Performance Analysis of UMA and NUMA Models

Vanya Rajput[1], Sanjay Kumar[2], V.K.Patle[3]
School of Studies in Computer Science
Pt.Ravishankar Shukla University,Raipur,C.G.
[1]vannu23@gmail.com
[2]sanraipur@rediffmail.com
[3]patlevinod@gmail.com

Abstract - Uniform memory access, non uniform memory access and cache only memory access are the three memory designs of multiprocessor. All the processors in UMA model share the physical memory uniformly. In the UMA architecture, access time to a memory location is independent from which processor makes the request. In the NUMA model memory access time depends relative to a processor and logically follow in scaling from symmetric multiprocessing (SMP) architectures. At the last COMA is memory model in which local memories at each node are used as cache. This is in contrast to using the local memories as actual main memory, as in NUMA organizations. All the models provide some facilities and have some drawbacks too. There are many criteria's on which performance of a multiprocessor system can be analyzed. In this paper we present the frame work of multiprocessor architecture, than we discuss the memory models after that we are going to compare performances of memory models.

***Keywords*— UMA, NUMA, OMP, Speedup, Memory Contention.**

I. Introduction

The concept of parallel processing is for the simultaneous use of more than one CPU to execute a program. Parallel processing makes a program run faster because there are more engines (CPUs) running it[1]. In computing, shared memory is memory that may be simultaneously accessed by multiple programs with an intent to provide communication among them or avoid redundant copies. Shared memory is an efficient means of passing data between programs. Depending on context, programs may run on a single processor or on multiple separate processors. View of data and the communication between processors can be as fast as memory accesses to a same location A shared memory system is relatively easy to program since all processors share a single. Distributed Shared Memory (DSM) is a memory area shared by processes running on computers connected by a network. DSM provides direct system support of the shared memory programming model[2]. Parallel architecture divides in two parts shared memory architecture and distributed memory architecture. In the shared memory all processors share the same memory while in the distributed memory architecture all processors have their own local memory. Further shared memory architecture divides in two memory models. UMA and NUMA are the examples of multiprocessor which works on the basis of shared memory.[3]

II. UMA

Uniform Memory Access (UMA) is a shared memory architecture used in parallel computers. The UMA Network consists of one or more access points and one or more UMA Network Controllers, interconnected through a broadband IP network. All the processors are same and have equal access times to all memory parts[4]

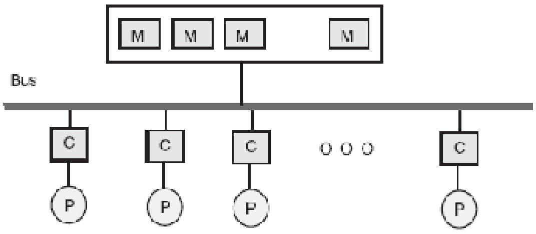

Fig1 UMA Architecture[8]

III. NUMA

NUMA is a memory model which is used in multiprocessing. It is an influential memory model for multiprocessing. In NUMA architecture each processor has its own local memory which are connected through network interfaces[5].In NUMA some part of memory are connected with different buses from other parts. NUMA uses two types of memories local and foreign memory so it will take more time to access some part of memory than other . Local memory is the memory that is on the same node as the CPU currently running the thread. Any memory that does not belong to the node on which the thread is currently running is foreign. Foreign memory is also known as *remote memory*.

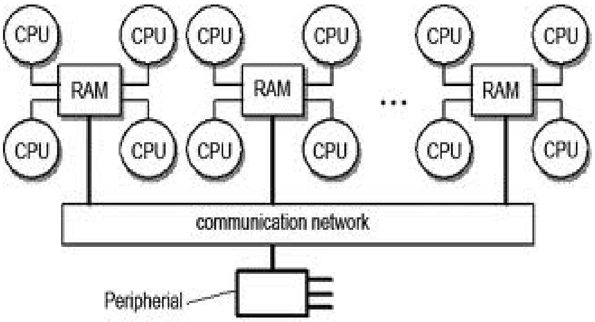

Fig.2 NUMA Architecture[9]

IV. EXPERIMENTAL SETUP

We use OpenMP for the evaluation of memory models. The programs are compiled using gcc with full optimizations and relaxed floating points options. Our datas are calculated on two systems with different memory architectures UMA and NUMA[6].

V. PERFORMANCE COMPARISON

Performance comparison of UMA and NUMA in the terms of memory contention

TABLE I

No of cores	(UMA)Memory Contention	(NUMA)Memory Contention
1	O	0
2	0.3	0.1
3	1.1	0.3
4	3.8	0.5
5	3.9	0.7
6	4.2	0.9
7	4.9	1.4
8	7.5	1.9
9		2.0
10		2.4
11		2.6
12		2.8
13		3.0
14		3.2
15		3.4
16		3.6

Memory contention means a situtíon in which two different programs, or two parts of a program, try to read items in the same block of memory at the same time[6].By this table we can see that NUMA has less memory contention instead of UMA[7].

Performance Comparison of UMA and NUMA in the terms of speedup

TABLE 2

No of cores	(UMA) Speedup	(NUMA) Speedup
1	1	1
2	1.4	1.8
3	1.3	2.4
4	0.8	2.8
5	1	3.2
6	1.1	3.3
7	1.2	3.0
8	0.9	2.6
9		2.7
10		2.8
11		2.9
12		3.0
13		3.1
14		3.2
15		3.3
16		3.4

Speedup formula

$$S_p - \frac{T_1}{T_p}$$

p is the number of processors

T_1 is the execution time of the sequential algorithm T_p is the execution time of the parallel algorithm with *p* processors[10]

Through this table we can see that number of cores that maximizes the speedup is 6 on NUMA[7]

VI.CONCLUSION

In this paper we compare the performance of UMA and NUMA in the terms of memory contention and speedup. The number of inputs required by the model is independent of any characteristic of the program such as number of cores, and depends only on the architecture of the memory systems. We used our model to derived the number of cores that maximizes the speedup. It can reduce the execution cost, by reducing theexecution time and number of cores required. This model can help to study the benefits of switching from an UMA system to NUMA system.

REFERENCES

[1] www.webopedia .com

[2] Taxonomy-Based Comparison of Several Distributed Shared Memory Systems ,* Ming-Chit Tam ,Jonathan M. Smith ,David J. Farber,Distributed Systems Laboratory, Dept. CIS, University of Pennsylvania, Philadelphia, PA 19104-6389, May 15, 1990

[3] A Real-Time Java Chip-Multiprocessor CHRISTOF PITTER, Vienna University of Technology, Austria And MARTIN SCHOEBERL,Vienna University of Technology, Austria

[4] Non-Unifrom Memory Access(NUMA) Nakul Manchanda and Karan Anand, New York University, {nm1157,ka804}@cs.nyu.edu

[5] Model-Based, Memory-Centric Performance and Power Optimization, ChunYi Su† Dong Li‡ Dimitrios S. Nikolopoulos§ Kirk W. Cameron† Bronis R. de Supinski\ Edgar A. Le´on\Department of Computer Science† Oak Ridge ‡ Lawrence Livermore\ Queen's University§Virginia Tech, VA 24060 National Laboratory National Laboratory of Belfast, {sonicat,cameron}@vt.edu Oak Ridge, TN 37831 Livermore, CA 94550 Belfast,Northern Ireland, UK lid1@ornl.gov {bronis,leon}@llnl.gov d.nikolopoulos@qub.ac.uk

[6] www.answers.com › Library › Science › Sci-Tech Dictionary

[7] B.Tundor and Y.M. Teo, A Practical Approach for Performance Analysis of Shared-Memory Programs, Proceedings of 25th IEEE on NUMA Multiprocessors, International Parallel & Distributed Processing Symposium, anchorange, USA, May 16-20,2011.

[8] siber.cankaya.edu.tr/ozdogan/...old/ceng505/node69.html

[9] numa.png people.sc.fsu.edu

[10] en.wikipedia.org/wiki/speedup