Simple and Efficient Parallel Programming for Distributed-memory Systems

Universidad de Valladolid

Ana Moreton-Fernandez Arturo Gonzalez-Escribano Diego R. Llanos

Grupo Trasgo
Universidad de Valladolid

Twelfth International Summer School on Advanced Computer Architecture and Compilation for High-Performance and Embedded Systems (ACACES), 10-16 July 2016, Fiuggi, Italy

Introduction

Several techniques and frameworks have been proposed to automatically generate parallel programs for hybrid distributed-memory systems from high-level parallel languages or sequential codes. These techniques should take into account the combination of data communication across distributed processes, and the exploitation of shared-memory models inside each process.

Trasgo is a new programming model and compilation framework to generate parallel programs from a high-level parallel specification, based on the SPMD (Single Program Multiple Data) model. It proposes the use of a global-view approach, with flexible and explicit mechanisms to deal with locality.

Trasgo

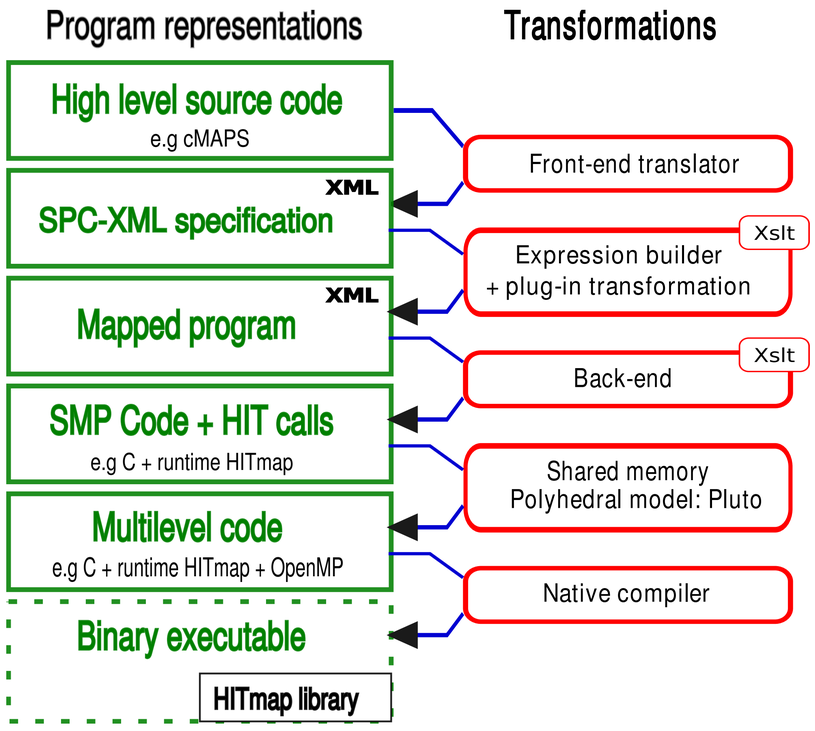

Figure 1 : Trasgo architecture.

Trasgo [1] is a source-to-source parallel compilation system for distributed- and shared-memory machines.

- **Front-end** generates SPC-XML intermediate code.
- **Expression builder** adds annotations, new tags, and transforms the initial SPC-XML document into a new one.
- **Back-end** translates this XML code into a target code. The code generated by Trasgo uses the Hitmap library [2] to perform efficient data distributions and communications.
- **Pluto** is a state-of-the-art tool based on polyhedral model, to automatically parallelize sequential parts [3]. We integrate it to parallelize code for shared-memory platforms.

Compiler-Runtime system

We have developed and implemented in Trasgo several techniques to transform a high-level code to a low-level program. For example, one of this techniques calculates automatically at runtime the communications needed in a distributed programming model between two SPMD blocks with distributed data structures. The technique is based on a compile-time analysis that, from a set of affine expressions generates tailored functions to calculate at runtime the set of indexes accessed to read or to write in a SPMD block.

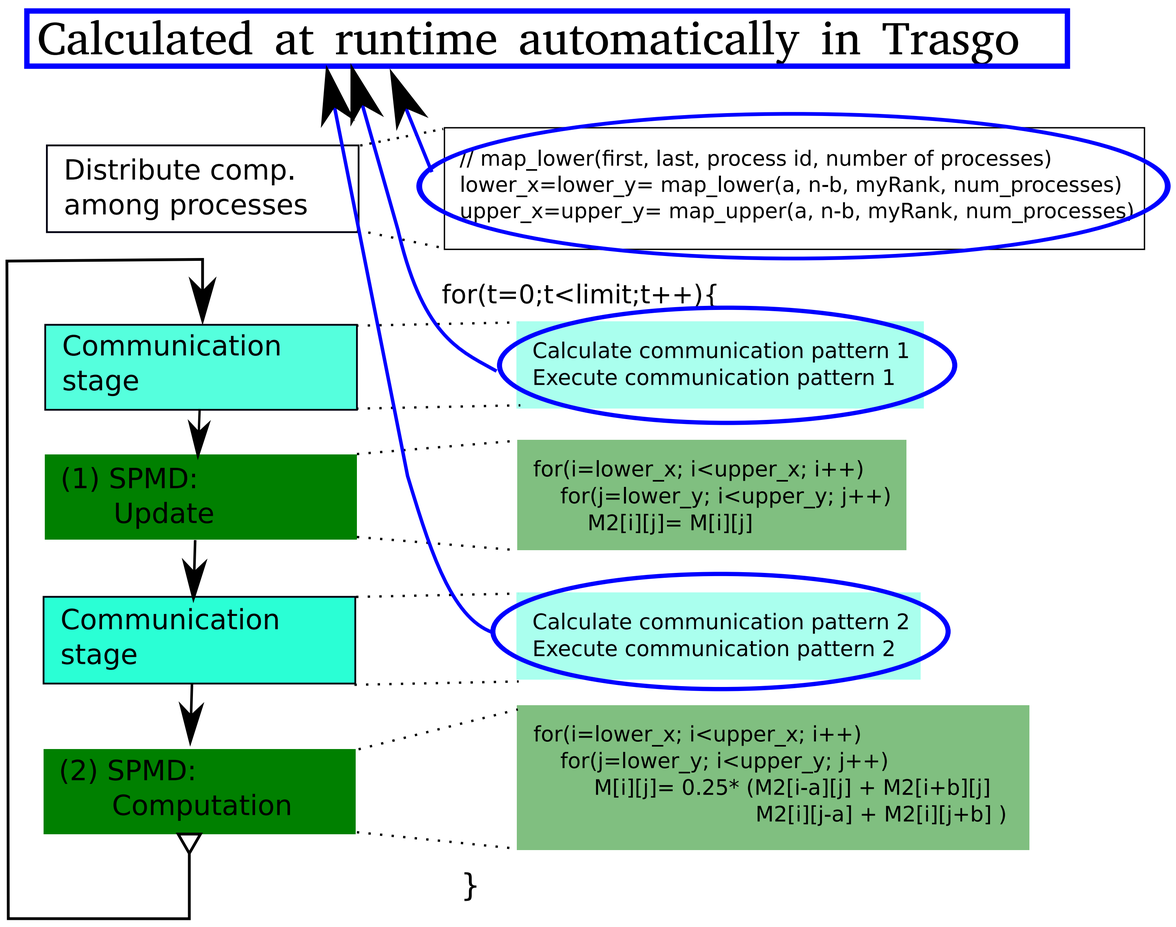

Figure 2 : Scheme of a code programmed with the SPMD model.

Performance

We show the sequential, communication calculation and communication execution times for several number of distributed processes, using our runtime approach in two representative applications.

The results indicate that our runtime calculation is insignificant compared with the sequential, and communication times.

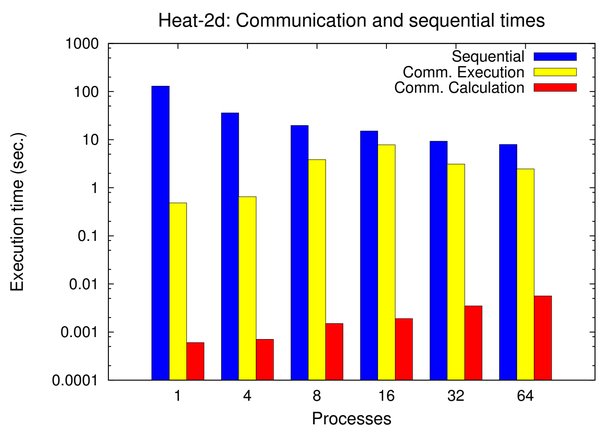

Figure 3 : Execution times for the Heat-2d example (Size=8000x8000, iter=500).

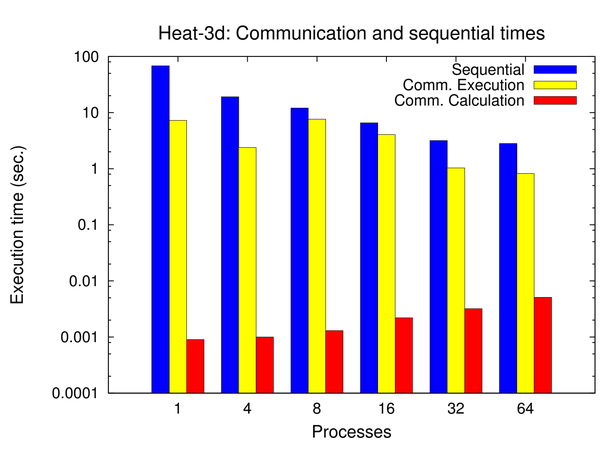

Figure 4 : Execution times for the Heat-3d example (Size=500x500x500, iter=100).

Conclusions

- We have designed and developed a framework to help to the programmer to deal with the main challenges on distributed-memory systems, data partition and communication.
- The techniques implemented in our framework enable to exploit many capabilities of the execution machines in an abstract way to the programmer (as for example, choosing the tile size at runtime).

References

[1] A. Gonzalez-Escribano and D. Llanos, "Trasgo: A nested-parallel programming system," The Journal of Supercomputing, vol. 58, no. 2, pp. 226–234, 2011.

[2] A. Gonzalez-Escribano, Y. Torres, J. Fresno, and D. Llanos, "An extensible system for multilevel automatic data partition and mapping," IEEE TPDS, vol 5, no. 5, pp. 1145–1154, 2013.

[3] U. Bondhugula, A. Hartono, J. Ramanujam, and P. Sadayappan, "A practical automatic polyhedral program optimization system". In ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), June 2008.

Acknowledgements This research has been partially supported by MICINN (Spain) and ERDF program of the European Union: HomProg-HetSys project (TIN2014-58876-P), and COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS). by the computing facilities of Extremadura Research Centre for Advanced Technologies (CETA-CIEMAT), funded by the European Regional Development Fund (ERDF). CETA-CIEMAT belongs to CIEMAT and the Government of Spain.