JÜLICH
FORSCHUNGSZENTRUM

JuRoPA

Juropa-JSC - HPC-FF
Batch Usage

U.Ehrhart@fz-juelich.de

Batch System - Usage Model

- ***Juropa is divided into two partitions:***
  - *Juropa-JSC (2208 nodes: `jj01c01` .. `jj25c96`)*
    - *32 nodes dedicated to interactive jobs*
    - *Up to ca.10 nodes dedicated to software or system tests resp.*
  - *HPC-FF (1080 nodes: `jf38c01` .. `jf59c54`)*
    - *16 nodes dedicated to interactive jobs*
    - *up to 64 nodes dedicated to special tests*
- ***Each user group is allowed to submit jobs only to the appropriate partition***
  - *Jobs submitted by user will automatically run in the appropriate partition (class) – controlled by group ID*
  - *Exceptions are possible on request*

JÜLICH
FORSCHUNGSZENTRUM

Batch System

- ***Moab / Torque - Batch Scheduler and Resource Manager***
  - *Manages policies, priorities, limits*
  - *Fulfills application resource requests*
  - *Starts jobs, manages output*
  - *Provides for advanced reservations, backfilling etc.*
  - *Job statistics and accounting*
  - *User commands for job submission, job query, cancel etc.*

Batch System - Limits

- ***Interactive jobs***
  - *no node sharing*
  - *max. number of nodes: 8 (default: 1)*
  - *max. wall-clock time:*
    - *JSC: 6h, default 30 min*
    - *HPCFF: 1h, default 30 min*
  - *max. running jobs: 15 per user (including batch jobs)*
  - *accounting: (number of nodes) x (connect-time)*

JÜLICH
FORSCHUNGSZENTRUM

Batch System - Limits

- ***Batch jobs***
  - *no node sharing*
  - *max. number of nodes: 1024 (default: 1)*
  - *max. wall-clock time: 24 h (default: 30 min)*
  - *max. number of running jobs: 15 per user*
  - *max. number of 'eligible' jobs is limited to 15 per user to avoid monopolising by single users*
  - *accounting is based on (number of nodes) x (wall-clock*
  - *Jobs requesting > 1024 nodes can be run on special request*
    - *not included in normal scheduling*
    - *will be run e.g. once a week if needed during nonprime time*
    - *Please contact sc@fz-juelich.de*

Batch System - How to run a Job

- ***Write a batch script including mpiexec***
  - *Example 1*

```
#!/bin/bash -x
#MSUB -l nodes=8:ppn=8
#MSUB -l walltime=4:00:00
#MSUB -e /home/jhome3/test_user/my-error.txt
#MSUB -o /home/jhome3/test_user/my-out.txt
### start of jobscript

cd $PBS_O_WORKDIR
echo "workdir: $PBS_O_WORKDIR"

# NSLOTS = nodes * ppn = 8 * 8 = 64
NSLOTS=64
echo "running on $NSLOTS cpus ..."

mpiexec -np $NSLOTS ./mpi_prog
```

pure MPI application will start 64 tasks on 8 nodes using 8 cores/node

Batch System - How to run a Job

- *Example 2*

```
#!/bin/bash –x
#MSUB -N hybrid_8x8_job
#MSUB -l nodes=4:ppn=8
#MSUB -v tpt=4
### start of jobscript ###
export OMP_NUM_THREADS=8
mpiexec -np 4 –exports=OMP_NUM_THREADS application.exe
```

This jobscript will start a hybrid application.exe on 4 nodes allocating 4 cpus/node and 8 threads per node

Batch System - How to run a Job

- *Example 3*

```
#!/bin/bash –x
#MSUB -N SMT_MPI_64x1_job
#MSUB -l nodes=4:ppn=16
### start of jobscript ###
mpiexec -np 64 application.exe>   $PBS_O_WORKDIR/out.
 $PBSJOBID
```

Application will be started on 4 nodes using 16 MPI tasks/node, where two MPI tasks will be executed on each core
Please see http://www.fz-juelich.de/jsc/juropa/usage for more detailed information concerning jobscripts and SMT usage.

Batch System - How to run a Job

- ***Step 3: Submit the job***
  - *Submit a job*

    msub <name of jobscript>

    or (if #MSUB options are not specified in the job script):

    ```
    msub -l nodes=16:ppn=8 -e /lustre/jhome5/zam/zdv113
      -o /lustre/jhome5/zam/zdv113 <job_script>
    ```

  - *Other useful options:*

    ```
    -l walltime=hh:mm:ss
    -j oe                  # combine stderr and stdout
    -M <mail_addr>         # send mail to <mail_addr>
    -m eab                 # e = on end, a abort, b begin
    -I                     # run an interactive job
    -v tpt=<threads>       # number of OpenMP threads
    -r y                   # define job as restartable
    ```

Batch System - How to run a Job

- *Submit several dependend jobs (job chain)*

  ```
  msub -l depend=<jobid> <jobscript>
  ```

  The job submitted will start after the job with dependend <jobid> has finished. A jobscript to submit a number of jobs each depending on the preceding one can be found in the ´´Quick Introduction´´ of the user online info for Juropa.

- *Submit an interactive job*

  ```
  msub -I -l nodes=2:ppn=8,walltime=00:15:00
  ```

  you will automatically have access to a node and can start your application right here

Batch System - Commands

- ***msub***
  - *Submit a job*
  - *returns job ID on success*

    Note: during times of high impact Moab might run into a timeout
- ***showq [-r | -i | -b] [-u <userid>]***
  - *Shows all, running, idle, or blocked jobs of all or specified user*
- ***mjobctl –c <job_ID> | ALL***
  - *Cancel queued or running job*
- ***mjobctl –h <job_ID>|-u <userid>***
  - *Put hold on specified or all jobs*
- ***checkjob [-v] <jobid>***
  - *Display detailed information on specified job*

Batch System - Commands

- ***showstart <jobid>***
  - *Shows estimated starttime of specified job*
    - *estimated starttime can change while jobmix changes, assigned nodes fail,…..*
    - *Prediction at a given instant only!*
- ***mjobctl - -help***
  - *Shows all options, e.g. to hold or resume holds on jobs*
- ***showbf -c jsc|hpcff***
  - *Shows resources available for immediate use*
- ***For detailed information on Moab commands please see:***

  http://www.adaptivecomputing.com/resources/docs/mwm/6-1/help.htm

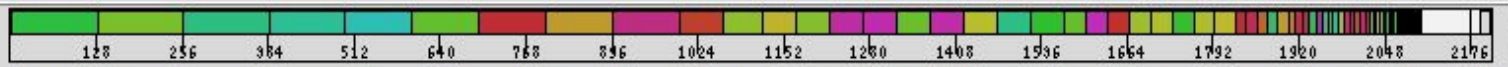
Batch System - Commands

- ***llview***
  - *Graphical view of jobs, usage, distribution of jobs*
  - *In house development by W.Frings*

,

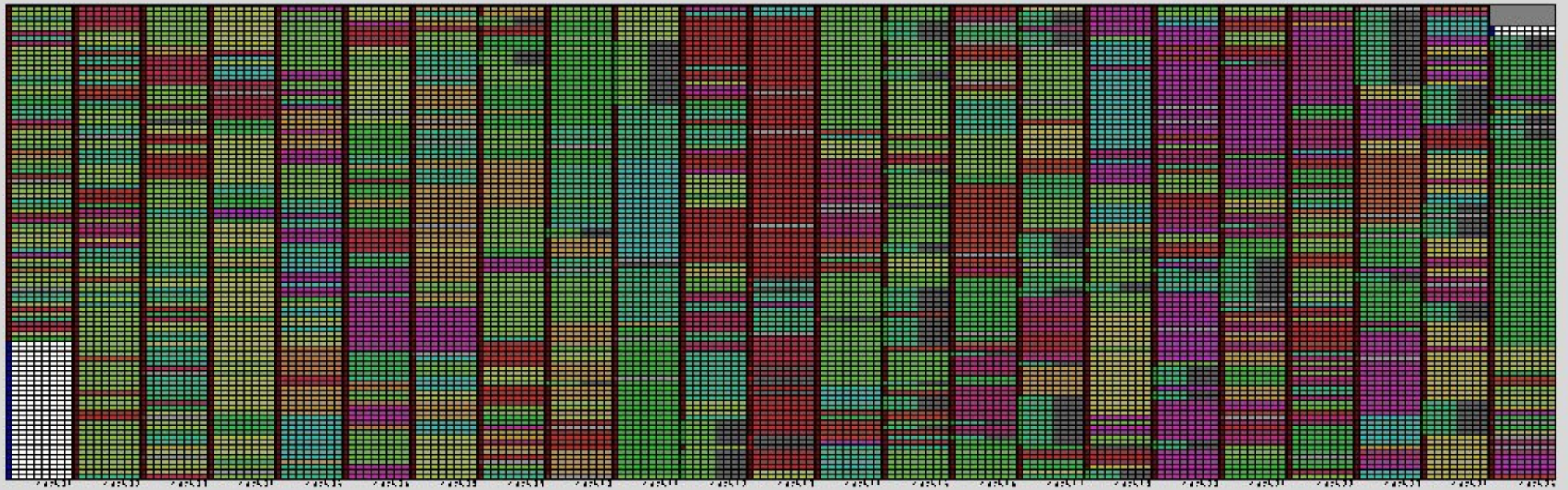
File Options | Step 60 s active | Search zdv077 | Last Update 05/12/10-14:45:01 next in 52 s Source WWW | Help

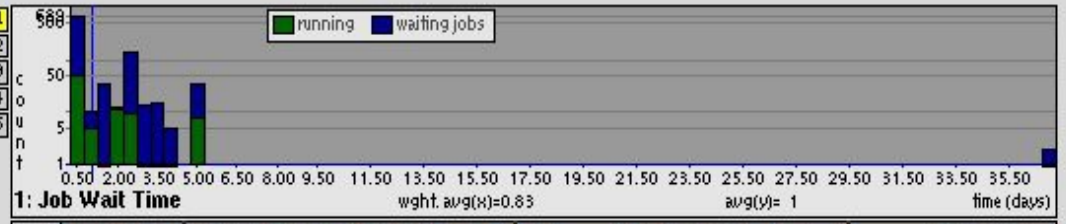
Nodes | Running | Waiting

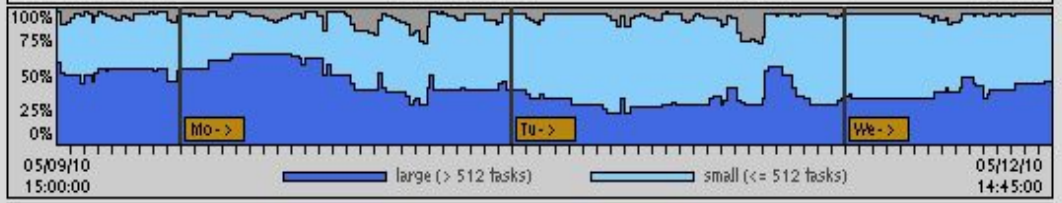


used: 95% 16800/17512,
free: 712, 89 nds (640 nshd)
#jobs (run/wait): 88/880

JUROPA







Machine: Juropa/HPC-FF
Memory: , #cpus: 26304,
speed = 2.93 GHz type = Intel Nehalem,
#frames = 24

peak = 300 TFLOPS
type = Nehalem
Date/Time: 05/12/10-14:45:01
Usable Nodes: 2189

05 updates, started at Wed May 12 14:44:00 2010

LLview

	CPUs	Userid	cpuh	wall	U	Class	Spec	TEnd
1.	1024	hpc08a	4.8h of	12:05		isc	n128.p04.t01	21:54
2.	1024	hes118	2.5h of	12:05		isc	n128.p08.t01	+00:15
3.	1024	hes118	4.6h of	12:05		isc	n128.p08.t01	22:06
4.	896	iias1100	4.3h of	10:05		isc	n112.p08.t01	20:29
5.	800	hss067	6.0h of	12:05		isc	n100.p08.t01	20:43
6.	800	hss067	5.7h of	12:05		isc	n100.p08.t01	21:01
7.	800	hss067	5.7h of	12:05		isc	n100.p08.t01	21:00
8.	800	hss067	7.4h of	12:05		isc	n100.p08.t01	19:23
9.	800	hss067	5.5h of	12:05		isc	n100.p08.t01	21:13
10.	512	izam0602	3.5h of	12:05		isc	n64.p08.t01	23:12
11.	480	hss067	8.0h of	12:05		isc	n60.p08.t01	18:42
12.	400	hss067	1.7h of	12:05		isc	n50.p08.t01	+01:01
13.	400	hss067	2.5h of	12:05		isc	n50.p08.t01	+00:15
14.	400	hss067	7.3h of	12:05		isc	n50.p08.t01	19:29
15.	400	hss067	5.8h of	12:05		isc	n50.p08.t01	20:54
16.	400	hss067	5.8h of	12:05		isc	n50.p08.t01	20:57
17.	400	hss067	7.4h of	12:05		isc	n50.p08.t01	19:23
18.	400	hss067	1.1h of	12:05		isc	n50.p08.t01	+01:39
19.	400	hss067	5.4h of	12:05		isc	n50.p08.t01	21:18
20.	400	hss067	5.9h of	12:05		isc	n50.p08.t01	20:51
21.	256	esc001	10.8h of	12:05		isc	n32.p08.t01	15:57
22.	256	esc001	10.9h of	12:05		isc	n32.p08.t01	15:50
23.	256	hpc08a	4.7h of	12:05		isc	n32.p04.t01	22:01
24.	256	esc001	10.6h of	12:05		isc	n32.p08.t01	16:07
25.	256	iias1100	0.3h of	1:05		isc	n32.p08.t01	15:24
26.	256	hss060	6.4h of	12:05		isc	n32.p08.t01	20:18
27.	256	hbn151	1.1h of	8:05		isc	n32.p08.t01	21:38
28.	240	izam0602	6.3h of	12:05		isc	n30.p08.t01	20:27
29.	128	hss069	0.3h of	11:11		isc	n16.p08.t01	+01:34

Batch System – Job Scheduling

- ***Batch usage statistics***
  - *> 2000 users*
  - *Application size from 1 up to 1024 nodes*
  - *Requested wallclocktime varies from a few minutes to 24h*
  - *Average of ~ 50000 jobs/month*
  - *Overall usage ~ 90%*

Batch System – Job Scheduling

- ***Job Scheduling Policies***

  - *Jobs are scheduled by priority*
  - *Job priority increases by number of nodes requested*
  - *Job priority increases due to waiting time (aging)*
  - *Scheduler runs in `backfilling`mode*
    - *Precise wallclocktime necessary*
  - *Priority < 0 for jobs without cpu quota*
    - *Affects already queued jobs a well*
    - *Estimated starttime (showstart) may change very often*

Batch System – CPU quota and Accounting

- ***Query current status of cpu quota***
  - `q_cpuquota <options>`
    - `q_cpuquota -?  shows all options available`
- ***Types of CPU quota***
  - *Fixed: a fixed amount of cpu quota can be used during the allocation period (refers to small quota amounts)*
  - *Monthly: jobs will be scheduled with normal priority until current, previous and next monthly quota is exhausted. CPU quota not used in this time frame is lost.*

Batch System – CPU quota and Accounting

- ***Policies in terms of cpu quota***
  - *All members of a group will be informed by mail if the group runs out of cpu quota or if new quota is assigned*
  - *Jobs will get a low priority (< 0) when cpu quota is used up.*
  - *Already queued jobs might not start due to decreased time limit*
    - *Set to hold*
    - *Released automatically if new cpu quota is assigned*

Batch System – CPU quota and Accounting

- ***Further Information***
  - *Preventive maintenance every second Thursdays*
    - *See ´Message of today´ on login*
  - *get status updates by subscribing to the system messages at the bottom of*
    - *http://juelich.de/jsc/CompServ/services/high_msg.html*
  - *online user JSC documentation:*
    - *http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUROPA*
  - *User support: sc@fz-juelich.de*
  - *HPCFF oerson of contact: A.Schnurpfeil*
    - *hpcff-support@fz-juelich.de*