Implantação em Nuvem de um Portal Web para Execução Remota de Algoritmos de Computação Científica

Lucas Ferreira da Silva[1], **Andrea S. Charão**[1]
Renata S. R. Ruiz[2], **Haroldo F. Campos Velho**[2]

[1] Universidade Federal de Santa Maria (UFSM)

[2]Instituto Nacional de Pesquisas Espaciais (INPE)

Resumo. *Este trabalho apresenta a implantação em nuvem de um portal Web que disponibiliza algoritmos de computação científica para execução remota. O portal é construído com um framework que utiliza Django para a aplicação Web e Celery como gerenciador de execução de tarefas. A implantação é em nuvem pública, utilizando a plataforma Heroku. Emprega-se configurações diferentes daquelas oficialmente documentadas para a modalidade gratuita do serviço, permitindo disponibilizar um portal completo de forma rápida, com limitações.*

1. Introdução

Pesquisadores da área de computação científica costumam conjugar esforços para tornar seus resultados (dados e software) disponíveis na Web. No âmbito de uma cooperação entre INPE e UFSM, desenvolveu-se um novo algoritmo para classificação de objetos astronômicos [Madalosso et al. 2016a] e, como forma de compartilhar esse resultado com pesquisadores da área, surgiu a ideia de disponibilizá-lo para execução remota, utilizando modernas tecnologias de desenvolvimento Web. Essa motivação levou ao desenvolvimento de uma solução mais genérica: um *framework* para construção de portais Web que permitam a execução remota de algoritmos de computação científica [Madalosso et al. 2016b]. Sua principal contribuição é prover uma solução simples para disponibilizar novos algoritmos, sob forma de programas executáveis remotamente.

O *framework* em questão foi aplicado à construção de um portal hospedado nas instituições dos próprios autores. Nesse processo, no entanto, há uma forte dependência dos setores responsáveis pela gestão de TI das instituições, que devem autorizar e habilitar acessos a serviços de rede necessários ao funcionamento do portal. Tomou-se, então, por hipótese, que implantar o portal em uma nuvem pública poderia ser uma alternativa para abreviar o processo, aproveitando que atualmente existem muitos provedores de computação em nuvem que permitem hospedar aplicações Web. Por outro lado, estima-se que esta solução implique em restrições que precisem ser avaliadas.

Assim, no presente trabalho, desenvolveu-se uma implantação em nuvem pública de um portal Web utilizando o *framework* proposto por [Madalosso et al. 2016b]. Nesta experiência, observou-se algumas particularidades que, estima-se, sejam úteis para pesquisadores que desejem reutilizar o *framework* ou que estejam confrontados com problemas semelhantes ao disponibilizar seus resultados. No restante deste texto, apresentam-se mais informações sobre o *framework* usado na construção do portal (seção 2), para depois apresentar o processo de implantação em nuvem (seção 3) e discutir os resultados obtidos (seção 4).

2. Arquitetura da Aplicação Web

O *framework* em questão implementa funcionalidades para os seguintes tipos de usuários:

- Usuário **anônimo**: pode visualizar informações sobre o portal e seu(s) algoritmo(s) e solicitar acesso como usuário registrado;
- Usuário **registrado**: pode executar experimentos no portal, selecionando o algoritmo desejado e, se necessário, enviando dados de entrada. Pode acompanhar monitorar o estado da execução de seus experimentos e obter os dados de saída;
- Usuário **administrador**: pode cadastrar novos algoritmos e realizar quaisquer outras operações no banco de dados do portal.

Para implementar tais funcionalidades, são empregados dois tipos de servidores (ver Figura 1): um servidor denominado *front-end*, que executa um servidor Web, e um ou mais servidores de execução, chamados *workers*. No servidor *front-end*, utiliza-se o *framework* Django para implementar os formulários de interação com os usuários e para orquestrar a interação com os *workers*. Estes, por sua vez, utilizam Celery, um gerenciador de execução de tarefas assíncronas e distribuídas. Entre esses servidores, ocorrem transferências de arquivos com dados de entrada e saída, conforme necessário.

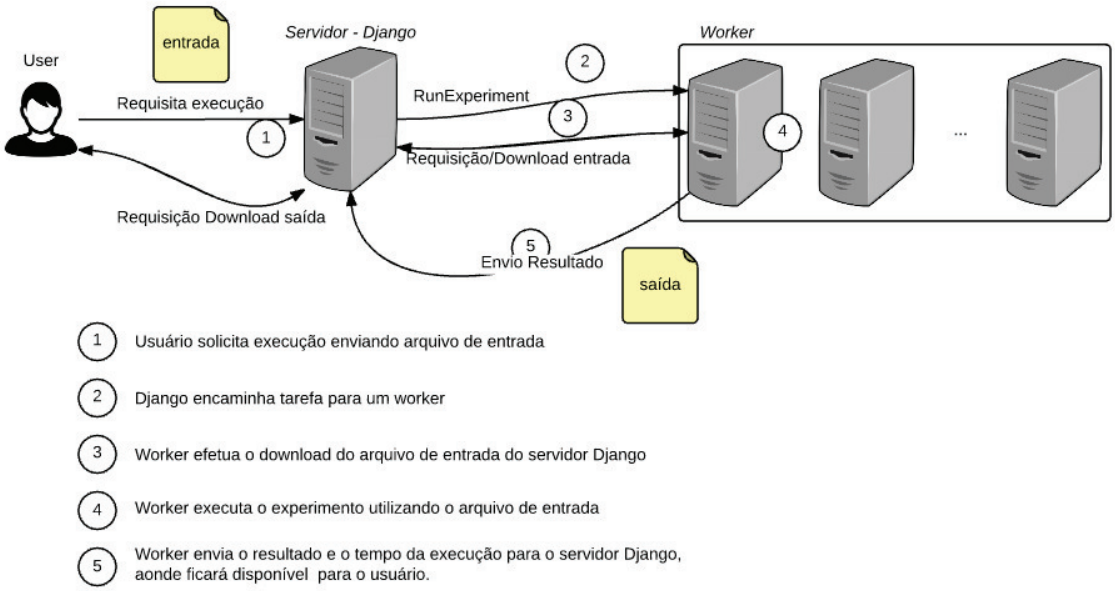

Figura 1. Arquitetura da aplicação Web [Madalosso et al. 2016b]

Para implantar uma aplicação Web usando este *framework*, é necessário instalar todas as dependências de software nos servidores e configurar os serviços com parâmetros de rede (URLs e portas) e credenciais do administrador. Isso provê as funcionalidades básicas descritas acima, que podem posteriormente ser estendidas.

3. Implantação em Nuvem

3.1. Escolha e Características do Serviço

Para a aplicação em questão, pode-se usar serviços em nuvem na modalidade IaaS (*Infrastructure as a Service*) ou PaaS (*Platform as a Service*). Exemplos de provedores na modalidade IaaS incluem Amazon EC2 e Microsoft Azure. Como PaaS, tem-se Heroku e Google App Engine, por exemplo. Os dois primeiros exemplos em IaaS já foram alvo

de investigações no contexto de aplicações de computação científica e de alto desempenho [Alves e Drummond 2014, Silva e de Oliveira 2016].

No presente trabalho, optou-se por explorar uma alternativa na modalidade PaaS, menos abordada em trabalhos envolvendo computação científica. Para isso, escolheu-se a plataforma Heroku, que oferece suporte a aplicações desenvolvidas em Django. Esta plataforma possui uma modalidade gratuita, assim como Amazon EC2, porém com a vantagem de não exigir que se informe um número de cartão de crédito para fins de verificação – o que é conveniente para instituições públicas de ensino e pesquisa.

Heroku implementa um modelo de execução baseado em contêineres virtualizados, chamados *dynos*, que oferecem o ambiente necessário para executar uma aplicação. Pode-se usar um ou mais *dynos* para uma única aplicação, distribuindo um ou mais serviços sobre eles (servidor Web, de banco de dados, de execução de tarefas, etc.). Na modalidade gratuita, dispõe-se de um *dyno* Web, que recebe tráfego HTTP, e um *dyno* para outros tipos de processos. Para ativar este segundo *dyno*, é necessário informar um número de cartão de crédito, embora não seja efetuada cobrança.

Para implantar aplicações na nuvem usando Heroku, deve-se fornecer o código-fonte da aplicação em um repositório Git, contendo uma descrição de suas dependências e um arquivo `Procfile`, que descreve os comandos a serem executados para lançar processos em cada tipo de *dyno*.

3.2. Configurações para Implantação

Inicialmente, procurou-se seguir as instruções para implantação de aplicações desenvolvidas em Python e Django [Salesforce.com 2016a], contidas na documentação oficial da plataforma Heroku. Uma etapa fundamental nestas instruções é sobre a obtenção da lista de dependências (`requirements.txt`), que compreende as bibliotecas e outros artefatos a serem automaticamente instalados na nuvem, para o correto funcionamento da aplicação.

No `Procfile`, pensou-se inicialmente em usar um *dyno* Web para o *front-end* e colocar o Celery no segundo *dyno*. Este esquema segue o que é sugerido na documentação oficial [Salesforce.com 2016b]. No entanto, a ativação do segundo *dyno* exigiria vinculação de um cartão de crédito. Então, buscou-se alternativas em outras fontes e descobriu-se uma forma de especificar um `Procfile` capaz de executar todos os serviços em um único *dyno*. Esta solução não é recomendável do ponto de vista de desempenho, mas mostrou-se suficiente para uma implantação com fins de demonstração.

O programa com o algoritmo a ser executado remotamente foi colocado em formato binário executável no repositório da aplicação. Foi necessário configurar uma nova localização para os arquivos com dados de entrada e saída, que são considerados conteúdos estáticos e devem estar em pastas específicas, para que possam ser encontrados pelo servidor Web na plataforma Heroku. Por fim, outro aspecto crítico foram as configurações para uso de um serviço externo de envio de e-mail, necessário para o cadastro de novos usuários, que recebem um e-mail com instruções para ativação de suas contas.

4. Resultados e Discussão

O portal implantado em nuvem encontra-se operacional e disponível em `http://demofof.herokuapp.com`. Conforme esperado, o processo de implantação foi abreviado na etapa de configurações de rede, que são realizadas automaticamente e permitem

disponibilizar uma aplicação em poucos segundos. Uma mesma etapa poderia levar vários dias, seguindo formalidades internas de setores de TI de grandes instituições. As etapas anteriores, referentes à configuração do `Procfile`, foram as que tomaram mais tempo, mas esta configuração pode ser reutilizada para abreviar implantações semelhantes.

Neste trabalho, a principal restrição observada foi a limitação de recursos na modalidade gratuita da plataforma Heroku, que não favorece a execução de aplicações de alto desempenho. De fato, execuções que se alonguem podem ser interrompidas por ultrapassar os limites da modalidade[1]. Em modalidades pagas, resta investigar se o desempenho seria suficiente para aplicações de computação científica.

5. Considerações Finais

Neste trabalho, explorou-se a plataforma de computação em nuvem Heroku, para implantar um portal Web voltado à execução remota de algoritmos de computação científica. Conseguiu-se realizar uma implantação pouco trivial, distante das configurações tipicamente utilizadas e oficialmente documentadas. Embora limitado pelos recursos da modalidade gratuita, o portal implantado pode ser utilizado para fins de demonstração. Em trabalhos futuros, pode-se explorar outros serviços em nuvem na modalidade PaaS para esta mesma aplicação, que satisfaz uma demanda recorrente entre pesquisadores de computação científica.

Referências

Alves, M. M. e Drummond, L. (2014). Análise de desempenho de um simulador de reservatórios de petróleo em um ambiente de computação em nuvem. In *Anais do 15º Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD-SSC'14)*.

Madalosso, O. M., Charão, A. S., de Campos Velho, H. F., e da Rocha Ruiz, R. S. (2016a). Multicore and many-core architectures in a parallel O(N log N) friends-of-friends algorithm for astronomical object classification. In *Proceedings of the 4th Conference of Computational Interdisciplinary Science*. Pan-American Association of Computational Interdisciplinary Sciences.

Madalosso, O. M., Charão, A. S., de Campos Velho, H. F., e da Rocha Ruiz, R. S. (2016b). A web portal framework for remote execution of high performance applications in astronomy. In *Proceedings of the 4th Conference of Computational Interdisciplinary Science*. Pan-American Association of Computational Interdisciplinary Sciences.

Salesforce.com (2016a). Deploying Python and Django apps on Heroku. Disponível em: `https://devcenter.heroku.com/articles/deploying-python#django-applications-on-heroku`.

Salesforce.com (2016b). Using Celery on Heroku. Disponível em: `https://devcenter.heroku.com/articles/celery-heroku`.

Silva, A. L. e de Oliveira, T. B. (2016). Avaliação de escalabilidade de aplicações de alto desempenho em núvem pública e privada. In *Anais do 17º Simpósio em Sistemas Computacionais de Alto Desempenho (WSCAD-SSC'16)*.

[1]Modalidades de preço da plataforma Heroku: https://www.heroku.com/pricing