تمرین درس پردازش موازی سری سوم

نام: سینا قادرمرزی شماره‌ی دانشجویی: ۹۳۱۳۱۰۳۸

در این تمرین موازی‌سازی الگوریتم غربال اراتستن با استفاده از MPI انجام گرفته است

روش استفاده شده

همان طور که می‌دانیم، در الگوریتم غربال اراتستن همه‌ی اعداد اول که از یک مقدار داده شده‌ی N کوچکتر هستند محاسبه می‌گردند. این الگوریتم شامل لیستی از اعداد ۲ تا N است که در یک فرایند ترتیبی اعدادی که مرکب هستند از لیست خط می‌خورند و اعداد باقی‌مانده به عنوان اعداد اول مشخص می‌گردند.

2, 3, 4, 5, 6, 7, 8, 9, 10, …, N

در ابتدا اولین عدد خط نخورده ۲ است : در پیمایش اول لیست همه‌ی اعدادی که به ۲ بخش‌پذیر اند خط می‌خورند.

2, 3, 4, 5, 6, 7, 8, 9, 10, …, N

عدد خط نخورده‌ی بعدی ۳ است : در پیمایش دوم لیست همه‌ی اعدادی که به ۳ بخش‌پذیر اند خط می‌خورند.

2, 3, 4, 5, 6, 7, 8, 9, 10, …, N

عدد خط نخورده‌ی بعدی ۵ است : در پیمایش سوم لیست همه‌ی اعدادی که بر ۵ بخش‌پذیر اند خط می‌خورند.

2, 3, 4, 5, 6, 7, 8, 9, 10, …, N

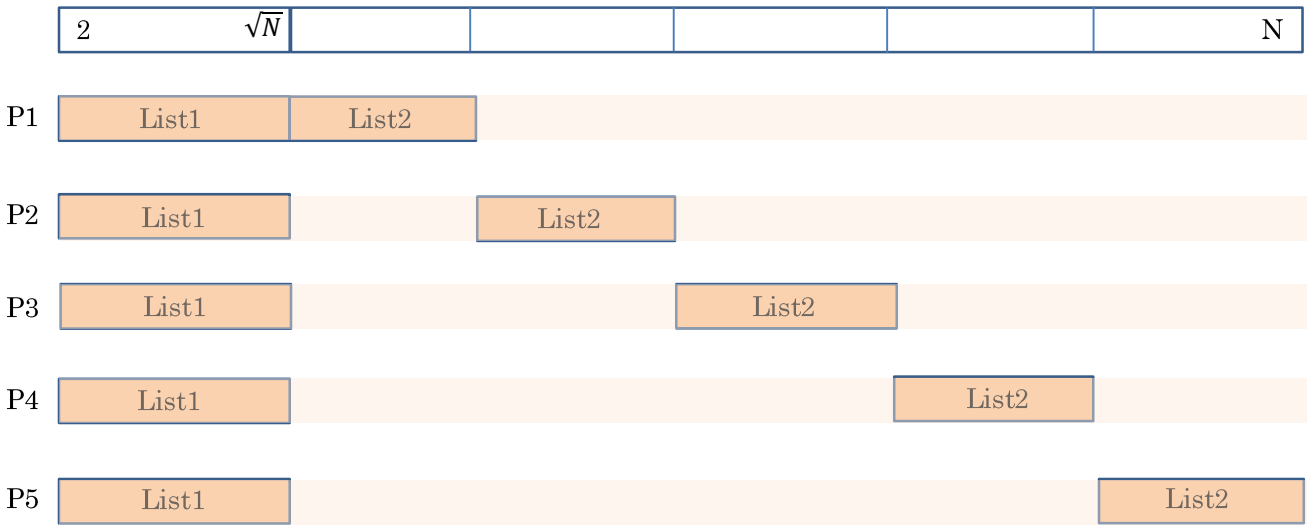
... به همین شکل در هر مرحله عدد خط نخورده‌ی بعدی مشخص می‌شود و لیست برای خط زدن اعدادی که بر آن بخش‌پذیر اند پیمایش می‌شود. با توجه به این که همه‌ی اعداد مرکب کوچکتر از N حداقل توسط یک عدد اول کوچکتر از $\sqrt{N}$ خط می‌خورند فرآیند یافتن اعداد اول از سر لیست حداکثر تا $\sqrt{N}$ لازم است ادامه پیدا کند و پس از آن می‌دانیم که همه‌ی اعداد مرکب خط خورده‌اند.

ایده‌ی کلی موازی سازی به این شکل است که با داشتن اعداد اول متناظر با هر پیمایش، قطعه‌های مختلف لیست می‌تواند به صورت مستقل و موازی پیمایش شوند. اما تعیین اعداد اول متناظر با هر پیمایش وابسته به پیمایش‌های قبلی است. می‌دانیم که جستجو برای این اعداد حداکثر تا $\sqrt{N}$ ادامه پیدا می‌کند بنابراین پس از

آن که این جستجو تا $\sqrt{N}$ انجام گرفت پیمایش مابقی لیست می‌تواند به قسمت‌هایی تقسیم شود و به صورت موازی توسط فرایندهای مختلف انجام گیرد.

برای این که این ایده‌ی موازی سازی را با استفاده از MPI پیاده نماییم می‌توان به این صورت عمل کرد که یکی از فرایندها اعداد اول از۲ تا $\sqrt{N}$ را محاسبه و برای فرایندهای دیگر ارسال نماید. اما راه‌حل بهتر این است که هرکدام از فرایندها مستقلاً به محاسبه‌ی اعداد اول از ۲ تا $\sqrt{N}$ بپردازند و در پی آن قطعه لیست مربوط به خود را نیز پردازش نمایند. و سپس همه‌ی فرایندها قطعه لیست پردازش شده را برای فرایند اصلی ارسال نمایند.

برای این منظور آرایه‌ای از اعداد به طول N-1 در نظر گرفته می‌شود که مقدار داخل آن، وضعیت عدد متناظر با اندیس آن خانه را نشان می‌دهد. که در ابتدا همه‌ی خانه‌های آن صفر است و پس از پردازش، اندیس‌های متناظر با اعداد مرکب مقدار ۱ به خود می‌گیرند. این لیست به دو قسمت اصلی تقسیم می‌شود. قسمت اول متناظر با اعداد از ۲ تا $\sqrt{N}$ و قسمت دوم متناظر با اعداد از $\sqrt{N}$ تا N است. قسمت اول توسط همه‌ی فرایندها محاسبه می‌شود و قسمت دوم به P قسمت مساوی (P تعداد فرایندها) تقسیم شده و هر یک از فرایندها یک قطعه از آن را محاسبه می‌نمایند. در برنامه، قسمت اول توسط آرایه‌ی list1 و هریک از قطعه‌های قسمت دوم توسط آرایه‌ی list2 نشان داده شده اند. کاری که هر فرایند انجام می‌دهد این است که پس از انجام عملیات مربوط به list1 (این کار توسط همه‌ی فرایندها انجام می‌گیرد)، عملیات مربوط به list2 (که خاص آن فرایند است را انجام می‌دهد) و به این شکل قسمتی از آرایه‌ی اصلی را که مربوط به آن فرایند است مقداردهی می‌نماید. در نهایت فرایند اصلی (Rank =0) قطعه‌های پردازش شده را دریافت و بر اساس آن اعداد اول در بازه‌ی مربوطه را چاپ می‌کند.



این برنامه بر روی یک Virtual Machine با سیستم عامل Ubuntu 15.04 و با اختصاص ۱ پردازنده‌ی چهار هسته‌ای و 2GB حافظه اجراشده است. برای کامپایل و اجرای این برنامه از بسته‌ی MPICH استفاده شده است.

برای کامپایل این برنامه در ترمینال Ubuntu دستور زیر را وارد می‌کنیم.

```
mpicc parallel_primes.c –o parallel_primes
```

دستور زیر را به منظور اجرای برنامه‌ی ایجاد شده با سایز وروی N وارد می‌کنیم

```
mpiexec –np 4 ./parallel_primes N 2
```

در اینجا ورودی 2 برای رفتن به حالت اندازه‌گیری speedup در نظر گرفته شده. برای مشاهده ی نتایج می‌توان به جای 2 عدد 1 را قرارداد.

با وارد کردن این دستور برنامه به صورت محلی بر روی ۴ فرایند اجرا می‌گردد.

تسریع‌های به دست آمده نسبت به حالت سریال، به ازای سایزهای مختلف ورودی و استفاده از چهار فرایند در جدول زیر آمده است.

N	1000	10000	100000	1000000	10000000
speedup	0.487076	2.529706	2.380798	3.590501	3.890513

برای این که فرایندها را بر روی Remote Host اجرا کنیم. مراحل زیر انجام شده است.

۱- ایجاد ۳ عدد Virtual machine بر روی VMWare با حافظه‌ی 256MB و نصب Ubuntu server بر روی آن‌ها

۲- اتصال این Virtual Machine ها از طریق برقراری یک شبکه‌ی مجازی

۳- نصب ssh و MPICH بر روی هر کدام از ماشین‌ها و برقرای ارتباط ssh بین آن‌ها

۴- تولید یک کلید بر روی ماشین اصلی و نصب این کلید بر روی ماشین‌های دیگر (به منظور ایجاد امکان اتصال ssh از ماشین اصلی به ماشین‌های دیگر بدون نیاز به پسورد)

۵- ایجاد یک host file بر روی ماشین اصلی شامل آدرس ماشین‌های دیگر

۶- فراخوانی mpirun با host file ساخته شده