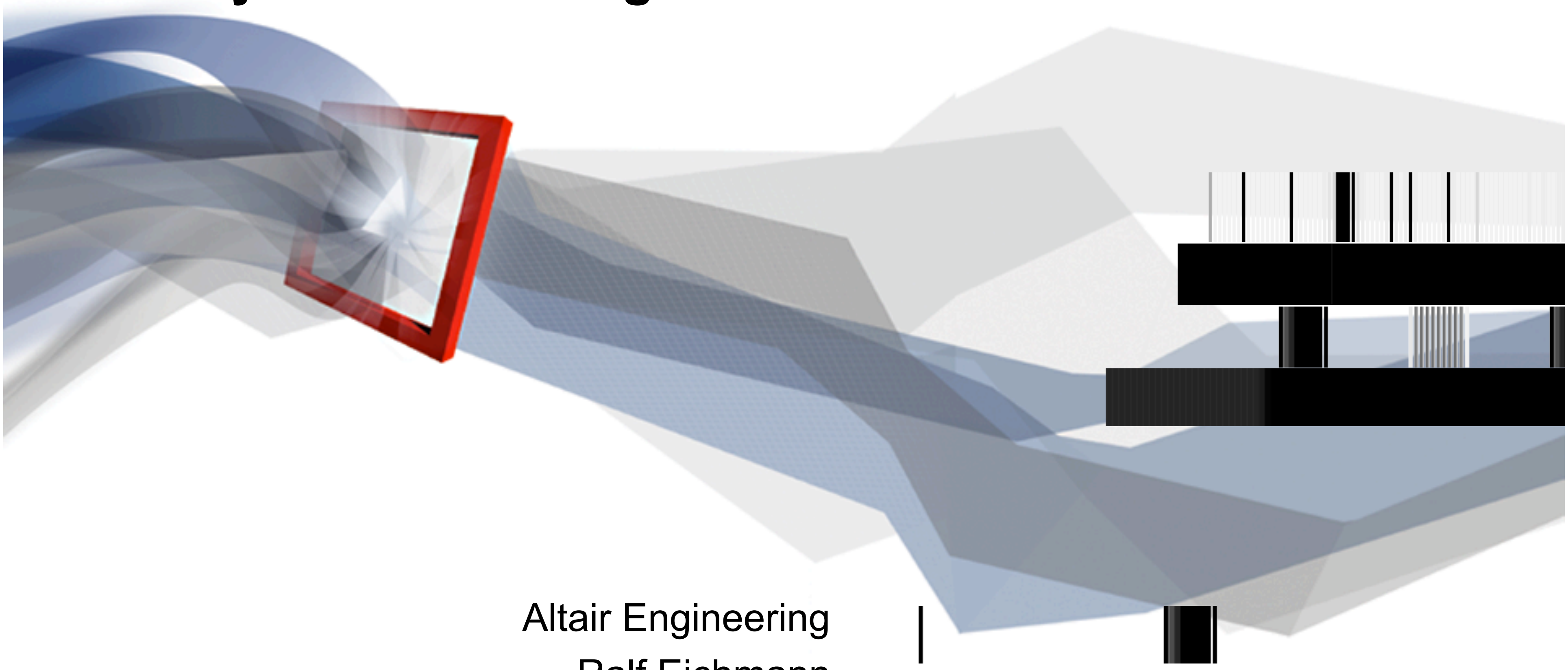
Workload Management using PBS Pro^{TM}

Portable Batch System, Professional Edition

One Day Class Covering PBS Pro v5.4 for IT Administrators



Altair Engineering

Ralf Eichmann

Technical Consultant Enterprise Computing

April 2005

Altair Engineering

The Shortest Distance Between Concept and Reality®

PBS Pro Training

- Logistics
- Outline
  - Altair and PBS Pro
  - Concepts and Terms
  - Anatomy of PBS Pro
  - Installation
  - Basic Configuration
  - Scheduling Strategies
  - Checking System Status
  - Log and Accounting Files
  - PBS Pro for Users

PBS - The Portable Batch System

- Optimal utilization of hardware and application software licenses
  - Fully configurable scheduler module
  - Arbitrary resources, fair-share, load balancing, priorities, backfilling, multi-clustering, preemption, use of idle workstations, and more
- Unified interface to all computing resources
  - All major UNIXes plus Windows 2000 and later supported
  - Heterogeneous environments supported
  - SMPs and clusters supported
  - Interactive jobs and parallel jobs supported
  - POSIX batch standard
- Sophisticated fault tolerance and security
- Professional services: Commercial support for all supported platforms and training available

PBS History

- NASA developed COSMIC NQS 15+ years ago to manage batch queuing on the supercomputers of the time, and it quickly became the *de facto* standard.
- Later, the introduction of parallel and distributed memory machines created a need for an NQS replacement.
- Many (35+) batch systems were developed by sites around the world, but none met all the needs of NASA and other large government labs.
- NASA embarked upon a project to produce a replacement for NQS. The new system had to be:
  - Capable of managing parallel and cluster systems as well as traditional supercomputers.
  - Extensible and maintainable.
  - Able to support any scheduling policy.
  - POSIX 1003.2d compliant
- The result was PBS: the Portable Batch System.

PBS History

- 1993-97: Developed for NASA to replace NQS
- 1996-97: Used as core enabling software in the NASA Metacenter (prototype Grid)
- 1998: DoD demonstrated PBS-based Metacenter at SC98 conference
- 1999: PBS and Globus used to create prototype of NASA's Information Power Grid (IPG)
- 2000: Commercial PBS Products Dept. formed within Veridian Corp.; released PBS Pro 5.0
- 2001: Released PBS Pro 5.1
- 2002: Released PBS Pro 5.2
- 2003: PBS Pro technology and engineering team acquired by Altair; Released PBS Pro 5.3
- February 2004: Released PBS Pro 5.4

Altair at a Glance

1985 Altair founded as an Engineering Services provider in Detroit, USA

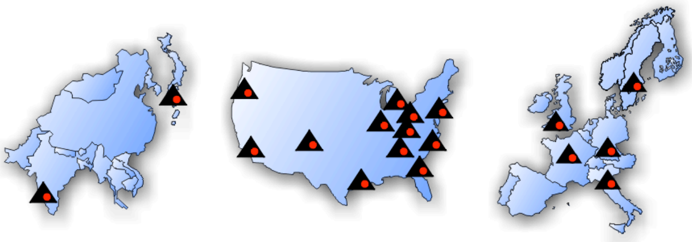
1989 Release of the first commercial software product Altair HyperMesh 1.0

1994 Release of OptiStruct 1.0, received „Technology Of The Year“ award

2003 Acquisition of the PBS Pro technology and development team; founded Altair Grid Technologies

today **Global product design and technology company**

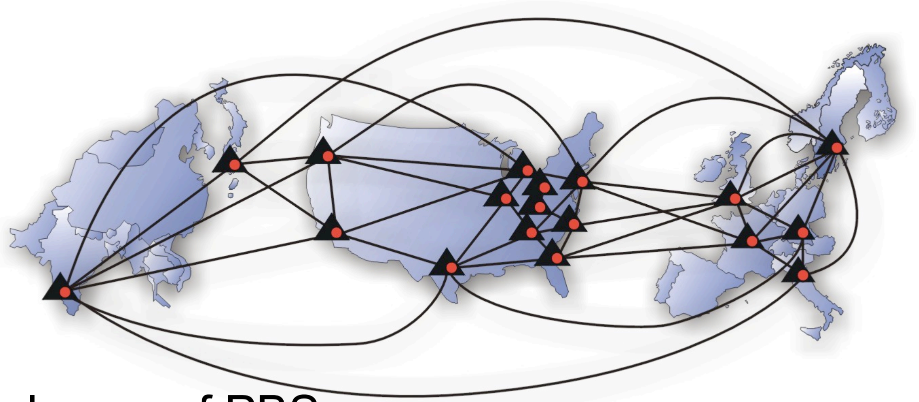
More than 20 offices world-wide
More than 800 employees
More than 4000 customers



Altair Engineering (www.altair.de), Altair Grid Technologies (www.pbspro.com), 2004

Altair and PBS Pro

] PBS Pro Sales and Technical Support
  - Altair Engineering provides world-wide sales and technical support for PBS Pro via offices in North America, Europe, and Asia.



] PBS Pro Engineering Team
  - Contains all the original developers of PBS
  - Focused on enhancing and supporting PBS Pro
  - Separate company: Altair Grid Technologies, LLC, operating as a subsidiary of Altair Engineering, Inc.
  - Offices in Mountain View, California

PBS Pro Training

] Logistics

] Outline

- Altair and PBS Pro
- Concepts and Terms
- Anatomy of PBS Pro
- Installation
- Basic Configuration
- Scheduling Strategies
- Checking System Status
- Log and Accounting Files
- PBS Pro for Users

Concepts and Terms

- A resource management or batch queuing system has three primary roles:
  - **Queuing** of work or tasks to be run on a computer. Users submit their jobs to the batch system where they are queued up until the system is ready to run them
  - **Scheduling**, or the process of selecting which jobs to run when and where, according to a predetermined policy. Sites try to balance competing needs and goals on the system; scheduling is often wrought with compromise. You can’t please all the users all the time...
  - **Monitoring**, tracking and reserving system resources, and enforcing usage policy. Covers user-level and system level monitoring; also monitoring of the scheduling algorithms to see how well they are meeting the stated goals

Concepts and Terms

- **Node**: Computer system with a single operating system image, a virtual memory space, one or more virtual CPU, and one or more IP address
  - Cluster node: 1:1 relation of virtual processor to task
  - Time-shared node: Can be over-committed
- **Job**: Basic execution object, consists of tasks
- **Queue**: Named container for jobs
  - Routing queue: Move jobs to an execution queue
  - Execution queue: Execute jobs
- Node, Queue, and Server **Attribute**: Provide control information
  - **Pre-defined**: node type, node state, queue type,…
  - Node **Property**: User-defined strings
  - Node **Resource**: User-defined, node level, value modifiable
  - Queue and Server **Resource**: Similar to node resources, but different level

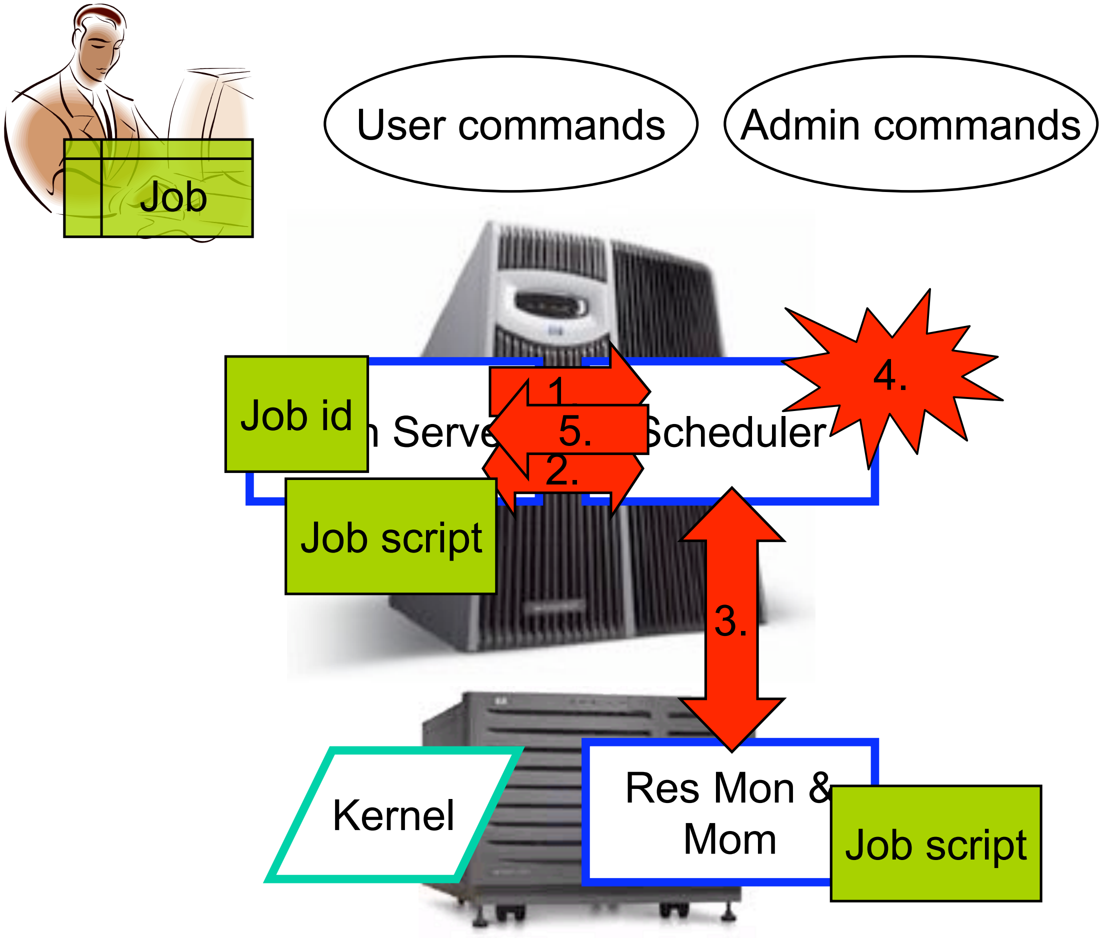
Concepts and Terms

- **Account**: Strings to group for charging of resource use
- **Administrator**, **manager**, **operator**: Different levels of privilege inside PBS Pro
- **Destination**: Location for jobs within PBS Pro, e.g. a queue or queue@server
- **File Staging**: File movement
- **Job Hold** and **Job Release**: Artificially restrict or allow jobs to be considered for scheduling
- **Job Owner**: User who submitted a job
- **Task**: POSIX session started by PBS Pro on behalf of a job
- **Users** and **Groups**: Establish level of control, uses names and IDs
- **Virtual Processor**: Number of tasks supported by a node, defaults to the number of physical processors

Anatomy of PBS Pro: Overview

- PBS Pro Daemons
  - Server (pbs_server)
  - Scheduler (pbs_sched)
  - Resource monitor and job executor (pbs_mom)
- PBS Pro Commands
  - User Commands
    - qsub, qstat, qdel, …
  - Administrator Commands
    - qmgr, pbsnodes, tracejob, …
- PBS Pro database
  - $PBS_HOME, usually /var/spool/PBS
- PBS Pro documentation
  - PDF files: Administrator Guide (AG), User Guide (UG), QSG, SCG, ERS
  - Man pages

Anatomy of PBS Pro: Overview



1. New job arrived message
2. Request all job/queue info from server
3. Request system resource info from mom
4. Select a job to run according to resources available and local scheduling policy
5. Request server to run job on host X (or N nodes)
6. Server sends job script to MOM to run
7. MOM starts and monitors job

Altair Engineering (www.altair.de), Altair Grid Technologies (www.pbspro.com), 2004

Anatomy of PBS Pro: The Server

- Maintains queues and jobs
- Communicates with:
  - Client commands
  - Mini-server
  - Resource Monitor
  - Scheduler
  - Other servers
    - Forwarding jobs
    - Status requests
    - Fail-over

Anatomy of PBS Pro: The Scheduler

] The PBS scheduler serves in the role of implementing the local site policy.
  - Queries list of running and queued jobs from the Server
  - Queries queue limits, etc. from the Server
  - Queries resource consumption and availability from MOM
  - Sorts available jobs according to local policy
  - Selects and runs jobs according to local policy, available resources, pending deadlines, etc.

Anatomy of PBS Pro: The Machine Oriented Mini-server

- MOM
  - Executes jobs at request of Server

- Resource Monitor
  - Monitors resource usage of running jobs
  - Enforces limits on jobs
  - Reports system resource limits, configuration (e.g. memory, CPU utilization, swap rate, etc.)

PBS Pro Training

- Logistics
- Outline
  - Altair and PBS Pro
  - Concepts and Terms
  - Anatomy of PBS Pro
  - Installation
  - Basic Configuration
  - Scheduling Strategies
  - Checking System Status
  - Log and Accounting Files
  - PBS Pro for Users

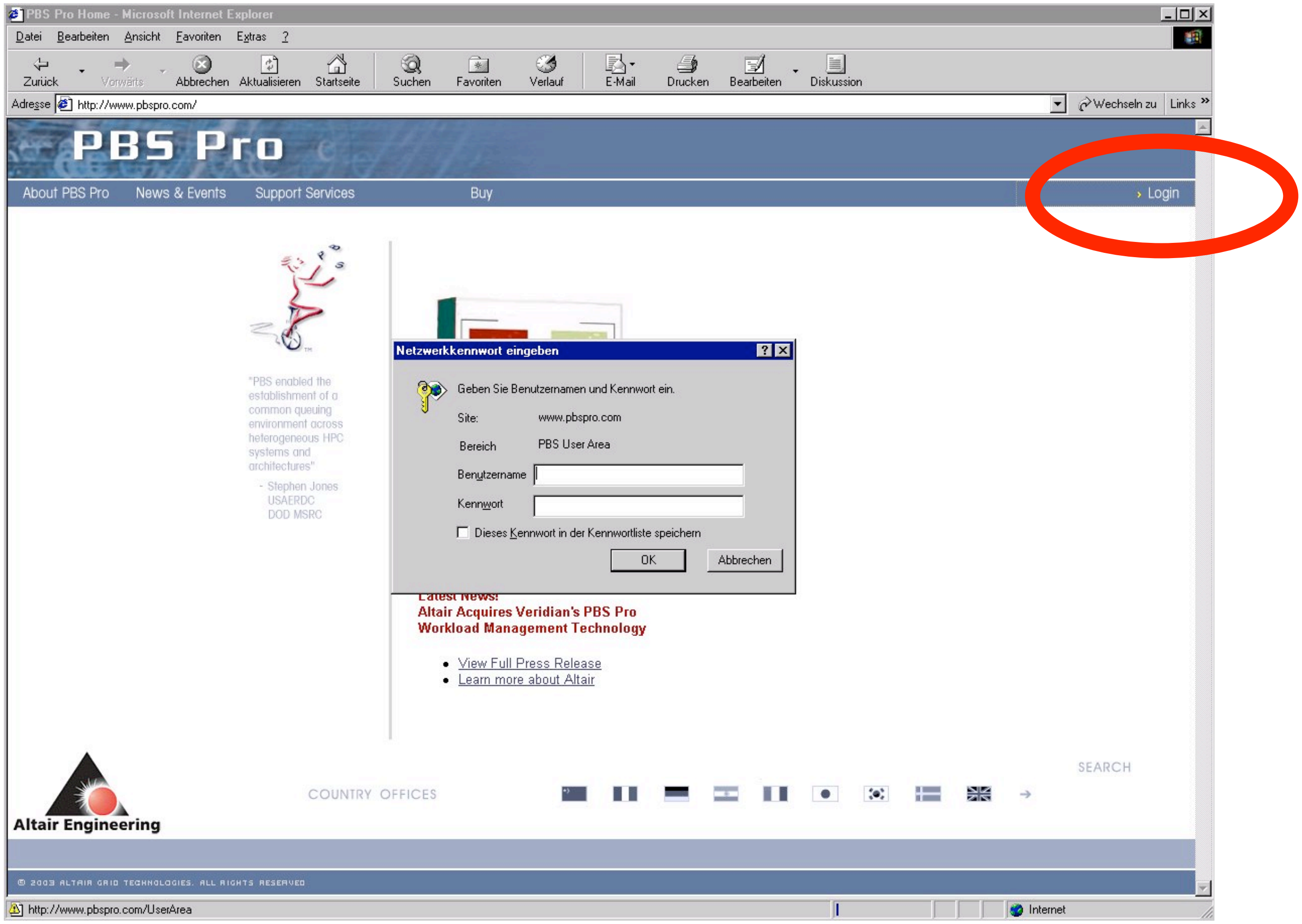
Installation: Get in Touch

- www.altair.de
- www.pbspro.com
- www.altair.com

- support@altair.de
- pbssupport@altair.de
- sales@pbspro.com
- support@pbspro.com

- +49 7031 6208 22

Installation: Log Into Your Account


PBS Pro Home - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten Verlauf E-Mail Drucken Bearbeiten Diskussion
Adresse http://www.pbspro.com/
Wechseln zu Links
PBS Pro
About PBS Pro
News & Events
Support Services
Buy
› Login
COUNTRY OFFICES
SEARCH
Altair Engineering
© 2003 ALTAIR GRID TECHNOLOGIES. ALL RIGHTS RESERVED
http://www.pbspro.com/UserArea
Internet

"PBS enabled the establishment of a common queuing environment across heterogeneous HPC systems and architectures"

- Stephen Jones USAERDC DOD MSRC

Netzwerkkennwort eingeben

Geben Sie Benutzernamen und Kennwort ein.

Site: www.pbspro.com

Bereich PBS User Area

Benutzername

Kennwort

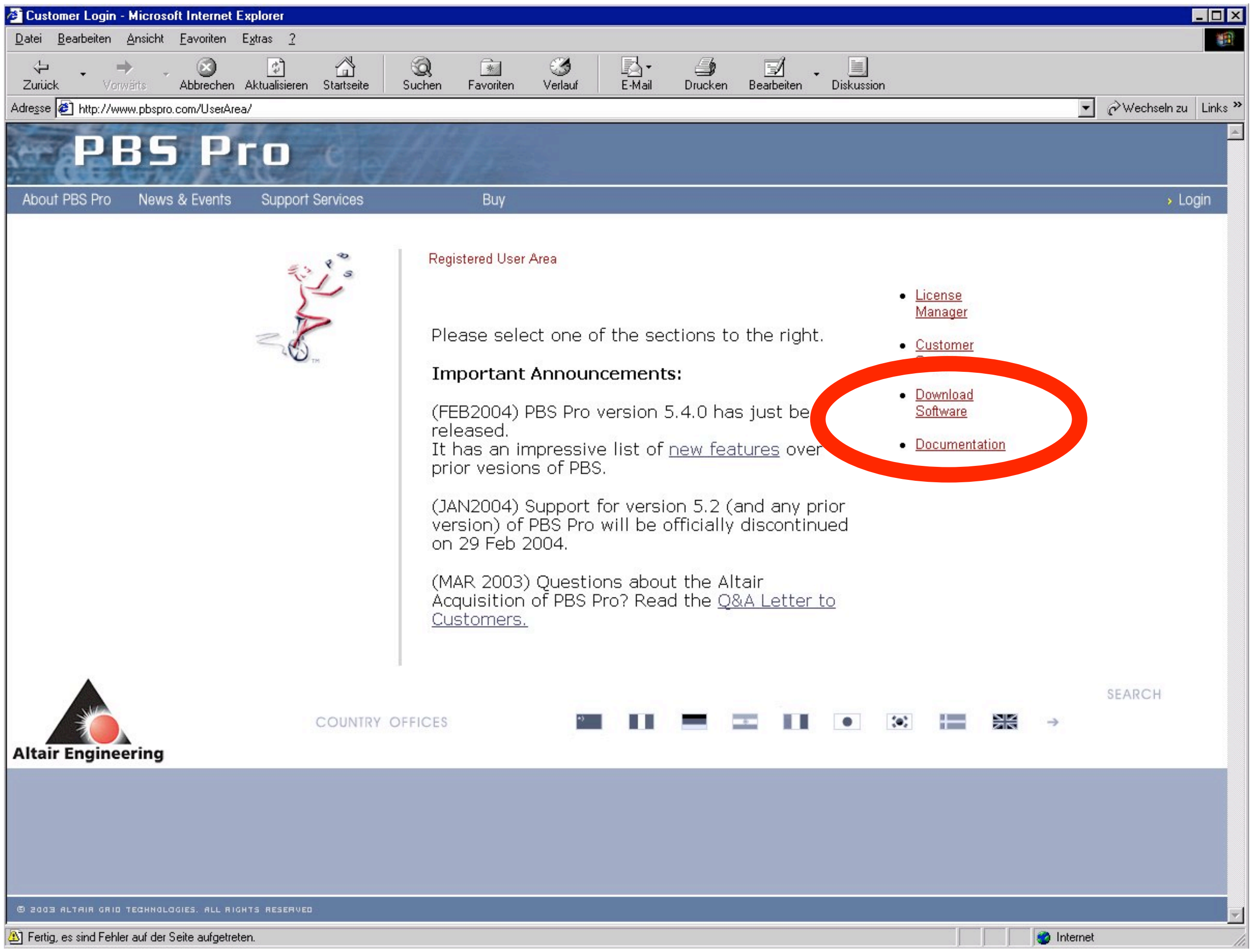
□ Dieses Kennwort in der Kennwortliste speichern

OK Abbrechen

Latest News!
Altair Acquires Veridian's PBS Pro Workload Management Technology

- View Full Press Release
- Learn more about Altair

Installation: Download Software and Docs


Customer Login - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück
Vorwärts
Abbrechen
Aktualisieren
Startseite
Suchen
Favoriten
Verlauf
E-Mail
Drucken
Bearbeiten
Diskussion
Adresse http://www.pbspro.com/UserArea/
Wechseln zu
Links
PBS Pro
About PBS Pro
News & Events
Support Services
Buy
Login
COUNTRY OFFICES
SEARCH
Altair Engineering
© 2003 ALTAIR GRID TECHNOLOGIES. ALL RIGHTS RESERVED
Fertig, es sind Fehler auf der Seite aufgetreten.
Internet

Registered User Area

Please select one of the sections to the right.

Important Announcements:

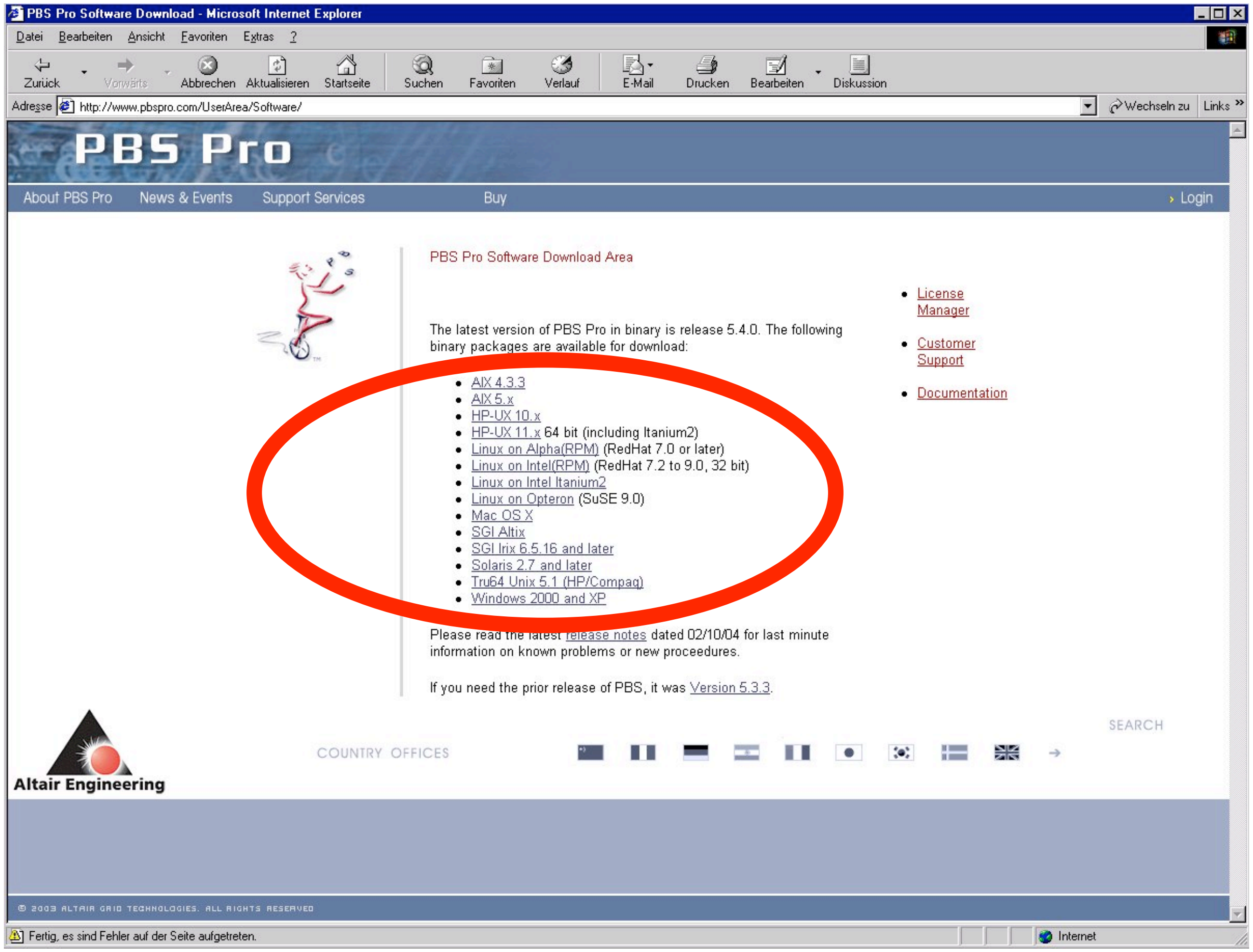
(FEB2004) PBS Pro version 5.4.0 has just be released.
It has an impressive list of new features over prior vesions of PBS.

(JAN2004) Support for version 5.2 (and any prior version) of PBS Pro will be officially discontinued on 29 Feb 2004.

(MAR 2003) Questions about the Altair Acquisition of PBS Pro? Read the Q&A Letter to Customers.

- License Manager
- Customer
- Download Software
- Documentation

Installation: Download Software and Docs


PBS Pro Software Download - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten Verlauf E-Mail Drucken Bearbeiten Diskussion
Adresse http://www.pbspro.com/UserArea/Software/
Wechseln zu Links
PBS Pro
About PBS Pro News & Events Support Services Buy › Login

PBS Pro Software Download Area

The latest version of PBS Pro in binary is release 5.4.0. The following binary packages are available for download:

- AIX 4.3.3
- AIX 5.x
- HP-UX 10.x
- HP-UX 11.x 64 bit (including Itanium2)
- Linux on Alpha(RPM) (RedHat 7.0 or later)
- Linux on Intel(RPM) (RedHat 7.2 to 9.0, 32 bit)
- Linux on Intel Itanium2
- Linux on Opteron (SuSE 9.0)
- Mac OS X
- SGI Altix
- SGI Irix 6.5.16 and later
- Solaris 2.7 and later
- Tru64 Unix 5.1 (HP/Compaq)
- Windows 2000 and XP

Please read the latest release notes dated 02/10/04 for last minute information on known problems or new proceedures.

If you need the prior release of PBS, it was Version 5.3.3.

- License Manager
- Customer Support
- Documentation

SEARCH
Altair Engineering
COUNTRY OFFICES
© 2003 ALTAIR GRID TECHNOLOGIES. ALL RIGHTS RESERVED
Fertig, es sind Fehler auf der Seite aufgetreten.
Internet

Installation: Planning

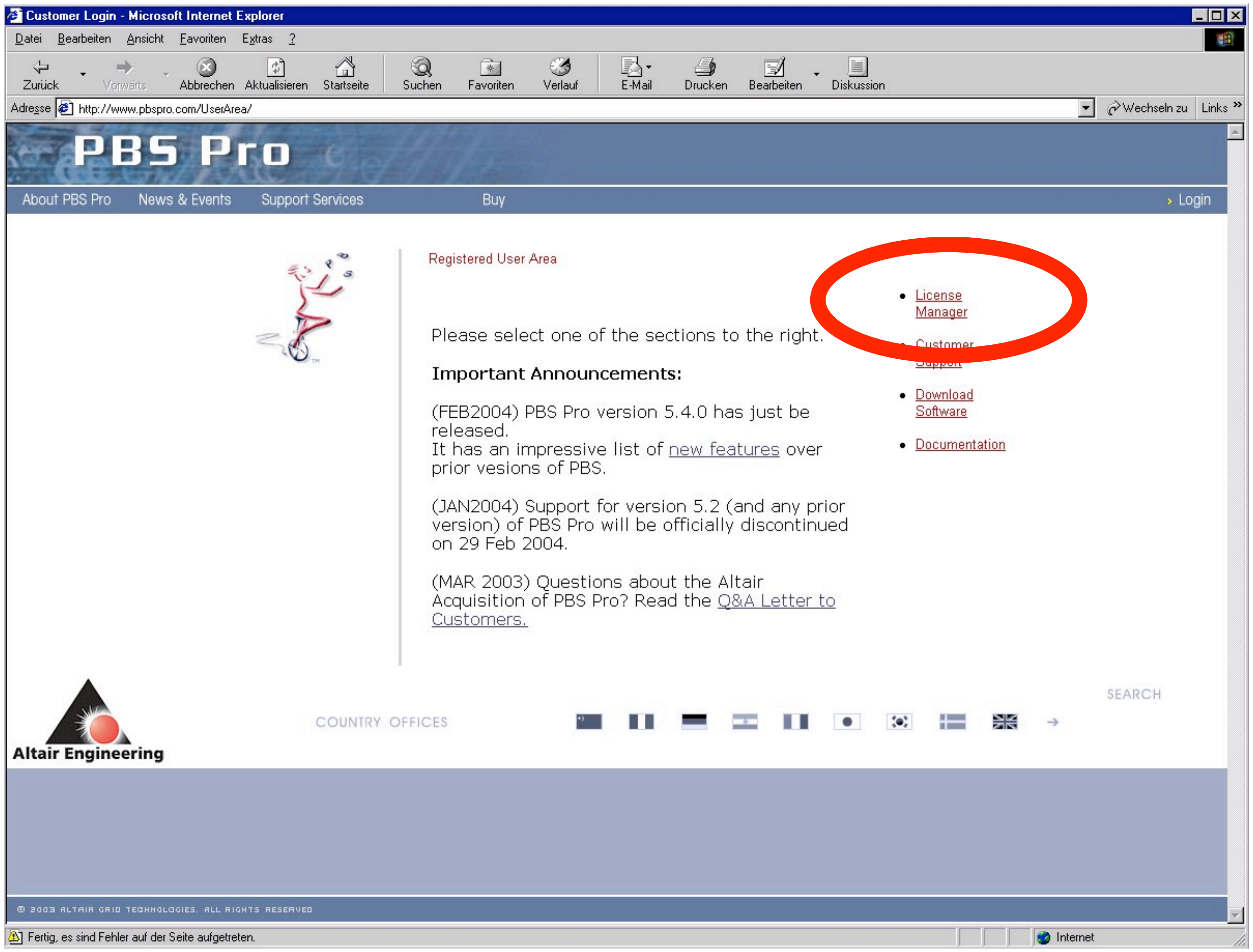
- Decide role for each computer system:
  - One or two PBS Pro servers
  - Execution nodes (need licenses)
  - Commands only (submit hosts)

- Choose place for PBS Pro binaries, libraries, man pages ($PBS_EXEC)
- Choose place for PBS Pro database ($PBS_HOME)

Installation: Installing the Software

- Download binary package
- Uncompress package into temporary directory
- Run the INSTALL script
  - Supply role of the computer: PBS Pro server, execution only, commands only
  - Supply place for PBS Pro commands ($PBS_EXEC)
  - Supply place for PBS Pro database ($PBS_HOME)
  - For execution only or commands only: Supply PBS Pro server name
- Review configuration files
- On the server: Run pbs_hostid
- Start daemons

Installation: Creating Keys From Licenses



http://www.pbspro.com/UserArea/

Registered User Area

Please select one of the sections to the right.

Important Announcements:

(FEB2004) PBS Pro version 5.4.0 has just be released.
It has an impressive list of new features over prior vesions of PBS.

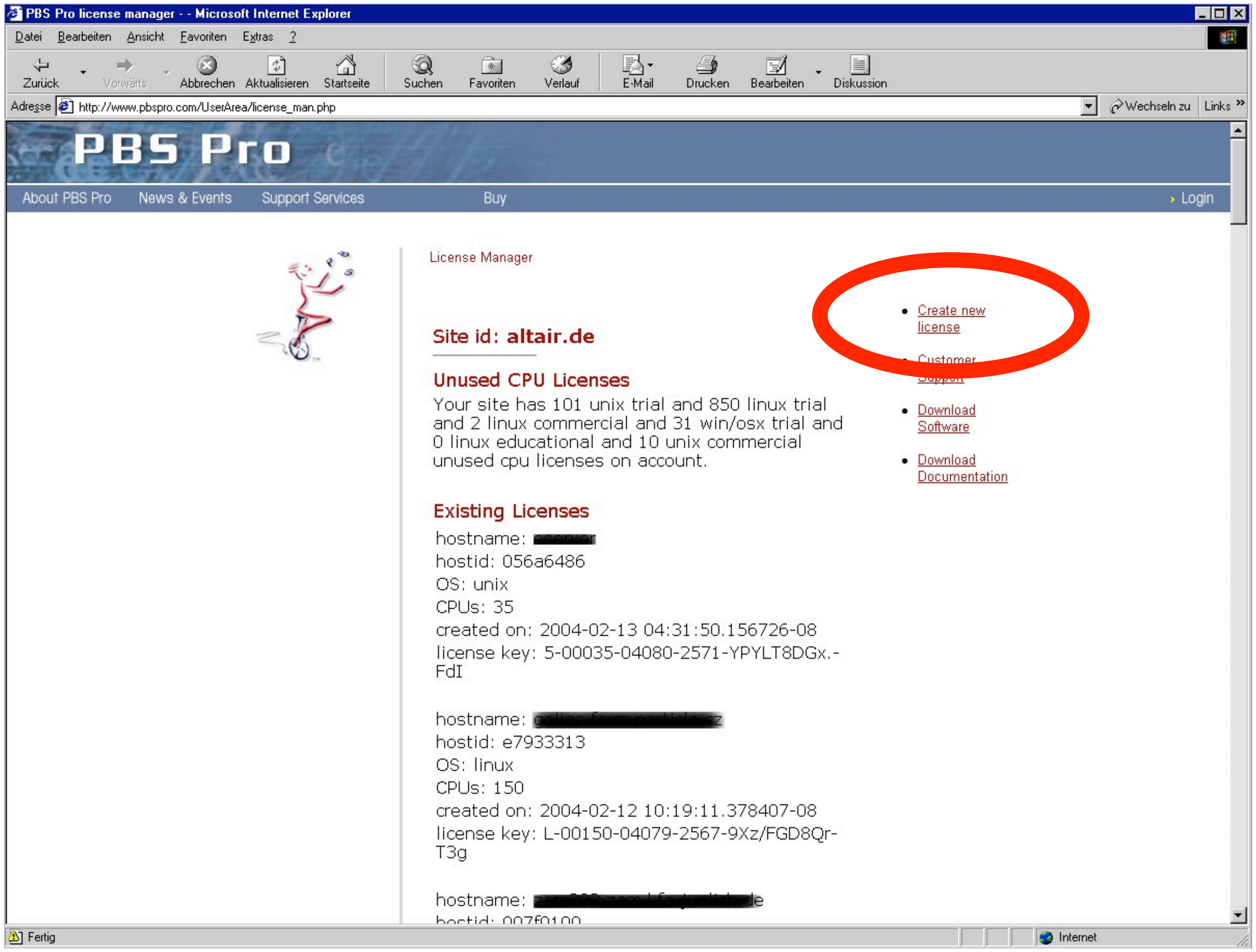
(JAN2004) Support for version 5.2 (and any prior version) of PBS Pro will be officially discontinued on 29 Feb 2004.

(MAR 2003) Questions about the Altair Acquisition of PBS Pro? Read the Q&A Letter to Customers.

- License Manager
- Customer Support
- Download Software
- Documentation

Fertig, es sind Fehler auf der Seite aufgetreten.

Installation: Creating Keys From Licenses


PBS Pro license manager - - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten Verlauf E-Mail Drucken Bearbeiten Diskussion
Adresse http://www.pbspro.com/UserArea/license_man.php
Wechseln zu Links
PBS Pro
About PBS Pro
News & Events
Support Services
Buy
Login
Fertig
Internet

License Manager

- Create new license
- Customer Support
- Download Software
- Download Documentation

Site id: altair.de

Unused CPU Licenses

Your site has 101 unix trial and 850 linux trial and 2 linux commercial and 31 win/osx trial and 0 linux educational and 10 unix commercial unused cpu licenses on account.

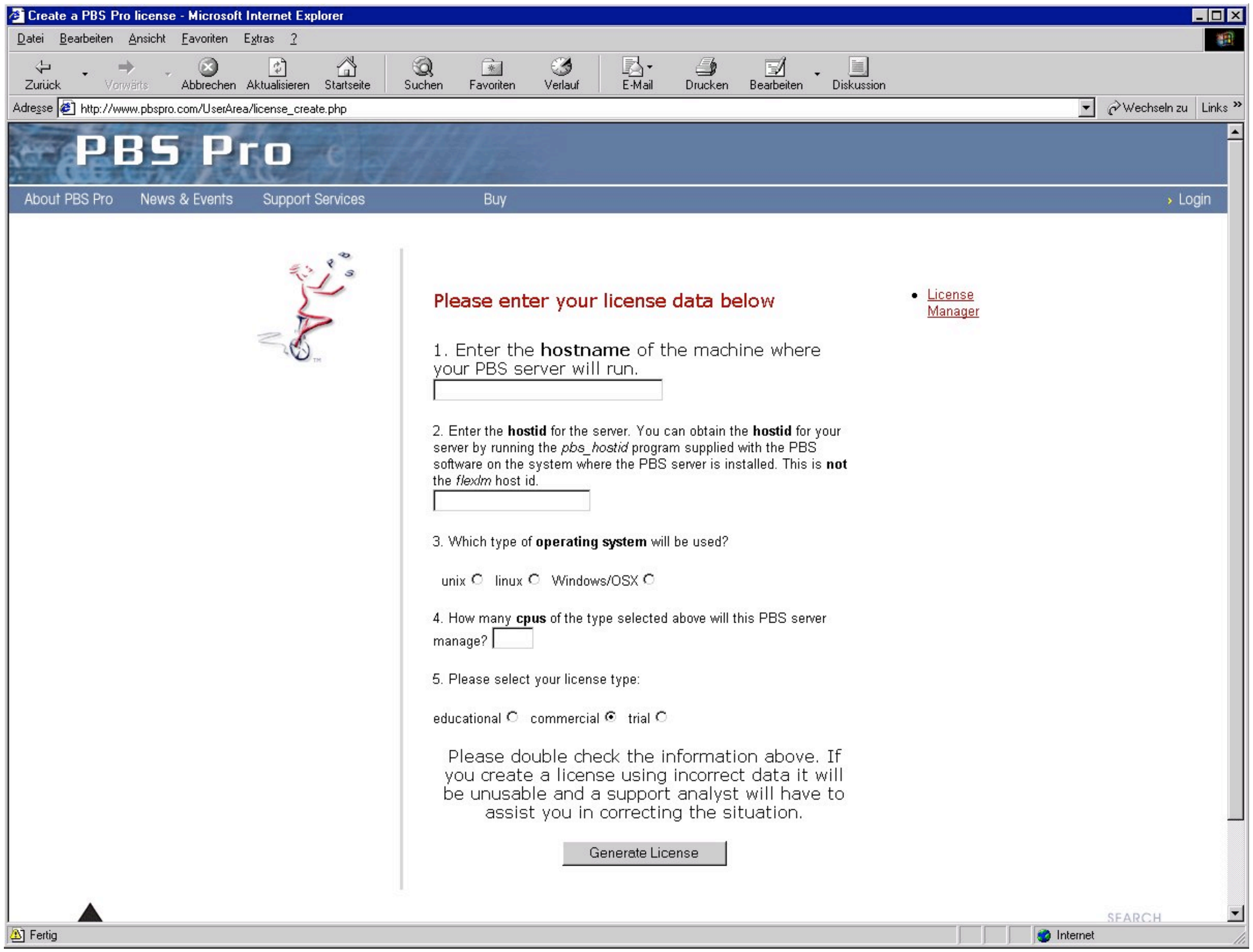
Existing Licenses

hostname: [redacted]
hostid: 056a6486
OS: unix
CPUs: 35
created on: 2004-02-13 04:31:50.156726-08
license key: 5-00035-04080-2571-YPYLT8DGx.-FdI

hostname: [redacted]
hostid: e7933313
OS: linux
CPUs: 150
created on: 2004-02-12 10:19:11.378407-08
license key: L-00150-04079-2567-9Xz/FGD8Qr-T3g

hostname: [redacted]
hostid: 007f0100

Installation: Creating Keys From Licenses



Create a PBS Pro license - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Vorwärts Abbrechen Aktualisieren Startseite Suchen Favoriten Verlauf E-Mail Drucken Bearbeiten Diskussion
Adresse http://www.pbspro.com/UserArea/license_create.php
Wechseln zu Links
PBS Pro
About PBS Pro News & Events Support Services Buy Login
License Manager
SEARCH
Fertig
Internet

Please enter your license data below

1. Enter the **hostname** of the machine where your PBS server will run.

2. Enter the **hostid** for the server. You can obtain the **hostid** for your server by running the *pbs_hostid* program supplied with the PBS software on the system where the PBS server is installed. This is **not** the *flexlm* host id.

3. Which type of **operating system** will be used?

unix ○ linux ○ Windows/OSX ○

4. How many **cpus** of the type selected above will this PBS server manage?

5. Please select your license type:

educational ○ commercial ● trial ○

Please double check the information above. If you create a license using incorrect data it will be unusable and a support analyst will have to assist you in correcting the situation.

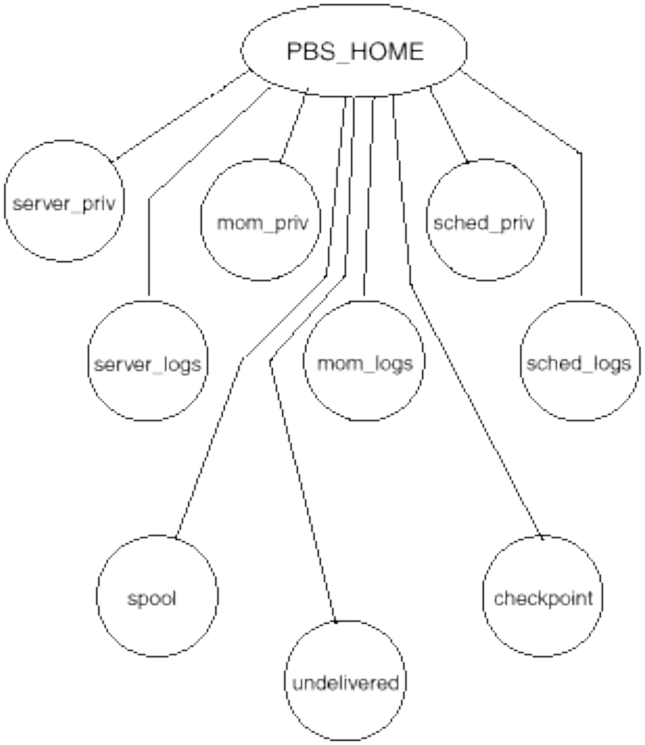
Generate License

PBS Pro Training

- Logistics
- Outline
  - Altair and PBS Pro
  - Concepts and Terms
  - Anatomy of PBS Pro
  - Installation
  - Basic Configuration
  - Scheduling Strategies
  - Checking System Status
  - Log and Accounting Files
  - PBS Pro for Users

Basic Configuration: Tools and Files

- General settings
  - /etc/pbs.conf
- Server, queue and node configuration:
  - qmgr command
- Licenses:
  - server_priv/license_file
- Node list:
  - server_priv/nodes
- Scheduling parameters:
  - sched_priv/sched_config
- Node attributes:
  - mom_priv/config



Basic Configuration: One Server and Nodes

- Install software on server
- Check local configuration
  - Install license key
- Start daemons on server
- Do basic configuration
  - Promote user to manager
  - Adjust server configuration
  - Add queue
  - Adjust access control lists
- Add nodes
  - Install software
  - Configure nodes

Basic Configuration: Server installation

- Install software on server
- Check configuration
  - /etc/pbs.conf

```
PBS_EXEC=/usr/pbs
PBS_HOME=/var/spool/PBS
PBS_START_SERVER=1
PBS_START_MOM=1
PBS_START_SCHED=1
PBS_SERVER=altgtest1
```

  - server_priv/license_file

```
L-00002-99999-1182-9WbwjlKlEO-gKO
```

- Start PBS Pro daemons

```
/etc/init.d/pbs start
```

- Check system status

```
qstat -Bf
```

Basic Configuration: Managers and Operators

- PBS Pro manager and operator are privileged levels of control for users.
- Manager: Highest level of control
  - Can create and delete queues and nodes
  - Can modify server, queue, and node configuration
  - Can work on any job
  - Allows PBS Pro administration largely without superuser privileges
- Operator: Can use some restricted capabilities
  - Can modify server and queue configuration, but not
    - node attributes
    - security-related attributes
    - scheduling-related attributes
  - Can work on any job

Basic Configuration: Server Configuration

- Use qmgr command, because server configuration is stored in binary files.
- Typical commands:
  - print server
  - list server
  - set server …
  - unset server …
- qmgr –c “p s”
- Reference: AG 6

```
[root@altgtest1 server_priv]# qmgr
Max open servers: 4
Qmgr: p s
# …
#
# Set server attributes.
#
set server scheduling = True
set server default_queue = workq
set server log_events = 511
set server mail_from = adm
set server query_other_jobs = True
set server resources_default.ncpus = 1
set server scheduler_iteration = 600
set server resv_enable = True

Qmgr: set server managers+=eichmann@altgtest1
```

Basic Configuration: Adding a Queue

- Use qmgr
- Enabled queue: Jobs may be submitted
- Started queue: Jobs can be routed or executed.

```
[root@altgtest1 server_priv]# qmgr
Max open servers: 4
Qmgr: p q workq
#
# Create queues and set their attributes.
#
#
# Create and define queue workq
#
create queue workq
set queue workq queue_type = Execution
set queue workq enabled = True
set queue workq started = True
```

Basic Configuration: Using Access Control Lists

- Restrict access to a PBS Pro server for users, groups, or hosts
- Consider using a flat uid scheme
- Primary GIDs considered only.

```
[root@altgtest1 server_priv]# qmgr
Max open servers: 4
Qmgr: s s flatuid=true
Qmgr: s s acl_host_enable=true
Qmgr: s s acl_hosts+=*.altair.de
Qmgr: s s acl_user_enable=true
Qmgr: s s acl_users="eichmann,waldeck"
```

- Restrict access to a PBS Pro queue for users, groups, or hosts

```
Qmgr: s q workq acl_group_enable=true
Qmgr: s q workq acl_groups=ec
```

Basic Configuration: Adding a Node

- Install software on node

- Add to server’s database during run-time:
  Use qmgr

```
[root@altgtest1 server_priv]# qmgr
Max open servers: 4
Qmgr: c n altgtest2
Qmgr: s n altgtest2 ntype=cluster
Qmgr: s n altgtest2 resources_available.ncpus=2
```

- Alternatively: Modify server_priv/nodes with text editor
  - Shutdown server (server overwrites nodes file from values in memory)
  - Modify nodes file
  - Restart server
- Order of nodes in the nodes file establishes default node sorting.

Basic Configuration: Node Configuration

- Use nodes file or qmgr
  - properties
  - limits for node level internally tracked resources
  - control attributes: state, ntype, license
  - Reference: AG 6.6, 6.7

```
[root@altgtest1 server_priv]# qmgr
Max open servers: 4
Qmgr: s n altgtest2 property=“fast,myri”
Qmgr: s n altgtest2 max_running=2
```

- mom_priv/config file
  - security
  - log level
  - load thresholds
  - …
  - Reference: AG 7.2

```
$clienthost altgtest1
$restricted *.altair.de
$logevent 255
$max_load 2.2
```

Basic Configuration: Two Servers and Nodes

- PBS Pro 5.4 supports a primary and a secondary server to build a high availability system
- Prerequisites:
  - Shared filesystem for both servers (NFS server, external RAID box)
  - Same architecture for both servers
- Active/passive configuration with heartbeat signal
- PBS Pro database access internally synchronized
- Reference: AG 6.15
- Caveats:
  - Routing queue setup
  - Peer queue setup
  - Mom database on secondary server
  - Scheduler on secondary server

```
PBS_EXEC=/usr/pbs
PBS_HOME=/var/spool/PBS
PBS_START_SERVER=1
PBS_START_MOM=1
PBS_START_SCHED=1
PBS_SERVER=altgtest1
PBS_PRIMARY=altgtest1
PBS_SECONDARY=altgtest2
```

PBS Pro Training

- Logistics
- Outline
  - Altair and PBS Pro
  - Concepts and Terms
  - Anatomy of PBS Pro
  - Installation
  - Basic Configuration
  - Scheduling Strategies
  - Checking System Status
  - Log and Accounting Files
  - PBS Pro for Users

Scheduling: Overview

- A site tries to balance competing demands on the systems:
  - Users want fast turnaround of their jobs.
  - Managers want the highest possible utilization of the system.
  - Administrators want a static system (set it up and leave it alone).
- The PBS Pro scheduler (called “standard”) is a sophisticated general purpose scheduler implementing a variety of (selectable) scheduling algorithms.
- Sites can edit the configuration file to change behavior (see AG 8)
- PBS administrators use input from management and feedback from the users to optimize the scheduling of a particular system to the local needs.
- Configuration is done through editing sched_priv/sched_config and other auxiliary text files.
- Scheduler re-reads its configuration file upon receiving SIGHUP.

Scheduling: Adding Properties

- Node properties can be added through qmgr or the nodes file.
- Property requests are honored for single-node and multi-node jobs.

Scheduling: Adding a Node-level Resource

- The PBS Pro scheduler supports arbitrary resources tracked by PBS Pro on node level.

- Define resource in server_priv/resourcedef

  ```
  lsdyna       type=long   flag=n
  ```

- Declare available amount in qmgr

  ```
  s n altgtest1 resources_available.lsdyna=2
  ```

- Include into scheduling in sched_priv/sched_config

  ```
  resources: “ncpus,mem,lsdyna”
  ```

- Scheduling based on resources_available.lsdyna, on resources_assigned.lsdyna, and on the requested amount.
- Node-level resources are honored for single-node jobs only.
- Reference: AG 9

Scheduling: Adding a Queue- or Server-level Resource

- The PBS Pro scheduler supports arbitrary resources tracked by PBS Pro on queue and server level.

- Define resource in server_priv/resourcedef	pamcrash type=long flag=q
- Declare available amount in qmgr	s s resources_available.pamcrash=8
- Include into scheduling in sched_priv/sched_config	resources: “ncpus,mem,pamcrash”

- Scheduling based on resources_available.pamcrash, on resources_assigned.pamcrash, and on the requested amount.
- Server-level resources are honored for both single-node jobs and multi-node jobs.
- Reference: AG 9

Scheduling: External Load Sensors on Node Level

- The PBS Pro scheduler supports arbitrary resources tracked externally on node level.

- Define resource in server_priv/resourcedef

```
scratch      type=size
```

- Implement and install load sensor in mom_priv/config

```
scratch !/usr/local/bin/scratch.pl
```

- Include into scheduling in sched_priv/sched_config

```
resources: “ncpus,mem,scratch”
mom_resources: “scratch”
```

- Load sensor returns available amount in supported units on stdout.
- Node-level resources are honored for single-node jobs only.
- Reference: AG 9

Scheduling: External Load Sensors on Server Level

- The PBS Pro scheduler supports arbitrary resources tracked externally on server level.

] Define resource in server_priv/resourcedef	hwu type=long
] Implement and install load sensor in sched_priv/sched_config	server_dyn_res:”hwu !/usr/local/bin/flex_hwu.pl”
] Include into scheduling in sched_priv/sched_config	resources: “ncpus,mem,hwu”

- Load sensor returns available amount in supported units on stdout.
- Server-level resources are honored for both single-node jobs and multi-node jobs.
- Reference: AG 9

Scheduling: Using Limits for Users and Groups

		server	queue	node
• max_running		x	x	x
• max_user_run		x	x	x
• max_user_run_soft	x	x		
• max_group_run		x	x	x
• max_group_run_soft	x	x		
• max_user_res.resource		x	x	
• max_user_res_soft.resource	x	x		
• max_group_res.resource		x	x	
• max_group_res_soft.resource	x	x		

] Soft and hard limits allow with one configuration:

- Full utilization in times of low competition
- Fair resource sharing in times of high competition

] Soft limits effective only in conjunction with preemption

Scheduling: Defaults and Limits for Queues and Server

- Default, minimum and maximum values for resources may be set per job for server and queues.
- Use:
  - Selective routing using routing queues (AG 6.11)
  - Implement custom policies, e.g.: Run no more than 6 jobs in queues “medium” and “long”.

- resources_min.res (Doc Bug: Not available at server level!)
- resources_max.res
- resources_default.res
- resources_available.res

- Server defaults are used if there is no queue default.
- Defaults can be used to enforce limits on resources not explicitly requested.
- Checks for min and max are performed before default is assigned.
- Min=max works like an exact requirement, even for string resources.

Scheduling: FIFO

- FIFO = First In First Out
- Configuration:
  - sched_priv/sched_config

    strict_fifo: true

- If used together with job_sort_key: Changes meaning to strict ordering
- No backfilling

Scheduling: Fair-sharing

- Automatically assign priorities to waiting jobs based on past usage.
- Configuration:
  - sched_priv/sched_config

    fair_share: true

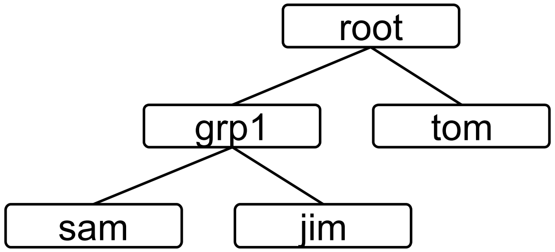
    fairshare_entity: euser

    fairshare_usage_res: cput

    half_life: 24:00:00

    sync_time: 1:00:00
  - sched_priv/resource_group
    - Create hierarchical tree of shares
- Available as preemption level
- Monitoring, tuning:
  - pbsfs (AG 11.5)

```
#name   id   parent   share
grp1    10   root     10
tom     11   root     10
sam     20   grp1     10
jim     21   grp1     20
```



#name	same level	global
root	100%	100%
tom	10/20=50%	100%*50%=50%
grp1	10/20=50%	100%*50%=50%
sam	10/30=33%	50%*33%=16%
jim	20/30=66%	50%*66%=33%

Scheduling: Fair-sharing Details

- Fair-sharing with empty sched_priv/resource_group file:
  - Even-sharing

- Entities not listed in the fair-share tree
  - sched_priv/sched_config

    unknown_shares: 10

    fairshare_enforce_no_shares: false (allow entities without shares to run jobs)

- Instantaneous fair-sharing based on jobs
  - Introduce server-level resource “jobct” with server default 1
  - Use jobct as fair-sharing resource
  - Use small half life (but larger than 2 minutes)

Scheduling: Time-dependent Scheduling

- Primetime and non-primetime queues
  - Run jobs during primetime or non-primetime only
  - sched_priv/sched_config

    primetime_prefix: p_

    nonprimetime_prefix: np_

    prime_spill: 0:00:00

    backfill_prime: false
  - sched_priv/holidays (AG 8.6)
- Dedicated queues
  - Run jobs during dedicated time only
  - sched_priv/sched_config

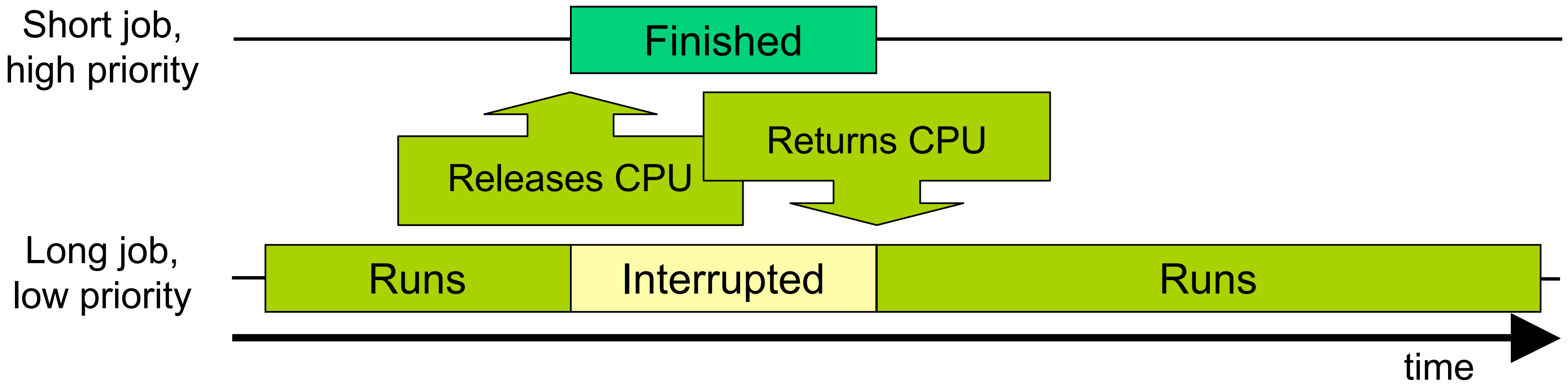
    dedicated_prefix: ded
  - Automatic backfilling at borders
  - sched_priv/dedicated time (AG 8.4)

Scheduling: More Time-dependent Scheduling

- Primetime and non-primetime scheduler configuration
  - AG 8.3
  - E. g. fair_share: true prime
fair_share: false non_prime
- Cron jobs
  - E. g. alter mom_priv/config and send SIGHUP to pbs_mom

Scheduling: Preemption

- Preempt a currently running job (preemptee) in order to run a high-priority job (preemptor).
- Methods:
  - Suspend and resume
  - Requeue and rerun
  - Checkpoint and restart
    - Use OS-level facility on SGI IRIX, Cray UNICOS
    - Use site-specific checkpoint facilities



Altair Engineering (www.altair.de), Altair Grid Technologies (www.pbspro.com), 2004

Scheduling: Preemption Details

- Configuration:
  - sched_priv/sched_config

    preemptive_sched: true

    preempt_queue_prio: 150

    preempt_prio: “express_queue, normal_jobs, server_softlimits”

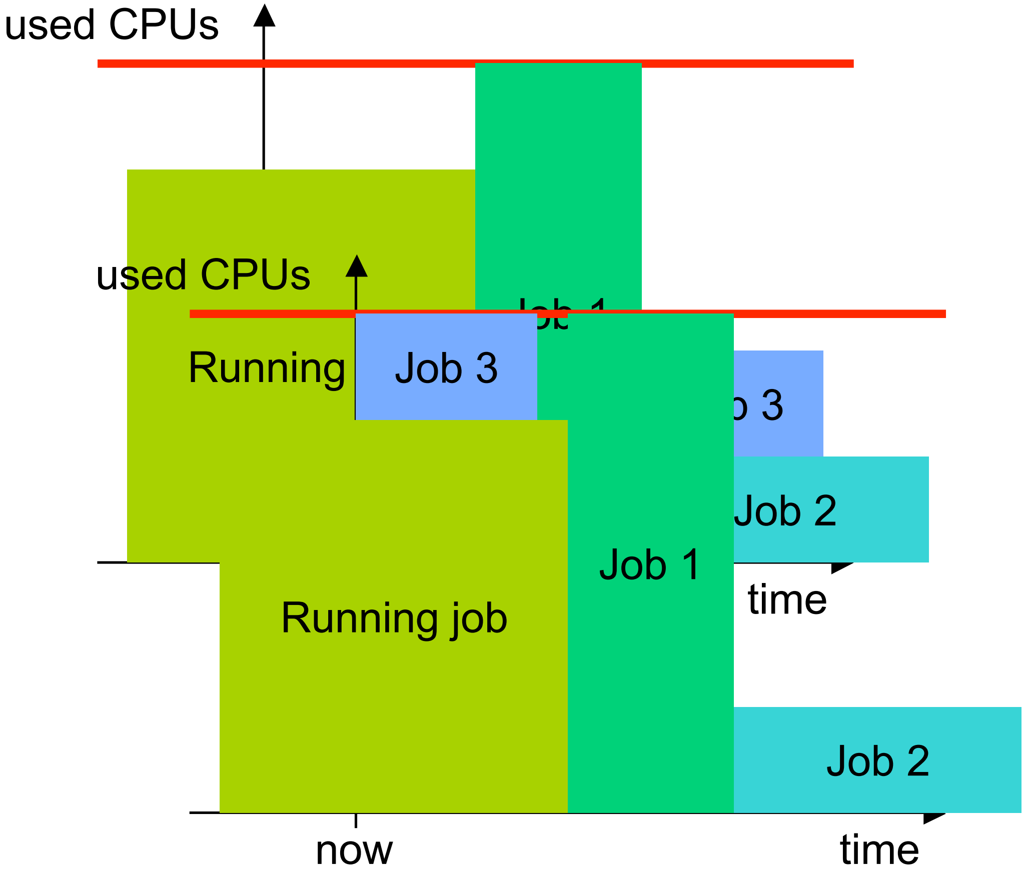
    preempt_order: “R 80 S”
- Every queue with a priority > preempt_queue_prio is an express queue.
- If there are not enough preemptees, then no job is preempted.
- Suspend and resume means send SIGSTOP and SIGCONT to all processes within the POSIX session of a job.
  - Reliable for single-node jobs only
  - SIGSTOP can be changed by $suspendsig in mom_priv/config
  - Consider using signal handlers
- Jobs running in an advance reservations cannot be preempted.

Scheduling: Backfilling

- Use “small amounts” of available resources for “small jobs” without delaying the most important “big job”.
- Applications:
  - Primetime and non-primetime borders
  - Dedicated and non-dedicated borders
  - Starving jobs
  - Advance reservations
- Configuration
  - sched_priv/sched_config

    backfill: true

    backfill_prime: true



Scheduling: Node Selection

- Limits on job number and resources
- Properties
- Load average
  - sched_priv/sched_config

    smp_cluster_dist: lowest_load
- Sort nodes on available resources
  - sched_priv/sched_config

    node_sort_key: mem low
  - any per-node resource can be used, plus sort_priority
  - multi-level key applied sequentially
- Order in server_priv/nodes

Scheduling: Node Grouping

- Build partitions based on a per-node resource
- Multi-node jobs will run within a single partition
- Advance reservations will reside within a single partition
- Configuration
  - Introduce per-node resource in server_priv/resourcedef:

    ```
    switch        type=string
    ```

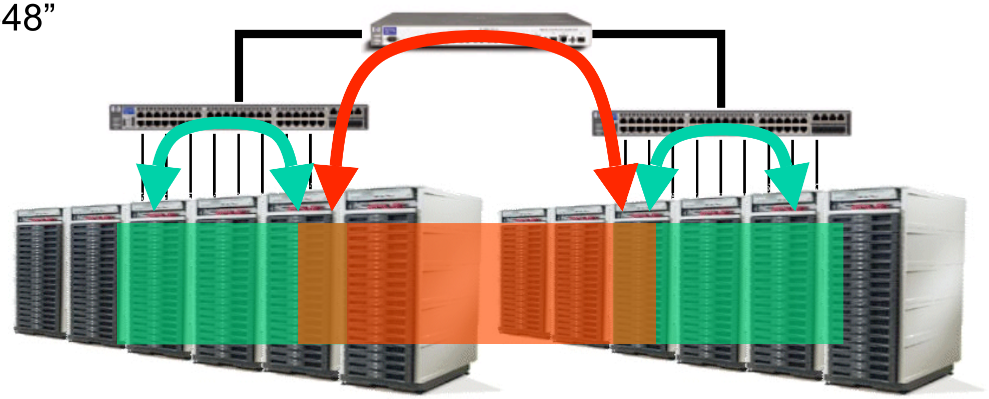
  - Assign value to resource for each node:

    ```
    active node “node01,node02,…,node24”
    s n resources_available.switch=sw1
    active node “node25,node26,…,node48”
    s n resources_available.switch=sw2
    …
    ```

  - Enable grouping in qmgr:

    ```
    s s node_group_enable=true
    s s node_group_key=switch
    ```

- Reference: AG 6.8



Scheduling: Starving Job Support

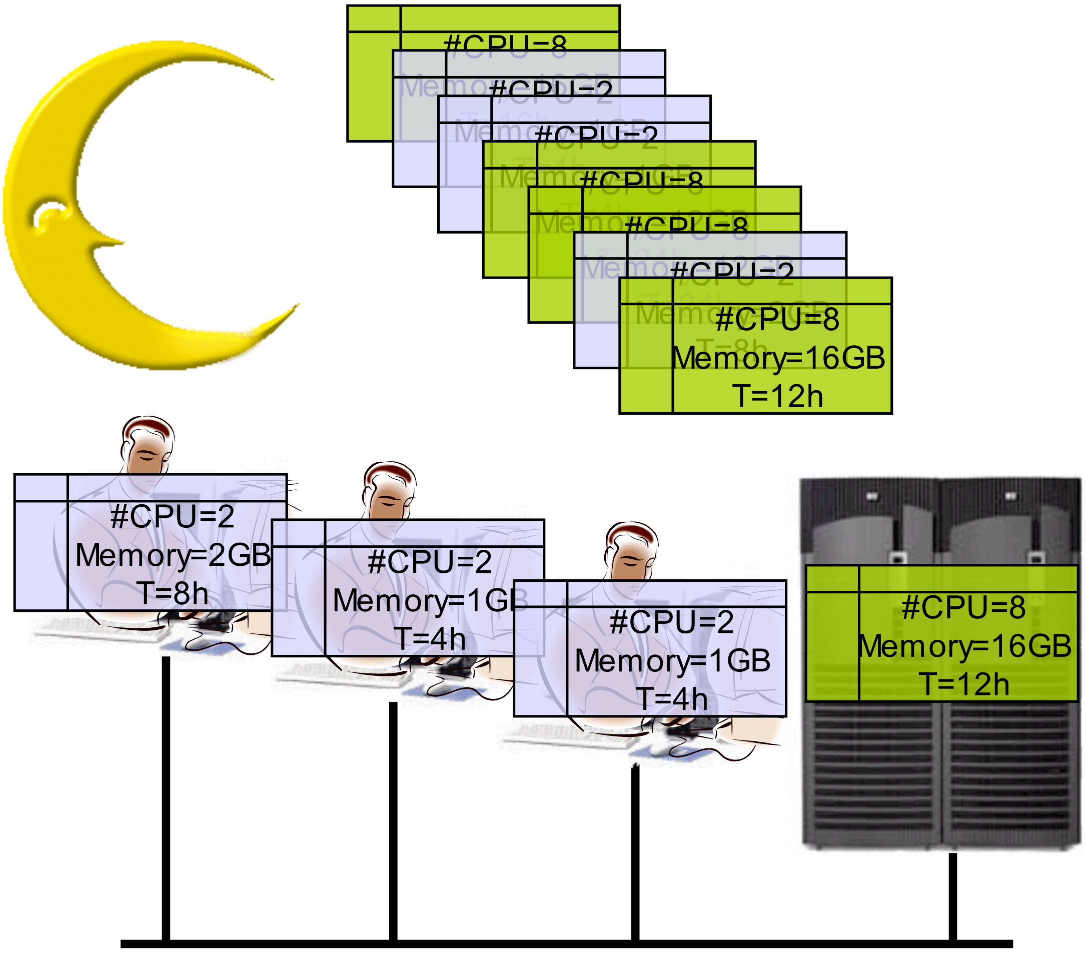
- Special support for jobs that wait very long for execution
  - Complex is drained until job can be started
  - Consider using backfilling
  - Starving jobs have their own preemption level, e. g.:

    preempt_prio: “starving_jobs, normal_jobs, fairshare”
- Configuration
  - sched_priv/sched_config

    ```
    help_starving_jobs: true
    max_starve: 24:00:00
    ```

Scheduling: Using Idle Workstations

- Use idle workstations to off-load small jobs from dedicated compute servers.
- Especially useful for large quantities of small and medium single-node jobs.
- Better support for multi-node jobs on workstation networks in 5.4.



Scheduling: Details on Using Idle Workstations

- Supported on most platforms, directly or via pbs_idled
- Default action, when node becomes non-idle:
  - singe-node jobs: suspend
  - multi-node jobs: do nothing
- Configuration
  - mom_priv/config

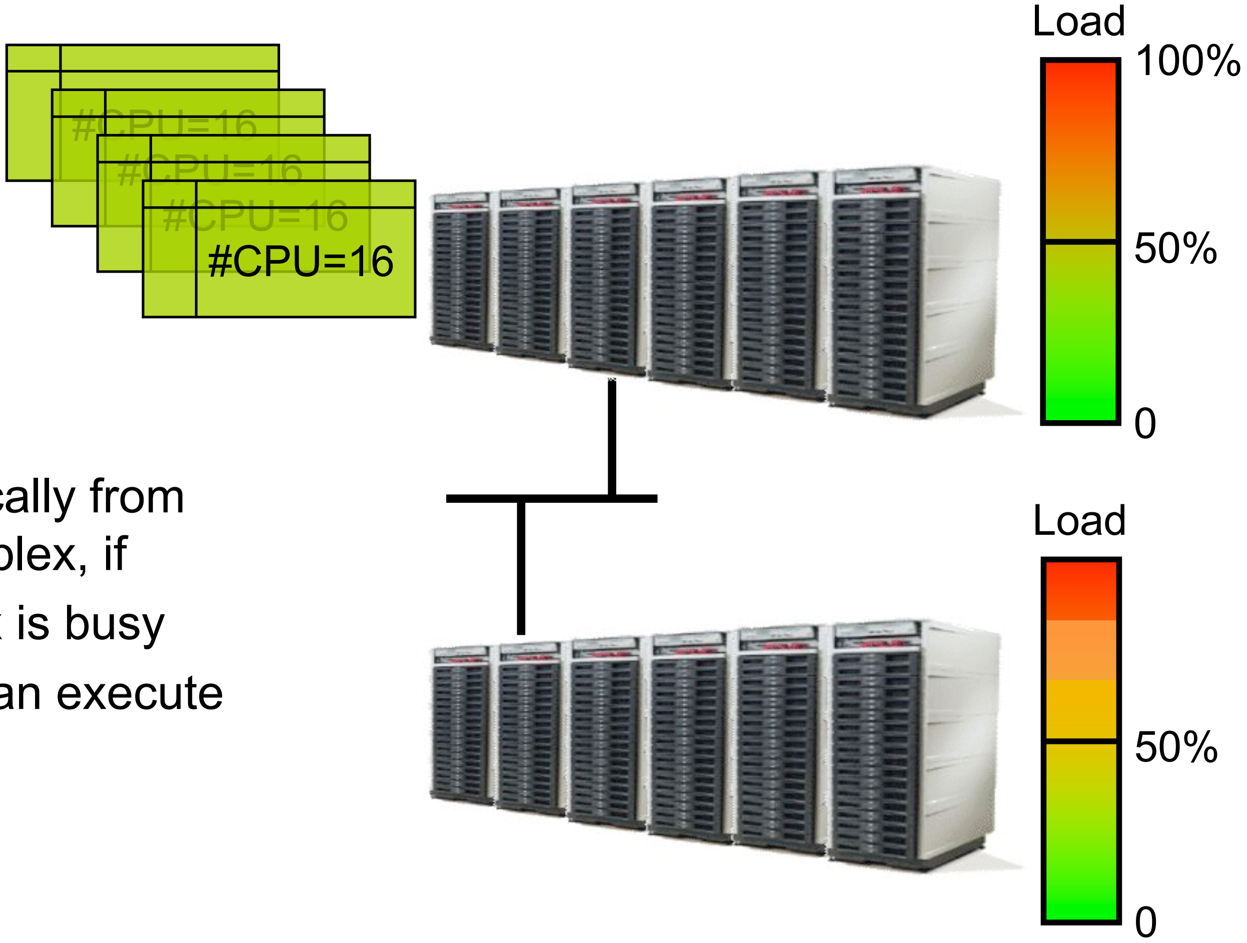
    $kbd_idle 1800 10 10

      - idle time, time to go non-idle, polling interval

    $action multinodebusy 0 requeue

      - requeues a multi-node job
      - non-rerunnable multi-node jobs are killed
- Reference: AG 7.7

Scheduling: Peer Scheduling



- Pulls jobs automatically from other PBS Pro complex, if
  - remote complex is busy
  - local complex can execute job immediately

Scheduling: Peer Scheduling Details

- Use same PBS Pro version on all peer complexes.
- Use flat UID scheme on all peer complexes.
- Make local root a manager of the remote PBS Pro complex (and vice versa).
- Map queues
  - sched_priv/sched_config

    peer_queue: “alienq workq@remote.domain”
- Local queue may be exclusive:
  - Allows special treatment of remote jobs, e.g. lower priority.
- Have peer queues for both remote servers in fail-over installations.
- Reference: AG 8.11

Scheduling: Advance Reservations

- Set of resources for specific users and for a limited period of time in the future.
- Act like a queue with ACLs and a life-time.
- Possible uses:
  - Interactive Debugging
  - Performance measurements
  - System maintenance
- Scheduler will backfill before advance reservations
- Jobs in advance reservations cannot be preempted.

Scheduling: Advance Reservation Details

- Node attribute to control reservations:
  - resv_enable
  - Defaults to true, if kbd_idle is not used, false otherwise
- Server attributes to control reservations:
  - Master switch: resv_enable
  - Control who can request advance reservations:
    - acl_resv_host_enable, acl_resv_hosts
    - acl_resv_group_enable, acl_resv_groups
    - acl_resv_user_enable, acl_resv_users
- If a advance reservation is requested by a user, then
  - Server checks the reservation related ACLs and resv_enable
  - Scheduler confirms or rejects the reservation
  - Server enables the reservation to accept jobs
  - Server starts the reservation at start time to execute jobs
  - Server kills jobs that run at end time
  - Server deletes the reservation

PBS Pro Training

- Logistics
- Outline
  - Altair and PBS Pro
  - Concepts and Terms
  - Anatomy of PBS Pro
  - Installation
  - Basic Configuration
  - Scheduling Strategies
  - Checking System Status
  - Log and Accounting Files
  - PBS Pro for Users

Checking System Status: Overview

- Daemon health
- Controlling the daemons
- Server and queue status
- Log files
- Job status
- Node status
- Accounting

Checking System Status: Daemon Status

] Check using the ps command:

```
# ps -eaf |grep pbs_
root 3428    691: /pbs/sbin/pbs_mom -r
root 3429    6:40 /pbs/sbin/pbs_sched
root 3430   20:32 /pbs/sbin/pbs_server
root 1808    0:00 grep pbs_
```

] Which daemons should be running?

	Server	Scheduler	Mom
primary server	yes	yes	(yes)
secondary server	yes	no	(yes)
execution node	no	no	yes
submit host	no	no	no

Checking System Status: Stopping Daemons

- Use /etc/init.d/pbs stop
- Use qterm to stop daemons
  - Terminate both failover servers: -f
  - Terminate all moms: -m
  - Terminate scheduler: -s
  - Termination type: -t immediate|delay|quick
    - Checkpoint, requeue rerunnable, kill non-rerunnable jobs: immediate
    - Checkpoint, requeue rerunnable, leave non-rerunnable jobs: delay
    - Leave running jobs in their state: quick (default)
  - Reference: AG 10.2.5
- Use the kill command
  - Consider SIGKILL and SIGTERM
  - See AG 10.2.7

Checking System Status: Starting Daemons

- Use /etc/init.d/pbs start
- Start daemons manually
  - Mom: ${PBS_EXEC}/sbin/pbs_mom
    - Poll for left-over jobs: -p
    - Kill left-over jobs: -r
    - Reference: AG 10.2.1
  - Server: ${PBS_EXEC}/sbin/pbs_server
    - After shutdown using qterm -t immediate: -t hot
    - After shutdown using qterm –t quick: -t warm (default)
    - Reference: AG 10.2.2
  - Scheduler: ${PBS_EXEC}/sbin/pbs_sched
    - Reference: AG 10.2.3

Checking System Status: Using qstat

- Ask for server status:
  - qstat –B

```
Server           Max   Tot   Que   Run   Hld   Wat   Trn   Ext Status
---------------- ----- ----- ----- ----- ----- ----- ----- ----- -----------
altgtest1            0     9     1     8     0     0     0     0 Active
```

  - qstat –Bf

- Ask for queue status:
  - qstat –Q

```
Queue            Max   Tot Ena Str   Que   Run   Hld   Wat   Trn   Ext Type
---------------- ----- ----- --- --- ----- ----- ----- ----- ----- ----- ----
high               0     0  no  no     0     0     0     0     0     0 Exec
long               0     8 yes yes     0     8     0     0     0     0 Exec
short              0     0 yes yes     0     0     0     0     0     0 Exec
medium               0     0 yes yes     0     0     0     0     0     0 Exec
```

  - qstat –Qf

Checking System Status: Server Log

- Verbosity controlled by log_events server attribute:
  - 0 means log nothing
  - 511 means log everything
  - Most useful settings: 63 or 127
  - Effective immediately after change in qmgr
  - Reference: AG 10.12
- One log file per day
  - server_logs/yyyymmdd
  - Accessible for root
  - Included in tracejob output

Checking System Status: Scheduler Log

- Verbosity controlled by log_filter in sched_priv/sched_config
  - 0 means log everything
  - 511 means log nothing
  - Reversed logic w.r.t. server and mom: This is a filter!
  - Send SIGHUP to pbs_sched after changes to sched_priv/sched_config
  - Reference: AG 10.12
- One log file per day
  - sched_logs/yyyymmdd
  - Accessible for root
  - Included in tracejob output

Checking System Status: Mom Log

- Verbosity controlled by $logevent in mom_priv/config
  - 0 means log nothing
  - 511 means log everything
  - Most useful settings: 63 or 127
  - Send SIGHUP to pbs_mom after changes to mom_priv/config
  - Reference: AG 10.12
- One log file per day
  - mom_logs/yyyymmdd
  - Accessible for root
  - Included in tracejob output

Checking System Status: Accounting Log

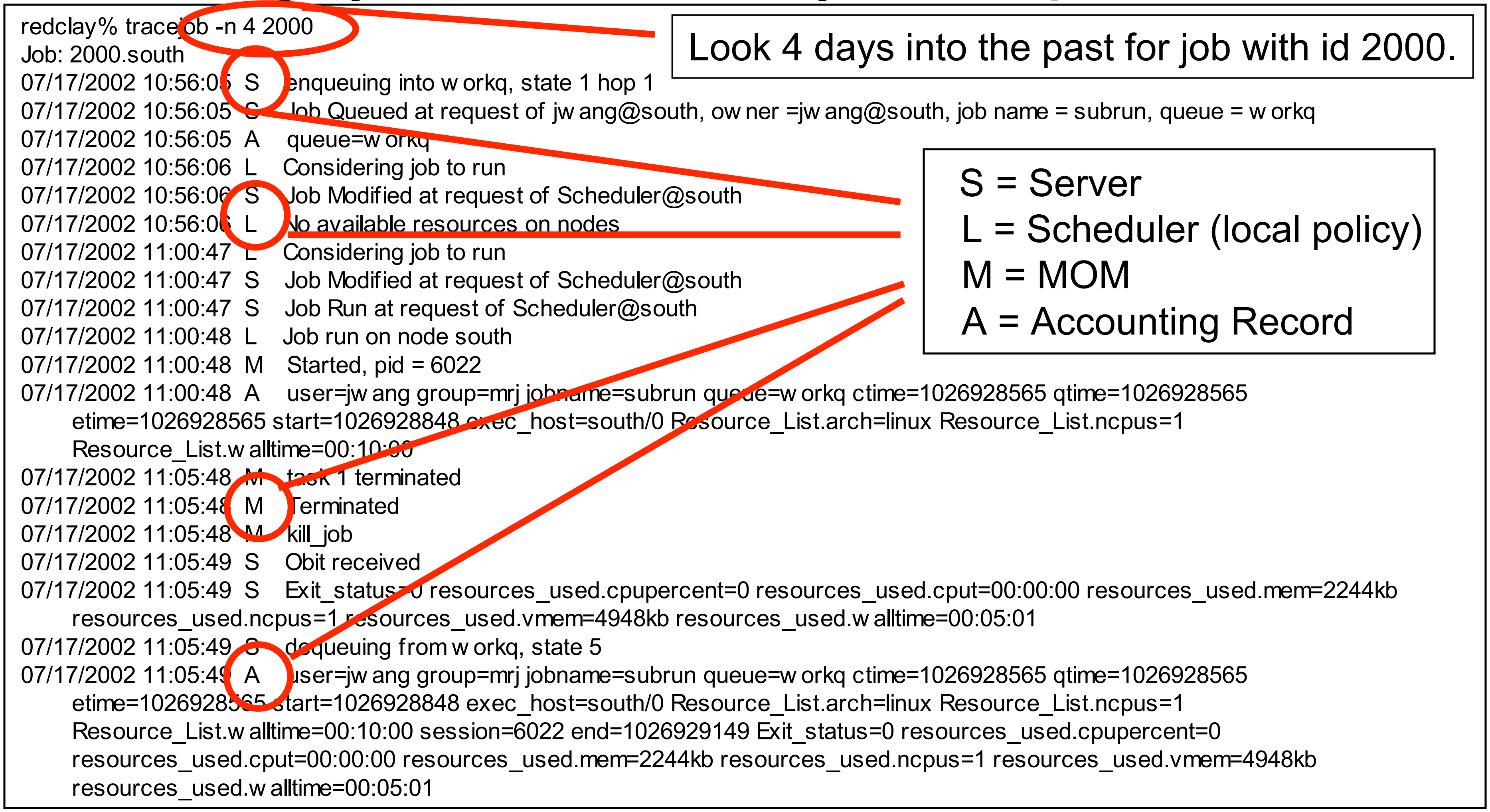
- One log file per day:
  - server_priv/accounting/yyyymmdd
- Format:
  - Date time;record_type;job_id;message_text
  - Date and time stamp: mm/dd/yyyy hh:mm:ss
  - Record type: Single character

    A - job was aborted by the server
    D - job was deleted by request
    E - job ended (terminated execution)
    C - job was checkpointed and held
    Q - job entered a queue
    R - job was rerun
    S - job execution started
    T - job was restarted from a checkpoint file
  - Job identifier
  - Message text: ASCII text string (whose content depends on the record type) in the format: blank separated keyword=value

Checking System Status: Job Status

- Monitor jobs in the complex:
  - Overview: qstat
  - Alternate Overview: qstat -a
  - Display nodes associated with jobs: qstat –n
  - Display job comments: qstat –s
  - Display using a single line per job: qstat –n1 (Bug!)
- Extensive Information concerning jobs:
  - qstat –f …
- Display history of a job:
  - tracejob [-n x] …
  - For users: server, scheduler, mom logs
  - For root: server, scheduler, mom, accounting logs

Checking System Status: tracejob Example



```
redclay% tracejob -n 4 2000
Job: 2000.south
07/17/2002 10:56:05  S    enqueuing into w orkq, state 1 hop 1
07/17/2002 10:56:05  S    Job Queued at request of jw ang@south, ow ner =jw ang@south, job name = subrun, queue = w orkq
07/17/2002 10:56:05  A    queue=w orkq
07/17/2002 10:56:06  L    Considering job to run
07/17/2002 10:56:06  S    Job Modified at request of Scheduler@south
07/17/2002 10:56:06  L    No available resources on nodes
07/17/2002 11:00:47  L    Considering job to run
07/17/2002 11:00:47  S    Job Modified at request of Scheduler@south
07/17/2002 11:00:47  S    Job Run at request of Scheduler@south
07/17/2002 11:00:48  L    Job run on node south
07/17/2002 11:00:48  M    Started, pid = 6022
07/17/2002 11:00:48  A    user=jw ang group=mrj jobname=subrun queue=w orkq ctime=1026928565 qtime=1026928565
      etime=1026928565 start=1026928848 exec_host=south/0 Resource_List.arch=linux Resource_List.ncpus=1
      Resource_List.w alltime=00:10:00
07/17/2002 11:05:48  M    task 1 terminated
07/17/2002 11:05:48  M    Terminated
07/17/2002 11:05:48  M    kill_job
07/17/2002 11:05:49  S    Obit received
07/17/2002 11:05:49  S    Exit_status=0 resources_used.cpupercent=0 resources_used.cput=00:00:00 resources_used.mem=2244kb
      resources_used.ncpus=1 resources_used.vmem=4948kb resources_used.w alltime=00:05:01
07/17/2002 11:05:49  S    dequeuing from w orkq, state 5
07/17/2002 11:05:49  A    user=jw ang group=mrj jobname=subrun queue=w orkq ctime=1026928565 qtime=1026928565
      etime=1026928565 start=1026928848 exec_host=south/0 Resource_List.arch=linux Resource_List.ncpus=1
      Resource_List.w alltime=00:10:00 session=6022 end=1026929149 Exit_status=0 resources_used.cpupercent=0
      resources_used.cput=00:00:00 resources_used.mem=2244kb resources_used.ncpus=1 resources_used.vmem=4948kb
      resources_used.w alltime=00:05:01
```

Look 4 days into the past for job with id 2000.

S = Server
L = Scheduler (local policy)
M = MOM
A = Accounting Record

Checking System Status: Node Status

- Monitor nodes with pbsnodes
  - List all nodes: pbsnodes –a
  - List one node or some nodes: pbsnodes …
  - List nodes marked in some way: pbsnodes -l

- Change status of nodes
  - Set nodes offline: pbsnodes –o …
  - Clear offline status: pbsnodes –c …

- Use xpbsmon
- Use some additional monitoring software
  - Ganglia (http://ganglia.sourceforge.net/)
  - Nagios (http://www.nagios.org/)

Checking System Status: pbs-report

- pbs-report is a Perl script installed in ${PBS_EXEC}/sbin
- Parses accounting logs and extracts information
- Use pbs-report –help or pbs-report –man to get help
- Reference: AG 11.20
- Build your own accounting scripts

Getting Help: PBS Pro Support

- Call
  - Ralf Eichmann: +49 7031 6208 39
  - Altair Germany support: +49 7031 6208 22
- Send Email
  - Ralf Eichmann: eichmann@altair.de
  - Altair Germany PBS Pro support: pbssupport@altair.de
  - Altair Germany support: support@altair.de
  - Altair PBS Pro support: pbssupport@altair.com
- More contact information at p2 in AG and UG.
- Have available:
  - qstat –Bf, qstat –Qf, pbsnodes –a, qstat -f output
  - log files of mom, server, scheduler
  - /etc/pbs.conf
  - Other relevant files (core, stdout, stderr,…)

PBS Pro Training

- Logistics
- Outline
  - Altair and PBS Pro
  - Concepts and Terms
  - Anatomy of PBS Pro
  - Installation
  - Basic Configuration
  - Scheduling Strategies
  - Checking System Status
  - Log and Accounting Files
  - PBS Pro for Users

PBS Pro for Users: System Status

- Use qstat
  - For server status: qstat –B
  - For queue status: qstat –Q
  - For job monitoring: qstat
qstat –f
qstat –s
qstat –a
qstat –n
- Use tracejob
  - No accounting information

PBS Pro for Users: Creating a Job

- Supported job types
  - batch and interactive jobs
  - single-node and multi-node jobs
  - non-blocking and blocking jobs
- Job submission
  - Job is described in a script (Unix shells, Perl, Python, etc.)
  - Information for PBS Pro in comment lines
  - Returns job id
- Sample:

```
#!/bin/sh
#PBS -l walltime=1:00:00
#PBS -l mem=400mb
#PBS -l ncpus=4
#PBS -j oe
cd ${PBS_O_WORKDIR}
./subrun
```

PBS Pro for Users: Submission Details

- Information on qsub options:
  - Man page
  - UG 4
- Information on local resources:
  - Standard resources: UG 4
  - Look at
    - qstat –Bf
    - pbsnodes –a
  - Ask administrator
  - Request by #PBS –l … in a job script
- Jobs may be submitted using “here documents” and ^D
- Job options may be specified as options to the qsub command line
  - These have precedence over options in the job script

PBS Pro for Users: Common Submission Options

Job name	#PBS –N name	
Job destination	#PBS –q queue@server	
Stdout destination	#PBS –o file_name	
Stderr destination	#PBS –e file_name	
Join stdout and stderr	#PBS –j	
Email notification	#PBS –m abe	
Expand variables	#PBS –v variable=value, variable2=value2	
Export all variables	#PBS –V	
Mark job rerunnable	#PBS –r n	(default: y)
Shell	#PBS –S shell	
Priority	#PBS –p 400	(-1023 … +1023)
Deferring execution	#PBS –a 200402262330	
Hold job	#PBS -h	

PBS Pro for Users: Boolean Resource Requests

- Boolean resource requests are supported for single-node jobs
- Use #PBS –l resc="…"
- Examples:
  - Use logical operators || && ==
    #PBS -l resc="((arch==hpux10) || (arch==irix6)) && (mem=1500mb)"
  - Establish preference by using more than one –l resc line:
    #PBS -l resc="(ncpus=16) && (walltime=1:00)“
    #PBS -l resc="(ncpus=8) && (walltime=2:00)“
    #PBS -l resc="(ncpus=4) && (walltime=4:00)“
- Note difference between "==“ (comparison) and "=“ (assignment)
- Relational operators > >= < <= are supported

PBS Pro For Users: Single-node and Multi-node jobs

- Single-node jobs
  - no #PBS –l nodes=… statement
  - All resources must be available on a single node
  - All properties and resources supported
  - Boolean resource requests supported
  - Beware of server or queue defaults, that turn jobs into multi-node jobs
- Multi-node jobs (see UG 9)
  - Use #PBS –l nodes=…
  - Supported per-node resources:
    - ncpus: processor count
    - ppn: processes per node
    - cpp: cpus per process
  - Properties and server-level resources supported
  - No boolean resource requests

PBS Pro for Users: Interactive and Blocking Jobs

- Interactive jobs:
  - qsub –I …
  - When resources are available, then the job will be run
  - Once started the job appears as a login session on the system
  - Resource limits will be enforced by PBS Pro

- Blocking submission
  - qsub –W block=true …
  - qsub will wait for the job to complete and exit with the job’s exit code
  - If SIGHUP, SIGINT, SIGQUIT, SIGTERM is received, then exit code 2
  - If the job is deleted before termination, then exit code 3
  - Applications:
    - Make dependencies
    - Custom process management software

PBS Pro for Users: Job Dependencies

- Establish dependencies among jobs
- Concurrent execution
  - First job: qsub –W depend=synccount:job_number …
  - Following jobs: qsub –W depend=syncwith:job_id …
- Before and after dependencies:
  - qsub –W depend=dependency_list:job_id:job_id…
  - Supported dependencies:
    - after
    - afterok, afternotok, afterany
    - before
    - beforeok, beforenotok, beforeany

PBS Pro for Users: Other Job Submission Options

- File staging
  - #PBS –W stagein=input@frontend:/home/tom/parameter1.dat
  - #PBS –W stageout=output@frontend:/home/tom/result1.dat
  - @ is separator between local file (on exec node) and remote file
  - Translated to rcp (or scp) calls

- Umask of stdout and stderr files
  - #PBS –W umask=022
  - Allow other people to view these files

- Suppress job identifier
  - #PBS -z

PBS Pro for Users: Altering and Deleting a Job

- Alter a job
  - Queued jobs: Most attributes can be changed
  - Running jobs: Resource limits cannot be changed (cput, walltime, ncpus, mem,…)
  - qalter has same options as qsub

- Delete a job
  - qdel job_id
  - Delay SIGKILL after SIGTERM: qdel –W delay 30 job_id
  - Delete job even if execution node cannot be contacted:
    qdel –W force job_id

PBS Pro for Users: Advance Reservations

- Submission examples:
  - pbs_rsub –R 1400 –E 1600 –l nodes=8 –U “tom@*”
  - pbs_rsub –R 1400 –D 08:00:00 –l ncpus=16 –G “cfd@*”
  - Pay attention to server defaults
  - Reference: UG 8.9
- Scheduler confirms or rejects reservation
- Scheduler enables confirmed reservations: Allows submission
- Use reservation id like a queue name
- Server starts confirmed reservations at start time: Jobs run
- Server deletes reservation (and jobs) at end time
- View reservations: pbs_rstat
  - Brief view: pbs_rstat –B
  - Extended view: pbs_rstat –f
- Delete reservation
  - pbs_rdel resv_id

PBS Pro for Users: Further Commands

- Hold and release waiting jobs
  - Restrict consideration of a job for scheduling
  - qhold, qrls
- Send a message to a job
  - Writes a message into the stdout or stderr file of a job
  - qmsg –O|-E “message_string” job_id
- Change order of two waiting jobs
  - qorder job_id1 job_id2
- Send a signal to a job
  - Application or job script should trap and process signal
  - qsig –s signal job_id
- Move a queued job to another destination
  - qmove queue@server job_id

Thank
you!

Additional Topics

- Advanced Configuration
  - Prologue and epilogue scripts
  - Mom limit enforcement
  - Job checkpointing
  - Jobs on failed nodes
  - Using scp
  - SGI Altix support
  - Globus support
- Graphical interfaces
- Troubleshooting
- Security
- PBS Pro software upgrade
- PBS Pro on Windows

Advanced Configuration: Prologue and Epilogue Scripts

- Prologue and Epilogue are scripts run by pbs_mom before and after a job
  - Name: mom_priv/prologue and mom_priv/epilogue
  - Run as root
  - Many arguments passed automatically by pbs_mom
  - Configure timeout for prologue and epilogue:
    - $prologalarm in mom_priv/config
    - Default: 30 seconds
  - Reference: AG 10.11
- Possible uses:
  - Cleanup MPI environment
  - Deliver files and cleanup scratch disk
  - Release application software licenses

Advanced Configuration: Mom Limit Enforcement

- pbs_mom may enforce the requested resource limits
- Configuration:
  - $enforce metric in mom_priv/config where metric is one of:
    - For memory: mem (default: off)
    - Absolute CPU usage: cpuaverage (default: off)
    - Weighted moving CPU usage: cpuburst (default: off)
  - See AG 7.8 and 7.9

Advanced Configuration: Job Checkpointing

- PBS Pro 5.4 supports site-specific checkpoint and restart through site-specific scripts executed by pbs_mom
- Configuration in mom_priv/config
  - periodic checkpointing: $action checkpoint time_out script args
  - checkpoint before server shutdown:
    $action checkpoint_abort time_out script args
  - restart: $action restart time_out script args
  - $restart_background true|false
  - $restart_transmogrify true|false
- Reference: AG 7.6
- For Linux systems:
  - BLCR (http://ftg.lbl.gov/checkpoint)
  - Meiosys Metacluster (http://www.meiosys.com/)
- Application-specific checkpointing (LS-DYNA, Pam-Crash, FLUENT, CFX, …)

Advanced Configuration: Jobs on Failed Nodes

- PBS Pro detects failed nodes automatically
- Default behavior: Jobs are left running, because there might be only a network issue
- Configure automatic requeueing using qmgr:
  - s s node_fail_requeue=300
  - Non-rerunnable jobs are killed
  - Reference: AG 6.4
- Manual intervention
  - To delete a job: qdel –W force job_id
  - To requeue a job: qrerun –W force job_id
    - Non-rerunnable jobs are killed
  - Reference: AG 13.5

Advanced Configuration: Using scp

] Using scp for file delivery

- Set up keys to allow password-less scp
- Configuration in /etc/pbs.conf

```
PBS_SCP=/usr/bin/scp
```

Advanced Configuration: SGI Altix Support

- SGI Altix is a cell-based SMP, i.e. non-uniform memory access
- Run jobs in “cpusets” on a single Altix node for optimal performance
- ProPack 2.2 required (look at ls /usr/lib/libcpuset.so*)
- Run pbs_mom.cpuset instead of pbs_mom
- Configure in mom_priv/config
- Exclusive cpusets
    - For bigger jobs (>= 1 node board, i.e. 2p and associated memory)
    - Both number of processors and memory are considered
    - cpuset_create_flags
- Shared cpusets
    - For small jobs
    - max_shared_nodes, cpuset_small_ncpus, cpuset_small_mem
- Reference: AG 7.10

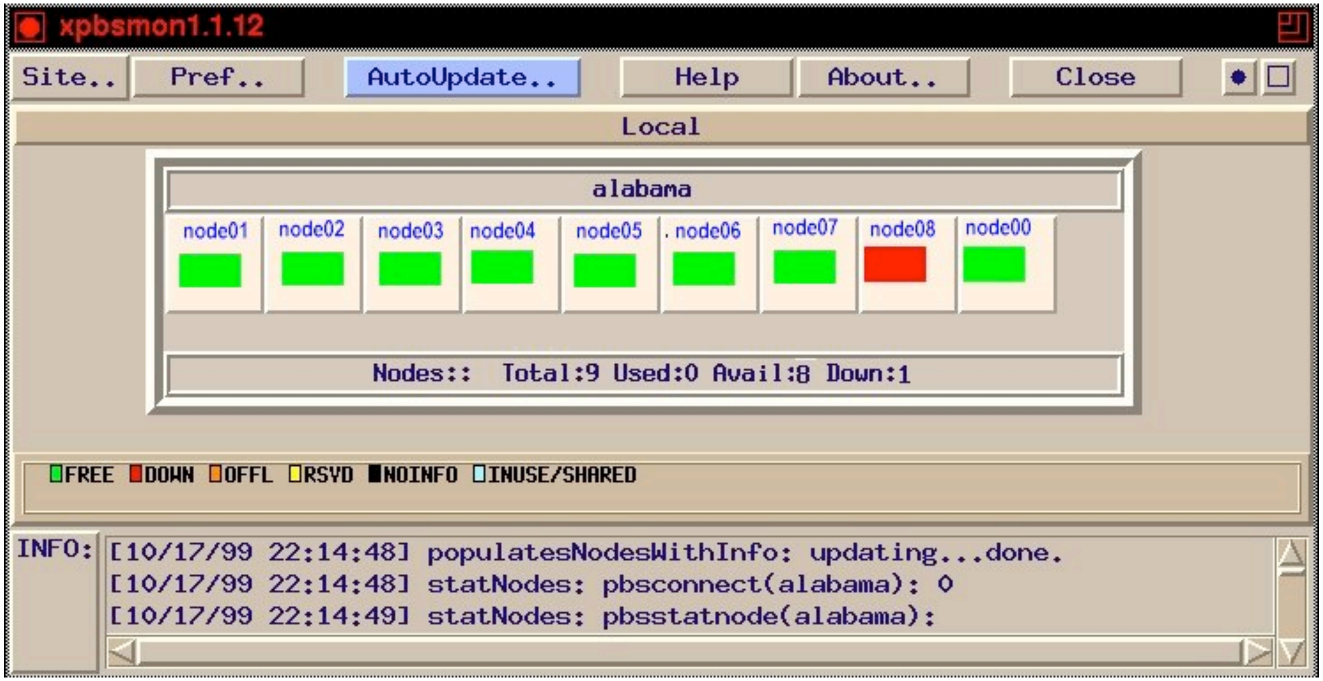
Advanced Configuration: Globus Support

- Globus: Standardized software for geographically distributed infrastructures, see www.globus.org
- Additional pbs_mom_globus required, not included in binary distribution
- Has its own configuration in mom_globus_priv and its own logs in mom_globus_logs
- Establish security infrastructure, e.g. generate proxy certificates
- Submit jobs to Globus via PBS Pro:
  - Specify gatekeeper via
    #PBS –l site=globus:globus-resource-name
  - Resource requirements and job status flags are translated
  - File staging through Global Access to Secondary Storage (GASS)

- Reference: UG 8.8, AG 6.14, 7.11

Graphical Interfaces: xpbsmon

- Quick overview of nodes status
- Configurable to show two or more complexes

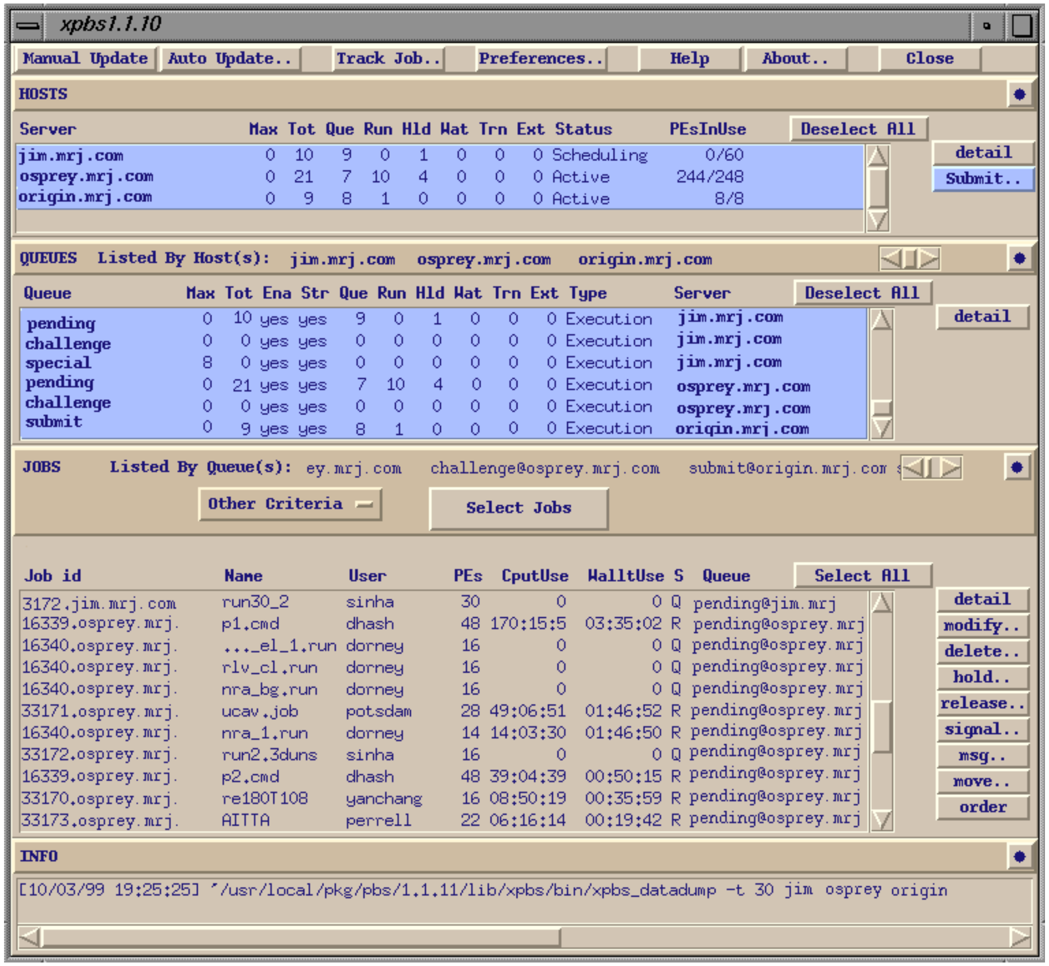


```
Nodes::  Total:9 Used:0 Avail:8 Down:1
```

```
[10/17/99 22:14:48] populatesNodesWithInfo: updating...done.
[10/17/99 22:14:48] statNodes: pbsconnect(alabama): 0
[10/17/99 22:14:49] statNodes: pbsstatnode(alabama):
```

Graphical Interfaces: xpbs

- Quick overview of server and queue status
- Allows job submission
- Supports most job operations
- Admin mode for queue operation and extended job operation:
  - xpbs -admin



Server	Max	Tot	Que	Run	Hld	Wat	Trn	Ext	Status	PEsInUse
jim.mrj.com	0	10	9	0	1	0	0	0	Scheduling	0/60
osprey.mrj.com	0	21	7	10	4	0	0	0	Active	244/248
origin.mrj.com	0	9	8	1	0	0	0	0	Active	8/8

Queue	Max	Tot	Ena	Str	Que	Run	Hld	Wat	Trn	Ext	Type	Server
pending	0	10	yes	yes	9	0	1	0	0	0	Execution	jim.mrj.com
challenge	0	0	yes	yes	0	0	0	0	0	0	Execution	jim.mrj.com
special	8	0	yes	yes	0	0	0	0	0	0	Execution	jim.mrj.com
pending	0	21	yes	yes	7	10	4	0	0	0	Execution	osprey.mrj.com
challenge	0	0	yes	yes	0	0	0	0	0	0	Execution	osprey.mrj.com
submit	0	9	yes	yes	8	1	0	0	0	0	Execution	origin.mrj.com

Job id	Name	User	PEs	CputUse	WalltUse	S	Queue
3172.jim.mrj.com	run30_2	sinha	30	0	0	Q	pending@jim.mrj
16339.osprey.mrj.	p1.cmd	dhash	48	170:15:5	03:35:02	R	pending@osprey.mrj
16340.osprey.mrj.	..._el_1.run	dorney	16	0	0	Q	pending@osprey.mrj
16340.osprey.mrj.	rlv_cl.run	dorney	16	0	0	Q	pending@osprey.mrj
16340.osprey.mrj.	nra_bg.run	dorney	16	0	0	Q	pending@osprey.mrj
33171.osprey.mrj.	ucav.job	potsdam	28	49:06:51	01:46:52	R	pending@osprey.mrj
16340.osprey.mrj.	nra_1.run	dorney	14	14:03:30	01:46:50	R	pending@osprey.mrj
33172.osprey.mrj.	run2.3duns	sinha	16	0	0	Q	pending@osprey.mrj
16339.osprey.mrj.	p2.cmd	dhash	48	39:04:39	00:50:15	R	pending@osprey.mrj
33170.osprey.mrj.	re180T108	yanchang	16	08:50:19	00:35:59	R	pending@osprey.mrj
33173.osprey.mrj.	AITTA	perrell	22	06:16:14	00:19:42	R	pending@osprey.mrj

```
[10/03/99 19:25:25] '/usr/local/pkg/pbs/1.1.11/lib/xpbs/bin/xpbs_datadump -t 30 jim osprey origin
```

Troubleshooting PBS Pro

- Server@thunder: Permission denied (13) in chk_file_sec, Security violation “/usr/spool/PBS/server_priv/jobs/” resolves to “/var”
  - Check file and directory permissions, i.e. on /var
  - See AG appendix C
  - Use pbs_probe (AG 11.3)
- Some submit hosts work, others do not
  - Check ACLs on server, queue, and reservations using qmgr
  - Check flat_uid using qmgr
- Some execution nodes do not execute jobs
  - Check mom and server logs for communication problems
  - Check licensing using pbsnodes
- Can’t see other people’s jobs
  - query_other_jobs server attribute set to false

Troubleshooting PBS Pro: Jobs Don’t Run

- Jobs don’t run
  - Look at
    - qstat –f job_id
    - tracejob job_id
  - Check user and group ACLs
  - Check daemons
  - Check scheduling server attribute
  - Check epilogue script and epilogue alarm time
  - “Hop count exceeded”: Loops in routing queue setup
  - “Job rejected by all possible destinations”: Invalid resource requirement
  - “No destination”: Server has no default_queue

Troubleshooting PBS Pro: File Delivery

] Stdout and stderr files are not delivered
- Check mom’s log file for reason
- Check /etc/hosts.equiv and user’s .rhosts file
- Check user authorization
- Check directories and permission (${PBS_HOME}/spool, and target)
- User’s login files on destination node do terminal action
  - Check for interactive login or batch job (UG 3.5)

```
if ( ! $?PBS_ENVIRONMENT ) then
 do terminal settings here
 run command with output here
endif
```

- Check PBS_RCP and PBS_SCP variables in /etc/pbs.conf

Troubleshooting PBS Pro: Job Exit Codes

- Any code 0 or greater (positive) is the return code of the top level shell.
- Negative codes are set by PBS:

-1	Job could not be executed, problem occured before the standard output/error files were created; the reason can typically be found in Mother Superior's log
-2	Job could not be executed, problem occured after the standard job files were created and the error message can be found in stderr.
-3	Job could not be executed, the reason is likely temporary and the job will be requeued
-4	Job terminate when Mom was restarted
-5	Job terminate when Mom was restarted, there is a checkpoint image
-7	Job could not be restarted from the checkpoint image
-10	User's UID was invalid/not found
-11	Job was rerun (qrerun)
-12	Job was checkpointed (and killed)

Security in PBS Pro

- Internal security
  - File and directory permissions
  - Security in the daemon's environment
- Host authentication
  - Uses credentials, and check host name and IP adress
- User authentication
  - Uses credentials
- Host, user, and group authorization
  - ACLs
  - mom_priv/config entries $clienthost and $restricted
  - User and group mapping, or flat uid scheme configured
- External security
  - Uses manager, operator, and user levels of privilege

Security in PBS Pro: Root Owned Jobs

- By default, root-owned jobs are not executed
- Configuration using qmgr:

```
s s acl_roots=“root@server1,root@server2”
```

Updating PBS Pro

- Overlay upgrade
  - Replace PBS Pro executables, retain PBS Pro database
  - Stop all daemons, install new software, start daemons
  - Most new versions of PBS Pro for Unix support overlay upgrades
    - 5.2.x to 5.4 or 5.3.x to 5.4
  - Reference: AG 5.1
- Migration upgrade
  - Replace both PBS Pro executables and database
  - Requires moving jobs to new installation
  - Requires duplicating configuration
  - Required for
    - Upgrade from older PBS Pro version for Unix (See Release Notes)
    - Upgrade PBS Pro for Windows
  - Reference: AG 5.2

PBS Pro on Windows Systems

- Supported operating systems:
  - Windows 2000, Windows XP, Windows 2003, professional and server versions, with recent service packs
- Main caveats
  - User Authentication
    - Use common user names
  - File management
    - Additional pbs_rshd daemon for rcp file movement
- Mixed Unix/Windows installation not recommended
- Reference: Release Notes, AG 4.5.1, 6.15.3, 10.3, 10.4, 13.9-13.12

- Supplementary software
  - Microsoft Services for Unix 3.5 (http://www.microsoft.com/windows/sfu/)