An Optimized Algorithm for Network on Chip

S. Subha
School of Information Technology &
Engineering,
VIT, Vellore, India

ABSTRACT

Network on a chip can be viewed as processors with various instruction sets residing on a chip. Programs issued to a particular processor type can be divided into sequential and parallel code. Each subtask is characterized by an estimated time for completion. This paper proposes a method to determine the topological arrangement of processors to minimize the total execution time. The tasks are assumed to be allocated based on the algorithm proposed in literature. The logical arrangement of processors is in a directed acyclic connected graph. This is achieved in a tree arrangement. The expression for total execution time in this topology is derived. The model is simulated and the model verified

General Terms

Computer Archtiecture

Keywords
Allocation in CMP, Optimization of scheduling, topology of n processors, task scheduling

1. INTRODUCTION

Network on Chip is an emerging concept. The basic architecture in it consists of many chips located on a single chip sharing other computer resources. The chips can be from the same family making it homogenous or different families making it heterogeneous. The resources that are shared among the chips are the memory, I/O, bus. On the memory side the main memory is shared among the processors. In addition the system has cache. Usually two levels of cache are definitely on the chip in current technology. These two levels of cache can be local to a chip or shared. Some research has been done in the topology of the cache in CMP. Nihalal architecture is assumed for cache design. The other important aspect of NOC is process scheduling. Without loss of generality this paper assumes a heterogeneous NOC. The scheduling algorithm proposed in [6] is assumed. The topology of the processors determines the communication cost. The total cost is determined for an optimal processor arrangement. This paper validates the proposed topology for a given environment.

The rest of the paper is organized as follows. Section 2 gives the motivation, section 3 the proposed architecture, section 4 simulation, section 5 conclusion, section 6 references.

2. MOTIVATION

Consider a NOC with four CPUs A, B, C, D. Let two of the CPUs A, B belong to one family. The rest two belong to different family of processors. Consider a job submitted to be executed on A, B. Let the job be submitted to CPU A. Let the distance between A and B is 6nm. Let the data transfer rate be one nanosecond for each slice of the task. The job can be divided into serial and parallel parts. The serial part has to be executed in serial execution fashion on A or B. The parallel part of the job which is usually interspersed with the serial parts can be scheduled on both the processors A and B. Assume that there is shared main memory and the cache is according to the Nihalal architecture. Consider one parallel piece of code of the job that is to be scheduled on A, B. Assume each of the processors has its own job queue. Let the average waiting time in the queue be 10 and 5 units of time. The parallel code gives an estimate of the total time required to process it as in uniprocessor system. Let this be 4 units of CPU time. Let there be three such four unit time slots needed for the code that is parallelized. Consider the traditional round robin allocation strategy. Let the allocation start in processor B. The three units are allocated as two time slices to B and one to A. The total time taken on any processor calculated as follows

Time taken on A + time taken on B + communication cost

Time taken on A in round robin = 10+4 = 14

Time taken on B in round robin = 2*5+2*4 = 18

Time taken in communication to B = 2*6*one nanosecond = 12ns

Assume each CPU time unit is one ns.

The total time taken = (14+18+12)ns = 44ns.

Now consider the following method. Let x_1 and x_2 units of the parallel job be distributed among A and B. The total time taken is given by the expression

T = 2 d_2 *time for data transfer + x_2 *(5+4) + x_1*(10+4)

Where d_2 is the distance between A and B. The objective is to minimize the value of T. This can be expressed as

min T

subject to

d_2 = 12

x_1 >=0

x_2 >= 0

$$x_1 + x_2 = 3 \qquad (1)$$

All the variables in (1) have integer solution. Based on the assumption made the optimization problem gives the following solution.

2 units to A and one unit to B. The total time taken is given by 2*6*1+2(14)+1(9)=12+28+9=39 ns.

An improvement in the performance is observed. The distance between the processors d_2 can be minimized based on the topology. The simplest connected graph would be a tree with one edge. This is the motivation of this paper.

3. PROPOSED MODEL

Consider a NOC with n number of processors. Let these belong to m distinct families. Let the number of processors in each family be $t_1, t_2, ..., t_m$. Then we have

$$t_1 + t_2 + ... + t_m = n \qquad (2)$$

Consider a program. Assume it can be scheduled on q number of processor families. The architecture of the system is assumed to accommodate such scheduling. The program consists of sequential and parallel code. The sequential part of the code can be executed on one processor only. The parallel portion of the code can be scheduled among the available processors. Assume a distributed memory model. Thus the program structure consists of the following:

Serial part-1

Parallel part-1

Serial part -2

Parallel part-2

…….

Serial part-q

Parallel part-q

Serial part-q+1 (3)

The serial part execution proceeds after the parallel part synchronizes in all processors. The total time is the sum of communication and computation cost. Consider the computation cost.

Consider the parallel part of the code that is to be scheduled on processor belonging to family f. Assume the separation of the parallel part is done on processor i. This might belong to any family. For the sake of simplicity assume it belongs to family f. Let the total amount of parallel part to be scheduled be c CPU units of time. Let there be $r = t_f$ number of processors belonging to family f. Assume each processor has its own CPU queue. Let the average waiting time in each of the processors belonging to family f be

$w_1, w_2, .., w_{t_f}$. Let the distance from processor i to the various r processors of the same family be $d_{i1}, d_{i2}, ..., d_{ir}$ respectively.

$p_{i1}, p_{i2}, ..., p_{ir}$ be the number of CPU units of service requested by processor i from various processors belonging to family f. The total time taken to process the c units is given by the following expression.

$$p_{i1}(1+w_1) + p_{i2}(1+w_2) + \ldots + p_{ir}(1+w_r) \qquad (4)$$

Assuming round robin schedule of the tasks within each of the r processors. The objective is to minimize the expression in (4). The objective function for this computation is given by

$$\min\left(\max\left(p_{i1}(1+w_1), p_{i2}(1+w_2), \ldots, p_{ir}(1+w_r)\right)\right)$$

subject to

$$p_{i1} \geq 0$$
$$p_{i2} \geq 0$$
$$\ldots.$$
$$p_{ir} \geq 0$$
$$w_1 \geq 0 \qquad (5)$$
$$w_2 \geq 0$$
$$\ldots$$
$$w_r \geq 0$$
$$p_{i1} + p_{i2} + \ldots + p_{ir} \approx c$$

The memory model chosen is distributed memory model and hence data transfer is required.

Now consider the round robin algorithm for scheduling the c units of job among the r processors. Assume r divides c. Each processor gets c/r units of job. The total time taken in this case is given by

$$\text{Max}\left((w_1+1)\frac{c}{r}, (w_2+1)\frac{c}{r}, \ldots, (w_r+1)\frac{c}{r}\right) \qquad (6)$$

A performance improvement is seen when

$$\min\left(\max\left(p_{i1}(1+w_1), p_{i2}(1+w_2), \ldots, p_{ir}(1+w_r)\right)\right)$$

<=

$$\text{Max }\left((w_1+1)\frac{c}{r}, (w_2+1)\frac{c}{r}, \ldots, (w_r+1)\frac{c}{r}\right) \qquad (7)$$

Consider the communication cost. Assume that the communication is between r processors of the same processor family. The objective is to minimize this cost. The simplest topology is to connect each of the r processors to each other. However this involves lot of book keeping. The next simplest topology is to arrange the r processors as a tree with r-1 edges. Let it be a binary tree. Let the height of the tree is h starting from level zero. Then the maximum communication is $2h\left(2^{\frac{h}{2}}\right)$ units which is obtained when the leaf nodes communicate with each other. As the height of the tree decreases this communication cost decreases. The minimum communication cost would be 2 units. The total cost would be sum of communication and computation cost. For a program with q parallel parts, the total cost would be

Total Cost=Computation cost + communication cost

i.e. Computation part_serial + Computation part parallel + communication part-parallel

i.e.

$$T_{s1} + T_{s2} + \ldots + T_{s(q+1)} + T_{p1} + 2T_{communication1} + \ldots + T_{pq} + 2T_{communicationq} \quad (8)$$

where T_{si} is the time for serial computation in i-th serial step, T_{pi} is the time for parallel computation in i-th parallel step, $T_{communicationi}$ is the communication time during i-th parallel computation. The objective function would be

$$\min\begin{pmatrix} T_{s1} + T_{s2} + \ldots + T_{s(q+1)} + T_{p1} + 2T_{communication1} + \\ \ldots + T_{pq} + 2T_{communicationq} \end{pmatrix} \quad (9)$$

The optimized value is realized based on the heuristics given earlier in this section. This involves a tree topology for the nodes with the scheduling given in equation (5).

4. SIMULATION

The proposed model was simulated with the parameters given in Table 1.

The proposed method gave the solution on substituting the values in (5) as (8, 6, 16, 9, 11). The maximum is 16 units. Assume communication takes six units of time. The total time taken is 3+16+(2*6) = 31 units of time. This solution was obtained by formulating the equations and solving it using MS-Excel Solver package. The traditiponal round robin technique would assign eight units of CPU request to each of the processors. Assuming round robin scheduling inside each processor, the total time taken is equal to (1+7)8+(1+10)8+(1+3)8+(1+6)8+(1+5)8+3+2*6 = 303 units. A performance improvement of 89% is seen in this example for parameters chosen.

Table 1 Parameters for simulation

Name of Parameter	Value
Total CPU units required	50
Number of processors	5
Waiting time on processor #1	7
Waiting time on processor #2	10
Waiting time on processor #3	3
Waiting time on processor #4	6
Waiting time on processor #5	5
Serial time	3 units for one serial task

5. CONCLUSION

An algorithm to arrange the processors and schedule tasks in them using round robin technique is proposed in this paper. The algorithm involves formulating an optimization function for minimizing the total time involving computation and communication costs. The proposed algorithm is simulated for an example and the results validated with an improvement of 89% over round robin technique of same topology.

6. REFERENCES

[1] Ceyda Oguz, M.Fikret Erca, etal, Heuristic Algorithms for Multiprocessor Task Scheduling in a Two-Stage-Flow-Shop, 2000

[2] Dror G.Feitelson, Job Scheduling in Multiprogrammed Parallel Systems, 1997

[3] Fang Wang, Scheduling in Multiprogrammed Parallel Systems , Research Report RC 19790 (87657), IBM T.J.Watson Research Center, 1997

[4] Shashi Kumar et al, A Network on Chip Design and methodology, Proceedings of ISVLSI'02

[5] Uwe Schwiegelshohn, Ramin Yahyapour, Analysis of First – Come- First –Serve Parallel Job Scheduling, Proceedings of the 9th SIAM Symposium on Discrete Algorithms, 1998

[6] S.Subha, A Scheduling Algorithm for Network on Chip, Proceedings of ACT 2009, pp-289-291