**PHẦN 1. JAVA CORE**

1. **Tài liệu**

OOP core:

<http://www.w3resource.com/java-tutorial/java-object-oriented-programming.php>

Concurrency:  
<http://winterbe.com/posts/2015/04/07/java8-concurrency-tutorial-thread-executor-examples/>

(*cái này nó viết bằng syntax của java 8*)

Collection(s)

(*Overview*) <http://cs.lmu.edu/~ray/notes/collections/>

(*compare*) <http://www.codejava.net/java-core/collections/java-collections-framework-summary-table>

(*performance*) <http://infotechgems.blogspot.com/2011/11/java-collections-performance-time.html>

I/O:

(*phần rút gọn*) <http://www.studytonight.com/java/java-io-stream.php>

(*phần đầy đủ*) <http://www.ntu.edu.sg/home/ehchua/programming/java/j5b_io.html>

(phần đầy đủ khá phức tạp, và nhiều phần, em đọc lướt thôi nhé)

Exception

<http://www.journaldev.com/1696/java-exception-handling-tutorial-with-examples-and-best-practices>

SQL & JDBC

Sử dụng được các lệnh Insert, select, update, delete. *(Tài liệu tìm trên mạng)*

JDBC: <http://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-select-update-and-delete-examples>

1. **Test:**

(Điểm đánh giá từ 0-10)

1. Sử dụng Set để tìm tập giao và tập hợp giữa 2 tập hợp (tự tạo ra 2 tập hợp, mỗi tập hợp 200.000 phần tử không trùng nhau và các phần tử là một số Integer ngẫu nhiên, 2 tập hợp phải có phần tử trùng nhau) trong khoảng thời gian ngắn nhất.

2. Cho một văn bản bất kỳ, đặt ở file input.txt. Hãy đọc file và đếm số lần xuất hiện của từng từ (mỗi từ phân biệt nhau bằng dấu cách) rồi ghi vào file output.txt.

3. Lập trình đếm từ đa luồng cho bài 2 với dữ liệu vào là một folder chứa nhiều file text. Hãy xử lý song song các file và tìm top 10 từ xuất hiện nhiều nhất, và top 10 từ xuất hiện ít nhất của toàn bộ dữ liệu có trong folder. Lưu ý, chỉ được chạy tối đa 6 luồng cùng lúc.

4. Sinh ngẫu nhiên 30.000 điểm trên mặt phẳng tọa độ 2 chiều số nguyên. Sao cho 8.000 điểm đầu tiên có cách điểm A(800, 800) một độ dài không quá 400 đơn vị, 10.000 điểm tiếp theo cách điểm B(4000,800) không quá 500 đơn vị, và 12.000 điểm cuối cùng cách điểm C(2400, 2400) không quá 600 đơn vị. Trộn ngẫu nhiên 30.000 điểm này, sau đó ghi ra file output4.txt

5. Trong file Maze.java có chứa bản đồ mê cung, hãy implement hàm solve dựa trên một phương pháp tìm đường nào đó. Hãy chọn các phương pháp có độ phức tạp (số bước thực hiện) càng nhỏ càng tốt và đánh dấu đường đi bằng cách thay số 0 bằng số 2 trên ma trận mê cung. Lưu ý, bài tập này có xét cách viết code, comment, viết java doc, đặt tên biến/hàm, tổ chức project …

**PHẦN 2. TOOLs**

1. **Tài liệu**

Làm quen với Linux (shell, file system, permission…):

<https://help.ubuntu.com/community/LinuxFilesystemTreeOverview>

<http://ryanstutorials.net/linuxtutorial/permissions.php>

<https://www-xray.ast.cam.ac.uk/~jss/lecture/computing/notes/out/commands_basic/>

Mã hóa, bảo mật với khóa công khai (SSH)

<https://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_h%C3%B3a_kh%C3%B3a_c%C3%B4ng_khai>

Mạng và các giao thức cơ bản (TCP/IP, port, SSH, http…)

<https://vi.wikipedia.org/wiki/TCP/IP>

<http://code.tutsplus.com/tutorials/http-the-protocol-every-web-developer-must-know-part-1--net-31177>

Sử dụng VPN (openVPN), IDE (eclipse, intellij), Maven, git

*Tài liệu tìm trên mạng*

1. **Test:**

(Điểm đánh giá từ 0-10)

1. Tạo maven project, sau đó add thêm 2 thư viện logback, và json để parse một dữ liệu json từ url <http://news.admicro.vn:10002/api/realtime?domain=kenh14.vn>.

Tạo một thread lấy dữ liệu từ url mỗi 1s một lần, tách lấy số user và lưu vào biến tĩnh S.

Tạo một thread khác quét biến S mỗi 2s một lần, và ghi lại sử dụng logback theo quy tắc sau:

* Lần đầu sẽ ghi giá trị biến S với level info
* Các lần sau nếu biến S lớn hơn 1.5% so với giá trị của biến S vào thời điểm trước đó thì ghi lại giá trị biến S với level Info
* Nếu quá lâu (12s) mà không ghi data lần nào thì sẽ buộc ghi giá trị biến S với level Debug

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0s | 2s | 4s | 6s | 8s | 10s | 12s | 14s | 16s | 18s | 20s | 22s |
| 1032 | 1022 | 1092 | 1091 | 1082 | 1080 | 1075 | 1072 | 1060 | 1059 | 1075 | 1076 |
| Info |  | Info |  |  |  |  |  | Debug |  | Info |  |

2. Tạo maven project với project encode là utf-8, sử dụng thêm thư viện jsoup, sau đó build toàn bộ project thành một file jar duy nhất chứa tất cả các thư viện đi kèm (sử dụng plug-in *jar-with-dependencies* của maven). Sử dụng thư viện jsoup lấy nội dung của url ”<http://dantri.com.vn>” và ghi nội dung ra một file với tên file là thời gian hiện tại. Sử dụng shell script của linux để chạy file jar mỗi 5 giây một lần.

3. Tạo một project viết một restful webservice (có thể sử dụng [sparkjava.com](http://sparkjava.com)) trả lại dãy số nguyên tố từ một đến n nhập vào từ param trên url của request (Ví dụ: <http://localhost:8080/prime?n=10000> sẽ trả lại danh sách các số nguyên tố từ 1 đến 10000).

4. Sử dụng guava cache để giảm số lần request muốn tính cùng một số của bài 3, nếu sau 10s không có request thì cache tự xóa.

5. Upload project bài 4 lên github.com. Sử dụng ssh-key để remote vào một máy khác sử dụng linux (nếu không có thì sử dụng máy ảo) mà không sử dụng password. Deploy project bài 4 thông qua git clone và build file jar bằng lệnh của maven trên máy đó.

**PHẦN 3. Scala/Python/Golang**

(Phần tranning này tùy vị trí, nếu cần thiết)

1. **Tài liệu**

Scala basic

* Tutorials:
  + <https://twitter.github.io/scala_school/>
  + <https://hoaiptvn.blogspot.com/search/label/Scala>
* Others:

<https://github.com/lauris/awesome-scala>

Python basic:

[**Python Cookbook, 3rd Edition.pdf**](https://vk.com/doc169496702_353141701?hash=e9c928518cdf27b23c&dl=01490c577abb6a6b29)

Go Lang basic:

go\_tutorial.pdf

<https://tour.golang.org/list>

1. **Test:**

(Điểm đánh giá từ 0-10)

Scala

1. Cho một văn bản text:

- Đếm số từ có trong văn bản đó

- Đếm số từ có length > 3 trong văn bản đó

- Đếm số từ có length > 5 trong văn bản đó.

2. Đọc một tập văn bản text, tính tf-idf cho các từ, tạo tập từ điển từ tập các từ bằng cách chặn ngưỡng tf-idf và lọc các từ rác ( ví dụ các từ chứa số, các từ stopword - các từ stopword được cho trước)

(\*\*\*nâng cao\*\*\*) : đánh index cho các từ trong từ điển vừa tạo được, convert các văn bản theo index đó và rút gọn bằng cach biểu diễn <index từ> : <số lần xuất hiện trong văn bản>

( map từ sang index theo 2 cách : broadcast một map vocab, cách 2 là sử dụng join dataframe)

3. Đọc log file, lưu những dòng log mà truy cập vào domain dantri.com.vn lại vào một bảng cassandra ( bảng đó có các trường : timetruycap, domain, path). (Chú ý sử dụng spark thao tác trên cassandra).

Python

1.

2.

3.

4.

Go Lang

1. Cho một mảng số int, đưa ra số nhỏ nhất, lớn nhất, các số lẻ, các số chẵn. Cộng mỗi phần tử của mảng bằng cách, nếu phần tử mảng đó chia hết cho 3 thì cộng 3, chia hết cho 5 thì cộng 5, chia hết cho 7 thì cộng 7, chia hết cho 9 thì cộng 9, còn không thì cộng 1. In lại mảng đó.

2. Đọc một tập văn bản text, tính tf-idf cho các từ, tạo tập từ điển từ tập các từ bằng cách chặn ngưỡng tf-idf và lọc các từ rác ( ví dụ các từ chứa số, các từ stopword,. ..)

(\*\*\*nâng cao\*\*\*) : đánh index cho các từ trong từ điển vừa tạo được, convert các văn bản theo index đó và rút gọn bằng cach biểu diễn <index từ> : <số lần xuất hiện trong văn bản>

3. Làm một server tiếp nhận http request, khởi tạo tại 127.0.0.1 port 6789. Khởi tạo 2 path là /hello và /bye. Khi gọi đến server thì truyền thêm tham số age= n (trong đó n là một số tuổi của một người) và tham số rand = r (r là một số tùy ý). Tính số x = ([sqrt( n!) ] \* r ) % 256. (trong đó [ ] là phần nguyên của một số ) . Trả lại cho client một xâu theo format :

câu chào + “con số may mắn của bạn là: “ x.

- Nếu path /hello thì xâu bắt đầu bằng “chào”

- Nếu path /hello thì xâu bắt đầu bằng “tạm biệt”.

(\*\*\*nâng cao\*\*\*) : tạo một cache lưu trữ kết quả, nếu như n và r gửi tới không đổi, thì lấy kết quả từ cache mà ko tính toán lại nữa.

**PHẦN 4. Hadoop/Spark/Cassandra**

1. **Tài liệu**

Tài liệu chung về bigdata:

Mapreduce (đọc Abstract, Introduction, Programming Model)

[https://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf](https://static.googleusercontent.com/media/research.google.com/en/archive/mapreduce-osdi04.pdf)

Bigtable (đọc Abstract, Introduction, Data Model)

<http://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf>

Hadoop:

https://web.stanford.edu/class/cs246/homeworks/tutorial.pdf

<https://www.tutorialspoint.com/hadoop/hadoop_tutorial.pdf> hoặc

https://www.tutorialspoint.com/hadoop/index.htm

Spark:

https://www.tutorialspoint.com/apache\_spark/index.htm

Cassandra: (file đính kèm)

introduction-to-cassandra.pdf

cassandra\_tutorial.pdf

Thử nghiệm

* Cài đặt hadoop/Spark/Cassandra trên local
* Sử dụng Hadoop/Spark/Cassandra shell
* Viết và thử chạy job trên Hadoop/Spark/Cassandra

1. **Test:**

(Điểm đánh giá từ 0-10)

Add vào file ~/.bashrc

export HADOOP\_HOME=/data/hadoop/hadoop-2.8.0

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export HADOOP\_INSTALL=$HADOOP\_HOME

export PATH=$PATH:$HADOOP\_HOME/bin:$HADOOP\_HOME/sbin

export HADOOP\_CONF\_DIR=${HADOOP\_HOME}"/etc/hadoop"

# set JAVA\_HOME

export JAVA\_HOME=/usr/java/latest

export PATH=$PATH:/usr/java/latest

1. Cài đặt thử k-mean trên môi trường phân tán sử dụng dữ liệu sinh ra của PHẦN 1. Bài 4

2. Sử dụng lệnh trên HDFS để put 10 file text bất kỳ trong thư mục 10.3.34.139/data/sample-data lên hdfs thư mục hdfs://user/< tên đăng nhập LDAP>.

Sau đó, gộp nội dung của 10 file này thành 1, và download xuống.

3. Sử dụng toàn bộ dữ liệu dữ liệu text trong thư mục: *10.3.34.139/data/sample-data* theo thứ tự trường có sẵn. Tạo thư mục lưu như bài 2, tạo file parquet ‘pageviewlog’ với tất cả các trường trong model. Chuyển toàn bộ dữ liệu text thành file parquest và thực hiện các job sau:

3.1 Lấy url đã truy cập nhiều nhất trong ngày của mỗi guid

3.2 Các IP được sử dụng bởi nhiều guid nhất *số guid không tính lặp lại, không tính số lần sử dụng chỉ lấy số lượng guid* (Lấy top 1000 và số guid duy nhất sử dụng)

3.3 Tính các guid mà có timeCreate – cookieCreate nhỏ hơn 30 phút

Để connect vào sử dụng câu lệnh cqlsh 10.3.34.139

4. Đọc dữ liệu file text từ thư mục 10.3.34.139/data/sample-data theo thứ tự trường có sẵn. Tạo keyspace theo tên đăng nhập LDAP, tạo table ‘pageviewlog’ với tất cả các trường trong model. Import toàn bộ data trong HDFS vào Cassandra. Với việc tối ưu cho câu truy vấn sau:

4.1 Liệt kê các url đã truy cập trong ngày của một guid (input: guid, date => output: ds url)

4.2 Các IP được sử dụng nhiều nhất của một guid (input: guid=> output: sort ds ip theo số lần xuất hiện)

4.3 Tìm thời gian truy cập gần nhất của một GUID (input: guid=> output: thời gian truy cập gần nhất)

4.4 Tính các guid mà có timeCreate – cookieCreate nhỏ hơn 30 phút

**PHẦN 5. SQL/MySQL**

1. **Tài liệu**

Tài liệu chung:

<http://www.tutorialspoint.com/sql/sql_tutorial.pdf>

1. **Test:**

(Điểm đánh giá từ 0-10)

Import employees database vào mysql trên local.

1. Truy vấn để liệt kê danh sách:

1.1. Liệt kê 10 nhân viên bắt đầu làm việc từ năm 1999

1.2. Đếm số nhân viên nữ có ngày sinh từ năm 1950 đến năm 1960

1.3. Lấy ra các nội dung sau của nhân viên có id = 10005: first\_name, last\_name, hire\_date, salary\_total. Trong đó salary\_total là tổng lương của nhân viên 10005 trong toàn bộ thời gian anh ta giữ chức vụ “Staff” – trong bảng titles.

1.4. Tìm xem người quản lý có tên là Margareta Markovitch hiện đang quản lý bao nhiêu nhân viên

1.5. Tìm xem tổng lương phải trả của mỗi phòng ban trong khoản thời gian from\_date = 1988-06-25 và to\_date 1989-06-25 (from\_date, to\_date từ bảng salaries) là bao nhiêu.

2. Cập nhật những nội dung sau:

2.1. Thăng chức cho nhân viên 10002 từ “Staff” lên “Senior Staff”

2.2. Hãy xóa phòng ban Production cùng toàn bộ nhân viên của phòng này, cùng với tất cả các dữ liệu có liên quan.

2.3. Thêm phòng ban mới “Bigdata & ML” và bổ nhiệm nhân viên có ID = 29005 lên làm quản lý.

3. Viết một Stored Procedure với input là tên nhân viên. Cần trả lại 2 result - kết quả :

3.1. Kết quả 1: Lấy ra id, full name, giới tính, title (hay chức vụ), tên phòng ban

3.2. Kết quả 2: Tính tổng lương của từng người có tên đó trong khoảng thời gian từ lúc nhận lương đến thời điểm hiện tại

4. Viết một Store Procedure để thuyên chuyển phòng ban cho một nhân viên nào đó, với chức vụ mới (không chuyển lên làm quản lý). Đồng thời trả lại một kết quả bao gồm:

ra id, full name, giới tính, title (hay chức vụ), tên phòng ban