Optimizing Performance on Linux Clusters Using Advanced Communication Protocols: How 10+ Teraflops Was Achieved on a 8.6 Teraflops Linpack-Rated Linux Cluster

Manojkumar Krishnan and Jarek Nieplocha
Computational Sciences and Mathematics Division
Pacific Northwest National Laboratory

Motivation

Problem
- Advancements in high-performance networks (Quadrics, Infiniband or Myrinet) continue to improve the efficiency of modern clusters.
- However, average application efficiency is a small fraction of peak system efficiency.

Primary performance degradation factors
- Processor-Network Speed Gap
- Processor-Memory Speed Gap

Approach
- Avoidance of unnecessary memory copies
- Latency hiding
Motivation: Processor-Network Speed Gap

1990

Cray X1
12.8GFLOP/s CPU
9 µs latency, 10GB/s (MPI)

Itanium-2/Myrinet cluster
6GFLOP/s CPU
6 µs latency, 550MB/s (MPI)

Today

64-CPU board
NCUBE/2
2MFLOP/s CPU
100µs latency (35 µs AM layer)
2.5MB/s B/W

Cray X1
12.8GFLOP/s CPU
9 µs latency, 10GB/s (MPI)

Improvement over 12 years
- CPU speed $10^4$
- B/W $10^{2-4}$
- Latency $10^1$

What it means for efficient communication?
- Application level
  - increase problem granularity to scale
  - avoid small messages
  - explore algorithmic opportunities for latency hiding
- Comm. system level
  - effective nonblocking communication
  - message aggregation

Motivation: Processor-Memory Speed Gap

What it implies for efficient communication?
- Avoid processor touching data unnecessarily e.g.: copying between user and special (DMA) buffers
- data packing (sparse, strided)
- format conversions

Instead, we want:
- zero-copy communication
- offload all/most data processing to the network adapter
- including noncontiguous data
- simplicity of RMA model attractive

Source: Steven S. Woo
Remote Memory Access (RMA) Model

- RMA is a programming model
  - Simple, fastest communication model than message passing
  - Closely aligned with RDMA capabilities of modern networks
  - Put/get model (e.g., ARMCI library)
    - Unlike MPI-1 RMA doesn’t require an explicit receive for every send
- MPI build on top of RMA/RDMA
  - E.g., MPI over Infiniband

Message passing 2-sided model

Remote memory access (RMA) 1-sided model

Data Movement Schemes

- These are two ends of spectrum
  - E.g., on Myrinet MPI tries to “register” user buffers with NIC on the fly
    - After handshaking between sender and receiver are zero-copy
    - NIC does handle MPI tag matching and queue management
  - RMA model is more favorable than MPI on these networks
    - Once the user registers communication buffer
    - Put/get operations handled by DMA engines on the NIC
    - No need to involve remote CPU

Copy-based, high CPU involvement e.g., IBM SP

Zero-copy, low CPU involvement e.g., Quadrics
Latency Hiding: Non-blocking Communication

- Overlapping communication with computations
  - RMA interfaces of HPC interconnects offer better overlap
    - Quadrics-Elan, Myrinet-GM, LAPI, IBA
  - “pure” non-blocking communication
    - > 90% overlap

Using Optimized Protocols

- Data Locality
  - SMP node 1
  - SMP node 2
  - network

- Shared memory
  - fastest communication protocol within SMP node

- RDMA
  - Fastest between SMP nodes
Mpp2 - IA64 Linux Cluster
Molecular Science Computing Facility
provides a high-performance computer with Intel Itanium2 processors, Quadrics interconnect, and HP RX2600 nodes, which supports a wide range of environmental molecular science.

- 1,976-processor system in 978 nodes, with 11.8 Teraflof peak.
- 6.8 Terabytes of memory
- 500 Terabytes of disk
- Quadrics QSNET-II network with Elan-4 adapter
- System allocated to users accomplishing peer-reviewed Grand Challenge science in support of DOE missions

Elan4 Communication Performance

- Asynchronous RMA Communication
  - No explicit cooperation to complete data transfers
- Avoid remote host processing
  - hardware support RDMA to read from or write to remote memory locations.
  - Smart NICs at 2 ends complete data transfers asynchronously
Example
Parallel Matrix Multiplication

- Existing parallel matrix multiplication algorithms rely on optimized broadcast and send/recv operations

- Our Algorithm - SRUMMA
  - Uses the principles discussed earlier
  - Fastest communication wherever possible
    - Shared memory within SMP
    - RMA (Uses ARMCI library)
  - Latency hiding mechanism
    - (pure) Nonblocking communication
    - Zero copy RMA (registered memory buffers)

Example
Parallel Matrix Multiplication (Cont.)

- Role of Zero-copy and non-blocking protocols
- Smart network contention algorithm using RMA

Effect of Network Contention

<table>
<thead>
<tr>
<th>Processors</th>
<th>0</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
<th>300</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregate GLOPs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>500</td>
<td>1000</td>
<td>1500</td>
<td>2000</td>
<td>2500</td>
<td>3000</td>
<td>3500</td>
</tr>
</tbody>
</table>

Using two sets of buffers to overlap communication and computation in matrix multiplication
Parallel Matrix Multiplication on the HP/Quadrics Cluster at PNNL
Matrix size: 40000x40000

Efficiency 92.9% w.r.t. serial algorithm and 88.2% w.r.t. machine peak on 1849 CPUs

Application Efficiency of 88.2% on 1849 CPUs

Asynchronous RMA
Non-blocking communication
Zero-copy protocols
Data Locality
Communication Contention

Conclusion

Remote Memory Access (RMA) protocols improves application performance

RMA approach offered more room for communication performance optimizations
- Relevant to modern clusters with high-performance interconnects with hardware-support RDMA
- E.g. asynchronous, zero-copy, more overlap, data locality

Parallel \texttt{dgemm} efficiency of 88.2% on IA64 Quadrics QsNet-II Linux cluster
- Over 10 TFlops on 1849 CPUs (Linpack 8.6 TFlops on 1960 CPUs)
- sequential \texttt{dgemm} 92.9%
- MPI based parallel \texttt{dgemm} efficiency 74%