GATS 1.0: A Novel GA-based Scheduling Algorithm for Task Scheduling on Heterogeneous Processor Nets

Mohammad Daoud
Concordia University
1455 de Maisonneuve Blvd. W.
Montreal, QC, Canada H3G 1M8
(+1) 514-848-2424 ext. 3117
m_daoud@ece.concordia.ca

Nawwaf Kharma
Concordia University
1455 de Maisonneuve Blvd. W.
Montreal, QC, Canada H3G 1M8
(+1) 514-848-2424 ext. 3117
kharma@ece.concordia.ca

ABSTRACT
We present a novel GA-based scheduling algorithm for heterogeneous processor networks that succeeds in generating task schedules with completion times that are 7% and 10.1% shorter than those produced by two of the best existing scheduling algorithms for heterogeneous networks of processors: HEFT [3] and DLS [2]. The new algorithm (GATS 1.0) achieves these results by employing an innovative genotype to phenotype encoding scheme and matching crossover and mutation operators. In addition, GATS 1.0 uses a simple fitness evaluation function and a small population, which makes it efficient (relative to classic GA implementations), as well as effective.

Categories and Subject Descriptors
C.1.2 **[Processor Architectures]**: Multiple Data Stream Architectures (Multiprocessors) – *multiple-instruction-stream, multiple-data-stream processors (MIMD)*; D.1.3 **[Programming Techniques]**: Concurrent Programming – *distributed programming, parallel programming*; D.4.1 **[Operating Systems]**: Process Management – *scheduling*; F.1.2 **[Computation by Abstract Devices]**: Modes of Computation – *parallelism and concurrency.*

General Terms
Algorithms, Performance, Design.

Keywords
Genetic algorithm, parallel processing, scheduling, heterogeneous systems.

1. INTRODUCTION
The problem of static task scheduling is defined as the process of allocating the tasks of an application to a network of processors, and arranging the execution of these tasks to minimize the completion time of the application. The objective of our research effort is to develop a mechanism that carries out static task scheduling and generates schedules with shorter completion time than those schedules produced by the best existing static scheduling algorithms.

GECCO'05, June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

The majority of published static scheduling algorithms are only suited to homogeneous processor networks. Little effort has been put into developing scheduling algorithms specifically for heterogeneous processor networks. It is easy to prove, using counterexamples, that the best existing heterogeneous scheduling algorithms [2, 3] generate sub-optimal schedules. Hence, there is much room for the development of better scheduling algorithms for heterogeneous processor networks.

2. PROBLEM DESCRIPTION
Static task scheduling for a Distributed Heterogeneous Computing System (DHECS) is the problem of assigning the tasks of a distributed application to a set of diversely capable machines, and specifying the start execution time of the tasks assigned to each processor. This must be done in a way that respects the precedence constrains among tasks. All the information needed to perform scheduling is assumed to be known in advance.

3. PROPOSED ALGORITHM
In this section, the Genetic Algorithm for Task Scheduling (GATS 1.0) is introduced. In GATS 1.0, an innovative genotype to phenotype encoding scheme is employed. The introduced encoding scheme ensures that the information encoded within the chromosome will always lead to a feasible schedule. Hence, this representation provides for a dense search space, and an efficient search process.

The first step in GATS 1.0 is the creation of the initial population. GATS 1.0 runs a list-based scheduling heuristic to create the first chromosome in the initial population. The schedule generated by the heuristic scheduling algorithm is encoded, and the resulting chromosome is inserted into the initial population. The schedule that is produced by the heuristic scheduling algorithm is located at an approximate area in the search space around the optimal schedule. GATS 1.0 searches that approximate area to improve the schedule.

In addition to the heuristic-generated chromosome, GATS 1.0 creates a set of processor-based chromosomes. For each individual processor in the system, exactly one processor-based chromosome is created and inserted into the initial population, such that all tasks in the encoded schedule are randomly allocated to that processor. The rest of the chromosomes in the initial population are created randomly.

Two novel genetic operators that are specially designed for the task scheduling problem are introduced: swap crossover and swap mutation. These two operators search the search space efficiently. Hence, the time required by the GATS algorithm to converge to an optimal or near-optimal schedule is minimized.

For a detailed description of the methodology of GATS 1.0, refer to [1].

4. RESULTS AND ANALYSIS

In this section, we present a performance comparison of GATS 1.0 and two algorithms form the literature. For this purpose, the GATS 1.0 algorithm, the HEFT algorithm and the DLS algorithm are simulated. A test set of 400 randomly generated task graphs and four different DHECSs are generated and used as the workload for evaluating the algorithms.

One of the performance metrics chosen for the comparison is the Normalized Schedule Length (NSL). For DHECS, the NSL of a given schedule is defined as the normalized schedule length to the lower bound of the schedule length. The NSL of a schedule is calculated using equation 1.

$$NSL = \frac{\text{Schedule Length}}{\sum_{t_i \in CP_{lower}} c_{i,a}} \qquad (1)$$

In equation 1, $c_{i,a}$ is defined as the computation cost of task t_i on processor p_a. CP_{lower}, is the critical path of the unscheduled Directed Acyclic Graph (DAG) of the application, based on the computation cost of tasks on the fastest processor p_a. The dominator of equation 1 is the summation of the computation costs of the tasks located on CP_{lower}, when they are executed on p_a. The average NSL values over several application DAGs, are used to study the performance of the algorithms.

The NSLs produced by the GATS 1.0 algorithm, the HEFT algorithm and the DLS algorithm for each DAG size are shown in table 1 and figure 1. The average NSL value of the GATS 1.0 algorithm, on all generated application DAGs, is shorter than that of the HEFT and DLS algorithms by 7% and 10.1% respectively. At higher application sizes, the performance of the GATS 1.0 algorithm degrades. At higher numbers of nodes, the size of the search space becomes larger and the GATS 1.0 algorithm requires more generations to find better schedules.

Table 1. Comparison of average NSL with respect to DAG size

DAG size	DLS	HEFT	GATS 1.0
20	3.08	3.01	2.50
40	3.61	3.49	3.20
60	4.01	3.86	3.64
80	4.22	4.09	3.93
100	4.45	4.29	4.15

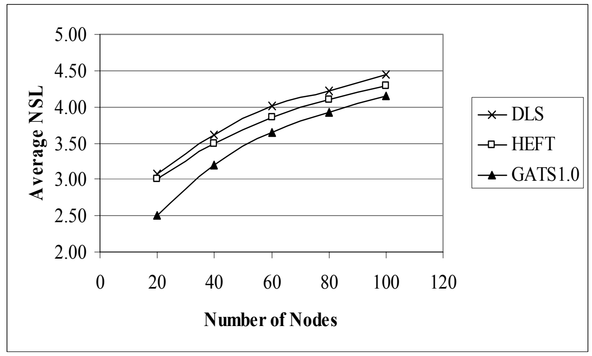

Figure 1. Average NSL with respect to DAG size

5. CONCLUSION

In this paper, a novel genetic algorithm for task scheduling called the Genetic Algorithm for Task Scheduling (GATS 1.0) is introduced. The GATS 1.0 algorithm efficiently optimizes task scheduling in DHECSs. First, the GATS 1.0 algorithm runs a list-based scheduling heuristic to generate a near-optimal schedule. The GATS 1.0 uses the list-based generated schedule to create its first population. Next, the GATS 1.0 algorithm evolves this population to find better optimal (or near-optimal) schedules. The GATS1.0 algorithm employs a set of genetic operators that are specifically designed for the task scheduling problem. These operators lead to efficient searching of the space of possible schedules. Hence, the time required by the GATS algorithm to find an optimal or near-optimal schedule is minimized.

The performance of the GATS 1.0 algorithm is compared to two of the best list-based scheduling algorithms for DHECSs. These two techniques are called the HEFT algorithm and the DLS algorithm. The GATS 1.0 algorithm outperforms both the HEFT and DLS algorithms. In terms of schedule length, the average NSL of the GATS 1.0 algorithm is shorter than that of the HEFT and DLS algorithms by 7% and 10.1%, respectively.

For the full text of the accepted paper, refer to [1].

6. REFERENCES

[1] Daoud, M. and Kharma, N. GATS 1.0: A Novel GA-based Scheduling Algorithm for Task Scheduling on Heterogeneous Processor Nets. http://www.ece.concordia.ca/~kharma/ResearchWeb/H2GS/GATS1.0.pdf

[2] Sih, G.C. and Lee, E.A. A Compile-Time Scheduling Heuristic for Interconnection-Constrained Heterogeneous Processor Architectures. *IEEE Trans. Parallel and Distributed Systems,4,* 2 (February 1993), 175-187.

[3] Topcuoglu, H., Hariri, S.,and Wu, M.Y. Performance-Effective and Low- Complexity Task Scheduling for Heterogeneous Computing. *IEEE Trans. Parallel and Distributed Systems, 13,* 3 (March 2002), 260-274.