6
Avaliação prática das novas Estratégias

No Capítulo anterior sugerimos duas estratégias de alocação de submissão de seqüências de consulta: Corretiva e Sob Demanda. Ambas utilizam a alocação da base de dados replicada em cada estação de trabalho e dividida em diversos fragmentos. A partir das sugestões propostas, mostraremos neste Capítulo alguns resultados analisando diferentes formas de execução. Pretendemos mostrar resultados para: estado de aceleração (*speedup*), distintas bases de dados e alteração no número de fragmentos. Cada um desses serão discutidos nas próximas seções.

O ambiente de execução para os testes pode ser visto no Apêndice 1. Para todos os testes apresentados neste Capítulo, foi utilizado o mesmo arquivo contendo 1.000 seqüências de consulta. Além disso, exceto nos testes que analisam diferentes bancos de dados, utilizamos as mesmas base de dados *nt* com distintos fragmentos.

6.1
Estado de Aceleração

A aceleração mede o ganho de tempo em um sistema paralelo à medida que novos recursos de *hardware* sejam adicionados ao ambiente, e isso sem alterar a complexidade do sistema (15). Desta forma, ao duplicar o poder de *hardware* em um sistema paralelo, se espera que o tempo total de execução diminua na mesma proporção. Quando isto ocorre, dizemos que houve uma aceleração linear no tempo. Embora em muitas aplicações paralelas isso não seja possível obter, em outras a aceleração pode ser melhor que linear.

Em nossos testes utilizamos uma variação de 2, 4 e 8 máquinas e mantivemos a mesma base de dados fragmentada em 48 fragmentos. Não foi possível executar testes com um número superior a oito máquinas, devido limitações do nosso agrupamento.

Nas Figuras 6.1 e 6.2 estão os gráficos para a estratégia Corretiva e Sob Demanda, respectivamente. Nessas é possível obter uma aceleração linear, à medida que o número de máquinas é duplicado. Isso ocorre pelo caráter adaptativo do balanceamento de carga disponível em ambas as estratégias. Essa adaptação não exige nenhum custo durante o processamento para diferentes configurações no número de máquinas.

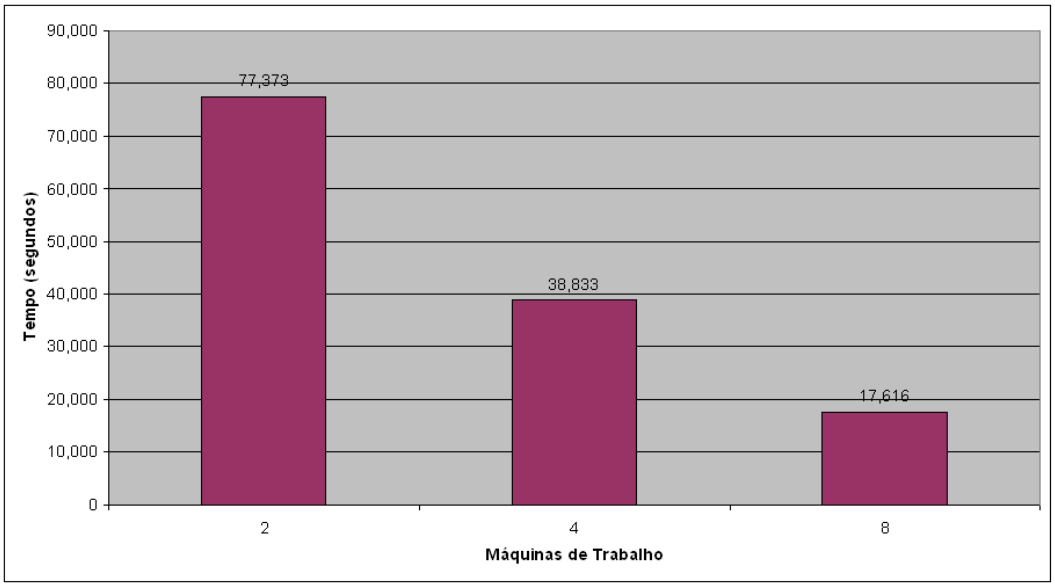

Figura 6.1: *Execução da estratégia Corretiva variando o número de máquinas de trabalho.*

6.2 Diferentes tamanhos da Base de Dados

A vantagem de utilizar bancos de dados fragmentados ocorrerá quando o ganho de tempo na paralelização no acesso ao disco superar o tempo necessário para executar a consulta contra todo o banco de dados não fragmentado, mesmo considerando os custos adicionais. Esse ganho tende a ser evidente em casos em que a base é maior que a memória primária. A paralelização no acesso ao disco deve trazer ganhos suficientes que superem os custos adicionais ao processo de fragmentação da base de dados, são eles: montagem e comunicação dos resultados e o elevado número de chamadas à ferramenta BLAST.

Nesse sentido, apresentaremos (a seguir) algumas execuções para as nossas estratégias variando entre as seguintes bases de dados: ecoli.nt, SwissProt, e nt. Todas podem ser obtidas no site do NCBI(4), e foram essas utilizadas

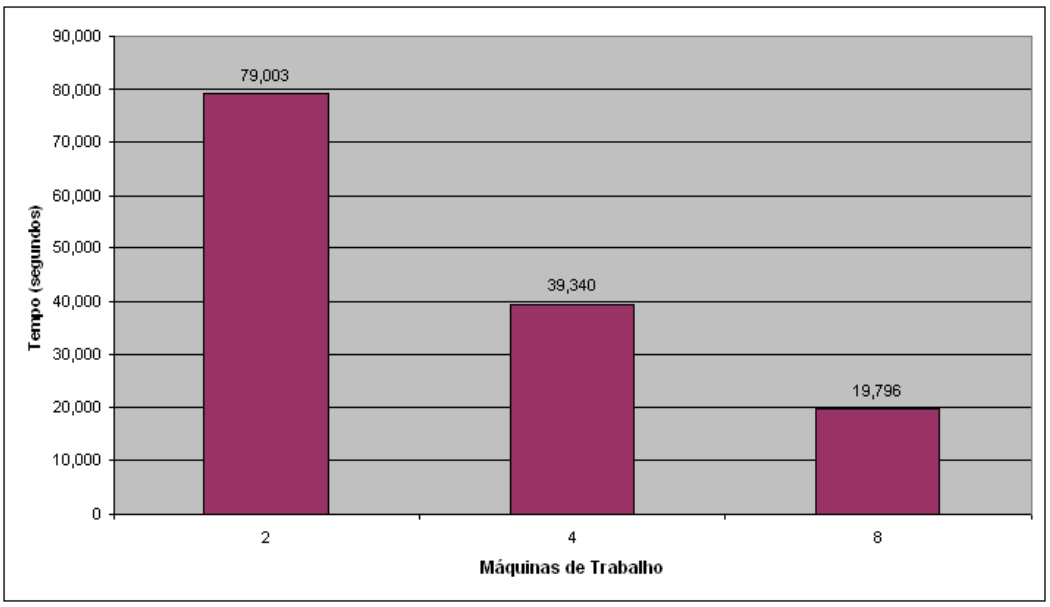

Figura 6.2: *Execução da estratégia Sob Demanda variando o número de máquinas de trabalho, com rS=32.*

a fim de variar a análise com diferentes tamanhos do banco de dados. A base ecoli.nt se refere ao genoma da Escherichia-coli. Já a segunda contém as informações da última versão da base de dados de aminoácidos SWISS-PROT. O nt contém algumas bases de dados do GenBank(3), do NCBI e outras, de forma não redundante. As informações dessas bases podem ser vistas na Tabela 6.1.

	Ecoli.nt	**SwissProt**	**nt**
Número de Seqüências	400	247,743	5.214.551
Número de caracteres	4.662.239	93.466.512	20.481.288.407
Tamanho do arquivo (MB)	4,6	116	20.000

Tabela 6.1: *Propriedade das bases de dados utilizadas nos testes.*

Iniciemos então apresentando na Figura 6.3 os resultados obtidos quando utilizada a base de dados *ecoli.nt* em 8 máquinas de trabalho. Para os resultados desta figura as estratégias propostas utilizaram 8 fragmentos, e em espécial para a Estratégia Sob Demanda, foi utilizado 32 seqüências por tarefa. Adicionamos também os resultados das estratégias Fragmentada e Replicada para que possam ser comparados.

Analisando os resultados da Figura 6.3, verificamos o que já era esperado para as estratégias Fragmentada e Corretiva, pois como a base *ecoli.nt* possui somente 4,6 MB (tamanho bem inferior à memória primária), a estratégia

Replicada possui melhor desempenho. O principal motivo para isso é o custo no aumento do número de execuções BLAST que irão ocorrer. Nesse teste utilizamos 1.000 seqüências de consulta e em uma estratégia Replicada foram 1.000 execuções. Contudo, usando a estratégia Fragmentada e Corretiva, foram 8.000 execuções para cada estratégia. Cada uma dessas execuções é uma chamada ao sistema e a criação de um novo processo, que considerando a base de dados menor que a memória, aumenta efetivamente o tempo total. Em relação à estratégia Replicada, as estratégias Fragmentada e Corretiva tiveram um custo adicional de somente 5% referente à comunicação e montagem dos relatórios. Este valor mostra mais uma vez que o aumento de tempo se refere principalmente ao número de execuções BLAST exigidas nas estratégias Fragmentada e Corretiva.

Inclusive, é por este mesmo motivo que é possível justificar o tempo final de execução da Estratégia Sob Demanda ter sido menor do que estratégia Replicada. Na Estratégia Replicada utilizada, toda tarefa é submetida assim que uma estação de trabalho se encontra ociosa. Contudo, cada tarefa terá somente uma seqüência de consulta, contabilizando um total de 1.000 execuções. Já no caso da Estratégia Sob Demanda (utilizando 32 seqüências por tarefa, conforme Figura 5.4), todas as 1.000 seqüências de consulta são divididas em aproximadamente 31 grupos com 32 seqüências. Cada grupo gera uma única execução BLAST para um fragmento. Como utilizamos 8 fragmentos, serão aproximadamente 248 execuções BLAST. Número bem inferior que as 1.000 execuções geradas pela estratégia Replicada. Embora a estratégia Replicada possa ser implementada para que várias seqüências sejam inseridas em uma única tarefa, nosso objetivo aqui é mostrar o ganho obtido quando várias seqüências são inseridas por tarefa.

Outra observação da Figura 6.3 é a proximidade no tempo final entre as estratégias Fragmentada e Corretiva. Devido ao pequeno tamanho da base de dados, não houve necessidade de ajuste no balanceamento de carga na Estratégia Corretiva. Isso mostra o baixo custo no controle para o balanceamento, já que a Estratégia Fragmentada não contém nenhuma implementação para tal.

Agora vamos analisar os resultados obtidos quando a base de dados SwissProt é utilizada, apresentada na Figura 6.4. A diferença no tempo final das estratégias diminui em relação à utilização da base ecoli.nt, pois embora ainda menor que a memória primária, o tamanho da base SwissProt é vinte

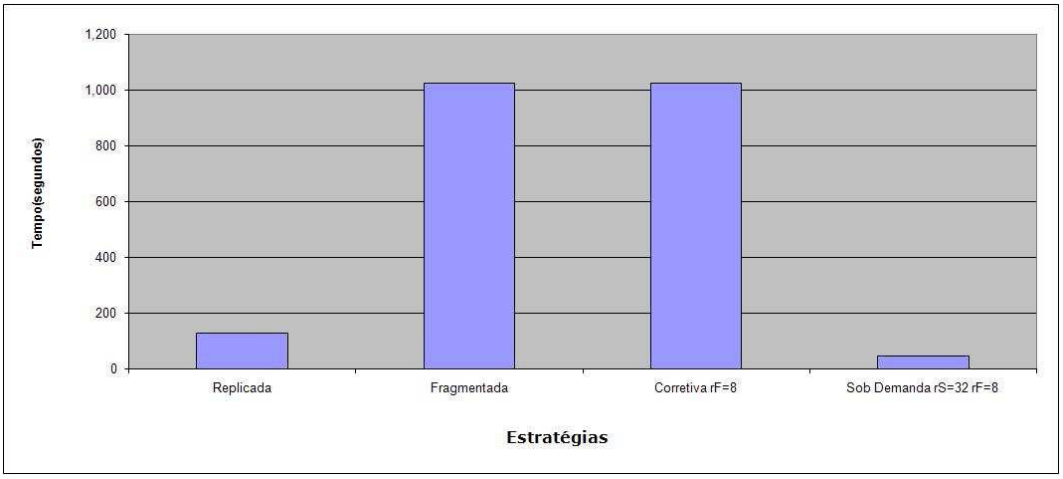

Figura 6.3: *Execução das estratégias utilizando a base de dados ecoli.nt.*

vezes maior que a base *ecoli.nt*.

Com relação aos resultados da Figura 6.4 existem duas importantes observações. A primeira é a pouca distinção nos resultados entre a estratégia Fragmentada e Corretiva, embora essa última ainda possua menor tempo. Novamente esta semelhança ocorreu pelo pequeno desvio de balanceamento encontrado, por se tratar de uma base de dados menor que a memória primária. Contudo, o pequeno desbalanceamento identificado foi tratado pela Estratégia Corretiva.

A segunda observação importante é a relação entre os tempos das estratégias Replicada e Sob Demanda, que difere dos resultados obtidos em relação à base de dados *ecoli.nt*. Para explicar isto, verificamos que o custo de execução total da ferramenta BLAST em cada máquina, entre estas duas estratégias, estão bem próximos quando utilizado a base de dados SwissProt. Já o tempo de comunicação e de montagem dos resultados teve grande aumento, devido ao crescimento da base de dados, o que conseqüentemente gera relatórios de saída maiores, gerando maior custo para a Estratégia Sob Demanda.

Como teste final na variação do banco de dados, comparamos todas as estratégias utilizando a base de dados *nt*, com tamanho extremamente maior em relação às outras bases. Para as execuções nas estratégias sugeridas foram utilizados 24 fragmentos, e na estratégia Fragmentada foram utilizados 8 fragmentos.

A Figura 6.5 apresenta os ganhos já esperados (discutidos no Capítulo

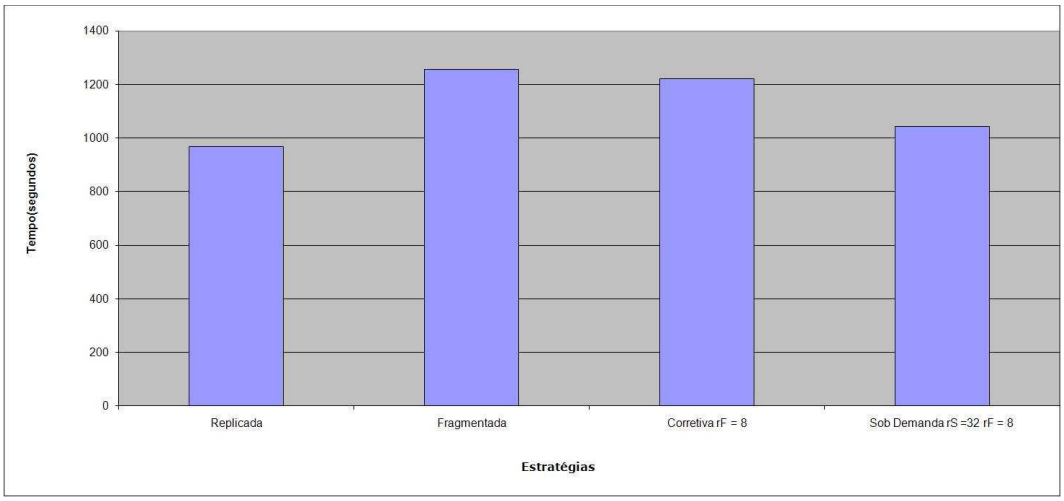

Figura 6.4: *Execução das estratégias utilizando a base de dados SwissProt*

5) quando utilizado as estratégias propostas em relação às outras. Contudo vale justificar o motivo pelo qual a estratégia Fragmentada teve tempo final superior à estratégia Replicada. Embora a base nt estivesse fragmentada em 8 fragmentos, cada um desses ainda possui 3 vezes o tamanho da memória primária. Nesse caso, para que a estratégia Fragmentada pudesse mostrar melhor desempenho, o número de máquinas deveria aumentar para que cada fragmento tivesse tamanho inferior à memória primária.

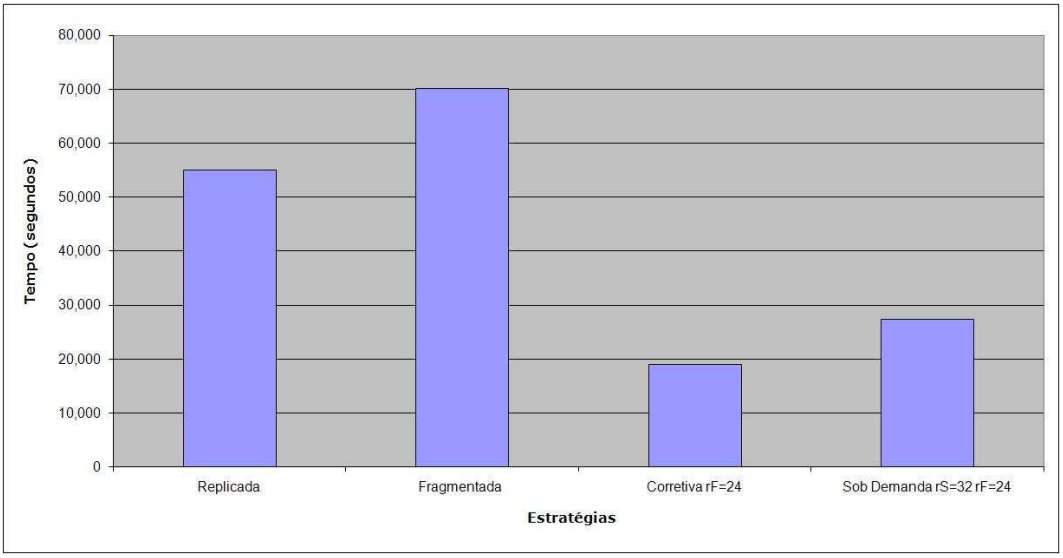

Figura 6.5: *Execução das estratégias utilizando a base de dados nt.*

6.3 Diferentes Números de Fragmentos da Base de Dados

Nesta seção analisaremos o comportamento das estratégias sugeridas variando a quantidade de fragmentos para o mesmo banco de dados nt. Nosso objetivo com este teste é observar até quanto vale a pena fragmentar a base de dados, considerando o paralelismo no acesso ao disco e o custo de montar os resultados gerados.

A Figura 6.6 apresenta o tempo total de execução para as Estratégias Corretiva e Sob Demanda, variando o número de fragmentos entre 8, 24, 48 e 96. Com o agrupamento utilizado, a partir de 24 fragmentos, cada um passa a ter tamanho menor que a memória primária.

Com a Figura 6.6, percebemos que uma vez que a base de dados seja dividida em fragmentos, o maior número de fragmentos pode não necessariamente diminuir o tempo de execução, mesmo que permita menor granularidade de balanceamento de carga. Assim, quando utilizados 96 fragmentos o custo de montagem e o execuções da ferramenta BLAST aumentou o tempo de execução final, comparado à execução utilizando 48 fragmentos.

É importante analisar a partir da Figura 6.6, a necessidade de obtenção do balanceamento de carga no agrupamento de computadores utilizado. A tendência é que em ambientes com maior desvio necessitem de maior número de fragmentos, possibilitando maior balanceamento.

6.4 Conclusão

Neste Capítulo fizemos vários testes a fim de analisar a execução das estratégias propostas e algumas observações podem ser obtidas. A primeira é que ambas as estratégias tiveram uma aceleração linear a medida que novos recursos de *hardware* são inseridos. Discutimos também, que das estratégias propostas, a Sob Demanda foi a que teve melhor desempenho para bancos de dados pequenos, pois consegue executar o BLAST uma única vez para várias seqüências de consulta.

Com as proximidades de tempo para banco de dados pequenos entre a estratégia Corretiva e a Fragmentada, verificamos que o custo para controle do balanceamento na Estratégia Corretiva é mínimo, sem valor relevante.

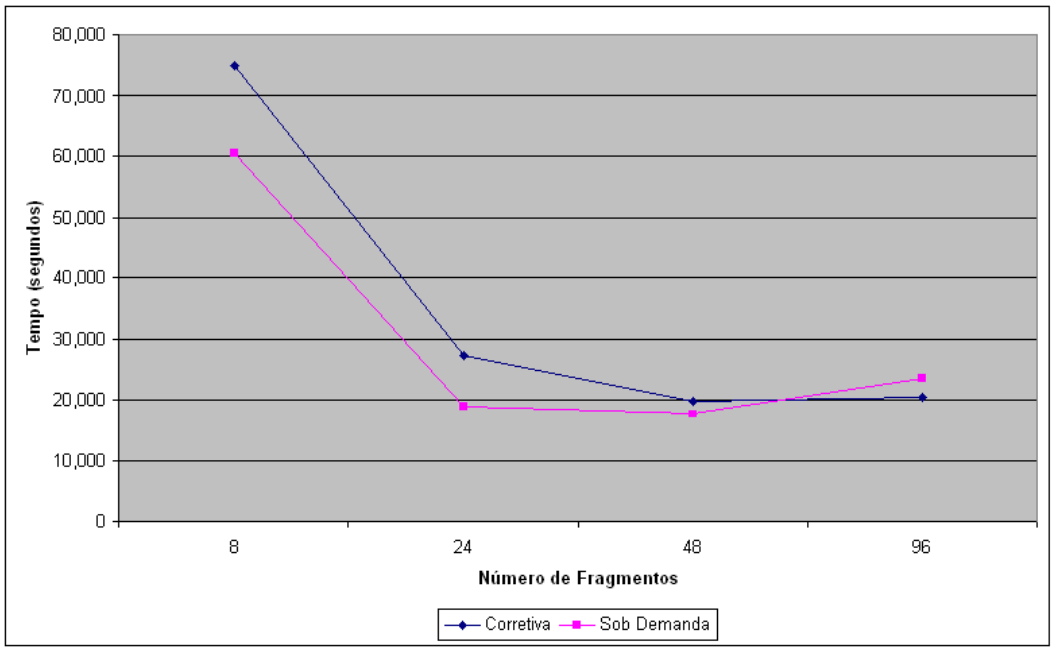

Figura 6.6: *Execução das Estratégias Corretiva e Sob Demanda alterando o número de fragmentos que compõem a base de dados.*

Verificamos que não é necessariamente o número de fragmentos da base de dados que poderá conduzir ao melhor desempenho. Em nossos exemplos, com o número de fragmentos acima de 48, houve um aumento no tempo final de execução.