1
Introdução

É grande o volume de dados advindos dos projetos das áreas de biotecnologia, bioinformática e biologia molecular. Um exemplo disso é a base EMBL(7), que contém algumas centenas de gigabytes, com crescimento correspondente a quadruplicar de tamanho a cada ano em termos de número de seqüências. Diante de tantos dados, se torna imprescindível que ferramentas computacionais agilizem o processo de análise na procura de informações que possam ser relevantes aos biólogos.

Dos atuais projetos de seqüenciamento de DNA, talvez a atividade mais exigida seja a comparação de seqüências. Trechos de nucleotídeos seqüenciados são comparados a várias outras seqüências em bancos de dados[1] para buscar informações que possam ser atribuídas a esses trechos. Com essa comparação, é possível inferir informações a partir das seqüências dos bancos de dados, das quais já se têm algum conhecimento prévio.

Para a atividade de comparar seqüências, as ferramentas da família BLAST (*Basic Local Alignment Search Tool*)(1) têm sido amplamente utilizadas com uso de heurísticas eficientes. Inclusive algumas específicas para seqüências de nucleotídeos (BLASTN) e outras para aminoácidos (BLASTP). Contudo, este conjunto de ferramentas se torna extremamente lento quando a base de dados não pode ser mantida inteiramente em memória principal (2, 30, 17).

No intuito de acelerar o processamento do BLAST, poderíamos simplesmente adquirir processadores mais rápidos. Contudo, enquanto a taxa de processamento desses duplica a cada 18 meses, os bancos de dados públicos de biosseqüências duplica a cada 16 meses (18). Dado que o tempo de pro-

[1] O termo utilizado neste manuscrito para bancos (ou bases) de dados diz respeito de modo geral a arquivos-texto não manipulados por um Sistema Gerenciador de Bancos de Dados (SGBD).

cessamento do BLAST é diretamente relacionado ao tamanho das bases de dados, a busca de estratégias para avaliar o BLAST com alto desempenho se tornou necessária. Em particular, este trabalho de pesquisa discute uma das abordagens que vem sendo utilizada, o uso de ambientes de computação distribuída e processamento paralelo para obter ganhos de desempenho durante o processamento do BLAST.

Na busca por estratégias que reduzam os custos de entrada e saída de dados (E/S), nada mais coerente do que buscar conceitos já estabelecidos na área de bancos de dados. Considerando bases de seqüências como bancos de dados contendo uma única tabela, e a possibilidade de distribuí-las em máquinas diferentes, lidamos com um problema conhecido de bancos de dados distribuídos: determinar a melhor forma de alocar fisicamente as bases de dados visando realizar acessos com melhor desempenho possível (26). Foram verificadas algumas estratégias de processamento já bastante difundidas (21, 16, 11), nas quais os bancos de dados se encontram fragmentados ou replicados em distintas estações de trabalho.

Quando se utilizam vários computadores trabalhando em paralelo, existe uma grande dificuldade em encontrar o melhor ajuste na distribuição de tarefas para cada máquina participante, no intuito de se evitar que alguma trabalhe mais do que outra. Quando há falta de equilíbrio dizemos que ocorre desbalanceamento de carga. Um fator importante a ser considerado na ferramenta BLAST para evitar este desbalanceamento é a possibilidade de seqüências de mesmo tamanho possuírem tempos de execução distintos quando comparadas ao mesmo banco de dados (11, 10). Por ser difícil prever esse desequilíbrio antes que a comparação do BLAST ocorra, métodos estáticos de balanceamento de carga devem ser evitados pois são pouco eficazes no balanceamento (10).

O objetivo principal deste trabalho é o estudo da execução da ferramenta BLAST em agrupamentos (*clusters*) de computadores, com uso de estratégias de bancos de dados distribuídos e paralelismo de E/S para obter melhor balanceamento de carga. Estaremos considerando para isso, que um agrupamento será formado por uma máquina dita coordenadora ou gerente, e outras ditas trabalhadoras.

As idéias tratadas neste trabalho tiveram origem em (9), onde foi sugerido que se estudasse a distribuição das bases de dados de forma a tirar

proveito das vantagens tanto das estratégias fragmentadas como replicadas. Nesta dissertação, explora-se particularmente uma das idéias sugeridas como trabalhos futuros em (9), onde cada estação de trabalho contém uma réplica da base de dados que é, por sua vez, dividida em partes distintas. Neste trabalho são definidas e implementadas algumas estratégias de paralelização do BLAST, com balanceamento de carga dinâmico baseado nesta alocação replicada (mostraremos mais adiante que a replicação total não é obrigatória) e alguns dos fragmentos considerados primários para execução local. Mais especificamente, os estudos aqui realizados fazem uso de métodos corretivos (quando um desequilíbrio de carga é detectado) e preventivos (com solicitação de tarefas sob demanda).

Neste trabalho a ênfase também foi dada em propor abordagens não intrusivas, isto é, as estratégias devem ser aplicadas de forma que o código fonte da ferramenta BLAST não seja alterado. Essa opção de não ser intrusiva tem algumas vantagens, por exemplo: (i) permite que as estratégias se apliquem a outras ferramentas semelhantes; (ii) não exige adaptações para futuras versões de implementação do próprio BLAST; e (iii) os usuário têm maior aceitação por propostas que não alteram a ferramenta básica a ser utilizada. Estas vantagens serão detalhadas mais adiante, nos próximos capítulos.

A pesquisa busca também alguns fatores de correção do processamento BLAST paralelo, são eles: robustez de processamento e estatísticas de alinhamento em bases fragmentadas. Em qualquer processamento em ambientes de computação paralela pode ocorrer que estações de trabalho se tornem inacessíveis; assim, implementamos estratégias que garantam a robustez do processamento permitindo que o sistema recupere-se de possíveis falhas. Referente ao uso de estatísticas de alinhamento em bases fragmentadas, propomos uma diferente utilização dos parâmetros disponíveis pelo BLAST, o que garante uma confiabilidade dos resultados obtidos em relação aos resultados de execução com a base de dados não fragmentada.

Embora existam inúmeros trabalhos na literatura (10, 30, 25, 36, 6) abordando a paralelização da ferramenta BLAST, as propostas desta dissertação buscam um balanceamento de carga e aproveitamento dos recursos computacionais, reduzindo o tempo de execução da ferramenta BLAST em relação aos outros trabalhos já apresentados. Essa melhora de desempenho faz uso de uma abordagem não intrusiva e mantendo a correção dos resultados. São apresentados vários resultados de testes práticos que permitem verificar

os ganhos obtidos.

O restante do texto desta dissertação está organizado conforme descrito a seguir:

- Capítulo 2: descreve as dificuldades e necessidades quando o BLAST é utilizado em agrupamentos de computadores, principalmente quando os bancos de dados são fragmentados;
- Capítulo 3: discute os fatores de desbalanceamento de carga durante a execução paralela da ferramenta BLAST;
- Capítulo 4: faz um estudo das abordagens referente à execução da ferramenta BLAST em *cluster* de computadores ;
- Capítulo 5: são apresentadas nossas soluções, com utilização dos conceitos de Banco de Dados Distribuídos;
- Capítulo 6: são mostrados os resultados experimentais obtidos com as propostas de estratégia do Capítulo 5;
- Capítulo 7: descrevem-se as contribuições e conclusões desta dissertação e as sugestões de trabalhos futuros.