HPC Benchmarking and Performance Evaluation with Real-Life Engineering Problem using ANSYS 12.1

AMIR KHAN
Lecturer
Institute of Space Technology, Islamabad

13-NOV-2014
Sequence of Presentation

- Introduction
- HPC Benchmarking
- Platform Description
- Softwares used
- Discussion on results
- Conclusion
Introduction

- The purpose of benchmarking and performance evaluation is to assess the performance and understand the characteristics of HPC platforms.
- The performance could be assessed using LINPACK (LINear algebra PACKage) which is used to solve many millions of linear algebraic equations or by solving a real-life problem using a commercial software/code.
- The main goal of such practice is the search for machines those are the best suitable for industrial/research projects.
- A good practice is to evaluate the performance of the computer cluster architecture, before its purchase, using the same application software which a company uses for their design.
HPC Benchmarking

- **HPC**
  High-performance computing (HPC) is the use of parallel processing for running advanced application programs efficiently, reliably and quickly

- **HPC benchmarking**
  Benchmarking is used to measure performance using a specific indicator
Types of HPC Architectures

**SHARED MEMORY PROCESSING (SMP)**
- Usually same PC with multi-core and multi-processors
- Scalable up to about 64 or 128 processors but costly
- Synchronization problem

**DISTRIBUTED MEMORY PROCESSING (MPP)**
- Distributed among physically different machines of same architecture
- Scalable up to larger no. of Processors
- Message passing overheads
# Platform Description

## Head Node (01 node)

<table>
<thead>
<tr>
<th><strong>CPU Type</strong></th>
<th><strong>AMD Opteron 6174, 12-core (Magny-cours, 2.2 GHz)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU numbers per node</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td>64</td>
</tr>
<tr>
<td><strong>Storage</strong></td>
<td>1 TB</td>
</tr>
</tbody>
</table>

## Compute Nodes (10 nodes)

<table>
<thead>
<tr>
<th><strong>CPU Type</strong></th>
<th><strong>Intel Xeon Westmere X5670 (2.93 GHz clockspeed, Hex-Core)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU numbers per node</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Total cores</strong></td>
<td>$10 \times 2 \times 6 = 120$</td>
</tr>
<tr>
<td><strong>Memory</strong></td>
<td><strong>48 GB per node</strong></td>
</tr>
<tr>
<td><strong>Hard disk</strong></td>
<td><strong>300 GB per node</strong></td>
</tr>
</tbody>
</table>
Sequence Diagram

Input file for $\Omega_1$ Calculates: $u_1$

Compute Node 1 $P_1$

Input file for $\Omega_2$ Calculates: $u_2$

Compute Node 2 $P_2$

Calculates: $K_1, K_2, f_1, f_2, K, u$

$K_1, u = f$

Head Node
# Platform Description

<table>
<thead>
<tr>
<th>Cluster Configuration</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System interconnect</strong></td>
<td>Mellanox 5200 QDR</td>
</tr>
<tr>
<td></td>
<td>Infiniband</td>
</tr>
<tr>
<td><strong>Storage system</strong></td>
<td>Panasas 12 file server</td>
</tr>
<tr>
<td></td>
<td>storage type</td>
</tr>
<tr>
<td><strong>Total RAM</strong></td>
<td>480GB</td>
</tr>
<tr>
<td><strong>Total cores</strong></td>
<td>120</td>
</tr>
<tr>
<td><strong>Cluster</strong></td>
<td>TB1.1</td>
</tr>
</tbody>
</table>
## Softwares used

<table>
<thead>
<tr>
<th>Benchmarking Software</th>
<th>ANSYS 12.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solver</td>
<td>Direct solver (dsparse)</td>
</tr>
<tr>
<td>Operating System</td>
<td>RHEL5</td>
</tr>
<tr>
<td>Job scheduler</td>
<td>PBS</td>
</tr>
<tr>
<td>Monitoring software</td>
<td>Ganglia</td>
</tr>
</tbody>
</table>
DISCUSSION ON RESULTS
Problem Description

- A static structural analysis of a square bar was carried out. The problem contains 8 millions DOFs. One end of the bar is fully constrained while the other end is subjected to the tensile loading. Number of cores varies from 24 to 120 cores.
> Time curve starts getting steady/horizontal from 120 cores shows that further increase of number of cores will not considerably decrease the computational time, rather it might increase
Computational Time save (Speedup)

No. of Cores

Percentage Time save

0.00
8.70
39.49
54.11
62.79
65.24

speedup

13/11/2014
Cluster Performance

29 % of Peak performance at 120 cores
Problem Description

- A steady state Thermal analysis problem was analyzed
- A bar with 2.4 MDOF was subjected to a constant temperature at one end as shown
- Number of cores varying from 6 to 30 cores
Elapsed Time

No. of Cores

Elapsed Time (seconds)

188 166 138 105

Time vs cores
Cluster Performance

35% of Peak performance at 30 cores
Conclusion

- Computational time save of 65.24% could be achieved while using 120 cores of TB1.1 blades comparing to 24 cores while dealing with 8MDOF problem.

- Computational time save of 44.09% could be achieved using 30 cores comparing to cores while handling a problem involving 2.4 MDOF.

- The results could be used to compare the performance of the other HPC systems blades to select the appropriate HPC system for an industry.
Thanks for your attention
Supplements

- Peak Performance of a cluster $120 \times 2.93 \times 4 = 1.4$ Tflops
- Speedup = $(t_2 - t_1)/t_1$

$$T_p = \frac{T_s}{p} + T_{Oh}(p)$$