Proposal Title :

Dynamic and Concurrent Jobs scheduling on Heterogeneous Systems

This proposal addresses the problem of scheduling dynamically multiple user jobs on heterogeneous systems. The resources have different capabilities and jobs have different requirements. A job is represented by a graph of tasks (DAG); jobs arrive randomly, have priorities and multiple soft deadlines. To maximize the performance of the system, it is essential to assign the resources to tasks (match) and order the execution of tasks on each resource (schedule).
The common approach is to set a fixed number of processors for each job, but it was demonstrated that a job may not use efficiently all processors due to inter-task constrains. The aim is not only to optimize the individual makespan, but also to achieve fairness, defined on the basis of the slowdown that each job would experience as a result of competing for resources with other jobs. The scheduler performance will be measured by a QoS metric that will be defined.

Heterogeneous computing (HC) is the coordinated use of various resources with different capabilities to satisfy the requirements of varying task mixtures. The heterogeneity of the resources and tasks in an HC system is exploited to maximize the performance or the cost-effectiveness of the system [1,2].
In this proposal, a job is composed by a set of related tasks and represented in a Direct Acyclic Graph (DAG). An important research problem is how to assign resources to tasks (match) and order the execution of tasks on the resources (schedule) to maximize some performance criterion of an HC system. Two different types of mapping are static and dynamic. Static mapping is performed when the applications are mapped in an off-line planning phase [3] e.g., planning the schedule for a set of production jobs. Dynamic mapping is performed when the applications are mapped in an on-line fashion [4], e.g., when tasks arrive at unknown intervals and are mapped as they arrive. In both cases, the mapping problem has been shown to be NP-complete [5,6,7]. Thus, the development of heuristic techniques to find near-optimal solutions for the mapping problem is an active area of research [8-14].
For the problem of scheduling several DAGs on a heterogeneous cluster, multiple job scheduling, [16] presents a static approach where the DAGs are known at the beginning of the scheduling. The authors compare four variants of solutions that incorporate all DAGs into a super DAG to the fairness policy that considers the DAGs individually. The aim was to optimize the overall makespan and to achieve fairness among DAGs. Fairness is achieved if the slowdowns experienced by each DAG due to resource sharing are equal. Fairness is a QoS measurement that results in higher-efficiency runs and has shown similar results in terms of overall makespan.
In [15] the former concept was extended to schedule multiple parallel task DAGs. The aim was again to minimize the overall makespan using constrained resource allocations to obtain fairness among DAGs. The aim of the fairness measurement is to allow shorter jobs and longer jobs equal access to resources. The DAGs are all known beforehand and are considered individually. Three policies based on three DAG metrics were proposed to obtain fairness.
In [17] the dynamic scheduling of multiple DAGs was introduced. DAGs arrive at different instants in time, which results in a more complex problem. The aim is to minimize the overall makespan without considering fairness because the jobs are related in the sense that only the collective result is meaningful.
In this research I want to extend the former work [17] in order to introduce, to the dynamic scheduling problem, QoS measures that reflect the quality of scheduling of each job individually, instead of the global makespan that is a collective measure.
Also, the common approach on scheduling multiple jobs is to assign a fixed set of processors for each job [15, 16]. I intend to explore the new approach presented in [17,18], that allows processors to participate in different jobs to reduced idle time and increase efficiency.

The first phase of the work consists in study single DAG schedulers in order to select the ones that give more efficient schedules. In [19] I have already evaluate the performance of list scheduling heuristics related to makespan, through comparison to metaheuristic algorithms, such as Ant Colony, Simulating Annealing and Tabu Search.
The work will start from the static scheduling of single DAGs in heterogeneous systems. In this context a widely study has been done and a new algorithm is being developed. For randomly generated DAGs the makespan is substantially reduced when compared to the reference algorithms of this field. Further research is required to obtain results for typical problems, e.g. Gaussian ilimination, FFT, etc, and to study the efficiency of the algorithm.

The target system is composed by a set of heterogeneous processing nodes and possibly heterogeneous connection network, i.e. different latency and bandwidth among network links.
With the aim of increasing the system performance, either homogeneous or heterogeneous, the aim is to study how a DAG scheduler can be extended to consider more than one job to schedule. In [18] we have already proposed an algorithm to statically schedule several jobs, but the optimization targets the global makespan, which is not the target when the jobs are unrelated.

To overcome this, the next step consists in the definition of QoS metrics, that will evaluate the schedules produced by the algorithms so that individual job metrics are evaluated. Other related metrics can be formulated to guide the scheduling algorithm. For example, based on deadline accomplishment, priority consideration, etc. An example of a set of independent task scheduling heuristics in order to maximize a Quality of Service (QoS) measurement was proposed in [20], which is based on the value accrued of completed tasks in a given interval of time. The value of a task is calculated based on the priority of the task and the completion time of the task with respect to its deadlines.

[1] M.M. Eshaghian (Ed.), Heterogeneous Computing, Artech House,Norwood, MA, 1996.
[2] R.F. Freund, H.J. Siegel, Heterogeneous processing, IEEE Comput. 26 (6) (1993) 13–17.
[3] T.D. Braun, H.J. Siegel, N. Beck, L. Boloni, R.F. Freund, D. Hensgen, M. Maheswaran, A.I. Reuther, J.P. Robertson, M.D. Theys, B. Yao, A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems, J. Parallel Distrib. Comput. 61 (6) (2001) 810–837.
[4] M. Maheswaran, S. Ali, H.J. Siegel, D. Hensgen, R.F. Freund, Dynamic mapping of a class of independent tasks onto heterogeneous computing systems, J. Parallel Distrib. Comput. 59 (2) (1999) 107–121.
[5] E.G. Coffman Jr. (Ed.), Computer and Job-Shop Scheduling Theory, Wiley, New York, NY, 1976.
[6] D. Fernandez-Baca, Allocating modules to processors in a distributed system, IEEE Trans. Software Eng. SE-15 (11) (1989) 1427–1436.
[7] O.H. Ibarra, C.E. Kim, Heuristic algorithms for scheduling independent tasks on non-identical processors, J. ACM 24 (2) (1977) 280–289.
[8] Kwok, Y.-K., Ahmad, I.: Static scheduling algorithms for allocating directed task graphs to multiprocessors. ACM Computing Surveys 31 (1999) 406-471
[9] Kwok, Y.-K., Ahmad, I.: On multiprocessor task scheduling using efficient state space search approaches. Journal of Parallel and Distributed Computing 65 (2005) 1515-1532
[10] Gyung-Leen, P.: Performance evaluation of a list scheduling algorithm in distributed memory multiprocessor systems. Future Generation Computer Systems 20 (2004) 249-256
[11] Topcuoglu, H., Hariri, S., Min-You, W.: Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Transactions on Parallel and Distributed Systems 13 (2002) 260-274
[12] Sinnen, O., Sousa, L.: List scheduling: Extension for contention awareness and evaluation of node priorities for heterogeneous cluster architectures. Parallel Computing 30 (2004) 81-101.
[13] Daoud M., Kharma N.: A high performance algorithm for static scheduling in heterogeneous distributed computing systems. Journal of Parallel and Distributed Computing, 68 (2008) 399-409.
[14] Barbosa, J., Morais, C., Nobrega, R., Monteiro, A.P.: Static scheduling of dependent parallel tasks on heterogeneous clusters. Burlington, IEEE CS Press, MA (2005)
[15] T. N’Takp´e and F. Suter. Concurrent scheduling of parallel task graphs on multi-clusters using constrained resource allocations. In International Symposium on Parallel & Distributed Processing, pages 1–8. IEEE Computer Society, 2009.
[16] H. Zhao and R. Sakellariou. Scheduling multiple dags onto heterogeneous systems. In International Symposium on Parallel & Distributed Processing, pages 1–14. IEEE Computer Society, 2006.
[17] Barbosa J., Moreira B., Dynamic scheduling of a batch of parallel task jobs on heterogeneous clusters, Parallel Computing (2011), doi:10.1016/j.parco.2010.12.004
[18] Barbosa J., Monteiro A.: A list scheduling algorithm for scheduling multi-user jobs on clusters, in High Performance Computing for Computational Science - VECPAR 2008, LNCS 5336, pp.123-136, 2008
[19] Arabnejad H, Barbosa J.: Performance Evaluation of List Based Scheduling on Heterogeneous Systems, Heteropar’11, page 440-449, june 2011.
[20] J.-K. Kima, S. Shivleb, H.J. Siegel, A.A. Maciejewski, T.D. Braun, M. Schneider, S. Tidemanc, R. Chittac, R.B. Dilmaghani, R. Joshi, A. Kaul, A. Sharmab, S. Sripada, P. Vangari, and S.S. Yellampalli. Dynamically mapping tasks with priorities and multiple deadlines in a heterogeneous environment. Journal of Parallel and Distributed Computing, 67:154–169, 2007.