A New Approach of Dynamic Load Balancing Scheduling Algorithm for Homogeneous Multiprocessor System

Mahfooz Alam, Department of Computer Science, Al - Barkaat College of Graduate Studies, Aligarh, India
Ankur Kumar Varshney, Department of Computer Science and Engineering, Institute of Technology and Management, Aligarh, India

ABSTRACT

Load balancing of parallel tasks in homogeneous as well as heterogeneous systems are major trouble for researches of both industry and academia. Load balancing were broadly classified into two categories namely static load balancing and dynamic load balancing. Nowadays, numerous of researches focus towards dynamic load balancing schemes for multiprocessor system. In this paper, the authors propose a dynamic load balancing strategy for homogeneous multiprocessor system and apply on cube based interconnection network named as Folded Crossed Cube network. The performance of folded crossed cube network gives better result in terms of diameter. Experimental results show that lesser load imbalance factor has been achieved along with execution time. By this algorithm, parallel jobs are solved with largest number of tasks. The merit of this algorithm is that when the number of tasks increases, the execution time decreases with lesser load imbalance factor.

KEYWORDS

Dynamic Scheduling, Execution Time, FCC, Ideal Load, Load Balancing, Load Imbalance Factor, Migration, Multiprocessor System

1. INTRODUCTION

In computing, scheduling is the scheme by which individual task by a few means is allocated to resources that complete the task. The task may be virtual computation elements such as threads, processes or data flows, which are in turn scheduled onto hardware resources such as processors, network links or extension cards. In other hands, scheduling is the process of allocating jobs in a particular way, mainly with the reference of sequence of their coming time or according to their computational time. An excellent scheduling strategy facilitates in growing the efficiency of a system performance and in utilization of available resources in the best possible way. A scheduler is what carries out the scheduling action. Schedulers are often executed so they stay all calculate resources busy (i.e., load balancing); allow several users to share system resources effectively, or to achieve an aim quality of service. But multiprocessor scheduler has to decide which process to run and which CPU to run it on. This additional dimension greatly complicates scheduling on multiprocessor systems. However, multiprocessor scheduling (MPS) algorithm is an NP-hard optimization problem. A scheduling algorithm (SDA) is the algorithm which states how much CPU time is assigned to Processes and Threads. The main aim of every scheduling algorithm is to complete a number of
criteria: reduce the makespan or scheduling length, execution time and maximization of speedup. SDA can be classified as; deterministic and non-deterministic scheduling algorithm. In Deterministic scheduling, the scheduler has an exact knowledge of the all jobs properties beforehand such as data are dependent or independent between the tasks, running time, etc., Deterministic scheduling problem is also known as NP-Complete (Singh et al., 2015). In Non-deterministic scheduling (NDS), the scheduler has no idea of the all jobs properties because it takes decisions during run time and scheduling is based upon on dynamic parameters which change during run time while deterministic scheduling is used on static parameters. NDS algorithm can apply on heterogeneous system (HES) or homogeneous system (HOS). Therefore, multiprocessor system (MTS) is either HES or HOM (Samad et al., 2016).

1.1. Heterogeneous System

HES can be defined as the interconnection of a number of processors, having dissimilar computational speed, and are joined with dissimilar speed links used for solving group of problems which have dissimilar computational necessities. All the processors in this type of computing system, usually incorporates specialized processing capabilities and thus tasks are allocating to them accordingly. Heterogeneity can be planned in two types, such as, Functional- level heterogeneity and Performance – level heterogeneity (Alam et al., 2015; Karimi et al., 2009). In functional – level heterogeneity systems combine general use processors, likes vector units, floating-point co-processors and input/ output processors. But all tasks may not be running on all functional units. In performance – level heterogeneity, system contain several interconnected common – use processors of varying speed. HES are commonly appropriate in the commercial and scientific research fields where, real-time security, robustness and high performance, is the main priority. A few examples of heterogeneous system are High Performance Computing (Cray XD1, SRC -6 and SRC-7) etc.

1.2. Homogeneous System

A cluster of processors joined to a high speed network for accomplishing the required task, is defined as parallel computing system. Homogeneous is a technique of parallel computing system. It comprises of group of like processors joined to a high speed network. Identical processors mean all the processors joined to the system are of same computational speed and processor complexity. To achieve maximal utilization of all the available resources tasks needs to be divided equally among all the processors present in homogeneous system. HOS is all accurately the same: cache sizes, equivalent frequencies, functions, etc. while every core in HES may have a dissimilar function, memory model, frequency, etc. So, it is easier to develop software for HOSs. In particular, the following situation must be fulfilled earlier than a system can be considered homogeneous:

- The hardware of every processor assures the storage illustration and the outputs for operations on floating point numbers (FPN) equally;
- If a FPN is conveyed between processors, the communication layer assures the correct transmittal of the floating point value;
- The equal storage illustration and the equal outputs for operations on floating point numbers is also guaranteed by the software on each processor.

In scheduling, the HOS is constant computation costs and constant communication costs (Hamidzadeh, et. al., 2013). There have been number of heuristic strategies proposed for scheduling tasks for execution on the processors of a homogeneous multiprocessor system (Casavant et al., 1988; Lilja et al., 1994) via task models that address a large range of task granularities.

This paper is classified as follows: part 2 presents of problem description of proposed algorithm and parts 3 presents of load balancing and their types, part 4 and 5 presents the previous and related
Reliability-Aware Proactive Energy Management in Hard Real-Time Systems: A Motivational Case Study

Model-Based Exploration of Societal Aging in the Netherlands

Stochastic Fatigue of a Mechanical System Using Random Transformation Technique
Seifedine Kadry (2013). Diagnostics and Prognostics of Engineering Systems: Methods and Techniques (pp. 182-188). www.igi-global.com/chapter/stochastic-fatigue-mechanical-system-using/69678?camid=4v1a