Seqüenciamento de DNA em Arquiteturas com Memória Distribuída

Daniela Saccol Peranconi, Gerson Geraldo H. Cavalheiro

Mestrado em Computação Aplicada - Universidade do Vale do Rio dos Sinos
Av. Unisinos, 950 – 93.022-000 – São Leopoldo, RS - Brasil
{danielap, gersonc}@exatas.unisinos.br

Palavras-chave

Seqüenciamento, DNA, aglomerados, memória compartilhada distribuída.

Seqüenciamento de DNA

A razão com a qual seqüências têm sido adicionadas às bases de dados públicas se dá de forma exponencial, aumentando rapidamente o tamanho destas bases. A busca nestas bases de dados por seqüências similares a uma seqüência dada é a mais importante operação primitiva na Biologia Computacional, servindo como base para operações mais complexas [SET 97]. O algoritmo de Smith-Waterman [SMI 81] é considerado o mais sensitivo na busca por seqüências semelhantes, apresentando os melhores resultados. No entanto, o longo tempo de computação exigido por este algoritmo limita sua utilização. Como alternativa para aumentar a velocidade, algumas heurísticas têm sido desenvolvidas, tais como FASTA [PEA 88] e BLAST [ALT 90]. Estes programas, porém, sacrificam a qualidade dos resultados.

Devido à demanda tanto por buscas mais rápidas quanto sensitivas, muito esforço tem sido aplicado no desenvolvimento de implementações eficientes do algoritmo de Smith-Waterman. Diversas soluções vêm sendo propostas, tanto em hardware (Paracel's GeneMatcher, Compugen's Bioaccelerator, TimeLogic's Decypher) quanto em software [GAL 90, ROG 01, MAR 01].

Objetivo do Trabalho

O desenvolvimento do presente trabalho objetiva a implementação de um mecanismo de suporte à comunicação em aglomerados, oferecendo uma visão de memória compartilhada entre os nós. A validação de tal mecanismo será obtida através da implementação do algoritmo de Smith-Waterman.

A implementação concorrente do algoritmo prevê a divisão do trabalho da aplicação em atividades concorrentes, denominadas tarefas. Desse modo, é preciso introduzir mecanismos de comunicação de dados e de sincronização de tarefas para permitir o controle da evolução do programa. Os mecanismos de comunicação permitem que dados produzidos por uma tarefa sejam colocados à disposição de uma outra tarefa. Os mecanismos de sincronização permitem a uma tarefa informar a outra que um dado encontra-se disponível ou verificar a disponibilidade de um determinado

dado. Com os mecanismos de sincronização é possível garantir que tarefas não sejam executadas antes que seus dados de entrada estejam disponíveis.

Portanto, a função da sincronização é de conciliar as datas de execução das tarefas em relação à produção/consumo de dados. O programa em execução consistirá, então, de um conjunto de tarefas, onde cada tarefa delimita uma seqüência de instruções elementares e define dois conjuntos de dados: os dados necessários para iniciar sua execução e os dados produzidos como resultado de sua execução, tal um grafo de fluxo de dados. A ordem com que as tarefas serão executadas é definida de acordo com a disponibilidade de seus dados de entrada. Ao seu término, uma tarefa produzirá um resultado que, eventualmente, poderá viabilizar a execução de outra tarefa.

Neste modelo de programação concorrente é realizada a implementação do algoritmo de Smith-Waterman. Este algoritmo manipula uma matriz M[n, m] de dados, inicialmente vazia. O cálculo inicia na posição M[0,0] desta matriz, evoluindo num processo de inundação até calcular a posição M[n-1, m-1]. Um ponto M[i,j], para ser calculado na matriz, necessita dos pontos M[i-1, j], M[i-1, j-1] e M[i, j-1]. O algoritmo concorrente define tarefas como sendo o cálculo de um bloco de pontos (submatriz de M). Portanto, o cálculo do bloco [k, l] necessita dos blocos [k-1, l], [k-1, l-1] e [k, l-1]. A sincronização está definida na dependência de dados entre as tarefas.

Por ser um algoritmo com ótimos resultados, mas com grande exigência computacional e longo tempo de computação, propõe-se um mecanismo de comunicação para ser utilizado em aglomerados, que seja o mais eficiente possível, de forma a reduzir ao máximo o overhead de comunicação entre as tarefas. Com este mecanismo e com a utilização de uma arquitetura paralela para execução do algoritmo de Smith-Waterman, acredita-se que os resultados encontrados serão satisfatórios tanto em questão de tempo quanto de sensitividade.

Referências

[ALT 90] ALTSCHUL, S. F. et al. Basic local alignment search tool. **J. Mol. Biol.**, 215:403--410, Oct. 1990.

[GAL 90] GALPER, A. R. et al. Parallel Similarity Search and Alignment with the Dynamic Programming Method. **KSL Report 90-74**, Stanford University, Apr. 1990.

[MAR 01] MARTINS, W. S. et al. A Multithreaded Parallel Implementation of a Dynamic Programming Algorithm For Sequence Comparison. **Pacific Symposium of Biocomputing**, 6:311--322, 2001.

[PEA 88] PEARSON, W. R. et al. Improved tools for biological sequence comparison. **Proc. Natl Acad. Sci. USA**, v. 85, p. 2444--2448, Apr. 1988.

[ROG 01] ROGNES, T. ParAlign: a parallel sequence alignment algorithm for rapid and sensitive database searches. **Nucleic Acids Res.**, v. 29, n. 7, p. 1647--1652, Feb. 2001.

[SET 97] SETUBAL, J. and MEIDANIS, J. **Introduction to Computational Molecular Biology**. PWS Publishing Company, CA, 1997.

[SMI 81] SMITH, T. F. et al. Identification of common molecular subsequences. **J. Mol. Biol.**, 147, p. 195--197, Mar. 1981.