ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (151-156), Month: October - December 2014, Available at: www.researchpublish.com

BASICS OF DISTRIBUTED OPERATING SYSTEM

[1]Puneet Bhalla, [2]Piyush Chutani, [3]Rohan Khanna

[1,2,3]Affiliated to Maharishi Dayanand University, Rohtak
[1,2,3]Dronacharya College of Engineering, Gurgaon, India

Abstract: **The Distributed Operating Systems focuses mainly on the concept of "shared memory". The term "shared memory" means each device connected in network has its own private memory. So in distributed system the data stored in distributed manner or say data is not localized at one centre location it is stored in distributed manner. Distributed system consist of set of process managing resources and connections between these processes and mappings of events controlling this distributed operating system into processes managing resources. The distributed operating system is one which looks like an ordinary centralized operating system but works with different independent processing units called central processing unit (CPU).**

Keywords: **Shared Memory, Data Localization, Classification, Scope, Features of Distributed OS.**

I. INTRODUCTION

A distributed operating system is consist of large number of independent, networked and separate nodes(computers) but this whole system is appeared as single system .In distributed operating system(DOS) multiple central processing units(CPU) are connected. To connect these nodes and how they communicate each other various classification scheme are defined. Such classification is called Flynn's classification.

II. HARDWARE CONCEPT

A. Flynn's classification

Flynn classified computers into four architectures on the basis of the number of concurrent instruction (or control) and data streams available in the architecture. The classification is shown in TABLE I:

a. Single Instruction, Single Data Stream(SISD)

A Single Instruction, Single Data Stream architecture is a sequential computer which exploits parallelism in neither the instruction streams nor data streams. This architecture consists of a single control unit (CU) which fetches a single Instruction Stream (IS) from the memory. The Control Unit then generates appropriate signals to direct single processing element (PE) to operate on single Data Stream (DS) indicating one operation at a time. The process is as shown in diagrammatic **Figure 1** below:

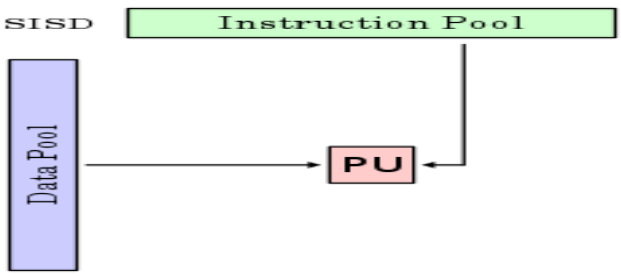


FIGURE 1: SISD ARCHITECTURE

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (151-156), Month: October - December 2014, Available at: www.researchpublish.com

Examples of SISD architecture include the traditional uniprocessor computers like a PC (Single Processing Unit) or the earlier mainframes.

b. Single Instruction, Multiple Data streams(SIMD)

A Single Instruction, Multiple Data stream architecture is a computer architecture which exploits multiple data streams against a single instruction stream to carry operations which may be naturally parallelized.

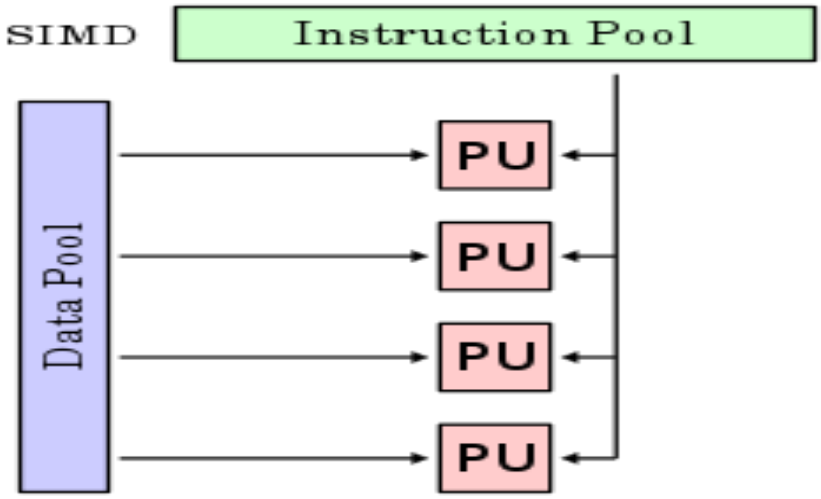


FIGURE 2: SIMD ARCHITECTURE

Examples of SIMD architecture include array processors and GPU's.

c. Multiple Instruction, Single Data stream(MISD)

A Multiple Instruction, Single Data stream architecture is a computer architecture in which multiple instructions operate on a single data stream. It is an uncommon architecture, generally used for fault tolerance. Heterogeneous systems operate on the same data stream and must agree on the result.

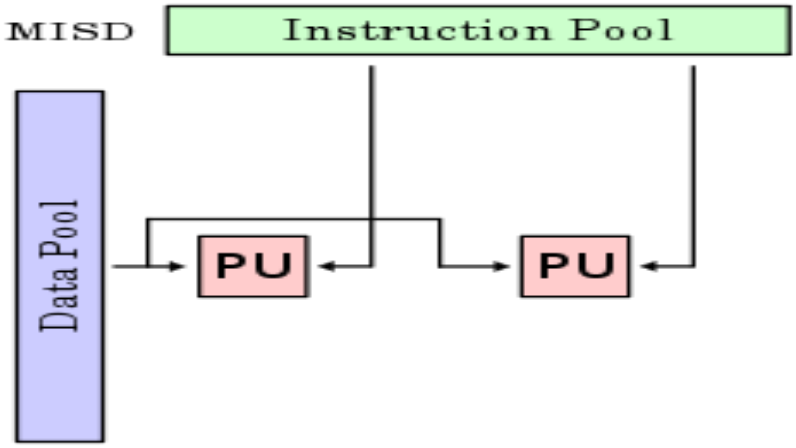


FIGURE 3: MISD ARCHITECTURE

Examples of this category are most probably hypothetical.

d. Multiple Instruction, Multiple Data streams(MIMD)

A Multiple Instruction, Multiple Data stream architecture is a computer architecture in which multiple autonomous processors simultaneously executes different instructions on different data. Distributed systems are generally recognized to be MIMD architectures; either exploiting a single shared memory space or a distributed memory space.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (151-156), Month: October - December 2014, Available at: www.researchpublish.com

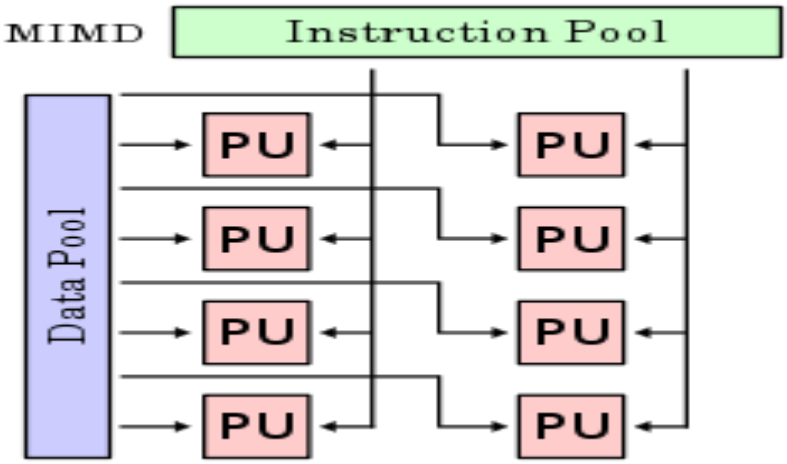


FIGURE 4: MIMD ARCHITECTURE

A multi-core superscalar processor is an excellent example of a MIMD processor.

Flynn's taxonomy (*multiprogramming context*)

	Single instruction	Multiple instruction	*Single program*	*Multiple program*
Single data	SISD	MISD		
Multiple data	SIMD	MIMD	SPMD	MPMD

TABLE I: FLYNN'S CLASSIFICATION OF COMPUTERS

B. Taxonomy of parallel and distributed system

I. ***Multiprocessor (shared memory):*** In this type the single virtual address space is shared by all CPU. It is also called Tightly Coupled system. In this system the delay experience when a message is send from one computer to another computer is short and data rate is high i.e. number bits per second that can be transferred is large. They are generally parallel system (working on single problem). FIGURE 5 depicts the classification of parallel and distributed computers.

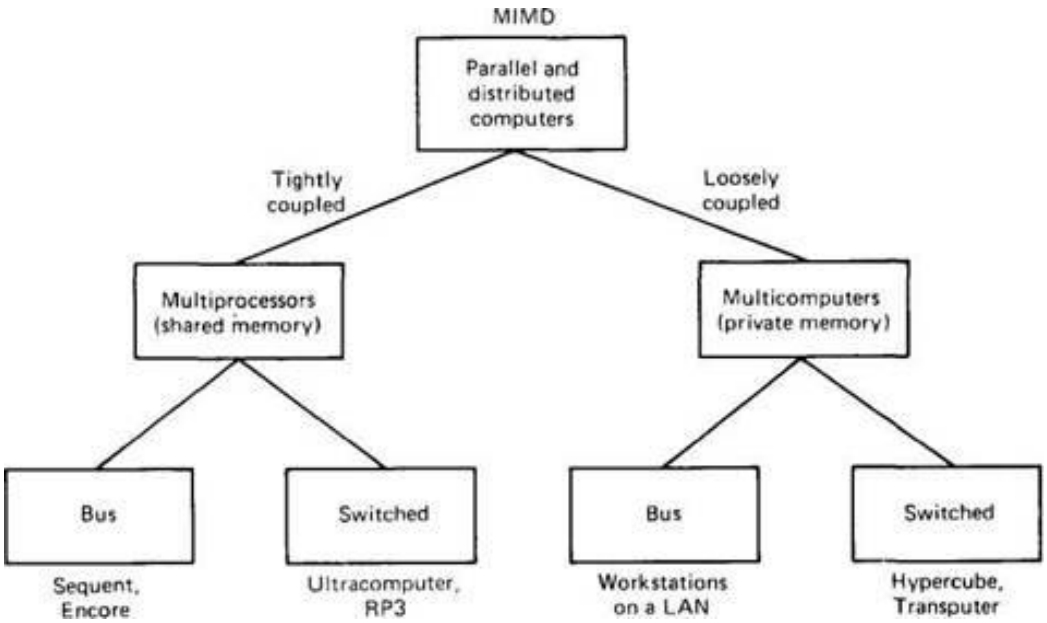


FIGURE 5: CLASSIFICATION OF PARALLEL AND DISTRIBUTED SYSTEMS

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (151-156), Month: October - December 2014, Available at: www.researchpublish.com

B.1. Types of Multiprocessors

a. Bus Based Multiprocessor: A number of CPUs are connected to a common bus with a memory module. The memory module is called cache memory. A simple configuration is to have high speed motherboard on to which CPU and memory card can be inserted. If CPU A writes a word to a memory and B reads it, it will read the just written value. This is called **coherent** property. The cache memory holds the most recently word. There is a problem with two or more cache memory. Super CPU A and B each read the same word into respected the word then A overwrites the word, when B read the same word it get same value from its cache not the value A just wrote it violating the coherent property.

B.1.1. Problem of cache memory is solved using two types of cache memory

- **Write through cache:** The solution is when the word is written on to cache it is written to memory as well such a cache is called Write through cache.
- **Snoopy cache:** When a word is written to cache the new value updated to network and value changes of caches of all computer connected to network.

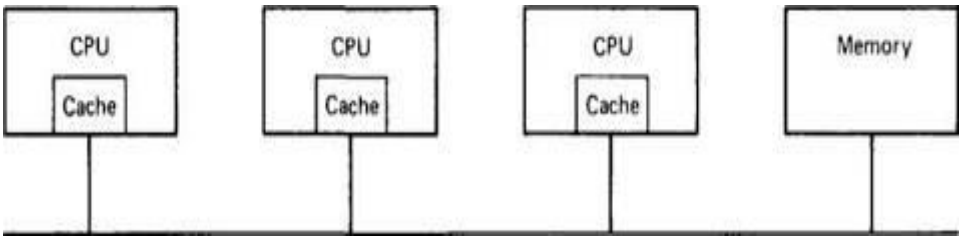
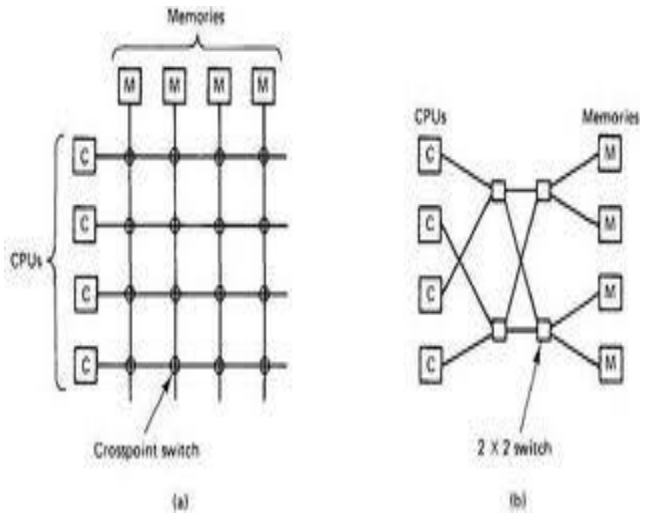


FIGURE 6: BUS BASED MULTIPROCESSOR

b. Switch Based Multi-processor: At every intersection is a tiny electronic cross point switch that can be opened and closed in hardware. In this memory is divided into modules and connect them to CPU's with crossbar switch.



No of CPU = n
Memory = n
Cross point = n^2

FIGURE 7: SWITCH BASED MULTIPROCESSOR

II. Multicomputer (private memory): In this type each computer has its own memory. It is also called loosely coupled system. In this system the delay experience when a message is sent from one computer to another computer is more and data rate is low. It is used as distributed system and work on different (unrelated) problem.

B.2 Types of Multicomputer

a. Bus Based Multi-computer: It is the collection of workstations (computers) on local area network (LAN). Each workstation has its own central processing unit (CPU) and private memory. CPU is connected to its private (local) memory. Each workstation connected to network.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 2, Issue 4, pp: (151-156), Month: October - December 2014, Available at: www.researchpublish.com

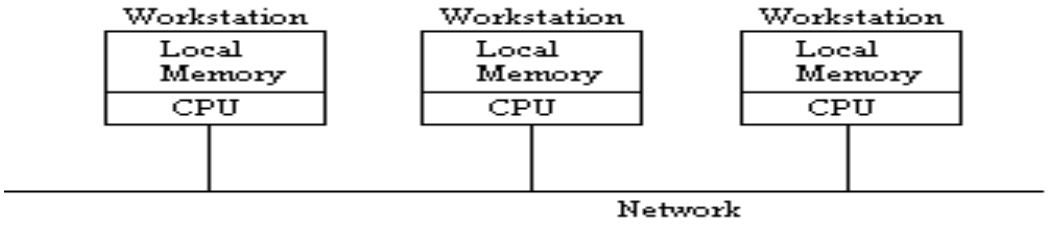


FIGURE 4: BUS BASED MULTICOMPUTER

b. Switch Based Multicomputer: In this type several computers interconnected with holding a property that each computer has directly access to its private memory.

Two types of topology are used

1. **Grid:** It is first topology and easy to understand. Grids are two dimensional (2 D). As from name it specifies that CPU connected in the form of grid. Grids are very easy to understand.

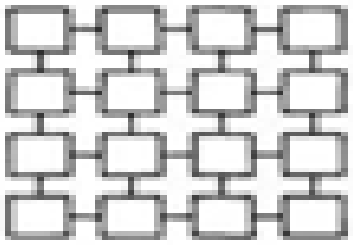


FIGURE 6: GRID

2. **Hypercube:** A hypercube is a n-dimensional cube. It is assumed as two simple cubes with 8 vertices and 12 edges. Here each vertex is CPU and each edge connects vertices (CPU). The edge connects two corresponding vertices of cubes.

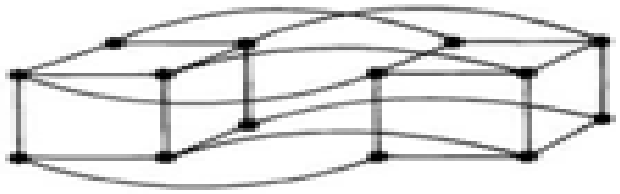


FIGURE 7: HYPERCUBE

III. SOFTWARE CONCEPT

The software concept includes three things that are defined below:

1. **Network operating system:** In network operated system a number of workstation are connected through local area network (LAN). In this the client send the request to file server and file sever examine and execute the request and then appropriate file or data send to client. In network operating system it is easy to add new client (user).

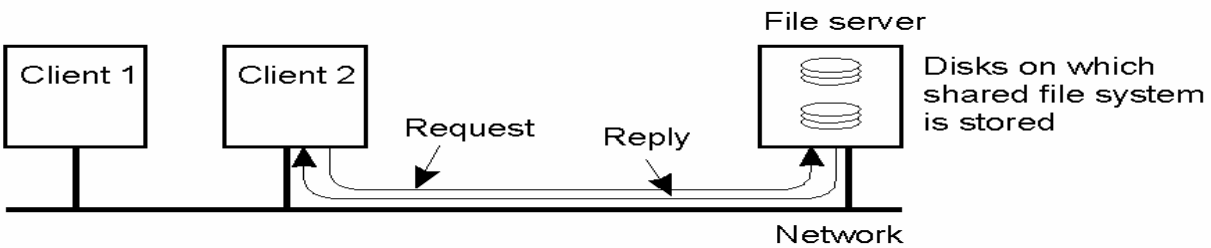


FIGURE 8: A TYPICAL CLIENT SERVER MODEL

2. Multiprocessor time sharing system: In multiprocessor time sharing system operating as UNIX timesharing system with multiple CPU. Each process is allotted equal time for execution.

3. Features of Distributed Systems

1. **Economics** : Microprocessor offers a better price and performance than main frames
2. **Speed:** A distributed system may have more total computing power than main frame.
3. **Inherit distribution:** Some application involves separated machines.
4. **Reliability:** If one server machine crashes the system can still survive.
5. **Incremental Growth:** Computing power can we added in small increments.

IV. CONCLUSION

This paper discusses what a Distributed Operating System means and how it has been classified according to ***Flynn's Classification.*** It focusses on the ***Hardware Concept of Distributed Systems*** followed by the ***Software Concepts.*** This paper discusses in brief the ***Taxonomy of Parallel and Distributed Systems,*** and concludes with the ***features and a few Advantages of Distributed Systems.***

- ***Advantages of Distributed System***

1. **Data Sharing:** Allow many users to access a common database. Data can be shared by all the users connected in network.
2. **Device Sharing:** Sharing of peripheral devices such as laser printers storage devices.
3. **Communication:** To enhance person to person communication by using e-mail.
4. **Flexibility:** To have the mixture of personal as well as shared computers perhaps of different sizes and let the job run on more appropriate system.

REFERENCES

[1] http://www2.cs.siu.edu/~cs401/Textbook/ch6.pdf
[2] http://ro.uow.edu.au/cgi/viewcontent.cgi?article=1035&context=compsciwp
[3] http://www.cs.helsinki.fi/u/jakangas/Teaching/DistSys/DistSys-08f-1.pdf
[4] https:/www.cs.columbia.edu/~smb/classes/s06-4118/l26.pdf