Terabyte Threat Analysis
Progress Report

Team Lima
For Paul Reid on behalf of BT

14th February, 2013

Contents

1 Overall Progress

As noted in the documents submitted on 12th February 2013, we have spent considerable time deliberating over the project and the best manner in which the brief should be converted to an implementation. The relatively open nature of the brief is most certainly appreciated, insofar that it provides flexibility as to the threats to be analysed, the technologies to make use of and the overall system structure.

However, our work on this has determined the system to be realised is not trivial; a finished product cannot be created by simply coupling some application programming interface (API) to a front-end Graphical User Interface (GUI). Difficulties abound when attempting to formalise the notion of a *threat* and the scope of such threats we are expected to consider; should threats be considered as granularly as individual customers or as globally as the entire network? Both topics are requirements upon which the design of all aspects of the project depend, in particular the data analysis and the visualisation components. Furthermore, we have discovered that even after the concept of a threat is settled, it remains a complex matter to transform such a definition into a technical system using off-the-shelf technologies, not to mention one which scales to processing large datasets in near real-time.

Consequently, our design work has included a period of research, during which time we have experimented with the various technologies we intend to use in order to familiarise ourselves with their interfaces and behaviour. This has primarily revolved around those technologies which will be utilised (by stipulation of our client, or otherwise) to process large datasets at scale, such as Hadoop and HBase. This has necessarily involved the writing of some preliminary programming code to experiment with these systems, and efforts have been made to make such code applicable to the future implementation wherever possible.

In addition, some members of the group have spent considerable time locating and digesting research papers from academic journals and other sources to determine the specific threats we wish to capture and how to go about doing this using MapReduce jobs for data analysis at scale. This work - following many meetings and continual adjustment to the proposed system design - has finally resulted in the design of a system which we are happy with and feel addresses the project brief most directly. Furthermore, we are aware the project is of a particularly short duration and has relatively few developers for a task of this magnitude. We feel the system design we have converged upon permits further development in the future, should BT wish to do so.

The considerable time spent in the research and design phase means implementation did not begin as early in the project as we had initially hoped. As of this writing, full implementations are now in progress. We do not feel at this stage that the time remaining to the final review and code completion will pose a significant issue due to the substantial design and project management work completed upfront. All members of the team are now familiar with all aspects of the project, the modular structure, and the manner in which aspects such as documentation and version control are to be undertaken. We feel the upfront planning will act to accelerate, rather than hinder, the implementation of the project and look forwarding to reporting positively in this regard in our final implementation report later this month.

The details that now follow concern the code and frameworks we have written and begun to use, respectively, within the project.

2 Module Progress

2.1 Importer Tool

The main bash script for converting nfcapd files into Comma Separated Values (CSV) format is complete, although testing will be required to ensure it works in all circumstances. The inclusion of five variables at the top of the file to determine directory locations ensures it works under any configuration, since these variables can be changed for any environment on any system. Naturally, there may be issues that cannot be preempted, such as a change to the *nfdump* command, or to the *hdfs fs* command, but it is hoped that Cloudera would not be releasing major updates to their software components without bumping the version number of CDH, at which point it is possible to properly plan and test any such upgrades first, to account for the slight breakage which should be expected to occur.

2.2 MapReduce Jobs

As of this point, no analytical MapReduce jobs have been created. However, proof of concept code has been used to ensure the group understands how MapReduce jobs are created, such as a distributed word-count example. This ensures the team is confident they understand how they can implement a given algorithm in a distributed manner, although the requirement of actually deciding on suitable algorithms to implement did cause significant delay in this phase. Since the Hadoop platform is relatively self-contained, reading from its own file system and writing to HBase, provided the CSV is placed in an expected place in the HDFS, and provided the HBase API remains the same, there is no other consideration that needs to be given to the portability or integration of this component.

2.3 HBase

The HBase schema has been decided and constructed, the group in agreement that it is suitable for the given task. No consideration needs to be given to implementation, other than a list of the commands required to build the same stores on the machine it is being implemented upon. The majority of the cleaner tool to maintain the data in this database is written already, using hard-coded variables to specify the age of the data to remove. These are able to be tweaked at will. Once the cron job is set up, this should also be an entirely self-contained step, since accessing HBase through ZooKeeper should require no added configuration. In addition, it is trivial to move the cleaner tool to another host provided a Java Virtual Machine (JVM) is present, as it will be packaged as a single Java archive (JAR) file containing all dependencies.

2.4 Monitor

The understanding as to what the Monitor is going to be tasked with has constantly changed through the project definition, as it is essentially the ‘glue’ between the backend and the frontend. Initially it was expected that it would have to run the algorithms, however these were pushed further into Hadoop. Then it was expected it would have to push results back into HBase, but then the PostgreSQL database was added for easier accessibility. Now it is clearer what the Monitor will do, however the fluidness of the definition means it has been very hard to create any code for it lest a specification change cause it to all become invalid. Provided the Monitor has access directly to HBase and PostgreSQL, integration is simple since there are no extra external sources to access.

2.5 Web Frontend

The core of the Python web framework backend has been implemented, showing the connection between the database and the HTML5 front-end. The front-end web interface itself has a very basic but functional layout which will be polished towards the end of the project. As of this point, no actual mock or sample data has been processed or visualised by the front-end but progress has been made in defining exactly how the data will be shown. Provided there are no problems with formatting the data for use within the JavaScript graph libraries, creation of the graphs and statistics to be displayed is simple since it is just the case of passing data correctly to the JavaScript libraries.

3 Milestone progress and future deliverables

As noted in the original *Project Plan* document, the milestone for delivery of this report is 14th February 2013, at 2pm. This report has been prepared and, by virtue of your reading it, delivered successfully.

Furthermore, by the second review meeting, on 15th February, we had initially committed to having a working demonstration of the core of our system, minus any GUI implementation. It would appear that we will have prototype code to demonstrate the design and implementation of individual modules, but at this time, we are unable to commit to a full data flow through all the back-end modules by the next review meeting. We will, of course, be able to speak extensively about the design of each module and the further time commitment required to implement them.

The hard deadline for completing the overall translation of concepts and designs to a concrete implementation is the third and final review meeting, on 28th February 2013. Active steps are in the process of being taken to ensure this deadline is comfortably met.