Performances of parallel codes

Performances of parallel codes can be evaluated by means of dedicated applications like profilers. A profiler is a software specifically designed to analyze the behaviour of parallel programs in order to improve and optimize them. There are many profilers around, some of them commercial, others open source. It is higly recommended to use a toolset for analyzing own applications, for such a good practise will make the code faster and more efficient, preventing waste of time.

In this section we present Scalasca, an open source toolset developed as a joint project of the Jülich Supercomputing Centre and theGerman Research School for Simulation Sciences. Scalasca has been installed on CRESCO HPC system for the Linux platform. The current installation supports GNU, Intel and PGI compilers with many MPI flavours. Moreover, users can also use Scalasca on CRESCO for analyzing hybrid (MPI + OpenMP) codes.

How to use Scalasca on CRESCO

For using Scalasca on CRESCO, users have to load the appropriate environment by typing in at the prompt the command **`scalasca`**. Examples and detailed information can be found in the **`/example`** and **`/doc`** directory as suggested by the command itself. Once loaded the Scalasca environment, a cursory look at the main features can be done by giving the command **`scalasca -h`**.

The basic steps in using Scalasca for analyzing a program involve three phases: program instrumentation, measurement execution and analysis report:

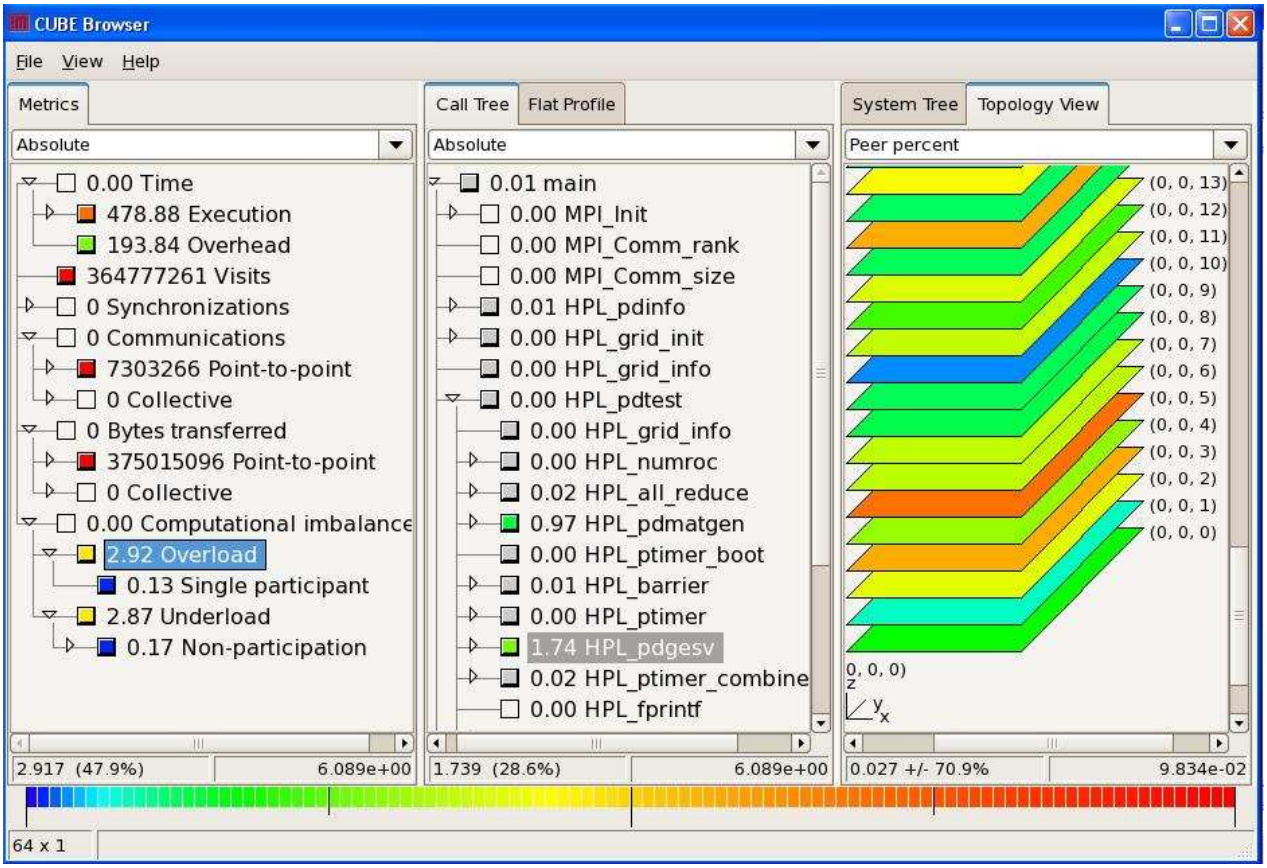
- **`scalasca -instrument`** `mpicc mympiprog.c -o mympiprog`
- **`scalasca -analyze`** `mpirun -np #cores ./mympiprog`

After successful execution of the job analysis a summary report file is created within a measurement directory called `epik_*`. The summary analysis report can be postprocessed by typing the command:

- **`scalasca -examine`** `epik_*`

Scalasca comes with the GUI CUBE which allows the programmer to analyse the main metrics of the executable for eventually proceeding with further investigations.

A snapshot of CUBE



It is possible to submit a Scalasca job to CRESCO in batch mode. This is accomplished by following the simple procedure described below.

- **Step 0: Optional**

  If the job will produce a huge amount of data or I/O, it should be better to use the ENEA GRID Parallel File System (GPFS).

```
cd ~/PFS/por
mkdir myfolder
cd myfolder
```

- **Step 1: Load Scalasca**

  Load Scalasca environment by typing

```
scalasca
```

- **Step 2: Instrument the code**

```
scalasca -instrument mpicc myprog.c -o myprog
```

- **Step 3: Submit to LSF**

  Use a submition script from inside the Scalasca environment

```
bsub -o myoutput -e myerror -n #cores -q quename ./submit.sh
```

  Where the script looks like:

```
#!/bin/sh

PROCS=`cat $LSB_DJOB_HOSTFILE |
wc -l`
```

```
# General Scalasca environment
variables
export SCAN_MPI_LAUNCHER=mpirun
export SCAN_MPI_RANKS=$PROCS


scalasca -analyze \
$SCAN_MPI_LAUNCHER "-np
$SCAN_MPI_RANKS" \
"--mca pls_rsh_agent blaunch.sh"
\
"--hostfile $LSB_DJOB_HOSTFILE"
\
-- ./myprog
```

Pay attention to double quote the strings denoting `mpirun` parameters. Often it is necessary to put a double hyphen "--" before the target executable. More details can be found in the Scalasca User Guide. Also, it is recommended to read the OPEN_ISSUES file.