FERRAMENTA BIOPROVIDER VOLTADA PARA GRANDES VOLUMES DE DADOS

Aluno: Gabriel Luis Caldeira Vieira
Orientador: Sérgio Lifschitz

Introdução

A comparação de biossequências é uma das tarefas mais importantes atualmente na área de Bioinformática, entretanto, ainda não existe um Sistema Gerenciador de Banco de Dados (SGBD) específico para essa aplicação, restando apenas o gerenciamento de buffer padrão fornecido pelo sistema operacional. As ferramentas da família BLAST são as mais utilizadas hoje em dia, e o seu desempenho é um fator muito importante, sendo que pequenas melhoras nestas podem trazer grandes benefícios.

Com esse propósito foi criado o BioProvider, uma ferramenta provedora de dados que realiza um gerenciamento de buffer eficiente para o BLAST. Essa ferramenta é utilizada de forma transparente ao BLAST, já que a comunicação entre ambos é feita através de um driver que substitui as chamadas de funções de leitura e escrita no arquivo do banco de dados.

Contudo, na época em que a ferramenta BioProvider foi desenvolvida, as bases de dados disponíveis dificilmente ultrapassavam 1GB, como descrito em Noronha, M. (2005). Portanto, ela não foi projetada para funcionar com uma base de dados de tamanho muito elevado e não se imaginava que isso poderia gerar problemas.

A ferramenta BLAST não trabalha bem com bases grandes, por isso, numa etapa de pré-processamento, um programa chamado *formatdb* é chamado para quebrar em pedaços menores os arquivos de índice e de anotações de tal forma que os arquivos sejam mais fáceis de manusear — o que impede o funcionamento do BioProvider, já que este foi desenvolvido sem levar em conta essa funcionalidade da ferramenta BLAST. Portanto, é necessário realizar modificações no código fonte do BioProvider a fim de adaptá-lo para esse caso em particular, que consiste em uma tarefa certamente trabalhosa, visto que são necessários conhecimentos bem aprofundados na linguagem de programação C, sistema operacional Linux, drivers de dispositivos e o algoritmo BLAST, assim como é imprescindível estudar a fundo o gerenciamento de buffer do BioProvider e suas demais funcionalidades.

Objetivos

Modificar o código atual do BioProvider a fim de que este possa tratar eficientemente as múltiplas divisões de uma base de dados volumosa (com tamanho em disco superior a 1GB), sem que ele deixe de funcionar de maneira transparente a ferramenta BLAST.

Para simplificar os testes realizados durante o desenvolvimento, a fim de que estes levem menos tempo para executar, deve-se limitar a memória RAM manualmente em 256MB ou 512MB. Como o desempenho do BioProvider depende tanto do tamanho do arquivo de entrada como da memória RAM disponível, é interessante também fazer as simulações variando esses dois parâmetros para analisar o seu efeito sobre o desempenho do programa.

Metodologia

Para tratar a possível quebra da base em "volumes" (nome atribuído às partições pela NCBI como pode ser lido no seu próprio site) foram consideradas várias soluções alternativas. Uma primeira opção seria fazer uma pequena alteração no código do programa *formatdb*, de tal forma que ele não quebrasse mais a base, mas isso acarretaria em perda de eficiência assim como uma alteração no código do BLAST (que não conseguiria abrir um arquivo muito grande). Logo essa solução não seria adequada visto que o BioProvider foi desenvolvido visando ser transparente ao BLAST.

A solução ideal encontrada é alterar a maneira como o BioProvider funciona, isto é, modificar o seu código fonte, adicionando um tratamento especial para o caso das bases grandes.

Para isso foi necessário estudá-lo a fundo, afim de identificar os pontos onde seriam necessárias alterações. Uma forma de fazer isto é tomando um exemplo e analisando os casos que resultariam em erro decorrentes da lógica de programação incorreta.

Então, seja uma base dividida em 4 blocos pelo BioProvider. Isso significa que esta base possuí 4 possíveis pontos de entradas onde os processos BLAST podem começar a ler do arquivo de sequências. As permutações criadas são simplesmente os arquivos PHR e PIN referentes a cada um desses pontos, já que um processo BLAST que começou em um certo ponto precisa ver o arquivo de índice e de anotações modificado em relação a este. Ou seja, quando um processo BLAST precisa ver a anotação de uma sequência, o BioProvider pergunta em qual ponto ele começou sua leitura e em seguida passa o par correto de arquivos índice e anotação.

Continuando o exemplo anterior, seja uma base dividida em 4 blocos mas grande o suficiente para ser quebrada em 3 volumes pelo *formatdb*. A étapa de quebra em volumes ocorre primeiro, ou seja, para cada um existiriam 4 permutações dos arquivos de índice e anotações. Embora não sejam criadas permutações do arquivo de sequências, ele também é quebrado em volumes.

O primeiro problema aparente é quando o buffer do anel chega no final de um volume do arquivo de sequências. O segundo ocorre quando um processo BLAST pedir um arquivo de índices ou de anotações. O BioProvider saberá apenas a permutação correta desse arquivo, mas não o volume.

No caso do primeiro problema, o buffer do anel deverá ser reposicionado para o início do próximo volume ou para o início do primeiro volume (isto é, se já tiver percorrido todos), de tal maneira que ele continue percorrendo o arquivo de sequências "ciclicamente". O BioProvider deverá ter controle do volume "corrente" que está sendo percorrido pelo buffer.

No segundo caso, dependendo da implementação do BioProvider, poderia ser que não fosse necessária nenhuma alteração no código fonte, caso o BioProvider devolvesse o nome dos arquivos pedidos apenas concatenando o sufixo da permutação correta (por exemplo, se o BLAST pedir o arquivo "nr.00.phr" e o BioProvider identificar a permutação desse bloco como sendo aquela de índice 2, basta que ele apenas concatene "_2" antes do ".phr" para devolver o arquivo correto "nr.00_2.phr"). Analisando o código fonte, pode-se concluir que ele não o faz dessa maneira - e isto é justamente o que deve ser alterado.

Conclusão

De fato, a ferramenta funciona como o esperado para tamanhos inferiores a esse limite, mas atualmente os pesquisadores dificilmente trabalham com bases que não o ultrapassem. Logo, antes que a ferramenta seja publicada como projeto de código aberto (na verdade, já existe uma versão estável para bases pequenas em um projeto no site SourceForge), seria interessante finalizar as alterações no código que permitirão ao BioProvider tratar as múltiplas partições (ou "volumes") de uma base de dados de tamanho superior a 1GB.

Concluí-se que a realização desse projeto consiste em uma tarefa bastante complexa, visto que são necessários conhecimentos bem aprofundados na linguagem de programação C, sistema operacional Linux, drivers e do algoritmo BLAST, além de um estudo a fundo da ferramenta BioProvider. Contudo, essa complexidade é justificável, pois trata-se de um projeto inovador que pode trazer muitos ganhos para a comunidade científica e para os laborátorios farmacêuticos, considerando a utilização frequente das ferramentas da família BLAST na área da Bioinformática.

Referências

1 - NORONHA, M.; **Controle da Execução e Disponibilização de Dados para Aplicativos sobre Sequências Biológicas: o Caso BLAST**. Rio de Janeiro, 2006. 83p. Dissertação de Mestrado, Departamento de Informática, PUC-Rio.

2 - CORBET, J.; RUBINI, A.; KROAH-HARTMAN, G. **Linux Device Drivers.** O'Reilly, 2005.

3 - http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/formatdb_fastacmd.html