Mapping and Characterization of Applications in Heterogeneous Distributed Systems

Jaime Yeckle Sánchez
Advisor: Wilson Rivera

Computing Information Sciences and Engineering
University of Puerto Rico, Mayaguez Campus
Mayagüez, PR 00681
jyeckle@ece.uprm.edu

Abstract

The problem of mapping tasks and communications onto multiple machines and networks in a heterogeneous computing environment has been shown to be NP complete. Therefore, the development of heuristic techniques to find near-optimal solutions is required. Many different types of mapping heuristics have been developed in recent years. However, selecting the best heuristic to use in any given scenario remains a difficult problem. Moreover, it is not possible to make a general mapping in a heterogeneous computing environment. In this paper we propose to characterize classes of applications with the objective of predicting their behavior. Using the insight provided by the characterization, we achieve a more realistic mapping for specific applications

1. Introduction

The steady decrease in cost and increase in performance of commodity workstations and personal computer have made it increasingly attractive to use clusters of such systems as compute servers instead of high-end parallel supercomputers. Due to the rapid advance in performance of commodity computers, when such clusters are upgraded by addition of nodes, they become heterogeneous. The issue of effective mapping of applications onto such heterogeneous clustered systems is therefore of great interest. Several recent research studies have addressed this problem. However the problem is NP complete [Coffman76], therefore, the development of heuristics techniques is required. Many factors make it difficult to select the best technique of mapping. These factors can be described as following:

1. A distributed computing environment has conflicting requirements: (i) While minimizing interprocessor communication tends to assign the entire computation to a single processor, load balancing tries to distribute the computation evenly among the processors. (ii) While real-time constraint uses has many processors as possible to maximize parallel execution, the precedence relationships limit parallel execution. (iii)The saturation effect suggests the use of fewer processors since inefficiency increases with the number of processors.

2. When one heuristic technique is presented and evaluated in the literature, typically, different assumptions are made about the underlying target platforms making comparisons problematic. Similarly, different assumptions about application models complicate comparisons. The algorithms should take into account characteristics of processors, network architecture and applications. Regarding applications, for example, the existing algorithms only consider the size property.

In this paper, we propose characterize classes of applications with the objective of predicting their behavior. In order to obtain this characterization of applications, we consider observations and statistical methods. Using the insight provided by the characterization, we propose a more efficient and realistic mapping for specific applications.

The rest of the paper is organized as follows. Section 2 begins with the description of different mapping schemes and taxonomy for describing matching and scheduling heuristics for heterogeneous computing systems. Section 3 presents the methodology for characterizing applications and the use for heterogeneous distributed systems. The paper concludes with comments about the proposed strategy and its extensions.

2. Taxonomy of Mapping

Mapping includes assigning (matching) each task to a machine and ordering (scheduling) the execution of the tasks on each machine. The mapping problem is extremely difficult to solve and generally intractable. Even the simplified subproblems constructed from the original mapping problem by imposing a variety of constraints still fall in the class of NP hard problems. The difficulty of solution varies with the inclusion or exclusion of preemption, the number of parallel processors, precedence constraints, etc.
Casavan [Casavan88] classify the various strategies for multiprocessor scheduling, task mapping, and resource allocation under a commo n, uniform set of terminology. The Figure 1 shows the structure of the hierarchical portion of the taxonomy. The strategies can be classified as being either static or dynamic.

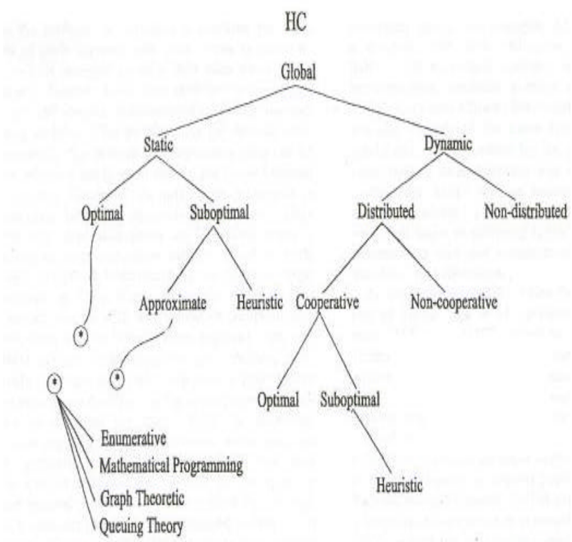

Figure 1. Taxonomy of mapping

Many algorithms have been published addressing the problem of matching and scheduling where several simplifying assumptions are common. Orduña [Orduña01], for example, describes a mapping scheme assuming all the network switches are attached to the same number of workstations, all the workstations are uniprocessors and only one process is mapped to each processor. These assumptions nevertheless determine the system's performance.
Another simplifying assumption is made in [Dixit94]. In this paper mapping is modeled with forward flow only. Programs with dynamic structures are not considered. Also, Ahmad and Kwok [Ahmad96] compared several algorithms for scheduling task graphs. The algorithms have different assumptions: bounded and unbounded number of processors and clusters, task duplication based scheduling and arbitrary processor network scheduling.

In the previous discussion, most of the approaches focus primarily on specific mapping strategies for a particular multiprocessor architectures. Some approaches attempt to take advantage of hardware characteristics, such as the interconnection network of architectures. Others take advantage of characteristics of processors and very few papers take advantage of characteristics of applications. Also exist other approaches, such as [Chaudhary98] which intent to generalize mapping and scheduling. By large, these approaches assume homogeneous conditions for many characteristics of applications or architecture of hardware. These approaches assume homogeneous conditions for many characteristics of applications or architecture of hardware. Therefore, a general mapping to all applications and characteristics of hardware and network is difficult to obtain. We propose a specific mapping considering hardware issues in the context of specific applications. To achieve this, we propose a characterization of the applications with the objective of predict behavior.

3. Characterization of Applications

In this section, we provide a new approach to mapping in heterogeneous distributed system considering the characterization of applications.

3.1 Characteristics of applications

The characteristics of the applications (which can be tasks or subtasks) can be defined as follows:

Application size: This mean if a task contains subtasks or not.
Application type: Types of applications to be mapped.
Communication patterns: This mean to the sources and destination subtasks for each data item to be transferred.
Data availability: The time at which input data needed by a subtask or output data generated by a sub task can be utilized varies in relation to subtask start and finish times: (a) is data available (to be forwarded) before a subtask completes, and (b) can a subtask begin execution before receiving all of its input data? As an example, a clustering non-uniform assumes that a subtask cannot send data to other

waiting subtasks until it completely finishes executing.

Deadlines: the applications have deadlines.

Execution time model: Most mapping techniques require an estimate of the execution time of each application on each machine. The two choices most commonly used are probabilistic and deterministic modeling. Probabilistic modeling uses a probability distribution for application execution times when make mapping decisions. Deterministic modeling uses a fixed (or expected) value, e.g., the average of ten previous executions of an application.

Task heterogeneity: For each machine exist different tasks with different properties (e.g., probability distribution) which make the execution times different.

Multiple versions: the applications have multiple versions that could be executed For example, an application that requires an FFT might be able to perform the FFT with either of two different procedures that have different precisions, execution times, and resource requirements.

Priorities: Priorities are generally assigned by the user (within some allowed range), but the relative weight given to each priority are usually determined by another party (e.g., a system administrator). Priorities and their relative weightings are required if the mapping strategy is preemptive.

Task profile: Task profiling specifies the types of computations occurring in an application based on the code for the task (or subtask) and the data to be processed. This information may be used by the mapping heuristic, in conjunction with analytical benchmarking to estimate task (or subtask) execution time.

Temporal distribution: It is mean static applications (the complete set of tasks to be mapped known a priori), dynamic applications (the tasks arrive in a real-time, non-deterministic manner), or it can be a combination of the two

3.2 Characterization

The predictions of behavior in applications are based on past observations. These observations can be obtained using simulations or in real time. The survey in real time is not advantage because real applications of interest might run for long periods of time and it is not feasible to perform a statistically significant number of experiments. In addition, using real resources makes it difficult to exp lore a wide variety of resource configurations. Finally, variations in resource load over time make it difficult to obtain repeatable results. Simulation is then the most viable approach to effectively evaluate scheduling algorithms.

Currently a few simulation packages are available, but they are not targeted to the simulation of distributed applications for the purpose of evaluating scheduling algorithms. These tools are often very complete and sophisticated but too complex and low-level for our purpose. Simgrid [Casanova01] is the tool with more success. Simgrid provides core functionalities that can be used to build simulators for the study of application scheduling in distributed environments. In this survey we consider Simgrid to perform simulations. In order to make simulations, we use a mapping algorithm which consider heterogeneity of processors and network.

3.3 Statistical method

In order to obtain the characterizations of applications using past observations we propose use statistical methods. This approach offers a number of advantages. First, a statistical method can compensate for many different factors, without requiring a distinct model for each of the different machine architectures. Second, statistical estimates will improve with time, as the number of previous observations increases. Finally, statistical schemes can be made to be computationally efficient, making them practical for use at run time. One potential criticism of statistical schemes is the need of a large number of past observations to obtain accurate estimates.

One important method to consider is nonparametric regression. Nonparametric regression has the advantage of being able to estimate equation of model, as a function of several parameters, without any knowledge of the function itself. Since we make no assumptions on the functional form, this prediction scheme does not require any knowledge of either the task of the target architecture, making it applicable in a very general sense.

Most of the previous work for heterogeneous distributed computing which mixing past observations and statistical methods are generally for execution time estimation. The SmartNET heterogeneous scheduling tool offers statistical execution time estimation technique, but no details of its implementation have yet been published [Kidd95]. Iverson [Iverson99] present nonparametric regression technique, for execution time estimation in heterogeneous distributing computing. Other authors [Hou94],[Shin94] using techniques, based on Bayesian decision theory, but these techniques are difficult to implement.

For prediction behavior of applications defined in this paper, we consider a function p(d) where d is a vector of parameters of the application such that size, etc. While the estimation algorithm does not know any details about the functional form of p(d), it does have a set of n previous observations of applications $\{(y(l)_i, d_i)\}_{i=1}^{n}$, where $y(l)_i$ is the prediction behavior of application for the vector d and ? is a function that depends of time, bandwidth, etc. These observations are assumed to contain some amount of random error e, such that

$$y(l)_i = p(d_i) + e_i$$

The goal of this problem is, given the function p(d), to obtain an estimate of the characteristics of this application, using the set of previous observations.

3.4 Integration

The scheme will operate in the following manner. A set of previous observations is maintained by the algorithm. Using this set of observations, the prediction behavior of application in each potential machine can be estimated, and then matching and scheduling algorithm can use these estimates to make a realistic and efficient mapping decision. After the task of the application is complete, the new parameters are added to the set of observations, to be used to improve future predictions. By storing past observations, the estimation algorithm is able to improve its estimates over time.

4. Conclusions and Future Work

In this paper, we have presented a review of different techniques used to perform matching and scheduling in heterogeneous distributed systems. We propose a characterization of specific applications with the objective to obtain a forecast behavior. The predictions of behavior in applications are made base on a combination of observations, simulation and modeling through statistical methods. The result can be utilized to generate a new mapping techniques more efficient and realistic. Our future work will be concerned with the development of mapping algorithms based on the new approach. We envision a set of mapping strategies and tools to reach peak performance of applications on heterogeneous distributed systems.

Acknowledgement

This work was supported by the UPRM-NSF PRECISE Project (NSF-EIA 99-77071) and the UPRM PhD program in Computing and Information Science and Engineering.

References

[Ahmad96] Ahmad, I., Kwok, K., and Wu, M., “Analysis, Evaluation, and Comparison of Algorithms for Scheduling Task Graphs on Parallel Processors,” in IEEE Transactions on Parallel and Distributed Systems (1996).

[Casanova01] Casanova, H. “Simgrid: a Tolkit for the Simulation of Application Scheduling. Computer Science Department University of California, San Diego,(Sept. 2001).

[Casavant88] Casavant, T., and Kuhl, J., “A Taxonomy of Scheduling in General Purpose Distributed Computing Systems. IEEE Transactions on Software Engineering, 14(2):42-45, (February 1988).

[Coffman76] Coffman, E., “Computer and Job-Shop Scheduling Theory,” John Wiley & Sons, New York, (NY, 1976).

[Chaudhary98] Chaudhary, V., and Aggarwal, J., “A Generalized Scheme for Mapping Parallel Algorithms. IEEE Transactions on Parallel and Distributed Systems, (1998).

[Dixit94] Dixit, V., and Panda, D.,. “Clustering and Intra-Processor Scheduling for Explicitly Parallel Programs on Distributed Memory Systems”. IEEE Transactions on Parallel and Distributed Systems, (Agust 1994).

[Hou94] Hou, C., and Shin, K.,. “Load Sharing with consideration of Future Task Arrivals in Heterogeneous Distributed Real Time Systems. IEEE Transactions on Computer,43(9);1076-90, (Sept. 1994).

[Iverson99] Iverson, M., “Statistical Prediction of Task Execution Times through Analytic Benchmarking for Scheduling in a Heterogeneous Environment. IEEE Transactions on Computer, vol 48 Nro 12, (December 1999).

[Kidd95] Kidd, T., Hensgen, D., Moore, L., “Studies in the Useful Predictability of Programs in a Distributed and Homogeneous Environment,” The Smartnet Home Page, http://papaya.nosc.mil:80/SmartNet/, 1995.

[Orduña01] Orduña, J., and Duato, F., “A New Task Mapping Technique for Communication-aware Scheduling Strategies”. IEEE Transactions on Parallel and Distributed Systems, (Set. 2001).

[Shin94] Shin, K., and Hou, C.,. “Design and Evaluation of Effective Load Sharing in Distributed Real Time systems. IEEE Transactions Parallel and Distributed Systemsr,5(7):704-19, (July. 1994)