



Improving Support for OpenSHMEM in TAU

Milestone 3

Subcontract Number: 4000146681

Contractor: ParaTools, Inc.
2836 Kincaid St.
Eugene, OR 97405
(541) 913-8797

Overview

This report documents the completion of Milestone 3 for Oak Ridge National Laboratory (ORNL). Completion of this milestone requires compliance with the following,

“Milestone 3: TAU with support for generating OTF2 traces for OpenSHMEM.”

The software was delivered to the Company on 6 September 2017 and can be downloaded from <http://tau.uoregon.edu/tau.tgz>.

Introduction

Users of the OpenSHMEM library need simple, yet powerful performance evaluation tools that capture high-utility information and present it in meaningful ways, such as how much time is spent in OpenSHMEM routines, when and where these routines are called in the source code, and on which processing elements (PEs). They need tools that minimize manual steps needed to generate performance data (e.g., work with unmodified binaries to reduce source instrumentation) and perform automatic analysis of key metrics (e.g., the extent and volume of communication). To address these concerns, we have extended the TAU Performance System® to better support and simplify the creation of performance traces for OpenSHMEM applications, by natively generating OTF2 traces so output from TAU may be viewed in the Vampir trace visualizer.

OpenSHMEM OTF2 Support

We have implemented support for native generation of OTF2 format traces of OpenSHMEM applications in TAU. This allows a user to observe detailed information on the execution of an application, including the times and durations of each OpenSHMEM call invoked, as well as the time, source, destination, and size of each communication. These traces are generated by TAU using the libotf2 2.1 library, which will be downloaded and installed by TAU when “-otf=download” is passed to the TAU configuration script. OTF2 support is enabled at runtime by setting the environment variable TAU_TRACE_FORMAT=otf2. Traces generated with this configuration are suitable for viewing with the commercially available Vampir trace visualizer.

Native OTF2 trace generation provides several advantages over the generation of traditional TAU traces. The OTF2 format provides event types specifically suited for representing OpenSHMEM one-sided communication, which are output by TAU when they are encountered in place of the traditional representation of one-sided communication as if it were two-sided MPI communication. OTF2 is also highly space-optimized, producing smaller files than equivalent traditional TAU traces.

We tested OTF2 support with various SHMEM implementations on multiple computing systems available to ParaTools, including Linux workstations and clusters, Cray XC30 and Cray XC40 systems, Godzilla (University of Oregon) and Talapas (University of Oregon). SHMEM implementations used were OpenSHMEM Reference 1.2 and 1.3, Sandia OpenSHMEM 1.3.3, and Cray SHMEM. We tested with the ISx integer sort application (<https://github.com/ParRes/ISx>), the NAS Parallel Benchmarks

(<https://github.com/openshmem-org/openshmem-npbs>), the GUPS random access benchmark (<https://github.com/openshmem-org/gups-shmem>), and miscellaneous kernels (<https://github.com/openshmem-org/openshmem-examples>).

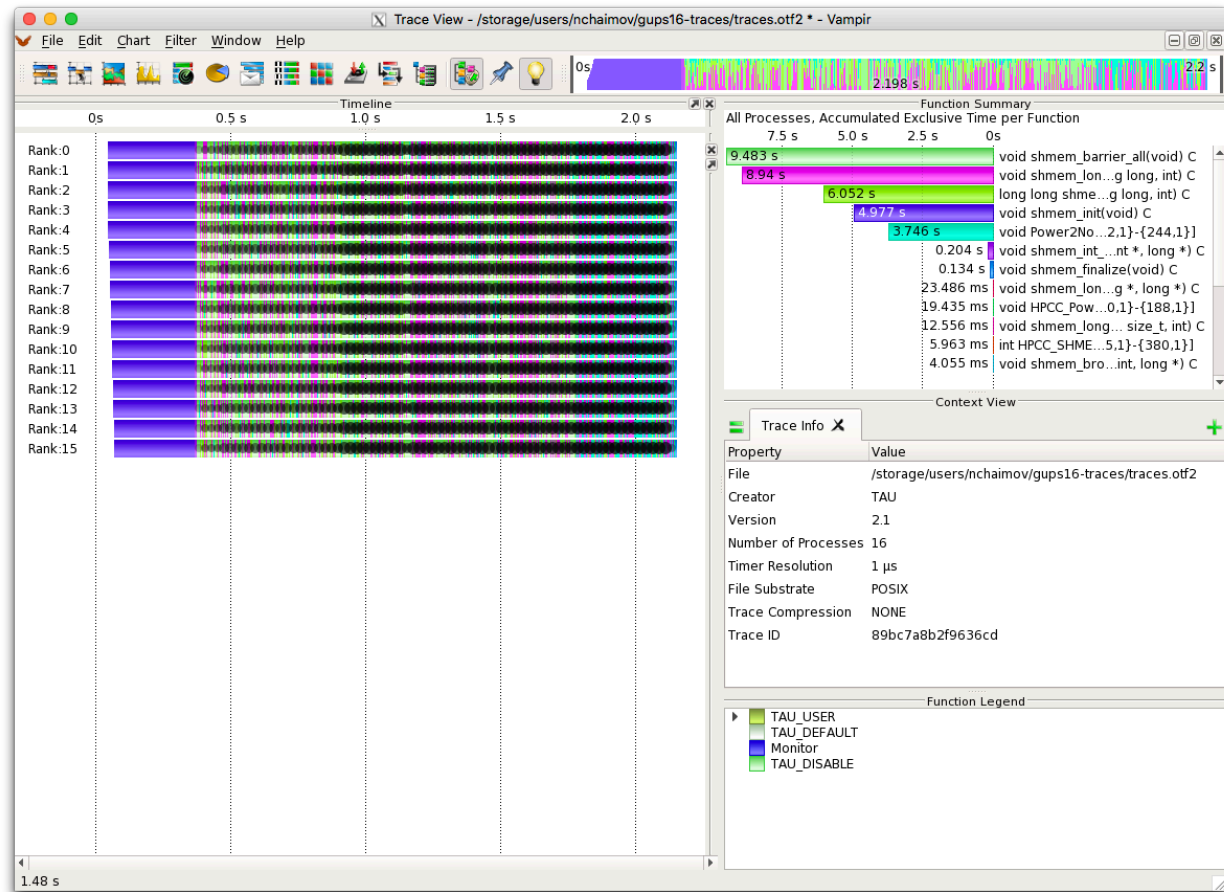


Figure 1: Vampir visualization of a TAU-generated OTF2 trace of GUPS on Godzilla with OpenSHMEM Reference 1.3.

Figure 1 shows Vampir visualizing an OTF2 trace of the GUPS benchmark executing on Godzilla with OpenSHMEM Reference 1.3. Colored regions of the timeline view indicate the functions that are executing at particular points in time during program execution, and the function summary chart on the right summarizes the time per function. Black circles on the timeline view indicate “bursts” of communication, with the size of circle being proportional to the message volume during the interval. Figure 2 shows the same trace zoomed in to reveal individual messages. Selecting a message reveals its type in the context view on the right: the selected message is a Put of 8 bytes.

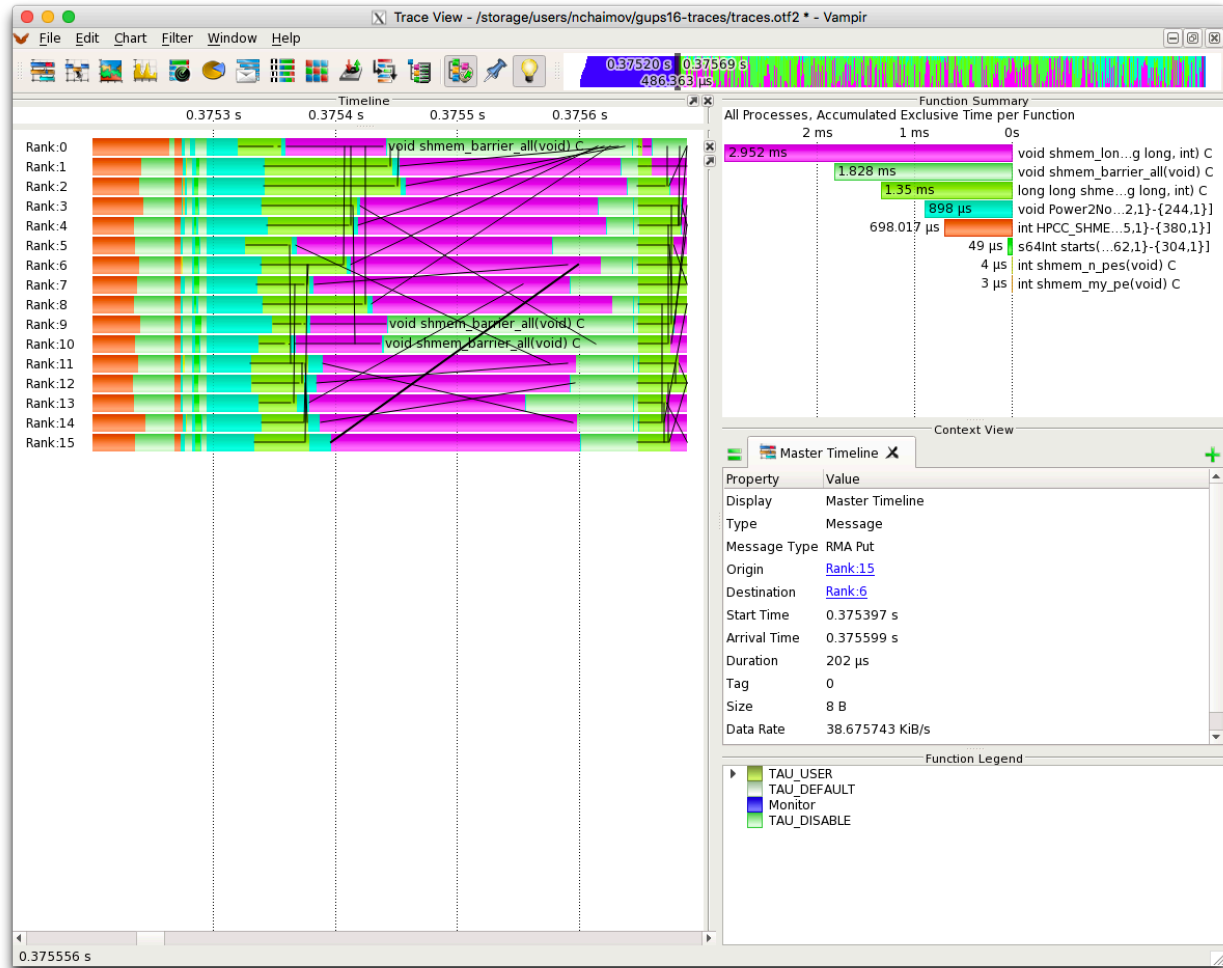


Figure 2: Vampir visualization of a TAU-generated OTF2 trace of GUPS on Godzilla with OpenSHMEM Reference 1.3, zoomed to show individual messages.

Summary

We have implemented support for generating OTF2 traces of OpenSHMEM applications in TAU. We have tested this support on various SHMEM implementations including OpenSHMEM Reference 1.2 and 1.3, Sandia OpenSHMEM 1.3.3, and Cray SHMEM. We have tested on multiple computing systems available to ParaTools including Linux workstations and clusters, Cray XC30 and Cray XC40 systems, Godzilla (University of Oregon) and Talapas (University of Oregon). We tested with the ISx integer sort application, the NAS Parallel Benchmarks, the GUPS random access benchmark, and miscellaneous kernels provided as tests with the OpenSHMEM reference implementation.