Available online at www.sciencedirect.com
ScienceDirect
The Journal of Systems and Software 80 (2007) 1005–1014
The Journal of Systems and Software
www.elsevier.com/locate/jss

Characterizing robustness in dynamic real-time systems

Dazhang Gu *, Lonnie Welch, Frank Drews, Klaus Ecker

Center for Intelligent, Distributed, and Dependable Systems, School of Electrical Engineering and Computer Science, Ohio University, Athens, Ohio 45701, USA

Available online 1 November 2006

Abstract

The problem of robust task allocation is motivated by the need to deploy real-time systems in dynamic operational environments. Existing robust allocation approaches employ coarse robustness metrics, which can result in poor allocations. This paper proposes a metric that accurately characterizes a system's robustness within feasible allocation regions. An allocation algorithm is provided to find allocations that are both feasible and robust; the robustness as measured by the metric is shown to have theoretical bounds. Experiments demonstrate that the algorithm produces good and scalable performance compared with several heuristic algorithms.

Keywords: Real-time system; Task allocation; Robustness; Dynamic environment

1. Introduction

A current area of active research in real-time computing is resource allocation with an objective of robustness. Robust allocation reduces the necessity of reallocations, which are time-consuming both to compute and to enact. Additionally, reallocations are not appropriate for stateful real-time applications whose complex states are costly to recover. A heuristic mixed-integer programming solution was provided to maximize the allowable increase in load for a static allocation in Gertphol et al. (2002). Several heuristic algorithms were proposed to find robust allocations for periodic task strings in Shestak et al. (2005). Heuristic and approximation algorithms were developed to find robust allocations for independent, periodic real-time tasks in Juedes et al. (2004), Gu et al. (2005).

This area of research is motivated by the need to deploy real-time systems in dynamic operational environments. Such systems can take inputs from the environment, causing their execution times to depend on unpredictable environmental factors. An example of real-time applications with the property is an air defense system that has filtering, situation assessment, and missile guidance tasks. Each task's resource needs depend on environmental factors such as the number of radar tracks and the number of launched missiles.

Existing robust allocation approaches such as Gertphol et al. (2002), Ali et al. (2003), Juedes et al. (2004) and Gu et al. (2005) have the shortcoming that they employ coarse robustness metrics, which can result in poor allocations. The robustness metric of Ali et al. (2003) was based on the $l-2$ norm. It measures the radius of maximum environment perturbation environment without violating feasible boundaries. However, the metric only partially characterizes feasible regions with an inner tangent sphere, and no algorithm was developed to optimize it. The max-of-min component of a workload vector among all feasible points was adopted as a robustness metric in Gertphol et al. (2002), Juedes et al. (2004), Gu et al. (2005). But it only partially characterizes feasible regions with an inner tangent rectangle. In addition, many existing approaches do not emphasize real-time scheduling and feasibility. For instance, CPUs were assumed to be fair-shared in

* Corresponding author. Tel.: +1 740 593 2913; fax: +1 740 593 0007.
E-mail addresses: gud@ohio.edu (D. Gu), welch@ohio.edu (L. Welch), drews@ohio.edu (F. Drews), ecker@ohio.edu (K. Ecker).

0164-1212/$ - see front matter
doi:10.1016/j.jss.2006.09.028

Gertphol et al. (2002). A special scheduler based on tightness was assumed by Shestak et al. (2005), but no feasibility guarantee was made.

The shortcoming is addressed in this paper. A new model that explicitly incorporates environmental factors is presented, which characterizes task execution time as a function of the environment. A metric that accurately characterizes the robustness of feasible regions of allocations is introduced. The metric accurately captures the entire feasible region by measuring its volume. It is able to consider allocation properties ignored by the previous metrics. A robust task allocation problem is defined based on the metric. Under this metric, it is shown that an allocation algorithm with fast running time and scalability can be developed to find feasible, robust allocations. This is useful for modern distributed systems that may contain hundreds of processors and thousands of tasks. A theoretical bound for its solution quality is derived, allowing guarantees to be made about the minimum robustness that can be achieved by the algorithm. The algorithm is experimentally compared with an exhaustive-search algorithm and with three heuristic algorithms (simulated-annealing, random-search, and hill-climbing). The experiments generate numerous random problem instances with small, medium, and large numbers of processors and tasks, and the approximation algorithm is shown to have fast running time and good performance. The paper is organized as follows. Section 2 describes the system model and defines the robust task problem. Section 3 introduces the new robustness metric. Section 4 presents the robust allocation algorithm and derives its theoretical analysis. Section 5 presents the experimental results.

2. System model

The real-time computing paradigm is introduced next to capture features of the problem. It is based on periodic real-time tasks characterized by environment-dependent execution time functions, as opposed to the traditional model with hard periodic real-time tasks characterized by worst case execution times. In this paradigm, occasional deadline misses could be tolerated when unpredictable environmental factors drive a demand on resources beyond their limits. This section begins the system model with a traditional model; then necessary extensions are made to incorporate environmental factors. An example follows to illustrate the model.

2.1. Traditional model

The system model used in this work is derived from the standard real-time periodic task model in Liu (2000), where a software system consists of a set of n periodic tasks $S = \{T_1, T_2, \ldots, T_n\}$. Each T_i is released periodically with a period of p_i and has a deadline equal to its period. The execution time of each task $T_i \in S$ is a constant e_i that represents the worst case execution time of T_i. The tasks are executed on a set of m identical processors $H = \{P_1, P_2, \ldots, P_m\}$.

2.2. Model extensions

The traditional model does not capture dynamic and unpredictable environment since the execution times are modeled as constants. It is inadequate for some real-time systems operating in these environments. For instance, a general distributed control system can have filter, analysis, action planning, and actuation tasks, and one or more of the tasks may contain algorithms and execution times that are affected by unpredictable environmental factors. Systems with these properties include air defense, surveillance, and intelligent vehicles. Accurate modeling of such systems will supply the necessary information to allocate resources to them so that they are resilient to many unpredictable scenarios.

The changing environmental factors that affect a system are modeled as l environmental variables, $\vec{w} = (w_1, w_2, \ldots, w_l)$. Assume that each variable has a defined value at each point in time. Let W_i be the range of w_i ($1 \leqslant i \leqslant l$); then $w_i : \mathbb{R} \to W_i$ describes the temporal behavior of variable w_i. We assume that the range W_i is a set of discrete numbers, starting from 0 to some maximum $W_i^{\max}$ (which may be infinity). The function of w_i however, cannot be known a priori, but at any given time t, each $w_i = w_i(t)$ becomes known as a constant value.

Each task $T_i \in S$ has an execution time $e_i(\vec{w})$ that is a function of these environmental variables. The system utilization is likewise a function of the environmental variables, $U(\vec{w}) = \sum_{i=1}^{n} \frac{e_i(\vec{w})}{p_i}$, and $U(0)$ corresponds to the portion of system utilization that is independent of environmental variables. Each task T_i can be allocated to any $P_j \in H$, because processors P_j are assumed to be identical. Rate monotonic scheduling is assumed to be used on each processor. An allocation M of the system is a many-to-one mapping of the task set to the processor set, $S \to H$.

For example, an air defense system contains filtering, situation assessment, and missile guidance tasks. These tasks depend on such factors as the number of radar tracks and the number of missiles. Thus, $l = 2$ in this example, where w_1 is the number of radar tracks and w_2 is the number of missiles. The system may be further characterized as follows: T_1, the filtering task, has an execution time function $e_1(w_1,w_2) = w_1 + w_2$; T_2, the situation assessment task, has $e_2(w_1) = w_1$; T_3, the missile guidance task, has $e_3(w_2) = w_2$. Task T_1, T_2, T_2 may have the same period $p_1 = p_2 = p_3 = 2$ seconds. There are 3 processors available: P_1, P_2, P_3. In one allocation, M, of the system, task T_1 is assigned to P_1, task T_2 is assigned to P_2, and task T_3 is assigned to P_3.

3. An accurate robustness metric

A robustness metric is used to measure the amount of environmental variation that can be sustained by an