



Silicon Graphics, Inc.

OpenSpeedShop™

Status, Issues, Schedule, Demonstration

Presented by:

**Jim Galarowicz**

**SGL Core Engineering**

**March 21, 2006**



# Trademark Acknowledgements

- Intel, Intel Inside (logos) and Itanium are trademarks of Intel Corporation in the United States, other countries, or both.
- Linux is a trademark of Linus Torvalds in the United States, other countries or both.
- Qt and the Qt logo are trademarks of Trolltech in Norway, the United States and other countries.
- Open|SpeedShop, SGI SpeedShop, IRIX, SGI and SGI Altix are trademarks of Silicon Graphics Inc.
- IBM is a registered trademarks of International Business Machines Corporation in the United States, other countries, or both.
- All other trademarks mentioned herein are the property of their respective owners

# Open|SpeedShop Presentation Agenda

## Agenda:

- Project Introduction/Overview
- Feature Status
- Development Issues
- Schedule/Milestones
- Questions
- Demonstration (If we have time)
- Supplemental Info
  - Commonly Used Term Definitions
  - Credits

# Open|SpeedShop Project Introduction

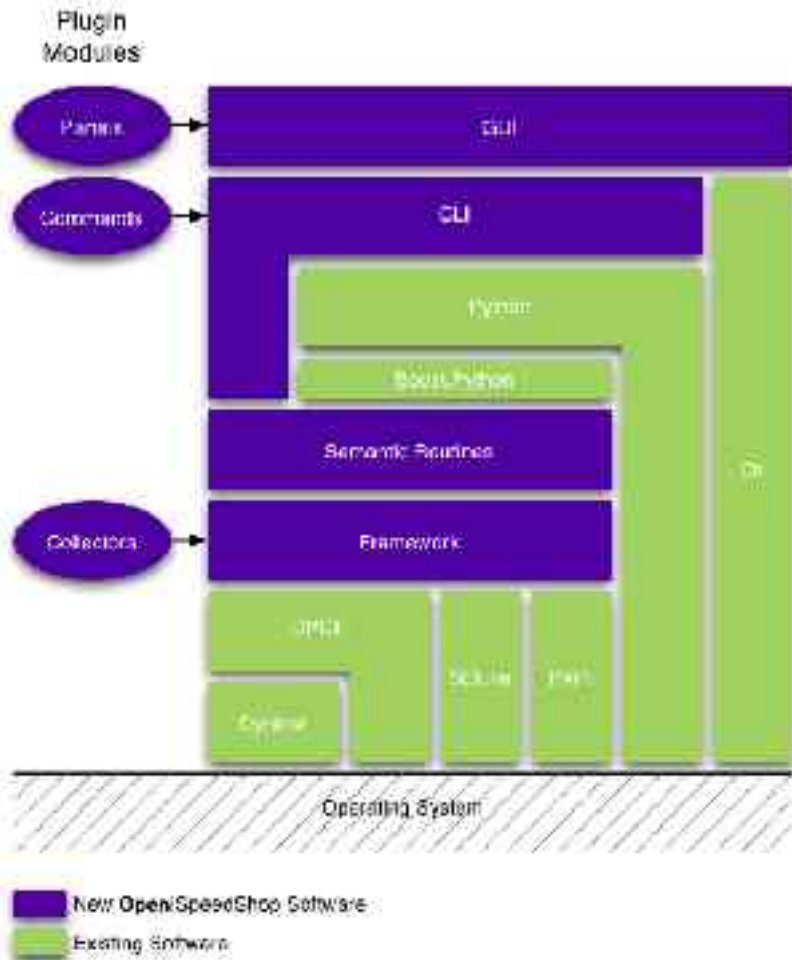
## Open|SpeedShop – an Open Source Performance Tool

- Based on software concepts from SGI® SpeedShop™ for the IRIX® O/S
  - Uses a set of experiments to help answer the question:
    - Where is my program (application) spending the most time and why?
  - Experiment Definition
    - A set of collectors and executables bound together to generate performance metrics.
  - Same type of experiments (sampling, event counting, call tracing)
  - No recompilation of application being analysed is needed
  - Minimal perturbation of the application being analysed
  - Support for batch performance analysis
  - Save performance measurement data file for later analysis

# Open|SpeedShop Project Introduction (continued)

- Open Source under GPL and LGPL licenses
- Uses Dynamic Instrumentation Technology
  - Dyninst: Universities of Wisconsin and Maryland
- Cluster Friendly
- Linux O/S Multi-Platform Support
  - Depends on availability of Dyninst
  - SGI target is SGI® Altix® system based on the Intel® Itanium™ chip
  - X86 and X86\_64 platforms also supported
- Implemented Using Open Source Components
  - **DPCL** – multi-platform cluster capability
  - **PAPI** – multi-platform hardware counter access
  - **SQLite** – lightweight multi-platform database
- DOE/NNSA co-funded - as part of ASC PathForward

# Open|SpeedShop Design Overview



- **Foundation of Open Source Components**

- Core Components (shown here)
- Build System (Autoconf, Automake, Libtool, Make & GCC)
- Source Control (CVS)
- Testing (Python)

- **Layered Approach**

- Open Source OS-Dependent Layer
- Non-Monolithic to Encourage Reuse
- Higher Levels of Abstraction Going Up

- **Extensible via Plugin Modules**

- Support New Data & Views
- Vendor Differentiation

# Open|SpeedShop Feature Status

- Three modes of operation: batch, interactive command line interface (CLI), and graphical user interface (GUI)
- Multiple platform single system image (SSI), traditional cluster, and SGI super cluster support
- Support for large numbers of processes, threads, and ranks\*
- Ability for users to create their own experiments via a well defined plug-in facility
- Attach to running processes, threads, and ranks to gather performance data\*
- Automatically attach to all processes in a MPI job to gather performance data
- View performance data for all of application's lifetime
- View performance data for the application in smaller time slices\*

\*Part or all of feature still in development and not available with Open|SpeedShop 0.875 release

# Open|SpeedShop Feature Status (Continued)

- View intermediate performance measurement data on demand (while the experiment is running)
- Save and restore performance experiment data and symbol information to do post experiment performance analysis
- Compare performance results for processes, threads, ranks, or to compare current experiment results with previous experiment results\*
- GUI Wizard facility and context sensitive help aid in the tools' usability
  - Multiple usability studies have been conducted
- Interactive CLI help facility which lists the CLI commands, syntax, and typical usage
- Python Scripting API which accesses Open|SpeedShop functionality corresponding to CLI commands

\*Part of all of feature is still in development and not available with Open|SpeedShop 0.875 release



# Open|SpeedShop Development Challenges

- *Development and Support for various Linux O/S variations*
  - Dyninst support required
  - Various lab environments/tools
  - Configuration issues/package differences/platform differences (lib64 versus lib)
- *Creating and Managing DPCL changes/fixes*
  - Getting these fixes approved and into DPCL sourceforge
  - 97 patches to DPCL so far
- *Using older versions of Dyninst and transitioning to newer versions*
  - Fixes for “older” versions of Dyninst
  - Trying to get newer versions of Dyninst to play well with DPCL
    - Time consuming – Dyninst changing rapidly
    - Takes time to verify new versions work with DPCL/openss

# Open|SpeedShop Development Challenges

- *Scalability Issues*
  - Threading internally in Open|SpeedShop
  - User Interface scaling issues – user friendly and fast
  - Performance Database file scaling issues
    - Make data retrieval and storage as fast as possible
  - Attaching to running processes in a timely manner
- *Supporting Various MPI Implementations*
  - SGI MPT, MPICH, LAMPI, OPENMPI
- *Development Time*
  - Schedules are tight

# Open|SpeedShop Milestones and Schedule

- *Infrastructure deliveries*
  - September and December, 2004
- *Performance Experiments Delivery Schedule*

|                                      |                              |
|--------------------------------------|------------------------------|
| – PC Sampling                        | 1 <sup>st</sup> Quarter 2005 |
| – User Time                          | 2 <sup>nd</sup> Quarter 2005 |
| – Hardware Counter                   | 3 <sup>rd</sup> Quarter 2005 |
| – MPI call tracing                   | 4 <sup>th</sup> Quarter 2005 |
| – MPI enhanced call tracing          | 1 <sup>th</sup> Quarter 2006 |
| – Input/Output call tracing          | 1 <sup>st</sup> Quarter 2006 |
| – Input/Output enhanced call tracing | 2 <sup>nd</sup> Quarter 2006 |
| – FPE tracing                        | 2 <sup>nd</sup> Quarter 2006 |
- *“Live” cvs repository source is available now:*
  - *<http://oss.sgi.com/openspeedshop>*

# Questions

## Questions:

Jim Galarowicz  
jeg@sgi.com

# Open|SpeedShop Demonstration

Open|SpeedShop  
Demonstration

## Supplemental Info

# Open|SpeedShop Terms and Definitions

**Experiment:** A set of collectors and executables bound together to generate performance metrics.

**Component(s):** A somewhat self-contained section of the Open|SpeedShop performance tool. This section of code does a set of specific related tasks for the tool. For example, the GUI component does all the tasks related to displaying Open|SpeedShop wizards, experiment creation, and results using a graphical user interface. The CLI component does similar functions but uses the interactive command line delivery method.

**Collector:** The portion of the tool containing logic that is responsible for the gathering of the performance metric (information). A collector is a portion of the code that is included in the experiment plugin.

**Metric:** The entity, which the collector/experiment is gathering. A time, occurrence counter, or other entity, which reflects in some way on the applications performance and is gathered by a performance experiment (by the collector).

**Framework:** The set of API functions that allows the user interface to manage the creation and viewing of performance experiments. It is the interface between the user interface and the cluster support and dynamic instrumentation components.

# Credits

- **Presentation**

- *William Hachfeld (wdh@sgi.com)*
- *Helen Stepanov (helens@sgi.com)*
- *Al Stipek (stipek@sgi.com)*
- *Jack Carter (jcarter@sgi.com)*

- **Partners**

- DOE/NNSA via ASC PathForward
  - LLNL, LANL, Sandia National Laboratories



# Credits Continued

## •Technology

- ***Dyninst***: jointly developed by the University of Wisconsin and the University of Maryland - <http://www.dyninst.org>
- ***DPCL*** : IBM® Open Source Project - <http://sourceforge.net/projects/dpcl>

# Credits Continued

- **Technology** (Continued)

- *Python: Open Source Project* - <http://www.python.org>
- **PAPI**: University of Tennessee - <http://icl.cs.utk.edu/papi>
- **SQLite**: Open Source Project - <http://www.sqlite.org>
- **Qt** ® : GUI toolkit - <http://www.trolltech.com/products/qt>