

Nginx V.S. NodeJS

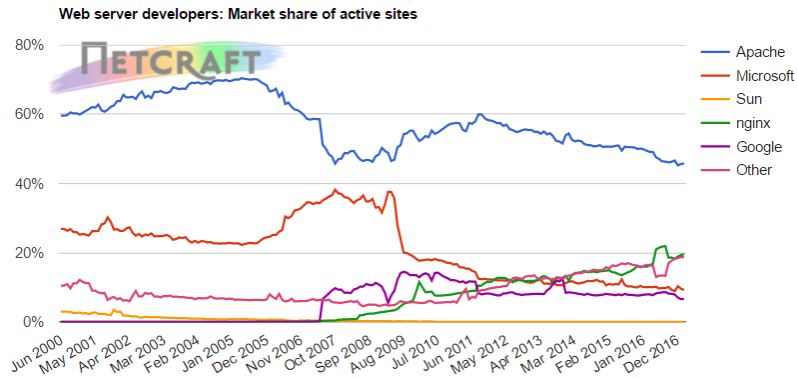
Dai Li

dl37@rice.edu

NodeJS and Nginx are some of the popular solutions for web servers recently. As Internet companies begin to handle various demands from customers, NodeJS and Nginx, because of their capability of handling high concurrency requests, have been winning more attention. This work will compare the features and performance of the two web servers and suggest some solutions to deploy suitable web servers according to website features.

Introduction

Whereas modern Internet greatly elevated information communication for business, entertainment and security, the massively increasing number of requests in Internet requires higher computation and I/O capability. On the backend, the server is the provider of data generation, update and exchange. Among servers, there are application servers that provides support for desktop applications, such as WebLogic and .Net Framework. While for web applications, there are specific web servers that process HTTP requests. From the birth of WWW to the current time of prospered mobile Internet, various servers rise and fall. IIS and Apache are some of the long-living survivors. But now they are facing the challenge from Nginx and NodeJS.



Developer	January 2017	Percent	February 2017	Percent	Change
Apache	78,707,037	45.67%	79,593,938	45.78%	0.11
nginx	33,331,358	19.34%	34,088,228	19.60%	0.27
Microsoft	16,601,302	9.63%	16,031,854	9.22%	-0.41
Google	11,372,796	6.60%	11,656,739	6.70%	0.11

Fig.1. Market share survey of web servers as of Feb 2017 by NetCraft

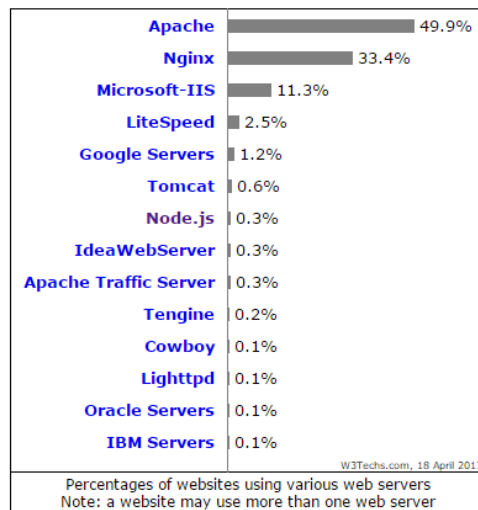


Fig.2. Market share survey as of April 2017 by W3techs

From the above figures we could see that Nginx is becoming more and more popular among websites. And NodeJS is also begin to gain share.

Nginx was created by Igor Sysoev in 2004. It is completely written in C. It uses asynchronous event-driven and non-blocking I/O. It has excellent performance when processing static files. It is also widely used as reverse proxy and load balancing. Recently it even started to support Javascript, making it more widely accepted by software engineers.

On the other hand, NodeJS was created by Ryan Dahl in 2009. It was written in C++ and has a good package manager (npm) to manage modules. This makes it a wonderful platform to run JavaScript on Google V8 engine. NodeJS also uses asynchronous event-driven and non-blocking I/O. Its usage of JavaScript as backend language enables large numbers of frontend engineers to participate in server work and make full-stack engineering a lot easier.

Currently, LAMP and LEMP stacks are the most widely used web service solution stacks. LAMP stands for Linux, Apache, MySQL and PHP. While in LEMP, Nginx takes the place of Apache. Such a stack could handle most internet server demands. For Nginx, it is especially good when there are a large number of concurrent requests with lower memory cost.

While more and more non-relational database is used in modern web applications, NodeJS is also becoming a good choice for full-stack solutions such as MEAN stack, which stands for MongoDB, Express, Angular and NodeJS. The MEAN stack is very agile and can handle present nice frontend pages as well as quick response from backend.

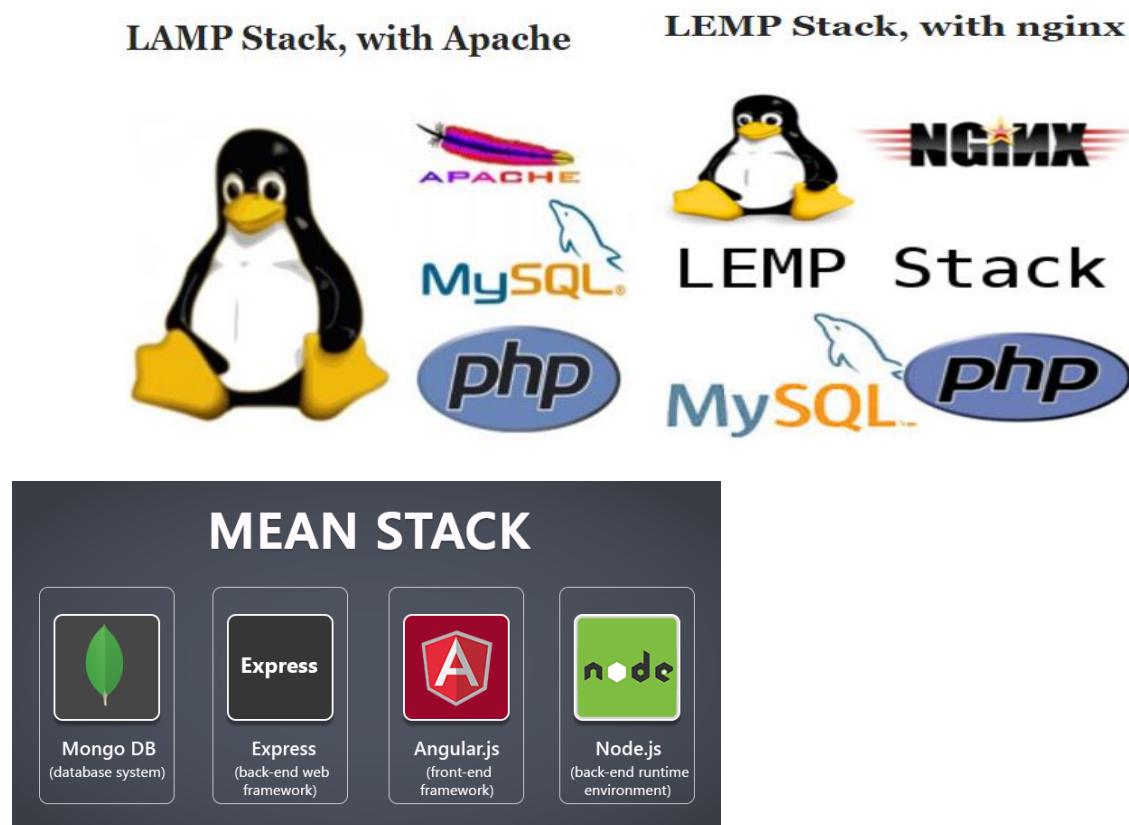


Fig.3. Popular web service stacks, LAMP, LEMP and MEAN

Features	Nginx	NodeJS
Synchronous?	No	No
I/O	Non-Blocking	Non-Blocking
Thread	Multi	Single
Support Language	PHP, JavaScript, etc	JavaScript Only
Usage	Reverse Proxy, Load Balancing, Server, ect	Server mainly

Table.1. Basic Feature Comparison of Nginx and NodeJS

Now we have some understanding of the mechanism of Nginx and NodeJS. We still need to investigate their performance under different situations.

Performance Comparison

Our most concerned issue would be # of requests a server could process in a second. Here are some tests regarding Nginx and NodeJS on performances.

Static File Serving

Test Target VirtualBox Server Config:

Xeon W3550 @3380Mhz with 4 cpu threads

1280MB DDR3-1536Mhz

20GB storage on 1TB Western Digital Black Caviar SATA disk

Servers Tested:

Centmin Mod Nginx v1.2.7 server with 4 worker_processes (port 80)

Centmin Mod node.js Cluster server (4 processes) (port 8080)

Siege benchmark test parameters:

Benchmark mode -b used

100 to 600 concurrent users @500 reps and 5s delay

Cache validation enabled

Using Siege benchmark Siege load test against a static html file which consisted of 50x lipsum auto generated paragraphs from lipsum.com in a html file named lipsum.html.

Result:

It can be observed that NodeJS has even better performance in serving static files than Nginx, which is often thought of very good at it. The transaction per second of NodeJS is almost twice as many as Nginx. It also has shorter average response time and transaction time.

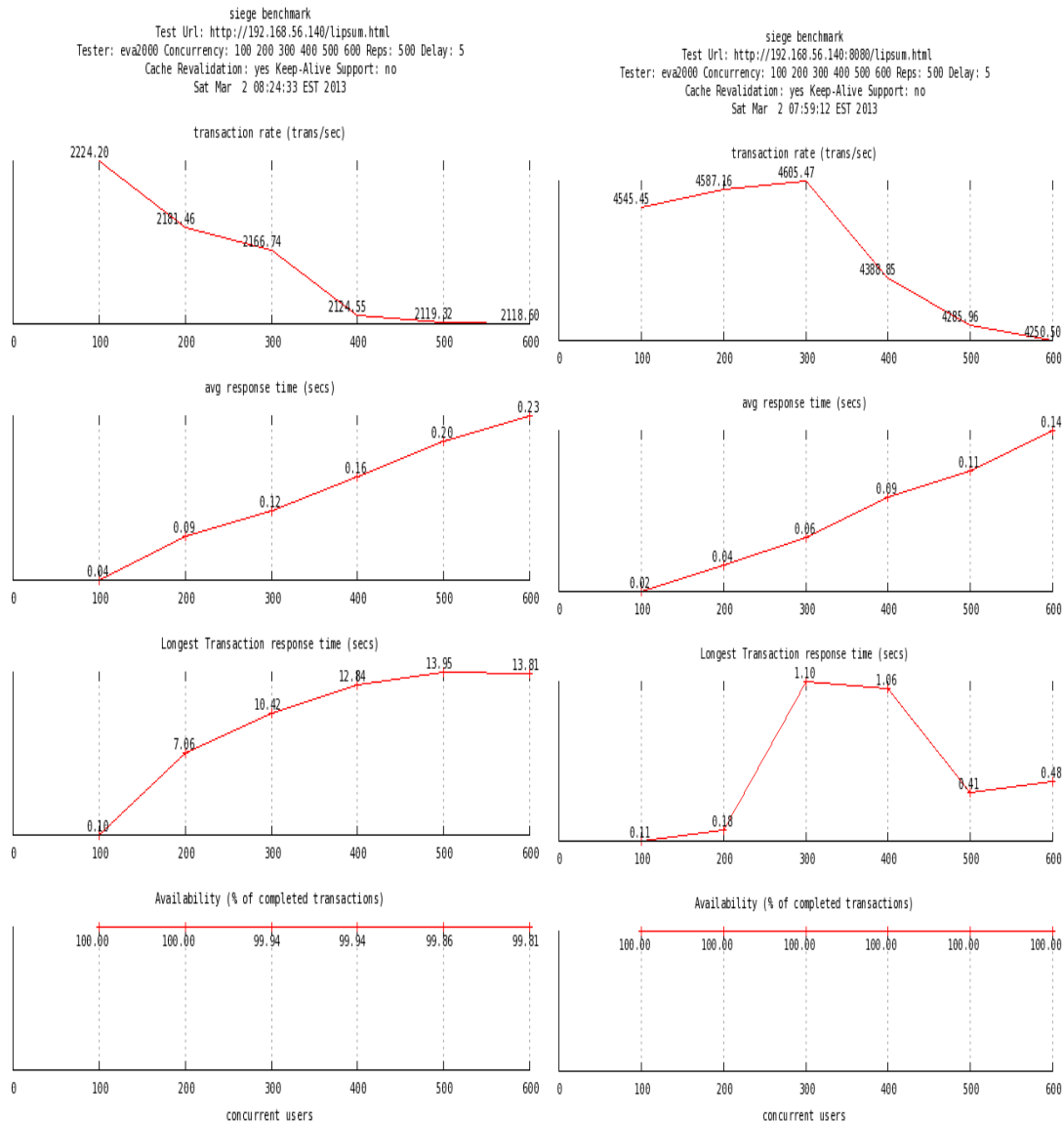


Fig.4. Performance comparison of Static file siege benchmarks

Another Comparison of Serving Static Files of Higher Load

While one's test result may vary from another's due to various request types, contents and personal settings, here we take a look at another comparison.

All test were ran locally on an:

Intel core i7-2600k machine of 4 cores and 8 threads.

Gentoo Linux is the operating system used to run the tests.

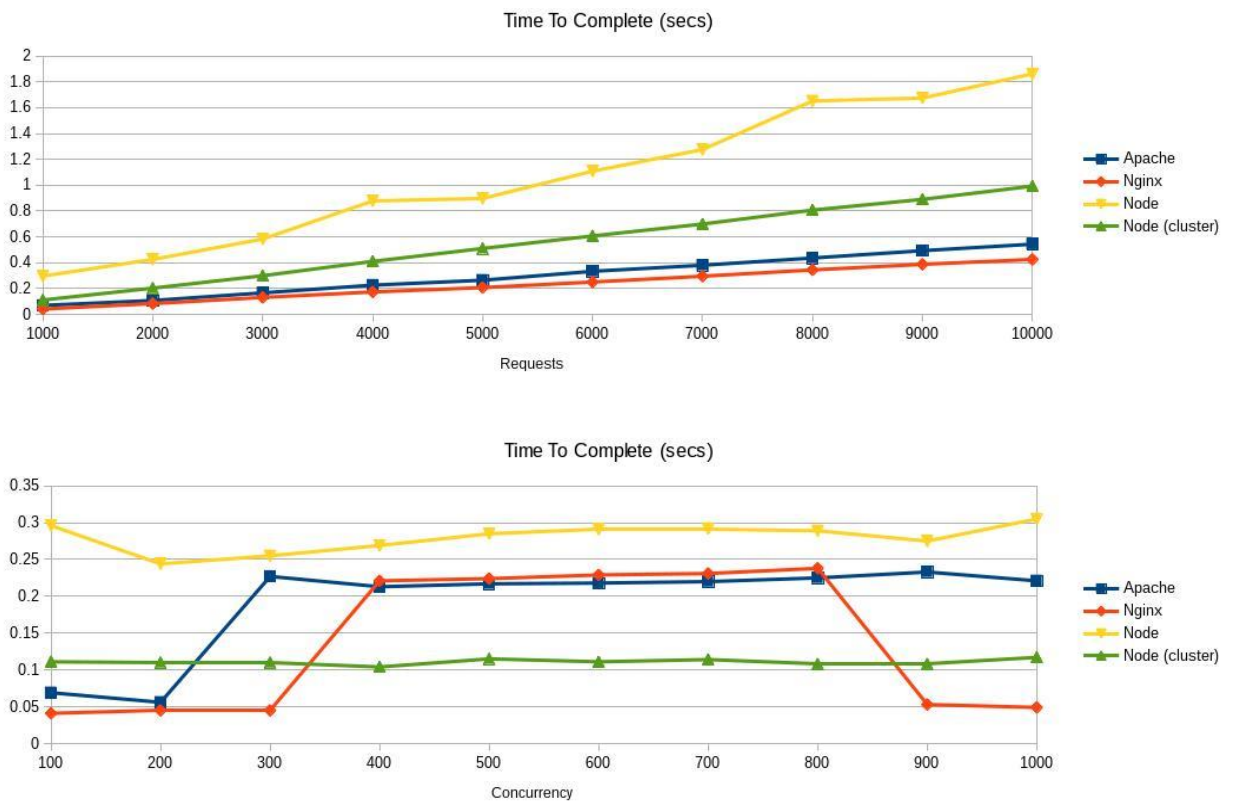
The tool used for benchmarking: ApacheBench, Version 2.3 <\$Revision: 1748469 \$>.

The tests included a series of benchmarks, starting from 1,000 to 10,000 requests and a concurrency of 100 to 1,000 – the results were quite surprising.

In addition, stress test to measure server function under high load was also issued.

As for the content, the main focus was about a static file containing a number of Lorem Ipsum verses with headings and an image.

Result



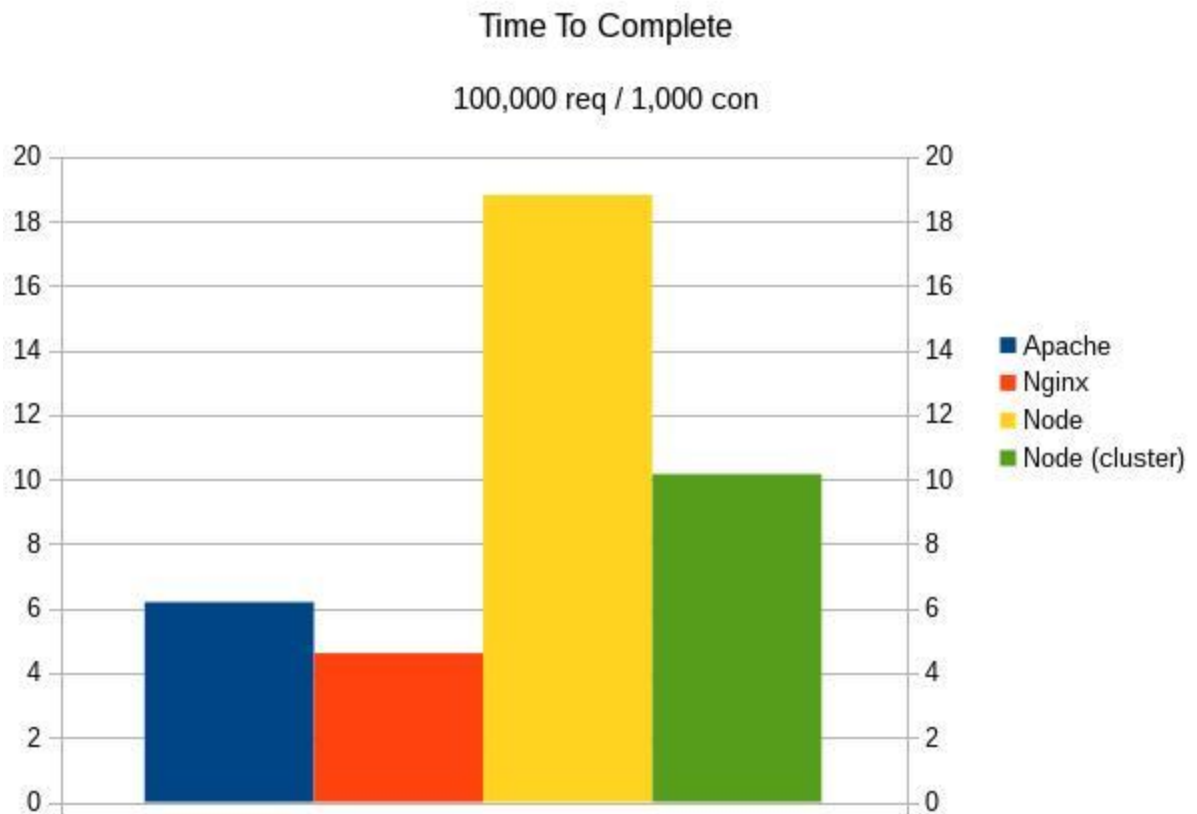


Fig.5. Performance comparison of Nginx, Apache, Node and Node with cluster

This result, which looks different from previous result, shows that Nginx is still the best server under high concurrency. The previous test was performed under relatively low concurrency and low request numbers, where NodeJS might outperform Nginx. But in a real enterprise level application like this test, Nginx is still among the best solutions.

Performance Comparison of Load Balancing

Setup

This test compare number of requests and memory usage of Node when running with Cluster module, iptables and Nginx as process load balancers.

Lets also compare how well these solutions distribute connections between workers/child processes.

Hardware & Software

Machine receiving the load is physical hardware with 8 CPU cores / 8GB RAM using Node.js v6.0.0.—cpuinfo

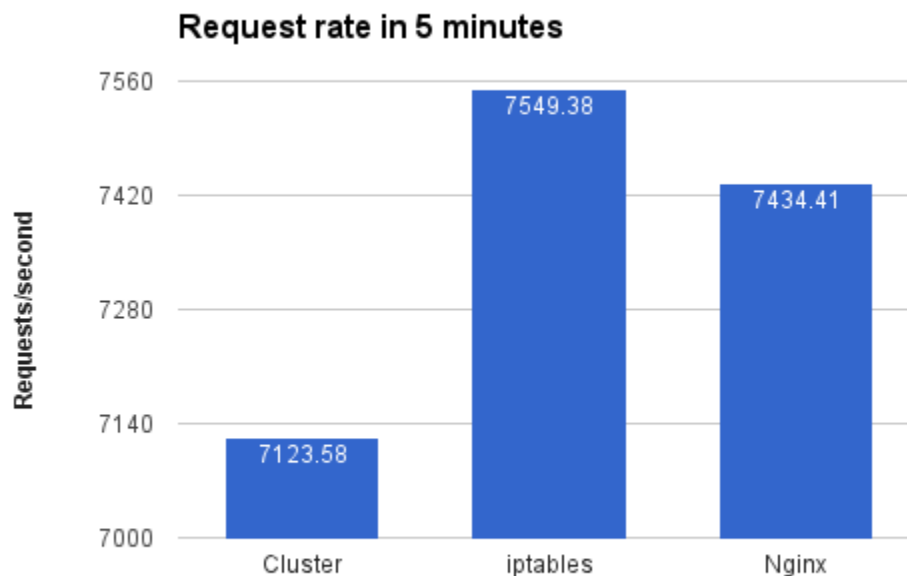
Machine generating the load has the same specs as above and is running Siege with this command and config

sysctl, ulimits and friends configured to do not be a bottleneck for the tests. The most important configs are these

iptables with nat and prerouting configuration to redirect from port 80 to Node.js workers on ports [8080–8087]

Nginx configured with 1 worker per core and upstream proxy pass to Node.js workers on ports [8080–8087]

Result



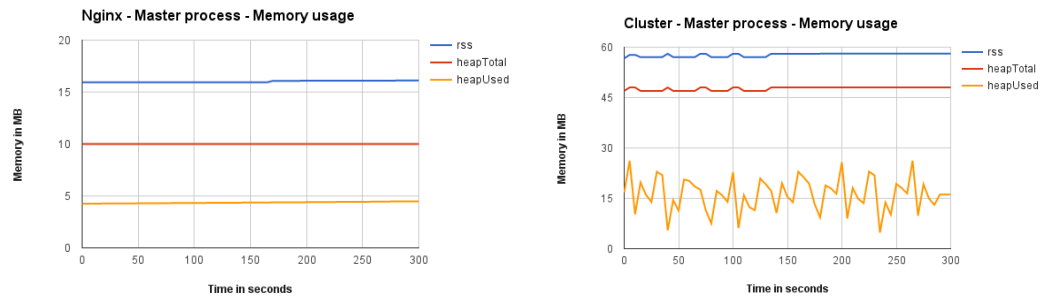


Fig.6. Load Balancer Performance of Nginx and NodeJS cluster

From the above result, we could observe that Nginx is a better load balancer in terms of request per second and memory occupation. While NodeJS cluster is a built in module in NodeJS and would work smoothly with JavaScript backend logics, Nginx still outperform it as a load balancer to NodeJS servers. It maybe because Nginx is written in C and uses basic kernel APIs.

Summary

As can be observed from several comparison examples, Nginx and NodeJS are both very fast web servers. Because of their asynchronous nature and non-blocking I/O mechanism, they support high concurrency and serve static files really well.

NodeJS is a good platform with npm capable of handling various open-source packages, making it easy to collaborate with database, middleware and even front-end. It has the advantage of being known to most front-end engineers thus has a good eco environment. It saves a lot of work to do a full-stack application with high concurrency and low computation.

Nginx is a mature web server used in a large number of companies as reverse proxy, load balancer and even main server. It supports even higher concurrency than NodeJS. A typical way is to deploy Nginx as reverse proxy and Tomcat as backend server to process business logic.

Circumstances	Nginx	NodeJS
---------------	-------	--------

Low&Medium concurrency static file server	Worse	Better
High concurrency static file server	Better	Worse
Reverse proxy	Better	Worse
Add-on Blocks	Less	More
Package Management	Worse	Better
Load Balancing	Better	Worse
Heavy Service with lots of CPU computation	Worse	Worse

Table.2. Performance Comparison of Nginx and NodeJS

Based on discussions above, this work listed table 2 as the performance comparison between Nginx and NodeJS. Nginx is widely recognized as a good server for static files, so is NodeJS. But NodeJS can also handle dynamic pages quite well. So using Nginx as front-end server and NodeJS as backend server seems a good way to combine their advantages. In other situations, if one want to create a light service with high concurrency, NodeJS is definitely the best choice.

To conclude, Nginx and NodeJS are showing good potential of offering microservices-friendly applications. Nginx is also suitable for enterprise-level applications, playing a role of reverse proxy and load balancing. Nginx is increasing rapidly in market share and would be good choice for business sites. But NodeJS has a full-stack solution for applications of certain demands. And NodeJS can improve its performance with Javascript engines of higher efficiency. Although Nginx seems good at this moment, I believe NodeJS has a bright future.

References

[1]Netcraft Feb 2017 web server survey:
<https://news.netcraft.com/archives/2017/02/27/february-2017-web-server-survey.html>

[2] Node.js process load balance performance: comparing cluster module, iptables and Nginx: <https://medium.com/@fermads/node-js-process-load-balancing-comparing-cluster-iptables-and-nginx-6746aaf38272>

[3] Apache Vs Nginx Vs Node.js And What It Means About The Performance Of WordPress Vs Ghost: <https://iwf1.com/apache-vs-nginx-vs-node-js-and-what-it-means-about-the-performance-of-wordpress-vs-ghost/>