AGENDA

1 NEEDS

2 JUMBO MEM
   - About the project
   - FAQ
   - Features
   - Design

3 TESTS IN XRQTC

4 QUESTIONS
**What alternatives we have for huge memory jobs?**

- Buy more memory or ultra-dense memory dimms.
- Buy a NUMA or ccNUMA machine.
- Parallelize your code (if you can).
- Use a "memory server".
Cluster global memory

- The application can swap over remote memory.
- We need an hierarchical level memory access.
- We are interested in use the available memory.
JumboMem

Feed your memory-hungry programs
About the project

- Developed by Los Alamos National Security, LLC (U.S. Department of Energy)
- Benchmarks: http://www.top500.org/system/7705
- Web: http://sourceforge.net/projects/jumbomem
- Web: http://www.c3.lanl.gov/pakin/software/jumbomem/
- Start: 2007
- Last update: 2010 (v. 2.1)
What kind of questions are in your mind?

- Is this software transparent to final user?
- What is the limitation?
- What is the performance?
- What about the scalability?
- How many nodes are you talking about?
- How many memory can it admin?
- Why JumboMem and not other software?
Scalability in Memory Capacity

One order of magnitude more memory than any existing study

- Memory Server
- GMS
- RRMP
- PGMS
- Dodo
- SAMSON
- Nswap
- HPBD
- dRamDisk
- Anemone
- JumboMem

Total memory (MB)
Scalability in Cluster Size

- JumboMem
- Memory Server
- GMS
- RRMP
- PGMS
- Dodo
- SAMSON
- Nswap
- HPBD
- dRamDisk
- Anemone

Number of nodes

One order of magnitude more nodes than any existing study
### TABLE I

**Comparison to previous work in terms of total nodes and total memory**

<table>
<thead>
<tr>
<th>Project</th>
<th>Nodes</th>
<th>Memory (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory Server [7]</td>
<td>4</td>
<td>28</td>
</tr>
<tr>
<td>GMS [8]</td>
<td>9</td>
<td>576</td>
</tr>
<tr>
<td>RRMP [9]</td>
<td>6</td>
<td>192</td>
</tr>
<tr>
<td>PGMS [10]</td>
<td>5</td>
<td>320</td>
</tr>
<tr>
<td>Dodo [3]</td>
<td>14</td>
<td>1,536</td>
</tr>
<tr>
<td>Nswap [12]</td>
<td>4</td>
<td>2,048</td>
</tr>
<tr>
<td>HPBD [13]</td>
<td>17</td>
<td>34,816</td>
</tr>
<tr>
<td>dRamDisk [14]</td>
<td>5</td>
<td>10,240</td>
</tr>
<tr>
<td>Anemone [15]</td>
<td>10</td>
<td>9,216</td>
</tr>
<tr>
<td><strong>JumboMem</strong></td>
<td><strong>256</strong></td>
<td><strong>1,048,576</strong></td>
</tr>
</tbody>
</table>
This software is special in many ways

- You don’t need to modify your code to access to whole memory of your cluster (IB island).
- You don’t have to modify your OS.
- You don’t need a virtual swap disc on remote server.
- The limits are given by the hardware (Low Latency Network).
Microbenchmark Performance

- **Altix**: 3210.7 MB/s
- **Local swap**: 1222.4 MB/s
- **JumboMem**: 1.2 MB/s
Easy to modify the computing network layer (MPI, SHMEM, ARMCI).

It allows to use several access algorithm (Pseudo NRU, true NRU, NRE, FIFO, random).

All nodes are master and slave, but not both at the same time (exclusive usage).
Restrictions

- It is not fault tolerance.
- It is no thread-safe (cannot handle fork process).
- Only works with 64-bit programs.
- The checkpointing options doesn’t work (tested with BLCR).
- It’s hard to get success with optimized binaries (MKL, PETSc, Atlas, etc.)
**Most useful options**

- `-np count, - nodes=count` Specify the number of nodes to use (min. 3)
- `-debug=level` Verbosity control
- `-pagesize=bytes` Designate a logical page size (default is 262144 - 256 KB). Applications with:
  - high degree of spatial locality (large pages).
  - low degree of spatial locality (small pages).
**Most useful options**

- `-reserve=bytes|percent%` Reserve either bytes or percent% of the available memory for non-JumboMem processes (default is 1%).

- `-slavemem=bytes` Specify explicitly the amount of memory that each slave process can serve.

- `-mastermem=bytes` Specify explicitly the amount of memory that the JumboMem master process is allowed to use.

- `-pages=count|percent%` Limit the number of logical pages that JumboMem is allowed to cache locally to either count or to percent% of the number of page mappings supported by the operating system (default is 70%).
ENVIRONMENT

Most of the options to the jumbomem script merely set environment variables that the JumboMem runtime library (libjumbomem.so) reads and processes.

- JMASYNC_EVICT (async-evict option)
- JM_DEBUG (debug option)
- JM_MASTERMEM (mastermem option)
- JMAPAGESIZE (pagesize option)
- JMSERVEMEM (reserve option)
- JM_SLAVEMEM (slavemem option)
### Summer 2011: Cluster IQTC04

- 36 nodes HP Proliant DL160 (12-cores Intel)
- 4GB*12Core*36Nodes = 1728 GB
- Infiniband x4 QDR
- SO SLES 11
<table>
<thead>
<tr>
<th>Tested applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Octave &amp; python WORK</td>
</tr>
<tr>
<td>Orca WORK</td>
</tr>
<tr>
<td>Molcas WORK</td>
</tr>
<tr>
<td>Gromacs WORK</td>
</tr>
<tr>
<td>Gaussian WORK</td>
</tr>
<tr>
<td>Molpro 2008.1 FAIL</td>
</tr>
</tbody>
</table>
Questions

Jordi Blasco (jblasco@xrqtc.com)  JumboMem
GET INVOLVED!

http://hpq.xrqtc.org

Jordi Blasco (jblasco@xrqtc.com)