A Runtime Analysis For Communication Calculation

Ana Moreton-Fernandez, Arturo Gonzalez-Escribano and Diego R. Llanos
Universidad de Valladolid, Spain
{ana | arturo | diego}@infor.uva.es

Universidad de Valladolid

Grupo Trasgo
Universidad de Valladolid

Introduction

- Several automatic code generation approaches transform sequential or high-level parallel codes, to low-level parallel programs for **distributed-memory clusters**.
- They abstract many issues related to the execution platform, while also deliver good performance.
- However, they generate a generic code that cannot take into account some specific details about the execution machine.

Related Work

- **Task-oriented approaches** imply performance penalties in distributed-memory systems because of: Management of distributed queues, synchronization and load balancing mechanisms, or data communications due to dynamic task scheduling and/or migration.
- **Static-scheduled approaches**: They are compile-time solutions with their corresponding constraints: Compile-time choices, or compiler scalability problems.

References

[1] A. Gonzalez-Escribano, Y. Torres, J. Fresno, and D.R. Llanos. An extensible system for multilevel automatic data partition and mapping. *IEEE TPDS*, 25(5):1145–1154, 2013. (doi:10.1109/TPDS.2013.83).

[2] Uday Bondhugula. Compiling affine loop nests for distributed-memory parallel architectures. In *2013 SC-International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12. IEEE, 2013.

[3] Martin Kong, Louis-Noël Pouchet, P Sadayappan, and Vivek Sarkar. Pipes: a language and compiler for task-based programming on distributed-memory clusters. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, page 39. IEEE Press, 2016.

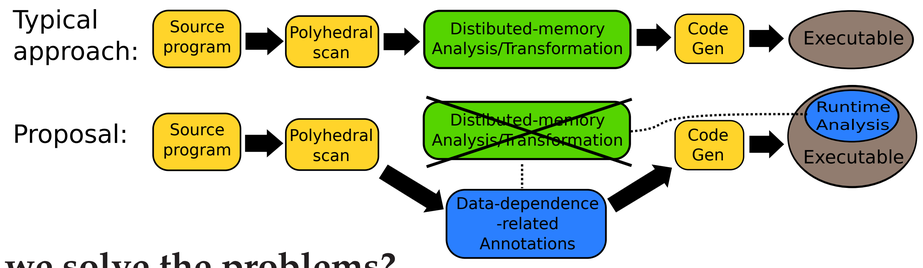
Proposal

Proposal:

We propose to move to runtime, part of the compile-time analysis needed to generate the communication code for distributed-memory systems, in order to better exploit the capacilities of the execution platforms.

Problems:

- Could we efficiently **represent** and manage polyhedra at runtime?
- Could we efficiently perform part of the compile-time **analysis** needed to generate communication code for distribute-memory systems at runtime?



How do we solve the problems?

- **Representation:** Using a distributed-memory-specialized Hierarchical Tiling Array library, named Hitmap [1], to manage hyperrectangular shapes at runtime. It provides:
  1. Domain index set operations such us intersections, or unions.
  2. Functionalities to determine the data mapped to any process at run-time, with no control data communication.
  3. Functionalities to store and reuse data communication patterns in an object.
- **Analysis:** Our approach calculates communication patterns by intersecting at runtime the index space read or written by a remote process, with the index space written or read by the local process. Communication patterns are added to an object.

Benefits:

- **Exact communications**:
  1. No control data exchange is needed among processes. At runtime, every process can determine the data space assigned, read, or written by the others.
  2. No redundant data communications. Replicated instances on the same index space can be purged when they are added to the communication object.
- **Coarse-grained communications**:
  1. Only one communication operation between each pair of processes is done.
  2. Choices such as the tile size, or the data partition method, can be decided at runtime. The communications patterns are adapted at construction.

Results

The time spent by our runtime calculation is **several orders of magnitude smaller** than the computation and the communication execution times for the tested cases. The technique currently supports affine and periodic expressions.

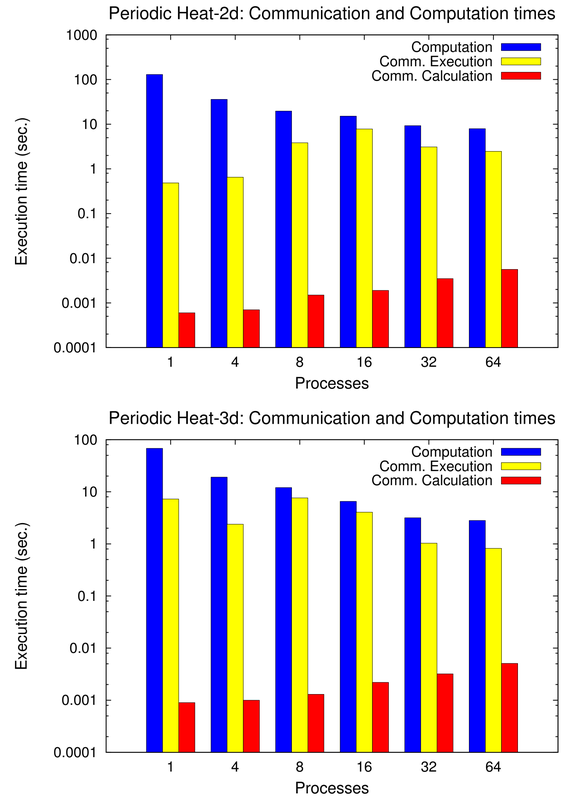


Figure 1: Execution times for the periodic Heat-2d (Size=8000x8000, iter=500), and Heat-3d examples (Size=500x500x500, iter=100).

Conclusion

We present a compiler/run-time approach to calculate communications for distributed-memory systems, that is able to adapt the program to the execution platform at runtime, delivering good performance.

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). CETA-CIEMAT belongs to CIEMAT and the Government of Spain.