RWTH AACHEN UNIVERSITY

The RWTH Compute Cluster Environment

Source: D. Both, Bull GmbH

Tim Cramer

29.07.2013

The RWTH Compute Cluster (1/2)

- **The Cluster provides ~300 TFlop/s**
  - → No. 32 in TOP500 (June 2011), no. 4 in Germany
  - → No. 142 in TOP500 (June 2013)
  - → 1358 Westmere EP nodes ("**MPI Partition**")
    - → 2x Xeon X5675 (6-core CPU) @ 3.06 GHz
      24 – 96 GB RAM, QDR Inifiniband (Full fat tree)
  - → 88 Nehalem EX nodes ("**SMP/BCS Partition**")
    - → 16x Xeon X7550 (8-core CPU) @ 2.00 GHz
      256 – 2048 GB RAM, QDR Infiniband (Full fat tree)
    - → Connected with proprietary BCS-Chips from Bull and consist of 4 physical 4-socket nodes
  - → 28 Nvidia nodes
    - → 2x Quadro 6000 (Fermi, 448 GPU cores)
    - → 2x 6 GB GPU memory, PCIe bus
  - → 9 Intel Xeon Phi nodes
    - → 2x Intel Xeon Phi @ 1 GHz (MIC, 60 cores)
    - → 2x 8 GB DDR5 memory, PCIe bus

Source: D. Both, Bull GmbH

The RWTH Compute Cluster (2/2)

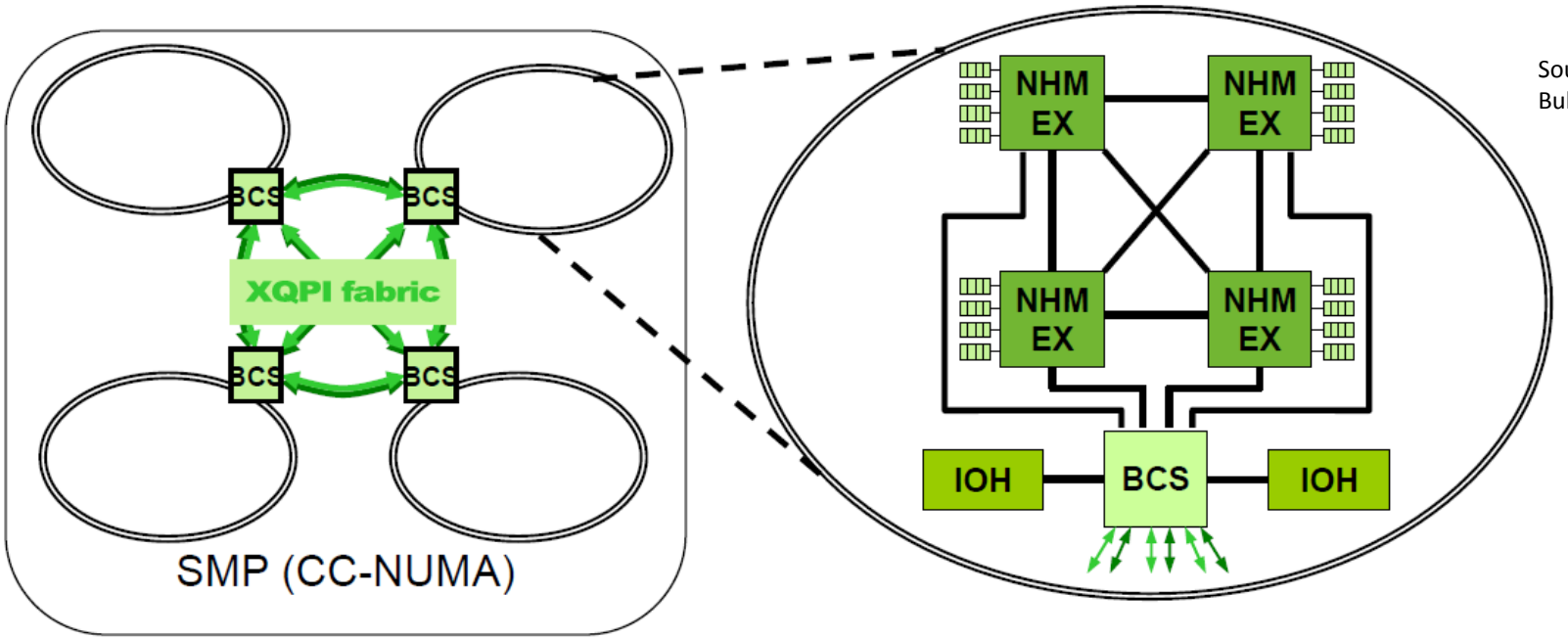
- **The Cluster provides ~3 PByte storage**
  - → 1.5 PByte parallel high performance file system
    - →Lustre ($HPCWORK)
    - →designed for high throughput
    - →(Group) Quota 1TB / 50,000 files
  - → 1.5 PByte NFS file system
    - →NetApp filer ($HOME / $WORK)
    - →Quota HOME:150 GB (1,000,000 files)
    - →Quota WORK: 250GB (1,000,000 files)
  - → only HOME is backuped
  - → no automatic cleanup for any file system

Source: D. Both, Bull GmbH

Fat Nodes (“SMP Partition”)

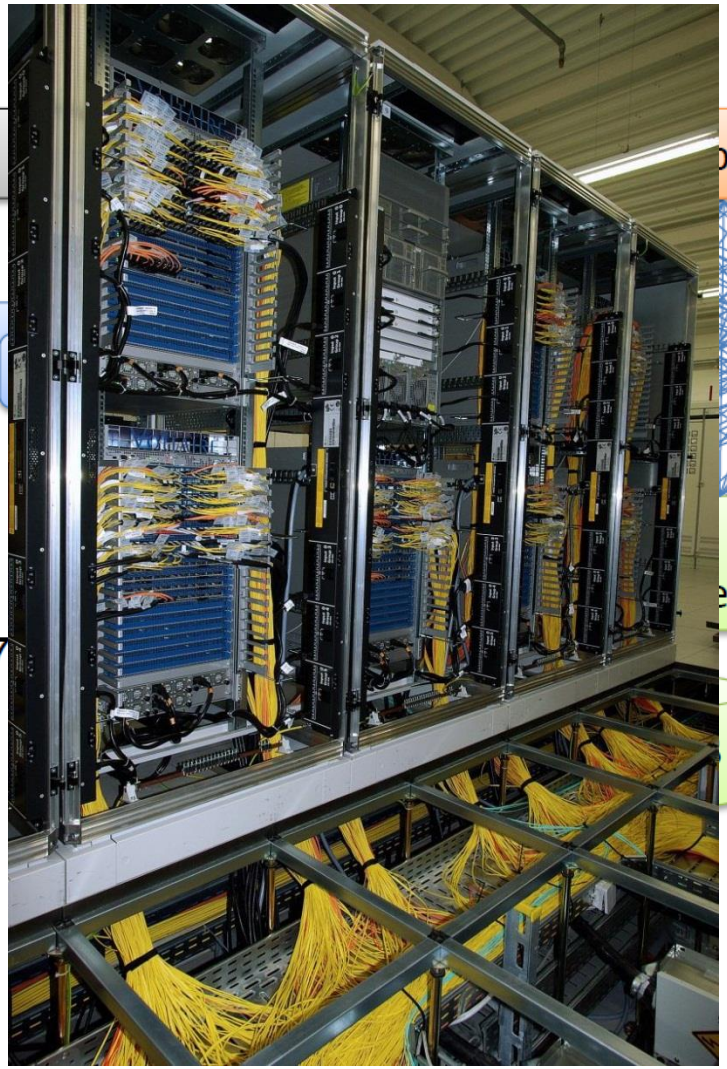
RWTH AACHEN UNIVERSITY

- **Bull Coherence Switch (BCS)**
  - → Two levels of NUMAness
  - → One node (128 Cores) consist of 4 physical 4-socket nodes
  - → Smallest job / granularity:
    - → Shared memory: 32 threads / MPI: 128 processes

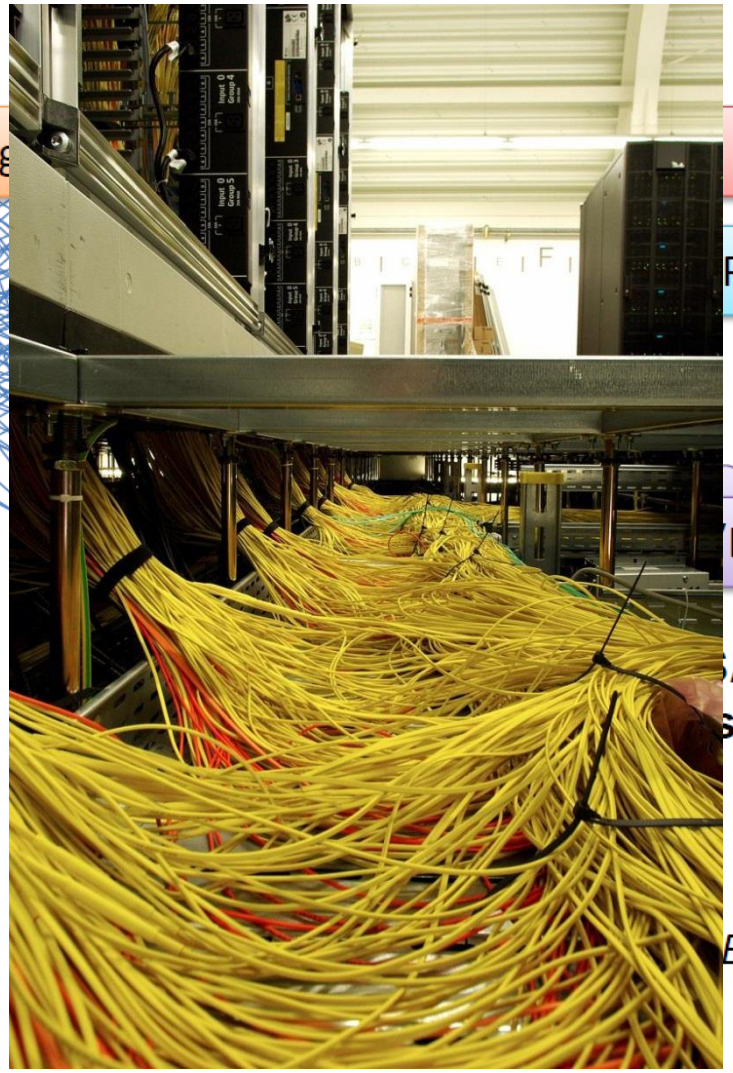


Source: Bull GmbH

Overview of the New Cluster



Source: D. Both, Bull GmbH



Source: D. Both, Bull GmbH

How to login

Frontends

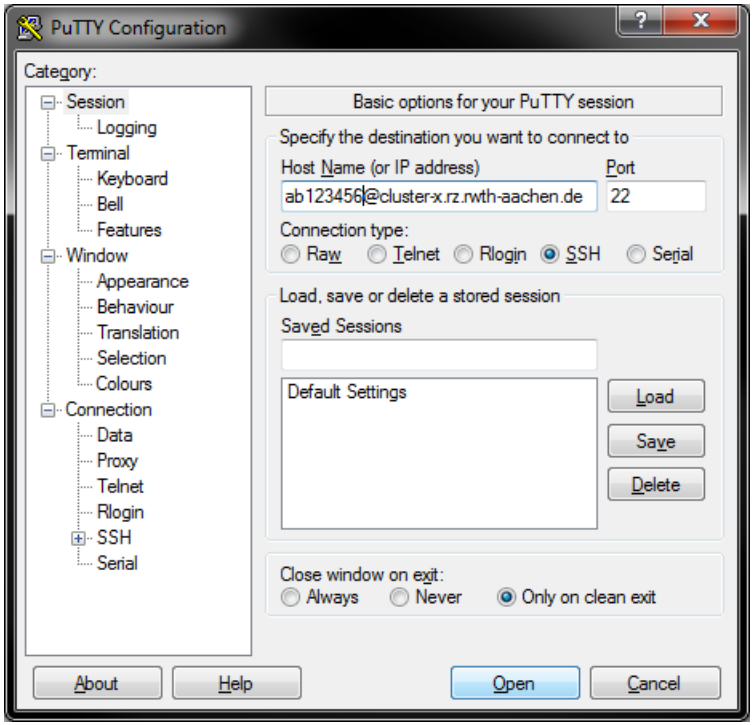
cluster.rz.RWTH-Aachen.DE	cluster2.rz.RWTH-Aachen.DE
cluster-x.rz.RWTH-Aachen.DE	cluster-x2.rz.RWTH-Aachen.DE
cluster-linux.rz.RWTH-Aachen.DE	cluster-linux-opteron.rz.RWTH-Aachen.DE
cluster-linux-xeon.rz.RWTH-Aachen.DE	cluster-linux-nehalem.rz.RWTH-Aachen.DE
cluster-linux-tuning.rz.RWTH-Aachen.DE	cluster-copy.rz.RWTH-Aachen.DE

- Use frontends to develop program, compile applications, prepare job scripts or debug programs
- Different frontends for different purposes
- cgroups activated for fair-share
- login / SCP File transfer:
  - `$ ssh [-Y] user@cluster.rz.rwth-aachen.de`

    `$ scp [[user@]host1:]file1 [...] [[user@]host2:]file2`

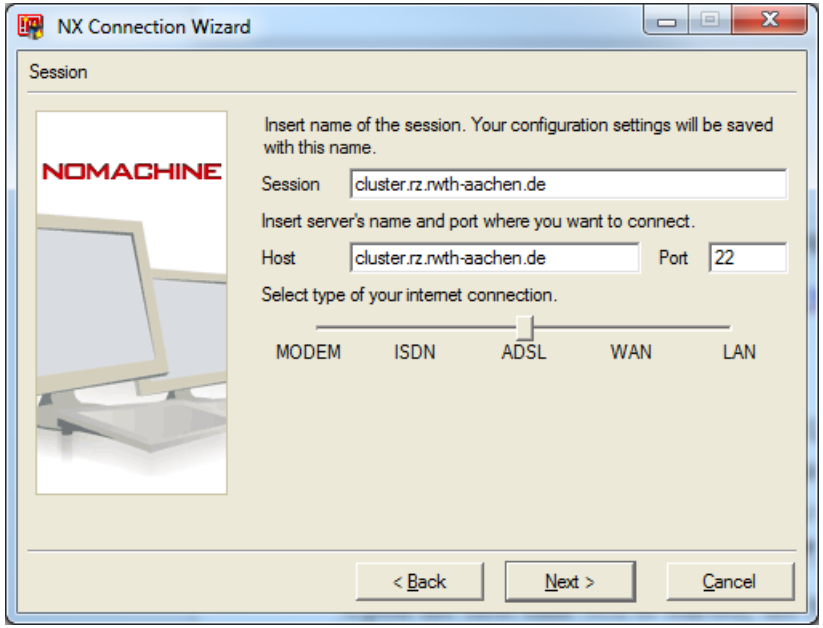
Login to a frontend / SSH

- Linux users can use a ssh connection out of a terminal:
  `$ ssh [-Y] <username>@cluster.rz.rwth-aachen.de`
- Windows users can use PuTTY from http://www.putty.org
  extract or install and configure it to connect to `cluster`, `cluster2` or `cluster-linux`



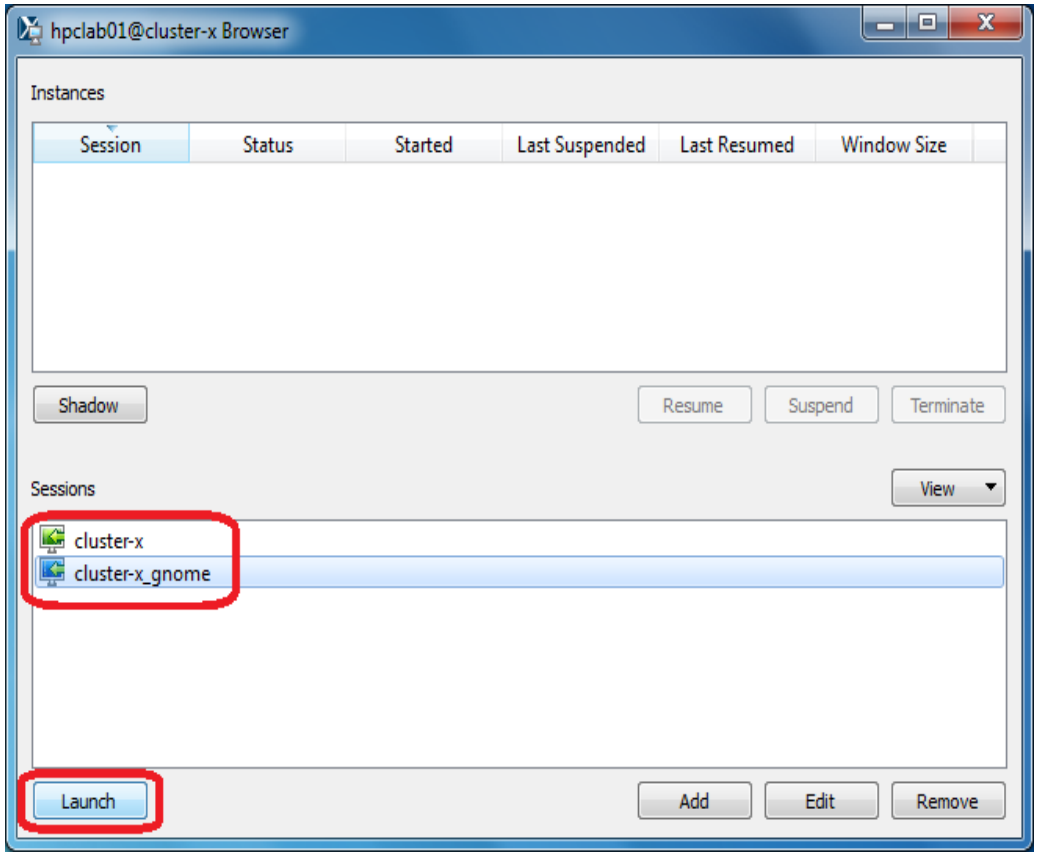
Login to a frontend / NX Client

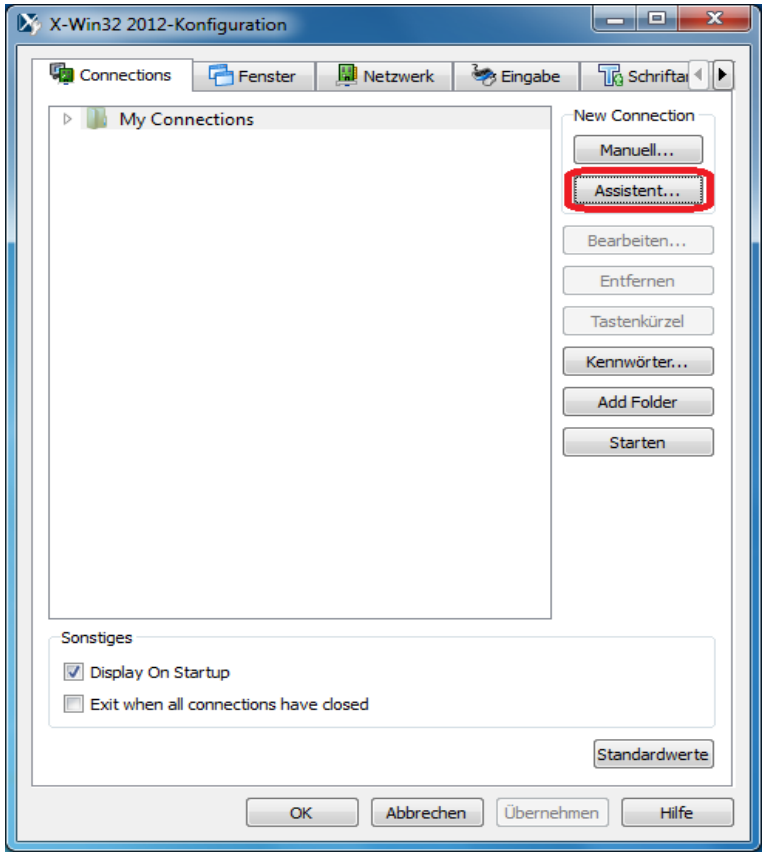
- We are running a NX server on two Linux frontend machines (cluster-x and cluster-x2).
- The NX allows you to run remote X11 sessions even across low-bandwidth network connections, as well as reconnecting to running sessions.
- Download the NX client from www.nomachine.com/download.
- Use the NX Connection Wizard to set up the connection.



Login to a frontend / X-Win32

- Alternative for NX client
- Better performance for Tuning Tools (e.g., Intel VTune Amplifier)





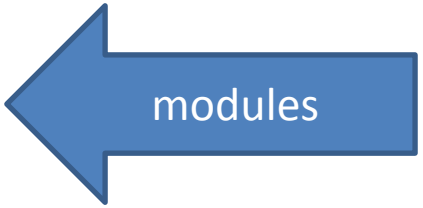
Live X-Win32

Module System (1/2)

- **Many compilers, MPIs and ISV software**
- **The module system helps to manage all the packages**
  - List loaded modules
    - `$ module list`
  - List available modules
    - `$ module avail`
  - Load / unload a software
    - `$ module load <modulename>`
    - `$ module unload <modulename>`
  - Exchange a module (Some modules depend on each other)
    - `$ module switch <oldmodule> <newmodule>`
  - Reload all modules (May fix your environment, especially with a NX session)
    - `$ module reload`
  - Find out in which category a module is:
    - `$ module apropos <modulename>`

Module System (2/2)

```
$ module avail
------------ /usr/local_rwth/modules/modulefiles/linux/x86-64/DEVELOP ------------
cmake/2.8.5(default)   inteltbb/4.1(default)
cuda/40                intelvtune/XE2013U02(default)
cuda/41                likwid/system-default(default)
cuda/50(default)       nagfor/5.2
ddt/2.6                nagfor/5.3.1(default)
ddt/3.0(default)       openmpi/1.5.3
gcc/4.3                openmpi/1.6.1(default)
gcc/4.5                openmpi/1.6.1mt
gcc/4.6                openmpi/1.6.4
gcc/4.7                openmpi/1.6.4mt
...                                    ...
------------ /usr/local_rwth/modules/modulefiles/GLOBAL ------------
BETA      DEPRECATED GRAPHICS   MATH      TECHNICS   VIHPS
CHEMISTRY  DEVELOP    LIBRARIES  MISC      UNITE
```



Environment Variable

RWTH AACHEN UNIVERSITY

- **For convenience we provide several environment variables**
  - Set by the module system

Variable	Function
$FC, $CC, $CXX	Compiler
$FLAGS_DEBUG	Compiler option to enable debug information.
$FLAGS_FAST	Enables several compiler optimization flags.
$FLAGS_OPENMP	Enables OpenMP support.
$MPIFC, $MPICC, $MPI	MPI compiler wrapper.
$MPIEXEC	The MPI command used to start MPI applications.
$FLAGS_MPI_BATCH	MPI options necessary for executing in batch mode.
$FLAGS_OPENMP	Compiler option to enable OpenMP support.
$OMP_NUM_THREADS	Sets the number of threads for OpenMP applications.
$FLAGS_MATH_INCLUDE	Include flags for mathematical libraries (e.g. Intel MKL)
$FLAGS_MATH_LINKER	Linker flags for mathematical libraries (e.g. Intel MKL)

Live demo module system / simple C program

Why to use the batch system?

- **Use of backend nodes via our batch system for large calculations**
  - Contra:
    - Jobs sometimes need to wait before they can start
  - Pro:
    - Nodes are not overloaded with too many jobs
    - Jobs with long runtime can be executed
    - Systems are also used at night and on the weekend
    - Fair share of the resources for all users
    - The only possibilty to handle such a big amount of compute nodes

LSF: Job Submission (1/3)

How to submit a job

- `$ bsub [options] command [arguments]`

General parameters

Parameter	Description
`-J <name>`	Job name
`-o <path>`	Standard out
`-e <path>`	Standard error
`-B`	Send mail when job starts running
`-N`	Send mail when job is done
`-u <mailaddress>`	Recepient of mails
`-P <projectname>`	Assign the job to the specified project (e.g. jara, integrative hosting costumers)
`-U <reservation>`	Use this for advanced reservations

LSF: Job Submission (2/3)

- **How to submit a job**
  - `$ bsub [options] command [arguments]`
- **Parameters for job limits / resources**

Parameter	Description
`-W <runlimit>`	Sets the hard runtime limit in the format [hour:]minute [default: 15]
`-M <memlimit>`	Sets a **per-process** memory limit in MB [default: 512]
`-S <stacklimit>`	Set a **per-process** stack size limit in MB [default: 10]
`-C <corefilesize>`	Set a **per-process** core file size limit in MB [default: 16]
`-x`	Request node(s) exclusive
`-R "select[hpcwork]"`	ALWAYS set if you using the HPCWORK (Lustre file system)

LSF: Job Submission (3/3)

- **How to submit a job**
  - `$ bsub [options] command [arguments]`
- **Parameters parallel jobs**

Parameter	Description
`-n <min_proc>[,max_proc]`	Submits a parallel job and specifies the number of processors required [default: 1]
`-a openmp`	Use this to submit a shared memory job (e.g. OpenMP)
`-a {open|intel}mpi`	Specify the MPI (remember to switch the module for Intel MPI)
`-R „span[hosts=1]“`	Request the compute slots on the same node
`-R „span[ptile=n]“`	Will span *n* proceses per node (hybrid)

- `$MPIEXEC $FLAGS_MPI_BATCH a.out`

LSF: Using the Magic Cookie

- **You can use the magic cookie `#BSUB` for a batch script `job.sh`**

```
#!/bin/zsh
#BSUB -J TESTJOB                    #Job name
#BSUB -o TESTJOB.o%J                #STDOUT, the %J is the job id
#BSUB -e TESTJOB.e%J                #STDERR, the %J is the job id
#BSUB -We 80                        #Request 80 minutes
#BSUB -W 100                        #Will run max 100 minutes
#BSUB -M 1024                       #Request 1024 MB virtual mem
#BSUB -u user@rwth-aachen.de        #Specify your mail
#BSUB -N                            #Send a mail when job is done
cd /home/user/workdirectory         #Change to the work directory
a.out                               #Execute your application
```

- **Submit this job**
  - `$ bsub < job.sh`
  - **Please note the <, with SGE this was not needed, with LSF it is**

LSF: Job Status (1/2)

Use `bjobs` to display information about LSF jobs

- `$ bjobs [options][jobid]`

```
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
3324    tc53084 RUN   serial     linuxtc02   ib_bull     BURN_CPU_1 Jun 17 18:14
3325    tc53084 PEND  serial     linuxtc02   ib_bull     BURN_CPU_1 Jun 17 18:14
3326    tc53084 RUN   parallel   linuxtc02   12*ib_bull  *RN_CPU_12 Jun 17 18:14
3327    tc53084 PEND  parallel   linuxtc02   12*ib_bull  *RN_CPU_12 Jun 17 18:14
```

Option	Description
-l	Long format – displays detailed information for each job
-w	Wide format - displays job information without truncating fields
-r	Displays running jobs
-p	Displays pending job and the **pending reasons**
-s	Displays suspended jobs and the suspending reason

- LSF can display the reasons for a pending job

LSF: Job Status (2/2)

- **Use `bpeek` to display stdout and stderr of an running LSF job**
  - `$ bpeek [options][jobid]`

```
<< output from stdout >>
Allocating 512 MB of RAM per process
Writing to 512 MB of RAM per process
PROCESS 0: Hello World!
PROCESS 1: Hello World!
[ application output ]
<< output from stderr >>
```

- **Remove a job from the queue**
  - `$ bkill [jobid]`
- **Remove all jobs from the queue**
  - `$ bkill 0`

Feedback and Documentation

RWTH Compute Cluster Environment

- HPC Users‘s Guide:

  http://www.rz.rwth-aachen.de/hpc/primer

- Online documentation (including example scripts):

  https://wiki2.rz.rwth-aachen.de/

- Full LSF documentation:

  http://www1.rz.rwth-aachen.de/manuals/LSF/8.0/index.html

- Man-Pages for all commands available

- In case of errors / problems let us know:

  servicedesk@rz.rwth-aachen.de

End