A Model of Distributed Computer System for Load Distribution

7M－7 †Naomaru Itoi ‡Koichi Asakura ‡ Toyohide Watanabe
† Department of Information Engineering, School of Engineering, Nagoya University
‡ Department of Information Engineering, Graduate School of Engineering, Nagoya University

1 Introduction

Recently, the distributed computer system, which consists of a collection of autonomous workstations(nodes) connected by a local area communication network, is commonly used. In order to control such a system efficiently, it is necessary to apply a load distributing technique to assign the newly created process to a suitable node.

Several researches about load distribution have been reported. However, the load distributing methods proposed before have the following problems:

1. They are concerned about only CPU load, or they pay slight attention to other resources such as memory, disk, network and so on;
2. On assigning a new process, the characteristics of the process are not focused on;
3. The influence, which the newly assigned process causes to other processes on the same node, is not taken into account.

In order to solve the above problems, we compose a model of distributed computer system and propose a process assignment method based on this model. Our model takes not only the CPU resource but also the memory resource into account, and can predict the system behavior successfully.

2 Formulation of System Behavior

In order to achieve an efficient load balancing, it is important to observe and formulate the behavior of distributed computer system. In other words, the information about the processing time, the number of memory page replacement and so on, is required at many circumstances.

For retrieving these characteristics of the system, we use two typical benchmark programs: one of which consumes the CPU resource mainly and another does the memory one. Two benchmark programs are executed under an arbitrary CPU load average or an arbitrary size of free memory, and then the processing times of these programs are measured.

Figures 1, 2 and 3 represent the results of benchmark programs. These graphs express the processing times for CPU load average (in Figure 1), the size of free memory and usage memory (in Figure 2), and the size of usage memory (in Figure 3) respectively.

Figures 1 and 3 indicate that the processing time of a program increases in proportion to the CPU load average or the size of used memory. Furthermore, Figure 2 explains that the processing time increases by the square order for the size of free memory when few memory is left. From these results, we can formulate the system behavior as follows:

1. At a certain CPU load average "x", the processing time of benchmark program can be approximated by
$$t_{cpu}(x) = c_1 x + c_2. \quad (1)$$
2. At a certain size "y" of free memory, and size "z" of used memory, the processing time can be approximated by
$$t_{mem}(y, z) = t_{mem1}(y, z) + t_{mem2}(z), \quad (2)$$
where from the result of Figure 2,
$$t_{mem1}(y, z) = \begin{cases} 0 & (y - z >= w_0) \\ c_3(c_4 - (y - z))^2 + c_5 & (y - z < w_0) \end{cases} \quad (3)$$
and from the result of Figure 3,
$$t_{mem2}(z) = c_6 z. \quad (4)$$

3 Process Assignment

Here, we explain our process assignment method based on the model of distributed computer system described above. In order to detect the accurate node for executing a newly created process, it is very important to analyze the following three topics:

1. The influence for newly created process, with respect to the system state;
2. The influence for distributed computer system, with respect to the execution of a new process;
3. The influence for other processes which have been executed before, with respect to the state change of system resources.

3.1 Preliminaries

The model of distributed system is shown in Figure 4. First of all, we define some terms used in the following discussion.

- "P" , "N" and "IP" mean a newly created process, node computer, processes which have been executed before, respectively.
- For characteristics of a process, we use the real processing time "p_{real}", CPU time "p_{cpu}" and size "p_{mem}" of used memory. These parameters can be retrieved through the accounting information.

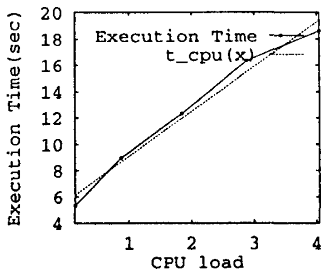

Figure 1: CPU bound

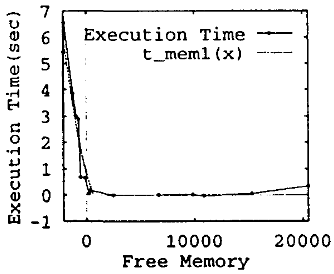

Figure 2: Memory bound (small free memory)

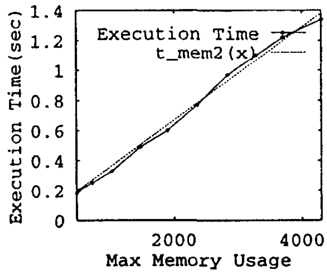

Figure 3: Memory bound (large free memory)

- The CPU time of benchmark program, which consumes CPU resource and memory resource, is expressed in "b_{cpu}" and "b_{mem}", respectively.
- The current CPU load average and size of free memory in the node are expressed in "$\text{Load}_{(old)}$" and "$\text{Mem}_{(old)}$".

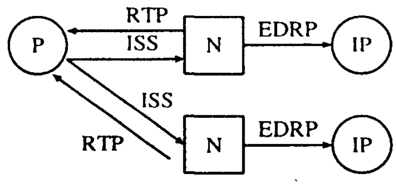

Figure 4: Model of distributed computer system

3.2 Predictions

We describe our process assignment strategy in detail.

Response Time Prediction (RTP)

First, we think about the influence from a system state of node to a newly created process. We estimate the response time of the process with the system state and characteristics of the process. Response Time Prediction (RTP) is calculated by the following equation (5). RTP is the sum of the times for CPU processing, paging, and memory access.

$$\begin{aligned}\text{RTP}(p_{cpu}, p_{mem}, b_{cpu}, \text{Load}_{(old)}, \text{Mem}_{(old)}) = \\ \frac{p_{cpu}}{b_{cpu}} t_{cpu}(\text{Load}_{(old)}) + t_{mem1}(\text{Mem}_{(old)} \\ -p_{mem}) + t_{mem2}(p_{mem})\end{aligned} \quad (5)$$

Influence to System State (ISS)

Second, we think about the influence from a newly created process to the distributed computer system. When the process is assigned to a certain node, the CPU load average "Load" of the node is increased, as

$$\text{Load}_{(new)} = \text{Load}_{(old)} + \frac{p_{cpu}}{p_{real}}. \quad (6)$$

In addition, the size "Mem" of free memory is decreased as

$$\text{Mem}_{(new)} = \text{Mem}_{(old)} - p_{mem}. \quad (7)$$

Efficiency Decrease Rate Prediction (EDRP)

Third, we think about the influence from a change of the system state to other processes which have been executed before. The rate "α" of new and old efficiencies of the processes, is calculated by the following equation:

$$\begin{aligned}\alpha &= \frac{\text{Load}_{(new)}}{\text{Load}_{(old)}} \times \frac{\text{Load}_{(old)}}{\text{Load}_{(old)} + \text{Mem}_{(old)}} \\ &\quad + \frac{\text{Mem}_{(new)}}{\text{Mem}_{(old)}} \times \frac{\text{Mem}_{(old)}}{\text{Load}_{(old)} + \text{Mem}_{(old)}} \\ &= \frac{\text{Load}_{(new)} + \text{Mem}_{(new)}}{\text{Load}_{(old)} + \text{Mem}_{(old)}}.\end{aligned} \quad (8)$$

3.3 Process Assignment Strategy

The node, which the process should be assigned to, is decided according to the three predicted values described above. We use the following equation for this decision. The execution efficiency of processes in the node is decreased by $(\alpha - 1)$. c_7 is a constant to keep balance between the newly created process and processes which have been executed.

$$\text{EVAL} = \text{RTP}(1 + c_7(\alpha - 1)) \quad (9)$$

The node whose value of EVAL is minimum in distributed computer system is selected for assignment of the newly generated process.

4 Conclusion

In this paper, we described the model of distributed computer system and the system behavior prediction method, that is required for the load distribution. We will ascertain the availability of this method by implementation, and will take account of the factors of other resources such as disk, network and so on, as our future work.