

#### a) Node.js简介

Node.js 诞生于 2009 年，由 Joyent 的员工 [Ryan Dahl](<https://github.com/ry>) 开发而成，之后 Joyent 公司一直扮演着 Node.js 孵化者的角色。由于诸多原因，Ryan 在 2012 年离开社区，随后在 2015 年由于 Node 贡献者对 es6 新特性集成问题的分歧，导致分裂出 io.js，并由 io.js 发布 1.0、2.0 和 3.0 版本。由于 io.js 的分裂最终促成了 2015 年 Node 基金会的成立，并顺利发布了 4.0 版本。Node.js 基金会的创始成员包括 Google、Joyent、IBM、Paypal、微软、Fidelity 和 Linux 基金会，创始成员将共同掌管过去由 Joyent 一家企业掌控的 Node.js 开源项目。此后，Node.js 基金会发展非常好，稳定的发布 5、6、7、8 等版本，截止发稿最新版本已经是 8.6，长期支持版本是 6.11。

Node.js 不是一门语言也不是框架，它只是基于 Google V8 引擎的 JavaScript 运行时环境，同时结合 Libuv 扩展了 JavaScript 功能，使之支持 io、fs 等只有语言才有的特性，使得 JavaScript 能够同时具有 DOM 操作(浏览器)和 I/O、文件读写、操作数据库(服务器端)等能力，是目前最简单的全栈式语言。

早在 2007 年，Jeff Atwood 就提出了著名的`Atwood定律`

> 任何能够用 JavaScript 实现的应用系统，最终都必将用 JavaScript 实现

目前 Node.js 在大部分领域都占有一席之地，尤其是 I/O 密集型的，比如 Web 开发，微服务，前端构建等。不少大型网站都是使用 Node.js 作为后台开发语言的，用的最多的就是使用 Node.js 做前端渲染和架构优化，比如 [淘宝](https://www.taobao.com/) 双十一、[去哪儿网](https://www.qunar.com/) 的 PC 端核心业务等。另外，有不少知名的前端库也是使用 Node.js 开发的，比如，[Webpack](https://github.com/webpack/webpack) 是一个强大的打包器，[React](https://github.com/facebook/react)/[Vue](https://github.com/vuejs/vue) 是成熟的前端组件化框架。

Node.js 通常被用来开发低延迟的网络应用，也就是那些需要在服务器端环境和前端实时收集和交换数据的应用（API、即时聊天、微服务）。阿里巴巴、腾讯、Qunar、百度、PayPal、道琼斯、沃尔玛和 LinkedIn 都采用了 Node.js 框架搭建应用。

另外，Node.js 编写的包管理器 npm 已成为开源包管理领域最好的生态，直接到 2017 年 10 月份，有模块超过 47 万，每周下载量超过 32 亿次，每个月有超过 700 万开发者使用 npm。

当然了，Node.js 也有一些缺点。Node.js 经常被人们吐槽的一点就是：回调太多难于控制（俗称回调地狱）和 CPU 密集任务处理的不是很好。但是，目前异步流程技术已经取得了非常不错的进步，从Callback、Promise 到 Async函数，可以轻松的满足所有开发需求。至于 CPU 密集任务处理并非不可解，方案有很多，比如通过系统底层语言 Rust 来扩展 Node.js，但这样会比较麻烦。笔者坚信在合适的场景使用合适的东西，尤其是在微服务架构下，一切都是服务，可以做到语言无关。如果大家想使 JavaScript 做 CPU 密集任务，推荐 Node.js 的兄弟项目 [fibjs] (<http://fibjs.org/>)，基于纤程(fiber，可以简单理解为更轻量级的线程)，效率非常高，兼容 npm，同时没有异步回调烦恼。

从这段介绍来看，解读要点如下

- Node.js 不是 JavaScript 应用，不是语言（JavaScript 是语言），不是像 Rails(Ruby)、Laravel(PHP) 或 Django(Python) 一样的框架，也不是像 Nginx 一样的 Web 服务器。Node.js 是 JavaScript 运行时环境
- 构建在 Chrome's V8 这个著名的 JavaScript 引擎之上，Chrome V8 引擎以 C/C++ 为主，相当于使用JavaScript 写法，转成 C/C++ 调用，大大的降低了学习成本
- 事件驱动（event-driven），非阻塞 I/O 模型（non-blocking I/O model），简单点讲就是每个函数都是异步的，最后由 Libuv 这个 C/C++ 编写的事件循环处理库来处理这些 I/O 操作，隐藏了非阻塞 I/O 的具体细节，简化并发编程模型，让你可以轻松的编写高性能的 Web应用，所以它是轻量（lightweight）且高效（efficient）的
- 使用 `npm` 作为包管理器，目前 `npm` 是开源库里包管理最大的生态，功能强大，截止到 2017年12月，模块数量超过 60 万+

大多数人都认为 Node.js 只能写网站后台或者前端工具，这其实是不全面的，Node.js 的目标是让并发编程更简单，主要应用在以网络编程为主的 I/O 密集型应用。它是开源的，跨平台，并且高效（尤其是 I/O 处理），包括 IBM、Microsoft、Yahoo、SAP、PayPal、沃尔玛及 GoDaddy 都是 Node.js 的用户。