Experiences with LSF and cpusets on the Origin3800 at Dresden University of Technology

Stefanie Maletti

TU Dresden – University Computer Center

email: maletti@urz.tu-dresden.de

Outline

1. HPC Resources at Dresden University of Technology and their Usage
2. Operational Concept of the Origin3800
3. Learning about LSF with cpusets
4. On the Way to the Enforcement of Our Operational Concept
5. Conclusions

1. HPC Resources at Dresden University of Technology and their Usage

SGI Origin2800 - rapunzel

since December 1996

one of the first Origin2000s shipped to Europe

48 CPUs MIPS R10000 with 195 MHz,
4 MB secondary cache
17 GB memory

Cray T3E - ratbert

since October 1998

placeholder for the new Origin3000

64 PEs (+ 2 Com, 2 OS)

128 MB memory per PE

1. HPC Resources at Dresden University of Technology and their Usage

Early SGI Origin3800 System

since August 2000
serial number L2000001 – first in Europe

- 64 CPUs MIPS R12000 with 400 MHz, 8 MB secondary cache
- 32 GB memory

SGI Origin3800 Today

128-CPU system romulus
since March 2001

- 128 CPUs MIPS R12000
- 64 GB memory

64-CPU system remus
since January 2002

- 64 CPUs MIPS R12000
- 64 GB memory

1. HPC Resources at Dresden University of Technology and their Usage

Origin2800

- Compute server for general purpose applications
- Approximately 500 users
- Applications covering technical fields such as
  - **FEM:** ABAQUS, ANSYS, MARC, NASTRAN
  - **Chemistry:** GAUSSIAN, GAMESS, MOLPRO
  - **Math. Libraries:** NAG, Maple, MATLAB, Mathematica
  - **CFD:** CFX, Fluent, Tascflow
- Development of parallel programs is encouraged by the Center of High Performance Computing
- Throughput for parallel jobs (esp. MPI jobs) is critical – turnaround time significantly varied from the time needed on a dedicated system
- Batch scheduler miser had been a way out, but sometimes miser failed to get the resources

1. HPC Resources at Dresden University of Technology and their Usage

T3E

- Parallel computer
- Only for projects – few users only
- MPI jobs and shmem programming
- Proven performance for parallel jobs

And what about the new Origin3800 systems?

128-CPU Origin3800: Parallel computer

64-CPU Origin3800: Server for large memory applications

2. Operational Concept of the Origin3800

128-CPU Origin3800 - romulus

- Parallel computer
- Only for projects
- Exclusive usage of processors and memory using LSF and cpusets
- Boot cpuset using 8 processors for the operating system and interactive tasks
- Switch between day and night/weekend operational mode by means of the checkpoint/restart facility
  - Express mode supporting application development and small production jobs during the daytime
  - Large production jobs during the night and at the weekend

2. Operational Concept of the Origin3800

Queues

- *Express:* application development and small production jobs with a maximum CPU time of 10 minutes and up to 16 processors
- *Short:* production jobs with a maximum CPU time of 5 hours and up to 64 processors per user
- *Long:* production jobs with a maximum CPU time of 5 hours and up to 120 processors per user

Our Goal

- Support of multiple parallel jobs of several users without interaction of personnel
- Turnaround time of parallel jobs should not exceed the time needed on a dedicated system by more than 10 percent
  - Realization: IRIX cpuset under control of LSF 4.1

3. Learning about LSF with cpusets

Installation and Configuration of LSF 4.1

- Initial installation and configuration of the LSF cpuset integration could not be carried out without special knowledge obtained from SGI and Platform Computing personnel
- Starting with the initial installation in January 2001, approximately 6 months elapsed before we were able to establish stable user operation

What did we observe during that time?

1. Because cpusets were not assigned exclusively, threads left their cpuset
2. Memory was not assigned exclusively
3. MPI jobs running with the MPI_DSM_MUSTRUN environment variable set, returned errors
4. Mixture of both sequential and parallel jobs led to fragmentation after some time (about 1 week), i.e. contiguous cpusets were no longer possible

3. Learning about LSF with cpusets

Configuration of LSF for cpusets

1. Selection of cpuset options

CPUSET_CPU_EXCLUSIVE
CPUSET_MEMORY_LOCAL
CPUSET_MEMORY_EXCLUSIVE

2. Setting of cpuset options

- As usual in the configuration file lsf.conf
  - Not effective
- Entering with the command line
  bsub –exsched „CPUSET_OPTIONS=CPUSET_CPU_EXCLUSIVE|
  CPUSET_MEMORY_LOCAL|CPUSET_MEMORY_EXCLUSIVE”
  - Exclusive assignment of cpusets, no threads left cpuset
  - However, the memory is still not exclusive

3. Learning about LSF with cpusets

- The described memory options caused the following problem:
- Job A requires more memory than its assigned cpuset provides
- Job A is assigned vacant CPUs
- Job B is assigned a cpuset without memory because LSF does not check the memory of free processors
- Job B (an MPI job) fails, as it is fixed to its cpuset by MPI_DSM_MUSTRUN

➢ Successor job will fail instead of the originator of the problem

3. Learning about LSF with cpusets

- Solution

- New memory options

- bsub –exsched „CPUSET_OPTIONS=CPUSET_CPU_EXCLUSIVE|CPUSET_MEMORY_LOCAL|CPUSET_MEMORY_MANDATORY|CPUSET_POLICY_KILL”
  - Only memory from nodes that are contained in the cpuset can be allocated
  - The process will be killed if more memory is needed but not available within the cpuset

Consequence:

1. User has to request as many processors as memory is required
2. Homogeneous memory configuration to work with LSF cpuset integration

3. Learning about LSF with cpusets

EXCLUSIVE
Defines a cpuset to be restricted.

MEMORY_LOCAL
Threads assigned to the cpuset will attempt to assign memory only from nodes within the cpuset. Assignment of memory from outside the cpuset will occur only if no free memory is available from within the cpuset. No restrictions are made on memory assignment to threads running outside the cpuset.

MEMORY_EXCLUSIVE
Threads not assigned to the cpuset will not use memory from within the cpuset unless no memory outside the cpuset is available. If, at the time a cpuset is created, memory is already assigned to threads that are already running, no attempt will be made to explicitly move this memory. If page migration is enabled, the pages will be migrated when the system detects that most references to the pages are non-local.

MEMORY_MANDATORY
The kernel will limit all memory allocations to nodes that are contained in this cpuset. If memory requests cannot be satisfied, the allocating process will sleep until memory is available. The process will be killed if no more memory can be allocated.

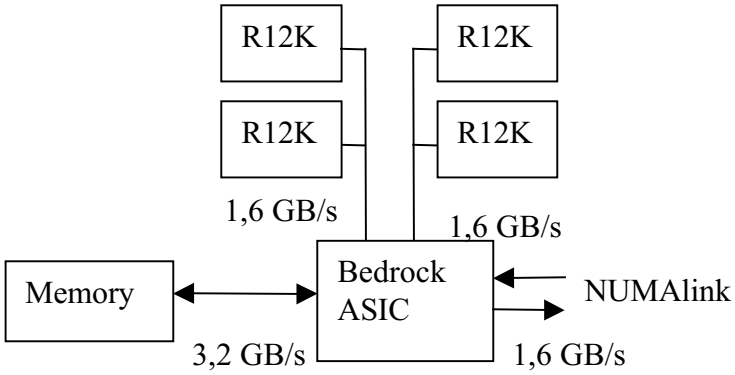
POLICY_KILL
Requires MEMORY_MANDATORY. The kernel will attempt to free as much space as possible from kernel heaps, but will not page user pages to the swap file. If all physical memory on the nodes contained in this cpuset are exhausted, the process will be killed.

4. On the Way to the Enforcement of Our Operational Concept

Exclusive usage of processors and memory works in LSF 4.1:

But take care of the memory!

- Memory is physically distributed onto nodes (C bricks)
- C brick of Origin3800: 4 CPUs share the same memory banks



➢ For sequential jobs with large memory allocation, this leads to bottle necks

4. On the Way to the Enforcement of Our Operational Concept

Difficulties with sequential jobs or jobs with a number of processors, which is not a multiple of 4

- LSF 4.1 with option MEMORY_MANDATORY:
  - Jobs with 1, 2 or 3 CPUs can assign memory from the entire C brick
  - Job is not killed until the memory from the entire C brick is exhausted

- LSF 4.2 with option MEMORY_MANDATORY:
  - For jobs with 1, 2 or 3 CPUs LSF rounds up to 4

And what about the boot cpuset?

- Boot cpuset for the operating system and interactive tasks is not exclusive
  - May cause memory problems
  - POLICY_KILL can hit the wrong job

4. On the Way to the Enforcement of Our Operational Concept

Fragmentation

- Occurs if a mixture of sequential and parallel jobs is running over a certain time
- The situation can occur that LSF can not assign contiguous cpusets and some jobs will fail

Defragmentation

- Workaround today: Checkpoint/restart
- Interaction of personnel is required to checkpoint the sequential jobs
- The released cpusets can be assigned to parallel jobs
- Restart of the sequential jobs in new cpusets

4. On the Way to the Enforcement of Our Operational Concept

Realizing the switch between day and night/weekend mode -

Queue windows versus suspend versus checkpoint/restart

- Queue window:
  - The system runs empty because a new job is not started with respect to the expected run limit exceeding its time frame
  - Setting the "IGNORE_DEADLINE" parameter, jobs are suspended when the time frame ends
- Suspend:
  - The respective resources (cpusets in LSF 4.1, memory in general) are not released
  - Contrary to our operational concept
- Checkpoint/restart:
  - The only way to switch between queues!
  - For administration purposes, checkpoint/restart is substituted by the automatic migration facility

4. On the Way to the Enforcement of Our Operational Concept

Checkpoint/restart facility plays a very important role

- Prerequisites:
  - Checkpoints have to be allowed for the queues
  - Checkpointing has to be enabled, especially for MPI jobs (mpirun script!)
  - Job has to be checkpointable
- LSF 4.1
  - Works fine
- LSF 4.2
  - Initial releases with the embedded cpuset reattach feature can not restart jobs because the assigning of cpusets fails
  - Outstanding problem until today

5. Conclusions

- Parallel applications under the control of LSF and exclusive cpusets get the best possible performance
- Switch between day and night/weekend operational mode is not currently realized – one job is critical, outstanding problems with LSF 4.2
- Memory still remains a problem for LSF (unequally distributed shared memory, hardware changes need new configuration)
  - Complete resource control with LSF or IRIX would be most desirable
- Fragmentation problems should be solved automatically

The usage of our Origin3800, which is fully accepted by our end-users, continues to grow

Workload about 70 % in spite of our restrictive operational concept

- Features such as cpusets show the right way to improve the performance of parallel jobs