



université PARIS-SACLAY

«PLACEMENT DE GRAPHE DE TÂCHES DE GRANDE TAILLE SUR ARCHITECTURES MASSIVEMENT MULTICŒURS» PAR KARL-EDOUARD BERGER

Discipline : MATHÉMATIQUES ET INFORMATIQUE, Laboratoire : LIST - Laboratoire d'Intégration des systèmes et des Technologies & PRISM - Laboratoire Parallélisme, Réseaux, systèmes, Modélisation

R é s u m é

Ce travail de thèse de doctorat est dédié à l'étude d'un problème de placement de tâches dans le domaine de la compilation d'applications pour des architectures massivement parallèles. Ce problème vient en réponse à certains besoins industriels tels que l'économie d'énergie, la demande de performances pour les applications de type flots de données synchrones. Ce problème de placement doit être résolu dans le respect de trois critères: les algorithmes doivent être capable de traiter des applications de tailles variables, ils doivent répondre aux contraintes de capacités des processeurs et prendre en compte la topologie des architectures cibles. Dans cette thèse, les tâches sont organisées en réseaux de communication, modélisés sous forme de graphes. Pour évaluer la qualité des solutions produites par les algorithmes, les placements obtenus sont comparés avec un placement aléatoire. Cette comparaison sert de

métrique d'évaluation des placements des différentes méthodes proposées. Afin de résoudre à ce problème, deux algorithmes de placement de réseaux de tâches de grande taille sur des architectures clusterisées de processeurs de type many-coeurs ont été développés. Ils s'appliquent dans des cas où les poids des tâches et des arêtes sont unitaires. Le premier algorithme, nommé Task-wise Placement, place les tâches une par une en se servant d'une notion d'affinité entre les tâches. Le second, intitulé Subgraph-wise Placement, rassemble les tâches en groupes puis place les groupes de tâches sur les processeurs en se servant d'une relation d'affinité entre les groupes et les tâches déjà affectées. Ces algorithmes ont été testés sur des graphes, représentants des applications, possédant des topologies de types grilles ou de réseaux de portes logiques. Les résultats des placements sont comparés avec un algorithme de placement, présent dans la littérature qui place des graphes de tailles modérée et ce à l'aide de la métrique définie précédemment. Les cas d'application des algorithmes de placement sont ensuite orientés vers des graphes dans lesquels les poids des tâches et des arêtes sont variables similairement aux valeurs qu'on peut retrouver dans des cas industriels. Une heuristique de construction progressive basée sur la théorie des jeux a été développée. Cet algorithme, nommé Regret Based Approach, place les tâches une par une. Le coût de placement de chaque tâche en fonction des autres tâches déjà placées est calculée. La phase de sélection de la tâche se base sur une notion de regret présente dans la théorie des jeux. La tâche qu'on regrettera le plus de ne pas avoir placée est déterminée et placée en priorité. Afin de vérifier la robustesse de l'algorithme, différents types de graphes de tâches (grilles, logic gate networks, series-parallèles, aléatoires, matrices creuses) de tailles variables ont été générés. Les poids des tâches et des arêtes ont été générés aléatoirement en utilisant une loi bimodale paramétrée de manière à obtenir des valeurs similaires à celles des applications industrielles. Les résultats de l'algorithme ont également été comparés avec l'algorithme Task-Wise Placement, qui a été spécialement adapté pour les valeurs non unitaires. Les résultats sont également évalués en utilisant la métrique de placement aléatoire.

A b s t r a c t

This Ph.D thesis is devoted to the study of the mapping problem related to massively parallel embedded architectures. This problem arises from industrial needs like energy savings, performance demands for synchronous dataflow applications. This problem has to be solved considering three criteria: heuristics should be able to deal with applications with various sizes, they must meet the constraints of capacities of processors and they have to take into account the target architecture topologies. In this thesis, tasks are organized in communication networks, modeled as graphs. In order to determine a way of evaluating the efficiency of the developed heuristics, mappings, obtained by the

heuristics, are compared to a random mapping. This comparison is used as an evaluation metric throughout this thesis. The existence of this metric is motivated by the fact that no comparative heuristics can be found in the literature at the time of writing of this thesis. In order to address this problem, two heuristics are proposed. They are able to solve a dataflow process network mapping problem, where a network of communicating tasks is placed into a set of processors with limited resource capacities, while minimizing the overall communication bandwidth between processors. They are applied on task graphs where weights of tasks and edges are unitary set. The first heuristic, denoted as Task-wise Placement, places tasks one after another using a notion of task affinities. The second algorithm, named Subgraph-wise Placement, gathers tasks in small groups then place the different groups on processors using a notion of affinities between groups and processors. These algorithms are tested on tasks graphs with grid or logic gates network topologies. Obtained results are then compared to an algorithm present in the literature. This algorithm maps task graphs with moderated size on massively parallel architectures. In addition, the random based mapping metric is used in order to evaluate results of both heuristics. Then, in a will to address problems that can be found in industrial cases, application cases are widen to tasks graphs with tasks and edges weights values similar to those that can be found in the industry. A progressive construction heuristic named Regret Based Approach, based on game theory, is proposed. This heuristic maps tasks one after another. The costs of mapping tasks according to already mapped tasks are computed. The process of task selection is based on a notion of regret, present in game theory. The task with the highest value of regret for not placing it, is pointed out and is placed in priority. In order to check the strength of the algorithm, many types of task graphs (grids, logic gates networks, series-parallel, random, sparse matrices) with various size are generated. Tasks and edges weights are randomly chosen using a bimodal law parameterized in order to have similar values than industrial applications. Obtained results are compared to the Task Wise placement, especially adapted for non-unitary values. Moreover, results are evaluated using the metric defined above.