由于Javascript的本身特性，需要考虑其适用场景，对大多数HTTP层面的应用来说应该是足够了，对于其他一些底层的设计，比如TCP服务器，可能需要其他很多特性，这时候可能自己添加module或者hack源码。  
本身node.js是一个比较年轻的东西，成熟度等或者适用范围比twisted要略逊一筹。

[发布于 2011-05-24](http://www.zhihu.com/question/19653241/answer/12584486" \t "_blank) [5 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 0反对，不会显示你的姓名

[http://pic4.zhimg.com/3da5ab7eb_s.jpgsherrik](http://www.zhihu.com/people/sherrik)，hey

[收起](javascript:;)

四月的时候上了一个模仿果壳的社区网站  
<http://t.youze.cc/>   
用Node.js写的，开发挺爽的~~~  
不过社区上了之后就一直没管了……

[发布于 2012-08-02](http://www.zhihu.com/question/19653241/answer/15006250" \t "_blank) [添加评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 2反对，不会显示你的姓名

[http://pic4.zhimg.com/6a0bbb0c95abdb2ace6805363b971427_s.jpg宝术](http://www.zhihu.com/people/tedeyang)，1688

[收起](javascript:;)

[黄修源](http://www.zhihu.com/people/dreamcog)、[孙立](http://www.zhihu.com/people/mrsunli) 赞同

劣势：单线程，不稳定，容易崩溃；编程模型局限于异步和事件，开发困难，调试困难，异常堆栈信息不友好，不利于团队开发，js语言特性不完整，需要太多hack。  
优势：在强调I/O事件的应用场景如socket通讯方面有天赋。  
=================================================  
写完这段话后大概三个月有个在线演唱会的需求,本来想用flash+netty实现,后来发生一些变故,改用nodejs+socket.io,效果很好.  
.  
nodejs在异步IO上提供了一个简单易用的体系,另外扩展了js的module设计以及一些必要的简单基础设施如Process,但是javascript语言终归是个不完整的语言,我无法想象一个连extends关键字都没有的语言能具备多好的代码组织性,事实上也是如此,<http://socket.io>是一个非常好的例子,我研究了它的源码,发现非js高手写不出这么结构良好的代码,太需要技巧了.  
异步编程模式对开发的不友好使他天生只能适用于某些不得不用的场合,因为人类要处理的事物其实大部分还是用同步描述的,而且容易理解和维护.试着写个嵌套五层callback的程序看看,除了IO之类的需求,我看不到其他非得要用异步给自己找别扭的必要性.  
另外nodejs的性能也不比java\python强,甚至要弱很多,即便在异步IO领域也是如此,目前还只能利用单核CPU单进程,单线程一点火星就崩溃了,只是函数式语言做异步会比较舒服,丢一个function过去就行了.  
在做完上面那个产品后,我深感nodejs应该会在网页游戏领域大放异彩,nodejs天生是为异步而生, 其他领域,我看还是没有足够的比较优势.

[编辑于 2012-07-27](http://www.zhihu.com/question/19653241/answer/13827087" \t "_blank) [6 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 15反对，不会显示你的姓名

[http://pic1.zhimg.com/5ed23a70c_s.jpg鲁塔弗](http://www.zhihu.com/people/lutaf)，<http://lutaf.com>

[收起](javascript:;)

[丁亮](http://www.zhihu.com/people/ding-liang-50)、[Tong Phil](http://www.zhihu.com/people/tong-phil)、知乎用户 [等人赞同](javascript:;)

前段时间正好在公司做了一个nodejs的讲座

nodejs的内涵 = 闭包语法糖+异步IO+高性能网络

nodejs的优点：快速开发一个可以用于生产环境的tcp server 服务程序

也可以用于一些简单的http 服务场景，类似comet，data proxy 之类

nodejs不适合做web开发，不适合做业务/逻辑复杂应用，没有什么特别的优点，反而比较琐碎，各种包类库支持很有限，性能也不好

nodejs目前比较火，是因为nodejs基于javascript的，语法简单，学习门槛低

所以，我不同意范凯的预测  
nodejs写点小接口服务可以，去陷入各种业务/需求编程，是很不明智的行为

[编辑于 2012-07-28](http://www.zhihu.com/question/19653241/answer/14967151" \t "_blank) [12 条评论](http://www.zhihu.com/question/19653241) [感谢](http://www.zhihu.com/question/19653241) [分享](http://www.zhihu.com/question/19653241) [收藏](http://www.zhihu.com/question/19653241) • [没有帮助](http://www.zhihu.com/question/19653241) •

[举报](http://www.zhihu.com/question/19653241)

赞同 4反对，不会显示你的姓名

[http://pic3.zhimg.com/bd7dbee8a_s.jpgMeteoric](http://www.zhihu.com/people/Meteoric_cry)，JavaScript/Lua/C++

[收起](javascript:;)

[寶蛋](http://www.zhihu.com/people/bao-dan-47)、[郑海](http://www.zhihu.com/people/zheng-hai-9-2)、[wilson](http://www.zhihu.com/people/liuweisan) [等人赞同](javascript:;)

不同的人用nodejs写的后台系统性能会差异相差十万八千里，因为它实在是太需要技巧了。  
  
node.js的大放异彩，我觉得会是在webgame上  
  
劣势：开源的组件库太多，更新太快，向下不兼容  
优势：异步IO