

# 分布式与云计算一期末考试

## 1、ACID & CAP & BASE 模型

ACID 特性对于大型的分布式系统来说，强调一致性。而 BASE 模型是传统 ACID 模型的反面，不同与 ACID，BASE 强调牺牲高一致性，从而获得可用性，数据允许在一段时间内的不一致，只要保证最终一致就可以了。

### 1.1.1、 ACID

事务正确执行的四个基本要素

- [1]. **原子性 (Atomicity)**: 全有或全无
- [2]. **一致性 (Consistency)**: 从一个状态转化到另一状态
- [3]. **隔离性 (Isolation)**: 不受其他事务影响，中间过程不可见
- [4]. **持久性 (Durability)**: 一旦完成，结果将永久存储

### 1.1.2、 CAP

指的是在一个分布式系统中，一致性，可用性，分区容错性三者不可兼得。

- [1]. **一致性 (Consistency)**: 数据一致更新，所有数据变动都是同步的
- [2]. **可用性 (Availability)**: 在集群中一部分节点故障后，集群整体是否还能响应客户端的读写请求，即好的响应性能。
- [3]. **分区容错性 (Partition tolerance)**: 以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在 C 和 A 之间做出选择。

### 1.1.3、 BASE

BASE 理论是对 CAP 理论的延伸，核心思想是即使无法做到强一致性（CAP 的一致性就是强一致性），但应用可以采用适合的方式达到最终一致性

- [1]. **基本可用 (Basically Available)**: 基本可用是指分布式系统在出现故障的时候，允许损失部分可用性，即保证核心可用。
- [2]. **软状态 (Soft state)**: 状态可以有一段时间不同步，异步。
- [3]. **最终一致 (Eventually consistent)**: 最终数据是一致的就可以了，而不是时时一致。

## 2、两阶段提交协议

二阶段提交（Two-phase Commit）是为了使基于分布式系统架构下的所有节点在进行事务提交时保持一致性而设计的一种算法。通常，二阶段提交也被称为是一种协议。在分布式系统中，每个节点虽然可以知晓自己的操作时成功或者失败，却无法知道其他节点的操作的成功或失败。当一个事务跨越多个节点时，为了保持事务的 ACID 特性，需要引入一个作为**协调者**的组件来统一掌控所有节点（称作**参与者**）的操作结果并最终指示这些节点是否要把操作结果进行真正的提交。因此，二阶段提交的算法思路可以概括为：**参与者将操作成败通知协调者，再由协调者根据所有参与者的反馈情报决定各**

**参与者是否要提交操作还是中止操作。**

所谓的两个阶段是指：第一阶段：**准备阶段（投票阶段）**和第二阶段：**提交阶段（执行阶段）**。

## **准备阶段**

事务协调者给每个参与者发送 **Prepare** 消息，每个参与者要么直接返回失败(如权限验证失败)，要么在本地执行事务，写本地的 **redo** 和 **undo** 日志，但不提交，到达一种“万事俱备，只欠东风”的状态。

可以进一步将准备阶段分为以下三个步骤：

- [1]. 协调者节点向所有参与者节点询问是否可以执行提交操作(**vote**)，并开始等待各参与者节点的响应。
- [2]. 参与者节点执行询问发起为止的所有事务操作，并将 **Undo** 信息和 **Redo** 信息写入日志。（注意：若成功这里其实每个参与者已经执行了事务操作）
- [3]. 各参与者节点响应协调者节点发起的询问。如果参与者节点的事务操作实际执行成功，则它返回一个“同意”消息；如果参与者节点的事务操作实际执行失败，则它返回一个“中止”消息。

## **提交阶段**

如果协调者收到了参与者的失败消息或者超时，直接给每个参与者发送回滚(**Rollback**)消息；否则，发送提交(**Commit**)消息；参与者根据协调者的指令执行提交或者回滚操作，释放所有事务处理过程中使用的锁资源。(注意:必须在最后阶段释放锁资源)

接下来分两种情况分别讨论提交阶段的过程。

1. 协调者节点从所有参与者节点获得的相应消息都为“同意”时：

- [1]. 协调者节点向所有参与者节点发出“正式提交(**commit**)”的请求。
- [2]. 参与者节点正式完成操作，并释放在整个事务期间内占用的资源。
- [3]. 参与者节点向协调者节点发送“完成”消息。
- [4]. 协调者节点受到所有参与者节点反馈的“完成”消息后，完成事务。

2. 如果任何参与者节点在第一阶段返回的响应消息为“中止”，或者协调者节点在第一阶段的询问超时之前无法获取所有参与者节点的响应消息时：

- [1]. 协调者节点向所有参与者节点发出“回滚操作(**rollback**)”的请求。
- [2]. 参与者节点利用之前写入的 **Undo** 信息执行回滚，并释放在整个事务期间内占用的资源。
- [3]. 参与者节点向协调者节点发送“回滚完成”消息。
- [4]. 协调者节点受到所有参与者节点反馈的“回滚完成”消息后，取消事务。

**不管最后结果如何，第二阶段都会结束当前事务。**

二阶段提交看起来确实能够提供原子性的操作，但是不幸的事，二阶段提交还是有几个缺点的：

- [1]. **同步阻塞问题**。执行过程中，所有参与节点都是事务阻塞型的。当参与者占有公共资源时，其他第三方节点访问公共资源不得不处于阻塞状态。

- [2]. **单点故障**。由于协调者的重要性，一旦协调者发生故障。参与者会一直阻塞下去。尤其在第二阶段，协调者发生故障，那么所有的参与者还都处于锁定事务资源的状态中，而无法继续完成事务操作。（如果是协调者挂掉，可以重新选举一个协调者，但是无法解决因为协调者宕机导致的参与者处于阻塞状态的问题）
- [3]. **数据不一致**。在二阶段提交的阶段二中，当协调者向参与者发送 `commit` 请求之后，发生了局部网络异常或者在发送 `commit` 请求过程中协调者发生了故障，这回导致只有一部分参与者接受到了 `commit` 请求。而在这部分参与者接到 `commit` 请求之后就会执行 `commit` 操作。但是其他部分未接到 `commit` 请求的机器则无法执行事务提交。于是整个分布式系统便出现了数据不一致性的现象。

### 3、Raft（最后一道大题）

观看动画 <http://thesecretlivesofdata.com/raft/>

### 4、云计算

云计算是一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池（资源包括网络，服务器，存储，应用软件，服务），这些资源能够被快速提供，只需投入很少的管理工作，或服务供应商进行很少的交互。

#### IaaS 基础设施即服务

将硬件设备等基础资源，包括计算、存储和网络等，封装成服务供用户使用，消费者通过网络可以从完善的计算机基础设施获得服务。

典型的如亚马逊的弹性计算云 EC2（Elastic Compute Cloud）和简单存储服务 S3（Simple Storage Service）。相较于传统的用户自行购置硬件的使用方式，IaaS 允许用户按需使用硬件资源，并且按量计费。从服务使用者的角度看，IaaS 的服务器规模巨大，用户能够申请的资源几乎是无限的；从服务提供者的角度看，IaaS 同时为多个用户提供服务，因而具有更高的资源利用率。

#### PaaS 平台即服务

是指将一个完整的应用开发平台，包括应用设计、应用开发、应用测试和应用托管，都作为一种服务提供给客户。在这种服务模式中，客户不需要购买硬件和软件，只需要利用 PaaS 平台，就能够创建、测试和部署应用和服务。

例如，Google App Engine 只允许使用 Python 和 Java 语言、基于称作 Django 的 Web 应用框架、调用 Google App Engine SDK 来开发在线应用服务。

#### SaaS 软件即服务

指将某些特定应用软件功能封装成服务，它是一种通过网络提供软件的模式，用户无需购买软件，而是向提供商租用基于 Web 的软件，来管理企业经营活动。如 Salesforce 公司提供的在线客户关系管理 CRM 服务。SaaS 既不像 IaaS 一样提供计算或存储资源类型的服务，也不像 PaaS 一样提供运行用户自定义应用程序的环境，它只提供某些专门用途的服务调用。

## 5、CDN 的工作原理

CDN 是构建在网络之上的内容分发网络，依靠部署在各地的边缘服务器，通过中心平台的负载均衡、内容分发、调度等功能模块，使用户就近获取所需内容，降低网络拥塞，提高用户访问响应速度和命中率。CDN 的关键技术主要有内容存储和分发技术。

CDN 的基本原理是广泛采用各种缓存服务器，将这些缓存服务器分布到用户访问相对集中的地区或网络中，在用户访问网站时，利用全局负载技术将用户的访问指向距离最近的工作正常的缓存服务器上，由缓存服务器直接响应用户请求。

## 6、P2P 按网络拓扑结构划分可分为哪几类，并简述每类的工作原理。

分为四类：

- [1]. **中心化拓扑**：是网络中资源和服务的索引都存储在中心服务器目录中，而资源和服务本身则存储在网络中的各结点中。Napster 是典型的 P2P 中心化拓扑结构系统
- [2]. **全分布式非结构化拓扑**：全分布式非结构化拓扑结构的 P2P 网络是一种重叠网络。重叠网络是在现有的网络体系架构上新加一层虚拟网络，并将虚拟网络中的每个结点与实际网络中的一些结点相连，从而实现与实际网络中各结点的联通。
- [3]. **全分布式结构化拓扑**：全分布式结构化拓扑采用分布式散列表(Distributed Hash Table, 简称 DHT) 来组织网络中的各结点，因此该拓扑结构的网络也称为 DHT 网络。DHT 网络是一个由广域范围大量结点共同维护的庞大散列表，散列表被分割成不连续的块，每个结点被分配一个属于自己的散列表块，并成为这个散列表块的管理者。典型的 DHT 网络案例有 Tapestry、Pastry、Chord、CAN 等
- [4]. **半分布式拓扑**：半分布式拓扑选择了性能较高的结点作为超级结点，在各个超级结点上存储其他部分结点的信息，检索算法仅在超级结点间转发，超级结点再将查询请求转发给适当的叶子结点。该拓扑结构的网络系统有较好的性能和扩展性、且支持复杂查询、易于管理维护。KaZaA 就是一款典型的半分布式拓扑的 P2P 文件共享软件。

## 7、分布式系统中间件的主要功能是什么？常见的中间件的技术有哪些？

中间件是指一个软件层、它提供了一个编程对象、同时屏蔽了底层网络、硬件、操作系统和编程语言的异构性。中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信。中间件在操作系统、网络和数据库之上，应用软件的下层，总的作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户灵活、高效地开发和集成复杂的应用软件。

常见的中间件技术有 OMG 组织的 Corba，微软的 COM，IBM 的 SOM, zookeeper, Apache Kafka, Facebook Thrift, Java RMI。

## 8、什么是分布式系统？分布式系统的构造面临哪些挑战？

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。具有组件的并发性、缺乏全局时钟、组件故障的独立性等特征。构造分布式系统的挑战是处理其组件

的异构性、开放性（允许增加和替换组件）、安全性、可伸缩性（用户数量增加时能正常运行的能力）、故障处理、组件的并发性和透明性问题。

## 9、私有云和共有云的区别以及各自服务的领域是什么？

私有云和公有云的区别：对数据的掌控程度不同。采用公有云服务的企业必须将数据托管于云服务商的数据中心，企业对数据的掌控力度自然减弱。一旦数据中心因自然灾害、人为因素或法律规范等各方面因素导致数据丢失，将对企业形成致命伤害。私有云由企业负责设置和维护云，在数据安全、数据备份等方面也有更多的可选择空间。公有云当然也具备数据安全服务和数据备份能力，但企业对此的控制力度较差，不能处于主导地位。

私有云和公有云各自服务的领域：公有云通过互联网为客户提供服务的云，所有的基础设施均由云服务提供商负责，用户只需能够接入网络的终端即可。无需做相应的投资和建设。而私有云是企业使用自有基础设施构建的云，它提供的服务仅供自己内部人员或分支机构使用。私有云的部署比较适合于有众多分支机构的大型企业或政府部门。大型企业数据中心的集中化趋势日益明显，私有云将会成为企业部署 IT 系统的主流模式。

## 10、云可以为企业带来哪些具体收益？请举例说明。

云计算的诸多潜在效益已经明朗。其**价格优势**颇具诱惑力。例如，美国礼来公司选择了亚马逊网络服务对某个开发中的药剂进行数据分析，而仅仅为此支付了 89 美元。如果由礼来公司独立完成这项工作，其研究人员将必须购买 25 台服务器<sup>8</sup>。通过削减服务器、软件许可证、维护、数据中心空间、电力和 IT 劳力等成本来节省开支，使用低廉的、按使用付费的支出方式来取代高昂的资产投资以降低成本，云计算在财务方面具有得天独厚的吸引力。

云也能提供**极为灵活的资源**。其独特的技术设计带来了无限的可扩展性。云可以根据需要快速聚集，并通过向某项任务分派多个服务器而逐步壮大，并且可以在不需要时收缩或消失。由此，云可适应零星的、周期性或临时性的工作，能够快速完成海量数据处理任务，以及执行软件研发与测试项目。在传统系统的计算需求超过其供应能力时，云也可以为其提供增补支持。作为一项运营开支，云服务可以避免资本支出审批流程，因此能够早于传统系统获得资金支持。以礼来公司为例，使用云避免了 3 个月的 IT 预算及其他相关审批流程。

## 11、分布式计算和网格计算的相同点和不同点？分布式计算和并行计算的优缺点？

分布式计算和网格计算的相同点和不同点：

### 相同点

都是利用网络把成千上万台计算机连接起来，组成一台虚拟的超级计算机，完成单台计算机无法完成的超大规模的问题求解。把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算机进行处理，最后把这些计算结果综合起来得到最终的结果

### 不同点

从云计算的描述可知，所谓的云计算可以被看成是网格计算和虚拟化技术的融合：即利用网格分布式计算处理的能力，将 IT 资源构筑成 1 个资源池，再加上成熟的服务器虚拟化、存储虚拟化技术，使用户可以实时地监控和调配资源。可以说云计算的概念涵盖了网格计算，并且加上了更多企业级安全的因素。

### **分布式计算和并行计算的优缺点：**

#### **优点：**

- a) 低廉的计算机价格和网络访问的可用性。
- b) 资源共享：分布式计算体系反映了计算结构的现代组织形式，每个组织在面向网络提供共享的资源。
- c) 可伸缩性：分布式计算有良好的伸缩性，对资源需求的增加可通过提供额外资源来有效解决。
- d) 容错性：可以通过资源复制（或镜像）维持故障情形下的可用性，与单机比较，分布式计算提供了容错功能。

#### **缺点：**

- a) 多点故障：因涉及多台计算机，且依赖于网络进行通信。
- b) 安全性能较差

## **12、 网格体系结构的基本要求包括哪些内容？**

网格体系结构的基本要求包括：

- a) 为应用提供单一分布资源全局视图和聚集各种分布的自治资源成为一个系统，不能破坏现有系统，不导致应用程序的重新编写；
- b) 不同组织之间的资源共享互操作是必不可少的，可以对资源直接访问而不单单是文件共享和交换，具体体现在各种资源构件之间和机制之间；
- c) 有一套标准的协议是至关重要的，在遵守标准协议的前提下，才能使允许跨节点的协同工作、允许基础下部组织共享成为可能；没有标准的协议，互操作和共享很难实现，重复开发和安装不可避免；
- d) 从 Grid 实现者的角度来看，网格体系结构是一个层次型中间件服务集合，按照现代软件系统的设计原则，以服务的方式为应用工具和应用的开发提供支持，而服务的实现必须能够自由地使用各个层次上的功能和服务进行构造。

## **13、 请阐述面向服务的体系结构和 Web Service 的关系。**

面向服务的体系结构简称 SOA，从本质上来说 SOA 是一种架构模式，而 Web 服务是利用一组标准实现的服务。SOA 不是 Web Service ,Web Service 是实现 SOA 的方式之一。Web Service 是技术规范，而 SOA 是设计原则。Web Service 中的 WSDL(Web 服务描述语言)，是一个 SOA 配套的接口定义标准，这是 Web Service 和 SOA 的根本联系。

14、 阐述软交换的功能主要表现在哪些方面？举例说明给下一代网络带来哪些优势？

软交换的功能主要表现在以下方面：

- a) 提供支持多种信令协议（包括 H. 248、H. 323、SIP\*、SCTP、ISUP+、INAP+、RADIUS、SNMP）的接口，实现 PSTN 和 IP 网/ATM 网间的信令互通和不同网关的互操作。支持开放的业务/应用接口功能。
- b) 业务提供功能，软交换主要处理实时业务，现阶段是语音，以后将处理视频和多媒体业务。还应该具有利用新的网络服务设施提供各种增值业务和补充业务的能力。
- c) 提供可编程的、逻辑化控制的开放的 API 协议，实现与外部应用平台的互联互通。
- d) 通过不同的逻辑与媒体层的网关交互，对网关设备或 IP/ATM 网的核心设备进行控制，完成融合网络中的呼叫控制，会话的建立、修改和拆除过程以及媒体流的连接机制。
- e) 提供网守功能，即接入认证与授权、地址解析和带宽管理功能。
- f) 资源控制功能和 QoS 管理功能。
- g) 操作维护功能，主要包括业务统计和告警等。
- h) 计费功能，应具有采集详细话单的功能。

给下一代网络带来带来的优势如下：

- a) 业主与呼叫控制分离，呼叫与承载分离：分离的目标是业主真正的独立于网络，灵活有效的实现业务的提供。在业务层用户可自行配置和定义自己的业务特征，不必关心承载业务的网络形式以及终端类型，使得业务和应用的提供有较大的灵活性。
- b) 业务升级与基础设备无关，不需要升级基础设备。
- c) 提供可扩展的业务平台，业务投放周期短。
- d) 提供开发的接口给第三方应用做二次开发。
- e) 地域概念消失，同一个软交换系统的用户可以位于任何一个网络可以到达的地方。

15、 虚拟化技术在云计算中的作用是什么？可以带来哪些方便？

虚拟化技术在云计算中的作用：虚拟化是云计算的基石，在虚拟化与云计算共同构成的一个整体架构中，虚拟化有效地分离了硬件与软件，而云计算则让人们将精力更加集中在软件所提供的服务上。云计算必定是虚拟化的，虚拟化给云计算提供了坚实的基础。

虚拟化技术带来的方便：可降低成本，可帮助企业创建具有高效率和成本效益并且具备自适应能力的 IT 系统，能够自动即时地提供由于业务环境发生变化所需要的新能力。

16、 请阐述网格计算和 Web Service 的关系。

网格计算与 Web Service 之间存在着密切的关系：Web Service 的核心是在大的异构网络上将各种应用连接起来，借助于 Web 标准 UDDI、WSDL 和 XML/SOAP 等将 Internet 从一个通信网络进一步发展到一个应用平台。当越来越多的 Web Service 实现后，应用的各种特征将会随之改变。一些应用就可以根据需要或者是根据可以得到的功能，从可得的服务中来动态构造。

OGSA 吸纳了许多 Web 服务标准，如 Web 服务描述语言（WSDL）、简单对象访问协议（SOAP）、

轻目录访问协议（LDAP）、Web 服务探测（WS-Inspection）等。这些标准用于定位、调度计算资源并保证它们的安全。OGSA 采用 Web 服务框架的优点是：一方面，通过注册和发现接口定义和终端实现描述以及动态产生特定接口绑定的代理，在异构环境中能够支持服务的动态发现和合成。另一方面，由于 Web 服务机制在商业应用领域广泛采用，OGSA 采用 Web 服务框架使人们能够利用许多现有的开发工具和扩展服务，如产生语言绑定的 WSDL 处理器。

有些人认为 Web Service 和 Grid Service 都需要发现服务，进而将服务绑定，以集成到应用之中，于是把 Web Service 和 Grid Service 当成一回事，这是错误的。Web service 强调永久服务的发现和调用，Web service 的服务是无状态的，但在网格环境中，必须也要支持瞬时服务实例，动态地创建和取消。可以说，Grid Service 只是用了 Web Service 的一层外皮。