

Fast and Accurate Evaluation of Embedded Applications for Many-core Systems



Felipe da Rosa, Ricardo Reis, Luciano Ost
 {frdarosa, reis}@inf.ufrgs.br
 ost@lirmm.fr



Sistemas embarcados multiprocessados são apontados com a solução mais viável para abordar as emergentes restrições de design em custo, potência e performance [Borkar and Chien 2011]. Todavia, conceber estes sistema impõe novos desafios para engenheiros de software, compreendendo entre outros: (i) definição de protocolos de comunicação entre processos, (ii) analisar e portar sistemas operacionais, (iii) possibilidade de melhor explorar modelos de programação para tratar de questões para processamento paralelo. [Marongiu and Benini 2012], (iv) desenvolvimento de drivers [Gray and Audsley 2012], (v) traduzir aplicações entre sistemas multiprocessados.

Esta crescente complexidade de software faz com que a verificação funcional torne-se mais difícil, como resultado, engenheiros de software estão dedicando-se para escalar a performance. Tornando a simulação crítica durante o desenvolvimento de software, principalmente nas fases iniciais, durante a exploração do espaço de design.

Este trabalho objetiva começar a tratar os desafios impostos na construção de simuladores *JIT-Based (Just-In-Time)* adequados para estimativas de software, aprimorando a capacidade de engenheiros de software melhor explorar o espaço de design em estágio inicial de desenvolvimento de sistemas. Neste trabalho, será apresentado um modelo chamado *Watchdog* visando fornecer

estimativas de energia e tempo de execução em uma metodologia integrável em qualquer simulador baseado em *JIT*

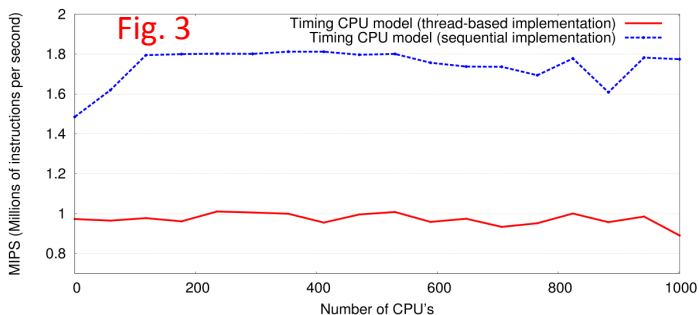
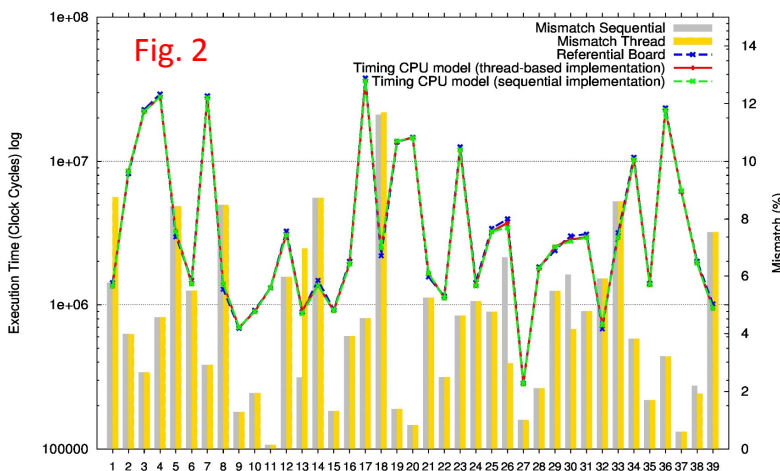
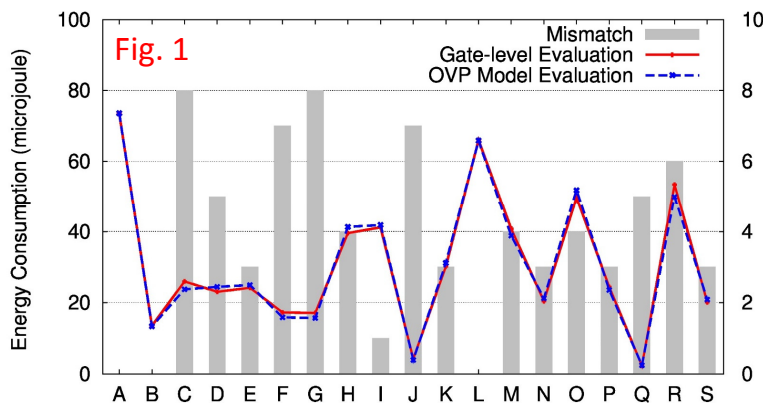
A metodologia proposta foca em um modelo baseado em eventos, observando as instruções individualmente executadas, simplificando a construção entorno do simulador. Adicionalmente, a abordagem proposta nesta tese é puramente *run-time*, isto significa que toda a computação necessária para implementar o modelo é realizada concomitantemente com a simulação, evitando enorme quantidade de uso de memória necessária para abordagens baseadas em *trace-driven* ao mesmo tempo que mantém a escalabilidade para sistemas *multi-processados*.

A fim de demonstrar a validade do modelo proposto, várias conjuntos de aplicações populares foram selecionados, dentre elas MiBench [Guthaus et al. 2001], Mälardalen WCET [Jan Gustafsson 2010], SPLASH-2 [Woo et al. 1995].

Os resultados mostram que a precisão do nosso modelo de tempo de execução varia de 0,6% a 11,5%, com 4,35% em de erro média. O modelo de energia atinge 0,01% a 8,6% de precisão dependendo do perfil de referência com um erro médio de 4,33%. Além disso, o modelo foi submetido a cenários de 1000 processadores mantendo um desempenho estável de 1,8 MIPS.

Figura 1 apresenta os 19 benchmarks usados (A até S) para validar a técnica, sendo o erro (*mismatch*) entre os dois representado na colunas em cinza, a linha vermelha o valor da avaliação de energia da fermenta de refecia e em pontilhado azul a estimativa gerada pelo modelo desenvolvido. A tabela 1 abaixo apresenta as aplicações utilizadas.

A - Bfsh	F - Crc	K - Fibonacci	P - Mdc
C - Binary search	G - Edn	L - Hanoi	Q - Peaksped
B - Bit Manipulation	H - Expint	M - Harm	R - UD
D - Bubble	I - Factorial	N - Insertsort	S - Usqrt
E - Counts	J - Fft	O - MatrixMult	



A figura 2, de forma semelhante à figura 1, é demonstrada a precisão do modelo de estimativa temporal utilizando-se 39 aplicações (1 até 39) e de forma semelhante a estimativa do modelo foi comparado com uma referencia. Neste caso foi utilizado uma placa de prototipação STM32F4DISCOVERY, contendo um Cortex-M4F. Para o modelo de estimativa temporal foram criados duas implementações visando aumentar o desempenho, uma utilizando pthreads e a outra apenas sequencial. Abaixo as aplicações selecionadas.

1. Adpcm	9. Counts	17. Fib	25. MatrixInver	33. Sha
2. Barnes	10. Crc	18. Fir	26. Mdc	34. Statistic
3. BasicMath	11. Dijkstra	19. Hanoi	27. Patricia	35. Stringsearch
4. Bfsh	12. Edn	20. Harm	28. Peaksped	36. Sw
5. BinarySearch	13. Expint	21. InsertSort	29. Petri	37. Tree
6. BitManipulation	14. Factorial	22. Jfdctint	30. Prime	38. Ud
7. Bubble	15. Fdct	23. Lms	31. Qsolver	39. Usqrt
8. Compress	16. Fft	24. Lu	32. Qsort	

Na ultima figura é apresentada velocidade de simulação para os dois modelos propostos em milhões de instruções por segundo, demonstrando a escalabilidade do modelo proposto para cenários com até 1000 processadores.

References:

- S. Borkar and A. A. Chien, "The future of microprocessors," *Commun. ACM*, vol. 54, no. 5, p. 67, May 2011.
- A. Marongiu and L. Benini, "An OpenMP Compiler for Efficient Use of Distributed Scratchpad Memory in MPSoCs," *IEEE Trans. Comput.*, vol. 61, no. 2, pp. 222–236, Feb. 2012.
- I. Gray and N. C. Audsley, "Challenges in software development for multicore System-on-Chip development," in *2012 23rd IEEE International Symposium on Rapid System Prototyping (RSP)*, 2012, pp. 115–121.
- B. Binkert et al., "The Gem5 Simulator," *SIGARCH Comput Arch. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.
- D. Sanchez and C. Kozyrakis, "ZSim: Fast and Accurate Microarchitectural Simulation of Thousand-core Systems," in *Proceedings of the 40th Annual International Symposium on Computer Architecture*, New York, NY, USA, 2013, pp. 475–486.
- M. R. Guthaus et al., "MiBench: A free, commercially representative embedded benchmark suite," in *2001 IEEE International Workshop on Workload Characterization*, 2001. WWC-4, 2001, pp. 3–14.
- A. B. Jan Gustafsson, "The Mälardalen WCET Benchmarks: Past, Present And Future," pp. 136–146, 2010.
- S. C. Woo et al., "The SPLASH-2 programs: characterization and methodological considerations," in *22nd Annual International Symposium on Computer Architecture*, 1995. Proceedings, 1995, pp. 24–36.