

6 Conclusões

Infra-estruturas de *software* concebidas com o intuito de dar apoio à implementação e integração de aplicações científicas em ambientes distribuídos e heterogêneos assumem cada vez um grau maior de importância devido, principalmente, ao surgimento de novas tecnologias – por exemplo, redes de comunicação de alta velocidade e estações de trabalho com grande poder de processamento – que têm servido como base de desenvolvimento para estas aplicações distribuídas.

O escopo de execução das aplicações em ambientes distribuídas e do gerenciamento de usuários e de recursos computacionais, tais como computadores, arquivos de dados e aplicações, tem ultrapassado o ambiente local, cruzando fronteiras administrativas e tecnológicas, provocando uma demanda nas características de interoperabilidade e portabilidade.

A heterogeneidade existente de equipamentos, de sistemas operacionais e de ambientes de desenvolvimento integrado (IDE - *Integrated Development Environment*), tem levado ao surgimento de especificações abertas, devido à necessidade de padronização desses sistemas. Essa mesma heterogeneidade, também leva à necessidade de fornecer às aplicações uma série de mecanismos de gerência de recursos e de transparências de distribuição.

Aliado a esse fato, a sofisticação das novas aplicações distribuídas sugere o alto grau de complexidade exigido dos serviços oferecidos e as dificuldades associados ao projeto dessas aplicações. Grande parte dessas dificuldades consiste em oferecer serviços capazes de manter as características necessárias a diversidade de tipos de dados e fluxos.

A análise efetuada na infra-estrutura de *software* para aplicações distribuídas de alto desempenho CSBase, demonstrou sua utilização com razoável sucesso no sentido de permitir que desenvolvedores criem rapidamente aplicações reunindo um conjunto particular de serviços, com uma quantidade aceitável de esforço e tempo. Entretanto, os níveis de desempenho e as características de gerência de recursos da implementação atual deste sistema ainda não são suficientes para as aplicações que hoje despontam, considerando-se que ele não consegue efetuar reserva dos recursos

necessários às aplicações. Tampouco se tem mostrado suficiente em abstrair dos desenvolvedores os efeitos de mudanças na utilização desses recursos, que ocorrem paralelamente ao ciclo de vida de uma aplicação, isto é, durante a distribuição ou em tempo de execução, devido, principalmente:

- A necessidade de utilização de recursos não ser conhecida antecipadamente pela aplicação, até que seja executada e avaliada;
- A prioridade com que os clientes solicitam ou os servidores tratam um pedido pode variar de acordo com as condições estáticas e dinâmicas do *hardware* e do *software*, tais como necessidade e disponibilidade de processador, disponível apenas em tempo de execução;
- A inexistência de um mecanismo que forneça alguma qualidade de serviço à aplicação e, ao mesmo tempo, abstraia desta a plataforma de *hardware* e *software* na qual será executada.

Para aplicações que não toleram interrupções no seu funcionamento – com a realização de reconfigurações seguidas por um reinício – a adaptação da aplicação ao novo ambiente, dinamicamente, mostra-se como a única alternativa viável. Neste contexto, o suporte oferecido pela proposta atual do *framework* CSBase ainda é insipiente. A pesquisa demonstrou que o *framework* CSBase é apropriado para requisições de aplicações de alto desempenho. Entretanto, sem a extensão proposta, o *framework* CSBase pode incorrer na má utilização dos recursos computacionais disponíveis nos aglomerados.

Várias abordagens têm sido propostas para preencher esta lacuna, que não é apenas do *framework* CSBase, mas de diversos outros sistemas com objetivo semelhante [14] [21] [22]. Em comum, a maioria dessas abordagens emprega técnicas de reflexão computacional, que mantêm uma representação interna da configuração do sistema, a qual pode ser manipulada de forma que mudanças na representação são refletidas no sistema real. Embora o emprego abrangente de reflexão computacional (de forma a permitir a adaptação de aspectos arbitrários da plataforma) exija uma reestruturação no projeto da infraestrutura de *software*, a adoção de elementos isolados (*ad-hoc*) com características reflexivas em tecnologias atualmente disponíveis aponta a

abordagem geral de reflexão como um caminho concreto para a próxima geração dessas plataformas [11] [26].

Este trabalho estende o *framework* CSBase oferecendo um novo serviço de gerenciamento de recursos distribuídos, baseado em objetos distribuídos, possibilitando a execução remota de componentes de *software* em uma coleção de computadores conectados através de aglomerados de computadores, dando suporte para o gerenciamento e a reserva de processador no sistema distribuído. A escolha de onde os componentes são executados é feita de forma implícita ou explícita pelo usuário, e leva em consideração características estáticas e dinâmicas do *hardware* e do *software* disponível no sistema distribuído.

Apresentou-se uma descrição detalhada de uma solução escalável para gerenciamento de recursos distribuídos em aglomerados de computadores. A solução proposta (CSBase estendido) implementa um mecanismo para aglomerados de computadores, integrando uma coleção heterogênea de recursos distribuídos, oferecendo suporte para gerenciamento de recursos. Ao contrário dessa solução, o mecanismo sem estas funcionalidades não oferece suporte para reserva de recursos.

Mostrou-se a implementação da solução, oferecendo suporte para reserva de recursos e escalonamento de tempo flexível (*soft real-time*). Os resultados experimentais demonstram que o serviço implementado pode vir a preencher o espaço deixado pelas deficiências verificadas na implementação da infraestrutura de *software* para aplicações de alto desempenho. Como resultado da extensão, obteve-se a integração entre a monitoração de recursos distribuídos, a iniciação remota de aplicações, e um mecanismo de reserva de processador que proporcionou uma melhor utilização das máquinas disponíveis.

Uma limitação no serviço proposto é que, a adoção de um mecanismo executável no mesmo nível da aplicação (*application-level*) implica em uma falta de granularidade mais fina na garantia de tempo de processador, que só é possível se este mecanismo for implementado no nível do sistema operacional (*kernel-level*).

Apesar dos resultados obtidos pelos gráficos apresentados terem sido satisfatórios, uma proposta de trabalho futuro é uma avaliação mais detalhada nos componentes internos do sistema, como a eficiência proporcionada pelos SSI's. Essa necessidade é bastante acentuada pelo fato do mecanismo manter informações aproximadas sobre o estado global dos recursos dos aglomerados, o que, provavelmente, deve provocar alguma sobrecarga no desempenho da rede, dos aglomerados, e do sistema como um todo.

Outro fato que merece uma avaliação futura é que usuários leigos, provavelmente não saberiam – nem desejariam – fornecer informações de reserva de recursos. Nesse caso o próprio o administrador do sistema poderia fornecer uma especificação dos requisitos da aplicação, e armazená-la no Repositório de Algoritmos junto com a própria aplicação. Essa reserva poderia agir tanto no sentido de privilegiar algumas aplicações, como de limitar o uso de recursos em outras. Como limitador de recursos, o mecanismo poderia ser estendido para aceitar limites explícitos de utilização dos recursos, flexibilizando o uso dos recursos mais importantes dos aglomerados.

Outra proposta para trabalho futuro é avaliar a extensão do sistema em outras plataformas – Linux por exemplo. A proposta da API de reserva de recursos é ser o mais portátil possível, contudo por questões operacionais não foi possível o teste em plataformas diferentes da apresentada.

Deverá ainda ser avaliada no futuro, a possibilidade do mecanismo proposto extrapolar as fronteiras dos aglomerados locais, atuando como um mecanismo de reserva de recursos em redes geograficamente distribuídas.

Por fim, seria interessante que, à partir da extensão objeto desta dissertação, fosse efetuada uma seleção automática da máquina (SSI) que executará a aplicação, permitindo um balanceando de carga de forma transparente aos usuários, possibilitando a integração do *framework* CSBase e de vários aglomerados locais heterogêneos.