



Official Power UP btrzx5

Bernhard Winkler

```
PROGRAM btrzx5
```

```
#include <stdgreetings.h>
```

```
#include „../*/greetings.h“
```

```
call Motivation(mandelbrot)
```

```
call Facts&Numbers(btrzx5)
```

```
call Setup(graphic)
```

```
call Software(btrzx5)
```

```
call ExampleProgram(FORTRAN,MPI)
```

```
call Queue(PBS)
```

```
call Questions(Audience)
```

```
END btrzx5
```

Subroutine Motivation(name)

Let's calculate (and draw) Mandelbrot's Julia Set

lots of computing work to be done

work can be done in parallel

„Getrennt Marschieren, vereint Schlagen“

(1866 (Schlacht bei Königgrätz) Graf v.Moltke (1800-1891))

→ ideal for Linux clusters

END Motivation

```
Subroutine Facts&Numbers(name)
```

```
NumberofNodes = 142; ProcsperNode = 2; CoresperProc = 4
```

```
TotalCores = NumberofNodes * ProcsperNode * CoresperProc = 1136
```

```
CPUType = „Intel E5520 2.26GHz (Nehalem/I7)“
```

```
RamperNode = 24GB
```

```
TotalRam = NumberofNodes * RamperNode = 3408
```

```
for (I=1,NumberofNodes); write(NodeName(I),'(node%03d)') I; done
```

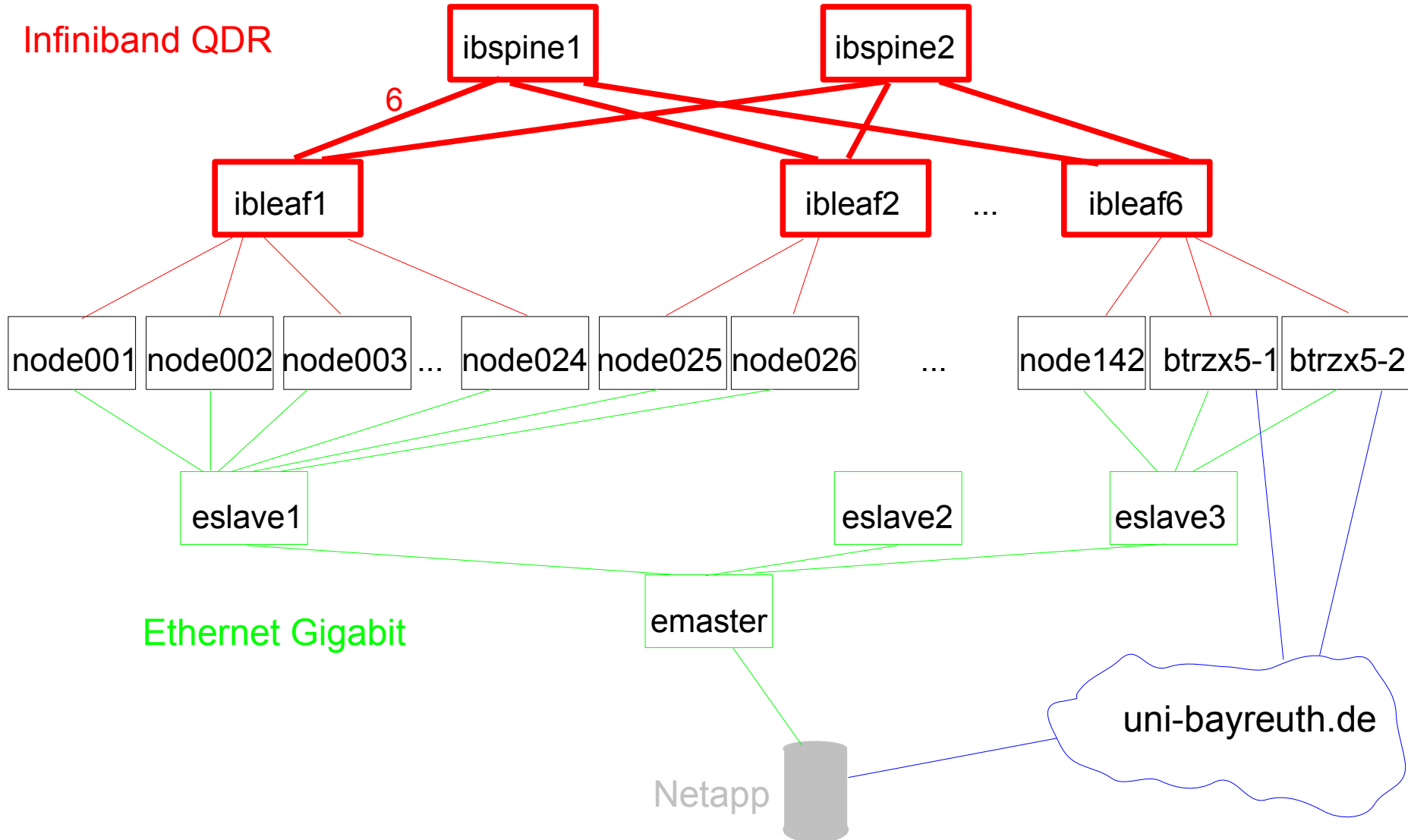
```
NumberofFrontends=2
```

```
FrontendName=(„btrzx5-1.rz.uni-bayreuth.de“, „btrzx5-2.rz.uni-bayreuth.de“)
```

```
FileSystems=(„/home“, „/scratch“, „/tmp“)
```

```
END Facts&Numbers
```

Infiniband QDR



SUBROUTINE Software(name)

KernelVersion = „LINUX 2.6.16.60-0.21-smp“

Distribution = „SUSE Linux Enterprise Server 10 SP2 (x86_64)“

Modules currently available:

compiler/intel

libs/intel/mkl

mpi/openmpi/1.3.2/intel

mpi/openmpi/1.3.2/gcc

libs/fftw/intel/3.2.3

→ add the following lines to your .tcshrc.own :

```
if (-r /etc/csh.cshrc.modules) then
```

```
    module load compiler/intel
```

```
    module load libs/intel/mkl
```

```
    module load libs/fftw/intel/3.2.3
```

```
    module load mpi/openmpi/1.3.2/intel
```

```
endif
```

END Software

8.10.2009

Dr. Bernhard L. Winkler

6

SUBROUTINE ExampleProgram(name,name1)

file blw.f:

```
program main

include 'mpif.h'

integer n, myid, numprocs, l
character*(MPI_MAX_PROCESSOR_NAME) procnam

call MPI_INIT( ierr )
call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
call MPI_Get_processor_name (procnam,l,ierr)
write(6,*) 'Process ',myid,' of ',numprocs,' is alive on ',
&      procnam(1:l)
call sleep (1)
call MPI_FINALIZE(rc)

stop
end
```

END ExampleProgram

SUBROUTINE Queue(name)

Commands to operate the Queuing system:

qstat : Show status of Jobs

qdel : Delete / terminate jobs

qsub : Submit a job

For best performance use as many Processing Cores per node as possible (8)

Example Jobfile:

```
#!/bin/tcsh
```

```
#PBS -l nodes=4:ppn=8
```

```
#PBS -j oe
```

```
cd $PBS_O_WORKDIR
```

```
date
```

```
$MPI_RUN -hostfile $PBS_NODEFILE ./blw
```

END Queue


```
PROGRAM btrzx5
#include <stdgreetings.h>
#include „../*/greetings.h“

call Motivation(mandelbrot)
call Facts&Numbers(btrzx5)
call Setup(graphic)
call Software(btrzx5)
call ExampleProgram(FORTRAN,MPI)
call Queue(PBS)
call Questions(Audience) ← To be supplied now!
END btrzx5
```