并行计算复习————第一篇 并行计算硬件平台：并行计算机_JCGuo的专栏-CSDN博客

并行计算复习

第一篇 并行计算硬件平台：并行计算机

Ch1 并行计算与并行计算机结构模型

1.1多核处理器与线程级并行

1.何谓多核处理器？

将功能复杂的单一核处理器划分为若干个功能相对简单的多个处理器内核，这些多处理器集中在一块芯片上，最初称为单芯片多处理器CMP，Intel公司将其商用名定为**多核处理器**

2.多核处理器的意义：

- 解决单处理器瓶颈：密集晶体管集成，功耗剧增；设计指令级并行体系结构来利用晶体管资源，但软件与硬件设计复杂
- 具有自己的优势：CMP设计验证周期短、开发风险成本低，相对较低的主频功耗也相对较低、单处理器程序移植容易，通过布局能够改善多处理器内核之前延迟和带宽

3.微处理器中的并行方式

- ILP：指令级并行，单处理器同时执行多条指令，包括乱序执行、分支预测、指令多发射、硬件预取等技术
- TLP：线程级并行，多处理器多线程执行
- 多任务OS：多进程多线程分时间片轮转或抢占式，OS管理
- SMT：同时多线程技术，超标量与多线程的结合，同时发射多个线程中的多条不相关指令
- CMP：单芯片多处理器
- 虚拟计算技术：异构平台，剥离指令集结构和处理器依赖关系（运行时虚拟化JVM、系统虚拟化）
- Intel超线程技术：单核心模拟双核心环境执行多线程，是一种SMT

1.2 并行计算机体系结构

1.并行计算机结构模型

（1）结构类型

- SISD：单指令流单数据流计算机（冯诺依曼机）
- SIMD：单指令流多数据流计算机
- MISD：多指令流单数据流计算机
- MIMD：多指令流多数据流计算机

（2）几种MIMD

- PVP并行向量处理机：多VP（向量处理器）通过交叉开关和多个SM（共享内存）相连
- SMP对称多处理机：多P/C（商品微处理器）通过交叉开关/总线和多个SM（共享内存）相连
- MPP大规模并行处理机：处理节点有商品微处理器+LM（分布式本地内存），节点间通过高带宽低延迟定制网络互联，异步MIMD，多个进程有自己的地址空间，通过消息传递机制通信
- COW工作站机群：节点是完整操作系统的工作站，且有磁盘
- DSM分布共享存储处理机：高速缓存目录DIR确保缓存一致性，将物理分布式LM组成逻辑共享SM从而提供统一地址的编程空间

注：对称指所有处理器都能同等地访问I/O很同样的运行程序（如OS和I/O服务程序），而非对称主从式是仅有主处理器运行OS和控制访问I/O并监控从处理器执行

2.并行计算机访存模型

- UMA（Uniform Memory Access）均匀存储访问：物理存储器被所有处理器均匀共享，所有处理器对所有SM访存时间相同，每台处理器可带有高速私有缓存，外围设备共享。
- NUMA非均匀存储访问：共享的SM是由物理分布式的LM逻辑构成，处理器访存时间不一样，访问LM或CSM（群内共享存储器）内存储器比访问GSM（群间共享存储器）快
- COMA（Cache-Only MA）全高速缓存存储访问：NUMA的特例、全高速缓存实现
- CC-NUMA（Coherent-Cache NUMA）高速缓存一致性NUMA：NUMA + 高速缓存一致性协议
- NORMA（No-Remote MA）非远程存储访问：无SM，所有LM私有，通过消息传递通信

3.Cache一致性协议

- 监听总线协议：总线连接通信，写无效和写更新策略
- 基于目录的协议：目录记录共享数据缓存状态，读缺失时查看目录D，写更新时通知目录D

4.其他并行计算概念

衡量并行计算机性能单位：

- PFLOPS：每秒1千万亿 (=10^15) 次的浮点运算
- TFLOPS：每秒1万亿 (=10^12) 次的浮点运算
- GFLOPS：每秒10亿 (=10^9) 次的浮点运算

TOP500前500名超级计算机排名指标(GFLOPS)：

- Rmax：Maximal LINPACK(Linear system package) performance achieved
- Rpeak：Theoretical peak performance

Ch2 并行计算机系统互连与基本通信操作

2.1 并行计算机互连网络

互连网络是并行计算机系统中各处理器与内存模块等之间传输的机制

1.静态互连

处理单元间有固定连接的网络，程序执行期间这种点到点的连接不变

- 一维线性阵列LA/LC：二邻近串联

- 二维网孔MC：四邻近连接（Illiac连接、2D环绕）
- 树连接TC：二叉树、星型网络、二叉胖树（节点通路向根节点方向逐渐变宽，解决通信瓶颈）
- 超立方HC：3立方、4立方
- 立方环：3立方顶点用环代替

2.动态互连

交换开关构成的，可按应用程序要求动态改变连接组态

- 总线：连接处理器、存储模块、I/O外围设备等的一组导线和插座，分时工作、多请求总线仲裁，多总线（本地、存储、数据、系统）和多层总线（板级、底板级、I/O级）
- 交叉开关：高带宽的开关控制的专用连接通路网络，NxN的开关网络同时只能接通N对源目的通信
- 多级互联网络MIN：每一级用多个开关单元，各级之间有固定的级联拓扑

3.标准网络互连

- FDDI光纤分布式数据接口
- 快速以太网
- Myrinet：商用千兆位包开关网
- InfiniBand：交换式通信结构

2.2-2.5 通信代价公式

1.选路

（1）消息格式

消息是由一些定长的**信包**组成，信包包括了

- 选路信息R
- 顺序号S
- 多个数据片D

（2）存储转发选路SF

SF中信包是基本传输单位，中间节点必须收齐信包中所有分片且存储在缓冲器后才可能传向下一节点

长度为m的信包，穿越l条链路，SF基本通信时间公式：

$$t_{comm}(SF)=t_s+(mt_w+t_h)l$$

$$t_{comm}(SF)=t_s+(mt_w+t_h)l$$

其中$t_s$$t_s$是启动时间，$t_h$$t_h$是节点延迟时间，$t_w$$t_w$是传输每个字节的时间（带宽倒数）

（3）切通选路CT

CT中信包切片传输（包头和数据片），类似流水线

长度为m的信包，穿越l条链路，CT基本通信时间公式：

$$t_{comm}(CT)=t_s+mt_w+lt_h$$

$$t_{comm}(CT)=t_s+mt_w+lt_h$$

2.SF一到多播送

（1）一维环

最远的节点是瓶颈：

$$tone-to-all(SF)=(ts+mtw)\lceil p/2\rceil$$

$$tone-to-all(SF)=(ts+mtw)\lceil p/2\rceil$$

（2）带环绕的Mesh

先完成行SF环绕播送，再完成列的SF环绕播送（即两次节点个数为$p-\sqrt{p}$的一维环SF）：

$$tone-to-all(SF)=2(ts+mtw)\lceil p-\sqrt{/2}\rceil$$

$$tone-to-all(SF)=2(ts+mtw)\lceil p/2\rceil$$

（3）超立方

同理带环绕的Mesh，可推知：

$$tone-to-all(SF)=3(ts+mtw)\lceil p13/2\rceil$$

$$tone-to-all(SF)=3(ts+mtw)\lceil p13/2\rceil$$

3.CT一到多播送

（1）一维环

CT通信时间与中继节点无关，采取先按高维播送，再按中维播送，最后按低维播送：

$$tone-to-all(CT)=\sum i=1\log(p)(ts+mtw+th\times p/2i)=(ts+mtw)\log(p)+th(p-1)$$

$$tone-to-all(CT)=\sum i=1\log(p)(ts+mtw+th\times p/2i)=(ts+mtw)\log(p)+th(p-1)$$

（2）带环绕的Mesh

$$tone-to-all(CT)=(ts+mtw)\log(p)+2th(p-\sqrt{-1})$$

$$tone-to-all(CT)=(ts+mtw)\log(p)+2th(p-1)$$

（3）超立方

$$tone-to-all(CT)=(ts+mtw)\log(p)$$

$$tone-to-all(CT)=(ts+mtw)\log(p)$$

2.SF多到多播送

（1）一维环

p-1次环路传播：

$$tall-to-all(SF)=(ts+mtw)(p-1)$$

$$tall-to-all(SF)=(ts+mtw)(p-1)$$

（2）带环绕的Mesh

先行环路多播，再列环路多播

$$tall-to-all(SF)=(ts+mtw)(p-\sqrt{-1})+(ts+mp-\sqrt{tw})(p-\sqrt{-1})=2ts(p-\sqrt{-1})+mtw(p-1)$$
$$tall-to-all(SF)=(ts+mtw)(p-1)+(ts+mptw)(p-1)=2ts(p-1)+mtw(p-1)$$

（3）超立方

$$tall-to-all(SF)=tslog(p)+mtw(p-1)$$
$$tall-to-all(SF)=tslog(p)+mtw(p-1)$$

Ch4 并行计算性能评测

4.1 基本性能指标（见书）

4.2 加速比性能定律

约定：

- pp是处理器数
- 问题规模WW=程序中串行分量WsWs+可并行部分WhWh
- ff为串行部分比例，f=Ws/Wf=Ws/W
- SS为加速比

1.Amdahl加速定律

固定负载加速比公式：

$$Slimp\to\infty S=Ws+WpWs+Wpp=1f+1-fp=1f$$
$$S=Ws+WpWs+Wpp=1f+1-fplimp\to\infty S=1f$$

若考虑并行额外开销W0W0：

$$Slimp\to\infty S=Ws+WpWs+Wpp+W0=1f+1-fp+W0W=1f+W0W$$
$$S=Ws+WpWs+Wpp+W0=1f+1-fp+W0Wlimp\to\infty S=1f+W0W$$

2.Gustafson

实际应用中增多了处理器不会固定问题规模，而是保持总时间不变的情况下去增大问题规模：

$$S=Ws+pWpWs+pWpp=Ws+pWpWs+Wp=f+p(1-f)$$
$$S=Ws+pWpWs+pWpp=Ws+pWpWs+Wp=f+p(1-f)$$

若考虑并行额外开销W0W0：

$$S=Ws+pWpWs+pWpp+W0=Ws+pWpWs+Wp+W0=f+p(1-f)1+W0/W$$
$$S=Ws+pWpWs+pWpp+W0=Ws+pWpWs+Wp+W0=f+p(1-f)1+W0/W$$

3.Sun&Ni定律

问题规模增加了，相应的存储容量也要增加p倍，令因子G(p)G(p)为存储容量增加到p倍时工作负载的增加，则有加速比：

$$S=\frac{W_s+G(p)W_p}{W_s+G(p)W_p}=\frac{f+(1-f)G(p)}{f+(1-f)G(p)/p}$$

$$S=\frac{W_s+G(p)W_p}{W_s+G(p)W_p}=\frac{f+(1-f)G(p)}{f+(1-f)G(p)/p}$$

若考虑并行额外开销W_0：

$$S=\frac{W_s+G(p)W_p}{W_s+G(p)W_p+W_{0p}}=\frac{f+(1-f)G(p)}{f+(1-f)G(p)/p+W_0/W}$$

$$S=\frac{W_s+G(p)W_p}{W_s+G(p)W_p+W_{0p}}=\frac{f+(1-f)G(p)}{f+(1-f)G(p)/p+W_0/W}$$