

CCS525 NOTEBOOK

liyu0x

2021.03-2021.07

目录

1 Distributed System	1
1.1 Construct And Middleware	1
1.2 why use DS	2
1.3 DS Goals	2
1.4 Transparency	3
1.5 Service Models	3
1.6 Architectural Models	3
1.7 Hardware Architecture	4
1.7.1 Homogenous Multicomputer System	6

1 Distributed System

Definition A A distributed system is one in which components located at networked computers communicate and coordinate their actions only by message-passing.(CDK)

Definition B A collection of independent computers that appears to its users as a single coherent system.(Tanenbaum and van Steen)

1.1 Construct And Middleware

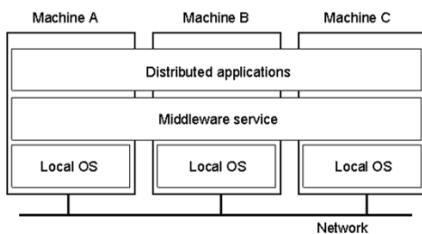


图 1: Distributed System Construct

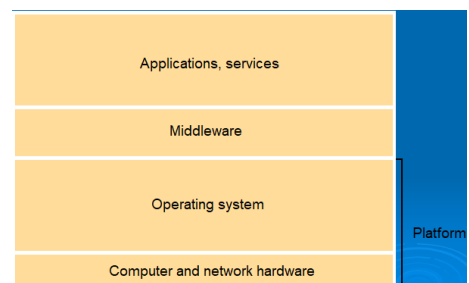


图 2: Distributed System Construct

Middleware A distributed system organized as middleware. The middleware layer extends over multiple machines. In an open middleware-based distributed system, the protocols used by each middleware layer should be the same, as well as the interfaces they offer to applications

1.2 why use DS

- Ever increasing need to SHARE resources and information. 对资源和信息分享的需求日益增长
- Rapid decreasing cost of powerful workstations. 大型工作站需要降低成本
- Widespread use of networks. 网络的广泛使用
- Maturity of software technology. 软件技术的成熟

1.3 DS Goals

- Efficiency
 - communication 通讯效率
 - load distribution 压力分布
- Flexibility
 - Users Viewpoint
 - * ease of use 便于使用
 - * consistency 一致性
 - * reliability 可靠性
 - System viewpoint
 - * modularity 模块化
 - * scalability 可扩展性
 - scale up: 垂直扩展, 增加 CPU 或存储
 - scale out: 水平扩展, 添加机器
 - * portability 可移植性
 - * interoperability 互操作性
- Robustness
 - Can handle exceptional situations and errors
- Consistency
 - A system is consistent if there is uniformity in using the system and the system behaviour is predictable. (eg. raft state machine)
 - Integrity of the system must be maintained with proper concurrency control.
 - Inconsistency can be due to lack of global info, potential replication and partitioning of data, possibility of component failures or complexity of interaction among modules. 不一致的原因

1.4 Transparency

There are have 8 properties.

Properties	Description
Access	Hide different in data representation and how a resource is accessed
Location	Hide where a resource is locate
Migration	Hide that a resource may move to another location
Relocation	Hide that a resource may be moved to another location while in use
Replication	Hide that existence of replicas and replica management from user
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

表 1: Transparency Table

1.5 Service Models

There are 3 types of Service,

Primitive Services Fundamental functions that the kernel must provide.3 minimum functions includes:

- communication
- Synchronization
- Process serving: manages the creation, deletion and tracking of processes.

url: <https://www.geeksforgeeks.org/service-primitives/>

System Servers Provide fundamental services for managing processes, file and process communication.Example:name server(DNS), directory server(ldap), network server, time server.

Value-added Services Services that are not essential in the implementation of a distributed system but are useful in supporting distributed applications.Example: group server to manage creation and termination of group activities.

1.6 Architectural Models

C-S C-S, Client-Server architecture, including:

- Clients invoke individual servers
- A service provided by multiple servers
- use Web proxy server
- peer to peer

Mobile Agents Variations on client-server model, send the applet code to client. Reason for using:

- Reduce network load and client processing power
- Overcome network latency
- Encapsulate protocols
- Execute asynchronously & autonomously
- Adapt dynamically
- Naturally heterogeneous
- Robust and fault-tolerant

Drawback:

- A computer virus is some kind of MA
- MA may be attacked/modified/deleted by hostile host

1.7 Hardware Architecture

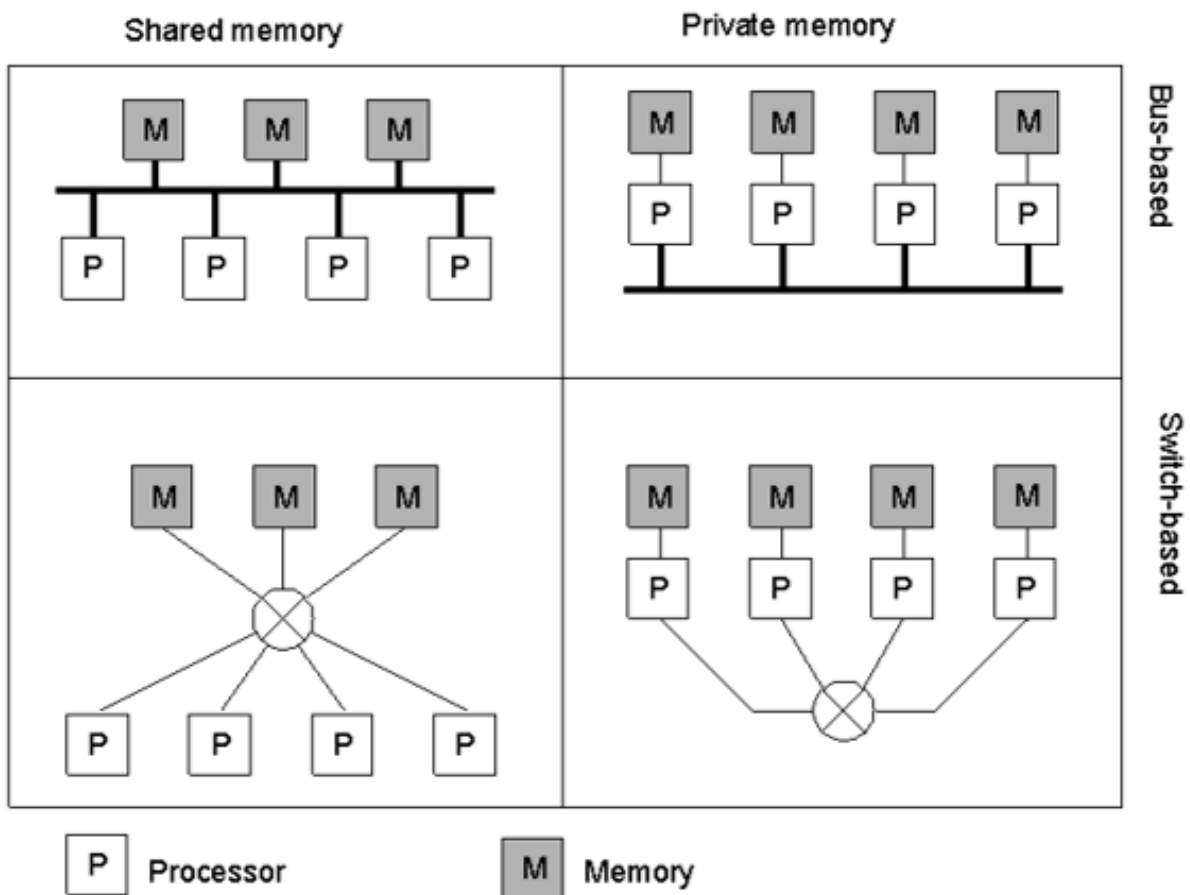


图 3: Different basic organizations and memories in distributed computer systems.

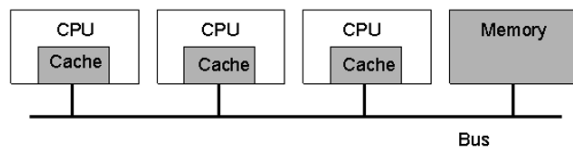


图 4: A bus-based multiprocessor

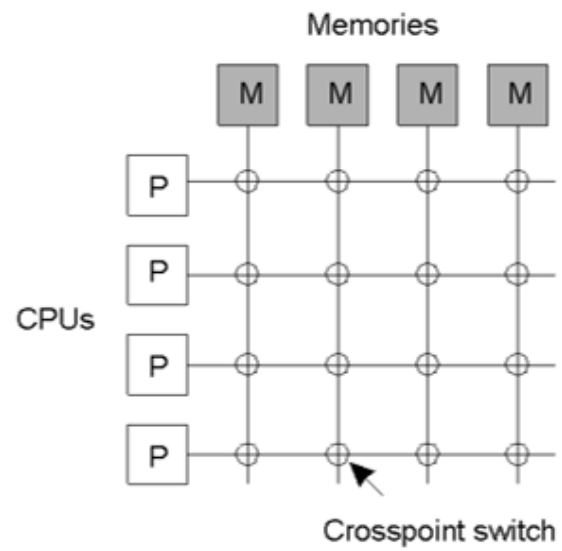


图 5: Crossbar Switch

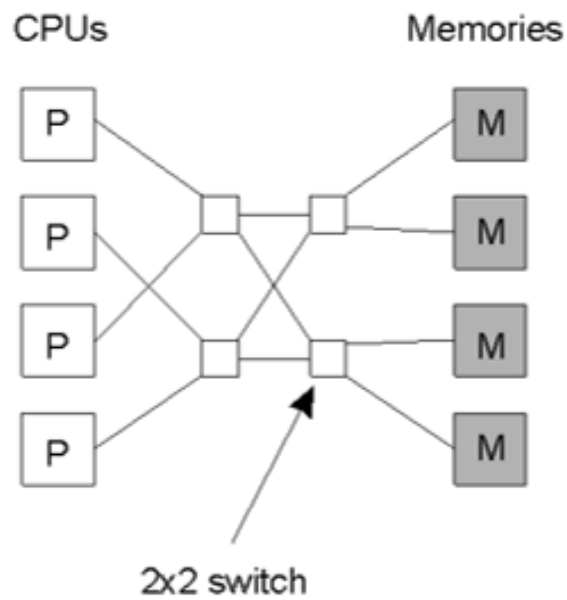


图 6: Omega Switching Network

1.7.1 Homogenous Multicomputer System

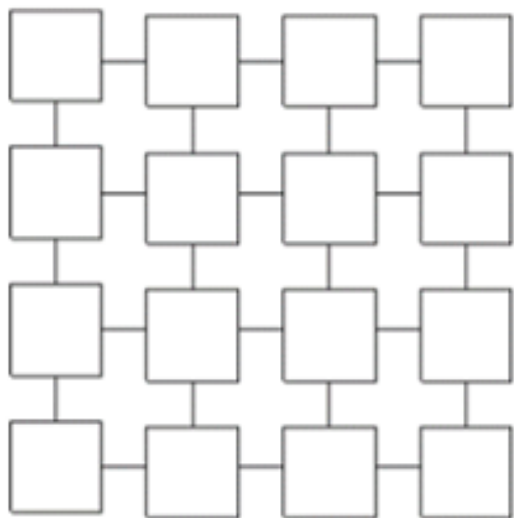


图 7: Grid

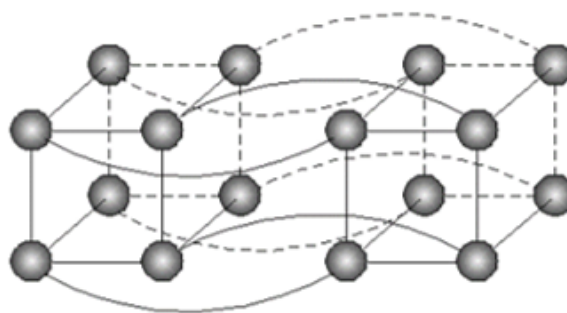


图 8: Hypercube