上海第二工业大学

并行计算
Parallel Computing

主讲人　陈林

Sep,2021

并行计算——结构·算法·编程

- 第一篇 并行计算的基础
  - 第一章 并行计算与并行计算机结构模型
  - 第二章 并行计算机系统互连与基本通信操作
  - 第三章 典型并行计算机系统介绍
  - 第四章并行计算性能评测

第四章 并行计算性能评测

- 4.1 并行机的一些基本性能指标
- 4.2 加速比性能定律
  - 4.2.1 Amdahl定律
  - 4.2.2 Gustafson定律
  - 4.2.3 Sun和Ni定律
- 4.3 可扩放性评测标准
  - 4.3.1 并行计算的可扩放性
  - 4.3.2 等效率度量标准
  - 4.3.3 等速度度量标准
  - 4.3.4 平均延迟度量标准

CPU的某些基本性能指标

- 工作负载
  - 执行时间
  - 浮点运算数: Flops
  - 指令数目: MIPS

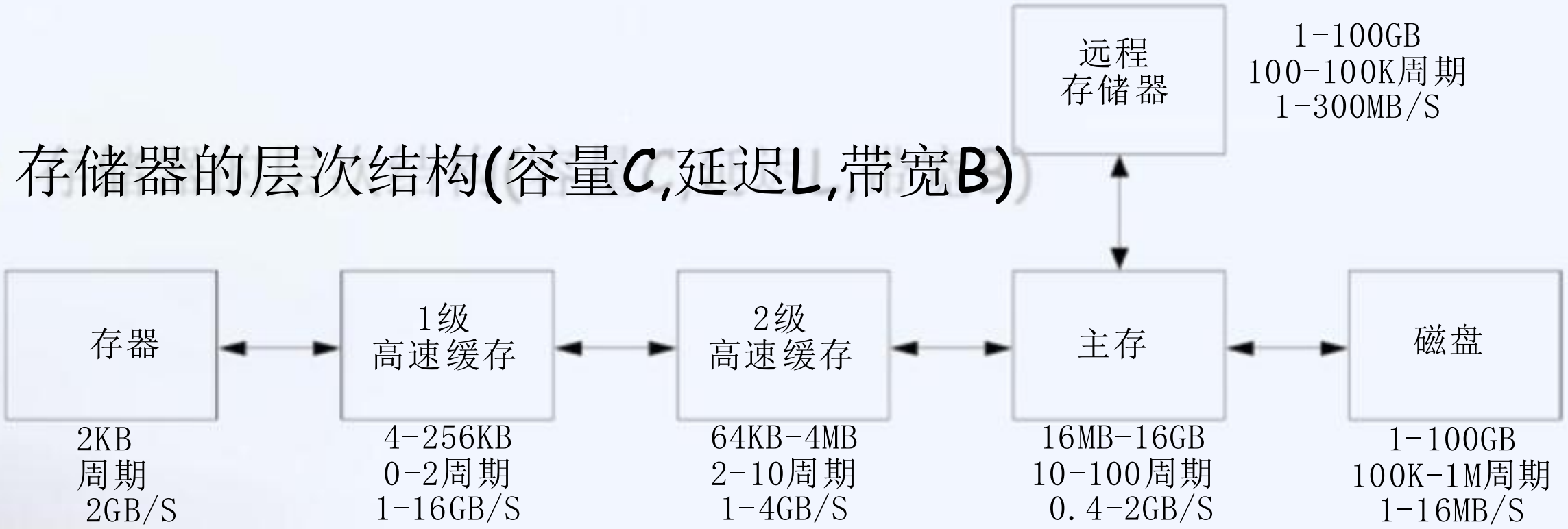
- 无重叠的假定下：并行执行时间包括 T_{comput} 计算时间，T_{paro} 并行开销时间，T_{comm} 相互通信时间

$$T_n = T_{comput} + T_{paro} + T_{comm}$$

  - T_{paro}：进程管理（如进程生成、结束和切换等），组操作（如进程组的生成与消亡等），进程查询（如询问进程的标志、等级、组标志和组大小等）
  - T_{comm}：同步（如路障、锁、临界区、事件等），通讯（如点到点通信、整体通信），聚合操作（如规约、前缀运算等）

存储器性能

- 存储器的层次结构(容量C,延迟L,带宽B)



- 估计存储器的带宽

RISC的加法可在单拍内完成，假定字长8bytes，时钟频率100MHz，则

带宽B = $3*8*100*10^6$ B/s= 2.4GB/s

并行与通信开销

- 并行和通信开销：相对于计算很大。

  PowerPC (每个周期 15ns 执行4flops;

  创建一个进程1.4ms 可执行372000flops)

- 开销的测量：乒--乓方法（Ping-Pong Scheme）节点0发送m个字节给节点1；节点1从节点0接收m个字节后，立即将消息发回节点0。总的时间除以2，即可得到点到点通信时间，也就是执行单一发送或接收操作的时间。

- 可一般化为热土豆法（Hot-Potato），也称为救火队法（Fire-Brigade) 0——1 —— 2 —— … —— n-1 —— 0

Ping-Pong Scheme

```
if （my_node_id =0） then /*发送者*/
        start_time =second（ ）
                send an m-byte message to node 1
                receive an m-byte message from node 1
        end_time = second（ ）
        total_time = end_time – start_time
        communication_time[i] = total_time/2
else if （my_node_id = 1） then /*接收者*/
                receive an m-byte message from node 0
                send an m-byte message to node 0
        endif
```

并行开销的表达式：点到点通信

- 通信开销$t(m) = t_0 + m/r_\infty$，m为消息长度（字节数）
- 通信启动时间t_0
- 渐近带宽r_∞：传送无限长的消息时的通信速率
- 半峰值长度$m_{1/2}$：达到一半渐近带宽所要的消息长度
- 特定性能π_0：表示短消息带宽
- 4个参数只有两个是独立的：$t_0 = m_{1/2}/r_\infty = 1/\pi_0$
- 以上是由Hockney提出的

并行开销的表达式：整体通信

- 典型的整体通信有：
  - 播送（Broadcasting）：处理器0发送m个字节给所有的n个处理器
  - 收集（Gather）：处理0接收所有n个处理器发来在消息，所以处理器0最终接收了m n个字节；
  - 散射（Scatter）：处理器0发送了m个字节的不同消息给所有n个处理器，因此处理器0最终发送了m n个字节；
  - 全交换（Total Exchange）：每个处理器均彼此相互发送m个字节的不同消息给对方，所以总通信量为mn^2个字节；
  - 循环移位（Circular-shift）：处理器i发送m个字节给处理器i+1，处理器n-1发送m个字节给处理器0，所以通信量为mn个字节。

机器的成本、价格与性/价比

- 机器的成本与价格
- 机器的性能/价格比Performance/Cost Ratio：系指用单位代价（通常以百万美元表示）所获取的性能（通常以MIPS或MFLOPS表示）
- 利用率（Utilization）：可达到的速度与峰值速度之比

算法级性能评测

- 加速比性能定律
  - 并行系统的加速比是指对于一个给定的应用，并行算法（或并行程序）的执行速度相对于串行算法（或串行程序）的执行速度加快了多少倍。
  - Amdahl 定律
  - Gustafson定律
  - Sun Ni定律
- 可扩放性评测标准
  - 等效率度量标准
  - 等速度度量标准
  - 平均延迟度量标准

第四章 并行计算性能评测

- 4.1 并行机的一些基本性能指标
- 4.2 加速比性能定律
  - 4.2.1 Amdahl定律
  - 4.2.2 Gustafson定律
  - 4.2.3 Sun和Ni定律
- 4.3 可扩放性评测标准
  - 4.3.1 并行计算的可扩放性
  - 4.3.2 等效率度量标准
  - 4.3.3 等速度度量标准
  - 4.3.4 平均延迟度量标准

Amdahl 定律（1）

- P：处理器数；
- W：问题规模（计算负载、工作负载，给定问题的总计算量）；
  - W_s 应用程序中的串行分量，f是串行分量比例（$f = W_s/W$，$W_s=W_1$）；
  - W_p 应用程序中可并行化部分，1-f为并行分量比例；
  - $W_s+W_p=W$；
- $T_s=T_1$：串行执行时间，T_p：并行执行时间；
- S：加速比，E：效率；
- 出发点：Base on Fixed Problem Size
  - 固定不变的计算负载；
  - 固定的计算负载分布在多个处理器上的，
  - 增加处理器加快执行速度，从而达到了加速的目的。

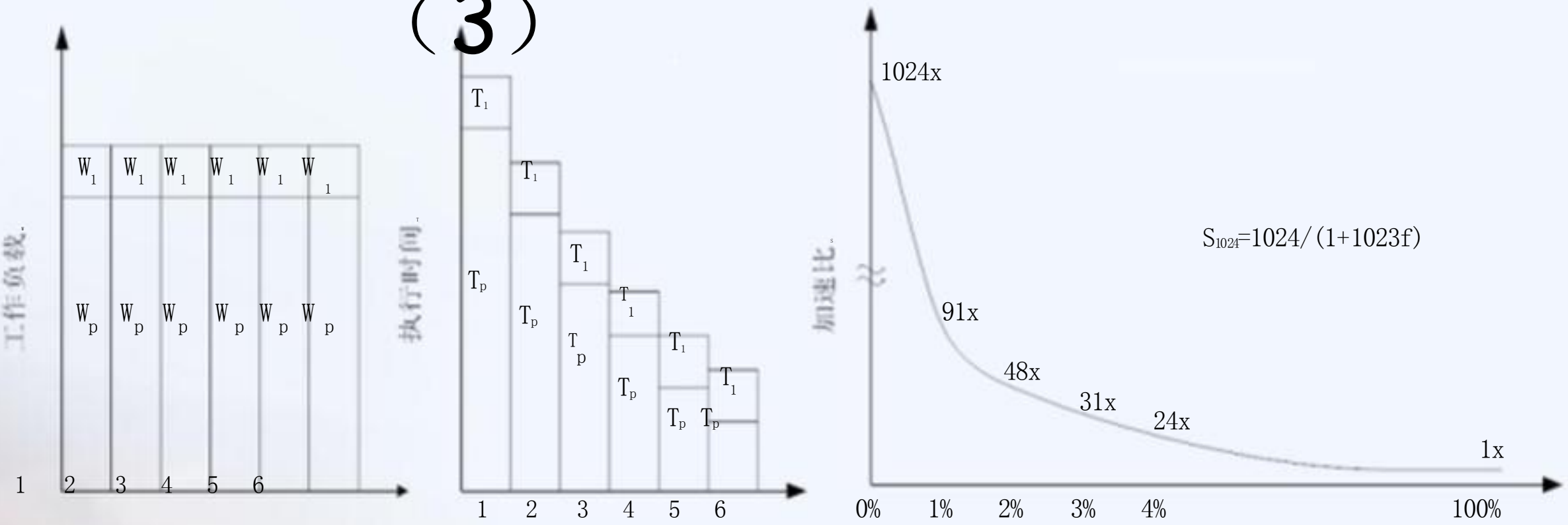
Amdahl定律（2）

- Amdahl’s Law （1967年）表明：
  - 适用于实时应用问题。当问题的计算负载或规模固定时，我们必须通过增加处理器数目来降低计算时间；
  - 加速比受到算法中串行工作量的限制。
  - Amdahl’s law: argument against massively parallel systems
- 公式推导

$$T_s = fW + (1-f)W \qquad T_p = fW + \frac{(1-f)W}{p}$$

$$S_p = \frac{W}{fW + \frac{(1-f)W}{p}} = \frac{p}{pf + 1 - f} = \frac{1}{\frac{(p-1)f + 1}{p}} \xrightarrow[p\to\infty]{} \frac{1}{f}$$

Amdahl’s law（3）



$S_{1024}=1024/(1+1023f)$

- 实际上并行加速不仅受限于串行分量，而且也受并行实现时的额外开销的限制：P124(P84 in old version)

Gustafson定律（1）

- 出发点：Base on Fixed Execution Time
  - 对于很多大型计算，精度要求很高，即在此类应用中精度是个关键因素，而计算时间是固定不变的。此时为了提高精度，必须加大计算量，相应地亦必须增多处理器数才能维持时间不变；
  - 表明：随着处理器数目的增加，串行执行部分f不再是并行算法的瓶颈。
- Gustafson加速定律（1988）：

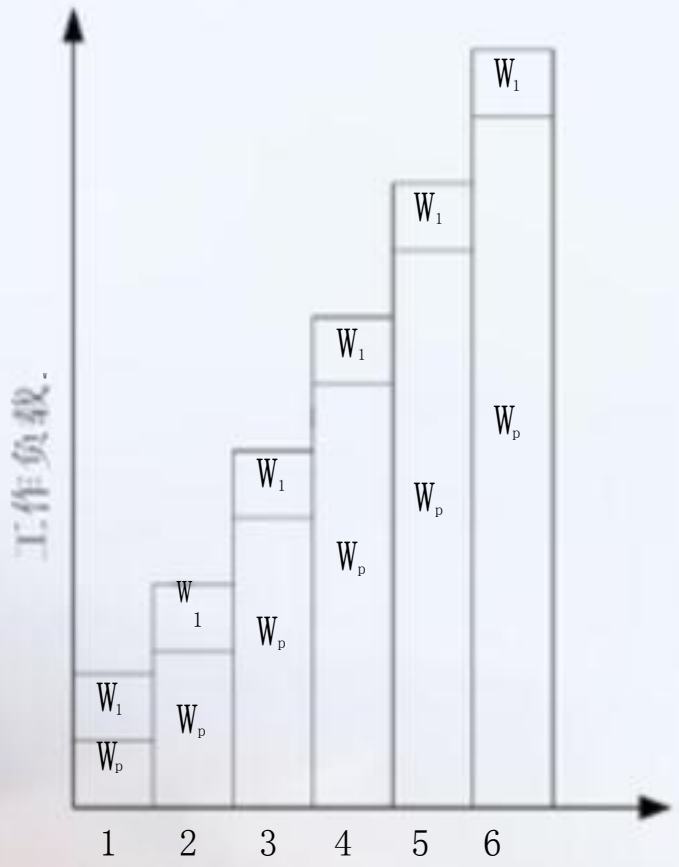
$$S' = \frac{W_S + pW_p}{W_S + p \cdot Wp/p} = \frac{W_S + pW_p}{W_S + W_P}$$

$$S' = f + p(1-f) = p + f(1-p) = p - f(p-1)$$

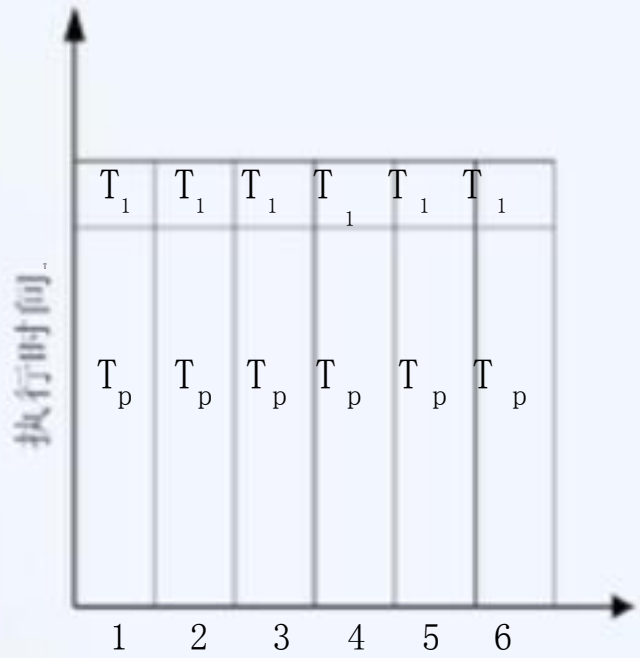
- 并行开销W_o：

$$S' = \frac{W_S + pW_P}{W_S + W_P + W_O} = \frac{f + p(1-f)}{1 + W_O / W}$$

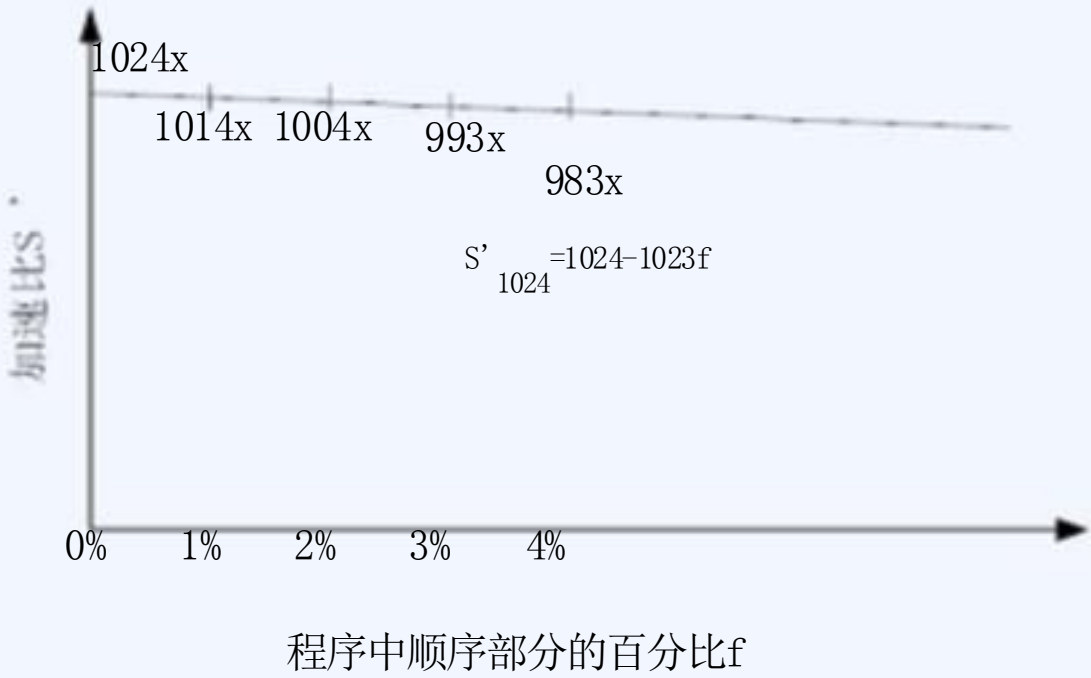
Gustafson定律（2）



(a)



(b)



(c)

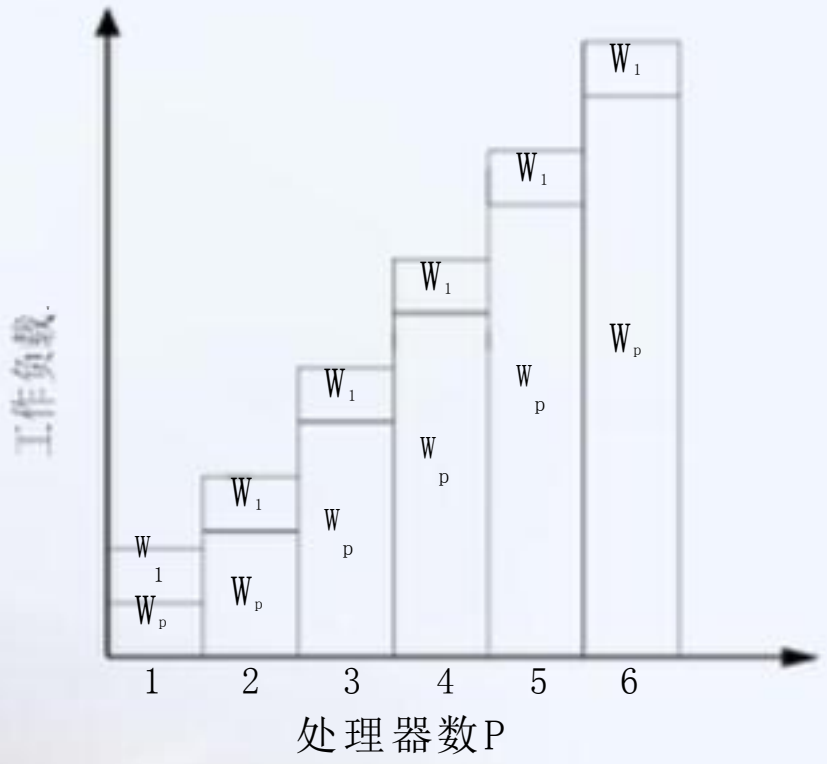
- 出发点：Base on Memory Bounding
  - 充分利用存储空间等计算资源，尽量增大问题规模以产生更好/更精确的解。是Amdahl定律和Gustafson定律的推广。
- 公式推导：
  - 设单机上的存储器容量为M，其工作负载W=fW+(1-f)W
    当并行系统有p个结点时，存储容量扩大了pM，用G(p)表示系统的存储容量增加p倍时工作负载的增加量。则存储容量扩大后的工作负载为W=fW+(1-f)G(p)W，所以存储受限的加速为

$$S'' = \frac{fW + (1-f)G(p)W}{fW + (1-f)G(p)W/p} = \frac{f + (1-f)G(p)}{f + (1-f)G(p)/p}$$

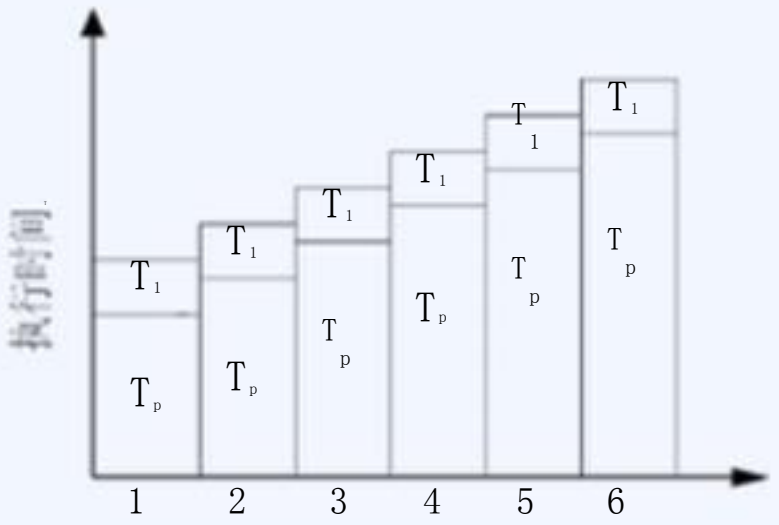
  - 并行开销W：

$$S'' = \frac{fW + (1-f)WG(p)}{fW + (1-f)G(p)W/p + W_o} = \frac{f + (1-f)G(p)}{f + (1-f)G(p)/p + W_o/W}$$

Sun 和 Ni定律（2）



(a)



(b)

- G（p）=1时就是Amdahl加速定律；
- G（p）=p 变为f + p（1-f），就是Gustafson加速定律
- G（p）>p时，相应于计算机负载比存储要求增加得快，此时Sun和 N i 加速均比Amdahl 加速和Gustafson 加速为高。

加速比讨论

- 参考的加速经验公式：$p/\log p \leqslant S \leqslant p$
- 线性加速比：很少通信开销的矩阵相加、内积运算等
- $p/\log p$的加速比：分治类的应用问题
- 通信密集类的应用问题：$S = 1/C(p)$

  这里$C(p)$是p个处理器的某一通信函数
- 超线性加速
- 绝对加速：最佳串行算法与并行算法
- 相对加速：同一算法在单机和并行机的运行时间

第四章 并行计算性能评测

- 4.1 并行机的一些基本性能指标
- 4.2 加速比性能定律
  - 4.2.1 Amdahl定律
  - 4.2.2 Gustafson定律
  - 4.2.3 Sun和Ni定律
- 4.3 可扩放性评测标准
  - 4.3.1 并行计算的可扩放性
  - 4.3.2 等效率度量标准
  - 4.3.3 等速度度量标准
  - 4.3.4 平均延迟度量标准

可扩放性评测标准（1）

- 并行计算的可扩放性（Scalability）也是主要性能指标
  - 可扩放性最简朴的含意是在确定的应用背景下，计算机系统（或算法或程序等）性能随处理器数的增加而按比例提高的能力
- 影响因素：处理器数与问题规模，还有
  - 求解问题中的串行分量；
  - 并行处理所引起的额外开销（通信、等待、竞争、冗余操作和同步等）；
  - 加大的处理器数超过了算法中的并发程度；
- 增加问题规模的好处：
  - 提供较高的并发机会；
  - 额外开销的增加可能慢于有效计算的增加；
  - 算法中的串行分量比例不是固定不变的（串行部分所占的比例随着问题规模的增大而缩小）。
- 增加处理器数会增大额外开销和降低处理器利用率，所以对于一个特定的并行系统（算法或程序），它们能否有效利用不断增加的处理器的能力应是受限的，而度量这种能力就是可扩放性这一指标。

可扩放性评测标准（2）

- 可扩放性:调整什么和按什么比例调整
  - 并行计算要调整的是处理数p和问题规模W，
  - 两者可按不同比例进行调整，此比例关系（可能是线性的，多项式的或指数的等）就反映了可扩放的程度。
- 与并行算法和体系结构相关
- 可扩放性研究的主要目的：
  - 确定解决某类问题用何种并行算法与何种并行体系结构的组合，可以有效地利用大量的处理器；
  - 对于运行于某种体系结构的并行机上的某种算法当移植到大规模处理机上后运行的性能；
  - 对固定的问题规模，确定在某类并行机上最优的处理器数与可获得的最大的加速比；
  - 用于指导改进并行算法和并行机体系结构，以使并行算法尽可能地充分利用可扩充的大量处理器
- 目前无一个公认的、标准的和被普遍接受的严格定义和评判它的标准

等效率度量标准（1）

- 令t_{ie}和t_{io}分别是并行系统上第i个处理器的有用计算时间和额外开销时间（包括通信、同步和空闲等待时间等）

$$T_e = \sum_{i=0}^{p-1} t_e^i = T_s \qquad T_o = \sum_{i=0}^{p-1} t_o^i$$

- T_p是p个处理器系统上并行算法的运行时间，对于任意i显然(假设)有

$T_p = t_{ie} + t_{io}$，且 $T_e + T_o = pT_p$

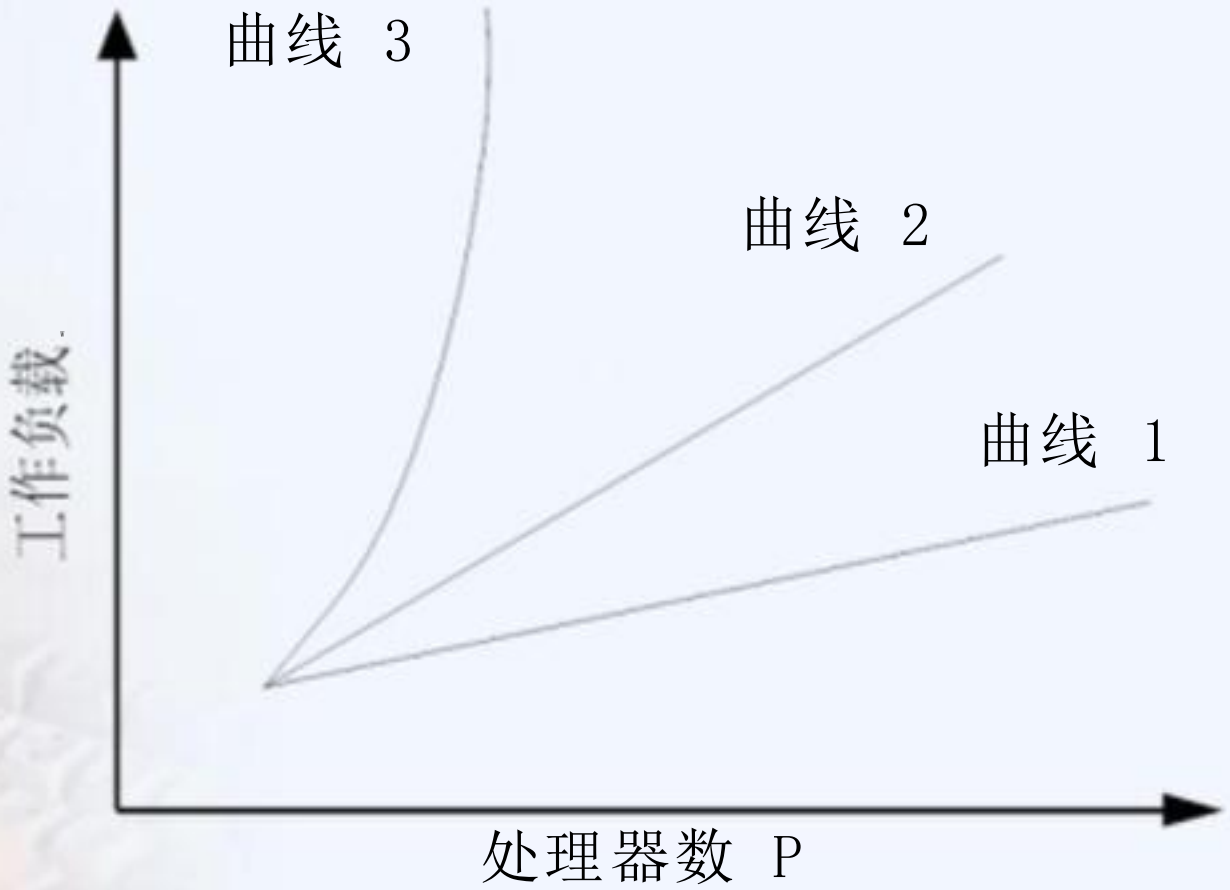
- 问题的规模W定义为最佳串行算法所完成的计算量，$W=T_e$

$$S = \frac{T_e}{T_p} = \frac{T_e}{\frac{T_e + T_o}{p}} = \frac{p}{1 + \frac{T_o}{T_e}} = \frac{p}{1 + \frac{T_o}{W}} \qquad E = \frac{S}{p} = \frac{1}{1 + \frac{T_o}{T_e}} = \frac{1}{1 + \frac{T_o}{W}}$$

- 如果问题规模W 保持不变，处理器数p增加，开销T_o增大，效率E下降。为了维持一定的效率（介于0与1之间），当处理数p增大时，需要相应地增大问题规模W的值。由此定义函数$f_E(p)$为问题规模W随处理器数p变化的函数，为等效率函数（ISO-efficiency Function）（Kumar1987）

等效率度量标准（2）

- 曲线1表示算法具有很好的扩放性；曲线2表示算法是可扩放的；曲线3表示算法是不可扩放的。
- 优点：简单可定量计算的、少量的参数计算等效率函数
- 缺点：如果T_o难以计算出（在共享存储并行机中）



等速度度量标准（1）

- 出发点：对于共享存储的并行机，T难以计算，如果速度能以处理器数的增加而线性增加，则说明系统具有很好的扩放性。
- p表示处理器个数，W表示要求解问题的工作量或称问题规模（在此可指浮点操作个数），T为并行执行时间，定义并行计算的速度V为工作量W除以并行时间T
- p个处理器的并行系统的平均速度定义为并行速度V除以处理器个数 p:

$$\overline{V} \quad \frac{V}{p} \quad \frac{W}{pT}$$

- W是使用p个处理器时算法的工作量，令W’表示当处理数从p增大到p’时，为了保持整个系统的平均速度不变所需执行的工作量，则可得到处理器数从p到p’时平均速度可扩放度量标准公式(介于0与1之间，比值越靠近1越好)

$$(p, p') \quad \frac{W/p}{W'/p'} \quad \frac{p'W}{pW'}$$

等速度度量标准（2）

- 优点：直观地使用易测量的机器性能速度指标来度量
- 缺点：某些非浮点运算可能造成性能的变化没有考虑
- 等速度度量标准的扩放性与传统加速比之间的关系：

当p=1时，等速度度量标准为

$$(p') \qquad (1,p') \frac{W/1}{W'/p'} \quad \frac{p'W}{W'} \quad \frac{T_1}{T_{p'}}$$

解决工作量为W的问题所需串行时间

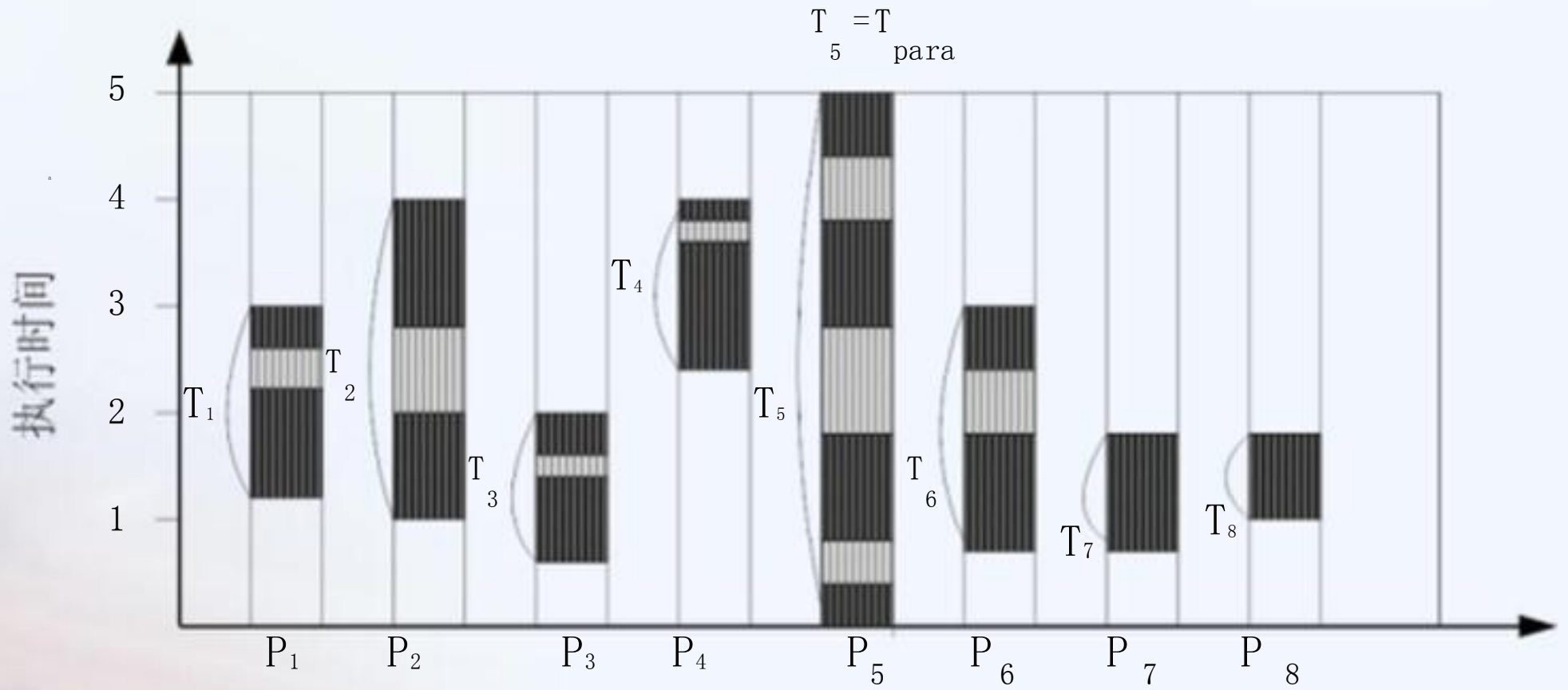
解决工作量为W 的问题所需并行时间

其主要差别：

加速比的定义是保持问题规模不变，标志对于串行系统的性能增加；扩放性定义是保持平均速度不变，标志对于小系统到大规模系统所引起的性能变化。

平均延迟度量标准（1）

- 一个并行系统执行的时间图谱



启动前与结束后空闲时间　　开销延迟 L_i　　运行时间 T_i

平均延迟度量标准（2）

- T_i为P的执行时间，包括延迟L_i，Pi的总延迟时间为“L_i+启动时间+停止时间”。定义系统平均延迟时间为

$$\overline{L}(W,p)=\sum_{i=1}^{p}\left(T_{para}-T_i+L_i\right)\Big/p$$

又有 $pT_{para}=T_o+T_{seq}$，$T_o=p\overline{L}(W,p)$

所以 $\overline{L}(W,p)=T_{para}-T_{seq}/p$

- $\overline{L}(W',p)$在p’个处理器上求解工作量为W’问题的平均延迟当处理数由p变到p’，而维持并行执行效率不变，则定义平均延迟可扩放度量标准为

$$\Phi(E,p,p')=\frac{\overline{L}(W,p)}{\overline{L}(W',p')}$$

- 该值在0、1之间，值越接近1越好。

平均延迟度量标准（3）

- 优点：平均延迟能在更低层次上衡量机器的性能
- 缺点：需要特定的软硬件才能获得平均延迟

Activity 5

. 文献阅读1:

M.D. Hill, M.R. Marty. Amdahl's law in the multicore era. Computer, 2008, 41 (7), 33-38.