Terabyte Threat Analysis
Final group report

Team Lima
For Paul Reid on behalf of BT

27th February, 2013

Abstract

This report concludes all formal paperwork provided by Team Lima over the course of the Terabyte Threat Analysis project. It aims to review the specification as per the start of the project, and reflect on exactly how much of it has come to fruition. It aims to describe deviations from this plan, and describe specifically how the final product has been produced. Documentation and the final codebase will follow at the point of the code freeze, serving specifically as the physical deliverables that complement this report.

Contents

1 Project Specification Retrospective

The first consideration in the completion of a product is how well it conforms to the original specification. In the case of this project, the initial specification was incredibly fluid and loose prior to the investigation phase. This was an intentional design decision that would allow alterations up until the rigidly structured report submission. This extra phase did allow us to research new technologies and become sure that the products would fit together in the way planned, but the introduction of so many unfamiliar products did stunt the team's ability to be confident it was a somewhat-optimal layout.

When the project specification's footnote numbers are taken into account, it is clear that there are a large number of software projects upon which we intended to hang our own project. Libraries and helpers of various kinds that would massage the data into the correct formats for the correct phases of the workflow. However, now that the product has been created and both the Java `/lib` folder and the HTML `/js` folder can be examined, approximately thirty libraries ended up being referenced. While it is the case that Hadoop is dependent on some number of these and we were not aware of that, it still represents a large amount of external dependencies. As with any libraries, we are aware that updates to the libraries may or may not be backwards compatible, and the more libraries are in the project, the more likely general development may need to version bump an existing library.

The project went into great detail as to how each module that fitted together would work. Tweaks to modules shifted the exact position in the chain in which certain processing would go, leading to slight variations in the exact workflow. Although flowcharts were drawn up, the documentation does indicate changes to exact storage mechanisms and formats, such as the removal of 'warning' states in order to simplify migration of data from one database to the other.

2 Project Successes and Failures

Ultimately, the project did result in a deliverable that the team believes is a good milestone on the roadmap to BT being able to analyse their router log data with ease. Given the complexity and enormity of being the first to tackle this problem, there are undoubtedly some issues with the implementation. The initial images in all team members' heads were clearly very different. From the very first meeting, there were concepts that this would be for data analysis, for visualisation, for detection of notable activities and even for creating internal tickets to the relevant departments on each issue. The unified concept took its inspiration from all of these areas combined, the broad scope of the requirements specification a testament to this.

Naturally, as with any new area to be explored, it was gradually apparent that certain actions were more complex than initially perceived, or even not possible with the given workflow. The format of Hadoop jobs, splitting the data up and processing it without the benefit of the lines above and below in certain edge cases for resolution led to some complex and slightly inefficient workarounds. This was necessary, however it is natural for pride in code to lead a writer to desire a more optimal and exact solution in these cases.

Since Hadoop is a highly distributed system, running across many different machines in order to execute jobs at its full capacity, it was initially suggested that some Amazon EC2 instances may help to test the project at scale on large datasets. Unfortunately, this was never able to be completed. The team was neither able to acquire the server time, nor the extra data that would have been needed to feed into the jobs. The non-Hadoop portion of the workflow was held remotely on a VPS for the purpose of using a common database instance, but ultimately the team had a single hour of data in a slightly different format than would be fed into the final product, in that it was the coalescing of results on multiple pieces of hardware. This was worked around to the best of the team's abilities, but it is still impossible to guarantee or even be strongly confident that the code written will work immediately upon integration, as we were unable to ascertain generated filenames, folder structures and the like. We have done our best to leave configurations capable of adapting to the operating environment, but this still cannot possibly cover 100% of system configurations.

The team had initially intended that a resolution to this would have been to demonstrate the product at the second client meeting, then gain feedback on exactly what portions would or would not work correctly when using BT's infrastructure. This, however, failed to materialise due to issues in actually having a working prototype at that time. Although the discussion during the meeting was nevertheless useful, it was evident that the fact the client would not see the project before the final result was being documented and polished would harm the team's ability to report positively on implementation possibility.

The one method used by the team to test involved Cloudera's CDH4 virtual machines. These came preinstalled with all of the tools required to spin up a Hadoop instance on a single machine, paying the performance cost associated with virtualisation. A second drawback, however, was the extreme RAM demands the machine made.

The virtual machine alone required over 3GB of RAM. When the host Operating System was added, it was clear that even on 4GB of RAM, this would be infeasible. This even led to one member of the team having to upgrade his RAM in order to have a usable build and test environment.

3 Salient points from development

Version control was incredibly beneficial. It was often the case that we all swarmed to work on the code at the exact same time, and sometimes even on the same file for certain reasons. Without a version control system, we would have struggled incredibly to coordinate. Although it would initially appear that the modularity of the entire project would allow for each developer have a relatively unique set of files to create and test, the amount of message passing between phases caused much blurring of lines. Developers would have to add extra data in the sender so their receiver would have a usable result, or have to modify the receiving component to take extra data that had to be passed at a given point. In addition, the sections were each their own package, however certain packages had common concepts of enumerations and status codes to the point where they would have to be modified in lockstep to remain functional.

It appears to have been a common consensus that, while the team appreciates the fact BT left the scope of the project so open, it would have been useful to have some guidance as to the exact target of the final product. The fact that the project was presented with the caveat that "*we[BT] don't exactly know what's possible*" led to us having to make mistakes travelling down routes that would not provide any of the facility required. Factoring in some of the Apache incubator's poor documentation caused confusion and unease about certain directions that the product could take or how to interact with their APIs.

Although it was initially believed that more data would simply be a 'nice addition,' and that we could either use the same data over and over as the test case or generate our own, it was clear that this was not true. For one thing, the file given was a coalescing of multiple routers, leading to duplicate rows and anomalies that meant we could not tune the parameters for our rule based functions on what was given. In addition, our own random data was harder to generate correctly than initially thought, and would also have not been possible to use to tune our variables since it were not real-world. This led to parameters having placeholder values in the final product, with stress placed on the fact that the value would require changing to be of any use in the real world.

In addition, the team was increasingly aware that the documents regarding language choice advise using Java unless circumstances lead to another being truly necessary. Creating the initial importer as a cron-launched shell script was an immediate and correct design choice. The importer required the ability to call command line conversion and import programs, and doing so natively was critical. The selection of python for the web based interface was initially controversial, although the final product does clearly show an interface with features far greater than Swing could ever provide, specifically given the client's description of the existing web-based statistics portal in Splunk.

4 Work distribution

The initial plan of the work distribution was relatively unchanged.

As intended, Aaron took control of the main web interface, coding the python and HTML sections, plus setting up the hooks for earlier components to notify it of changes. Since he would be reading from the PostgreSQL database, he also designed its schema.

Jan and Ernest together set up the entire Hadoop infrastructure and its interactivity with HBase. As the writers to HBase, they defined the key and value layout for the database. This was a significant amount of work with new technologies, although Ernest did have some benefit in experience in the base concepts.

Alex was initially in control of the cleaner tool, and tasked with extra work where there were opportunities to contribute. Near the end of the project, the Monitor was a critical package to complete and he was able to assist in creating some of the classes for transferring the data between the databases.

Matt and Simon were initially willing to take on the role of documentation and general 'sysadmin' tasks for the project, as well as implementing the importer and Monitor. Matt wrote the base framework for the monitor and handled the connections to the two databases, while Simon initially wrote the shell script importer, then moved on to the Monitor where alterations were needed. Between the two of them, they produced a significant portion of all paperwork submitted to the client, coupled with reviewing the generated Javadoc and comments with the intention of producing a comprehensive package of paperwork the client could use for implementation.

5 Team Conclusion

Overall, Team Lima is very pleased to have been given the opportunity to do a project such as this one. In comparison to some of the possible projects, this had a much larger scope to make a significant benefit in the real world. We hope that what we've been able to produce helps BT understand what their data set is useful for, and we look forward to being able to present the final code and documentation at the end of the project.

The team agrees that this project was complex, and that there were certain roadblocks associated with the research-driven nature of the product, but we are confident that building on what we've been able to achieve should set the foundations for a more intelligent and wide-reaching analysis tool for the netflow data.

Since the start, the group was always drawn to the mindset of using the most modern and fully featured tools available for development. From Apache incubator projects that are very new and innovative to git and Google Docs for collaborating in real time, the entire period has been a true example of how much the workload can be eased by identifying relevant open source or otherwise free tools and libraries.

The project was not without its downsides, and the team has spent many a late night trying to push code and paperwork forward to meet deadlines, but that is simply a natural part of projects and the inherent belief that certain components will take less time to implement than they do.

Most importantly, Team Lima would like to extend thanks to both the Computer Laboratory and BT for the experience, and hope that the result of these six weeks is a fitting culmination to a period of dedicated work by all six members.