Linux Control Groups
Support for
Univa Grid Engine

Daniel Gruber

dgruber@univa.com

… a Little Bit of History

Sun Grid Engine

… a Little Bit of History

Sun Grid Engine

Oracle acquired Sun

… a Little Bit of History

Sun Grid Engine

Oracle acquired Sun

Oracle Grid Engine

… a Little Bit of History

Sun Grid Engine

Oracle acquired Sun

Oracle Grid Engine

Developers moved to Univa

… a Little Bit of History

Sun Grid Engine

↓ *Oracle acquired Sun*

Oracle Grid Engine

> 3 years
ACTIVE
Development

↓ *Developers moved to Univa*

Univa Grid Engine

… a Little Bit of History

Sun Grid Engine

↓ *Oracle acquired Sun*

Oracle Grid Engine

> 3 years
ACTIVE
Development

↓ *Developers moved to Univa*

Univa Grid Engine

2013: Univa got all source code /
IP / customer support for SGE / OGE from Oracle

Many Features added ...

Job Classes

ShareTree on Job Array Tasks

Cloud Bursting

Configure PE Sorting Order

PostgreSQL Spooling

Better Job Runtime Limit Control

License Orchestrator Support

JSV Enhancements

Web GUI For Accounting: UniSight included

NUMA Aware Scheduling

Better MPI Support

RSMAP

Easier Debugging

Faster JSV Job Submission Time

Active Roadmap!

World Wide Support

Improved Scalability

Cray XC-30 Support

...and many more

HP CMU Support

GPU Support

Completely New User Guide

MacOS X Support

Per Socket Memory Management

Cache Size Reporting

Memory Affinity Settings

Intel Xeon Phi Support

Hundreds of Bug Fixes

Completely New Admin Guide

Completely New Installation Guide

Many Features added ...

Job Classes

ShareTree on Job Array Tasks

Cloud Bursting

Configure PE Sorting Order

PostgreSQL Spooling

Better Job Runtime Limit Control

License Orchestrator Support

JSV Enhancements

Web GUI For Accounting: UniSight included

NUMA Aware Scheduling

Better MPI Support

Cgroups Support in Univa Grid Engine

RSMAP

Easier Debugging

Faster JSV Job Submission Time

Active Roadmap!

World Wide Support

Improved Scalability

Cray XC-30 Support

...and many more

HP CMU Support

GPU Support

Completely New User Guide

MacOS X Support

Per Socket Memory Management

Cache Size Reporting

Memory Affinity Settings

Intel Xeon Phi Support

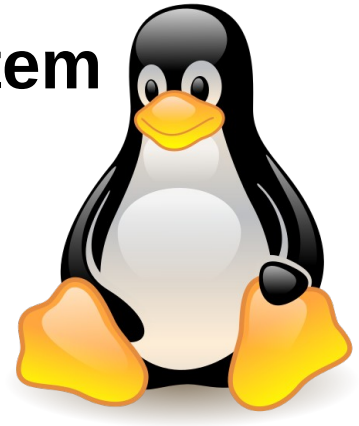
Hundreds of Bug Fixes

Completely New Admin Guide

Completely New Installation Guide

About Linux cgroups

- **Linux kernel enhancement**
- **Aggregates** / **partitions** sets of tasks into **hierarchical** groups with specialized behavior
- **Cgroup** associates a set of tasks (and all child processes) with kernel subsystems
- **Subsystem** is a Linux resource controller that schedules resources or applies limits
- **Hierarchy** is a set of cgroups arranged in a tree
- Communication / configuration through **file system interface**



Why Grid Engine Needs it

- Irrevocable **CPU isolation** and **NUMA domain isolation**
- Safer job **suspension**
- Safer job **reaping**
- More possibilities to **limit main memory** and **virtual memory** of jobs

Supported Linux Subsystems

- **Cpuset**: Assigning set of CPUs and memory nodes to a set of tasks.
  - cpuset.cpus → List of CPUs in the cpuset
  - cpuset.mems → List of memory nodes in that cpuset
- **Memory**: Isolates memory behavior of a group of tasks from the remaining system.
  - memory.limit_in_bytes → Hard memory limit for tasks
  - memory.soft_limit_in_bytes → Soft memory limit for tasks
  - memory.memsw.limit_in_bytes → Hard virtual memory limit
- **Freezer**: Starting / stopping sets of tasks without signaling them.
  - Freezer.state → Set and get the status of the tasks

UGE Features using cgroups

- Main memory limitation: **qsub -l m_mem_free**
- Virtual memory limitation: **qsub -l h_vmem**
- Automatic cpuset creation: **qsub -binding**
- NUMA domain isolation: **qsub -mbind cores:strict**
- Safe process reaping:
  - **job gets deleted (qdel / h_rt / …)**
- Process suspension without sending signals::
  - **qmod -sj**
  - **subordination**
  - **suspend_threshold**

Grid Engine Configuration

- Homogeneous cluster: Global configuration
- Heterogeneous cluster: Host local configuration

- Configuration Example: **qconf -mconf global**

```
 3 auto_user_oticket            0
 4 rlogin_command               builtin
 5 execd_params                 PTF_MIN_PRIORITY=20,PTF_MAX_PRIORITY=0, \
 6                              SET_LIB_PATH=true, KEEP_ACTIVE=ERROR
 7 rlogin_daemon                builtin
 8 cgroups_params               cgroup_path=/sys/fs/cgroup cpuset=true mount=true \
 9                              freezer=true freeze_pe_tasks=true killing=true \
10                              forced_numa=true h_vmem_limit=true \
11                              m_mem_free_hard=true m_mem_free_soft=true \
12                              min_memory_limit=200M
13 enforce_project             false
14 rsh_command                 builtin
```

Automatic cgroup Creation

- **Try to mount** cpuset/memory/freezer (**mount=true**)
- **Creation of UGE subdirectory** in cgroup subsystems of execution host, if it does not exist yet.

  $cgroup_path/cpuset/UGE
  $cgroup_path/memory/UGE
  $cgroup_path/freezer/UGE

- During job **startup**:

  Creation of a per job / array job task directory:
  $cgroup_path/<subsystem>/UGE/<jobno>.<taskid>

  Applying settings to cgroup
  Putting job (shepherd) into cgroup

- During job **shutdown**:

  Removal of all processes (killing) and destruction of cgroup

Example: cpuset Subsystem

- Configuration needs to have set **cpuset=true**
- Job needs to request core binding:
  - Directly: **-binding** submission parameter
  - Indirectly: RSMAP topology masks
- Example:

```
daniel@mint14:~$ qsub -b y -binding linear:1 sleep 12345
Your job 5 ("sleep") has been submitted
daniel@mint14:~$ cat /sys/fs/cgroup/cpuset/UGE/5.1/cpuset.cpus
0
daniel@mint14:~$ qsub -b y -binding linear:1 sleep 12345
Your job 6 ("sleep") has been submitted
daniel@mint14:~$ cat /sys/fs/cgroup/cpuset/UGE/6.1/cpuset.cpus
1
```

Main Memory Limitation

- Limiting main memory consumption
- Main memory request: **-l m_mem_free=512M**
- Different behavior:
  - **Soft limit** → Linux kernel allows to exceed if there is no memory pressure in the system
  - **Hard limit** → Out of memory for the job processes if exceeded
- Limiting virtual memory consumption: **-l h_vmem=768M**
- Prevent issues with too low memory requests: **min_memory_limit=300M**
- Limit access to NUMA node:
  - Requires: Core and memory binding
  - Configuration: **forced_numa=true**

RSMAP Resource Type

- New **Univa Grid Engine 8.1** resource type RSMAP: Resource Map
- Like a **bag of strings** (can be anything)
- Manage **co-processors** and other per host or global resources
- **Isolate** multiple jobs on one host from each other (no performance degradation due to other jobs):
  - Now with **cgroups support**
- Schedule jobs to near resources – **NUMA aware scheduling** (lower latency)
- More **flexible resource request** (decoupling from slots):
  - per HOST request possible

RSMAP Resource Type

- Resource declaration: **qconf -mc**

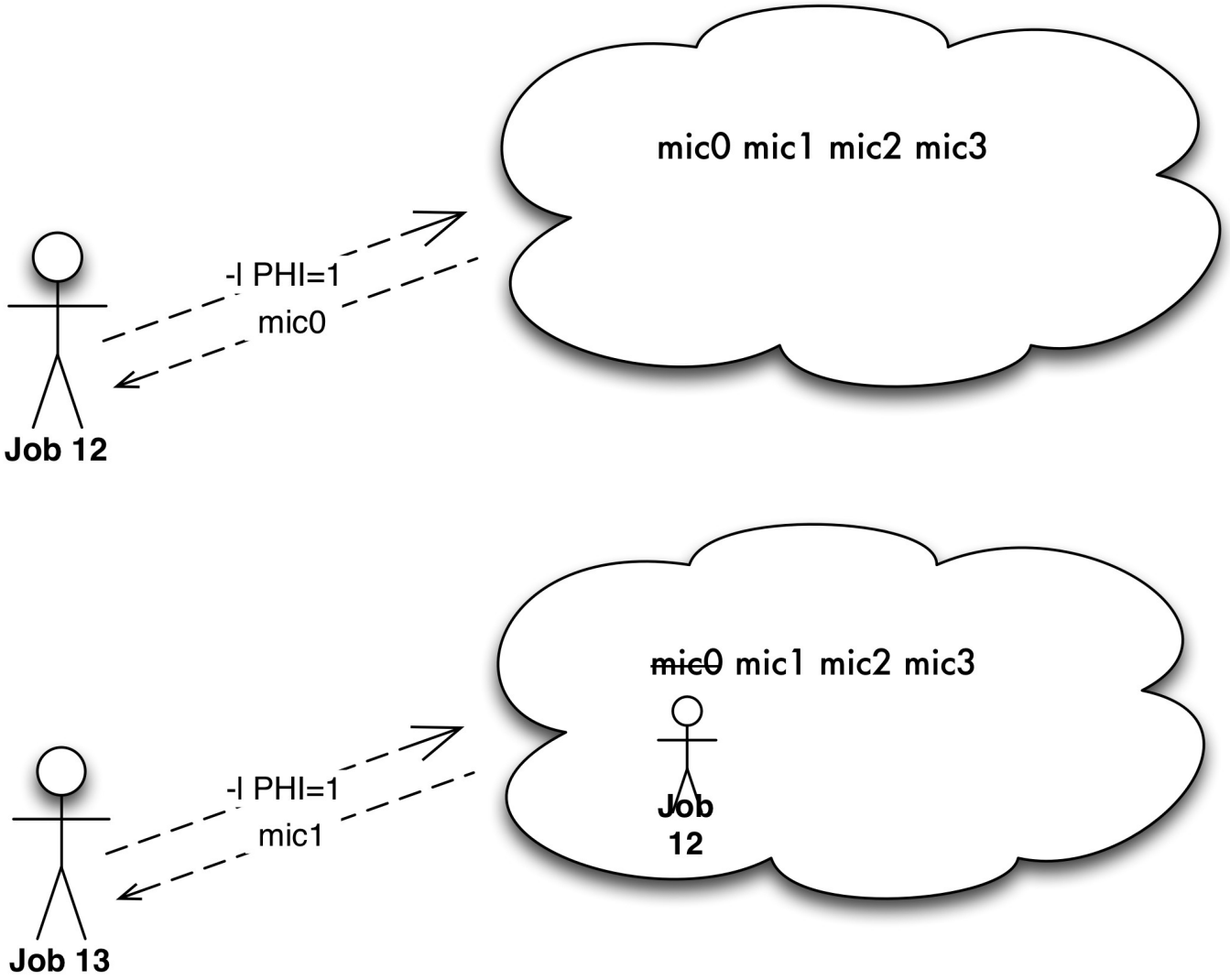
```
1 #name               shortcut   type         relop requestable consumable default  urgency
2 #-------------------------------------------------------------------------------------------
3 PHI                 PHI        RSMAP        <=    YES         HOST       0        10000
4 arch                a          RESTRING     ==    YES         NO         NONE     0
```

- Initialization of resource capacity:

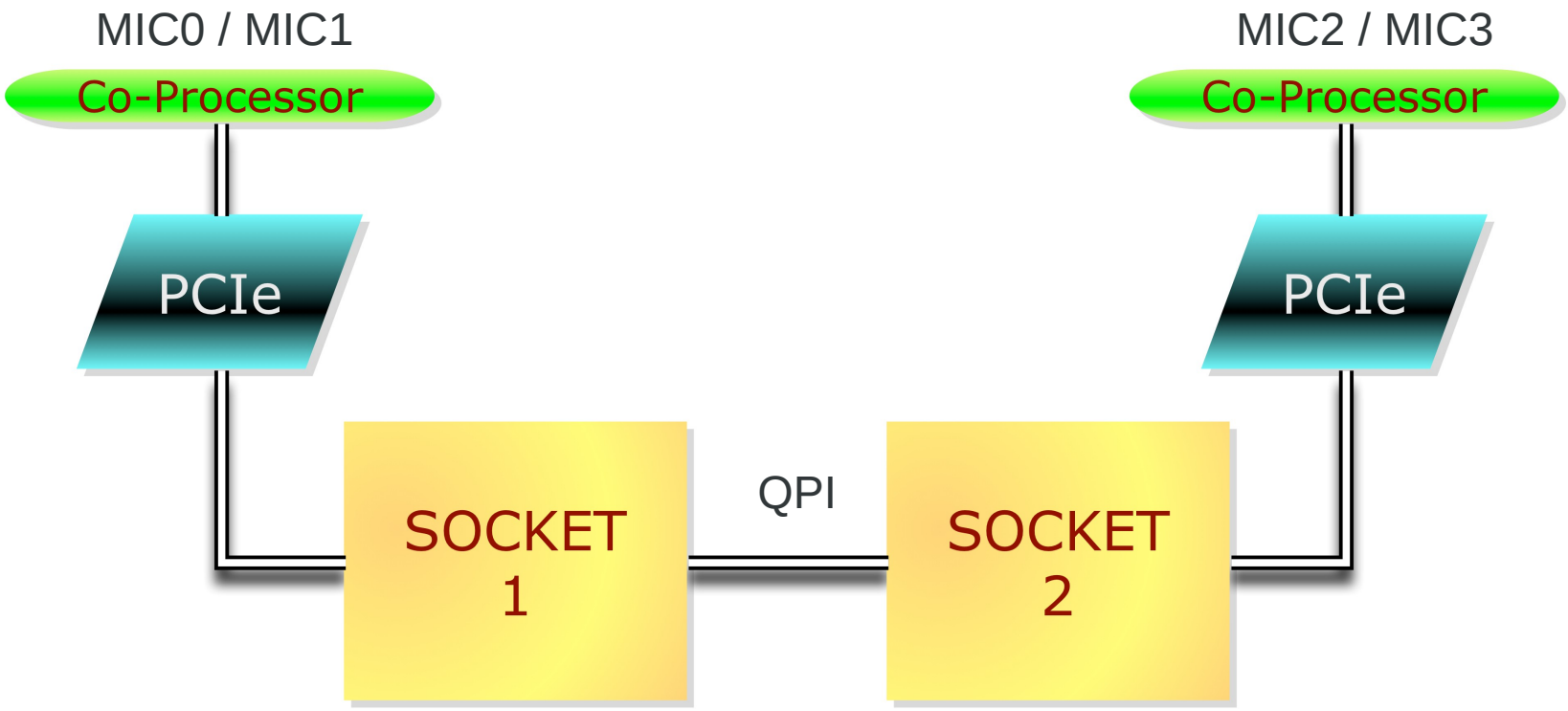
```
daniel@mint14:~$ qconf -mattr exechost complex_values "PHI=4(mic0 mic1 mic2 mic3)" mint14
Unable to find "PHI" in "complex_values" of "exechost" - Adding new element.

daniel@mint14 modified "mint14" in exechost list
```

Mapping Jobs to Resources



PCIe NUMA Device Locality



RSMAP Topology Masks

- Provides a **mapping** of a **resource** and specific compute **cores** (sockets).
- Attaches to each instance (string) a set of **allowed cores**.
- Topology string format (like m_topology): S C
- 2 Socket 4 Cores: SCCCCSCCCC
- **Excluding** cores when **mic0** is selected: mic0:S**cccc**SCCCC
- **PCIe NUMA device locality**
  - Memory and PCIe devices attached to sockets
  - Intel Xeon Phi offloading
  - Memory transfer between host and GPU
  - Faster access to discs / network / other PCIexpress resources

Example: Topology Masks

- Configuration of 4 Intel Xeon Phi cards on dual socket host.

```
-bash-4.1$ qconf -se maui
hostname              maui
load_scaling          NONE
complex_values        PHI=4(mic0:SCCCCccccScccccccc mic1:SccccCCCCScccccccc \
                      mic2:SccccccccSCCCCcccc mic3:SccccccccScccCCCC), \
```

- Based on /sys/class/mic/mic0/**local_cpulist** / **numa_node**

Example: Topology Masks

- During job submission: Specification of **how many** Intel Xeon Phis are going to be used.

```
-bash-4.1$ qsub -b y -l PHI=1 sleep 1234
Your job 7004 ("sleep") has been submitted
-bash-4.1$ qstat -j 7004 | grep mic
resource map           1:    PHI=maui=(mic1)
-bash-4.1$ qsub -b y -l PHI=1 sleep 1234
Your job 7005 ("sleep") has been submitted
-bash-4.1$ qstat -j 7005 | grep mic
resource map           1:    PHI=maui=(mic2)
```

- Check **cgroups cpuset** assignment when job is running

```
-bash-4.1$ cat /sys/class/mic/mic1/device/local_cpulist
0-7,16-23
-bash-4.1$ cat /sys/class/mic/mic2/device/local_cpulist
8-15,24-31
-bash-4.1$ cat /cgroup/cpuset/UGE/7004.1/cpuset.cpus
4-7,20-23
-bash-4.1$ cat /cgroup/cpuset/UGE/7005.1/cpuset.cpus
8-11,24-27
```

Job Reaping

- Reaping based on cgroups: **killing=true**
- Cgroup contains **tasks** file: List of all processes which belong to cgroup (job). Uses **cpuset** cgroup without any limitation.
- Loops around **tasks** file entries and kill each process until tasks file is empty
- **Safe**: Guarantees that only process IDs which belong to job (i.e. are in one cgroup) are signaled.

Summary

- Cgroups in Univa Grid Engine **isolates jobs**
- **No performance degradation** due to badly behaving jobs
- Schedules jobs **near** resources (RSMAP topology masks + memory binding)
- **Restrict main memory** usage
- Complementary safe **suspend / resume**
- Complementary safe **cleanup** of jobs processes
- Future enhancements: Based on **customer feedback**

Thank you very much for your attention!

Questions?

dgruber@univa.com