Vampir 5.0 – Getting Started

Vampir is a software tool for analyzing the performance of parallel applications. It visualizes runtime information by means of event traces gathered by monitoring software like Vampirtrace, TAU, and KOJAK (see below for further details).

The Vampir tool translates tracefile information into a variety of graphical representations that give developers a better understanding of performance issues concerning their parallel application. Vampir allows for quick focusing on appropriate levels of detail which allows the detection and explanation of various performance bottlenecks such as load imbalances and communication deficiencies.

This documentation gives a brief description on how to install and start Vampir. Furthermore, it provides an overall view on the key features of the tool.

1 Installation

To run the Vampir tool, a platform specific executable and a license file are needed. Demo versions are available from `http://www.vampir-ng.com`. Please, contact `service@vampir-ng.com` for buying permanent licenses.

The easiest way to install Vampir is to put the executable program and the corresponding license file *lic.dat* into the same directory. The directory can be either user specific or system wide. Upon startup, Vampir scans the binary search path for license files. Additionally, the environmental variable "VAMPIR_LIC_FILE" can be used to point directly to a license file. After that start Vampir by simply executing the binary.

2 Generation of Trace Data

The generation of tracefiles for the Vampir performance visualization tool requires a working monitoring system to be attached to your parallel program. A new version of our monitoring facility is currently in developement and will be available soon, as part of the Vampir tools framework.

In addition you can use the monitoring components of either TAU or KOJAK which are both free software and available for many platforms. Both monitors are able to write program traces that can be analyzed and displayed by Vampir.

TAU: `http://www.cs.uoregon.edu/research/tau/home.php`
KOJAK: `http://www.fz-juelich.de/zam/kojak`

3 Loading Trace Data

After having started Vampir the main window is visible. To open a tracefile, select "Open Tracefile..." in the "File" menu. It is possible to filter the files which are visible

in the list. The filename input field allows to determine the mask for the files that should be displayed. Clicking the “Filter” button updates the list whenever a new mask is specified. Using “*.vtf” for example, shows all files in the classical Vampir trace format. All file types can be displayed by using “*”.

Vampir can also be started with command line options. Typing “vampir <trace-file>” on the command line starts the tool and immediately loads the given tracefile. The tracefiles to be loaded have to be compliant with one of the OTF or VTF tracefile formats.

The loading process is indicated by a progress bar. It can be interrupted by clicking on the “Pause” button and continued by clicking on the “Resume” button. Figure 1 shows this display. After loading has been completed the status bar beneath the main menu will display the tracefile’s name.

Figure 1: Main Window While Loading

4 Timeline Displays for Event Visualization

Vampir supports different display types for illustrating trace data. They can be divided into timeline and statistical displays. Timeline displays show detailed event based information for arbitrary time intervals while statistical displays reveal accumulated measures which were computed from the corresponding event data. There are global displays, which show information for a number of processes, as well as process specific displays.

4.1 Timeline

The timeline display presents runtime information for processes which is displayed on a timeline axis. It provides detailed information about executed functions and communication/synchronization operations. Certain classes of functions, e.g. MPI, Computation etc. are grouped and illustrated in the same color. In Vampir terminology, these groups are called “activities”. They are defined by the trace generator or by the programmer individually.

Event timelines are available for a collection of processes as well as for individual processes. The corresponding windows can be opened via the “Global Displays-> Timeline” and “Process Displays->Timeline” menu entries in the main window. The “Process Displays->Timeline” menu entry remains inactive until at least one process is

selected. To select a process click on its label in the global timeline or on its bars in the global activity chart. Figure 2 shows the global timeline with two selected processes.

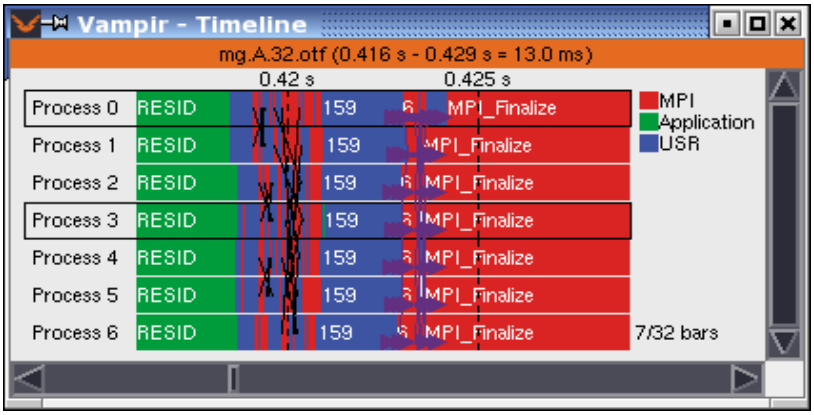

Figure 2: Global Timeline With Two Selected Processes

Process timelines show the different levels of function calls in a stacked bar chart and in contrast to the global timeline have the opportunity to add process-specific counters to the display. Figure 3 shows a process timeline.

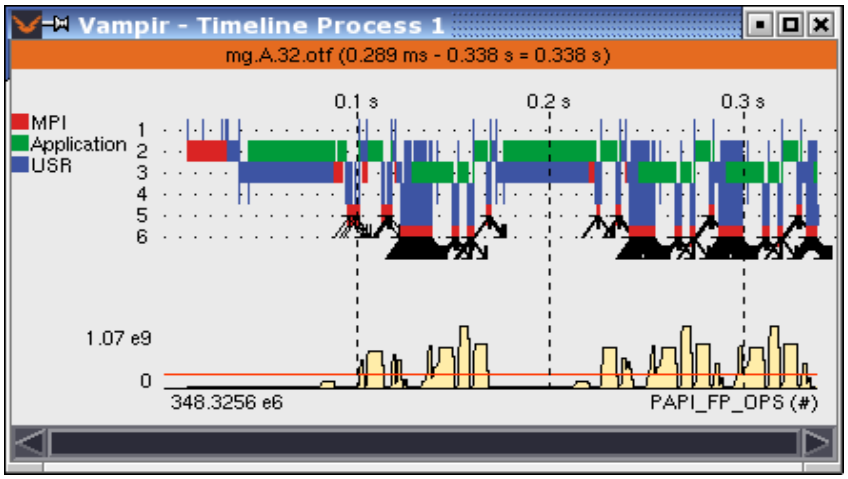

Figure 3: Process Timeline For Process 1

4.2 Summary Timeline

Clicking on “Global Displays->Summary Timeline...” opens a window which displays the number of processes that are actively involved in a given activity at a certain time. The display is useful for studying communication overhead and load imbalance issues from a high level perspective. Figure 4 shows the summary timeline.

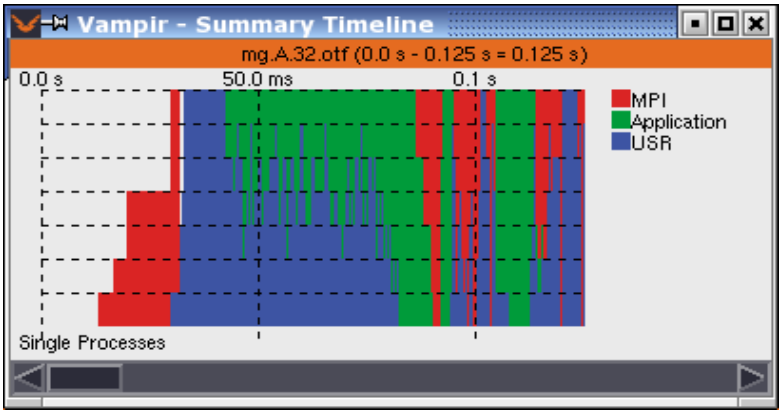

Figure 4: Summary Timeline

4.3 Counter Timeline

Depending on the monitoring infrastructure, counter measurements can be included in the trace data. Click on “Global Displays->Counter Timeline...” to visualize the measurements of a single counter like *number of cash misses* or *number of floating point operations* for multiple processes over time. Figure 5 shows the counter timeline. Additionally, counter presentations are accessible in single process timelines via “Process Displays->Timeline...”. In this case multiple counter types for one process can be studied simultaneously (see figure 3).

5 Statistics Displays for Accumulated Performance Measures

Click on “Global Displays->Message Statistics...” to view message statistics information. This display consists of a table where the rows are the sending processes and the columns the receiving ones. The default entries show the sum of the message lengths transfered between two processes but can be changed with the context menu entry “Display” (e.g. into minimum/maximum rate or duration). The statistics for collective operations found under “Global Displays - Collective Op. Statistics...” are equivalent in structure to the message statistics only that they shows information for collective

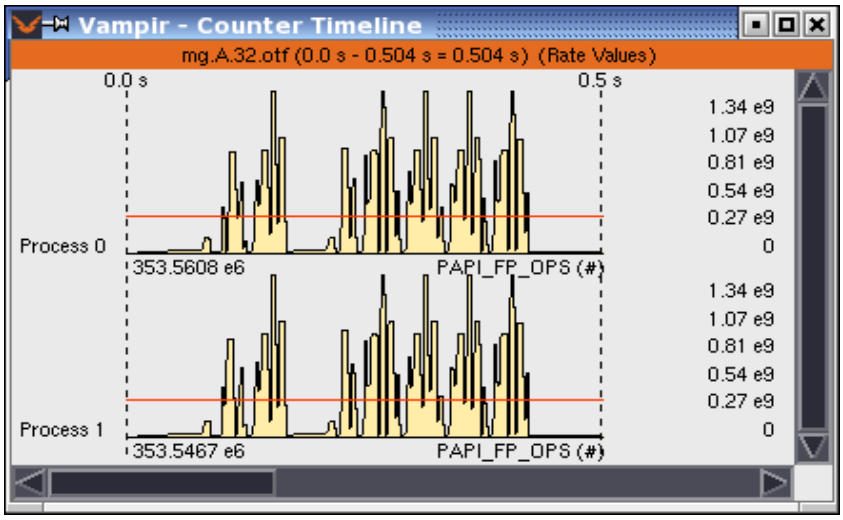

Figure 5: Counter Timeline

communication like MPI_Bcast or MPI_Barrier). Using “Global Displays - I/O Events Statistics...” opens the file I/O statistics. This displays structure is also equivalent to the message statistics but the processes which perform I/O operations are the columns and the accessed I/O files are the rows.

5.1 Summary Chart, Activity Chart and Process Profile

To open an overview of the time consumed by the different activities or the number of activity occurrences use the following displays. For the overview on the time/occurences over all processes click on “Global Displays - Summary Chart...”. Figure 6 shows this display.

To view the information of the activities for every process in seperate bars use “Global Displays - Activity Chart...” and for every process in a seperate window choose “Process Displays - Activity Chart...”. The process activity chart menu entry remains inactive until at least one process is selected. To select a process click on its label in the global timeline or on its bars in the global activity chart. The process profile opened with “Global Displays - Process Profile...” limits the view to one activity for every process in seperate bars. The displays can be changed, by context menu options, into a variety of possible formats (e.g. exclusive or inclusive times, times per process,...).

5.2 Call Tree

The call tree has the function of illustrating the structure of function calls of the trace in a tree-like representation. The display reveals information about the number of calls

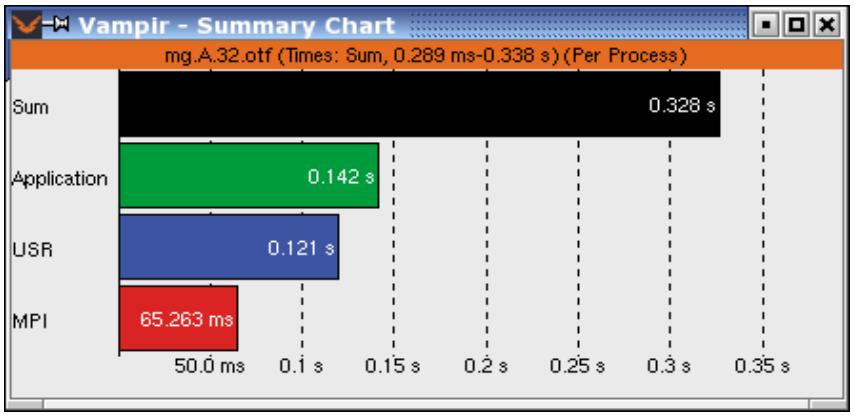

Figure 6: Summary Chart

for a function (first value range on the right side) and the times spent in the different calls (second value range on the right side).

6 Information Filtering and Reduction

Because of the large amount of information that can usually be found in tracefiles it is very useful to reduce the currently displayed information according some filter criteria.

There are different ways of filtering. It is possible to limit the displayed information to a certain choice of processes or to specific types of communication events (e.g. to certain types of messages, collective operations or I/O events).

The process filter is used either because some processes behave similarly or a problem was already found in a subset of processes and is to be analyzed in more detail. The process filter shows all processes arranged in a tree like representation. It is possible to select or deselect single processes or whole groups of processes. Process groups can be defined while tracing and then are displayed in the process filter window.

7 Customization

The appearance of the program (e.g. the color style for the entire program or for some displayed components), the starting behavior or some file search paths can be altered by choosing the “Preferences” sub-menu. If the configuration of the program was changed with the preferences menu and Vampir is closed, a dialog is opened which asks for saving the actual configuration. The standard configuration is saved in the file *“vampir.cnf”* in the directory `$HOME/.VAMPIR_defaults/`. If this folder and a default configuration file do not allready exist when starting Vampir, they are created automatically in the users home directory.