

Performance HPC Linux  
Bull Echirolles

# Workbook Compilers

C.Berthelot  
Christophe.Berthelot@bull.net

Copyright (©) Bull S.A.S. 2008



- Equipment list
- First compilation and help to port
- Optimization
- MPI

- Equipment list
  - Hardware
  - Software
- First compilation and help to port
- Optimization
- MPI

# Hardware

## NovaScale 3045

- ▶ Number of processor: 4 CPUs Itanium 2 (8 cores)
- ▶ Frequency: 1.6Ghz
- ▶ Memory size: 32G

## Networking equipment

- ▶ Infiniband network

# Software

- ▶ Bull Linux AS4 V5.1 FIX10
- ▶ Compiler: Fortran **9.1.045**, 10.0.023 and 10.1.008
- ▶ Compiler: C **9.1.049** and 10.0.023 and 10.1.008
- ▶ MPI: mpibull2-1.2.9-2.t
- ▶ POV-Ray
- ▶ NPB benchmark

## modules:

```
>module list
Currently Loaded Modulefiles:
1) oscar-modules/1.0.3      2) intel/cc/9.1.049
3) intel/fc/9.1.045        4) intel/ld/9.1.049
5) intel/cmkl/9.0.018      6) mpibull2/1.2.9-2.t
```

- Equipment list
- First compilation and help to port
  - Setting up
  - Test switching default
  - Test switching High level optimization
  - Test Efence vs CB
  - Test reporting
- Optimization
- MPI

## Setting up

- ▶ `cd $HOME`
- ▶ `cp /tmp/tp_formation.tgz .`
- ▶ `tar -zxvf tp_formation.tgz`
- ▶ `cp TP/portage`

# Test switching default

## Compile

- ▶ make clean
- ▶ make SWITCH

## Run

```
./portage_switch
```

## Record

- ▶ Time for switching
- ▶ Time for switching2
- ▶ Time for tiling



# Test switching default

## Compile

- ▶ make clean
- ▶ make SWITCH

## Run

```
./portage_switch
```

## Record

- ▶ Time for switching
- ▶ Time for switching2
- ▶ Time for tiling

# Test switching default

## Compile

- ▶ make clean
- ▶ make SWITCH

## Run

```
./portage_switch
```

## Record

- ▶ Time for switching  $6.36E - 2s$
- ▶ Time for switching2  $2.46E - 02s$
- ▶ Time for tiling

# Test switching default

## Compile

- ▶ make clean
- ▶ make SWITCH

## Run

```
./portage_switch
```

## Record

- ▶ Time for switching  $6.36E - 2s$
- ▶ Time for switching2  $2.46E - 02s$
- ▶ Time for tiling  $4.98E - 02s$

# Test switching High level optimization

## Compile

- ▶ make clean
- ▶ make SWITCH\_FLAGS="-O3"

## Run

```
./portage_switch
```

## Record

- ▶ Time for switching
- ▶ Time for switching2

# Test switching High level optimization

## Compile

- ▶ make clean
- ▶ make SWITCH\_FLAGS="-O3"

## Run

```
./portage_switch
```

## Record

- ▶ Time for switching
- ▶ Time for switching2

# Test switching High level optimization

## Compile

- ▶ make clean
- ▶ make SWITCH\_FLAGS="-O3"

## Run

```
./portage_switch
```

## Record

- ▶ Time for switching  $2.43E - 02s$
- ▶ Time for switching2  $2.42E - 02s$

# Efence vs CB first test

## Compile

- ▶ make clean
- ▶ make TEST

## Run

```
./portage_test
```

## Result

- ▶ Ok or not ?

# Efence vs CB first test

## Compile

- ▶ make clean
- ▶ make TEST

## Run

```
./portage_test
```

## Result

- ▶ Ok or not ?
- ▶ It seems OK

```
Time for test 1 : 1.5000000E-03 s  
A(M,M)= 1.711544025050047E-002  
A(0,0)= 0.000000000000000E+000  
== End subroutine test1 ==
```



## Compile

- ▶ make clean
- ▶ make TEST FLINKS="-lefence"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?

## Compile

- ▶ make clean
- ▶ make TEST FLINKS="-lefence"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?
- ▶ No

```
Electric Fence 2.2.0 Copyright (C) 1987-1999 Bruce Perens <bruce@perens.com>
Time for test 1 : 1.6000000E-03 s
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image                PC                Routine                Line                Source
portage_test          4000000000004160  Unknown                Unknown              Unknown
portage_test          4000000000002D40  Unknown                Unknown              Unknown
```

## Compile

- ▶ make clean
- ▶ make TEST\_FLAGS="-g -traceback" FLINKS="-lefence"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?

## Compile

- ▶ make clean
- ▶ make TEST\_FLAGS="-g -traceback" FLINKS="-lefence"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?
- ▶ No but we have:

```
Electric Fence 2.2.0 Copyright (C) 1987-1999 Bruce Perens <bruce@perens.com>
Time for test 1 : 0.1762000      s
forrtl: severe (174): SIGSEGV, segmentation fault occurred
Image                PC                Routine                Line                Source
portage_test          40000000000005470  test1_                191  portage.f90
portage_test          400000000000060D0  MAIN_                299  portage.f90
```

## Compile

- ▶ make clean
- ▶ make TEST\_FLAGS="-CB -g -traceback"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?

## Compile

- ▶ make clean
- ▶ make TEST\_FLAGS="-CB -g -traceback"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?
- ▶ No but we have:

```
Time for test 1 : 0.2447000      s
forrtl: severe (408): fort: (2): Subscript #2 of the array A has value
101 which is greater than the upper bound of 100
Image                PC                Routine                Line                Source
portage_test          400000000000065A0      test1_                191      portage.f90
portage_test          40000000000008010      MAIN_                299      portage.f90
```

# Efence vs CB second test

## Compile

- ▶ make clean
- ▶ make TESTE\_FLAGS="-g -traceback" FLINKS="-lefence"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?

# Efence vs CB second test

## Compile

- ▶ make clean
- ▶ make TESTE\_FLAGS="-g -traceback" FLINKS="-lefence"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?
- ▶ It seems ok:

```
Electric Fence 2.2.0 Copyright (C) 1987-1999 Bruce Perens <bruce@perens.com>  
A(M,M)= 1.711544025050047E-002  
A(0,0)= 0.000000000000000E+000  
== End subroutine test2 ==
```



## Efence vs CB second test

### Compile

- ▶ make clean; make TESTE\_FLAGS="-CB -g -traceback"

### Run

```
./portage_test
```

### Result ?

- ▶ Ok or not ?

# Efence vs CB second test

## Compile

- ▶ make clean; make TESTE\_FLAGS="-CB -g -traceback"

## Run

```
./portage_test
```

## Result ?

- ▶ Ok or not ?
- ▶ No

```
      A(M,M)= 1.711544025050047E-002
forrtl: severe (408): fort: (3): Subscript #2 of the array A has value 0
which is less than the lower bound of 1
Image                PC                Routine                Line                Source
portage_test          40000000000006DE0      test2_                228      portage.f90
portage_test          40000000000007910      MAIN_                303      portage.f90
```

# Test reporting 1/2

## Compile

- ▶ module purge; module load intel/fc/10.1.008
  - ▶ make clean
  - ▶ make APPEL FLAGS="-O3 -opt-report-fileapp.txt -opt-report-phase hlo"
- 
- ▶ Look file app.txt and opt\_report.log

# Test reporting 1/2

## Compile

- ▶ module purge; module load intel/fc/10.1.008
- ▶ make clean
- ▶ make APPEL FLAGS="-O3 -opt-report-fileapp.txt -opt-report-phase hlo"

- ▶ Look file app.txt and opt\_report.log

```
-bash-3.00$ more opt_report.log  
<portage.f90;235:256;SWP;appelle_fct_;0>
```

```
    Loop body has a function call => cannot be pipelined
```

# Test reporting 1/2

## Compile

- ▶ `mv opt_report.log opt_report.log1; make clean`
  - ▶ `make PIPE_FLAGS="-O3 -opt-report-fileapp.txt -opt-report-phase hlo"`
- 
- ▶ Look file `opt_report.log`

# Test reporting 1/2

## Compile

- ▶ `mv opt_report.log opt_report.log1; make clean`
- ▶ `make PIPE_FLAGS="-O3 -opt-report-fileapp.txt -opt-report-phase hlo"`

- ▶ Look file `opt_report.log`

```
-bash-3.00$ more opt_report.log
<portage.f90;280:285;SWP;swp_test_;0>
    Resource II      =      5
    Recurrence II   =      1
    Minimum II      =      5
    Scheduled II    =      5
    Estimated GCS II =    32
    Percent of Resource II needed by arithmetic ops      = 40%
    Percent of Resource II needed by memory ops          = 40%
    Percent of Resource II needed by floating point ops  = 100%
    Number of stages in the software pipeline =    10
```

- Equipment list
- First compilation and help to port
- Optimization
  - Setting up
  - Initial compilation
  - High level optimization
  - Profile Guides optimization
  - Putting it all together
  - Inter procedural optimization
- MPI

## Setting up

- ▶ `cd $HOME`
- ▶ `cd $HOME/TP/povray/povray31/source/unix`



# Initial compilation with gcc

## Compile

- ▶ make clean
- ▶ make usegcc

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed

# Initial compilation with gcc

## Compile

- ▶ make clean
- ▶ make usegcc

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed 68seconds

# Initial compilation with icc

## Compile

- ▶ make clean
- ▶ make useicc

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed

# Initial compilation with icc

## Compile

- ▶ make clean
- ▶ make useicc

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed `60seconds`

# High level optimization

## Compile

- ▶ make clean
- ▶ make useicc CF="-O3"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed

# High level optimization

## Compile

- ▶ make clean
- ▶ make useicc CF="-O3"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed 56seconds

# Profile Guides optimization: prof\_gen

## Compile

- ▶ make clean
- ▶ make useicc CF="-prof\_gen"

## Run

```
./povbench ../../scenes/objects/pawns.pov  
./povbench ../../scenes/advanced/skyvase.pov  
./povbench ../../scenes/advanced/chess2.pov
```

# Profile Guides optimization: prof\_use

## Compile

- ▶ make clean
- ▶ make useicc CF="-prof\_use"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed



# Profile Guides optimization: prof\_use

## Compile

- ▶ make clean
- ▶ make useicc CF="-prof\_use"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed 68seconds

# Putting it all together

## Compile

- ▶ make clean
- ▶ make useicc CF="-O3 -prof\_use"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed

# Putting it all together

## Compile

- ▶ make clean
- ▶ make useicc CF="-O3 -prof\_use"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed 64seconds

# Inter procedural optimization

## Compile

- ▶ make clean
- ▶ make useicc CF="-fast -prof\_use"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed

# Inter procedural optimization

## Compile

- ▶ make clean
- ▶ make useicc CF="-fast -prof\_use"

## Run

```
./povbench ../../scenes/objects/pawns.pov
```

## Record

Record time elapsed 63seconds

- Equipment list
- First compilation and help to port
- Optimization
- MPI
  - Setting up
  - Run Bench NAS CG

## Setting up

- ▶ `cd $HOME/TP/NPB3.2.1/NPB3.2-MPI`

# Run Bench NAS CG

## Compile

- ▶ make cg CLASS=B NPROCS=4

## Run with Infini band

- ▶ Run it
- ▶ Record time elapsed



# Run Bench NAS CG

## Compile

- ▶ make cg CLASS=B NPROCS=4

## Run with Infini band

- ▶ Run it
- ▶ Record time elapsed 63.17seconds

(To be continued)

- ▶ (c) Copyright Bull. All rights reserved
  - ✓ Users Restricted Rights - Use, duplication or disclosure restricted.
  - ✓ Any copy of these documents should keep all copyright, logos and other proprietary notices contained herein.
  - ✓ This publication may include technical inaccuracies or typographical errors.
  - ✓ This publication is provided "AS IS" without any warranty either expressed or implied including but not limited to the implied warranties of merchantabilities or fitness of the described product.
  - ✓ Course Material Licensing Terms : No sublicensing rights.
  - ✓ For other licensing needs, please contact Bull