Lehrstuhl für Systeme und Informationsverwaltung

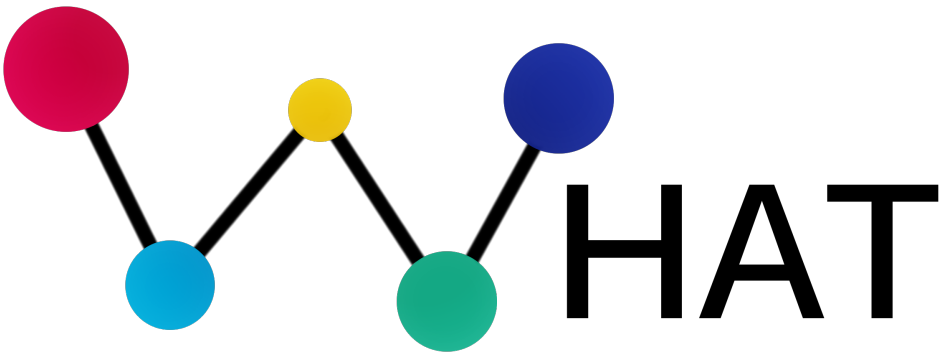
Prof.Dr.Ing. Klemens Böhm
Hoang Vu Nguyen
Marco Neumann

Karlsruher Institut für Technologie

Praxis der Softwareentwicklung (PSE)
WS 2012/2013

Visualizing and Statistically Analyzing Access Behavior to Scientific Databases

QA & Testing



WareHouse Analyzing Tool

February 28, 2013

Functional Specification	**Alexander Noe**
Design	**Jonathan Klawitter**
Implementation	**Anas Saber**
QA / Testing	**Nikolaos Alexandros Kurt Moraitakis**
Final	**Lukas Ehnle**

E-M@il: pse10-group14-ws12@ira.uni-karlsruhe.de

1 Introduction

This document is about the testing phase of the program WHAT WareHouse Analyzing Tool. It contains descriptions of what was tested and how, which results tests produced and what bugs or problems were found with them. There are also some comments about the programming and code style in general.

The tests are grouped by tasks of the program, which mostly were also separate packages. The sections of this document follow this structure and complete it with the general sections concerning the whole program.

The functions which have to show the correct web pages or produce correct SQL query where tested by hand. This means that for a configuration and data we tested, if correct looking MySQL queries where produced and correct web pages or charts where shown. But more about this in the specific sections.

The last section (11) is about complying with the requirements made in the functional specifications.

Contents

2 General

2.1 JavaDoc

2.2 CheckStyle

2.3 Coding style

hardcoded strings, task separation, reuse of code, . . .

3 Facade, Helper

4 Configuration file

This section is about the part of the system reading the configuration file.

TODO

5 Parser

TODO: Struktur ordnen und gegebenfalls erweitern

5.1 Parsing

5.1.1 Task determination

When having too many threads (10+), sometimes a thread stopped working. This caused the counter for finished tasks to be stuck under the number of threads and the program not to get finished. So we implemented a watchdogtimer which replaces a thread by a new one, if said thread doesn't finish a line for a predefinied amount of time. (standard = 2 seconds)

5.1.2 Negative or too high poolsize

The parser works with a pool of threads, which are created when parsing. In the ParserMediator there is a variable poolsize, which is the used poolsize for parsing. When using a negative or a really high number (tried Integer.MAX_VALUE), the program crashes. So we implemented a check for the poolsize, which looks if it is between 1 and 50 and stops parsing if it isn't.

5.1.3 NullPointerExceptions when correct file doesn't exist

sp_parser.Logfile had a check, if the file which is given to it is actually existent. It used the file-class from java and got a NullPointerException, which was caught and used as an indicator, that there is no file for the correct name.

This worked, but to improve our program, we used a class called FileHelper, which creates the file-object one time and returns it when needed, because other parts of the program need the file too. It didn't return a NullPointerException when the file doesn't exist, so this check got overrun and the parser "thought" that there is a correct file which wasn't. This caused the program to crash. Fixed by checking for null instead of catching a NullPointerException

5.1.4 .trim()

Some log files contain unneeded whitespace, which produce incorrect data. This was fixed by adding the `trim`-command in StringRow.split() and StringMapRow.split()

5.1.5 Anonymous Proxy

Sometimes the city- and country-name of an IP returned Anynomous Proxy, which is now replaced by "other" to fit in with the other undefined countries and cities.

5.2 Check for missing/empty parts

The parser now checks if a part of the line which is parsed is missing or empty. If it is, the line is deleted and linesDeleted gets incremented.

5.3 Only accept when enough lines submitted

Added "final static double CORRECT", a double between 0 and 100, which indicates how many percent of the lines need to be submitted correctly for the ParserMediator to return true to the facade.

5.4 Reset

The parser now resets after he finished parsing a logfile, so that another file can be parsed.

. . .

5.5 Automic tests

5.5.1 Parsing

- smallParseTestStandardSize - parse 10 lines, standard (5) poolsize
- smallParseTestSize10 - parse 10 lines, poolsize 10
- smallParseTestTooBig - parse 10 lines, poolsize 100 -> Error #2: ParserMediator.poolsizeParsing is bigger than 50.
- negativePoolsize - parse 10 lines, poolsize -2 -> Error #1: ParserMediator.poolsizeParsing is smaller than 1.
- nonexistentFile - try to parse nonexistent file -> Error #5: The path is wrong.
- wrongFile - try to parse from an empty .txt file -> Error #5: The path is wrong.
- flawedFile - try to parse from a file with a typo in "type" -> Error #11: The configuration file got a different format.
- mediumParseTestStandardSize - parse 1000 lines, standard (5) poolsize
- mediumParseTestSize50 - parse 1000 lines, poolsize 50
- bigParseTestStandardSize - parse 30k lines, standard (5) poolsize
- veryBigParseTestStandardSize (IGNORED) - parse a whole month, standard (5) poolsize, ignored because it may take hours to finish.
- smallParseTestEmptyLine - parse 10 lines with an empty line in between
- testIntegerMistake - parse 10 lines with year "2013apples" instead of "2013"

- testMissingPart - parse 10 lines with one line with a missing statement
- testEmptyPart - parse 10 lines with one empty statement and one empty year
- doubleParsing - parse 1000 lines twice from seperate files

5.6 Test scenario

6 Chart requests

Behandelt das paket chart requests

7 Data access

7.1 Connection pool

Known problem: No driver there if to often started and not returned. TODO

7.2 TODO

TODO

8 Web

Web seite betreffende Dinge. Meiste wird durch sehen getestet. Aber eben auch Dinge die sicherstellen, dass bei falscher Config nix passiert, dass ohne daten nix passiert

charts einfach so einfuegen

adminzeugs

sprachen

etc

9 Charts

Alles was mit den Charts also, der Anzeige D3 und so zusammenhaengt.

10 Performance

We should test the performance and than check how we could get it better.

10.1 Loading

Parsing of 1k lines with old upload schema:

1. 19048ms
2. 22033ms
3. 11020ms

Parsing of 1k lines with new upload schema:

1. 11016ms
2. 10017ms
3. 11017ms

Lukas hat noch mehr Zeiten, werde diese dann Eintragen und was gescheites drauss machen. Das oben war nur um meine zu speichern.

Werd dann noch Test machen, wie lange es dauert, ganze Monate hochzuladen und danach, wie lange eine Chart anfrage braucht. Sobald der Upload wieder geht

10.2 Extracting

11 Requirements

TODO: wie gut wir die sachen erfuellt oder nicht erfuellt haben, was es mehr gibt oder fehlt und so zeugs