HPC Applications Scalability

Gilad@hpcadvisorycouncil.com
Applications Best Practices

- LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator
- D. E. Shaw Research Desmond
- NWChem
- MPQC - Massively Parallel Quantum Chemistry Program
- ANSYS FLUENT and CFX
- CD-adapco STAR-CCM+
- CD-adapco STAR-CD
- MM5 - The Fifth-Generation Mesoscale Model
- NAMD
- LSTC LS-DYNA
- Schlumberger ECLIPSE
- CPMD - Car-Parrinello Molecular Dynamics
- WRF - Weather Research and Forecast Model
- Dacapo - total energy program based on density functional theory
- Lattice QCD
- Open Foam
- GAMESS
- HOMME
- OpenAtom
- PopPerf
Applications Best Practices

- Performance evaluation
- Profiling – network, I/O
- Recipes (installation, debug, command lines etc)
- MPI libraries
- Power management
- Optimizations at small scale
- Optimizations at scale
HPC Applications
Small Scale
ECLIPSE Performance Results - Interconnect

- **InfiniBand enables highest scalability**
  - Performance accelerates with cluster size

- **Performance over GigE and 10GigE is not scaling**
  - Slowdown occurs beyond 8 nodes

![Schlumberger ECLIPSE (FOURMILL)](image)

*Lower is better*  
*Single job per cluster size*
• **Input Data: T3A**
  – Resolution 36KM, grid size 112x136, 33 vertical levels
  – 81 second time-step, 3 hour forecast

• **InfiniBand DDR delivers higher performance in any cluster size**
  – Up to 46% versus GigE, 30% versus 10GigE
MPQC Performance Results - Interconnect

- **Input Dataset**
  - MP2 calculations of the uracil dimer binding energy using the aug-cc-pVTZ basis set

- **InfiniBand enables higher scalability**
  - Performance accelerates with cluster size
  - Outperforms GigE by up to 124% and 10GigE by up to 38%

*Lower is better*
NAMD Performance Results – Interconnect

- **ApoA1 case - benchmark comprises 92K atoms of lipid, protein, and water**
  - Models a bloodstream lipoprotein particle
  - One of the most used data sets for benchmarking NAMD
- **InfiniBand 20Gb/s outperforms GigE and 10GigE in every cluster size**
  - InfiniBand provides higher performance up to 79% vs GigE and 49% vs 10GigE
CPMD Performance - MPI

- Platform MPI shows better scalability over Open MPI

Lower is better

These results are based on InfiniBand
• **Platform MPI demonstrates higher performance versus MVAPIC**
  – Up to 25% higher performance
  – Platform MPI advantage increases with increased cluster size

**MM5 Benchmark Results - T3A**

*Higher is better*

**Single job on each node**
• **Platform MPI and Open MPI provides same level of performance**
  – Platform MPI has better performance for cluster size lower than 20 nodes
  – Open MPI becomes better with 24 nodes
    • Higher configurations than 24 nodes were not tested

These results are based on InfiniBand
• **Test case**
  - Single job over the entire systems
  - Input Dataset (A-Class)

• **HP-MPI has slightly better performance with CPU affinity enabled**

---

**STAR-CD Benchmark Results**

(A-Class)

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Total Elapsed Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>9000</td>
</tr>
<tr>
<td>8</td>
<td>5000</td>
</tr>
<tr>
<td>16</td>
<td>3000</td>
</tr>
<tr>
<td>20</td>
<td>2000</td>
</tr>
<tr>
<td>24</td>
<td>1000</td>
</tr>
</tbody>
</table>

*Lower is better*
CPMD – MPI Profiling

- **MPI_AlltoAll is the key collective function in CPMD**
  - Number of AlltoAll messages increases dramatically with cluster size

![MPI Function Distribution](chart.png)

- **MPI Function Distribution**
  - (C120 inp-1)

**Chart Description**
- **Y-axis**: Number of Messages (Millions)
- **X-axis**: MPI Functions
  - MPI_Allgather
  - MPI_Allreduce
  - MPI_Alltoall
  - MPI_Bcast
  - MPI_Send
  - MPI_Recv

**Legend**
- 4 Nodes
- 8 Nodes
- 16 Nodes
• Majority of MPI messages are large size
• Demonstrating the need for highest throughput
• MPI_Test, MPI_Recv, and MPI_BARRIER/Bcast show the highest communication overhead
OpenFOAM MPI Profiling – MPI Functions

- Mostly used MPI functions
  - MPI_Allreduce, MPI_Waitall, MPI_Isend, and MPI_recv
  - Number of MPI functions increases with cluster size

**MPI Profiling of OpenFOAM**
(Number of MPI messages)

<table>
<thead>
<tr>
<th>MPI Functions</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI_Allreduce</td>
<td>5.00E+06</td>
</tr>
<tr>
<td>MPI_Waitall</td>
<td>1.00E+07</td>
</tr>
<tr>
<td>MPI_Recv</td>
<td>1.50E+07</td>
</tr>
<tr>
<td>MPI_Isend</td>
<td>2.00E+07</td>
</tr>
<tr>
<td>MPI_Irecv</td>
<td>2.50E+07</td>
</tr>
<tr>
<td>MPI_Probe</td>
<td>3.00E+07</td>
</tr>
<tr>
<td>MPI_Bsend</td>
<td>3.50E+07</td>
</tr>
<tr>
<td>MPI_Send</td>
<td>4.00E+07</td>
</tr>
<tr>
<td>MPI_Send</td>
<td>4.50E+07</td>
</tr>
</tbody>
</table>

- 8 Nodes
- 16 Nodes
- 24 Nodes
ECLIPSE - Interconnect Usage

- **Total server throughput increases rapidly with cluster size**

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Data Sent</th>
<th>Timing (s)</th>
<th>Data Transferred (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Nodes</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
<tr>
<td>8 Nodes</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
<tr>
<td>16 Nodes</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
<tr>
<td>24 Nodes</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
</tbody>
</table>

This data is per node based
Most data related MPI messages are within 8KB-256KB in size
- Number of messages increases with cluster size

Shows the need for highest throughput to ensure highest system utilization
STAR-CD MPI Profiling – Data Transferred

- Most point-to-point MPI messages are within 1KB to 8KB in size
- Number of messages increases with cluster size
Most data related MPI messages are within 256B-1KB in size
Typical MPI synchronization messages are lower than 64B in size
Number of messages increases with cluster size
**ECLIPSE Performance - Productivity**

- **InfiniBand increases productivity by allowing multiple jobs to run simultaneously**
  - Providing required productivity for reservoir simulations
- **Three cases are presented**
  - Single job over the entire systems
  - Four jobs, each on two cores per CPU per server
  - Eight jobs, each on one CPU core per server
- **Eight jobs per node increases productivity by up to 142%**

![Schlumberger ECLIPSE (FOURMILL)](chart)

*Higher is better*  

*InfiniBand*
MPQC Performance - Productivity

- **InfiniBand increases productivity by allowing multiple jobs to run simultaneously**
  - Providing required productivity for MPQC computation
- **Two cases are presented**
  - Single job over the entire systems
  - Two jobs, each on four cores per server
- **Two jobs per node increases productivity by up to 35%**

**MPQC Benchmark Result**
(aug-cc-pVDZ)

![Graph showing MPQC Benchmark Result](chart)

- Higher is better

InfiniBand
LS-DYNA Performance - Productivity

**LS-DYNA - 3 Vehicle Collision**

- Jobs per Day vs. Number of Nodes for 1 Job, 2 Parallel Jobs, 4 Parallel Jobs, and 8 Parallel Jobs.
- Higher is better.

**LS-DYNA - Neon Refined Revised**

- Jobs per Day vs. Number of Nodes for 1 Job, 2 Parallel Jobs, 4 Parallel Jobs, and 8 Parallel Jobs.
- Higher is better.
• **Test cases**
  - Single job over the entire systems
  - 2 jobs, each runs on four cores per server

• **Running multiple jobs simultaneously improves FLUENT productivity**
  - Up to 90% more jobs per day for Eddy_417K
  - Up to 30% more jobs per day for Aircraft_2M
  - Up to 3% more jobs per day for Truck_14M

• **As bigger the # of elements, higher node count is required for increased productivity**
  - The CPU is the bottleneck for larger number of servers

![FLUENT 12.0 Productivity Result](chart.png)
MM5 Performance with Power Management

- **Test Scenario**
  - 24 servers, 4 processes per node, 2 processes per CPU (socket)
- **Similar performance with power management enabled or disabled**
  - Only 1.4% performance degradation

*MM5 Benchmark Results - T3A*

- **Higher is better**
- **InfiniBand DDR**
MM5 Benchmark – Power Cost Savings

- Power management saves $673/year for the 24-node cluster
- As cluster size increases, bigger saving are expected

![Power Cost Comparison Graph]

$\text{$/year} = \text{Total power consumption/year (KWh)} \times 0.20$

STAR-CD Benchmark – Power Consumption

- Power management reduces 2% of total system power consumption.

**STAR-CD Benchmark Results (AClass)**

<table>
<thead>
<tr>
<th>Power Consumption (Watt)</th>
<th>Power Management Enabled</th>
<th>Power Management Disabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>6000</td>
<td>6500</td>
<td></td>
</tr>
<tr>
<td>6500</td>
<td>7000</td>
<td></td>
</tr>
<tr>
<td>7000</td>
<td>7500</td>
<td></td>
</tr>
<tr>
<td>7500</td>
<td>8000</td>
<td></td>
</tr>
</tbody>
</table>

Lower is better

InfiniBand DDR
STAR-CD Performance with Power Management

- **Test Scenario**
  - 24 servers, 4-Cores/Proc
- **Nearly identical performance with power management enabled or disabled**

![STAR-CD Benchmark Results](chart)

**STAR-CD Benchmark Results**

(AClass)

<table>
<thead>
<tr>
<th>Total Elapsed Time (s)</th>
<th>Power Management Enabled</th>
<th>Power Management Disabled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1600</td>
<td>1600</td>
</tr>
</tbody>
</table>

*Lower is better*
MQPC Performance with CPU Frequency Scaling

- **Enabling CPU Frequency Scaling**
  - Userspace – reducing CPU frequency to 1.9GHz
  - Performance – setting for maximum performance (CPU frequency of 2.6GHz)
  - OnDemand – Maximum performance per application activity
- **Userspace increases job run time since CPU frequency is reduced**
- **Performance and OnDemand enable similar performance**
  - Due to high resource demands from the application

![MPQC Performance Graph](image)

**Benchmark Dataset**

- **Elapsed Time (s)**
  - ug-cc-pVDZ
  - ug-cc-pVTZ

**InfiniBand DDR**

**24 Node Cluster**
NWChem Benchmark Results

- Input Dataset - Siosi7
- Open MPI and HP-MPI provides similar performance and scalability
  - Intel MKL library enables better performance versus GCC

NWChem Benchmark Result (Siosi7)

Higher is better

InfiniBand QDR
NWChem Benchmark Results

- Input Dataset - H2O7
- AMD ACML provides higher performance and scalability versus the default BLAS library

![NWChem Benchmark Result](H2O7 MP2)

*Lower is better*
Tuning MPI For Performance Improvements

- **MPI Performance tuning**
  - Enabling CPU affinity
    - mca mpi_paffinity_alone 1
  - Increasing eager limit over infiniBand to 32K
    - mca btl_openib_eager_limit 32767
- **Performance increase of up to 10%**

![NAMD (ApoA1) Performance Chart](chart.png)

*Higher is better*
HPC Applications
Large Scale
Acknowledgments

• The following research was performed under the HPC Advisory Council activities
  – Participating members: AMD, Dell, Jülich, Mellanox NCAR, ParTec, Sun
  – Compute resource: HPC Advisory Council Cluster Center, Jülich Supercomputing Centre

• For more info please refer to
  – www.hpcadvisorycouncil.com
• High-Order Methods Modeling Environment (HOMME)
  - Framework for creating a high-performance scalable global atmospheric model
  - Configurable for shallow water or the dry/moist primitive equations
  - Serves as a prototype for the Community Atmospheric Model (CAM) component of the Community Climate System Model (CCSM)
  - HOMME supports execution on parallel computers using either MPI, OpenMP or a combination of MPI/OpenMP
  - Developed by the Scientific Computing Section at the National Center for Atmospheric Research (NCAR)
• **POP (Parallel Ocean Program) is an ocean circulation model**
  – Simulations of the global ocean
  – Ocean-ice coupled simulations
  – Developed at Los Alamos National Lab
• **POPperf is a modified version of POP 2.0 (Parallel Ocean Program)**
• **POPperf improves POP scalability on large processor counts**
  – Re-writing of the conjugate gradient solver to use a 1D data structure
  – The addition of a space-filling curve partitioning technique
  – Low memory binary parallel I/O functionality
• **Developed by NCAR and freely available to the community**
Objectives and Environments

- The presented research was done to provide best practices
  - HOMME and POPperf scalability and optimizations
  - Understanding communication patterns

- HPC Advisory Council system
  - Dell™ PowerEdge™ SC 1435 24-node cluster
  - Quad-Core AMD Opteron™ 2382 (“Shanghai”) CPUs
  - Mellanox® InfiniBand ConnectX HCAs and switches

- Jülich Supercomputing Centre - JuRoPA
  - Sun Blade servers with Intel Nehalem processors
  - Mellanox 40Gb/s InfiniBand HCAs and switches
  - ParTec ParaStation Cluster Operation Software and MPI
• **PSP_ONDEMAND**
  - Each MPI connection needs a certain amount of memory by default (0.5MB)
  - PSP_ONDEMAND disabled
    • MPI connections establish when application starts
    • Reduce overhead to start connection dynamically
  - PSP_ONDEMAND enabled
    • MPI connections establish per need
    • Reduce unnecessary message checking from large number of connections
    • Reduce memory footprint

• **PSP_OPENIB_SENDQ_SIZE and PSP_OPENIB_RECVQ_SIZE**
  - Define default send/receive queue size (Default is 16)
  - Changing them can reduce memory footprint
Lessons Learn for Large Scale

• Each application needs customized MPI tuning
  - Dynamic MPI connection establishment should be enabled if
    • Simultaneous connections setup is not a must
    • Some MPI function needs to check incoming messages from any source
  - Dynamic MPI connection establishment should be disabled if
    • MPI_Alltoall is used in the application
    • Number of calls to check MPI_ANY_SOURCE is small
    • Enough memory in the system

• At different scale, different parameters may be used
  - PSP_ONDEMAND shouldn’t be enabled at very small scale cluster

• Different MPI libraries has different characteristics
  - Parameters change for one MPI may not fit to other MPIs
HOMME Performance Results
(Standard.nl, ndays=12)

- **PSP_ONDEMAND=1** (no memory allocation)
- **No PSP_ONDEMAND** (with memory allocation), **PSP_OPENIB_SENDO & RECVQ_SIZE=8**
- **No PSP_ONDEMAND** (with memory allocation), **PSP_OPENIB_SENDO & RECVQ_SIZE=16**
HOMME Performance at Higher Scale...

HOMME Performance Results
(Standard.nl, ndays=12)
HOMME Scalability

HOMME Scalability Results
(Standard.nl, ndays=12)

Scalability

Number of Cores

512 1152 1975 3456 4608 6912 13824

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
POPperf Performance Results

- **No PSP_ONDEMAND** (with memory allocation)
- **No PSP_ONDEMAND** (with memory allocation), PSP_OPENIB_SENDQ & RECVQ_SIZE=8
- **PSP_ONDEMAND=1** (No memory allocation)
POPperf Scalability Results

Scalability

Number of Cores

271  541  764  1312  2009  2162  2442
Summary

• HOMME and POPperf performance depends on low latency network

• InfiniBand enables both HOMME and POPperf to scale
  – Tested over 13824 cores
  – Same scalability is expected at even higher core count

• Optimized MPI settings can dramatically improve application performance
  – Understanding application communication pattern
  – MPI parameter tuning
Acknowledgements

- Thanks to all the participated HPC Advisory members
  - Ben Mayer and John Dennis from NCAR who provided the code and benchmark instructions
  - Bastian Tweddell, Norbert Eicker, and Thomas Lippert who made the JuRoPA system available for benchmarking
  - Axel Koehler from Sun, Jens Hauke and Hugo R. Falter from ParTec who helped review the report
Thank You
HPC Advisory Council

www.hpcadvisorycouncil.com
info@hpcadvisorycouncil.com