Mapping Parallel Task Graphs on cluster platforms

A. Kaci[1,2], A. Nakib[1], HN. Nguyen[2], and P. Siarry[1]

[1] Université de Paris-Est Créteil, Laboratoire Images, Signaux et Systèmes Intelligents (LiSSi, E. A. 3956), 122 rue Paul Armangot, 94400 Vitry-sur-Seine, France
nakib@univ-paris12.fr
siarry@univ-paris12.fr
[2] Bull SAS, Les Clayes-sous-Bois, France
ania.kaci@bull.net
huy-nam.nguyen@bull.net

Abstract

Many scientific applications can be structured as Directed Acyclic Graphs (DAGs). Some applications exhibit both data and task parallelism, and arise in many domains. Scheduling these applications on heterogeneous parallel platforms is a challenging task due to the large number of mapping possibilities.
Task scheduling has been largely studied in the literature and is generally a NP-hard problem [5]. For mixed parallel applications, scheduling on cluster platforms consists in allocating a cluster and a subset of its resources to tasks that optimize a considered criterion. In the case of a single homogeneous cluster, the most common objective is to minimize the schedule length which is directly responsible for the application completion time (makespan).
In the literature, scheduling algorithms can be classified as schedulers for independent tasks and schedulers for dependent tasks. Herein, we are interested on the scheduling of dependent tasks. Moreover, each task depends on data computed by other tasks. In this type of scheduling, we take into account the communication costs of transferred data.
The computational platforms considered in our work constitute a multi-cluster environment where heterogeneous clusters are fully connected by a high-speed network. Each cluster is composed of a set of homogeneous processors.
Many programming languages representing workflow of scientific applications can be translated into a DAG G={V,E} where V=$\{v_k/k \in \{1,...N\}\}$ is a set of vertices representing N tasks and E=$\{e_{ij}/(i,j) \in \{1,...,N\} \times \{1,...,N\}\}$ is a set of edges between vertices representing precedence constraints between tasks.
Each task allocated to a cluster is executed with different execution times since this execution time varies with the number of processor cores. It is restricted to run within a single cluster because of the inter-cluster communication cost. To model the parallel execution time of task i on c_{ij} processors of cluster j, we use Amdahl's law [1].
Due to the complexity of the problem, most efforts are concentrated on heuristics and metaheuristics. One of the most commonly scheduling technique found in the literature is DAG clustering [6]. The first phase consists in grouping tasks (clusters). Therefore, the size of the problem is reduced. Each group is allocated to the same cluster aiming at reducing communication cost. In the second phase, each group is allocated to a cluster according to the objective function.

In our work, we propose a two-phase DAG clustering and a hierarchical task mapping using a hybrid method based on a branch and bound algorithm.
The DAG clustering is performed in two phases. The first one is an iterative strategy which consists in identifing critical paths of the DAG. Each time a critical path is identified, it is deleted from the DAG and forms a new group. This procedure is executed until the graph is empty.
To identify critical paths, computation cost on the speedest cluster is used for the computation of earliest and latest start times of each task by traversing the DAG in a breadth-first manner. Communication times are ignored. At the end of the first phase, all the tasks within a group are sequential.
To take advantage of the multiple homogeneous processors within a cluster, a second clustering is performed. This phase consists in merging the groups found in the first clustering phase. It takes into account the data flow between groups of tasks and the target platform. Two groups are merged

if they can share an available cluster and there are data dependencies between them. Following the order of group construction, a maximum number of processors is computed for tasks belonging to the same level of two different groups, those two groups are merged if possible.

An example of the DAG clustering is shown in Fig. 1. The blue rectangles illustrate the groups constructed in the first phase. The dashed one illustrates a merging phase. To facilitate the graph analysis, two dummy tasks 1 and 11 are added. These two tasks are a predecessor and a successor (respectively) of all other tasks in the graph.

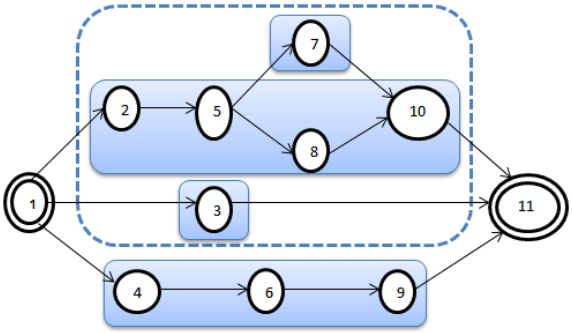

Fig. 1. A two phase DAG clustering example

The mapping phase is performed on the reduced graph. A formulation of the problem on the reduced graph is proposed. The objective is to optimize the completion time. For the mapping phase, a branch and bound algorithm [2] is performed inside each group of tasks hierarchically.
Due to the infaisability of the best mapping considered in the computation of critical paths, the branch and bound algorithm gives new values of estimated start times based on the clustering found earlier. A new step of the algorithm is executed until the maximum time is reached.
For simulation, we consider synthetic task graphs created by Daggen [3]. This proposed mapping is compared with FCFS and M-HEFT algorithms [4].

References

1. Amdahl, Gene M. "Validity of the single processor approach to achieving large scale computing capabilities." Proceedings of the April 18-20, 1967, Spring joint computer conference. ACM, 1967.

2. Brown, A. P. G., and Z. A. Lomnicki. "Some applications of the" branch-and-bound" algorithm to the machine scheduling problem." OR (1966): 173-186.

3. Casanova, Henri, Frédéric Desprez, and Frédéric Suter. "On cluster resource allocation for multiple parallel task graphs." Journal of Parallel and Distributed Computing 70.12 (2010): 1193-1203.

4. N'takpé, Tchimou, Frédéric Suter, and Henri Casanova. "A comparison of scheduling approaches for mixed-parallel applications on heterogeneous platforms." Parallel and Distributed Computing, 2007. ISPDC'07. Sixth International Symposium on. IEEE, 2007.
5. Garey, Michael R., and David S. Johnson. Computers and intractability. Vol. 174. San Francisco: Freeman, 1979.

6. Gerasoulis, Apostolos, and Tao Yang. "A comparison of clustering heuristics for scheduling directed acyclic graphs on multiprocessors." Journal of Parallel and Distributed Computing 16.4 (1992): 276-291.