Tools Hands-on

Judit Gimenez
judit@bsc.es

NERSC - Berkeley, August 2016
Getting a trace with Extrae
**Extrae features**

- **Parallel programming models**
  - MPI, OpenMP(*), pthreads, OmpSs, CUDA, OpenCL, Java, Python...

- **Platforms**
  - Intel, Cray, BlueGene,, Fujitsu Sparc, Xeon Phi,, ARM, Android...

- **Performance Counters**
  - Using PAPI

- **Link to source code**
  - Callstack at MPI routines
  - OpenMP outlined routines
  - Selected user functions (Dyninst)

- **Periodic sampling**

- **User events (Extrae API)**
How does Extrae work?

**Symbol substitution through LD_PRELOAD**
- Specific libraries for each combination of runtimes
  - MPI
  - OpenMP
  - OpenMP+MPI
  - ...

**Dynamic instrumentation**
- Based on DynInst (developed by U.Wisconsin/U.Maryland)
  - Instrumentation in memory
  - Binary rewriting

**Alternatives**
- Static link (i.e., PMPI, Extrae API)
## Extrae overheads

<table>
<thead>
<tr>
<th>Event</th>
<th>Average values</th>
<th>Cori (intel – gnu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event</td>
<td>120-200 ns</td>
<td>120 - 140 ns</td>
</tr>
<tr>
<td>Event + PAPI</td>
<td>725 ns – 1 us</td>
<td>730 - 725 us</td>
</tr>
<tr>
<td>Event + callstack (1 level)</td>
<td>600 ns</td>
<td>570 - 518 ns</td>
</tr>
<tr>
<td>Event + callstack (6 levels)</td>
<td>2 us</td>
<td>1.8 us</td>
</tr>
</tbody>
</table>
How to use Extrae

1. Adapt the job submission script

2. (Optional) Tune the Extrae XML configuration file
   - Examples distributed with Extrae at $EXTRAE_HOME/share/example

3. Run with instrumentation

For further reference check the Extrae User Guide

- Also distributed with Extrae at $EXTRAE_HOME/share/doc
- http://www.bsc.es/computer-sciences/performance-tools/documentation
Log in to cori and copy the example to your home

@ your laptop

```
> ssh -X <USER>@cori.nersc.gov
```

Copy the examples to your home folder:

@ cori.nersc.gov

```
> cp -r ~jEdit/handson_extrae $HOME

> ls -l $HOME/handson_extrae
  ... lulesh
  ... mpi_ping
  ... extrae
```
Adapt the job script to load Extrae (LD_PRELOAD)

> vi $HOME/handson_extrae/lulesh/job1n.sh

job1n.sh

#!/bin/bash -l
#SBATCH --partition=debug
#SBATCH --nodes=1
#SBATCH --time=00:10:00
#SBATCH --license=SCRATCH  #note: specify license need for the file systems your job needs, such as SCRATCH,project

module swap PrgEnv-intel PrgEnv-gnu

srun -n 27 ./lulesh2.0 -i 10 -p -s 65
Adapt the job script to load Extrae (LD_PRELOAD)

> vi $HOME/handson_extrae/lulesh/job1n.sh

---

```bash
#!/bin/bash -l
#SBATCH --partition=debug
#SBATCH --nodes=1
#SBATCH --time=00:10:00
#SBATCH --license=SCRATCH  #note: specify license need for the file systems your job needs, such as SCRATCH,project

module swap PrgEnv-intel PrgEnv-gnu
module load extrae

TRACE=../extrae/trace.sh
export TRACE_NAME=lulesh27_1node.prv

srun -n 27 $TRACE ./lulesh2.0 -i 10 -p -s 65
```
Adapt the job script to load Extrae (LD_PRELOAD)

> vi $HOME/handson_extrae/lulesh/job1n.sh

```bash
#!/bin/bash -l
#SBATCH --partition=debug
#SBATCH --nodes=1
#SBATCH --time=00:10:00
#SBATCH --license=SCRATCH  #note: specify license need for the file systems your job needs, such as SCRATCH, project
module swap PrgEnv-intel PrgEnv-gnu
module load extrae

TRACE=../extrae/trace.sh
export TRACE_NAME=lulesh27_1node.prv
srun -n 27 $TRACE ./lulesh2.0 -i 10 -p -s 65
```

trace.sh

```bash
#!/bin/bash
export EXTRA_HOME=${EXTRA_DIR}
export EXTRA_CONFIG_FILE= $HOME/handson_extrae/extrae/extrae.xml

export LD_PRELOAD=${EXTRA_HOME}/lib/libmpitrace.so
# For C apps
#export LD_PRELOAD=${EXTRA_HOME}/lib/libmpitracef.so
# For Fortran apps

## Run the desired program
*$
```

Pick tracing library
Choose depending on the application type

<table>
<thead>
<tr>
<th>Library</th>
<th>Serial</th>
<th>MPI</th>
<th>OpenMP</th>
<th>pthread</th>
<th>CUDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>libseqtrace</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>libmpitrace[f]₁</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>libompitrace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>libpttrace</td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>libcudatrace</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✔</td>
</tr>
<tr>
<td>libompitrace[f]₁</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>libptmpitrace[f]₁</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>libcudampitrace[f]₁</td>
<td>✔</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>

¹ include suffix “f” in Fortran codes
Extrae XML configuration

```xml
<mpi enabled="yes">
  <counters enabled="yes" />
</mpi>

<openmp enabled="yes">
  <locks enabled="no" />
  <counters enabled="yes" />
</openmp>

<pthread enabled="no">
  <locks enabled="no" />
  <counters enabled="yes" />
</pthread>

<callers enabled="yes">
  <mpi enabled="yes">1-3</mpi>
  <sampling enabled="no">1-5</sampling>
</callers>
```

Trace MPI calls + HW counters

Trace call-stack events @ MPI calls
Extrae XML configuration (II)

```xml
<counters enabled="yes">
  <cpu enabled="yes" starting-set-distribution="cyclic">
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L1_DCM,PAPI_L3_TCM,PAPI_BR_CN,PAPI_BR_UCN
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_BR_MSP,PAPI_SR_INS,PAPI_LD_INS,RESOURCE_STALLS:SB
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS,PAPI_TOT_CYC,PAPI_L2_DCM,RESOURCE_STALLS:ROB,RESOURCE_STALLS:RS
    </set>
    <set enabled="yes" domain="all" changeat-time="500000us">
      PAPI_TOT_INS,PAPI_TOT_CYC,RESOURCE_STALLS
    </set>
  </cpu>
  <network enabled="no" />
  <resource-usage enabled="no" />
  <memory-usage enabled="no" />
</counters>
```

Select which HW counters are measured
<buffer enabled="yes">
    <size enabled="yes">500000</size>
    <circular enabled="no" />
</buffer>

<sampling enabled="no" type="default" period="50m" variability="10m" />

<merge enabled="yes"
    synchronization="default"
    tree-fan-out="16"
    max-memory="512"
    joint-states="yes"
    keep-mpits="yes"
    sort-addresses="yes"
    overwrite="yes"
> $TRACE_NAME$
</merge>
Run with instrumentation

Submit your job

@ cori.nersc.gov

```
> cd $HOME/handson_extrae
> sbatch job1n.sh
```

– Once finished (check with “squeue”) you will have the trace:
  • Lulesh27_1node.{prv,pcf,row} (3 files)
First steps of analysis

**Start Paraver**

Start Paraver

```
$ source $HOME/handson_extrae/setup_bsctools.sh
$ wxparaver &
```

**Load the trace**

Load the trace

**Follow Tutorial #3**

Follow Tutorial #3

- Introduction to Paraver and Dimemas methodology
Is the application efficient?

**Click on “mpi_stats.cfg”**
- Check the Average for the column labeled “Outside MPI”
  - Parallel efficiency is high?

**Click on “2dh_usefulduration.cfg”**
- Histogram of duration of the computing regions
  - Vertical lines or dispersion? Fast/slow processes?

**Click on “2dh_useful_instructions.cfg”**
- Histogram of instructions of the computing regions
  - Work is well distributed?
The trace is HUGE?

Chop a region of interest from the trace-file

1. Filter the trace discard very small regions

$PARAVER_HOME/bin/paramedir -f filter.xml <trace.prv>

Example file in cori:
~judit/handson-exrae/toolsCfgs/filter.xml

• This gets a trace showing only the most relevant regions for the whole run

2. On the filtered trace: File → Load configuration → cfgs/General/view/useful-duration.cfg
   • Identify a periodic pattern / small region of interest

3. Apply the “scissors” tool → Cut the region of interest from the big trace
   • This gets a small trace with all the details for a small time interval
Installing Paraver in your laptop
Installing Paraver in your laptop (I)

Download the Paraver binaries to your laptop

@ your laptop

> scp <USER>@cori.nersc.gov:~jedit/packages/<VERSION> $HOME

Pick your version

- Linux 64 bits: wxparaver-4.6.2-linux-x86_64.tar.gz
- Linux 32 bits: wxparaver-4.6.2-linux-x86_32.tar.gz
- Mac: wxparaver-4.6.2-mac.zip
- Windows: wxparaver-4.6.2-win.zip
Installing Paraver in your laptop (II)

Uncompress the package into your home directory

```shell
@ your laptop

> tar xvzf wxparaver-4.6.2-linux-x86_64.tar.gz -C $HOME
> ln -s $HOME/wxparaver-4.6.2-linux-x86_64 $HOME/paraver
```

Download Paraver tutorials and uncompress into the Paraver directory

```shell
@ your laptop

> scp <USER>@cori.nersc.gov:~judit/packages/paraver-tutorials.tar.gz $HOME
> tar xvzf $HOME/paraver-tutorials.tar.gz -C $HOME/paraver
```
Check that everything works

Start Paraver

@ your laptop

> $HOME/paraver/bin/wxparaver &

Check that tutorials are available

Click on Help → Tutorials