System Analysis using the System Profiler#

Analysis Introduction#

So you have received a kernel instrumentation log file, or you have been asked to start using the system profiler to expose and understand the inner workings of a system. This page looks at some of the details of how to get going with this type of system level analysis.

Capturing a trace log file#

There is no use in discussing System Profiling without first discussing how one goes about acquiring an instrumented kernel trace log file.

Short of having the log file thrust upon you by someone else, there are two ways to capture your own trace log file:

- Use the tracelogger command line utility
- Use the IDE to capture it via the qconn target agent

There is currently a significant difference in functionality and performance. The tracelogger utility provides the lowest latency data capture by allowing a direct to shared memory data capture (via the -M and -S flags), which should be used by systems that are already under significant CPU load. There is limited event filtering capability provided by the tracelogger utility.

The IDE trace capture, accessed via the 'Log' command in the Target Navigator view, provides a direct event filter and a streaming capability direct to a host system; however, it will have a higher oprational overhead, require that a target be connected to the network, and be capable of running the qconn target agent.

Once the log file is captured (using either method), it can be analysed using the IDE System Profiler tools.

System Profiler IDE components#

Several components make up the System Profiler tooling in the IDE, and the most appropriate location to perform trace log file analysis is within the System Profiler perspective (or your own perspective composed of System Profiler views).

Editor Panes
The System Profiler editor is structured so that nearly all tooling is structured around the active System Profiler editor used when you "open" a *.kev trace log file. For navigable types of data analysis, custom editor panes are created that represent the log file in different ways, or with a different focus on a specific aspect of the log file data.

Editor panes can be toggled and the display can also be split so that you can do comparisons between different views of the data at the same "time point" by locking the different panes together. It is also possible to open a new editor window to completely decouple your display.

Editor Toolbar
The editor toolbar, like all editor toolbars, is tied to the current System Profiler behaviour. This means that in order for any of these actions to be active, you must first select an editor before issuing a command. This toolbar provides common navigation controls, such as allowing zoom.

Views

The views associated with System Profiling generally show snapshots of information that are not tied to a timespan duration, but are result oriented and can assist with point navigation in the associated editor. The information tends to help drive and direct navigation.

Search
The editor provides a custom search capability that is specialized for instrumentation trace log files. Users create re-usable conditions that identify event characteristics that can then be used for search and other tools, such as filters and statistics.

Quick "view" overview#

Trace Log/Raw Event Data/Properties
These are the primary views to use when you want to look at the detail for the individual events. The size of this view is limited to a 'window' around the current editor selection, and can be controlled by a setting in the Preference panel.

For analysis purposes, you can synchronize and unsynchronize the view with the editor filters. This allows you to look "out of band" at events that you may have removed from the main editor display. This is particularly interesting when you filter out interrupt events, for example.

Filters / Timeline State Colours
These views help to highlight specific information in a timeline display by removing elements from the display and use colour to emphasize the information that is most relevant.

The filter view is displayed when you select "Filters ..." in the Timeline editor pane, and it controls the filters that can be generally applied to the Timeline display. The filters play a significant role in decimating a log file down to a set of processes that are of particular relevance to what is being examined.

General/Event Owner/Condition Statistics & Client Server
These views are used to help condense information contained in a trace log to bundles of data that can be used to further refine analysis. All of these views provide the ability to examine data in a global manner (an entire log file), or to look at a specific user selected region.

Why Running/Thread State
These views provide automated support for manual backtracking based on a particular point in time as selected in the log file. The Thread State view extracts all of the thread states for a selected time, while the Why Running view performs an automated backtrack to determine what caused the currently running thread to run.

Performing an analysis

There is no magic formula for analysing log files, and in most cases, the analysis will ultimately end up examining the specific sequence of system events. However, there are some techniques that can be used to effectively narrow down into an area of interest when you are looking for "odd behaviour":

Look at the big picture
Use the System Summary view and the General Statistics view to get a quick feel for what kind of work is dominating the system and where to proceed next. Here are some questions you may want to ask the provider of the log file:

- Is this a system that should be maximizing the CPU but continues to have idle time?
- Is this a multi-CPU system that should be evenly balanced?
- Is there a lot of HW interaction expected or only a little?

Use handrails and fenceposts

Generally, you are taking traces during periods of time where something of interest is occuring. To help drive the analysis, using two navigation techniques will help you orient yourself:

- Handrails are features that help guide you towards something you are interested in
- Fenceposts are markers along the way that you note (or leave) to help backtrack

Translated to log files, this means looking for abnormal transitions in the log and leaving bookmarks behind to get you back to those places if you need to revisit them.

- Use the Bookmark action to leave yourself interesting markers
- An increase/decrease in CPU consumption (CPU Usage editor pane)
- An increase/decrease in event generation (search, condition statistics)
- Mis-matched running/ready ratios (general statistics)
- Significant difference in max./avg. state times (general statistics)

Trust your eyes, use your brain
More than anything else, the visualization tools provide guidance for your eyes and brain to take over, and look for trends and repetitions in the data.

- Adjust timeline event colours and size to make data "pop"
- Look for groups of event markers that start or stop repeating
- Watch for expanding/contracting groups of events

Use your system knowledge
You may frequently be asked to help analyse logs where you have no knowledge of the system. In these cases, you need to leverage your Neutrino system knowledge to help direct you to the true root of a problem. From here you can:

- Look into the clients for servers with high CPU consumption (client/server CPU statistics)
- Look for synchronization mis-use (sempahore mapping to condvars)
- Identify pre-emption by looking at priorities
- Weed out bad polling behaviour ... asynchronous notifications rock!
    - Nanosleep states
    - TimerTimeout kernel calls

When in doubt, start looking anywhere
You can certainly use the above techniques to intelligently narrow down your focus to a "better" starting point for event level analysis. However, in most large software systems, the log file will contain enough information that you can likely start at any point, and start providing valuable feedback.

Common Questions and Answers#

Q: I specified 5 seconds of capture to tracelogger but I don't get 5 seconds of data.
A: It is likely that you are missing the -n0 option and are filling your kernel buffers before you encounter the time frame.

Q: How do I get data out of the System Profiler tables?
A: All of the tables should support copy and paste, though for some tables it may not be possible to use a direct copy, and you will need to use a right click menu action to perform this. Once you copy the data, it will be transformed into CSV format. You can import this into a spreadsheet, or if you want to paste it directly, you can use the Data > Text to Columns menu item to convert the pasted data into proper data cells.

Q: Why am I not getting any priority (or partition) information?

A: You need to capture state events in wide mode for this to occur. In particular you really only need to capture the RUNNING state in wide mode because only the RUNNING state really matters (though READY is handy to have as well).

Q: Why does my CPU usage not line up with the RUNNING state statistics?
A: This is because the CPU usage takes into consideration, as much as it can, the effects of interrupt processing on the system, in addition to the RUNNING state of the thread.