

# ENTENDENDO SERVIDORES WEB DE ALTA DEMANDA – NODE.JS

SOLER, L. H., NOGUEIRA, W. G. H., OLIVEIRA, T. M. B., SERAFIM, R. C.

## Introdução / Objetivo

Na última década pudemos notar que o número de usuários da Internet continua crescendo: saltou de aproximadamente 360 Milhões em 2000 para quase 2 Bilhões em 2010, o que representa um crescimento de 444,8%. Isso nos indica que cerca de 29% da população mundial em 2010 já tem acesso à Internet (INTERNET WORLD STATS, 2011).

Esse crescimento exagerado deve-se a uma série de fatores, dos quais podemos destacar:

- Recursos de Hardware e Software que estão em constante processo de evolução e amadurecimento além da diminuição dos custos de implantação de tecnologias para Internet;
- Crescimento vertiginoso da diversidade de serviços baseados em web como redes sociais, portais de notícias, servidores de e-mail, jogos, etc.
- Aumento da quantidade de dispositivos com suporte a conexão com a Internet, como celulares, computadores portáteis, tablets, e até eletrodomésticos;
- Melhoria nas tecnologias de acessibilidade tanto para usuários quanto para outros sistemas;
- Diminuição dos custos de acesso à Internet banda larga, expansão da área de cobertura dos provedores de acesso e aumento da velocidade das conexões.

O aumento do número de usuários da Internet exige que os provedores de serviços na web invistam em estruturas que possam suportar um volume imprevisível de transações, ou seja, que possam ser expandidas sem muitas complicações.

Sistemas que precisam responder uma quantidade de acessos muito grande estão presentes em redes sociais, em jogos, em servidores de download, em serviços do governo, nos Correios, etc.

Em todos esses casos, há uma preocupação com a escalabilidade dos servidores, sendo que cada um aplica um modelo com tecnologias distintas, tanto de hardware quanto de software. Esse tema foi escolhido, pois é notório que a questão da otimização de servidores é relevante no contexto atual.

Nesse estudo iremos analisar uma solução de servidor web de alto desempenho baseada no Node.js, comparando-a com uma solução padrão de mercado baseada em Apache HTTP Server e PHP. Será feito um teste prático de desempenho das duas soluções, ambas rodando em condições semelhantes e com suas respectivas configurações padrão.



## Materiais e Métodos

O ambiente de testes foi composto pelos seguintes recursos:

### Hardware:

- **Placa-Mãe** ASUS P5GC-MX/1333;
- **Processador:** Intel Core 2 Duo CPU E4600 @ 2.40GHz;
- **Memória:** 2048MB RAM; **Disco Rígido:** 30 GB;
- **Placa de Vídeo:** NVIDIA GeForce 8600 GT.

**Sistema operacional:** TurnKey Core versão 11.1-lucid-x86.

### Software:

- **make** versão 3.81-7ubuntu1;
- **g++** versão 4:4.4.3-1ubuntu1;
- **apache2** versão 2.2.14-5ubuntu8.4;
- **PHP5** versão 5.3.2-1ubuntu4.9;
- **libapache2-mod-PHP5** versão 5.3.2-1ubuntu4.9;
- **Node.js** versão 4.1.0;
- **ApacheBench** versão 2.3;
- **free** versão 3.2.8.



Dois testes comparativos foram realizados: desempenho (requisições por segundo) e uso de memória principal.

O teste foi realizado através da ferramenta *ApacheBench* (*ab*), que mede o desempenho de um servidor *web* através do envio de um dado número de requisições concorrentes.

Os parâmetros utilizados foram:

- **-n20000:** 20.000 (vinte mil) requisições no total;
- **-c1000:** 1.000 (mil) requisições concorrentes;
- **'http://127.0.0.1:80/':** Endereço do servidor.

O uso de memória do sistema foi medido através da ferramenta *free*, que indica a quantidade de memória principal utilizada pelo sistema em intervalos determinados.

Os parâmetros utilizados foram:

- **-s0.5:** Verificação a cada 0,5 (meio) segundo;
- **-c60:** 60 (sessenta) medições totalizando trinta segundos de teste.

O código utilizado para o teste com o Node.js foi:

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World');
}).listen(80);
```

O código utilizado para o teste com o Apache HTTP Server e com o PHP foi:

```
<?php
echo "Hello World";
```

## Resultados

A representação dos resultados dos testes foi feita da seguinte maneira:

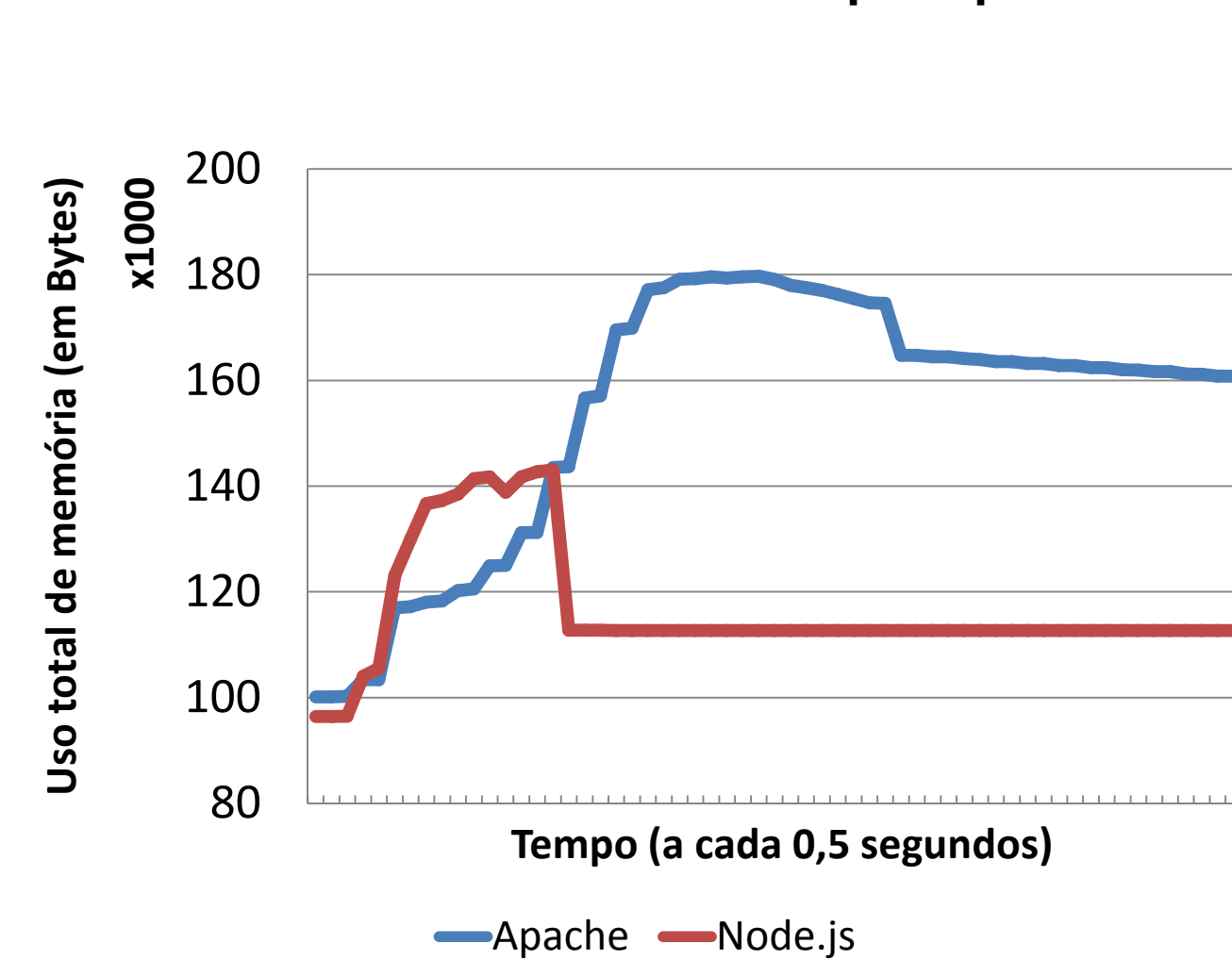
**Testes de desempenho:** Os resultados gerados pela ferramenta ApacheBench traduzidos e tabulados para facilitar a comparação (Tabela 1);

**Testes de memória:** Os dados dos testes de memória representam todas as ocorrências da quantidade de memória principal em uso pelo sistema durante os testes (Gráfico 1).

Tabela 1 – Teste de desempenho

	Node.js	Apache HTTP Server
Tamanho do documento	11 bytes	11 bytes
Tempo total do teste	5,835 s	15,970 s
Falhas	0	849
Total de dados transferidos	1.500.000 bytes	4.379.172 bytes
Dados HTML transferidos	220.000 bytes	216.986 bytes
Requisições por segundo (média)	3427,75	1252,38
Tempo por requisição (média)	0,292 ms	0,798 ms
Taxa de transferência	251,06 Kb/s	267,79 Kb/s

Gráfico 1 - Uso de memória principal



## Discussão e Conclusão

Devemos destacar que existem diferenças no funcionamento de cada um dos servidores utilizados nas soluções de teste, o que influenciou nos resultados dos testes.

Em relação ao desempenho, pudemos notar que o Node.js obteve resultados melhores em relação ao Apache HTTP Server, tanto no uso de memória principal quanto no número de requisições que esse pode suportar simultaneamente.

Em relação à programação, o código do Node.js implementa o servidor *web* como um método do módulo *"http"*. Esse método é usado para manipular as requisições manualmente, com poucas abstrações implícitas.

Já o Apache HTTP Server apresenta várias abstrações e otimizações por padrão. Isso agiliza o desenvolvimento do *software* na maioria dos casos, mas a implantação dessas características consome recursos do sistema. Abstrações tornam a produtividade inversamente proporcional ao desempenho.

Em relação à disponibilidade, O Node.js é um software recente e dificilmente encontramos provedores de hospedagem *web* que tenham suporte a ele. A combinação de Apache HTTP Server e PHP já é mais comum, sendo encontrada em boa parte dos provedores de hospedagem, e a preços acessíveis.

Em relação à documentação, O JavaScript é uma linguagem de programação padronizada pela ECMA e pela ISO, e é utilizada para a programação no Node.js. É uma linguagem muito difundida por ser utilizada para programação *client-sided* em páginas da *web*, e sempre foi criticada por implantar recursos problemáticos e com comportamentos imprevisíveis.

A linguagem PHP não é padronizada por nenhuma norma estrita.

Embora as duas linguagens tenham problemas específicos, a curva de aprendizado de ambas é muito baixa se comparada à de outras linguagens mais formais como C++ e Java.

Concluimos que o Node.js é uma escolha relevante quando se trata da resolução de problemas em que o desempenho é uma característica crítica da aplicação.

Utilizá-lo pode não ser interessante quando se trata de uma aplicação mais simples, pois a falta de abstrações diminui a produtividade desnecessariamente.

Para esse caso, a combinação do Apache HTTP Server com PHP é mais recomendada, pois ela se adapta a uma grande parte dos casos em que se precisa criar aplicações para *web*.

Também não podemos descartar a possibilidade de se integrar as duas soluções, utilizando-as em conjunto e delegando operações adequadas às características de cada tecnologia.

## Referências

Flórida: Wikimedia Foundation, **Node.js**. <<http://en.wikipedia.org/wiki/Nodejs>>, acesso em 9 de mai de 2011.

HIRATA, R., **Otimizando Servidores Web de Alta Demanda**. 2002. 202 p. Mestrado em Ciência da Computação – UNICAMP, Campinas, 2002.

Internet World Stats, **World Internet Usage and Population Statistics**. Disponível em: <<http://www.internetworldstats.com/stats.htm>>, acesso em 20 de abr de 2011.

KEGEL, D., **The C10k Problem**. Disponível em: <<http://www.kegel.com/c10k.html>>, acesso em 20 de abr de 2011.

LEITE, E. M., **Introdução ao Node.js**. In: CAMPUS PARTY, 4, 2011, São Paulo. Disponível em: <<http://www.youtube.com/watch?v=XPjogYLUNg>>, acesso em 2 de mai de 2011.

PHP Documentation Group, **Manual do PHP**. Disponível em: <[http://br.php.net/manual/pt\\_BR/preface.php](http://br.php.net/manual/pt_BR/preface.php)>, acesso em 9 de mai de 2011.

The Apache Software Foundation, **Apache HTTP Server**. Disponível em: <[http://projects.apache.org/projects/http\\_server.html](http://projects.apache.org/projects/http_server.html)>, acesso em 9 de mai de 2011.

TIOBE Software, **TIOBE Programming Community Index for May 2011**. Disponível em: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>, acesso em 2 de mai de 2011.

W3Techs, **Usage of web servers for websites**. Disponível em: <[http://w3techs.com/technologies/overview/web\\_server/all](http://w3techs.com/technologies/overview/web_server/all)>, acesso em 9 de mai de 2011.