OpenSpeedShop

www.openspeedshop.org

Open Source, Multi-Platform Linux® Performance Tool

- No need to recompile; run existing binaries under OpenSpeedShop to view several types of performance data
- Comprehensive performance analysis for sequential, POSIX threaded, MPI, OpenMP, and hybrid applications
- Supports both first analysis steps as well as deeper analysis options for performance experts
- Easy to use GUI and fully scriptable through a command line interface and Python
- Supports Linux systems with Intel, AMD, ARM, GPU, Intel Phi, Power PC, Power processors, including Cray and IBM systems
- In production use on all major cluster, Cray and IBM platforms at LANL, LLNL, SNL and many HPC sites around the world

AN INTRODUCTION TO OPENSPEEDSHOP

OpenSpeedShop (OISS) is an open source multi-platform performance tool enabling performance analysis of HPC applications running on both single node and large scale Intel, AMD, ARM, Intel Phi, Power PC, Power, GPU processor based systems, including Cray and IBM platforms. OISS is a community effort by the Krell Institute with current direct funding from DOE NNSA. Argo Navis Technologies, LLC, a for-profit company working with Krell, recently completed a Phase II NASA SBIR to further develop OISS. OISS builds on top of a broad list of community infrastructures, most notably Dyninst and MRNet from UW, libmonitor from Rice, and PAPI from UTK.

OISS gathers and displays several types of information to aid in solving performance problems, including: a high-level summary of performance, program counter sampling for a lightweight flat profile to pinpoint where the slowdowns occurred, call path profiling to add caller/callee context and locate critical time consuming paths, access to the machine hardware counter information, input/output tracing for finding I/O performance problems, MPI function call tracing for MPI load imbalance detection, memory analysis, POSIX thread tracing, NVIDIA CUDA analysis, and OpenMP analysis. OISS offers a command-line interface (CLI), a graphical user interface (GUI) and a python scripting API user interface.

The base functionality includes:

- High-level Overview/Summary
- Program Counter Sampling
- Support for Call Path Analysis
- Hardware Performance Counters
- MPI Profiling and Event Tracing
- I/O Call Profiling and Tracing
- OpenMP Profiling and Analysis
- Memory Analysis
- POSIX Thread Analysis
- NVIDIA CUDA Tracing and Analysis

In addition, OISS is designed to be modular and extensible. It supports several levels of plug-ins which allow users to add their own performance experiments. The infrastructure and base components of OISS are released as open source code primarily under LGPL.

To run an OISS experiment on an application, simply choose the performance analysis information desired and run the application under OISS to gather the performance information. Once OISS has gathered the selected type of performance analysis information OISS then presents the performance information in a detailed report that allows the user to easily relate the performance information back to their application source code.

There are powerful algorithms included in OISS which can isolate ranks, processes, or threads that are performing outside the norm. These outlying performing entities are identified, so the user can examine outliers and resolve performance issues more rapidly.

AVAILABLE TODAY

- OpenSpeedShop Development Version on github
- OISS Installation and Support Contracts

KRELL institute

NEW RELEASE 2.4.0 AVAILABLE

OpenSpeedShop

Preferred build and install through spack package manager:
https://github.com/spack/spack

Downloads for legacy build and install:
www.openspeedshop.org/downloads

- Download source code and instructions on how to build and install

Experiments supported by O|SS

- High-level Overview/Summary
- Program Counter (PC) Sampling
- Exclusive and Inclusive User Time
- CPU Hardware Performance Counter
- MPI Profiling and Event Tracing
- I/O Call Profiling and Tracing
- OpenMP Profiling and Analysis
- Memory Analysis
- POSIX Thread Analysis
- NVIDIA CUDA Tracing

Analysis features supported by O|SS

- Cluster Analysis of MPI, OpenMP, and Threaded Applications
- Load Balance Views
- Comparisons of Multiple Runs, Ranks or Threads
- Hybrid (MPI and OpenMP) Analysis Support
- AVX512 Vector Instruction Detection and Display

www.openspeedshop.org

- Intel, AMD, ARM, Power, Cray and IBM platform support

FEATURES

- No need to recompile the user's application
- View performance information per library, per function, per loop, per statement, and per vector instruction
- Summary experiment creates comma separated list (csv) files containing application level overview performance information on MPI, OpenMP, I/O, Memory usage, kokkos, and hardware performance counters
- Four user interface options: batch (immediate), interactive command line interface (CLI), graphical user interface (GUI), and Python scripting API
- Supports multi-platform single system image (SSI), traditional clusters, Cray and IBM
- Scales to large numbers of processes, threads, and ranks
- View performance data using multiple customizable views
- Compare performance results
  - > Between processes, threads, or ranks
  - > Between a previous experiment and current experiment
- GUI with context sensitive help
- Option to automatically group like performing processes, threads, or ranks
- Support for installation via spack package manager
- Legacy comprehensive installation scripts
- Save and restore performance experiment data and symbol information for post experiment performance analysis
- Interactive CLI help facility lists the commands, syntax, and typical usage

UNDER DEVELOPMENT

- New Graphical User Interface
- Summary/Overview Experiment Enhancements

LOOK FOR US AT SC18

- Demonstrations in OpenISpeedShop Booth 2840