SGI Origin

A ccNUMA Highly Scalable Server

Fall 2005, High Performance Computing

Georg Apitz
Goals

- Follow-on system to PowerChallenge
- Scale beyond 36 processors
  - *infrastructure for higher performance per processor*
- Cache-coherent memory model
- Low entry level and incremental cost
  - *compared to high-performance SMP*
SGI Origin 2000 architecture

- DSM, cc maintained via a directory-based protocol
SGI Origin 2000 nodes

- Up to 512 nodes, 1TB of Main Memory

- Up to 4GB Main Memory
- Full 4-way crossbar
- Access to Interconnect Network (6 ports)
- Access to IO subsystem
- Processors

SGI Origin 2000 nodes
Key Aspects

- Low memory latency
- Remote latency/local latency ratio low
- Hard- and software features for page migration
  - majority local
  - hardware memory reference counters
  - block copy engine
  - mechanisms to reduce cost of TLB updates
  - clean-exclusive state
Interconnect Network

- based on SGI SPIDER chip
  - 6 pairs of unidirectional links per router
  - 4 virtual channels per physical channel
  - congestion control (msg. can switch between virt. channels)
  - 256 priority levels for messages
  - programmable routing tables

32/64/128 processors ((hierarchical fat) bristled hypercubes)
CC Protocol

- Similar to Stanford DASH
  - non-blocking
  - memory requests satisfied immediately
  - request forwarding for three party transactions
- Clean-Exclusive (CEX) processor cache state
  - efficient execution of read-modify-write accesses
CC Protocol cont’d

- Upgrade requests
  - *move lines from shared to exclusive state*
    - without transferring the memory data
- backoff intervention
  - *request might be delayed but data is guaranteed*
- directory poisoning
  - *mark modified pages poisoned*
IO Subsystem

- 8 XIO ports connected to 2 nodes and 6 XIO cards
- 2 virtual channels per physical channel
- Supports allocated bandwidth of msg. from partic. devices
- CRC checking on each packet
Performance

- Microbenchmarks
  - utilizes more than half of the available memory bandwidth/node
  - small # of processors limits fetch-and-increment increase to f-and-I latency

- Applications
  - good results on NAS benchmarks
    - superlinear speedup due to larger cache size/bandwidth available

<table>
<thead>
<tr>
<th></th>
<th>M op/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 P</td>
<td>4.0</td>
</tr>
<tr>
<td>2 P</td>
<td>7.4</td>
</tr>
<tr>
<td>4 P</td>
<td>6.1</td>
</tr>
<tr>
<td>8 P</td>
<td>10.0</td>
</tr>
<tr>
<td>16 P</td>
<td>19.3</td>
</tr>
<tr>
<td>32 P</td>
<td>23.0</td>
</tr>
</tbody>
</table>

\[
\text{LL/SC} \quad \begin{array}{cccccccc}
4.9 & 2.3 & 0.84 & 0.23 & 0.12 & 0.09 \\
\end{array}
\]
Conclusions

- Tightly integrated DSM structure
  - *local accesses seen as optimization of memory reference*
- Highly scalable (1 - 512 nodes)
- Highly modular
- Bristled fat hypercube network
  - *high bi-section bandwidth, low latency interconnect*
- Low latency to local memory
- Low remote/ local memory latency ratio
Questions