

# Chapter 5

## An Algorithm for Task Scheduling in Heterogeneous Distributed Systems Using Task Duplication

**Amrit Agrawal**

*Jaypee University of Information Technology, India*

**Pranay Chaudhuri**

*Jaypee University of Information Technology, India*

### ABSTRACT

*Task scheduling in heterogeneous parallel and distributed computing environment is a challenging problem. Applications identified by parallel tasks can be represented by directed-acyclic graphs (DAGs). Scheduling refers to the assignment of these parallel tasks on a set of bounded heterogeneous processors connected by high speed networks. Since task assignment is an NP-complete problem, instead of finding an exact solution, scheduling algorithms are developed based on heuristics, with the primary goal of minimizing the overall execution time of the application or schedule length. In this paper, the overall execution time (schedule length) of the tasks is reduced using task duplication on top of the Critical-Path-On-a-Processor (CPOP) algorithm.*

### 1. INTRODUCTION

A heterogeneous distributed computing system (HDCS) consists of a set of processors or nodes of varying computing power, connected by a high speed network. An efficient scheduling of the

tasks of an application on the available processors is one of the key factors for achieving high performance. There are two types of task allocation schemes: static and dynamic. In static task allocation scheme, the decision about which tasks should be allocated to which processors is made prior to the actual start of that job execution. In

DOI: 10.4018/978-1-4666-2065-0.ch005

dynamic task allocation scheme, the decision about the number of processors to be made available for a job and the task assignment to these processors is made during the job execution and the decision can vary according to the existing conditions (current loads, node failures, priorities, etc.) of the network. In recent years, HDCS has emerged as a popular platform to execute computationally intensive applications with diverse computing needs. The problem of mapping (including matching and scheduling) tasks and communications is a very important issue since an appropriate mapping can truly exploit the parallelism of the system thus achieving large speedup and high efficiency (Braun & Siegel, 1998). It deals with assigning (matching) each task to a machine and ordering (scheduling) the execution of the tasks on each machine in order to minimize some cost function. The most common cost function is the total schedule length. This policy has a very high overhead leading to a degradation of the overall system performance. The above situation can be avoided by the use of simple and constant time heuristics. Efficient application scheduling is critical for achieving high performance in heterogeneous computing systems. These heuristics are classified into a variety of categories such as list scheduling algorithms, task duplication based algorithms, clustering algorithms and guided random search methods. Task assignment is known to be an NP-complete problem. The natural approach, therefore, is to develop good heuristics that will allocate the tasks to the available processors and minimize the schedule length.

**List based scheduling:** According to their priority, list-scheduling heuristic maintains a list of all tasks of a given graph. It has two phases: task prioritizing phase for selecting the highest-priority ready task and the processor selection phase for selecting a suitable processor that minimizes a predefined cost function. Some of the examples are the Modified critical Path (MCP) (Wu & Gajski, 1990), Dynamic level Scheduling (Sih & Lee, 1993), Mapping Heuristic (MH) (Rewini

& Lewis, 1990), Insertion Scheduling Heuristic (Kruatrachue & Lewis, 1998), Earliest Time first (EFT) (Hwang, Chow & Anger, 1989), and Dynamic Critical Path (DCP) (Kwok & Ahmad, 1996) algorithms. List-scheduling heuristics are generally more practical and provide better performance results at a lower scheduling time than the other groups.

**Task Duplication based Heuristics:** The idea behind duplication-based scheduling algorithms is to schedule a task graph by mapping some of its tasks redundantly, which reduces the communication overhead (Ahmad & Kwok, 1994; Park, Shirazi & Marquis, 1997; Chung & Ranka, 1992). The only change is to selection strategies of the tasks for duplication in different duplication-based algorithms.

**Clustering based heuristics:** Another class of DAG scheduling algorithms is based on a technique called clustering (Gerasoulis & Yang, 1992; Kim & Browne, 1998; Yang & Gerasoulis, 1994). The basic idea of clustering based algorithm is to group heavily communicated tasks into the same cluster. Tasks grouped into the same cluster are assigned to the same processor in an effort to avoid communication costs.

**Guided random search algorithms:** The task scheduling problem is a search problem where the search space consists of an exponential number of possible schedules with respect to the problem size. Guided random search algorithms are a class of search algorithms based on enumerative techniques with additional information used to guide the search. They have been used extensively to solve very complex problems. A Genetic algorithm (GA) (Davis, 1991) is a type of evolution computations that is commonly used.

## **2. TASK-SCHEDULING PROBLEM**

An application program can be represented by a Directed Acyclic Graph (DAG),  $G = (V, E)$ , where  $V$  is the set of  $n$  tasks.  $E$  is the set of  $e$  directed edges.

8 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/algorithm-task-scheduling-heterogeneous-distributed/69028](http://www.igi-global.com/chapter/algorithm-task-scheduling-heterogeneous-distributed/69028)

## Related Content

---

### SONATA: Social Network Assisted Trustworthiness Assurance in Smart City Crowdsensing

Burak Kantarci, Kevin G. Carrand Connor D. Pearsall (2016). *International Journal of Distributed Systems and Technologies* (pp. 59-78).

[www.irma-international.org/article/sonata/143903](http://www.irma-international.org/article/sonata/143903)

### Rough Sets and Boolean Reasoning

Hung Son Nguyen (2008). *Rough Computing: Theories, Technologies and Applications* (pp. 38-69).

[www.irma-international.org/chapter/rough-sets-boolean-reasoning/28467](http://www.irma-international.org/chapter/rough-sets-boolean-reasoning/28467)

### A Novel Load Balancing Technique for Smart Application in a Fog Computing Environment

Mandeep Kaur and Rajni Aron (2022). *International Journal of Grid and High Performance Computing* (pp. 1-19).

[www.irma-international.org/article/a-novel-load-balancing-technique-for-smart-application-in-a-fog-computing-environment/301583](http://www.irma-international.org/article/a-novel-load-balancing-technique-for-smart-application-in-a-fog-computing-environment/301583)

### Grid-Based Nuclear Physics Applications

Frans Arickx, Jan Broeckhove, Peter Hellinckx, David Dewolf and Kurt Vanmechelen (2009). *Handbook of Research on Grid Technologies and Utility Computing: Concepts for Managing Large-Scale Applications* (pp. 195-205).

[www.irma-international.org/chapter/grid-based-nuclear-physics-applications/20521](http://www.irma-international.org/chapter/grid-based-nuclear-physics-applications/20521)

### Context Related Software Under Ubiquitous Computing

N. Raghavendra Rao (2012). *Grid and Cloud Computing: Concepts, Methodologies, Tools and Applications* (pp. 602-614).

[www.irma-international.org/chapter/context-related-software-under-ubiquitous/64505](http://www.irma-international.org/chapter/context-related-software-under-ubiquitous/64505)