1024

workers: 1 2 4 8

8.3443894.6352012.3482712.0836038.7475034.4316952.2265152.1111568.2695764.265122.1649182.0525778.5799314.3280462.1781832.0513748.4405594.2708522.1930892.0473458.3069644.3795092.2136562.0827138.1853634.3262552.1513892.0327518.6158354.3440832.1430322.0831128.3077724.2469712.1012742.1058258.3452344.3841672.1971082.176542

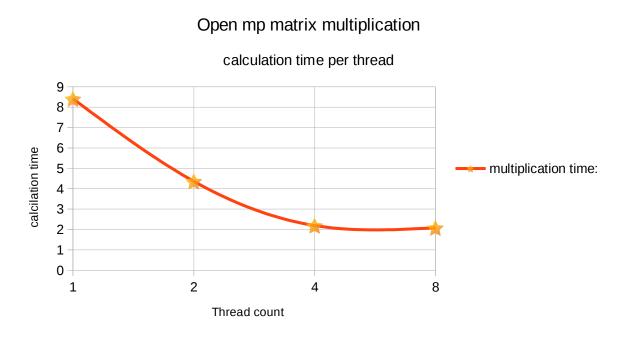
8.4143126 4.3611899 2.1917435 2.0826998 1 2 3 4

threads: multiplication time:

1 8.414313
2 4.36119
4 2.191744

8

2.0827



۱. تبدیل مدل سریال ضرب ماتریس ها به مدل پارالل **open mp** تنها با اضافه کردن سه خط به کد برنامه و اضافه کردن یک فلگ به دستور کامیایل امکان پذیر است.

۲. در مدل **omp** برنامه، بهینه ترین روش، استفاده از **for** موازی در تابع ضرب است. حالت دیگر **task** بود که کد برنامه خیلی تغییر می کند.

۳. در directive دستور for تمام حالت های scheduling ارزیابی شدند که به این علت که عملیات درون for کاملا همریخت هستند، تغییر محسوسی در زمان عملیات مشاهده نشد. البته انتظار می رود روش guided بهترین کارایی را (شاید در ماتریس های خیلی بزرگ تر) داشته باشد.

بنیازی به تعریف private بودن متغیر ها نیست چون این کار به صورت پیش فرض انجام می شود. همچنین هیچ data race و یا false sharing وجود ندارد.
 البته شاید بتوان cache miss را کمی کاهش داد که detect کردن این موضوع بسیار دشوار است.

Speed up.۵ کسب شده خطی است و فقط در ۸ thread به دلیل وجود ۴ هسته فیزیکی تنها حدود /۱۰ ثانیه سرعت محاسبات کاهش یافت

تحلیل برنامه: (مرحله ۳)

تحلیل برنامه: (مرحله ۲)

reducer.۱ یک عمل ریاضی است که به صورت اتمیک انجام می گیرد

پس در ساده ترین حالت به وسیله قرار دادن عملیات در **mutex** این کار امکان پذیر خواهد بود.

اگر بخواهیم این کار را به صورت بهینه تر انجام دهیم، اگر به قفل های سخت افزاری پردازنده دسترسی داشته باشیم می توانیم از آن ها استفاده کنیم.

۲. هرسه این ابزار ها مزیت هایی دارند که می توان از آن ها در موقعیت های مختلفی استفاده کرد:

مثلا **pthread** در زمانی که بخواهیم یک الگوریتم مشخص و دارای پیچیدگی کم را پیاده سازی کنیم بهترین ابزار است چون به برنامه نویس این امکان را می دهد تا حداکثر موازی سازی را انجام دهد در حالی که از تمام اتفاقاتی که در برنامه می افتد نیز آگاه است

Openmpبهترین گزینه است، چون با استفاده از قابلیت هایpthread با کمترین تغییر ات بهترین گزینه است. یکی از مزیت های ompموازی سازی بسیار سریع حلقه این امر امکان پذیر است. یکی از مزیت های مهم ompموازی سازی بسیار سریع حلقه های for است. چیزی که در Cilkمحدودیت های زیادی دارد

کامپایلر Cilk برای نصب و راه اندازی به حدود ۱ ساعت زمان نیاز دارد، در حالی که دو مورد دیگر نصب شده هستند. اما در برنامه های functional و یا برنامه هایی که بیشتر به صورت task پیاده سازی شده اند. Cilk کارایی خوبی ارایه می دهد.

در کل استفاده از **open MP** قطعا بسیار سریع تر و راحت تر از موار دیگر است. بدون شک