Intralayer Communication for Tree-Based Overlay Networks

PSTI 2013

Tobias Hilbrich (tobias.hilbrich@tu-dresden.de)
Joachim Protze (protze@rz.rwth-aachen.de)
Bronis R. de Supinski (bronis@llnl.gov)
Martin Schulz (schulzm@llnl.gov)
Matthias S. Müller (mueller@rz.rwth-aachen.de)
Wolfgang E. Nagel (wolfgang.nagel@tu-dresden.de)
Various Parallel Tools use TBONs

- **Performance analysis:**
  - Periscope, TAUoverMRNet

- **Application launching:**
  - LaunchMON

- **Debugging:**
  - DDT, STAT

- **Runtime correctness checking:**
  - MUST

→ Common goal: scalable communication between central tool node and increasing number of application nodes
Common Patterns on TBONs

- **Aggregation**
  - Sum up all leaf values
  - Min/max of all leaf values

- **Filtering**
  - Filter leaf values for given criterion

- Reduction of workload for root node
Irregular Patterns on TBONs

- Aggregation
  - Calculate minimal product of two leaf values

- Filtering
  - Filter leaf values depending on data value of rank 0 ($V[0]$)

- Burden of work for root node
… using Intra-Layer Communication

- **Aggregation**
  - Calculate minimal product of two leaf values

- **Filtering**
  - Filter leaf values depending on value of unit 0 ($V[0]$)

→ Load distributed to tree

Diagram:
- For Aggregation:
  - Tree structure with nodes labeled 0, 1, 2, 3, 4, 6, 2, 1, 2, 3, 12, 2.
  - Nodes marked with "*" indicating the aggregation operation.

- For Filtering:
  - Similar tree structure with nodes labeled 0, 1, 2, 3, 4, 6, 2, 1, 2, 3, 12, 2.
  - Nodes marked with "V[0]" indicating the filtering operation.

*Intralayer Communication for Tree-Based Overlay Networks*

Joachim Protze | RWTH Aachen University, Center for Computing and Communication
Our use case: P2P Matching

- Using standard binary TBON:
  - Forward send and recv event to common parent node
  - $\rightarrow \frac{1}{2}$ of matches at root node

- With intra-layer communication:
  - Forward send event to parent parent node of destination
  - $\rightarrow$ Work distributed to first tool layer
Our use case: Collective Matching

Using standard binary TBON:
→ Matching vector sizes with sizes given at collective’s root node

Matching vector sizes with sizes given at collective’s root node

½ of matches at root node

With intra-layer communication:
→ Forward vector sizes to tool nodes

Forward vector sizes to tool nodes

Work distributed to first tool layer
Implementation

- **Tool Infrastructure GTI**
  - Infrastructure forwards events to defined destinations
  - Communication currently implemented with MPI
  - Tool nodes wait for incoming messages (events) to take action
Performance Study: Synthetic P2P Matching

**System:** 1,944 nodes of Xeon 5660; 24GB/node; QDR InfiniBand

**Metric:** “Slowdown” as Runtime-with-tool/Reference-runtime

**Variables:** $p$, fan-in, and tool version with centralized manager

**Benchmark:**
- Loop of MPI_Sendrecv (…, 1 /*count*/, MPI_INT, (rank+i)%size /*dest*/, … , (rank-i)%size /*src*/, MPI_COMM_WORLD) /*i is loop counter*/

---

**Graph: Slowdown vs #Processes**

- 16 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 32 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 64 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 128 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 256 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 512 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 1024 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 2048 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
- 4096 processes: Old (Filter) fan-in 4, Intra fan-in 64, Intra fan-in 32, Intra fan-in 16, Intra fan-in 8, Intra fan-in 4
**System:** 1,944 nodes of Xeon 5660; 24GB/node; QDR InfiniBand

**Metric:** “Slowdown” as Runtime-with-tool/Reference-runtime

**Variables:** $p$ and tool version with centralized manager

**Benchmark:**
- Loop of MPI_Gatherv (..., 1 /*count*/, MPI_INT, ..., $i$%size /*root*/, MPI_COMM_WORLD) /*$i$ is the loop counter*/
Conclusions

- **Advantages:**
  - Improved scalability for distributed analyses
  - Combination of distributed and centralized analysis

- **Scalability:**
  - Performance study shows scale-independent overhead for moderate fan-in

- **Future work:**
  - Analyzing and optimizing process placement
  - Multi-hop communication for non-MPI communication
Thanks for your attention!

Questions?

For more information on MUST visit our Tutorial „Debugging MPI and Hybrid/Heterogeneous Applications at Scale” On November 17th at SC’13