行政院國家科學委員會補助專題研究計畫成果報告

※ ※※※※※※※※※※※※※※※※※※※※※※※
※
※
※
※以隨機過程來研究網狀多處理器的處理器配置策略
※ ※
※
※ ※※※※※※※※※※※※※※※※※※※※※※※
※

計畫類別：■個別型計畫　　□整合型計畫
計畫編號：NSC　89-2218-E-039-001
執行期間：89 年　8 月 1　日至 90 年 7 月 31 日

計畫主持人：吳　帆 助理教授
共同主持人：

本成果報告包括以下應繳交之附件：

□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位：中國醫藥學院醫務管理研究所資訊組

中　華　民　國　91 **年 元 月七　日**

行政院國家科學委員會專題研究計畫成果報告
國科會專題研究計畫成果報告撰寫格式說明
Preparation of NSC Project Reports

計畫編號：NSC 89-2218-E-039-001
執行期限：89 年 8 月 1 日至 90 年 7 月 31 日

主持人：吳 帆　中國醫藥學院醫務管理研究所資訊組
計畫參與人員：林俊榮,孫漢屏中國醫藥學院醫務管理研究所資訊組

一、中文摘要

在這研究中，我們提出一個新的資料結構—R-array 來表示網狀多處理機 透過R-array的統計資料，配置程序可以很快知道是否目前所檢查的處理機可以成為空閑子網的基底。在R-array的基礎下，我們可以隨機過程來研究處理器配置。這研究可以精確找出過去從未算出的，在不同負載下，找出空閑處理器空間的機率；這方法也是第一個能精確算出其執行成本的處理器配置策略。

關鍵詞：網狀結構、馬可夫鏈結、隨機程序、處理器配置

Abstract

In this study, we propose a new data structure—the R-array to represent the mesh at first. With the statistical information of R-array, the allocation process can quickly know whether the scanned processor can serve as a base of a free submesh. Based on the R-array, the stochastic process to analyze the processor allocation scheme is developed. The probabilities, which are never computed out before, to find the free submesh under different workloads can then be accurately calculated. In addition, the proposed scheme becomes the first one whose execution costs of allocation processes can be precisely computed. This article provides guidance for report writing under the Grant of National Science Council beginning from fiscal year 1998.

Keywords: mesh, Markov chain, stochastic process, processor allocation.

二、緣由與目的

Gaining speed in solving problems by a number of processors has drawn considerable attention in recent years. Due to its regular and simple structure, the mesh-connected multiprocessor is one of the most suitable architectures for the multiprocessor system of small and medium size. The mesh has exhibited a high potential as a supercomputer but at a lower price to efficiently implement algorithms for image processing, matrix operations, partial differential equations, and so on [7]. Based on the mesh topology, many prototype and commercial systems, such as ILLIVAC IV [2], Tera Computer System [1], Dash [14], Intel Touchstone Delta [10] and Paragon [11], J-Machine [16] and T3D [12] have been marketed or built.

Recently, a lot of schemes have been proposed for processor allocations in the mesh. Li and Cheng [9] proposed the two-dimensional buddy scheme, which is a generalization of the one-dimensional buddy system used in the memory management. Chuang and Tzeng [4] proposed the Frame-Sliding (FS) scheme capable of applying to a non-square mesh. Ding and Bhuyan [5] modified the FS scheme into the Adaptive-Scan (AS) scheme. The AS scheme can fully recognize the free spaces; however, its search time is slow. Zhu proposed two full-recognition schemes: The first scheme [19] includes the first-fit and best-fit processes. Zhu recently proposed another full-recognition scheme [20]. The scheme represents the mesh by the *area tree*, a variant of quadtree [18] that is used to represent binary images in computers.

三、研究報告應含的內容

The execution cost of the allocation process generally counts on the number of accessed units. Without loss of generality, the analyses of the leapfrog scheme are considered in terms of the number of elements in the *R*-array that are accessed.

Assume that a task $T(w, h)$ comes up a mesh $M(W, H)$ whose current workload is 钋, where $0 \le$ 钋 ≤ 1, $0 \le w \le W$ and $0 \le h \le H$. For simplicity, the occupied condition among processors is assumed to be statistically independent. Let A be the event that processor p_a is occupied, and let B be the event that another processor p_b is occupied. Then $P(AB) =$ 钋×钋, and $P(A/B) =$ 钋贞

Consider the first-fit process in the leapfrog scheme. Concentrating the movement of the process, Fig. 1 is the transition diagram to describe the movement of the first-fit process in row r, where $0 \le r \le H-h$, the circles denote the possible states that the process will move to, and the directed links represent the permitted transition among these states. The states in this figure can be classified into three types, according to the meaning the state denotes:

Type I: This type of states contains only one state (i.e., state F) in it. State F (or *Found* state) denotes the situation that the first-fit process finds the currently scanned processor is a free base;

Type II: This type of states also contains only one state (i.e., state N) in it. State N (or *Next-row* state) denotes the situation that the first-fit process fails to find the free base in row r and will move to the next higher row;

Type III: This type of states contains $(W-w+1)$ states (i.e., from state 0 to state $(W-w)$) in it. State t where $0 \le t \le W-w$, denotes the situation that the allocation process reaches processor $[t, r]$ and will check whether this processor is a free base or not. All the states in type III are called *transient* or *searching* states. When the first-fit process reaches these states, the process stays in these states temporarily and will continue moving rightward until it reaches either state F or state N.

Clearly, the state in which the allocation process will stay at some time in the future is a random variable. Furthermore, since the occupied conditions of processors are assumed to be independent, the next state that the allocation process will move to is only dependent on the current state. Therefore, the state-transition diagram in Fig. 6 is reduced to a Markov chain and can be properly described through the use of the theory of stochastic (or random) process [13]. A variety of natural questions present themselves for the Markov chain to be answered is: What is the probability and how much is the cost that the first-fit process moves from state 0 to state F or to state N?

The first-fit process at the searching state t will move to one of the three types of states in the next move. That is, the next state is (1) state F, (2) state N, (3) or another searching state $(t+s)$, where $1 \le s \le W-w-t$. Let $¢_{t,k}$ and $C_{t,k}$ (where $k = F$ or N, or $t < k \le W-w$) respectively denote the one-step transition probability and the one-step transition cost that the first-fit process moves from state t to state k. Note that for simplifying the expressions, we use the variable 闪 such that 闪泌+钬铀= 1.

Lemma 1: Assume the first-fit process is currently at searching state t, where $0 \le t \le W-w$. Then $¢_{t,F} = (2^{-w})^h$, and $C_{t,F} = h$.

Lemma 2: Assume the first-fit process is currently at searching state t. Then $¢_{t,t+s} = (\pm^{s}2 + f(s,w) \times 2^{s}\pm) \times \sum_{k=1}^{h}(2^{-w})^{k-1}$, and $C_{t,t+s} = \sum_{k=1}^{h}(k \times (2^{-w})^{k-1}) / \sum_{k=1}^{h}(2^{-w})^{k-1}$, where $0 \le t < W-w$, $1 \le s \le W-w-t$, and f is an indication function such that $f(x, y) = 0$, if $x \ge y$; otherwise (i.e., $x < y$), $f(x, y) = 1$.

Lemma 3: Assume the process is currently at state t. Then $¢_{t,N} = 1 - \sum_{k=t+1}^{W-w} ¢_{t,k} - ¢_{t,F}$, and $C_{t,N} = \sum_{k=1}^{h} k \times (2^{-w})^{k-1} / \sum_{k=1}^{h}(2^{-w})^{k-1}$, where $0 \le t \le W-w$.

Theorem 1: When a task $T(w, h)$ attends the mesh $M(W, H)$, the probability that the first-fit process finds that there is no free base for the task is $(\,'_{0,N})^{H-h+1}$.

Theorem 2: When a task $T(w, h)$ attends the mesh $M(W, H)$, the probability that the first-fit process does not find a free base from row 0 to row (r 讣 1) but finds a free base in row r is $(\ /\ _{0,N})^r \times\ /\ _{0,F}$, where $1 \le r \le H-h$.

Corollary 1: Assume there is no free base for task $T(w, h)$ in mesh $M(W, H)$. The expected cost that the first-fit process finds there is no a free base for the task is $(H-h+1) \times\ \bar{}\ _{0,N}$.

Let $\bar{}\ _{t,F}$ 汹纡 $0 \le t \le W-w$) be the average cost when the first-fit process multiple-step moves from state t to state F, given that there exist free base(s) in this row.

Corollary 2: Assume there exists free base(s) for task $T(w, h)$ in mesh $M(W, H)$. The expected cost that the first-fit process does not find a free base from row 0 to row (r 讣 1) but finds a free base in row r is ($r \times\ \bar{}\ _{0,N} +\ \bar{}\ _{0,F}$), where $0 \le r \le H-h$.

Lemma 4: Assume the best-fit process is currently at state t. Then $¢_{t,F^t} = ¢_{t,F}$, and $C_{t,F^t} = C_{t,F}$, where $0 \le t \le W$ 讣 w.

Theorem 3: When a task $T(w, h)$ attends the mesh $M(W, H)$, the expected cost of the first-fit process is: $\sum_{r=0}^{H-h-1}\{\ (\ /\ _{0,N}\ )^r \times\ /\ _{0,F} \times (r \times\ \bar{}\ _{0,N} +\ \bar{}\ _{0,F}))+(\ /\ _{0,N}\ )^{H-h+1} \times (H-h+1) \times\ \bar{}\ _{0,N}$.

Theorem 4: The expected cost of the best-fit process in the leapfrog scheme is $(H-h+1) \times\ ¤_{0,N}$.

Theorem 5: When a submesh of size $w \times h$ is allocated or released, the expected cost that the leapfrog scheme takes to update the R-array is $w \times h + 1/(W$ 讣 $w+1) \times\ \sum_{k=0}^{W-w}\{k \times (\pm^k + ²^k) +\ \sum_{i=1}^{k-1}(i \times (²^i \times \pm^i + \pm \times ²^i))\} \times h$.

四、結論與建議

The R-array is a simple and statistical array to represent the occupied configuration of the mesh system. The statistical data plays as a guide to direct the allocation process to the next candidate processor.

This study is the first one to propose the precise analyses for the dynamic behavior of the processor allocation in the interconnection networks. The regular and simple structure of the mesh provides the platform to let the theory of the random walk can be applied. The analytical results show that the first-fit process takes very small cost to find the free base when the workload is smaller than 40% and that the best-fit process takes decreasing costs to find all the free bases when the workload is increasing. These results match the observation about the behaviors of the first-fit and best-fit processes. In addition, we also compute the probability whether the allocation process can find the free base. This result is the same for each full-recognition scheme, and can explain the myth why the upper bound of the system utilization of all the proposed schemes that allocate continuous spaces to tasks is hardly above 60%.

五、參考文獻

1. R. Alverson, *et al.*, The Tera computer system, *in* "Proc. 4th ACM Intl. Conf. on Supercomputing," pp. 1-6, 1990.
2. G. Barnes. *et al.* The Illiac IV computers, *IEEE Trans. on Comput.* C 17 (8) (Aug. 1968), 746-757.
3. S. Chittor, R. Enbody, Performance degradation in large wormhole-routed interprocessor communication networks, *in* "Proc. of the 1990 Intl. Conf. on Parallel Processing," vol. I, pp. 424-428, 1990.
4. P. Chuang and N. Tzeng, Allocating precise submeshes in mesh connected systems, *IEEE Trans. on Parallel and Distrib. systems*, 5 (2) (Feb. 1994), 211-217.
5. J. Ding and L. N. Bhuyan, An Adaptive submesh allocation for two-dimensional mesh connected systems, *in* "Proc. of the 1993 Intl. Conf. on Parallel Processing," vol. II, pp. 193-200, 1993.
6. V. Gupta and A. Jayendran, A flexible processor allocation strategy for mesh

connected parallel system, *in* "Proc. of the 1996 Intl. Conf. on Parallel Processing," vol. III, pp. 166-173, 1996.
7. R. Hockney and C. Jesshope, "Parallel Computers 2: Architecture, Programming and Algorithms," IOP Publishing Ltd, 1988.
8. T. Juang, Y. Tseng, Y. Chen, and C. Tsai, An Adaptive Processor Allocation Strategy in a Partitionable Two-Dimentional Buddy System, *in* "Proc. on the 1997 Intl. Conf. on Algorithms and Architectures for Parallel Processing," pp. 345-352, 1997
9. K. Li and K. Cheng, A two-dimensional buddy system for dynamic resource allocation in a partitionable mesh connected system, *J. Parallel and Distrib. Comput.* 12 (May 1991), 79-83.
10. Intel Corp. "A Touchstone DELTA System Description," 1991.
11. Intel Corp. "Paragon network queuing system manual," 1993.
12. R. Kessler and J. Schwartzmeier, CRAY T3D: A new dimension for Cray Research, *in* "Compcon Spring93," pp. 176-182, 1993.
13. L. Kleinrock, "Queuing System vol 1: Theory," Wiley-Interscience, 1975.
14. D. Lenoski, J. Laudon, *et al.* The Stanford DASH multiprocessor. *IEEE Trans. on Comput.* C 25 (3) (Mar. 1992), 63-79.
15. W. Liu, V. Lo, K. Windisch, and B. Nitzberg. Non-contiguous processor allocation algorithms for distributed memory multicomputers. *in* "Proc. ACM Intl. Conf. Supercomputing," pp. 227-236, 1994.
16. M. Noakes, D. Wallach and W. Dally, The J-machine multi-computer: An architectural evaluation. *in* "Intl. Symp. on Computer Architecture," pp. 224-235, 1993.
17. E. Smirni, C. Childers, E. Rosti and L. Dowdy, Thread Placement on the Intel Paragon: Modeling and Experimentation, *in* "Proc. of the 3rd Intl. Workshop on Modeling, Analysis, and Simulation of Computer and Telecom. Systems," pp. 226 –231, 1995.
18. P. Strobach and M. Siemens, Quadtree-structured recursive plane decomposition coding of images, *IEEE Trans. on Signal Processing*, 39 (6) (Jun. 1991), 1380-1397.
19. Y. Zhu, Efficient processor allocation strategies for mesh-connected parallel computers, *J. of Parallel and Distrib. Comput.* 16 (Dec. 1992), 328-377.

[1] Y. Zhu, Fast processor allocation and dynamic scheduling for mesh multiprocessors, *J. of Computer Systems Science & Engineering*. 11 (May 1996), 99-107.

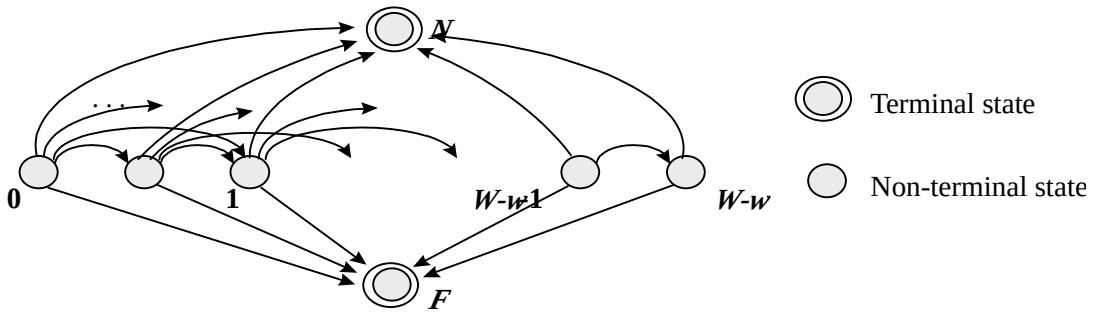

Fig. 1. The Markov chain depicting the behavior of the first-fit process in a row.

[2]