第一章 计算机系统的基本概念

- 层次机构：按照计算机语言从低级到高级的次序，把计算机系统按功能划分成多级层次结构，每一层以一种不同的语言为特征。这些层次依次为：微程序机器级，传统机器语言机器级，汇编语言机器级，高级语言机器级，应用语言机器级等。
- 虚拟机：用软件实现的机器。
- 翻译：先用转换程序把高一级机器上的程序转换为低一级机器上等效的程序，然后再在这低一级机器上运行，实现程序的功能。
- 解释：对于高一级机器上的程序中的每一条语句或指令，都是转去执行低一级机器上的一段等效程序。执行完后，再去高一级机器取下一条语句或指令，再进行解释执行，如此反复，直到解释执行完整个程序。
- 计算机系统结构：传统机器程序员所看到的计算机属性，即概念性结构与功能特性。
- 透明性：在计算机技术中，把这种本来存在的事物或属性，但从某种角度看又好像不存在的概念称为透明性。
- 计算机组成：计算机系统结构的逻辑实现，包含物理机器级中的数据流和控制流的组成以及逻辑设计等。
- 计算机实现：计算机组成的物理实现，包括处理机、主存等部件的物理结构，器件的集成度和速度，模块、插件、底板的划分与连接，信号传输，电源、冷却及整机装配技术等。系统加速比：对系统中某部分进行改进时，改进后系统性能提高的倍数。
- Amdahl 定律：当对一个系统中的某个部件进行改进后，所能获得的整个系统性能的提高，受限于该部件的执行时间占总执行时间的百分比。
- 程序的局部性原理：程序执行时所访问的存储器地址不是随机分布的，而是相对地簇聚。包括时间局部性和空间局部性。
- CPI：每条指令执行的平均时钟周期数。
- 测试程序套件：由各种不同的真实应用程序构成的一组测试程序，用来测试计算机在各个方面的处理性能。
- 存储程序计算机：冯·诺依曼结构计算机。其基本点是指令驱动。程序预先存放在计算机存储器中，机器一旦启动，就能按照程序指定的逻辑顺序执行这些程序，自动完成由程序所描述的处理工作。
- 系列机：由同一厂家生产的具有相同系统结构、但具有不同组成和实现的一系列不同型号的计算机。
- 软件兼容：一个软件可以不经修改或者只需少量修改就可以由一台计算机移植到另一台计算机上运行。差别只是执行时间的不同。
- 向上（下）兼容：按某档计算机编制的程序，不加修改就能运行于比它高（低）档的计算机。
- 向后（前）兼容：按某个时期投入市场的某种型号计算机编制的程序，不加修改地就能运行于在它之后（前）投入市场的计算机。
- 兼容机：由不同公司厂家生产的具有相同系统结构的计算机。
- 模拟：用软件的方法在一台现有的计算机（称为宿主机）上实现另一台计算机（称为虚拟机）的指令系统。
- 仿真：用一台现有计算机（称为宿主机）上的微程序去解释实现另一台计算机（称为目标机）的指令系统。
- 并行性：计算机系统在同一时刻或者同一时间间隔内进行多种运算或操作。只要在时间上相互重叠，就存在并行性。它包括同时性与并发性两种含义。
- 时间重叠：在并行性概念中引入时间因素，让多个处理过程在时间上相互错开，轮流重叠地使用同一套硬件设备的各个部分，以加快硬件周转而赢得速度。
- 资源重复：在并行性概念中引入空间因素，以数量取胜。通过重复设置硬件资源，大幅度地提高计算机系统的性能。
- 资源共享：这是一种软件方法，它使多个任务按一定时间顺序轮流使用同一套硬件设备。

- 耦合度：反映多机系统中各计算机之间物理连接的紧密程度和交互作用能力的强弱。
- 紧密耦合系统：又称直接耦合系统。在这种系统中，计算机之间的物理连接的频带较高，一般是通过总线或高速开关互连，可以共享主存。
- 松散耦合系统：又称间接耦合系统，一般是通过通道或通信线路实现计算机之间的互连，可以共享外存设备（磁盘、磁带等）。计算机之间的相互作用是在文件或数据集一级上进行。
- 异构型多处理机系统：由多个不同类型、至少担负不同功能的处理机组成，它们按照作业要求的顺序，利用时间重叠原理，依次对它们的多个任务进行加工，各自完成规定的功能动作。同构型多处理机系统：由多个同类型或至少担负同等功能的处理机组成，它们同时处理同一作业中能并行执行的多个任务。

第二章 计算机指令集结构

- 堆栈型机器：CPU 中存储操作数的单元是堆栈的机器。
- 累加器型机器：CPU 中存储操作数的单元是累加器的机器。
- 通用寄存器型机器：CPU 中存储操作数的单元是通用寄存器的机器。
- CISC：复杂指令集计算机
- RISC：精简指令集计算机
- 寻址方式：指令系统中如何形成所要访问的数据的地址。一般来说，寻址方式可以指明指令中的操作数是一个常数、一个寄存器操作数或者是一个存储器操作数。
- 数据表示：硬件结构能够识别、指令系统可以直接调用的那些数据结构

第三章 流水线技术

- 流水线：将一个重复的时序过程，分解成为若干个子过程，而每一个子过程都可有效地在其专用功能段上与其它子过程同时执行。
- 单功能流水线：指流水线的各段之间的连接固定不变、只能完成一种固定功能的流水线。
- 多功能流水线：指各段可以进行不同的连接，以实现不同的功能的流水线。
- 静态流水线：指在同一时间内，多功能流水线中的各段只能按同一种功能的连接方式工作的流水线。当流水线要切换到另一种功能时，必须等前面的任务都流出流水线之后，才能改变连接。
- 动态流水线：指在同一时间内，多功能流水线中的各段可以按照不同的方式连接，同时执行
- 多种功能的流水线。它允许在某些段正在实现某种运算时，另一些段却在实现另一种运算。部件级流水线：把处理机中的部件进行分段，再把这些部件分段相互连接而成。它使得运算操作能够按流水方式进行。这种流水线也称为运算操作流水线。
- 处理机级流水线：又称指令流水线。它是把指令的执行过程按照流水方式进行处理，即把一条指令的执行过程分解为若干个子过程，每个子过程在独立的功能部件中执行。
- 处理机间流水线：又称为宏流水线。它是把多个处理机串行连接起来，对同一数据流进行处理，每个处理机完成整个任务中的一部分。前一个处理机的输出结果存入存储器中，作为后一个处理机的输入。
- 线性流水线：指各段串行连接、没有反馈回路的流水线。数据通过流水线中的各段时，每一个段最多只流过一次。
- 非线性流水线：指各段除了有串行的连接外，还有反馈回路的流水线。
- 顺序流水线：流水线输出端任务流出的顺序与输入端任务流入的顺序完全相同。
- 乱序流水线：流水线输出端任务流出的顺序与输入端任务流入的顺序可以不同，允许后进入流水线的任务先完成。这种流水线又称为无序流水线、错序流水线、异步流水线。
- 吞吐率：在单位时间内流水线所完成的任务数量或输出结果的数量。
- 流水线的加速比：使用顺序处理方式处理一批任务所用的时间与按流水处理方式处理同一批任务所用的时间之比。
- 流水线的效率：即流水线设备的利用率，它是指流水线中的设备实际使用时间与整个运行时间的比值。
- 数据相关：考虑两条指令 *i* 和 *j*，*i* 在 *j* 的前面，如果下述条件之一成立，则称指令 *j* 与指令*i* 数据相关：
    1. 指令 *j* 使用指令 *i* 产生的结果；
    2. 指令 *j* 与指令 *k* 数据相关，而指令 *k* 又与指令 *i* 数据相关。
- 名相关：如果两条指令使用了相同的名，但是它们之间并没有数据流动，则称这两条指令存在名相关。
- 控制相关：是指由分支指令引起的相关。它需要根据分支指令的执行结果来确定后面该执行哪个分支上的指令。
- 反相关：考虑两条指令 *i* 和 *j*，*i* 在 *j* 的前面，如果指令 *j* 所写的名与指令 *i* 所读的名相同，则称指令 *i* 和 *j* 发生了反相关。
- 输出相关：考虑两条指令 *i* 和 *j*，*i* 在 *j* 的前面，如果指令 *j* 和指令 *i* 所写的名相同，则称指令 *i* 和 *j* 发生了输出相关。换名技术：名相关的两条指令之间并没有数据的传送，只是使用了相同的名。可以把其中一条指令所使用的名换成别的，以此来消除名相关。
- 结构冲突：因硬件资源满足不了指令重叠执行的要求而发生的冲突。

- 数据冲突：当指令在流水线中重叠执行时，因需要用到前面指令的执行结果而发生的冲突。
- 控制冲突：流水线遇到分支指令或其它会改变 PC 值的指令所引起的冲突。
- 定向：用来解决写后读冲突的。在发生写后读相关的情况下，在计算结果尚未出来之前，后面等待使用该结果的指令并不见得是马上就要用该结果。如果能够将该计算结果从其产生的地方直接送到其它指令需要它的地方，那么就可以避免停顿。
- 写后读冲突：考虑两条指令 i 和 j，且 i 在 j 之前进入流水线，指令 j 用到指令 i 的计算结果，而且在 i 将结果写入寄存器之前就去读该寄存器，因而得到的是旧值。
- 读后写冲突：考虑两条指令 i 和 j，且 i 在 j 之前进入流水线，指令 j 的目的寄存器和指令 i的源操作数寄存器相同，而且 j 在 i 读取该寄存器之前就先对它进行了写操作，导致 i 读到的值是错误的。
- 写后写冲突：考虑两条指令 i 和 j，且 i 在 j 之前进入流水线，，指令 j 和指令 i 的结果单元（寄存器或存储器单元）相同，而且 j 在 i 写入之前就先对该单元进行了写入操作，从而导致写入顺序错误。这时在结果单元中留下的是 i 写入的值，而不是 j 写入的。
- 链接技术：具有先写后读相关的两条指令，在不出现功能部件冲突和 Vi 冲突的情况下，可以把功能部件链接起来进行流水处理，以达到加快执行的目的。
- 分段开采：当向量的长度大于向量寄存器的长度时，必须把长向量分成长度固定的段，然后循环分段处理，每一次循环只处理一个向量段。
- 半性能向量长度：向量处理机的性能为其最大性能 $R\infty$ 的一半时所需的向量长度。
- 向量长度临界值：向量流水方式的处理速度优于标量串行方式的处理速度时所需的向量长度的最小值。

第四章 指令级并行

- 指令级并行：简称 ILP。是指指令之间存在的一种并行性，利用它，计算机可以并行执行两条或两条以上的指令。
- 指令调度：通过在编译时让编译器重新组织指令顺序或通过硬件在执行时调整指令顺序来消除冲突。
- 指令的动态调度：是指在保持数据流和异常行为的情况下，通过硬件对指令执行顺序进行重新安排，以提高流水线的利用率且减少停顿现象。是由硬件在程序实际运行时实施的。
- 指令的静态调度：是指依靠编译器对代码进行静态调度，以减少相关和冲突。它不是在程序执行的过程中、而是在编译期间进行代码调度和优化的。
- 保留站：在采用 Tomasulo 算法的 MIPS 处理器浮点部件中，在运算部件的入口设置的用来保存一条已经流出并等待到本功能部件执行的指令（相关信息）。
- CDB：公共数据总线。动态分支预测技术：是用硬件动态地进行分支处理的方法。在程序运行时，根据分支指令过去的表现来预测其将来的行为。如果分支行为发生了变化，预测结果也跟着改变。
- BHT：分支历史表。用来记录相关分支指令最近一次或几次的执行情况是成功还是失败，并据此进行预测。
- 分支目标缓冲：是一种动态分支预测技术。将执行过的成功分支指令的地址以及预测的分支目标地址记录在一张硬件表中。在每次取指令的同时，用该指令的地址与表中所有项目的相应字段进行比较，以便尽早知道分支是否成功，尽早知道分支目标地址，达到减少分支开销的目的。
- 前瞻执行：解决控制相关的方法，它对分支指令的结果进行猜测，然后按这个猜测结果继续取指、流出和执行后续的指令。只是指令执行的结果不是写回到寄存器或存储器，而是放到一个称为 ROB 的缓冲器中。等到相应的指令得到"确认"（即确实是应该执行的）后，才将结果写入寄存器或存储器。
- ROB：ReOrder Buffer。前瞻执行缓冲器。
- 超标量：一种多指令流出技术。它在每个时钟周期流出的指令条数不固定，依代码的具体情况而定，但有个上限。
- 超流水：在一个时钟周期内分时流出多条指令。
- 超长指令字：一种多指令流出技术。VLIW 处理机在每个时钟周期流出的指令条数是固定的，这些指令构成一条长指令或者一个指令包，在这个指令包中，指令之间的并行性是通过指令显式地表示出来的。
- 循环展开：是一种增加指令间并行性最简单和最常用的方法。它将循环展开若干遍后，通过重命名和指令调度来开发更多的并行性。

第五章 存储系统

- 多级存储层次：采用不同的技术实现的存储器，处在离 CPU 不同距离的层次上，各存储器之间一般满足包容关系，即任何一层存储器中的内容都是其下一层（离 CPU 更远的一层）存储器中内容的子集。目标是达到离 CPU 最近的存储器的速度，最远的存储器的容量。
- 全相联映象：主存中的任一块可以被放置到 Cache 中任意一个地方。
- 直接映象：主存中的每一块只能被放置到 Cache 中唯一的一个地方。
- 组相联映象：主存中的每一块可以放置到 Cache 中唯一的一组中任何一个地方（Cache 分成若干组，每组由若干块构成）。
- 替换算法：由于主存中的块比 Cache 中的块多，所以当要从主存中调一个块到 Cache 中时，会出现该块所映象到的一组（或一个）Cache 块已全部被占用的情况。这时，需要被迫腾出其中的某一块，以接纳新调入的块。
- LRU：选择最近最少被访问的块作为被替换的块。实际实现都是选择最久没有被访问的块作为被替换的块。
- 写直达法：在执行写操作时，不仅把信息写入 Cache 中相应的块，而且也写入下一级存储器中相应的块。
- 写回法：只把信息写入 Cache 中相应块，该块只有被替换时，才被写回主存。
- 按写分配法：写失效时，先把所写单元所在的块调入 Cache，然后再进行写入。
- 不按写分配法：写失效时，直接写入下一级存储器中，而不把相应的块调入 Cache。
- 命中时间：访问 Cache 命中时所用的时间。
- 失效率：CPU 访存时，在一级存储器中找不到所需信息的概率。
- 失效开销：CPU 向二级存储器发出访问请求到把这个数据调入一级存储器所需的时间。
- 强制性失效：当第一次访问一个块时，该块不在 Cache 中，需要从下一级存储器中调入 Cache，这就是强制性失效。
- 容量失效：如果程序在执行时，所需要的块不能全部调入 Cache 中，则当某些块被替换后又重新被访问，就会产生失效，这种失效就称作容量失效。
- 冲突失效：在组相联或直接映象 Cache 中，若太多的块映象到同一组（块）中，则会出现该组中某个块被别的块替换（即使别的组或块有空闲位置），然后又被重新访问的情况。
- 2：1Cache 经验规则：大小为 N 的直接映象 Cache 的失效率约等于大小为 N /2 的两路组相联 Cache 的实效率。
- 相联度：在组相联中，每组 Cache 中的块数。
- Victim Cache：位于 Cache 和存储器之间的又一级 Cache，容量小，采用全相联策略。用于存放由于失效而被丢弃（替换）的那些块。每当失效发生时，在访问下一级存储器之前，先检查 Victim Cache 中是否含有所需块。
- 故障性预取：在预取时，若出现虚地址故障或违反保护权限，就会发生异常。非故障性预取：在预取时，若出现虚地址故障或违反保护权限，不发生异常。
- 非阻塞 Cache：Cache 在等待预取数据返回时，还能继续提供指令和数据。
- 尽早重启动：在请求字没有到达时，CPU 处于等待状态。一旦请求字到达，就立即发送给 CPU，让等待的 CPU 尽早重启动，继续执行。
- 请求字优先：调块时，首先向存储器请求 CPU 所要的请求字。请求字一旦到达，就立即送往 CPU，让 CPU 继续执行，同时从存储器调入该块的其余部分。
- 虚拟 Cache：地址使用虚地址的 Cache。
- 多体交叉存储器：具有多个存储体，各体之间按字交叉的存储技术。
- 存储体冲突：多个请求要访问同一个体。
- TLB：一个专用高速存储器，用于存放近期经常使用的页表项，其内容是页表部分内容的一个副本。

第六章 输入输出系统

- 响应时间：从用户键入命令开始，到得到结果所花的时间。
- 可靠性：指系统从某个初始参考点开始一直连续提供服务的能力，它通常用平均无故障时间来衡量。
- 可用性：指系统正常工作的时间在连续两次正常服务间隔时间中所占的比率。
- 可信性：指服务的质量，即在多大程度上可以合理地认为服务是可靠的。
- RAID：廉价磁盘冗余阵列或独立磁盘冗余阵列。
- 分离事务总线：将总线事务分成请求和应答两部分。在请求和应答之间的空闲时间内，总线可以供给其它的 I/O 使用。采用这种技术的总线称为分离事务总线。
- 通道：专门负责整个计算机系统输入/输出工作的专用处理机，能执行有限的一组输入输出指令。
- 通道流量：指一个通道在数据传送期间，单位时间内能够传送的数据量。
- 虚拟 DMA：它允许 DMA 设备直接使用虚拟地址，并在 DMA 传送的过程中由硬件将虚拟地址转换为物理地址。
- 异步 I/O：允许进程在发出 I/O 请求后继续执行，直到该进程真正访问这些数据而它们又尚未就绪时，才被挂起。

第八章 多处理机

- 集中式共享多处理机：也称为对称式共享存储器多处理 SMP。它一般由几十个处理器构成，各处理器共享一个集中式的物理存储器，这个主存相对于各处理器的关系是对称的，
- 分布式共享多处理机：它的共享存储器分布在各台处理机中，每台处理机都带有自己的本地存储器，组成一个“处理机-存储器”单元。但是这些分布在各台处理机中的实际存储器又合在一起统一编址， 在逻辑上组成一个共享存储器。这些处理机存储器单元通过互连网络连接在一起，每台处理机除了能访问本地存储器外，还能通过互连网络直接访问在其他处理机存储器单元中的 “远程存储器”。
- 通信延迟：通信延迟 = 发送开销 + 跨越时间 + 传输时间 + 接收开销。
- 计算/通信比：反映并行程序性能的一个重要的度量。在并行计算中，每次数据通信要进行的计算与通信开销的比值。
- 多 Cache 一致性：多处理机中，当共享数据进入 Cache，就可能出现多个处理器的 Cache 中都有同一存储器块的副本，要保证多个副本数据是一致的。
- 监听协议：每个 Cache 除了包含物理存储器中块的数据拷贝之外，也保存着各个块的共享状态信息。Cache 通常连在共享存储器的总线上，各个 Cache 控制器通过监听总线来判断它们是否有总线上请求的数据块。
- 目录协议：用一种专用的存储器所记录的数据结构。它记录着可以进入 Cache的每个数据块的访问状态、该块在各个处理器的共享状态以及是否修改过等信息。
- 写作废协议：在处理器对某个数据项进行写入之前，它拥有对该数据项的唯一的访问权。
- 写更新协议：当一个处理器对某数据项进行写入时，它把该新数据广播给所有其它 Cache。这些 Cache 用该新数据对其中的副本进行更新。
- 栅栏同步：栅栏强制所有到达该栅栏的进程进行等待。直到全部的进程到达栅栏，然后释放全部进程，从而形成同步。
- 旋转锁：处理机环绕一个锁不停地旋转而请求获得该锁。
- 同时多线程：是一种在多流出、动态调度的处理器上同时开发线程级并行和指令级并行的技术，它是多线程技术的一种改进。
- 细粒度多线程技术：是一种实现多线程的技术。它在每条指令之间都能进行线程的切换，从而使得多个线程可以交替执行。通常以时间片轮转的方法实现这样的交替执行，在轮转的过程中跳过处于停顿的线程。
- 粗粒度多线程技术：是一种实现多线程的技术。只有线程发生较长时间的停顿时才切换到其他线程。
- SMP：对称式共享存储器多处理
- MPP ： 即大规模并行处理，按照当前的标准，具有几百台～几千台处理机的任何机器都是大规模并行处理系统。

第九章 机群系统

- 机群：是一种价格低廉、易于构建、可扩放性极强的并行计算机系统。它由多台同构或异构的独立计算机通过高性能网络或局域网互连在一起，协同完成特定的并行计算任务。从用户的角度来看，机群就是一个单一、集中的计算资源。
- 单一系统映象：包含四重含义。
    1. 单一系统。尽管系统中有多个处理器，用户仍然把整个机群视为一个单一的计算系统来使用。
    2. 单一控制。逻辑上，最终用户或系统用户使用的服务都来自机群中唯一一个位置。
    3. 对称性。用户可以从任一个结点上获得机群服务，也就是说，对于所有结点和所有用户，除了那些具有特定访问权限的服务与功能外，所有机群服务与功能都是对称的。
    4. 位置透明。用户不必了解真正提供服务的物理设备的具体位置。
- 高可用性机群：当系统中某些结点出现故障的情况下，仍能继续对外提供服务。它采用冗余机制，当系统中某个结点由于软、硬件故障而失效时，该结点上的任务将在最短的时间内被迁移到机群内另一个具有相同功能与结构的结点上继续执行。
- 负载均衡机群：机群能够根据系统中各个结点的负载情况实时地进行任务分配。它专门设置了一个重要的监控结点，负责监控其余每个工作结点的负载和状态，并根据监控结果将任务分派到不同的结点上。
- 高性能计算机群：通过高速的商用互连网络，将数十台乃至上千台 PC 机或工作站连接在一起，可以提供接近甚至超过传统并行计算机系统的计算能力，但其价格却仅是具有相同计算能力的传统并行计算机系统的几十分之一。
- Beowulf 机群：使用普通的硬件加上 Linux 操作系统、再加上 GNU 开发环境以及 PVM/MPI 共享库所构建的机群。它一方面集中了那些相对较小的机器的计算能力，能够以很高的性能价格比提供与大型机相当的性能，另一方面也保证了软件环境的稳定性。