Escalonamento de Processos Sequenciais e Paralelos em um Cluster Dedicado a Simulações Biológicas

V. da Fonseca Vieira, R. Weber dos Santos
Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora
vfvieira@gmail.com, rodrigo.weber@ufjf.edu.br

Resumo

Neste trabalho foram estudadas diferentes políticas de escalonamento de processos em um pequeno cluster de computadores dedicado a simulações de modelos biológicos. Para isto o perfil típico de carga de trabalho neste cluster dedicado foi reproduzido artificialmente, o qual leva em conta três diferentes tipos de processos: sequenciais leves, sequenciais pesados e processos paralelos pesados baseados na biblioteca MPI. O cluster de computadores utilizado é baseado em Linux e foi montado com o pacote NPACI Rocks. Para o escalonamento de processos foi utilizado o Sun Grid Engine (SGE), que acompanha o NPACI Rocks. O SGE oferece integração com o MPI e permite a criação de filas de processos com características distintas. Foi realizado um estudo comparativo entre o comportamento de diferentes políticas de escalonamento submetidas à carga de trabalho em questão. As métricas adotadas e os objetivos desejados foram os de redução do tempo médio de execução dos processos, aumento da taxa média de processos executados e redução do tempo ocioso dos processadores do cluster. Esta avaliação nos permitiu estabelecer uma forma eficiente para gerenciar os recursos computacionais deste cluster de computadores dedicado.

1. Introdução

A compreensão dos fenômenos biofísicos que caracterizam a funcionalidade do coração facilitam o desenvolvimento e testes de novas drogas, de novos equipamentos médicos e de novas técnicas de diagnóstico não invasivo para as diversas doenças cardíacas. Para isso, a modelagem computacional desses fenômenos tem se mostrado uma ferramenta bastante eficiente. No entanto a simulação destes modelos biológicos pode se mostrar bastante complexa e computacionalmente custosa, o que implica na necessidade de uso de uma grande quantidade de recursos computacionais.

O trabalho apresentado a seguir tem como objetivo auxiliar as atividades desenvolvidas no Projeto Parafisio [1], na área de Modelagem Computacional aplicada a eletrofisiologia cardíaca, em execução na Universidade Federal de Juiz de Fora.

Neste laboratório foi montado um pequeno cluster de computadores a fim de se criar uma boa infra-estrutura para realização das simulações cardíacas. Porém, para que os recursos do cluster sejam devidamente utilizados pelos seus diversos processos concorrentes é necessário que haja uma política clara e eficiente de escalonamento dos recursos entre os processos [5].

Neste trabalho apresentamos um estudo comparativo realizado com diversas políticas de escalonamento utilizando o escalonador de processos Sun Grid Engine [6]. O estudo foi feito com uma carga típica de simulações artificialmente gerada sobre o cluster em questão. O objetivo é estabelecer uma política de escalonamento que se adeque ao contexto típico de simulações do coração, reduzindo o tempo médio de execução dos processos, aumentando a taxa média de processos executados e reduzindo o tempo ocioso dos processadores do cluster para as simulações ocorridas no laboratório.

2. O Ambiente Distribuído Utilizado

O desenvolvimento e a execução das simulações cardíacas demandam um grande poder computacional. Por isso, para que essas simulações possam ser realizadas adequadamente foi necessário o uso de um pequeno cluster de computadores. O cluster utilizado em nosso laboratório dispõe de sete máquinas AMD Athlon 64, com 3GHz de clock e 2 Gb de memória principal. A conexão entre as máquinas é realizada através de um switch 3Com de 1 Gbps.

Para configuração e administração do cluster de computadores foi adotado o kit NPACI Rocks [2], baseado em Linux. O NPACI Rocks é de livre distribuição e consitui uma

coleção de softwares integrada à distribuição Red Hat. O Rocks nos permitiu utilizar uma arquitetura assimétrica de cluster. Assim, existe uma máquina central, denominada frontend e uma série de máquinas a ele ligadas através de um switch, denominadas nós. Para a comunicação entre os processos paralelos nas simulações foi utilizado o padrão Message Passing Interface (MPI) [3], que permite uma boa comunicação entre os processos espalhados pelos diversos nós do cluster.

Como escalonador de processos foi utilizado o Sun Grid Engine (SGE) [6], que acompanha o NPACI Rocks. O SGE é uma ferramenta para gerenciamento de recursos em ambientes UNIX de computação distribuída e se mostrou uma solução bastante atrativa pois, além de reunir diversas características de bons escalonadores de processos, tem a facilidade de ser um software livre. Entre essas características podemos citar um bom gerenciamento de recursos, uma boa integração com as bibliotecas de passagem de mensagens em aplicações paralelas MPI e MPICH, criação de múltiplas filas e uma interface amigável e de fácil utilização [6]. O SGE auxilia no gerenciamento da carga de trabalho sobre o cluster de computadores, administrando diferentes políticas de escalonamento a fim de maximizar o uso do cluster e aumentar a vazão de processos sobre o cluster.

3. Carga de Trabalho Artificial Simulada

As simulações biológicas realizadas em questão são programas desenvolvidados para o Sistema Operacional Linux e implementados na linguagem de programação C. Várias dessas simulações são implementadas em paralelo, utilizando o padrão Message Passing Interface (MPI) para passagem de mensagens e utilizando o MPICH além do PETSc (Portable, Extensible Toolkit for Scientific Computation). Maiores detalhes sobre as simulações são obtidas em [4].

Portanto, para a realização do estudo comparativo entre as diversas políticas de escalonamento no SGE foi simulada uma carga de trabalho artificial a qual visa reproduzir uma utilização real do cluster, com as diferentes simulações que normalmente são utilizadas.

3.1. Os tipos de Simulações Reproduzidos

Foram reproduzidos três tipos de simulação de modelos biológicos. O primeiro tipo (tipo 1) é formado por processos sequenciais. Cada execução desse tipo de simulação é feita em aproximadamente 90 segundos. O segundo tipo (tipo 2) é formado por processos sequenciais. Cada execução desse tipo de simulação é feita em aproximadamente 936 segundos. O terceiro tipo (tipo 3) é formado por processos paralelos. Cada execução desse tipo de simulação leva aproximadamente 2350 segundos, utilizando sete nós.

3.2. A Carga de Trabalho

A simulação da carga de trabalho foi feita através de um pequeno programa executado por 11000 segundos. Esse programa, durante sua execução, submete os diferentes tipos de processos ao SGE, em intervalos de tempos aleatórios obtidos através de uma distribuição gaussiana. A média e o desvio padrão para a distribuição em cada tipo de processo são descritos a seguir:

- Para o tipo 1 foi utilizada uma média de 120 segundos com desvio padrão de 5 segundos;
- Para o tipo 2 foi utilizada uma média de 936 segundos com desvio padrão de 30 segundos;
- Para o tipo 3 foi utilizada uma média de 2460 segundos com desvio padrão de 90 segundos;

Cada intervalo de tempo foi obtido à partir da seguinte fórmula [7]:

$$g = m + d \times (\sqrt{-2 \times log(u1)} \times sen(2 \times \pi \times u2))$$

onde g é o número aleatório com distribuição gaussiana a ser obtido, m é a média desejada, d é o desvio padrão desejado, u1 e u2 são números aleatórios distintos com distribuição uniforme. Para a simulação dos três cenários, os números aleatórios u1 e u2 foram obtidos à partir de uma mesma semente, garantindo que os números aleatórios com distribuição gaussiana obtidos fossem sempre os mesmos.

4. Políticas de Escalonamento Utilizadas

Foram produzidos três cenários, os quais implementam as três políticas de escalonamento estudadas, utilizando o escalonador de processos SGE. Esses cenários foram produzidos baseando-se na utilização de filas.

A configuração das filas foi feita utilizando o Qmon [6]. O Qmon é uma interface gráfica fornecida pelo SGE que permite a realização de uma grande parte das tarefas do SGE. O Qmon permite a submissão de tarefas de diferentes características, a monitoração das tarefas submetidas ao SGE e a administração de limites de recurso e tempo de processamento para diferentes usuários. Com o Qmon é possível também alterar as políticas de escalonamento em cada fila baseando-se em diversos atributos como disponibilidade de memória, arquitetura dos processadores e taxa média de processamento em seus nós [6].

Para a criação dos cenários a serem estudados foram criadas filas utilizando o Qmon. Cada uma dessas filas destina-se à execução de um tipo de simulação biológica (tipo1, tipo2 ou tipo 3). A cada fila corresponde um conjunto de nós disponíveis para a execução do tipo a ela associado.

O conjunto de nós disponível para cada fila em cada cenário é apresentado na figura 1 e descrito a seguir:

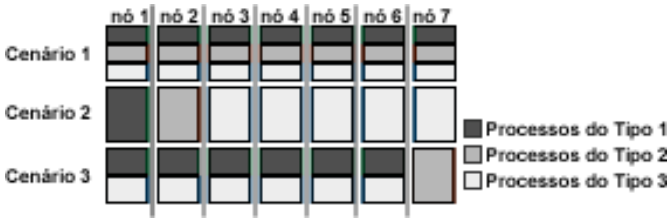

Figura 1. A divisao de nos para cada tipo de processo nos tres cenarios

- Cenário 1:
  Fila para processos do tipo 1: Nó 1 ao 7;
  Fila para processos do tipo 2: Nó 1 ao 7;
  Fila para processos do tipo 3: Nó 1 ao 7;

- Cenário 2:
  Fila para processos do tipo 1: Nó 1;
  Fila para processos do tipo 2: Nó 2;
  Fila para processos do tipo 3: Nó 3 ao 7;

- Cenário 3:
  Fila para processos do tipo 1: Nó 1 ao 6;
  Fila para processos do tipo 2: Nó 7;
  Fila para processos do tipo 3: Nó 1 ao 6;

Para cada uma das filas foi utilizado o algoritmo First-Come, First-Served, ou seja, priorizou-se a execução dos processos que solicitassem a CPU em ordem de chegada [9]. Assim não foi usada preempção e um processo que tivesse sua execução iniciada não poderia ser interrompido pelo escalonador. Como trata-se de um cluster de computadores foi utilizado o escalonamento global, ou seja, o escalonador tem uma visão global do cluster para a realização do escalonamento [8]. Nenhuma decisão de escalonameto foi atrelada ao código das aplicações. Essas decisões foram tomadas em tempo de execução e, por isso, o algoritmo utilizado pode ser considerado dinâmico [8]. Como não foi utilizado nenhum mecanismo de migração de processos entre os processadores, não foi necessária a realização de checkpointing para a execução das aplicações [6]. Cada fila está associada a um tipo de processo e a um número de processadores. Os processos de mesmo tipo, i.e. associados a uma determinada fila, não executam concorrentemente em um mesmo nó. Entretanto é possível que haja mais de uma execução de um tipo de simulação ocorrendo simultaneamente em uma mesma fila, desde que sejam realizadas em nós distintos. Dessa forma, em um nó, há concorrência entre processos apenas se eles forem de tipos diferentes.

5. Resultados

A carga de trabalho artificial simulada foi executada no cluster de computadores do Projeto Parafisio durante 11000 segundos. A distribuição gaussiana utilizada para cada tipo de simulação resultou na submissão de 93 instâncias de simulações do tipo 1, 12 instâncias de simulações do tipo 2 e 5 instâncias de simulações do tipo 3.

Dessa forma o tempo total de processamento de simulações do tipo 1 foi de 8370 segundos (93 instâncias X 90 segundos). O tempo total de processamento de simulações do tipo 2 foi de 11232 segundos (12 instâncias X 936 segundos). O tempo total de processamento de simulações do tipo 3 foi de 82250 segundos (5 instâncias X 2350 segundos X 7 nós). Assim, temos que o tempo total de processamento é de 101852 segundos (8370 + 11232 + 82250). Como o tempo de processamento foi de 77000 segundos (11000 segundos X 7 nós) a carga total submetida é de aproximadamente 132%, o que pode ser considerada uma carga alta. Para cada instância submetida ao SGE foi observado os tempos inicial e final de execução. Assim, pôde-se calcular o tempo de execução de cada processo. Após 11000 segundos de execução uma série de resultados foi obtida.

Tabela 1: Tempo médio de execução de cada tipo e número de instâncias completadas no cenário 1

Cenário 1			
	Tipo 1	Tipo 2	Tipo 3
Tempo médio	847	1050	12365
Instâncias completadas	90	10	0

Tabela 2: Tempo médio de execução de cada tipo e número de instâncias completadas no cenário 2

Cenário 2			
	Tipo 1	Tipo 2	Tipo 3
Tempo médio	105	1010	2450
Instâncias completadas	93	10	4

Tabela 3: Tempo médio de execução de cada tipo e número de instâncias completadas no cenário 3

Cenário 3			
	Tipo 1	Tipo 2	Tipo 3
Tempo médio	1020	936	6073
Instâncias completadas	61	12	1

As tabelas acima apresentam, para cada cenário, o tempo médio de execução de cada simulação em segundos e o número de instâncias com execução completada. No primeiro cenário o tempo médio de execução de processos do tipo 3 (12365 segundos) foi maior do que o tempo de simulação (11000 segundos), por isso nenhuma execução deste tipo de processos foi completada.

5.1. Analisando os dados obtidos

As tabelas comparativas a seguir permitem uma melhor análise dos dados obtidos com a carga de trabalho simulada:

Tabela 4: Comparação do tempo médio(em segundos) entre os três cenários

Tempo Médio de Execução das Simulações			
	Cenário 1	Cenário 2	Cenário 3
Tipo 1	847	105	1020
Tipo 2	1050	1010	936
Tipo 3	12365	2450	6073

A tabela anterior apresenta o tempo médio de execução de cada tipo de simulação em cada cenário em segundos. Nesta tabela observa-se que, para todos os tipos de simulações, o cenário 2, o qual permite que os processos sejam executados em nós distintos, apresenta os menores tempos médios de execução. Pode-se observar também que, para as filas com maior concorrência foram obtidos maiores tempos médios.

Tabela 5: Comparação da taxa de execução(em instâncias por 1000 segundos) entre os três cenários

Taxa de execução dos processos			
	Cenário 1	Cenário 2	Cenário 3
Tipo 1	8,1818	8,4545	5,5454
Tipo 2	0,9091	0,99	1,0909
Tipo 3	0,0808	0,4073	0,1645

A tabela anterior apresenta a taxa de execução de cada tipo de simulações em cada cenário, em número de instâncias executadas por 1000 segundos. Pode-se observar que, mesmo em cenários onde concorriam com simulações do tipo 3, as simulações do tipo 1 e as simulações do tipo 2 não tiveram grandes variações na taxa para os diferentes cenários. A simulação da carga artificial no cluster do Projeto Parafisio permite várias análises.

No cenários 1 e no cenário 3 pode-se observar que o tempo de execução das simulações do tipo 3 foi muito maior que no cenário 2. Isso ocorre devido à necessidade de sincronização das simulações paralelas. As aplicações do tipo 1 e do tipo 2 também apresentaram aumento no tempo médio de execução no cenário 1 e no cenário 3 porém a taxa de execução nestes cenários não sofreu grandes variações. Isso ocorreu devido à disponibilidade de nós para execuções simultâneas de simulações do tipo 1 e para simulações do tipo 2 (já que estas aplicações necessitam de apenas 1 nó para serem executadas).

Assim, pode-se dizer que, para uma carga de trabalho semelhante à encontrada no Projeto Parafisio, a política de escalonamento criada no cenário 2 é uma solução satisfatória para escalonamento de processos utilizando filas, baseando-se na separação de recursos por nós. Neste cenário o tempo médio aproxima-se do tempo médio de execução de cada simulação utilizando os 7 nós do cluster.

6. Conclusões

Este trabalho apresenta o resultado de testes realizados em um cluster de computadores dedicado a simulações de modelos biológicos. Esses testes foram realizados sobre o escalonador de processos SGE em um cluster composto por sete máquinas AMD Athlon 64 3000+, 2 Gb RAM.

Foi simulado no cluster uma carga de trabalho artificial que se aproxima à carga real típica do laboratório. Estabeleceu-se diferentes políticas de escalonamento entre os processos simulados, com o objetivo de encontrar uma política de escalonamento que permitisse a minimização do tempo médio de execução das aplicações e, também, a maximização da taxa de execução dessas aplicações no cluster.

Assim, foi simulada uma carga de trabalho composta de aplicações de três tipos: aplicações sequenciais leves, aplicações sequenciais pesadas e aplicações paralelas pesadas baseadas no padrão MPI. Após a simulação da carga de trabalho no cluster observou-se que, reservando-se cinco nós para as aplicações paralelas pesadas, um nó para a aplicação sequencial leve e um nó para a aplicação sequencial pesada foram obtidos resultados bastante satisfatórios, com tempos médios mais baixos e taxas de execução mais altos, em relação às outras políticas de escalonamento. Esses resultados permitiram a adoção desta política de escalonamento no cluster, fazendo com que o uso dos recursos disponíveis no laboratório seja mais eficiente.

Referências

[1] Fisiocomp, Laboratório de Fisiologia Computacional: http://fisiocomp.ufjf.br. último acesso em Junho de 2006.

[2] Rocks Clusters: http://www.rockscluster.org/Rocks. último acesso em Março de 2006.

[3] The MPI Standard: http://www.mpi-forum.org. último acesso em Maio de 2006.

[4] R. W. d. Santos, F. O. Campos, R. S. Oliveira. Performance comparsion of parallel geometric and algebraic multigrid preconditioners for the bidomain equations. *Lecture Notes in Computer Science*, 3991:76–83, 2006. Berlin-Heidelberg.

[5] J. Sloan. *High Performance Linux Clusters*. O'Reilly, 2005.

[6] SunMicrosystems. *N1 Grid Engine Administration Guide*, Maio 2005.

[7] A. R. P. Júnior, M. E. de A. Freitas. Geração de números aleatórios. http://www.cefetsp.br/sinergia/5p13c.html. último acesso em Maio de 2006.

[8] T. L. CASAVANT, J. G. KUHL. *A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems.* Fevereiro, 1988. IEEE Transactions on Software Engineering, p.141-154.

[9] A. SILBERCHATZ, P. B. GALVIN. *Operating Systems Concepts.* Addison-Wesley, 1998.