**《并行计算》课程实验报告**

**实验3：基于华为云环境的MPI并行编程**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 郑佳艺 | | 院系 | | 计算机科学与技术 | | | 学号 | | 2021110666 | |
| 任课教师 | | 郝萌 | | | | 指导教师 | 张伟哲 | | | | |
| 实验地点 | | G001 | | | | 实验时间 | 2023/10/18 | | | | |
| 实验课表现 | | 出勤、表现得分 | |  | | 实验报告  得分 |  | | 实验总分 | |  |
| 操作结果得分 | |  | |
| **一、实验目的** | | | | | | | | | | | |
| 要求：需分析本次实验的基本目的，并综述你是如何实现这些目的的？  1．了解华为云环境的使用过程；  2．掌握利用华为云环境搭建小型集群环境的过程；  3．掌握MPI程序设计的基本编写、编译与运行方法;  4．了解集群环境下N体问题的并行程序设计方法；  5．掌握利用加速比、运行时间、效率等测度分析并行程序性能；  6．掌握素数计算程序的基本原理；  7．掌握串行素数计算程序的MPI并行优化方法。  实现过程：  本次实验中，我购买了华为云服务器，并按照要求进行配置。之后远程登录服务器，进行环境配置；  通过老师所给出的实验指导书，了解了N体问题的定义，以及对应的串行计算和并行计算的计算方法。了解性能评估参数：并行算法的加速比定义和并行算法的效率定义。最后配置MPI环境，编译nbody程序，按照运行命令运行程序并改变相应的参数，完成实验数据的记录；  理解了N-2素数计算的串行代码，及其计算原理。按照代码修改提示将代码并行处理，结合2中的代码编译以及文件传送和三台主机并行运算命令行，完成实验数据的记录。修改代码中负载均衡的问题，重复上述步骤，记录实验数据。 | | | | | | | | | | | |
| **二、实验内容** | | | | | | | | | | | |
| 该部分填写在实验过程中，你都完成了哪些工作。  1．华为云实验环境说明：首先进行基础配置、网络配置、高级配置，远程登陆，进行环境配置和创建用户，最后进行免密配置，实现三个不同主机之间的互联。在免密配置中要生成本地密钥，然后添加公钥到所有主机，安装依赖包，配置环境变量，执行source命令，使之生效。  2．N体问题是研究物体之间相互作用力产生的效果的问题。该实验通过计算机模拟计算物体受到的总的引力。老师所给出的并行代码算法描述如下：  获取分配给本进程的物体的初始信息localparticles；  获取应用程序中所有物体的信息allparticles；  for (每一个时间步){  计算所有物体对分配给本进程的物体的作用力并据此更新localparticles的本进程的物体的信息；  将本进程信息localparticles保存到发送缓冲区sendbuf，同时更新allparticles中的部分信息；  for ( i = 0; i < m - 1; i++ ) { //对每个进程  send sendbuf to 本进程的下一个进程；  recv recvbuf from 本进程的前一个进程；  用recvbuf中的信息更新allparticles中的部分信息；  }  使所有进程在此处同步；  }  在本实验中我了解了MPI程序设计的基本编写、编译与运行方法，以及计算相应的性能评估参数。编译nbody程序，按照运行命令运行程序并改变相应的参数，完成了实验数据的记录。  3．实验三中，我理解了N-2素数计算的串行代码及其计算原理。按照代码修改提示将代码并行处理。我按照以下的代码修改提示完成了对代码的修改，对prime-c程序并行化：   1. 调用MPI头文件； 2. 在并行处理之前调用MPI\_Init()，MPI\_Comm\_size()，MPI\_Comm\_rank() 函数； 3. 在并行处理之后调用MPI\_Finalize()； 4. 调用MPI\_Reduce()收集各进程的计算结果； 5. 进程0打印最终的计算结果； 6. 如果N=100000，则素数的数量应为9592。可以申请1个进程来运行程序，检查计算结果是否正确； 7. 如果程序计算结果无误，就需要评估其性能。利用MPI\_Wtime()来统计时间，并利用进程0来输出程序的执行时间； 8. 可以使用命令行输入参数 N，支持设置不同的进程数来运行程序，避免每次修改 N 后重新编译与拷贝操作。   int n = atoi(argv[1]);   1. 编译与运行命令仿照 N 体问题中的进行操作。   完成实验数据的记录。我修改了代码中负载均衡的问题，结合2中的代码编译以及文件传送和三台主机并行运算命令行，记录实验数据执行代码并记录数据，计算加速比。  之后按照提示，解决程序的负载均衡问题，再次运行程序，记录数据并计算加速比。  由于线程号大的线程分配到的数据大，计算困难，所以通过增加线程号小的线程计算量，减少线程号大的线程的计算量来完成计算的负载均衡的问题。 | | | | | | | | | | | |
| **三、实验结果** | | | | | | | | | | | |
| **N体问题MPI并行程序剖析与性能分析：**  实验一：**单机**上，数据规模为6000时，随每机进程数变化的运行时间；   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 进程数 | 1 | 2 | 3 | 4 | | 时间 | 7.500409 | 3.375833 | 4.471386 | 4.087556 |   实验二：**相同数据规模为6000**，随每机进程数变化的运行时间   |  |  |  |  | | --- | --- | --- | --- | | 每机进程数 | 单机 | 双机 | 三机 | | 1 | 7.500409 | 3.761481 | 2.516216 | | 2 | 3.375833 | 1.904001 | 1.298974 | | 3 | 4.471386 | 2.866867 | 2.173173 | | 4 | 4.087556 | 2.527404 | 2.099484 |   实验三：**每机1个进程**，随数据规模变化的n-body并行程序运行时间。   |  |  |  |  | | --- | --- | --- | --- | | 粒子数n | 单机 | 双机 | 三机 | | 150 | 0.004823 | 0.005548 | 0.015196 | | 300 | 0.019002 | 0.016102 | 0.016374 | | 600 | 0.075368 | 0.054370 | 0.033125 | | 1200 | 0.300542 | 0.157572 | 0.107290 | | 2400 | 1.204283 | 0.610015 | 0.416872 | | 4800 | 4.800841 | 2.413520 | 1.611246 | | 9600 | 19.199679 | 9.614335 | 6.422893 |   根据数据计算加速比与效率：Sp=T1/Tp Ep=Sp/P  实验一：**单机**上，粒子数为6000，随进程数变化加速比（Sp）统计、效率统计。  加速比（Sp）统计表   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 进程数 | 1 | 2 | 3 | 4 | | 加速比 | 1 | 2.221795 | 1.677424 | 1.834937 |   效率（Ep）统计表   |  |  |  |  |  | | --- | --- | --- | --- | --- | | 进程数 | 1 | 2 | 3 | 4 | | 效率 | 1 | 1.110898 | 0.559141 | 0.458734 |   图1 单机上，粒子数为6000，随进程数变化加速比（Sp）效率（Ep）统计  实验二：**粒子数为6000**，随每机进程数变化的加速比与效率。  加速比（Sp）统计表   |  |  |  |  | | --- | --- | --- | --- | | 每机进程数 | 单机 | 双机 | 三机 | | 1 | 1 | 1.994004 | 2.980829 | | 2 | 2.221795 | 3.939288 | 5.774102 | | 3 | 1.677424 | 2.616239 | 3.451363 | | 4 | 1.834937 | 2.967634 | 3.572501 |   图2 粒子数为6000，随每机进程数变化的加速比  效率（Ep）统计表   |  |  |  |  | | --- | --- | --- | --- | | 每机进程数 | 单机 | 双机 | 三机 | | 1 | 1 | 0.997002 | 0.993610 | | 2 | 1.110898 | 0.984822 | 0.962350 | | 3 | 0.559141 | 0.436040 | 0.383485 | | 4 | 0.458734 | 0.370954 | 0.297708 |   图3 粒子数为6000，随每机进程数变化的效率  实验三：**每机1个进程**，随数据规模变化的n-body并行程序加速比和效率。  加速比（Sp）统计表   |  |  |  |  | | --- | --- | --- | --- | | 粒子数n | 单机 | 双机 | 三机 | | 150 | 1 | 0.869322 | 0.317386 | | 300 | 1 | 1.180102 | 1.160498 | | 600 | 1 | 1.386206 | 2.27526 | | 1200 | 1 | 1.907331 | 2.801212 | | 2400 | 1 | 1.974186 | 2.888856 | | 4800 | 1 | 1.989145 | 2.979583 | | 9600 | 1 | 1.996985 | 2.989257 |   图4 每机1个进程，随数据规模变化的n-body并行程序加速比  效率（Ep）统计表   |  |  |  |  | | --- | --- | --- | --- | | 粒子数n | 单机 | 双机 | 三机 | | 150 | 1 | 0.434661 | 0.105795 | | 300 | 1 | 0.590051 | 0.386833 | | 600 | 1 | 0.693103 | 0.75842 | | 1200 | 1 | 0.953666 | 0.933737 | | 2400 | 1 | 0.987093 | 0.962952 | | 4800 | 1 | 0.994572 | 0.993194 | | 9600 | 1 | 0.998492 | 0.996419 |   图5 每机1个进程，随数据规模变化的n-body并行程序效率  **利用MPI对素数计算程序进行并行优化：**  N在不同取值下的运行时间   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 进程数 | 串行执行 | 1 | 2 | 4 | 6 | | N | | 100000 | 1.058535 | 0.004168 | 0.000434 | 0.068222 | 0.107380 | | 200000 | 3.973990 | 0.009300 | 0.009334 | 0.076914 | 0.083870 | | 400000 | 15.006640 | 0.023530 | 0.040555 | 0.032013 | 0.114971 | | 800000 | 56.828075 | 0.061462 | 0.060502 | 0.088315 | 0.143362 |   N在不同取值下的加速比   |  |  |  |  |  | | --- | --- | --- | --- | --- | | N进程数 | 进程数1 | 进程数2 | 进程数4 | 进程数6 | | 100000 | 253.967131 | 2439.02074 | 15.5160359 | 9.857841 | | 200000 | 427.310753 | 425.754232 | 51.6679668 | 47.38274 | | 400000 | 637.766256 | 370.031809 | 468.767063 | 130.5254 | | 800000 | 924.605041 | 939.275974 | 643.470249 | 396.3957 |   修改程序负载均衡之后：  N在不同取值下的运行时间   |  |  |  |  |  | | --- | --- | --- | --- | --- | | N 进程数 | 1 | 2 | 4 | 6 | | 100000 | 0.002370 | 0.001570 | 0.087316 | 0.060298 | | 200000 | 0.004965 | 0.007931 | 0.020411 | 0.107377 | | 400000 | 0.011667 | 0.010852 | 0.055624 | 0.099576 | | 800000 | 0.029441 | 0.036901 | 0.080002 | 0.130971 |   N在不同取值下的加速比   |  |  |  |  |  | | --- | --- | --- | --- | --- | | N 进程数 | 进程数1 | 进程数2 | 进程数4 | 进程数6 | | 100000 | 446.639241 | 674.226115 | 12.1230359 | 17.55506 | | 200000 | 800.400806 | 501.070483 | 194.698447 | 37.00969 | | 400000 | 1286.24668 | 1382.84556 | 269.787142 | 150.7054 | | 800000 | 1930.2359 | 1540.0145 | 710.333179 | 433.8982 | | | | | | | | | | | | |
| **四、思考题** | | | | | | | | | | | |
| 思考题1：从算法层面上分析素数计算程序负载不均衡的原因，并提出改进方法。  程序需要检查从2到N的整数，为此首先将整数分解为P个子列表。例如，假设P为4，N为15：  Part 0: 2, 6, 10, 14  Part 1: 3, 7, 11, 15  Part 2: 4, 8, 12  Part 3: 5, 9, 13  由于在分配内存时按序分配，会导致偶数线程中的数据全部都是偶数，直接在j=2的时候直接能判断不是素数，所以进度快；而奇数线程中没有偶数，计算量大。  改进方法：按块分配任务，总量为N，P个线程，则线程i负责（i\*N/P，（i+1）\*N/P）每个线程均有偶数与奇数，负载相比之前更加均衡。    思考题2：根据该实验，简述并行程序编程和优化的步骤。  并行程序编程：  1. 确定并行度：确定程序中可以并行执行的部分，以及如何将程序分解为多个并行任务。  2. 确定通信方式：确定并行任务之间的通信方式，以便它们可以协同工作。  3. 选择编程模型：选择适合问题的编程模型，例如MPI，OpenMP，Pthread等共享内存、分布式内存或混合模型。  4. 编写并调试代码  5. 优化代码：通过使用高效算法、减少通信和避免竞争条件等技术来优化代码，同时注意不同线程之间的负载均衡问题，在代码实现的过程中要加以考虑。  并行程序优化角度：  1．确定瓶颈：确定程序中最耗时的部分。  2．减少通信：减少任务之间的通信量，以减少通信开销。  3．避免竞争条件：避免多个任务同时访问共享资源，以避免竞争条件。  4．调整负载平衡：调整任务之间的负载平衡，以确保所有处理器都得到充分利用。 | | | | | | | | | | | |
| **五、实验心得体会** | | | | | | | | | | | |
| 在本次实验中，我初步了解并使用了MPI进行编程，并分析了程序的负载问题。  在实验二中，我发现并不是线程数越多，加速比和效率越高。因为除了计算部分，进程数多了会有额外的通信开销，而且比物理核心数多的话，会有进程切换开销等等，因此并不是进程数越多，就一定能越快。  在实验三中，我们分析了负载均衡的问题，因此在并行化的同时我们要考虑平均性与并行化的合理性，分配给每个进程相似的任务量，才能更好的利用并行计算来提高程序运行加速比与效率。 | | | | | | | | | | | |
| 指导教师评语：  日期： | | | | | | | | | | | |