



可靠性分析设计

51CTO学院：邹月平

 **51CTO学院**

● 系统可靠性

系统可靠性是系统在规定的时间内及规定的环境条件下，完成规定功能的能力，也就是系统无故障运行的概率。根据国家标准《软件工程 产品质量 第1部分：质量模型》（GB/T 16260.1-2006）的规定。

功能性	可靠性	易用性	效率	维护性	可移植性
适合性 准确性 互用性 依从性 安全性	容错性 易恢复性 成熟性	易学性 易理解性 易操作性	时间特性 资源特性	可测试性 可修改性 稳定性 易分析性	适应性 易安装性 一致性 可替换性

● 系统可靠性指标

1. 平均无故障时间

例如，假设同一型号的1000台计算机，在规定的条件下工作1000小时，其中有10台出现故障。这种计算机千小时的可靠度 R 为 $(1000-10)/1000=0.99$ ，失效率为 $10/(1000 \times 1000)=1 \times 10^{-5}$ ， $MTTF=1/(1 \times 10^{-5})=10^5$ 小时。

2、平均故障修复时间

假设修复率(常数)，修复率是指单位时间内可修复系统的平均次数，则：

$$MTTR = 1 / \mu$$

● 系统可靠性指标

3. 平均故障间隔时间

因为两次故障之间必然有修复行为，因此，平均故障间隔时间（Mean Time Between Failure, MTBF）中应包含MTTR。于是有：

$$MTBF = MTTR + MTTF$$

在实际应用中，一般MTTR很小，所以通常认为 $MTBF \approx MTTF$ 。

系统可用性是指在某个给定时间点上系统能够按照需求执行的概率，其定义为：

$$\text{可用性} = MTTF / (MTTR + MTTF) \times 100\% = MTTF / MTBF \times 100\%$$

● 系统可靠性指标

3. 平均故障间隔时间

因为两次故障之间必然有修复行为，因此，平均故障间隔时间（Mean Time Between Failure, MTBF）中应包含MTTR。于是有：

$$MTBF = MTTR + MTTF$$

在实际应用中，一般MTTR很小，所以通常认为 $MTBF \approx MTTF$ 。

系统可用性是指在某个给定时间点上系统能够按照需求执行的概率，其定义为：

$$\text{可用性} = MTTF / (MTTR + MTTF) \times 100\% = MTTF / MTBF \times 100\%$$

● 串联系统可靠性

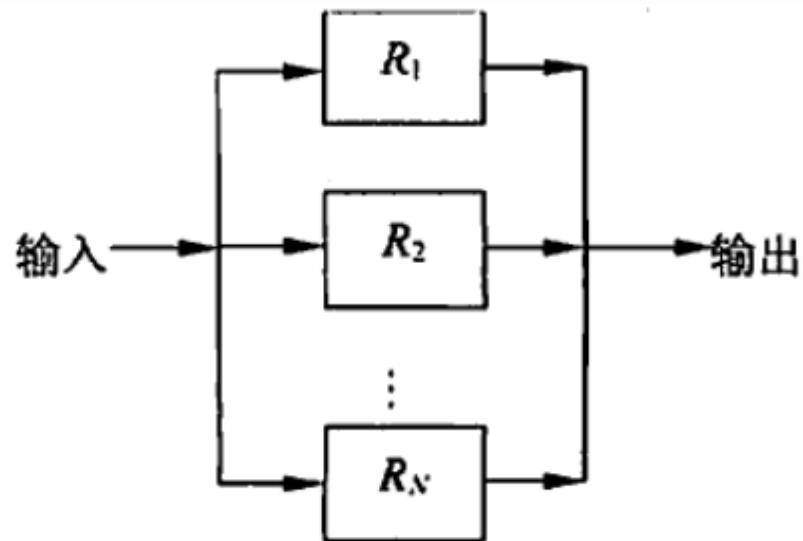
假设一个系统由n个子系统组成，当且仅当所有的子系统都能正常工作时，系统才能正常工作，这种系统称为串联系统。



如果系统的各个子系统的可靠度分别用 R_1, R_2, \dots, R_n 表示，则系统的可靠度为：

$$R = R_1 \times R_2 \times \dots \times R_n$$

● 并联系统可靠性



$$R = 1 - (1 - R_1) \times (1 - R_2) \times \dots \times (1 - R_n)$$

● 串联系统可靠性

假设一个系统由n个子系统组成，当且仅当所有的子系统都能正常工作时，系统才能正常工作，这种系统称为串联系统。



如果系统的各个子系统的可靠度分别用 R_1, R_2, \dots, R_n 表示，则系统的可靠度为：

$$R = R_1 \times R_2 \times \dots \times R_n$$

● 冗余

主要的冗余技术有结构冗余（硬件冗余和软件冗余）、信息冗余、时间冗余和冗余附加四种。

1. 结构冗余

结构冗余是常用的冗余技术，按其工作方式，可分为静态冗余、动态冗余和混合冗余三种。

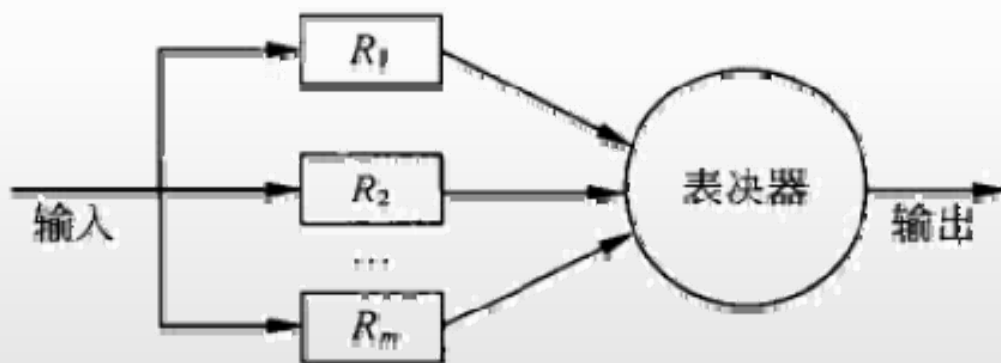
（1）静态冗余。静态冗余又称为屏蔽冗余或被动冗余，常用的有三模冗余和多模冗余。

（2）动态冗余。动态冗余又称为主动冗余，通过多重模块待机储备，当系统检测到某工作模块出现错误时，就用一个备用的模块来顶替它并重新运行。

（3）混合冗余。先静态后动态。

● 冗余

m模冗余系统由m ($m=2n+1$, $n>1$) 个相同的子系统和一个表决器组成，经过表决器表决后，m个子系统中占多数相同结果的输出作为系统的输出。



● 冗余

2. 信息冗余

信息冗余是在实现正常功能所需要的信息外，再添加一些信息，以保证运行结果正确性的方法。例如，检错码和纠错码就是信息冗余。

3. 时间冗余

时间冗余是以时间（即降低系统运行速度）为代价以减少硬件冗余和信息冗余的开销来达到提高可靠性的目的。在某些实际应用中，硬件冗余和信息冗余的成本、体积、功耗、重量等开销可能过高，而时间并不是太重要的因素时，可以使用时间冗余。时间冗余的基本概念是重复多次进行相同的计算，或称为重复执行（复执），以达到故障检测的目的。

● 冗余

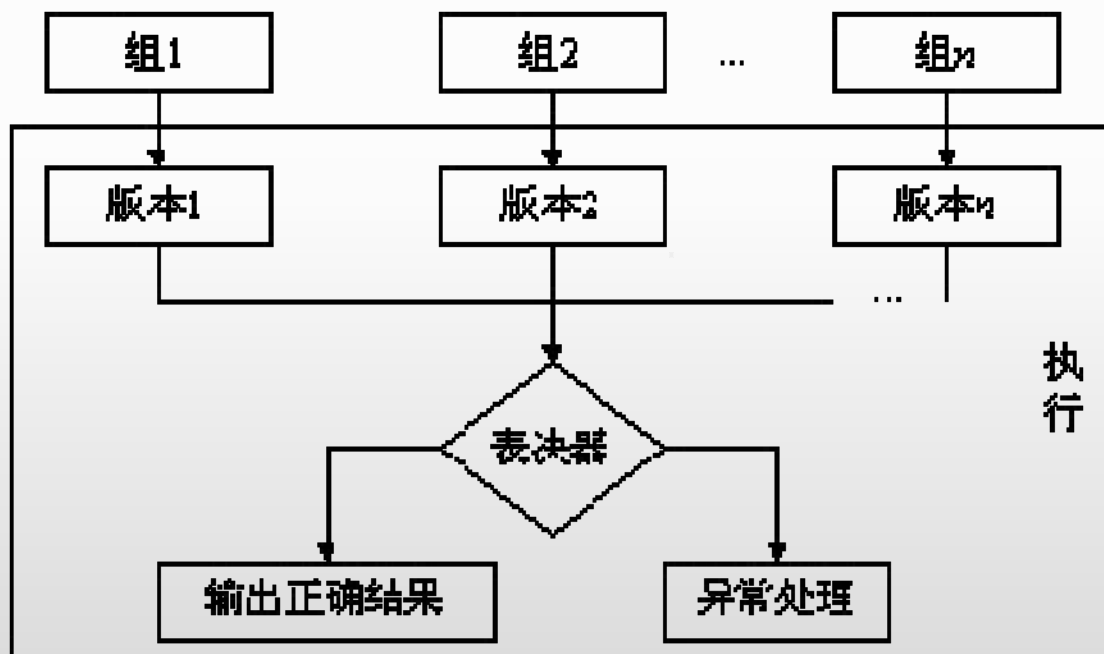
4. 冗余附加

冗余附加是指为实现上述冗余技术所需的资源和技术，包括程序、指令、数据，以及存放和调用它们的空间等。

● 软件容错技术

1、N版本程序设计

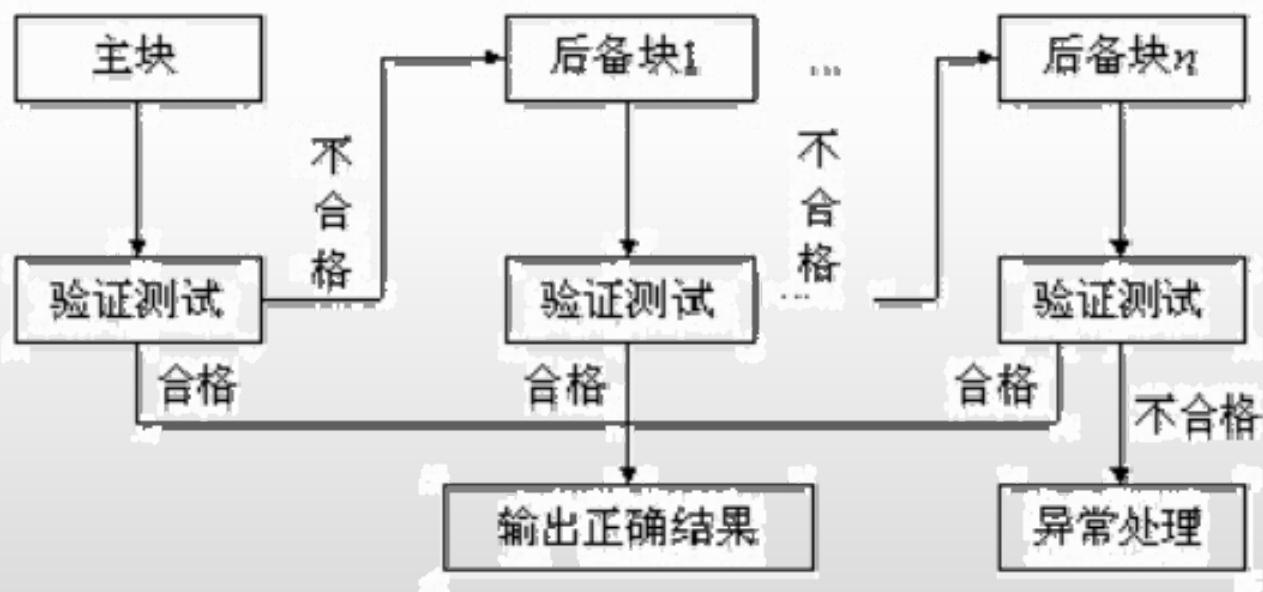
N版本程序设计是一种静态的故障屏蔽技术，采用前向恢复的策略。



● 软件容错技术

2、恢复块方法

恢复块方法是一种动态的故障屏蔽技术，采用后向恢复策略。



● 软件容错技术

3 防卫式程序设计

通过在程序中包含错误检查代码和错误恢复代码，使得一旦发生错误，程序就能撤消错误状态，恢复到一个已知的正确状态中去。

● 双机容错技术

在双机容错系统中，两台服务器一般区分为主系统和从系统（备用系统），两台服务器互为主从关系。每台服务器都有自己的系统盘（本地盘），安装操作系统和应用程序。每台服务器至少安装两块网卡，一块连接到网络上，对外提供服务；另一块与另一台服务器连接，用以侦测对方的工作状况。同时，每台服务器都连接在共享磁盘阵列上，用户数据存放在共享磁盘阵列中，当一台服务器出现故障时，另一台服务器主动替代工作，保证网络服务不间断。整个网络系统的数据通过磁盘阵列集中管理，极大地保护了数据的安全性和保密性。

● 双机容错技术

双机容错系统根据两台服务器的工作方式不同，可以有三种不同的工作模式，分别是双机热备模式、双机互备模式和双机双工模式。

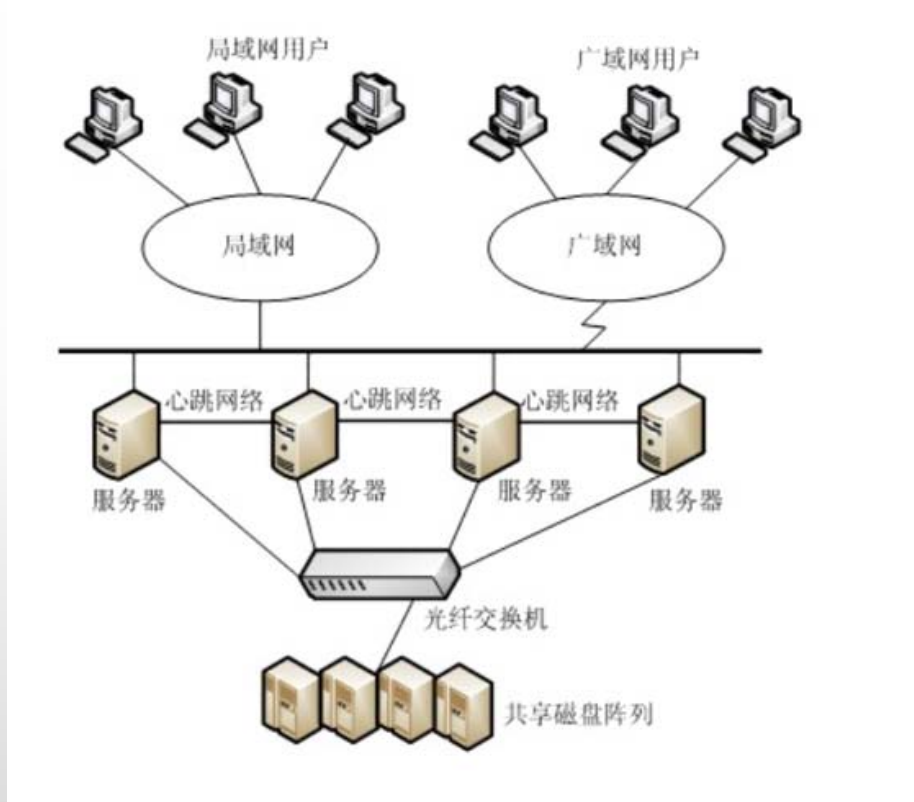
(1) 双机热备模式。正常情况下，一台服务器处于工作状态（主系统），另一台服务器处于监控准备状态（备用系统）。典型应用有证券资金服务器或行情服务器等。双机热备模式的主要缺点在于，备用系统长期处于后备的状态，存在一定的计算资源浪费。

● 双机容错技术

(2) 双机互备模式。两台服务器均处于工作状态，也就是说，两台服务器同时运行，但彼此均设为备用系统。当某一台服务器出现故障时，另一台服务器可以在短时间内将故障服务器的应用接管过来，从而保证了应用的持续性。双机互备模式的主要缺点是对服务器的性能要求比较高。

(3) 双机双工模式。双机双工模式是集群（cluster）技术的一种形式，两台服务器均处于工作状态，同时为前端客户机提供相同的应用服务，以保证整体系统的性能，实现负载均衡和互为备份。

● 集群技术



● 集群技术

(1) 服务器组。在高可用性集群系统中，每个节点的计算机必须有自己的CPU、内存和磁盘，每台计算机都需要安装操作系统和集群软件程序。

(2) 对外提供服务的网络。高可用性集群系统中的计算机一般采用TCP/IP网络协议与客户端相连，每台计算机上都有自己的应用服务，客户端必须通过集群计算机中的网络通路来得到自己的服务。

(3) 心跳信号通路。在高可用性集群系统中，每个节点必须有心跳接口，用于节点之间互相监视和通信，以取得备用计算机的工作状态。常见的心跳信号可通过串行通信线路（RS-232）、TCP/IP网络和共享磁盘阵列互相传递信息。心跳线路最好使用两条不同的通信线路，达到监视线路冗余的效果。

● 集群技术

(4) 数据共享磁盘阵列。在高可用性集群系统中，由于运行的都是关键业务，所以使用的存储服务器都应该是企业级的存储服务器，这些存储服务器应具有先进技术来保证其数据安全。共享磁盘阵列是各节点计算机之间维持数据一致性的桥梁，各节点在集群软件的控制下不会同时访问共享磁盘阵列。

● 集群技术

(1) **高性能计算集群**是利用高速互连网络将一组PC机（或工作站）连接起来，在并行程序设计和集成开发环境支持下，统一调度和协调处理，实现高效并行处理的系统。

(2) **负载均衡集群**是一种动态均衡，它通过一些工具实时地分析数据包，掌握网络中的数据流量状况，把任务合理分配出去。根据负载均衡的位置不同，可以将负载均衡分为客户端负载均衡和服务器端负载均衡。客户端负载均衡是指客户端的均衡器根据集群的负载情况，主动选择由集群中的哪台计算机为其提供服务；服务器端负载均衡又可根据执行负载均衡的方式不同，分为集中式负载均衡和分布式负载均衡。

● 集群技术

在集中式负载均衡方式下，均衡器位于集群中一台计算机上，它根据当前集群的负载状态对负载进行集中式分配；在分布式负载均衡方式下，有多个均衡器位于集群中不同的计算机上，由这些均衡器根据其均衡策略和集群中各计算机的当前负载状态，以分布式协商的机制分配负载。

(3) **高可用性集群**能够很好地保证各种故障情况下应用系统访问的连续性。在高可用性集群中，应用系统的任何一个服务都可以运行在集群系统中的任何一个节点中，当这个节点出现故障时，运行在这个节点上的所有服务都可以在定义好的其他节点中启动运行，而用户感觉不到有任何变化。高可用性集群技术适用于对应用系统有严格高可靠性要求的企业、政府、军队、重要商业网站或数据库应用等用户。

● 高并发下的高可用性技术

1、CDN，Content Delivery Network，内容分发网络。尽可能避开互联网上有可能影响数据传输速度和稳定性的瓶颈和环节，使内容传输的更快、更稳定。通过在网络各处放置节点服务器所构成的在现有的互联网基础之上的一层智能虚拟网络，CDN系统能够实时地根据网络流量和各节点的连接、负载状况以及到用户的距离和响应时间等综合信息将用户的请求重新导向离用户最近的服务节点上。CDN解决因分布、带宽、服务器性能带来的访问延迟问题，适用于站点加速、点播、直播等场景。使用户可就近取得所需内容，解决 Internet 网络拥挤的状况，提高用户访问网站的响应速度和成功率。

尽可能的减少资源在转发、传输、链路抖动等情况下顺利保障信息的连贯性。因为很难自建大量的缓存节点，所以通常使用CDN运营商的服务。目前国内的服务商很少，而且按流量计费，价格也比较昂贵。

● 高并发下的高可用性技术

2、负载均衡

(1) 负载均衡技术分类。

①基于http的负载均衡。

②基于DNS的负载均衡。

③基于NAT（Network Address Translation，网络地址转换）的负载均衡。

④反向代理负载均衡。

● 高并发下的高可用性技术

①基于http的负载均衡：根据用户的http请求计算出一个真实的web服务器地址，并将该web服务器地址写入http重定向响应中返回给浏览器，由浏览器重新进行访问。

➤ 优点：比较简单

➤ 缺点：浏览器需要两次请求服务器才能完成一次访问，性能较差。

● 高并发下的高可用性技术

②基于DNS的负载均衡：

- 优点：将负载均衡的工作交给了DNS，省却了网站管理维护负载均衡服务器的麻烦，同时许多DNS还支持基于地理位置的域名解析，将域名解析成距离用户地理最近的一个服务器地址，加快访问速度，改善性能。
- 缺点：目前的DNS解析是多级解析，每一级DNS都可能缓存记录，当某一服务器下线后，该服务器对应的DNS记录可能仍然存在，导致分配到该服务器的用户访问失败。负载均衡效果并不是太好。

● 高并发下的高可用性技术

③基于NAT（Network Address Translation，网络地址转换）的负载均衡：将一个外部IP地址映射为多个内部IP地址，对每次连接请求动态地转换为一个内部节点的地址，将外部连接请求引到转换得到地址的那个节点上，从而达到负载均衡的目的。

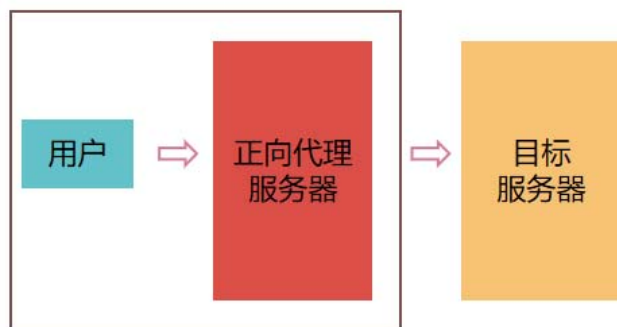
- 优点：在响应请求时速度较反向服务器负载均衡要快。
- 缺点：当请求数据较大（大型视频或文件）时，速度较慢。

● 高并发下的高可用性技术

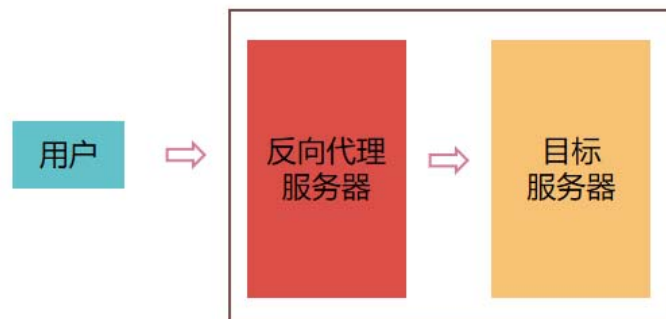
④反向代理负载均衡。反向代理是作用在服务器端的，是一个虚拟ip(VIP)。对于用户的一个请求，会转发到多个后端处理器中的一台来处理该具体请求。

- 优点：部署简单，处于http协议层面。
- 缺点：用了反向代理服务器后，web 服务器地址不能直接暴露在外，因此web服务器不需要使用外部IP地址，而反向代理服务作为沟通桥梁就需要配置双网卡、外部内部两套IP地址。

● 高并发下的高可用性技术



正向代理



反向代理

● 高并发下的高可用性技术

(2) 负载均衡的算法

静态均衡算法：

①轮询法：

将请求按顺序轮流地分配到每个节点上，不关心每个节点实际的连接数和当前的系统负载。

- 优点：简单高效，易于水平扩展，每个节点满足字面意义上的均衡；
- 缺点：没有考虑机器的性能问题，根据木桶最短木板理论，集群性能瓶颈更多的会受性能差的服务器影响。

● 高并发下的高可用性技术

②随机法：

将请求随机分配到各个节点。由概率统计理论得知，随着客户端调用服务端的次数增多，其实际效果越来越接近于平均分配，也就是轮询的结果。优缺点和轮询相似。

● 高并发下的高可用性技术

③源地址哈希法：

源地址哈希的思想是根据客户端的IP地址，通过哈希函数计算得到一个数值，用该数值对服务器节点数进行取模，得到的结果便是要访问节点序号。采用源地址哈希法进行负载均衡，同一IP地址的客户端，当后端服务器列表不变时，它每次都会落到到同一台服务器进行访问。

- 优点：相同的IP每次落在同一个节点，可以人为干预客户端请求方向，例如灰度发布；
- 缺点：如果某个节点出现故障，会导致这个节点上的客户端无法使用，无法保证高可用。当某一用户成为热点用户，那么会有巨大的流量涌向这个节点，导致冷热分布不均衡，无法有效利用起集群的性能。所以当热点事件出现时，一般会将源地址哈希法切换成轮询法。

● 高并发下的高可用性技术

④加权轮询法：

不同的后端服务器可能机器的配置和当前系统的负载并不相同，因此它们的抗压能力也不相同。给配置高、负载低的机器配置更高的权重，让其处理更多的请；而配置低、负载高的机器，给其分配较低的权重，降低其系统负载，加权轮询能很好地处理这一问题，并将请求顺序且按照权重分配到后端。

加权轮询算法要生成一个服务器序列，该序列中包含n个服务器。n是所有服务器的权重之和。在该序列中，每个服务器的出现的次数，等于其权重值。并且，生成的序列中，服务器的分布应该尽可能的均匀。比如序列{a, a, a, a, a, b, c}中，前五个请求都会分配给服务器a，这就是一种不均匀的分配方法，更好的序列应该是：{a, a, b, a, c, a, a}。

● 高并发下的高可用性技术

⑤加权随机法：

与加权轮询法一样，加权随机法也根据后端机器的配置，系统的负载分配不同的权重。不同的是，它是按照权重随机请求后端服务器，而非顺序。

⑥键值范围法：

根据键的范围进行负债，比如0到10万的用户请求走第一个节点服务器，10万到20万的用户请求走第二个节点服务器……以此类推。

- 优点：容易水平扩展，随着用户量增加，可以增加节点而不影响旧数据；
- 缺点：容易负债不均衡，比如新注册的用户活跃度高，旧用户活跃度低，那么压力就全在新增的服务节点上，旧服务节点性能浪费。而且也容易单点故障，无法满足高可用。

● 高并发下的高可用性技术

动态负载均衡算法：

①最小连接数法：

根据每个节点当前的连接情况，动态地选取其中当前积压连接数最少的一个节点处理当前请求，尽可能地提高后端服务的利用效率，将请求合理地分流到每一台服务器。俗称闲的人不能闲着，大家一起动起来。

- 优点：动态，根据节点状况实时变化；
- 缺点：提高了复杂度，每次连接断开需要进行计数；

实现：将连接数的倒数当权重值。

● 高并发下的高可用性技术

②最快响应速度法：

根据请求的响应时间，来动态调整每个节点的权重，将响应速度快的服务节点分配更多的请求，响应速度慢的服务节点分配更少的请求，俗称能者多劳，扶贫救弱。

➤ 优点：动态，实时变化，控制的粒度更细，跟灵敏；

➤ 缺点：复杂度更高，每次需要计算请求的响应速度；

实现：可以根据响应时间进行打分，计算权重。

③观察模式法：

观察者模式是综合了最小连接数和最快响应度，同时考量这两个指标数，进行一个权重的分配。

● 性能评价指标

性能指标，是软、硬件的性能指标的集成。包括计算机、路由器、交换机、网络、操作系统、数据库、WEB服务器。

性能指标	具体内容
计算机	时钟频率（主频）、运算速度、运算精度、内存的存储容量、存储器的存取周期、数据处理速率（ProcessingDataRate, PDR);吞吐率、各种响应时间、各种利用率、RASIS特性，即可靠性（Reliability）、可用性（Availability）、可维护性（Sericeability）、完整性和安全性（Integraity and Security);平均故障响应时间、兼容性、可扩充性、性能价格比。
网络	设备级性能指标、网络级性能指标、应用级性能指标、用户级性能指标、吞吐量。

● 性能评价指标

性能指标	具体内容
操作系统	系统的性能指标有系统的可靠性、系统的吞吐率（量）、系统响应时间、系统资源利用率、可移植性。
数据库管理系统	包括数据库本身和管理系统两部分，有数据库的大小、数据库中表的数量、单个表的大小、表中允许的记录（行）数量、单个记录（行）的大小、表上所允许的索引数量、数据库所允许的索引数量、最大并发事务处理能力、负载均衡能力、最大连接数等等。
Web服务器	最大并发连接数、响应延迟、吞吐量。

典型真题

●对计算机评价的主要性能指标有时钟频率、（）、运算精度和内存容量等。对数据库管理系统评价的主要性能指标有（）、数据库所允许的索引数量和最大并发实物处理能力等。

A、丢包率 B.端口吞吐量

C.可移植性 D.数据处理速率

A.MIPS B.支持协议和标准


C.最大连接数 D.时延抖动

典型真题

试题分析

性能指标，是软、硬件的性能指标的集成。在硬件中，包括计算机、各种通信交换设备、各类网络设备等；在软件中，包括：操作系统、协议以及应用程序等。

对计算机评价的主要性能指标有：时钟频率（主频）；运算速度；运算精度；内存的存储容量；存储器的存取周期；数据处理速率PDR（processing data rate）；吞吐率；各种响应时间；各种利用率；RASIS特性（即：可靠性Reliability、可用性Availability、可维护性、完整性和安全性；平均故障响应时间；兼容性；可扩充性；性能价格比。



典型真题

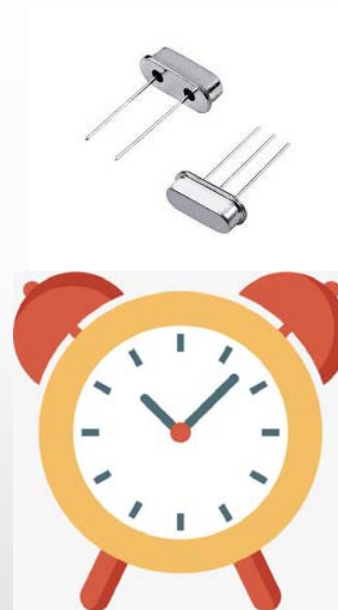
衡量事务处理能力、负载均衡能力、最大连接数等等。

参考答案：D、C

● 性能评价指标

计算机的运算速度与许多因素有关，例如，机器的主频、执行什么样的操作，以及主存的速度等。

主频	主频又称为时钟频率，在很大程度上决定了计算机的运算速度。CPU的工作节拍是由主时钟来控制的，主时钟不断产生固定频率的时钟脉冲，这个主时钟的频率就是CPU的主频。 主频=外频×倍频
CPU时钟周期	一个时钟脉冲所需要的时间。
机器周期	取指令、存储器读、存储器写等，这每一项工作称为一个基本操作。完成一个基本操作所需要的时间称为机器周期。一般情况下， 一个机器周期由若干个时钟周期组成。



● 性能评价指标

CPI

每条指令执行所用的时钟周期数是一个平均值。在现代高性能计算机中，由于采用各种并行技术，使指令执行高度并行化，常常是一个系统时钟周期内可以处理若干条指令。

MIPS (Million Instructions Per Second, 每秒百万条指令平均执行速度) :

$$\text{MIPS} = \text{指令条数} / (\text{执行时间} \times 10^6) = \text{主频} / \text{CPI}$$

典型真题

某台计算机的CPU主频为1.8GHz，如果2个时钟周期组成1个机器周期，平均3个机器周期可完成1条指令，则该计算机的指令平均执行速度为（ ） MIPS。

A.300

B.600

C.900

D.1800

典型真题

试题解析：

计算机的CPU主频为1.8GHz，2个时钟周期组成1个机器周期，平均3个机器周期可完成1条指令，则执行一条指令需要 $2 \times 3 = 6$ 个时钟周期，CPU

的主频为1.8GHz，因此执行速度为 $1800/6 = 300$ MIPS。

参考答案：A

● 阿姆达尔解决方案

1

阿姆达尔定律可以用来计算处理器平行运算之后效率提升的能力或所占总执行时间的比例。

2

阿姆达尔定律定义了采用特定组件所取得的加速比。假设使用某种改进了的组件，系统的性能就会得到提高。

阿姆达尔(Amdahl)定律量化定义了通过改进系统中某个组件的性能，使系统整体性能提高的程度。假设某一功能的处理时间为整个系统运行时间的60%，若使该功能的处理速度提高至原来的5倍，则根据阿姆达尔定律，整个系统的处理速度可提高至原来的()倍。

A.1.333 B.1.923

C.1.5

D.1.829

● 性能优化

对于**数据库应用系统**，造成性能不好的原因可能有数据库连接方式、系统应用架构、数据库设计、数据库管理、网络通信等，基于这些原因，可以采取修改应用模式、建立历史数据库、利用索引技术和分区技术等优化措施，需要调整的参数主要包括CPU和主存使用状况、数据库设计、进程或线程状态、硬盘剩余空间、日志文件大小等；

对于**Web应用系统**，性能瓶颈可能有客户端程序、网关接口、数据库互连等，

可以采取的优化措施主要有改善应用程序的性能和数据库连接、进行流量管理与负载均衡、使用Web交换机和Web缓存等，需要调整的参数主要包括

系统的可用性、响应时间、并发用户数，以及特定应用占用的系统资源等。

典型真题

为了优化系统的性能，有时需要对系统进行调整。对于不同的系统，其调整参数也不尽相同。例如，对于数据库系统，主要包括CPU/内存使用状况、（ ）、进程/线程使用状态、日志文件大小等。对于应用系统，主要包括应用系统的可用性、响应时间、（ ）、特定应用资源占用等。

A. 数据丢包率

B. 端口吞吐量

C. 数据处理速率

D. 查询语句性能

A. 并发用户数

B. 支持协议和标准

C. 最大连接数

D. 时延抖动

典型真题

试题分析

为了优化系统性能，有时需要对系统进行调整。对于数据库系统，性能调整主要包括CPU／内存使用状况、优化数据库设计、优化数据库管理以及进程/线程状态、硬盘剩余空间、日志文件大小等；对于应用系统，性能调整主要包括应用系统的可用性、响应时间、并发用户数以及特定应用的系统资源占用等。

参考答案：D、A

● 系统性能经典评估方法

时钟频率法	计算机的时钟频率在一定程度上反映了机器速度。显然，对同一种机型的计算机，时钟频率越高，计算机的工作速度就越快。
指令执行速度法	在计算机发展的初期，曾用加法指令的运算速度来衡量计算机的速度。因为加法指令的运算速度大体上可反映出乘法、除法等其他算术运算的速度，而且逻辑运算、转移指令等简单指令的执行时间往往设计成与加法指令相同，因此，加法指令的运算速度有一定代表性。表示机器运算速度的单位是MIPS。

● 系统性能经典评估方法

等效指令速度法	也称为吉普森或混合比例算法，是通过各类指令在程序中所占的比例进行计算后得到的计算机运算速度。
数据处理速率法 (PDR)	采用计算PDR值的方法来衡量机器性能，PDR值越大，机器性能越好。PDR与每条指令和每个操作数的平均位数以及每条指令的平均运算速度有关。PDR主要对CPU和主存储器的速度进行度量，不适合衡量机器的整体速度，不能全面反映计算机的性能，因为它没有涉及Cache、多功能部件等技术对性能的影响。
综合理论性能法	该方法是首先计算出处理部件每个计算单元的有效计算率，再按不同字长加以调整，得出该计算单元的理论性能，所有组成该处理部件的计算单元的理论性能之和即为最终的计算机性能。

● 基准程序法


经典评估方法性能评估方法主要是针对CPU（有时包括主存）的性能，但**没有考虑**诸如I/O结构、操作系统、编译程序的效率等对系统性能的影响，因此，**难以准确评估**计算机系统的实际性能。

把应用程序中用得**最多、最频繁**的那部分核心程序作为评估计算机系统性能的标准程序，称为**基准测试程序（benchmark）**。基准程序法是目前一致承认的测试系统性能的**较好方法**。

典型真题

计算机系统性能评估中，（ ）考虑了各类指令在程序中所占的比例。（ ）考虑了诸如I/O结构、操作系统、编译程序的效率对系统性能的影响，可以较为准确评估计算机系统的实际性能。

- | | |
|------------|------------|
| A. 时钟频率法 | B. 等效指令速度法 |
| C. 综合理论性能法 | D. 基准程序法 |
| A. 时钟频率法 | B. 等效指令速度法 |
| C. 综合理论性能法 | D. 基准程序法 |



典型真题

试题分析

略

参考答案：B、D

技术成就梦想

51CTO学院