AVALIAÇÃO PRÁTICA DA FERRAMENTA BIOPROVIDER

Aluno: Waldecir Faria
Orientador: Sérgio Lifschitz

Introdução

Em 2006 foi feito um estudo [1] voltado para a área de gerenciamento de banco de dados para bioinformática sobre uma ferramenta capaz de controlar e acelerar o acesso a um banco de sequências pelo programa BLAST [2] chamada BioProvider. Este BLAST pode, dado um banco de sequências e um conjunto de sequências de consulta, aproximar quais sequências tem maior semelhança entre si com uma velocidade considerável comparado com algoritmos como o de Smith-Waterman, assim sendo vastamente usado por pesquisadores.

Um dos trabalhos futuros que não havia sido explorado dizia respeito ao uso da ferramenta na presença de bases de dados mais volumosas. Além disso, como o *kernel* dos sistemas operacionais pode se alterar com o tempo, seria necessário gerar uma versão mais robusta da ferramenta. Por fim, recentemente o NCBI passou a adotar uma versão nova de BLAST, que necessita de avaliação para verificar a adequação do uso com o Bioprovider

Objetivos

Checar se esta ferramenta continua funcionando para sistemas operacionais atuais, corrigir *bugs,* atualizar o programa, acelerar a execução e tornar o acesso à memória de programas BLAST mais eficiente através de um gerenciamento de *buffer e de* processos de forma não intrusiva.

Metodologia

Primeiramente foi preciso conferir se o *software* continuava funcional. Baseado nos *scripts* e nas documentações disponíveis foi criado um novo *script* de preparação para a execução do BLAST na presença do BioProvider. Este script mostrou que o BioProvider continua funcionando corretamente numa versão de *Ubuntu* com *kernel* mais atual que a do primeiro estudo (2.6.32-21-generic).

Este *script* basicamente prepara as execuções para o BLAST consultar um banco de sequências gerenciado pelo BioProvider. Para se fazer este acesso é preciso usar o formatdb do pacote de programas do BLAST visando preparar o banco para ser processado pelo programa blastall, que com parâmetros padrões, gera três arquivos: um arquivo de sequências (.psq), um de índices(.pin) e um de anotações (.phr).

Como o BioProvider faz o controle dos dados lidos a partir destes arquivos, principalmente o de sequências, ele precisa executar o formatdb uma vez e armazenar o tamanho deste arquivo de sequências para saber quantos blocos serão precisos a partir da quantidade de memória disponível para o seu anel de dados.

O anel é, resumidamente, uma lista encadeada circular de blocos que funciona como uma espécie de *buffer* do banco de sequências a serem consultadas. Os blocos são trechos do banco carregados na memória para acesso futuro. Quando todos os processos ativos processarem um determinado bloco, este pode ser substituído por um novo. O importante para que o BLAST funcione corretamente assim é que todo o bloco se inicie com um começo de sequência, desta forma, mesmo se lendo o banco fora de ordem, o resultado da execução será correto ao se terminar de processar todos os blocos.

Ao usar o BioProvider, os processos BLAST leem o arquivo gerenciado por ele de maneira diferente da habitual. Caso se deixe o sistema operacional controlar os outros dois arquivos sem se fazer nada enquanto o BioProvider controla apenas o acesso ao arquivo de sequências, ocorreria um problema, pois o arquivo de índices poderia estar apontando para

uma posição do arquivo sequências gerenciado pelo BioProvider errado. Para se solucionar este empecilho os arquivos sofrem uma "permutação" para cada tipo de visão que este pode ter.

O número de permutações deste é determinado pelo número de blocos necessários para se armazenar o banco no anel de memória, porque cada processo BLAST pode iniciar a leitura de um bloco qualquer, desde que este bloco comece com um início de sequência e no final todo o arquivo seja processado.

Estas permutações também são formatadas com o formatdb e suas informações são carregadas para o BioProvider através de um arquivo de configurações. O *driver* usado para passar informações de maneira não intrusiva é carregado no *kernel* e são gerados três arquivos de dispositivos de caracteres, gerenciados por este driver, que serão usados no lugar dos três arquivos originais gerados pelo formatdb. Desta maneira é possível controlar as informações lidas pelo BLAST de maneira transparente para o mesmo.

Com este novo *script* o BioProvider foi testado inicialmente com o banco pataa (de aproximadamente 150 Mbytes – versão de 2006). Para mostrar o momento que o BioProvider pode realmente auxiliar na execução do programa ao invés de o tornar mais lento, estes testes foram feitos com diferentes tamanhos de memória RAM disponível, tamanhos da sequência de consulta e a quantidades de BLASTs acessando o arquivo concorrentemente.

Os resultados da execução com o pataa mostraram que o BioProvider realmente pode acelerar a execução do programa BLAST em alguns casos, particularmente ao se processar bases grandes e ao se rodar muitos processos concorrentemente. Além disto, cabe ressaltar que o resultado dos testes obtidos com e sem o uso do programa provedor de dados são os mesmos.

Posteriormente partimos para um teste com um banco maior. Porém quando este passa de 2GBytes, o BLAST o quebra em bancos menores de até 2GBytes e usa um arquivo *alias* para os lerem como se fossem um único banco. Como o BioProvider inicialmente foi criado se baseando que o banco tinha sempre apenas um volume, ele ainda não consegue processar bancos de tal tamanho.

Então foram feitos testes com um banco de tamanho menor que 2GBytes e maior que o pataa. Este banco foi criado com 5 milhões de sequências aleatórias do env_nr gerados por um script. Assim como feito anteriormente, os testes foram feitos com diferentes configurações, além disto, eles também foram executados em diferentes máquinas com diferentes versões de sistema operacional.

Conclusões

Com estes testes se pode confirmar que o que foi dito durante o estudo da criação do provedor de dados continua válido em máquinas mais atuais.

O uso do BioProvider é realmente proveitoso quando o banco de sequências a ser consultado não cabe inteiramente na memória, pois a manipulação genérica oferecida pelo sistema operacional não pode otimizar o acesso a estas informações como o BioProvider faz, caso contrário o BioProvider tornaria a execução do BLAST mais lenta do que a sua execução normal.

Também com os testes se confirmou que o BioProvider e o seu driver podem funcionar em versões anteriores de *kernel* (9.1 e 9.04).

Referências

1 - NORONHA, Maíra Ferreira de; LIFSCHITZ, Sérgio. Controle da Execução e Disponibilização de Dados para Aplicativos sobre Seqüências Biológicas: o Caso BLAST. Rio de Janeiro, 2007. 83p.

2 – KORF, Ian; YANDELL, Mark; BEDELL, Joseph. BLAST. O'Reilly Media, 2003, 368pp.