Chrome Performance Analysis

ASSIGNMENT 4 – Final Report

***SOEN 6611:* Software Measurement**

Winter 2015

Dhruv Ohri : 27149220
Firas Mourtada : 27127995
Jasmin Abou-Auda : 26776868
Parikshit Pandya : 27585195
Saleheh Nargesian : 26302181

Contents

1. Executive Summary

Google Chrome is one of the most important web browsers that managed to rise significantly within a short period of time. Since the browser's first release, it managed to attract a large base of users. According to statscounter.com, Chrome is ranked as the highest browser in use reaching a percentage of 40.43% compared to other browsers from all platforms [2]. Our ultimate objective as Google Chrome's manager is to maintain the current user base and work towards expanding it further more. In order to achieve that objective, our goal is to ensure high performance of the Google Chrome across multiple platforms and under different environments. This report represents an analytical study, which focuses on measuring the performance of Chrome in order to identify the performance trend over time and verify whether it improved or not.

Our study was primarily based on the performance issues reported relating to Chrome and on the level of coupling and cohesion of classes that directly related to the identified performance issues. The increased number of users of Chrome has surfaced a lot of bugs and issues across different platforms with ranging severities and under different conditions. We identified few questions in the study in order to assist us in identifying the direct measures as well the confounding factors that affect those measures. Then, we formed a hypothesis to test against the measures. Our main aim was to filter and identify the relative bugs in order to measure Chrome's performance more precisely and to provide a good fitting statistical model.

2. Goal, Question, Measures

We are utilizing the Entity-Question list in order to identify what exactly we intend to learn about in regards to quality improvement. We are considering three quality criteria of Chrome that are performance, security/safety and usability. However, our core focus is to examine Chrome form the prospect of performance.

2.1 Performance:

- How does controlling memory usage affect the overall performance of Chrome?
- Does high memory consumption affect the performance of Google Chrome?
- Is the performance of Chrome affected by different platforms?

Outcome Measures:
Number of resolved/fixed bugs that are directly related to performance
Direct Measures
Number of bugs found based on different keyword search criteria
Confounds Controlling Measures
1. Capacity of internal memory. 2. Processor speed
Hypothesis to be tested:
Increased memory consumption in Google Chrome leading to 100% CPU/GPU usage is positively correlated with high number of crash reports.
Measure of quality outcome:
Reliability: Crash free running of Google Chrome.

2.2 Security:

- **Is browsing in Google Chrome considered secure and safe?**

Direct Measures	
1	Number of bugs reported related to/stating privacy / Number of bugs stating
2	Number of issues reported related to viruses / malicious code
Confounds Controlling Measures	
	1. Issues severity. 2. Level of developers' experience involved in high severity issues The more experienced the developer is, the higher the rate of severity issues being fixed thus it will help control the confound.
Hypothesis to be tested:	
	Disabling privacy and security settings in Google Chrome is positively correlated with the number of privacy and security issue reported.
Measure of quality outcome:	
	Security: Provide a safe and secure browsing.

2.3 Usability

- **How does increased testing improve the usability of Chrome?**

Direct Measures	
1	Number of issues reported related to UI failed tests
2	The number of tasks succeeded versus the number of failed tasks
Confounds Controlling Measures	
	1. Level of developers' experience 2. Level of testers skills.
Hypothesis to be tested:	
	Increased acceptance testing will lead to less number of UI issues per build.
Measure of quality outcome:	
	Usability: Enhancing the usability by providing faster and convenient browsing

An important question shall be asked as why performance, security/safety and usability very important for success of Google Chrome? The following highlights our motivation towards selecting these qualities:

- **Performance:**
  - Application's performance is one of the biggest factors that determine its success or failure in the market.
  - Chrome has the biggest user base that dominates 40% of the market [2]. Clearly, performance is a crucial quality that should always be measured and improved.

- **Security/Safety:**
  - Users increased awareness towards digital/information security and safety poses an important constraint on software companies to ensure the security and safety of the their products. Chrome being a web browser even places a higher constraint on the security and safety qualities in order to guarantee it is trustworthy to use.

- **Usability:**
  - Applications that are graphically simplistic, more user-friendly and easy to learn tend to be easier for users to work with and adopt.

3. Raw Data

Chrome issue tracker is the main source of the raw data that we relied on to mine in order to be able to extract the relevant information that help us in achieving our measures. Our analysis process involves both a manual and automated research mechanism in order to identify the actual issues that are relevant to our study. Considering performance as the core focus of this study and the most important factor out of all the three qualities identified earlier, we started exploring the Chrome issue tracker manually for us to identify the specific keywords that would help us fetch all the relevant bugs. The following table summarizes the search criteria identified pertaining to performance along with the number of issues identified for those criteria:

Keyword Search Criteria	Number of Issues
100% CPU	**2378**
Memory Crash	**7626**
High Memory	**3357**

As part of the manual research and to ensure the relevance of the issues identified in the above table, we divided the issues among team members to identify the relevance each of which verifying between 2000-2500 issues. Our analyses concluded that most of the issues identified ranging from **85% to 90% were of relevance**. The other ten percent to fifteen percent of the issues issues are categorized under the same search criteria however did not include information that is relevant to performance. After identifying the actual issues we intended to work with, we proceeded with building the raw data. We downloaded a unique set of all the issues identified excluding the duplicates that are shared among different search criteria.

4. Attribute Extraction

We started the attribute extraction phase by identifying the actual issues' attributes that of relevance to our study. These attributes were bug identifier, status, operating system, summary, description, comments, Chrome version, processor speed and internal memory. This was the automated process of our research. We utilized regular expressions to find exact pattern matches of several keywords relating to performance. The main attributes used for information search were the issue summary, description and the comments.

After extracting all the issues we used summary and description to find out all the issues that were directly related to performance, based on those we get a table with following attributes:

- Each bug is uniquely identified by the bug id and all the other attributes are related to it.
  Bug Id acts as a primary key to uniquely identify each issue.
  Scale of bug id is Ordinal

- Performance of google Chrome varies along different therefore it has been identified as one of the attributes to be extracted in order to identify the operating system against which the issue is reported.
  Scale of operating systems is Nominal

- Internal memory and processor speed are two confounding factors which help to analyze how performance is affected by them.
  (Example: Low processor speed will take lot of time process each thread that is being opened for each individual tab in Chrome)
  Scale for both of the above is absolute.

- Based on the information extracted from the above attributes, we identified the total number of bugs fixed that directly related to our outcome measure. The fixed status is one values in the attributes status. Scale of status is Nominal

5. Link and Store

The bellow analytics database schema depicts the table structure utilized to link and store the extracted information.

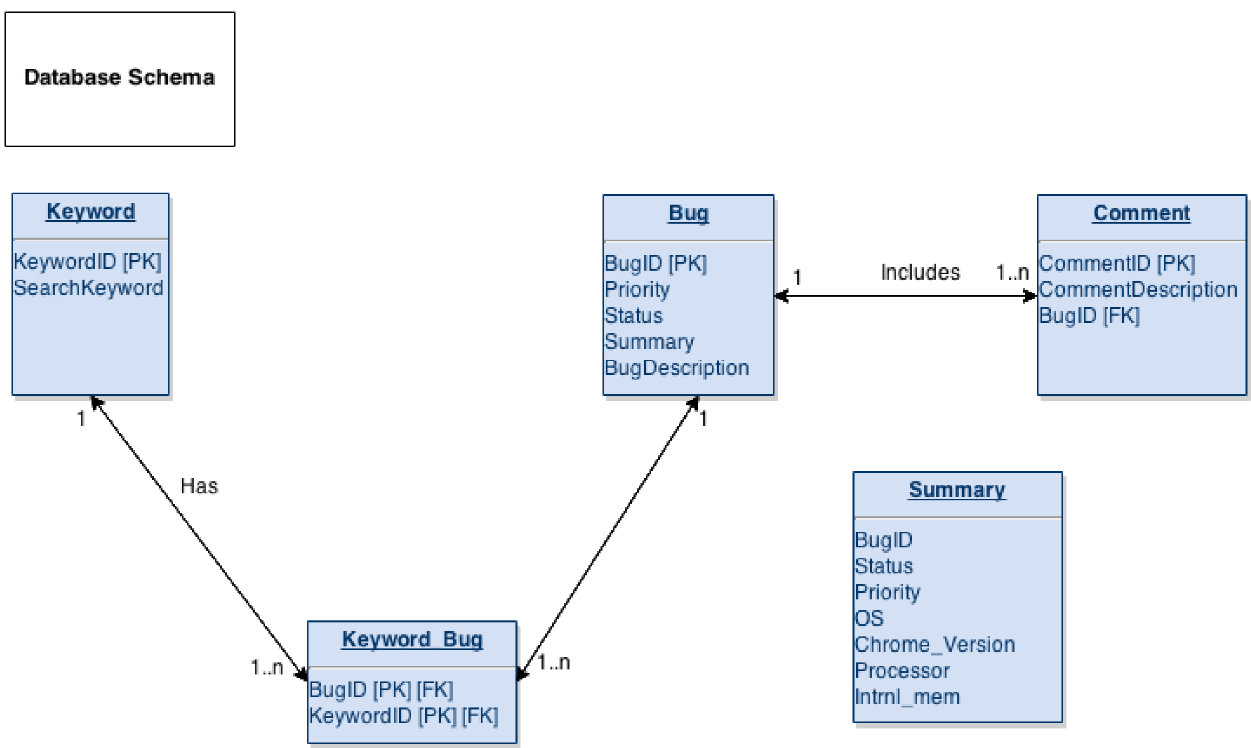

Figure 1: Analytics database schema

Summary table:

Column	Type	Modifiers
Bug id	Integer	Not null
Status	Varchar(10)	Default null
OS	Varchar(10)	Default null
Chrome_Version	Varchar(10)	Default null
Processor	Varchar(20)	Default null
Intrnl_mem	integer	Default 0

6. Descriptive Statistics:

Based on the outcome measure used which was finding out the number of fixed bugs out of the total number of bugs, we plotted a graph for all the status below to get an idea about the number of fixed bugs. Figure 2 is a box plot of total number bugs versus different bugs' status, from the plots below we can clearly interpret that:

- 30% of bugs are fixed (Maximum count)
- The outliers of all plots have almost no values.
- Graph is normally distributed

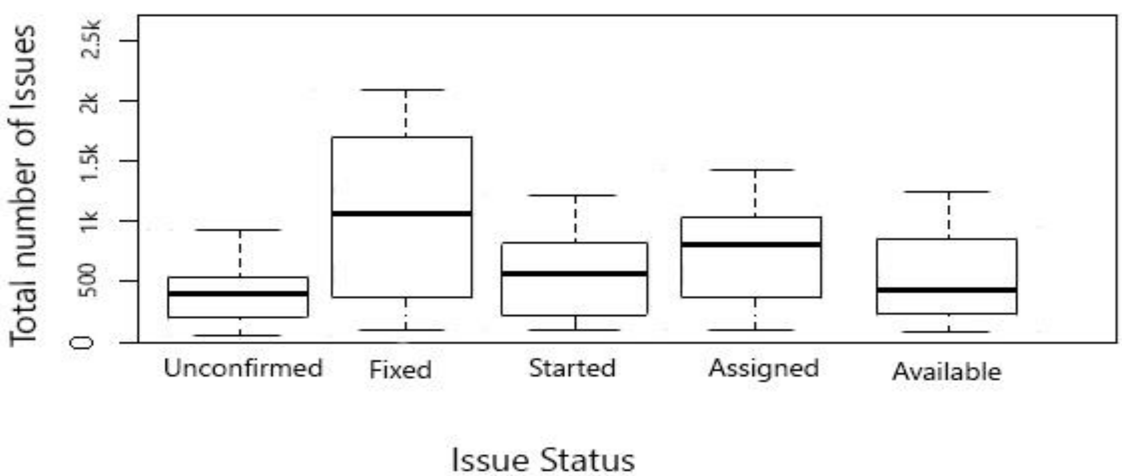

Figure 2: Box plot of bugs' status versus total number of bugs

Figure 3 is a normal Q-Q plot of the bug status versus the total number of bugs from the analysis in RStudio, which clearly shows that data is normally distributed which justifies our analysis.

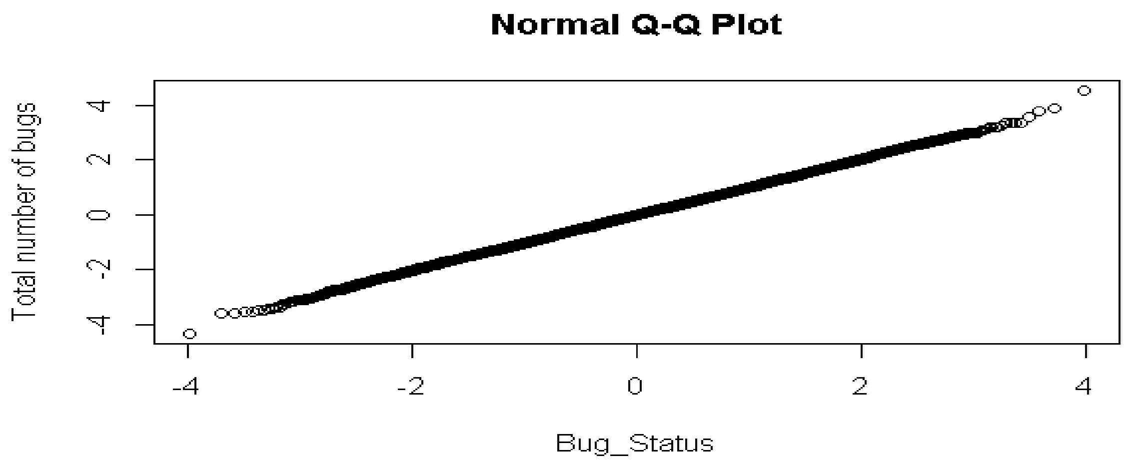

Figure 3: Normal Q-Q plot of the bug status versus the total number of bugs

Figure 4 is a box plot for cohesion and coupling of all classes across all versions together, the box plot below clearly shows that:

- Graphs are left skewed
- Low coupling between classes, only two classes have high coupling (median), so across all versions coupling is less between the classes.
- Classes are getting less cohesive.

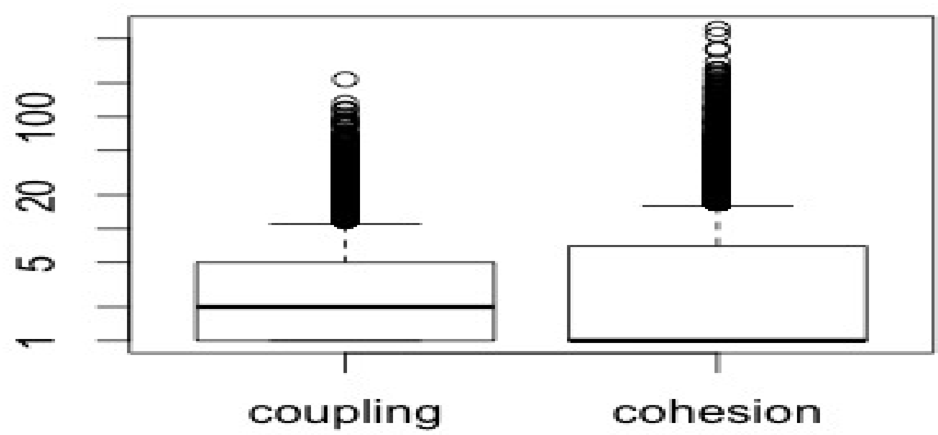

Figure 4: Box plot of coupling and cohesion for all releases

Analysis:

- First, based on our summary from the second assignment, we found the number of fixed bugs and made an attempt to find out the number of files that were altered in them.

- Second, the number of files that were changed helped us to fetch all the classes for different releases of Chrome being worked upon that we managed to get in our summary table in assignment 2.
- Third, when compared the cohesion and coupling of those classes we were able to interpret that they were less coupled and less cohesive thus for a class to be stated as a "GOOD" class it has to have low coupling but high cohesion. Conclusion was that since there was not much affect in cohesion and coupling thus these changes didn't had much effect of the performance of Google Chrome.

7. Statistical Model

Residuals:

Min	1Q	Median	3Q	Max
0.301	3.049	3.121	3.293	3.4

Coefficients:

Predictor	Estimated	Std. Error	P-val	Devnc
log(unconfirmed)	0.07	0.001	0.07	0
log(fixed)	0.30	0.004	0.00	2631
log(Started)	0.19	0.002	0.05	7
log(Assigned)	0.25	0.003	0.04	368
log(Available)	0.21	0.002	0.01	78

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual deviance: 4827
Null deviance: 9654

deviance: 0.51

Predictors that are of statistically significant ($P < .05$):

1. Fixed bugs
2. Assigned
3. Available

- Fixed bugs will help support our outcome. The numbers of assigned and available values help us predict the number of current issues that are being worked upon or are ready to be worked upon.
- The bugs that are ready to be worked upon can be used for future studies about performance of Chrome.

We used a logistic regression model and found that the deviance was equal 51%, which falls within the range of 30% to 70% that is the criteria of a good model. Hence a value of 51% justifies that our approach of analyzing the performance was correct and consequently a correct model has been developed for analyzing the data.

8. Threats to Validity

Few factors that affect the measures validity are summarized as follows:

- Not all the issues included information related to the confounding factors, which may have an indirect effect on the measures of our analysis and may result in an incorrect conclusion.
- Using an absolute count as an outcome measure to conclude things might not be statistically correct.

9. Recommendation and Actions

- Ensure proper memory management:

  After analyzing all the data we found that one of the major factors behind performance of Google Chrome was improper memory management. A number of issues revealed that for a different operating systems when 25 or more tabs were opened CPU usage would go beyond 100 % in one OS and 200% in another OS because for each open tab there was one single process dedicated to each one of them. We also found that particularly in case of MAC for each opened tab in Chrome would utilize 1GB of ram thus for a machine with low hardware specifications memory usage would increase subsequently with each opened process. Thus a proper memory management would ensure smooth running of Google Chrome.

- Focus on high priority bugs

  We saw that only 30% of bugs were fixed out of total number, but from those 30% percent of fixed bugs if we ensure that the amount of priority bugs are fixed quickly then we can significantly improve the performance of Chrome over time, even though the amount of fixed bugs were less.

- Focus should be on improving performance on all operating systems

  Another thing that we saw common in most of the issues that they were not specific to one operating system, thus they were affecting all the operating system which support Google Chrome.

  Once the bugs are fixed, Chrome should be tested on all the different operating systems rather than specific to just one operating system on which the issue was raised because majority of issues raised had problems on all operating systems, so changes made in a bug might affect the other operating systems as well.

10. Conclusion:

- Total number of relevant bugs: 13,361

- 30 % percent of the bugs were fixed.
- Median is at 6680.
- Numbers of fixed bugs is less than median

Even though number of fixed bugs is less than the median value, we can conclude that performance of Chrome has increased over time but not significantly.

References

[1] N. Fenton and J. Bieman, *Software Metrics*. Hoboken: Taylor and Francis, 2014.

[2] Gs.statcounter.com, *Top 9 Browser from Mar 2014 to Apr 2015*. 2015. [Online]. Available: http://gs.statcounter.com/#all-browser-ww-monthly-201403-201504-bar. [Accessed: 12- Apr-2015].