



7

SOA软件架构

7.1 分布式系统概述

- 定义

分布式系统属组件分布在网络计算机上且通过消息传递进行通信和动作协调的系统。

- 分布式系统特征

并发性、缺乏全局时钟、故障独立性。

构造和使用分布式系统的主要动力来源是资源共享。

7.1 分布式系统概述

- 并发性

在一个计算机网络中，程序并发执行，在需要时共享资源，系统处理共享资源的能力应该随着网络资源的增加而增加；如何协调并发执行的共享资源的企业程序是一个重要问题。

- 缺乏全局时钟

因为网络上计算机同步时钟的准确性受到限制，所以程序需要协调时仅能通过交换消息来协调它们的动作；通过网络发送消息作为唯一通信方式的直接结果，同步是重要的问题。

- 故障独立性

所有计算机都可能发生故障；

网络故障导致与之互联的计算机的隔离。计算机中程序无法检测网络出现了故障还是网络运行的比通常慢；

计算机的故障或系统中程序的一场终止并不能立即被与之通信的组件感知，系统中每个组件对单独地出现故障，而其他组件还在运行。

7.1 分布式系统概述

- 分布式系统实例
- **因特网**是一个巨大的多种类型计算机网络的互连集合。因特网上计算机程序通过传递消息交互，采用了一种公共通信手段。因特网协议是一项重大的技术成果，它使得一个在运行的程序能给在任何地方的程序发送消息。
- **企业内部网**是因特网的一个部分，实现独立管理，具有边界，通过配置能够执行本地安全策略。防火墙作用是防止未授权消息的发出和进入来保护企业内部网，防护墙是通过过滤进出消息实现其功能。
- **移动计算**[Kleinrock1997]是指用户在移动中执行计算任务的能力或访问他们所处环境以外的位置的能力。

- 技术挑战

- 异构性
- 开放性
- 安全性
- 并发性
- 可扩展性
- 故障与容错

- IBM定义

- Web服务是一种接口，它描述了一组操作，可以使用标准的XML消息传递技术，经由网络访问这些操作。Web服务可完成指定的任务或任务组。
- Web服务采用标准的、形式化的XML概念进行描述，称之为服务描述。服务描述提供了与该服务进行交互所必须的所有细节，包括消息格式(即操作的细节)、传输协议和位置。
- 接口的实质是隐去了服务的实现细节，以致于它的使用可以独立于实现它的硬件或软件平台，独立于用来编写它的编程语言。允许并促进了基于Web服务的应用成为松散耦合的、面向构件的、跨技术的实现。

- Microsoft定义

Web服务是一个向其他应用提供数据和服务的应用逻辑单元。应用程序通过无处不在的Web协议和数据格式访问Web服务，如HTTP、XML和SOAP，而无需关心每个Web服务是如何实现的。

- SUN定义

Web服务是软件构件。这类构件具有被发现、可组合和重组的特性，用于解决用户的问题或要求，Java语言和XML是Web服务的最重要技术。

7.2 Web服务简介

从业务、商务、技术视角看

- 业务视角

Web服务就是关于集成的技术，即集成同一机构内的应用功能，或集成商务伙伴之间的应用。

- 商务视角

Web服务是一商务过程或是商务过程中的步骤。内部的或外部的商务伙伴都可经由网络使用这项服务，以达到某种业务目标。

- 技术视角

Web服务只是一个或多个相关操作的集合，这些操作可以由网络访问，并可以用服务描述来描述。

7.3 面向服务的体系结构

- TCP / IP：作为网络开放平台能够支持诸如客户/服务器计算的体系结构；
- 因特网：基于HTML和HTTP，它提供了第一个真正全球开放的、可移植的用户界面；
- Java：实现开放和可移植的编程；
- XML：实现开放和可移植的数据交换；
- web服务：实现基于开放式标准的广泛集成。

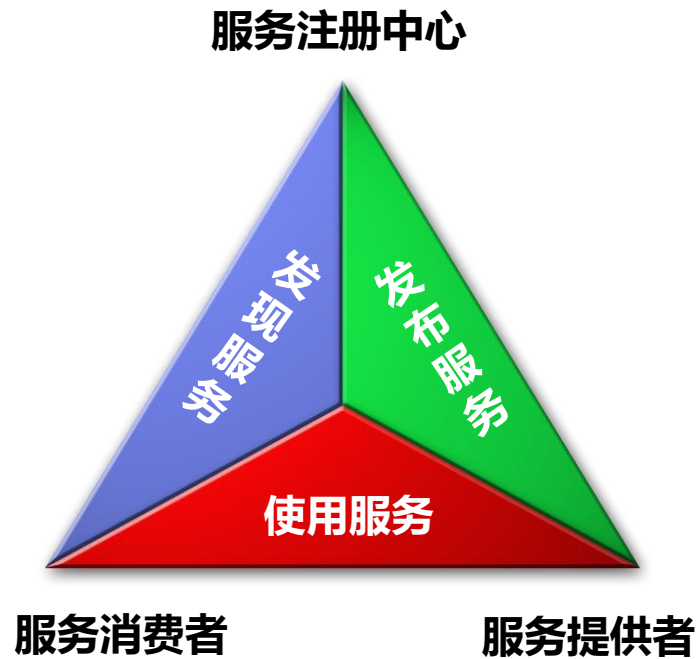
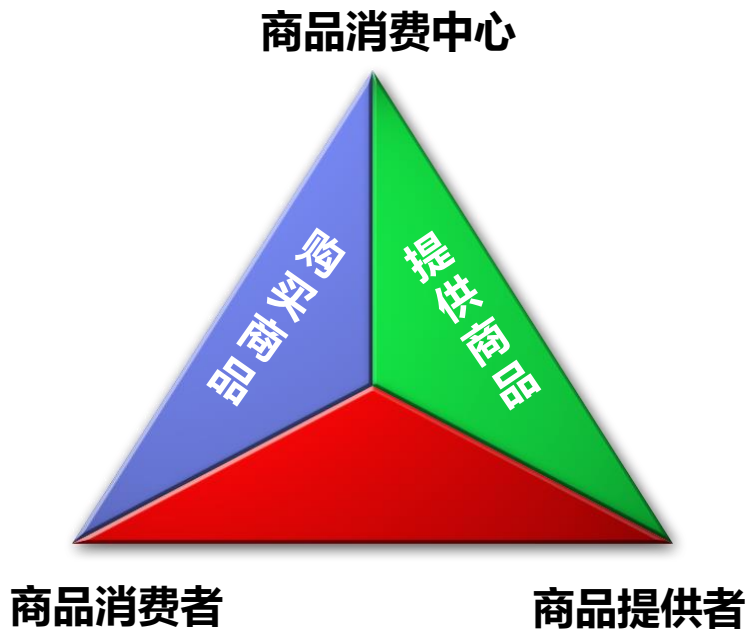
7.3 面向服务的体系结构

- 软件应用形态



7.3 面向服务的体系结构

- 商品消费—软件服务



7.3 面向服务的体系结构

- 服务提供者

一个可通过网络寻址的实体，它接受和执行来自使用者的请求。它将自己的服务和接口契约发布到服务注册中心，以便服务使用者可以发现和访问该服务。

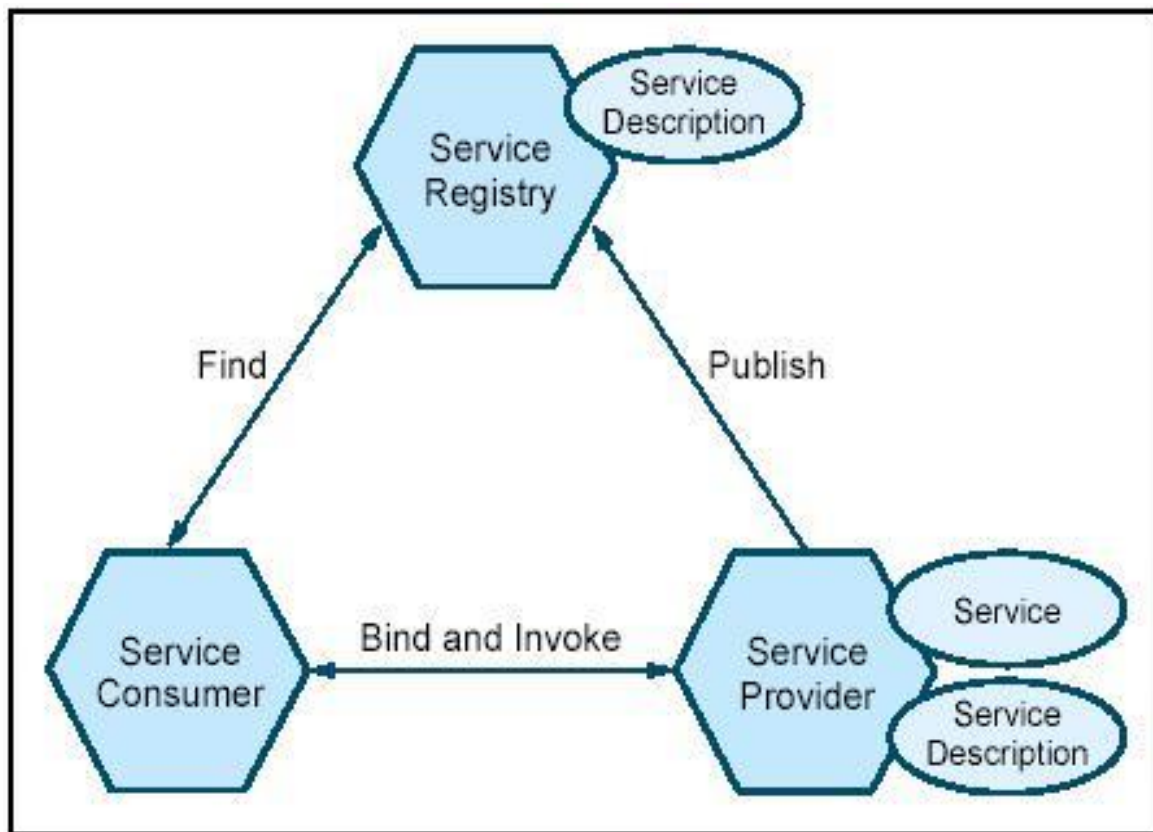
- 服务使用者

一个应用程序、一个软件模块或需要一个服务的另一个服务。它发起对注册中心中的服务的查询，通过传输绑定服务，并且执行服务功能。服务使用者根据接口契约来执行服务。

- 服务注册中心

服务发现的支持者；它包含一个可用服务的存储库，并允许感兴趣的服务使用者查找服务提供者接口。

7.3 面向服务的体系结构



7.4 面向服务的体系结构

- 面向服务的体系结构中的每个实体都扮演着服务提供者、使用者和注册中心这三种角色中的某一种（或多种）。
- 面向服务的体系结构中的操作包括：
 - **发布(Publish)**: 为了使服务可访问，需要发布服务描述以使服务使用者可以发现它。
 - **发现(Find)**: 服务请求者定位服务，方法是查询服务注册中心来找到满足其标准的服务。
 - **绑定(Bind)和调用(invoke)**: 在检索到服务描述之后，服务使用者继续根据服务描述中的信息来调用服务。

7.4 面向服务架构SOA

- 定义

- 一种体系结构风格，它将应用程序的不同功能单元—服务(service)，通过服务间定义良好的接口和契约(contract)联系起来。
- 接口采用中立的方式定义，独立于具体实现服务的硬件平台、操作系统和编程语言，使得构建的系统中的服务可以使用统一和标准的方式进行通信。这种具有中立的接口定义的特征称为服务之间的**松耦合**。

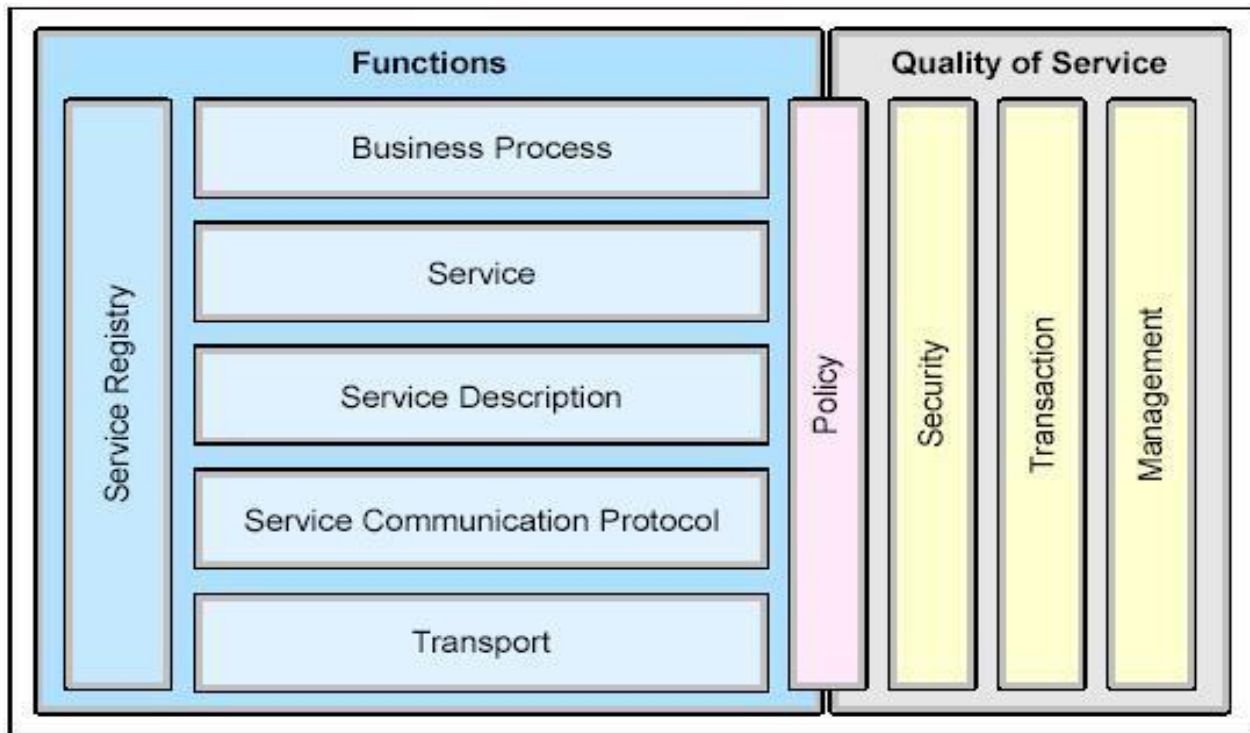
SOA 的体系架构指导描述了体系结构的式样、要素和关系，以及在整个开发生命周期中他们是如何被识别、指定和管理的。通常体系结构的目标，特别是体系结构建模，提供了一个适当的抽象级别，在此级别上，可以容易地识别体系结构的要素，并且对要素进行控制。

7.4 面向服务架构SOA

- Web Services的目标是即时装配、松散耦合以及自动集成。
- 服务 (service) : 整个SOA实现的核心。SOA的基本元素是服务, SOA 指定一组实体(服务提供者、服务消费者、服务注册表、服务条款、服务代理和服务契约), 这些实体详细说明了如何提供和消费服务。这些服务是可互操作的、独立的、模块化的、位置明确的、松耦合的并且可以通过网络查找其地址。

- 服务规范有三个规范要素，根据服务的建模类型可对它们进行选择：
- **结构规范**定义了可以调用的操作和由这些操作销毁或创造出的消息。
- **行为规范**表示服务客户和所指定服务之间的任意预期的有意义的协议或会话。
- **策略规范**表示服务的策略主张和约束。策略主张可能包括安全性、可管理性等等。

- SOA协议栈



7.4 面向服务架构SOA

- 功能性方面—服务
- **传输协议**：用于将来自服务使用者的服务请求传送给服务提供者，并且将来自服务提供者的响应传送给服务使用者。
- **通信协议**：一种经过协商的机制，通过这种机制，服务提供者和服务使用者可以就将要请求的内容和将要返回的内容进行沟通。
- **服务描述**：一种经过协商的模式，用于描述服务是什么、应该如何调用服务以及成功地调用服务需要什么数据。
- **服务**：描述供使用的操作。

- 功能性方面—流程
- 业务流程

一个服务的集合，可以按照特定的顺序并使用一组特定的规则进行调用，以满足业务要求；

业务流程本身也看作是服务，这样就产生了业务流程可以由不同粒度的服务组成的观念。

- 功能性方面—服务注册中心
- 服务注册中心

一个服务和数据描述的存储库；

服务提供者可以通过服务注册中心发布它们的服务，而服务使用者可以通过服务注册中心发现或查找可用的服务；

服务注册中心可以给需要集中式存储库的服务提供其他的功能。

服务质量方面

- 策略是规则集

在这些条件和规则之下，服务提供者可以使服务可用于使用者。

策略既有功能性方面，也有与服务质量有关的方面；因此，在功能和服务质量两个部分中都有策略功能。

- 安全性是规则集：应用于调用服务的服务使用者的身份验证、授权和访问控制。
- 事务是属性集：应用于一组服务，以提供一致的结果。如果要使用一组服务来完成一项业务功能，则所有的服务必须都完成，或者没有一个完成。
- 管理是属性集：应用于管理提供的服务或使用的服务。

Web服务策略：提供使 Web 服务应用程序能够指定策略信息所需的机制

- 框架

提供了一种灵活且可扩展的语法，用于表达基于 XML Web 服务的系统中实体的功能、要求和一般特性。定义了一个框架和一个模型，用于将这些属性表达为策略，策略定义为一组策略替换选项，其中每个策略替换选项又是一组策略断言。某些策略断言指定了一些传统的要求和功能，这些要求和功能最终将出现在网络中（如身份验证方案、传输协议选择）。

- Web 服务安全性基本要求
 - **机密性(Confidentiality)**是指信息对没有经过授权的个人、实体或进程的不可用性或不公开性，并保证消息内容不对没有经过授权的个人公开。
 - **授权(Authorization)**是指权限的授予，包括根据访问权限授予访问权和保证发送方被授权发送消息。
 - **数据完整性(Data integrity)**是指数据没有以未经授权的方式或被未经授权的用户不可察觉的改变或者破坏的性质，从而确保消息在传送的过程中不会被偶然或故意修改

- Web 服务中的 QoS 的主要需求
 - 可用性
 - 可靠性
 - 完整性
 - 安全性
 - 常规性

7.5 Web服务

- 性能：指根据吞吐量和延迟对其进行测量。吞吐量的值较大且延迟的值较小表示 Web 服务性能良好。吞吐量表示在给定时间段内被服务的 Web 服务请求数。延迟是发送请求和接收响应之间的往返时间。
- 可用性：指 Web 服务是否存在或是否已就绪可供立即使用。可用性表示服务可用的可能性。较大的值表示服务一直可供使用，而较小的值表示无法预知在某个特定时刻服务是否可用。与可用性有关的还有修复时间(time-to-repair, TTR)。TTR表示修复已经失效的服务要花费的时间。理想情况下，较小的 TTR 值是合乎需要的。

- 可靠性：指表示能够维护服务和服务质量的程度。每月或每年的失效次数是衡量 Web 服务可靠性的尺度。在另一种意义上，可靠性是指服务请求者和服务提供者发送和接收的消息的有保证和有序的传送。
- 完整性：指 Web 服务如何维护交互相对于最初情况的正确性。适当地执行 Web 服务事务会实现正确的交互。一个事务是指一系列将被当作单个工作单元的活动。要使事务成功，必须完成所有的活动。如果一个事务未完成，那么所做的全部更改都被回滚。

- 安全性：指通过验证涉及到的各方、对消息加密以及提供访问控制来提供机密性和不可抵赖性。由于 Web 服务调用是发生在公共的因特网上，安全性的重要性已经有所增加。根据服务请求者的不同，服务提供者可以用不同的方法来提供安全性，所提供的安全性也可以有不同的级别。
- 常规性：指 Web 服务与规则、法律一致，遵循标准和已建立的服务级别协议。Web 服务使用许多标准，如SOAP、UDDI 和 WSDL。要正确调用服务请求者请求的服务，就必须严格遵守服务提供者所提供的正确版本的标准,例如，SOAP 版本 1.2。

- 面向服务体系结构特征
 - 自包含和模块化
 - 互操作性
 - 松散耦合
 - 位置透明
 - 可组合性
 - 明确定义的接口

7.6 SOA基本特征

- 服务的封装(encapsulation)

将服务封装成用于业务流程的可重用组件的应用程序函数。它提供信息或简化业务数据从一个有效的、一致的状态向另一个状态的转变。封装隐藏了复杂性。服务的API保持不变，使得用户远离具体实施上的变更。

- 服务的重用(reuse)

服务的可重用性设计显著地降低了成本。为了实现可重用性，服务只工作在特定处理过程的上下文(context)中，独立于底层实现和客户需求的变更。

- 服务的互操作(interoperability)

通过服务之间既定的通信协议进行互操作。主要有同步和异步两种通信机制。SOA提供服务的互操作特性更利于其在多个场合被重用。

7.6 SOA基本特征

- 服务是自治的(Autonomous)功能实体

服务是由组件组成的组合模块，是自包含和模块化的。SOA强调提供服务的功能实体的完全独立自主的能力。SOA强调实体自我管理和恢复能力。

常见恢复的技术，如事务处理(Transaction)，冗余部署(Redundant Deployment)和集群系统(Cluster)。

- 服务之间的松耦合度(Loosly Coupled)

服务请求者到服务提供者的绑定与服务之间应该是松耦合的。服务请求者不知道提供者实现的技术细节，比如程序设计语言、部署平台等等。服务请求者往往通过消息调用操作，请求消息和响应，而不是通过使用 API 和文件格式。

7.6 SOA基本特征

- 服务是位置透明的(location transparency)

服务是针对业务需求设计的。需要反应需求的变化，即所谓敏捷（agility）设计。实现业务与服务分离，就必须使得服务的设计和部署对用户来说是完全透明的。

- 明确定义的接口(well defined interface)

Web服务使应用功能得以通过标准化接口（WSDL）提供，并可基于标准化传输方式（HTTP和JMS）、采用标准化协议（SOAP）进行调用。



End of Session
Thank you!