INF01113 - Organização de Computadores B
Primeiro Trabalho Prático

Cristiano Medeiros Dalbem - 173362
Instituto de Informática
Universidade Federal do Rio Grande do Sul
cristiano.dalbem@inf.ufrgs.br

29 de outubro de 2009

1 Resultados dos Benchmarks

Segue as tabelas com as saídas das simulações no programa SimpleScalar.

1.1 Quesito 1 - Preditor de Desvios

Multiplicação de Matrizes		
Configuração (predição de desvios)	IPC (sim_IPC)	Número de Ciclos (sim_cycle)
not taken	1,6418	4691408
bimodal	2,0146	3823298
perfect	2,0171	3818596

Cyclic Redundancy Check		
Configuração (predição de desvios)	IPC (sim_IPC)	Número de Ciclos (sim_cycle)
not taken	0,7447	81180
bimodal	1,3918	43435
perfect	1,8140	33324

Rijndael		
Configuração (predição de desvios)	IPC (sim_IPC)	Número de Ciclos (sim_cycle)
not taken	0,4667	16862
bimodal	0,5420	14520
perfect	0,5769	13642

1.2 Quesito 2 - Grau de Superescalaridade

Multiplicação de Matrizes		
Configuração (número de ULAs)	IPC (sim_IPC)	Número de Ciclos (sim_cycle)
1	1,0326	7459047
2	1,8283	4212933
4	2,0146	3823298

Cyclic redundancy check		
Configuração (número de ULAs)	IPC (sim_IPC)	Número de Ciclos (sim_cycle)
1	0,8554	70666
2	1,2905	46842
4	1,3918	43435

Rijndael		
Configuração (número de ULAs)	IPC (sim_IPC)	Número de Ciclos (sim_cycle)
1	0,4376	17984
2	0,5219	15079
4	0,5420	14520

2 Discussão de Resultados

2.1 Por Benchmark

Explicarei aqui o efeito das variações da arquitetura na eficiência com que a máquina processa cada um dos benchmarks.

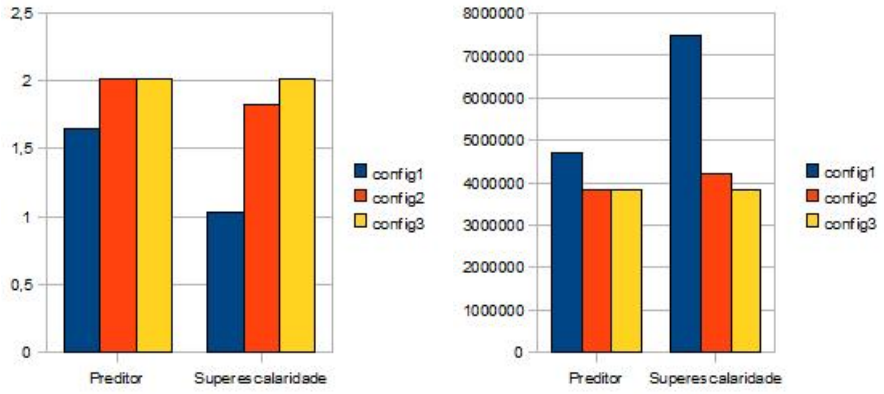
Abaixo, uma tabela com estatísticas de tipos de operações realizadas por cada benchmark, o que ajuda em entendermos a razão das variações obtidas nos testes.

	MM	CRC	RIJNDAEL
Load	898317 (**11.66%**)	12988 (**21.49%**)	8840790 (**23.51%**)
Store	146089 (**1.9%**)	10815 (**17.89%**)	2971213 (**7.90%**)
Desvio Incondicional	150519 (**1.95%**)	4567 (**7.56%**)	216508 (**0.58%**)
Desvio Condicional	136344 (**1.77%**)	5412 (**8.95%**)	1956969 (**5.20%**)
Aritmética de Inteiros	6371210 (**82.72%**)	26661 (**44.10%**)	23619223 (**62.81%**)

2.1.1 Multiplicação de Matrizes

Para o caso do benchmark de Multiplicação de Matrizes temos o resultado, que já era esperado, de que a maior melhoria que se conseguiria seria na otimização da superescalaridade, i.e. adicionando-se ULAs à arquitetura. Isso se dá pois este é um programa de "força bruta", que executa uma enorme quantidade de cálculos númericos (aritmética de inteiros) em detrimento a outros tipos de intruções. Isso explica o fato de se ter péssimo desempenho com 1 ULA e um ótimo melhoramento à medida que acrescemos ULAs.

Ainda assim há uma melhoria de desempenho com técnicas de predição de desvios, pois este é um código com muitos loops para a varredura das matrizes, e é bastante útil não termos erros nas previsões. (ver Figura 1)

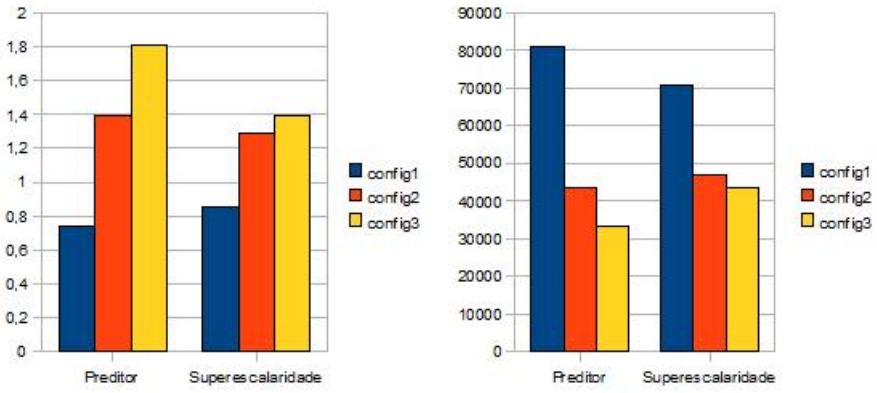


(a) IPC por método (b) Número de ciclos por método

Figura 1: Estatísticas de benchmark sobre Multiplicação de Matrizes

2.1.2 Cyclic Redundancy Check

Aqui temos um caso oposto ao anterior: este algoritmo tem, em relação aos outros dois algoritmos, porcentagem bastante alta de presença de instruções de desvio. Isso explica o porquê de melhorias na predição de desvios melhorar consideravelmente os resultados. A partir do momento que temos uma alta probabilidade de acertar o desvio, a pipeline não desperdiça grandes quantidades de instruções já resolvidas, aproveitando o ganho de eficiência gerado pelo uso de pipelines, como visto em aula (ver Figura 2).

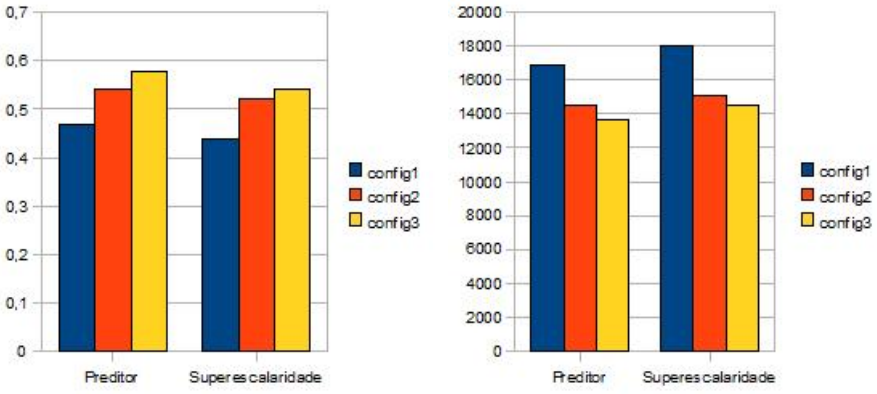


(a) IPC por método (b) Número de ciclos por método

Figura 2: Estatísticas de benchmark sobre Cyclic Redundancy Check

2.1.3 Rijndael

Neste caso não há melhorias significativas com qualquer um dos métodos de melhoria de desempenho estudados, já que não há quantidades significativas de desvios nem de operações aritméticas em detrimento às outras (ver Figura 3).



(a) IPC por método (b) Número de ciclos por método

Figura 3: Estatísticas de benchmark sobre Rijndael

2.2 Por Quesito

Pontuarei aqui rapidamente as características dos comportamentos apresentados nos testes realizados levando em consideração os dois quesitos estudados.

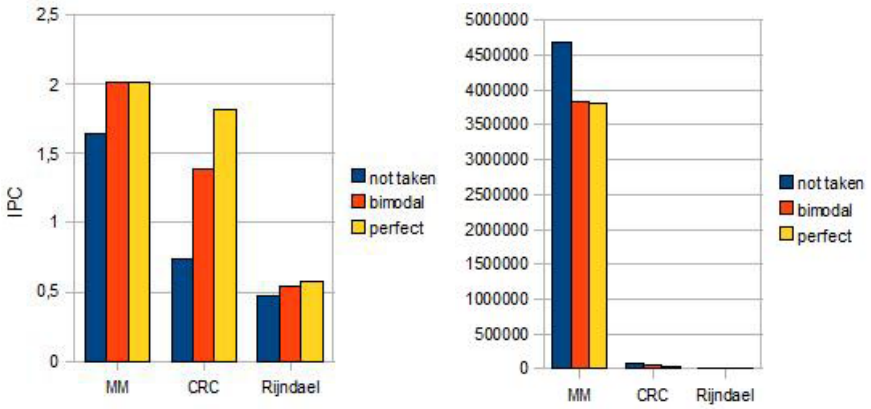
2.2.1 Preditor de Desvios

Neste exemplo temos claramente que a mais notável melhoria foi no benchmark CRC em razão de seu grande número de desvios. Não é utilizado nenhum método complexo de predição: simula-se um sistema que não faz predição, um idealizado que sempre acerte a predição, e um que utiliza a técnica bimodal. Essa técnica consiste em utilizar dois bits que representarão uma memória das últimas decisões tomadas e, como que utilizando esse número como medidor de "humor"("estou bem humorado, estou certando todas"ou "meu humor está baixo, não tenho acertado"), decide se fará a suposição de que o desvio seja tomado ou não.

No primeiro e no terceiro benchmarks percebemos que não há diferença significativa entre o bimodal e o ideal. Já no CRC podemos notar uma variação mais significativa, e isso se deve à grande importância que tem a Predição de Desvios para esse algoritmo; na realidade, como já foi comentado, esse algoritmo tem alto número de desvios, o que torna transparente tal efeito (ver Figura 4).

2.2.2 Grau de Superescalaridade

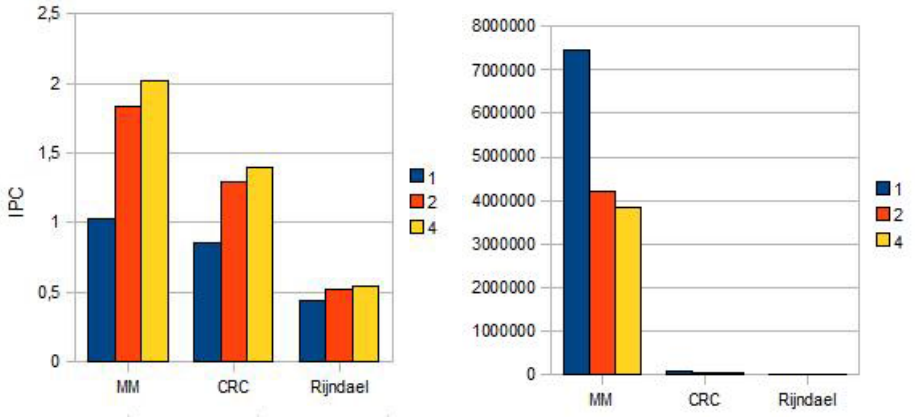
Aqui podemos notar claramente como o melhoramento desse quesito afeta especialmente o desempenho dependendo do tipo de programa executado. Como já comentado anteriormente, o algoritmo de multiplicação de matrizes é perfeito para paralelização e uso de múltiplas ULAs, já que há um trabalho de "força bruta"com uma enorme quantidade de operações de cálculo - o que pode ser notado facilmente pelo gráfico da Figura 5(b) - além de baixa dependência de dados nas leituras e escritas. Todos estes fatores, além da baixa quantidade de desvios, contribuem na otimização da eficiência do pipeline, o que se reflete notavelmente na eficiência geral (ver Figura 5).



(a) IPC por configuração

(b) Número de ciclos por configuração

Figura 4: Estatísticas de variação em Predição de Desvios



(a) IPC por configuração

(b) Número de ciclos por configuração

Figura 5: Estatísticas de variação em Grau de Superescalaridade