

Investigação de Técnicas de Projeto de Cache de Dados para Sistemas Embarcados

Giancarlo C. Heck

Larsis / PPGInf / UFPR

22 de agosto de 2008

Dissertação de Mestrado
Orientador: Prof. Roberto A. Hexsel

Roteiro

- ▶ Introdução
- ▶ Ambiente de Simulação
- ▶ Modelos de Cache
- ▶ Análise de Resultados
- ▶ Conclusão

Introdução

- ▶ Desempenho dos processadores e memórias - 35 a 55% *versus* 7%
- ▶ Redução da diferença com o aprimoramento das memórias cache - redução do tempo médio de acesso à memória
- ▶ Sistemas embarcados - domínio do mercado de microprocessadores (98%) - desempenho, consumo de energia, espaço ocupado, custo
- ▶ Avaliação das melhorias de cache

Ambiente de Simulação - SimpleScalar

- ▶ Conjunto de simuladores altamente configuráveis
- ▶ Simulador *sim-outorder*
- ▶ Hierarquia de memória de três níveis

Ambiente de Simulação - Parâmetros

► Parâmetros Variáveis

PARÂMETROS	Valores
Tamanho da cache (KB)	1, 2, 4, 8 e 16
Largura do bloco (bytes)	8, 16, 32 e 64
Associatividade da cache	1
Modelo de CPU	1, 2, 4 e 8

- Permitem investigação realista
- Representam principais características

Ambiente de Simulação - Parâmetros

► Parâmetros Fixos

MODELO DE CPU	1	2	4	8
Escalaridade	1	2	4	8
Tamanho da LSQ	4	8	16	32
Tamanho da RUU	8	16	32	64

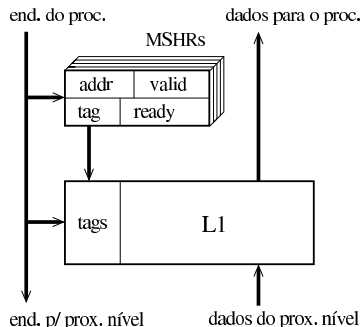
- Palavras de dados e instruções de 32 bits
- Cache L1 diretamente mapeada
- Barramento de dados de 8 bytes

Ambiente de Simulação - CommBench

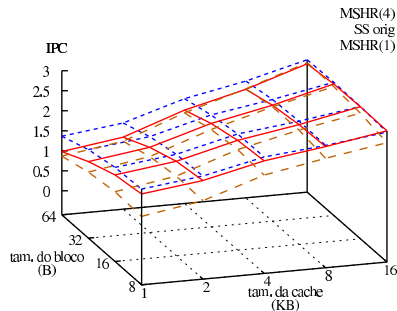
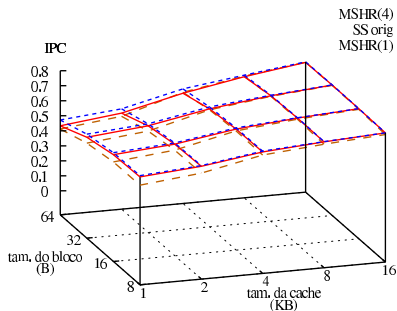
- ▶ Aplicações comuns em processadores de rede
- ▶ CAST-128 - criptografia aritmética (enc e dec)
- ▶ DRR - escalonamento (*Deficit Round Robin*)
- ▶ FRAG - fragmentação de pacotes IP
- ▶ JPEG - compressão de imagens (enc e dec)
- ▶ Utilização de *scripts* para coletar os 63.165 resultados (armazenados em BD)

Cache Não-Bloqueante (MSHR)

- ▶ Reduz degradação de desempenho
- ▶ MSHRs (*Miss Status Holding Registers*)
- ▶ N° de MSHRs = N° faltas pendentes
- ▶ Permite múltiplos acessos p/ mesmo endereço

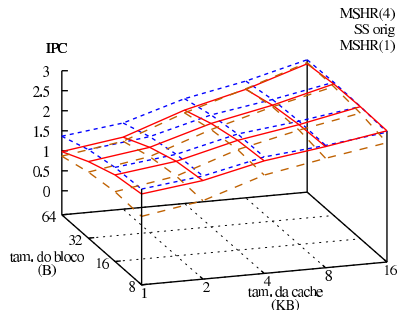
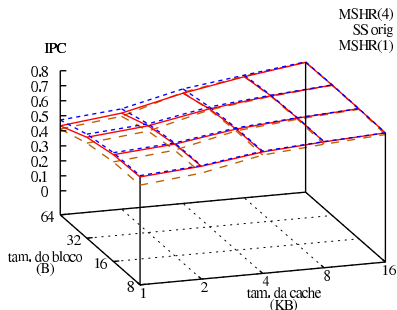


Resultados do MSHR - Largura 1



- ▶ Efeitos de uma cache não-bloqueante são mínimos com largura 1
- ▶ Em um ciclo qualquer existe no máximo um *load* ou *store* na interface de memória
- ▶ Pouca diferença no desempenho entre o SS_orig e o modelo com MSHR

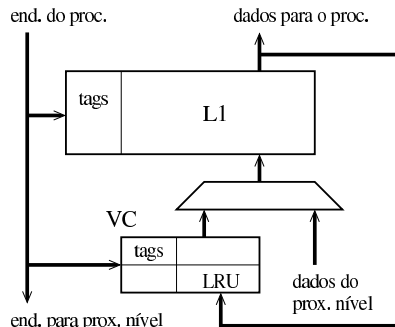
Resultados do MSHR - Largura 8



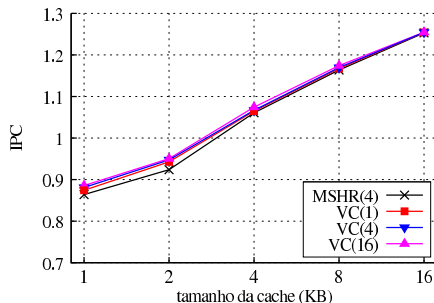
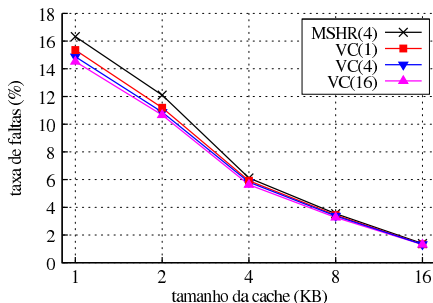
- ▶ Largura 8 impõe uma alta demanda ao sistema de memória
- ▶ É possível executar, concorrentemente, até oito referências à memória
- ▶ Caches pequenas e com poucos blocos melhoram o desempenho em 0,5 IPC

Cache de Vítima (VC)

- ▶ Pequena cache totalmente associada.
- ▶ Localizada entre a L1 e a memória
- ▶ Opera em paralelo à L1
- ▶ Reduz faltas por conflito

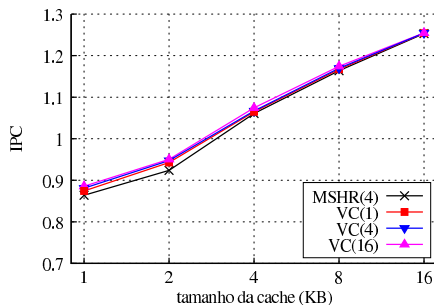
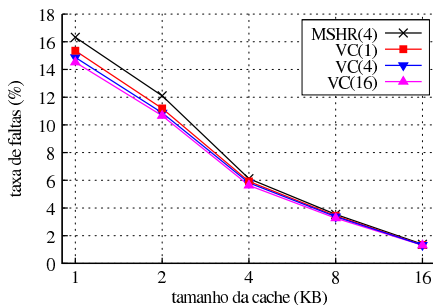


Resultados da VC - Taxa de Faltas e IPC



- ▶ Resultados da taxa de faltas e do IPC da VC comparados ao modelo com 4 MSHRs
- ▶ Valores médios dos seis programas de teste com processador de largura 2 e blocos de 16 bytes

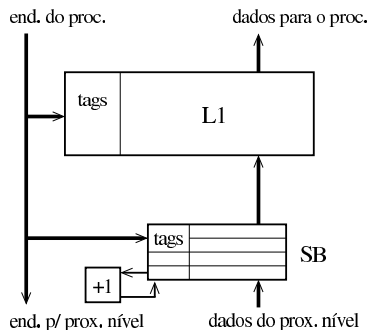
Resultados da VC - Taxa de Faltas e IPC



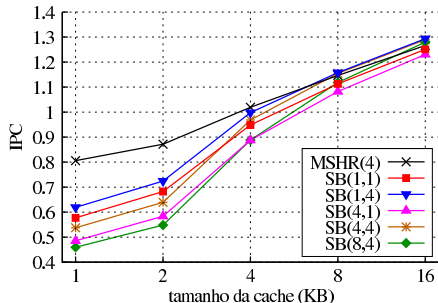
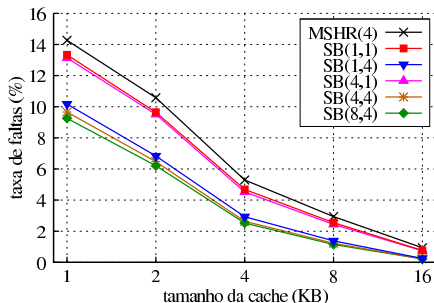
- ▶ Cache de 1 KB + VC(16) melhora a taxa de faltas em 11%
- ▶ Ganhos mais evidentes para caches pequenas
- ▶ Ganhos no IPC correspondem diretamente aos das taxas de faltas

Stream Buffer (SB)

- ▶ Fila de registros que efetuam busca antecipada
- ▶ Localizado entre a L1 e a memória
- ▶ Reduz faltas compulsórias e por capacidade
- ▶ *tags* são verificadas em todos os blocos

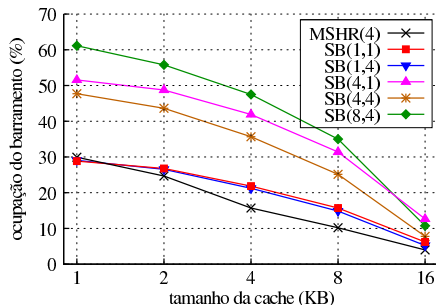
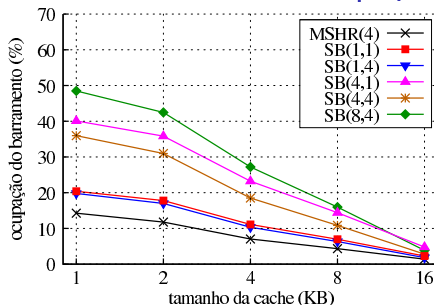


Resultados do SB - Taxa de Faltas e IPC



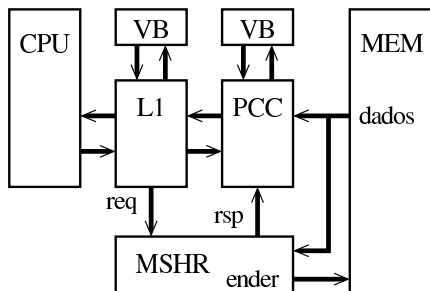
- Organização dos *buffers* é dada como SB(*prof.*, *assoc.*)
- Taxa de faltas (esq.) e IPC (dir.) para um processador de largura 2 e tamanho de bloco de 32 bytes
- SB(1,1) e SB(1,4) causam as maiores taxas de faltas enquanto apresentam os melhores desempenhos

Resultados do SB - Ocupação do Barramento



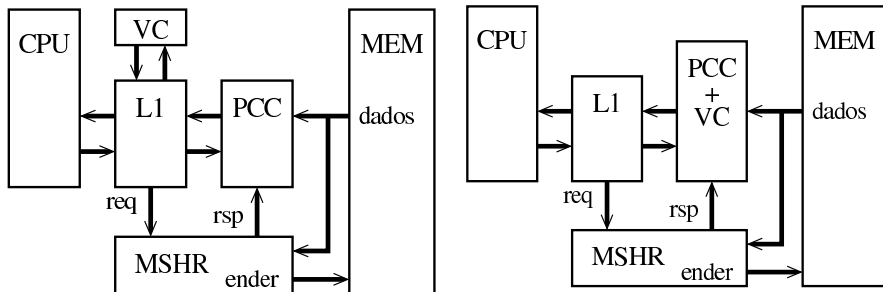
- ▶ Processadores de largura 1 (esq.) e 4 (dir.) e tamanho de bloco de 32 bytes
- ▶ SB(1,x), ocupação $\approx 30\%$ maior que a do modelo base
- ▶ SBs com 4 ou mais blocos tem uma elevadíssima ocupação

Cache de Controle de Poluição (PCC)



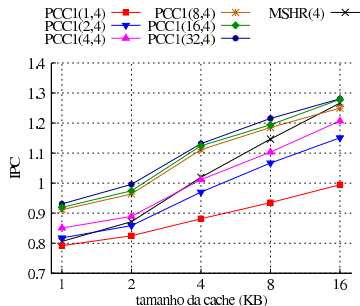
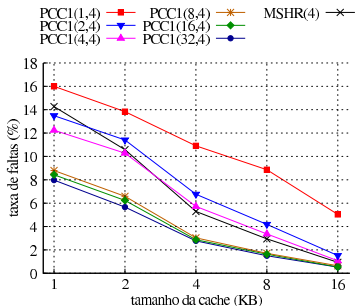
- ▶ “Poluição” provoca excesso de faltas
- ▶ Pequena cache totalmente associativa junto à L1
- ▶ Blocos são alocados primeiro na PCC
- ▶ Na segunda referência a um bloco ele vai para a L1

PCC1 e PCVC



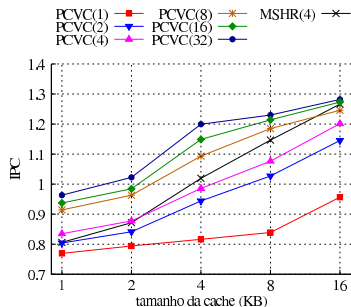
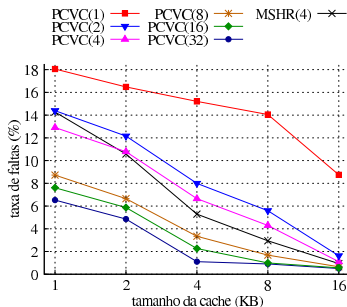
- ▶ $PCC1 = PCC + VC$ junto à L1
- ▶ Lógica de controle complexa na PCC1
- ▶ PCVC = PCC e VC em uma única estrutura

Resultados da PCC1 - Taxa de Faltas e IPC



- ▶ Larg.=2, t_b=32 e média dos programas de teste
- ▶ PCC1(*Nº de blocos na PCC*, *Nº de blocos na VC*)
- ▶ PCC1s grandes reduzem $\approx 50\%$ a taxa de faltas e produzem ganhos no IPC de 11% até 14% (8-32 blocos)

Resultados da PCVC - Taxa de Faltas e IPC



- ▶ Larg.=2, t_b=32 e média dos programas de teste
- ▶ Ganhos de 39%, 47% e 54% na taxa de faltas
- ▶ Ganhos de 13%, 16% e 19% no IPC (cache de 1 KB e PCVC de 8, 16 e 32 blocos)

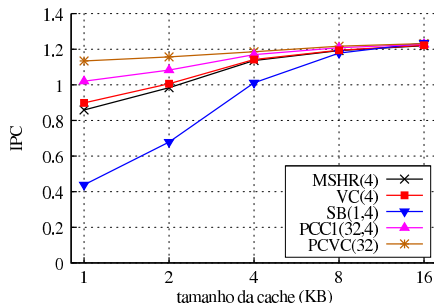
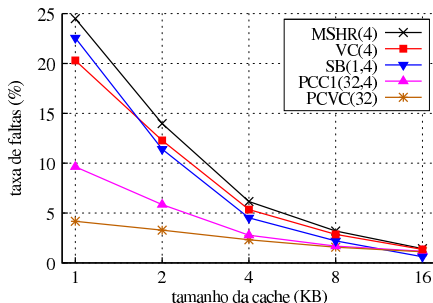
PCC1 *versus* PCVC - Ganhos no IPC

TAMANHO DA CACHE	1K	2K	4K	8K	16K
IPC base MSHR(4)	0,81	0,87	1,02	1,15	1,27
PCC1(4,4)/BASE	1,06	1,02	0,99	0,96	0,95
PCVC(4)/BASE	1,04	1,01	0,97	0,94	0,95
PCC1(32,4)/BASE	1,15	1,14	1,11	1,06	1,01
PCVC(32)/BASE	1,20	1,17	1,18	1,07	1,01
PCC1(32,4)/BASE*2	—	0,96	0,86	0,86	0,85
PCVC(32)/BASE*2	—	1,10	1,00	1,05	0,97
PCVC(32)/BASE*4	—	—	0,95	0,89	0,95
PCVC(32)/BASE*8	—	—	—	0,84	0,81
PCVC(32)/BASE*16	—	—	—	—	0,78

PCC1 *versus* PCVC

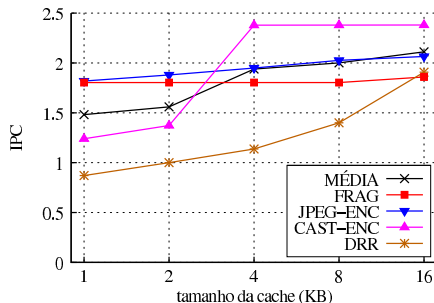
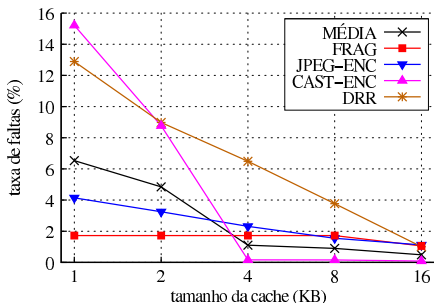
- ▶ Com base nos resultados das simulações pode-se dizer que a PCVC é um modelo mais efetivo do que a PCC1 porque:
 1. Ela é mais simples —possui apenas dois conjuntos de *tags* ao invés de três
 2. É menor —somente n blocos no PCVC ao invés de $PCC(n) + VC(m)$ blocos, e
 3. Tem melhor desempenho tanto em termos de taxa de faltas quanto de IPC

Resultados Individuais dos Programas de Teste - JPEG



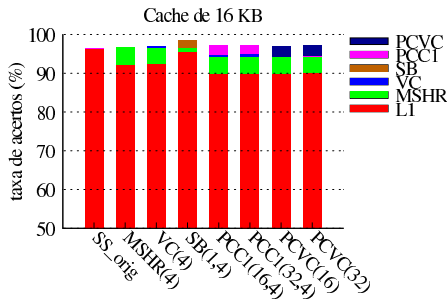
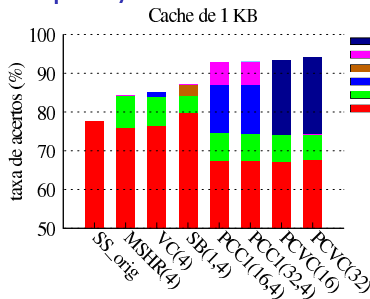
- ▶ Desempenho com JPEG ENC, larg.=2, t_b=32
- ▶ Resultados promissores para caches pequenas e esta classe de programas
- ▶ PCC1 e PCVC: redução da taxa de faltas em 61% e 77% e ganho no IPC de 35% e 43% (cache de 1 KB)

Resultados Individuais dos Programas de Teste - PCVC(32)



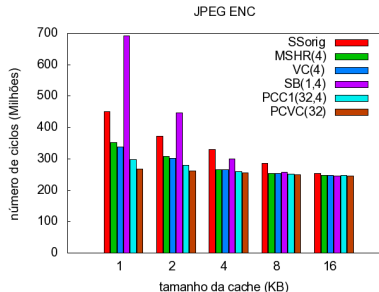
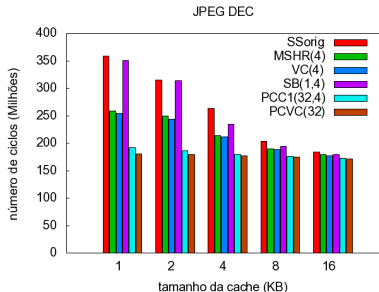
- ▶ Desempenhos individuais e média dos programas de teste
- ▶ Processador de largura 4 com blocos de 32 bytes

Composição da Taxa de Acertos Total



- ▶ Composição da taxa de acertos para JPEG DEC, caches de 1 e 16 KB, larg.=2, t_b=16
- ▶ Migração dos acertos (redução dos acertos na L1)
- ▶ Cache de 16 KB comporta prat. todo o conjunto de dados
- ▶ PCC1 aumenta o reúso dos blocos na VC (loc. temporal)

Tempo de Execução - JPEG DEC e ENC



- ▶ Número de ciclos total de execução dos programas, caches de 1 a 16 KB, larg.=2, t_b=32
- ▶ Comportamento comum - SS_orig, MSHR e VC
- ▶ Menores tempos de execução - PCC1 e PCVC

Cálculo dos Tamanhos Ocupados

- ▶ PCVC(32) $\Rightarrow$ 32 blocos de 32 bytes = 1K + controle

$$\text{L1 de 1K} + \text{PCVC}(32) = 2\text{K} == 2\text{K}$$

$$\text{L1 de 2K} + \text{PCVC}(32) = 3\text{K} \approx 4\text{K}$$

$$\text{L1 de 4K} + \text{PCVC}(32) = 5\text{K} < 8\text{K}$$

- ▶ PCVC(16) $\Rightarrow$ 16 x 32 = 512 + controle

$$\text{L1 de 1K} + \text{PCVC}(16) = 1,5\text{K} \approx 2\text{K}$$

$$\text{L1 de 2K} + \text{PCVC}(16) = 2,5\text{K} < 4\text{K}$$

$$\text{L1 de 4K} + \text{PCVC}(16) = 4,5\text{K} \ll 8\text{K}$$

- ▶ PCVC(8) $\Rightarrow$ 8 x 32 = 256 + controle

$$\text{L1 de 1K} + \text{PCVC}(8) = 1,25\text{K} < 2\text{K}$$

$$\text{L1 de 2K} + \text{PCVC}(8) = 2,25\text{K} \ll 4\text{K}$$

$$\text{L1 de 4K} + \text{PCVC}(8) = 4,25\text{K} \ll 8\text{K}$$

Conclusão

- ▶ Cache de Vítima é apropriada para aplicações que possuem um grande conjunto de dados e boa localidade temporal
- ▶ Stream Buffer teve um desempenho abaixo do esperado. Pode apresentar melhor desempenho em um sistema com barramento mais complexo ou se colocado entre a cache L2 e a memória principal

Conclusão

- ▶ Caches de Controle de Poluição testadas mostraram resultados promissores
- ▶ O novo modelo, a PCVC, superou os resultados da cache mais complexa, a PCC1
- ▶ A PCVC obteve o melhor desempenho quando comparada a caches muito maiores

Trabalhos Futuros

- ▶ O estudo de novos modelos de memória cache que não foram contemplados nesta pesquisa e de técnicas de projeto para redução do consumo de energia serão considerados em um trabalho futuro
- ▶ Pretende-se ainda testar o modelo PCVC com programas de teste de aplicações multimídia

Investigação de Técnicas de Projeto de Cache de Dados para Sistemas Embarcados

Giancarlo C. Heck

giancarlo@inf.ufpr.br