

Qlikview Load Script Profiler v1

Chris Cugliotta

March 4, 2016

Contents

1	Introduction	2
2	How to use	2

1 Introduction

It is important to know how much RAM a load script will consume. Sometimes, the entire loading algorithm will seem stable, but then one bad statement causes the whole operation to blow up. This tool aims to reveal these bad statements. Moreover, it aims to quantify the RAM usage of each individual step in a load script, to provide a complete picture of RAM usage over time.

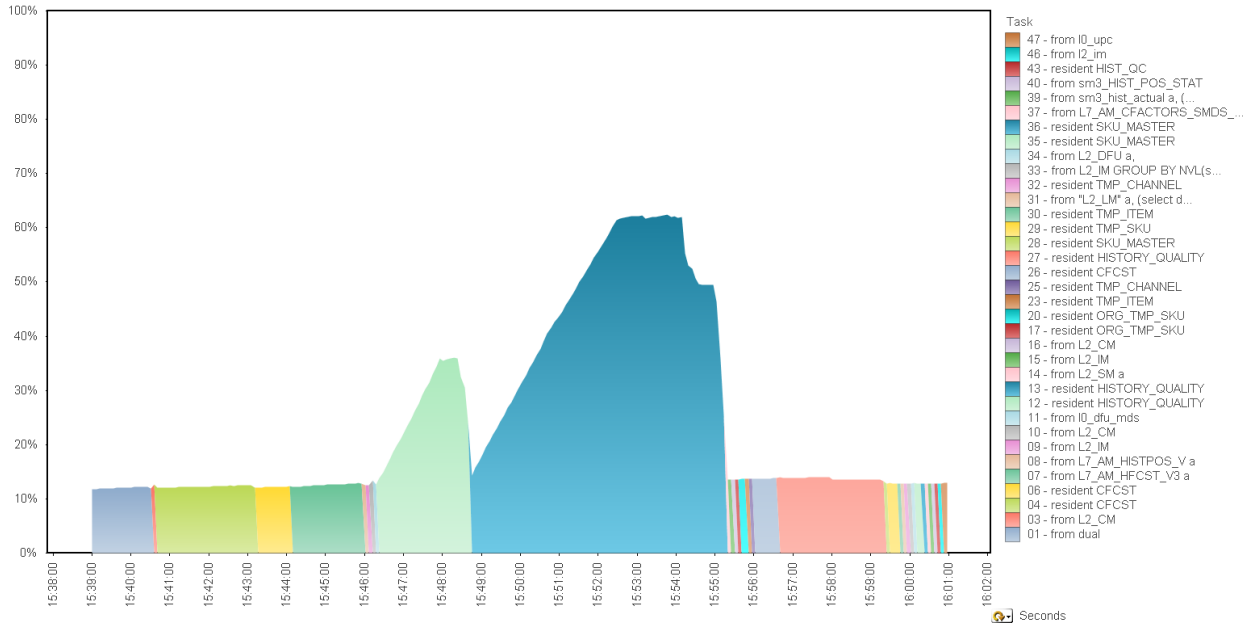


Figure 1: RAM usage vs. time

In the example above, we see two RAM spikes in teal and blue. Using the legend on the right-hand side, we can see these spikes correspond to the 12th and 13th statements in the load script. These statements are both *resident group by* statements, which often consume a large amount of RAM if the aggregated row count is too high.

This caused our client's Windows machine to crash. We fixed this issue by modifying our algorithm. Specifically, we offloaded the *group by* calculations to the Oracle machine, since it had stronger hardware than the Windows machine. In other words, we replaced the *resident group by* statements with equivalent *sql select group by* statements. This tool gave us the visibility to detect and fix the problematic statements.

2 How to use

This report has two inputs:

- **Memory log:** This is a comma-separated vector containing periodic timestamps and memory usage. It is created by a batch script called `memory.bat`, which will quietly monitor and record the memory used by `qv.exe`.
- **Load script log:** This is a Qlikview log file. Upon reloading a report, it will be generated automatically if you have the Settings → Document Properties → General → Generate Logfile checkbox checked.

The load script will cross-reference these two inputs, and produce a graph similar to Figure 1. To ensure the two inputs are properly formatted, the following steps are recommended:

1. Open up a report in `qv.exe`.
Note: Make sure you only have 1 instance of `qv.exe` running!
2. Go to Settings → Document Properties → General tab, and make sure Generate Logfile is checked.
3. In the load script, make sure `TimestampFormat = 'DD-MMM-YY hh:mm:ss TT'`.
4. Double click `memory.bat`. A command prompt should appear. It's running a loop, recording a snapshot of `qv.exe`'s memory footprint periodically over time. Don't x-out. Let it run quietly in the background.
Note: `memory.bat` and `qv.exe` should be running on the same Windows user!
5. In Qlikview, begin a reload.
6. When finished, x-out of the command prompt.
7. At this point, you should have two files: `memory.csv` (created by `memory.bat`) and `report.log` (created by the 'Generate Logfile' checkbox). The file names may be different.
8. Open the memory profiler report. In the input box, specify the appropriate input file paths. Then reload the document.