

Performance Tools

OVERVIEW

Parallel architectures enable to target more complex and ambitious problems each year. But in many cases, the achieved performance is far away from what the theoretical values promised us. Performance analysis tools allow application developers to identify and characterize the inefficiencies that caused a poor performance. We consider that this analysis must be the first step towards the optimization of an application. Optimizing without a previous analysis could be like driving without directions as it could mean wasting efforts improving parts of the code that were not the real performance bottlenecks.

In 1991 we started to work on the development of performance analysis tools, initially only for internal use and distributed since 2000. Flexibility, simplicity and the ability to interact between qualitative and quantitative information were the keys we considered most important in the design of the tools 15 years ago. These features allow us to keep working with the same tools.

OBJECTIVES

The main objective of the team is to be able to use efficiently our tools to solve any performance problem or question we face (directly as users of the tools or through external users). This means that our tools should be easily adaptable to new platforms, environments or programs, and should be able to scale, extend and evolve in the same way that applications, platforms and programming models do.

We consider that performance analysis is, in some sense, still an art where the experience and intuition of the analyst drives the analysis and determines the quality of the results. For this reason a second objective is to work in the definition of methodologies and procedures that would simplify and facilitate the process of extracting information from the performance data. Our belief is that if performance analysis do not require a special skill or expertise, more people would be interested in applying it.

PROJECTS/AREAS

The set of performance tools that we develop, named CEPBA-Tools, is currently comprised by:

- **Paraver:** A very powerful performance visualization and analysis tool based on traces that can be used to analyse any information that is expressed on its input trace format.
- **Dimemas:** Simulation tool for the parametric analysis of the behaviour of message-passing applications on a configurable parallel platform.
- **Instrumentation packages:** Set of programs and libraries to generate or translate Paraver and Dimemas traces. We have packages for instrumenting different programming models (MPI, OpenMP, mixed) under different platforms (IBM-AIX, Linux, SGI-IRIX, HP-Alpha) and translators from IBM-AIXtrace and LTT formats.
- **Utilities:** Set of small programs to process Paraver traces to cut, summarize, translate or accumulate the performance data. They can be used independently but they are also integrated within Paraver.