

Домашнее задание по HDFS

В этом домашнем задании Вам предстоит поближе познакомиться с [распределенной файловой системой HDFS] (http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html). Ваша цель - написать скрипты, которые будут делать HTTP запросы к различным демонам HDFS или вызвать стандартные утилиты командной строки (shell) с целью получения запрашиваемой информации.

Можно использовать любой скриптовый язык программирования. Самый простой вариант - использовать bash, а также утилиты curl, grep, wc и т.п.

Решение каждой задачи помещайте в отдельных директориях (в домашней директории на клиентском сервере) с именами hw1_task01, hw1_task02 и т.д. Там, где не указано иное, решение надо оформить в виде исполняемого скрипта `run.sh`.

> Даже если разрабатываете на Python, `run.sh` может содержать только вызов нужного скрипта, например:

```
>
>   #!/usr/bin/env bash
>
>   python my_python_script.py $*
```

> Тут \$* обеспечит передачу параметров, с которыми был вызван `run.sh` в `my_python_script.py`.

Задачи

AA (0.1 балла)

Запишите в свою домашнюю директорию в HDFS файл `README`, содержащий:

* в первой строке - ваш e-mail

* во второй строке - имя и фамилию

Не путайте строки и имя с фамилией. Задача проверяется по наличию файла.

BB (0.1 балла)

Создайте текстовый файл размером не больше 5 MB. Желательно именно с текстом, а не с html, xml и прочей разметкой. Можно откуда-то скопировать, можно скачать.

> Хорошо подходит для этого lib.ru, вот так можно получить всего "Евгения Онегина":

```
>
>   $ wget "http://lib.ru/LITRA/PUSHKIN/p4.txt"
> только возможно придется поправить кодировку одним из способов:
```

```
>
>   $ cat p4.txt | iconv -f windows-1251 -t utf8 p4_utf.txt
>   $ cat p4.txt | iconv -f koi8-r -t utf8 p4_utf.txt
```

> Главное, чтобы в итоге команда:

```
>
>   $ more <file>
> выдавала читаемый текст.
```

Создайте в домашней директории в HDFS директорию text/full и запишите туда полученный файл. Выделите 50 первых строк файла в отдельный файл (удобно с помощью команды head), запишите в директорию text/sample. Задача проверяется по наличию директорий и файлов. *Не качайте все одно и то же!*

01 (0.1 балла)

На вход скрипту подается имя файла, на выходе нужно получить имя сервера или IP-адрес, с которого будет читаться первый блок данных (реплик может быть несколько, засчитываться будет любой из них). Пример:

```
$ ./run.sh /data/access_logs/big_log/access.log.2015-12-10
mipt-node01.atp-fivt.org
```

02 (0.1 балла)

На вход скрипту подается имя файла, на выходе нужно получить первые 10 байт этого файла (hadoop fs и hdfs dfs использовать нельзя)

```
$ ./run.sh /data/access_logs/big_log/access.log.2015-12-10
41.190.60.
```

03 (0.1 балла)

На вход скрипту подается полный путь до файла в HDFS, на выходе нужно получить размер файла в блоках (см. hdfs fsck -h).

```
$ ./run.sh /data/access_logs/big_log/access.log.2015-12-10
8
```

04 (0.1 балла)

На вход скрипту подается идентификатор блока, на выходе нужно получить имя сервера (если их несколько, то выбрать любой), где хранится данный блок и физический путь в локальной файловой системе до этого блока данных. (О том, как зайти на ноды кластера, написано в материалах Семинара по HDFS)

```
$ ./run.sh blk_1075127191
bds03.vdi.mipt.ru:/dfs/dn/current/BP-76251478-10.55.163.141-
1427134131440/current/finalized/subdir21/subdir35/blk_1075127191
```

05 (0.2 балла)

Большие файлы на кластере делятся на блоки определенного размера. Нужно выяснить какой дополнительный объем используется в HDFS для хранения данных при использовании одной реплики (т.е. без дублирования данных). Для проведения экспериментов и создания файлов разного размера предлагается использовать утилиту dd (см. [пример использования] (<http://unix.stackexchange.com/questions/101332/generate-file-of-a-certain-size>)) + решение задачи 4. На вход программы подается размер файла в байтах, на выходе программы должно быть одно число, равное числу байт для физического хранения этого файла (без учета хранения файлов с расширением .meta). Из полученного результата вычесть объем исходного файла (в байтах).

```
$ ./run.sh <int>
123
```

Исследовательская задача (0.5 балла),

06 (0.5 балла)

На выходе – отчет, принимается только в случае сдачи в срок всех предыдущих задач.

Допустим у нас имеется 32GB RAM на сервере, где у нас установлен сервис Namenode. 6GB RAM зарезервировано на различные сервисы для мониторинга состояния кластера и поддержки работоспособности системы. Объем метаданных на 1 блок – 200 байт.

1. В предположении 3-кратной репликации оценить максимальный объем данных, которые можно хранить на кластере (стандартный размер блока: 64 или 128 MB).
2. Сколько данных можно хранить на кластере, если Secondary Namenode работает на той же машине, что и Namenode?
3. Как изменится этот показатель в предположении того, что средний размер файла в блоках – 1.5?
4. Допустим к нам пришел заказчик и сказал, что собирается хранить и обрабатывать на наших серверах 120PB данных. Какой минимальный объем оперативной памяти должен быть установлен при вышеуказанных параметрах реплицирования, размера блока и средней длины файла в блоках?