

# Dealing with parallelism at run-time

Arturo Gonzalez-Escribano

Elche, 30 Enero 2017



**Grupo Trasgo**  
Universidad de Valladolid



Departamento de  
**Informática**  
Universidad de Valladolid



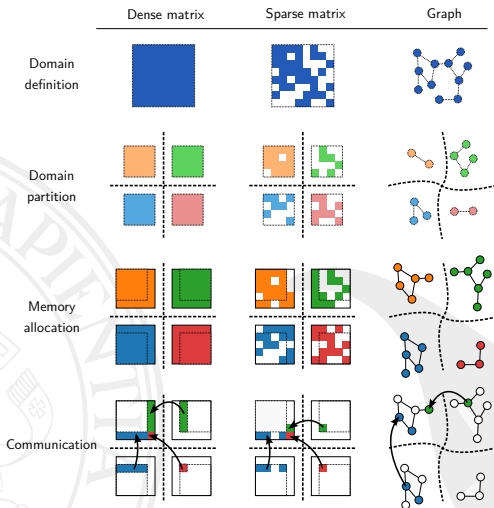
---

**Universidad de Valladolid**

# The run-time layer

- ▶ Parallel compilers and programming systems
  - ▶ Coordination, synchronization, communication
  - ▶ Mechanisms abstracted in a portable run-time layer
- ▶ Augmented run-time layer
  - ▶ More sophisticated technology
  - ▶ Extract or manage parallelism
- ▶ Advantage: Use run-time information
  - ▶ Dynamic parameters
  - ▶ Program state and behaviour
  - ▶ Execution platform details

# Distributed applications using indexed structures



# Run-Time Data Distribution (RTDD)

- ▶ Heterogeneous platforms

- ▶ Distributed clusters with nodes of different kinds
- ▶ Nodes with more than one multicore-CPU type
- ▶ Accelerators with different computation capabilities

- ▶ Indexed data structures

- ▶ Dense arrays: Limits per dimension, strides
- ▶ Graphs, sparse matrices: Sparse index structure

- ▶ Data distribution policies at run-time

- ▶ Partition and mapping of index domains
- ▶ Interchangeable distribution policies
- ▶ Load balance using information about the processing elements

# Trasgo framework

- ▶ Parallel programming framework

- ▶ High-level input parallel language
- ▶ Code transformation system, generates C code

<http://trasgo.infor.uva.es/>

- + ▶ Hitmap: Run-time layer

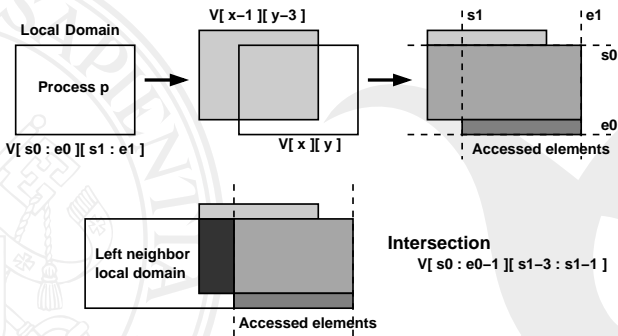
- ▶ Indexed domains and data management
- ▶ Map-Layouts: Partition and distribution modules
- ▶ Communication functions: On top of MPI

- ▶ Example of distributed array

```
Map distrib = Map( [n][m], heterogenBlocks, squareTopo );  
double M[][], aux[][];  
ArrayMap( M, distrib );  
ArrayMap( aux, distrib );
```

# Communication calculations at run-time

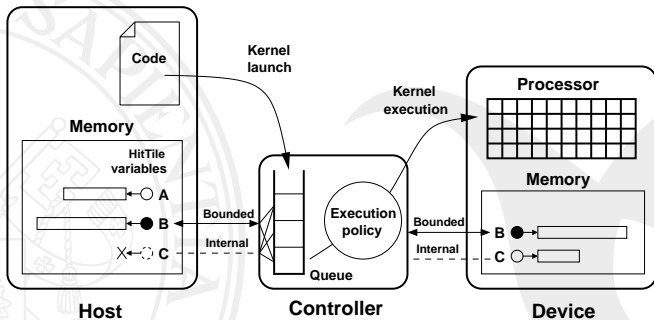
- ▶ SPMD code with affine access expressions on arrays
  - ▶ Generate code to apply access expressions to index domains
  - ▶ Query distribution objects for local/remote domains
  - ▶ Calculate intersection of local and remote domains
  - ▶ Build reusable comm. objects: Method to marshall/execute
  - ▶ Exact and coarse-grained comm: One calculation per process



# Distributing on accelerators

## ▶ Controllers model

- ▶ Transparent distribution with accelerators: GPUs to CPU-cores
- ▶ Generic, or highly specialized kernel definitions
- ▶ Automatic data mapping and communication
- ▶ Internally use native programming systems: Efficiency



# hlpp 2017

Valladolid, 10-11 Julio, 2017

Deadline: 2 Abril



## The 10th Int. Symp. on High Level Parallel Programming

### + ▶ Aims and Scope

- ▶ High-level parallel programming
- ▶ Concepts, tools, and applications
- ▶ Software quality, programming productivity

### ▶ Topics

- ▶ Programming and performance models
- ▶ Declarative programming: Functional, logical, data-flow, ...
- ▶ Algorithm skeletons and patterns
- ▶ Efficient code generation, auto-tuning, and optimization
- ▶ Model-driven software engineering for parallel systems
- ▶ Heterogeneous/hierarchical platforms with accelerators
- ▶ Teaching experience with high-level tools and methods