Guia para configuração geral do servidor

Observações gerais

O servidor fará o papel de servidor web e de base para execução de tarefas de diferentes tipos. Finalmente, este guia assume que você tenha um conhecimento razoável de sistemas linux e saiba lidar com comandos como “sudo”, “cd”, “mkdir”, “mv”, “cp”, “rm”, etc.

Sistema Operacional

O sistema utilizado atualmente é o **Ubuntu Server 32-bit**, em sua versão **8.04 LTS** (*long term support*). A escolha pela versão 32-bit, mesmo com a máquina suportando 64-bit, se deve ao fato de que o **Java** tem performance pior em 64-bit, o que é indesejado para o site e outras ferramentas. Para testar versões LTS mais recentes, basta visitar o site do Ubuntu, em http://www.ubuntu.com/, baixar o ISO (imagem do CD) correspondente.

Pré-instalação

Para que a instalação seja mais eficiente e fácil, é necessário que o computador tenha acesso à internet durante a instalação, através de uma conexão de rede com DHCP (por exemplo, ligando o computador via cabo a um roteador). Com isso, o processo de instalação poderá fazer o download dos pacotes mais recentes, quando necessário e você também poderá instalar novos pacotes diretamente do terminal.

Instalação

O primeiro passo é inicializar a máquina com o disco de instalação do Ubuntu e seguir as orientações gerais do processo de instalação, **exceto** pelas especificidades que indico a seguir:

- **Partições**: esta é uma questão muito importante e é preciso *muito cuidado e atenção*. Os computadores, normalmente, já vêm com uma versão do Windows instalada e oficial. Embora vá ser pouco ou nunca utilizada, não queremos perder essa instalação. Assim, é necessário **particionar manualmente** o HD, para que a partição do Windows seja redimensionada para um tamanho menor (digamos, 30 a 50 GB) e as partições necessárias à instalação do Ubuntu sejam criadas. O esquema de partição que temos utilizado é um dos mais simples, sendo composto por *três partições*, sendo uma para **swap** (com o tamanho máximo igual ao dobro da memória RAM da máquina, até o limite de 6GB), outra do tipo **ext3** para a **raiz** (ponto de montagem é “/”, para arquivos do sistema em geral, máximo de 20GB é suficiente) e a última também do tipo **ext3** para os **arquivos de usuários e do site** (ponto de montagem “/home”, com o espaço restante do HD).
- **Serviços internet**: em um dado momento da instalação, pede-se ao usuário para informar quais serviços de internet quer instalar. Se for possível, marque os serviços de **DNS**, e-mail (**Mail**) e o **OpenSSH**. O tipo de perfil do serviço de email é “Servidor Internet”.

Pós-instalação

Após a instalação básica do sistema, serão necessárias alguns ajustes antes de prosseguir. As ações são:

1. Na versão *Server*, o ambiente gráfico não é instalado por padrão e temos que fazê-lo manualmente. Execute o comando `sudo apt-get install ubuntu-desktop`
2. Execute o comando a seguir e siga as instruções padrão quando pedidas: `sudo apt-get install kate perl libdate-calc-perl libpar-packer-perl expect expect-dev php5-dev python-dev python-xml python-wxglade python-setuptools tcsh g++ libdb1-compat sun-java6-bin sun-java6-jdk sun-java6-plugin apt-file apache2 mysql-server-5.0 php5 php5-mysql phpmyadmin php5-gd php-image-graph firestarter linux-kernel-devel apcupsd apcupsd-cgi gcc build-essential xserver-xorg-dev subversion grsync xinetd sshfs vsftpd startupmanager`
3. O sistema irá pedir algumas informações. São elas:
   - MySQL: informe a senha que preferir.
   - PHPMyAdmin: escolha todas as opções começando por "apache".
   - Java: concorde com as licenças.
4. Em seguida, execute `sudo apt-file update` para atualizar as informações de pacotes do sistema e/ou faça as atualizações automáticas, se o sistema indicar.
5. Reinicie a máquina (`sudo reboot`), para entrar no modo gráfico.

Driver da Placa de Vídeo (extra)

Em algumas máquinas com placa de vídeo NVidia é necessário compilar o driver, pois este não consta na distribuição do Ubuntu. Só faça isso se o sistema não instalar automaticamente, o que você perceberá pela qualidade da imagem (baixa resolução indica falta do driver específico). Os passos para a instalação do driver são:

1. Baixar o instalador do driver correspondente no site da NVidia (http://www.nvidia.com/page/drivers.html).
2. Execute `sudo apt-get install linux-headers-`uname -r` gcc gcc-3.4 build-essential xserver-xorg-dev`
3. É preciso sair do modo gráfico. Para isso, alterne para o modo texto (via `Ctrl+Alt+F1`) e execute: `sudo /etc/init.d/gdm stop` e depois `sudo killall X`.
4. Vá para a pasta em que o driver foi salvo e execute o comando `chmod 755 <nome-do-arquivo>`, para torná-lo executável.
5. Execute o instalador do driver. Se não der certo, execute `sudo update-rc.d -f gdm remove` e tente executar o instalador novamente. Neste caso, depois da instalação será necessário executar `sudo update-rc.d gdm defaults`. Se ainda não der certo, recorra ao Google.
6. **[atenção]** A cada atualização do kernel, talvez seja necessário reinstalar o driver, refazendo os passos acima.

Instalação de pacotes necessários para o site e demais ferramentas utilizadas no projeto

Agora podemos instalar os aplicativos e bibliotecas extras, necessárias. Com isto, teremos, entre outras coisas, o servidor web Apache, o PHP5, o Java 1.6, o MySQL, o Perl e o Python.

- Faça o download do arquivo http://search.cpan.org/CPAN/authors/id/E/EL/ELIJAH/Time-SoFar-1.00.tgz, acesse a pasta via terminal, descompacte o arquivo `Time-SoFar-1.00.tgz` e execute a série de comandos: `perl Makefile.pl`, `make` e `sudo make install`.
- **[extra]** Se você optou pela versão 64-bit do Ubuntu Server, então será necessário executar alguns passos extras:
  1. Baixe o arquivo http://ubuntuguide.net/wp-content/uploads/2012/06/getlibs-all.deb e instale `getlibs-all.deb` dando um duplo clique sobre ele ou através do comando (via terminal) `dpkg -i getlibs-all.deb`.

2. Em seguida, baixe os arquivos http://packages.ubuntu.com/hardy/i386/libgtk2.0-0/download e http://www2.frugalware.org/mirror/old-releases.ubuntu.com/ubuntu/pool/universe/d/db2/libdb2_2.7.7.0-10_i386.deb, instale com o comando `getlibs -i libgtk2.0-0_2.12.9-3ubuntu2_i386.deb libdb2_2.7.7.0-10_i386.deb`.
3. Se faltarem outras bibliotecas, na execução de algum aplicativo, utilizar o **ia32-libs** para ver dependências de aplicativos 32-bit. O comando é `ldd <aplicativo>`. Encontre as bibliotecas (online) e as instale com o comando **getlibs** acima.
4. Opcionalmente, pode-se tentar fazer isso de modo automático, com a busca e instalação das bibliotecas necessárias. Tente o comando `getlibs <aplicativo>`. Nem sempre é suficiente, porém.

Configuração do Ambiente

Algumas ações são necessárias para configurar o ambiente adequadamente.

Geral

1. Crie a pasta "`tools`" dentro da pasta "`/home`". Será necessário usar o modo super-usuário para isso; aproveite para dar as permissões adequadas para o usuário que será usado para rodar o parser (talvez baste dar a propriedade da pasta para esse usuário).
2. Descompacte o arquivo "`bin.zip`" dentro de `/home/tools`, de modo a ter `/home/tools/bin`.
3. Descompacte o arquivo "`dbparser.zip`" de modo a ter a pasta `/home/tools/dbparser`.
4. Para que os scripts possam ser executados globalmente (ou seja, em qualquer pasta no terminal), acrescente ao fim do arquivo `/etc/bash.bashrc` (utilize "sudo"):
   ```
   # User scripts
   export PATH=$PATH:/home/tools/bin"
   ```

Ações finais (opcionais)

- Apagar pastas temporárias que ainda restarem (após descompactar os pacotes instalados).
- Atualize o idioma do sistema, através do menu "Sistema->Adminstração->Suporte a Idiomas". Procure "Portugues" na lista de idiomas, marque sua opção e clique em "Ok".
- Reinicie e deixe o Ubuntu rodar a atualização automática de pacotes (se for o caso).

Guia para a análise sintática automática utilizando o parser do Bikel

Observações gerais

O objetivo deste documento é descrever como realizar, passo-a-passo, a análise sintática de textos etiquetados. Espera-se que o usuário tenha conhecimento básico sobre utilização de comandos Linux/Unix, tais como "cd", etc. Além disso, este guia conta com os scripts desenvolvidos para facilitar esta tarefa, o analisador de Dan Bikel, alguns aplicativos instalados na configuração da máquina (tais como o *tcsh*; ver o guia para Configuração do Servidor). Ademais, outras informações estão disponíveis nos guias específicos do parser, na pasta /home/tools/dbparser/userguide/.

Scripts necessários

/home/tools/bin/tb-backgroundParsing.pl
Este é o script que gerencia todo o processo de análise. A idéia deste script é tornar a análise mais eficiente e evitar que ela demore ou "quebre" (abandone a análise por algum erro imprevisto). Para isso, o script prepara o texto e o divide em partes, cada uma contendo uma sentença. O script então dispara análises para cada parte, controlando a finalização e o início de cada uma, a partir do controle dos processos disparados. O número de sentenças que são analisadas "em paralelo" é configurável, na linha de comando do script, como veremos adiante. Para funcionar, este script depende de outros, descritos a seguir.

/home/tools/bin/tb-preparseFromPOS.pl
Este script gera, a partir de um arquivo etiquetado (no formato "palavra/ETIQUETA"), um arquivo com formato apropriado (chamado "preparse") para submeter ao analisador de Dan Bikel.

/home/tools/bin/tb-underscoreSubTags.pl
Este script troca hífens por traços de sublinhado (e vice-versa), permitindo que o analisador considere o sistema completo de etiquetas (incluindo as subetiquetas separadas por hífen).

/home/tools/bin/tb-splitPreparse.pl
Este script separa o arquivo "preparse" em arquivos contendo 1 sentença cada. Por exemplo, se o texto possui 1000 sentenças, serão gerados 1000 arquivos, um para cada sentença.

/home/tools/bin/tb-assignFlatTree.pl
Este script gera uma estrutura sintática "flat", ou seja, completamente nivelada, com cada elemento aparecendo como filho de um único nó raiz. Ele é utilizado para sentenças que o analisador não consegue analisar ("quebra") ou que estão tomando muito tempo (o limite é configurável na linha de comando, por exemplo, 10 minutos por sentença). Nestes casos, mesmo que o analisador consiga produzir alguma saída, ela acaba sendo mais difícil de corrigir do que fazer a análise da sentença "do zero". Com essa estratégia, o processamento segue, sem muita perda de tempo.

/home/tools/bin/tb-fixPostParse.pl
Este script faz tarefas gerais de pós-análise, limpando certas "sujeiras" deixadas pelo analisador.

/home/tools/bin/tb-retrieveSubTags.pl
Este script recupera as sub-etiquetas das palavras, retiradas para que a análise automática possa ser

feita (o analisador de Dan Bikel não trabalha com elas).

/home/tools/bin/tb-adjust-POS.pl
Este script ajusta textos etiquetados, fazendo quebras de sentenças automaticamente (e que podem, eventualmente, ser equivocadas), entre outras adaptações (inclusão de ID para cada sentença, etc.). Caso o texto a ser analisado não tenha ainda passado por este script é preciso fazê-lo. Por exemplo, os textos etiquetados que fazem parte da Busca Web com CorpusSearch, no site do CTB, foram todos submetidos a este script.

/home/tools/bin/tb-rmID.pl
Este script é utilizado para remover as informações de ID de cada sentença, para que possam ser geradas novas identificações, se for o caso. Esta tarefa é feita manualmente, geralmente *após a revisão da análise sintática*. Outro uso para este script é preparar arquivos de treinamento para o analisador, *que não devem conter estas informações de ID*.

/home/tools/bin/tb-addID.pl
Este script gera identificações para cada sentença do texto analisado sintaticamente. Esta tarefa é feita manualmente, geralmente *após a revisão da análise sintática*.

Aplicativos e outros arquivos necessários

/home/tools/dbparser/
Pasta que contém o analisador sintático de Dan Bikel, arquivos de treinamento, arquivos de configuração para o processamento do Português, entre outros. Arquivos fundamentais para a análise de textos em português são:

settings/classicalport.properties (propriedades para textos em português)
train.obj.gz (treinamento do analisador baseado na anotação do Corpus Tycho Brahe)

Treinamento do parser

O parser de Bikel funciona com base em um "treinamento", ou seja, é preciso prepará-lo para a análise de novos textos, fornecendo textos já anotados sintaticamente. Em geral, quanto mais textos anotados fornecidos, melhor. Além disso, é muito importante que o corpus de textos anotados esteja *consistente*, ou seja, que as análises estejam corretas e coerentes em todos.

Atualmente, no CTB, o parser foi treinado com um corpus composto por mais de 500 mil palavras (/home/tools/dbparser/train_003sub.psd), a partir dos textos já anotados do corpus. O parser foi então treinado, gerando o arquivo /home/tools/dbparser/train_003sub.obj.gz. Portanto, atualmente não há necessidade de retreinar o parser, exceto se você for utilizar outro sistema de anotação. Sugere-se aguardar uma boa quantidade de novas sentenças, por exemplo, acima de 2 mil, antes de retreinar o parser. Isto não é uma regra absoluta, mas pela nossa experiência (que foi pequena, diga-se), retreinar o parser com pouca quantidade de novas sentenças pode levar a um resultado pior nas análises, como pudemos verificar em alguns testes.

Para ver qual arquivo de treinamento está sendo utilizado, em qualquer momento, abra uma janela de terminal, vá para a pasta /home/tools/dbparser/ e execute ls -l train.obj.gz. Você verá para qual arquivo de treinamento este link está apontando no momento. Sempre que gerar um novo arquivo de treinamento, basta apontar o link simbólico para ele.

Enfim, para treinar o parser, siga os seguintes passos:

1. Preparação do corpus de treinamento:

i. Junte todos os textos anotados em um só arquivo (através de um editor como o *Kate* ou similar, compatível com UTF-8).
ii. É preciso retirar os "ID" dos textos. Abra uma janela de terminal, vá para a pasta onde está o arquivo e execute:

    tb-rmID.pl nome-do-arquivo-único > train_999.psd.

    Sendo "999" a numeração consecutiva à última utilizada ("001", "002", etc.). Isso irá gerar um novo arquivo, sem os "ID". O arquivo com os "IDs" pode ser apagado. Cuidado para não sobrepor um arquivo de treinamento anterior, pois pode ser necessário continuar a utilizá-lo, caso a performance do analisador caia com o novo treinamento.
iii. Mova este arquivo criado para a pasta /home/tools/dbparser/.

2. Para treinar o parser, abra um terminal, vá para a pasta /home/tools/dbparser/ e execute:

    bin/train 500 settings/classicalport.properties ctb_train_999.psd.

    Aguarde o processamento. Se tudo funcionar bem, aparecerá um "Have a nice day!" ao final.
3. Serão gerados dois novos arquivos, com os nomes ctb_train_999.observed.gz e ctb_train_999.obj.gz. Este último é o arquivo que será usado pelo parser. Para que este seja utilizado pelo script tb-backgroundParsing.pl, é preciso criar um link simbólico com o nome de ctb.obj.gz, na mesma pasta (/home/tools/dbparser/). Para isso, execute[1] (ainda na janela do terminal): ln -s ctb_train_999.obj.gz ctb.obj.gz. Note que no lugar de "999" deve vir sempre o número associado ao corpus de treinamento.

Com isso, tudo está pronto para a execução do parser, cujos passos são apresentados a seguir.

Passo 1: ajuste do arquivo etiquetado

Se o texto já passou por este script, você pode pular este passo. Caso contrário, abra uma janela de terminal, vá para a pasta contendo o texto e execute:

/home/tools/bin/tb-adjustPOS.pl <nome-do-texto>

Será criado um novo arquivo, com o mesmo nome acrescido da extensão ".cs". Este é o arquivo que deverá ser submetido ao analisador.

> *Dica*: vale a pena fazer um "passeio" no texto a ser submetido, para quebrar sentenças muito longas (por exemplo, com orações coordenadas), que certamente serão mal analisadas ou até não analisadas pelo analisador. Com isso, a performance do analisador será melhor e tarefa de revisão será mais fácil. Para fazer isso, utilize um editor tal como *Kate* ou similar (compatível com UTF-8). Depois, na correção da análise sintática, pode-se juntar as orações novamente, se for o caso.

Passo 2: split de etiquetas

Se o seu sistema de etiquetas prevê etiquetas compostas para casos como os preposição + artigo ("do", "das", etc.), verbo + clítico ("vende-se", etc.) entre outros, pode ser que você tenha o interesse de separar estes segmentos, enviando-os para o parser como elementos isolados (por exemplo, "do/P+D → d@/P @o/D", "das/P+D-F-P → d@/P @as/D-F-P", "vende-se/VB+CL → vende-/VB -se/CL"), para que sejam nós terminais na árvore. Se for este seu interesse, é preciso abrir o texto etiquetado em um editor de texto apropriado, fazendo-se manualmente o "split" das palavras e

1 Se o o link já existir, exclua-o (utilizando o comando rm ctb.obj.gz ou pelo gerenciador de arquivos).

etiquetas compostas. Outra opção é deixar como está, com tais elementos formando terminais sincréticos na análise sintática.

Passo 3: executando o analisador

Abra uma janela de terminal (ou continue na mesma do *Passo 1*) , vá para a pasta contendo o texto etiquetado e execute:

```
tb-backgroundParsing.pl <max_heap> <max_active_processes>
                        <max_parse_time> <pos-file>
```

Sendo que:

<max_heap>	Tamanho da pilha (em MB) (até 500)
<max_active_processes>	Quantidade máxima de análises simultâneas
<max_parse_time>	Tempo máximo de espera na análise de uma sentença (minutos)
<pos-file>	Arquivo etiquetado a processar

Por exemplo, para servidores Dell Precision T3500, que possuem 6GB de memória RAM e processador Xeon Quad, tenho utilizado:

```
tb-backgroundParsing.pl 500 12 10 <pos-file>
```

Ou seja, pilha de 500MB, 12 sentenças em paralelo e 10 minutos de tempo limite por sentença. Com esta configuração, para dar uma idéia, o texto do Ortigão (o_001) – em torno de 1800 sentenças – foi processado em apenas 3 horas.

Passo 4: após a análise

Ao terminar, o script terá gerado dois arquivos novos, na mesma pasta em que está o texto etiquetado. Os arquivos são:

<pos-file>.log	Arquivo com o registro das análises (a saída do analisador)
<pos-file>.psd	Texto analisado sintaticamente

Neste momento, você pode eventualmente renomear o arquivo, caso ache necessário.

Números de sentença ("IDs")

Após definir o nome do arquivo, pode ser necessário *inserir os números de sentença* ("ID"). Para isso, abra um *terminal*, vá até a pasta onde está o arquivo e execute:

```
tb-addID.pl arquivo-psd > file.tmp
```

Isso vai criar o arquivo "file.tmp", que é uma cópia do arquivo anotado, porém contendo as referências para cada sentença. Abra este arquivo em um editor de textos (*Kate*) ou *preferencialmente* no CorpusDraw e veja está correto. Confirmando isso, substitua o arquivo sem IDs através do comando:

```
mv file.tmp arquivo-psd
```

Revisão

Se a revisão for feita por várias pessoas, será necessário dividir o arquivo em partes, o que pode ser feito utilizando o editor *Kate* ou outro similar (compatível com UTF-8). Sugere-se utilizar a referência sobre o número da sentença (o ID), como base para a separação das partes.

Observações finais

O script "tb-backgroundParsing.pl" dá a opção de acompanhar o andamento da análise, enviando um e-mail no início, informando o total de sentenças, além de e-mails de hora em hora, informando o número de sentenças analisadas e um e-mail ao final do processo. Para habilitar essa opção, é preciso editar o script, substituindo as ocorrências de "youremail\@thedomain" pelo e-mail desejado e descomentando as linhas (retirando o sinal de "#" do início delas). **Cuidado ao fazer isso, ok!**