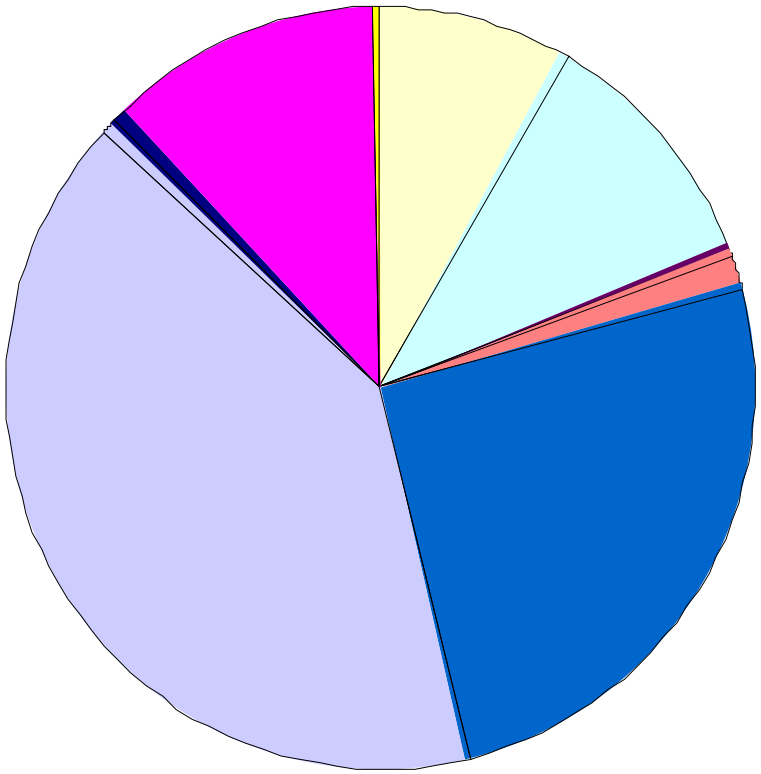
# Grid Engine - A Batch System for DESY

*Andreas Haupt,*
*Peter Wegner*
*15.6.2005*
*DESY Zeuthen*

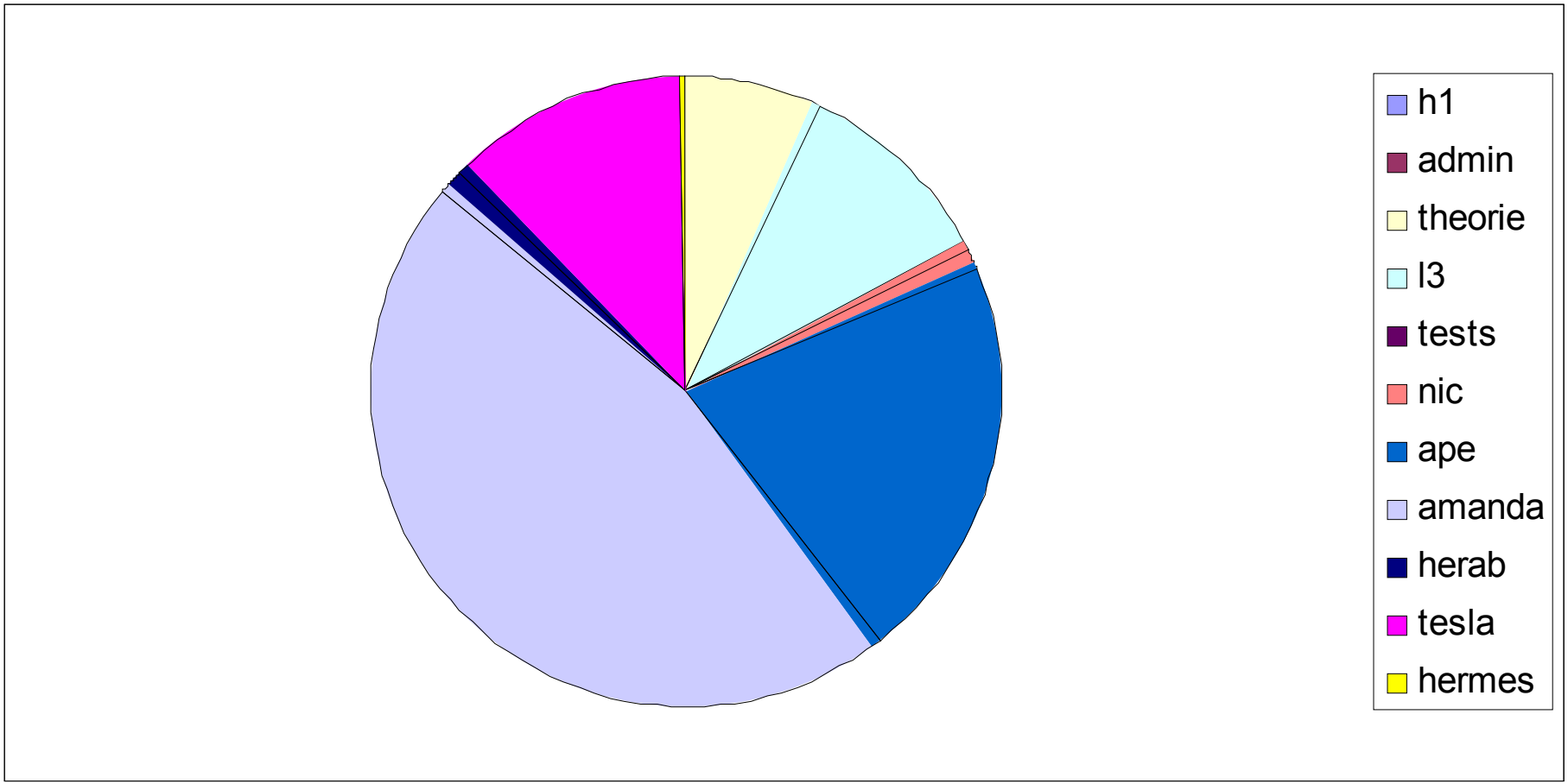
# Introduction

- **Motivations for using a batch system**
  - more effective usage of available computers (e.g. reduce idle cpus dramatically)
  - usage of resources 24h/day
  - assignment of resources according to policies (who gets how much CPU when)
  - quicker execution of tasks (system knows most powerful least loaded nodes)
- **Our goal:**
  - You tell the batch system a script name (preferred) or executable and what you need in terms of disk space, memory, CPU time
  - The batch system guarantees fastest possible turnaround

# PC Farm Usage 2003, cpu time



# PC Farm Usage 2003, wallclock time



# Benefits using the GE Batch System

- **For users:**
  - jobs get executed on the most suitable (least loaded, fastest) machine
  - fair scheduling according to defined sharing policies
  - no one can overuse the system and provoke system degradation
  - users need no knowledge of host names and queues where their jobs can run
- **For administrators:**
  - one time allocation of resources to users, projects, groups
  - no manual intervention to guarantee policies
  - reconfiguration of the running system (to adapt to changing usage pattern)
  - easy monitoring of hosts and jobs

# The Grid Engine Batch System

## Components of the system

- **Queues**
  - contain information on number of jobs and job characteristics that are allowed on a given host
  - Jobs need to fit into a queue to get executed.
- **Resources**
  - Features of hosts or queues that are known to GE.
  - Resource attributes are defined in so called (host, queue and user defined) complexes
- **Projects**
  - contain lists of users (usersets) that are working together. The relative importance to other projects may be defined using shares.

# The Grid Engine Batch System

## Components of the system

- **Policies**
  - Algorithms that define, which jobs are scheduled to which queues and how the priority of running jobs has to be set.
  - GE knows functional, share based, urgency based and override policies
  - Only functional and share based policies are used at DESY
- **Shares**
  - GE can use a pool of tickets to determine the importance of jobs.
  - The pool of tickets owned by a project/job etc. is called share
  - Functional: garanties fair share at a moment in time
  - Share Tree: garanties fair share over a period of time

# Projects

- **Every user must have a project**
  - Default GE project is normally identical with your primary unix group
  - If user's primary unix group is not a GE project, the default project is “other”
- **Users can be member of more than one project**
  - Additionial unix groups that are defined as GE projects
  - All jobs run under the default GE project by default
  - If you do not want your job to run under the default project, select one with qsub / qrsh parameter “-P <project>”

# Batch Farms

- **Linux Farms, after complete migration**

| Farm | CPUs | Memory | /tmp | Architecture |
|---|---|---|---|---|
| ice | 80 * PIII @ 800 Mhz | 512 MB | ~ 25 GB | ia32 |
| globe | 100 * Xeon @ 3.2 Ghz | 2 GB | ~ 58 GB | ia32 |
| heliade | 60 * Opteron @ 2.4 Ghz | 4 GB | ~ 58 GB | amd64/ia32 |

**All hosts are in a fair share environment!**

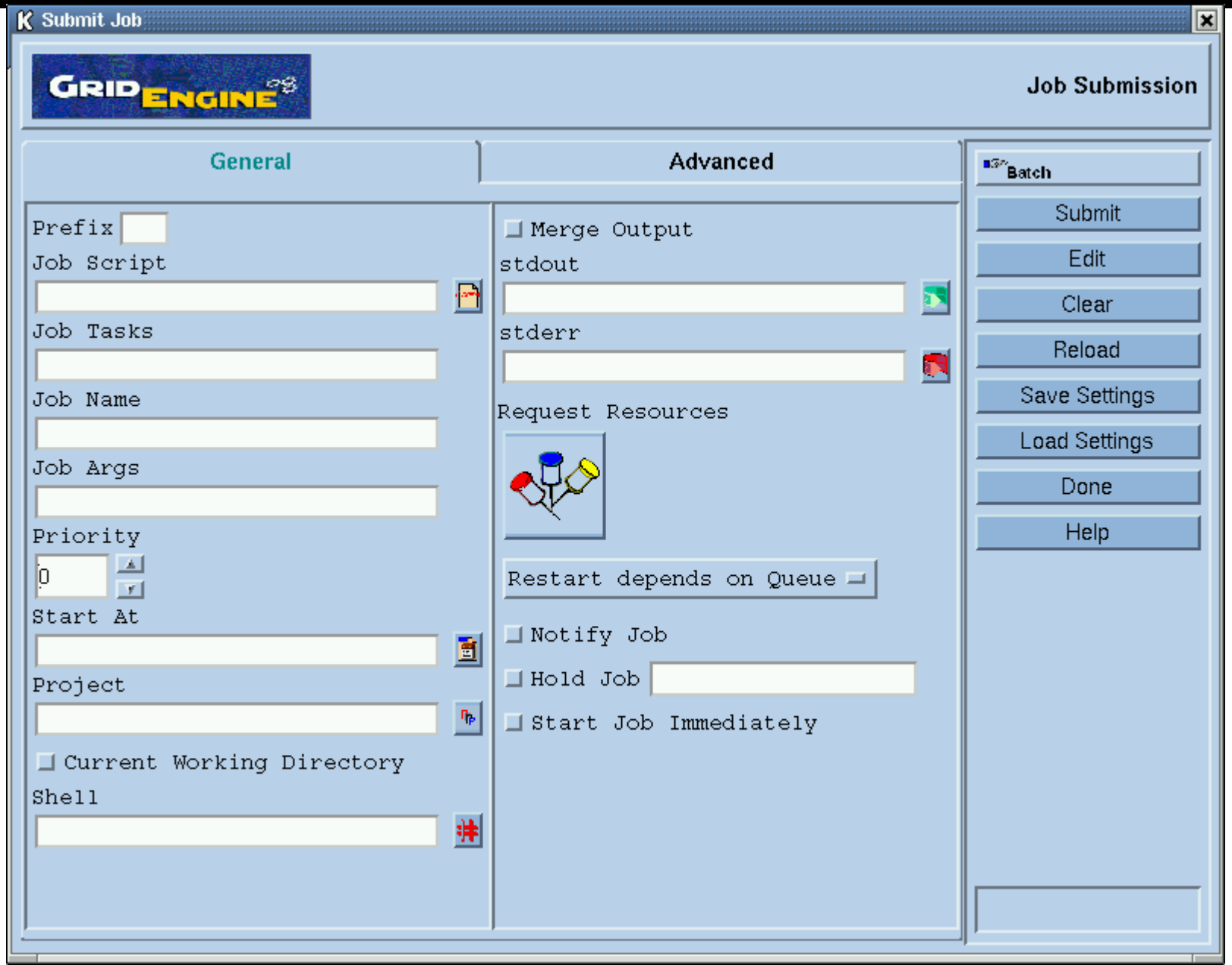
# Submitting Jobs

- **Requirements for submitting jobs**
  - logged in on a submit host (all linux desktops, public login hosts)
  - have a valid kerberos 5 ticket (verify with `klist`), otherwise obtain a new one (`kinit`)
  - ensure that in your .[t]cshrc or .zshrc no commands are executed that need a terminal (tty) (users have often a stty command in their startup scripts)
  - you are within batch if the env variable JOB_NAME is set or if the env variable ENVIRONMENT is set to BATCH
- **Submitting a job**
  - specify what resources you need (-l option) and what script should be executed

    ```
    qsub -l h_cpu=1:00:00 job_script
    ```
  - for more options consult the man page
  - alternatively use the graphical interface to submit jobs

    ```
    qmon &
    ```

# The Submit Window of qmon



# Job Submission and File Systems

- **Current working directory**
  - the directory from where the qsub command was called. STDOUT and STDERR of a job go into files that are created in **$HOME**. Because of quota limits and archiving policies that is **not recommended.**
  - With the -cwd option to qsub the files get created in the current working directory.
- **Local file space**
  - /tmp is guaranteed to exist on all linux nodes and has typically a capacity of 20GB and more
  - $TMP[DIR] is a unique directory below /tmp, that gets erased at the end of the job. Normal jobs should use this as job scratch space, if possible

# A simple Job Script

```
#!/bin/zsh
#$ -S /bin/zsh
#
#$ -l h_cpu=0:30:00
#$ -l h_vmem=512M
#$ -l tmp_free=15G
#$ -j y
#$ -m abe
DATADIR=/afs/ifh.de/group/h1/...
echo "using working directory $TMPDIR"
cp $DATADIR/large_input $TMPDIR
cd $TMPDIR
/path/to/executable
cp large_out $DATADIR
```

otherwise the default shell would be used

the cpu time limit for this job

the memory limit for this job

job needs 15 GB in $TMPDIR

merge STDOUT and STDERR

send mail when job crashes/starts/finishes

# Advanced usage of qsub

- **Option files**
  - instead of giving qsub options on the command line, users may store those in `.sge_requests` files in their $HOME or current working directories (not recommended as it will be forgotten very often...)
  - content of a sample `.sge_requests` file:

```
-cwd -S /usr/local/bin/perl -j y -l h_cpu=24:00:00
```

- **Array jobs**
  - GE allows to schedule n identical jobs with one qsub call using the -t option:

```
qsub -t 1-10 array_job_script
```

  - within the script use the variable SGE_TASK_ID to select different inputs and write to distinct output files (SGE_TASK_ID is 1...10 in the example above)
- **Job dependencies**
  - qsub parameter `-hold_jid <job id>` lets the job wait in the queue until the job with `<job id>` has finished successfully

# Abnormal Job Termination

- **Termination because of exceeded limits**
  - jobs can catch an XCPU signal (CPU soft limit exceeded). In that case termination procedures can be executed, before the SIGKILL signal is sent
  - When other soft limits are broken, SIGUSR1 is sent to the job
- **Signalling the end of the job**
  - with the qsub option -notify a SIGUSR2 signal is sent to the job one minute before the job is killed
- **Restart after the job has crashed**
  - `qmod -c <job id>`
  - Use with care and only if you understand why the job has crashed!
  - Every crashed job causes a mail to the GE admins!

# Useful GE commands

- **qstat - Job status**
  - `qstat` (basic output)
  - `qstat -u <user_id>` (show jobs for one user)
  - `qstat -ext` (show extended information – projects, tickets, usage, ...)
  - `qstat -j <job id>` (scheduling and other information for this job)
- **qhost - show host status**
- **qdel - deletes job**
  - `qdel <job id>`
- **qalter - change of qsub resources**
- **qselect - show queues which can run with specified resources**
  - `qselect -l h_vmem=2G`
- **qhold, qrls - hold and release job**
- **See also: man pages!!!**

# Complexes

- **Request complexes with qsub / qrsh parameter**
  - -l complex[=value]
- **Currently defined complexes**
  - architecture (**arch**), e.g.: ia32, amd64
  - Operating system (**os**), e.g.: sl3
  - Memory limit **(s_vmem, h_vmem)**, e.g.: 1.5G, 500M
  - File size limit (**s_fsize, h_fsize**), e.g.: 3G
  - CPU time limit (**s_cpu, h_cpu**), e.g.: 4:00:00 (4 hours), 600 (600 seconds)
  - Free temp space **(tmp_free)** e.g.: 6G
- **Usage:**
  - `qsub -l complex_1[=value_1] ... -l complex_n[=value_n] \ jobscript`
- **Example:**
  - `qsub -l h_vmem=512M -l h_cpu=30:00:00 -P theorie \ jobscript`
  - `qrsh -l h_vmem=1.3G -l arch=amd64`

# Major changes SGE 5.3 – GE 6.0

- **Changes since SGE 5.3**
  - All hosts are running SL3!
  - Command qssh does not exist any more – use qrsh
  - Complex tmpfree renamed to tmp_free
  - No “Level 6” space on execution hosts any more (/data)
  - No direct use of farms and subfarms – just specify the job's requirements

# Smooth migration to GE 6.0u4

- **Setting GE environment (only up to July 1st)**
  - Please test your jobs for GE 6.0 compliance immediately
  - ini sge6
  - qsub ...

**On July, 1st GE 6.0 will be the one and only batch system at DESY Zeuthen**

**http://www-zeuthen.desy.de/computing/services/batch/index.html**