Algoritmos numéricos empregando uma técnica de escalonamento para multigrafos

Gabriel A. L. Paillard[2], Felipe M. G. França [1], Christian Lavault [3]

[1]Universidade Federal do Ceará (UFC)
Instituto Universidade Vitual

[2]Universidade Federal do Rio de Janeiro (UFRJ)
Programa de Engenharia de Sistemas e Computação (PESC/COPPE)

[3]Université Paris Nord
Laboratoire d'Informatique de Paris Nord

gabriel@virtual.ufc.br,felipe@cos.ufrj.br,lavault@lipn.univ-paris13.fr

Resumo. *Este artigo apresenta o histórico da aplicação de uma técnica baseada em multigrafos para gerar todos os números primos em um dado intervalo de inteiros. De Eratóstenes, que elaborou o primeiro crivo (há mais de 2000 anos), para a atual geração de computadores paralelos, que permitiram atingir limites maiores no intervalo ou obter resultados anteriores em menor tempo, a geração de números primos ainda representa um domínio atraente de pesquisa e desempenha um papel central na criptografia. Neste trabalho mostramos o emprego do escalonamento por múltiplas inversões de arestas em três algoritmos totalmente distribuídos, empregados no crivo de um intervalo* $[2, n]$*, para encontrar todos os números primos.*

1. Introdução

Este trabalho se atenta à geração de números primos menores que um determinado limite n, usando o crivo da roda (*wheel sieve*) de forma distribuída [Pritchard 1982]. Os algoritmos baseados no crivo da roda podem ser muito eficientes para determinar a primalidade de inteiros que pertencem a um determinado intervalo finito $[2, n]$, para valores suficientemente grandes de n e quando o teste de primalidade é realizado em todos os números do intervalo, mas requer técnicas de pré-processamento [Mairson 1977]. Este artigo trata de três algoritmos de crivo totalmente distribuídos usando escalonamento por inversão de múltiplas arestas (SMER) [Barbosa et al. 2001], os quais possuem complexidade computacional $O(n + \sqrt(n))$.

O objetivo principal em paralelizar esse tipo de algoritmo é incrementar o alcance nos limites da geração de números primos e/ou alcançar esses limites em um tempo de execução mais curto. A primeira paralelização de um algoritmo de peneira foi realizada em 1987 [Bokhari 1987], que paralelizou o crivo de Eratóstenes. Este trabalho foi motivado pelo teste de uma nova máquina paralela (*Flex / 32*), já que este tipo de algoritmo é ideal para testar performances de uma nova arquitetura, de uma máquina sequencial ou paralela, como referência (*benchmarking*).

O presente artigo discute a aplicação da técnica SMER para encontrar todos os primos por crivo (de forma distribuída) em um dado intervalo $[1, n]$ [1], usando as

[1]o número 1 precisa ser incluído no intervalo para a aplicação do algoritmo do crivo de roda.

propriedades do escalonamento por múltiplas inversoes de arestas [Barbosa et al. 2001, Barbosa and Gafni 1989].

Alguns outros algoritmos distribuídos que geram todos os números primos podem ser encontrados em [Cosnard and Philippe 1989], os quais empregam as propriedades do teorema de Dirichlet.

2. Referencial teórico e Algoritmos

O crivo da roda, derivado do algoritmo de Pritchard [Pritchard 1981], opera basicamente gerando um conjunto de números que não são múltiplos dos primeiros números primos de k. O crivo, aplicado no conjunto resultante da roda, elimina os números não primos que permanecem no conjunto. Esta é a idéia básica da roda (*wheel*) que foi empregada como uma classe reduzida de resíduos $\mathrm{mod}(\Pi_k)$, onde Π_k denota o produto dos primeiros k números primos [Pritchard 1982]. $\mathcal{W}_k$ denota a roda k, que é definida como:

$$\mathcal{W}_k = \{x \mid 1 \leq x \leq \Pi_k \text{ e } (x, \Pi_k) = 1\} \tag{1}$$

onde (x, Π_k) é o maior divisor comum dos inteiros x e Π_k.

Nesta perspectiva propusemos um algoritmo distribuído baseado numa técnica de escalonamento por múltiplas inversões de arestas o qual foi publicado em [Paillard 2005a]. O algoritmo de Escalonamento por Inversão de Arestas (SER), proposto por [Barbosa and Gafni 1989], consiste em um mecanismo de controle de concorrência de acesso a recursos compartilhados em sistemas distribuídos. A solução de [Barbosa and Gafni 1989], propõe modelar os processos que compartilham recursos como os nós de um grafo direcionado G. Os processos que compartilham recursos possuem uma aresta entre si. O algoritmo pode ser descrito como:

> Dado um grafo não orientado G, inicialmente oriente G por uma orientação acíclica – que pode ser obtida facilmente se os nós possuem identificadores únicos, basta que, em cada nó, as arestas apontem para os nós que possuem identificadores de valor maior que seu próprio identificador - nós que são sorvedouros nesta primeira orientação tem o direito de operar inicialmente. Após a execução, apenas os nós que eram sorvedouros invertem suas arestas, assim, os nós que passarem a ser sorvedouros nesta nova orientação – que também é acíclica, ganharão o direito de operar. As inversões sucedem-se de maneira que todos os nós terão sua chance de obter o recurso compartilhado. [Barbosa and Gafni 1989].

Como um exemplo da aplicabilidade do SER, considere o problema dos filósofos de jantar de Dijkstra sob alta carga, isto é, no caso os filósofos estão ou ”famintos” ou ”comendo” (nenhum estado ”pensando”). Tal sistema pode ser representado por um conjunto $\{P_1, \ldots, P_N\}$ de N processos, onde cada processo compartilha um recurso tanto com seu processo anterior quanto com seu processo subsequente. Assim, tomando a configuração original onde $N = 5$ e definindo uma orientação acíclica sobre o anel de 5 nós, a dinâmica SER resultante onde $P = 5$ e $M = 2^2$ é ilustrada na Fig. 1.

[2]M representa o número de vezes que cada processo acessára os recursos compartilhados.

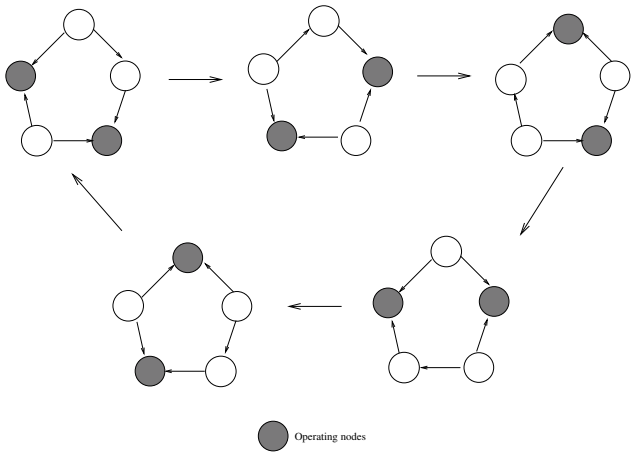

Figure 1. Dinâmica do SER aplicada ao Jantar dos Filósofos sob alta carga.

SMER (Escalonamento por Múltiplas Inversões de Arestas) é uma generalização do SER (Escalonamento por Inversões de Arestas) onde as taxas de acesso aos recursos atômicos são pré-especificadas junto aos processos. Num compartilhamento distribuído de recursos o sistema distribuído é representado por um multigrafo $\mathcal{M} = (V, \mathcal{E})$. Em contraste com o SER, múltiplas arestas podem existir entre quaisquer dois nós i e j (i, $j \in V$). Na dinâmica do SMER podemos ter $e_{i,j} \geq 0$ arestas conectando nós i e j; tais nós conectados são chamados de vizinhos e $\mathcal{E}$ representa o conjunto de todas as arestas direcionadas e$_{i,j}$.

Em [Paillard et al. 2005] foi proposto uma versão distribuída do crivo da roda, onde um processo mestre coordenava os processos remanescentes durante a geração de números primos. O mesmo foi implementado empregando uma biblioteca de interface de passagem de mensagens (*lam-mpi 7.0.6 library*) [Burns et al. 1994] e as medições de tempo da implementação sequencial e distribuída do crivo da roda foram comparadas, junto com uma implementação seqüencial e distribuída da crivo de Eratóstenes. Em [Paillard 2005b] uma versão totalmente distribuída de geração de números primos foi introduzida, mas sem empregar o escalonamento por múltiplas inversões de arestas. Finalmente em [Paillard 2005a] apresentamos um novo crivo utilizando o escalonamento por múltiplas inversões de arestas, de forma distribuída, mas sem ainda ter conseguido aplicá-la ao algoritmo do crivo da roda; no [Paillard et al.] foi apresentado um novo algoritmo distribuído empregando o crivo da roda como técnica de base para gerar todos os números primos de um determinado intervalo $1..n$.

3. Pespectivas de pesquisa

O próximo passo fundamental para este trabalho de pesquisa consiste em implementar os seguintes trabalhos: [Paillard et al.], [Paillard 2005a] e [Paillard 2005b] e comparar os resultados com as versões sequenciais [Pritchard 1982] para analisar os eventuais ganhos obtidos, visto que nos trabalhos propostos não existe tempo de pré-processamento. Também os tempos de execução das três implementações distribuídas serão comparados entre si.

4. Conclusões

O tema apresentado continua tendo uma relevância fundamental na área de computação numérica onde qualquer avanço pode representar melhorias significativas no desempenho dos algoritmos de base empregados em áreas diversas como segurança de informações.

References

Barbosa, V., Benevides, M., and França, F. (2001). Sharing resources at nonuniform access rates. *Theory of Computing Systems*, 34(1):13–26.

Barbosa, V. and Gafni, E. (1989). Concurrency in heavily loaded neighborhood-constrained systems. *ACM Transactions on Programming Languages and Systems*, 11(4):562–584.

Bokhari, S. (1987). Multiprocessing the sieve of eratosthenes. *IEEE Computer*, 20(4):50–58.

Burns, G., Daoud, R., and Vaigl, J. (1994). LAM: An open cluster environment for MPI. In *Proceedings of Supercomputing Symposium*, pages 379–386.

Cosnard, M. and Philippe, J.-L. (1989). Génération de nombres premiers en parallèle. *La lettre du transputer*, pages 3–12.

Mairson, H. (1977). Some new upper bounds on the generation of prime numbers. *Communications of the ACM*, 20(9):664–669.

Paillard, G. (2005a). A distributed prime sieving algorithm based on scheduling by multiple edge reversal. In *ISPDC 2005 (International Symposium on Parallel and Distributed Computing)*, pages 139–146, Lille, France.

Paillard, G. (2005b). A fully distributed prime numbers generation using the wheel sieve. In *PDCN 2005 (Parallel and Distributed Computing and networks)*, pages 651–656, Innsbruck, Autriche.

Paillard, G., Lavault, C., and França, F. (2005). A distributed prime sieving algorithm based on scheduling by multiple edge reversal. In *The 4th International Symposium on Parallel and Distributed Computing*.

Paillard, G. A. L., França, F. M. G., and Lavaut, C. A distributed wheel sieve algorithm. In *33rd IEEE International Parallel and Distributed Processing Symposium, 21st Workshop on Advances in Parallel and Distributed Computational Models*.

Pritchard, P. (1981). A sublinear additive sieve for finding prime numbers. *Communications of the ACM*, 24(1):18–23.

Pritchard, P. (1982). Explaining the wheel sieve. *Acta Informatica*, 17:477–485.