

Resultados Obtidos

Foram realizados testes para avaliar o desempenho do NCBI BLAST usando o BioProvider. Os testes foram feitos em um computador Pentium 4 com processador de 3 GHz de velocidade e 512 *megabytes* de memória RAM. Foram usados os bancos de dados *nr* e *env_nr*, de proteínas, disponíveis em NCBIa (2006). O banco *nr* é um dos mais importantes para os biólogos e possui informações de diversos organismos. Em formato FASTA, o banco possui 1,74 *gigabytes*, sendo que o arquivo de seqüências gerado possui 1,2 *gigabytes*. Já o banco *env_nr* possui tamanho mediano, sendo seu arquivo em formato FASTA de 250 *megabytes* e o de seqüências de aproximadamente 190 *megabytes*. As seqüências de entrada utilizadas nos testes foram obtidas de 3 bancos de dados diferentes: *ecoli.aa*, *ptaa* e *swissprot*, disponíveis em NCBIa (2006). Foram criados 3 conjuntos de 50 seqüências de entrada. O primeiro conjunto possui seqüências de tamanhos semelhantes, variando entre 432 e 492 aminoácidos. O segundo conjunto possui seqüências de tamanhos variados, de 15 a 1320 aminoácidos. Já o terceiro conjunto possui as mesmas seqüências de tamanhos variados, entretanto ordenadas da menor até a maior.

Para avaliar o desempenho dos processos BLAST, foram realizados testes com diferentes variáveis, como o tamanho do anel, o número de processos em execução, as seqüências de entrada, o número de seqüências por bloco no banco de dados (e, conseqüentemente, o número de blocos) e estratégias de atendimento dos processos. Cada conjunto de testes tem como objetivo avaliar a influência de uma das variáveis, consistindo em rodadas de teste para cada valor analisado da variável. Em cada rodada de testes foram criados 50 processos BLAST, iniciados em intervalos de tempo fixos, um depois do outro. Deste modo, objetivou-se simular um ambiente no qual, de tempos em tempos, usuários executam processos BLAST. Cada processo utilizou na comparação uma seqüência de entrada retirada

do conjunto de seqüências, sendo que o processo n utiliza a seqüência n . Antes da execução dos testes um processo BLAST é executado para ler o banco de dados, o que impede que haja grandes discrepâncias quando os mesmos testes forem repetidos, já que parte do banco de dados estará sempre na memória quando o teste for iniciado. Os resultados de cada teste foram comparados com os mesmos obtidos pelo BLAST em execução normal, sem a utilização do BioProvider.

Para facilitar a comparação entre os resultados, foram usados nos testes o conjunto de seqüências de entrada com tamanhos semelhantes e a estratégia de atendimento dos processos que não trata de *starvation*. O uso do conjunto de seqüências de entrada com tamanhos aleatórios e a estratégia de impedir *starvation* produzem efeitos que variam dependendo do banco de dados a ser lido, logo, foram analisados em testes separados.

As diferentes variáveis usadas nos testes e seus valores são resumidos na tabela abaixo:

Teste	Banco	Núm. Blocos	Tam. Memória	Tam. Anel	Seqüências de Entrada	Intervalo entre Processos	Estratégia de Atendimento
1	Nr	5	Varia	50 M	Conjunto 1	1 min	Não tratar <i>starvation</i>
2	Nr	5	512 M	Varia	Conjunto 1	1 min	Não tratar <i>starvation</i>
3	Nr	5	512 M	50 M	Varia	1 min	Não tratar <i>starvation</i>
4	Env_nr	Varia	512 M	25 M	Conjunto 1	10 seg	Não tratar <i>starvation</i>
5	Nr	5	512 M	25 M	Conjunto 1	1 min	Varia

Tabela 1: Variáveis usadas nos testes e seus valores

Para variar o tamanho da memória disponível no computador, foi modificado o arquivo de configuração do programa GRUB (*boot loader*), limitando a memória da máquina para diferentes valores.

Nos testes realizados, a estratégia usada de não tratar *starvation* consiste em dar prioridade aos processos que estão lendo informações fora do anel, em seguida priorizar o processo mais atrasado do anel ou cuja requisição de leitura é da menor posição do arquivo, caso o processo não esteja na etapa de leitura do anel. Como o *starvation* não é tratado, o atendimento a novos processos nunca é proibido.

Os dados analisados nos testes são os tempos médios e totais de execução dos processos BLAST, o número médio de *page faults* que impactaram em leituras feitas ao disco e o número médio de mudanças de contexto, nas quais um processo BLAST é interrompido durante a execução, dando lugar a outro processo. Os dados foram obtidos usando-se o programa *time* do Linux. Os dados apresentados nos resultados são calculados como valores médios por processo BLAST, exceto o tempo total de execução. Para o cálculo do número médio de *page faults*, são somados o número de *page faults* dos processos BLAST ao número de *page faults* do processo provedor e divididos em seguida pelo número de processos BLAST, para se obter o valor médio correspondente ao sistema BLAST + BioProvider.

4.1.

Variação do Tamanho da Memória

Para analisar o impacto do tamanho da memória no desempenho dos processos, foram realizados testes variando o tamanho da memória RAM disponível no computador. A tabela a seguir apresenta os valores das variáveis nas diferentes rodadas de testes usando o BioProvider.

Banco	Núm. Blocos	Tam. Memória	Tam. Anel	Seqüências de Entrada	Intervalo entre Processos	Estratégia de Atendimento
Nr	5	512 M	50 M	Conjunto 1	1 min	Não tratar <i>starvation</i>
Nr	5	256 M	50 M	Conjunto 1	1 min	Não tratar <i>starvation</i>
Nr	5	128 M	50 M	Conjunto 1	1 min	Não tratar <i>starvation</i>

Tabela 2: Valores das variáveis no teste 1

Os mesmos testes foram realizados sem a utilização do BioProvider. Entretanto, foi verificado que, com 128 *megabytes* de memória RAM, a rodada de testes sem o BioProvider não pôde terminar, já que não havia terminado após 30 horas de execução. Os resultados dos testes são apresentados nos gráficos a

seguir, comparados com o desempenho dos processos BLAST sem a utilização do BioProvider.

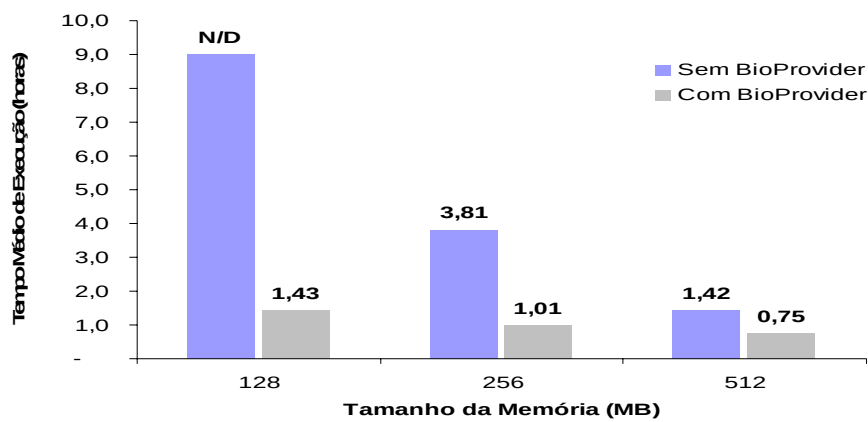


Figura 1: Gráfico dos tempos médios, teste 1

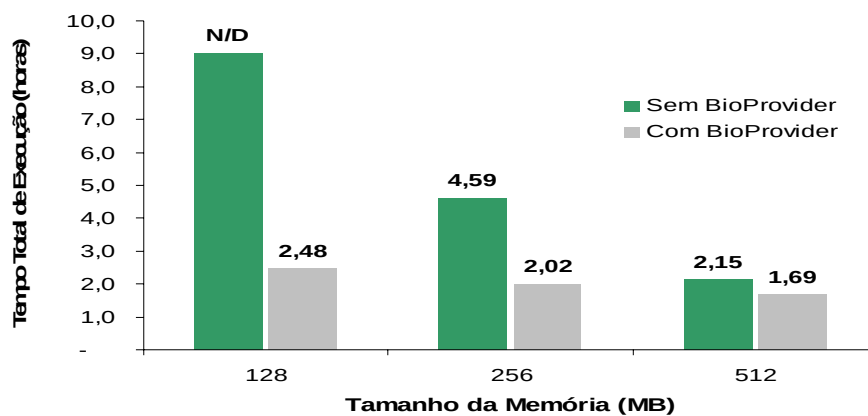


Figura 2: Gráfico dos tempos totais, teste 1

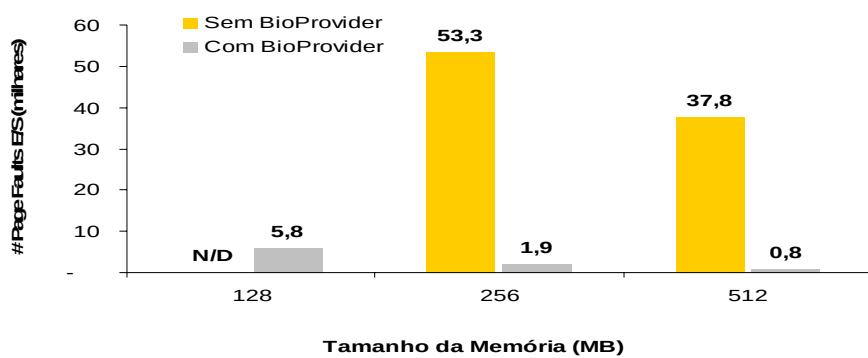


Figura 3: Gráfico dos números de *page faults*, teste 1

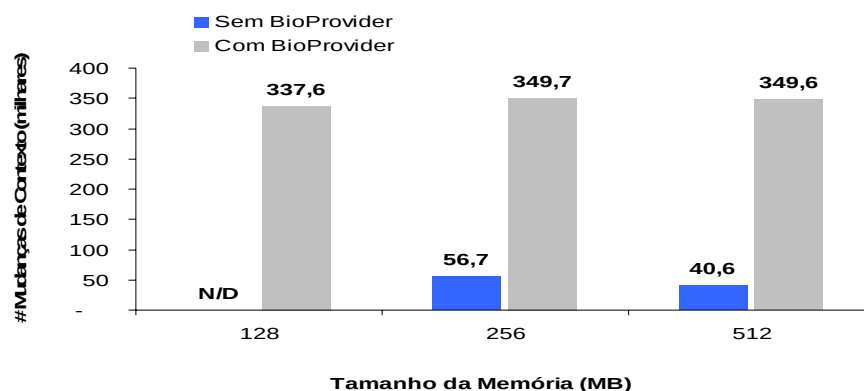


Figura 4: Gráfico dos números de mudanças de contexto, teste 1

Percebe-se que o tamanho da memória é um fator muito importante no desempenho dos processos BLAST, já que, quanto menor é a memória, pior é o desempenho destes. Por outro lado, quando o BioProvider é utilizado, os processos sofrem uma influência muito menor do tamanho da memória. Logo, quanto menor é a memória, maior é a vantagem da utilização do BioProvider. Além de possibilitar a execução de processos BLAST em situações caso contrário impeditivas, o BioProvider fez com que os tempos total e médio de execução dos processos diminuíssem respectivamente 56% e 73% no ambiente com 256 *megabytes* de memória, e 21% e 48% no ambiente com 512 *megabytes* de memória. Observa-se que a melhora de desempenho obtida com o BioProvider é atribuída ao menor número de *page faults* resultante. Apesar de haver um grande número de mudanças de contexto, devido à necessidade dos processos BLAST de serem bloqueados e interrompidos à espera de cada resposta a pedidos de leitura, este fato não impediu os processos de apresentarem melhor desempenho com o BioProvider.

4.2.

Variação do Tamanho do Anel em Memória

Para analisar o impacto do tamanho do anel no desempenho dos processos, foram realizados testes variando este. A tabela a seguir apresenta os valores das variáveis nas diferentes rodadas de testes usando o BioProvider.

Banco	Núm. Blocos	Tam. Memória	Tam. Anel	Seqüências de Entrada	Intervalo entre Processos	Estratégia de Atendimento
Nr	5	512 M	25 M	Conjunto 1	1 min	Não tratar starvation
Nr	5	512 M	50 M	Conjunto 1	1 min	Não tratar starvation
Nr	5	512 M	100 M	Conjunto 1	1 min	Não tratar starvation
Nr	5	512 M	250 M	Conjunto 1	1 min	Não tratar starvation
Nr	5	512 M	350 M	Conjunto 1	1 min	Não tratar starvation

Tabela 3: Valores das variáveis no teste 2

Os resultados dos testes são apresentados nos gráficos a seguir, comparados com o desempenho dos processos BLAST sem a utilização do BioProvider.

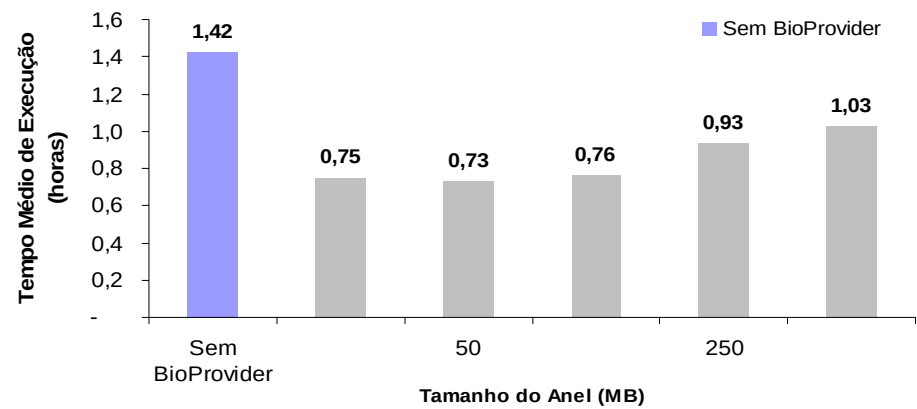


Figura 5: Gráfico dos tempos médios, teste 2

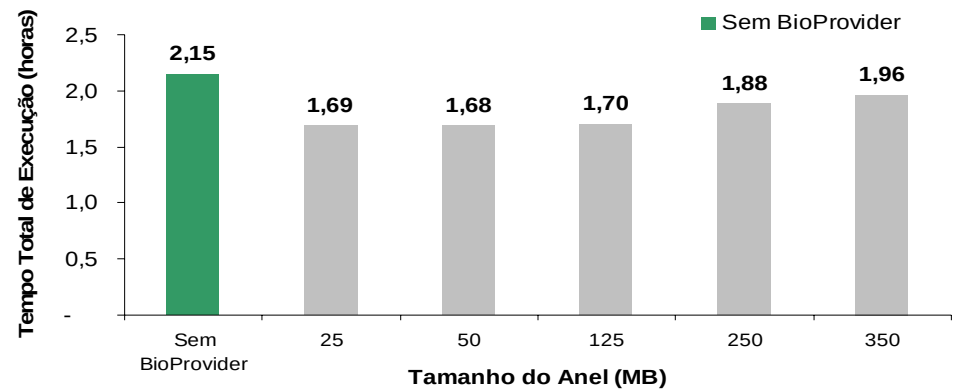


Figura 6: Gráfico dos tempos totais, teste 2

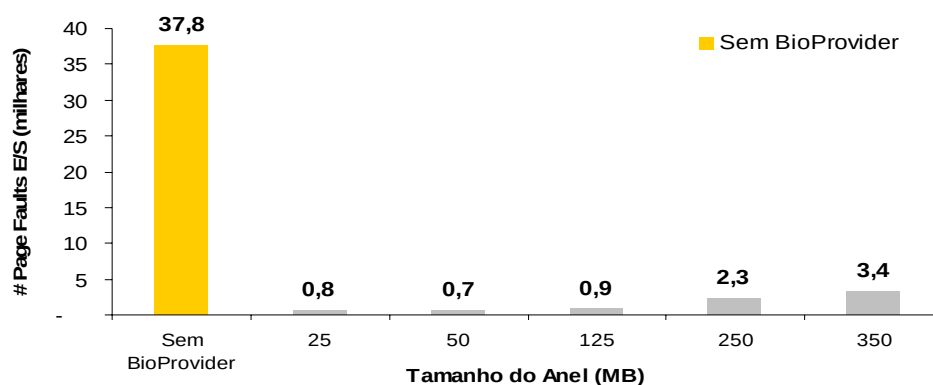


Figura 7: Gráfico dos números de *page faults*, teste 2

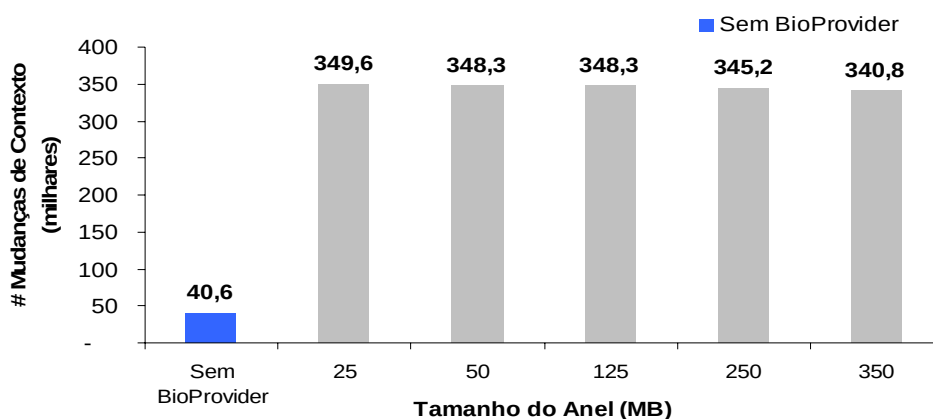


Figura 8: Gráfico dos números de mudanças de contexto, teste 2

Verifica-se que o desempenho dos processos piora quando o anel é muito grande, o que é esperado, já que passa a haver falta de memória para a execução dos processos. Ao mesmo tempo, pode ser notada uma ligeira piora de desempenho quando o anel é muito pequeno. Isto pode ser explicado pela necessidade de mais atualizações no anel para que os processos continuem suas leituras e pela necessidade ocasional de espera de alguns novos processos quando não há início de bloco no anel. O uso do BioProvider fez com que o tempo total de execução dos processos diminuísse entre 8% e 21% nos testes realizados, e o tempo médio de execução diminuísse entre 28% e 48%.

4.3.

Variação das Seqüências de Entrada

As consultas de entrada influenciam bastante no desempenho do BLAST. Para analisar este impacto, foram realizados testes variando as consultas usadas pelo BLAST, podendo ser do conjunto de seqüências de tamanhos semelhantes, o das de tamanhos aleatórios ou este último ordenado pelo tamanho das seqüências. A tabela a seguir apresenta os valores das variáveis nos testes.

Banco	Núm. Blocos	Tam. Memória	Tam. Anel	Seqüências de Entrada	Intervalo entre Processos	Estratégia de Atendimento
Nr	5	512 M	50 M	Conjunto 1	1 min	Não tratar starvation
Nr	5	512 M	50 M	Conjunto 2	1 min	Não tratar starvation
Nr	5	512 M	50 M	Conjunto 3	1 min	Não tratar starvation

Tabela 4: Valores das variáveis no teste 3

Os resultados são mostrados nos gráficos a seguir.

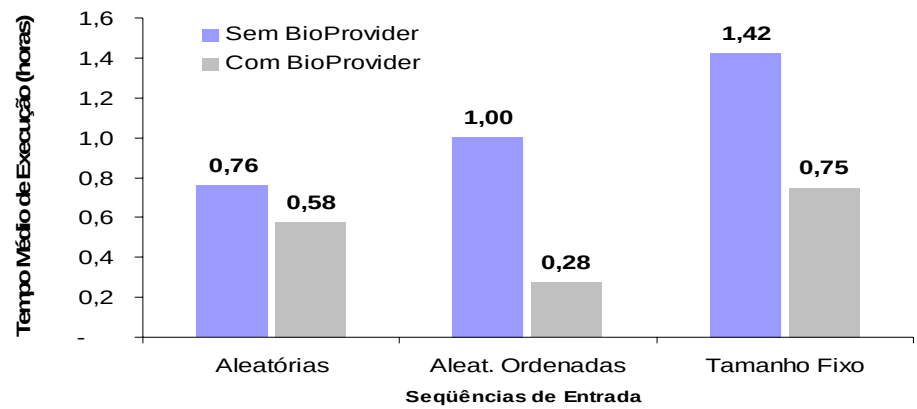


Figura 9: Gráfico dos tempos médios, teste 3

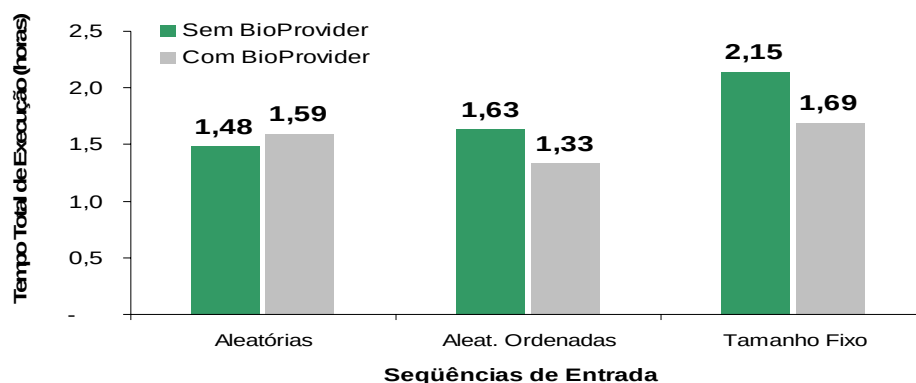


Figura 10: Gráfico dos tempos totais, teste 3

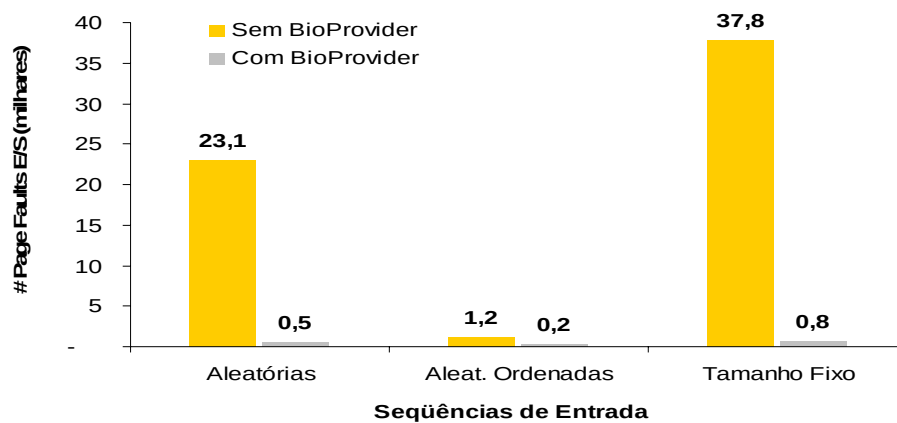
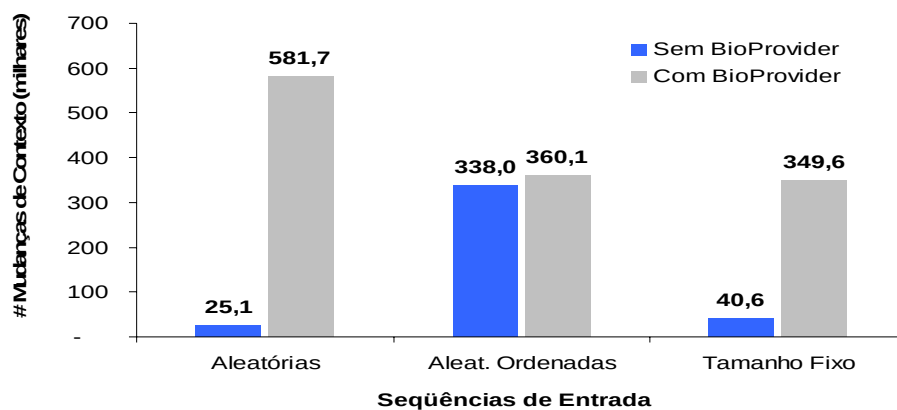
Figura 11: Gráfico dos números de *page faults*, teste 3

Figura 12: Gráfico dos números de mudanças de contexto, teste 3

Os resultados mostram uma situação na qual houve uma ligeira queda de desempenho usando-se o BioProvider, que foi verificada no tempo total de execução dos processos com sequências de entrada de tamanhos aleatórios.

Apesar de o tempo médio de execução ter diminuído em 24%, o tempo total aumentou em 7%. Como foi observado em Lemos et al. (003), a principal razão da piora de desempenho é porque processos BLAST com seqüências de entrada menores são, de modo geral, mais rápidos, e têm que esperar a leitura dos processos mais lentos para que o anel seja atualizado. Como os tamanhos das seqüências de entrada do conjunto variam muito, este fato influenciou bastante nos resultados. Por outro lado, se processos com tamanhos de seqüências de entrada semelhantes são executados juntos, o problema da aleatoriedade no tamanho das seqüências é diminuído. Isto foi feito na rodada de testes usando-se o conjunto de seqüências ordenadas pelo seu tamanho, já que os primeiros processos usaram seqüências de entrada menores e apenas os últimos usaram seqüências maiores. Nesse teste, o tempo total de execução diminuiu em 18% e o tempo médio diminuiu em 72%. Se todos os processos BLAST possuem seqüências de entrada de tamanho semelhante, o problema também é minimizado.

Finalmente, a rodada de teste na qual os processos usam seqüências de entrada de tamanhos semelhantes representa mais um exemplo no qual os processos executando ao mesmo tempo apresentam velocidades semelhantes. Nesse teste, o uso do BioProvider fez com que o tempo total de execução diminuísse em 21% e o tempo médio diminuísse em 48%.

4.4.

Variação do Número de Blocos

Foram realizados testes para analisar o impacto do número de seqüências por bloco, e conseqüentemente do tamanho do bloco, no desempenho dos processos BLAST. O banco *env_nr* foi dividido, primeiramente, em 8 blocos com o mesmo número de seqüências. Como o arquivo de seqüências tem 190 *megabytes*, cada bloco terá aproximadamente 24 *megabytes*, sendo próximo do tamanho do anel usado, de 25 *megabytes*. Devido ao tamanho dos blocos, é provável que, quando alguns novos processos começarem a ler do anel, estes encontrem um início de bloco apenas no final do mesmo, fazendo com que deixem de aproveitar alguns dados já presentes no *buffer*. Na segunda rodada de testes, o banco foi dividido em 14 blocos, fazendo com que cada bloco possua

aproximadamente 14 *megabytes*. Desse modo, o problema descrito será menor, pois haverá mais inícios de bloco no arquivo de seqüências. A tabela a seguir apresenta os valores das variáveis usadas nos testes.

Banco	Núm. Blocos	Tam. Memória	Tam. Anel	Seqüências de Entrada	Intervalo entre Processos	Estratégia de Atendimento
Env_nr	8	512 M	25 M	Conjunto 1	10 seg	Não tratar <i>starvation</i>
Env_nr	14	512 M	25 M	Conjunto 1	10 seg	Não tratar <i>starvation</i>

Tabela 5: Valores das variáveis no teste 4

Os resultados são mostrados nos gráficos a seguir.

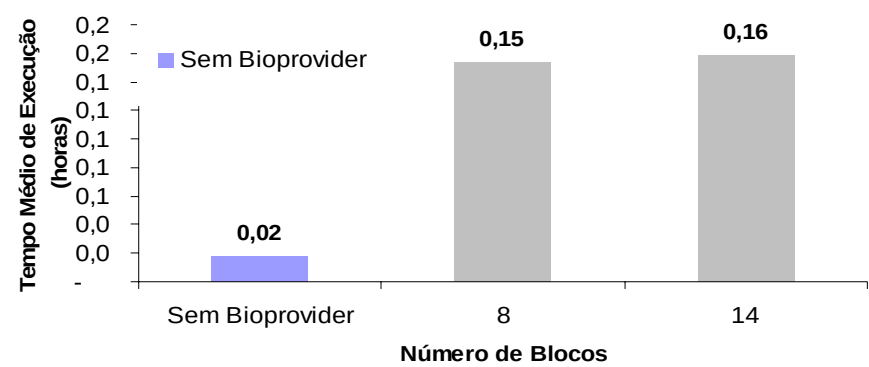


Figura 13: Gráfico dos tempos médios, teste 4

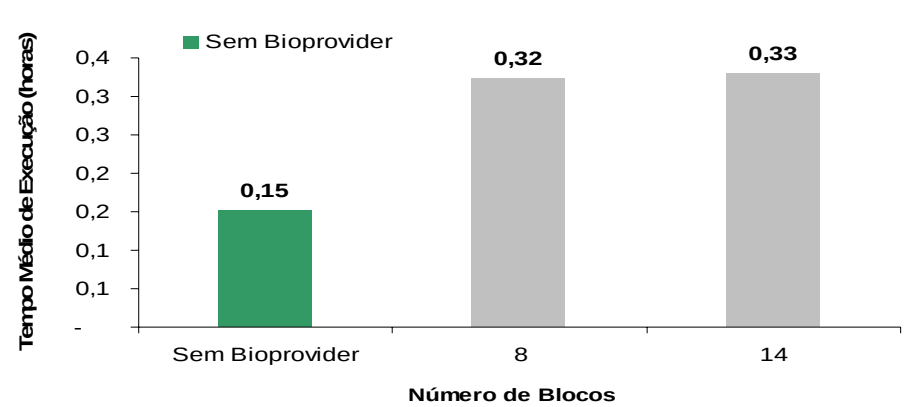


Figura 14: Gráfico dos tempos totais, teste 4

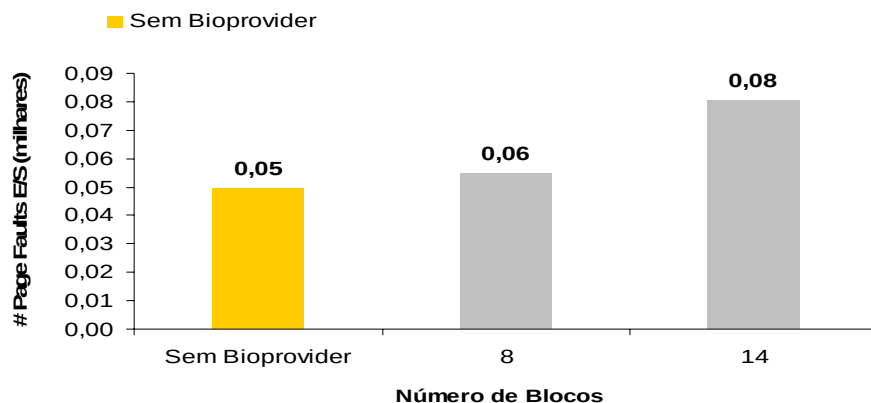
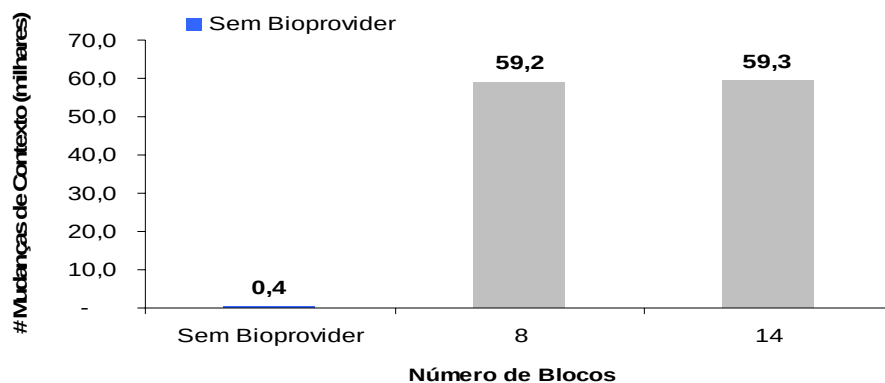
Figura 15: Gráfico dos números de *page faults*, teste 4

Figura 16: Gráfico dos números de mudanças de contexto, teste 4

Os testes variando o número de blocos do banco apresentaram desempenhos semelhantes, sendo que a divisão em 8 blocos gerou resultados ligeiramente melhores. Apesar de que, no momento em que iniciaram a leitura do anel, os processos BLAST tenham tido maiores chances de ler o conteúdo já disponível no anel quando o banco foi dividido em 14 blocos, este fato não compensou o maior número de *page faults* resultantes de leituras de um número maior de instâncias dos arquivos de índices e anotações. Conclui-se, então, que nem sempre vale a pena aumentar o número de blocos do banco.

Estes testes são também exemplos de situações nas quais o uso do BioProvider não é vantajoso. Como o arquivo de seqüências do *env_nr* é bem menor que a memória disponível no computador, o gerenciamento de *buffer* realizado pelo sistema operacional foi satisfatório, pois foi possível manter grande parte do arquivo em memória durante a execução dos processos. Usando o

BioProvider, o tempo médio de execução dos processos nas duas rodadas aumentou aproximadamente oito vezes, sendo que o tempo total foi o dobro.

4.5.

Varição da Estratégia de Atendimento dos Processos

Foram realizados testes para avaliar a estratégia de atendimento de processos que trata o *starvation*, que acontece quando alguns processos nunca são atendidos, em detrimento de outros. Em tentativas de criar uma situação na qual há *starvation*, o computador foi sobrecarregado e não foi possível obter resultados. Entretanto, foi possível verificar a influência do uso da estratégia no desempenho dos processos em situações nas quais naturalmente não ocorreria *starvation*.

Na estratégia que evita a ocorrência de *starvation*, o tempo durante o qual novos processos podem ser atendidos e o tempo durante o qual novos processos não são atendidos são, ambos, proporcional ao parâmetro t . Foram realizadas rodadas de testes com dois valores diferentes de t , cujos resultados foram comparados com a mesma rodada usando a estratégia que não trata de *starvation*. A tabela a seguir apresenta os valores das variáveis usadas nos testes.

Banco	Núm. Blocos	Tam. Memória	Tam. Anel	Seqüências de Entrada	Intervalo entre Processos	Estratégia de Atendimento
Nr	5	512	25	Conjunto 1	1 min	Não tratar <i>starvation</i>
Nr	5	512	25	Conjunto 1	1 min	$t = 4 \text{ seg.}$
Nr	5	512	25	Conjunto 1	1 min	$t = 30 \text{ seg.}$

Tabela 6: Valores das variáveis no teste 5

Os resultados são mostrados nos gráficos a seguir.

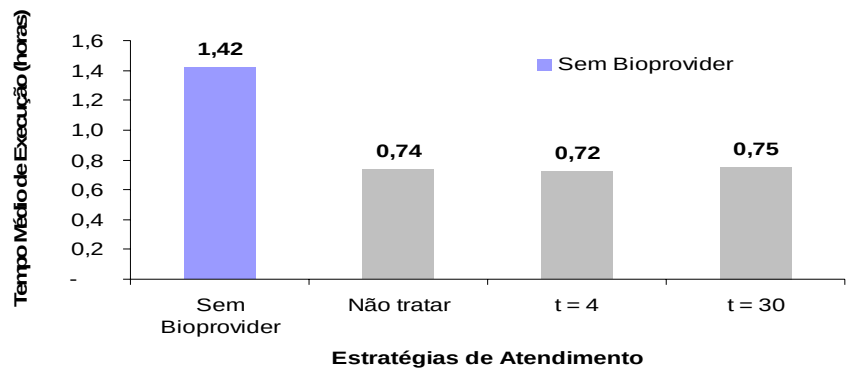


Figura 17: Gráfico dos tempos médios, teste 5

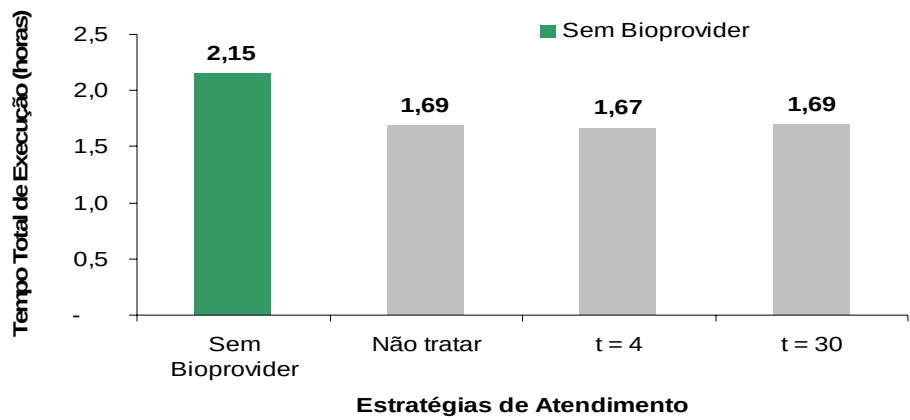


Figura 18: Gráfico dos tempos totais, teste 5

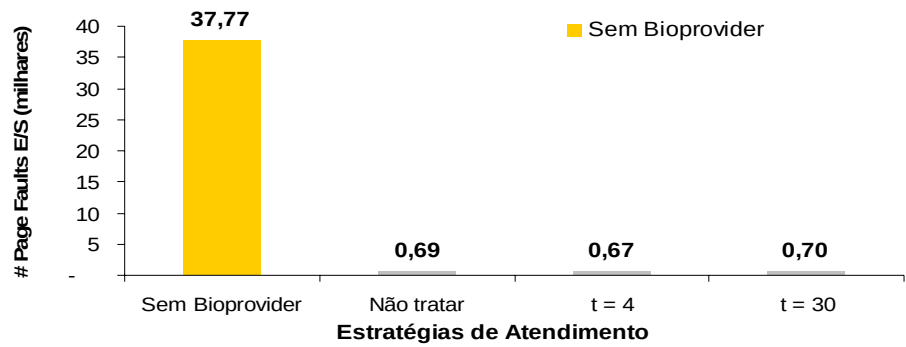


Figura 19: Gráfico dos números de *page faults*, teste 5

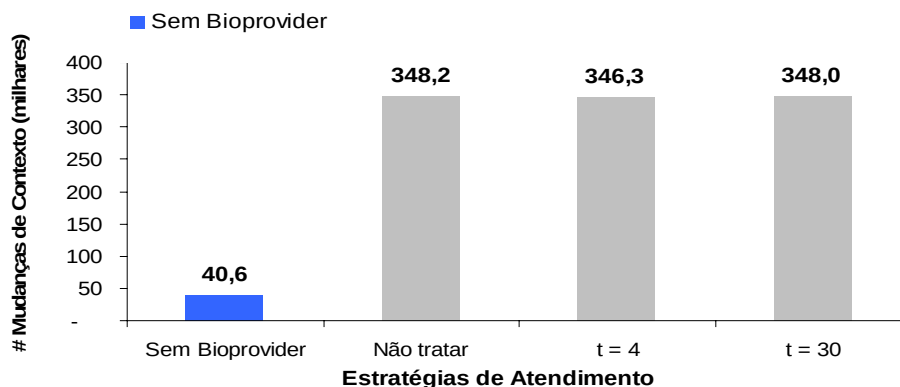


Figura 20: Gráfico dos números de mudanças de contexto, teste 5

Verifica-se que o desempenho dos processos não foi prejudicado quando a estratégia que impede *starvation* foi usada. Logo, em situações semelhantes é vantajoso utilizar esta estratégia, para garantir que nenhum processo deixe de ser atendido em detrimento de outros.

4.6.

Considerações Finais

Neste capítulo foram realizados testes para avaliar o desempenho do NCBI BLAST usando o BioProvider e identificar a influência de certos fatores no desempenho do BLAST. Na maioria dos testes realizados os processos BLAST apresentaram melhores desempenhos usando-se o BioProvider. Entretanto, foram apresentadas também situações nas quais o uso deste não é vantajoso.

Nos testes realizados na seção 4.1, foi possível verificar que o tamanho da memória é um fator muito importante no desempenho dos processos BLAST, mas quando o BioProvider é utilizado, os processos sofrem uma influência menor do tamanho da memória. O desempenho dos processos usando a ferramenta foi muito superior ao desempenho dos mesmos sem usá-la, sendo que esta possibilitou também a execução de processos BLAST em situações caso contrário impeditivas.

Na seção 4.2, foi observado que o tamanho do anel é um fator importante para o desempenho dos processos usando-se o BioProvider. Ao mesmo tempo que o anel não deve ser muito grande, pois pode prejudicar os processos se sobrar

pouca memória, o anel também não deve ser muito pequeno, pois isto faz com que os processos fiquem mais tempo bloqueados aguardando atualizações do anel.

Os testes da seção 4.3 mostraram que o BioProvider produz melhores resultados quando os processos BLAST em execução no mesmo momento possuem seqüências de entrada de tamanhos semelhantes.

Na seção 4.4, foi observada a influência do aumento do número de blocos nos quais o banco é dividido. Foi possível concluir que há situações nas quais não vale a pena aumentar o número de blocos do banco, pois isso acarreta em um número maior de *page faults* durante a leitura das diferentes instâncias dos arquivos de índices e anotações. Os testes são também exemplos de situações nas quais o uso do BioProvider não é vantajoso, pois o arquivo de seqüências do banco é bem menor que a memória disponível no computador.

Finalmente, nos testes da seção 4.5, foi possível verificar que a estratégia de bloquear o atendimento de novos processos de tempos em tempos não deteriorou o desempenho do BLAST, sendo então vantajosa em situações semelhantes quando se desejar garantir que o anel possa sempre ser atualizado.