

به نام خدا

گزارش کار آزمایش دوم درس برنامه نویسی چند هسته ای

امیرمحمد پیرحسین لو ۹۵۳۱۰۱۴

مهدی صفری

A. add elements of a row together

size	1 MB		461.57 MB		تسریع
thread- chunk size	1	128	1	128	
1	0.0019	0.0023	0.6938	0.6875	
4	0.0015	0.0019	0.3842	0.3522	
8	0.0013	0.0014	0.3657	0.3583	
16	0.0013	0.0014	0.3604	0.3581	

B. add elements of a column together

size	1 MB		461.57 MB		تسریع
thread- chunk size	1	128	1	12	
1	0.0070	0.0067	5.3243	6.2052	
4	0.0069	0.0049	3.8877	3.5165	
8	0.0045	0.0040	3.2234	3.6146	
16	0.0046	0.0040	3.2812	3.3358	

C. add elements of a block together

size	1 MB		461.57 MB		تسریع
thread- chunk size	1	128	1	128	
1	0.0074	0.0078	0.8468	3.1739	
4	0.0494	0.0569	1.4	1.3469	
8	0.0479	0.0466	1.3362	1.4010	
16	0.0149	0.1318	3.3338	3.1960	

- D. تجزیه ستونی زیرا miss cache در آن خیلی زیاد است.
- E. زمان اجرای کل عملیات زیاد می شود اما تاثیری در سرعت ندارد زیرا سرعت به الگوریتم و سخت افزار بستگی دارد که در هر دو حالت یکسان هستند.
- F. وقتی که تعداد نخ ها برابر ۸ است بیشترین میزان تسریع را داریم . تعداد نخ ها برای دست یابی به میزان بیشترین تسریع به تعداد هسته های cpu بستگی دارد.
- G. روش ایستا، زیرا نخ ها کار یکسانی انجام می دهند و با گذر زمان محاسبات نخ ها پیچیده تر نمی شود.
- H. در این برنامه تجزیه سطری بهتر عمل می کند زیرا میزان cache hit خیلی زیاد است .
- تجزیه یک بعدی cache hit بیشتری دارد در مقابل هم تجزیه دو بعدی این امکان را می دهد تا بتوان کار ها را بین نخ ها متعادل (balanced) تقسیم کرد.

I. Collapse addition

size	1 MB		461.57 MB		تسریع
thread-chunk size	1	128	1	128	
1	0.0040	0.0020	1.4370	0.8130	
4	0.0060	0.0020	0.6360	0.5280	
8	0.0030	0.0020	0.6830	0.4680	
16	0.0020	0.0030	1.2590	0.4690	

در این روش نسبت به همه ی روش های قبلی (۳ روش صفحه قبل) بهبود بیشتری حاصل شد و زمان اجرا به شکل چشمگیری کاهش یافت. به دلیل پیاده سازی ویژگی collapse توسط خود openmp و استفاده از library های از قبل آماده و optimize می توان تا حدودی این افزایش کارایی را توجیه کرد.