

AVALIANDO O POTENCIAL DE REUSO DE VALORES NA ARQUITETURA ARM

RODRIGO COSTA DE MOURA; GIOVANE DE OLIVEIRA TORRES; MAURICIO LIMA PILLA

Universidade Federal de Pelotas – {rcmoura, gdotorres, pilla}@inf.ufpel.edu.br

1. INTRODUÇÃO

Neste trabalho é apresentado o estudo sobre a viabilidade de implementação de reuso de traços de instruções sobre processadores ARM. O Reuso de Valores é uma técnica que explora a redundância de execução, reaproveitando sequências de execuções recorrentes (PILLA, 2007). É uma técnica não especulativa, isto é, trabalha apenas valores com conhecidos. Conforme a execução da computação acontece, as entradas e saídas são armazenadas em uma tabela indexada, para posterior consulta. Na próxima vez que essa computação for executada, suas entradas serão comparadas com as entradas da tabela. Caso sejam iguais, os valores de saída armazenados na tabela são copiados diretamente para os registradores de saída. Dessa forma, alguns dos estágios do *pipeline* não são utilizados durante o reuso, causando uma importante economia de recursos. Dentre as abordagens que utilizam reuso de valores, a diferença básica entre elas está na granularidade (PILLA, 2004), como reuso de instruções, de blocos básicos ou de traços de instruções. A Figura 1 mostra um mecanismo de reuso de instruções atuando em paralelo com o *pipeline*. Como vantagens das técnicas de reuso destacam-se a economia de recursos, pois instruções reusadas não precisam ser executadas; resultados ficam disponíveis mais cedo; e as instruções que causam dependências, se reusadas, antecipam a execução das instruções delas dependentes.

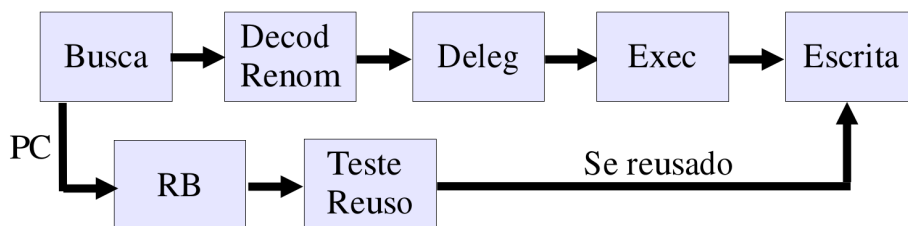


Figura 1. Pipeline com mecanismo de reuso

A Previsão de Valores é uma técnica especulativa que visa aumentar o desempenho de processadores antecipando a execução das próximas computações, aproveitando-se da redundância de instruções (PILLA, 2004). Consiste em prever os valores das próximas computações baseado nos valores das execuções anteriores. Com isso é possível que uma execução aconteça antes que os operandos, ou parte deles, estejam disponíveis. Outra estratégia que pode ser adotada é a combinação das técnicas de Reuso e Previsão de valores. Em PILLA (2004) foi proposto um mecanismo de reuso especulativo denominado *Reuse through Speculation on Traces* (RST). Na arquitetura RST o reuso é feito em uma granularidade mais alta, no nível de traços, que são sequências dinâmicas de instruções (GONZALEZ et al., 1999), (PILLA, 2007), (LAURINO, 2007). Dessa forma, o mecanismo tenta utilizar várias instruções de

uma só vez. Adicionalmente, os operandos que formam os traços podem ser especulados, aumentando assim o tamanho médio e a quantidade de traços que podem ser reusados.

A arquitetura ARM (*Advanced Risc Machine*) (SEAL, 2000) é uma categoria de processadores superescalares do tipo RISC (*Reduced Instruction Level Parallelism*), que prezam pela simplicidade e baixo consumo de energia. A simplicidade dos processadores ARM e suas características que lhe conferem baixo consumo energético, tornaram esses processadores referência para computação embarcada e dispositivos portáteis. São facilmente encontrados em celulares, *smartphones*, *tables*, calculadoras, computadores de bordo, impressoras, equipamentos de rede, etc.

A cada dia novos dispositivos portáteis são desenvolvidos, bem como equipamentos com pequenas unidades de processamento para atividades específicas. O aumento das funções que cada equipamento pode realizar, tornam os projetos cada vez maiores e mais complexos. Nesses casos, é necessário adotar soluções práticas e modulares. Em um *smartphone*, por exemplo, para processar os dados dos acelerômetros, da câmera, e gerenciar uma conexão de rede, é menos custoso utilizar um processador de uso geral do que desenvolver *hardware* especializado para gerenciar cada uma dessas tarefas. As características e funcionalidades providas pelos processadores ARM se mostraram muito adequadas para o desenvolvimento de projetos de dispositivos móveis e sistemas embarcados em geral, como pode ser comprovado pela sua grande disseminação nesses seguimentos. Junto com o crescimento das funcionalidades providas pelos dispositivos móveis, cresce também a necessidade de aumento do poder de processamento destes dispositivos. Neste sentido, a aplicação de técnicas de Reuso de Valores nos processadores ARM, é uma alternativa que propõe trazer ganhos de desempenho para essa categoria de processadores.

Um dos principais desafios em implementar reuso para arquiteturas ARM está relacionado à execução condicional de instruções. No conjunto de instruções ARM (KNAGGS; WELSH, 2004), estas podem ser executadas condicionalmente ou não, dependendo de um predicado e do estado dos *bits* de condição quando do momento de execução. Assim, este trabalho tem o intuito de estimar o potencial de reuso na arquitetura ARM considerando o impacto das instruções condicionais. Com isso, será possível definir quais características um mecanismo de reuso para essa arquitetura deve conter, bem como o ganho de desempenho esperado.

2. METODOLOGIA

Como forma de estimar o potencial de reuso em processadores ARM, no primeiro momento, este trabalho busca identificar o potencial de reuso no nível de instruções. Para isso foram executados os *benchmarks* Mibench (GUTHAUS et al. 2001) sobre o simulador SimPanalyzer, utilizado para simular um processador ARM em um computador de uso geral. Com essa execução foi possível coletar traços de execução de instruções, para posterior análise.

A análise feita consistiu em verificar o percentual de repetições de instruções, as quais possuam os mesmos valores de entrada e, conseqüentemente, os mesmo valores de saída. Nesse momento foram desconsideradas instruções de desvio e instruções de acesso à memória, as quais necessitam de mecanismos adicionais para implementar o reuso (LAURINO et al., 2005), (PILLA, 2004).

Num segundo momento serão identificados os traços de instruções e seus contextos de entrada e saída. Após a análise dos resultados, será feita a identificação da redundância existente, para quantificar o potencial de ganho com o uso das técnicas de reuso de traços. Com isso, objetiva-se analisar o impacto das instruções condicionais e avaliar mecanismos de reuso considerando essas instruções.

Como objetivo futuro é proposta a modificação de um simulador da arquitetura ARM para que este passe a prover um mecanismo de reuso. O objetivo é testar as diferenças de desempenho com as técnicas de reuso e previsão de valores, bem como a combinação delas, em um mecanismo de reuso especulativo (PILLA, 2007).

3. RESULTADOS E DISCUSSÃO

No presente trabalho, o MiBench foi executado sobre um simulador ARM para serem extraídos os resultados. Para isso, foram selecionados 6 *benchmarks* para a realização dos testes:

CRC: Executa um CRC (*Cyclic Reduncancy Check*) de 32 *bits* em um arquivo. CRCs são usados geralmente para verificar possíveis erros em transmissões de dados;

Dijkstra: Algoritmo de busca pelo menor caminho entre duas arestas em um grafo;

Jpeg encode/decode: JPEG é um formato comprimido de imagens. Os algoritmos incluídos no pacote são as etapas de *encode* (compressão) e *decode* (descompressão);

Quicksort: Algoritmo que ordena em ordem crescente uma lista de números;

Stringsearch: Procura por palavras em frases utilizando um algoritmo de comparação *case insensitive*;

Tiff2bw: Filtro de imagem que transforma uma imagem colorida em outra monocromática.

A análise da execução destes *benchmarks* consistiu em identificar a parcela de execução redundante. Esta parcela, apresentada no Gráfico 1, consiste no percentual de instruções com potencial para serem reutilizadas por um mecanismo de Reuso de Valores. Os valores obtidos variaram de 3% a 15%, o que representa o percentual de instruções redundantes.

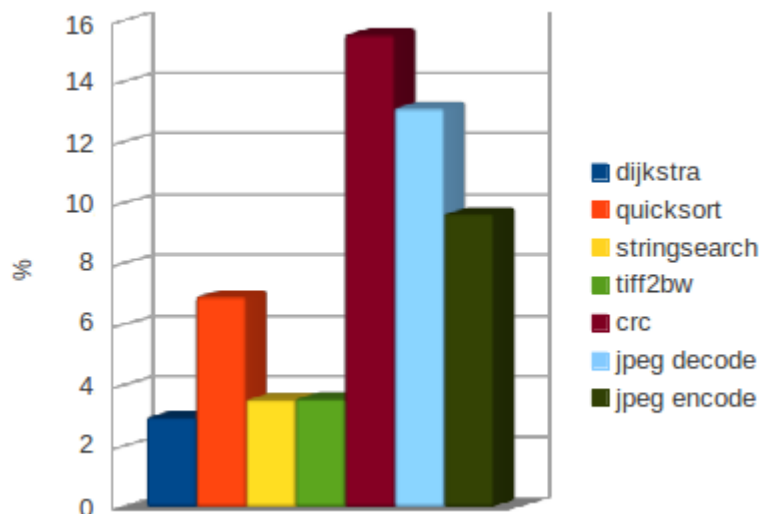


Gráfico 1: Redundância

4. CONCLUSÕES

A análise dos resultados confirma a existência de execução redundante que poderia ser reduzida utilizando um mecanismo de reuso. Porém, a redundância identificada teve grande variação entre os *benchmarks* avaliados, o que dificulta a definição do potencial de ganho do Reuso de Valores em processadores ARM. Essa variação pode estar associada às características específicas das aplicações, o que deve ser avaliado na continuidade deste trabalho.

Estudos semelhantes realizados sobre outras arquiteturas (PILLA, 2007), identificaram redundâncias superiores às encontradas no presente estudo. Dessa forma, como continuidade será feita a avaliação da redundância com os *benchmarks* utilizados neste trabalho sobre diferentes arquiteturas, a fim de validar a metodologia utilizada e identificar comportamentos específicos das diferentes arquiteturas.

5. REFERÊNCIAS BIBLIOGRÁFICAS

PILLA, M. L.; CHILDERS, B. R.; FRANÇA, F. M. G.; COSTA, A. T. da; NAVAUX, P. O. A. Limits for a feasible speculative trace reuse implementation. **IJHPSA**, [S.l.], v.1, n.1, p.69–76, 2007.

LAURINO, L. S.; PILLA, M. L.; SANTOS, T. S. G. dos; NAVAUX, P. O. A. Reuso de Tracos com Loads em Arquiteturas Superescalares. **WSCAD**, [S.l.], 2005.

GONZALEZ, A.; TUBELLA, J.; MOLINA, C. Trace-Level Reuse. **ICPP**, [S.l.: s.n.], p.30–, 1999.

GUTHAUS, M.; RINGENBERG, J. ERNST, D.; AUSTIN, T.; MUDGE, T.; BROWN, R. MiBench: A free, commercially representative embedded benchmark suite. **IEEE 4th Annual Workshop on Workload Characterization**, Austin, Texas, 2001.

LAURINO, L. S. **Reuso especulativo de tracos com instruções de acesso a memória**. 2007. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal do Rio Grande do Sul.

PILLA, M. L. **RST: Reuse through Speculation on Traces**. 2004. Phd thesis — Universidade Federal do Rio Grande do Sul.

KNAGGS, P.; WELSH, S.. ARM: Assembly Language Programming. **School of Design, Engineering & Computing**, Bournemouth University, 2004.

SEAL, D. **Arm Architecture Reference Manual**. Addison-Wesley Longman Publishing Co., 2000.