Sun Grid Engine Update

SGE Workshop 2007, Regensburg
September 10-12, 2007

Andy Schwierskott
Sun Microsystems

What is Grid Computing?

- The network is the computer™
  - Distributed resources
  - Management infrastructure
  - Targeted service or workload
- Utilization & performance ↑, costs & complexity ↓
- Examples:
  - Aggregating desktops for computation, aka cycle stealing
    - e.g. SETI@Home, use engineers' desktop at night
  - Managing an entire rack from a single interface
  - Rendering and simulation “farms”

What Sun Grid Engine does in Grid Computing

- Helps solving problems horizontally
  - High Performance [Technical] Computing
  - Data center optimization
- Examples:
  - EDA, modeling, transaction validation, MCAD
- Increasing utilization, reduce turnaround times
  - 10%-25% is typical, go up to 90%++
  - Cycle stealing
- ==> Intelligently automate batch and interactive job distribution for jobs running from seconds to days and weeks

Target Industries & Typical Workloads

Industries	Computing Tasks

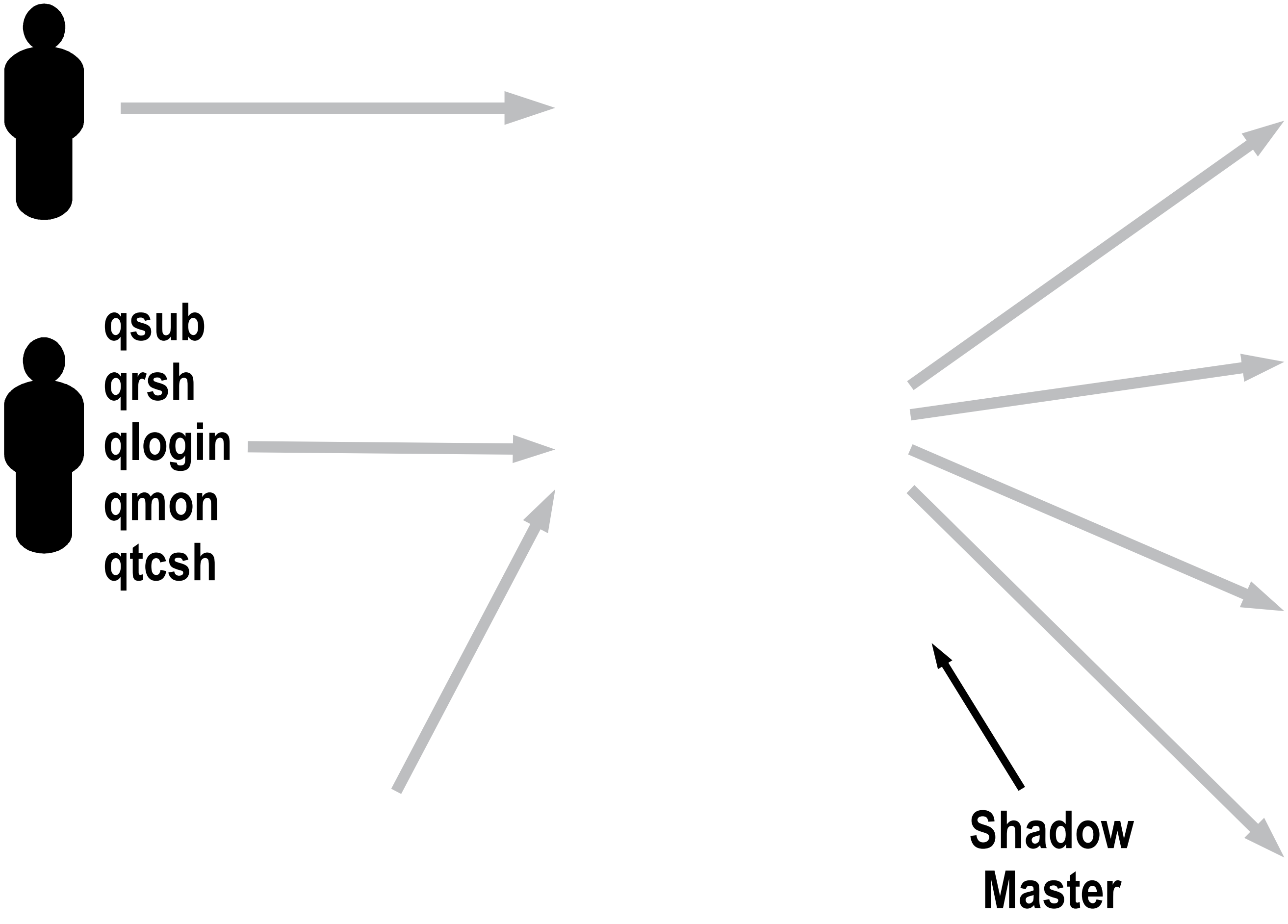
Sun Grid Engine

Resource Selection	Enterprise Allocation and Prioritization Policies Extensible Workload to Resource Matching
Resource Control	Customizable System Load and Access Regulation Definable Job Execution Contexts
Resource Accounting	Web-based Reporting and Analysis Open and Integratable Data Source

Sun Grid Engine

Ease of Administration	Hierarchical Configuration Integration with N1 Systems Management Products
3rd Party Software Integration	Standards-Compliant Full CLI Functionality
Heterogeneous Environments	Wide commercial OS support

Sun Grid Engine Components

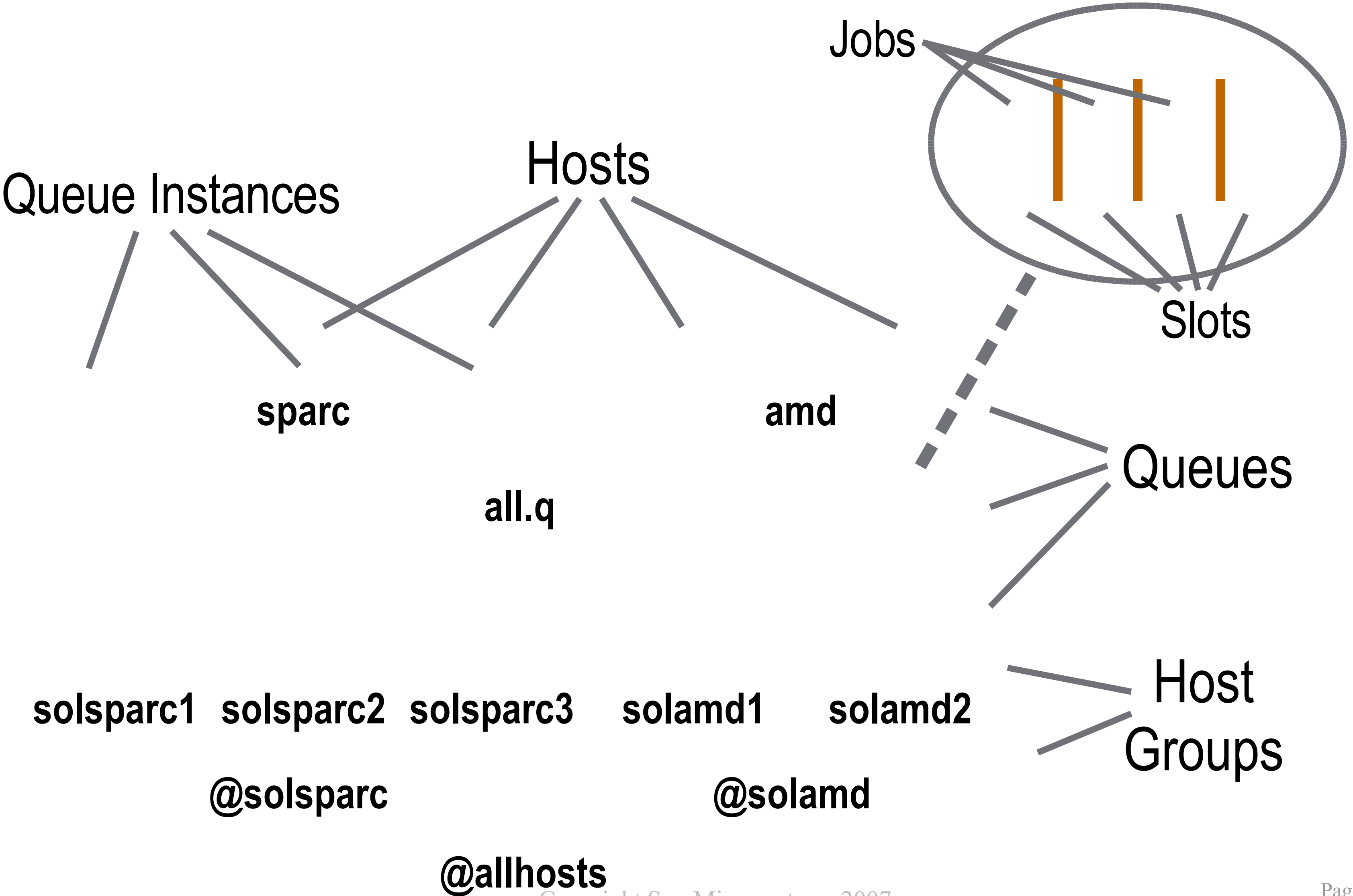


Copyright Sun Microsystems 2007

Sun Grid Engine 6

- SGE 6.0 released in 2004
  - Sites slowly adopt new functionality
  - ... and even quite a few customers still run SGE 5.3
- Powerful functionality was added to SGE 6.0
  - Cluster Queues, Host groups
  - Resource Reservation and Backfilling
  - New scheduling policies (urgency, wait time)
  - Accounting and Reporting console (ARCo)
  - Microsoft Windows Support (6.0u4)
  - Improved scalability, qstat-XML (6.0u4)
- Started significant architectural changes
  - multi-threaded qmaster, new communication library

Cluster Queues and Host Groups



Jobs

Queue Instances

Hosts

Slots

sparc

amd

Queues

all.q

Host Groups

solsparc1 solsparc2 solsparc3 solamd1 solamd2

@solsparc

@solamd

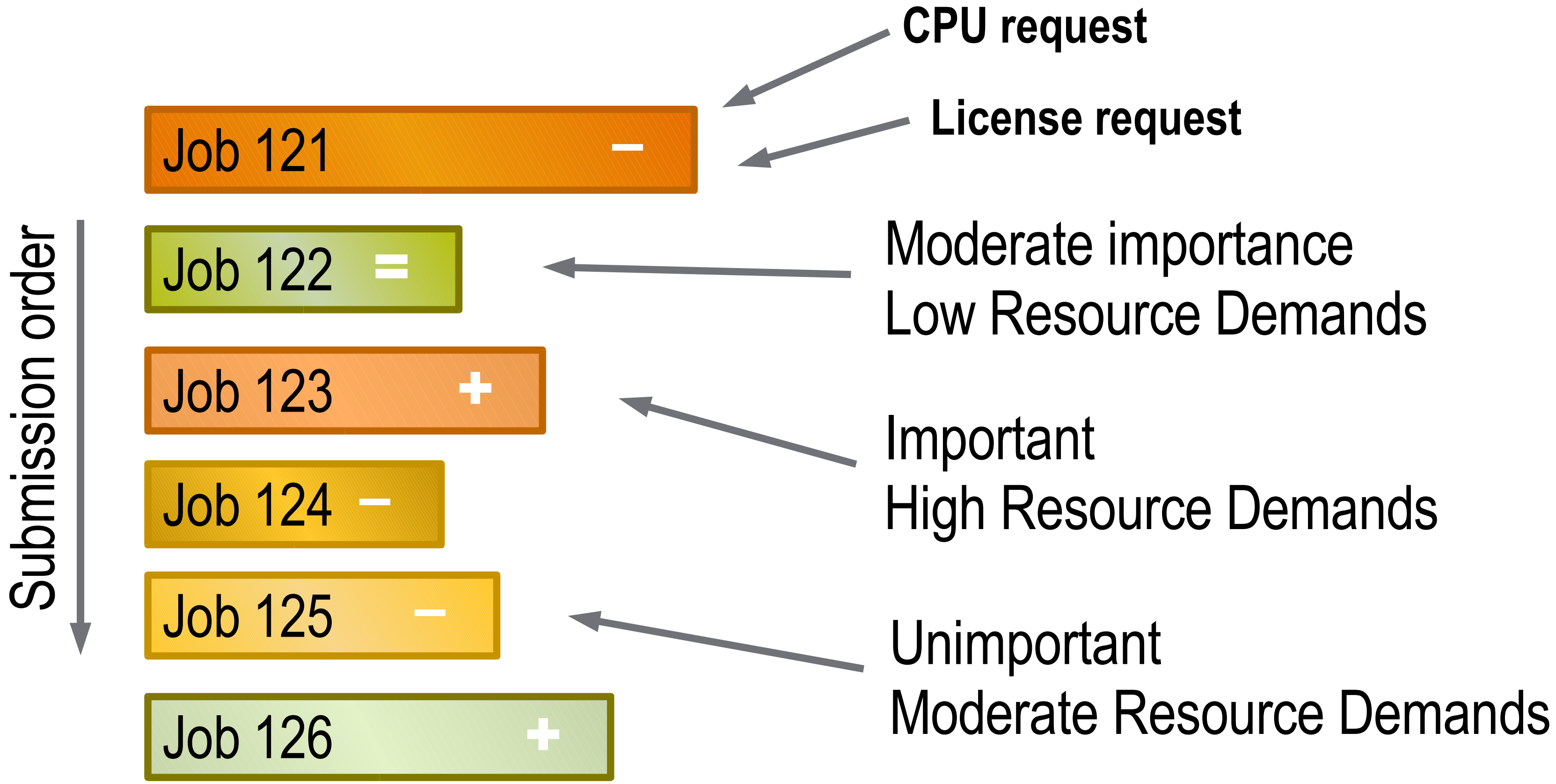
@allhosts

Resource Reservation

- Jobs may need several resources
  - Smaller jobs keep those resources busy
  - Priority inversion
- Resource Reservation
  - Allows a job to gather resources
  - Runs when all the resources are available
- Backfilling
  - Makes sure remaining resources are used
  - Fills gaps with “smaller” jobs

Resource Reservation Example

Pending Jobs List:



CPU request

License request

Job 121 −

Job 122 =

Job 123 +

Job 124 −

Job 125 −

Job 126 +

Submission order

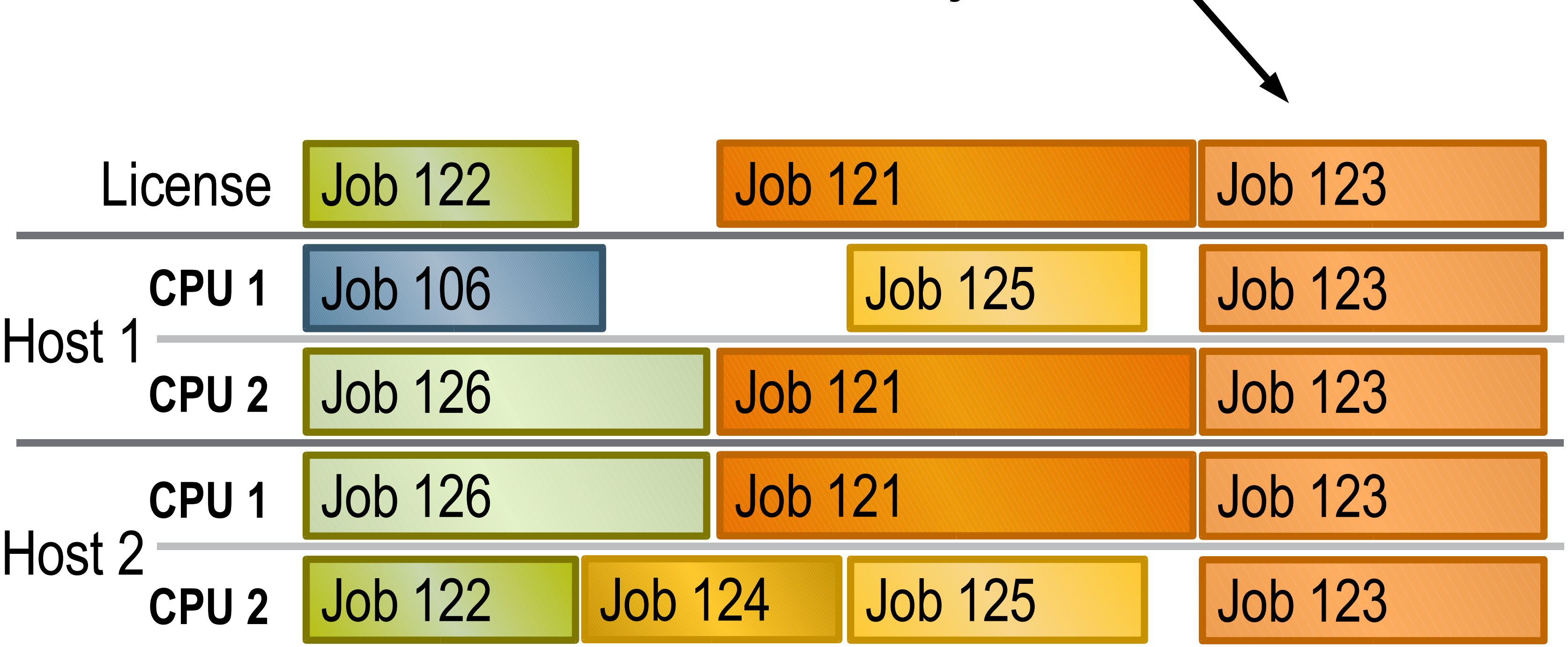
Moderate importance
Low Resource Demands

Important
High Resource Demands

Unimportant
Moderate Resource Demands

Without Resource Reservation

Highest priority job runs last!



License: Job 122, Job 121, Job 123

Host 1 CPU 1: Job 106, Job 125, Job 123

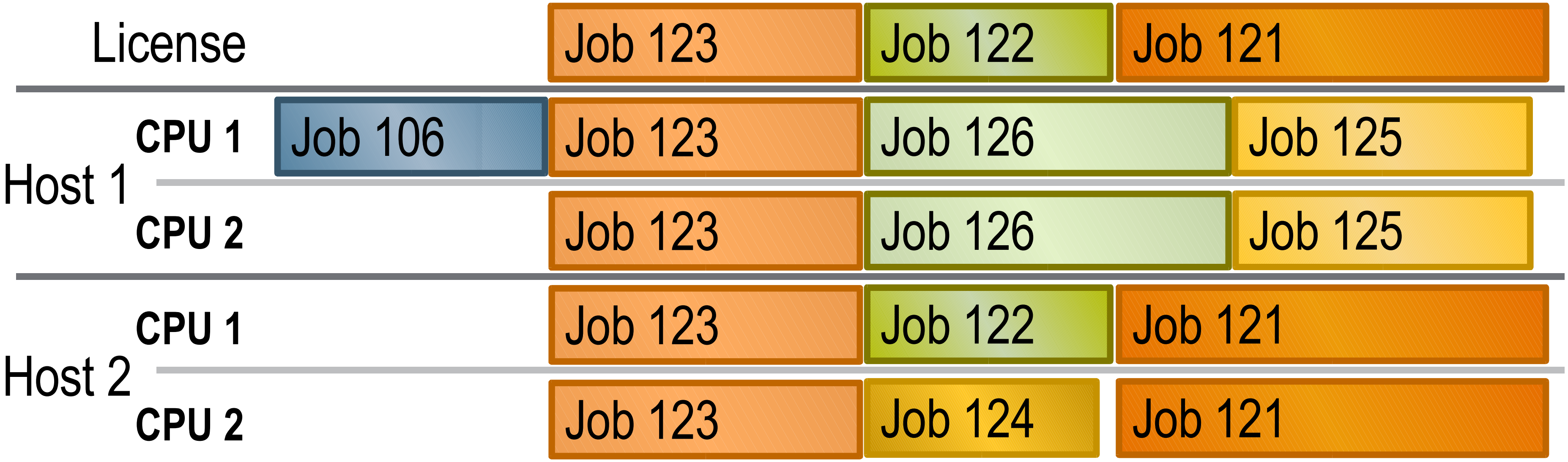
Host 1 CPU 2: Job 126, Job 121, Job 123

Host 2 CPU 1: Job 126, Job 121, Job 123

Host 2 CPU 2: Job 122, Job 124, Job 125, Job 123

With Resource Reservation

Right job order,
but less efficient!



License: Job 123, Job 122, Job 121

Host 1

CPU 1: Job 106, Job 123, Job 126, Job 125

CPU 2: Job 123, Job 126, Job 125

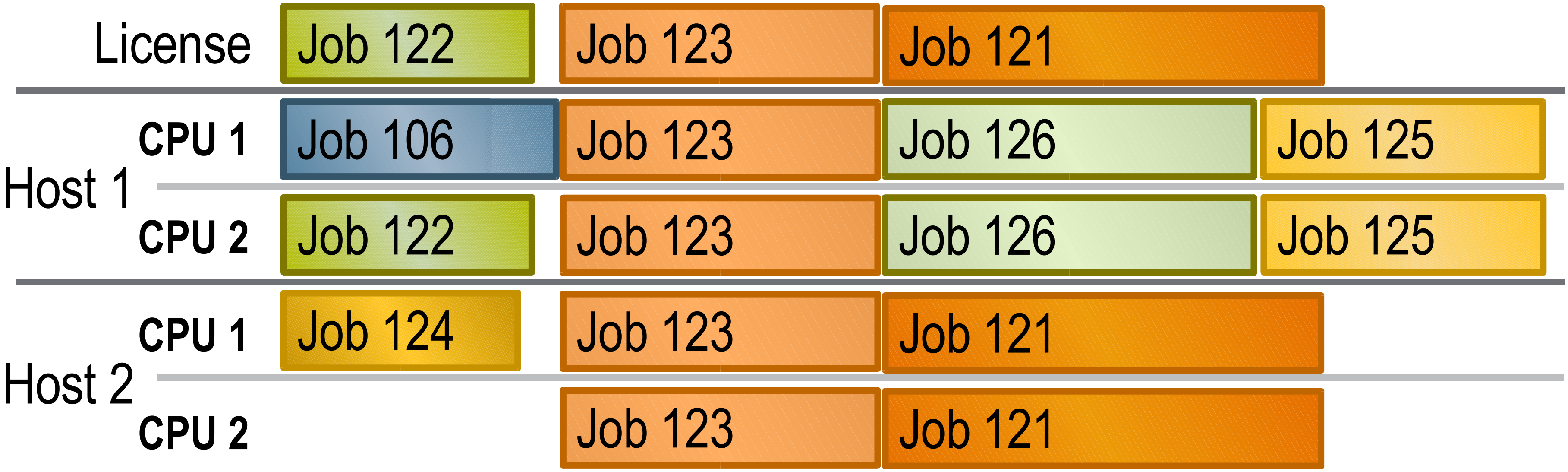
Host 2

CPU 1: Job 123, Job 122, Job 121

CPU 2: Job 123, Job 124, Job 121

Resource Reservation w/ Backfilling

Best trade-off between job order and efficiency



License: Job 122, Job 123, Job 121

Host 1

CPU 1: Job 106, Job 123, Job 126, Job 125

CPU 2: Job 122, Job 123, Job 126, Job 125

Host 2

CPU 1: Job 124, Job 123, Job 121

CPU 2: Job 123, Job 121

Entitlement Policy Components

- Hierarchical
  - Users
  - Projects
  - Arbitrary groups
- Historical
- *Fair-share*

- Categorical
  - Users
  - Departments
  - Projects
  - Jobs
- Non-historical

- Out-of-band
  - Users
  - Departments
  - Projects
  - Jobs
- Unlimited

Urgency Policy Components

- Increases as the deadline approaches

- Guarantees that a job will run eventually

- Resources can have urgencies
- Makes sure expensive resources are fully used

Combining Policies

- Each policy normalized between 0 and 1 before combining using weight factors
  - Default: $w_{psx} > w_{urg} > w_{tix}$
- Best practice: separate weights by 10x
  - e.g. 1, 10, 100

$$(w_{urg} \times n_{urg}) + (w_{tix} \times n_{tix}) + (w_{psx} \times n_{psx})$$

n_{tix} = normalized Entitlement

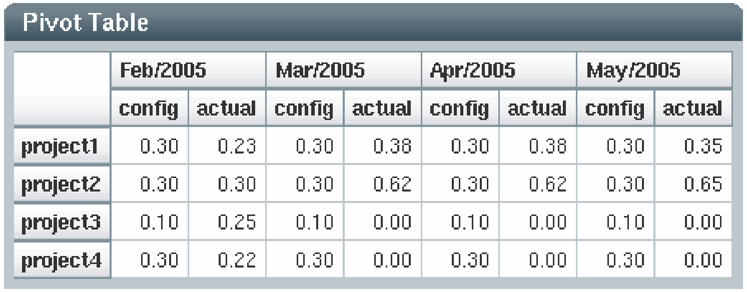
n_{urg} = normalized Urgency

n_{psx} = normalized Custom

Accounting and Reporting

- ARCo: Accounting and Reporting Console
  - Fine-grained resource accounting
    - Stored in RDBMS in well-defined schema
    - Standard SQL access for 3rd party tools
    - Customizable and extensible
  - Web-based console tool
    - Generate reports, queries, etc.
    - Customizable queries and report formats
    - Spreadsheet report generation for offline analysis

Customizable Results View



Pivot Table

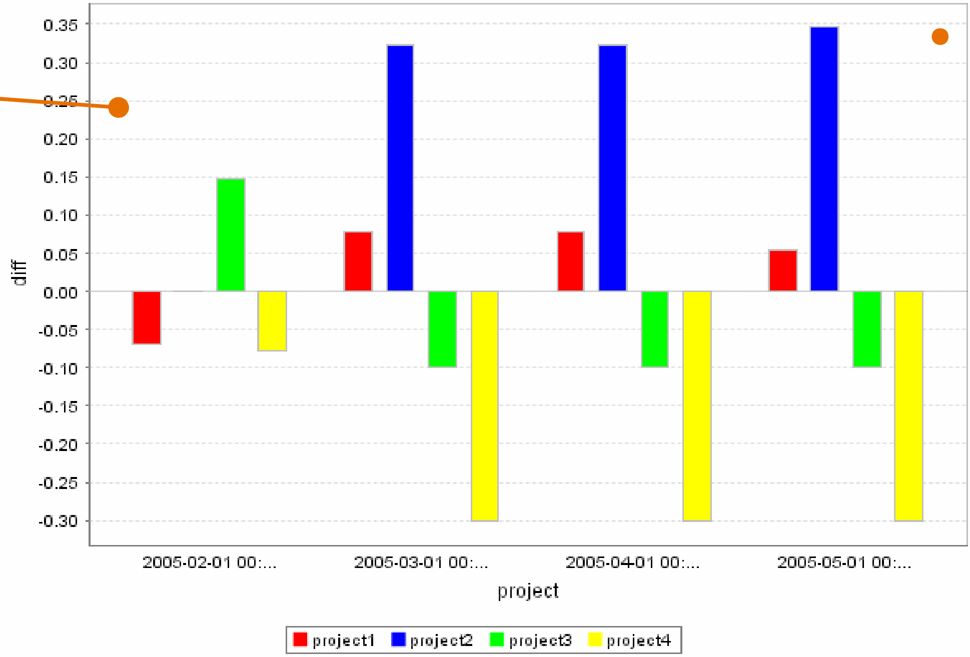
	Feb/2005		Mar/2005		Apr/2005		May/2005	
	config	actual	config	actual	config	actual	config	actual
project1	0.30	0.23	0.30	0.38	0.30	0.38	0.30	0.35
project2	0.30	0.30	0.30	0.62	0.30	0.62	0.30	0.65
project3	0.10	0.25	0.10	0.00	0.10	0.00	0.10	0.00
project4	0.30	0.22	0.30	0.00	0.30	0.00	0.30	0.00

Tables

- Simple
- Pivot
- Definable fields
- Customizable headings

Graphs

- Line Chart
- Bar Chart
- Pie Chart
- 3-D or flat



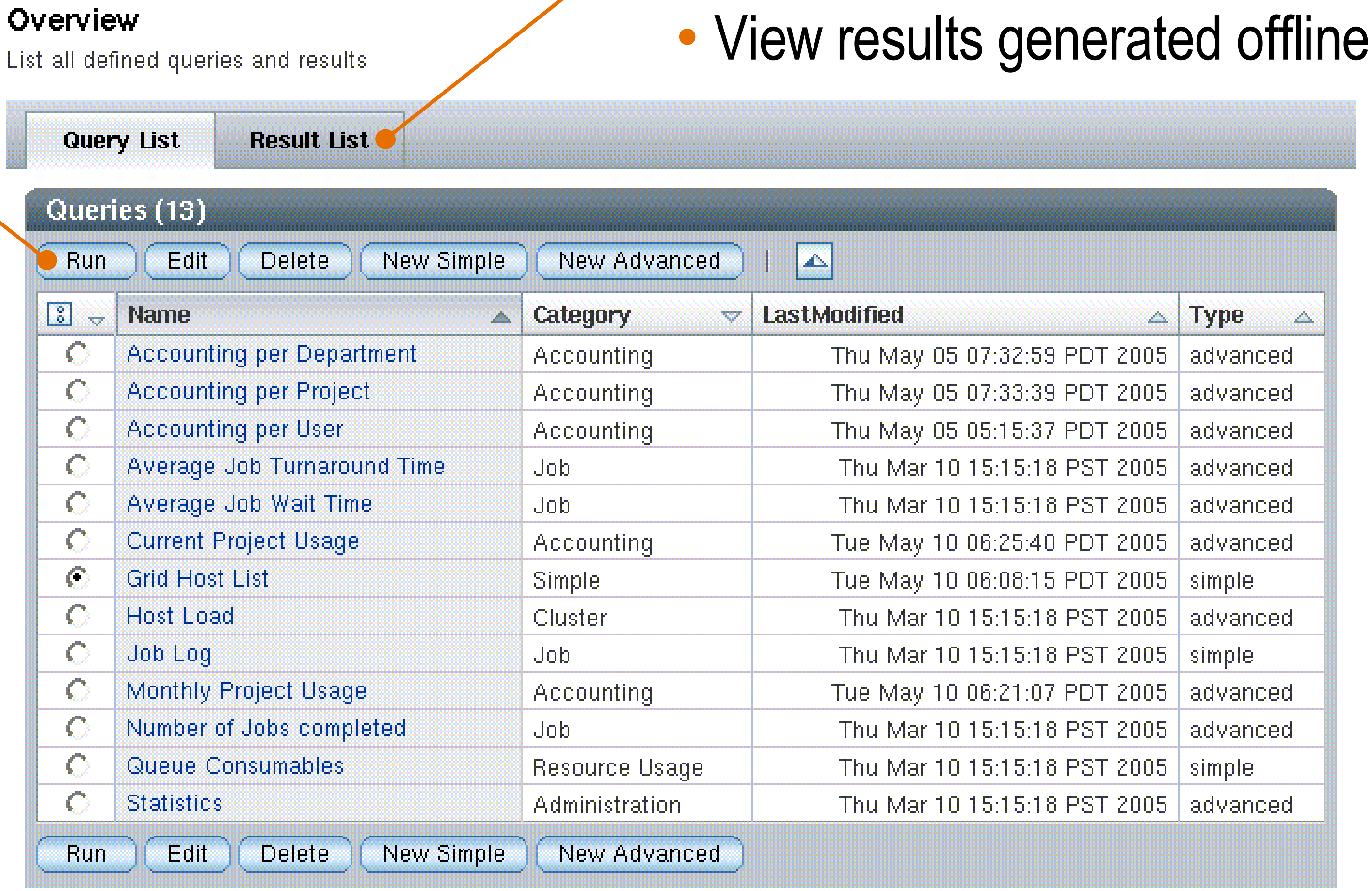
Accounting and Reporting Console

Query List

- Run by ordinary users
- Create, Edit by privileged users

Result List

- Save new results
- View results generated offline



Overview

List all defined queries and results

	Name	Category	LastModified	Type
○	Accounting per Department	Accounting	Thu May 05 07:32:59 PDT 2005	advanced
○	Accounting per Project	Accounting	Thu May 05 07:33:39 PDT 2005	advanced
○	Accounting per User	Accounting	Thu May 05 05:15:37 PDT 2005	advanced
○	Average Job Turnaround Time	Job	Thu Mar 10 15:15:18 PST 2005	advanced
○	Average Job Wait Time	Job	Thu Mar 10 15:15:18 PST 2005	advanced
○	Current Project Usage	Accounting	Tue May 10 06:25:40 PDT 2005	advanced
●	Grid Host List	Simple	Tue May 10 06:08:15 PDT 2005	simple
○	Host Load	Cluster	Thu Mar 10 15:15:18 PST 2005	advanced
○	Job Log	Job	Thu Mar 10 15:15:18 PST 2005	simple
○	Monthly Project Usage	Accounting	Tue May 10 06:21:07 PDT 2005	advanced
○	Number of Jobs completed	Job	Thu Mar 10 15:15:18 PST 2005	advanced
○	Queue Consumables	Resource Usage	Thu Mar 10 15:15:18 PST 2005	simple
○	Statistics	Administration	Thu Mar 10 15:15:18 PST 2005	advanced

DRMAA - Distributed Resource Management Application API

- Standard from the Open Grid Forum (OGF)
  - Submit, monitor, control jobs
  - Language & platform agnostic
- ISV's
  - “Grid-enable” their applications
  - Avoid DRM/Grid system lock-in
- In-house developers
  - Integrate Grid tasks into workflow, orchestration, online apps, etc.

DRMAA

- http://www.drmaa.org/
- Working group goals
  - Easy to use
  - Universally implementable
- Sun Grid Engine Bindings
  - C binding – supported
  - Java binding – supported
  - Perl binding – not supported by Sun
  - Python binding – not supported by Sun
  - Ruby binding – not supported by Sun

DRMAA Command-line Parity

To the qmaster

DRMAA Application Portability

- Stick to DRMAA specification
  - Be careful with native specification
    - Use job category instead
- DRMS/DRMAA info routines
- Adoption is growing
  - Sun Grid Engine
  - Condor
  - Gridway
  - Torque
  - UNICORE
  - EGEE

Further functionality added with SGE 6

- Microsoft Windows Support (6.0u4)
  - Windows 2000, Windows Server 2003, XP Pro
- Greatly improved scalability
  - Reduce job turnaround times
  - Handle more jobs, bigger clusters
  - Reduce memory footprint of master host daemons
- Started significant architectural changes
  - multi-threaded qmaster, new communication library

Security

- System can be installed with CSP (Certificate Security Protocol) enabled
  - Based on OpenSSL library
  - Client and daemons are authenticated to each other
  - Communication is encrypted
- ssh can be configured for “qrsh” command and for startup of parallel jobs

Sun Grid Engine 6.1

- SGE 6.1 released May 8, 2007
  - Free download from http://sun.com/gridware
  - Continued courtesy binary availability through open source project
  - Current patch level SGE 6.1u2
- Resource Quotas (RQS) – major new feature

Supported Platforms with SGE 6.1

	Master Host	Compute Host
solaris	Solaris 8, 9, 10 on SPARC Solaris 9, 10 on x86 Solaris 10 on x64	Solaris 8, 9, 10 on SPARC Solaris 9, 10 on x86 Solaris 10 on x64
Linux	Linux kernel 2.4-2.6 on x86/x64 (virtually any distribution, glibc >= 2.3.2)	Linux kernel 2.4-2.6 on x86/x64/IA64 (any distribution,glibc >= 2.3.2)
Microsoft Windows		Windows 2000/XP Pro, 2000/2003 Server
		Mac OS X 10.4 on PPC+x86
AIX L		AIX 5.1, 5.3
hp invent		HP-UX 11.xx
sgi		Irix 6.5

Dropped OS support in SGE 6.1

- Solaris 7 (Sparc), all Sparc 32-bit (“sol-sparc”)
- Solaris 8 (x86)
- Linux distributions with glibc version < 2.3.2, e.g.
  - > RH Linux 7.2, some very early RH 8.0
  - > RHEL 2.1
  - > => we provide Linux x86+x64 “unsupported” courtesy binaries through open source project
  - > => offer official support for a limited time for Linux, possibly Solaris – need setup special contract
- Apple Mac OS X 10.2+10.3 on PPC
- IBM AIX 4.3

Linux – a special support challenge

- Broad variety of distributions
  - RedHat, Suse, Ubuntu, Debian, Knoppix, JDS
  - Incompatibilities/weirdnesses:
    - e.g. Suse Linux 9.3 comes with different library levels than Suse Enterprise Linux 9.3
  - It's not just a glibc version issue
    - Startup script specialties between vendors and releases
    - Many small fixes have been done over the years
  - Motif library (qmon only)
    - Need libXm.so.3 from openmotif-2.2.3 RPM package or higher
  - No issue: the Linux threading library: “old” threading library vs. the newer “NTPL” library. No known issues with SGE though the old lib has some known bugs

New in Grid Engine 6.1

Resource Quotas

- Ability to implement the following kinds of rules:
  - > “Limit all users except Bob to run 10 jobs on queue X”
  - > “Every user is restricted to 2GB memory per Linux host, except Bob is restricted to 4GB memory per Linux host”
- Limits defined by
  - > Users/usergroups, projects
  - > Parallel environments, hosts/hostgroups, queues
  - > Resource attributes = max value
    - > Job slots, licenses, memory, etc.
- Firewall-style configuration

Resource Quota Rules

- Expressed using rules within a *rule set*
  - Group of rules, evaluated in order
  - Only the first applicable rule is used
- Example: “all users restricted to 15 slots in all.q, except user bob is restricted to 10 slots”

```
{
 name rule_set_1 →
 description Example rule set #1
 enabled TRUE
 limit users bob queues all.q to slots=10
 limit users * queues all.q to slots=15
}
```

Resource Quota Rule Sets

- All rule sets are evaluated – order does not matter
- The most restrictive is used
- Example:

`rule_set_1` ⇒ limit user “bob” to 5 slots

`rule_set_2` ⇒ limit user “bob” to ∞ slots (i.e. no rule)

`rule_set_3` ⇒ limit user “bob” to 3 slots

---

limit user “bob” to 3 slots

Resource Quotas

- Flexible limit definitions
  - Wildcards and logical NOTs
    - Users *, !bob
  - Group-wide and per-member
  - Static
    - slots=10
  - Dynamic (only on host level in 6.0)
    - slots=$num_proc * 2
  - Weighted Sum
    - slots=$num_proc * 2 - 1

Copyright Sun Microsystems 2007

Use case: some users limited to 10 slots per host

```
# qconf -srqs 10_slots_per_host
{
  name          10_slots_per_host
  description   limit a few users to 10 slots per host
  enabled       TRUE
  limit         users {A,B,C,D} hosts {*} to slots=10
}                         ^---- {} each of these users is limited to 10 slots per host
```

```
# qquota -u \*
resource quota rule   limit              filter
--------------------------------------------------------------------------
10_slots_per_host/1 slots=1/10          users D hosts bilbo
10_slots_per_host/1 slots=2/10          users D hosts lis
10_slots_per_host/1 slots=1/10          users D hosts brag
10_slots_per_host/1 slots=1/10          users D hosts carc
10_slots_per_host/1 slots=1/10          users D hosts nori
10_slots_per_host/1 slots=1/10          users D hosts angbor
10_slots_per_host/1 slots=1/10          users D hosts es-ergb01-01
```

Use case: Limit license use per project

```
# qconf -srqs F_lics_limit
{
   name           F_lics_limit
   description    Limit the use of the F00* licenses to one per project
   enabled        TRUE
   limit          projects {*} to F001=1,F002=1,F003=1
}                          ^----- {} expresses “per”
```

```
# qconf -se global | grep complex_values
complex_values      F001=100,F002=100,F003=100
```

```
# qconf -sc |egrep “^#|F00”
#name  shortcut   type      relop requestable consumable default  urgency
#------------------------------------------------------------------------
F001     F001       INT       <=    YES          YES          0          0
…
```

Use case: Limit license use for some projects to an upper limit

```
# qconf -srqs F_lics_limit
{
   name              F_lics_limit
   description       Limit the use of the F00* licenses to one for given projects
   enabled           TRUE
   limit             projects p1,p2,p3  to F001=1,F002=1,F003=1
}                             ^----- projects p1,p2,p3 together may not use more ...

# qconf -se global | grep complex_values
complex_values       F001=100,F002=100,F003=100

# qconf -sc |egrep “^#|F00”
#name  shortcut   type       relop requestable consumable default  urgency
#------------------------------------------------------------------------------
F001     F001       INT        <=    YES          YES          0         0
...
```

More Resource Quota Rules

-
  ```
  limit users * hosts * to license1=10
  ```
  - Global limit of 10 uses of license1

-
  ```
  limit users {*} hosts * to \
  license1=10
  ```
  - Each user has a global limit of 10 uses of license1

-
  ```
  limit users * hosts {*} to \
  license1=10
  ```
  - Global limit of 10 uses of license1 on each host

-
  ```
  limit users {*} hosts {*} to \
  license1=10
  ```
  - Each user is limited to 10 uses of license1 on each host

Boolean Expressions for String, Host and Queue Resource Requests

- AND (“&”), OR (“|”), and NOT (“!)
- Parenthesis “(“ and “)” are supported
- Examples – no blanks allowed
  - -l arch='sol-x86|sol-amd64'
    - Solaris x86 or Solaris AMD64
    - (Works with N1GE 6.0)
  - -l arch='sol-*&!sol-sparc'
    - Solaris except SPARC 32 bit
  - -l arch='!lx*&!*x86*'
    - Not Linux and not arch containing “x86”

Use cases: Boolean Expressions

- Works for “qsub -q” switch as well
  - qsub -q “big|medium@@hgrp[12]”
  - Equivalent to
  - qsub -q big@@hgrp1,big@@hgrp2,medium@@hgrp1,medium@@hgrp2
- Can also be used for the hostname attribute
  - qsub -l “h=gridhost00?&!gridhost005”
  - Matches: gridhost000-gridhost009 except gridhost005
- Be careful to properly quote wildcard expressions in command line (shell may do substitutions)

Solaris 10 Dtrace script

- See <sge_root>/dtrace for README and script
- bottleneck analysis first-aid kit for administrators
  - relevant indices about masters network traffic, file and scheduling activities in a single view
  - helps to understand reasons for unsatisfactory throughput
  - suited even in large production systems due to minimum interference of Dtrace
  - Solaris 10 required on the Grid Engine master node only

DRMAA in SGE 6.1

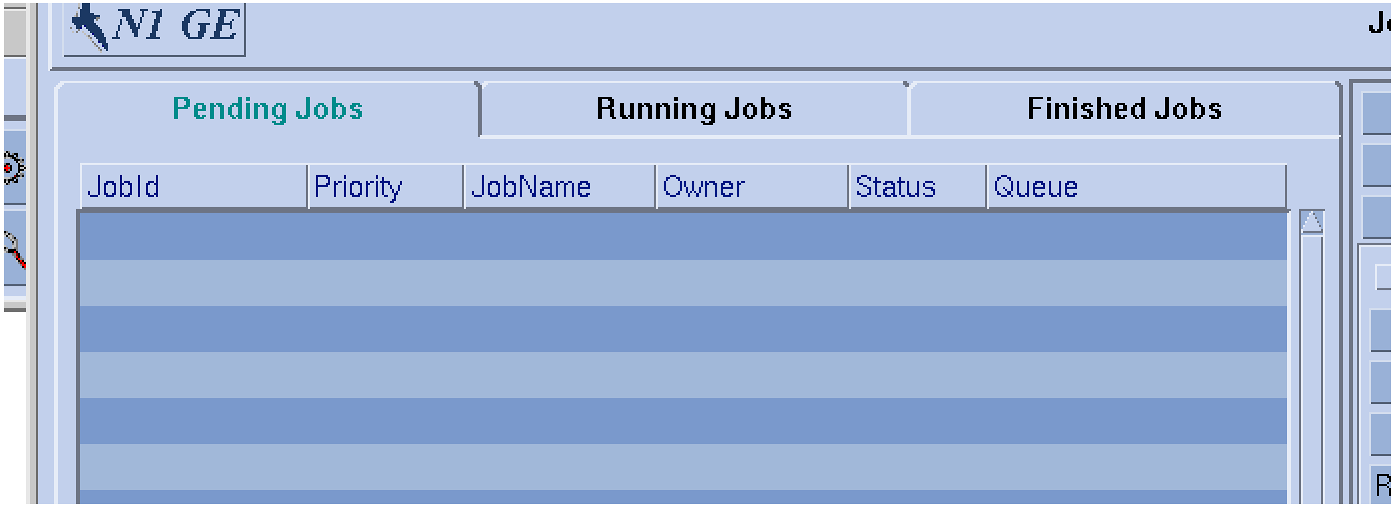
- 1.0 C binding specification implementation
  - 0.97 included for backward compatibility
  - Minor, but incompatible change from 0.97
- 1.0 Java[TM] language binding specification
  - New in Sun Grid Engine 6.1
  - 0.5 included for backward compatibility
  - Minor, but incompatible change from 0.5
  - Built as wrapper around the C binding implementation
-

Smaller enhancements

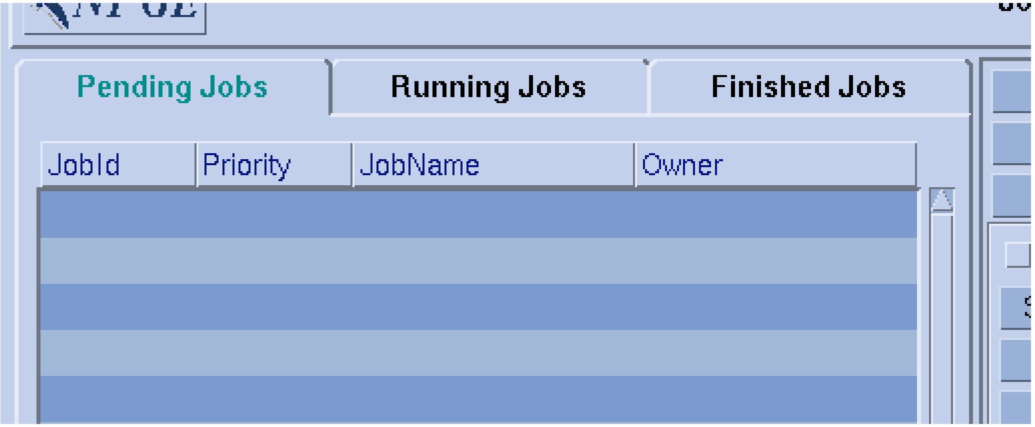
- qsub -wd <directory> switch
  - Specifiy job working directory
  - Pre SGE 6.1: only “qsub -cwd” available
  - Also supported in qmon job submission dialog
- Windows GUI job support now via boolean complex attribute `display_win_gui` request
- ~/Qmon resource file – specify job view qmon dialog
  - `Qmon*job_form*columnWidths`
  - `Qmon*job_form*visibleColumns`
  - -> see example next slide

Qmon job output customization

Default



```
Qmon*job_form*columnWidths:     8,8,15,15,17,16
Qmon*job_form*visibleColumns:   4
```



Install script changes

- New switches for inst_sge install script
  - -v - print version (bug in 6.1 FCS)
  - -copycerts - copy local certificates to given hosts
  - -winupdate - add Windows GUI display features to an existing execd installation
  - -s – install submit host (copies certs in CSP mode)
- Improved behavior of parallel automated installation
  - Template in
    - <sge_root>/util/install_modules/inst_template.conf

Need to know (1)

- New software name: Sun Grid Engine 6.1
- Same license as N1GE 6.0: License of Sun Software Portfolio (SSP)
  - Free, unlimited commercial use
  - No support entitlement (requires license)
- SGE 6.1 available for download and on DVD
  - http://www.sun.com/software/swportfolio/get.jsp
- Patch matrix
  - Approx. 15-20 patches for full set of distribution
  - Patch matrix is part of every patch README file

Need to know (2)

- Documentation for SGE 6.1 only available online
  - http://docs.sun.com/app/docs/coll/1017.4
- Linux RPM packages available (all: x86, x64, IA64)
  - Patches will be delivered with tar.gz patches to avoid patch id inflation
- Free 30-day email evaluation support available
  - See product home page on sun.com:
    - http://www.sun.com/software/gridware/
  - http://www.javelinfeedback.com/sun/index.jsp?pi=c2b00c871c1f86177ac800c779c76fab

Need to know (3)

- Grid Engine open source project and HOWTOs
  - > http://gridengine.sunsource.net
  - > http://gridengine.sunsource.net/howto/howto.html
- Community wiki of Grid Engine:
  - > http://gridengine.info

Coming: Advance Reservation (AR)

- “An advance reservation is a possibly limited or restricted delegation of a particular resource capability over a defined time interval, obtained by the requester from the resource owner through a negotiation process.” (GRAAP-WG)
- Spec at: http://gridengine.sunsource.net/nonav/source/browse/~checkout~/gridengine/doc/devel/rfe/AdvanceReservationSpecification.html
- Courtesy binary preview release available at Grid Engine open source project site since May 2007.
  - Becomes supported part of next SGE release

Advance Reservation Functionality

Part 1

- an AR has start_time, end_time/duration
- Diagnose tool to query granted ARs (qrstat)
- granted ARs is identified by a unique Handle (ID) and optional name
- AR has a user ACL list (-u switch)
- One AR can be utilized by multiple jobs from multiple users
- Job can use less or all of the reserved resources

Advance Reservation Functionality

Part 2

- AR request allows all qsub(1) request switches (e.g. -l/-q/-pe/-masterq/-ckpt/-now)
- AR are only granted if resource is available. Calendars are considered for verification, load thresholds not (e.g. host may be down at reservation time)
- Job accounting contains AR ID
- ARCo reporting is extended to cover AR event logs

AR - examples

Reserve a slot in queue all.q on host1 or host2

```
% qrsub -q all.q -l "h=host1|host2" -u $USER -a 01121200  -d 1:0:0
```

Reserve 4 slots on a host with arch=sol-sparc64

```
% qrsub -pe alloc_pe_slots 4 -l h=sol-sparc64 -u $USER -a 01121200 -d 1:0:0
```

```
% qstat
queuename                    qtype resv/used/tot. load_avg arch             states
-----------------------------------------------------------------------------------
all.q@brag                   BIPC  4/1/20         0.02     sol-sparc64
    16 0.55500 Sleeper   roland    r    11/8/2007 11:48:26    1
```

AR - examples

```
% qrstat
AR-ID name      owner state start at                    end at                  duration
----------------------------------------------------------------------------------------
    192 project1 user1   r      12/14/2006   14:47:23 12/14/2006 14:57:33 0:10:10
    193          user2   w      12/18/2006   10:00:00 12/19/2006  10:00:10 24:0:10

% qrstat -ar 193
================================================================================
id:                         193
ar_name:
submission_time:            Mon Nov 27 17:11:34 2006
owner:                      user1
acl_list:                   user1,user2
start_time:                 Mon Dec 18 10:00:00 2006
end_time:                   Tue Dec 19 10:00:10 2006
duration:                   24:0:10
granted_slots:              all.q@host1=2,all.q@host2=1
resource_list:              myapp1=1,myapp2=1
```

Sun Grid Engine Update

Andy Schwierskott

andy.schwierskott@sun.com