Performance Engineering using MVAPICH and TAU

Sameer Shende
University of Oregon

MVAPICH User's Group (MUG) Meeting
11:15 am – 11:45 am, Wednesday, August 8, 2018
Ohio Supercomputing Center, Columbus, Ohio

Acknowledgments

- **The MVAPICH2 team The Ohio State University**
  - http://mvapich.cse.ohio-state.edu
- **TAU team at the University of Oregon**
  - http://tau.uoregon.edu

Outline

- Introduction
- The MPI Tools Interfaces and Benefits
- Integrating TAU and MVAPICH2 with MPI_T
- Use Cases
- TAU Performance System®

Broad Challenge

Can we design dynamic and adaptive point-to-point communication mechanisms that can deliver the best

1. *Communication performance*
2. *Overlap of computation and communication*
3. *Memory footprint*

TAU Performance System®

- **Tuning and Analysis Utilities (22+ year project)**
- **Comprehensive performance profiling and tracing**
  - Integrated, scalable, flexible, portable
  - Targets all parallel programming/execution paradigms

- **Integrated performance toolkit**
  - Instrumentation, measurement, analysis, visualization
  - Widely-ported performance profiling / tracing system
  - Performance data management and data mining
  - Open source (BSD-style license)
  - Uses performance and control variables to interface with MVAPICH2
- **Integrates with application frameworks**
- **http://tau.uoregon.edu**

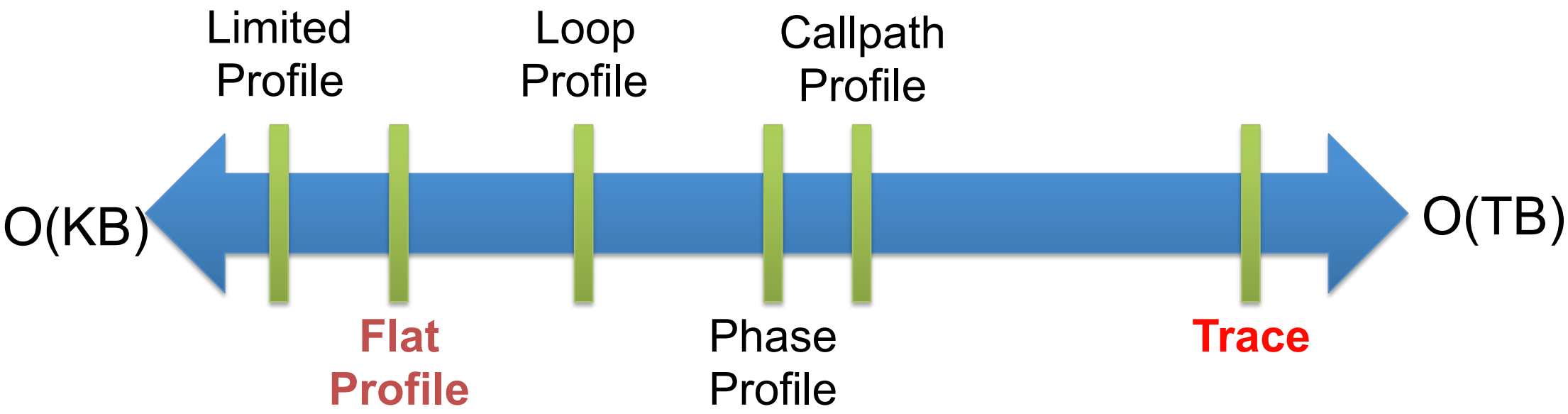
http://tau.uoregon.edu/mug18.pdf

Understanding Application Performance using TAU

- **How much time** is spent in each application routine and outer *loops*? Within loops, what is the contribution of each *statement*?
- **How many instructions** are executed in these code regions?
  Floating point, Level 1 and 2 *data cache misses*, hits, branches taken?
- **What is the memory usage** of the code? When and where is memory allocated/de-allocated? Are there any memory leaks?
- **What are the I/O characteristics** of the code? What is the peak read and write *bandwidth* of individual calls, total volume?
- **What is the contribution of each *phase*** of the program? What is the time wasted/spent waiting for collectives, and I/O operations in Initialization, Computation, I/O phases?
- **How does the application *scale***? What is the efficiency, runtime breakdown of performance across different core counts?
- **How can I tune MPI for better performance?** What performance and control does MVAPICH2 export to observe and control its performance?

http://tau.uoregon.edu/mug18.pdf

How much data do you want?



All levels support multiple metrics/counters

http://tau.uoregon.edu/mug18.pdf

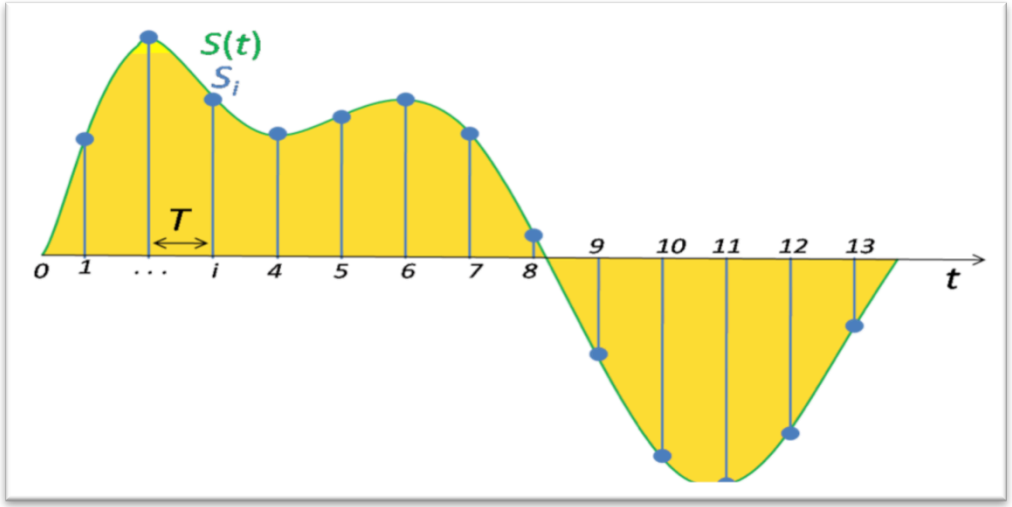
Performance Data Measurement

Direct via Probes

```
call TAU_START('potential')
// code
call TAU_STOP('potential')
```

- Exact measurement
- Fine-grain control
- Calls inserted into code

Indirect via Sampling



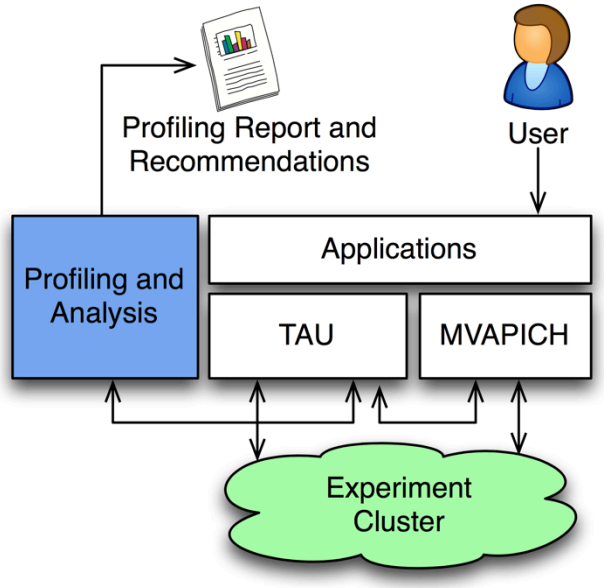
- No code modification
- Minimal effort
- Relies on debug symbols (**-g** option)
- TAU_SAMPLING=1

http://tau.uoregon.edu/mug18.pdf

Outline

- Introduction
- The MPI Tools Interfaces and Benefits
- Integrating TAU and MVAPICH2 with MPI_T
- Use Cases
- TAU Performance System®

MVAPICH2 and TAU



- **TAU and MVAPICH2 are enhanced with the ability to generate recommendations and engineering performance report**
- **MPI libraries like MVAPICH2 are now "reconfigurable" at runtime**
- **TAU and MVAPICH2 communicate using the MPI-T interface**

http://tau.uoregon.edu/mug18.pdf

MPI_T support with MVAPICH2

- Support performance variables (PVAR)
  - Variables to track different components within the MPI library
- Initial support for Control Variables (CVAR)
  - Variables to modify the behavior of MPI Library

Memory Usage:
- current level
- maximum watermark

InfiniBand N/W:
- #control packets
- #out-of-order packets

Pt-to-pt messages:
- unexpected queue length
- unexp. match attempts
- recvq. length

Registration cache:
- hits
- misses

Shared-memory:
- limic/ CMA
- buffer pool size & usage

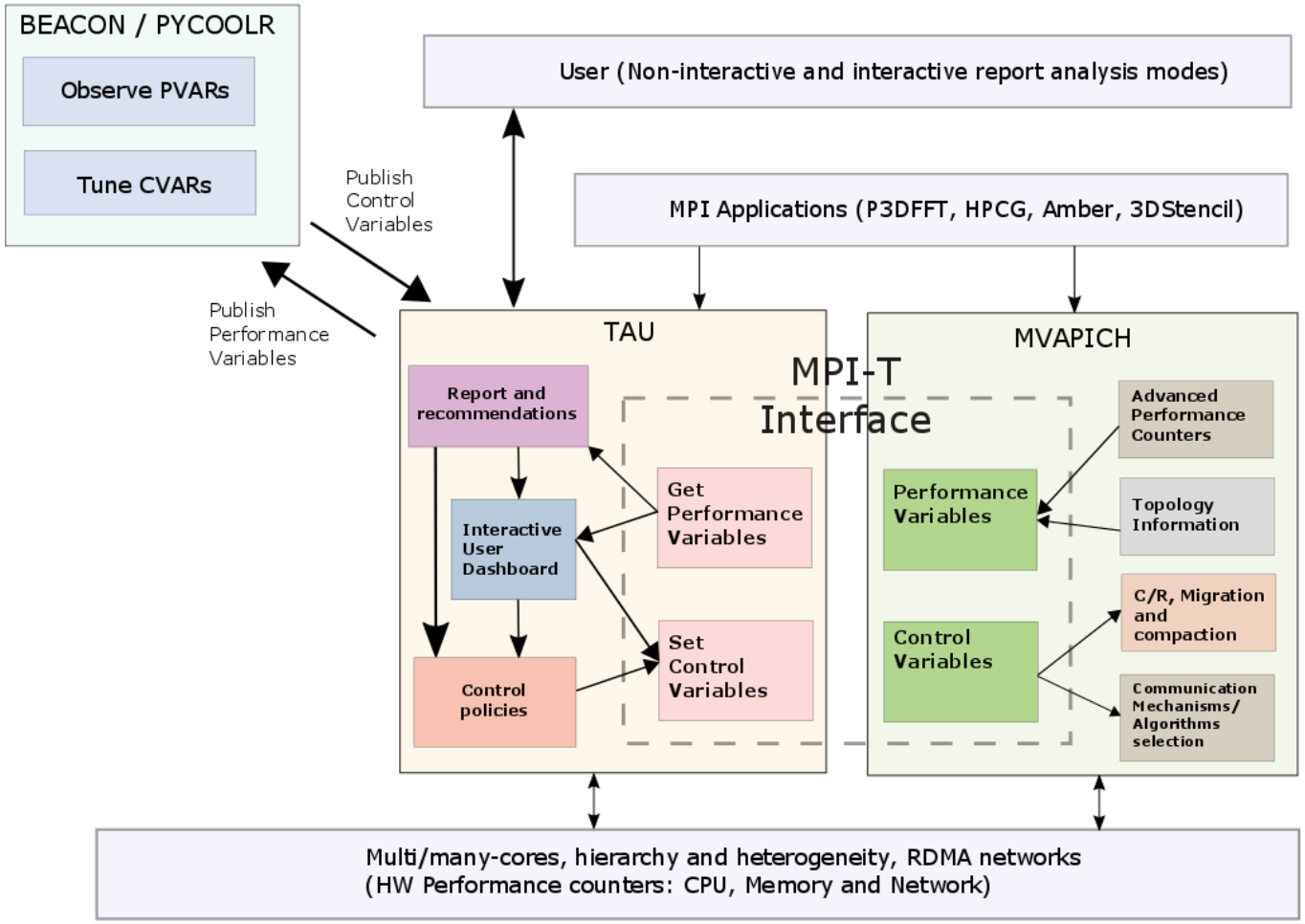
Collective ops:
- comm. creation
- #algorithm invocations
[Bcast – 8; Gather – 10]
…

http://tau.uoregon.edu/mug18.pdf

Outline

- Introduction
- The MPI Tools Interfaces and Benefits
- Integrating TAU and MVAPICH2 with MPI_T
- Use Cases
- TAU Performance System®

Interacting TAU with MVAPICH2 through MPI_T Interface

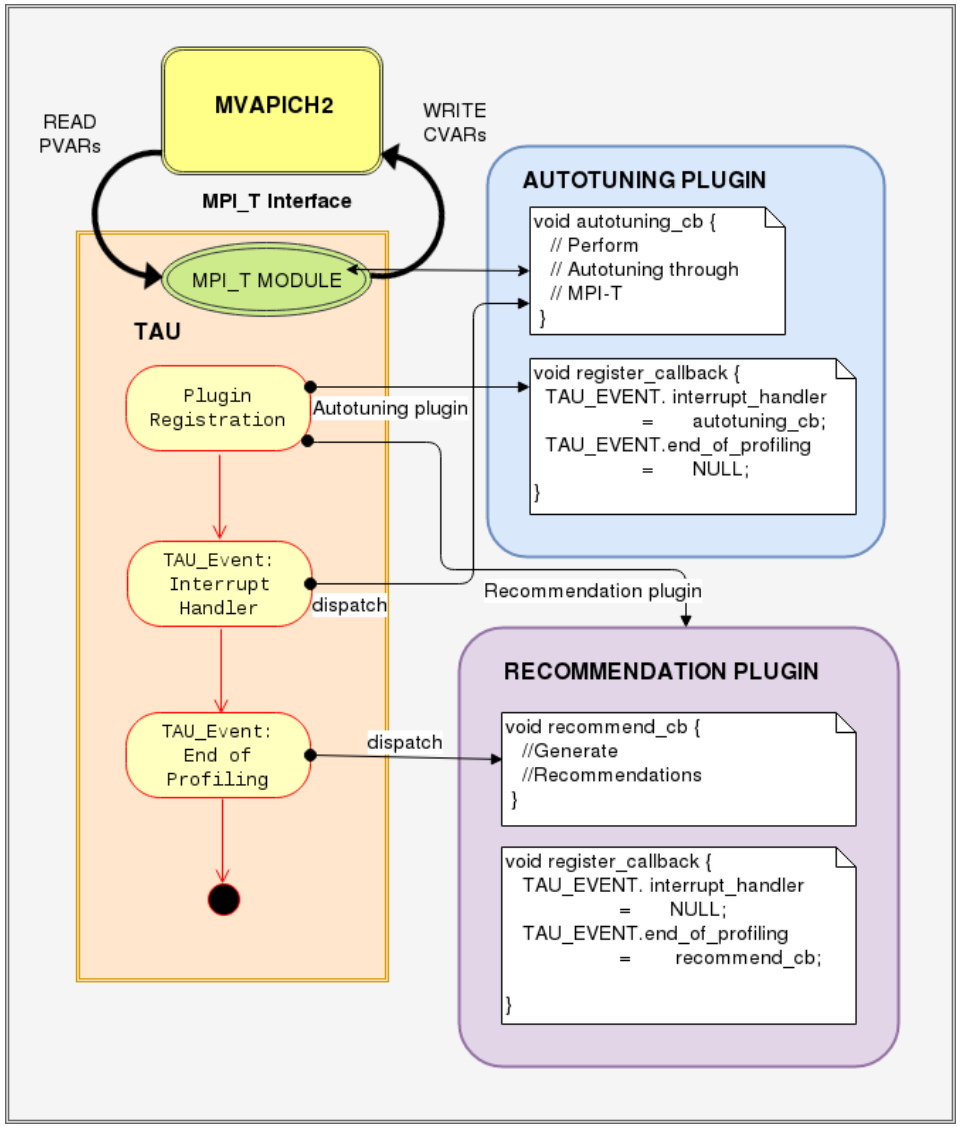


- **Enhance existing support for MPI_T in MVAPICH2 to expose a richer set of performance and control variables**
- **Get and display MPI Performance Variables (PVARs) made available by the runtime in TAU**
- **Control the runtime's behavior via MPI Control Variables (CVARs)**
- **Add support to MVAPICH2 and TAU for interactive performance engineering sessions**

http://tau.uoregon.edu/mug18.pdf

Plugin-based Infrastructure for Non-Interactive Tuning

- **Performance data collected by TAU**
  - Support for PVARs and CVARs
  - Setting CVARs to control MVAPICH2
  - Studying performance data in TAU's ParaProf profile browser
  - Multiple plugins available for
    - Tuning application at runtime and
    - Generate post-run recommendations



http://tau.uoregon.edu/mug18.pdf

Enhancing MPI_T Support

- **Introduced support for new MPI_T based CVARs to MVAPICH2**
  - MPIR_CVAR_MAX_INLINE_MSG_SZ
    - Controls the message size up to which “inline” transmission of data is supported by MVAPICH2
  - MPIR_CVAR_VBUF_POOL_SIZE
    - Controls the number of internal communication buffers (VBUFs) MVAPICH2 allocates initially. Also, MPIR_CVAR_VBUF_POOL_REDUCED_VALUE[1] ([2…n])
  - MPIR_CVAR_VBUF_SECONDARY_POOL_SIZE
    - Controls the number of VBUFs MVAPICH2 allocates when there are no more free VBUFs available
  - MPIR_CVAR_IBA_EAGER_THRESHOLD
    - Controls the message size where MVAPICH2 switches from eager to rendezvous protocol for large messages
- **TAU enhanced with support for setting MPI_T CVARs in a non-interactive mode for uninstrumented applications**

http://tau.uoregon.edu/mug18.pdf

MVAPICH2

- **Several new MPI_T based PVARs added to MVAPICH2**
  - mv2_vbuf_max_use, mv2_total_vbuf_memory etc
- **Enhanced TAU with support for tracking of MPI_T PVARs and CVARs for uninstrumented applications**
  - ParaProf, TAU's visualization front end, enhanced with support for displaying PVARs and CVARs
  - TAU provides tau_exec, a tool to transparently instrument MPI routines
    - Uninstrumented:
      % mpirun –np 1024 ./a.out
    - Instrumented:
      % mpirun –np 1024 tau_exec [options] ./a.out
      % paraprof

http://tau.uoregon.edu/mug18.pdf

PVARs Exposed by MVAPICH2

TAU: ParaProf Manager

File Options Help

- Applications
  - Standard Applications
    - Default App
      - Default Exp
        - lulesh.ppk
          - TIME
  - Default (jdbc:h2:/home

TrialField	Value
MPI_T PVAR[0]: mem_allocated	Current level of allocated memory within the MPI library
MPI_T PVAR[10]: mv2_num_2level_comm_success	Number of successful 2-level comm creations
MPI_T PVAR[11]: mv2_num_shmem_coll_calls	Number of times MV2 shared-memory collective calls were invoked
MPI_T PVAR[12]: mpit_progress_poll	CH3 RDMA progress engine polling count
MPI_T PVAR[13]: mv2_smp_read_progress_poll	CH3 SMP read progress engine polling count
MPI_T PVAR[14]: mv2_smp_write_progress_poll	CH3 SMP write progress engine polling count
MPI_T PVAR[15]: mv2_smp_read_progress_poll_success	Unsucessful CH3 SMP read progress engine polling count
MPI_T PVAR[16]: mv2_smp_write_progress_poll_succ...	Unsucessful CH3 SMP write progress engine polling count
MPI_T PVAR[17]: rdma_ud_retransmissions	CH3 RDMA UD retransmission count
MPI_T PVAR[18]: mv2_coll_bcast_binomial	Number of times MV2 binomial bcast algorithm was invoked
MPI_T PVAR[19]: mv2_coll_bcast_scatter_doubling_all...	Number of times MV2 scatter+double allgather bcast algorithm was invoked
MPI_T PVAR[1]: mem_allocated	Maximum level of memory ever allocated within the MPI library
MPI_T PVAR[20]: mv2_coll_bcast_scatter_ring_allgather	Number of times MV2 scatter+ring allgather bcast algorithm was invoked
MPI_T PVAR[21]: mv2_coll_bcast_scatter_ring_allgath...	Number of times MV2 scatter+ring allgather shm bcast algorithm was invoked
MPI_T PVAR[22]: mv2_coll_bcast_shmem	Number of times MV2 shmem bcast algorithm was invoked
MPI_T PVAR[23]: mv2_coll_bcast_knomial_internode	Number of times MV2 knomial internode bcast algorithm was invoked
MPI_T PVAR[24]: mv2_coll_bcast_knomial_intranode	Number of times MV2 knomial intranode bcast algorithm was invoked
MPI_T PVAR[25]: mv2_coll_bcast_mcast_internode	Number of times MV2 mcast internode bcast algorithm was invoked
MPI_T PVAR[26]: mv2_coll_bcast_pipelined	Number of times MV2 pipelined bcast algorithm was invoked
MPI_T PVAR[27]: mv2_coll_alltoall_inplace	Number of times MV2 in-place alltoall algorithm was invoked
MPI_T PVAR[28]: mv2_coll_alltoall_bruck	Number of times MV2 brucks alltoall algorithm was invoked
MPI_T PVAR[29]: mv2_coll_alltoall_rd	Number of times MV2 recursive-doubling alltoall algorithm was invoked
MPI_T PVAR[2]: num_malloc_calls	Number of MPIT_malloc calls
MPI_T PVAR[30]: mv2_coll_alltoall_sd	Number of times MV2 scatter-destination alltoall algorithm was invoked
MPI_T PVAR[31]: mv2_coll_alltoall_pw	Number of times MV2 pairwise alltoall algorithm was invoked
MPI_T PVAR[32]: mpit_alltoallv_mv2_pw	Number of times MV2 pairwise alltoallv algorithm was invoked
MPI_T PVAR[33]: mv2_coll_allreduce_shm_rd	Number of times MV2 shm rd allreduce algorithm was invoked
MPI_T PVAR[34]: mv2_coll_allreduce_shm_rs	Number of times MV2 shm rs allreduce algorithm was invoked
MPI_T PVAR[35]: mv2_coll_allreduce_shm_intra	Number of times MV2 shm intra allreduce algorithm was invoked
MPI_T PVAR[36]: mv2_coll_allreduce_intra_p2p	Number of times MV2 intra p2p allreduce algorithm was invoked
MPI_T PVAR[37]: mv2_coll_allreduce_2lvl	Number of times MV2 two-level allreduce algorithm was invoked
MPI_T PVAR[38]: mv2_coll_allreduce_shmem	Number of times MV2 shmem allreduce algorithm was invoked
MPI_T PVAR[39]: mv2_coll_allreduce_mcast	Number of times MV2 multicast-based allreduce algorithm was invoked
MPI_T PVAR[3]: num_calloc_calls	Number of MPIT_calloc calls
MPI_T PVAR[40]: mv2_reg_cache_hits	Number of registration cache hits
MPI_T PVAR[41]: mv2_reg_cache_misses	Number of registration cache misses
MPI_T PVAR[42]: mv2_vbuf_allocated	Number of VBUFs allocated
MPI_T PVAR[43]: mv2_vbuf_allocated_array	Number of VBUFs allocated
MPI_T PVAR[44]: mv2_vbuf_freed	Number of VBUFs freed
MPI_T PVAR[45]: mv2_ud_vbuf_allocated	Number of UD VBUFs allocated
MPI_T PVAR[46]: mv2_ud_vbuf_freed	Number of UD VBUFs freed
MPI_T PVAR[47]: mv2_vbuf_free_attempts	Number of time we attempted to free VBUFs
MPI_T PVAR[48]: mv2_vbuf_free_attempt_success_time	Average time for number of times we sucessfully freed VBUFs
MPI_T PVAR[49]: mv2_vbuf_free_attempt_success_time	Average time for number of times we sucessfully freed VBUFs
MPI_T PVAR[4]: num_memalign_calls	Number of MPIT_memalign calls
MPI_T PVAR[50]: mv2_vbuf_allocate_time	Average time for number of times we allocated VBUFs
MPI_T PVAR[51]: mv2_vbuf_allocate_time	Average time for number of times we allocated VBUFs

http://tau.uoregon.edu/mug18.pdf

CVARs Exposed by MVAPICH2

TAU: ParaProf Manager

File Options Help

- Applications
  - Standard Applications
    - Default App
      - Default Exp
        - lulesh.ppk
          - TIME
  - Default (jdbc:h2:/home

TrialField	Value
Local Time	2016-08-16T10:11:04-07:00
MPI Processor Name	cerberus.nic.uoregon.edu
MPIR_CVAR_ABORT_ON_LEAKED_HANDLES	If true, MPI will call MPI_Abort at MPI_Finalize if any MPI object handles have been leaked. For example,...
MPIR_CVAR_ALLGATHERV_PIPELINE_MSG_SIZE	The smallest message size that will be used for the pipelined, large-message, ring algorithm in the MPI_...
MPIR_CVAR_ALLGATHER_LONG_MSG_SIZE	For MPI_Allgather and MPI_Allgatherv, the long message algorithm will be used if the send buffer size is ...
MPIR_CVAR_ALLGATHER_SHORT_MSG_SIZE	For MPI_Allgather and MPI_Allgatherv, the short message algorithm will be used if the send buffer size is...
MPIR_CVAR_ALLREDUCE_SHORT_MSG_SIZE	the short message algorithm will be used if the send buffer size is <= this value (in bytes)
MPIR_CVAR_ALLTOALL_MEDIUM_MSG_SIZE	the medium message algorithm will be used if the per-destination message size (sendcount*size(sendtyp...
MPIR_CVAR_ALLTOALL_SHORT_MSG_SIZE	the short message algorithm will be used if the per-destination message size (sendcount*size(sendtype)) ...
MPIR_CVAR_ALLTOALL_THROTTLE	max no. of irecvs/isends posted at a time in some alltoall algorithms. Setting it to 0 causes all irecvs/isen...
MPIR_CVAR_ASYNC_PROGRESS	If set to true, MPICH will initiate an additional thread to make asynchronous progress on all communicati...
MPIR_CVAR_BCAST_LONG_MSG_SIZE	Let's define short messages as messages with size < MPIR_CVAR_BCAST_SHORT_MSG_SIZE, and mediu...
MPIR_CVAR_BCAST_MIN_PROCS	Let's define short messages as messages with size < MPIR_CVAR_BCAST_SHORT_MSG_SIZE, and mediu...
MPIR_CVAR_BCAST_SHORT_MSG_SIZE	Let's define short messages as messages with size < MPIR_CVAR_BCAST_SHORT_MSG_SIZE, and mediu...
MPIR_CVAR_CH3_EAGER_MAX_MSG_SIZE	This cvar controls the message size at which CH3 switches from eager to rendezvous mode.
MPIR_CVAR_CH3_ENABLE_HCOLL	If true, enable HCOLL collectives.
MPIR_CVAR_CH3_INTERFACE_HOSTNAME	If non-NULL, this cvar specifies the IP address that other processes should use when connecting to this pr...
MPIR_CVAR_CH3_NOLOCAL	If true, force all processes to operate as though all processes are located on another node. For example,...
MPIR_CVAR_CH3_ODD_EVEN_CLIQUES	If true, odd procs on a node are seen as local to each other, and even procs on a node are seen as local t...
MPIR_CVAR_CH3_PORT_RANGE	The MPIR_CVAR_CH3_PORT_RANGE environment variable allows you to specify the range of TCP ports ...
MPIR_CVAR_CH3_RMA_ACC_IMMED	Use the immediate accumulate optimization
MPIR_CVAR_CH3_RMA_GC_NUM_COMPLETED	Threshold for the number of completed requests the runtime finds before it stops trying to find more co...
MPIR_CVAR_CH3_RMA_GC_NUM_TESTED	Threshold for the number of RMA requests the runtime tests before it stops trying to check more reques...
MPIR_CVAR_CH3_RMA_LOCK_IMMED	Issue a request for the passive target RMA lock immediately. Default behavior is to defer the lock reque...
MPIR_CVAR_CH3_RMA_MERGE_LOCK_OP_UNLOCK	Enable/disable an optimization that merges lock, op, and unlock messages, for single-operation passive ta...
MPIR_CVAR_CH3_RMA_NREQUEST_NEW_THRESHOLD	Threshold for the number of new requests since the last attempt to complete pending requests. Higher ...
MPIR_CVAR_CH3_RMA_NREQUEST_THRESHOLD	Threshold at which the RMA implementation attempts to complete requests while completing RMA oper...
MPIR_CVAR_CHOP_ERROR_STACK	If >0, truncate error stack output lines this many characters wide. If 0, do not truncate, and if <0 use a ...
MPIR_CVAR_COLL_ALIAS_CHECK	Enable checking of aliasing in collective operations
MPIR_CVAR_COMM_SPLIT_USE_QSORT	Use qsort(3) in the implementation of MPI_Comm_split instead of bubble sort.
MPIR_CVAR_CTXID_EAGER_SIZE	The MPIR_CVAR_CTXID_EAGER_SIZE environment variable allows you to specify how many words in th...
MPIR_CVAR_DEBUG_HOLD	If true, causes processes to wait in MPI_Init and MPI_Initthread for a debugger to be attached. Once the ...
MPIR_CVAR_DEFAULT_THREAD_LEVEL	Sets the default thread level to use when using MPI_INIT.
MPIR_CVAR_DUMP_PROVIDERS	If true, dump provider information at init
MPIR_CVAR_ENABLE_COLL_FT_RET	DEPRECATED! Will be removed in MPICH-3.2 Collectives called on a communicator with a failed process...
MPIR_CVAR_ENABLE_SMP_ALLREDUCE	Enable SMP aware allreduce.
MPIR_CVAR_ENABLE_SMP_BARRIER	Enable SMP aware barrier.
MPIR_CVAR_ENABLE_SMP_BCAST	Enable SMP aware broadcast (See also: MPIR_CVAR_MAX_SMP_BCAST_MSG_SIZE)
MPIR_CVAR_ENABLE_SMP_COLLECTIVES	Enable SMP aware collective communication.
MPIR_CVAR_ENABLE_SMP_REDUCE	Enable SMP aware reduce.
MPIR_CVAR_ERROR_CHECKING	If true, perform checks for errors, typically to verify valid inputs to MPI routines. Only effective when M...
MPIR_CVAR_GATHERV_INTER_SSEND_MIN_PROCS	Use Ssend (synchronous send) for intercommunicator MPI_Gatherv if the "group B" size is >= this value....
MPIR_CVAR_GATHER_INTER_SHORT_MSG_SIZE	use the short message algorithm for intercommunicator MPI_Gather if the send buffer size is < this value...
MPIR_CVAR_GATHER_VSMALL_MSG_SIZE	use a temporary buffer for intracommunicator MPI_Gather if the send buffer size is < this value (in bytes...
MPIR_CVAR_IBA_EAGER_THRESHOLD	0 (old) -> 204800 (new), This set the switch point between eager and rendezvous protocol
MPIR_CVAR_MAX_INLINE_SIZE	This set the maximum inline size for data transfer
MPIR_CVAR_MAX_SMP_ALLREDUCE_MSG_SIZE	Maximum message size for which SMP-aware allreduce is used. A value of '0' uses SMP-aware allreduce ...

http://tau.uoregon.edu/mug18.pdf

Using MVAPICH2 and TAU

- To set CVARs or read PVARs using TAU for an uninstrumented binary:

```
% export TAU_TRACK_MPI_T_PVARS=1
% export TAU_MPI_T_CVAR_METRICS=
      MPIR_CVAR_VBUF_POOL_REDUCED_VALUE[1],
      MPIR_CVAR_IBA_EAGER_THRESHOLD
% export TAU_MPI_T_CVAR_VALUES=32,64000
% export PATH=/path/to/tau/x86_64/bin:$PATH
% mpirun -np 1024 tau_exec -T mvapich2,mpit   ./a.out
% paraprof
```

http://tau.uoregon.edu/mug18.pdf

VBUF usage without CVARs

TAU: ParaProf: Context Events for: node 0 - mpit_withoutcvar_bt.C.1k.ppk

Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamples	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	3,313,056	3,313,056	3,313,056	0	1	3,313,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	0	0
mv2_ud_vbuf_freed (Number of UD VBUFs freed)	0	0	0	0	0	0
mv2_ud_vbuf_inuse (Number of UD VBUFs inuse)	0	0	0	0	0	0
mv2_ud_vbuf_max_use (Maximum number of UD VBUFs used)	0	0	0	0	0	0
mv2_vbuf_allocated (Number of VBUFs allocated)	320	320	320	0	1	320
mv2_vbuf_available (Number of VBUFs available)	255	255	255	0	1	255
mv2_vbuf_freed (Number of VBUFs freed)	25,545	25,545	25,545	0	1	25,545
mv2_vbuf_inuse (Number of VBUFs inuse)	65	65	65	0	1	65
mv2_vbuf_max_use (Maximum number of VBUFs used)	65	65	65	0	1	65
num_calloc_calls (Number of MPIT_calloc calls)	89	89	89	0	1	89
num_free_calls (Number of MPIT_free calls)	47,801	47,801	47,801	0	1	47,801
num_malloc_calls (Number of MPIT_malloc calls)	49,258	49,258	49,258	0	1	49,258
num_memalign_calls (Number of MPIT_memalign calls)	34	34	34	0	1	34
num_memalign_free_calls (Number of MPIT_memalign_free calls)	0	0	0	0	0	0

http://tau.uoregon.edu/mug18.pdf

VBUF usage with CVARs

TAU: ParaProf: Context Events for: node 0 - bt-mz.E.vbuf_pool_16.1k.ppk

Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamp...	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	1,815,056	1,815,056	1,815,056	0	1	1,815,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	0	0
mv2_ud_vbuf_freed (Number of UD VBUFs freed)	0	0	0	0	0	0
mv2_ud_vbuf_inuse (Number of UD VBUFs inuse)	0	0	0	0	0	0
mv2_ud_vbuf_max_use (Maximum number of UD VBUFs used)	0	0	0	0	0	0
mv2_vbuf_allocated (Number of VBUFs allocated)	160	160	160	0	1	160
mv2_vbuf_available (Number of VBUFs available)	94	94	94	0	1	94
mv2_vbuf_freed (Number of VBUFs freed)	5,479	5,479	5,479	0	1	5,479
mv2_vbuf_inuse (Number of VBUFs inuse)	66	66	66	0	1	66
mv2_vbuf_max_use (Maximum number of VBUFs used)	66	66	66	0	1	66
num_calloc_calls (Number of MPIT_calloc calls)	89	89	89	0	1	89
num_free_calls (Number of MPIT_free calls)	130	130	130	0	1	130
num_malloc_calls (Number of MPIT_malloc calls)	1,625	1,625	1,625	0	1	1,625
num_memalign_calls (Number of MPIT_memalign calls)	56	56	56	0	1	56
num_memalign_free_calls (Number of MPIT_memalign_free calls)	0	0	0	0	0	0

TAU: ParaProf Manager

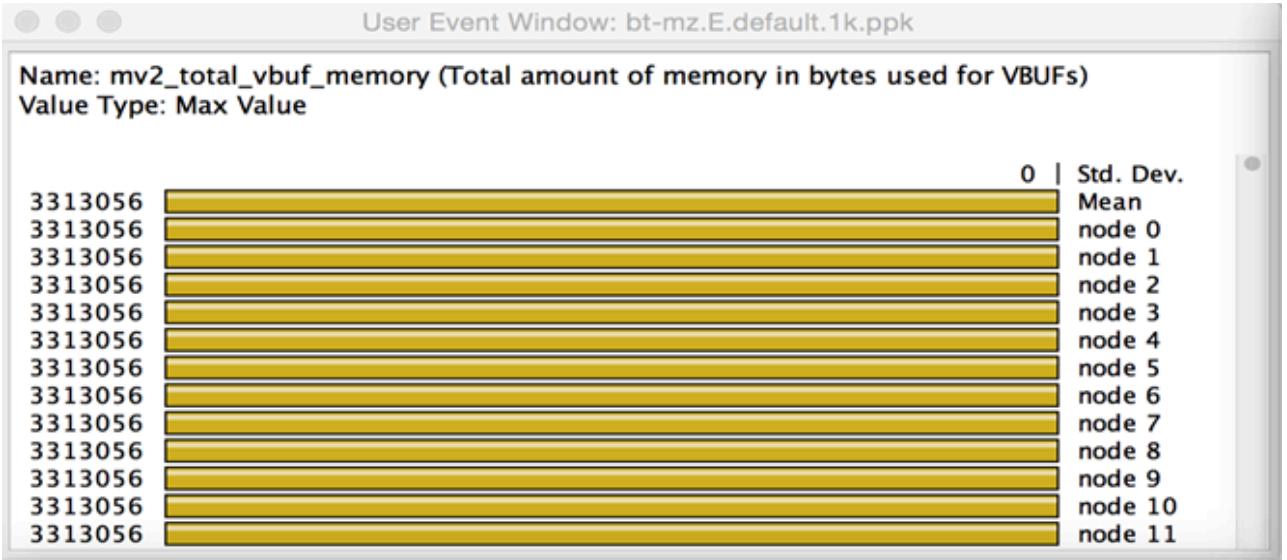
- Applications
  - Standard Applications
    - Default App
      - Default Exp
        - bt-mz.E.vbuf_pool_16.1k.pp
          - TIME

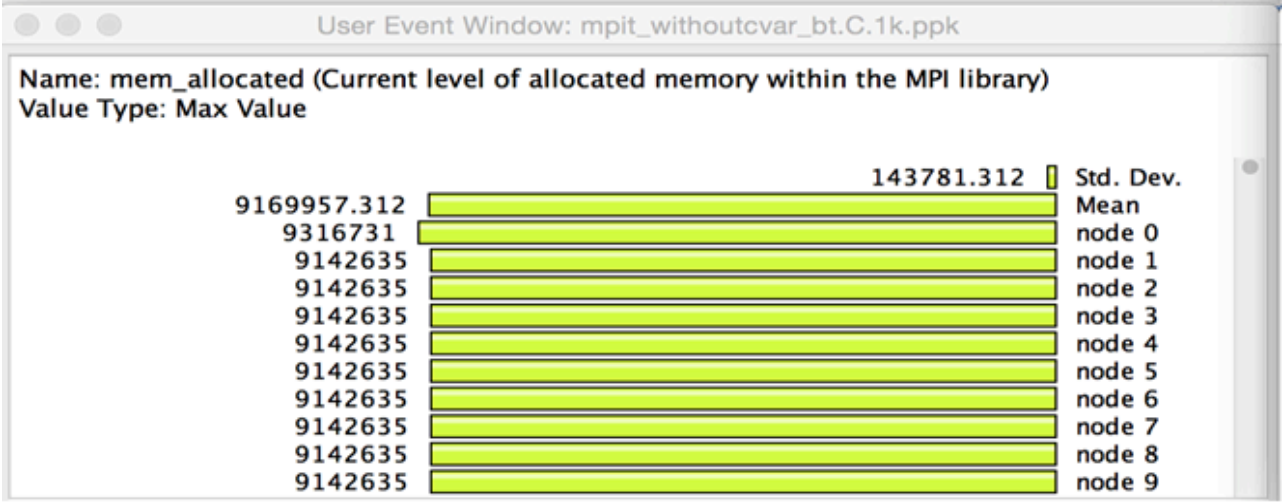
TrialField	Value
MPI Processor Name	c526-502.stampede.tacc.utexas.edu
MPIR_CVAR_VBUF_POOL_SIZE	0 (old) -> 16 (new), This set the size of the VBUF pool

Total memory used by VBUFs is reduced from 3,313,056 to 1,815,056

http://tau.uoregon.edu/mug18.pdf

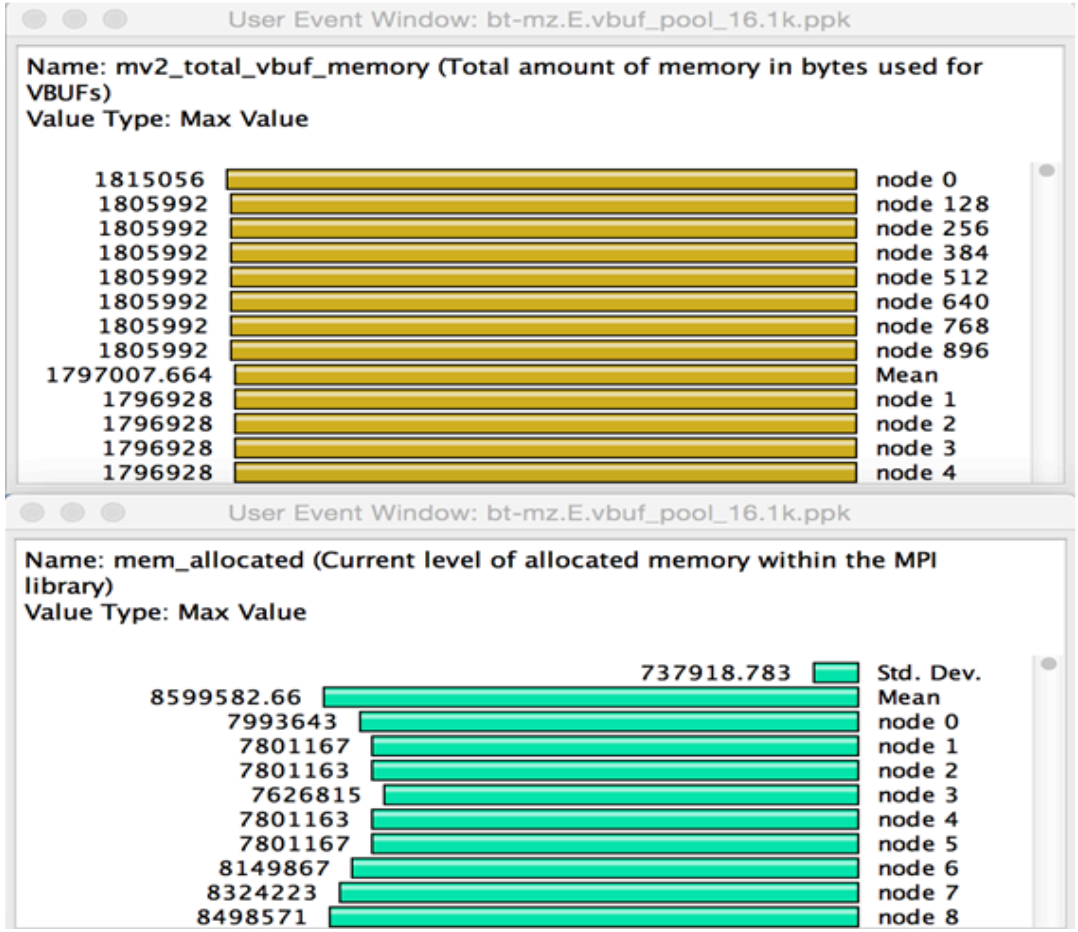
VBUF Memory Usage Without CVAR





http://tau.uoregon.edu/mug18.pdf

VBUF Memory Usage With CVAR




```
% export TAU_TRACK_MPI_T_PVARS=1
% export TAU_MPI_T_CVAR_METRICS=MPIR_CVAR_VBUF_POOL_SIZE
% export TAU_MPI_T_CVAR_VALUES=16
% mpirun -np 1024 tau_exec -T mvapich2  ./a.out
```

http://tau.uoregon.edu/mug18.pdf

TAU Performance System

Instrumentation

- Fortran, C++, C, UPC, Java, Python, Chapel, Spark
- Automatic instrumentation

Measurement and analysis support

- MPI, OpenSHMEM, ARMCI, PGAS, DMAPP
- pthreads, OpenMP, OMPT interface, hybrid, other thread models
- GPU, CUDA, OpenCL, OpenACC
- Parallel profiling and tracing
- Use of Score-P for native OTF2 and CUBEX generation

Analysis

- Parallel profile analysis (ParaProf), data mining (PerfExplorer)
- Performance database technology (TAUdb)
- 3D profile browser

http://tau.uoregon.edu/mug18.pdf

Instrumentation

Add hooks in the code to perform measurements

Source instrumentation using a preprocessor

- Add timer start/stop calls in a copy of the source code.
- Use Program Database Toolkit (PDT) for parsing source code.
- Requires recompiling the code using TAU shell scripts (tau_cc.sh, tau_f90.sh)
- Selective instrumentation (filter file) can reduce runtime overhead and narrow instrumentation focus.

Compiler-based instrumentation

- Use system compiler to add a special flag to insert hooks at routine entry/exit.
- Requires recompiling using TAU compiler scripts (tau_cc.sh, tau_f90.sh…)

Runtime preloading of TAU's Dynamic Shared Object (DSO)

- No need to recompile code! Use **mpirun tau_exec ./app** with options.
- Requires dynamic executable (link using **−dynamic** on Theta).

TAU Execution Command (tau_exec)

Uninstrumented execution

- % mpirun -np 256 ./a.out

Track GPU operations

- % mpirun –np 256 tau_exec –cupti ./a.out
- % mpirun –np 256 tau_exec –cupti -um ./a.out (for Unified Memory)
- % mpirun –np 256 tau_exec –opencl ./a.out
- % mpirun –np 256 tau_exec –openacc ./a.out

Track MPI performance using MPI_T interface

- % mpirun -np 256 tau_exec –T mpit ./a.out

Track OpenMP, and MPI performance (MPI enabled by default)

- % export TAU_OMPT_SUPPORT_LEVEL=full;
  % export TAU_OMPT_RESOLVE_ADDRESS_EAGERLY=1
- % mpirun -np 256 tau_exec –T ompt,tr6,mpi –ompt ./a.out

Track memory operations

- % export TAU_TRACK_MEMORY_LEAKS=1
- % mpirun –np 256 tau_exec –memory_debug ./a.out (bounds check)

Use event based sampling (compile with –g)

- % mpirun –np 256 tau_exec –ebs ./a.out
- Also –ebs_source=<PAPI_COUNTER> -ebs_period=<overflow_count>

http://tau.uoregon.edu/mug18.pdf

Using TAU

TAU supports several measurement and thread options

Phase profiling, profiling with hardware counters (papi), MPI library, CUDA, Beacon (backplane for event notification – online monitoring), PDT (automatic source instrumentation) …

Each measurement configuration of TAU corresponds to a unique stub makefile and library that is generated when you configure it

To instrument source code automatically using PDT

Choose an appropriate TAU stub makefile in <arch>/lib:

% source ~tg457572/tau.bashrc

% export TAU_MAKEFILE=$TAU/Makefile.tau-mvapich2-icpc-mpi-pdt

% export TAU_OPTIONS= '-optVerbose …' (see tau_compiler.sh)

Use tau_f90.sh, tau_cxx.sh, tau_upc.sh, or tau_cc.sh as F90, C++, UPC, or C compilers respectively:

% mpif90 foo.f90 changes to

% tau_f90.sh foo.f90

Set runtime environment variables, execute application and analyze performance data:

% pprof (for text based profile display)

% paraprof (for GUI)

http://tau.uoregon.edu/mug18.pdf

Choosing TAU_MAKEFILE

```
% source ~tg457572/tau.bashrc
% ls $TAU/Makefile.*
```

/home1/00494/tg457572/pkgs/tau_latest/x86_64/lib/Makefile.tau-mvapich2-icpc-mpi-pdt
/home1/00494/tg457572/pkgs/tau_latest/x86_64/lib/Makefile.tau-mvapich2-icpc-mpi-pthread-pdt
/home1/00494/tg457572/pkgs/tau_latest/x86_64/lib/Makefile.tau-mvapich2-icpc-ompt-mpi-pdt-openmp
/home1/00494/tg457572/pkgs/tau_latest/x86_64/lib/Makefile.tau-mvapich2-icpc-papi-mpi-pdt-mpit
/home1/00494/tg457572/pkgs/tau_latest/x86_64/lib/Makefile.tau-mvapich2-icpc-papi-ompt-mpi-pdt-openmp
/home1/00494/tg457572/pkgs/tau_latest/x86_64/lib/Makefile.tau-pdt

For an MPI+F90 application with MPI, you may choose

Makefile.tau-mvapich2-icpc-mpi-pdt

- Supports MPI instrumentation, papi, and PDT for automatic source instrumentation

```
% export TAU_MAKEFILE=$TAU/Makefile.tau-mvapich2-icpc-mpi-pdt
% tau_f90.sh matrix.f90 -o matrix
OR with build systems:
% make CC=tau_cc.sh CXX=tau_cxx.sh F90=tau_f90.sh
% cmake -DCMAKE_Fortran_COMPILER=tau_f90.sh
      -DCMAKE_C_COMPILER=tau_cc.sh -
DCMAKE_CXX_COMPILER=tau_cxx.sh
% mpirun -np 1024 ./matrix
% paraprof
```

http://tau.uoregon.edu/mug18.pdf

Configuration tags for tau_exec

```
% ./configure -pdt=<dir> -mpi -papi=<dir>; make install
```

Creates in $TAU:

```
Makefile.tau-papi-mpi-pdt(Configuration parameters in stub makefile)
shared-papi-mpi-pdt/libTAU.so

% ./configure -pdt=<dir> -mpi; make install  creates
Makefile.tau-mpi-pdt
shared-mpi-pdt/libTAU.so
```

To explicitly choose preloading of shared-<options>/libTAU.so change:

```
% mpirun -np 256 ./a.out     to
% mpirun -np 256  tau_exec -T <comma_separated_options> ./a.out

% mpirun -np 256  tau_exec -T papi,mpi,pdt ./a.out
```

Preloads `$TAU/shared-papi-mpi-pdt/libTAU.so`

```
% mpirun -np 256  tau_exec -T papi ./a.out
```

Preloads `$TAU/shared-papi-mpi-pdt/libTAU.so` by matching.

```
% mpirun -np 256 tau_exec -T papi,mpi,pdt -s ./a.out
```

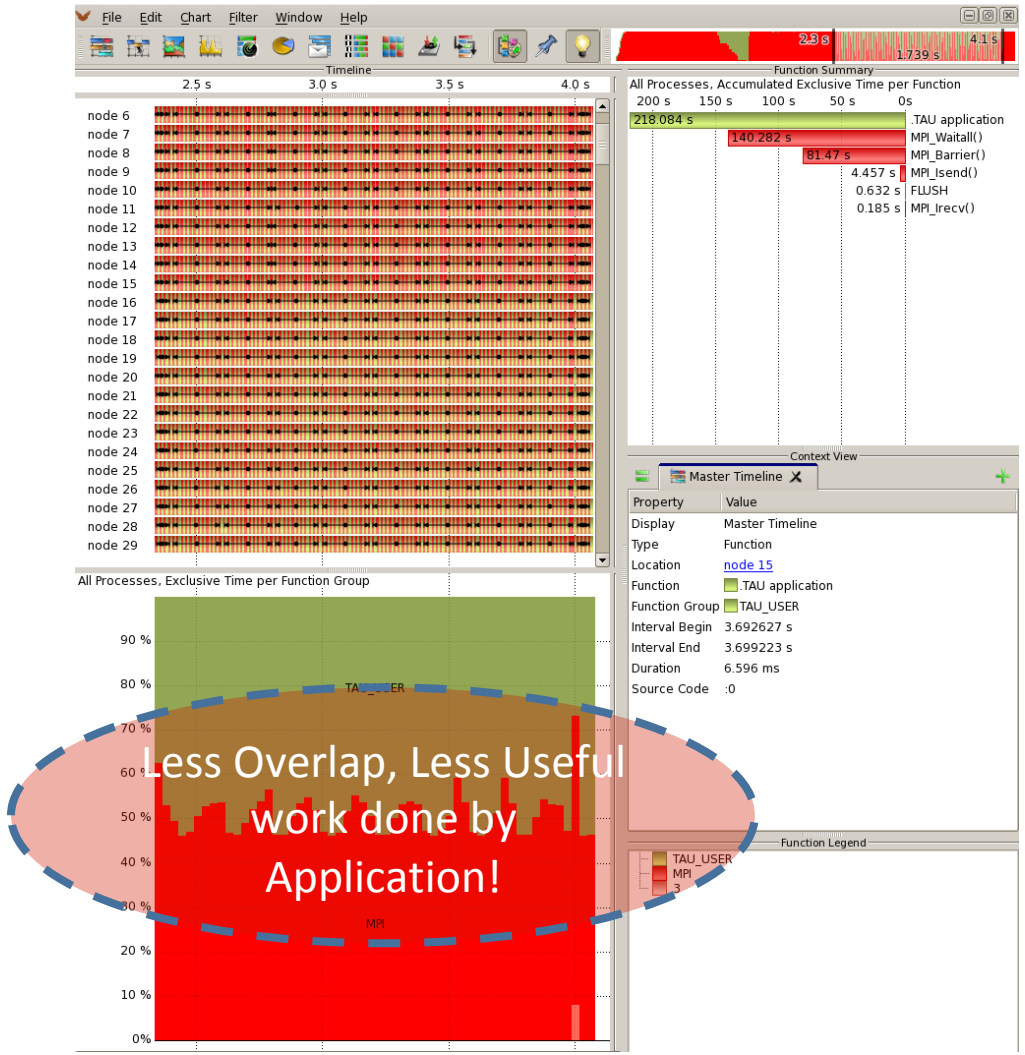
Does not execute the program. Just displays the library that it will preload if executed without the **–s** option.

NOTE: -mpi configuration is selected by default. Use –T serial for Sequential programs.

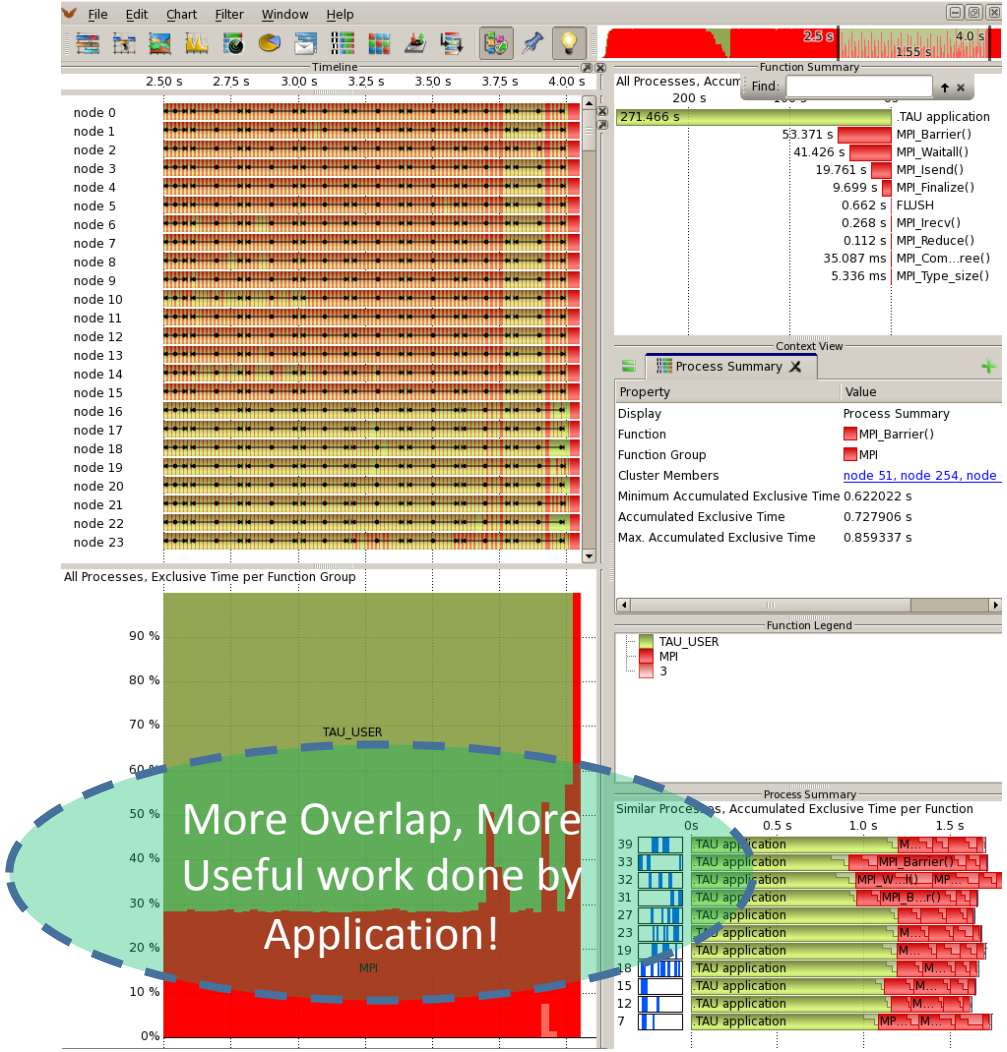
http://tau.uoregon.edu/mug18.pdf

Introspecting Impact of Eager Threshold on 3D Stencil Benchmark

Default



Optimized



http://tau.uoregon.edu/mug18.pdf

Other Runtime Environment Variables

Environment Variable	Default	Description
TAU_TRACE	0	Setting to 1 turns on tracing
TAU_CALLPATH	0	Setting to 1 turns on callpath profiling
TAU_TRACK_MEMORY_FOOTPRINT	0	Setting to 1 turns on tracking memory usage by sampling periodically the resident set size and high water mark of memory usage
TAU_SELECT_FILE		Specify the path to runtime selective instrumentation file for filtering events using exclude and include lists of routines and/or files.
TAU_CALLPATH_DEPTH	2	Specifies depth of callpath. Setting to 0 generates no callpath or routine information, setting to 1 generates flat profile and context events have just parent information (e.g., Heap Entry: foo)
TAU_SAMPLING	0	Setting to 1 enables event-based sampling.
TAU_TRACK_SIGNALS	0	Setting to 1 generate debugging callstack info when a program crashes
TAU_COMM_MATRIX	0	Setting to 1 generates communication matrix display using context events
TAU_THROTTLE	1	Setting to 0 turns off throttling. Enabled by default to remove instrumentation in lightweight routines that are called frequently
TAU_THROTTLE_NUMCALLS	100000	Specifies the number of calls before testing for throttling
TAU_THROTTLE_PERCALL	10	Specifies value in microseconds. Throttle a routine if it is called over 100000 times and takes less than 10 usec of inclusive time per call
TAU_COMPENSATE	0	Setting to 1 enables runtime compensation of instrumentation overhead
TAU_PROFILE_FORMAT	Profile	Setting to “merged” generates a single file. “snapshot” generates xml format
TAU_METRICS	TIME	Setting to a comma separated list generates other metrics. (e.g., TIME,ENERGY,PAPI_FP_INS,PAPI_NATIVE_<event>:<subevent>)

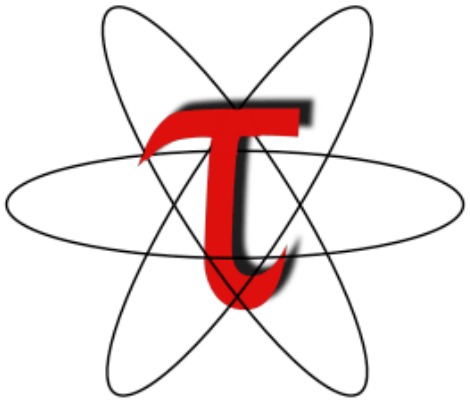
http://tau.uoregon.edu/mug18.pdf

Runtime Environment Variables (contd.)

Environment Variable	Default	Description
TAU_TRACK_MEMORY_LEAKS	0	Tracks allocates that were not de-allocated (needs –optMemDbg or tau_exec –memory)
TAU_EBS_SOURCE	TIME	Allows using PAPI hardware counters for periodic interrupts for EBS (e.g., TAU_EBS_SOURCE=PAPI_TOT_INS when TAU_SAMPLING=1)
TAU_EBS_PERIOD	100000	Specifies the overflow count for interrupts
TAU_MEMDBG_ALLOC_MIN/MAX	0	Byte size minimum and maximum subject to bounds checking (used with TAU_MEMDBG_PROTECT_*)
TAU_MEMDBG_OVERHEAD	0	Specifies the number of bytes for TAU's memory overhead for memory debugging.
TAU_MEMDBG_PROTECT_BELOW/ ABOVE	0	Setting to 1 enables tracking runtime bounds checking below or above the array bounds (requires –optMemDbg while building or tau_exec –memory)
TAU_MEMDBG_ZERO_MALLOC	0	Setting to 1 enables tracking zero byte allocations as invalid memory allocations.
TAU_MEMDBG_PROTECT_FREE	0	Setting to 1 detects invalid accesses to deallocated memory that should not be referenced until it is reallocated (requires –optMemDbg or tau_exec –memory)
TAU_MEMDBG_ATTEMPT_CONTINUE	0	Setting to 1 allows TAU to record and continue execution when a memory error occurs at runtime.
TAU_MEMDBG_FILL_GAP	Undefined	Initial value for gap bytes
TAU_MEMDBG_ALINGMENT	sizeof(int)	Byte alignment for memory allocations
TAU_EVENT_THRESHOLD	0.5	Define a threshold value (e.g., .25 is 25%) to trigger marker events for min/ max

http://tau.uoregon.edu/mug18.pdf

Download TAU from U. Oregon



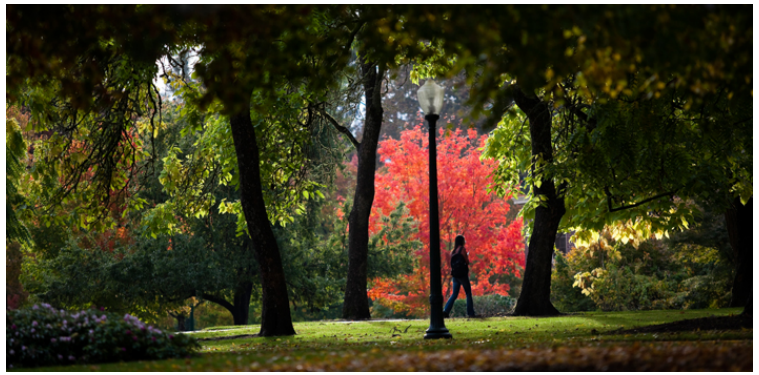
http://www.hpclinux.com [OVA file]

http://tau.uoregon.edu/tau.tgz

for more information

Free download, open source, BSD license

PRL, University of Oregon, Eugene



www.uoregon.edu

http://tau.uoregon.edu/mug18.pdf

Support Acknowledgments

US Department of Energy (DOE)

- ANL
- Office of Science contracts, ECP
- SciDAC, LBL contracts
- LLNL-LANL-SNL ASC/NNSA contract
- Battelle, PNNL and ORNL contract

CEA, France

Department of Defense (DoD)

- PETTT, HPCMP

National Science Foundation (NSF)

- SI2-SSI, Glassbox

Intel Corporation

NASA

Partners:

- University of Oregon
- The Ohio State University
- ParaTools, Inc.
- University of Tennessee, Knoxville
- T.U. Dresden, GWT
- Jülich Supercomputing Center