

Convergence of Parallel Architecture

AUTHOR- Ms. P.ANITHA
CO-AUTHOR- DR.G. RAMESH
M.KAMARUNISHA

INTRODUCTION-

Parallel machines have been developed with several distinct architecture. In this section, we will discuss different parallel computer architecture and the nature of their convergence.

Communication Architecture

Parallel architecture enhances the conventional concepts of computer architecture with communication architecture. Computer architecture defines critical abstractions (like user-system boundary and hardware-software boundary) and organizational structure, whereas communication architecture defines the basic communication and synchronization operations. It also addresses the organizational structure.

Programming model is the top layer. Applications are written in programming model. Parallel programming models include –

- Shared address space
- Message passing
- Data parallel programming

Shared address programming is just like using a bulletin board, where one can communicate with one or many individuals by posting information at a particular location, which is shared by all other individuals. Individual activity is coordinated by noting who is doing what task.

Message passing is like a telephone call or letters where a specific receiver receives information from a specific sender.

Data parallel programming is an organized form of cooperation. Here, several individuals perform an action on separate elements of a data set concurrently and share information globally.

Shared Memory

Shared memory multiprocessors are one of the most important classes of parallel machines. It gives better throughput on multiprogramming workloads and supports parallel programs.

In this case, all the computer systems allow a processor and a set of I/O controller to access a collection of memory modules by some hardware interconnection. The memory capacity is increased

by adding memory modules and I/O capacity is increased by adding devices to I/O controller or by adding additional I/O controller. Processing capacity can be increased by waiting for a faster processor to be available or by adding more processors.

All the resources are organized around a central memory bus. Through the bus access mechanism, any processor can access any physical address in the system. As all the processors are equidistant from all the memory locations, the access time or latency of all the processors is same on a memory location. This is called **symmetric multiprocessor**.

Message-Passing Architecture

Message passing architecture is also an important class of parallel machines. It provides communication among processors as explicit I/O operations. In this case, the communication is combined at the I/O level, instead of the memory system.

In message passing architecture, user communication executed by using operating system or library calls that perform many lower level actions, which includes the actual communication operation. As a result, there is a distance between the programming model and the communication operations at the physical hardware level.

Send and **receive** is the most common user level communication operations in message passing system. Send specifies a local data buffer (which is to be transmitted) and a receiving remote processor. Receive specifies a sending process and a local data buffer in which the transmitted data will be placed. In send operation, an **identifier** or a **tag** is attached to the message and the receiving operation specifies the matching rule like a specific tag from a specific processor or any tag from any processor.

The combination of a send and a matching receive completes a memory-to-memory copy. Each end specifies its local data address and a pair wise synchronization event.

Convergence

Development of the hardware and software has faded the clear boundary between the shared memory and message passing camps. Message passing and a shared address space represents two distinct programming models; each gives a transparent paradigm for sharing, synchronization and communication. However, the basic machine structures have converged towards a common organization.

Data Parallel Processing

Another important class of parallel machine is variously called – processor arrays, data parallel architecture and single-instruction-multiple-data machines. The main feature of the programming model is that operations can be executed in parallel on each element of a large regular data structure (like array or matrix).

Data parallel programming languages are usually enforced by viewing the local address space of a group of processes, one per processor, forming an explicit global space. As all the processors communicate together and there is a global view of all the operations, so either a shared address space or message passing can be used.

Fundamental Design Issues

Development of programming model only cannot increase the efficiency of the computer nor can the development of hardware alone do it. However, development in computer architecture can make the difference in the performance of the computer. We can understand the design problem by focusing on how programs use a machine and which basic technologies are provided.

In this section, we will discuss about the communication abstraction and the basic requirements of the programming model.

Communication Abstraction

Communication abstraction is the main interface between the programming model and the system implementation. It is like the instruction set that provides a platform so that the same program can run correctly on many implementations. Operations at this level must be simple.

Communication abstraction is like a contract between the hardware and software, which allows each other the flexibility to improve without affecting the work.

Programming Model Requirements

A parallel program has one or more threads operating on data. A parallel programming model defines what data the threads can **name**, which **operations** can be performed on the named data, and which order is followed by the operations.

To confirm that the dependencies between the programs are enforced, a parallel program must coordinate the activity of its threads.

REFERENCES;

- 1.EVOLUTION AND CONVERGENCE OF PARALLEL ARCHITECTURES-LECT.PDF.
2. Disruptive technologies; the convergence of new paradigm in architecture-phlippemorel,henriette bier-springer.
- 3.Akl, S. G., The Design and Analysis of Parallel Algorithms, Prentice–Hall, 1989
4. Akl, S. G., Parallel Computation: Models and Methods, Prentice–Hall, 1997.
- 5.Almasi, G. S., and A. Gottlieb, Highly Parallel Computing, Benjamin/Cummings, 2nd ed., 1994.
- 6.Bertsekas, D. P., and J. N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice–Hall, 1989.
- 7.Codenotti, B., and M. Leoncini, Introduction to Parallel Processing, Addison–Wesley, 1993.

8. IEEE Computer, journal published by IEEE Computer Society: has occasional special issues on parallel/distributed processing (February 1982, June 1985, August 1986, June 1987, March 1988, August 1991, February 1992, November 1994, November 1995, December 1996). IEEE Concurrency, formerly IEEE Parallel and Distributed Technology, magazine published by IEEE Computer Society.
9. Cichlow, J. M., Introduction to Distributed and Parallel Computing, Prentice–Hall, 1988
10. DeCegama, A. L., Parallel Processing Architectures and VLSI Hardware, Prentice–Hall, 1989.
- Desrochers, G. R., Principles of Parallel and Multiprocessing, McGraw-Hill, 1987.
- Duato, J., S. Yalamanchili, and L. Ni,
11. Interconnection Networks: An Engineering Approach, IEEE Computer Society Press, 1997. Flynn, M. J., Computer Architecture: Pipelined and Parallel Processor Design, Jones and Bartlett, 1995.
12. Proc. Symp. Frontiers of Massively Parallel Computation, sponsored by IEEE Computer Society and NASA. Held every 1 1/2–2 years since 1986. The 6th FMPC was held in Annapolis, MD, October 27–31, 1996, and the 7th is planned for February 20–25, 1999.
13. Fountain, T. J., Parallel Computing: Principles and Practice, Cambridge University Press, 1994. Hockney, R. W., and C. R. Jesshope, Parallel Computers, Adam Hilger, 1981. Hord, R. 14.M., Parallel Supercomputing in SIMD Architectures, CRC Press, 1990.
15. Hord, R. M., Parallel Supercomputing in MIMD Architectures, CRC Press, 1993.
16. Hwang, K., and F. A. Briggs, Computer Architecture and Parallel Processing, McGraw-Hill, 1984.
17. Hwang, K., Advanced Computer Architecture: Parallelism, Scalability, Programmability, McGrawHill, 1993