

第一章 分布式系统介绍

1.1 初识分布式系统

1.1.1 分布式系统的定义

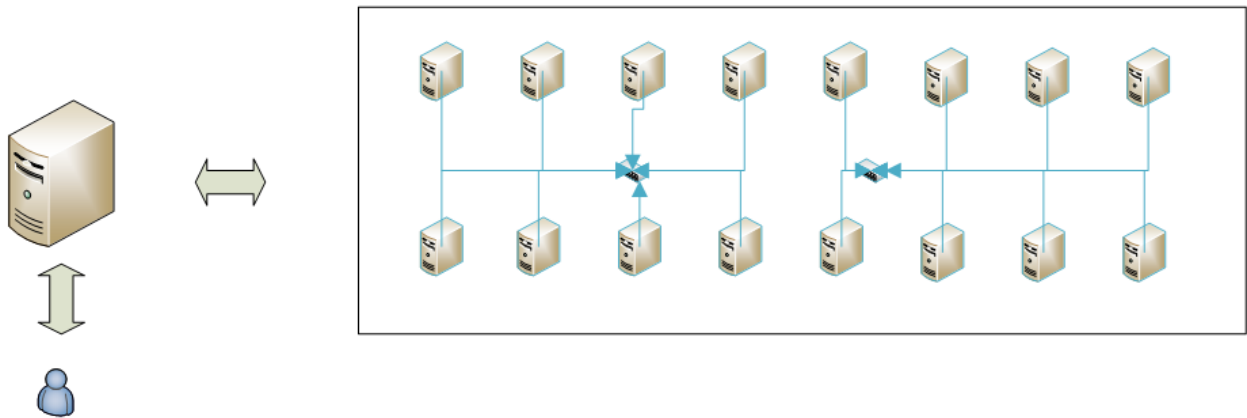
引用Distributed Systems Concepts and Design(Third Edition)中的定义：

A distributed system one in which components located at networked computers communicate and coordinate their actions only by passing messages.

这句话指明：

1. 组件分布在网络计算机上；
2. 组件之间仅仅通过消息传递来通信并协调行动。

分布式系统示意图如下：



由上图我们可以看出，对于用户来说，其只面对单一服务器，而背后的集群对于用户来说是完全透明的。分布式系统相当于一个超级计算机一样。

对于分布式系统的定义，有如下理解：

- 分布式系统一定是由多个节点组成的系统，一般来说一个节点就是一台计算机。
- 各个节点之间不是孤立的，而是相互连通的。
- 节点上部署了我们的组件，相互之间的操作有协同。

1.1.2 分布式系统的意义

分布式系统的意义：

1. 升级单机处理能力的性价比越来越低。
2. 单机处理能力存在瓶颈。
3. 出于稳定性和可用性考虑。

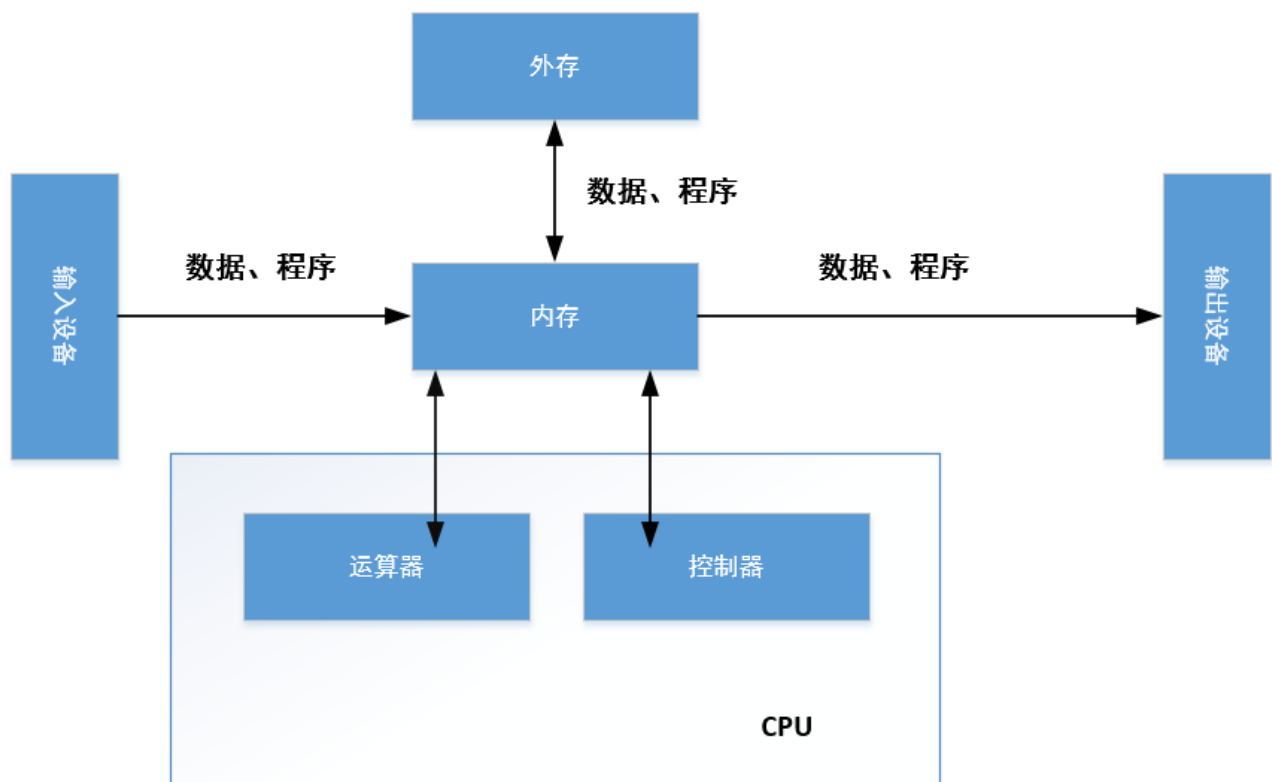
原因：

1. 摩尔定律告诉我们：当价格不变时，每隔18个月，集成电路上可容纳的晶体管数据会增加一倍，性能也将提升一倍。这意味着，随着时间的推移，单位成本的支出所能购买的计算能力在提升，但是在同一时间内，购买的处理器性能越高，付出的成本越高，性价比越低。
2. 同样固定时间点，单颗处理器有自己的性能瓶颈。
3. 单机系统一旦出现问题，系统就不可用了，而分布式系统可以解决这个问题。

1.2 分布式系统的基础知识

1.2.1 组成计算机的5要素

组成计算机的基本元素包括输入设备、输出设备、运算器、控制器、存储器。如下图：



1.2.2 线程与进程的执行模式

1.2.2.1 阿姆达尔定律

多线程程序不容易写，但是其带来的好处却是显而易见的。单核时代，程序随着CPU的更换而变快，而多核年代，程序的并发和并行就显得很重要。通过阿姆达尔定律可以很好的看到程序中的并行部分对于增加CPU核心来提升速度存在限制。

$$S(N) = \frac{1}{(1 - P) + \frac{P}{N}}$$

其中

- P指的是程序中可并行部分的程序在单核上执行时间的占比。
- N表示处理器的个数（总核心数）。
- S(N)是指程序在N个处理器（总核心数）相对在单个处理器（单核）中的速度提升比。

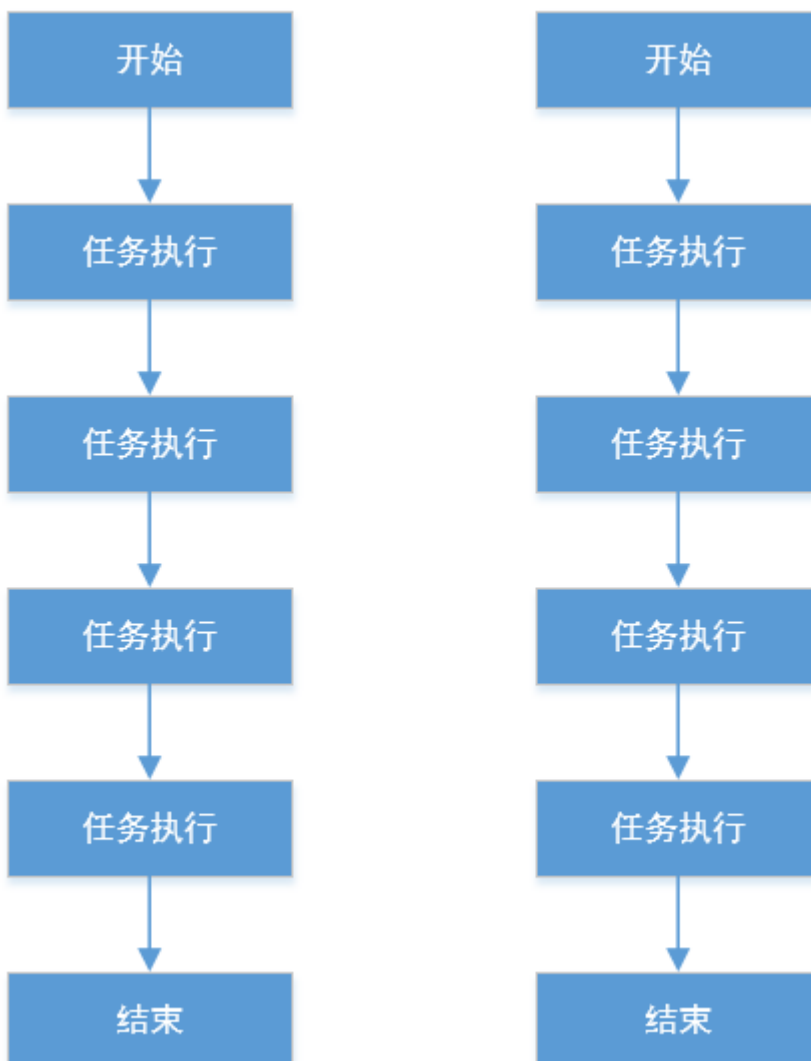
阿姆达尔定律变形得

$$S(N) = \frac{1}{1 - \frac{(N-1) * P}{N}}$$

当N趋近于无穷时，分母趋近于1-P，此时获得S(N)的最大值。因此，也就是说，程序中的可并行代码的比例决定你增加处理器（总核心数）所能带来的速度提升上限。

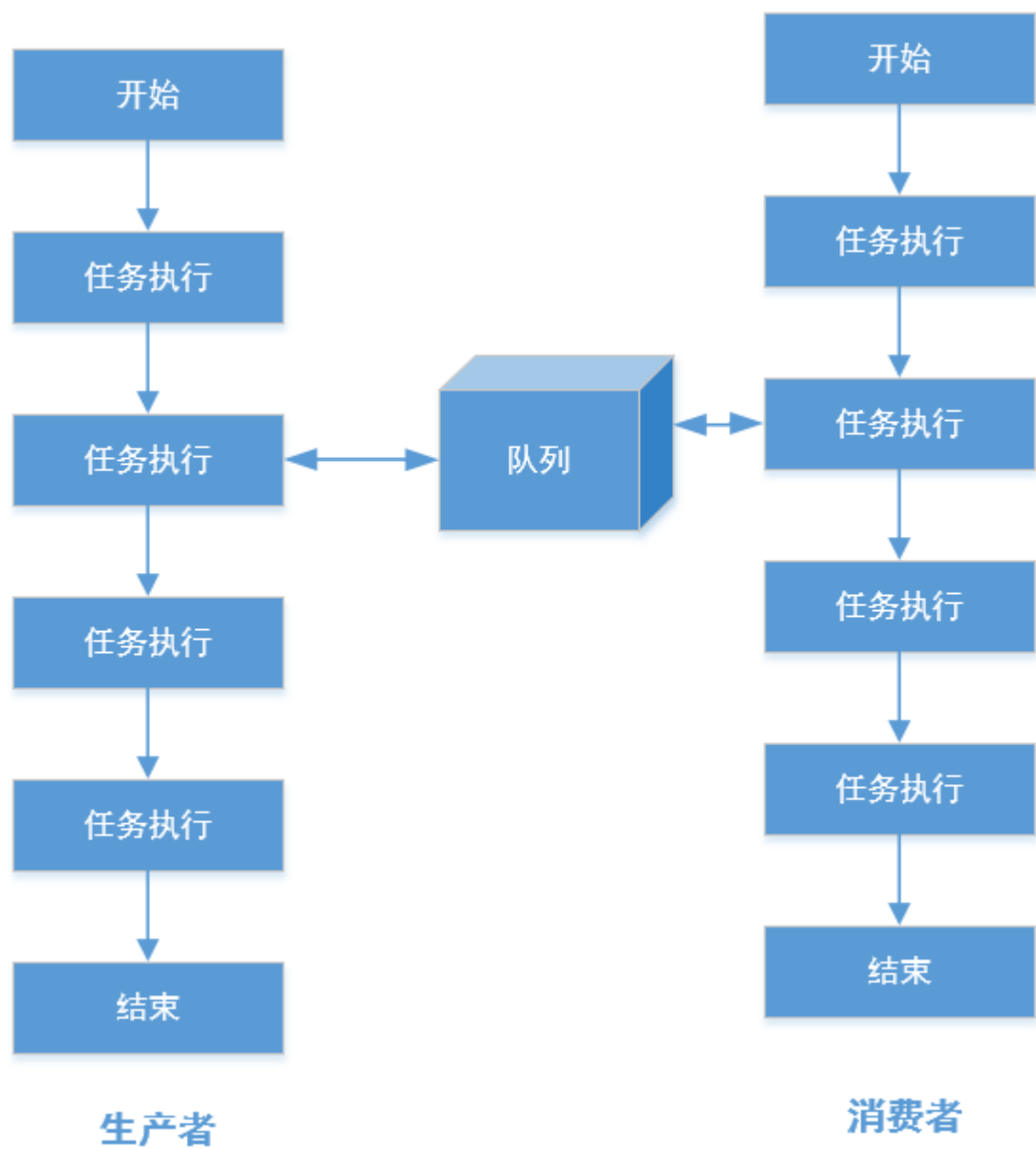
1.2.2.2 互不通信的多线程模式

在多线程程序中，多个线程会在系统中并发执行。如果线程之间不需要处理共享数据，也不需要进行动作协调，那么将会按照如下模式进行。



1.2.2.3 基于共享容器协同的多线程模式

一些场景中我们需要多个线程之间对共享的数据进行处理。例如经典的生产者和消费者的例子，如下图



对于这种在多线程环境下对同一数据的访问，我们需要保证访问的正确性。对于存储数据的容器或者对象，有线程安全和不安全之分，而对于线程不安全的容器或对象，一般可以通过加锁或者通过Copy On Write的方式控制并发访问。使用加锁方式时，如果数据在多线程中的读写比例很高，则一般采用读写锁而非简单的互斥锁。

1.2.2.4 通过事件协同的多线程模式

即所谓的同步互斥。

1.2.2.5 多进程模式

关于多进程与单进程多线程的异同，请查看操作系统相关知识。

单线程和单线程多进程的程序在遇到机器故障、OS问题或者自身进程问题时，会导致整个功能不可用。

对于多进程的系统，如果遇到机器故障或者OS问题，也会导致整个功能不可用，但是如果只是一个进程有问题，那么有可能保持系统的部分功能正常执行。

多级系统同，如果遇到某些机器故障、OS问题或者某些机器的进程问题，那么可能部分功能失效，但是有机会保证整体功能可用。

1.2.3 网络通信基础知识

1.2.3.1 OSI与TCP/IP网络模型

详见计算机网络知识。

1.2.3.2 网络IO实现方式

BIO、NIO、AIO

1.BIO方式

BIO即Blocking IO，采用阻塞的方式实现。一个Socket套接字需要使用一个线程来进行处理。发生建立连接、读数据、写数据的操作时，都可能发生阻塞。

优点：简单

缺点：一个线程只能处理一个Socket。

工作方式如下图：