第一章 分布式系统的特征

分布式系统：其组件分布在联网的计算机上，组件之间通过传递消息进行通信和动作协调的系统。

分布式系统的重要特征：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：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、透明性和提供的服务质量的问题。

分布式系统：其硬件或软件组件分布在联网的计算机上，组件之间通过传递消息进行通信和动作协调的系统。

分布式系统的显著特征

|  |  |
| --- | --- |
| 并发 | 用户可以在各自的计算机上工作，在必要时共享诸如Web页面或文件之类的资源。 |
| 缺乏全局时钟 | 网络上的计算机与时钟同步所达到的准确性是有限的，即没有一个正确时间的全局概念。 |
| 故障独立性 | 网络故障导致网上互联的计算机的隔离，但这并不意味着它们停止运行，事实上，计算机上的程序不能够检测到网络是出现故障还是网络运行得比通常慢。类似的，计算机的故障或系统中程序的异常终止（崩溃），并不能马上使与它通信的其它组件了解。系统的每个组件会单独地出现故障，而其它组件还在运行。 |

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。

分布式系统的例子

|  |  |
| --- | --- |
| Web搜索 |  |
| 大型多人在线游戏（Massively Multiplayer Online Game，MMOG） |  |
| 金融交易 | 金融交易系统的重点：对感兴趣数据项的通信和处理。  事件：在分布式系统中，感兴趣数据项。  分布式基于事件的系统  复杂事件处理（Complex Event Processing，CEP）：交易系统必须处理各种各样的事件流，这些事件流高速到达，经常需要实时处理来检测表示交易机会的模式。这在过去曾经是手工处理的，但在竞争压力下变成自动处理。 |

推动分布式系统发展的趋势

|  |  |
| --- | --- |
| 出现了泛在联网技术 | 防火墙的作用：保护企业内部网，防止未授权的消息进出网络。  互联网服务提供商（Internet Service Provider，ISP）：给个体用户和小型组织提供宽带链接和其他类型连接的公司，使他们能获得互联网上任何地方的服务；同时提供诸如电子邮件和Web托管等本地服务。  企业内部网通过主干网实现互相链接。  主干网：具有高传送能力的网络链接，通常采用卫星连接、光缆和其他高带宽线路。 |
| 出现了无处不在计算 | 移动计算：用户在移动或访问某个非常规环境时执行计算任务的性能。  位置感知（上下文感知的计算）：为用户在其移动时提供访问资源（如打印机）或方便的利用附近的销售点的情形在不断增加。  无处不在计算：对在用户的物理环境（包括家庭、办公室和其他自然环境）中存在的多个小型、便宜的计算设备的利用。  无处不在：小型计算设备最终将在不会引人注意的日常物品中普及。也就是说，它们的计算形为将透明地紧密捆绑到这些日常物品的物理功能上。  自发互操作  服务发现：让访问者的设备在访问地的网络上通信，并将设备与合适的本地设备相关联。 |
| 对多媒体设备的需求增加 | 多媒体支持：以集成的方式支持多种媒体类型的能力。  离散类型媒体。例如：图片、正文消息  连续类型媒体。例如：音频、视频  吞吐量的单位：帧/秒  网络播放（webcasting）：在互联网上广播连续媒体（典型的是音频或视频）的能力。 |
| 把分布式系统作为一个设施 | 数据中心：网络化的设施，为用户或机构提供对拥有大量数据的数据仓库的访问。  计算基础设施  操作系统虚拟化  云计算（cloud computing）  云：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。  集群计算机（cluster computer）：互联的计算机集合，它们紧密协作提供单一的、集成的高性能计算能力。  刀片服务器（blade server）：最小的包含处理和（主存）存储能力的计算元素。  刀片系统由包含在一个刀片机架中的大量刀片服务器组成。  集群服务器的总目的：提供一系列的云服务，包括高性能计算能力、大容量存储（例如通过数据中心）、丰富的应用服务（如Web搜索——Google依赖大容量集群计算机体系结构来实现其搜索引擎和其他服务）。  网格计算 |

构造分布式系统的主要动机：资源共享

计算机支持协同工作（Computer Supported Cooperative Working，CSCW）

服务：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。

服务器：在连网的计算机上的一个运行程序（一个进程），这个程序接收来自其他计算机上正在运行的程序的请求，执行一个服务并适当地做出响应。

客户：发出请求的进程。

客户-服务器计算：整个方案。

（客户）调用那个服务器上的操作：；客户发送一个要执行的操作请求。

远程调用：客户和服务器之间的完整交互，即从客户发送一个请求到它接收到服务器的应答。

进程

客户对象

服务器对象

一个客户对象调用了一个服务器对象上的方法：在用面向对象语言实现的分布式系统中，资源被封装成对象，并由客户对象访问。

构造分布式系统的挑战

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 异构性 | 多样性和差别   |  | | --- | | 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：软件层，它提供了一个编程抽象，同时屏蔽了底层网络、硬件、操作系统和编程语言的异构性。  公共对象请求代理（Common Object request Broker，CORBA）  Java远程方法调用（Remote Method Invocation，RMI）  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 | | 分布式事务处理 |   移动代码：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。  虚拟机 |
| 开放性 | 分布式系统的开放性主要取决于新的资源共享服务能被增加和供多种客户程序使用的程度。  发布（关键接口）  征求意见文档（Requests For Comments，RFC）  万维网联盟（World Wide Web Consortium，W3C）  开放的分布式系统：按……的方式支持资源共享的系统。 |
| 安全性 | 信息资源的安全性   |  |  | | --- | --- | | 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 |   没解决的挑战   |  |  | | --- | --- | | 拒绝服务攻击 | 用大量无意义的请求攻击服务，使得重要的用户不能使用它。 | | 移动代码的安全性 |  | |
| 可伸缩性 | （系统是）可伸缩的：资源数量和用户数量激增，系统仍能保持其有效性。  可伸缩分布式系统的设计面临的挑战   |  |  | | --- | --- | | 控制物理资源的开销 | 当对资源的需求增加时，应该可以花费合理的开销扩展系统以满足要求。 | | 控制性能损失 |  | | 防止软件资源用尽 |  | | 避免性能瓶颈 | 算法是分散型  规模问题是分布式系统开发中面临的主要问题。 | |
| 故障处理 | 当硬件或软件发生故障时，程序可能会产生不正确的结果或者在它们完成应该进行的计算之前就停止了。  分布式系统的故障是部分的，也就是说，有些组件出了故障而有些组件运行正常。  处理故障的技术   |  |  | | --- | --- | | 检测故障 | 面临的挑战：如何在有故障出现的情况下进行管理，这些故障不能被检测到但可以被猜到。 | | 掩盖故障 | 有些被检测到的故障能被隐藏起来或降低它的严重程度。 | | 容错 |  | | 故障恢复 |  | | 冗余 | 利用冗余组件，服务可以实现容错。 | |
| 并发性 | 在分布式系统中，代表共享资源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。 |
| 透明性 | 迁移透明性（被移动透明性替换）   |  |  | | --- | --- | | 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（例如，哪个建筑物或IP地址）就能够访问它们。 | | 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需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。 |   网络透明性：最重要的两个透明性是访问透明性和位置透明性，它们的有无对分布式资源的利用有很大影响。 |
| 服务质量 | 系统主要的非功能特性，即影响客户和用户体验的服务质量是   |  | | --- | | 可靠性 | | 安全性 | | 性能：满足及时性保证的能力。  源于及时性和计算吞吐量 |   适应性：满足变化的系统配置和资源可用性。  时间关键性数据：要求以固定速度处理或从一个进程传送到另一个进程的数据流。  QoS（Quality of Service，服务质量）：系统满足……的截止时间的能力，它的实现取决于所需要的计算和网络资源在相应时刻的可用性。 |

万维网

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 超文本  链接（超链接）  开放  内容  Web基于的主要的标准技术组件   |  |  |  | | --- | --- | --- | | 超文本标记语言（HyperText Markup Language，HTML） | 页面在Web浏览器上显示时指定其内容和布局的语言。 |  | | 统一资源定位器（Uniform Resource Locator，URL）（统一资源标识符（Uniform Resource Identifier，URI）） | 用于识别文档和保存成Web一部分的其他资源。 | URL结构：   |  | | --- | | 模式:模式特定的标识符 |   HTTP URL结构：   |  | | --- | | http://服务器名[:端口][/路径名][?查询][#片段] |   服务器名表示成一个域名系统（Domain Name System，DNS）名字。  默认端口号是80。  查询成分  片段标识符  发布资源  内容管理系统 | | 具有标准交互规则（超文本传输协议，HTTP）的客户-服务器系统体系结构 |  | 主要特征   |  |  | | --- | --- | | 请求-应答交互 | 方法 | | 内容类型 |  | | 一次请求一个资源 |  | | 简单的访问控制 |  | |   动态页面   |  | | --- | | 表单  （服务器）处理（用户的输入）  公共网关接口（Common Gateway Interface，CGI）程序：Web服务器上运行的为客户生成内容的程序。  下载的代码  异步  AJAX（Asynchronous Javascript And XML） |   Web服务   |  | | --- | | 可扩展标记语言（Extensible Markup Language，XML）是一种以标准的、结构化的、特定于应用的格式表示数据的方式。  自描述的：包含数据元素的名字、类型和结构。  REST（Representational State Transfer）体系结构 |   对Web的讨论   |  | | --- | | 资源描述框架  语义Web：由互连的元数据资源组成的Web。 | |