# 进程

在这一章中，我们将研究各种不同类型的进程是如何影响分布式系统工作的。进程的概念起源于操作系统中，它是用来定义一个程序的运行的。在操作系统层面，进程的管理和调度可能是最重要的课题。然而，当在分布式系统中讨论时，进程的一些其他课题变得更为重要。

我们首先重点讨论线程和它在分布式系统的作用。事实证明，线程不但在多核多处理器环境中对性能有重要的作用，更有助于构造客户端和服务端。在很多情况下，我们看到线程被进程所替代，并且使用底层操作系统来保证保护和促进通信。然而，当性能受到威胁时，线程将扮演着重要的角色。

近些年，虚拟化的概念开始流行起来。虚拟化允许应用程序及其完整的环境(包括操作系统)与其他应用程序并发运行，但高度独立于底层硬件和平台，从而实现高度的可移植性。另外，虚拟化可以帮助隔离由故障和安全引起的错误。这对于分布式系统是很重要的概念，我们将单独讨论。

客户-服务器组织架构在分布式系统中很重要。在这一章节，我们将仔细研究客户端和服务端的典型组织。我们将注意到服务端的典型设计，包括那些在object-based分布式系统中使用的类型。一个应用很广泛的Web服务是Apache，我们将单独了解。服务端的集群组织也很重要，特别是他们需要合作得像是单一系统在工作一样。我们将介绍类似的例子，包括广域服务PlanetLab。

在广域分布式系统中，一个重要的课题是机器间的晋城移动。进程转移或者代码转移可以帮助实现可伸缩性，或者帮助动态配置客户端和服务端。关于代码转移还有其如何实现的，我们将在这一章节讨论。

## 3.1线程

虽然进程组成了分布式系统的基本单元，但实践表明，分布式系统以进程作为最小粒度并不足够充分（就是还有线程呗）。作为替代，有进程控制的多个线程组成了更小的粒度，这能更好的构建分布式应用并获得更好的性能。在这一节中，我们将关注在分布式系统中的线程是如何发挥作用的并解释它为什么这么重要。关于线程以及如何通过线程构建应用可以参考

[Lewis and Berg, 1998; Stevens, 1999; Robbins and Robbins, 2003]。这些很好的从概念上诠释了多线程并发程序。

**线程的介绍**

为了理解分布式系统中的线程，我们需要首先理解进程是什么，并且进程和线程是什么关系。为了运行一个程序，操作系统会创造一系列虚拟进程，每一个会运行不同的程序。为了跟踪这些进程，操作系统有一个进程对照表，包括存储CPU的寄存器只，内存映射，打开文件，计数器，privileges，等。这些条目构成了进程的上下文。

进程上下文可以看作是硬件处理器上下文的软件模拟。硬件处理器上下文包括硬件处理终端，以及存储CPU处理现场的最小信息。这个进程上下文包括程序计数器，或者其他的寄存器值比如栈指针。

进程通常定义为一个运行的程序，也即正在操作系统虚拟处理器上运行的程序。一个关键问题是，操作系统会很小心的保证独立的进程之间不会相互干扰的彼此。换句话说，多个进程共享同一个cpu以及其他硬件资源这件事，是透明的。通常，操作系统需要硬件的支持来完成这种分割操作。

这种透明性是有代价的。比如，当一个进程被创建时，操作系统必须创建一个完全独立的地址空间。这种分配可以意味着初始化内存段，例如，将数据段归零，将相关程序复制到文本段，并为临时数据建立堆栈。这样，在两个进程间切换CPU需要消耗性能。除了在寄存器临时存放数据外（包括程序寄存器和栈指针），操作系统同样会修改内存管理单元（MMU）的寄存器和使地址转换缓存无效，例如在转换后备缓冲区(TLB)中。另外，如果操作系统需要支持超过他内存大小的进程数量，那么需要在内存和磁盘间进行内存交换。

像进程一样，线程也会执行一段代码，并独立于其他线程。然而，不同于进程，如果实现并发透明性会导致性能下降，则不会尝试实现高度的并发透明性。因此，线程系统通常保持最小的信息来允许CPU可以被其他线程共享。特别是，一个**线程上下文**通常和进程上下文的组成没有区别，除了一些关于线程管理的信息。例如，线程系统可以跟踪一个线程当前在互斥量变量上被阻塞的事实，以便不选择它执行。关于其他线程的不必要的信息是不必理会的。因此，保护数据不受单个进程内线程的不适当访问完全留给应用程序开发人员。因为我们发现一个处理器的上下文包含在线程上下文中，而线程上下文包含在进程上下文中。

正如我们刚才概述的，部署线程有两个重要含义。首先，多线程应用的系统性能很难比同样的单线程系统性能差。实际上，多线程可以获得更多的性能。其次，由于线程不能自动的像进程一样保护自己不受其他线程干扰，多线程应用的独立需要更多的智慧和努力。合适的设计和事情的简单化是有帮助的选择。不幸的是，从目前的实践来看，这个原则并不好理解。

**在非分布式系统中的线程使用**

在讨论分布式系统中线程的功能之前，我们首先讨论在非分布式系统中线程的功能。使用多线程系统有很多好处。

最重要的一个优势是，作为但线程进程，当一个阻塞方法被调用时，进程将会被阻塞。为了说明这一点，考虑一个应用程序，例如电子表格程序，并假设用户持续地、交互地想要更改值（举个访问数据库的栗子更好，比如需要向数据库刷10分钟的数据，进程将阻塞10分钟）。电子表格程序的一个重要特性是，它维护不同单元格之间的功能依赖关系，这些单元格通常来自不同的电子表格。因此当一个单元更改时，其他依赖的单元会自动更新。当用户更改一个单元时，这种更改会出发一系列计算。如果只有一个进程进行控制，在程序等待输入之前，计算是不能够完成的。像这样，当依赖的计算完成之前，提供数据是不容易的。一个简单的处理方式是，获取两个线程：一个负责处理用户输入，另一个负责更新。同时，第三个线程将数据备份。

另一个多线程的优点是当在一个多处理器或者多核系统上运行程序是，可以进行并行计算。当数据都被存储在共享的内存时，每一个线程都被分配给了不同的CPU和处理器。如果涉及得当，这种并行计算可以是透明的：这个进程同样可以在单核系统中运行，尽管这样运行会比较慢。在多核和多处理器价格变得相对便宜的情况下，多线程的并行处理变得很重要了。这种计算机系统通常发生在客户-服务端系统的服务端，但目前也广泛应用于智能手机等设备。