# 容错

分布式系统区别于单机系统的一个特征是部分故障的概念:系统的一部分正在故障，而其余部分继续运行，而且结果似乎是正确的。分布式系统设计的一个重要目标是构建一个能够自动从部分故障中恢复而不会严重影响整体性能的系统。特别是，当发生故障时，在进行维修时，系统应继续以可接受的方式运行。换句话说，分布式系统应该是容错的。

在本章中，我们将进一步研究实现容错的技术。在提供了一些一般背景之后，我们将首先通过进程组查看进程弹性。在这种情况下，多个相同的进程协作提供单个逻辑进程，以确保其中一个或多个进程可以在客户端没有注意到的情况下失败。进程组中一个特别困难的地方是在组成员之间就执行客户机请求的操作达成共识。到目前为止，Paxos是一种常用的，但相对复杂的算法，我们通过从头开始构建它来解释。同样，我们仔细审查了在何种情况下能够达成一致意见。

实现容错和可靠的通信是密切相关的。除了可靠的客户机-服务器通信之外，我们还关注可靠的组通信，尤其是原子组播。在后一种情况下，消息被传递给组中的所有非故障进程，或者根本不传递给任何进程。拥有原子多播使容错解决方案的开发变得容易得多。

原子性在许多应用程序中都很重要。在本章中，我们将关注所谓的分布式提交协议，通过该协议，一组进程可以联合提交它们的本地工作，也可以集体中止并返回到以前的系统状态。

最后，我们将研究如何从失败中恢复。特别是，我们要考虑何时以及如何保存分布式系统的状态，以便稍后恢复到该状态。

8.1容错介绍

容错一直是计算机科学研究的热点。在本节中，我们首先介绍与处理故障相关的基本概念，然后讨论故障模型。本文还讨论了冗余是故障处理的关键技术。有关分布式系统容错的更一般信息，请参见，例如[Jalote, 1994;Shooman, 2002]或[Koren和Krishna, 2007]。

**基本概念**

为了理解容错在分布式系统中的作用，我们首先需要更仔细地研究分布式系统容错的实际含义。容错与所谓的**可靠系统**密切相关。可靠性这个术语涵盖了分布式系统的一些有用需求，包括以下内容[Kopetz和Verissimo, 1993]:

•可用性

•可靠性

•安全

•可维护性

**可用性**定义为系统可以立即使用的属性。一般来说，它指的是系统在任何给定时刻正常运行并可以代表用户执行其功能的概率。换句话说，一个高可用性的系统很可能在某个特定的时刻工作。

**可靠性**是指系统能够连续运行而不发生故障的特性。与可用性相反，可靠性是根据时间间隔而不是瞬间来定义的。一个高度可靠的系统很可能在相当长的一段时间内不间断地工作。与可用性相比，这是一个细微但重要的区别。如果一个系统平均每小时宕机一毫秒，看上去是随机的，那么它的可用性超过99.9999%，但仍然是不可靠的。同样，一个系统不会崩溃，但每年8月都会关闭两个星期，它的可靠性很高，但只有96%的可用性。这两个是不一样的。

**安全**是指当系统暂时不能正常运行时，不会发生灾难性事件的情况。例如，许多过程控制系统，例如那些用于控制核电站或将人送入太空的系统，都需要提供高度的安全。如果这样的控制系统发生短时间的失效，后果可能都是灾难性的。过去的许多例子(可能还有更多的例子)表明，构建安全的系统是多么困难。

最后，**可维护性**是指一个失败的系统可以多么容易地修复。一个高度可维护的系统也可能显示出高度的可用性，特别是如果故障可以自动检测和修复。然而，正如我们将在本章后面看到的，从失败中自动恢复说起来容易做起来难。

通常，可靠的系统还需要提供高度的安全性，特别是涉及到完整性等问题时。我们将在下一章讨论安全性。

当一个系统不能兑现它的承诺时，它就会失败。特别是，如果分布式系统的设计目的是为其用户提供许多服务，那么当无法(完全)提供其中一个或多个服务时，系统就会失败。错误是系统状态中可能导致故障的部分。例如，当在网络上传输数据包时，可以预期一些数据包在到达接收端时已经损坏。

错误的原因称为**故障**。显然，找出导致错误的原因是很重要的。例如，错误的或不好的传输介质很容易导致数据包损坏。在这种情况下，相对容易排除故障。然而，在无线网络中，恶劣的天气条件也可能导致传输错误。改变天气来减少或防止错误是有点棘手的。

另一个例子，一个崩溃的程序很明显是一个失败，这可能是因为程序输入了一个包含一个编程错误的代码分支(例如，，程式错误)。这个bug通常是由程序员引起的。换句话说，是程序员的错误导致变成失败的。

建立可靠的系统与控制故障密切相关。正如Avizienis等人[2004]所解释的那样，可以对预防、容忍、消除和预测故障进行区分。就我们的目的而言，最重要的问题是容错，这意味着即使存在故障，系统也可以提供服务。例如，通过对传输包应用纠错码，可以在一定程度上容忍相对较差的传输线路，并降低错误(损坏的包)导致故障的可能性。

故障通常分为瞬态、间歇或永久性故障。**瞬态故障**发生一次，然后消失。如果操作重复，故障就会消失。一只鸟飞过微波发射器的光束可能会导致某些网络上的比特丢失(更不用说一只烤鸟了)。如果传输超时并重试，它可能会在第二次工作。

间歇性故障是从发生开始，然后自动消失，然后重新出现，依此类推。连接器上的松动接触经常会导致间歇性故障。间歇性故障由于诊断困难，造成严重后果。通常，当故障医生出现时，系统可以正常工作。

永久性故障是指在替换错误组件之前一直存在的故障。烧坏的芯片、软件bug和磁盘头崩溃都是永久性故障的例子。