Consistent Checkpointing for High Performance Clusters

Toshihiro Nishioka
MRI Systems, Inc.
3-2-22 Harumi, Chuo-ku,
Tokyo 104-0053, JAPAN
nishioka@rwcp.or.jp

Atsushi Hori
Real World Computing Partnership
1-6-1 Takezono, Tsukuba-shi
Ibaraki 305-0032, JAPAN
hori@rwcp.or.jp

Yutaka Ishikawa
Real World Computing Partnership
1-6-1 Takezono, Tsukuba-shi
Ibaraki 305-0032, JAPAN
ishikawa@rwcp.or.jp

Abstract

This paper describes a consistent checkpointing (CCP) technique for high performance clusters. The proposed CCP method uses local disks of cluster nodes to obtain high scalability. It is evaluated using NAS parallel benchmark programs. The evaluation results show that the scalability is slightly degraded because the I/O performance of the local disks varies widely, but it is far better than the case using a centralized NFS server as a stable storage.

1. Introduction

Checkpointing is a technique to reduce damages from system crashes by saving a snapshot of computation to a stable storage system and restoring it after a system crash. On parallel computers (including clusters), checkpointing is more important since they are less reliable than single machine computers and primarily long running programs are executed on them.

Increasing the number of computation nodes is a basic approach to save the computation time. But when a centralized storage system such as NFS is used as a stable storage system, increasing the node number although increases the checkpointing overhead because a total checkpoint size usually increases. This is contradictory. By using local disks of computation nodes as a stable storage system, it is expected that a high scalability of checkpointing can be obtained.

2. Implementation

We have developed a CCP on a cluster management software, called SCore-D[1, 2], which is implemented on top of the Unix operating system. Consistency of checkpoint[3] is insured by using network preemption technique[1], which is a context switch of network, used to implement a gang scheduler of SCore-D.

3. Evaluation environment

Table 1 shows the principal specifications of our evaluation platform (hereafter, PCC). Every node has a local disk of same type, which has almost 5 MB/s I/O performance when used as Linux file system. NAS parallel benchmark programs (hereafter, NPB) were used as application programs for the evaluations.

<table>
<thead>
<tr>
<th>Table 1. PCC specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Nodes</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>Disk</td>
</tr>
<tr>
<td>Local OS</td>
</tr>
</tbody>
</table>

512 KB cache, 4000 rpm
4. Evaluation Results

Checkpointing performance is evaluated by per-node-bandwidth (hereafter, bandwidth) defined as follows:

\[ p = \frac{\sum_{i=1}^{n} s_i}{t/n}, \]  

(1)

where \( s_i \) is the data size saved at node \( i \). \( t \) is the total time elapsed in the checkpoint.

Figure 1 shows the bandwidth when checkpointing NPB programs changing the number of nodes. The programs are executed as it is checkpointed periodically. In figure 1, the X-axis shows the number of nodes and the Y-axis shows the bandwidth. Each plotted graph connects the points plotted for a program executed on a different number of nodes.

The figure 1 shows that the bandwidth scatters in the range of 1.5 ~ 3.5 MB/s and not at a constant rate. It also shows that the bandwidth decreases according to the increase of the number of nodes. Although it is more scalable compared to the NFS result (section 5), this is not an ideal result about scalability.

5. Using NFS

Table 2 shows bandwidth when using NFS, where 'size' is the average checkpointing file size, 'elapsed' is the total time elapsed in the checkpointing, and 'bandwidth' is the bandwidth per node. In this evaluation, all computation nodes of PCC are connected to an NFS server with 100 MB/s Ethernet. The result shows that the bandwidth is almost halved when the number of nodes is doubled.

<table>
<thead>
<tr>
<th>program</th>
<th>IS</th>
</tr>
</thead>
<tbody>
<tr>
<td># of nodes</td>
<td>4</td>
</tr>
<tr>
<td>size (MB)</td>
<td>24.19</td>
</tr>
<tr>
<td>elapsed (sec)</td>
<td>102.37</td>
</tr>
<tr>
<td>bandwidth (MB/s)</td>
<td>0.236</td>
</tr>
</tbody>
</table>

6. Conclusions

This paper proposed a new scheme of checkpointing which uses local disks of computation nodes as a stable storage system to obtain scalability. The evaluation results show that the scalability is slightly degraded. We think the primary reason of that the writing performance of disk drives varies widely. Nevertheless, the proposed CCP is far more scalable compared to the one using NFS as a stable storage.

References