Interference-Aware Scheduling with 2D-Torus as a Case Study

Raphaël BLEUSE, Giorgio LUCARELLI, Grégory MOUNIÉ, Denis TRYSTRAM

To meet up the challenge of an increasing computational demand with a limited energy budget, the architecture of high performance computing platforms grows in complexity. This complexity mainly arises from the scale of the machines, the heterogeneity of the resources, and the structure of the interconnection network. The architectural evolutions of the network pose a big challenge since the network is shared by both internal and external communications of the jobs. Sharing a multi-purpose network begets complex interactions, and it has a strong impact on performances. More precisely, there are two main types of interleaved communication flows: the flows induced by data exchanges for computations (i.e., within jobs), and the flows related to I/O (i.e., jobs to external storage).

We propose here a new direction for limiting such complex interactions by adding geometric constraints to the scheduling problem.

We model a platform with two sets: nodes dedicated to computations, and nodes that are entry points to a high performance file system. The nodes are identified by a fixed numbering. These nodes communicate via a network with a given *topology*. The localization of every node within the topology is known.

A set of jobs has to be scheduled on the platform. Each job requires a fixed number of computing nodes, some I/O nodes (either a number of nodes or a subset), a certain time to be processed, and it is *independent* of every other job. Once a job has been allocated to some nodes, it runs until completion. Finally, any computing node is able to process at most one job at any time.

We consider two levels of communications: ***compute* communications** (first level) are induced by data exchanges for computations. Such communications occur between two computing nodes within a job. ***I/O* communications** (second level) are induced by data exchanges between computing nodes and I/O nodes.

We introduce two constraints for the first level: *Contiguity* and *Convexity*. We define the distance between any two nodes as the minimum number of hops between these nodes. Targeting the second level, three metrics are associated to the distance: *Compacity*, *Proximity* and *Locality*. These metrics reflect how far from I/O nodes an allocation is. They implicitly take into account external communication and potential congestion.

The Generic problem is then defined as an optimization problem with the platform (nodes and topology) and jobs' description as input. The objective is to minimize compacity or proximity along with makespan or throughput. The problem is constrained by enforcing convexity, contiguity or locality.

The second part of this work is to solve several instances for the Generic problem. We present the analysis for the 1D-torus (complexity, dominancy of some properties, greedy algorithms), and we study in detail algorithms for the 2D-torus. We consider a fixed routing scheme (by dimension), and we propose an approximation algorithm with a constant ratio.

• *R.* BLEUSE, *G.* LUCARELLI, *G.* MOUNIÉ, *and D.* TRYSTRAM *are with Univ. Grenoble Alpes. Email: first.last@imag.fr*