高级体系结构知识点

作者: 杨森

2017 年 8 月 19 日

基础

名词解释/填空

1. (微处理器技术) 的发展带来了计算机设计的复兴, 既强调 (体系结构) 方面的创新, 也重视技术改进的高效应用.(2006)
2. Throughout this range in price and capabiity, the desktop market tends to be driven to optimize (*price-performance*).(2006)
3. Power Consumption in Computer Systems Power consumption in modern systems is dependent on a variety of factors, including the chip (*clock frequency*), (*efficiency*), the disk drive speed, disk drive utilization, and (*DRAM*).(2013)
4. 服务器的三个关键特征:(可用性),(可扩展性),(吞吐能力).(2006)
5. 一个事件从启动到完成的事件, 称为 (响应时间), 也称为 (执行时间); 仓库级计算机的操作人员可能关心的是 (吞吐量), 也就是给定时间内完成的总工作量.(2006)
6. **计算机体系结构**: 计算机系统结构是指 (传统机器程序员) 看到的计算机属性, 即概念性结构和功能特性.(2001、2005、2008、2013)
7. 存储程序计算机 (冯诺依曼结构) 的特点是: 机器以 (运算器) 为中心; 采用存储程序原理; 存储器是按照 (地址) 访问的、线性编址的空间; 控制流由 (指令流) 流产生; 指令有操作码和地址码组成;数据以 (二) 进制编码表示, 采用二进制运算;由 (运算器),(存储器),(输入/输出设备),(控制器) 组成 (2007、2009、2012、2015)
8. 程序的局部性原理是指: 程序总是趋向于使用最近使用过的 (数据) 和 (指令), 程序执行时所访问的存储器地址不是随机分布的, 而是相对簇集, 包括 (时间局部性) 和 (空间局部性):(2008、2012、2015)
   (a) **时间局部性**: 程序即将用到的信息很可能是目前正在用的信息.(2002)
   (b) 空间局部性: 程序即将用到的信息很可能与目前正在使用的信息在空间上邻近.
9. **软件兼容 (Software Compatable)**: 软件兼容是指一台计算机上的程序不加修改就可以搬到另一台计算机上正常运行.(2002)
10. 所谓系列机就是指具有相同的 (体系结构), 但具有不同 (组成) 和 (实现) 的一些列不同型号的机器. 我们把不同厂家生产的具有相同体系结构的计算机称为 (兼容机).(2008、2010、2013)
11. **并行性 (parallelism)**: 并行性是指计算机系统在同一时刻或者同一时间间隔内运行多种运算或者操作, 包括同时性和并发性两种含义, 同时性是指两个或两个以上的时间在同一时刻发生, 并发性是指两个或两个以上的事件在同一时间间隔内发生.(2002)
12. **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 (加速比), 受限于该部件的 (执行时间) 占总执行时间的 (百分比).(2004、2010、2011、2014、2016)
13. 器件发生故障的概率与 (时间) 的关系可以使用**学习曲线**来说明。(2011)
14. 我们把 DRAM 这种不供电数据丢失的存储器称为 (易失性) 存储器，把磁盘这种不供电数据也能够保存数据的存储器称为 (非易失性) 存储器。(2015)

简答题

1. 简述存储程序计算机的特点: 机器以 (运算器) 为中心；采用存储程序原理；存储器是按照 (地址) 访问的、线性编址的空间；控制流由 (指令流) 流产生；指令有操作码和地址码组成；数据以 (二) 进制编码表示, 采用二进制运算；由 (运算器),(存储器),(输入/输出设备),(控制器) 组成 (2007、2009、2012、2015)

指令系统

名词解释/填空

1. 设计指令系统包括 (寻址方式)，(操作类型)，(指令集操作)，(指令系统编码) 等方面。(2011、2016)
2. **RISC** 含义是 (精简指令集计算机),**CISC** 含义是 (复杂指令集计算机).(2008、2015)
3. **寄存器-寄存器型 (Load/Store 型)** 指令结构: 操作数都来自通用寄存器组.(2001、2003)
4. 指令系统的基本要求:
   (a) 完整性: 在有限可用的存储空间内, 对于任何可解的问题, 编制计算程序时, 指令系统提供的功能足够使用.
   (b) **规整性**: 指令系统的规整性主要包括对称性和均匀性, 对称性是指所有与指令系统有关的存储单元的使用、操作码的设置等都是对称的；均匀性对于各种不同的操作数类型、字长、操作种类和数据存储单元, 指令的设置都要同等对待.(2001、2004)
   (c) **正交性**: 指令中各个不同含义的字段, 如操作类型、数据类型、寻址方式字段等, 在编码时应该互不相关、相互独立.(2002)
   (d) 高效率: 指令的执行速度快、使用频率高.

流水线

名词解释/填空

1. 流水线分类:

   (a) 部件级流水线: 把处理机中的部件进行分段, 在把这些分段相互连接而成.

   (b) 处理机级流水线: 又称指令流水线 (Instruction Pipeline), 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.

   (c) 系统级流水线: 又称为宏流水线 (Macro Pipeline), 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.

   (d) 单功能流水线 *(Unifunction Pipeline)*: 流水线各段之间连接固定不变, 只能完成单一固定功能.

   (e) **多功能流水线 (Multifunction Pipeline)**: 流水线各段之间可以进行不同的连接, 以实现不同功能.(2004)

   (f) 静态流水线 *(Static Pipeline)*: 同一时间内, 多功能流水线的各段只能按照同一种功能的连接方式工作.

   (g) 动态流水线 *(Dynamic Pipeline)*: 同一时间内, 多功能流水线的各段可以按照不同的方式连接.

   (h) 线性流水线 *(Linear Pipeline)*: 流水线各段串行连接、没有回馈.

   (i) 非线性流水线 *(Nonlinear Pipeline)*: 各段除了有串行的连接外, 还有反馈回路.

   (j) 顺序流水线 *(In-order Pipeline)*: 流水线输出端任务流出顺序与输入端任务流入顺序完全相同.

   (k) 乱序流水线 *(Out-order Pipeline)*: 流水线输出端任务流出顺序与输入端任务流入顺序可以不同.

2. 对于采用指令流水线的处理器, 其 CPI 公式为: CPI 流水线 = 流水线理想 CPI + (结构相关停顿导致的停顿) + (数据相关导致的停顿) + (控制相关导致的停顿).(2005、2007、2013)

3. 流水线的理想 CPI 是衡量流水线 (最高) 性能的一个指标.(2006)

4. 相关分类

   (a) **结构相关**: 当指令在同步重叠执行过程中, 硬件资源满足不了指令重叠执行的要求, 发生资源冲突.(2001、2002、2003)

   (b) **数据相关**: 当一条指令需要用到前面的执行结果, 而这些指令均在流水线中重叠执行是产生数据相关; 对于顺序指令 i 和 j, 指令 i 和 j 存在数据相关是指, 指令 j 的 (操作数寄存器或存储器) 是指令 i 要写入的寄存器或者 (存储单元)。(2006、2010)

   (c) 控制相关: 当流水线遇到分支指令和其他能够改变 PC 值的指令就会发生控制相关.

   (d) **反相关**: 如果指令 j 的 (目标寄存器) 是指令 i 要 (访问) 的 (寄存器) 或 (存储单元).(2011、2014、2016、2017)

   (e) 输出相关: 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.

5. **吞吐率 (ThroughPut, TP)**: 单位时间内流水线所完成的任务数量或者输出结果的数量.(2002)
6. **定向 (Forwarding)**: 在发生写后读相关的情况下, 将计算结果从起产生的地方直接从到其他指令需要它的的地方.(2003、2017)
7. **分支延迟槽 (Brancy Delay Slot)**: 存放分支指令的后继指令，无论分支指令成功与否，都会执行分支延迟槽中的指令.(2017)
8. **异常 (Exception)**: 指令正常的执行顺序得到改变.(2017)

向量处理机

名词解释/填空

1. 通过时间: 第一个任务从进入流水线到流出结果的时间段.
2. 排空时间: 最后一个任务从进入流水线到流出结果的时间段.
3. 流水线的**链接 (pipeline chaining)** 是将流水线计算机中多个 (功能部件) 按照指令要求链接在一起, 构成一个 (长流水线), 减少各个功能部件流水线 (加载/建立) 和 (排空) 的时间, 提高流水线执行的效率; 向量流水线的链接是在有 (数据 (写后读)) 相关的 (两) 条指令之间, 将产生数据的指令部件的 (结果) 直接送到使用数据部件的 (向量寄存器/输入).(2007、2009、2010、2011、2013、2014、2016)

指令级并行

名词解释/填空

1. 当指令之间不存在 (相关) 时，它们在流水线中是可以重叠起来并行执行的，这种指令序列中存在的潜在并行性称为 (指令级并行)。(2015)
2. **基本块 (Basic Block)**: 一段连续的代码除了入口和出口, 没有其他的分之指令和转入点, 这称其为基本快.(2003)
3. This potential overlap among instructions is called (*instruction-level parallelism (ILP)*) since the instructions can be evaluated in (*parallel*).(2006)
4. 静态调度 *(Static Scheduling)*: 在编译期间而非程序执行过程中进行代码的调度和优化.
5. **动态调度 (Dynamic Scheduling)**: 在程序的执行过程中, 依靠专门硬件对代码进行调度.(2004)
6. **乱序流出 (Out of order issue)**: 程序中的 (指令) 不是按照排列的顺序流出, 而是当指令需要的资源条件得到满足时即 (流出).(2001、2010)
7. 乱序执行 *(Out of order Execution)*: 指令的执行顺序与程序顺序不同.
8. 乱序完成 *(Out of order completion)*: 指令的完成顺序与程序顺序不同.
9. **分支预测缓冲 (Branch Prediction Buffer, BPB, BHT)**: 仅使用一片存储区域, 记录最近一次或几次的分支特征的历史, 包括两个步骤, 分支预测和预测为修改.(2004)
10. 分支目标缓冲 *(Branch Target Buffer, BTB)*: 将分支成功的分支指令和它的分支目标地址都放到一个缓冲区中保存起来, 缓冲去以分支指令的地址作为标识.
11. 在前瞻执行 (speculation) 中, 指令的流出过程是 (顺序) 的, 但结束过程的确认是 (顺序) 的, 再定序缓冲用于保存执行完毕但尚未顺序确认的指令.(2009)
12. **再定序缓冲 (ReOrder Buffer)**: 暂存指令执行的结果, 使其不直接写回到寄存器或者存储器, 能够在分支错误的情况下恢复现场.(2001)
13. **超标量技术 (Superscalar)**: 在每个时钟周期流出多条指令, 但流出的指令的条数不固定.(2003、2017)
14. **全局代码调度 (Globa code scheduling)**: 把分支之前的指令调度到分支之后, 或者把分支之后的指令调度到分支之前, 称之为 (全局) 代码调度, 其目的实在保持代段 (数据) 相关和 (控制) 相关不变的前提下, 将一段内部包含 (控制相关) 相关的代码段压缩到尽可能最短.(2005、2017)
15. **软流水**:(循环重组技术), 使得 (循环体) 由原来不同循环中指令组成. 是一种用于程序中 (循环结构) 的指令 (重组) 技术, 又称为 (符号化循环展开).(2002、2005、2007、2010)

存储系统

名词解释/填空

1. 在存储层次中, “Cache-主存” 层次为了弥补主存 (速度) 不足, “主存-辅存” 层次为了弥补主存 (容量) 的不足.(2008、2011、2012、2015、2016)

2. **虚拟 cache**: 虚拟 Cache 是指可以直接用虚拟地址进行访问的 Cache, 其标识存储器中存放的是虚拟地址, 进行地址检测用的也是虚拟地址.(2001、2009)

3. **存储体冲突**:(多体交叉存储器) 中, 两次独立的存储器访问请求在 (一个存储周期) 中访问 (同一个) 存储体, 就造成存储体冲突.(2001、2009、2011、2014、2016)

4. Cache 的三种类型不命中

   (a) **强制性不命中 (Compulsory Miss)**: 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, 这就是强制性不命中.(2002、2004、2010)

   (b) **容量不命中 (Capacity Miss)**: 程序执行时所需的块不能全部调入 Cache, 则当某些快被替换后, 若又重新被访问, 就会发生容量不命中.(2003)

   (c) 冲突不命中 *(Conflict Miss)*: 在组相联或者直接映像 Cache 中, 若太多的块映像到同一组 (块) 中, 则会出现某个块被别的块替换, 然后又被重新访问的情况, 就会发生冲突不命中.

5. 映像规则 (2005)

   (a) **全相联映像 (Fully Associative)**: 把主存中任意一块放置到 Cache 中任意一个位置

   (b) **直接映像 (Direct Mapping)**: 主存中每一块放置到 Cache 中唯一位置

   (c) **组相联映像 (Set Associative)**: 主存中每一块放置到 Cache 中唯一一个组中任意位置

6. 预取方式

   (a) 寄存器预取: 把数据取到寄存器中

   (b) *Cache* 预取: 把数据取到 Cache 中

   (c) 故障性预取: 预取时, 若出现 (虚地址故障) 或 (违反保护权限), 就会发生异常

   (d) **非故障性预取 (非绑定预取,nonbinding)**: 当出现虚地址故障或违反保护权限时, 不发生异常, 而是放弃预取, 转为空操作. 非绑定预取能够返回 (最新数据值), 并且保证对数据实际的存储器访问的是最新的数据项 (2008、2009)

7. TLB 是一个专用的 (高速缓存部件), 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).(2007、2012)

简答

1. 简述两级 *Cache* 的工作原理 (2001): 答: 在原有 Cache 和存储器之间增加另一级 Cache，构成两级 Cache，这就可以把第一级 Cache 做的足够小，使其能够和快速 CPU 的时钟周期相匹配；同时把第二级 Cache 做的足够大，使其能捕获更多本来需要到主存去的访问，降低实际不命中开销，克服 CPU 和主存之间的性能差距，使存储器和 CPU 性能匹配。

2. 简述 TLB 的工作原理 (2001): TLB 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。进行地址转换时，直接查找 TLB，只有在 TLB 不命中时，才需要访问内存中的页表，也成为块表或地址变换缓冲器，是一种能够实现快速地址变换的技术。

互连网络

名词解释/填空

1. 在拓扑上, 互连网络为输入和输出两组节点之间提供了一组互连 或映像.
2. **均匀混洗 (shuffle)**: 输入左移一位;**超立方体路由**: 特定为置反;设互联网的输入为 $(x_{k-1},\cdots,x_1,x_0)$, 则均匀混洗输出是 y=$(x_{k-2},\cdots,x_1,x_0,x_{k-1})$, 超立方体输出是 y=$(x_{k-1},\cdots,\bar{x_j},\cdots,x_1,x_0)$。(2010)
3. 静态互连网络: 由点和点直接相连而成
4. 动态互连网络: 由开关通道 实现, 可以动态改变结构
5. 节点度: 与节点相连接的边的数目称为节点度, 这里的边表示链路或通道, 进入节点的通道数为入度, 从节点出来的通道数为出度 (节点度 = 入度 + 出度)(2008).
6. **网络直径**: 网络中任意两个节点间 (最短) 路径长度的 (最大值) 叫做 (网络直径).(2002、2004、2006、2009、2010、2012、2014、2016、2017)
7. **等分宽度**: 将网络切成 (任意相等两半的各种切法中), 沿切口的 (最小通道边数) 称为等分带宽.(2009、2011、2013、2014、2015)
8. 对于一个网络, 如果从任何一个节点看, 拓扑结构都一样, 则称此网络为对称网络.
9. 路由: 在网络通信中对路径的选择与指定.
10. **虚拟自适应**: 将一个 (物理通道) 分成几个 (虚拟的通道), 根据后续各虚拟通道的 (资源或网络) 情况自适应选择后续通道。(2003、2007、2012、2015)
11. **虫孔路由 (Worm hole)**: 把信息包分成 (小片), 片头带 (目的地址), 所有片以不可分离的 (流水方式) 通过片缓冲区进行传输路由.(2001、2004、2007、2012)

多处理机

名词解释/填空

1. MIMD 计算机分类、存储器系统结构

   (a) 集中式共享存储器结构 *(Centralized Shared-Memory Architecture)*: 处理器较少，共享一个集中式的物理存储器。因为主存单一，且对各处理器的关系是对称的，所以称其为**对称式共享存储器多处理机 (Symmetric shared-memory Multi processor, SMP)**，因此也称为 **UMA(Uniform Memory Access)** 结构。

   (b) 分布式存储器多处理机: 存储器在物理上是分布的，支持大规模的多处理机系统，具有两种存储器系统结构和处理器通信方式：

   - 把物理上分离的所有存储器进行统一的 (共享逻辑空间) 进行编址，任何处理器都可访问任何存储单元，这类系统称为**分布式共享存储器系统 (Distributed Shared-Memory, DSM)**，此处共享是 (地址空间上) 的共享，不具有集中的存储器，也被称为 **NUMA**,，这是因为其 (访存时间) 取决于 (数据) 在存储器中的 (存放位置)。(2011、2013、2014、2015、2017)
   - 把每个结点中存储器编址为一个 (独立的地址空间)，不同结点地址空间独立，每个结点中存储器只能有本地处理器进行访问，远程处理器不能对其进行直接访问，这种计算机系统多以 (机群) 形式存在。

2. **Home 结点 (宿主结点)**: 是指存放 (需要访问) 的 (存储单元) 及 (相应的目录项) 所在的结点.(2008、2009、2016)

3. 本地结点 *(local node)*: 发出 (访问请求) 的结点.

4. 远程结点 *(remote node)*: 拥有 (被访问存储块副本) 的结点.

5. 拥有者 *(owner)* 是指拥有唯一的 (*Cache* 块副本) 的处理器.(2008)

6. **栅栏同步**: 栅栏强制所有的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) 全部的进程, 从而形成同步.(2003、2007、2013、2016)

7. 同时多线程 (Simultaneous Multi Threading, SMT) 是同时实现 (指令级) 和线程级的并行, 每拍有 (多个指令槽), 可以安排多个线程的 (多条指令) 同时流出。(2005、2009、2011、2014、2015)

8. 基本硬件原语: 原子交换 *(Atomic Exchange)* 将一个存储单元的值和一个寄存器的值进行交换; 测试并置定 *(test_and_set)* 先测试一个存储单元的值, 如果符合条件则修改其值; 读取并加 *1(fetch_and_increment)* 返回存储单元的值并自动增加该值; 指令对 *LL/SC*, 从第二条指令的返回值判断该指令执行是否成功, 这两条指令之间不能插入其他对存储单元进行操作的其他指令.

9. 大规模机器的同步方法:(2010、2012、2013、2014、2017)

   (a) 硬件方法：

   - **排队锁**: 通过硬件向量等方式进行排队和同步控制。

- **硬件原语**: 引进一种原语 (可以是 Fetch_and_increment) 减少栅栏技术时所需的时间, 从而减小串行形成的瓶颈。

(b) 软件方法:

- **延迟等待旋转锁**: 加锁失败时, 延迟的时间指数增大;
- **排队锁**: 用数组将要进行同步的进程排队, 按照排序进行同步操作;
- **组合树栅栏**: 通过组合树 (Combining Tree) 减少栅栏中进程读取 release 标志形成的冲突。