

برنامه‌سازی موازی

گزارش پروژه‌ی چهارم



سید پارسا حسینی‌نژاد 810196604

کیمیا خیری 810196606

سوال اول

میخواهیم برنامه‌ای بنویسیم که به دو روش سریال و موازی بزرگترین عنصر یک آرایه‌ی ۲ به توان ۲۰ خانه‌ای به همراه ایندکس آن را پیدا کند.

در حالت سریال باید روی تمام عناصر این آرایه حلقه‌ی `for` بزنیم و عنصری که از تمام عناصر دیگر بزرگتر است را پیدا کنیم. پیاده‌سازی روش سریال در تصویر زیر آمده است.

```
for (i = 0; i < DATA_SIZE; i++)
{
    if (a[i] > maxS) {
        maxS = a[i];
        maxSIdx = i;
    }
}
```

در حالت موازی آرایه را به ۴ بخش تقسیم می‌کنیم و مسئولیت پیدا کردن ماکزیموم در هر بخش را به یک ترد می‌سپاریم. در نهایت بزرگترین عدد از میان ۴ عدد پیدا شده ماکسیموم آرایه است. برای پیاده‌سازی به روش موازی از کتابخانه‌ی `openMP` استفاده شده که این پیاده‌سازی در تصویر زیر قابل مشاهده است.

```
#pragma omp parallel num_threads(4) shared(a, maxP, maxPIdx) private(maxLocal, maxPIdxLocal)
{
    maxLocal = a[0];
    #pragma omp for
    for (i = 0; i < DATA_SIZE; i++) {
        if (a[i] > maxLocal) {
            maxLocal = a[i];
            maxPIdxLocal = i;
        }
    }

    #pragma omp critical (maxP)
    {
        if (maxP < maxLocal) {
            maxP = maxLocal;
            maxPIdx = maxPIdxLocal;
        }
    }
}
```

در نهایت speedup روش موازی نسبت به سریال چیزی حدود 3.7 برابر بدست آمد.

```
~/Desktop/Parallel Programming/CA4/Q1 > g++ -o main main.cpp -fopenmp
~/Desktop/Parallel Programming/CA4/Q1 > ./main
Kimia Khabiri: 810196606 - Parsa Hoseininejad: 810196604
Parallel output: 1049086.750000 with index: 158058, Serial output: 1049086.750000 with index: 158058
Serial Run time = 3701
Parallel Run time = 996
Speedup = 3.715863
```

سوال دوم

میخواهیم برنامه‌ای بنویسیم که به دو روش سریال و موازی آرایه‌ای با ۲ به توان ۲۰ عدد ممیز شناور را مرتب کند.

در روش سریال quick sort روی تمام آرایه اعمال شده است. پیاده‌سازی تابع quicksort و اعمال آن روی تمامی آرایه در کدهای ضمیمه قابل مشاهده است.

جهت بهبود بخشیدن در سرعت مرتب‌سازی آرایه، در روش موازی به جای اعمال quicksort روی تمامی آرایه، آرایه را به ۴ بخش تقسیم کردیم و به هر ترد وظیفه‌ی مرتب کردن یکی از بخش‌ها به وسیله‌ی روش quick sort را سپرده‌ایم. در نهایت خروجی‌های تولید شده توسط ترد‌ها را به وسیله‌ی merge sort مرتب کردیم و به آرایه‌ی مرتب نهایی رسیدیم. مراحل گفته شده با استفاده از کتابخانه‌ی openMP پیاده‌سازی شده و در کد ضمیمه قابل مشاهده است.

در نهایت speedup روش موازی نسبت به سریال چیزی حدود 2.8 برابر بدست آمد.

```
~/Desktop/Parallel Programming/CA4/Q2 g++ -o main main.cpp -fopenmp
~/Desktop/Parallel Programming/CA4/Q2 ./main
Kimia Khabiri: 810196606 - Parsa Hoseininejad: 810196604
Serial Run time = 221119
Parallel Run time = 78897
Speedup = 2.802629
```

سوال سوم

میزان تسریع در حالت static چیزی حدود 1.9 برابر و در حالت های دینامیک حدود 3.2 برابر است. تفاوت هم به این دلیل است که در حالت static به ترد های مختلف میزان کار متفاوتی میرسد و کار به خوبی بین ترد ها پخش نشده که از میزان اجرای هر ترد در بخش static این موضوع مشخص است. اما در حالت دینامیک، کارها بین ترد ها تقریباً به صورت یکسان پخش میشود (همان طور که از زمان اجرای هر ترد مشخص است) و به همین دلیل برنامه بهتر عمل میکند.

```
~/Desktop/Parallel Programming/CA4/Q3 g++ -o 3_1 question3_1.cpp
~/Desktop/Parallel Programming/CA4/Q3 ./3_1
Serial timing for 100000 iterations
Time Elapsed 26053 mSecs Total=32.617277 Check Sum = 100000
```

```
~/Desktop/Parallel Programming/CA4/Q3 g++ -o 3_2 question3_2.cpp -fopenmp
~/Desktop/Parallel Programming/CA4/Q3 ./3_2
OpenMP Parallel Timings for 100000 iterations, static
Thread report time: Thread 0 had time 2078, Thread 1 had time 5940, Thread 2 had time 9773, Thread 3 had time 13272,
Time Elapsed 13274 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 2071, Thread 1 had time 5914, Thread 2 had time 9855, Thread 3 had time 13507,
Time Elapsed 13507 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 2062, Thread 1 had time 6049, Thread 2 had time 10155, Thread 3 had time 13308,
Time Elapsed 13309 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 2231, Thread 1 had time 6171, Thread 2 had time 9973, Thread 3 had time 13547,
Time Elapsed 13548 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 2092, Thread 1 had time 6296, Thread 2 had time 10067, Thread 3 had time 13499,
Time Elapsed 13499 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 2138, Thread 1 had time 6307, Thread 2 had time 10149, Thread 3 had time 13694,
Time Elapsed 13694 mSecs Total=32.617277 Check Sum = 100000
```

```
~/Desktop/Parallel Programming/CA4/Q3 g++ -o 3_2 question3_2.cpp -fopenmp
~/Desktop/Parallel Programming/CA4/Q3 ./3_2
OpenMP Parallel Timings for 100000 iterations, (dynamic, 1000)
Thread report time: Thread 0 had time 7632, Thread 1 had time 8092, Thread 2 had time 7820, Thread 3 had time 7981,
Time Elapsed 8093 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 8280, Thread 1 had time 7832, Thread 2 had time 8190, Thread 3 had time 8002,
Time Elapsed 8280 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 7965, Thread 1 had time 8298, Thread 2 had time 8511, Thread 3 had time 8168,
Time Elapsed 8511 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 8455, Thread 1 had time 8120, Thread 2 had time 8671, Thread 3 had time 8317,
Time Elapsed 8671 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 8847, Thread 1 had time 8696, Thread 2 had time 8424, Thread 3 had time 8556,
Time Elapsed 8847 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 8869, Thread 1 had time 8731, Thread 2 had time 8797, Thread 3 had time 9007,
Time Elapsed 9007 mSecs Total=32.617277 Check Sum = 100000
```

```
~/Desktop/Parallel Programming/CA4/Q3 g++ -o 3_2 question3_2.cpp -fopenmp
~/Desktop/Parallel Programming/CA4/Q3 ./3_2
OpenMP Parallel Timings for 100000 iterations, (dynamic, 2000)
Thread report time: Thread 0 had time 8112, Thread 1 had time 7481, Thread 2 had time 8388, Thread 3 had time 7768,
Time Elapsed 8389 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 7717, Thread 1 had time 7426, Thread 2 had time 8028, Thread 3 had time 8319,
Time Elapsed 8319 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 7421, Thread 1 had time 8340, Thread 2 had time 7688, Thread 3 had time 8029,
Time Elapsed 8340 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 8180, Thread 1 had time 8379, Thread 2 had time 8663, Thread 3 had time 7735,
Time Elapsed 8663 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 8795, Thread 1 had time 8642, Thread 2 had time 9067, Thread 3 had time 8165,
Time Elapsed 9067 mSecs Total=32.617277 Check Sum = 100000
Thread report time: Thread 0 had time 8382, Thread 1 had time 8066, Thread 2 had time 8691, Thread 3 had time 9031,
Time Elapsed 9031 mSecs Total=32.617277 Check Sum = 100000
```