**Distributed computation**

Distributed computing is a field of computer science that studies distributed systems. A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages.

Message passing sends a message to a process (which may be an actor or object) and relies on the process and the supporting infrastructure to select and invoke the actual code to run. Message passing differs from conventional programming where a process, subroutine, or function is directly invoked by name. Message passing is key to some models of concurrency and object-oriented programming.

Message passing is used anywhere in modern computer software. It is used as a way for the objects that make up a program to work with each other and as a way for objects and systems running on different computers (e.g., the Internet) to interact. Message passing may be implemented by various mechanisms, including channels.

The components interact with each other in order to achieve a common goal. Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and independent failure of components. Examples of distributed systems vary from SOA-based systems to massively multiplayer online games to peer-to-peer applications.

A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs. There are many alternatives for the message passing mechanism, including RPC-like connectors and message queues. An important goal and challenge of distributed systems is location transparency.

Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.

The word distributed in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing. While there is no single definition of a distributed system, the following defining properties are commonly used:

• There are several autonomous computational entities, each of which has its own local memory.

• The entities communicate with each other by message passing.

In this article, the computational entities are called computers or nodes.

A distributed system may have a common goal, such as solving a large computational problem. Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users.

Other typical properties of distributed systems include the following:

• The system has to tolerate failures in individual computers.

• The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program.

• Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input.

Client/Server System : The Client-server architecture is a way to provide a service from a central source. There is a single server that provides a service, and many clients that communicate with the server to consume its products. In this architecture, clients and servers have different jobs. The server's job is to respond to service requests from clients, while a client's job is to use the data provided in response in order to perform some tasks.

Peer-to-Peer System : The term peer-to-peer is used to describe distributed systems in which labor is divided among all the components of the system. All the computers send and receive data, and they all contribute some processing power and memory. As a distributed system increases in size, its capacity of computational resources increases. In a peer-to-peer system, all components of the system contribute some processing power and memory to a distributed computation.

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

• In parallel computing, all processors may have access to a shared memory to exchange information between processors.

• In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

**Glossary**

|  |  |  |
| --- | --- | --- |
| 1 | distributed computing | розподілені обчислення |
| 2 | distributed system | розподілені система |
| 3 | distributed programming | програмування розподілених систем |
| 4 | distributed algorithm | розподілений алгоритм |
| 5 | message passing | передача повідомлень |
| 6 | invoke | викликати |
| 7 | subroutine | підпрограма |
| 8 | function | функція |
| 9 | directly invoke | безпосередній виклик |
| 10 | models of concurrency | моделі паралелізму |
| 11 | object-oriented programming | об'єктно-орієнтоване програмування |
| 12 | channel | канал |
| 13 | SOA-based | Сервісно-орієнтована архітектура |
| 14 | peer-to-peer | “рівний до рівного” мережа рівноправних вузлів |
| 15 | RPC | Виклик віддалених процедур |
| 16 | message queue | «черга повідомлень» - відкладений обмін повідомленнями |
| 17 | location transparency | «прозорість розташування» - логічні імена для ідентифікації ресурсів у мережі, які визначаються також за місцем розташування |
| 18 | process | процес - контейнер системних ресурсів, призначених для підтримки виконання програми |
| 19 | local memory | локальна пам'ять |
| 20 | node | вузол (мережі) |
| 21 | tolerate failures | Відмовостійкість – властивість системи функціонувати при наявності помилок |
| 22 | Client/Server System | Клієнт-серверна система |
| 23 | concurrent computing | “конкурентне обчислення» - несинхронізоване виконання потоків |
| 24 | parallel computing | паралельні обчислення |
| 25 | shared memory | пам'ять спільного використання |
| 26 | distributed memory | розподілена пам'ять |