

Abordagens Paralelas para *Model Checking* de Redes de Autômatos Estocásticos*

Lucas G. Oleksinski, Fernando L. Dotti

Faculdade de Informática – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

Avenida Ipiranga, 6681, Prédio 32 – 90.619-900 – Porto Alegre – RS – Brasil

lucas.oleksinski@acad.pucrs.br, fernando.dotti@pucrs.br

Resumo. *Verificar modelos é uma técnica que permite garantir corretude da especificação de sistemas concorrentes em estágios iniciais de desenvolvimento. Entretanto, é custosa em termos de tempo e espaço requerendo algoritmos eficientes. Este artigo apresenta abordagens paralelas de verificação para modelos descritos em Redes de Autômatos Estocásticos e propriedades definidas na lógica temporal Computation Tree Logic.*

1. Introdução

A verificação de modelos é uma técnica de validação automática que permite a verificação de propriedades sob sistemas concorrentes de estados finitos [Clarke et. al. 1999] e provê informações sobre a evolução temporal dos mesmos para depuração.

As abstrações de uma realidade são definidas através de formalismos de modelagem e as propriedades à serem verificadas são descritas em lógicas temporais. Redes de Autômatos Estocásticos (*Stochastic Automata Networks* – SAN) é um formalismo Markoviano estruturado para modelagem de sistemas concorrentes de estados finitos primeiramente focado à avaliação de desempenho de sistemas distribuídos e protocolos de rede, como: escalonamento de processos em arquiteturas NUMA [Chanin et. Al. 2006], redes *wireless Ad Hoc* [Dotti et. Al. 2006], entre outros. Devido aos interessantes aspectos de modelagem que apresenta, esforços foram investidos na construção de um verificador de propriedades descritas na lógica temporal *Computation Tree Logic* (CTL) [Correa et. Al. 2012].

Entretanto, verificar um modelo é uma tarefa altamente custosa em termos de processamento e espaço, pois requer exaustiva avaliação da realidade em consideração [Clarke et. al. 1999]. Este trabalho propõe duas abordagens paralelas para verificação de propriedades CTL em ambientes com processamento e memória distribuídos.

2. Abordagens Paralelas para Verificação de Modelos SAN

Para implementação da ferramenta em questão, utilizou-se como base o algoritmo de satisfação de propriedades CTL descrito em [Baier and Katoen 2008]. Através da compilação do modelo, gera-se o Descritor Markoviano o qual representa implicitamente uma Cadeia de Markov, sendo utilizado como função de transição

*Projeto Financiado pelo CNPq e FAPERGS

[Clarke et. al. 1999] do sistema. O espaço de estados atingível do modelo é gerado através de um algoritmo de saturação [Sales and Plateau 2009] e é armazenado em uma estrutura de Diagramas de Decisão Multi-valorada (MDD), sendo eficiente no que tange à armazenamento e manipulação.

Considerando os altos custos envolvidos em termos de tempo para a técnica de verificação de modelos, duas abordagens paralelas foram desenvolvidas seguindo o padrão mestre/escravo. Para ambas as abordagens, réplicas do espaço de estados são feitas em conjunto com o Descritor Markoviano, habilitando as máquinas empregadas a calcular estados sucessores de um dado estado localmente, evitando demasiada troca de mensagens.

A primeira abordagem foca na distribuição das tarefas relacionadas à computação dos operadores da lógica temporal adotada. Como existem operadores que requerem cálculos de ponto fixo, rodadas de sincronismo é necessário, havendo pouca comunicação se comparado à cálculos de estados sucessores realizados sob espaços de estados distribuídos, os quais são altamente dependentes.

Sendo o tempo de verificação de fórmulas CTL linear ao tamanho da fórmula [Baier and Katoen 2008], a segunda abordagem realiza a escolha e distribuição de porções da árvore sintática que representa a propriedade para verificação em paralelo. Desta forma, grupos de máquinas são empregados na computação de diferentes partições da árvore sintática. Ainda, as tarefas relacionadas à verificação dos operadores de lógica temporal de maior custo computacional são divididas entre as máquinas que compõem cada grupo, executando-se os mesmos passos descritos para a primeira abordagem.

Os resultados preliminares mostram ganhos de desempenho próximos ao ideal para ambas as abordagens, uma vez que a computação dos operadores de maior custo é particionada e poucas rodadas de comunicação para sincronia entre as máquinas são necessárias. Mesmo as técnicas no trabalho descritas parecendo pouco eficientes no que tange à memória, é necessário considerar que estruturas intermediárias necessitam ser criadas na estrutura MDD para se atingir o resultado da verificação, sendo seu crescimento relacionado unicamente à computação dos operadores de lógica temporal, a qual é particionada e distribuída, tendo consumo reduzido em cada máquina empregada.

Referências

- Baier, C. and Katoen, J-P. (2008) *Principles of Model Checking*. MIT Press, Cambridge, Massachusetts.
- Clarke, E. M., Grumberg, O. and Peled, A. D. (1999) *Model Checking*. MIT Press, Cambridge, Massachusetts.
- Correa, C., Dotti, F., Fernandes, P., Maruani, E., Oleksinski, L. and Sales, A. (2012) “Um Verificador de Modelos Descritos em Redes de Autômatos Estocásticos”. In *XIII Workshop de Testes e Tolerância à Falhas (WTF)*, Ouro Preto - Minas Gerais, Brazil, pages 115-128.
- Sales, A. and Plateau, B. (2009) “*Reachable state space generation for structured models which use functional transitions*”. In 6th International Conference on the Quantitative Evaluation of Systems, pages 269-278.
- Chanin, R., Corrêa, M., Fernandes, P., Sales, A., Scheer, R. and Zorzo, A. F. (2006) “*Analytical Modeling for Operating System Schedulers on NUMA Systems*”. In *Electronic Notes in Theoretical Computer Science (ENTCS)*, pages 131-149.
- Dotti, F. L., Fernandes, P., Sales, A. and Santos, O. M. (2005) *Modular analytical performance models for ad hoc wireless networks*. In 3rd Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks. IEEE Computer Society, pages 164-173.