

SSC114 – Arquitetura de Computadores – 2º sem/2018

Trabalho Prático

Data da disponibilização: 28/10/18 - Data de entrega: 05/12/18

Grupos de 4 pessoas

O objetivo deste trabalho prático é permitir que os alunos assimilem melhor o conhecimento da disciplina.

Para atingir o objetivo, os grupos podem escolher uma das duas possibilidades especificadas abaixo:

- 1) Desenvolvimento de um material que será utilizado como um Recurso Educacional Aberto (REA) ¹ no contexto de paralelismo em nível de instrução; ou
- 2) Análise de desempenho de diferentes arquiteturas através da obtenção de métricas utilizando programas padrões (benchmarks).

Na opção (1) deve-se desenvolver um REA que ilustre o funcionamento de pelo menos um (ou mais do que um) dos principais tópicos de pipeline estudados, sendo:

- Definição básica de pipeline e os registradores intermediários: definição e exemplos de funcionamento;
- Dependência de Dados: definição, exemplo do problema e solução por *forwarding*;
- Dependência de Controle: definição, exemplo do problema e solução com predição dinâmica;
- Escalonamento de instruções: implementações *Scoreboarding* ou Tomasulo.

O REA pode ser de diferentes maneiras. Algumas alternativas são:

- Um simulador do comportamento de um algoritmo ou técnica;
- Uma sequência de ilustrações, como uma história em quadrinhos, como por exemplo em <https://jvns.ca/zines/#perf>;
- Uma página WEB com explicações e desenhos (<https://jvns.ca/blog/2016/12/03/how-much-memory-is-my-process-using/>)

Na opção (2), deve-se definir os programas que servirão como benchmark² para comparação e quais arquiteturas serão comparadas. Pode-se avaliar arquiteturas tais como AMD, Intel, ARM, etc. As métricas de avaliação podem ser, por exemplo, número de ciclos utilizado pelo programa, número de instruções finalizadas, CPI (e IPC), número de instruções de desvio, predições de desvio erradas, falhas na cache, etc. Para fazer essa análise, pode-se utilizar a ferramenta *perf*, também conhecida como *Performance Counters for Linux* (PCL). Maiores informações sobre a utilização da *perf* podem ser obtidas em: <http://www.brendangregg.com/perf.html> e também em <https://jvns.ca/zines/#perf>.

¹ <http://www.rea.net.br/site/faq/#a2>

² [https://pt.wikipedia.org/wiki/Benchmark_\(computação\)](https://pt.wikipedia.org/wiki/Benchmark_(computação))

Os benchmarks definidos devem representar algumas situações, como por exemplo, códigos com loops (que serão convertidos em instruções de desvio), códigos com leitura e escrita na mesma variável (que será convertido em instruções com dependência de dados), códigos que fazem muitos acessos a memória, códigos com muita computação, etc.

Para a análise de desempenho, deve-se repetir a execução de cada *benchmark* pelo menos 10 vezes, coletando todas as métricas e calcular a média e o intervalo de confiança (IC) para cada métrica. Sugere-se um nível de nível de confiança de 95% para o cálculo do IC. Após a obtenção dos dados e do cálculo das médias e ICs, deve-se analisar os resultados e concluir.

Tanto para a opção (1) como para a opção (2), o grupo deve fazer uma **monografia** com as seguintes seções: Introdução (descrevendo o trabalho escolhido), Desenvolvimento do trabalho, Conclusões e Bibliografia.

Todos os códigos desenvolvidos e demais arquivos gerados no desenvolvimento do trabalho devem ser compactados junto com a monografia e submetidos no Moodle na tarefa “Trabalho Prático”.

Caso surjam quaisquer dúvidas, entrem em contato com a professora e/ou monitor da disciplina.