46

Parallel MRP system

T.Tsukishima, M.Sato, M.Takata, H.Matoba
Production Information System Department
Production Engineering Research Laboratory,
Hitachi, Ltd.
292 Yoshida-cho, Totsuka-ku, Yokohama-shi 244 Japan
TEL : 81-45-860-1651, FAX : 81-45-860-1621
E-mail : tsuki@perl.hitachi.co.jp

Abstract

Within an environment of rapid demand fluctuations, production planning changes are frequent and reductions in the plan proposal/revision cycle are sought. Also, an enlarged production planning scale is urgently sought to deal with integrated management of multiple, globally spread production points. To handle these demands, rapid processing of large scale data has become a major theme of Material Requirements Planning (MRP). In this paper a Parallel MRP System, using a loosely connected parallel computer, is described. A newly developed parallel algorithm, data transfer method and load balancing method are used and the system can reduce the 10 hours or more required by a conventional main frame computer for MRP calculations, on the 120,000 item level, to within 9 minutes using a structure with 6 WS servers. The processing speed of the 6 machine structure is 5.98 times the speed of 1 machine and the capabilities increase in a linear fashion. In this way simulated applications are possible for MRP in the workstation integrated management of multiple factories.

Keywords

Production Planning, MRP, Parallel Processing

1 INTRODUCTION

Within an environment of rapid demand fluctuations, production planning changes caused by design changes, part shortages and other problems are frequent

Computer Applications in Production and Engineering. F. Plonka and G. Olling (Eds.)

and reductions in the plan proposal/revision cycle are sought. Also, an enlarged production planning scale is urgently sought to deal with the integrated management of multiple, globally spread production points and other changes. To handle these demands, rapid processing of large scale data has become a major theme of Material Requirements Planning (MRP).

In relation to this, the research detailed here has carried out the development of a Parallel MRP System which allows the rapid processing of large-scale data, based on the premise of using a parallel processing computer.

2 TARGETS FOR INCREASING MRP PROCESSING SPEED

The targets for increasing MRP processing speed are given in Figure 1. The horizontal axis is processing time and the vertical axis is number of items, which gives an idea of the scale of the MRP, with the MRP processing speed being indicated by the line in the figure. Traditionally an MRP on the scale of 100,000 items would require at least ten hours on a mainframe, and be carried out once a week. With this being carried out several times a day, if an attempt was made to carry out the integrated management of multiple factories then more than 100 times the processing speed would be required. This research has set a goal of a speed increase with a wide range from 10- to over 100-times the traditional speed, by increasing the number of processing units of a parallel computer.

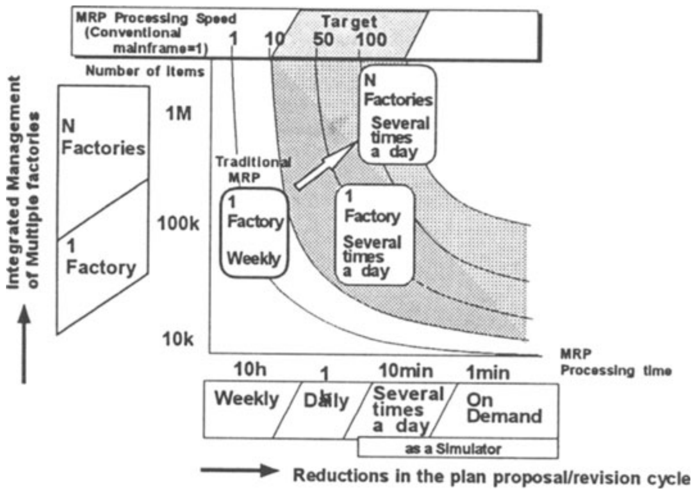


Figure 1 Targets for increasing MRP processing speed.

3 BASIC STRUCTURE AND TECHNICAL PROBLEMS

In MRP, in order to repeat the calculation of the requirements for all items that make up the product, dividing the items that must be calculated amongst the processing units and carrying out the processing in parallel is the basic idea behind this system. The basic structure of this system is given in Figure 2. The hardware consists of multiple Work Stations (WS) connected to a high-speed network. From here on work stations connected to a network will be referred to as Processing Elements (PE). This system has for each PE one MRP calculation process that calculates the require amount of items, and overall a single control process which controls the entire progress. In order to increase the processing speed, the necessary data is moved to the MRP calculation processes internal memory before the calculation commences, and reside there.

The technical problems are the three points listed below. Details will be explained in the following sections.

(1) High-speed data access,
(2) Load balance, and
(3) Parallel algorithm.

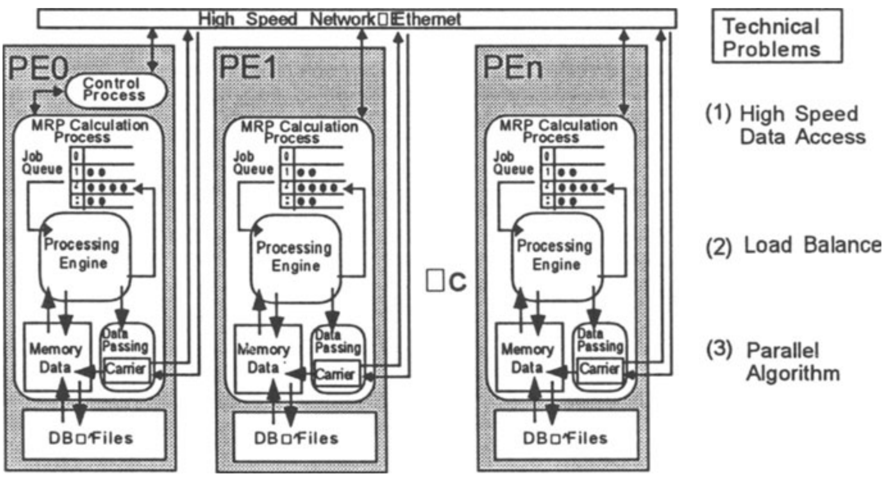


Figure 2 The Basic structure and technical problems.

4 HIGH-SPEED DATA ACCESS

A loosely connected parallel computer has been selected this time. The data is spread amongst the PEs and stored in their memories. Generally, with a loosely connected computer the data transfer between PEs is more than ten times greater in comparison to accesses confined to within a PE and it becomes an overhead for parallel processing. In this system, a reduction in the overhead that accompanies the parallelization has been achieved by transferring the requirements data for multiple items as a group, as shown in Figure 3. It will be referred to as the Requirements Carrier. The effectiveness of this buffering transferal is detailed along with the evaluation of the load balance method in the next section.

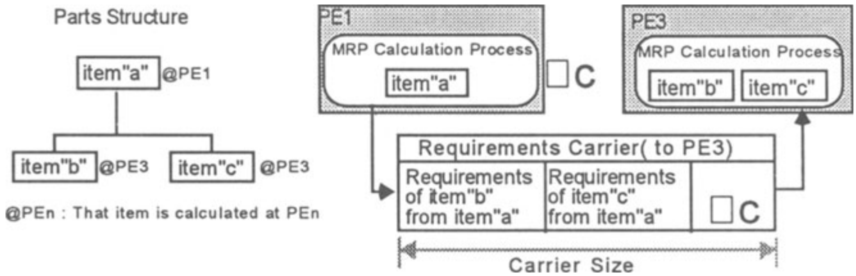


Figure 3 Data Transfer by the Requirements Carrier.

5 LOAD BALANCE

5.1 Load balance method

In order to increase the effectiveness of parallel processing it is necessary to make the amount of processing of each PE uniform. Basically with MRP, the number of items to be calculated is made uniform, but in developing this system the following two points were taken into consideration.

(i) When parent items and child items are allotted to different PE, a transfer of requirements data occurs. It is possible that if not only the number of items are unified for each PE here, but also parent/child items are allotted to the same PE wherever possible, then there will be a decrease in the overhead from data transfers between PE.

(ii) In the course of the actual process there will be items that do not require to be calculated because of an inventory allocation and so on. Because of this, there is the possibility that even if each item is allocated uniformly beforehand the load balance will be lost and there will be a decrease in processing speed.

Next, detailed explanations of the two load balance methods developed by this system will be given.

Parent/child based method

The parent/child based method is a load balance method developed in order to investigate the possibility (i) given above. This method processes parent and child in the same PE wherever possible, as well as making the number of items uniform between PE, and is a static load balance method which decides beforehand which PE should process each item, before execution of the MRP.

Dynamic load balance method

This is a load balance method developed to solve the problem (ii) given above. With a loosely connected parallel computer, dynamic load balance methods of the type which centrally manage the jobs in one PE result in a large overhead from data transmission. In relation to this, this system uses a method of leveling the load by moving items during the processing that have been assigned beforehand by static load balance. An outline of this function is shown in Fig. 4.

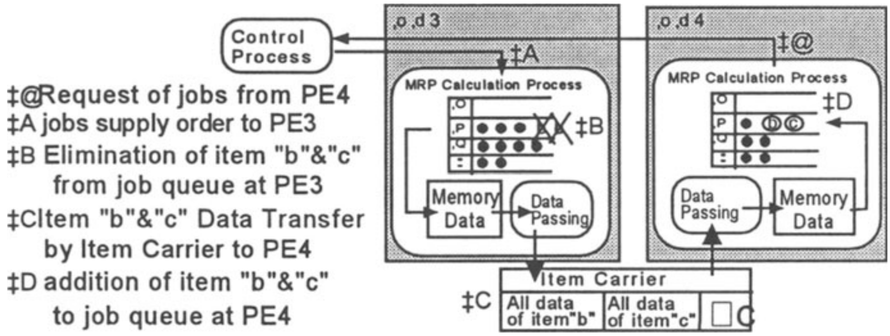


Figure 4 Dynamic load balance method.

5.2 Evaluation of methods

Parent/child based method

An evaluation will be given of the reduction in MRP calculation time by assigning parent/child items to the same PE. This result comes from the ability to reduce the amount of data transferred. As the transmission overhead per item here varies depending upon the number of items loaded on the Requirements Carrier shown in Figure 2, this result is affected by the carrier size.

The simple balance method here does not take into account parent/child relationships and is a method that only makes the number of items uniform.

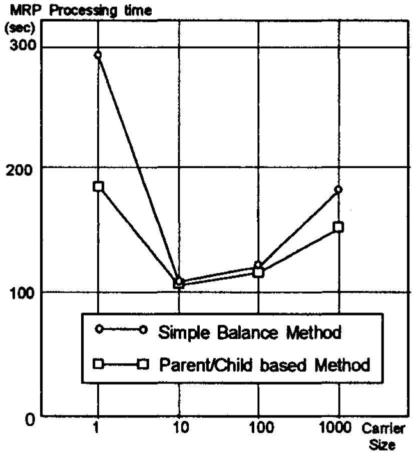
The results of the experiments in Table 1 are shown in Figure 5(a). For a carrier size of 10 the difference between both is only a few percent. Because a greater reduction in the inter-PE data transmission overhead was achieved than originally predicted, the difference between them was reduced. Taking into consideration the processing time of the load balance program itself, the simple load balance method which only makes the number of items uniform is of sufficient use.

Dynamic load balance

Under the conditions in Table 1, this method's performance was evaluated, with the load balanced unevenly so that for each level the assigned units of a specific PE was three times that of the others (Figure 5(b)). As will be explained later, with the hierarical parallel method, where the processing of each PE is done level-by-level at the same time, performance does degrade when there is a lack of uniformity in the load between levels. However, by adding the dynamic load balance function performance can be maintained of upto 90% of that at the time when the load is balanced uniformly.

Table 1 Experimental Condition

Hardware	*MPS*	*Parts Num.*	*Parts Structure*	*Inventory Data*
WS(131 MIPS) x 6	2000	120,000	12 levels	Nothing



(a) Carrier size and processing time

Figure 5 Performance of load balance method.

6 PARALLEL ALGORITHM

6.1 Two methods of Parallel Algorithm

In MRP there exists a restriction on the order of calculations such that the calculation for an item must be commenced after the calculation of requirements for that item's parent has been completed. In a parallel algorithm a synchronous mechanism is provided which fulfills the above restriction on the order of calculations. Furthermore there is a need to reduce two overheads mentioned below which is results from the parallelization.

(a) Processing time required by the synchronous mechanism itself

(b) The waiting time that occurs in the synchronization

In this system two methods have been developed for the parallel algorithm, the "hierarchical parallel method" and the "part parallel method", and their comparative evaluation was carried out. The details of these two methods are given below:

Hierarchical parallel method

The hierarchical parallel method simply applies the level-by-level calculation order, which uses the low level code of conventional non-parallel processing, as it is to the parallel algorithm. An outline of the process is shown in Figure 6. The calculation procedure repeats the calculation of items synchronizing them at each level.

Do until the final level
‡@When level n is finished, report and suspend. (a MRP Calculation Process)
‡A When all finish level n, order to resume. (Control Process)
‡B start level (n+1). (a MRP Calculation Process).

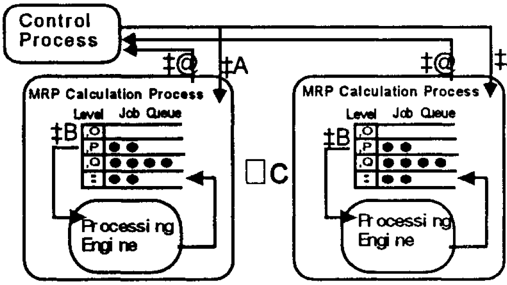


Figure 6 Hierarchical parallel method

Part parallel method

The part parallel method is a new method which differs from the level-by-level calculation order which uses conventional low level code. An outline of the process is shown in Figure 7.

By controlling the progress of the calculation at the item level in this way, items that can be calculated are regularly stored in the job queue, reducing the chance of synchronization waits occurring. Also, this method does not use any low level code. Because of this, while conventionally when there was some design change several hours were required to process updates to the low level code, now there is no need to carry out this processing. In other words, there is a smooth integration of design and manufacture.

Do until all items are finished
‡@Write the child items' requirements data directly or via the network.
‡ACheck whether all parent items have written to each child item when writing.
‡BIf so,add the child item in the job queue at the PE.

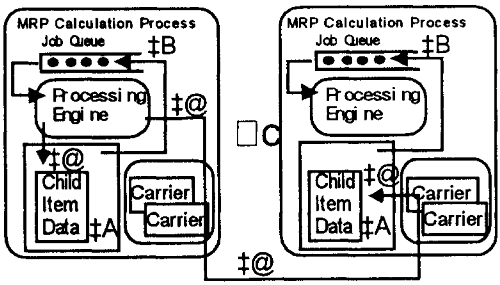


Figure 7 Part parallel method

6.2 Evaluation of the parallel algorithm method

A comparative evaluation of the hierarchical parallel method and part parallel method was carried out from the following viewpoint:

(1) Synchronization mechanism

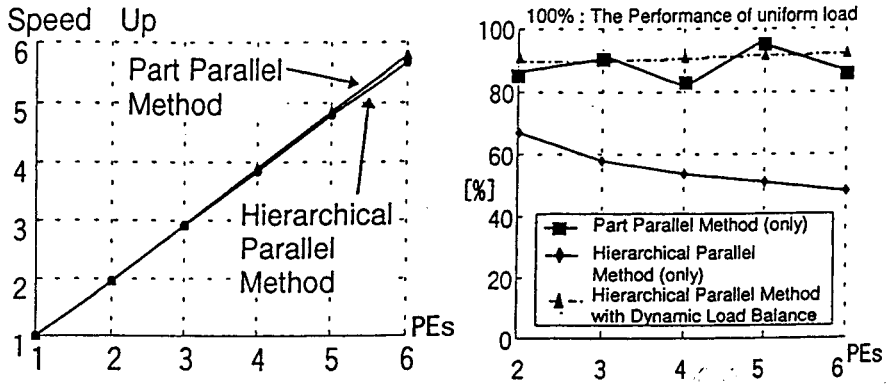
The amount of calculation for each MRP calculation unit was made uniform for each level in advance. Table 1 shows the experiment conditions, and Figure 8(a) shows the experiment results.

The performance of the part parallel method was equal to or greater than the hierarchical parallel method. From this, it can be seen that although the part parallel method's synchronization mechanism is complex, the processing time when the program is actually installed is the same as that of the hierarchical parallel method.

(2) Synchronization wait

The performance, under the conditions in Table 1, when the load is balanced so that for each level the number of items assigned to a particular PE is three times that of others is shown in Figure 8(b).

The part parallel method reduced the synchronization wait to the same level as the hierarchical parallel method using dynamic load balancing. Even if the amount of calculation for each PE is not uniform at each part level, if the load is balanced so that the total amount of calculation is uniform, as with these experiments, the part parallel method is thought to be superior from the simplicity of its mechanism.



(a) Performance of uniform load at each level (Totally uniform)

(b) Performance of ununiform load at each level (Totally uniform)

Figure 8 Evaluation of parallel algorithm

7 OVERALL PERFORMANCE EVALUATION

Based on the evaluation of each process method in the previous sections, an evaluation of the parallel MRP system's overall performance was carried out. The experiment conditions are shown in Table 1. The following was gained from the experiment results.

1) As shown in Figure 9, the system reduced the 10 hours or more required by a conventional main frame computer for MRP calculations, on the 120,000 item level, to within 9 minutes (67 times the processing speed including DB access) using a structure with 6 WS servers.

2) As shown in Figure 8(a) and 9, the processing speed of the 6 machine structure is 5.98 times the speed of 1 machine and the capabilities increase in a linear fashion. Also, a highly efficient parallelization of 95% or more was achieved for a wide variety of Bill of Materials (BOM) forms such as shown in Table 2.

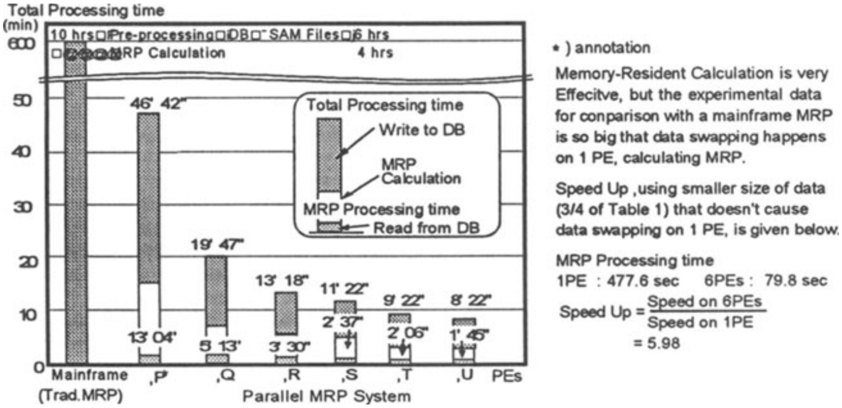
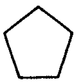


Figure 9 Performance of parallel MRP system

Table 2 Parallel efficiency (6PE against 2PE) with a wide variety of BOM form

BOM Form	*Rough Sketch*	*Parts Num.*	*Speed Up(6PE/2PE) (Ideal = 3)*	*Efficiency (SpeedUp/Ideal)*
				

8 CONCLUSION

A parallel MRP system has been developed for the high-speed processing of large-scale data. This system can reduce the 10 hours or more required for conventional MRP calculations, on the 120,000 item level, to within 9 minutes using a structure with 6 WS servers. Also, the capabilities increase in a linear fashion.

REFERENCES

D.N.P.Murthy, L.Ma (1991) MRP with Uncertainty: A Review and Some Extensions. *International Journal of Production Economics*, **25** pp.51-64

D.Turbide (1995) MRP II. *IIE SOLUTION* **July** pp.28-31

W.Chamberlain, G.Thomas (1995) The Future of MRP II. *IIE SOLUTION* **July** pp.32-39

N.Carriero, D.Gelernter (1990) How to write Parallel Programs. The M.I.T. Press, Boston

BIOGRAPHIES

Takahiro Tsukishima was born in 1965. He received a B.S. degree in 1988 and a M.S. degree in 1990 in Precision Engineering from Tokyo University. He is a researcher at the Production Research Engineering Laboratory, Hitachi, Ltd. His present interests include supply chain management and parallel processing. He is a member of JSPE and JIMA.

Masahiro Sato was born in 1969. He graduated from Hitachi Keihin College of Technology in 1994. He is a researcher at the Production Research Engineering Laboratory, Hitachi, Ltd. His present interests include supply chain management and parallel processing. He is a member of JIMA.

Masahito Takata was born in 1966. He graduated from Hitachi Keihin College of Technology in 1988. He is a researcher at the Production Research Engineering Laboratory, Hitachi, Ltd. His present interests include supply chain management and network computing.

Hideaki Matoba was born in 1954. He received a B.S. degree in 1977 and a M.S. degree in 1979 in Mechanical Engineering from Kobe University. He is a senior researcher at the Production Research Engineering Laboratory, Hitachi, Ltd. His present interests include supply chain management and agent systems. He is a member of JSME and JIMA.