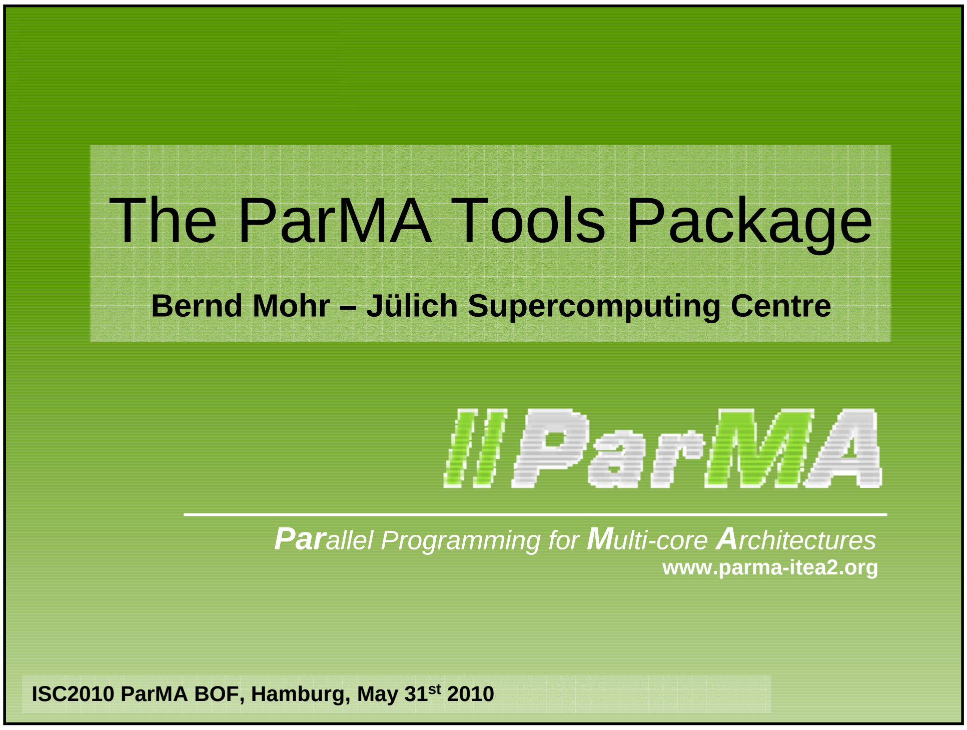


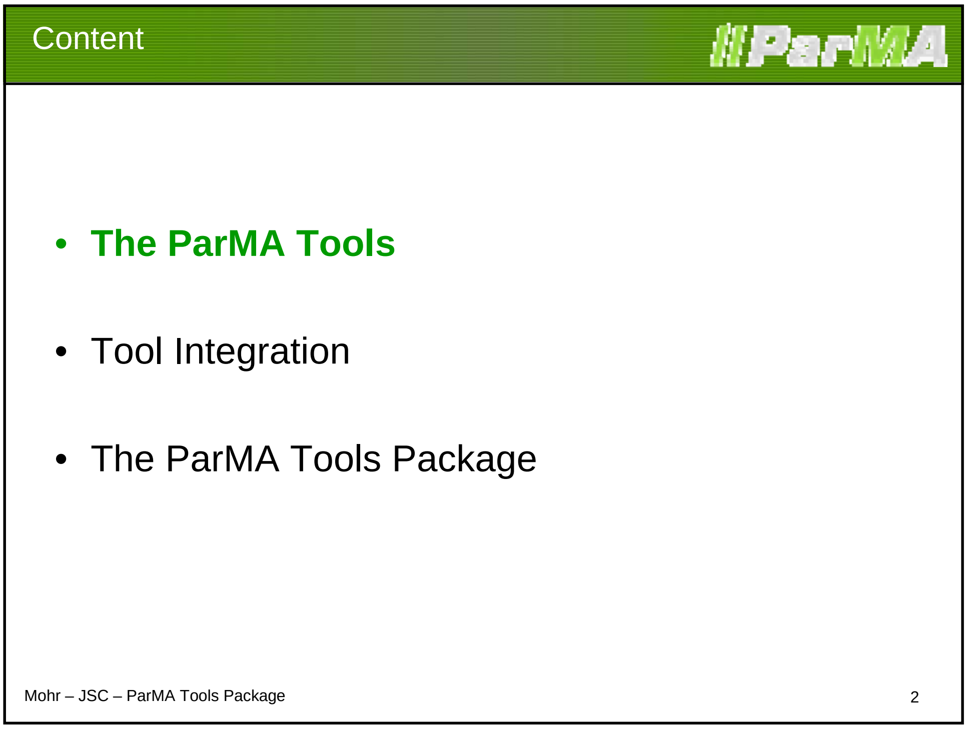
The ParMA Tools Package

Bernd Mohr – Jülich Supercomputing Centre

***Par**allel Programming for **M**ulti-core **A**rchitectures*

www.parma-itea2.org

ISC2010 ParMA BOF, Hamburg, May 31st 2010



Content

- **The ParMA Tools**
- Tool Integration
- The ParMA Tools Package

Goals

- **Starting point of the ParMA project (2007)**
  - Established MPI debugging and performance analysis tools of German and UK partners
  - New innovative tools from French partners
- **Achieved results (2010)**
  - **Adaptation to + enhancements for multi-core systems**
    - Support for multi-threaded programming (e.g., OpenMP and POSIX threads)
  - General enhancements (e.g., MPI-2 and I/O support)
  - **Integration of tools to coherent environment**
  - Based on feedback from real-world, industrial applications

Established Tools

Universität Stuttgart

- **Marmot** MPI correctness checker
- Open MPI / Peruse introspection interface

Technische Universität Dresden / GWT

- **Marmot** MPI correctness checker
- **VampirTrace** trace measurement system
- **Vampir** / **VampirServer** event trace visualizer

Jülich Supercomputing Centre

- **KOJAK** / **Scalasca** automatic trace-based performance analysis
- **CUBE** visualization browser

Allinea (unfunded partner)

- **DDT** parallel debugger

New Innovative Tools

UVSQ (France)

- **MAQAO** code and memory analysis on x86_64 architecture
- **DECAN** decremental analysis of machine code instructions
- **MTL** memory tracing library
- **MADRAS** binary instrumentation

It Sudparis (France)

- **STEP** can automatically transform OpenMP into MPI (and hybrid MPI/OpenMP) programs
- Allows to move shared-memory applications (limited by the power of one node) on to a cluster

DDT Parallel Debugger

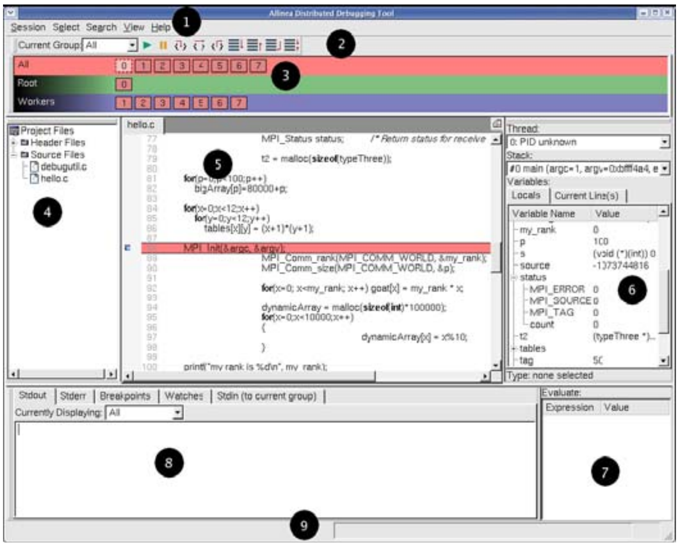
- **DDT**
  - Graphical parallel debugger

SCALE TO NEW HEIGHTS

- Commercial product of Allinea
- Portable across many platforms
- Support for parallel programs
  - Multi-tasking (MPI)
  - Multi-threading (OpenMP, pthreads)
  - GPUs
  - At scale

Example: DDT Main Screen

❶ Menu bar
❷ Process controls
❸ Process group window
❹ File window
❺ Code window
❻ Variable window
❼ Evaluate window
❽ Output window
❾ Status bar



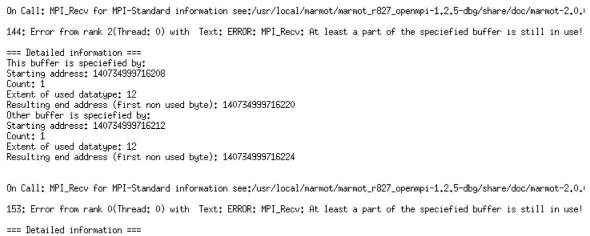
```
                MPI_Status status;       /* Return status for receive

                t2 = malloc(sizeof(typeThree));

    for(p=0;p<100;p++)
        bigArray[p]=80000+p;

    for(x=0;x<12;x++)
        for(y=0;y<12;y++)
            tables[x][y] = (x+1)*(y+1);

    MPI_Init(&argc, &argv);
                MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
                MPI_Comm_size(MPI_COMM_WORLD, &p);

                for(x=0; x<my_rank; x++) goat[x] = my_rank * x;

                dynamicArray = malloc(sizeof(int)*100000);
                for(x=0;x<10000;x++)
                {
                                dynamicArray[x] = x%10;
                }

    printf("my rank is %d\n", my_rank);
```

Variable Name	Value
my_rank	0
p	100
s	(void (*)(int)) 0
source	-1073744816
status	
MPI_ERROR	0
MPI_SOURCE	0
MPI_TAG	0
count	0
t2	(typeThree *)...
tables	
tag	50

MARMOT MPI Correctness Checker

- **MARMOT** checks
  - Correct usage of MPI calls
  - Portability
  - MPI resource usage
- Open source
- Developed in cooperation of ZIH and HLRS
- Supports
  - C, C++ and Fortran
  - MPI, or hybrid OpenMP/MPI

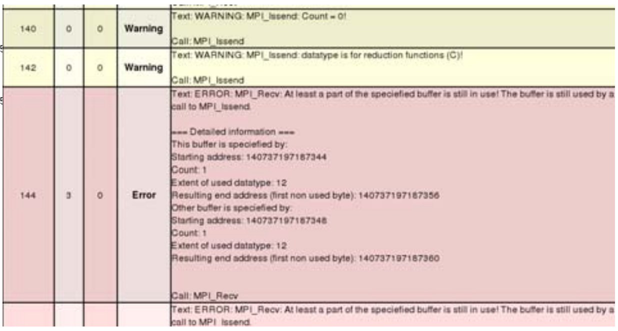
Example: MARMOT Analysis



```
On Call: MPI_Recv for MPI-Standard information see:/usr/local/marmot/marmot_r827_openmpi-1.2.5-dbg/share/doc/marmot-2.0.

144: Error from rank 2(Thread: 0) with  Text: ERROR: MPI_Recv: At least a part of the speciefied buffer is still in use!

=== Detailed information ===
This buffer is speciefied by:
Starting address: 140734999716208
Count: 1
Extent of used datatype: 12
Resulting end address (first non used byte): 140734999716220
Other buffer is speciefied by:
Starting address: 140734999716212
Count: 1
Extent of used datatype: 12
Resulting end address (first non used byte): 140734999716224


On Call: MPI_Recv for MPI-Standard information see:/usr/local/marmot/marmot_r827_openmpi-1.2.5-dbg/share/doc/marmot-2.0.

153: Error from rank 0(Thread: 0) with  Text: ERROR: MPI_Recv: At least a part of the speciefied buffer is still in use!

=== Detailed information ===
This buffer is speciefied by:
Starting address: 140737469793568
Count: 1
Extent of used datatype: 12
Resulting end address (first non used byte): 1407374697935
Other buffer is speciefied by:
Starting address: 140737469793572
Count: 1
Extent of used datatype: 12
Resulting end address (first non used byte): 1407374697935
```

- Plain text

- HTML



140	0	0	**Warning**	Text: WARNING: MPI_Issend: Count = 0! Call: MPI_Issend
142	0	0	**Warning**	Text: WARNING: MPI_Issend: datatype is for reduction functions (C)! Call: MPI_Issend
144	3	0	**Error**	Text: ERROR: MPI_Recv: At least a part of the speciefied buffer is still in use! The buffer is still used by a call to MPI_Issend. === Detailed information === This buffer is speciefied by: Starting address: 140737197187344 Count: 1 Extent of used datatype: 12 Resulting end address (first non used byte): 140737197187356 Other buffer is speciefied by: Starting address: 140737197187348 Count: 1 Extent of used datatype: 12 Resulting end address (first non used byte): 140737197187360 Call: MPI_Recv
				Text: ERROR: MPI_Recv: At least a part of the speciefied buffer is still in use! The buffer is still used by a call to MPI_Issend.

Vampir Tool Environment

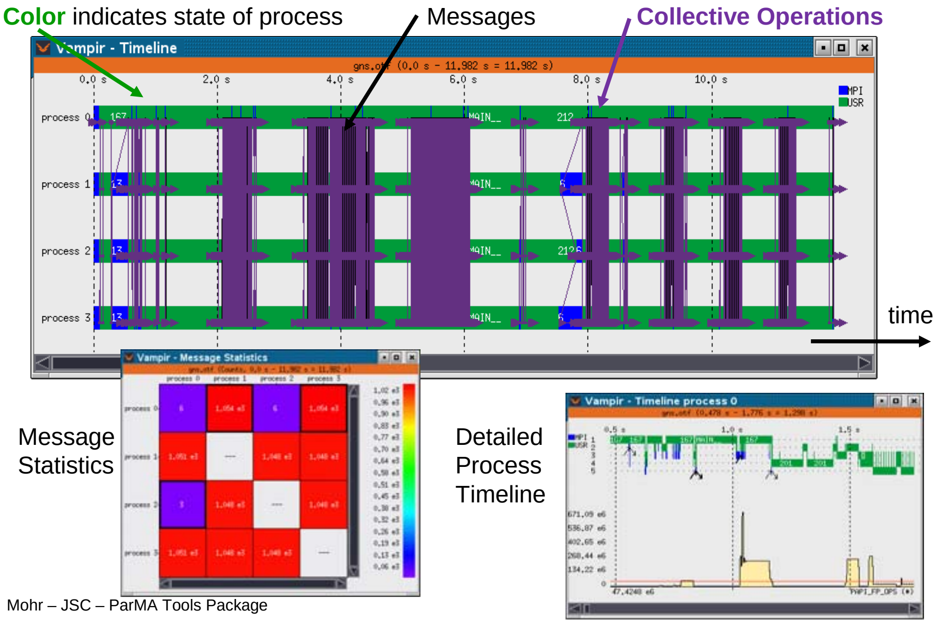
- **VampirTrace**
  - Open source
  - Event trace measurement system
  - Instruments C, C++, and Fortran
    - MPI, (simple) OpenMP, or hybrid, I/O
  - Collects event traces in OTF format
- Post-mortem visual analysis
  - Developed originally by Jülich and since 1997 by ZHR/ZIH of TU Dresden
  - Commercial distribution by GWT-TUD
  - **Vampir**: sequential visualizer
  - **VampirServer**: distributed client / parallel server architecture

VAMPIR

ZIH
Center for Information Services & High Performance Computing

GWT forschung+innovation

Example: Vampir Display of GNS code



Color indicates state of process

Messages

Collective Operations

time

Message Statistics

Detailed Process Timeline

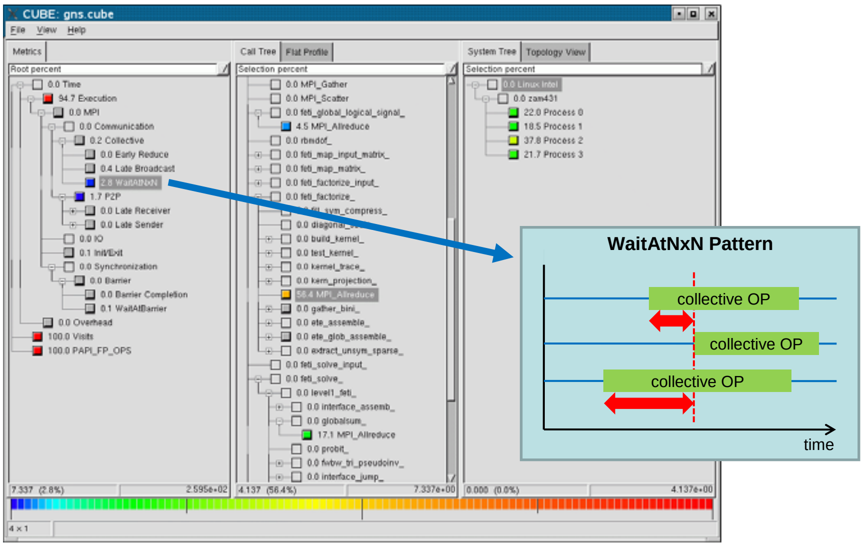
KOJAK / Scalasca Toolset

- Open source
- Jülich Supercomputing Centre

- Automatic performance analysis
  - Instrument C, C++, and Fortran parallel applications
    - Based on MPI, (simple) OpenMP, or hybrid
  - Collect call-path-profiles and EPILOG event traces
  - Scan trace for event patterns representing inefficiencies
    - **KOJAK**: sequential analysis
    - **Scalasca**: parallel analysis
  - Categorize and rank inefficiencies found
  - Visualize via **CUBE** browser

Example: KOJAK Analysis of GNS code

Identified Problems **Where in source?** **Which processes are affected?**



```
Root percent
0.0 Time
  94.7 Execution
    0.0 MPI
      0.0 Communication
        0.2 Collective
          0.0 Early Reduce
          0.4 Late Broadcast
          2.8 WaitAtNxN
        1.7 P2P
          0.0 Late Receiver
          0.0 Late Sender
      0.0 IO
      0.1 Init/Exit
      0.0 Synchronization
        0.0 Barrier
          0.0 Barrier Completion
          0.1 WaitAtBarrier
  0.0 Overhead
100.0 Visits
100.0 PAPI_FP_OPS
7.337 (2.8%)    2.595e+02
```

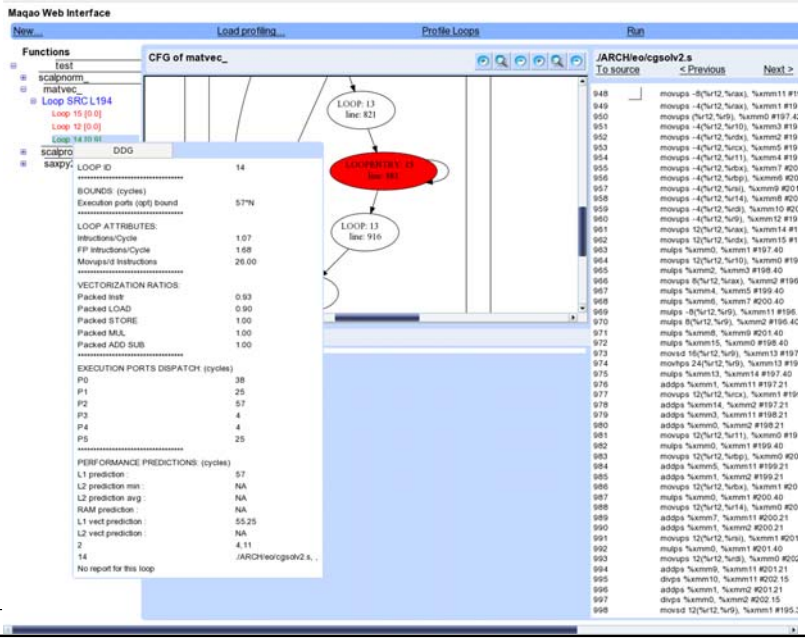
```
Selection percent
0.0 MPI_Gather
0.0 MPI_Scatter
0.0 feti_global_logical_signal_
  4.5 MPI_Allreduce
0.0 rbmdof_
0.0 feti_map_input_matrix_
0.0 feti_map_matrix_
0.0 feti_factorize_input_
0.0 feti_factorize_
  0.0 fil_sym_compress_
  0.0 diagonal_
  0.0 build_kernel_
  0.0 test_kernel_
  0.0 kernel_trace_
  0.0 kern_projection_
  58.4 MPI_Allreduce
  0.0 gather_bini_
  0.0 ete_assemble_
  0.0 ete_glob_assemble_
  0.0 extract_unsym_sparse_
0.0 feti_solve_input_
0.0 feti_solve_
  0.0 level1_feti_
    0.0 interface_assemb_
    0.0 globalsum_
      17.1 MPI_Allreduce
    0.0 probit_
    0.0 fwbw_tri_pseudoinv_
    0.0 interface_jump_
4.137 (56.4%)    7.337e+00
```

```
Selection percent
0.0 Linux Intel
  0.0 zam431
    22.0 Process 0
    18.5 Process 1
    37.8 Process 2
    21.7 Process 3
0.000 (0.0%)    4.137e+00
4 x 1
```

MAQAO Toolset

- **MAQAO (Modular Assembly Quality Analyzer and Optimizer)**
  - Combines static analysis of assembly code with dynamic analysis of execution traces
  - Focus on loop-level
- **MADRAS (Multi Architecture Disassembler ReAssembler)**
  - Instrumentation of binary files for MAQAO
- **MTL (MAQAO Trace Library)**
  - Memory trace library integrated in MAQAO, for threaded codes
  - Detect important inefficiencies (e.g., false sharing, strided access) or opportunities for optimizations (setting thread affinity according to reuse among threads)
- **DECAN (Decremental Analysis)**
  - Automatic detection of performance anomalies (e.g., inefficient memory access) via iterative modification of machine code instructions of hot functions
- Université de Versailles St-Quentin-en-Yvelines

Example: MAQAO Toolset

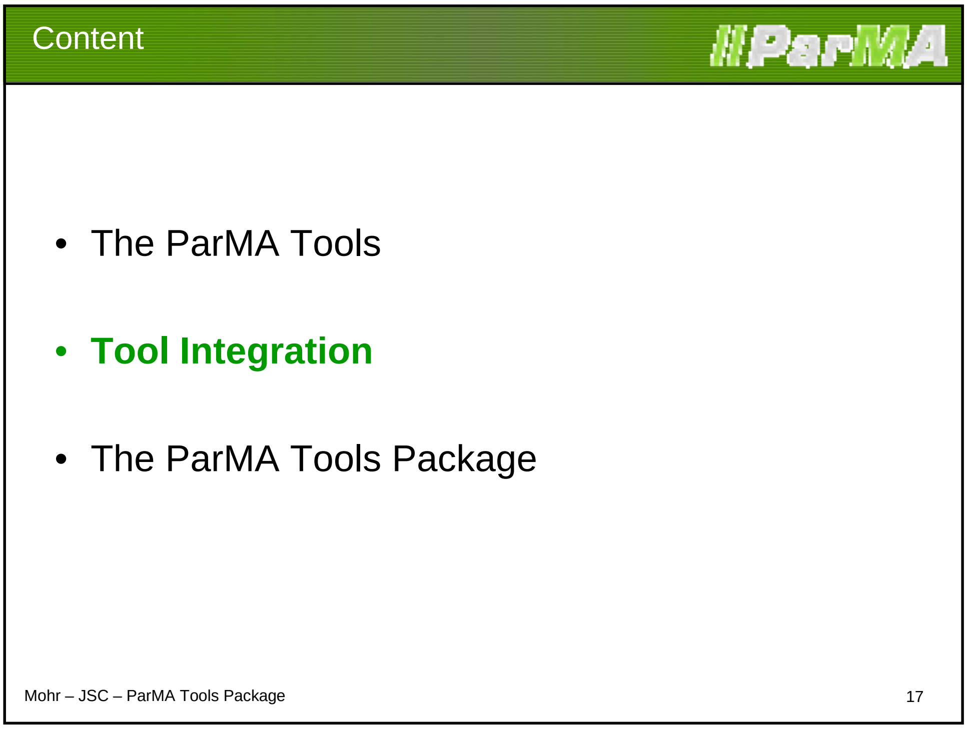


```
LOOP ID                                14
**********************************
BOUNDS: (cycles)
Execution ports (opt) bound            57*N
**********************************
LOOP ATTRIBUTES:
Intructions/Cycle                      1.07
FP Intructions/Cycle                   1.68
Movups/d Instructions                  26.00
**********************************
VECTORIZATION RATIOS:
Packed Instr                           0.93
Packed LOAD                            0.90
Packed STORE                           1.00
Packed MUL                             1.00
Packed ADD SUB                         1.00
**********************************
EXECUTION PORTS DISPATCH: (cycles)
P0                                     38
P1                                     25
P2                                     57
P3                                     4
P4                                     4
P5                                     25
**********************************
PERFORMANCE PREDICTIONS: (cycles)
L1 prediction :                        57
L2 prediction min :                    NA
L2 prediction avg :                    NA
RAM prediction :                       NA
L1 vect prediction :                   55.25
L2 vect prediction :                   NA
2                                      4, 11
14                                     ./ARCH/eo/cgsolv2.s, ,
No report for this loop
```

```
movups -8(%r12,%rax), %xmm11 #1
movups -4(%r12,%rax), %xmm1 #19
movups (%r12,%r9), %xmm0 #197.4
movups -4(%r12,%r10), %xmm3 #19
movups -4(%r12,%rdx), %xmm2 #19
movups -4(%r12,%rcx), %xmm5 #19
movups -4(%r12,%r11), %xmm4 #19
movups -4(%r12,%rbx), %xmm7 #20
movups -4(%r12,%rbp), %xmm6 #20
movups -4(%r12,%rsi), %xmm9 #201
movups -4(%r12,%r14), %xmm8 #20
movups -4(%r12,%rdi), %xmm10 #2
movups -4(%r12,%r9), %xmm12 #19
movups 12(%r12,%rax), %xmm14 #1
movups 12(%r12,%rdx), %xmm15 #1
mulps %xmm0, %xmm1 #197.40
movups 12(%r12,%r10), %xmm0 #19
mulps %xmm2, %xmm3 #198.40
movups 8(%r12,%rax), %xmm2 #196
mulps %xmm4, %xmm5 #199.40
mulps %xmm6, %xmm7 #200.40
mulps -8(%r12,%r9), %xmm11 #196.
mulps 8(%r12,%r9), %xmm2 #196.4
mulps %xmm8, %xmm9 #201.40
mulps %xmm15, %xmm0 #198.40
movsd 16(%r12,%r9), %xmm13 #197
movhps 24(%r12,%r9), %xmm13 #19
mulps %xmm13, %xmm14 #197.40
addps %xmm1, %xmm11 #197.21
movups 12(%r12,%rcx), %xmm1 #19
addps %xmm14, %xmm2 #197.21
addps %xmm3, %xmm11 #198.21
addps %xmm0, %xmm2 #198.21
movups 12(%r12,%r11), %xmm0 #19
mulps %xmm0, %xmm1 #199.40
movups 12(%r12,%rbp), %xmm0 #20
addps %xmm5, %xmm11 #199.21
addps %xmm1, %xmm2 #199.21
movups 12(%r12,%rbx), %xmm1 #20
mulps %xmm0, %xmm1 #200.40
movups 12(%r12,%r14), %xmm0 #20
addps %xmm7, %xmm11 #200.21
addps %xmm1, %xmm2 #200.21
movups 12(%r12,%rsi), %xmm1 #201
mulps %xmm0, %xmm1 #201.40
movups 12(%r12,%rdi), %xmm0 #20
addps %xmm9, %xmm11 #201.21
divps %xmm10, %xmm11 #202.15
addps %xmm1, %xmm2 #201.21
divps %xmm0, %xmm2 #202.15
movsd 12(%r12,%r9), %xmm1 #195.
```

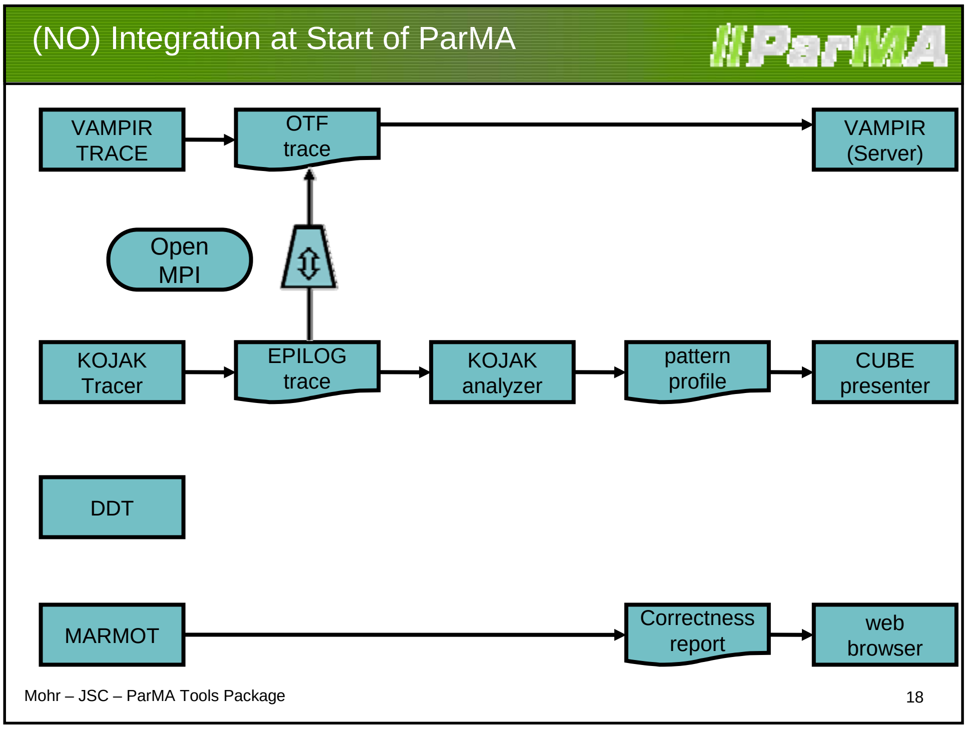
ParMA Tools ⇔ Competition

- DDT (Allinea) and Vampir (GWT-TUD)
  - Well established European HPC software vendors
- Marmot (HLRS / ZIH)
  - Competing tools either
    - vendor-specific only (NEC)
    - or of limited functionality (MPICH checker)
- Vampir (ZIH / GWT-TUD)
  - Competing tools either
    - vendor-specific only (Intel Trace Analyzer)
    - or less portable / un-supported (BSC Paraver)
- KOJAK/Scalasca (JSC)
  - World-leading product (no competitors)
- **Tool integrations** (DDT/Marmot, Vampir/Marmot, Scalasca/Vampir, …)
  - **UNIQUE!**



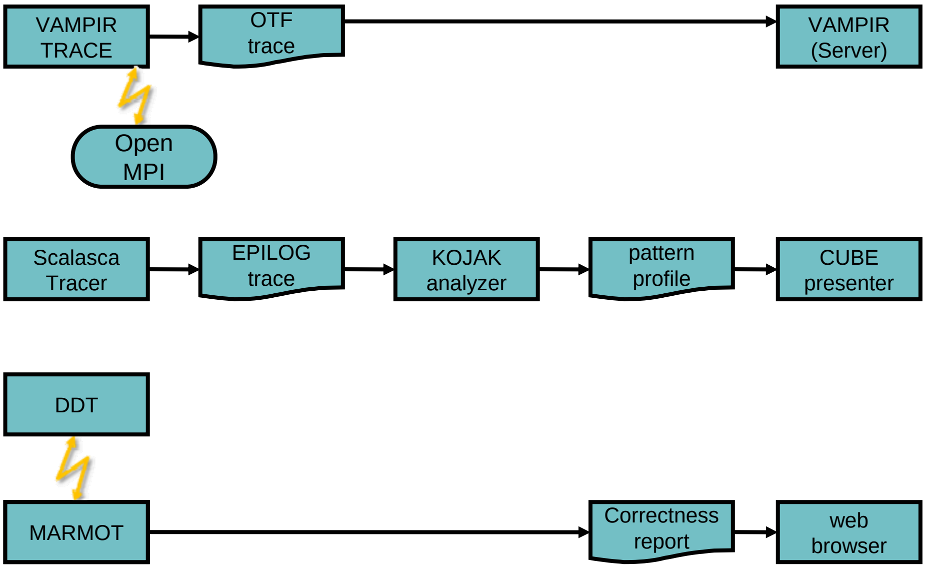
Content

- The ParMA Tools
- **Tool Integration**
- The ParMA Tools Package

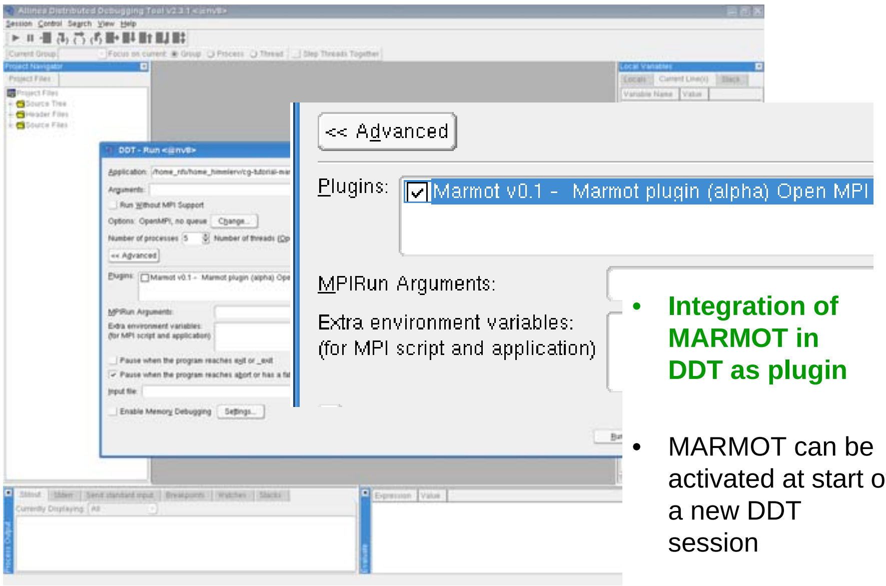


(NO) Integration at Start of ParMA

Final Tool Integration



Marmot ⇨ DDT Integration I



<< Advanced

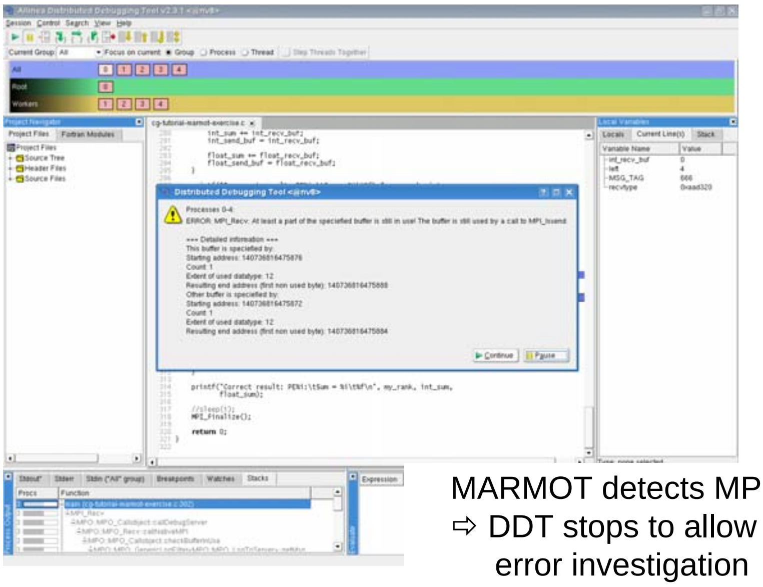
Plugins: ☑ Marmot v0.1 - Marmot plugin (alpha) Open MPI

MPIRun Arguments:

Extra environment variables:
(for MPI script and application)

- **Integration of MARMOT in DDT as plugin**
- MARMOT can be activated at start of a new DDT session

Marmot ⇨ DDT Integration II



```
      int_sum += int_recv_buf;
      int_send_buf = int_recv_buf;

      float_sum += float_recv_buf;
      float_send_buf = float_recv_buf;
   }
```

Distributed Debugging Tool <@nv8>

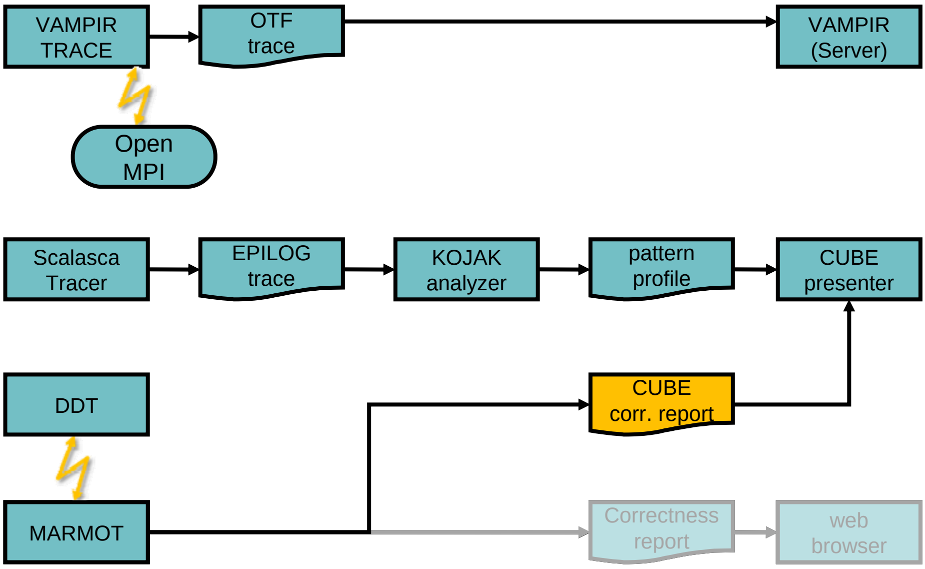
Processes 0-4:
ERROR: MPI_Recv: At least a part of the specified buffer is still in use! The buffer is still used by a call to MPI_Isend.

```
*** Detailed information ***
This buffer is specified by:
Starting address: 140736816475876
Count: 1
Extent of used datatype: 12
Resulting end address (first non used byte): 140736816475888
Other buffer is specified by:
Starting address: 140736816475872
Count: 1
Extent of used datatype: 12
Resulting end address (first non used byte): 140736816475884
```

```
   }

   printf("Correct result: PE%i:\tSum = %i\t%f\n", my_rank, int_sum,
          float_sum);

   //sleep(1);
   MPI_Finalize();

   return 0;
}
```

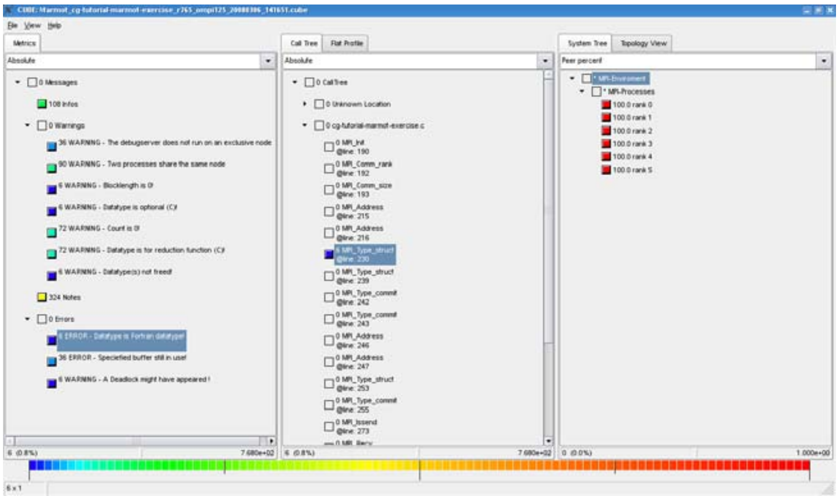
MARMOT detects MPI issue
⇨ DDT stops to allow further error investigation

Final Tool Integration



Marmot ⇨ CUBE

- **MARMOT result analysis using CUBE browser**



CUBE: Marmot_cg-tutorial-marmot-exercise_r765_ompi125_20080306_141651.cube

Metrics — Absolute

- 0 Messages
  - 108 Infos
  - 0 Warnings
    - 36 WARNING - The debugserver does not run on an exclusive node
    - 90 WARNING - Two processes share the same node
    - 6 WARNING - Blocklength is 0!
    - 6 WARNING - Datatype is optional (C)!
    - 72 WARNING - Count is 0!
    - 72 WARNING - Datatype is for reduction function (C)!
    - 6 WARNING - Datatype(s) not freed!
  - 324 Notes
  - 0 Errors
    - 6 ERROR - Datatype is Fortran datatype!
    - 36 ERROR - Specified buffer still in use!
    - 6 WARNING - A Deadlock might have appeared !

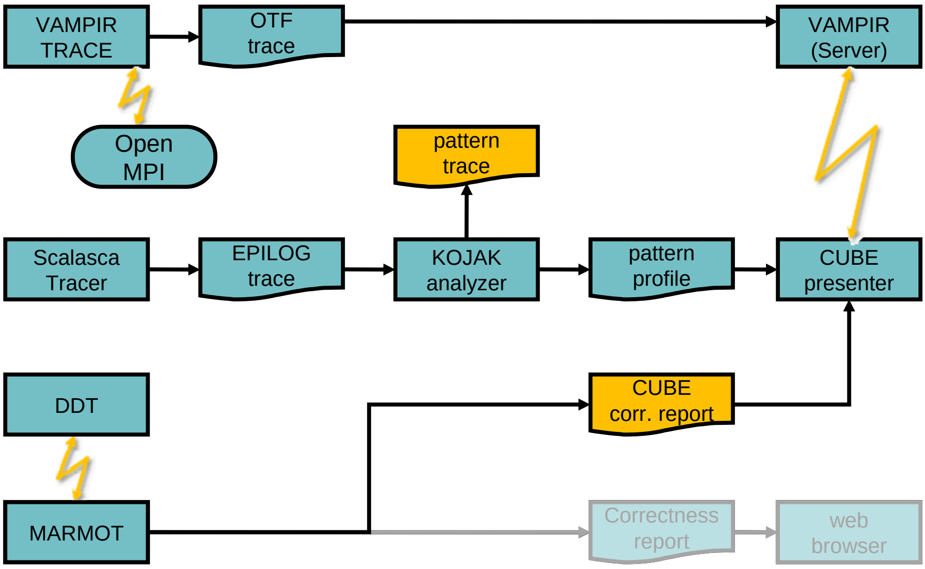
Call Tree — Absolute

- 0 Call Tree
  - 0 Unknown Location
  - 0 cg-tutorial-marmot-exercise.c
    - 0 MPI_Init @line: 190
    - 0 MPI_Comm_rank @line: 192
    - 0 MPI_Comm_size @line: 193
    - 0 MPI_Address @line: 215
    - 0 MPI_Address @line: 216
    - 6 MPI_Type_struct @line: 230
    - 0 MPI_Type_struct @line: 239
    - 0 MPI_Type_commit @line: 242
    - 0 MPI_Type_commit @line: 243
    - 0 MPI_Address @line: 246
    - 0 MPI_Address @line: 247
    - 0 MPI_Type_struct @line: 253
    - 0 MPI_Type_commit @line: 255
    - 0 MPI_Issend @line: 273
    - 0 MPI_Recv

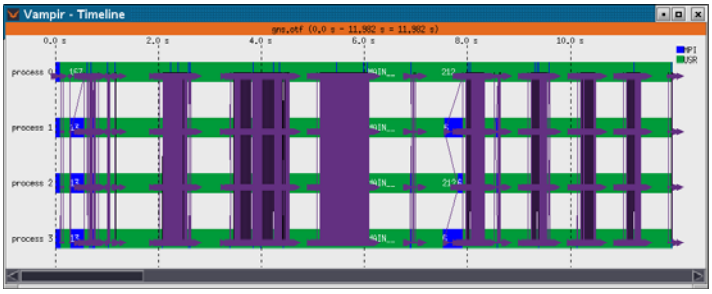
System Tree — Peer percent

- * MPI-Environment
  - * MPI-Processes
    - 100.0 rank 0
    - 100.0 rank 1
    - 100.0 rank 2
    - 100.0 rank 3
    - 100.0 rank 4
    - 100.0 rank 5

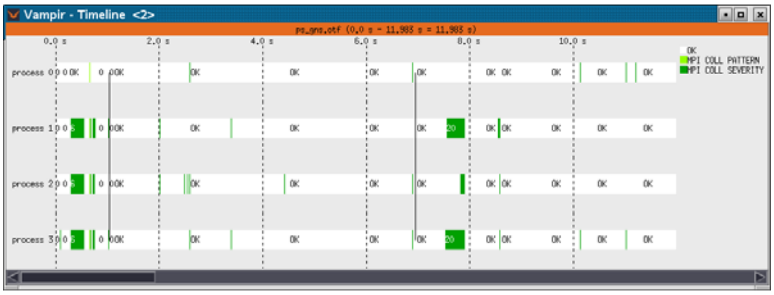
Final Tool Integration



VAMPIR ⇨ KOJAK via Pattern Traces

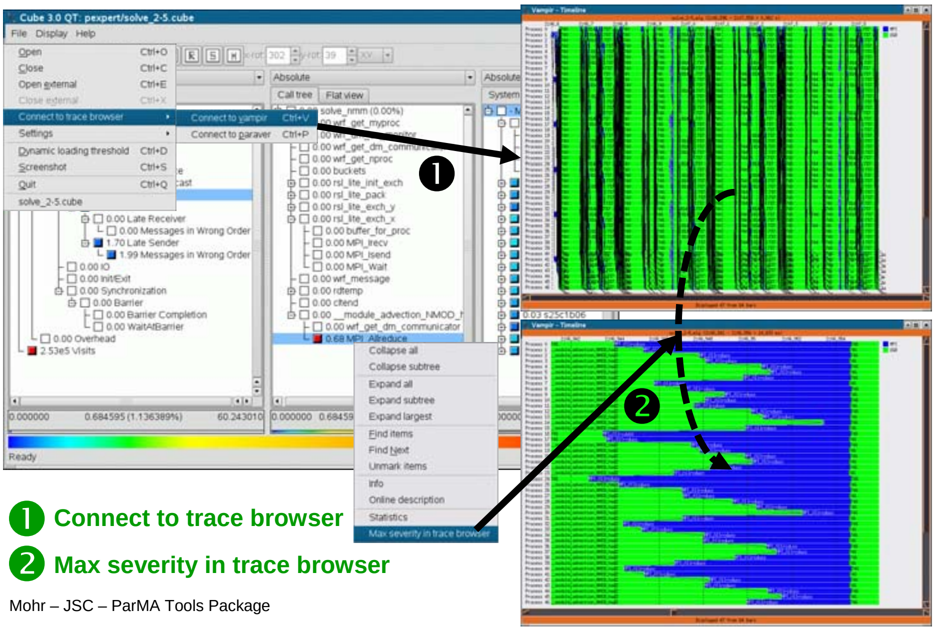


Original Vampir event trace



Pattern trace generated by KOJAK analysis highlighting problematic areas

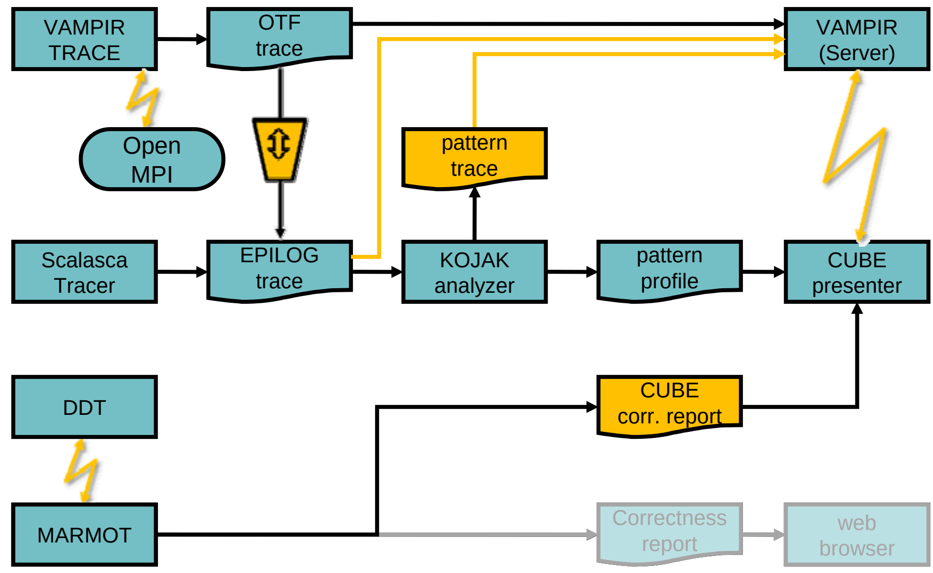
KOJAK ⇨ Vampir Integration



❶ Connect to trace browser

❷ Max severity in trace browser

Final Tool Integration (D3.all.5)



Vampir ⇔ KOJAK/Scalasca Integration

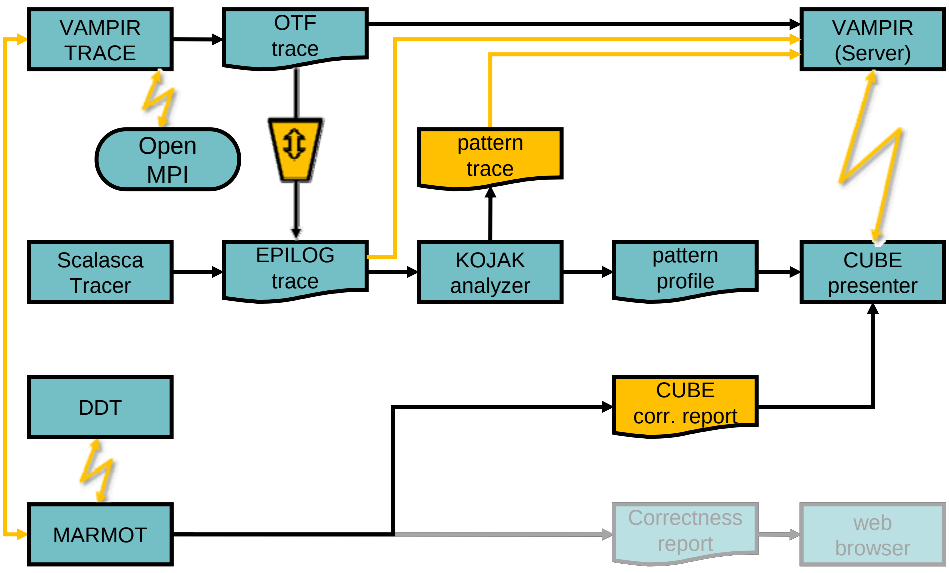
- VampirServer can read
  - KOJAK sequential EPILOG traces
  - Scalasca parallel EPILOG traces

  in addition to native OTF traces

- New OTF-to-EPILOG converter
  - Also: beta version of KOJAK’s event pattern trace analyzer based on new Œ interface (common OTF/EPILOG reading interface)

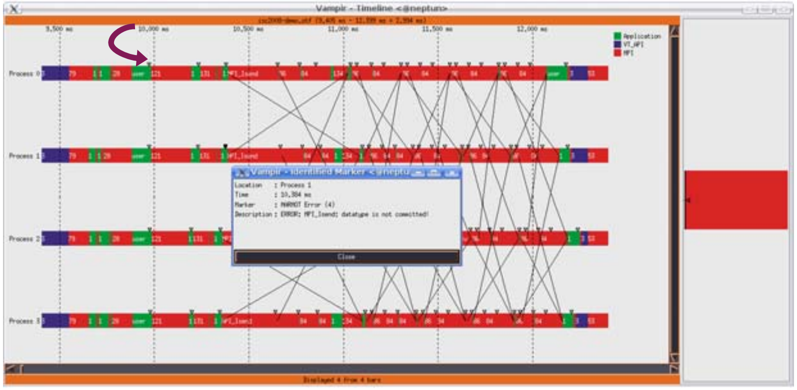
- **New** BMBF SILC **project**:
  - **full integration** of **Scalasca** and **VampirTrace** measurement systems and trace formats

Final Tool Integration



Marmot ⇨ Vampir Integration

- **Uni**versal **M**PI **C**orrectness **I**nterface for integration of MPI correctness tools into other tools (UniMCI)
- Implemented by Marmot (Producer) and VampirTrace (Consumer)



```
Location    : Process 1
Time        : 10,384 ms
Marker      : MARMOT Error (4)
Description : ERROR: MPI_Isend: datatype is not commited!
```

Content

- The ParMA Tools
- Tool Integration
- **The ParMA Tools Package**

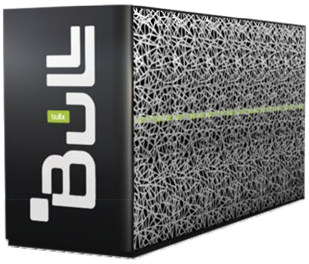
ParMA Tools Package I

- **UN**ified **I**ntegrated **T**ool **E**nvironment
- UNITE website: **http://apps.fz-juelich.de/unite/**
- Lower bar for inexperienced users and admins
  - **Common** usage and installation **documentation**
  - **Download, build and install**
    all ParMA tools **in one package:**
    - UNITE
      package installer and
      module package
    - OTF-1.6.5
    - pdtoolkit-3.15
    - cube-3.3
    - Scalasca-1.3.1
    - Vampirtrace-5.8.2
    - UniMCI-1.0.1
    - Marmot-2.4
    - Vampir-5.x or 7.x **(*)**
    - VampirServer-1.x, 2.x **(*)**

(*) Valid icense required

ParMA Tools Package II

- Extensively tested on
  - Itanium/IA32/x86_64 platforms with various MPI libraries (MPICH1, MPICH2, OpenMPI, Intel MPI, LAM, BullMPI, Parastation MPI, SGI MPT, ...)
  - AIX and Solaris clusters
- Already in use on Bull Nova and production machines of JSC, ZIH, RWTH, HLRN, (and soon LRZ, HLRS, …)



- Future work (beyond ParMA):
  - Integration of rest of ParMA tools (DDT, MAQAO, STEP)
  - Integration of other tools (Paraver, TAU, …)
  - More platforms (Cray XT, IBM BlueGene, NEC)