说明

目前，计算机系统在迅猛发展。在1945 – 1985年，现代计算机时代刚刚开始，那时的计算机都是非常大而且昂贵的。除此之外，由于计算机之间连接的缺乏，使得每个计算机只能独立的运行。

80年代中期开始，两个技术改变了这一现状。第一个就是微处理器。期初，只有8位的处理器，发展到后来有 16位，32位，以及64位等。我们可以通过多核处理器来编写并行程序。现在我们的处理器悠着30-40年前大型机的处理能力，然而造价却只有之前的千分之一。

第二项技术是高速的计算机网络的发明。Local-area networks 或 LANS（局域网）可以使得一栋大楼内的成百上千的计算机在毫秒量级进行通信。并且很大的数据可以以每秒数十亿比特的速度进行传输（千兆网络）。Wide-area networks 或 WANS (广域网）可以让地球上数百万计算机以 数十K到数百兆的速度进行传播。

同样的，计算机系统的迷你化也是我们关注的焦点，手机就是最好的例子。用传感器，内存和cpu包装的这些设备已经和完备的计算机相差无异。当然，他们也有网络通信的能力。 沿着这条发展线路，那些插入式设备同样找到了市场。那些像电源适配器一样大小的设备，可以通过直接插入的方式，并产生和near-desktop 相近的性能。

这些成果不但证明这些技术可行，并能够容易构成一个由很多计算机组成的计算机网络系统。由于这些计算机是在地理层面分散的，故我们将此系统称为分布式系统。这个系统可以包含几个或者数百万台计算机。这些计算机的连接可以是通过有线的，无线的，以及混合这两种。由于分布式系统总会有新的计算机加入，或者老的计算机退出，所以网络的拓扑和性能是一直在改变的，所以我们称之为动态的系统。

在这一章，我们将通过几个知名的系统 对分布式系统进行初探并了解其设计目标。

## 分布式系统是个什么鬼东西

分布式系统的定义出现在很多的文献中，然而他们的定义都不那么尽如人意。我们这里给出一个松散的定义：

*分布式系统是由一些各自独立的计算机组成的集合，对使用者而言，它确是一个单一的系统。*

这个定义参考了分布式系统的两个特性。一个是分布式系统的计算机是能够彼此运行。这些独立的计算机我们称之为节点，可以是一个硬件，也可以是一个软件。第二个特性是从他的使用者（人 或者 应用）来说明的。这些独立的计算机必须要协作，如何构建这种协作关系是分布式系统的核心内容。这里强调下，我们并没有对节点（独立的计算机）的类型做任何假设。原则上，从高性能主机到小的传感器设备都是可以的。同样的，我们也没对节点之间的连接方式做任何假设。

特性1：独立计算机组成的集合

大部分现代操作系统是由许多高性能的计算机或者是可插拔式的(plug)计算机或者是一些小的设备组成的。一个基本的原则是这些计算机可以独立运行，但是如果他们只是独立运行，这个系统就不能称之为分布式系统了。实际上，这些节点彼此之间可以交换信息，并为了共同的目标而存在的。一个节点收到传入的信息后，反应并加工之后，将导致更深远层次的消息传递。

一个重要的现象是，每个独立的计算机都有他自己的一套时间。或者说，我们不能假设这里有一个统一的时钟概念。缺乏统一的全局时钟会导致基本的同步和协调问题，关于这一点，我们将在第六章阐述。事实上，我们必须要管理好这个这些节点的关系和关联。说白了，我们需要知道哪些节点在集群，哪些已经排除了，或者哪些节点可以直接相连等等。

管理节点集群的关系是很困难的，即使只涉及权限控制。为了方便阐述，我们先搞清楚开放集群和关闭集群。在开放集群中，每个节点都是可以加入到集群的，这表明所有节点都可以互相通信。相反是闭合集群，只有集群内的节点可以互相通信，并有另外一套机制控制其他节点的加入或离开。

很明显的，这种权限控制很困难。这种处理节点是否可以进入集群的机制很重要，如果设计不得当，将成为集群可扩展性的瓶颈，这点我们将在第九章阐述。第二点，每个节点都必须确认自己和其他的正确的集群，正确的节点进行通信，而不是那些恶意的搞破坏的节点。最后，由于集群成员可以和非集群成员通信，这里将会面临信任问题（是否相信那些非成员节点）。

考虑到集群的结构，经验表明，分布式集群很大程度都是一个覆盖网络（overlay network）。每个节点都是由一个进程组成，这个进行有其他的进程的列表，可以向其他进程发送小心。通常来讲，与节点最近的邻居节点是最先被通信的。小心是通过 TCP/UDP 协议发送的，高级别的设备也同样适用，这点我们将在第四章阐述。下面介绍两种 覆盖网络。

结构化覆盖网络： 每个节点都会有一套设定好的邻居节点用来通信

非结构化覆盖网络： 每个节点都随机选择其他节点用来通信

无论怎样，覆盖网络应该是连通的，即任意两个节点都可以通过某个路径进行通信。一个众所周知的覆盖网络是 P2P (peer-to-peer)网络。覆盖网络的例子我们将在第二章以及后续章节进行阐述。通过以上我们应该能认识到，集群的组织构建是需要下很大功夫的，同样也是分布式网络管理很重要的组成部分。

特性2：单一的系统

正如之前提到的，一个分布式系统，应该是一个单一的系统。一些情况下，研究者表示应该有一个单一系统的视角，即终端用户感知不到他们正在处理的数据，程序是分散在不同区域的。从某些方面来说，单一系统的视角的要求太高了，在我们定义的分布式系统里，我们称之为连贯的。粗略的讲，分布式系统的连贯性表示的是它表现的和使用者的预期一致。更特殊的来讲，单一系统的连贯性表示，集群中的节点作为一个整体，无论在什么时间，地点以及用户和系统是如何交互的情况下，都表现的非常一致。

提供单一连贯性视角是非常有挑战性的。例如，它要求终端用户应当无法分辨程序运行在哪台计算机上，或者无法感知到这个任务的一部分已经派发到其他地方的另外一个进程去执行。同样的，用户不用关心数据会存储到哪里，以及复制一些数据用来提升性能。这就叫做分布式系统的透明性，这是一个重要的分布式系统的设计目标，我们将在1.2节进行讨论。从某种意义来说，这类似于类unix系统中的文件系统的操作，它屏蔽了文件，设备，主存以及网络的不同。

然而，争取一个连贯性系统需要一个重要的权衡。由于我们不能忽略分布式系统有许多网络节点，这有可能导致集群的一部分有可能会失效。这表示有些应用将会执行成功，而有些则会失败，这种情况是需要处理的。部分失败在任何一个复杂系统中都是固有的，在分布式系统中更是难以掩藏的。就像某个图灵奖获得者说的“你将不会知道是计算机哪个部分的故障导致你计算机的故障”。

## 中间件和分布式系统

为了帮助分布式系统的发展，分布式系统通常会构建一个分离的软件层，这个软件层放置在各自计算机的操作系统的上层，并属于分布式系统的一部分。如下图1.1

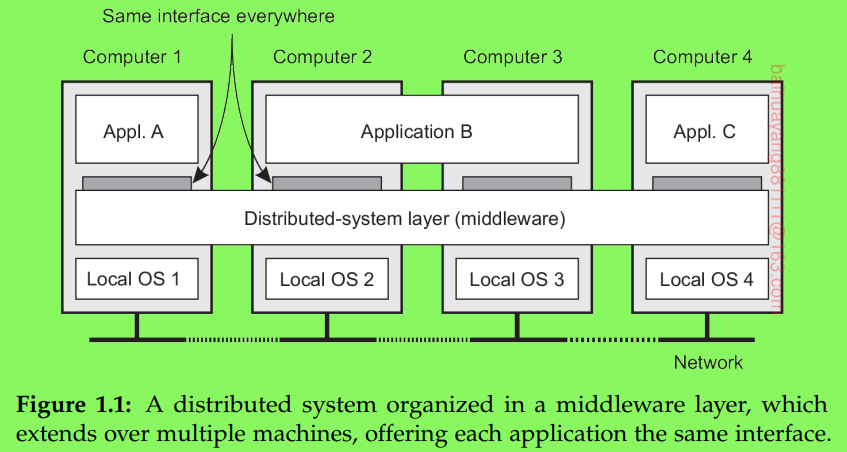


图1.1 中有四个网络计算机和三个应用，B应用分布在计算机2和3上。每个应用都提供相同的接口。分布式系统为单个分布式应用的不同组件提供了方法，也为不同的应用提供了通信的手段。同时，它还尽可能的隐藏了不同硬件和操作系统的不同。

某种意义来说，中间件对于分布式系统的意义，就相当于操作系统对于计算机的意义。即一个在网络中分配、提供和管理资源的一个组件。仅次于资源管理，它能提供类似操作系统的服务：

内部通信的设备

安全服务

Accounting services

掩饰并恢复错误

他们唯一的不同在于，中间件是存在于网络环境中的。事实表明，大部分服务是都是应用有所帮助的。基于这种原因，中间件可以当做是一个常用组件和函数的容器，已经不需要由应用程序单独实现。为了证明这些观点，让我们短暂的考虑几个典型中间件服务的例子。

**通信**：一个大众化的通信服务被称作RPC（远程过程调用）。一个RPC服务可以使得应用在远程计算机执行一个方法，但看起来就好像在本地调用一样。开发者无需在特殊的编程语言中特殊化函数的写法，RPC子系统会生成必要的代码来执行远程调用。

**事务**：许多应用使用分布在很多计算机上的各种各样的服务。中间件通常会利用原子化的事务操作，对这些服务的调用提供特殊的支持。在这种情况下，应用开发者只需要保证远程服务被调用，并且遵循了某种协议，剩下的中间件会保证服务被调用或者没有被调用。

**服务组成**：现在有一种趋势，将已经存在的程序组合而成为新的应用。这在Web应用中体现很广泛。基于Web的中间件可以帮助Web服务提供的服务调用标准化，是方法调用在一个特殊的顺序。一个关于Web服务混搭的例子是：Web 组合并聚合不同来源的数据。有名的混搭是哪些基于谷歌地图的旅行计划应用和天气预报应用。

**可靠性**：最后一个例子，这有一个有价值的研究，是关于如何提供建设可靠分布式系统的方法。Horus toolkit 让一个开发人员构建一组进程，可以使得任何一个进程发送的消息，要么被所有进程收到，要么所有进程都收不到。这个结果可以大大简化分布式系统应用，并且通常作为部分中间件的实现。

## 设计目标

这一节，我们讨论4个重要的目标，这些是分布式系统的建设应该满足的。分布式系统应当是的资源可以容易访问；应当隐藏资源是分布在一个网络中；应当对外开放（使得其他节点可以加入）；可扩展。

**支持资源分享**：

分布式系统的一个重要的设计目标是是的用户或者应用能够容易的访问远程资源。这里的资源可以使任何事物，但典型的是外围设备，存储设备，数据，文件，服务和网络等。分享资源的原因有很多。一个明显的原因是经济成本。例如：将数据存储在一个可共享的设备中要比为每一个用户单独存储一个资源的成本低。

用户和资源的连接是的信息的交换变得容易，这也要归功于简单的网络协议（这里的简单应该是只方便）。互联网的连接使得地理上分布很广泛的人可以在一起工作，如用于协作编辑、电话会议等。

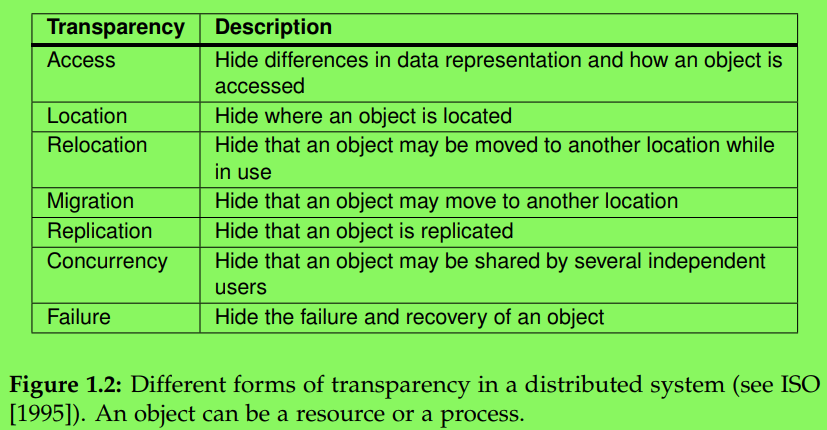
然而，分布式资源分享最好的体现还是在peer-to-peer的文件分享上，就像BitTorrent这个软件。这个分布式系统使得用户通过互联网分享文件变得容易。Peer-to-Peer 网络通常和分布式文件介质相关，如音频和视频。在其他的情况下，这个技术已经用在通过主机间的软件升级，服务备份，数据同步等大数据的分布式网络中。

**使得分布式透明：**

分布式系统的一个重要的目标是掩藏进程和资源是分布在很远距离的事实。换句话说，对用户和应用来说，分布式系统使得资源和进程的分布变得透明。

**分布式透明的类型：**

分布式系统在很多方面有透明这个概念，几个比较重要的如下图1.2所示。我们用Object表示一个进程或者一个资源。



**访问透明** 对数据展现方式的不同以及数据访问的隐藏。基本上，我们想要隐藏机器架构的不同，但更重要的是我们将对不同机器和操作系统的展现方式加以隐藏。举个栗子，一个分布式系统会有多个操作系统，每个操作系统都有自己的文件命名特性。不同的命名特性，不同的文件操作，不同的底层进程通信都是应该对用户和应用透明的（隐藏）。

一组重要的透明类型涉及到进程和资源的位置信息。位置信息透明的意思是，用户或应用无法分辨一个进程或资源的具体物理位置。命名在位置透明的功能里起到了重要的作用。特别是，位置透明通常可以通过设置资源的逻辑路径来实现，资源的位置命名并不是秘密的编码的。举个例子就是 URL ，<http://www.prenhall.com/index.html>，这个url并没有具体给出Prentice Hall 的Web服务地址。这个URL也没有给出index.html是否一直在现在的地址或者最近才移动过来的。就是说，可能整个网址是从一个数据中心搬移过来的，然而用户却没有发现。第二个例子是重定向透明，这一点在云计算中十分重要，稍后我们将讲解这部分。

重定向透明指的是分布式系统，而迁移透明更多的是指由人产生的迁移，并且不影响通信和操作。例如移动中打电话，移动中开电话会议等。

同样的，复制在分布式系统中也扮演者重要的作用。例如，复制的资源可以增加可靠性，并且把资源复制到访问路径可以增加性能。复制透明需要将存在多份相同资源的事实隐藏起来，并且多份资源可以保证一份出问题时，其他的复制资源可以进行代替。为了对用户隐藏复制，我们需要将所有的复制资源进行相同的命名。因此，一个支持复制透明的系统应当也支持位置透明，否则不太容易实现。

我们提到分布式系统的一个重要目标是资源分享。在通信层次上，资源分享体现在一种合作层面上。例如，两个独立的用户需要访问服务器的同一个资源或数据库的同一张表。在这种情况下，让用户感觉不到另外有人在访问同样的资源是十分重要的。这种叫做**并发透明**。一个重要的课题是这种并发访问需要让资源保持一致性。这可以通过加锁的机制，例如通过给需要的资源加入排它锁。一个更完善的方案是使用事务，但这在分布式系统是是很难实现的，这会导致可扩展性的问题。

其次，提供**错误透明**也是十分重要的。这表示，用户或应用将感知不到系统的某台机器发生错误，而系统将会自然而然的进行修复。错误掩盖在分布式系统中是一个很艰难的课题并且经常是不可实现的，我们将在第八章阐述。错误掩盖最主要的困难就是无法判断一个进程究竟是死亡了，或者只是相应太慢。正如我们访问一个运行很繁忙的服务器时，浏览器最终超时并报告服务不可用。而我们根本无法盘点这个服务就是挂掉了还是网络拥塞的原因。