پنجمین کنفرانس اطلاعات و دانش
دانشگاه شیراز، ۱ الی ۳ خردادماه ۱۳۹۲

ارائه روش موازی از غربال چرخ برای تولید اعداد اول

پیمان نویدی[۱]، غلامحسین دستغیبی فرد[۲]، احسان بزرگپوری[۳]

دانشکده مهندسی برق و کامپیوتر دانشگاه شیراز، *paymanild@gmail.com[1] , dstghaib@shirazu.ac.ir[2] , ehsan108@gmail.com[3]*

تولید اعداد اول با روش موازی در زمان کمتر ، جهت بگاری در امنیت انتقال داده های کامپیوتری

چکیده

این مقاله الگوریتم موازی برای تولید اعداد اول با استفاده از غربال چرخ تا یک حد مشخص را معرفی می نماید. غربال چرخ یک روش گرافیکی از غربال ارتستون است که اعداد شبه اول را از اعداد مرکب جدا می کند. امروزه با پیشرفت محاسبات موازی، امکان پردازش دستورالعمل ها بصورت همزمان توسط چندین کامپیوتر، برای کاهش زمان پردازش میسر است. ما در این مقاله الگوریتم موازی را مطرح می کنیم که با پیچیدگی زمانی بسیار پایینی اعداد اول از یک تا n را بدست می آورد. نتایج حاصل از تحقیقات ما فرمولی را بدست می آورد که احتمال تشخیص اول یا مرکب بودن یک عدد را مشخص می کند. امروزه اعداد اول نقش مهمی در رمزنگاری داشته و هنوز هم موضوعی داغ و موردعلاقه محققان است.

مقدمه

ما در این مقاله می خواهیم راجع به تولید اعداد اول صحبت کنیم. اعداد اول برای مهندسان وطراحان نرم افزارهای مهندسی بسیار مهم و حیاتی است، چرا که یکی از کاربردهای اصلی اعداد اول در مسائل امنیتی و ایمنی ارتباطات رایانه ای به ویژه شبکه های مبادلاتی الکترونیک است.

ما می خواهیم الگوریتم موازی مطح کنیم که اعداد اول بازه یک تا n را با استفاده از فاکتور چرخ بدست می آورد و این محدودیت الگوریتم های ترتیبی را به حداقل می رساند و یا به عبارتی دیگر زمان اجرا را کاهش می دهد. فاکتور چرخ کمک می کند اعداد را در چرخ های پی در پی قرار دهیم و بر اساس الگویی که به چرخ ها می دهیم اعداد مرکب را غربال کرده و اعداد شبه اول را تولید کنیم. سپس در یک فیلتر دیگر اعداد شبه اول را تمیز کرده و باقیمانده اعداد، نتیجه نهایی اعداد اول تولید شده در بازه یک تا n است.

ما در این روش کار را با چند پردازنده شروع می کنیم. ابتدا پردازنده اول الگویی بر اساس مدل چرخی که می خواهیم تولید کنیم، ایجاد می کند. سپس الگو را برای همه Broadcast می کند. پردازنده های دیگر براساس الگوی دریافت شده و آدرس شروع و پایان در آرایه اصلی (که با استفاده از شماره rank هر پردازنده بدست می آید)، اعداد چرخ مربوط به خود را تولید و غربال را تبطق فرمول تعیین اعداد شبه اول انجام می دهند. در نهایت هر چرخ غربال، دارای تعدادی اعداد شبه اول است که حداکثر یک سوم طول چرخ است.

الگوریتم موازی غربال

در این الگوریتم ابتدا الگویی از نوع حلقه ای که قرار است هر پردازنده آن را محاسبه کند توسط پردازنده 0 ساخته می شود. سپس الگو برای تمام پردازنده های دیگر Broadcast می شود. پردازنده ها با توجه به الگو، شماره rank و شماره iteration اندیس ابتدا و انتهای آرایه چرخ را محاسبه می کنند. سپس اعدادی را که در فرمول زیر صدق کند، غربال کرده و در نهایت یک سری اعداد شبه اول بدست می آید.

pattern_array[k]+ring_mode(rank+i*proc)

چون ممکن است تعدادی عدد مرکب در اعداد شبه اول باشد، یک تابع که اول بودن عدد را بررسی می کند، بر روی اعداد شبه اول اعمال شده و در پایان اعداد اول نهایی تولید می شود.

```
for(i=0;i<iteration;i++){
 start=ring_mode*(rank+i*procs)+1;
 end=ring_mode*(rank+i*procs+1);
 k=0;
 for(j=start;j<=end && k<=pc;j=pattern_array[k]
       +ring_mode(rank+i*proces)){
  if(isprime(j)==”true”)
    f++;
  k++; }}
```

نتایج و آزمایشات

ما برای اینکه به نتایج محسوسی برسیم، الگوریتم را بر روی سیستمی با مشخصات پایین تست کردیم. هدفمان رسیدن به کمترین زمان ممکن با کمترین سیستم موجود است. در جدول زیر زمان بدست آمده از اعداد یک میلیون تا دو میلیارد نشان داده شده است.

Max number	Second	Ring mode
1.000.000	0.009	2310
10.000.000	0.068	2310
50.000.000	0.189	30030
100.000.000	0.637	30030
500.000.000	1.515	30030
1.000.000.000	3.344	30030
1.500.000.000	5.197	510510
2.148.226.080	**7.188**	**510510**

نتیجه گیری

این الگوریتم موازی کمک می کند تا اعداد اول بازه ۱ تا n را با سرعت بسیار بالایی تولید کنیم. الگوریتم برای داده های بزرگ بخوبی کار می کند و با افزایش تعداد پردازنده ها متناسب با اعداد ورودی و نوع حلقه، نتایج بسیار خوبی را بدست می آورد. این الگوریتم تنها الگوریتمی است که از طریق ارسال الگو و تقسیم کار برای هر پردازنده کار می کند . iteration ها کمک می کند بیکاری پردازنده ها به حداقل برسد. الگوریتم مشابهی در [۷] انجام شده که با استفاده از fork و بصورت درختی پیاده سازی شده است.

امید است در آینده بتوانیم با انجام آزمایشات بیشتر برای اعداد بزرگتر، به نتایج بهتری برسیم. همچنین با دسترسی به متغیر هایی از حافظه، که امکان ذخیره اعداد بزرگتری را در ختیارمان قرار دهد، صورت مسئله را بزرگتر کنیم و الگوریتم را با زمان کمتری برای اعداد بالاتر از دو میلیارد آزمایش کنیم.

منابع

[1] S.H. Bokhari. Multiprocessing the sieve of eratosthenes. IEEE Computer, 20(4):50–58, 1987.
[2] G. Paillard and C. Lavault. Le crible de la roue en distribu ´e. In MAJECSTIC 2003 (MAnifestation des Jeunes Chercheurs en STIC), Marseille, October 2003.
[3] C. Lavault. ´Evaluation des algorithmes distribu´es analyse, complexit´e, m´ethodes. ´Editions Herm`es, Paris, 1995.
[4] H.G. Mairson. Some new upper bounds on the generation of prime numbers. Communications of the ACM, 20(9):664–669, 1977.
[5] J. Sorenson. An introduction to prime numbers sieves. Technical Report 909, University ofWisconsin, Computer Sciences Department, January 1990.
[6] G. Paillard, "A Fully Distributed Prime Number Generation Using The Wheel Sieve", Universit´e Paris XIII, UMR CNRS 7030.
[7] P. Pritchard. Explaining the wheel sieve. Acta Informatica, 17:477–485, 1982.
[8] G. Paillard, C. Lavault, and F. Franc¸a. A smerbased distributed prime sieving algorithm. Technical Report 2004-04, Universit´e Paris Nord, Laboratoire d'Informatique de Paris Nord, July 2004.