Abstract

As the scale of the computing systems and applications is increasing, the performance of the interconnect has become key for application performance. An efficient implementation of the MPI layer is essential for realizing optimal communication performance. This work presents the performance aspects of the MPI layer implementation on the Cray XC system. Specifically, a set of guidelines are defined for the expected performance of the MPI collective algorithms. We analyze how these guidelines were adhered to and also highlight where there is scope for improvement. We expect our analysis can be used as a feedback to the MPI developers to further tune the MPI on Cray XC and as a guide to the application developers for their application performance projections and analysis.

Introduction

The performance and scaling of parallel applications on large-scale systems is highly dependent on the effective communication performance achieved by the MPI protocol. Performance guidelines for the expected behavior of MPI collectives have been defined in the following papers [1], [2] and [3]. A performance guideline usually defines a common-sense performance expectation based on semantic functionality of the collective. For example, one performance guideline states that a call to MPI_Allgather on n data elements should "not be slower than a combination of a call to MPI_Gather with n data elements followed by a call to MPI_Broadcast with n data elements.

Performance Guideline Violations on Theta

Performance guideline violations for different collectives can be noticed from the figures. For example, the performance of the Allgather is compared with the different collective combinations that essentially achieve the same functionality. Allgather has higher latency than Allbcast. This observation is true across all the node sizes tested for message sizes between 4KB and 512KB. Gather and Scatter are slower than Reduce for message sizes greater than 1MB, whereas for smaller message sizes, Reduce is faster than Gather and Scatter. Broadcast is consistently slower than Gather and Scatter, and the slopes of the curves of Broadcast and Gather/Scatter are different. There is a sudden jump in the scaling curve for Gather/Scatter at specific node sizes.

Why Consistency?

- Why performance consistency?
- Why performance consistency?
- Why performance consistency?
- Why performance consistency?
- Why performance consistency?
- Why performance consistency?

Table 1: Performance guideline violations for Allgather in Cray MPI using 256 processes (1 process per node)

| Message Size | Nodes | Allgather | Algbcast | Reduce | Algbcast
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>512 B</td>
<td>64</td>
<td>400 nsec</td>
<td>350 nsec</td>
<td>300 nsec</td>
<td>350 nsec</td>
</tr>
<tr>
<td>2048 B</td>
<td>128</td>
<td>1200 nsec</td>
<td>1000 nsec</td>
<td>800 nsec</td>
<td>1000 nsec</td>
</tr>
<tr>
<td>8192 B</td>
<td>256</td>
<td>4800 nsec</td>
<td>4000 nsec</td>
<td>3500 nsec</td>
<td>4000 nsec</td>
</tr>
<tr>
<td>16384 B</td>
<td>512</td>
<td>9600 nsec</td>
<td>8000 nsec</td>
<td>7000 nsec</td>
<td>8000 nsec</td>
</tr>
</tbody>
</table>

References