

MDS: MÓDULO DE DISTRIBUIÇÃO DE SIMULAÇÕES. APLICAÇÕES COM O PROGRAMA LMPROT

Fábio Pinho Moreira¹, Francis Weslei dos Reis Pereira², Paulo Sérgio Pereira³, Rodrigo dos Santos Camargo⁴ e Roosevelt Alves da Silva⁵

¹⁻⁵ Universidade do Vale do Paraíba - Faculdade de Ciência da Computação

Av. Shishima Hifumi, 2911 CEP 12244-000 São José dos Campos - SP – Brasil

¹fapimo@gmail.com, ²franciswesley@gmail.com, ³psergio.p@terra.com.br, ⁴rodbrsjc@gmail.com,

⁵roossilva@univap.br

Resumo - O presente trabalho reporta um módulo desenvolvido para distribuir e gerenciar de forma sincronizada a execução e manipulação de entradas/saídas de programas de simulações via internet. Inicialmente desenhado para utilização em massa e de forma colaborativa. Em particular, o módulo foi aplicado para a execução do programa de simulação molecular LMProt, cujos resultados permitiram conferir sua eficácia na distribuição simultânea das simulações (independentes) a diversos núcleos de processamento (CPU's). O sistema foi desenvolvido em Java, utilizando a ferramenta Eclipse, e testado usando a plataforma Linux.

Palavras-chave: Sistemas Distribuídos, Protein Folding, *Docking*, LMProt.

Área do Conhecimento: Ciência da Computação e Biologia.

Introdução

A possibilidade de se empregar os recursos oferecidos pela internet em aplicações envolvendo a distribuição de tarefas em núcleos de processamento tem sido destacada atualmente pelo alto poder de processamento que se pode alcançar. Existe um elevado número de CPU's que muitas vezes estão ociosas e podem ser acessadas diretamente pela rede, formando uma excelente ferramenta para processamento paralelo e/ou distribuído. As vantagens deste tipo de aplicação tornam-se relevantes principalmente quando as tarefas desejadas podem ser realizadas de forma independente. Sistemas desta natureza são denominados de Sistemas Distribuídos (SD), que consiste basicamente em usufruir, de forma independente, o uso de uma coleção de computadores autônomos interligados através de uma rede de comunicação e gerenciados através de softwares capazes de compartilhar os recursos do sistema (hardware, software e dados).

Dentre as vantagens que tornam este tipo de aplicação bastante atrativa pode-se destacar: a) seu aspecto incremental, isto é, novos computadores e linhas de comunicação podem ser consecutivamente acrescentadas ao sistema e; b) tolerâncias às falhas são tratadas através da replicação de processos em unidades de computação distintas.

Atualmente, projetos audaciosos envolvendo SD têm sido propostos para contribuir no desempenho de simulações moleculares envolvendo estudos de problemas biológicos. O tratamento e a predição de doenças e anomalias

são pretensões que estão por traz das simulações [1], visto que o conhecimento das bases moleculares envolvidas no organismo permite o desenho de novos fármacos e também a predição efeitos colaterais, minimizando os riscos ao ser humano. A importância de tais estudos chama atenção da população e leva a formação de inúmeros colaboradores aos programas por meio da concessão de suas CPU's para a execução das simulações via SD. Em particular, o projeto de sistema distribuído denominado *foldings@home*, tem permitido entender o enovelamento de diversas proteínas através de simulações via dinâmica molecular [2-4]. O projeto conta atualmente com mais de um milhão de CPU's de colaboradores associados por todo o mundo.

Neste trabalho, um Módulo de Distribuição de Simulações (MDS) foi desenvolvido visando aplicações similares às descritas acima. O módulo, no momento, tem sido preparado para simulações Monte Carlo com o programa LMProt [5] (*Local Moves for Proteins*), desenvolvido pelo Grupo de Computação Biomédica da UNIVAP. As simulações por este programa permitem abordar problemas relacionados ao enovelamento de proteínas e problemas de *docking* (interação entre moléculas). O uso de SD para execução de programas envolvendo estudos de *docking* tem sido também muito recentemente aplicado por Fernández-Recio *et al.* [6], cujos resultados permitiram reduzir o tempo efetivo das simulações de forma surpreendente.

Materiais e Métodos

Para o desenvolvimento do módulo MDS, a ferramenta Eclipse [7] foi empregada a fim de facilitar e aumentar a produtividade no processo de codificação do sistema na linguagem Java [8]. O módulo completo consiste de três tipos de módulo:

- Cliente
- Servidor
- Interface – Servidor

O módulo cliente é responsável por interagir com o programa LMProt [5] e comandá-lo. O módulo servidor é responsável por comandar e monitorar cada módulo cliente que está conectado. Através da interface - servidor, é possível definir quais tarefas cada cliente vai executar e monitorá-los. Após a realização da tarefa, cada cliente envia para o servidor o resultado com todas as informações necessárias para futuras análises.

Resultados e Discussão

Abaixo são apresentados os componentes principais da interface do MDS da versão atual:

1. Interface gráfica

- Tela principal do sistema; recursos disponíveis são apresentados de forma que haja uma manipulação fácil e rápida do sistema (Fig.1);

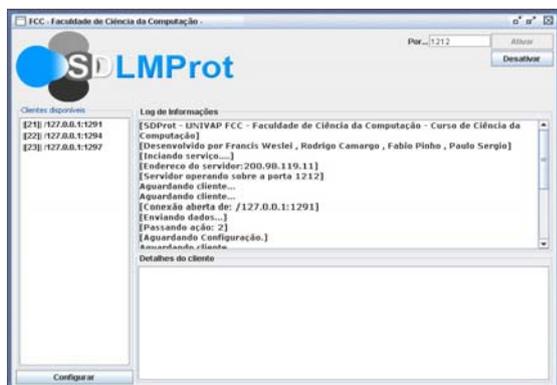


Fig.1 – Interface gráfica do Módulo de Distribuição de Simulações (MDS). O rótulo “SDLMProt” da interface denota a aplicação do atual Sistema Distribuído para aplicações com o programa LMProt.

2. Botões Ativar e Desativar (Fig.2)

- Localizados no topo a direita da Fig.1, os botões de ativar e desativar permitem que o usuário a qualquer momento inicie ou desative o funcionamento do sistema. Um campo editável ao lado dos botões informa a porta TCP que o servidor irá operar;



Fig.2 – Botões para ativar e desativar o sistema. O campo editável permite definir a porta TCP de operação do sistema.

3. Detalhes dos clientes (Fig.3)

- Localizado na base à direita da Fig. 1, campo não editável, mostra as informações que cada cliente possui entre elas o endereço e a porta que o cliente esta utilizando para se conectar ao servidor;

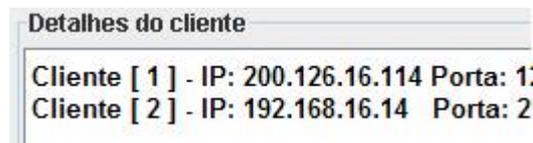


Fig.3 – Caixa de texto para exibir os detalhes de cada cliente.

4. Clientes disponíveis (Fig.4)

- Campo localizado à esquerda da Fig.1, exibe todos os clientes conectados que estão disponíveis para as simulações;

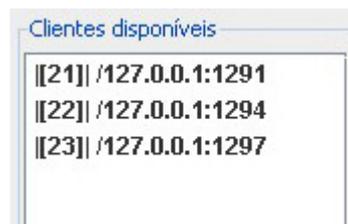


Fig.4 – Lista de clientes ativos no servidor, configurados de forma independente.

5. Configurar (Fig.5)

- Localizado na base à esquerda da Fig.1, o botão configurar é utilizado para enviar a configuração inicial dos testes de simulações feita nos clientes e executar a tarefa inicial do teste, isto é, execução do programa LMProt;



Fig.5 – Botão de envio de configurações ao(s) cliente(s) selecionado(s).

6. Log de informações (Fig.6)

- Localizado no centro da Fig.1, campo de texto não editável, apresenta várias informações do processo de funcionamento do sistema, entre elas, as relacionadas às conexões e desconexões de cada cliente, as de mudança de estado do servidor, e informações decorrentes de exceções causadas tanto pelo o servidor quando o cliente;

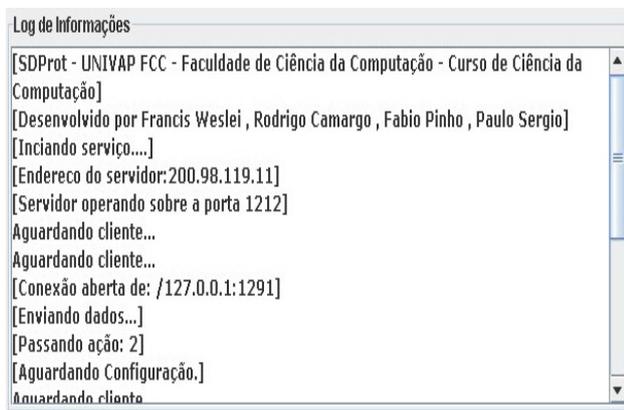


Fig.6 – Log de informações sobre todo o processo de distribuição do sistema; envolve informações do envio e recebimento de dados, da execução de tarefas e status dos processos entre os clientes e servidor.

A premissa para a construção do MDS foi principalmente oferecer uma plataforma capaz de ser empregada para a execução de outros processos além daqueles envolvendo o programa LMProt. Isto permitiu obter uma plataforma com característica de integração bastante flexível, ou seja, um código facilmente adaptável a outras necessidades independente do programa a ser distribuído. A forma independente a qual foi desenvolvido cada um dos componentes do MDS permite que cada cliente seja gerenciado/manipulado individualmente ou simultaneamente, facilitando inclusive a análise dos resultados que cada cliente obtém.

Testes preliminares com o MDS foram realizados usando um conjunto de 6 CPU's do Lab.6 da Faculdade de Ciências da Computação - UNIVAP, sendo o sistema LINUX Fedora Core 5 a plataforma empregada para oferecer maior performance na execução do programa LMProt.

Conforme os testes realizados, os clientes foram capazes de realizar todas as tarefas submetidas pelo MDS, tarefas que vão desde o recebimento de informações e execução do algoritmo LMProt a obtenção dos resultados e o envio dos mesmos ao servidor. A interação do MDS com o programa LMProt ocorreu sem falhas, sendo todos os requisitos solicitados pelo programa atendido de forma plena para todos os clientes.

Conclusão

Todos os pré-requisitos solicitados para execução do programa LMProt de maneira distribuída, foram satisfeitos. A distribuição de tarefas, execução do LMProt com tarefas diferentes, controle, monitoração e coleta de resultados foram executadas com sucesso. O MDS operou sob a plataforma Linux de maneira eficiente, obtendo sincronização e desempenho nos processos executados entre os clientes (CPU's) e o servidor.

Pretensões Futuras

Devido à arquitetura robusta sob a qual o módulo foi construído (adaptável), pretende-se que a interface gráfica do mesmo seja manipulada de forma interativa a fim de permitir configurá-la com especificidades que cada programa desejado para distribuição requeira. Além disso, o módulo deve se comunicar ou ser chamado por outros módulos que irão compor o Sistema de Simulação Molecular (SSM) em desenvolvimento pelo Grupo de Computação Biomédica da UNIVAP. Um deles é o módulo de interface gráfica (MIG) que recentemente foi desenvolvido (trabalho apresentado no INIC), cuja principal função é manipular os parâmetros de entrada do programa LMProt, automaticamente, de modo a gerar arquivos de entrada distintos e independentes para a distribuição do MDS, e fornecer todos gráficos envolvendo as análises dos arquivos de saída em tempo real.

Referências

- [1] Projetos que fazem uso de sistemas distribuídos para aplicações na biologia computacional. Disponível em: <http://www.distributedcomputing.info/ap-sciences.html>. Acesso em: 12.agosto.2006.
- [2] Snow C. and Pande V. S. *Kinetic Definition of Protein Folding Transition State Ensembles and Reaction Coordinates*. *Biophysical Journal* **91**: 14-24 (2006).

- [3] Kasson PM et al. *Ensemble molecular dynamics yields submillisecond kinetics and intermediates of membrane fusion*. PNAS **103** (32): 11916-11921 (2006).
- [4] Jayachandran G., Vishal V., and Pande V. S. *Folding Simulations of the Villin Headpiece in All-Atom Detail*. Phys. Chem. B **108** (33): 12267-2270 (2004).
- [5] daSilva RA, Degrève L and Caliri A. *LMProt: An Efficient Algorithm for Monte Carlo Sampling of Protein Conformational Space*. Biophysical Journal **87**:1567-1577 (2004).
- [6] Fernández-Recio J, Totrov M and Abagyan R. *Identification of Protein–Protein Interaction Sites from Docking Energy Landscapes*. J. Mol. Biol. **335**: 843–865 (2004).
- [7] Eclipse, IDE aberta para a construção de programas de computador. Disponível em: <http://www.eclipse.org>. Acesso em: 12.agosto.2006.
- [8] Java 2 – Fundamentos, Swing e JDBC. Ivan Mecenas, Editora Alta Books.