

## UMA PROPOSTA PARA AVALIAÇÃO DO EFEITOS DE DECISÕES DE PROJETO NA TÉCNICA DE REUSO DE TRAÇOS EM ARQUITETURAS ARM

GIOVANE DE OLIVEIRA TORRES<sup>1</sup>; LAÉRCIO LIMA PILLA<sup>2</sup>; MAURÍCIO LIMA PILLA<sup>3</sup>

<sup>1</sup>*Universidade Federal de Pelotas – gdotorres@inf.ufpel.edu.br*

<sup>2</sup>*Universidade Federal de Santa Catarina – laercio.pilla@ufsc.br*

<sup>3</sup>*Universidade Federal de Pelotas – pilla@inf.ufpel.edu.br*

### 1. INTRODUÇÃO

Atualmente, com as diversas evoluções tecnológicas ocorridas, existe uma grande difusão de portáteis que demandam melhor desempenho e redução no consumo de energia. Hoje, é possível encontrar arquiteturas que ocupam pouca área fisicamente, possuindo baixo consumo energético. Estas características citadas referem-se às arquiteturas ARM (*Advanced RISC Machine*), presentes em grande parte dos dispositivos portáteis atuais (KORNAU, 2010). Com isto, é importante que se busquem meios para melhorar este tipo de arquiteturas. Uma das possibilidades para isto está na técnica de reuso de traços (GONZALEZ; TUBELLA; MOLINA, 1999), que tem como objetivo reaproveitar sequências de instruções já executadas anteriormente. Com isso, espera-se reduzir o tempo de execução das aplicações, bem como economizar energia neste tipo de arquiteturas.

Arquiteturas ARM são processadores do tipo RISC (*Reduced Instruction Set Computer*). Uma das particularidades de destaque neste tipo de arquiteturas é a presença das instruções condicionais no conjunto de instruções. Todas as instruções são consideradas condicionais, o que indica que estas serão executadas ou não dependendo de uma determinada condição, a qual é indicada no *assembly* de ARM como um sufixo composto por duas letras na instrução. O processador executa ou não uma instrução condicional dependendo os quatro *bits* mais significativos do CPSR, os quais representam as quatro *flags* condicionais do registrador de *status*.

Existe um sufixo para determinar que uma instrução executará sempre independente das condições (sufixo *a1*) que pode ser omitido da instrução. 4 *bits* dos 32 de uma instrução são empregados para determinar sua condição, o que permite um total de 16 possíveis condições (KNAGGS; WELSH, 2004). Os benefícios destas instruções são os seguintes (ARM, 2011):

- Redução da quantidade de instruções de desvio, acarretando em melhorias na densidade do código;
- Instruções de desvio demandam mais ciclos de processador. Nos processadores ARM que não possuem *hardware* com predição de desvios, normalmente se gastam três ciclos de processador para preencher o *pipeline* do processador.

A técnica de reuso de traços está inserida dentro do contexto de reuso de valores – tendo como objetivo reexecutar sequências de instruções as quais se repetem durante a execução de uma aplicação.

Este mecanismo possui o seguinte funcionamento: primeiramente, um traço é construído na medida em que as instruções são executadas. Enquanto é formado um traço, seus contextos de entrada e saída (os registradores envolvidos

no traço com seus respectivos valores) são construídos observando os operandos das instruções que compõem o traço. Estes são armazenados em um *buffer*, a fim de permitir o posterior reuso. Quando a primeira instrução de um traço que já foi armazenado for preparada para ser executada, paralelamente o *hardware* responsável pelo reuso de traços inicia a comparação do contexto atual com o traço já armazenado. Se os contextos de entrada forem iguais, o traço poderá ser reutilizado, o que potencialmente evita a execução de instruções que compõem o traço. Para reusar um traço, basta utilizar o contexto de saída que está armazenado no *buffer*. Caso os contextos de saída forem diferentes, deve-se seguir o fluxo normal de execução de instruções. A Figura 1 mostra o exemplo de um mecanismo de reuso acoplado ao *pipeline* de uma arquitetura.

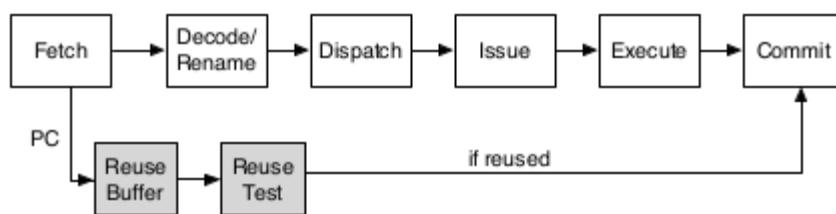


Figura 1. *Pipeline* com mecanismo de reuso (PILLA, 2007).

As principais vantagens que podem ser apontadas com a utilização de um mecanismo de reuso são as seguintes:

- Economia de recursos, dado que alguns passos do pipeline não são executados;
- Os resultados ficam disponíveis mais cedo;
- Redução nas dependências de dados, já que instruções dependentes podem ser executadas em paralelo.

Logo, o principal objetivo do trabalho é avaliar quais são os efeitos de uma implementação da técnica de reuso de traços em arquiteturas ARM. O que este trabalho difere de relacionados que estudaram técnicas de reuso GONZALEZ; TUBELLA; MOLINA, 1999; PILLA, 2004) é o estudo em ARM, já que esta apresenta um conjunto de instruções condicional.

## 2. METODOLOGIA

Para a realização deste trabalho, faz-se necessária a implementação do mecanismo de reuso de traços em arquiteturas ARM para sua avaliação. Assim, este trabalho visa dar continuidade a trabalhos anteriores que tiveram este mesmo objetivo.

A primeira etapa para a execução do trabalho proposto é avaliar a aplicabilidade de simuladores do estado da arte para efetuar este estudo. Isto deve-se ao fato de que os trabalhos anteriores (TORRES, 2015; MOURA, 2015) utilizaram-se da ferramenta Sim-Panalyzer (MUDGE, 2001), a qual não possui mais continuidade, além de possuir pouca documentação.

Além da pesquisa e análise de um novo simulador, propõe-se o estudo de *benchmarks* diferentes dos utilizados anteriormente para a realização deste projeto. Em trabalhos anteriores, o conjunto de *benchmarks* MiBench (GUTHAUS, 2001) foi utilizado para extração dos principais resultados. Neste caso, a exploração de aplicações diferentes para simulação tem como objetivo verificar se

os resultados encontrados anteriormente serão similares no que diz respeito a quantos traços são reusáveis, bem como a quantidade de instruções média por traço.

Escolhidas as ferramentas, o próximo passo é efetuar a implementação do mecanismo de reuso em um simulador de arquiteturas ARM. O diferencial de trabalhos anteriores está no fato de que esta implementação será feita em paralelo com a execução do simulador, o que indica que diversos fatores deverão ser considerados, o que incluem: (i) Tempo de acesso ao *buffer* de reuso, (ii) tempo de comparação entre os traços e (iii) acoplação do mecanismo com o *pipeline* do simulador.

Com a implementação consolidada e validada, o principal foco está em analisar maneiras para a técnica de reuso de traços obter ganhos em desempenho, além de ponderar questões referentes ao consumo de energia. Assim, esta etapa da metodologia tem como meta avaliar os ganhos e perdas que a implementação desta técnica irá obter.

### 3. RESULTADOS E DISCUSSÃO

Como trata-se de uma proposta, este artigo não apresenta resultados concretos. O que espera-se obter como resultado é uma avaliação sobre os efeitos da técnica de reuso de traços implementada em um simulador de arquiteturas ARM.

Logo, espera-se que seja possível encontrar uma maneira eficiente de implementação e configuração da técnica de reuso de traços para atingir bons resultados no que diz respeito à questões de desempenho e/ou consumo de energia.

Inicialmente, espera-se obter os percentuais de instruções reusáveis para verificar a quantidade de reuso encontrada nos *benchmarks* simulados, além do tamanho médio do traço que é formado. Depois, pretende-se configurar a melhor maneira para aproveitar o reuso, as quais incluem: (i) tamanho do *buffer*, (ii) política para substituição de traços e (iii) quantidade de registradores por traço.

Por fim, a implementação do mecanismo de reuso deverá ser testada em paralelo à execução normal dos programas. Assim, será possível avaliar qual o impacto da técnica de reuso sobre a arquitetura ARM, permitindo análise comparando o simulador não modificado com o simulador implementando o mecanismo de reuso. Nesta etapa, espera-se obter resultados referentes à desempenho e consumo de energia, a fim de avaliar se a técnica de reuso trouxe um impacto positivo nas arquiteturas ARM.

### 4. CONCLUSÕES

Este artigo apresentou uma proposta de trabalho para avaliar efeitos de implementar a técnica de reuso de traços em arquiteturas ARM. Para que isso seja possível, foi proposta a utilização de um simulador de arquiteturas que será modificado para dar suporte ao mecanismo de reuso. Para a extração de resultados, foi proposta a utilização de um conjunto de *benchmarks* que será executado, o qual ainda deverá ser escolhido.

Com isto, espera-se que a execução deste trabalho encontre meios eficientes para implementação da técnica de reuso com a finalidade de obter ganhos, tanto em desempenho como em consumo energético. Para buscar isto, foi proposto que uma série de testes fossem simulados, os quais envolvem

verificar o reuso potencial das aplicações, como é estruturado o traço e as diferentes configurações que podem ser feitas no *buffer*.

## 5. REFERÊNCIAS BIBLIOGRÁFICAS

- ARM. **ARM Information Center – Condition code suffixes**. Acesso em: junho de 2015. Disponível em: <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0473c/CEGBHJCJ.html>.
- GONZALEZ, A.; TUBELLA, J.; MOLINA, C. Trace-level reuse. In: **INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING**, Aizu-Wakamatsu City, Japan, 1999. Anais... Washington, DC: IEEE, 1999. p.30–37.
- KNAGGS, P.; WELSH, S. **ARM: Assembly Language Programming**. Bournemouth, England: School of Design, Engineering & Computing, Bournemouth University, 2004.
- KORNAU, T. **Return oriented programming for the ARM architecture**. 2010. Dissertação (Mestrado em Ciência da Computação) - Ruhr-Universitat Bochum, Bochum, Germany.
- MUDGE, T.; AUSTIN, T.; GRUNWALD, D. **Sim-Palyzer 2.0 Reference Manual**. Ann Arbor, MI, USA: Electrical Engineering and Computer Science – University of Michigan, 2001.
- GUTHAUS, M.; RINGENBERG, J.; ERNST, D.; AUSTIN, T.; MUDGE, T.; BROWN, R. Mibench: A free, commercially representative embedded benchmark suite. In: **WWC-4. 2001 IEEE INTERNATIONAL WORKSHOP ON**, Austin: Texas, 2001. Washington, DC: IEEE, 2001. p.3–14.
- MOURA, R. **Potencial de reuso de traços em Arquiteturas ARM**. 2015. Tese (Mestrado em Ciência da Computação) – Universidade Federal de Pelotas, Pelotas, Brasil.
- PILLA, M. **RST: Reuse through Speculation on Traces**. 2004. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil.
- PILLA, M. L.; CHILDERS, B. R.; FRANCA, F. M. G.; DA COSTA, A. T. Limits for a feasible speculative trace reuse implementation. **International Journal of High Performance Systems Architecture**, Genève, Switzerland, v.1, n.1, p.69–76, 2007.
- TORRES, G. **Um estudo sobre os efeitos da técnica de reuso de traços em arquiteturas ARM nas questões de desempenho**. 2015. Monografia (Bacharel em Ciência da Computação) – Universidade Federal de Pelotas, Pelotas, Brasil.