

Parallel and Distributed Computing

ompP Tutorial

The objective of this tutorial is to familiarize the students with the ompP profiling tool. Throughout this tutorial you should consult the User Guide and Manual of ompP.

1 ompP

ompP is a profiling tool for OpenMP applications written in C/C++ or FORTRAN. ompP's profiling report becomes available immediately after program termination in a human-readable format.

2 Usage

ompP is implemented as a static library that is linked to your application. To capture OpenMP execution events, ompP relies on source-to-source instrumentation. To instrument your application and link it with ompP's monitoring library, simply prefix any compile or link command with `kinst-ompp`, i.e., on a shell prompt:

```
$> gcc -fopenmp foo.c bar.c -o myapp  
becomes  
$> kinst-ompp gcc -fopenmp foo.c bar.c -o myapp
```

3 Examples

Download the example programs `mult-sync.c`, `mult-threads.c` and `mult-good.c`. For each one of them do:

- Compile the example with ompP support as described above;
- Run the executable;
- Check the profiling file generated.

For each one of the profiling files generated compare in particular:

- the global execution times;
- the synchronization times;
- the cost of creating and destroying threads.

Note: The tool that instruments the code uses the same file every time. Before each instrumentation you should remove the file `opari.rc`