

OpenMP 4.0 Support In Oracle Solaris Studio

Ruud van der Pas
Distinguished Engineer
Architecture and Performance, SPARC Microelectronics

SC'14 OpenMP BOF
November 18, 2014

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

What Is Oracle Solaris Studio ?

- ***Supported on Linux too***
- Compilers and tools,
- It is actually a *comprehensive software suite*, including:
 - C, C++ and Fortran compilers
 - Various tools (performance analyzer, thread analyzer, code analyzer, debugger, etc)
- Platforms supported
 - Processors: SPARC, x86
 - Operating Systems: Solaris, Linux

Oracle Solaris Studio Features

www.oracle.com/technetwork/server-storage/solarisstudio/features/index.html

Oracle Solaris Studio Features

Sun Netra Carrier-Grade Systems

Networking and Data Center Fabric Products

Sun Storage Software

Sun Ray Products


Sun x86 Servers

Virtual Desktop Infrastructure


OpenStack

ORACLE SOLARIS STUDIO


COMPILER SUITE




C, C++ Compilers
Utilizes advanced code generation technology to optimize apps for highest performance on Oracle's Sun servers (SPARC & x86).



Fortran Compiler
Optimize compute-intensive application performance.




Debugger
Ensures app stability with event handling and multi-thread support.




Performance Library
Maximizes compute-intensive app performance across SPARC and x86 using advanced numeric solver libraries.


ANALYSIS SUITE



Performance Analyzer
Provides unparalleled insight into your app, allowing you to identify bottlenecks and improve performance by orders of magnitude.



Code Analyzer
Ensures app reliability by detecting app vulnerabilities, including memory leaks and memory access violations.



Thread Analyzer
Simplifies complex parallel programming errors by detecting hard to pinpoint race and deadlock conditions.

INTEGRATED DEVELOPMENT ENVIRONMENT Increases Developer Efficiency

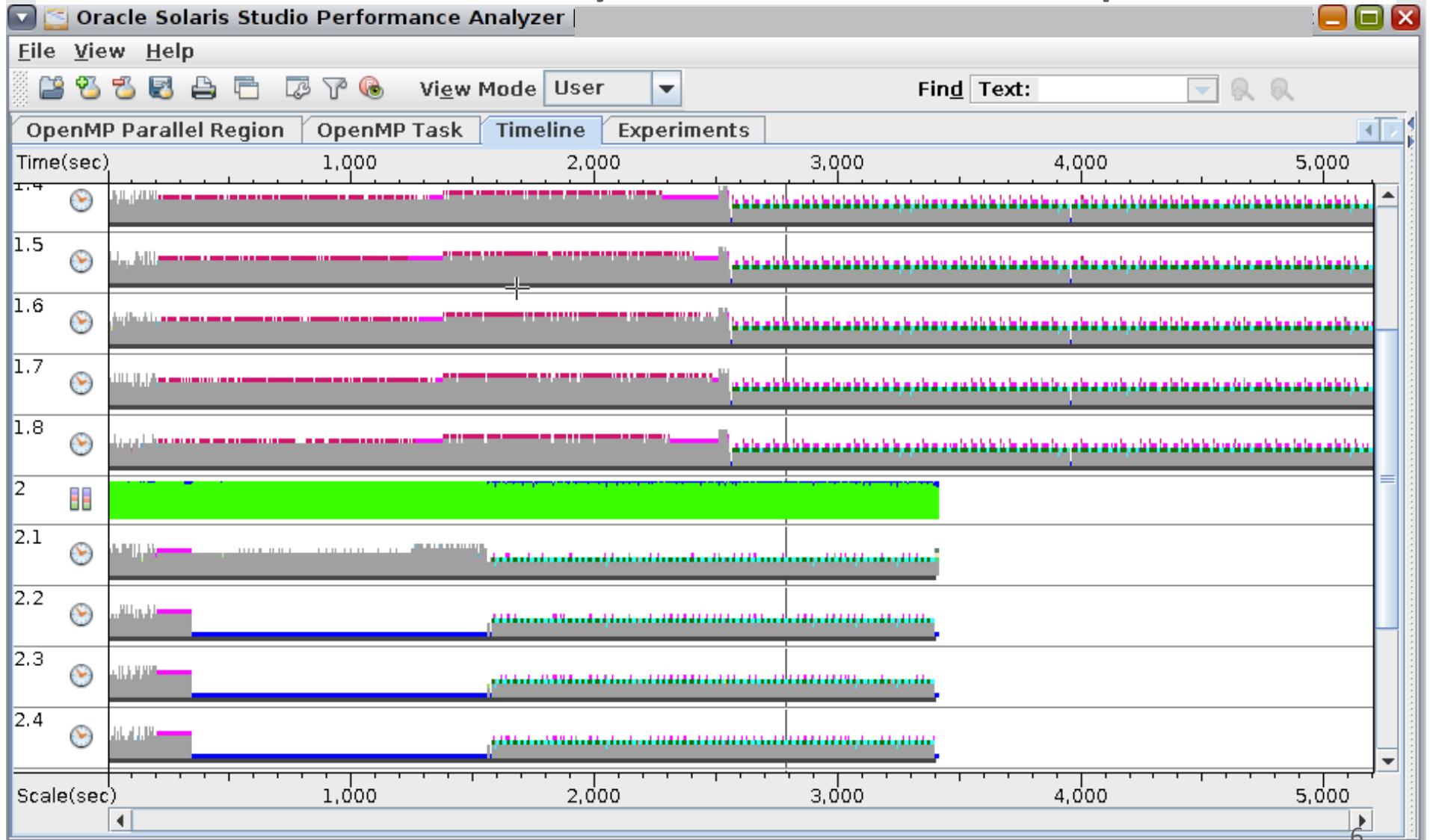
<http://www.oracle.com/technetwork/server-storage/solarisstudio/overview/index.html>

- **Performance Library:** Maximizing application performance, the Performance Library provides a set of optimized, high-speed mathematical subroutines for solving linear algebra and other numerically intensive problems. It contains enhanced and newly added standard routines such as BLAS1/2/3, CBLAS, LAPACK, Sparse BLAS, SPSOLVE, SuperLU, FFTPACK, and VFFTPACK.

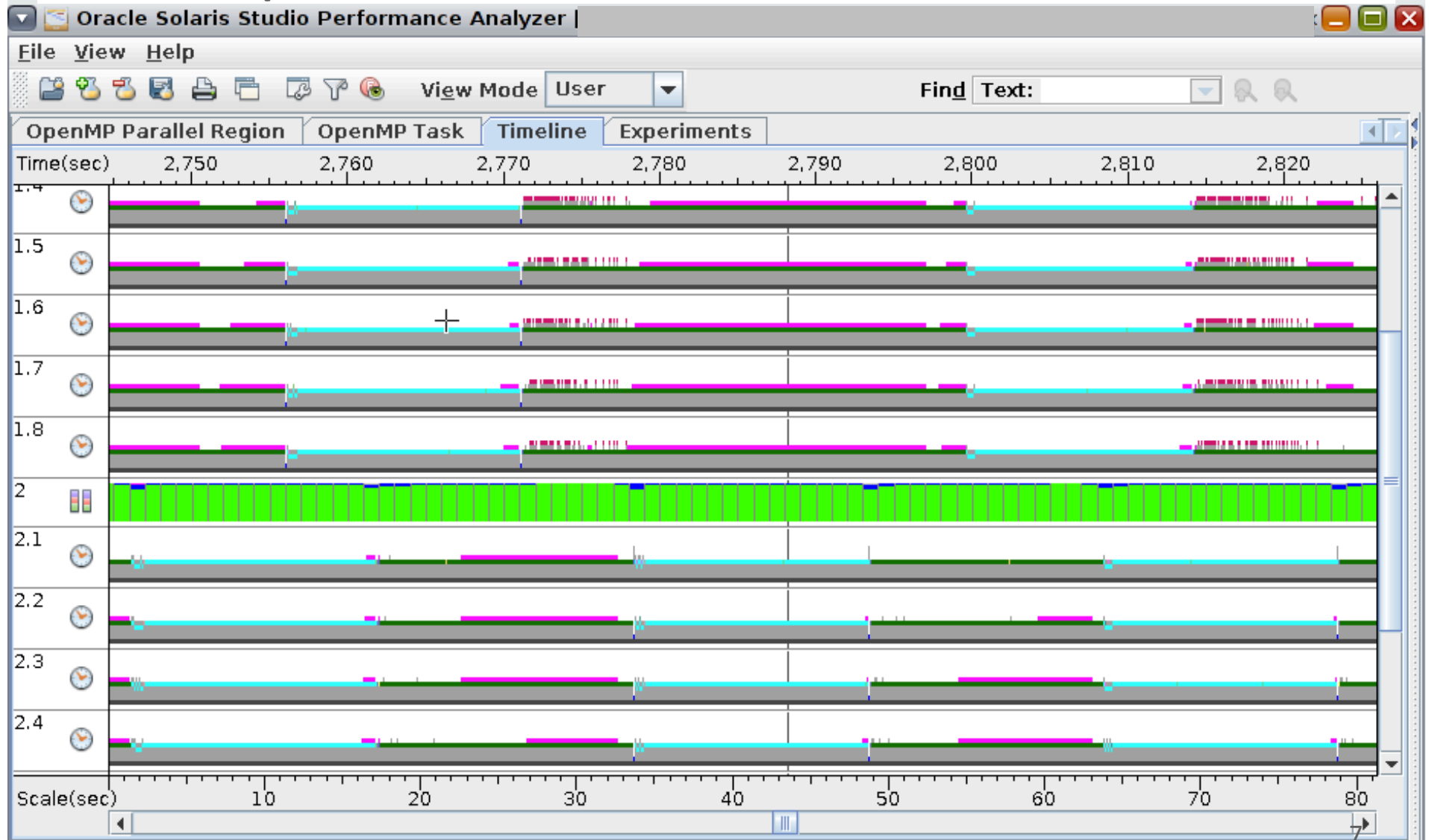
OpenMP Specific Support/1

- Compiler
 - Various options and environment variables
 - Autoscopying
 - Compiler Commentary
 - General feature to inform about optimizations, but also specific to OpenMP
- Performance Analyzer
 - OpenMP “states”, metrics, etc
 - Task and region specific information

Performance Analyzer - Profile Comparison



A Comparison– More Detail



OpenMP Specific Support/2

- Thread Analyzer
 - Data race and deadlock detection
- Performance Library
 - Uses OpenMP internally
- Debugger

Oracle Solaris Thread Analyzer

The screenshot displays the Oracle Solaris Thread Analyzer interface. The left sidebar contains a 'Views' panel with options: Welcome, Overview, Races (selected), Timeline, Dual Source, Experiments, and More Views... Below this is a 'No Active Filters' section with instructions on how to add filters. The main window is divided into three panes. The top-left pane lists three race conditions (Race #1, #2, #3) with their Vaddr and access details. The top-right pane, titled 'Data for Selected Race', shows details for 'Race 1/3 -- Callstack Trace 1/1', including the Vaddr and the type of access (Write). The bottom-right pane shows the source code for the selected race, highlighting the lines where the data race occurs. Two large red arrows point from the 'Races' list to the 'Data for Selected Race' pane and the source code pane. A red text box at the bottom states: 'Shows source lines where data races occur'.

Races List:

- Race #1, Vaddr :0xffbfff648
 - Access 1: Write, line 23 in "matrix_multiply_race.c", main
 - Access 2: Write, line 23 in "matrix_multiply_race.c", main
 - Total Callstack Traces: 1
 - Callstack Trace #1
- Race #2, Vaddr :0xffbfff31c
 - Access 1: Write, line 25 in "matrix_multiply_race.c", main
 - Access 2: Write, line 25 in "matrix_multiply_race.c", main
 - Total Callstack Traces: 1
- Race #3, Vaddr :0xffbfff318
 - Access 1: Write, line 26 in "matrix_multiply_race.c", main
 - Access 2: Write, line 26 in "matrix_multiply_race.c", main
 - Total Callstack Traces: 1

Data for Selected Race (Race 1/3 -- Callstack Trace 1/1):

Vaddr: 0xffbfff648

Access 1

Type: Write

main, line 23 in "matrix_multiply_race.c"
main, line 20 in "matrix_multiply_race.c"
_start + 0x00000108

Access 2

Type: Write

main, line 23 in "matrix_multiply_race.c"
main, line 20 in "matrix_multiply_race.c"
_start + 0x00000108

Shows source lines where data races occur

Local Host: clpt41 | Remote Host: | Working Directory: ../Race | Compare: off | Filters: off

Oracle Solaris Thread Analyzer

The screenshot displays the Oracle Solaris Thread Analyzer interface. The left sidebar contains a 'Views' menu with options: Welcome, Overview, Races, Timeline, Dual Source (selected), Experiments, and More Views... Below this is a 'No Active Filters' section with a filter icon and instructions. The main window is divided into three panes. The top pane shows a list of 'Accesses' with a table of memory access details. The middle pane displays the source code for 'matrix_multiply_race.c', with a red arrow pointing to line 23, which is highlighted. The bottom pane shows the 'Data for Selected Race' details, including the race ID, address, and access type (Write). A large orange box with the text 'Show dual source code' is overlaid on the right side of the interface, with a red arrow pointing to the source code pane.

Accesses

Accesses	Source File: matrix_multiply_race.c
0	16. B[i][j] = 20;
0	17. }
0	18. }
0	19. }
30	20. #pragma omp parallel for shared(A,B,C) num_threads(4)
0	<Function: main -- Parallel loop from line 20 [\$_d1A20.main]>
0	21. for (i=0; i<n; i++)
0	22. {
10	23. for (j=0; j<n; j++)
0	24. {
10	25. sum = 0;
10	26. for (k=0; k<n; k++)
0	27. {
0	28. sum = sum + (A[i][k] * B[k][j]);
0	29. C[i][j] = sum;
0	30. }
0	31. }
0	32. }

Data for Selected Race

Id: Race 1/3 -- Callstack Trace 1/1
Vaddr: 0xffbfff648

Access 1

Type: Write
main, line 23 in "matrix_multiply_race.c"
main, line 20 in "matrix_multiply_race.c"
_start + 0x00000108

Access 2

Type: Write
main, line 23 in "matrix_multiply_race.c"
main, line 20 in "matrix_multiply_race.c"
_start + 0x00000108

No Active Filters

To add a filter, select a row from a view (such as Functions) and then click on the toolbar Filters icon.

Local Host: clpt41 | Remote Host: | Working Directory: .../Race | Compare: off | Filters: off

Some Features Useful For OpenMP Apps

- Stack overflow check (**-xcheck=stkovf** option)
- Static, compile time, checks (**-xvpara** option)
- Run time checks (**SUNW_MP_WARN** variable)
- Run time controls (e.g. behaviour of idle threads)
- Autoscopying (“**default(__auto)**” clause)
- Run time data race and deadlock detection
- Debugger/dbx support for OpenMP

Just Released (November 11, 2014)

Oracle Solaris Studio 12.4

Full Support For OpenMP 4.0

Get Your Free Download Now

New Features in Studio 12.4

- **Full support for OpenMP 4.0**
- C++ 2011 support
- Significant improvements in:
 - Performance Analyzer (both functionality and GUI)
 - Code Analyzer
 - Integrated Development Environment

***Check the web site for more details
and your free download***

OpenMP 4.0 Support

Full OpenMP 4.0 Support/1

- Thread Affinity
- Tasking Extensions
- Error Handling
- User Defined Reductions
- ...

Full OpenMP 4.0 Support/2

- SIMD
 - Constructs are accepted, but may not generate SIMD instructions
- Accelerators
 - Constructs are accepted, but code is executed on host

Thank You And Stay Tuned !

Ruud van der Pas

ruud.vanderpas@oracle.com

**[http://www.oracle.com/technetwork/
server-storage/solarisstudio/overview/index.html](http://www.oracle.com/technetwork/server-storage/solarisstudio/overview/index.html)**

Hardware and Software Engineered to Work Together

ORACLE®