

INTERFACE GRÁFICA, PARTICIONAMENTO DA APLICAÇÃO E MECANISMO GERADOR DE ARCABOUÇOS PARA A CMB-SIMULATION. Geraldo Francisco Donegá Zafalon, Renata Spolon Lobato. – Ciência da Computação – Bacharelado em Ciência da Computação – Departamento de Ciências de Computação e Estatística – Instituto de Biociências, Letras e Ciências Exatas – Campus de São José do Rio Preto.

O advento da computação tem mostrado que é possível a realização de tarefas antes inimagináveis. Uma das áreas que mais cresce dentro da computação é a do processamento paralelo e distribuído, que se mostra bastante útil e com grandes vantagens em relação a computação tradicional, dita sequencial, visto o ganho de desempenho que se pode obter com o paralelismo.

A demanda por tarefas ou aplicações, cada vez mais complexas, provoca uma busca por um sistema computacional com um desempenho ainda melhor. A análise de desempenho objetiva medir o grau de eficiência de um sistema computacional, para que se saiba se esse sistema será eficiente para a determinada aplicação.

Como formas de medir o desempenho de um determinado sistema, as técnicas de aferição (*benchmarks*, coleta de dados e protótipos) e as técnicas de modelagem (solução analítica do modelo ou através de simulação) são utilizadas. Devido a complexidade dos modelos propostos para os sistemas computacionais mais complexos, a análise por simulação sequencial se torna inviável, dando lugar a simulação paralela.

A concepção de grandes sistemas, em geral, requer grandes montantes de dinheiro e necessita-se, a partir disso, possuir a garantia de que o sistema irá funcionar de forma ideal. Com isso, aparece a simulação de sistemas, como uma alternativa confiável e barata em relação à construção de um sistema real. A facilidade de se realizar uma simulação é representada pelo fato de não se necessitar do sistema pronto, nem mesmo um protótipo. Porém, muitas vezes é bastante complexo, principalmente para quem não possui conhecimentos em simulação, conceber modelos que garantam fidelidade em relação ao caso real. Porém, a complexidade pode aumentar ainda mais, quando se fala do paradigma de simulação distribuída.

Na simulação sequencial [Soa90], quem assume um papel bastante importante é a lista de eventos futuros [Mac87], que funciona como coordenadora da execução dos eventos. Para a simulação distribuída, essa lista de eventos necessitou sofrer mudanças, para que se pudesse garantir a consistência dos resultados em relação ao tempo. Para isso, houve o desenvolvimento de protocolos de sincronização entre os processos participantes da simulação, que são os protocolos conservativo e otimista.

Nos protocolos conservativos, os eventos são executados segundo a ordem que eles ocorrem na simulação [Mis86], com isso o sistema não permite correr riscos. Contrariamente, os protocolos otimistas [Car00, Fuj00, Jef85] permitem a execução até que um erro de causa e evento ocorra. Mas para isso os protocolos otimistas devem implementar um mecanismo que faça o sistema regredir até um estado seguro (*rollback*).

O desenvolvimento deste trabalho abordou a revisão bibliográfica sobre simulação e simulação distribuída, um estudo detalhado sobre a CMB-Simulation e de *frameworks* para interfaces gráficas em Java. As bibliotecas da UNESP, além do material disponível nos periódicos *on-line* da Capes (cujo acesso poderá ser efetuado de qualquer computador dos laboratórios do Departamento de Ciências de Computação e Estatística do IBILCE/UNESP), supriram as necessidades da revisão bibliográfica.

Efetuada a revisão e os estudos sobre programação, foi feita a especificação da interface gráfica. Durante essa fase, foram utilizadas metodologias de desenvolvimento de interfaces gráficas [Dix04].

A interface gráfica, como pode ser vista na figura 1, foi concebida em linguagem Java, pois deve-se ao fato de que Java é uma linguagem que executa em qualquer plataforma de *software*. Também foi realizada a modelagem utilizando os diagramas de casos de uso da UML (*Unified Modeling Language*), como pode ser visto na figura 2, antes da concepção da interface, para que se pudesse obter uma visão panorâmica de como ficaria o projeto da interface.

Através dessa interface, o modelador ou usuário pode especificar o modelo do sistema a ser estudado, compondo e parametrizando a rede de filas correspondente. Essa interface também sugere ao modelador qual o particionamento da rede de filas em processos lógicos deve ser adotado. Esses processos lógicos serão utilizados na simulação distribuída conservativa.



Figura 1. Tela principal da interface gráfica.

Todo um estudo foi realizado para se compreender formas eficientes de se particionar o modelo. Os testes de particionamento usaram os modelos de filas apresentados por Sacchi [Sac05, Uls99, Mor00] e também outros modelos da literatura de simulação distribuída e avaliação de desempenho.

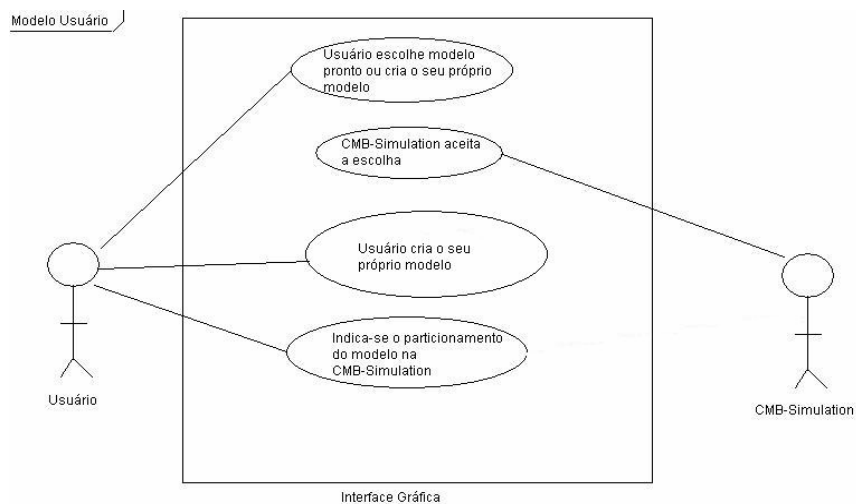


Figura 2. Caso de Uso da UML para modelagem da interface gráfica.

Além do resultado mostrado na figura 1, existem os resultados das funcionalidades da interface gráfica. Na figura 3, por exemplo, é mostrada a funcionalidade do gerador de aplicação utilizado para um dos modelos de filas prontos da CMB-Simulation, para possível alteração dos dados do modelo pronto.

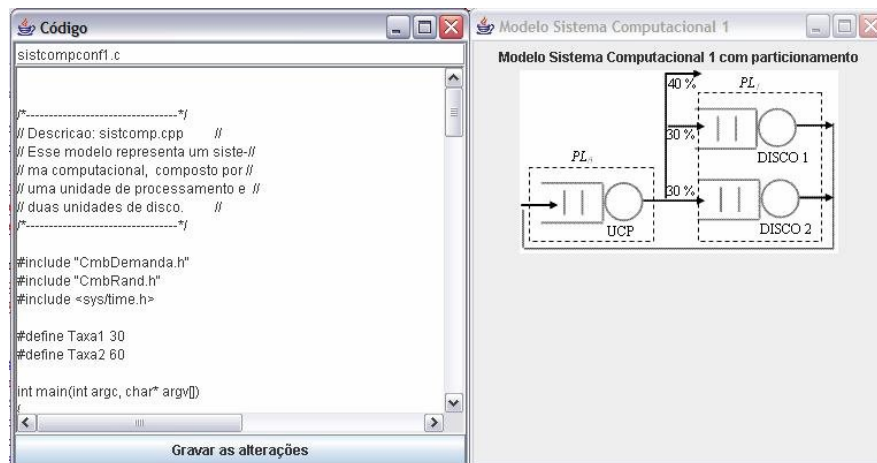


Figura 3. Funcionalidade da CMB-Simulation para um modelo pronto.

Além disso, o usuário pode criar o seu próprio modelo e particioná-lo. Na figura 4, pode ser vista a tela para a criação do modelo do usuário e obtendo como resultado, um arcabouço de código em linguagem C, contendo os principais comandos para se executar um programa de simulação na CMB-Simulation. Nessa tela, o usuário pressiona os botões (fila/centro de serviço) e uma outra tela (figura 5) é chamada, onde o usuário indica a disciplina da fila, do centro de serviço e em qual outro elemento ele se ligará.

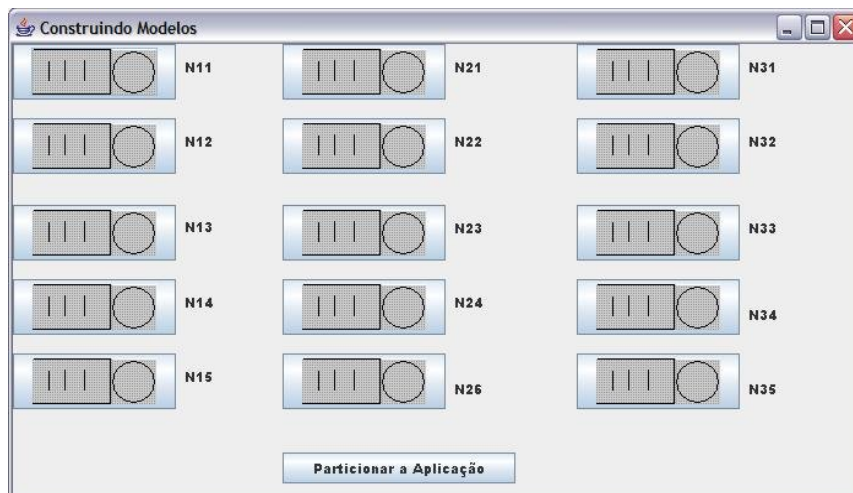


Figura 4. Funcionalidade para a criação do modelo do usuário.

O desenvolvimento dessa interface também contribuiu para a obtenção dos resultados desejados para o trabalho encontrado em [Lob06].

Os resultados obtidos permitem concluir que a extensão da CMB-Simulation, no que tange a concepção da sua interface gráfica e o mecanismo gerador de arcabouços, facilita, e muito, o usuário que deseja criar um programa de simulação distribuída, mesmo que esse usuário possua apenas conhecimentos básicos sobre o assunto.

Assim, o uso da ferramenta, com a sua interface gráfica, é de grande valia para o aprendizado de simulação paralela e distribuída.

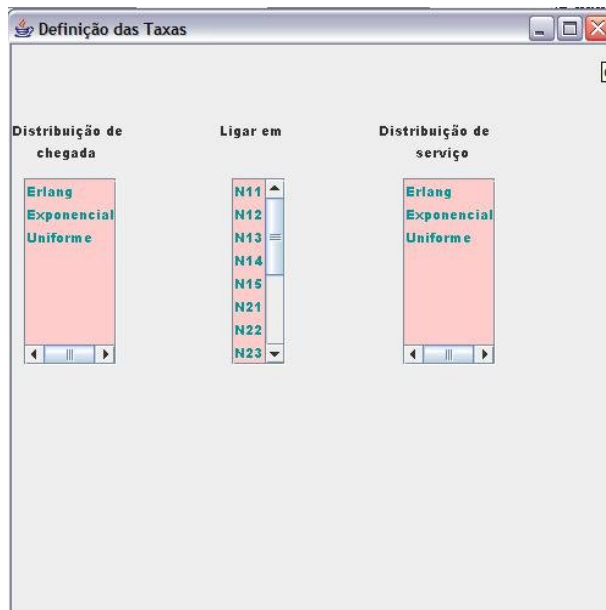


Figura 5. Escolha das disciplinas

Referências Bibliográficas

- [Car00] C. D. Carothers, D. Bauer, S. Pearce. ROSS: A High-Performance, Low Memory, Modular Time Warp System. *Proceedings of The 14th Workshop on Parallel and Distributed Simulation (PADS'2000)*. Pag. 53-60. 2000.
- [Dix04] A. Dix, J. Finlay, G. D. Abowd, R. Beale. Human-Computer Interaction. Pearson Education Limited. Terceira Edição. 2004.
- [Fuj00] R. M. Fujimoto. Parallel and Distributed Simulation Systems. John Wiley & Sons, Inc. 2000.
- [Jef85] D. R. Jefferson. Virtual Time. *IEEE Transactions on Programming Languages and Systems*. Num. 3. Vol. 7. Pag. 404-425. 1985.
- [Lob06] R. S. Lobato, G. F. D. Zafalon, R. S. Ulson, M. O. S. Balieiro. Abordagem Conservativa para Simulação Distribuída de Modelos de Redes de Fila. Anais da XXXII Conferência Latinoamericana de Informática, Santiago-Chile, 2006.
- [Mac87] M. H. MacDougall. Simulating Computer Systems. MIT Press. 19876
- [Mis86] J. Misra. Distributed Discrete-Event Simulation. *ACM Computing Surveys*. Num. 1. Vol. 18. Pág. 39-65. 1986.
- [Mor00] J. C. M. J. Morselli. Mecanismo para troca dinâmica de protocolos de sincronização. Instituto de Física de São Carlos – USP. 2000.
- [Sac05] R. P. S. Sacchi. ETW: Um Núcleo Básico de Simulação Distribuída Otimista. Departamento de Computação e Estatística, Universidade Federal de Mato Grosso do Sul. 2005
- [Soa90] L. F. G. Soares. Modelagem e Simulação Discreta de Sistemas. VII Escola de Computação - São Paulo. 1990.
- [Uls99] R. S. Ulson. Simulação Distribuída em Plataformas de Portabilidade: Viabilidade de Uso e Comportamento do Protocolo CMB. Instituto de Física de São Carlos, Universidade de São Paulo. 1999.