

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**UM SISTEMA WEB PARA EXECUÇÃO
REMOTA DE APLICAÇÕES DE ALTO
DESEMPENHO**

TRABALHO DE GRADUAÇÃO

Otávio Migliavacca Madalosso

Santa Maria, RS, Brasil

2015

UM SISTEMA WEB PARA EXECUÇÃO REMOTA DE APLICAÇÕES DE ALTO DESEMPENHO

Otávio Migliavacca Madalosso

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da
Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para
a obtenção do grau de

Bacharel em Ciência da Computação

Orientadora: Prof^a. Dr^a. Andrea Schwertner Charão

Santa Maria, RS, Brasil

2015

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**UM SISTEMA WEB PARA EXECUÇÃO REMOTA DE APLICAÇÕES DE
ALTO DESEMPENHO**

elaborado por
Otávio Migliavacca Madalosso

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:

Andrea Schwertner Charão, Dr^a.
(Presidente/Orientadora)

silva non se sabe, Prof^a. Dr^a. (UFSM)

incognito anon, Prof. Dr. (UFSM)

Santa Maria, 08 de Outubro de 2015.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

UM SISTEMA WEB PARA EXECUÇÃO REMOTA DE APLICAÇÕES DE ALTO DESEMPENHO

AUTOR: OTÁVIO MIGLIAVACCA MADALOSSO

ORIENTADORA: ANDREA SCHWERTNER CHARÃO

Local da Defesa e Data: Santa Maria, 08 de Outubro de 2015.

Algumas áreas de pesquisa utilizam constantemente algoritmos que demandam alto desempenho dos seus ambientes de execução. Ocasionalmente, surgem algoritmos novos, com diferentes propriedades, que se propõem a resolver um problema de forma mais eficiente e/ou completa. Infelizmente, é comum que esses algoritmos fiquem restritos a ambientes institucionais, limitando muito a sua visibilidade para a comunidade de pesquisa. Este trabalho tem como objetivo criar um portal que permita ao usuário solicitar a execução remota de um algoritmo de acordo com as configurações que o sistema oferecer.

Palavras-chave: Computação de alto desempenho. Programação Web. !!!!!!!!!!!!!!!1. !!!!!!!!!!!!!!!3. !!!!!!!!!!!!!!!2.

SUMÁRIO

1 INTRODUÇÃO.....	6
1.1 Objetivos.....	6
1.1.1 Objetivo Geral.....	6
1.1.2 Objetivos Específicos	6
1.2 Justificativa	7
2 FUNDAMENTOS E REVISÃO DE LITERATURA.....	8
2.1 New Zeland eScience Infrastructure - NeSI	8
2.2 MediGRID	8
2.3 Framework Django	9
2.4 Friends of Friends	9
3 DESENVOLVIMENTO.....	11
3.1 Ambiente de Desenvolvimento	11
3.1.1 Sistema Operacional	11
3.1.2 Ferramentas de Programação	11
3.1.3 Third Party Apps	11
3.1.4 Controle de Versão - Repositório	12
3.2 Organização do Projeto	12
3.2.1 <i>manage.py</i>	12
3.2.2 <i>Static Files</i>	12
3.2.3 <i>Media url</i>	12
3.2.4 <i>views.py</i>	13
3.3 Modelos de dados	13
3.4 Funcionalidades.....	13
3.4.1 Usuário Anonimo	13
3.4.2 Usuário registrado	13
3.4.3 Administrador.....	14
3.5 Tasks	14
3.6 Design	14
3.7 Administração	14
4 PRÓXIMAS ETAPAS.....	15
REFERÊNCIAS	16

1 INTRODUÇÃO

Algoritmos com grande custo computacional são facilmente encontrados em áreas como meteorologia, biologia e astronomia. Esses algoritmos possuem a característica de utilizar um nível elevado de processamento para concluir sua execução, e consequentemente, seus tempos de execução podem variar dependendo da máquina aonde estão sendo executados.

É comum pesquisadores destas e de outras áreas desenvolverem novas implementações de algoritmos utilizados por seus colegas. Implementações essas que podem trazer muitos benefícios para outros pesquisadores que necessitam deste tipo de solução. Infelizmente, é comum essas implementações ficarem restritas a ambientes privados, não por questões de licença, mas simplesmente pela ausência de um método prático para disponibilizar a nova ferramenta ao público.

Com isso, surge a ideia de desenvolver um portal web que permita ao usuário o cadastro de um experimento, no qual ele poderá ditar os dados de entrada desse experimento, e qual algoritmo (disponível no sistema) ele deseja utilizar para processar os dados. Depois de requisitar o experimento, o sistema deve providenciar sua execução e quando finalizar, retornar o resultado do experimento ao usuário que o requisitou.

Para este portal, será utilizado um algoritmo desenvolvido para a área de astronomia, uma versão do algoritmo Friends-of-Friends de complexidade $n \cdot \log(n)$ paralelizada através do framework OpenMP.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo deste trabalho é criar um portal web que possibilite aos usuários cadastrados no sistema executar algoritmos presentes no sistema segundo suas configurações e disponibilizar o resultado da execução após o término da mesma.

1.1.2 Objetivos Específicos

- Estudo de frameworks web para ser utilizado no desenvolvimento.
- Desenvolvimento front-end do servidor.

- Administração das execuções requisitadas.
- Atualização dos estados das requisições no sistema.

1.2 Justificativa

O projeto é capaz de gerar benefícios significativos para a comunidade de pesquisa de diversas áreas, criando um modelo de ambiente que facilite a divulgação e teste de resultados de algoritmos alternativos para resolução de problemas comuns.

Além de servir como modelo, o projeto disponibilizará um algoritmo que se enquadra na categoria alvo do projeto: a versão de complexidade $n \cdot \log(n)$ e paralela do friends-of-friends.

2 FUNDAMENTOS E REVISÃO DE LITERATURA

Os trabalhos citados a seguir se relacionam com a proposta deste trabalho pois compartilham características como:

- Garantir ao usuário a capacidade de fazer seus experimentos em um ambiente que está além da capacidade normal de processamento de um equipamento pessoal.
- Disponibilizar softwares já configurado para o ambiente aonde será executado.
- Disponibilizar soluções para problemas que não são oferecidas ao público geral.

2.1 New Zeland eScience Infrastructure - NeSI

O NeSI(NEW ZELAND ESCIENCE INFRASTRUCTURE, ????) provê plataformas de grande capacidade computacional para auxiliar pesquisadores na Nova Zelândia. Atualmente eles possuem 5 ambientes disponíveis em diferentes instalações. Cada um desses ambientes possui hardware e software capazes de resolver problemas relacionados a áreas de pesquisa, especialmente relacionados a química e bioinformática.

Um desses ambientes por exemplo é o FitzRoy, que dispõe de várias ferramentas para o ambiente de programação como compiladores para as linguagens C/C++, Fortran e Python, além de *debuggers*, ferramentas de *profilling*. Além disso também conta com bibliotecas úteis que auxiliam em questões como entrada e saída de dados e problemas matemáticos. Fora da área de programação, o ambiente oferece também aplicações de simulações que requerem alto desempenho do hardware.

Cada um dos ambientes disponíveis possui seu próprio site institucional com sua respectiva documentação e formas de contato, além disso o próprio NeSI oferece suporte a usuários através da sua equipe.

2.2 MediGRID

O MediGRID (MEDIGRID, ????) é um portal focado em pesquisa na área da bio-medicina, ele oferece aplicações e infraestrutura para os pesquisadores cadastrados no portal realizarem experimentos conforme a sua necessidade e a disponibilidade do sistema.

O objetivo principal do MediGRID quando foi desenvolvido era ser uma plataforma de integração middleware ligando serviços eScience com as pesquisas de biomedicina. Porém hoje ele executa tarefas em 3 áreas majoritárias: Biomedicina, processamento de imagens e pesquisa clínica.

2.3 Framework Django

Django(DJANGO PROJECT, ????) é um framework para criação de aplicações web que encoraja o desenvolvimento ágil, em alto nível e com design pragmático. Foi criado inicialmente para manter o portal de notícias online do Lawrence Journal World pelos programadores Adrian Holovaty, Simon Willison e Jacob Kaplan-Moss.

Por se tratar de um portal inicialmente desenvolvido para administrar notícias, o framework lida muito bem com gerenciamento de conteúdo e agilidade quando é necessário fazer alterações no sistema.

2.4 Friends of Friends

Simulações de N-corpos têm sido utilizadas para promover vários avanços na compreensão de questões relevantes em astrofísica, como por exemplo o processo de formação e evolução de estruturas do Universo. Este tipo de simulação tem um papel fundamental(BERTSCHINGER, 1998; G. EFSTATHIOU M. DAVIS, 1985) no estudo da evolução cósmica em tópicos como a distribuição de matéria escura em grande escala, a formação de halos de matéria escura, e a formação e evolução de galáxias e aglomerados.

A manipulação e análise da grande quantidade de dados produzidos em tais simulações também é algo desafiador. Neste contexto, é essencial o desenvolvimento de técnicas computacionais eficientes para extrair informação significativa a partir dessas fontes de dados, em um período apropriado de tempo.

Etapas importantes neste tipo de análise são a identificação de halos de matéria escura e o estudo do espectro da energia potencial gravitacional de tais objetos. Uma abordagem para este tipo de análise consiste em usar o algoritmo de percolação Friends-of-Friends (FoF) (HUCHRA J. P., 1982) . A ideia básica deste algoritmo é a seguinte: considere uma esfera de raio R ao redor de cada partícula do conjunto total; se dentro desta esfera existirem outras partículas, elas serão consideradas pertencentes ao mesmo grupo e serão chamadas de amigas.

Em seguida, toma-se uma esfera ao redor de cada amiga e continua-se o procedimento usando a regra "qualquer amigo de meu amigo é meu amigo". O procedimento para quando nenhuma amiga nova puder ser adicionada ao grupo.

Na 15ª edição da Escola Regional de Alto Desempenho do Rio Grande do Sul (XV ERAD-RS)(referenciar), foram publicados resultados de execuções de uma nova implementação do Friends-of-Friends(OTÁVIO MIGLIAVACCA MADALOSSO, 2015) cuja complexidade computacional era reduzida em relação as versões anteriores, resultando em redução do tempo de processamento.

3 DESENVOLVIMENTO

Aqui serão descritas as atividades realizadas para atingir os objetivos propostos. Essas atividades serão distribuídas em sessões de acordo com o segmento do projeto que mais se relacionam, sendo eles:

- Ambiente de Desenvolvimento
- Organização do Projeto
- Modelos de dados
- Funcionalidades
- Design
- Administração

3.1 Ambiente de Desenvolvimento

3.1.1 Sistema Operacional

Para o desenvolvimento do projeto está sendo utilizado o sistema operacional Ubuntu 14.04(REF?), que contém um interpretador da linguagem Python(REF?), sendo necessário apenas fazer a instalação do framework Django 1.8(REF?) para ter um ambiente pronto para começar o desenvolvimento do projeto. Além disso, também foi instalado o sistema de gerenciamento de pacotes pip(REF), que foi utilizado para instalação de outras ferramentas (descritas abaixo) que auxiliaram no desenvolvimento.

3.1.2 Ferramentas de Programação

Para auxiliar na tarefa de programação do sistema, foi utilizado o editor de texto Sublime Text 2 (REF!), um blabla.... utilizando também o plugin Anaconda... Django algo...

3.1.3 Third Party Apps

Durante o decorrer do desenvolvimento observou-se que o projeto poderia se beneficiar de outras aplicações django. Django-registration-redux 1.2 foi a ferramenta utilizada para lidar

com a tarefa de registro de usuários novos. Django Crispy foi utilizada para criar e fazer a verificação de formulários que o usuário deve preencher para executar algum algoritmo ou alterar algum dado de seu perfil. Celery é o

Falar dos pacotes que foram utilizados até agora, (Redux registration, Crispy, celery json-django)

3.1.4 Controle de Versão - Repositório

Para realizar o controle de versões foi utilizado o *Git*¹, também utilizado pelo projeto original.

3.2 Organização do Projeto

O framework Django tem uma proposta de criar aplicações fáceis de serem reutilizadas por outros projetos, e por conta disso tem uma abordagem bastante metódica e pouco flexível em relação a distribuição dos arquivos que pertencem ao projeto. Consequentemente, a organização utilizada é a mesma proposta pela documentação do framework, que pode ser observada na figura abaixo (incluir figura)

3.2.1 *manage.py*

explicar a funcionalidade do manage e sua importancia para todo o projeto

3.2.2 *Static Files*

Os arquivos estáticos que o projeto mantém são todos organizados dentro do diretório "static in pro", lá são mantidos arquivos que precisam estar disponíveis por qualquer documento html, em grande maioria são arquivos .css, .js e imagens comuns às páginas do projeto.

3.2.3 *Media url*

Media URL é a variável responsável por indicar ao sistema qual diretório será a raiz dos arquivos que serão manipulados pelo sistema. Dentro desse diretório raiz existe uma pasta "Users" e dentro dela, uma pasta com o id de cada usuario cadastrado no banco de dados. Para cada experimento criado por um usuário, em seu respectivo diretório será criado um novo dire-

¹ O repositório com o código pode ser encontrado em <https://github.com/Madalosso/TG>

tório cujo nome será o identificador do experimento, e dentro desse diretório serão mantidos os arquivos de entrada e saída do experimento.

3.2.4 *views.py*

Esse arquivo contém as funções que definem telas a serem exibidas para os utilizadores do sistema. Como padrão toda função nesse arquivo recebe com um *request* e retorna alguma resposta html, essa resposta na maioria dos casos utiliza arquivos presentes no diretório "Templates".

3.3 Modelos de dados

adicionar imagem de algum software mostrando as relações entre os tipos de dados?
explicar o que cada classe / atributo tem como objetivo

3.4 Funcionalidades

As funcionalidades do sistema podem ser divididas até agora em 3 grupos distintos.

3.4.1 Usuário Anônimo

Um usuário anônimo possui a permissão de acessar áreas de informação a respeito do sistema, de contato com o administrador do sistema, e da área de registro, onde pode solicitar o registro e, seguindo as orientações apresentadas, fazer login como um usuário registrado.

3.4.2 Usuário registrado

O usuário registrado tem permissão de criar experimentos no portal, para isso ele faz o *upload* de um arquivo que será utilizado como entrada no algoritmo selecionado. Além disso o usuário também pode monitorar o estado dos experimentos que ele requisitou e fazer o *download* dos arquivos de cada experimento, tanto o arquivo de entrada, como a saída(se houver) do algoritmo.

3.4.3 Administrador

O administrador tem as mesmas capacidades do que um usuário registrado e detem privilégios de acesso ao painel de administração do django. Isso permite a ele cadastro de novos algoritmos no sistema, de novos experimentos padrão e a editar qualquer informação que o sistema detenha no banco de dados.

3.5 Tasks

Escrever como foi feito a manipulação e controle de tarefas (Celery).

3.6 Design

Explicar uso do Bootstrap, jQuery, tentativa de usar ajax e o porque não funcionou(por hora)

4 PRÓXIMAS ETAPAS

Como continuação do trabalho realizado até o momento, são planejadas as seguintes atividades:

1. fazer a execução das tarefas em uma máquina diferente da que mantém o portal.
2. Avaliar novas idéias de funcionalidades e implementar as que forem validadas.
3. tornar site responsivo via bootstrap.

REFERÊNCIAS

- BERTSCHINGER, E. Simulations of structure formation in the universe. **Annu. Rev. Astron. Astrophys** **36**, [S.l.], 1998.
- DJANGO Project. Accessed: 2015-08-14, <https://www.djangoproject.com/>.
- G. EFSTATHIOU M. DAVIS, S. D. M. W. C. S. F. Numerical techniques for large cosmological N-body simulations. **Astrophysical Journal Supplement Series (ISSN 0067-0049)**, vol. **57**, [S.l.], 1985.
- HUCHRA J. P., G. M. J. Groups of galaxies. I - Nearby groups. **Astrophysical Journal, Part 1**, vol. **257**, June 15, 1982, p. 423-437. Research supported by Cambridge University, [S.l.], 1982.
- MEDIGRID. Accessed: 2015-08-14, http://www.medigrid.de/index_en.html.
- NEW Zeland eScience Infrastructure. Accessed: 2015-08-14, <https://www.nesi.org.nz/>.
- OTÁVIO MIGLIAVACCA MADALOSSO, A. S. C. Implementação do algoritmo Friends of Friends de complexidade $n \cdot \log(n)$ para classificação de objetos astronômicos. **ERAD-RS XV**, [S.l.], 2015.