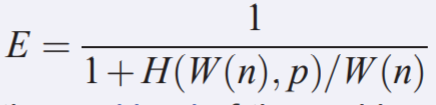
Assignments 2019

1. The execution time of a parallel algorithm using p processors is T(p)=(n^4+n^2)/p+p∙n and the workload of the problem is T(1)=n^4+n^2. What is the scalability of this parallel algorithm? In other words, if p increases, how fast must n increase in order to keep the same efficiency? If the problem size n is increased, can we keep the computing time constant by using more processors? What happens to the parallel efficiency in that case?

**Answer:**

**Scalability means how well the performance of a parallel algorithm scales up with the increase in the number of processors and in the problem size.**

**Iso-efficiency: increase the size of the problem when the number increases in order to keep a constant efficiency where W(n) and H are the workload of the problem and the total overhead respectively. Answer: keep H/W=constant=O(1)**

****

**In this problem, in order to keep a constant efficiency, we must keep H/W=constant=O(1)**

**The (sequential) workload is W= (n^4+n^2)**

**The total overhead is H=p\*T(p)-W=(n^4+n^2) +p^2∙n-(n^4+n^2)= p^2∙n**

**The parallel efficiency is kept constant if H/W is kept constant, thus (p^2∙n)/ (n^4+n^2)=O(1), then n=O(p^(2/3))**

**When problem size is increased, executing time can be kept constant by increasing the amount of processors proportional to the increased work. However, it's impossible in general! This means that to keep a constant efficiency the parallel execution time will increase even if p increases.**

1. What are the main features/components in parallel computer architectures? Briefly describe the Shenwei Taihu Light and the Summit (IBM Power System AC922) in terms of these features/components.

**Answer:**

**Hardware: multi-core and accelerators**

**Parallel & Distributed Computers Architectures**

**Cluster of SMPs**

* **神威·太湖之光：**

**神威·太湖之光超级计算机由40个运算机柜和8个网络机柜组成。每个运算机柜比家用的双门冰箱略大，打开柜门，4块由32块运算插件组成的超节点分布其中。每个插件由4个运算节点板组成，一个运算节点板又含2块“申威26010”高性能处理器。一台机柜就有1024块处理器，整台“神威·太湖之光”共有40960块处理器。每个单个处理器有260个核心，主板为双节点设计，每个CPU固化的板载内存为32GBDDR3-2133，总核数>10.6 millions[1]。**

**系统峰值性能：125.436PFlops  
实测持续运算性能：93.015PFlops  
处理器型号："申威26010" 众核处理器  
整机处理器个数：40960个  
整机处理器核数：10649600个  
系统总内存：1310720 GB  
操作系统：Raise Linux  
编程语言：C、C++、Fortran  
并行语言及环境：MPI、OpenMP、OpenACC等  
SSD存储：230TB  
在线存储：10PB，带宽288GB/s  
近线存储：10PB，带宽32GB/s**

* **IBM Power System AC922：**

**IBM Power System AC922包含种类繁多的下一代 I/O 架构，包括：PCIe gen4、CAPI 2.0、OpenCAPI 和 NVLINK。与 x86 服务器采用的陈旧的 PCIe gen3 相比，这些互连可为当今的数据密集型工作负载提供 2-5.6 倍的带宽[2]。**

**IBM Power System AC922采用据称是最适合企业 AI 的最佳 GPU，堪称最适合它的最佳平台。AC922 还采用了 POWER9 CPU 以及提供 NVLink GPU 的 NVIDIA Tesla V100。这种搭配可提供高达 5.6 倍的性能。它是唯一能在 CPU 和 GPU 之间提供这种 I/O 性能的服务器。这将能够提供大规模的吞吐能力来支持 HPC、深度学习和 AI 工作负载[2]。**

**虽然 POWER9 CPU 的运行速度极快，但它们确实能够充分发挥其周围所有组件的潜力。Power 9 专为 AI 时代而构建，与 x86 竞争产品相比，可提供超过 5.6 倍的 I/O 性能，支持超过 2 倍的线程。Power 9 提供 16、18、20 和 22 核配置，AC922 服务器最多支持 44 核配置[2]。**

**AI 模型越来越大，大多数 x86 服务器中的 GPU 内存容量很快将无法满足其需求。AC922 中 CPU 与 GPU 的一致性解决了这些问题，允许加速后的应用将系统内存用作 GPU 内存。此外，它还能通过消除数据移动需求和定位需求来简化编程工作。此外，通过利用 5.6 倍速的 NVLink 互连线路，CPU 与 GPU 之间的内存共享不会像 x86 服务器那样带来 PCIe 3 速度瓶颈[2]。**

**AC922 是业界首款采用下一代业界标准 PCIe 互连技术的服务器。第四代 PCIe 提供的数据带宽大约是 x86 服务器中采用的第三代 PCIe 互连线路的2倍[2]。**

1. Consider the following OpenMP program.

(a) Is this a correct program? What should be modified to make it more efficient?

/\* C program: Compute the matrix-vector multiplication y=Ax, A is an n-by-n matrix, x and y are vectors of length n. \*/

#include <stdio.h>

#include "omp.h"

#define n 10000

int i, j;

double A[n][n], x[n], y[n];

#pragma omp parallel for

for (i=0; i<n; i=i+1)

for (j=0; j<n; j=j+1)

y[i]=y[i] + A[i][j]\*x[j];

**Answer:**

**No, this is not a correct program, we can change it as the following in order to make it more efficient.**

**#include <stdio.h>**

**#include "omp.h"**

**#define n 10000**

**int i, j;**

**double A[n][n], x[n], y[n];**

**#pragma omp parallel for**

**for (i=0; i<n; i=i+1)**

**for (j=0; j<n; j=j+1)**

**wait(mutex)**

**y[i]=y[i] + A[i][j]\*x[j];**

**signal(mutex)**

(b) Can we use OpenMP directive to execute the following loop in parallel? Please explain.

for (i=2; i<n; ++i)

y[i] = (y[i-2] + y[i-1])/2;

**Answer:**

**No, we can not use OpenMP directive to execute the above loop in parallel. Because loops like**

**...**

**for(i=1;i<n;++i){**

**a[i] = a[i-1]+b[i];**

**...**

**}**

**...**

**and**

**...**

**for(i=1;i<n;++i){**

**a[i] = 0.5\*(a[i-1]+a[i+1]);**

**...**

**}**

**...**

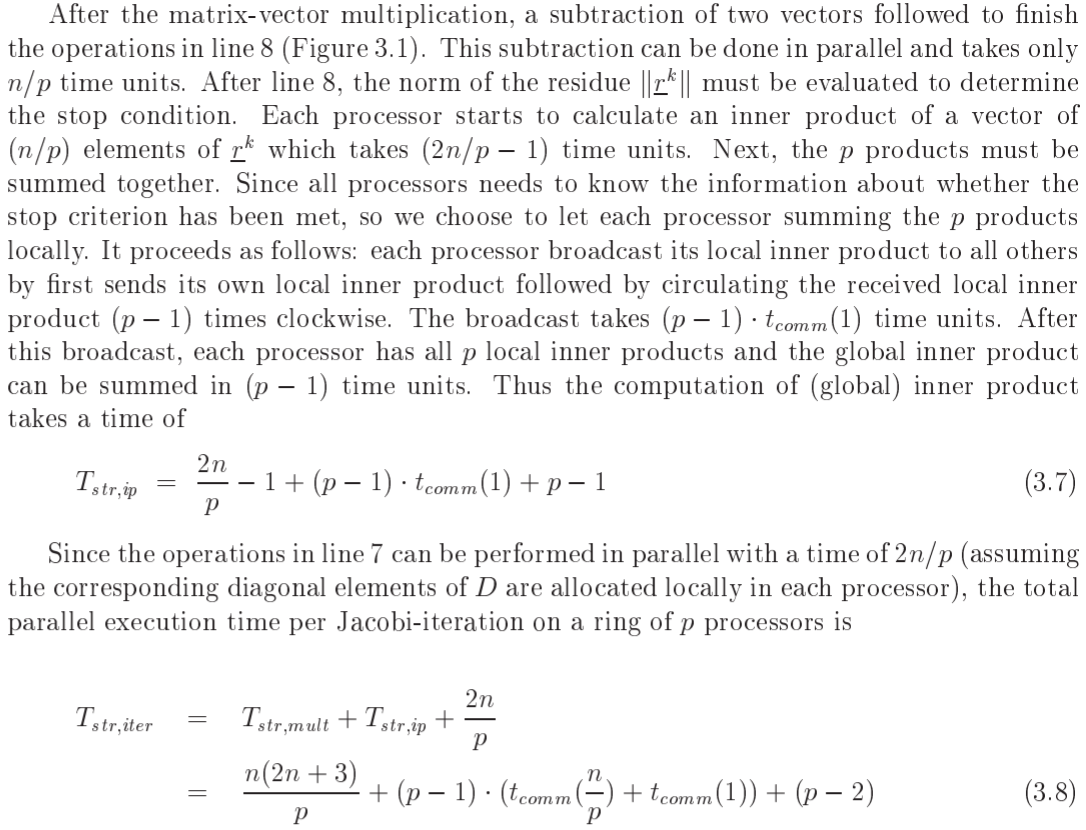
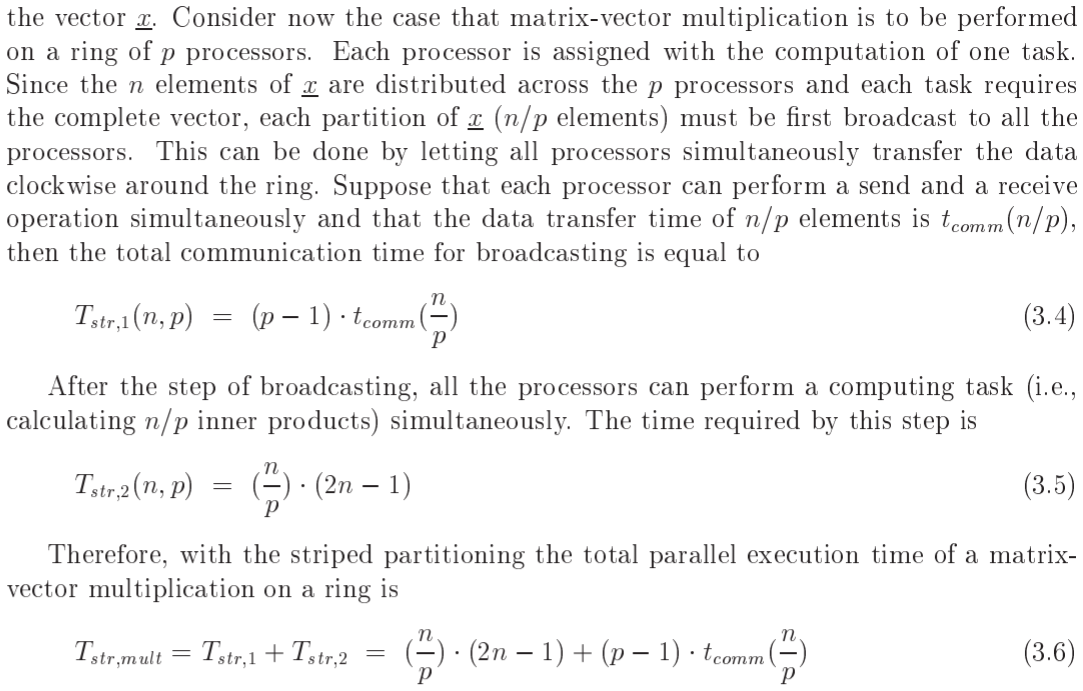
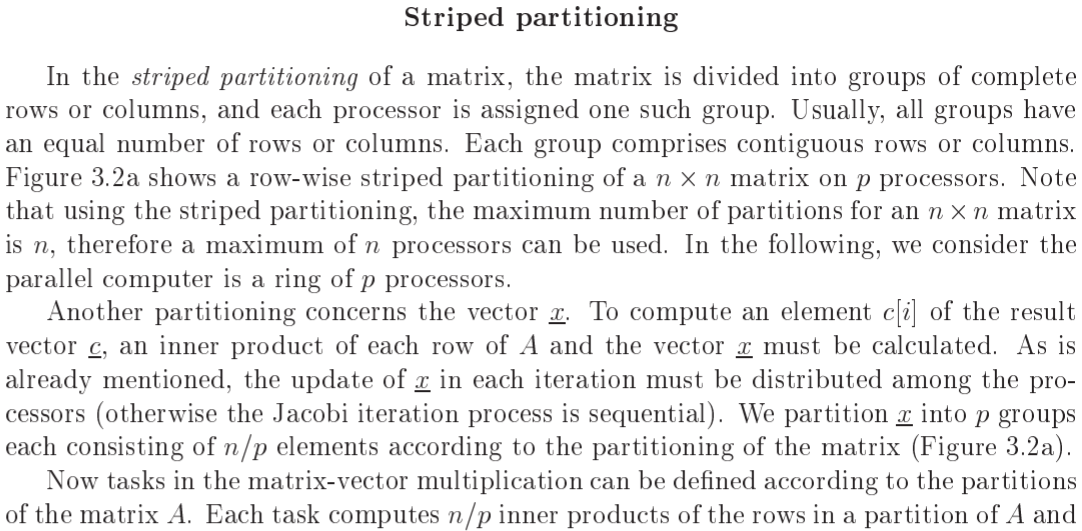
**cannot be parallelized**

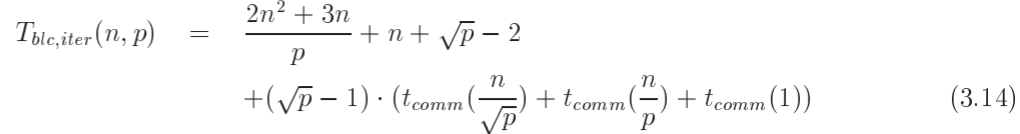
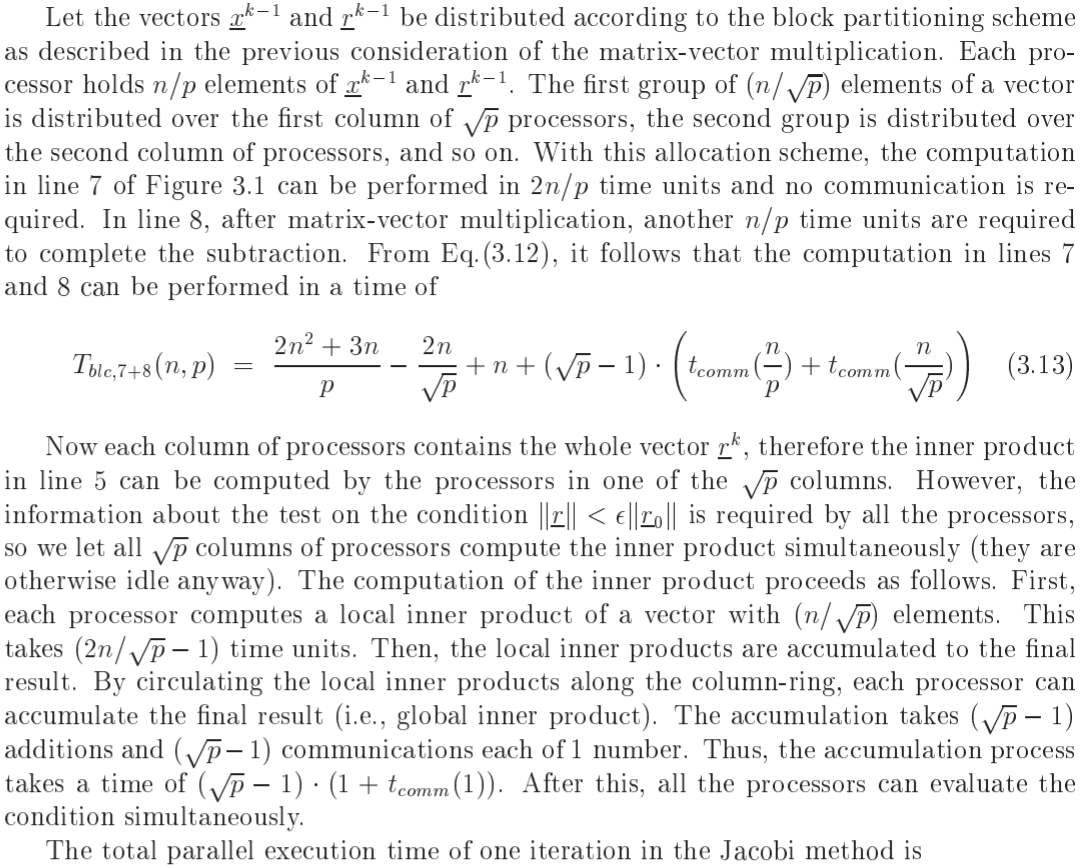
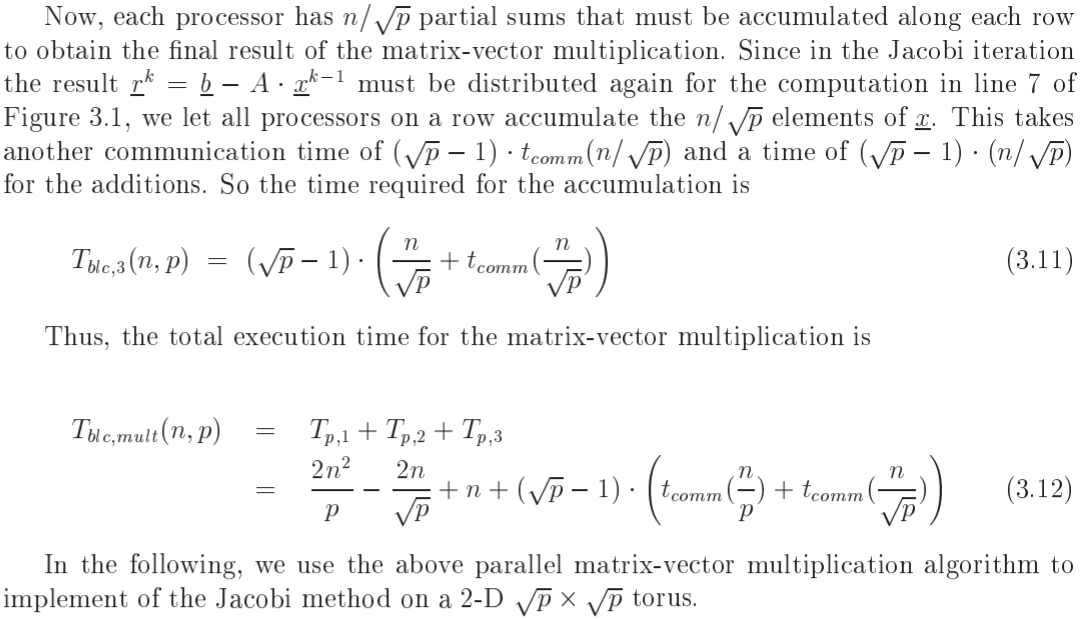
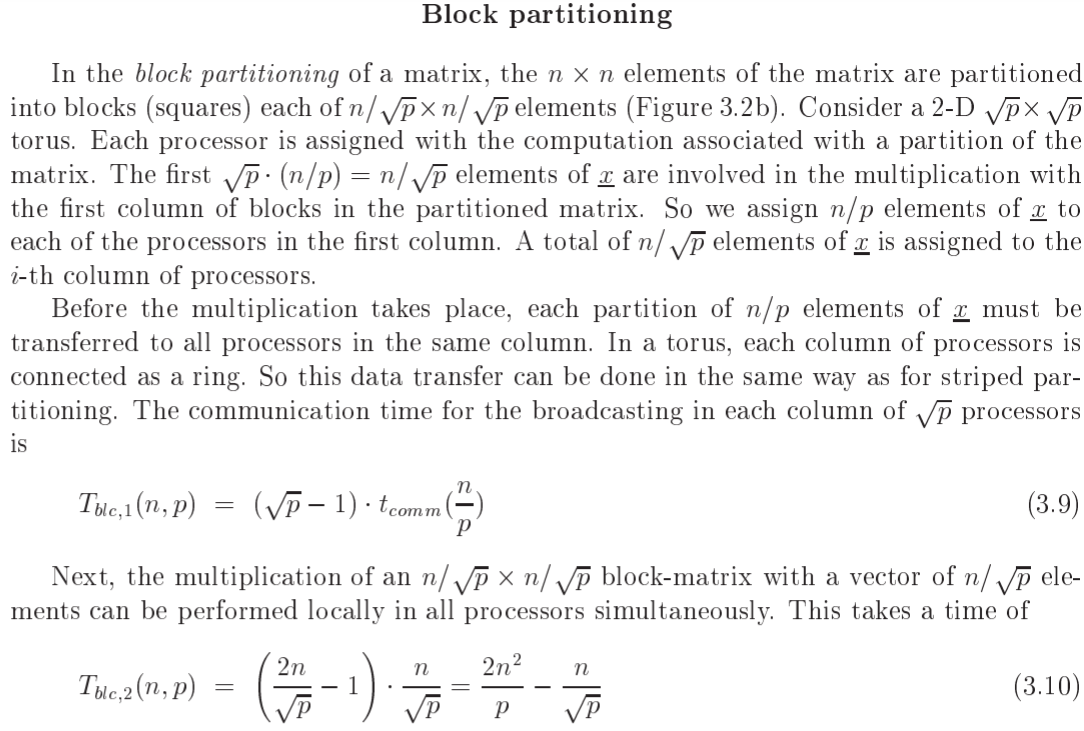
1. Consider a 2-dimensional grid of n by n points. Suppose that we want to parallelize the Jacobi method using the stripe-partitioning and block-partitioning for a parallel computer with P processors. Give the data locality (ratio) of the stripe-partitioning and the block-partitioning

**Answer:**

**Data locality:**

**A parallel algorithm with a large ratio is said to have a high data locality. Higher data locality means that the parallel algorithm is more efficient.**





1. What makes efficient parallelization of a Multi-Grid algorithm more complicated as compared to the Jacobi or Conjugate Gradient algorithm?

**Answer:**

**Jacobi, Conjugate Gradient, or any other sparse-matrix-vector-multiply-based algorithm can only move information one grid-cell at a time, due to local structure of stencil，so it takes O() steps to get information across the grid.**

**Faster convergence in O(1) steps requires moving information across grid faster than to one neighboring grid cell per step. Decreasing error by fixed factor c<1 takes** Ω**(log n) steps can result convergence to a fixed error ε < 1 takes** Ω **(log n ) steps.**

**Multi-Grid replace problem on fine grid by an approximation on a coarser grid.**

**Multi-Grid solve the coarse grid problem approximately, and use the solution as a starting guess for the fine-grid problem, which is then iteratively updated. Multi-Grid solve the coarse grid problem recursively, i.e. by using a still coarser grid approximation, etc.**

**Coarse grid solution being a good approximation to the fine grid makes Multi-Grid a more efficient parallelization algorithm**

1. The force calculation is the dominant computational cost in a conventional (straight-forward implementation) of NBody simulation. For the parallelization of the Barnes-Hut NBody simulation algorithm, will it be sufficient to have only an efficient parallelization of the force calculation step? Explain your answer.

**Answer:**

**No, it is not sufficient to have only an efficient parallelization of the force calculation step.**

**The steps of Barnes-Hut Algorithm as follows:**

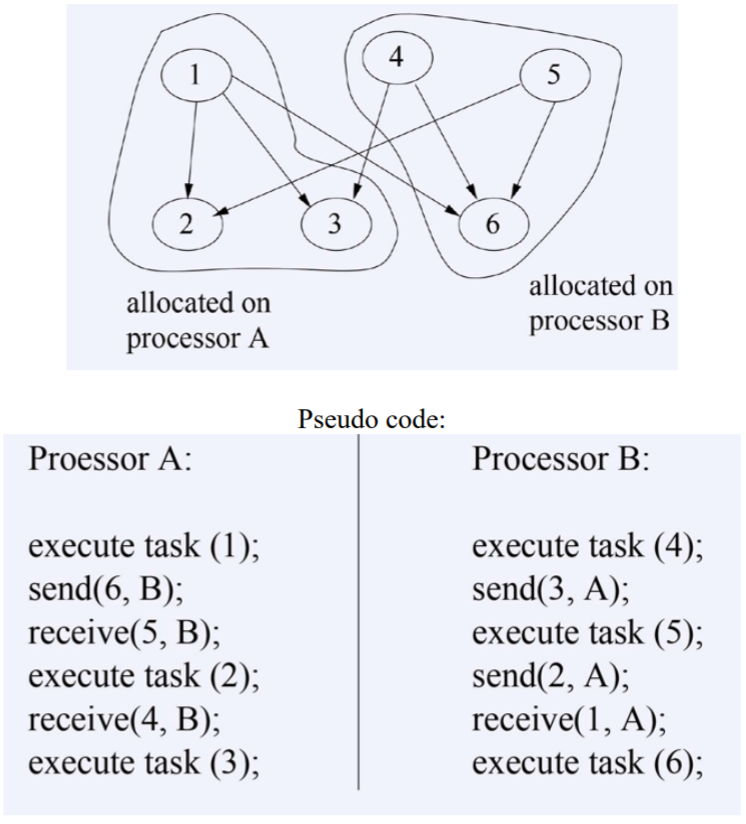
**1) Build the QuadTree using QuadTreeBuild, cost = O( N log N) or O(b N)**

**2) For each node, a sub-square in the QuadTree, Compute the CM and total mass (TM) of all the particles it contains, cost = O( N log N) or O(b N)**

**3) For each particle, traverse the QuadTree to compute the force on it, using the CM and TM of “distant” sub-squares, cost depends on accuracy desired but still O(N log N) or O(bN).**

**Data Insertion in nodes is equally important as force calculation.**

1. We use 2 processors to execute the following task dependence graph, there are 6 tasks, each processor will execute 3 tasks as shown in the figure. Suppose the send and receive are synchronous, is the following pseudo code correct for the parallel execution of the task graph? If not, how would you change it?



**Answer:**

**No, more than one tasks are mapped onto the same processor: Some combination of task-scheduling and the sequence of send and receives may cause deadlock.**

**Processor A:**

**Execute task(1);**

**Send(6,B)**

**Receive(5,B)**

**Execute task(2)**

**Receive(4,B)**

**Execute task(3)**

**Processor B:**

**Execute task(5);**

**Send(2,A)**

**Receive(1,A)**

**Execute task(4)**

**Send(3,A)**

**Execute task(6)**

1. (a) What do we understand under the class of so-called “hierarchical methods or algorithms”? What do these methods have in common?

**Answer:**

**In this class, we have learnt hierarchical methods including Multi-grid, FFT, Multi-level graph partitioning, Barnes-Hut Nbody alg. Hierarchical methods divide the problems into several small blocks. These blocks can be calculated in different processers. hierarchical methods is suitable for parallel computer architectures.**

(b) Compare the parallelization of the Multi-Grid method and the Barnes-Hut algorithm, what makes the parallelization of the Barnes-Hut algorithm more complicated?

**Answer:**

**Multi-Grid replace problem on fine grid by an approximation on a coarser grid.**

**Multi-Grid solve the coarse grid problem approximately, and use the solution as a starting guess for the fine-grid problem, which is then iteratively updated. Multi-Grid solve the coarse grid problem recursively, i.e. by using a still coarser grid approximation, etc.**

**Barnes-Hut Algorithm is called “A Hierarchical O(n log n) force calculation algorithm”. Barnes-Hut Algorithm is good for low accuracy calculations typically. It’s accuracy can easily improved, by changing a single parameter θ, but the cost increases as well.**

**The steps of Barnes-Hut Algorithm as follows:**

**1) Build the QuadTree using QuadTreeBuild**

**2) For each node, a sub-square in the QuadTree, Compute the CM and total mass (TM) of all the particles it contains**

**3) For each particle, traverse the QuadTree to compute the force on it, using the CM and TM of “distant” sub-squares.**

**Multi-Grid is a coarse grid solution, so makes the parallelization of the Barnes-Hut algorithm more complicated.**

(c) Can you think of a problem in your (future) application which might benefit from a hierarchical approach? (in the discussion about the last question, you don’t have to give a solution, just some ideas or thoughts are sufficient).

**Answer:**

**Most neural network is multilevel and have hierarchical structure. The Feedforward and back propagation of these neural networks can be calculated by hierarchical approach, which can accelerate the calculation process.**

**参考文献**

1. **揭秘太湖之光：纯国产如何称雄TOP500？．网易新闻．2016-07-16**
2. **IBM Power System AC922 - 详细信息 - 中国. available at** [**https://www.ibm.com/cn-zh/marketplace/power-systems-ac922/details**](https://www.ibm.com/cn-zh/marketplace/power-systems-ac922/details)