André Spanguero Kanayama

Distribuição de carga em execução de scripts no sistema VisionDataset

Brasil

2013, v-0.02

André Spanguero Kanayama

Distribuição de carga em execução de scripts no sistema VisionDataset

Desenvolvimento de um método para a execução de scripts no VisionDataset, sistema cujo desenvolvimento já foi iniciado anteriormente.

Universidade de São Paulo – USP

Instituto de Matemática e Estatística

Orientador: Prof. Dr. Roberto Hirata Jr.

Brasil

2013, v-0.02

"Simple' does not mean 'easy'. I have learned
that the things that seem the simplest are often the most powerful of all."
(Christie Golden, Thrall: Twilight of the Aspects)

Sumário

Parte I

Objetiva

Introdução

Com o advento da Internet, cada vez mais usa-se ferramentas colaborativas on-line para o desenvolvimento científico. Com o objetivo de facilitar esta colaboração, surgiu o VisionDataset. Criado por Bruno Klava, este sistema tinha como objetivo centralizar as imagens utilizadas nos projetos desenvolvidos pelo grupo de pesquisa em visão computacional do IME-USP.

O sistema já está sendo utilizado, mas ainda possui muitos recursos que podem ser adicionados, inclusive a execução de scripts em python para o processamento das imagens armazenadas no banco de dados. Porém, a execução de scripts em imagens muito grandes pode durar muito tempo consumindo muitos recursos da máquina, o que pode prejudicar o desempenho para os outros usuários, caso seja simplesmente executado no mesmo servidor da aplicação, portanto seria necessário pensar em algum tipo de solução que melhorasse o desempenho e não afetasse o sistema.

Meu trabalho consiste em realizar a execução de scripts no VisionDataset de forma eficiente, para isso utilizarei outras máquinas, no princípio apenas virtuais que reportam para o servidor o seu estado, sendo assim ele pode escolher em qual delas, e como fará o processamento das imagens automaticamente, sendo que o usuário verá apenas o resultado final.

1 Conceitos e Tecnologias

1.1 Máquinas virtuais

Uma máquina virtual é um computador que funciona dentro de outro, através de software. A principal vantagem de uma máquina virtual é seu isolamento em relação ao servidor (também conhecido como *host*), sendo assim mesmo que aconteça algum problema com a máquina virtual, ainda é possível gerenciá-la remotamente. A máquina virtual não precisa ter o mesmo sistema operacional que o *host*, sendo assim temos também a opção de ter vários sistemas operacionais rodando simultaneamente em um computador só, diminuindo custos com infra-estrutura.

"As máquinas virtuais podem ser divididas em três tipos:

- Tipo 1 Sistema em que o monitor é implementado entre o hardware e os sistemas convidados (guest system).

- Tipo 2 Nele o monitor é implementado como um processo de um sistema operacional real, denominado sistema anfitrião (host system).

- Tipos Híbridos Os monitores de tipo 1 e 2 raramente são usados em sua forma conceitual em implementações reais. Na prática, várias otimizações são inseridas nas arquiteturas apresentadas, com o objetivo principal de melhorar o desempenho das aplicações nos sistemas convidados. Como os pontos cruciais do desempenho dos sistemas de máquinas virtuais são as operações de E/S, as principais otimizações utilizadas em sistemas de produção dizem respeito a essas operações.

Outra importante categoria de máquinas virtuais são as máquinas virtuais para computadores fictícios projetados para uma finalidade específica. Atualmente a mais importante máquina virtual desta família é a JVM (máquina virtual Java). Existem simuladores para ela em quase todos os computadores atuais, desde computadores de grande porte até telefones celulares, o que torna as aplicações Java extremamente portáveis. Uma importante vantagem sem duvida de se escrever código para uma máquina virtual é a de se poder compilar o código sem que seja perdida a portabilidade, melhorando-se a velocidade em relação à programação interpretada, que também é portátil, porém mais lenta, já que neste caso cada linha será traduzida e executada em tempo de execução, e no caso da máquina virtual cada mnemônico da máquina virtual é convertido no equivalente em linguagem de máquina (ou assembly) da máquina real."(http://pt.wikipedia.org/wiki/Máquina_virtual)

1.2 PostgreSQL

O PostgreSQL é um sistema gerenciador de bancos de dados objeto-relacional. Ele possui o código aberto e é capaz de rodar em vários sistemas operacionais, além de ser muito confiável, sendo um dos sistemas gerenciadores de bancos de dados mais usados no mundo. Sua versão mais recente é capaz de lidar com tabelas de até 32 TB sendo que em cada tabela podem haver até 1600 colunas.

1.3 Maven

O Maven é uma tecnologia desenvolvida pela Apache Software Foundation e que é utilizada para gerenciamento e compilação de projetos feitos na linguagem Java. Com ele é possível descrever as dependências de um projeto em um arquivo, chamado pom.xml, e quando a compilação for realizada, todas elas são verificadas e, caso não sejam satisfeitas, os pacotes necessários são baixados automaticamente, tornando a compilação e o compartilhamento do código muito mais fáceis. Além disso o Maven oferece inúmeras ferramentas para testes e documentação.

1.4 Aplicação Web

Uma aplicação web é um sistema projetado para ser utilizado através do navegador. Esse sistema fica em um servidor, no qual é feito o processamento das requisições, sendo assim é possível utilizá-lo à partir de qualquer computador, independentemente do sistema operacional e do hardware. Para entender o funcionamento da aplicação web, especialmente no caso do VisionDataset, é necessário conhecer algumas tecnologias:

1.4.1 Servidor HTTP Apache

"Servidor web pode ser um programa de computador responsável por aceitar pedidos HTTP de clientes, geralmente os navegadores, e servi-los com respostas HTTP, incluindo opcionalmente dados, que geralmente são páginas web, tais como documentos HTML com objetos embutidos (imagens, etc.) ou um computador que executa um programa que provê a funcionalidade descrita anteriormente."
(http://pt.wikipedia.org/wiki/Servidor_web)

O Apache é um servidor HTTP livre e atualmente é o mais utilizado no mundo. Com ele é possível redirecionar um endereço no *host* para a porta 8080 das máquinas virtuais, utilizada pelo Tomcat.

1.4.2 Apache Tomcat

Um servlet é uma classe Java utilizada para gerar páginas HTML dinâmicas. Ele é responsável por receber chamadas HTTP, processá-las e devolver uma resposta.

O Tomcat, também desenvolvido pela Apache Software Foundation, é um container de servlets, ou seja, gerencia o ciclo de vida, dá suporte ao multithread, segurança, e suporte para páginas. O Tomcat tem a capacidade de rodar integradamente com um servidor HTTP Apache.

1.4.3 JavaServer Pages (JSP)

JavaServer Pages é uma tecnologia criada em 1999 e que permite a criação de páginas web dinamicamente. Com ela é possível acessar o banco de dados no servidor e exibir a página de acordo com seu conteúdo. Para poder executar JSPs é preciso usar um container de servlets, como o Tomcat.

1.5 VisionDataset

Como define o criador do sistema, Bruno Klava: "Este projeto tem por objetivo a criação de uma base de imagens para o grupo de pesquisa em visão computacional do IME/USP. Tal base de imagens servir a para centralizar num mesmo local as imagens utilizadas pelos membros do grupo, facilitando a utilização das mesmas imagens nos trabalhos desenvolvidos pelo grupo e permitindo a comparação de resultados de diferentes técnicas".

O VisionDataset é um sistema web feito em Java e utiliza o sistema gerenciador de banco de dados PostgreSQL, e o framework Apache Struts. Com ele é possível criar e gerenciar usuários e álbuns de imagens, fazer o upload de arquivos em massa e criar anotações e *tags* em imagens. Posteriormente, Rafael de Oliveira Lopes Gonçalves fez algumas melhorias, entre elas a integração com o sistema de segmentação de imagens SegmentIt.

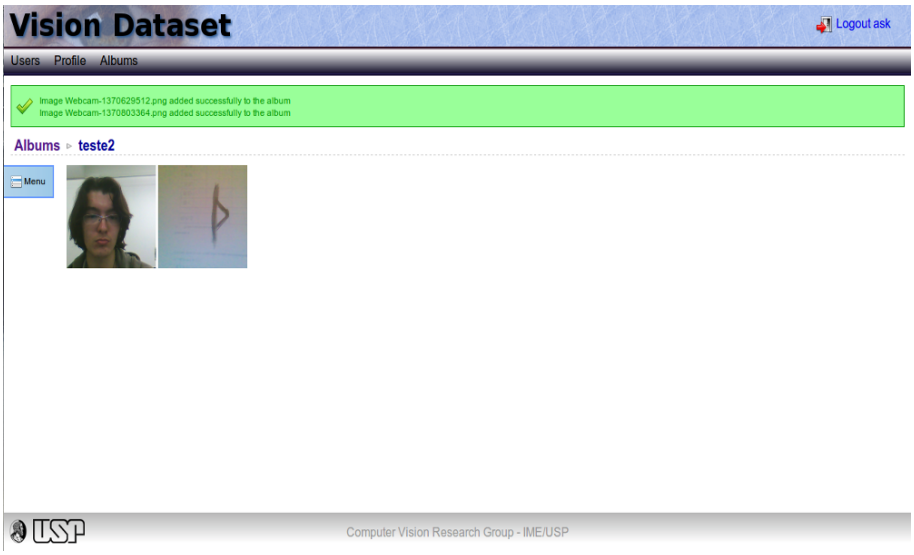

Figura 1 – Screenshot do VisionDataset

2 Atividades Realizadas

2.1 Pesquisa sobre ferramentas similares

No início do trabalho pesquisei tecnologias semelhantes ao VisionDataset para saber se há alguma com o mesmo propósito ou com recursos que poderia agregar. Estas tecnologias são:

- AdessoWiki
  Uma plataforma colaborativa para programação e escrita de documentos, focados no processamento de imagens. Possui um método para execução de scripts em python muito eficiente, porém não funciona como um repositório de imagens.

- Harvard Dataverse Network
  É um repositório de dados dedicado à colaboração científica. Porém não é voltado a imagens e não possui um sistema de edição e execução de scripts.

Além disso, revi algumas tecnologias que haviam sido pesquisadas previamente por Fábio Tsuguta Matsumoto e que na época não supriam as necessidades dos usuários do VisionDataset, como CAIMAN(CAncer IMage ANalysis), BASE (BioArray Software Environment), CARMEN (Code, Analysis, Repository and Modelling for e-Neuroscience), LabKey Server e MediGRID.

2.2 Configuração do ambiente

Para rodar e disponibilizar o VisionDataset no servidor da Rede Vision do IME-USP foi criada uma máquina virtual dedicada apenas à essa tarefa. Pude então instalar o Tomcat, o PostgreSQL (sistema gerenciador de banco de dados utilizado pelo VisionDataset) e outros pacotes usados pelo sistema. Com os pacotes instalados, configurei o Maven para usar as versões mais atuais do PostgreSQL e do Tomcat. Tendo tudo configurado na máquina virtual, tive então que configurar o Apache no servidor web físico para poder atribuir um endereço de acesso externo ao Tomcat da máquina virtual. Para tanto, foi feito um ProxyPass no Apache, adicionando-se as seguintes linhas ao arquivo de configuração na pasta *sites-available*:

```
ProxyPass /VisionDatasetdev http://192.168.231.207:8080/VisionDatasetdev
ProxyPassReverse /VisionDatasetdev http://192.168.231.207:8080/VisionDatasetdev
```

2.3 Correção no envio de e-mails

Tendo o sistema rodando no servidor, percebeu-se que ele não enviava e-mails para os usuários, o que não se pode ignorar, já que a criação de novos usuários depende da confirmação do e-mail deste. Percebi então que o sistema estava configurado para rodar no servidor antigo, utilizando o servidor de e-mails da própria Rede Vision. Fiz então uma modificação no código para que ele enviasse os e-mails utilizando uma conta no Gmail (serviço de e-mails da Google) dedicada apenas para esta tarefa, esta conta pode também servir como um canal de comunicação entre o adimistrador do sistema e o usuário.

As alterações foram feitas na classe:

`br.usp.ime.vision.dataset.util.Email`

2.4 Execução de scripts na linguagem Python através do sistema

Na página de imagens coloquei uma caixa para inserção de texto, ao digitar o script em Python a ação é executada no servidor onde está o VisionDataset e a saída é exibida na próxima página. Para que funcionasse, criei a classe br.usp.ime.vision.dataset.actions.ShowScript, que é chamada pelo arquivo /textitjsp correspondente para a exibição do resultado.

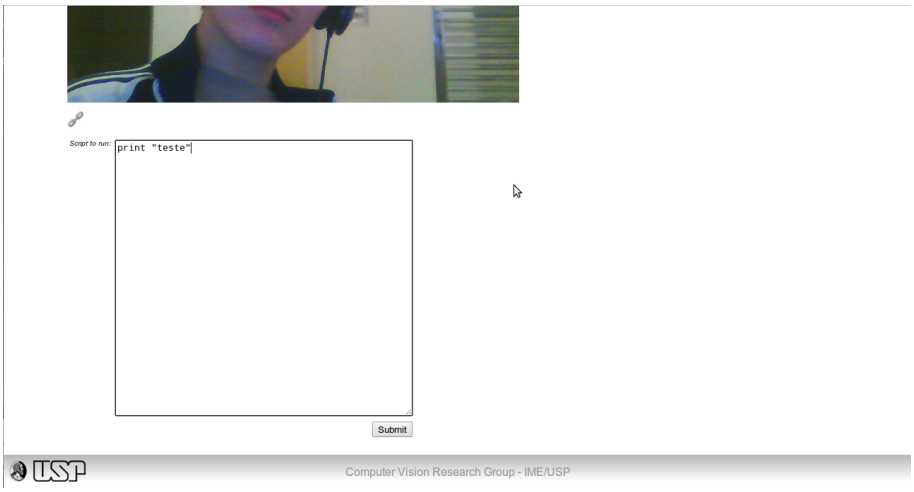

Figura 2 – Tela de edição do script.

2.5 Criação de daemon e script para comunicação entre máquinas

Criei um daemon em python, chamado machinestatusdaemon que utiliza sockets para se comunicar com as outras máquinas. Usando a biblioteca psutil consigo obter diversas informações sobre o uso da máquina, como, por exemplo, o uso dos processadores

e da memória. Minha intenção é colocar este daemon em cada máquina disponível para o processamento dos dados, assim o servidor ao executar um script consegue obter tais informações sobre o uso delas e assim ter a melhor decisão sobre como delegar o processamento.

2.6 Criação da estrutura para salvar os scripts no banco de dados

Crei as seguintes classes:

`br.usp.ime.vision.dataset.dao.impl.ScriptDAOImpl`
`br.usp.ime.vision.dataset.util.ScriptUtils`
`br.usp.ime.vision.dataset.servlet.ImagesServlet`
`br.usp.ime.vision.dataset.entities.ImageScript`
`br.usp.ime.vision.dataset.entities.Script`

Elas contêm os atributos e métodos necessários para a inserção e recuperação dos scripts no banco de dados.

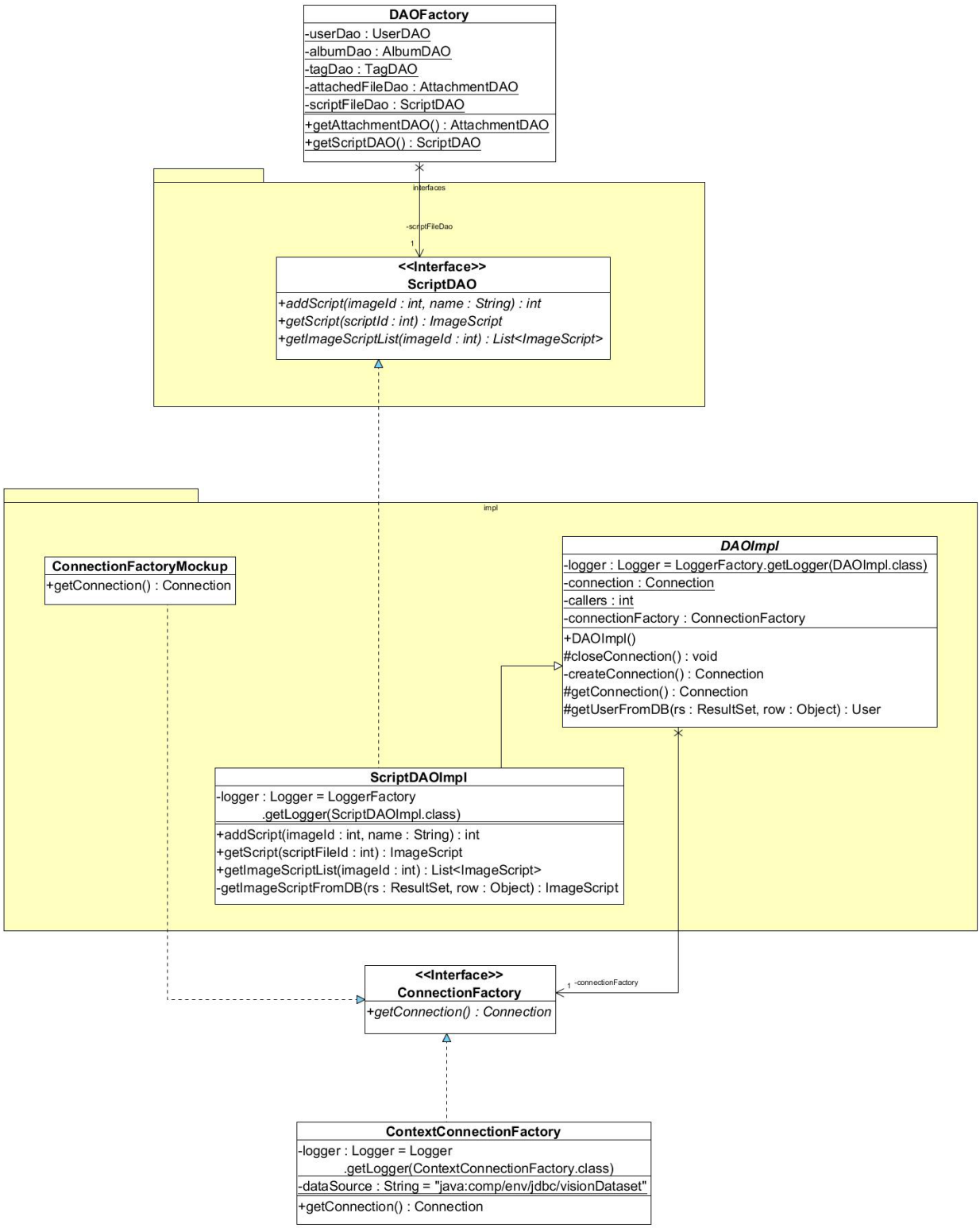

Figura 3 – Diagrama de classes da DAO dos scripts.

3 Próximas atividades

3.1 Interface para uso de scripts

O que estou fazendo agora é uma interface para que o usuário possa lidar com os scripts (editar, salvar, recuperar, etc.) através das páginas *jsp*.

3.2 Definir como distribuir o processamento

Decidir como escolher em quais máquinas rodar o processamento da imagem, e se possível, fazer esse processamento em mais de uma máquina, para economizar recursos e tempo. Para isso serão necessários alguns estudos para saber o que levar em conta na escolha das máquinas e também fazer alterações no modo como os scripts são executados. Além disso será preciso que todas as máquinas enxerguem o mesmo banco de dados.

3.3 Melhorar a interface e a usabilidade

Se sobrar tempo, pretendo melhorar a interface e a usabilidade do VisionDataset, bem como a própria execução de scripts, podendo inclusive adicionar mais opções de linguagem.

Conclusão

Referências Bibliográficas

- KLAVA, B. *Base de imagens do grupo de Visão Computacional do IME/USP*. 2010. 3 p. - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo.
- MATSUMOTO, F. T. *Ferramentas semelhantes ao VisionDataset*. 2012. 6 p. - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo.
- *Máquina Virtual*s. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2013. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Máquina_virtual&oldid=36412439>. Acesso em: 20 ago. 2013.
- *Servidor Web*. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2005. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Servidor_Web&oldid= Acesso em: 20 ago. 2013.
- *Servidor Apache*. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2013. Disponível em: <http://pt.wikipedia.org/w/index.php?title=Servidor_Apache&old Acesso em: 2 set. 2013.
- LOTUFO, R. A.; MACHADO R. C. *Adessowiki*. 2009. Disponível em: <http://adessowiki.fee.un Acesso em: 21 ago. 2013.
- *The Dataverse Network Project*. 2013. Disponível em: <http://thedata.org>. Acesso em: 21 ago. 2013
- Seshadri. G. *Understanding JavaServer Pages Model 2 architecture*. 1999. Disponível em: <http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>. Acesso em 12 set. 2013.
- *JavaServer Pages*. In: WIKIPÉDIA, a enciclopédia livre. Flórida: Wikimedia Foundation, 2013. Disponível em: <http://pt.wikipedia.org/w/index.php?title=JavaServer_Pages&o Acesso em: 12 set. 2013.
- *About PostgreSQL*, 2013. Disponível em: <http://www.postgresql.org/about/>. Acessado em: 12 set. 2013.
- *What is Maven?*, 2013. Disponível em: <http://maven.apache.org/what-is-maven.html>.Acessa em: 12 set. 2013.

Parte II

Subjetiva

4 Desafios e frustrações

5 Disciplinas relevantes

6 Próximos passos