NODE.JS

ESTUDO TECNOLÓGICO E DESENVOLVIMENTO FULL-STACK JAVASCRIPT DE PLATAFORMA DE COMPETIÇÕES EM PROBLEMAS ALGORÍTMICOS

ALUNO: GUSTAVO STOR DE AGUIAR
ORIENTADOR: FERNANDO DA FONSECA DE SOUZA

Recife, Abril de 2015

SUMÁRIO

1. CONTEXTO

1.1 UM POUCO SOBRE A WEB

A World Wide Web é, de longe, o serviço mais utilizado da Internet. Através dele, conseguimos obter conteúdo e nos conectar a pessoas em poucos milissegundos. Todo esse conteúdo e essas pessoas a quem nos conectamos estão disponíveis em páginas na Web, e a ordem de grandeza dessas páginas existentes é a mesma que a quantidade de pessoas no mundo [1]. É como se cada pessoa do mundo fosse responsável por uma página virtual. A importância da World Wide Web nos dias de hoje é de tamanha magnitude que o desenvolvimento de aplicações Web passou a ser um dos conhecimentos mais requisitados no mercado de trabalho em tecnologia de informação.

Mas tudo é feito de escolhas, e nem sempre são fáceis. Desenvolver um servidor na Web começa sempre com uma escolha: que tecnologia usar? Um iniciante poderia escolher a que provê um conjunto de ferramentas ou frameworks que facilite o desenvolvimento; um desenvolvedor mais experiente poderia optar por uma que suporte mais conexões paralelas, ou que seja otimizada no gerenciamento de requisições. Existem inúmeras linguagens de programação com suporte a desenvolvimento Web, algumas com seus respectivos frameworks que auxiliam no desenvolvimento: Django, em Python; Rails, em Ruby; JSP; PHP; ASP.NET. A escolha da tecnologia utilizada depende muito do desenvolvedor, do(s) objetivo(s) e da determinação do desenvolvedor em alcançar o(s) determinado(s) objetivo(s).

O estado da arte do desenvolvimento Web é uma linha turva, pois dizer o que é mais apropriado depende muito, como já foi dito, dos requisitos. Não muito longe disso, porém, é comum querer comparar tecnologias em determinados aspectos específicos. “Qual é a melhor tecnologia a utilizar quando queremos otimizar o tempo de resposta de uma requisição?”, “Qual é a tecnologia mais propícia para iniciantes?”, “Que tecnologia podemos usar para desenvolver um servidor Web que facilite a persistência de conexão cliente-servidor?” ou “É possível uma única instância de servidor aguentar 10 mil conexões paralelas?” são perguntas comuns e que podemos,

neste caso, responder precisamente qual ou quais tecnologias disponíveis se encontram no estado da arte para o determinado requisito.

O último ponto levantado – uma implementação de servidor Web que consiga manter 10 mil conexões paralelas – é frequentemente revisitado em pesquisas e levantamentos das tecnologias que melhor atendem as demandas de grandes servidores em produção. O problema C10K [2] relata justamente isso: o gargalo não é mais o hardware, pois com uma máquina 2000MHz, com 4GB de memória RAM e uma placa de rede de 1000Mbits/s, dividindo esses recursos para 10000 clientes, disponibilizaríamos, em um simples cálculo baseado na distribuição uniforme dos recursos, 200KHz de processamento, 400KB de memória virtual e uma conexão de 100Kbits/s para cada usuário, o que pode ser bastante suficiente para a maioria das páginas na Web. Mas o mundo não é tão ideal assim, e nem tudo depende do hardware. Chegamos em um ponto em que o problema maior é desenvolver o software que consiga ser o mais eficiente possível na distribuição dos recursos. Numa era em que 5 segundos de espera é muito tempo, otimização dos recursos pode ser a melhor arma para quem quer se sobressair.

O problema que mencionamos foca no lado do cliente, mas desenvolvimento Web também deve se importar com o lado do desenvolvedor. Nesse âmbito, existem práticas que podem facilitar muito o desenvolvimento. Primeiramente, a tecnologia escolhida deve dispor de um framework de desenvolvimento que seja continuamente mantido por uma equipe ou comunidade de desenvolvedores – dependendo de sua natureza privada ou pública. Esse aspecto é fundamental no mundo da computação, pois códigos se tornam obsoletos muito rapidamente. Erros que comprometem a segurança são encontrados a todo instante e as tecnologias devem acompanhar a passo rápido essas mudanças e correções. Outro fundamental pré-requisito é que a tecnologia empregada disponha de técnicas atuais de programação. Desenvolver um servidor Web em Perl, como era comum no fim da década de 90, seria um pesadelo nos dias de hoje [3]. Por fim, muitos defendem a ideia de que o desenvolvimento se torna mais prático e modular quando utilizamos a mesma tecnologia para desenvolver o front-end e o back-end do servidor Web.

Tendo em vista todos os problemas levantados, uma nova tecnologia foi proposta e implementada há poucos anos, e a

comunidade de desenvolvedores dela está cada vez maior – recentemente chegou a ultrapassar o Ruby on Rails. Focada em desenvolvimento de servidores Web em JavaScript, a plataforma Node.js foi implementada em cima do mecanismo V8, desenvolvido pelo Google [4]. V8 é um mecanismo altamente otimizado de JavaScript que funciona como um compilador JIT (Just-In-Time) e transforma o código alto-nível em código de máquina [5]. Assim como o Google Chrome, que roda em cima do mecanismo V8, o Node.js incorporou essa ***engine*** do Google e desenvolveu uma plataforma para desenvolvimento Web, que trás consigo todos os benefícios de JavaScript no lado do servidor, e todos os benefícios da rapidez do V8 como base estrutural da tecnologia.

Os benefícios de JavaScript no servidor vão muito além do escopo de desenvolvimento prático, como a reutilização de códigos modulares no cliente e no servidor. JavaScript é uma linguagem monolítica, movida a eventos, que consegue realizar operações não-bloqueantes e, com isso, simular paralelismo. Por ser monolítica, só temos um processo single-thread rodando a todo instante, então não devemos nos preocupar com gerenciamento de múltiplas threads. Por ser não-bloqueante, qualquer operação de entrada ou saída em JavaScript desbloqueia instantaneamente o "processo" (ou *mensagem*, nome mais apropriado) e processa a próxima requisição da fila. Quando a operação de entrada ou saída retornar um resultado, este é colocado no fim da fila, e eventualmente será processado. Como a maioria das operações de servidor são funções de requisição ou escrita, o processamento de cada mensagem se dá extremamente rápido e um cliente que requisitou poucos Kbytes de informação não precisará esperar por outro que requisitou algo bem maior um pouco antes dele. Por isso, é comum dizer que JavaScript é movido a ***Event loop***. Sem sombra de dúvidas, é uma ótima alternativa a se considerar na hora de começar um novo projeto.

1.2 UM POUCO SOBRE COMPETIÇÕES ALGORÍTMICAS NA WEB

No ensino fundamental e médio existem várias olimpíadas que as escolas incentivam o aluno a participar: OBM (Olimpíada Brasileira de Matemática), OBF (Olimpíada Brasileira de Física), OBQ (Olimpíada Brasileira de Informática), para citar alguns. Poucos

sabem, porém, que existe a OBI – Olimpíada Brasileira de Informática. E não para aí, existe a IOI, que é a Olimpíada Internacional de Informática, a ICPC, que é para universitários, e várias outras competições virtuais e algumas presenciais, mas apenas em outros países, como a USACO (USA Computing Olympiad). Poucos sabem as oportunidades que surgem com uma medalha nessas competições, que são tão bem vistas por grandes empresas tecnológicas.

O mundo digital está tão em vigor hoje em dia que não é nada improvável que, muito em breve, jovens aprendam a programar na escola. Mas enquanto esse dia não chega, a Web é repleta de oportunidades de preparação para essas competições. Enquanto que na Rússia e na China os alunos são mentorados desde cedo e são informados de várias oportunidades que sobressaem os assuntos do ensino fundamental e médio, no Brasil não há incentivo algum para o aluno, tanto por falta de divulgação das oportunidades quanto por falta de conteúdo em português para estes jovens que decidem aprender.

Com isso, muitos talentos são perdidos e, os poucos que conseguem se engajar nessas atividades durante a graduação, acabam por ter pouco tempo para nivelar o conhecimento que jovens de outras nações acumularam durante os anos de treinamento. Isso se reflete, por exemplo, no desempenho do Brasil na ICPC (*Internacional Collegiate Programming Competition*), que nunca obteve uma colocação entre os 10 primeiros. Enquanto isso, países do leste Europeu, da Ásia e Estados Unidos se revezam entre os primeiros colocados.

Oportunidades de aprendizado na Web são o que não falta. O Codeforces, plataforma russa, mas com idioma também disponível em inglês, promove competições regulares que, atualmente, contam com uma média de 8 mil competidores por competição. O TopCoder, que não é só uma plataforma de competição, mas também as promove, também é em inglês. Essas competições fornecem um conjunto de questões, e o participante deve escrever um programa de computador em uma linguagem de sua preferência (geralmente C, C++ ou Java) que resolva esse problema. Uma série de recursos de gamificação são aplicados a estas competições que as deixam mais emocionantes e iterativas, como a possibilidade de desafiar outras pessoas.

2. PROPÓSITO

O trabalho proposto será dividido em duas etapas. A primeira etapa será um estudo aprofundado da plataforma Node.js, suas vantagens e comparações com outras tecnologias em diversos aspectos, através das fontes mais atualizadas na literatura de desenvolvimento Web. Será abordado não somente o fluxo de desenvolvimento de aplicações Web em alto-nível por meio dessa tecnologia, como também o processo de execução a nível de plataforma – o que normalmente é omitido do desenvolvedor. Este estudo minucioso do comportamento da plataforma a baixo-nível será extremamente importante para a compreensão das vantagens e desvantagens do uso dessa tecnologia com relação a outras opções disponíveis. Além disso, ficará mais intuitivo a interpretação da análise de dados feita em cima da aplicação desenvolvida na segunda etapa.

A segunda etapa é uma intersecção entre o escopo teórico, que já terá sido abordado, e o escopo prático. Será desenvolvida uma aplicação em Node.js utilizando as melhores práticas de desenvolvimento para essa plataforma específica, como o uso, por exemplo, de MEAN [6] – MongoDB, Express.js, Angular.js e Node.js, que são um conjunto de ferramentas que se encaixam bem para um desenvolvimento Web produtivo. O escopo teórico entra à medida em que explicamos a implementação dos diversos componentes que irão compor a aplicação desenvolvida, bem como as decisões técnicas que tomamos durante o desenvolvimento.

A aplicação desenvolvida será uma plataforma brasileira de competições de problemas algorítmicos. O objetivo dessa plataforma é disseminar a ideia de programação competitiva no Brasil, além de incentivar jovens ainda não ingressados no ensino superior a cursar computação. A maior parte das grandes empresas de tecnologia seguem uma abordagem nas entrevistas, tanto de estágio quanto de emprego, de medir o conhecimento do aluno de computação através de muitos desses problemas que são comumente vistos em competições de programação – o que é por si só um grande incentivo para participar. Além disso, uma plataforma competitiva em português, e com competições periódicas e exclusivas também em nossa língua nativa é algo bastante requisitado por competidores brasileiros e que deve atrair

olhares não só destes, mas também de jovens que desejam iniciar sua jornada em computação.

Através do estudo detalhado da plataforma sugerida, com uma implementação eficiente e detalhada de uma aplicação utilizando essa tecnologia, este trabalho terá cumprido seu principal objetivo: incentivar desenvolvedores a questionar, medir requisitos, pesquisar e procurar alternativas antes de começar seus projetos. Afinal, desenvolvimento de sistemas de informação é uma via de mão dupla: de um lado, vai o conhecimento e a determinação; do outro, volta a recompensa.

3. Cronograma

Tarefas	Abril			Maio				Junho				Julho			
Revisão Bibliográfica	■	■	■	■	■										
Desenvolvimento da aplicação				■	■	■	■	■	■	■	■				
Testes e Análise de Performance				■	■	■	■	■	■	■	■	■	■		
Elaboração do relatório final								■	■	■	■	■	■		
Preparação da defesa								■	■	■	■	■	■	■	
Defesa														■	

4. Referências

[1] **KUNDER, Maurice De**. World Wide Web Size: daily esimated size of the world wide web. Disponível em: http://www.worldwidewebsize.com. Acesso em: 25 de Abril de 2015.

[2] **KEGEL, Dan**. The C10K problem. Disponível em: http://www.kegel.com/c10k.html. Acesso em: 25 de Abril de 2015.

[3] **MYHRVOLD, Conor**. The Fall Of Perl, The Web's Most Promising Language. Disponível em: http://www.fastcompany.com/3026446/the-fall-of-perl-the-webs-most-promising-language. Acesso em: 25 de Abril de 2015.

[4] **MARDAN, Azat**. Practical Node.js: Building Real-World Scalable Web Apps. Apress, 2012.

[5] **WILSON, Chris**. Performance Tips for JavaScript in V8. Disponível em: http://www.html5rocks.com/en/tutorials/speed/v8/. Acesso em: 26 de Abril de 2015.

[6] MEAN: Full-Stack JavaScript Using MongoDB, ExpressJS, NodeJS and AngularJS. Disponível em: http://mean.io/. Acesso em: 26 de Abril de 2015.

5. Possíveis Avaliadores

- Ana Carolina Brandão Salgado
- Márcio Lopes Cornélio
- Patrícia Tedesco
- Paulo Cunha
- Paulo Gonçalves

6. Assinaturas

Fernando da Fonseca de Souza
Orientador

Gustavo Stor de Aguiar
Aluno