Handling CPU affinity of Code in the Simulation of Parallel Applications

Frédéric Suter

October 15, 2015

1 Motivations

Analyzing and understanding the performance behavior of parallel applications on various compute infrastructures is a long-standing concern in the High Performance Computing community. When the targeted execution environments are not available, simulation is a reasonable approach to obtain objective performance indicators and explore various hypothetical scenarios. In the context of applications implemented with the Message Passing Interface (MPI), two simulation methods have been proposed, on-line simulation and off-line simulation, both with their own drawbacks and advantages. In off-line simulation, a trace of a previous execution of the application is "replayed" on a simulated platform. The main advantage when compared to on-line simulation is that replaying the execution can be performed on a single computer. This is because the replay does not entail executing any application code, but merely simulating computation and communication delays. While the simulation of communications is a well-covered area, with the proposition of accurate models [1], that of computations usually remains rather simple. Indeed, all off-line simulators of MPI applications rely on time-stamped traces, i.e., each traced event is associated to a date, obtained on a homogeneous and at scale platform. It is then possible to apply a uniform scaling when simulating a target platform with slower/faster processors. But, as a result, trace acquisition is not easily scalable since a homogeneous platform may not be available at the required scale.

To solve the above trace acquisition scalability problem, we proposed the Time-Independent Trace Replay Framework[2]. This is achieved by eliminating time-stamps altogether from application traces. Then combined this framework with ScalaTrace, a tool that produces compact traces[3]. An application then corresponds to a list of *blocks* performed by each process of an MPI application.

While this framework allows for the acquisition of large traces on any platforms, e.g., heterogeneous multi-clusters, it raises two important questions: "**How to take the CPU affinity of computing blocks during the simulation?**" and "**How to calibrate the simulator with an appropriate CPU processing rate for the target without any timing information?**" In our previous works, we computed an *average* instruction rate on a *calibration instance* of the studied application. This rate was then used by the simulator. However, this method is a rough estimation and hinders the accuracy of our simulation.

2 Subject

The main objective of this internship is to leverage the block structure of the framework proposed in[3] to better take their CPU affinity in the calibration of the CPU processing rate for the off-line simulation of MPI applications. To achieve this, the candidate will have to study and seize earlier work on that topic. Then s/he will analyze existing execution traces, and maybe acquire new ones, to propose a characterization of the respective CPU affinity. Depending on the first obtained results, the candidat may have to propose a way to extrapolate information gathered on calibration instances to target, i.e., large instances of the studied application(s). Finally the candidate will conduct a thourough evaluation of the impact of the CPU affinity on the accuracy of simulation results obtained thanks to the SimGrid toolkit [4] with regard to actual experimentations.

To favor an Open Science approach, the candidate will be encouraged to document and made available all the processes and data used during the internship. This may be eased by tools such as the org mode of the Emacs editor.

3 Environment

The candidate will be jointly hosted by the research team of the IN2P3 Computing Center (CC IN2P3)[1] and the AVALON team[2] of the Laboratoire de l'Informatique du Parallélisme (UMR 5668) at the École Normale Supérieure de Lyon. CC IN2P3 is a service and research unit belonging to CNRS (USR 6402). A major French research infrastructure, it is responsible for providing computing and data storage resources for researchers involved in corpuscular physics experiments. The main services offered by CC-IN2P3 are the storage and processing of large volumes of data and the transfer of these data over very high- speed international networks. CC-IN2P3 is one of eleven centers worldwide engaged in the primary processing of LHC data and one of only four centers that will provide storage and data processing capacity for all four LHC experiments.

References

[1] Clauss PN, Stillwell M, Genaud S, Suter F, Casanova H, Quinson M. Single Node On-Line Simulation of MPI Applications with SMPI. *Proc. of the 25th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Anchorage, AK, 2011.

[2] Casanova H, Desprez F, Markomanolis G, Suter F. Simulation of MPI Applications with Time-Independent Traces. *Concurrency and Computation: Practice and Experience*, 27(5):1145-1168, April 2015.

[3] Casanova H, Gupta A, Suter F. Toward More Scalable Off-Line Simulations of MPI Applications. *Parallel Processing Letters*, 25(3):1541002, September 2015.

[4] Casanova H, Giersch A, Legrand A, Quinson M, Suter F. Versatile, Scalable, and Accurate Simulation of Distributed Applications and Platforms. *Journal of Parallel and Distributed Computing*, 74(10):2899-2917, October 2014.

[1] `http://cc.in2p3.fr`

[2] `http://avalon.ens-lyon.fr/`