Future Generation Computer Systems 28 (2012) 745–754

Contents lists available at SciVerse ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Power-aware linear programming based scheduling for heterogeneous computer clusters

Hadil Al-Daoud [a], Issam Al-Azzoni [b], Douglas G. Down [a,*]

[a] Department of Computing and Software, McMaster University, Canada

[b] King Saud University, College of Computer and Information Sciences, Riyadh, Saudi Arabia

ARTICLE INFO

Article history:
Received 29 June 2010
Received in revised form
25 February 2011
Accepted 7 April 2011
Available online 15 April 2011

Keywords:
Power-aware scheduling
Grid computing
Linear programming
Heterogeneous computing systems

ABSTRACT

In the past few years, scheduling for computer clusters has become a hot topic. The main focus has been towards achieving better performance. It is true that this goal has been attained to a certain extent, but on the other hand, it has been at the expense of increased energy consumption and consequent economic and environmental costs. As these clusters are becoming more popular and complex, reducing energy consumption in such systems has become a necessity. Several power-aware scheduling policies have been proposed for homogeneous clusters. In this work, we propose a new policy for heterogeneous clusters. Our simulation experiments show that using our proposed policy results in significant reduction in energy consumption while performing very competitively in heterogeneous clusters.

1. Introduction

Optimizing performance in computer clusters has been a topic of interest in a number of recent research papers. A computer cluster is constructed by networking various machines with different capabilities and coordinating their use to execute a set of tasks. It is true that research has been, to a certain extent, successful in accomplishing this goal but on the other hand, energy consumption has been mostly neglected.

There is often a trade-off between performance and energy consumption. Thus, good performance can be attained but often at the expense of an undesired level of energy consumption. This is because better performance can be achieved by keeping all machines on all the time in order to handle peak load conditions and improve system responsiveness. Since peak load conditions typically happen infrequently and as a result, most of the time the cluster is underutilized, energy consumption can be reduced significantly just by taking advantage of the time during which the cluster is underutilized.

Reducing energy consumption in computer clusters has become a necessity for many reasons. First of all, for a large cluster which consumes a significant amount of energy, it is necessary to use expensive cooling equipment. Cooling equipment can consume up to 50% of the total energy consumption in some commercial servers (see [1]). Also, because of the growing cost of electricity, reducing energy consumption has become an economic necessity (see [2]). Furthermore, reducing energy consumption helps the environment since gas emissions during electricity production are reduced (see [3]).

In this paper, we attempt to develop scheduling policies which aim to reduce energy consumption in computer clusters. Computer clusters can be homogeneous or heterogeneous. In our study, we consider heterogeneous clusters. Widespread availability of low-cost, high performance computing hardware and the rapid expansion of the Internet and advances in computing networking technology have led to an increasing use of heterogeneous computing (HC) systems (see [4]). Such systems are constructed by networking various machines with different capabilities and coordinating their use to execute a set of tasks.

Scheduling for such systems is complicated due to several factors. The state of the system dynamically changes and a scheduling policy should adapt its decisions to the state of the system. Another factor contributing to the complexity of scheduling for clusters is related to the heterogeneous nature of such systems. These systems interconnect a multitude of heterogeneous machines (desktops with various resources: CPU, memory, disk, etc.) to perform computationally intensive applications that have diverse computational requirements. Performance may be significantly impacted if information on task and machine heterogeneity is not taken into account by the scheduling policy.

In our earlier work [5,6], we have developed several scheduling policies that perform competitively in heterogeneous systems.

* Corresponding author.
E-mail address: downd@mcmaster.ca (D.G. Down).

0167-739X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.future.2011.04.001

The policies use the solution to an allocation linear programming problem (LP) which maximizes the system capacity. However, machine power consumption is not considered. In this paper, we suggest a power-aware scheduling policy (the power-aware Linear Programming based Affinity Scheduling policy (LPAS)). The proposed policy also uses the solution to an allocation LP which takes into consideration machine power consumption. Our experiments show that our policy provides significant energy savings.

The policy uses the arrival and execution rates to find the maximum capacity. Also, the policy uses information on the power consumption of each machine in order to find an allocation of the machines which results in the maximum energy saving. However, there are cases where obtaining such information is not possible or there is a large degree of uncertainty. In this paper, we also suggest a power-aware policy for structured systems that only requires knowledge of the ranking of machines with respect to their power efficiencies. Structured systems are a special kind of heterogeneous systems that are common for cluster environments. These are defined in Section 6.

The organization of the paper is as follows. Section 2 gives the workload model in detail. Section 3 describes several scheduling policies. The power-aware LPAS policy is described in Section 4. In Section 5, we present the results obtained in our simulation experiments including simulation results for realistic cluster models. In Section 6, we describe a power-aware scheduling policy for structured systems. Section 7 gives an overview of related work. Section 8 concludes the paper.

A preliminary version of this work appeared in [7].

2. Workload model

In our model for a computer cluster (see Fig. 1), there is a dedicated front-end scheduler for assigning incoming tasks to the back-end machines. Let the number of machines in the system be J.

It is assumed that the tasks are classified into I classes. Tasks of class i arrive to the front-end at the rate α_i. Let α be the arrival rate vector, the ith element of α is α_i. The tasks are assumed to be independent and atomic. In the literature, parallel applications whose tasks are independent are sometimes referred to as Bag-of-Tasks applications (BoT) (as in [8]) or parameter-sweep applications (as in [9]). Such applications are becoming predominant for clusters and grids (see [10,11]).

While determining the exact task execution time on a target machine remains a challenge, there exist several techniques that can be used to estimate an expected value for the task execution time (see [12,13], for example). The policies considered in this paper exploit estimates on mean task execution times rather than exact execution times. Furthermore, in computer clusters and grids, tasks that belong to the same application are typically similar in their resource requirements. For example, some applications are CPU bound while others are I/O bound. In fact, several authors have observed the high dependence of a task's execution time on the application it belongs to and the machine it is running on. They argue for using application profile information to guide resource management (see [4]). We follow the same steps and assume that the tasks are classified into groups (or classes) with identical distributions for the execution times.

Let $\mu_{i,j}$ be the execution rate for tasks of class i at machine j, hence $1/\mu_{i,j}$ is the mean execution time for class i tasks at machine j. We allow $\mu_{i,j} = 0$, which implies machine j is physically incapable of executing class i tasks. Each task class can be executed by at least one machine. Let μ be the execution rate matrix, having (i, j) element $\mu_{i,j}$. Our workload model is similar to the workload model in [6].

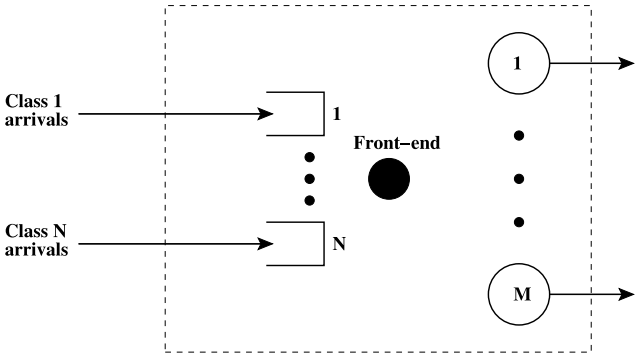

Fig. 1. The cluster system model.

We note that performance monitoring tools such as NWS [14] and MonALISA [15] can be used to provide dynamic information on the state of the cluster system. Furthermore, these tools anticipate the future performance behaviour of an application including task arrival and machine execution rates.

At this stage, we introduce the machine power consumption model. We assume that at any point in time a machine can be either busy or in a low power state. For the time being, we assume that each machine has a single operating frequency for the busy state. We will relax this when we discuss Dynamic Voltage Scaling in Section 5.3. Each machine may have different power consumption when executing different classes of tasks. Let $M_{i,j}$ be the power consumption of machine j when executing a task of class i (it is measured in terms of the energy consumed per time unit). In addition, we assume that a machine is put into a low power state when it is not executing any task. Let B_j be the power consumption of machine j when it is in a low power state. We assume that $B_j \ll M_{i,j}$. Our power consumption model is similar to the one considered in [16].

3. Current policies

A scheduling policy that is applicable to our workload model is the classical First-Come-First-Served (FCFS) policy. FCFS is used in major schedulers (such as [17,18]). An advantage of FCFS is that it does not require any dynamic information on the state of the system. However, FCFS only performs well in systems with limited task heterogeneity and under moderate system loads. As the application tasks become more heterogeneous and the load increases, performance degrades rapidly (see [5]). Furthermore, FCFS ignores machine power consumption and thus may result in severe energy wastage.

Another candidate scheduling policy is the Pick-the-Most-Efficient (PME) policy. The policy uses a greedy approach for assigning tasks to machines. It is defined as follows. When a machine j becomes available, it is assigned a class i task where the power efficiency of machine j on class i is the maximum amongst those classes with at least one task waiting. The power efficiency of a machine j on class i tasks is defined as $\mu_{i,j}/M_{i,j}$. The PME policy only requires dynamic information on the machine execution rates and power consumption. It does not take into account information on the task arrival rates.

4. The power-aware LPAS policy

The power-aware LPAS policy requires solving two allocation linear programming (LP) problems. The first LP does not take power consumption into account. It is the same LP that is used in the other LPAS-related policies (see [5,6]). This LP is solved for the purpose of obtaining the maximum capacity of the system λ^*. This value is used in the second LP.