2.

Avaliação e Comparação de Ferramentas de Software.

De um modo geral, *benchmarking* [50] é entendido como "um processo sistemático e contínuo de avaliação dos produtos, serviços e processos de trabalho das organizações". Por avaliação entende-se o processo sistemático de determinação do mérito, do valor e da significância de algo, considerado um conjunto de padrões.

Em computação, efetuar um *benchmark* pode ser entendido como o ato de executar um conjunto de avaliações sobre determinadas aplicações, de modo a obter o desempenho relativo de cada uma delas. Essas avaliações são efetuadas utilizando-se um conjunto padrão de testes e devem permitir, a partir da análise dos resultados obtidos, que seja realizada alguma comparação entre as mesmas.

A utilização de alguma metodologia, que defina um conjunto de ações para efetuar as avaliações, e o estabelecimento de fatores e critérios a considerar em um benchmark, constituem a chave para obter um resultado mais confiável.

Este capítulo tem por objetivo mostrar os fatores e critérios que devem ser considerados na avaliação de ferramentas de software, em particular aquelas voltadas à execução do BLAST em paralelo, de modo a permitir que se faça uma comparação entre elas, ou benchmark, de modo a indicar qual delas é mais apropriada em uma determinada aplicação.

2.1

Avaliação e Comparação de Ferramentas de *Software*

Por definição, *benchmarking* constitui um processo de mensuração e comparação do desempenho de produtos e serviços [7]. Por exemplo, em

computação, a partir da utilização de ferramentas conhecidas como *benchmark*, são efetuadas diversas medições com o objetivo de comparar diferentes sistemas computacionais, como é o caso do TPC-C, que emula os componentes fundamentais de um ambiente de processamento de transações, e possui como métrica para comparação de desempenho o número de transações online (OLTP) efetuadas. [33].

Assim, *benchmarking* é, antes de tudo, um processo de comparação, cujo objetivo é estabelecer qual produto ou serviço possui o melhor desempenho. Para isso é necessário que se identifique o que se quer efetivamente avaliar, ou seja, que fatores podem ser considerados importantes e que possam ponderar alguma decisão.

Em uma avaliação de ferramentas de *software*, os seguintes fatores podem ser considerados:

a) Desempenho:

   Trata-se da "atuação desejada ou observada de um indivíduo, ou grupo. na execução de uma tarefa, cujos resultados são posteriormente analisados para avaliar a necessidade de modificação ou melhoria" [11]. Transportando o conceito para a área da computação, pode ser associado à velocidade com que uma tarefa é executada por uma ferramenta de software;

b) Abrangência:

   Tem como significado: chegar a, alcançar, atingir [11]. Em computação pode ser associada ao tamanho do domínio em que uma ferramenta pode ser utilizada;

c) Funcionalidade:

   Pode ser entendido como algum instrumento que possua uma função de uso prático e de compreensão imediata [12]. No caso das ferramentas de software, pode ter como significado a finalidade para a qual uma determinada aplicação foi desenvolvida, e que uma vez utilizada atenda ao que dela seja desejado;

d) Usabilidade:

É um termo usado para definir a facilidade com que as pessoas podem empregar uma ferramenta ou objeto a fim de realizar uma tarefa específica e importante [12]. Aplica-se em computação à forma como um programa pode ser utilizado por um usuário, e que facilidades ele oferece para que se possa fazer uso de toda a sua potencialidade;

e) Acurácia:

Interpretado como "Isento de falha, erro ou defeito" [11]. Liga-se ao fato de uma aplicação apresentar sempre os mesmos resultados para um mesmo conjunto de dados, e esses resultados serem considerados corretos, de acordo com a especificação adotada para aquela aplicação.

f) Facilidade

De significado semelhante ao item d, pode ser relacionada à "ausência de obstáculos e dificuldades" [4]. Dessa forma uma aplicação não oferece nenhum problema para o entendimento de quem vai utilizar a aplicação com algum objetivo.

g) Adequabilidade

Pode ser entendido como a "capacidade de tornar-se ajustado, adaptar-se, amoldar-se", ou seja, uma aplicação tem que atender efetivamente às necessidades de um usuário, tanto em termos de sua instalação, considerada a infra-estrutura existente (equipamentos, sistema operacional, *middleware*), quanto ao seu uso.

Os fatores acima listados estão incluídos no padrão ISO 9126-1 [13], que institui normas de qualidade para produtos de software, e que inclui alguns outros tais como Manutenibilidade e Portabilidade, que apesar de importantes não são pertinentes à presente discussão. A adequabilidade e acurácia, incluídas nas normas integradas à funcionalidade, podem ser vistas aqui como fatores à parte, e que devem ser analisados separadamente.

Efetuar uma avaliação e comparação de ferramentas de software exige que se tenha alguma motivação. Caso essa motivação seja utilizar uma ferramenta BLAST que permita atender ao processamento de um grande volume de dados

oriundos de seqüenciamento de genomas, então é necessário que ela seja robusta, e que possua um bom desempenho.

Uma vez estabelecidas as ferramentas que satisfazem a possíveis restrições que possam ser impostas, então é necessário definir uma estratégia que seja adequada a uma confrontação. Em se tratando de ferramentas BLAST, por exemplo, o que normalmente se adota como fator de avaliação é o tempo de execução, ou tempo de resposta, ou ainda a vazão (quantidade de trabalho executada por unidade de tempo) [19, 23, 24, 25, 26].

É necessário, então, que se estabeleçam métricas adequadas à comparação entre as ferramentas, além de um programa padrão de testes que procure demonstrar que uma ou mais ferramentas destacam-se em relação às demais. Para isso é fundamental que se considere alguns critérios que permitam uma avaliação segura, dentre eles [14]:

a) Validade, Significação:
   Deve permitir efetuar medidas corretas de forma correta, e que mostrem ao final um significado concreto;

b) Consistência, Precisão, Confiabilidade:
   Os dados auferidos devem descrever realmente o que se está medindo;

c) Claro Entendimento:
   Deve ser simples, de modo a facilitar o entendimento por quem tiver interesse;

d) Compreensão:
   Que devem capturar o que é realmente importante aos objetivos desejados;

e) Não Redundância:
   Que não se deve gastar tempo com medições que tornem aos mesmos resultados.

Cabe aqui uma diferenciação do que seja medida, medição e métrica. Embora semelhantes, esses termos tem significados um tanto diferentes. De um modo geral: medida é simplesmente um padrão utilizado para se efetuar uma

avaliação; medição é o ato de quantificar um produto ou processo utilizando algum padrão; e métrica é uma medida capaz de permitir a comparação, por exemplo, de duas ferramentas de *software* [15].

Vários são os instrumentos que podem ser utilizados para efetuar medições em engenharia de *software*, dentre os quais os seguintes podem ser citados: observação, reconhecimento cognitivo, estimação, monitoração, testes, auditagem, *log*, *checklist*, experimentos, entre outros [16].

Por outro lado, é importante observar que pode existir mais de uma forma de se efetuar medições [16]. Não se tem como especificar se as métricas adotadas para uma determinada avaliação são mais ou menos adequadas ou corretas do que outras, mas sim que devem levar em consideração a sua isenção e a sua relevância.

Existem basicamente 3 (três) tipos de métricas a considerar [17]:

a) Métricas de Processo:

   Levam em consideração as atividades durante a fase de desenvolvimento e utilização do *software*. Basicamente estão ligadas a estimativas, como por exemplo: tamanho, possíveis defeitos, utilização;

b) Métricas de Produto:

   Procuram medir a qualidade do produto resultante da fase de desenvolvimento. São exemplos: manutenibilidade, usabilidade, etc.;

c) Métricas de Recursos:

   Têm como objetivo medir as entidades requeridas para a execução do processo.

Recursos e produtos estão normalmente associados aos processos. Cada processo utiliza recursos de *hardware*, de *software*, assim como pessoas necessárias ao seu desenvolvimento. São exemplos: velocidade do *hardware* e qualidade do s*oftware* [17].

Para efeito de estabelecimento de métricas, os atributos de um software que se deseja medir podem ser classificados em internos e externos [17]. Atributos internos podem ser medidos com maior facilidade, pois em geral são concretos, como por exemplo: tamanho e custo. Já os atributos externos não são tão simples,

principalmente por dependerem muito do ambiente em que uma aplicação é executada, como por exemplo: usabilidade, testabilidade e eficiência.

O problema inicial que se tem ao estabelecer métricas é definir quais objetivos se deseja alcançar com base nas possíveis medições a serem efetuadas. Uma vez definidos os objetivos, deve-se determinar o que se deseja obter como respostas durante o processo de medição, e como obtê-las a partir dos dados disponíveis [17].

Para que as métricas possam ser aferidas é possível a utilização de alguma ferramenta automatizada. No entanto, um dos problemas que se pode ter em adotar ferramentas automatizadas de avaliação, é que muitas vezes elas são capazes de medir tudo, mas falham no que realmente é necessário [17]. O que acontece é que elas não possuem um objetivo específico que seja, efetivamente, do interesse de quem vai utilizar o programa. Assim, por exemplo, ao se efetuar um *benchmark* de diversas ferramentas de comparação de seqüências biológicas, digamos NCBI BLAST [34], FASTA [36] e WU-BLAST [35], existe um interesse quanto ao número de alinhamentos obtidos para um mesmo conjunto de seqüências de consulta e uma mesma base de dados biológica, em um determinado período de tempo, e não no número de operações de ponto flutuante executadas pelo processador naquele período.

Ferramentas que se destinam a efetuar medições serão mais bem sucedidas caso sejam projetadas tendo-se em mente os objetivos aos quais se destinam. Por exemplo, para o presente trabalho foi necessária a criação de uma pequena ferramenta que permitiu medir o número de leituras e gravações em disco, durante a avaliação das ferramentas BLAST em paralelo, utilizando, como fonte de dados, um dos arquivos virtuais existentes no diretório "/proc" do Linux.

Assim, com relação aos objetivos de análise e comparação de ferramentas de *software*, deve-se [17]:

a) Listar os objetivos que se deseja atingir no processo de avaliação;
b) Para cada um dos objetivos, levantar questões que devem ser respondidas para saber se um determinado objetivo foi alcançado;
c) Decidir o que deve ser medido, para que as perguntas formuladas tenham uma resposta conveniente.

Um cuidado que se deve ter, no entanto, é conhecer e aceitar os limites que são indiretamente impostos quando se procede a uma avaliação. É saber, por exemplo, que nem tudo que se deseja medir é exeqüível [14], ou que nem sempre se tem total controle sobre o ambiente em que uma avaliação está sendo efetuada. Por exemplo, quando se deseja executar uma ferramenta distribuída em um ambiente de *cluster* em modo exclusivo, é possível obter junto à equipe de suporte da instalação uma autorização para isso; Tal autorização, no entanto, não seria exeqüível em um ambiente de *grid*, já que tanto os equipamentos quanto os usuários estão geograficamente distribuídos, e não sabem da existência do outro.

Uma vez estabelecidas as métricas que se deseja obter, considerando os fatores citados anteriormente, além de outros elementos pertinentes à avaliação, elas devem ser então detalhadas. Para esse detalhamento é necessário que se façam constar algumas das informações que estão listadas abaixo [14]:

a) Dados técnicos que não permitam ambigüidades sobre as medidas a serem adotadas;
b) O motivo pelo qual se optou pela utilização de uma medida;
c) A unidade considerada como padrão para efetuar a medição (metros, horas);
d) Como efetuar a medição.

Por exemplo, é sabido que operações de entrada e saída possuem um custo bastante elevado, principalmente considerando aplicações que demandam muitos acessos a disco. Dessa maneira a definição de rotinas, que consigam minimizar tais acessos, pode proporcionar uma economia considerável no tempo de execução para aquela aplicação, tornando mais eficiente uma determinada ferramenta em relação a alguma outra. Assim, ao analisar o desempenho de duas ou mais ferramentas, a medição do "número de acessos" efetuados por uma ferramenta pode ser determinante para se conhecer o "porquê" de uma ferramenta ser melhor do que alguma outra, quando efetuando um processo de comparação entre as mesmas. Adicionalmente é necessário que se tenha um ambiente computacional que proporcione a "igualdade de condições" de processamento, assim como a definição de um procedimento, como, por exemplo, a "utilização de uma ferramenta de software", que possibilite efetuar tais medições.

2.2.

Utilização no Contexto do Trabalho

Como foi citada anteriormente, a presente dissertação inclui duas avaliações de ferramentas BLAST voltadas ao processamento distribuído. A primeira tem por finalidade verificar a eficácia do balaBLAST em realizar o balanceamento de carga durante a sua execução, enquanto a segunda tem como propósito realizar um benchmark de ferramentas BLAST em paralelo, de modo a identificar aquela com melhor desempenho, incluindo o balaBLAST.

Reportando ao Capítulo 1 e à Seção 2.1 da presente dissertação, uma vez que existem dois objetivos perfeitamente definidos, é possível formular algumas questões a respeito dos fatores que podem ter alguma influencia sobre os mesmos. Por exemplo, no caso da verificação do balanceamento de carga, podem ser citados: o tipo do ambiente distribuído; o número de equipamentos utilizados; a capacidade de processamento dos equipamentos; a latência; o tamanho da base de dados biológica; entre outros. Assim, em relação ao tamanho da base de dados biológica, pode-se ser argüir, por exemplo, em quê o seu tamanho poderia influenciar no balanceamento de carga durante a execução do balaBLAST.

Uma vez estabelecidos os fatores citados anteriormente, é possível definir o que medir, a partir das respostas aos questionamentos efetuados, e como fazê-lo. Aproveitando o questionamento mostrado no exemplo anterior, uma das opções poderia ser a medição do número de seqüências processadas, assim como da quantidade de leituras em disco efetuadas, para cada um dos equipamentos pertencentes ao ambiente distribuído utilizado, considerando um conjunto de execuções do balaBLAST com bases de dados de diferentes tamanhos.

2.3

Conclusão

Neste capítulo o que se procurou apresentar, de um modo geral, foram os fatores e critérios que devem ser considerados ao avaliar e comparar ferramentas

de software. Foi mostrado que não basta somente verificar se o desempenho de uma ferramenta é melhor que o da outra, mas também considerar a qualidade das ferramentas avaliadas. A observação de fatores tais como: abrangência, funcionalidade, usabilidade, acurácia, facilidade e adequabilidade, além de critérios tais como: validade, significação, consistência, precisão, confiabilidade, claro entendimento, compreensão e não redundância, além do desempenho, constitui a chave para que seja efetuada uma avaliação robusta de uma ferramenta de software.

Os fatores e critérios aqui apresentados, também considerados no roteiro para avaliação e comparação de ferramentas BLAST em paralelo constante do Anexo 3, serviram de aporte às avaliações descritas nos Capítulos 4 e 5, mas principalmente à avaliação de desempenho e comparação de ferramentas BLAST, apresentada no Capítulo 5, no qual se expõe, de maneira prática, a aplicabilidade de alguns conceitos aqui mostrados.