Achieving Computational I/O Efficiency in a High Performance Cluster Using Multicore Processors

Li Ou, Xin Chen, Xubin (Ben) He
Tennessee Tech University

Christian Engelmann, Stephen L. Scott
Oak Ridge National Lab
Multicore Processors

Simple Cluster Architecture with Multicore Processors

- Core
  - L1 Cache
- Core
  - L1 Cache
- L2 Cache
- OR
- L2 Cache
- Memory
- NIC
- Switch
- Memory
- NIC
Multicore Processor in Cluster Node

- Shared L2 cache: more cache misses.
- Shared memory bus.
- Shared I/O path
Simple Parallel I/O of Multicore Processors

- A large number of I/O operations.
- Multiple separated buffers.
- Multiple noncontiguous disk accesses.
- Poor I/O performance.
Collective I/O

- Optimize for internodes coordination.
- Additional memory copy.
- In memory permutation.

Source: Rajeev Thakur, William Gropp, Ewing Lusk, “Optimizing Noncontiguous Accesses in MPI-IO”
Asymmetric Computation for Multicore processors
Asymmetric Collective I/O for Multicore Processors

- Computing cores does not commit I/O requests.
- Coordinator aggregate I/O operations from each core.
- Contiguous access from coordinator.
- Coordinator allocates one buffer: no memory copy and permutation.
### Asymmetric Collective I/O

<table>
<thead>
<tr>
<th>Computing Core</th>
<th>Coordinator core</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>char *read (file, size)</code></td>
<td></td>
</tr>
<tr>
<td>{</td>
<td></td>
</tr>
<tr>
<td>Inform coordinator;</td>
<td></td>
</tr>
<tr>
<td>Barrier;</td>
<td></td>
</tr>
<tr>
<td>Wait message from</td>
<td></td>
</tr>
<tr>
<td>coordinator;</td>
<td></td>
</tr>
<tr>
<td>Return buffer address;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

```
char *read (file, size)
{
    Barrier;
    Aggregate I/O operation;
    Allocate a contiguous buffer;
    Send I/O read;
    Assign buffer to each core;
    Wake up each core with buffer address;
    return buffer address;
}
```
Hierarchical Collective I/O

- Two level hierarchy.
- Intranode Asymmetric: among cores.
- Internode Symmetric: among coordinators.
Implementations

- Integrate into MPI I/O.
- Same interface as MPI collective IO.
- Distinguish intra-node and inter-node operations.
Node-sensitive Dataset Partitions

- **Inter-node**: each node is assigned a contiguous dataset.
- **Intra-node**: Coordinator assigns sub dataset to each core.
Simulation Methodology

- Simulate using SimpleScalar 3.0 and DiskSim 3.0.
- Modify SimpleScalar to support multicores.
- Dispatch I/O requests of SimpleScalar to DiskSim to simulate disk accesses.
Conclusions

• Asymmetric computation architecture.

• Asymmetric Collective I/O for Multicore processors within a node.

• Two level hierarchies for inter-node and intra-node collective I/O.
Future Work

• Implement asymmetric collective I/O within MPI-IO.

• Use parallel I/O benchmark (BTIO) to compare performance.

• Extend our idea to support I/O operations on RDMA.
Acknowledgements

• Laboratory Directed Research and Development Program of Oak Ridge National Laboratory.


• U.S. National Science Foundation under Grant No. CNS-0617528.
Questions and Comments?
Achieving Computational I/O Efficiency in a High Performance Cluster Using Multicore Processors

Li Ou, Xin Chen  
Xubin (Ben) He  
Tennessee Tech University

Christian Engelmann  
Stephen L. Scott  
Oak Ridge National Lab