

1.分布式系统概述

学生：张帅豪 18030100101

老师：李龙海

分布式系统定义

一个分布式系统由多个通过网络互联的独立自治的计算节点组成，这些计算节点基于消息传递机制进行相互协作，以完成共同的目标。

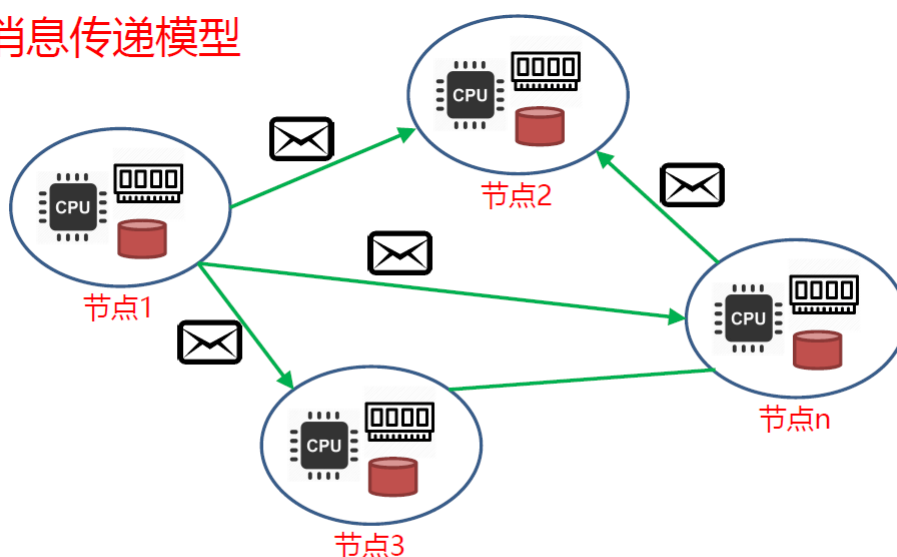
从普通用户角度看，分布式系统是计算节点内聚在一起的一个整体，用户在使用系统功能的时候，往往无法察觉分布式系统的内部构成和节点之间的协作关系。

1. 多个计算节点：计算节点一般指单个计算机，也可以是计算机中的一个进程、线程或虚拟机
2. 网络互联：节点之间可以通过有线、无线等任意网络通信方式互联。对物理拓扑结构不做明确限定。
3. 独立自治：每个节点都有自己独立的CPU、独立的时钟，发生错误的时机和模式也相互独立。并发：不同节点的动作是同时进行的。
4. 消息传递模型：并非内存共享模型

分布式计算

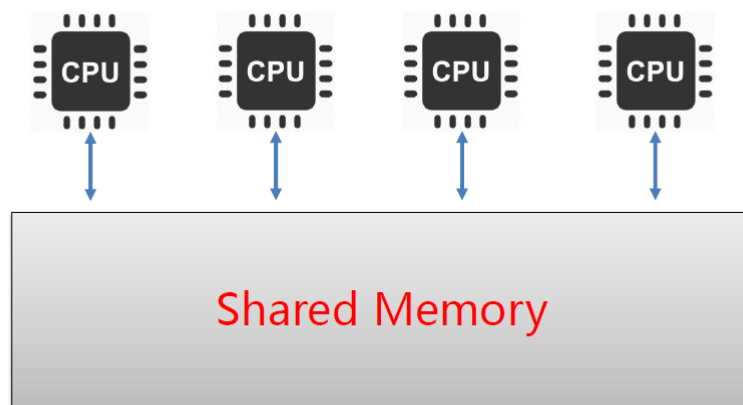
多个通过网络互联的计算节点通过相互协作共同完成计算任务

■ 消息传递模型



1. 节点之间没有公共状态，必须通过相互发送消息来进行协作。
2. 内部运算速度比消息传递速度高几个数量级。通信复杂度是影响效率的重要因素。
3. 系统设计时必须应对局部失效、消息延迟/丢失等错误。

■ 共享内存模型



1. 不同节点之间共享公共状态；
2. 节点之间通过写-读共享存储器中的公共状态来隐式地进行通信。

不同层次的并行计算		提高计算能力
<ul style="list-style-type: none"> □指令级并行：多指令并行；单指多数并行（向量指令） □CPU多核并行：多线程编程 □多CPU并行（一致性内存访问）：多线程编程 □多CPU并行（非致性内存访问）：超级计算机 □基于GPU的并行：单指多数并行；CUDA、OpenCL □多机并行：基于消息传递的分布式计算（share nothing） 	====>	<ul style="list-style-type: none"> □提高存储能力 □提高网络吞吐能力（并发访问能力） □提高可靠性（解决局部失效问题） □提高安全性（解决被局部攻击问题） □提高可扩展性（解决瓶颈问题） □实现资源共享 □实现跨越时空的协同服务（发挥不同节点的优势）

例子



设计挑战举例：节点染色问题

Algorithm P3C: 3-colouring a path.

Repeat forever: c代表节点当前的染色值，初始值为节点ID

- Send message c to all neighbours.
- Receive messages from all neighbours.
Let M be the set of messages received.
- If $c \notin \{1, 2, 3\}$ and $c > \max M$:
Let $c \leftarrow \min(\{1, 2, 3\} \setminus M)$.

现状

- 实现资源共享的分布式系统（分布式存储）
 - Web系统、DNS系统
 - 网络文件系统：NFS、HDFS
 - P2P资源共享系统：BitTorrent、 μ Torrent、eMule
 - 区块链、比特币
- 高性能计算系统：Map-Reduce、Spark、TensorFlow
- 云计算
- 网格计算
- 集群计算
- 分布式信息系统：跨企业应用系统、金融应用系统
- 无处不在计算
- 现实生活中分布式系统已经得到了广泛应用
- 近年来的热点技术背后都以分布式系统作为后盾
 - 云计算（边缘计算、雾计算）
 - 物联网
 - 大数据（分布式采集、存储、处理）
 - 人工智能（高性能分布式计算、GPU）
 - 区块链
 - 可能没听过的一些热点：SOA、微服务、P2P、网格计算、无处不在计算（Ubiquitous Computing）
 - 大型Web系统：HW/BAT的后台是什么？
- 计算机科学发展到今天，分布式计算应该成为和操作系统同等重要的专业基础课程。
- 分布式计算领域还大有可为

- 现实生活中分布式系统已经得到了广泛应用
- 近年来的热点技术背后都以分布式系统作为后盾
 - 云计算（边缘计算、雾计算）
 - 物联网
 - 大数据（分布式采集、存储、处理）
 - 人工智能（高性能分布式计算、GPU）
 - 区块链
 - 可能没听过的一些热点：SOA、微服务、P2P、网格计算、无处不在计算（Ubiquitous Computing）
 - 大型Web系统：HW/BAT的后台是什么？
- 计算机科学发展到今天，分布式计算应该成为和操作系统同等重要的专业基础课程。
- 分布式计算领域还大有可为