

MPI矩阵乘法

元凯文 200110213

一、实验内容

用MPI（进程间通信框架）实现矩阵乘法的并行计算

MPI简介：MPI,即 Message Passing Interface，用于进程间通信的编程框架，六个常用 API（初始化，获取当前进程标识，通信器包含进程数，结束 MPI，点对点通信两个接口函数）

二、设计方案

1. 划分

按列划分，每个进程需要计算的 $c[i][j]$ 个数为 $size \times size \div (proNum - 1)$ ，其中 $size$ 为规模， $proNum$ 为进程总个数，减去1为主进程不参与计算

2. 通信

采用点对点通信，借助MPI_Send,MPI_Recv等直接将子进程计算的MPI_INT类型结果数组发送给主进程

三、主要代码

```
if (myid != 0)
{
    int start = (myid - 1) * block; //当前进程需要计算的首个位置
    int end = start + block;       //当前进程需要计算的末尾位置
    int px = 0; //游标表明当前已计算个数

    while (start < end)
    {
        int i = start % size;
        int j = start / size;
        temp[px] = 0;
        for (int k = 0; k < size; k++)
        {
            temp[px] += A[i][k] * B[k][j];
        }
        //printf("In %d:temp[%d]=%d\n",myid,px,temp[px]);
        px++;
        start++;
    }

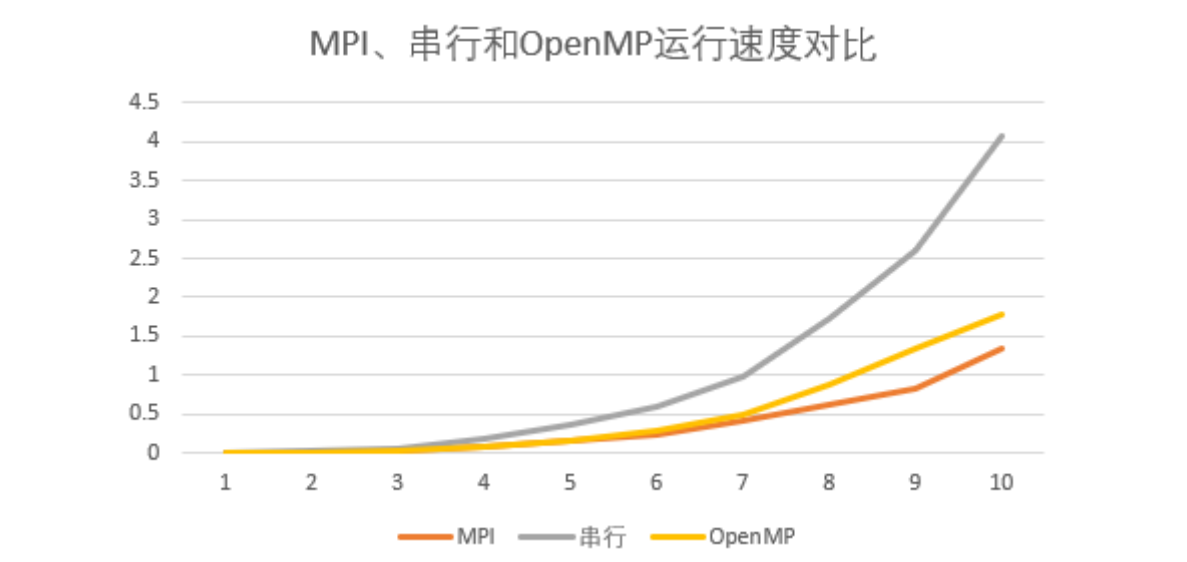
    MPI_Send(temp, px, MPI_INT, 0, 99, MPI_COMM_WORLD); //通信发送
}
else
{
    double start=MPI_Wtime();
    for (source = 1; source < numprocs; source++)
    {
```

```
        MPI_Recv(temp, block, MPI_INT, source, 99, MPI_COMM_WORLD, &status);
//通信接收
int start = (source - 1) * block;
int end = start + block;
int px = 0;

//拼合各进程计算结果
while (start < end)
{
    int i = start % size;
    int j = start / size;
    C[i][j] = temp[px];
    start++;
    px++;
}
}
```

三、数据处理

数据规模	100	200	300	400	500	600	700	800	900	1000
MPI	0.002	0.011	0.033	0.071	0.154	0.241	0.415	0.622	0.819	1.329
串行	0.002	0.017	0.063	0.170	0.361	0.584	0.979	1.735	2.611	4.081
OpenMP	0.001	0.011	0.032	0.089	0.154	0.290	0.482	0.866	1.341	1.772



mpi需要理论内存为273KB(size=100)，4个进程实际内存为 600KB左右

四、总结

通过MPI通信框架，可以实现更加安全和高效的进程间数据通信，其中不仅有点对点通信，还有集合通信，对于矩阵乘法这样有较多共享数据的计算而言，似乎集合间通信更合理一些。

