

Manatee

Описание

Manatee - система управления базы данных написано на языке C++, разработанная с целью того, чтобы эффективно индексировать объёмные корпуса. Система основана на идее обратной индексации (хранение индексов всех позиций слов в данном тексте). Корпуса, индексированные с помощью Manatee, обработаны посредством языка CQL.

Установка (инструкции написано для убунту 16.04, 64-бит)

- Требуется использовать 64-бит операционную систему.
- Требуется по крайней мере 1ГГц процессор и 1гб оперативной памяти (рекомендуется 4гб для нормального использования, и 8гб для больших корпусов (выше 1гб))

Предпосылки:

1. Установить python 2.7. (Инструкция находится по этой ссылке <https://www.python.org/>)
2. Установить python-cheetah
 - a. В терминале с правами root введите команду - *'apt-get install python-cheetah'*
3. Установить PCRE
 - a. В терминале с правами root введите команду - *'apt-get install libpcre3 libpcre++-dev'*
4. Установить apache2
 - a. В терминале с правами root введите команду - *'apt-get install apache2'*
 - b. В терминале с правами root введите команду - *'apt-get install apache2-dev'*
5. Установить python-simplejson
 - a. В терминале с правами root введите команду - *'apt-get install python-simplejson'*
6. Установить libltdl7
 - a. В терминале с правами root введите команду - *'apt-get install libltdl7'*

*****Для простоты можно просто написать данную строку в терминале с правами root - *'apt-get install libpcre3 libpcre++-dev apache2 apache2-dev python python-cheetah python-simplejson libltdl7'***

Скачать:

1. Из архива (<http://corpora.fi.muni.cz/noske/deb/1604/>) скачать следующие:
 - a. Finlib(<http://corpora.fi.muni.cz/noske/deb/1604/finlib/>)
 - b. Manatee-open(<http://corpora.fi.muni.cz/noske/deb/1604/manatee-open/>)
 - c. manatee-open-python(<http://corpora.fi.muni.cz/noske/deb/1604/manatee-open/>)
 - d. Bonito-open(<http://corpora.fi.muni.cz/noske/deb/1604/bonito-open/>)
 - e. Bonito-open-www(<http://corpora.fi.muni.cz/noske/deb/1604/bonito-open/>)
 - f. python-signalfd(<http://corpora.fi.muni.cz/noske/deb/1604/python-signalfd/>)
 - g. Antlr(<http://corpora.fi.muni.cz/noske/deb/1604/antlr3c/>)

Распаковка:

1. Используя терминал, находясь в папке с .deb архивами, введите команду -
`'dpkg -i finlib_2.35.2-1_amd64.deb manatee-open_2.139.3-1ubuntu1_amd64.deb
manatee-open-python_2.139.3-1ubuntu1_amd64.deb
bonito-open_3.88.9-1_all.deb bonito-open-www_3.88.9-1_all.deb
python-signalfd_0.1-1ubuntu1_amd64.deb libantlr3c_3.4-1_amd64.deb'`

****Версии могут смениться. Если команда не работает, проверьте версии своей установки.*

Дополнительные инструкции:

1. Если желаете использовать корпус susanne, то можно здесь скачать (<http://corpora.fi.muni.cz/noske/deb/1604/manatee-open/>)
2. Потом введите команду - `'dpkg -i
manatee-open-susanne_2.139.3-1ubuntu1_amd64.deb'`

Bonito

Описание

Bonito - программа написана на языке python, служащая веб-интерфейсом и посредником между низкоуровневыми функциями Manatee и пользователями. Для генерации веб-страниц, пользуется шаблонизатор, который впоследствии обеспечивает лёгкие изменения формата и внешнего вида веб-страниц.

Установка

См. Инструкции для установления Manatee***

Конфигурация корпуса

Установка корпуса

1. Переместитесь в папку 'bonito' с помощью команды - `'cd /var/www/bonito/'`
2. С правами root откройте файл `run.cgi` с помощью текстовым редактором, например - `'gedit run.cgi'` или `'nano run.cgi'`
 - a. Переменная `'corplist'` содержит список доступных корпусов
 - i. например - `corplist = ['susanne', 'bnc', 'rnc']`
 - b. Переменная `'corpname'` определяет корпус, который показан по умолчанию.
 - i. например - `corpname = 'rnc'`
 - c. Переменная `os.environ['MANATEE_REGISTRY']` показывает путь к файлу конфигурации корпуса.
 - i. например - `os.environ['MANATEE_REGISTRY'] = '/corpora/registry'`

Установка Bonito на Apache2

1. Вы должны были уже установить у себя apache2 в предыдущем шагу. Если не установлены:
 - a. В терминале с правами root введите команду - `'apt-get install apache2'`
 - b. В терминале с правами root введите команду - `'apt-get install apache2-dev'`
2. Потом введите команду - `'/etc/init.d/apache2 start'` или `'/etc/init.d/httpd start'`

3. Откройте любой браузер и напишите в строке - *'localhost/bonito'*? чтобы войти в сайт.

Kontext

Описание

Kontext - интерфейс корпусных запросов, основанный на корпусном поисковике Manatee. Изначально Kontext являлся вариантом корпусного интерфейса Bonito, но постепенно стал более самостоятельным.

Установка

Предпосылки

1. Установить python 2.7. (Инструкция находится по этой ссылке <https://www.python.org/>)
2. Установить
 - а. В терминале с правами root введите команду - *'sudo apt-get install libxml2-dev libxslt-dev python-dev'*
3. Установить python-cheetah
 - а. В терминале с правами root введите команду - *'sudo apt-get install python-cheetah'*
4. Установить apache2
 - а. В терминале с правами root введите команду - *'sudo apt-get install apache2'*
5. Установить lxml и дополнительные библиотеки
 - а. В терминале с правами root введите команду - *'pip install lxml'*
6. Установить werkzeug
 - а. В терминале с правами root введите команду - *'pip install werkzeug'*
7. Установить PyICU
 - а. В терминале с правами root введите команду - *'pip install PyICU'*
8. Установить Markdown
 - а. В терминале с правами root введите команду - *'pip install markdown'*
9. Установить openpyxl
 - а. В терминале с правами root введите команду - *'pip install openpyxl'*
10. Установить mod-wsgi
 - а. В терминале с правами root введите команду - *'pip install mod-wsgi'*

- b. В терминале с правами root введите команду - *'sudo apt-get install libapache2-mod-wsgi'*

c.

11. Установить систему хранения ключа-значения

- a. По умолчанию поддерживаются redis и SQLite
- b. Redis (В терминале с правами root введите команду) - *'apt-get install redis-server'*
- c. *'pip install redis'*
- d. Библиотека SQLite должна быть установлена по умолчанию
 - i. SQLite (В терминале с правами root введите команду) - *'apt-get install SQLite'*
 - ii.

*** Находясь в папке kontekst, вы тоже можете вести данный строку в командной строке, чтобы все предпосылочные требования установить - *'pip install -r requirements.txt'*

Установить Manatee (см. инструкции по установлению Manatee)*

Скачать Kontext

- 1. В терминале переместитесь в папку, где хотите хранить Kontext (например /var/www/) с помощью командой - *'cd'*
- 2. В терминале с правами root введите команду - *'wget <https://github.com/czcorpus/kontext>'*

Изменение правил

- 1. Добавить главного юзера в группу - *'sudo adduser <username> www-data'*
- 2. Изменить владение группой - *'sudo chown -R www-data:www-data /var/www/'*
- 3. Изменить правил - *'sudo chmod -R g+wrX /var/www/'*

Mod-wsgi

- 1. Переместитесь в папку sites-enabled - *'cd /etc/apache2/sites-enabled/'*
- 2. Создайте файл kontekst.conf - *'sudo gedit kontekst_config'*
- 3. Скопируйте и вставьте нижеуказанный текст в файл (заполните места, где написано /path/to/... с своими собственными путями) :
 - a. *<VirtualHost *:80>*
 ServerName my.domain
 DocumentRoot /path/to/kontext/public

 Alias /files /path/to/kontext/public/files

```
<Directory /path/to/kontext/public/files>
    Order deny,allow
    Allow from all
</Directory>
```

```
WSGIScriptAlias / /path/to/kontext/public/app.py
WSGIDaemonProcess kontext_app processes=2 threads=15
display-name=%{GROUP}
WSGIProcessGroup %{GLOBAL}
</VirtualHost>
```

b. например:

```
<VirtualHost *:80>
    ServerName kontext.test
    DocumentRoot /var/www/kontext/public

    Alias /files /var/www/kontext/public/files
    <Directory /var/www/kontext/public/files>
        Order deny,allow
        Allow from all
    </Directory>
```

```
WSGIScriptAlias / /var/www/kontext/public/app.py
WSGIDaemonProcess kontext_app processes=2 threads=15
display-name=%{GROUP}
WSGIProcessGroup %{GLOBAL}
</VirtualHost>
```

4. редактировать etc/hosts с командой - *'sudo gedit /etc/hosts'*

Напишите - 127.0.0.1 kontext.test (или с другим именем, указанным ServerName)

Конфигурация

1. В папке Kontext/conf/ создайте файл config.xml - *'gedit config.xml'*

2. В файле введите нижеуказанный текст:

```
<kontext>
    <theme>
        <name>default</name>
        <css>style.css</css>
        <logo>logo.png</logo>
```

```
<logo_mouseover>logo_s.png</logo_mouseover>
<logo_href>http://ruscorpor.ru</logo_href>
<logo_inline_css>width: 300px;height: 150px</logo_inline_css>
<fonts>
```

```
<item>https://fonts.googleapis.com/css?family=Cousine:400|Roboto:300,400,400italic,700,700italic|Roboto+Condensed:400,700&subset=latin,latin-ext</item>
```

```
</fonts>
</theme>
<global>
<deployment_id>20151031_v4</deployment_id>
<manatee_path>/var/lib/manatee</manatee_path>
<umask>002</umask>
<debug>>false</debug>
<maintenance>>false</maintenance>
```

```
<profile_log_path>/optional/path/used/for/profiling/output/if/debug/level/is/2</profile_log_path>
```

```
<max_attr_list_size>30</max_attr_list_size>
<translations>
<language label="česky">cs_CZ</language>
<language label="English">en_US</language>
</translations>
<error_report_url>http://acmecorpora.org/support</error_report_url>
<error_report_params>
<param name="queryurl">@_get_current_url</param>
</error_report_params>
<upload_cache_dir>/tmp/kontext-upload</upload_cache_dir>
<periodic_tasks
conf="/var/www/kontext/conf/beatconfig.py">celery</periodic_tasks>
<conc_line_max_group_num>99</conc_line_max_group_num>
<calc_backend
conf="/var/www/kontext/conf/celeryconfig.py">celery</calc_backend>
<action_path_prefix>/kontext</action_path_prefix>
<static_files_prefix>/files</static_files_prefix>
<intl_polyfill_url>https://cdn.polyfill.io/v2/polyfill.js?features={0}</intl_polyfill_url>
<use_conc_toolbar>1</use_conc_toolbar>
</global>
<mailing>
```

```
<smtp_server>acme.localdomain</smtp_server>
<sender>acme@localdomain</sender>
</mailing>
<logging>
<path>/var/log/kontext/application.log</path>
<file_size>1000000</file_size>
<num_files>5</num_files>
<values>
<item>environ:REMOTE_ADDR</item>
<item>environ:HTTP_USER_AGENT</item>
<item>date</item>
<item>user</item>
<item>user_id</item>
<item>action</item>
<item>params</item>
<item>settings</item>
<item>proc_time</item>
<item>pid</item>
</values>
</logging>
<corpora>
<calc_pid_dir>/var/www/kontext/corpora/calc/</calc_pid_dir>
<manatee_registry>/var/lib/manatee/registry/</manatee_registry>
<users_subcpath>/var/www/kontext/corpora/subc/</users_subcpath>
<freqs_cache_dir>/var/www/kontext/corpora/freq_cache/</freqs_cache_dir>
<freqs_cache_ttl>3600</freqs_cache_ttl>
<freqs_cache_min_lines>100</freqs_cache_min_lines>
<colls_cache_dir>/var/www/kontext/corpora/colls_cache/</colls_cache_dir>
<conc_dir>/var/www/kontext/corpora/conc/</conc_dir>
<default_corpora>
<item>susanne</item>
<item>spec/susanne</item>
<item>anothercorpus</item>
</default_corpora>
<speech_files_path>/var/www/data</speech_files_path>
<empty_attr_value_placeholder>[empty]</empty_attr_value_placeholder>
<multilevel_freq_dist_max_levels>10</multilevel_freq_dist_max_levels>
<use_db_whitelist>1</use_db_whitelist>
<corplist title="RNC Corpora">
```



```

<corpus ident="susanne" sentence_struct="s"/>
</corplist>
</corpora>
<plugins>
<db>
<module>redis_db</module>
<db_path extension-by="default"> var/lib/redis/6379</db_path>
<host extension-by="default">127.0.0.1</host>
<port extension-by="default">6379</port>
<id extension-by="default">1</id>
</db>
<auth>
<module>default_auth</module>
<auth_cookie_name>kontext_session</auth_cookie_name>
<anonymous_user_id>0</anonymous_user_id>
<login_url>kontext.test/user/login</login_url>
<logout_url>kontext.test/user/logout</logout_url>
</auth>
<getlang></getlang>
<query_storage>
<module>default_query_storage</module>
<js_module>ucnkQueryStorage</js_module>
<history_max_query_size>250</history_max_query_size>
<page_num_records extension-by="default">10</page_num_records>
<page_append_records>5</page_append_records>
<num_kept_records>10</num_kept_records>
</query_storage>
<conc_persistence>
<module>default_conc_persistence</module>
<ttl_days extension-by="default">100</ttl_days>
<anonymous_user_ttl_days
extension-by="default">7</anonymous_user_ttl_days>
</conc_persistence>
<locking>
<module>redis_locking</module>
<ttl extension-by="default">20</ttl>
<num_attempts extension-by="default">10</num_attempts>
</locking>
<conc_cache>

```

```

<module>default_conc_cache</module>
<cache_dir extension-by="default">/var/local/corpora/cache</cache_dir>
</conc_cache>
<live_attributes></live_attributes>
<settings_storage>
<module>default_settings_storage</module>
<excluded_users>
    <item>1</item>
    <item>3</item>
</excluded_users>
</settings_storage>
<sessions>
<module>default_sessions</module>
<ttl>14400</ttl>
</sessions>
<corparch>
<module>default_corparch</module>
<js_module>defaultCorparch</js_module>
<file>/var/www/kontext/conf/config.xml</file>
<root_elm_path>/corpora/corplist</root_elm_path>
<tag_prefix extension-by="default">+</tag_prefix>
<max_num_hints extension-by="default">10</max_num_hints>
</corparch>
<export>
<csv>default_csv</csv>
<xlsx>default_xlsx</xlsx>
</export>
<subc_restore></subc_restore>
<taghelper></taghelper>
<user_items>
<module>default_user_items</module>
</user_items>
<menu_items>
<module>default_menu_items</module>
<data_path
extension-by="default">/var/www/kontext/conf/main-menu.sample.json</data_path>
</menu_items>
<applicationBar></applicationBar>
<footerBar></footerBar>

```

```
<subcmixer></subcmixer>
<syntaxViewer></syntaxViewer>
</plugins>
</kontext>
```

Установить Nodejs и Grunt и настроить Kontext

1. Установить Nodejs - `'curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -`
`sudo apt-get install -y nodejs`
`sudo apt-get install -y build-essential'`
`'sudo npm install npm -g'`
2. Установить Grunt - `'npm install -g grunt-cli'`
3. Находясь в папке Kontext/ введите команду - `'sudo npm install; grunt production'`

Войдите в браузер и введите в строку 'YOUR ServerName/YOUR_WSGIScriptAlias' или просто 'YOUR ServerName'

Создание юзеров в Redis: