Westerdals

Oslo School of Arts,
Communication and Technology

Student project, Big Data MA120,
Fall 2016
Westerdals Oslo School of
Arts, Communication and Technology

September 27, 2016

1 General information

This assignment counts 75% towards the final grade for the course. The project can be completed individually by each student or in the group of up to two students. All the assignment materials must be delivered via It's Learning well before the deadline. For any delay, a valid absence documentation must be presented to the school administration.

The assignment consists of two parts:

- An implementation reflected in a code base (delivered as a zip/tar.gz);
- A report describing the implementation, structure and student reflection.

Both parts together affect the grading. This project delivery is not anonymized. Any assumptions made by students must be clearly stated in the report. You are free to use original Hadoop with Java or choose any other language that works with Hadoop streaming. Report should include clear descriptions of the code. You can use the schools's Hadoop cluster(optional) that runs on the following machines (distributed mode):

- bigdata01.ad.nith.no (10.32.16.12) - namenode
- bigdata02.ad.nith.no (10.32.16.16) - datanode
- bigdata03.ad.nith.no (10.32.16.13) - datanode

The dataset is small enough and does not require you to run on a cluster.

If interested, send your username to instructor at ¡mailto:taknai@westerdals.no¿ and your user will be set up with passwordless login between the cluster nodes, so logging from one of these to the other does not require password.

NB! As of this writing the machines are reachable only if you physically reside in the campus Fjerdingen.

2 The Assignment: Exploring Stackoverflow

Stack Overflow is a Q&A (question and answer) site for professional and enthusiast programmers. It's built and run by you as part of the Stack Exchange network of Q&A sites. Chances are high that you will find an answer to a Hadoop related questions there.

2.1 Dataset

The dataset is an anonymized dump of all user-contributed content on the Stack Exchange network. In this course we are narrowing the scope to subset of the Stack Exchange dataset: **Database Administrators** (DBA). Database Administrators Stack Exchange is a question and answer site for people interested in finding an answer to a question they might have. You can check it out at: dba.stackexchange.com

The dataset is formatted as a separate archive consisting of XML files. The archive includes:

- Badges.xml
- Comments.xml
- PostHistory.xml
- PostLinks.xml
- Posts.xml
- Tags.xml
- Users.xml
- Votes.xml

For complete schema information, see the included `readme.txt` in appendix A.

2.1.1 Obtaining the dataset

The dataset is available from: archive.org/download/stackexchange/dba.stackexchange.com.7z

2.2 Tasks

This section describes the actual tasks to be performed by each student/-group. The **bolded** word before the task text suggests the name of the task.

2.2.1 Warmup

(a) **WordCount**. Count the words in the body of questions (note the *PostTypeId*). This is the classic WordCount. The resulting data should include counts for each word, that is, how many times each word appears in the body of questions.

(b) **Unique words**. Write a Hadoop MapReduce job that outputs words in the question titles. The output should contain all words used in the title of questions, only once. No count, just the word. That will the dictionary over titles of the questions.

(c) **>10**. How many questions are there which have more than 10 words in their titles?

(d) **Stopwords**. Based on a), exclude the stopwords from titles (an example list can be obtained at: http://j.mp/STOPWORDS). We refer to the output of this task as to popular words.

(e) **Pig top 10**. Write a Pig script that selects the top 10 list words after you remove the stopwords in step d).

(f) **Tags**. Write a MapReduce job that creates a dictionary over the unique tags (that is unique tags in the whole dataset).

2.2.2 Discover

(a) **Counting**. How many unique users are in the dataset?

(b) **Unique users**. Write a Hadoop MapReduce job that outputs unique users in the dataset.

(c) **Top DBAs**. Write a Hadoop MapReduce job that outputs top 10 users in terms of their reputation.

(d) **Top questions**. Write a Hadoop MapReduce job that outputs top 10 questions in terms of their reputation.

(e) **Favourite questions**. Write a Hadoop MapReduce job that outputs top 10 questions in terms of their *FavouriteCount*.

(f) **Average answers**.Calculate an average number of answers per question. You choose whether you want to user MapReduce or Pig.

(g) **Countries**. Discover users by countries, that is the output should be country and the number of users.

(h) **Names**. What is the most popular name of a user? List top 10 names. Hint: you may want to split the name by space.

(i) **Anwswers**. How many questions do have at least one answer?

2.2.3 Numbers

(a) **Bigram**. A pair of adjacent words is called a bigram. For example, "big data" or "fast car" are examples of bigrams. Find the most common bigram in the titles of the questions.

(b) **Trigram**. Do the same for word-level trigrams, i.e. a three words that appear consecutively.

(c) **Combiner**. In 1a) you created a WordCounter. Add a combiner to it. What are implications of having a combiner?

(d) **Useless**. The word "useless" is pretty much useless without any context. Count how many questions contain this word in the **body**.

2.2.4 Search engine

(a) **Title index**. Search engines maintain an index over dictionary with reference to the documents where they appear. The purpose is to quickly locate docuemnts so when you search it is very fast to locate the documents. Documents in the context of the project is a question.

In this task we create an index over titles and bodies of questions (including answers). What we are after is having a simple index that lists publications in which a searching term/s occur/s. The reference to the question in which the terms appear should be the attribute key.

Lets have a look at an example (for readability the example is stripped down to minimum). Given the xml below:

```
<posts>
  <row Id="10" Title="Cannot obtain secure database connection".../>
  <row Id="25" Title="How can I secure my database connection?".../>
     [...]
</posts>
```

The index entries (output) for these two posts would be:

```
can 25
cannot 10
connection 10,25
database 10,25
how 25
i 25
my 25
obtain 10
secure 10, 25
```

In this dictionary we use the term as key and list of post ids as values in which the term appears. Our index is simple, it does not weight posts based on the number of occurrence of the terms in the posts.

3 Practical parts

Use your own input format class that reads the stream of XML. You can set the class name by calling 'setInputFormatClass()' method of the Job class in your MapReduce client (driver) code.

3.1 Groups

Students are encouraged to work in groups. Groups can consist of up to 2 students. Deadline for group registration is Tuesday, 4th of October. Send your registration to: taknai@westerdals.no. Guidance hour (veiledningstime): Wednesday, 5th of October, for each team individually. Deadline for project delivery: **Friday, 21th of October, 23:55**.
Part-time students normally have extra time to deliver the project. Deadline for part-time students is: **Friday, 18th of November, 23:55**.

3.2 Structuring the results

Tasks should be independent of each other. You should not mix the implementation of one task with another. This may interfere and affect the results of the output unintentionally. Therefore, you may organize your zipped project file for each task. Some tasks depend on each other. You can copy the folder to the depending task change the code there.

Appendices

A Appendix readme.txt

The following `readme.txt` describes the dba dataset obtained from Stack Exchange.

```
- Format: 7zipped
- Files:
  - **badges**.xml
      - UserId, e.g.: "420"
      - Name, e.g.: "Teacher"
      - Date, e.g.: "2008-09-15T08:55:03.923"
  - **comments**.xml
      - Id
      - PostId
      - Score
      - Text, e.g.: "@Stu Thompson: Seems possible to me - why not
         try it?"
      - CreationDate, e.g.:"2008-09-06T08:07:10.730"
      - UserId
  - **posts**.xml
      - Id
      - PostTypeId
         - 1: Question
         - 2: Answer
      - ParentID (only present if PostTypeId is 2)
      - AcceptedAnswerId (only present if PostTypeId is 1)
      - CreationDate
      - Score
      - ViewCount
      - Body
      - OwnerUserId
      - LastEditorUserId
      - LastEditorDisplayName="Jeff Atwood"
      - LastEditDate="2009-03-05T22:28:34.823"
      - LastActivityDate="2009-03-11T12:51:01.480"
      - CommunityOwnedDate="2009-03-11T12:51:01.480"
```

```
    - ClosedDate="2009-03-11T12:51:01.480"
    - Title=
    - Tags=
    - AnswerCount
    - CommentCount
    - FavoriteCount
- **posthistory**.xml
        - Id
        - PostHistoryTypeId
                    - 1: Initial Title - The first title a
                      question is asked with.
                    - 2: Initial Body - The first raw body text a
                      post is submitted with.
                    - 3: Initial Tags - The first tags a question
                      is asked with.
                    - 4: Edit Title - A question's title has been
                      changed.
                    - 5: Edit Body - A post's body has been
                      changed, the raw text is stored here as
                      markdown.
                    - 6: Edit Tags - A question's tags have been
                      changed.
                    - 7: Rollback Title - A question's title has
                      reverted to a previous version.
                    - 8: Rollback Body - A post's body has
                      reverted to a previous version - the raw
                      text is stored here.
                    - 9: Rollback Tags - A question's tags have
                      reverted to a previous version.
                    - 10: Post Closed - A post was voted to be
                      closed.
                    - 11: Post Reopened - A post was voted to be
                      reopened.
                    - 12: Post Deleted - A post was voted to be
                      removed.
                    - 13: Post Undeleted - A post was voted to be
                      restored.
                    - 14: Post Locked - A post was locked by a
                      moderator.
                    - 15: Post Unlocked - A post was unlocked by a
```

```
            moderator.
        - 16: Community Owned - A post has become
            community owned.
        - 17: Post Migrated - A post was migrated.
        - 18: Question Merged - A question has had
            another, deleted question merged into
            itself.
        - 19: Question Protected - A question was
            protected by a moderator
        - 20: Question Unprotected - A question was
            unprotected by a moderator
        - 21: Post Disassociated - An admin removes
            the OwnerUserId from a post.
        - 22: Question Unmerged - A previously merged
            question has had its answers and votes
            restored.
- PostId
- RevisionGUID: At times more than one type of
    history record can be recorded by a single action
    . All of these will be grouped using the same
    RevisionGUID
- CreationDate: "2009-03-05T22:28:34.823"
- UserId
- UserDisplayName: populated if a user has been
    removed and no longer referenced by user Id
- Comment: This field will contain the comment made
    by the user who edited a post
- Text: A raw version of the new value for a given
    revision
        - If PostHistoryTypeId = 10, 11, 12, 13, 14,
            or 15 this column will contain a JSON
            encoded string with all users who have
            voted for the PostHistoryTypeId
        - If PostHistoryTypeId = 17 this column will
            contain migration details of either "from
            <url>" or "to <url>"
- CloseReasonId
        - 1: Exact Duplicate - This question covers
            exactly the same ground as earlier
            questions on this topic; its answers may
```

```
                    be merged with another identical question.
                  - 2: off-topic
                  - 3: subjective
                  - 4: not a real question
                  - 7: too localized
- **postlinks**.xml
  - Id
  - CreationDate
  - PostId
  - RelatedPostId
  - PostLinkTypeId
    - 1: Linked
    - 3: Duplicate
- **users**.xml
  - Id
  - Reputation
  - CreationDate
  - DisplayName
  - EmailHash
  - LastAccessDate
  - WebsiteUrl
  - Location
  - Age
  - AboutMe
  - Views
  - UpVotes
  - DownVotes
- **votes**.xml
  - Id
  - PostId
  - VoteTypeId
    - ‘ 1‘: AcceptedByOriginator
    - ‘ 2‘: UpMod
    - ‘ 3‘: DownMod
    - ‘ 4‘: Offensive
    - ‘ 5‘: Favorite - if VoteTypeId = 5 UserId will be populated
    - ‘ 6‘: Close
    - ‘ 7‘: Reopen
    - ‘ 8‘: BountyStart
    - ‘ 9‘: BountyClose
```

```
    - ‘10‘: Deletion
    - ‘11‘: Undeletion
    - ‘12‘: Spam
    - ‘13‘: InformModerator
- CreationDate
- UserId (only for VoteTypeId 5)
- BountyAmount (only for VoteTypeId 9)
```