Core binding

tight rein your user processes
The RDlab is part of the practices of excellence focused in promoting research and development of IT projects at the Departments of the Politechnical University of Catalonia.

The Department of Llenguatges i Sistemes Informàtics (LSI) was founded on 1986. Initially, the LSI grouped several departments related to the software area from different university schools.

In order to provide better IT support to their teachers and researchers, on 1987 the Laboratori de Càlcul del Departament de LSI (LCLSI) was created. This laboratory, initially formed by 2 people, grew in associated people - up to 12 on 2010 - as well as in skills and experience.

At the end of 2010, the LSI Department decides to make a brave and innovative bet in order to boost its research groups and the project development: the Laboratori de Recerca i Desenvolupament (RDlab) was born.
RDlab: Scope

• The Research+Development Lab is in charge of providing all the IT support to the LSI Department's research groups. We offer personal, integral and high-quality support to all teaching and research academic staff as well as their collaborators and projects.

• The RDlab must be the key factor which allows boosting the quantity and quality of the research, as well as the development of projects at the LSI Department.
RDlab: Business, Projects and Future

- Business (IT)
  - Server and services management
  - High Performance Computing

- Projects
  - Research/Educational
  - Technology Transfer
  - Self developments

- Future
  - You!

http://rdlab.lsi.upc.edu
rdlab@lsi.upc.edu
The LSI cluster

• The LSI Department cluster uses Open Grid Scheduler as resource manager
• The 90% of all the executed jobs request just 1 slot
• Ratio slots-core 1:1
• PROBLEM: Some users jobs consume more cores than the actual amount of slots requested
The LSI cluster

- The LSI Department uses Open Grid Scheduler as resource manager.
- The 90% of all the executed jobs request just 1 slot.
- PROBLEM: Some users’ jobs consume more cores than the actual amount of slots requested.

### Whole Cluster Current Information

- **Running jobs**: 13
- **Queued jobs**: 0
- **Used slots**: 13/1612

### Cluster Execution time

- **Successfull jobs**: 5610/8 jobs
- **Failure**: 20.581 jobs

### Cluster memory usage

- **Max**: 72.07 GB
- **Min**: 4.23 GB
- **Mean**: 21.34 GB

### Cluster slots-per-job usage

- **< 1 GB or less**: 84.22%
- **1 GB to 4 GB**: 15.00%
- **5 GB to 6 GB**: 0.77%
- **7 GB or more**: 0.00%

### Top Ten Node Usage

- **Usage**: 3093.5 GB

### Top Ten Users Usage

- **Usage**: 3093.5 GB

### Top Ten Average Memory Usage*

- **Usage**: 3093.5 GB

### Top Ten Average Users Execution Time*

- **Usage**: 3093.5 GB

*Approximate Calculation for page speed improvement
The LSI cluster

- The LSI Department cluster uses Open Grid Scheduler as resource manager.
- The 90% of all the executed jobs request just 1 slot.
- Ratio slots-core 1:1.
- PROBLEM: Some users' jobs consume more cores than the actual amount of slots requested.
The LSI cluster

• The LSI Department cluster uses Open Grid Scheduler as resource manager
• The 90 % of all the executed jobs request just 1 slot
• Ratio slots-core 1:1
• PROBLEM: Some users jobs consume more cores than the actual amount of slots requested
Grid Engine default behaviour

- 1 slot
- Multi-threaded
- Core Binding OFF

java

SCHEDULER

node100
node101
node102
node103
...
nodeX
...
nodeN
Grid Engine default behaviour

- nodeX
- 1 Job
- 1 Slots consumidos
- 8 cores utilizados
- 8 threads
Grid Engine default behaviour

- 1 Job
- 1 Slots consumidos
- 8 cores utilizados

nodeX

java

- 8 threads
Grid Engine default behaviour

- 1 Job
- 1 Slots consumidos
- 8 cores utilizados

nodeX

java

- 8 threads
Core binding

- Mechanism of processor affinity (hwlock) integrated within Grid Engine
- Available on all Grid Engine forks
- Allow users to target their jobs execution at core level
- Takes into account the CPU Topology

root@node111:~# loadcheck -cb
Your SGE uses hwloc for core binding functionality!
Amount of sockets: 2
Amount of cores: 12
Topology: SCCCCCCCCCCCCC
Core binding by default

• Setting a global JSV script
• The JSV shellscript is available on every Grid Engine fork
• The script checks the number of requested slots by user jobs and binds them to physical cores
• Execution hosts are configured with as many slots as real cores they have
• CPU load threshold is not needed anymore
Grid Engine & core binding

- Single-thread
- Core Binding ON
- Multi-threaded
- 8 threads
- pe make 4
- 10 threads
Grid Engine & core binding

nodeX

- .java
  - 8 threads
  - ScCCCCCSCCCCC
- .C
  - 1 thread
  - ScCCCCCSCCCCC
- .py
  - 10 threads
  - 4 slots (make)
  - SCCccccSCCCCCC
What about MPI?

• Core binding integration is only available to parallel environments with allocation rule `pe_slots`
• OpenMPI, mpich2, … have, indeed, hwlock support, but they lack tight integration with Grid Engine
Conclusions

• Every job uses 1 core by default
• Jobs can use only as many cores as slots they request
• User processes use cores exclusively
• Different user processes don’t share physical core
• User processes run always on same cores so they don’t do context switch
Demo!