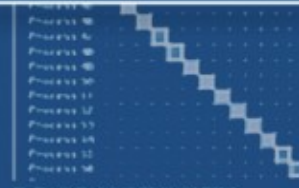




SOFTWARE

- + 19.56 updatex
- + 399.70 updateien
- + 0.00 gene
- 0.00 <<iteration loop>>
- + 447.52 genbc

PRODUCTIVITY



FAST SOLUTIONS

- PAPI_L1_ICM
- PAPI_L2_DCM
- PAPI_L2_ICM
- PAPI_L1_TCM

Marmot MPI Correctness Tool: Overview

For the 4th VI-HPS Tuning Workshop

Tobias Hilbrich

GWT-TUD GmbH

tobias.hilbrich@zih.tu-dresden.de

September 2009

- Tool to check for correct MPI usage at runtime
 - Checks parameters passed to MPI
 - Checks conformance to MPI Standard
 - Monitors MPI resource usage
- A library written in C++, is linked to the application
- Does not require source code modifications
- Requires an additional process used as *DebugServer*
- Support for Fortran and C bindings of MPI-1.2
- Results are shown in a log file (different types supported)

Linking and Compilation with Marmot

- Use the Marmot compiler wrappers to compile and link:
 - Replace compiler calls by appropriate wrapper
 - For C/C++ use `marmotcc` or `marmotcxx`
 - For Fortran use `marmotf77` or `marmotf90`
 - Source code instrumentation added automatically (usually)

Execution with Marmot

- Execution with Marmot requires one additional process (used for global view)
 - Instead of “`mpirun -np n`” call “`mpirun -np n+1`”
 - Marmot's checks cause overhead, thus, performance may decrease

- MARMOT supports 3 different log file types:
 - ASCII
 - HTML
 - CUBE
- Desired log file type is selected with an environment variable (`export MARMOT_LOGFILE_TYPE=...`)
- ASCII and HTML logs may become very large and are unable to give an overview
- CUBE log offers an overview along with all the detailed information

datatype.c:

```
(47) MPI_Init (&argc, &argv);
(48) MPI_Comm_rank (MPI_COMM_WORLD, &rank);
(49) MPI_Comm_size (MPI_COMM_WORLD, &size);
(50) MPI_Type_contiguous (2, MPI_INT, &cont2Int);
(51)
(52) if (rank == 0)
(53)     MPI_Send (s_buf, 1, cont2Int, 1, MSG_TAG_1, MPI_COMM_WORLD);
(54)
(55) if (rank == 1)
(56)     MPI_Recv (r_buf, 1, cont2Int, 0, MSG_TAG_1, MPI_COMM_WORLD, &status);
(57)
(58) MPI_Type_free (&cont2Int);
(59) MPI_Finalize();
```

Linking and Compiling

- `marmotcc datatype.c -o datatype.exe` *#Compile and link*
- (On some systems you may have to load a module first)

Configuring Marmot (optional)

- `export MARMOT_LOGFILE_TYPE=1` *#Use HTML logging*

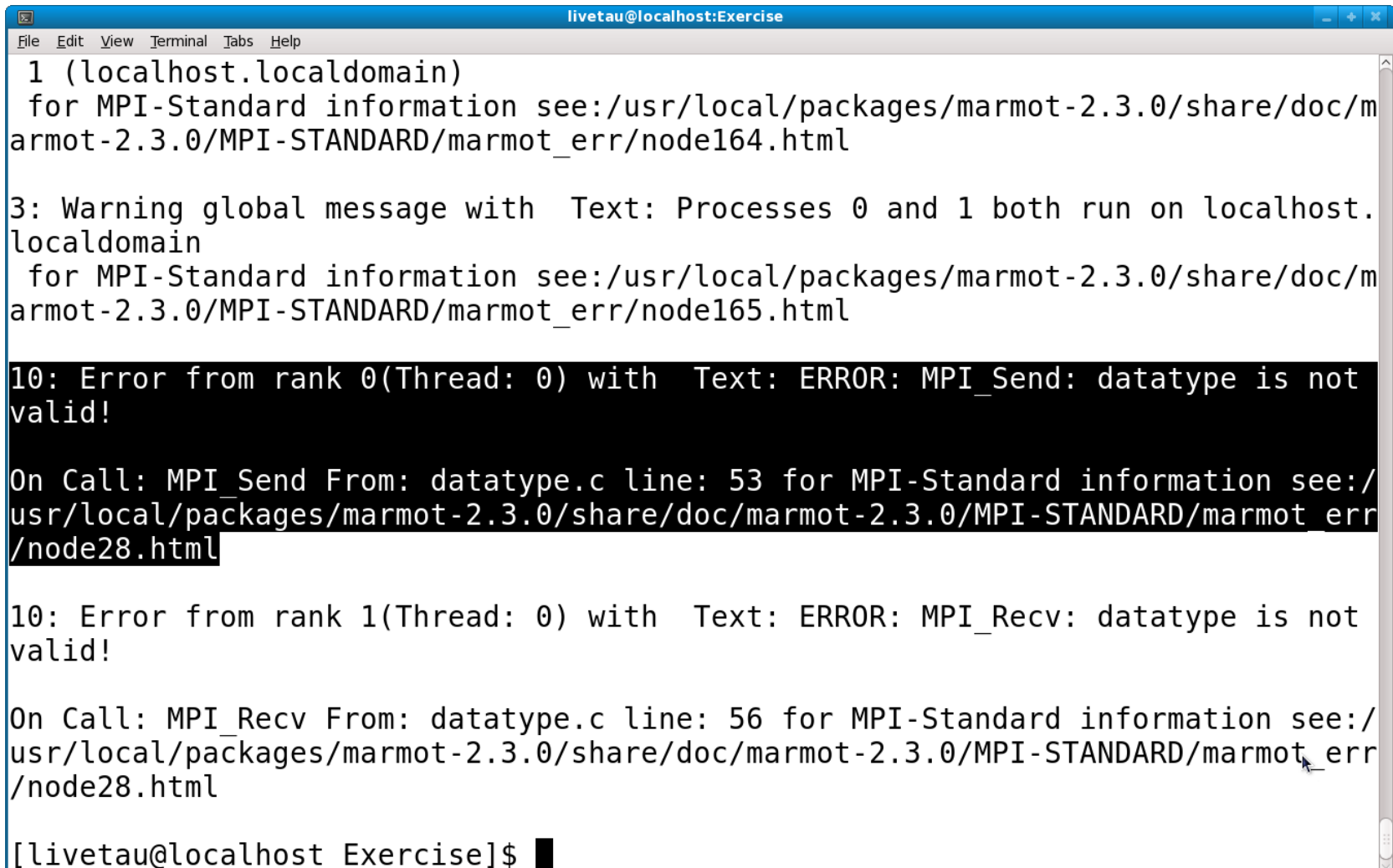
Running

- `mpirun -np 3 datatype.exe` *#Execute with 2(!) app. processes*

Example – Text Output



```
export MARMOT_LOGFILE_TYPE=0
```

A terminal window titled "livetau@localhost:Exercise" showing the output of an MPI program. The output includes a warning about processes running on localhost, followed by two error messages from ranks 0 and 1. The error messages indicate that the MPI_Send and MPI_Recv functions received an invalid datatype. The terminal prompt is "[livetau@localhost Exercise]\$".

```
1 (localhost.localdomain)
for MPI-Standard information see:/usr/local/packages/marmot-2.3.0/share/doc/marmot-2.3.0/MPI-STANDARD/marmot_err/node164.html

3: Warning global message with Text: Processes 0 and 1 both run on localhost.localdomain
for MPI-Standard information see:/usr/local/packages/marmot-2.3.0/share/doc/marmot-2.3.0/MPI-STANDARD/marmot_err/node165.html

10: Error from rank 0(Thread: 0) with Text: ERROR: MPI_Send: datatype is not valid!

On Call: MPI_Send From: datatype.c line: 53 for MPI-Standard information see:/usr/local/packages/marmot-2.3.0/share/doc/marmot-2.3.0/MPI-STANDARD/marmot_err/node28.html

10: Error from rank 1(Thread: 0) with Text: ERROR: MPI_Recv: datatype is not valid!

On Call: MPI_Recv From: datatype.c line: 56 for MPI-Standard information see:/usr/local/packages/marmot-2.3.0/share/doc/marmot-2.3.0/MPI-STANDARD/marmot_err/node28.html

[livetau@localhost Exercise]$
```

Example – HTML Output



export MARMOT_LOGFILE_TYPE=1

MARMOT HTML Logfile - Konqueror						
/home/livetau/workshop-marmot/Exercise/Marmot_datatype.exe_20090807_130509.html						
				default: 1000 microseconds)		
0	Global	0	Information	Text: MARMOT_MAX_TIMEOUT_ONE = 0 (maximum message time, default: 0 microseconds)	Unknown	
0	Global	0	Information	Text: MARMOT_MAX_TIMEOUT_TWO = 0 (maximum message time, default: 0 microseconds)	Unknown	
0	Global	0	Information	Text: MARMOT_LOGFILE_PATH = (path of Marmot log file output, default:)	Unknown	
0	Global	0	Information	Text: MARMOT_ERRCODES_SET = (not set) (not functional yet)	Unknown	
0	Global	0	Information	Text: End of the environmental variables info.	Unknown	
0	Global	0	Information	Text: Thread Synchronisation is disabled.If you are using multiple threads errors might occur	Unknown	
3	Global	0	Warning	Text: Debugserver runs on same node as process 0 (localhost.localdomain)	Unknown	Infos see MPI-Standard
3	Global	0	Warning	Text: Debugserver runs on same node as process 1 (localhost.localdomain)	Unknown	Infos see MPI-Standard
3	Global	0	Warning	Text: Processes 0 and 1 both run on localhost.localdomain	Unknown	Infos see MPI-Standard
10	0	0	Error	Text: ERROR: MPI_Send: datatype is not valid! Call: MPI_Send	datatype.c line: 53	Infos see MPI-Standard
10	1	0	Error	Text: ERROR: MPI_Recv: datatype is not valid! Call: MPI_Recv	datatype.c line: 56	Infos see MPI-Standard

Example – CUBE Output



export MARMOT_LOGFILE_TYPE=2

The screenshot shows the Cube 3.2 QT interface with three main panels:

- Metric tree:** Shows a hierarchy of metrics. The '1 ERROR - Datatype is not valid!' entry is highlighted in blue.
- Call tree:** Shows a call stack. The entry '1 MPI_Recv @line: 56' is highlighted in blue.
- System tree:** Shows the MPI environment structure. The entry '1 rank 1' is highlighted in blue.

At the bottom, a status bar indicates 'Selected *1 MPI_Recv @line: 56*'. A color bar at the very bottom shows a gradient from blue to red.

Right click for detailed information (as HTMLs)

What

Where in Code

Which process

- Call site
- Called region
 - Location
 - Source code
 - Online Description
 - Info