

SI2-SSI: Collaborative Research: A Software Infrastructure for MPI Performance Engineering: Integrating MVAPICH and TAU via the MPI Tools Interface

H. Subramoni, B. Ramesh, P. Kousha and D.K. Panda The Ohio State University http://mvapich.cse.ohio-state.edu

#### S. Shende, A. D. Malony, and S. Ramesh University of Oregon http://tau.uoregon.edu

#### **Research Challenges**

Creating an MPI programming infrastructure that can integrate performance analysis capabilities more directly, through the MPI Tools Information Interface (MPI\_T), monitor Performance metrics during run time, and deliver greater optimization opportunities for scientific applications.



#### **Proposed Approach**

The proposed tool infrastructure combines performance monitoring support through **BEACON/PYCOOLR** and a highly customizable plugin infrastructure in TAU to enable fine-grained performance analysis and control. **MVAPICH2** enables the full use of this infrastructure by exposing a vast array of **PVARs and runtime tunable CVARs** 



#### **Usage Scenarios**

#### Scenario #1 - Non-interactive mode





# Enabling customization and runtime control for MPI\_T

#### Enhanced MPI\_T support in MVAPICH2

- Added MPI\_T PVARs for various MPI collectives (Bcast, Reduce, Allreduce etc) to measure
  - bytes sent/received
  - number of invocations of the operation
- Added PVARs and CVARs for host and device based **MPI** operations
- Added MPI\_T PVAR timers to measure the time taken and the number of calls pertaining to various collective algorithms (allreduce, barrier, reduce etc)
- Added support for dynamic MPI\_T PVAR counter arrays where each index in the array represents a counter for a "bucket" or a user specified message range
- Added new CVARs that can be tuned at **run-time**
- MPIR\_CVAR\_USE\_GPUDIRECT\_RECEIVE\_LIMIT -Allows configuring GPUDIRECT receive limit in MVAPICH2 at run-time
- MPIR\_CVAR\_CUDA\_IPC\_THRESHOLD Tunes the usage of IPC communication for intra-node operations
- MPIR\_CVAR\_GPUDIRECT\_LIMIT Configures GPUDIRECT limit to tune the hybrid design that uses pipelining and GPUDirect RDMA for maximum

### **Enabling Customization**

- TAU states can be *named* or *generic*
- TAU distinguishes named states in a way that allows for separation of occurrence of a state from the action associated with it
  - Function entry for "foo" and "bar" represent distinguishable states in framework
- TAU maintains an internal map of a list of plugins associated with each state

#### **TAU Plugin Map**



## Phase-based Recommendation

- MiniAMR: Benefits from hardware offloading using SHArP hardware offload protocol supported by MVAPICH2 for MPI Allreduce operation
- Recommendation Plugin:
  - Registers callback for "Phase Exit" event
  - Monitors message size through PMPI interface
  - $\circ~$  If message size is low and execution time inside MPI Allreduce is significant, a recommendation is generated on ParaProf (TAU's GUI) for the user to set the CVAR enabling SHArP

#### Per-thread, Per-phase Recommendation Generated as Metadata on ParaProf

😣 🖻 🔍 Metadata for n,c,t 7,0,0	
Name	Value
TAU MEMDBG PROTECT BELOW	off
TAU MEMDBG PROTECT FREE	off
TAU MPI T ENABLE USER TUNING POLICY	off
TAU OPENMP RUNTIME	on
TAU OPENMP RUNTIME EVENTS	on
TAU OPENMP RUNTIME STATES	off
TAU OUTPUT CUDA CSV	off
TAU PAPI MULTIPLEXING	off
TAU PROFILE	on
TAU PROFILE FORMAT	profile
TAU RECOMMENDATION PHASE ALLOCATE	MPI T RECOMMEND SHARP USAGE: No perfomance benefit foreseen with SHArP usage
TAU RECOMMENDATION PHASE DEALLOCATE	MPLT RECOMMEND SHARP USAGE: You could see potential improvement in performance by enabling MV2 ENABLE SHARP in MVAPICH version 2.3a and above
TAU RECOMMENDATION PHASE DRIVER	MPLT RECOMMEND SHARP USAGE: You could see potential improvement in performance by enabling MV2 ENABLE SHARP in MVAPICH version 2.3a and above
TAU RECOMMENDATION PHASE INIT	MPLT RECOMMEND SHARP USAGE: No perfomance benefit foreseen with SHArP usage
TAU RECOMMENDATION PHASE PROFILE	MPLT RECOMMEND SHARP USAGE: You could see potential improvement in performance by enabling MV2 ENABLE SHARP in MVAPICH version 2.3a and above
TAU REGION ADDRESSES	off
TAU SAMPLING	off
TAU SHOW MEMORY FUNCTIONS	off
TAU SIGNALS GDB	off
TAU THROTTLE	on
TAU THROTTLE NUMCALLS	100000
TAU THROTTLE PERCALL	10
TAU TRACE	off
TAU TRACE FORMAT	tau
TAU TRACK CUDA CDP	off
TAU TRACK CUDA ENV	off
TAU TRACK CUDA INSTRUCTIONS	
TAU TRACK CUDA SASS	off
TAU TRACK HEADROOM	off
TAU TRACK HEAP	off
TAU TRACK IO PARAMS	off
TAU TRACK MEMORY FOOTPRINT	off

#### performance

### Plugin Infrastructure



- Fully-customizable plugin infrastructure based on callback event handler registration for salient states inside TAU:
  - Function Registration / Entry / Exit
  - Phase Entry / Exit
  - Atomic Event
  - **Registration / Trigger**
  - Init / Finalize Profiling
- Interrupt Handler ○ MPI\_T
- Application can define its own "trigger" states and associated plugins

# **Enabling Runtime Control**

- TAU defines a plugin API to deliver access control to the internal plugin map
- User can specify a regular expression to control plugins executed for a class of named states at runtime
  - Access to map on a process is serialized: application is expected to access map through main thread

**TAU Plugin API** 

```
int main()
```

MPI\_Init();

.... . . . . .

#### /\* Add a regular expression for MPI\_Wait\* calls \*/ TAU\_ADD\_REG\_EXPR(MPI\_Wait\*);

/\* TAU\_DISABLE\_PLUGIN (Tau\_plugin\_state s, char \* named\_state, uint plugin\_id) \*/ TAU\_DISABLE\_PLUGIN (TAU\_PLUGIN\_STATE\_FUNCTION\_ENTRY, "MPI\_Wait\*", 0); . . . . .

#### MPI Finalize(); return 0;

#### Future Work & Research Dissemination

- Add support for MPI\_T PVAR timer arrays for various collective operations to measure time taken for a set of message ranges in conjunction with TAU
- Add support to generate tuning recommendations within the MPI library using collected PVAR information in conjunction with TAU
- Use of TAU's latest plugin architecture to enable the fine-grained tuning of large-scale production applications
- Enhance TAU's support for PVARs bound to MPI objects
- Explore the use of state-of-the-art statistical analysis techniques to generate smarter performance recommendations
- Enhance TAU's support for the in-situ analysis of MPI\_T PVAR data using the latest plugin architecture

# Acknowledgment

This work was supported by the NSF under the ACI-1450440 & ACI-1450471 grants. This work used the Extreme Science and Discovery Environment (XSEDE) which is supported by National Science Foundation grant number ACI-1053575. This work used allocation grants TG-ASC090010 & TG-NCR130002.



