Memory Server Architecture for Parallel and Distributed Computers

Bala Dhandayuthapani Veerasamy

***Abstract*—** In the past programming life, we were mostly using sequential programming. But, today's life style is going with more faster than the past decades. Also, solving problems on the computers are enormous. Parallel computer can executes two or more job within a same period of time. The effective performance of a program on a computer relies not just on the speed of the processor but also on the ability of the memory system to feed data to the processor [1]. Thus parallel computers are required more memory space than the normal computers. This paper encompasses the innovative thought on memory architectures for parallel computers.

***Keywords*—** Distributed Memory, Hybrid Memory, MPP, Server Memory, Shared Memory, SMP.

I. Introduction

THE term processor, or microprocessor, refers to the central processing unit (CPU) [3],[6]. It's a single chip responsible for the execution of instructions given by a computer program, memory management and address translation, integer and floating point operations, and cache management. Uniprocessors have a single processor. Application instructions and hardware calls are executed in order, one at a time (sequentially). The system appears to run concurrent processes, but the processor actually switches back and forth between instructions.

Two events are said to be concurrent if they occur within the same time interval. Two or more tasks executing over the same time interval are said to execute concurrently [4]. Tasks that exist at the same time and perform in the same time period are concurrent. Concurrent tasks can execute in a single or multiprocessing environment. In a single processing environment, concurrent tasks exist at the same time and execute within the same time period by context switching. In a multiprocessor environment, if enough processors are free, concurrent tasks may execute at the same instant over the same time period. The determining factor for what makes an acceptable time period for concurrency is relative to the application.

Concurrency techniques are used to allow a computer program to do more work over the same time period or time interval. Rather than designing the program to do one task at a time, the program is broken down in such a way that some of the tasks can be executed concurrently. In some situations, doing more work over the same time period is not the goal. Rather, simplifying the programming solution is the goal. Sometimes it makes more sense to think of the solution to the problem as a set of concurrently executed tasks. Memory performance begins to affect overall system performance in two instances [5]. The first instance occur when the system is unable to retrieve and store data from physical memory fast enough, or when the system is forced to travel to main memory frequently. The second, and more likely, case is that the demand for physical memory by all currently running applications, including the kernel, exceeds the available amount. Hence, the need for parallel computer memory architectures commenced to solve memory issue. There are three different architectures usually allowed parallel computers to perform operations. There are shared memory architectures, distributed memory architectures, and hybrid memory architectures.

A. Shared Memory

Shared memory (See Fig. 1) parallel computers [3] are varying widely, but generally have in common the ability for all processors to access all memory as global address space. It has multiple processors can operate independently but share the same memory resources. Changes in a memory location effected by one processor are visible to all other processors.

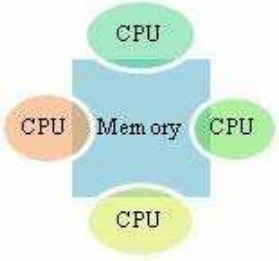

Fig.1 Shared Memory Model has four CPU, can access centralized memory through global address space.

Shared memories machines can be divided into two main classes based upon memory access times are UMA and NUMA. Uniform Memory Access (UMA).This is commonly represented by Symmetric Multiprocessor (SMP) machines. It has identical processors. It has equal access and access times to memory. It is sometimes called CC-UMA - Cache Coherent UMA. Cache coherent means if one processor updates a location in shared memory, all the other processors know

Department of Computing, Mekelle University, Mekelle, Ethiopia (Phone: +251 914740551; e-mail: dhanssoft@gmail.com).

about the update. Cache coherency is accomplished at the hardware level. Non-Uniform Memory Access (NUMA). It is often made by physically linking two or more SMPs. One SMP can directly access memory of another SMP. All processors will not have equal access time to all memories. It can access the memory across link as very slower. If cache coherency is maintained, then may also be called CC-NUMA - Cache Coherent NUMA.

B. Distributed Memory

Like shared memory systems, distributed memory systems vary widely but share a common characteristic. Distributed memory systems require a communication network to connect inter-processor memory [7]. (See Fig. 2)

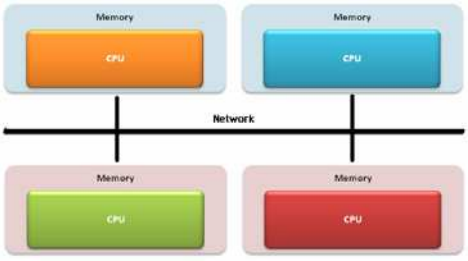

Fig. 2 Distributed Memory Access Model, has a CPU with local memory, can distribute memory among networked computers.

Processors have their own local memory. Memory addresses in one processor do not map to another processor, so there is no concept of global address space across all processors. Because each processor has its own local memory, it operates independently. Changes it makes to its local memory have no effect on the memory of other processors. Hence, the concept of cache coherency does not apply. When a processor needs access to data in another processor, it is usually the task of the programmer to explicitly define how and when data is communicated. Synchronization between tasks is likewise the programmer's responsibility. Massively parallel processors (MPP) usually use a specially designed network. Clusters of workstations usually use system/local area networks. Grid computers use the Internet as the networks.

C. Hybrid Distributed-Shared Memory

The combinations of shared and distributed memory architectures are called hybrid distributed-shared memory. The largest and fastest computers use both shared and distributed memory architectures. (See Fig. 3)

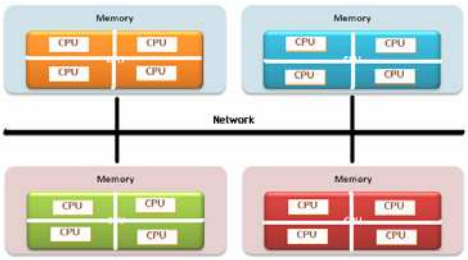

Fig. 3 Hybrid Distributed Shared Memory Model also called Hybrid memory. It is the combination of distributed and shared memory.

The shared memory component is usually a cache coherent SMP machine. Processors on a given SMP can address that machine's memory as global. The distributed memory component is the networking of multiple SMPs. SMPs knows only about their own memory - not the memory on another SMP. Therefore, network communications are required to move data from one SMP to another. Current trends seem to indicate that this type of memory architecture will continue to prevail and increase at the high end of computing for the foreseeable future.

II. SMP's, MPP's, Architectures

A. Symmetric Multiprocessing

Symmetric MultiProcessing (See Fig. 4) [7],[2] is a multiprocessing architecture in which multiple CPUs, residing in one cabinet, share the same memory. It is the tightly-coupled process of program tasks being shared and executed, in true parallel mode, by multiple processors who all work on a program at the same time. The term SMP is so closely associated with shared memory that it is sometimes misinterpreted as standing for "shared memory parallel". By design, all processors in an SMP system have equal claim on the shared resources time. SMP systems range from two to as many as 32 or more processors. However, if one CPU fails, the entire SMP system is down. Clusters of two or more SMP systems can be used to provide high availability. SMP systems allow any processor to work on any task no matter where the data for that task is located in memory; with proper operating system support, SMP systems can easily move tasks between processors to balance the workload efficiently. In SMP, CPUs are assigned to the next available task or thread that can run concurrently.

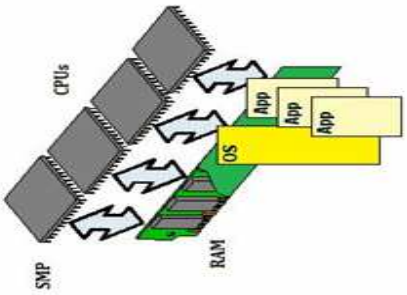

Fig. 4 SMP Memory Model

B. Massively Parallel Processing

Massively Parallel Processing (MPP) [7], in its most common form, is a loosely-coupled process of program tasks being shared and processed by multiple processors. The distinction is that each processor, with its own operating system and memory, works in a coordinated manner to process the different parts of the program. This is

accomplished with the help of message passing libraries and subroutines, such as those specified by MPI or PVM3. This variety of supercomputer avoids the memory bottleneck problems by having memory that is dedicated to each processor, but adds a layer of complication for programmers who write the parallel applications that utilize them (See Fig. 5). In MPP operation, the problem is broken up into separate pieces, which are processed simultaneously.

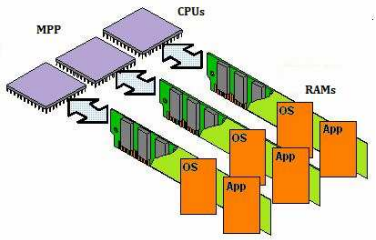

Fig. 5 MPP Memory Model

III. Server Memory Architectures

In the world, we are using different kinds of server technology to serve specific data to its client such as mail server, file server, video server, music server, web server. Likewise there should be a server used at the network called memory server (See Fig. 6). The server computer architectures will have multiprocessing environment with two different memories, one is shared local memory and the other is shared and distributed memory. And the server computer will connect with the hybrid distributed-shared memory architecture. All the server processors will use shared local memory. Shared and distributed memory will be used by hybrid distributed-shared memory architecture computers over the network.

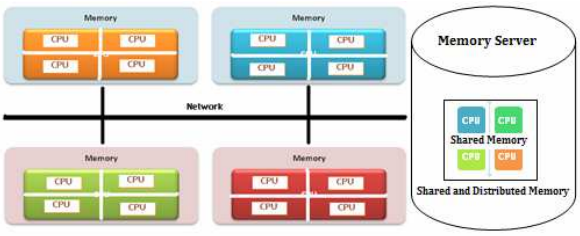

Fig. 6 Memory Server Architectures has hybrid distributed and shared memory with server memory.

This memory will be used moreover for client/server architecture programs or any distributed environments. This server should allow the any client computer to access primary memory through software or programs. It can be used for any software, which is sharable from server machine, any middleware programs, any server side scripting. Moreover the program that is accessible from server can utilize the server memory of shared and distributed rather than using the client memory of local or global memory. In this architecture, server is responsible for the details associated with data communication among computers. It usually allows non-uniform memory access times, so some client will utilize more memory and some may not.

IV. Conclusion

This architecture allows software or programs to take advantage of memory located on the Internet, on corporate and organization intranets, and on networks. It usually involves network programming in one form or another such as parallel and distributed programs, server side scripting, middleware programs, broker architectures and any client server related programs. This architecture is very powerful to solve scarcity of memory usage over the computer networks.

References

[1] Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar, Introduction to Parallel Computing, Second Edition, Addison Wesley (January 2003)
[2] Bernd Mohr, Introduction to Parallel Computing, Computational Nanoscience, 2006
[3] Hesham El-Rewini, Mostafa Abd-El-Barr, Advanced Computer Architecture And Parallel, A John Wiley & Sons, Inc Publication (2005).
[4] Maurice Herlihy, Nir Shavit, The Art of Multiprocessor Programming, Elsevier (2008).
[5] Mike Loukides, Gian-Paolo D. Musumeci, System Performance Tuning, Second Edition, O'Reilly (2002).
[6] Mostafa Abd-El-Barr, Hesham El-Rewini, Fundamentals of Computer Organization and Architecture, A John Wiley & Sons, Inc Publication (2005).
[7] Tobias Wittwer, An Introduction to Parallel Programming, VSSD (2006).

Bala Dhandayuthapani Veerasamy was born in Tamil Nadu, India in the year 1979. The author was awarded his first masters degree M.S in Information Technology from Bharathidasan University in 2002 and his second masters degree M.Tech in Information Technology from Allahabad Agricultural Institute of Deemed University in 2005. He has published more than fifteen peer reviewed technical papers on various international journals and conferences. He has managed as technical chairperson of an international conference. He has an active participation as a program committee member as well as an editorial review board member in international conferences. He is also a member of an editorial review board in international journals.
He has offered courses to Computer Science and Engineering, Information Systems and Technology, since 8 years in the academic field. His academic career started in reputed engineering colleges in India. At present, he is working as a Lecturer in the Department of Computing, College of Engineering, Mekelle University, Ethiopia. His teaching interest focuses on Parallel and Distributed Computing, Object Oriented Programming, Web Technologies and Multimedia Systems. His research interest includes Parallel and Distributed Computing, Multimedia and Wireless Computing. He has prepared teaching material for various courses that he has handled. At present, his textbook on "An Introduction to Parallel and Distributed Computing through java" is under review and is expected to be published shortly. He has the life membership of ISTE (Indian Society of Technical Education).