 Scalable Compression and Replay of Communication Traces in Massively Parallel Environments

Michael Noeth, Frank Mueller, Martin Schulz, Bronis R. de Supinski

Problem Motivation

- How can communication traces be gathered in peta-scale computing?
  - need scalable, loss-less approach
  - objective: near constant-size traces
- help understanding communication patterns
- assist in procurement
- rapid prototyping of communication needs
- Current communication analysis tools fall in 2 classes:
  1. aggregation methods 
  2. flat traces 

Our Approach

- Record Traces
  - Use MPI profiling layer
  - Compress at task level
  - Compress at node level
- Replay Traces
  - Inverse of merging algorithm

Task level compression framework

- Umpire: PMPI wrapper generator
  - Initialization wrapper
  - Tracing wrapper
  - Termination wrapper
- Task-level compression of MPI calls
  - Provides load scalability
  - Interoperable w/ cross-node framework

Cross-Node Framework Interoperability

- Single Program, Multiple Data (SPMD) nature of MPI codes
- Maintain structure of calling sequences
- stack walk signatures
- Match operations across tasks by manipulating parameters
- Source / destination offsets
- Reques handles
- Event aggregation
- Special handling of MPI_Waits

Cross-Node Compression Framework

- Invoked after application termination
- Merges operation queues produced by task-level framework
- Job-size scalability

- Reduction over binary radix tree
  - Cross-node framework merges operation queues of each task
  - Merge algorithm supports merging two queues at a time
  - Radix layout facilitates compression (constant stride b/w nodes)
  - Need a control mechanism to order merging process

Experimental Results

- Near constant size for fully compressed traces

NAS PB experiments, codes fall into 3 classes:

1. Constant size traces: EP, IS, and DT
   - Trace file size: Near constant size
   - Memory Usage: Near constant size

2. Sub-linear traces: MG, LU
   - Trace file size: sub-linear
   - Memory Usage: Sub-linear

3. Non-scalable traces: FT, BT, CG
   - Trace file size: not (yet) scalable
   - Memory Usage: Non-Scalable

NAS PB Codes – Output times

- EP, IS, DT = near constant
- MG, LU sub-linear
- FT, BT, CG = Non-scalable

Contributions and Future Work

- Scalable approach to capture full trace of communication
- Scalable replay mechanism
- Trace analysis determine inefficient MPI usage
- Assist in procurement via rapid replay
- Use to address task mapping problem