学习笔记

baikaishui

2024 年 9 月 19 日

摘要

Chapter 1

分布式系统特征

1.1 简介

分布式系统是其组件分布在连网的计算机上，组件之间通过传递消息进行通信和动作协调的系统，重要特征：组件的并发性、缺乏全局时钟、组件故障的独立性。构造分布式系统的挑战是处理其组件的异构性、开放性(允许增加或替换组件)、安全性、可伸缩性（用户的负载或数量增加时能正常运行的能力)、故障处理、组件的并发性、透明性和提供服务质量的问题。

构造和使用分布式系统的主要动力来源于对共享资源的期望

复杂事件处理：Complex Event Processing

感兴趣数据项在分布式系统中称为事件

防火墙的作用是保护企业内部网，防止未授权的消息进出网络，通过过滤到达消息和外发消息来实现。可以在源和目的地进行过滤。

互联网服务提供商（Internet service Provider,ISP）是给个体用户和小型

云被定义成一组基于互联网的应用，并且足以满足大多数用户需求的存储和计算服务的集合，这使得用户能大部分或全部免除本地数据存储和应用软件的使用。网格计算也被看成是一种云计算，可以看成是云计算更通用模式的先驱，侧重于科学计算。

服务表示计算机系统中管理相关资源并提供功能给用户和应用的一个单独的部分；客户与服务器之间的完整交互，即从客户发送一个请求到它

分布式系统与云计算的区别：云计算主要侧重资源的使用方式，包括硬件资源、软件资源，分布式系

接收到服务器的应答，称为一个**远程调用**。在面向对象语言实现的分布式系统，资源被封装成对象，并由客户访问，这时，称一个客户对象调用了一个服务对象上的方法。客户与服务器仅仅是针对一个请求中扮演的角色而言，客户是主动的，服务器是被动的；服务器是持续运行的，而客户所持续的时间只是客户所属的那部分应用程序持续的时间。

1.1.1 挑战

异构性

- 网络
- 计算机硬件
- 操作系统
- 编程语言
- 由不同开发者完成的软件实现

中间件是指一个软件层，它提供了编程抽象，同时屏蔽了底层网络、硬件、操作系统和编程语言的差异性。除了解决异构性的问题外，中间件为服务器和分布式应用的程序员提供了一致的计算模型，这些模型包括远程对象调用、远程事件通知、远程 SQL 访问和分布式事务处理。**移动代码**是指能从一台计算机发送到另一台计算机，并在目的计算机上运行的代码，Java applet 是一个例子。

开放性

计算机系统的开放性是决定系统能否以不同的方式被扩展和重新实现的特征，分布式系统的开放性取决于新的共享服务能被增加和供多种客户程序使用的程度。除非软件开发者能获得系统组件的关键软件接口的规约和文档，否则无法实现开放性。开放的分布式系统主要强调可扩展性，从硬件层上，通过在网络上增加计算机实现硬件层次上扩展；通过引入新的服务、重新实现旧的服务实现在软件层次上的扩展，最终使用应用程序共享资源。

开放的分布式系统的特征总结如下：

- 发布系统的关键接口是开放系统的特征
- 开放的分布式系统是基于一致的通信机制和发布接口访问共享资源的
- 开放的分布式系统能用不同销售商提供的异构硬件和软件构造。

安全性

信息资源的安全性包括三部分：机密性（防止泄露给未授权的个人）、完整性（防止被改变或被破坏）、可用性（防止对访问资源的手段的干扰）。

可伸缩性

分布式系统可在不同规模下有效且高效地运转。如果资源数量和用户数量激增，系统仍能保持其有效性，那么该系统就称为可伸缩的。

主要面临的挑战：控制物理资源的开销；控制性能损失；防止软件资源用尽；避免性能瓶颈。

故障处理

处理故障的技术：检测故障，掩盖故障，容错，故障恢复，冗余。

系统的可用性性是对系统可用时间的比例的一个度量指标。

并发性

在分布式系统中，代表共享对象的任何一个对象必须负责确保它在并发环境中操作正确，这不仅适用于服务器，也适用于应用中的对象。为了使对象在并发环境能安全使用，它的操作必须在数据保持一致的基础上同步。

透明性

透明性被定义成对用户和应用程序屏蔽分布式系统的组件的分离性，使系统被认为是一个整体，而不是独立组件的组合。透明性的含义对系统软件的设计有重大影响。

- 访问透明性：用相同的操作访问本地资源和远程资源
- 位置透明性：不需要知道资源的物理或网络位置
- 并发透明性：几个进程能并发地使用共享资源进行操作且互不干扰
- 复制透明性：使用资源的多个实例提升可靠性和性能，而用户和应用程序员无须知道副本的相关信息。
- 故障透明性：屏蔽错误，不论是硬件组件故障还是软件组件故障，用户和应用程序都能够完成它们的任务
- 移动透明性：资源和客户能够在系统内移动而不会影响用户或程序的操作
- 性能透明性：当负载变化时，系统能被重新配置以提高性能
- 伸缩透明性：系统和应用能够进行扩展而不改变系统结构或应用算法。

服务质量

系统的非功能特性，即影响客户和用户体验的服务质量是可靠性、安全性和性能。满足变化的系统配置和资源可用性的适应性已被公认为服务质量的一个重要方面。可靠性和安全性问题是设计大多数计算机系统时的关键，服务质量的性能方面源于及时性和计算吞吐量。

Chapter 2

系统模型

2.1 简介

每类模型试图对分布式系统设计的一个相关方面给出抽象、简化但一致的描述。物理模型从计算机及其互联的网络方面考虑系统的硬件组成；体系结构模型从系统的计算元素执行的计算和通信任务方面来描述系统；基础模型抽象的观点描述分布式系统的某个方面，如交互模型，它考虑在系统元素之间通信的结构与顺序，故障模型考虑一个系统可能不能正确操作的方式，安全模型考虑如何保护系统使其不受到正确操作的干扰或不被窃取数据。

分布式系统的困难和威胁主要包括：

- 使用模型的多样性
- 系统环境的多样性
- 内部问题，包括非同步的时钟、冲突的数据更新、多种涉及系统单个组件的软硬件故障模式
- 外部威胁：包括对数据完整性、保密性的攻击以及服务拒绝攻击

2.2 物理模型

物理模型是从计算机和所用的网络技术的特定细节中抽象出来的分布式系统底层硬件元素的表示

2.3 体系结构模型

一个系统的体系结构是用独立指定的组件以及这些组件之间的关系来表示的结构。整体目标是确保结构满足现在和将来可能的需求。主要关心的是系统的可靠性、可管理性、适应性和性价比。

2.3.1 体系结构元素

分布式系统的基础构建块，有必要考虑下面四个关键问题:

- 在分布式系统中，通信的实体是什么
- 它们如何通信，特别是使用什么通信范式
- 它们在整个体系结构中扮演什么角色，承担什么责任
- 它们怎样被映射到物理分布式基础设施上

通信实体，从面向系统的角度看，包括进程和结点。面向问题，包括: 对象，在分布式面向对象的方法中，一个计算由若干交互的对象组成，这些对象代表分解给定问题领域的自然单元; 组件: 类似于对象，也通过接口访问，关键区别在于组件不仅指定其接口而且给出关于其他组件/接口的假设，其他组件/接口是组件完成它的功能必须有的; web 服务: web 服务与对象和组件相关，也是采用基于行为封装和通过接口访问的方法，但通过利用 web 标准表示和发现服务，web 服务本质上被集成到万维网中。

通信范型主要包括三种: 进程间通信、远程调用、间接通信。进程间通信指的是用于分布式系统进程之间通信的相对底层的支持，包括消息传递原语、直接访问由互联网协议提供的 API 和对多播通信的支持。远程调用代表分布式系统中最见的通信范型，覆盖一系列分布式系统中通信实体之

间基于双向交换的技术，包括调用远程操作、过程或方法。请求－应答协议用于支持客户－服务器计算。

远程过程调用（Remote Procedure Call,RPC），在 RPC 中，远程计算机上进程中的过程能被调用，好像它们是在本地地址空间中的过程一样。

远程方法调用（Remove Method Invocation,RMI）应用于分布式对象环境，一个发起调用的对象能调用一个远程对象中的方法。

间接通信包括：组通信，是支持一对多通信的多方通信范型；发布－订阅系统，共享同一个关键的特征，即提供一个中间服务，有效确保由生产者生成的信息被路由到需要这个信息的消息者；消息队列，提供了点对点服务，生产者进程能发送消息到一个指定的队列，消息者进程能从队列中接收消息，队列临产者和消费者的中介；元组空间：进程能把任意的结构化数据项（元组）放在一个持久元组空间，其他进程可以指定感兴趣模式，从而可以在元组空间读或者删除元组；分布式共享内存：提供一种抽象，用于支持在不共享内存的进程之间共享数据。

通信实体		通信范型		
面向系统的实体 结点 进程	面向问题的实体 对象 组件 web 服务	进程间通信 消息传递 套接字 多播	远程调用 请求-应答 RPC RMI	间接通信 组通信 发布-订阅 消息队列 元组空间 DSM

角色与责任：客户-服务器是讨论分布式系统最常引用的体系结构；对待体系结构作为对等方进行协作交互，不区分客户和服务器或运行它们的计算机，促使对等系统发展的主要观点是一个服务的用户所拥有的网络和计算资源也能被投入使用以支持那个服务。

放置考虑的问题是诸如对象或服务这样的实体是怎样映射到底层的物理分布式基础设施上的，需要考虑实体间的通信模式、给定机器的可靠性和它们当前的负载、不同机器之间的通信质量等。主要关注以下放置策略：

- 将服务映射到多个服务器
- 缓存，代理服务器的目的是通过减少广域网和 web 服务器的负载，提高服务的可用性和性能。

- 移动代码
- 移动代理，是一个程序，它从一台计算机移动到网络上另一台计算机，代表某人完成诸如信息搜集之类的任务，最后返回结果。

其他模式：代理模式是主要用于支持远程过程调用或远程方法调用的过程位置透明性。web 服务中的业务代理（brokerage）支持互操作性的体系结构模式，由服务提供者、服务请求者与服务代理（提供与请求的服务一致的服务）三部分组成。反射（reflection）模式作为支持内省（系统的动态发现的特性）和从中调停（动态修改结构或行为的能力）的手段而被持续地使用。在一个反射系统中，标准的服务接口在基础层可供使用，但元层接口也可以提供对涉及服务实现的组件及组件参数的访问。

2.3.2 相关的中间件解决方案

中间件任务是为分布式系统的开发提供一个高层的编程抽象，并且，通过分层，对底层基础设施中的异构性提供抽象，从而提升互操作性和可移植性。除了编程抽象外，中间件也能够提供分布式系统的基础设施服务。分布式程序正确行为在很多层面上依赖检查、错误校正机制和安全手段。

2.4 基础模型

所有模型都共享的设计需求：实现进程及网络性能和可靠性特征，确保系统中资源的安全性。

2.4.1 交互模型

分布式算法定义了组成系统的每个进程所采取的步骤，包括它们之间的消息传递。**通信信道的性能**，延迟：从一个进程开始发送消息到另一个进程开始接收消息之间的间隔时间称为延迟；计算机网络带宽是在给定时间内网络能传递的信息总量；抖动是传递一系列消息所花费的时间的变化值；时钟漂移率（clock drift rate）指计算机时钟偏离绝对参考时钟的比率。

异步分布式系统对进程执行速度、消息传递延迟和时钟漂移率没有限制。

2.4.2 故障模型

故障模型定义了故障可能发生的方式，以便理解故障所产生的影响，包括遗漏故障、随机故障和时序故障。遗漏故障类错误指进程或通信通道不能完成它应该做的动作；在异步系统中，超时只能表明进程没有响应，它可能崩溃了，也可能是执行速度慢，或者消息还没有到达。随机故障用于描述可能出现的最坏的故障，此时可能发生任何类型的错误。时序故障发生在同步分布式系统中。

2.4.3 安全模型

通过保证进程和用于进程交互的通道的安全以及保护所封装的对象免遭未授权访问可实现分布式系统的安全。

Chapter 3

网络和网际互联

计算机网络所基于的原理包括协议分层、包交换、路由以及数据流等，网际互连技术使得异构网络可以集成在一起。

3.1 简介

对网络的需求:

- 性能，影响两个互连计算机间消息的传输速度两个参数，延迟和点到点的数据传输率。延迟是指执行发送操作之后和到达目标计算机之前的这段时间；数据传输率是指一旦传输过程开始，数据在网络上两台计算机间传输的速度。传输率主要由它的物理特征决定的，而延迟主要由软件开销、路由延迟和与负载有关有统计因素决定。过载是指在网络上传输的数据过多。
- 可伸缩性。为了适应互联网下一阶段的发展，技术人员正在对寻址和路由机制进行一些实质性的改变
- 可靠性。通信错误的检测和校正通常由应用级软件完成。
- 安全性。防火墙在网关上运行，所谓网关是企业内部网入口点处的计算机。
- 移动透明性

- 服务质量
- 组播

网络这部分内容讲述的并不深，不再做过多的记录，只是把一些基本概念重述一下。

自适应路由，网络两点间的通信的最佳路由会周期性的重新评估，评估时会考虑到当时的网络流量以及故障情况。路由算法包括两部分内容：一是决定每个数据包穿梭于网络时所应经过的路径；二是必须通过监控流量和检测配置变化或故障来动态地更新网络的知识。

路由器信息协议（Router Information Protocol）：通过发送自己路由表信息的概要和邻接结点相互交互网络信息。周期性地并且只要本地路由表发生改变，就将自己的路由表发给邻接的所有可访问的路由器；当从邻接路由器接收这样的表时，如果接收到的表中给出了到达一个新目的地的路由，或对于已有的一个目的地更好的路由，则用新的路由更新本地的路由。

向量-距离算法可以用多种方法改进。开销，也被称为度量，可以根据链路的实际带宽来计算；可以修改算法，以增加信息收敛的速度。

经验表明，当网络的负载超过了其能力的 80%，系统的吞吐量会因为数据包丢失而下降，除非控制高负载链路的使用。

UDP 数据报被封装在一个 IP 数据包中，它具有一个包含源端口号和目的端口号的短的头部、一个长度域和一个校验和，它不提供舆保证；TCP 建立了一个双向的通信通道，包含额外机制以保证可靠性：排序，TCP 发送进程将流分割成数据片断序列，然后将之作为 IP 数据包发送，每个 TCP 都有一个序号，它在该片断的第一个字节给出流中的字节数；流控制，发送方管理不能使接收方或者中间结点过载，这通过片断确认机制完成，确认片断中的窗口大小域指定了在下一个确认之前允许发送方传送的数据量。

Chapter 4

进程间通信

特征:

同步和异步每个消息目的地与一个队列相关，发送进程将消息添加到远程队列中，接收进程从本地队列中移除消息。在同步形式的通信中，发送进程和接收进程之间在每一个消息上阻塞。在异步情况下，send 操作是非阻塞的，只要消息被复制到本地缓冲区，发送进程就可以继续进行其他处理，消息的传递与发送进程并行进行。receive 操作有阻塞型和非阻塞型两种形式。在不阻塞操作中，接收进程在发出 recieve 操作后可继续执行它的程序，该操作在后台提供一个缓冲区，但它必须通过轮循或中断独立接收缓冲区已满的通知。在多线程情况下，阻塞型 recieve 缺点较少。

消息目的地可以通过名字使用服务，可以使用服务重定位，但不能迁移，迁移指在系统运行时移动服务所在的位置。

可靠性：如果一个点对点消息服务在丢失了“合理”数量的数据包后，仍能保证发送消息，那么该服务就被称为可靠的。

排序：有些应用要求消息要按发送的顺序发送。

UDP 数据报通信的一些问题:

- 消息大小，接收进程要指定固定大小的用于接收消息的字节数组。底层 IP 协议最大数据包为 216 字节。大多数情况下，消息大小被限制为 8KB 左右。
- 阻塞，支持阻塞和非阻塞

- 超时，选择适当的超时不容易
- 任意接收，recieve 不指定消息的来源

UPD 故障模型：

- 遗漏故障：消息偶尔会丢失
- 排序，消息可能没有按顺序发送

4.1 覆盖网络

一个面向特定应用的虚拟网络能建立在已有网络上并为特定的应用进行优化，而不改变底层网络的特征。每个虚拟网络有它自己特定的寻址模式、协议和路由算法，它们被重新定义以满足特定类应用的需要。

覆盖网络（overlay network）是一个结点和虚拟链接组成的虚拟网络，它位于一个底层网络之上，提供一些独有的功能：

- 满足一类应用需求的服务或一个特别高层的服务，如多媒体内容的分发；在一个给定的联网环境中的更有效的操作，如在一个自组织网络中的路由；
- 额外的特色，如组播或安全通信。

Chapter 5

远程调用

请求-应答协议描述了一个基于消息传递的范型,该协议支持在客户/服务器计算中遇到的消息双向传输。请求-应答协议的故障模型包括: 存在遗漏故障和没有保证消息按照其发送顺序进行传输, 除此之外, 该协议还会遇到进程故障问题。

实现基于 TCP 流的请求-应答协议的原因之一是期望避免实现多包协议, 因为 TCP 流可以传输任意长度的参数和结果。如果应用不要求 TCP 提供的所有机制, 那么更有效的方法是定制一个基于 UDP 实现的协议。

HTTP 方法: 每个用户请求指定使用服务器资源的方法和该资源的 URL, 应答则说明该请求的状态:

GET: 请求在参数中给出的 URL 对应的资源。

HEAD: 与 GET 相同, 但不返回任何数据。

POST: 指定资源的 URL, 该资源可处理在请求消息体中提供的数据。

PUT: 要求请求中提供的数据在存储时以指定的 URL 所标识的资源, 要么作为现有资源修改, 要么作为一种新资源。

OPTIONS: 服务器提供给客户端能够应用到给定 URL 及其特定需求的方法列表

TRACE: 服务器返回请求的消息, 用于诊断目的。

DELETE: 服务器删除给定 URL 所标识的资源。

PUT 和 DELETE 操作是幂等操作。

5.1 远程过程调用

远程过程调用使分布式编程和传统编程相似，即实现了高级的分布透明性，底层 RPC 系统隐藏了分布式环境重要的部分，包括对参数和结果的编码和解码、消息传递以及保留过程调用要求的语义。

RPC 调用语义：**或许调用语义**远程方法可能执行一次或者根本不执行，当没有任何容错措施时，就启用了或许语义。它可能遇到的故障模型：遗漏故障，如果调用或结果消息丢失；系统崩溃，由于包含远程的服务器出现故障。或许语义仅对那些可以接受偶然调用失败的应用是有用的。**至少一次调用语义**，调用者可能收到返回的结果，也可能收到一个异常。异常信息通知调用者没有接收到执行结果。至少一次调用语义可以通过重发请求消息来达到，它屏蔽了调用或结果消息的遗漏故障。其可能的故障模型为：由于包含远程对象的服务器故障而引起的系统崩溃；随机故障。如果服务器操作中的操作被设计成幂等操作，则至少一次是可以接受的。**至多一次调用语义**：调用者可以接收返回的结果，也可以接收一个异常。异常信息通知调用者没有收到执行结果。

容错措施

Chapter 6

基本概念

- 统一资源定位器: Uniform Resource Lockor,URL
- 统一资源定位符: Uniform Resource Indentifier,URI

Chapter 7

问题

- 联网的计算机如何做到时间同步
-