www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

OmpSs-2 Hands-on

Rosa M. Badia, Xavier Martorell, Xavier Teruel, and Orestis Korakitis

Barcelona, September 6th 2022

Extract and configure oss-2-ee package

(1) Extracting the sources

```
$ tar -xvzf oss-2-ee.tar.gz
<list of extracted files>
```

(2) Main directory

```
$ cd oss-2-ee
$ ls
02-examples
03-fundamentals
04-mpi+ompss-2
05-cuda+ompss-2
06-openacc+ompss-2
local
configure
configure-dc
paraver-cfgs
README.md
```

(3) Configure

```
$ source configure
Basic configuration...
  Mercurium compiler at /apps/ompss/bin
  Extrae library at /apps/extrae/bin
  Paraver utility at /apps/wxparaver/bin
  ...
Aditional libraries...
MPI library at /opt/mpi/bullxmpi/.../lib
MKL library at /opt/.../mkl/lib/intel64/
ATLAS library at /opt/.../ATLAS/3.9.51/lib
```

(0) Remainder: exercises are in /apps/PM/training/PUMPS-2022

$ cp /apps/PM/training/PUMPS-2022/oss-2-ee.tar.gz $HOME

Building oss-2-ee package

(1) nn-session / exercise

```
$ cd 02-examples/cholesky-openblas
$
```

(2) Directory contents

```
$ ls
cholesky.c
cholesky.h
Makefile
README.rst
```

(3) README.rst file (script)

```
$ vi README.rst
$
```

(4) Run “make” will create…

```
$ make
Building: program
Creating: mutirun.sh, run-once.sh, trace.sh and
extrae.xml
```

- » an executable
- » different scripts to run your programs
- » an extrae.xml file (as default to get your Paraver traces)

```
$ ls
cholesky.c    cholesky.h    cholesky
extrae.xml    Makefile      multirun.sh
README.rst    run-once.sh   trace.sh
```

Running oss-2-ee package (CTE Power 9) [1]

Checking script configuration

```
$ vi run-once.sh
$
```

run-once.sh

```
#!/bin/bash
#SBATCH --job-name=oss-2-ee
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=12
#SBATCH --qos=debug
#SBATCH --gres=gpu:1
PROGRAM=cholesky
#INSTRUMENT=./xxxxx.sh
$INSTRUMENT ./$PROGRAM 4096 512 1
```

Submitting the script

```
$ sbatch run-once.sh
Submitted batch job 1234567
```

Check submitted jobs with:
$ squeue

Exercise directory

```
$ ls
cholesky.c      cholesky.h      cholesky
extrae.xml      Makefile        multirun.sh
README.rst      run-once.sh     trace.sh
```

Directory after execution

```
$ ls
cholesky.c          cholesky.h          cholesky
extrae.xml          Makefile            multirun.sh
oss-2-ee_nnn.err    oss-2-ee_nnn.out    README
run-once.sh         trace.sh
```

Running oss-2-ee package (CTE Power 9) [2]

Checking script configuration

```
$ vi run-once.sh
$
```

run-once.sh

```
#!/bin/bash
#SBATCH --job-name=oss-2-ee
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=12
#SBATCH --qos=debug
#SBATCH --g   =gpu:1
PROGRAM=chole
#INSTRUME
$INSTRUME
```

```
##SBATCH --qos=debug
#SBATCH --reservation = PUMPS+AI22
```

Submitting the script

```
$ sbatch run-once.sh
Submitted batch job 1234567
```

Exercise directory

```
$ ls
cholesky.c      cholesky.h      cholesky
extrae.xml      Makefile        multirun.sh
README.rst      run-once.sh     trace.sh
```

Directory after execution

```
$ ls
cholesky.c          cholesky.h          cholesky
extrae.xml          Makefile            multirun.sh
oss-2-ee_nnn.err    oss-2-ee_nnn.out    README
run-once.sh         trace.sh
```

Instrumenting oss-2-ee package

Checking configuration scripts

```
$ vi run-once.sh trace.sh
$
```

run-once.sh

```
#!/bin/bash
#SBATCH --job-name=oss-2-ee
INSTRUMENT=./trace.sh  or  INSTRUMENT=./graph.sh
#SBA      --qos=debug
#SBA      --gres=gp
PROGRAM=cholesky
#INSTRUMENT=./xxxxxx.sh
$INSTRUMENT./$PROGRAM 4096 512 1
```

trace.sh

```
#!/bin/bash
. . .
export EXTRAE_CONFIG_FILE=extrae.xml
export LD_PRELOAD=/path/to/extrae/lib?

$*
```

Exercise directory

```
$ ls
cholesky.c      cholesky.h      cholesky
extrae.xml      Makefile        multirun.sh
README.rst      run-once.sh     trace.sh
```

Submitting the script

```
$ sbatch run-once.sh
Creating: cholesky.prv, cholesky.pcf cholesky.raw
```

Compile and execute (examples)

Exercise's location: 02-examples

```
~oss-2-ee:$ ls
02-examples                 03-fundamentals
04-mpi+ompss-2              05-cuda+ompss-2
06-openacc+ompss-2          ...
```

Compile and execute (guidelines)

- Code is completely annotated, you DON'T need to modify it. Review source code, check different directives and their clauses
- Check scalability → compute speed-up
- Obtain traces, explore them
- Check the README file (more goals)

Hands-on session's contents

~~Exercise 2.1: cholesky [-mkl][-openblas]~~
Exercise 2.2: matmul [ ] [-dep] ~~[-red] [-wdep]~~

Remember to configure your system

```
$ source configure
Basic configuration...
```

Check running scripts (before submit)

```
$ vi run-once.sh
$
```

- Scripts run-once.sh, multi-run.sh or trace.sh
- Job scheduler configuration front-matter: queue, hardware resources, etc.
- Other runtime's options used in the script

Fundamentals

Exercise's location: 03-fundamentals

```
~oss-2-ee:$ ls
02-examples                 03-fundamentals
04-mpi+ompss-2              05-cuda+ompss-2
06-openacc+ompss-2          ...
```

OmpSs-2 Fundamentals (guidelines)

- Code are incomplete
  - » Lack of clauses (in some cases)
  - » Lack of directives (is some others)
- Check scalability → compute speed-up
- Obtain traces, explore them
- Check the README file (more goals)

Hands-on session's contents

Continue previous session's exercises
Exercise 3.1: dotproduct [-dep] [-red]
Exercise 3.2: axpy ~~[-]~~ ~~[-dep]~~ [-wdep]

Remember to configure your system

```
$ source configure
Basic configuration...
```

Check running script (before submit)

```
$ vi run-once.sh
$
```

- Scripts run-once.sh, multi-run.sh or trace.sh
- Job scheduler configuration front-matter: queue, hardware resources, etc.
- Other runtime's options used in the script

OmpSs-2 + CUDA

Exercise's location: 05-cuda+ompss-2

```
oss-2-ee:$ ls
02-examples                 03-fundamentals
04-mpi+ompss-2              05-cuda+ompss-2
06-openacc+ompss-2          ...
```

OmpSs-2 + CUDA (guidelines)

- Review source code, check different directives and their clauses
- Check scalability → compute speed-up
- Obtain traces, explore them
- Check the README file (more goals)

Hands-on session's contents

Continue previous session's exercises
Exercise 5.1: SAXPY kernel
Exercise 5.2: Matmul kernel
Exercise 5.3: Nbody kernel

Remember to configure your system

```
$ source configure
Basic configuration...
```

Check running script (before submit)

```
$ vi run-once.sh
$
```

- Scripts run-once.sh, multi-run.sh or trace.sh
- Job scheduler configuration front-matter: queue, hardware resources, etc.
- Other runtime's options used in the script

OmpSs-2 + OpenACC

Exercise's location: 06-openacc+…

```
oss-2-ee:$ ls
02-examples                 03-fundamentals
04-mpi+ompss-2              05-cuda+ompss-2
06-openacc+ompss-2          ...
```

OmpSs-2 + OpenACC (guidelines)

- Review source code, check different directives and their clauses
- Check scalability → compute speed-up
- ~~Obtain traces, explore them~~
- Check the README file (more goals)

Hands-on session's contents

Continue previous session's exercises
Exercise 6.1: SAXPY kernel
Exercise 6.2: Matmul kernel
Exercise 6.3: Nbody kernel

Remember to configure your system

```
$ source configure
Basic configuration...
```

Check running script (before submit)

```
$ vi run-once.sh
$
```

- Scripts run-once.sh, multi-run.sh
- Job scheduler configuration front-matter: queue, hardware resources, etc.
- Other runtime's options used in the script

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Intellectual Property Rights Notice

The User may only download, make and retain a copy of the materials for his/her use for non-commercial and research purposes. The User may not commercially use the material, unless has been granted prior written consent by the Licensor to do so; and cannot remove, obscure or modify copyright notices, text acknowledging or other means of identification or disclaimers as they appear. For further details, please contact BSC-CNS.

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Thank you!

For further information please contact

https://www.linkedin.com/in/xteruel

Intellectual Property Rights Notice

The User may only download, make and retain a copy of the materials for his/her use for non-commercial and research purposes. The User may not commercially use the material, unless has been granted prior written consent by the Licensor to do so; and cannot remove, obscure or modify copyright notices, text acknowledging or other means of identification or disclaimers as they appear. For further details, please contact BSC-CNS.

Visualizing paraver traces [1/5]

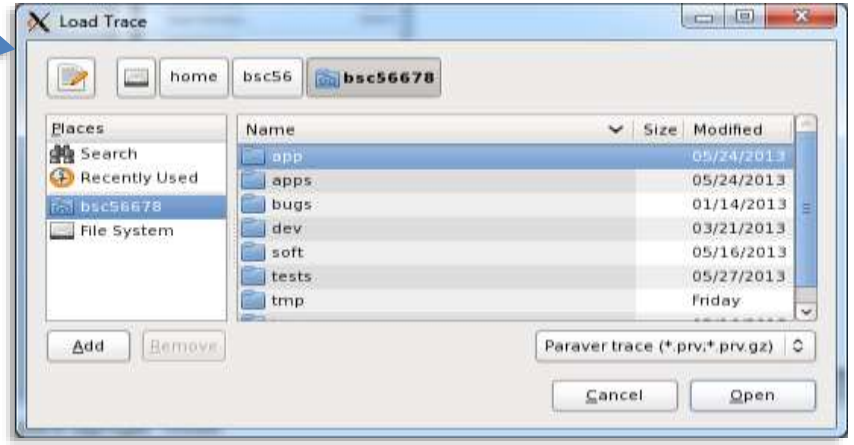
Running paraver tool

```
$ paraver
$
```

— Load a Paraver trace

Suite exercise directory

```
$ ls
cholesky.c          cholesky-d
cholesky.h          cholesky-i
cholesky-i.pcf      cholesky-i.prv
cholesky-i.raw      cholesky-p
extrae.xml          Makefile
multirun.sh         README.rst
run-once.sh         trace.sh
```



Visualizing paraver traces [2/5]

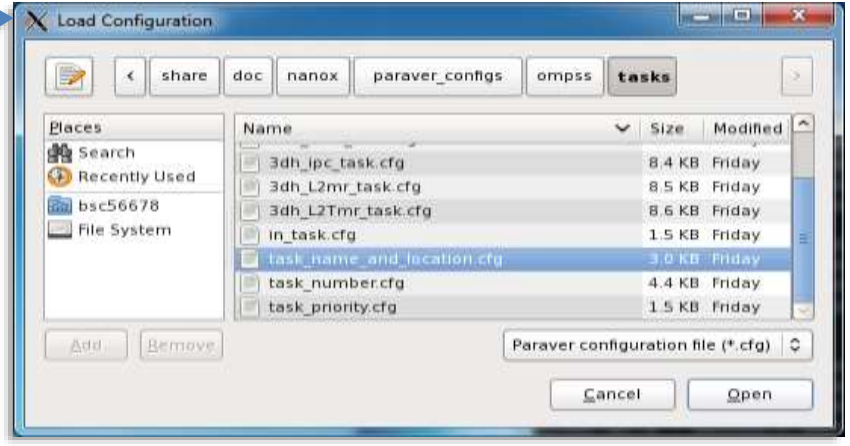
Running paraver tool

```
$ paraver
$
```

— Load a Paraver trace
— Load a configuration file

Suite root directory

```
$ ls
01-examples              02-beginners
03-gpu-devices           04-mpi+ompss
common-files             configure.sh
paraver-cfgs             README.rst
```



Visualizing paraver traces [3/5]

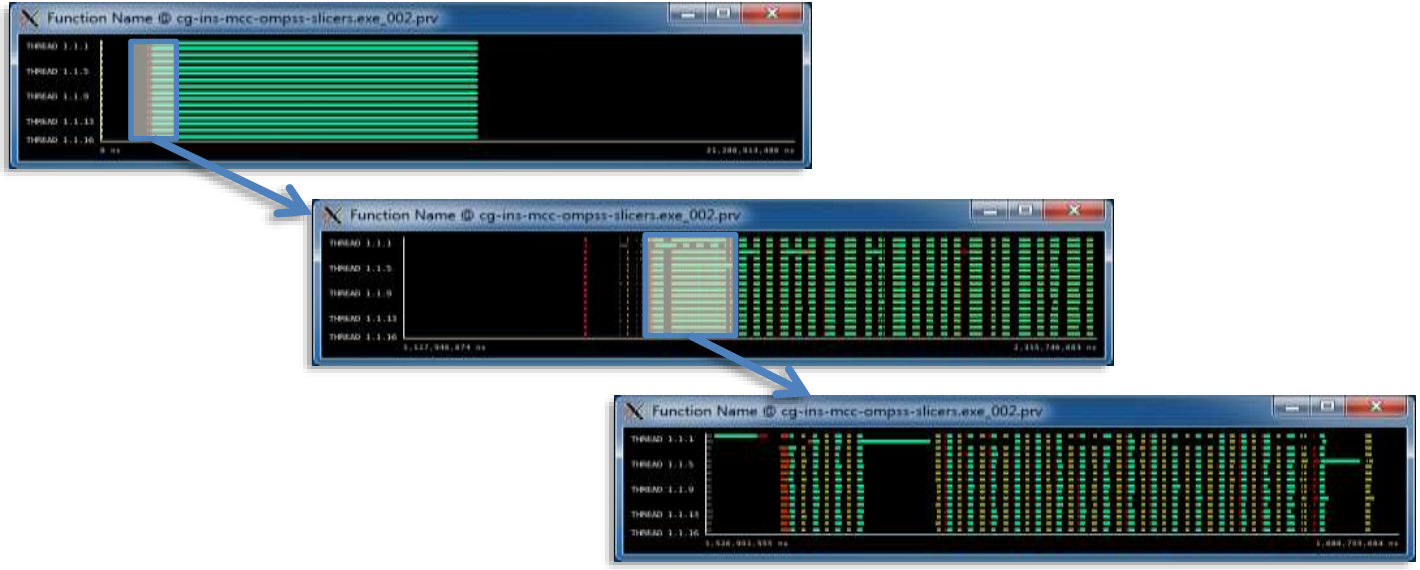
Running paraver tool

```
$ paraver
$
```

- Load a Paraver trace
- Load a configuration file
- Trace analysis (zoom in, details)

Suite root directory

```
$ ls
01-examples             02-beginners
03-gpu-devices          04-mpi+ompss
common-files            configure.sh
paraver-cfgs            README.rst
```



Visualizing paraver traces [4/5]

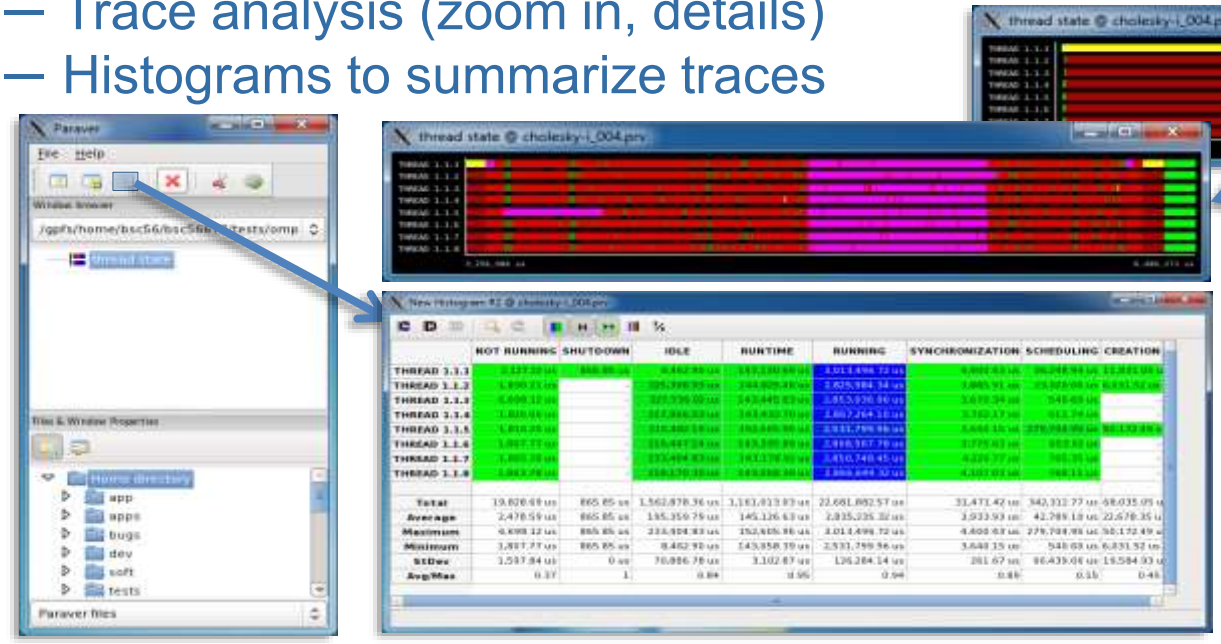
Running paraver tool

```
$ paraver
$
```

- Load a Paraver trace
- Load a configuration file
- Trace analysis (zoom in, details)
- Histograms to summarize traces

Suite root directory

```
$ ls
01-examples              02-beginners
03-gpu-devices           04-mpi+ompss
common-files             configure.sh
paraver-cfgs             README.rst
```



Visualizing paraver traces [5/5]

Running paraver tool

```
$ paraver
$
```

- Load a Paraver trace
- Load a configuration file
- Trace analysis (zoom in, details)
- Histograms to summarize traces
- Other configuration files
  - » ompss / runtime / thread_state.cfg
  - » ompss / runtime / nanos_API.cfg
  - » ompss / tasks / task_name_and_location.cfg
  - » ompss / cuda / …
  - » hwc / papi / performance / …

Suite root directory

```
$ ls
01-examples             02-beginners
03-gpu-devices          04-mpi+ompss
common-files            configure.sh
paraver-cfgs            README.rst
```

- more info about paraver instrumentation tool

http://pm.bsc.es/ompss-docs/user-guide