Evento	Salão UFRGS 2022: FEIRA DE INOVAÇÃO TECNOLÓGICA DA UFRGS - FINOVA
Ano	2022
Local	Campus Centro - UFRGS
Título	Sim2PIM: framework completo de simulação para processamento em memória
Autores	AUGUSTO EXENBERGER BECKER BRUNO ENDRES FORLIN PAULO CESAR SANTOS DA SILVA JUNIOR
Orientador	LUIGI CARRO

RESUMO

TÍTULO DO PROJETO: Sim2PIM: Framework Completo de Simulação para Processamento em Memória.

Aluno: Augusto Exenberger Becker
Orientador: Luigi Carro

RESUMO DAS ATIVIDADES DESENVOLVIDAS PELO BOLSISTA

Com a desaceleração da Lei de Moore e o aumento da diferença de perfomance entre processadores e tecnologias de memória, estratégias para sustentar os ganhos de perfomance tornam-se necessárias. Nos últimos anos, arquiteturas para Processamento em Memória (PIM) foram propostas, com o objetivo de suprir essa demanda [1-4]. Dessa forma, se tornam necessárias ferramentar para avaliar o desempenho de arquiteturas PIM e sua viabilidade em situações do mundo real.

O objetivo desse projeto foi expandir o simulador Sim2PIM [5], permitindo simular programs com instruções de processamento em memória PIM, compostos por múltiplas *threads*, mantendo o compromisso com a precisão e os baixos *Overheads,* presentas na solução original. O Simulador Sim2PIM é baseado no uso de contadores de Hardware para trechos executados no processador original (*host*) e no *offload* de instruções PIM para o código que representa a arquitetura específica. Dessa forma, diferencia-se de outros simuladores disponíveis atualmente, baseados em simular o sistema completo (*host* e arquitetura PIM) [6,7] ou no uso de arquivos *trace* [8] para extraír o funcionamento do processador *host.* Essa diferença permite obter dados mais precisos do *host,* ao mesmo tempo em que se introduzem menos *overheads* ao processo [5].

Para a expansão do simulador, foi projetada uma nova arquitetura, focada em manter o isolamento entre os módulos da simulação, para garantir a precisão das medidas. Na arquitetura proposta inicialmente, o código compilado é instrumentado, trocando chamadas para criação de threads por chamadas do simulador, que isolam cada thread em um *core* e iniciam os contadores de Hardware. Além disso, chamadas de instruções PIM também são instrumentadas. A figura 1 apresenta o diagrama de execução e overheads para um aplicação multithread no simulador Sim2PIM.

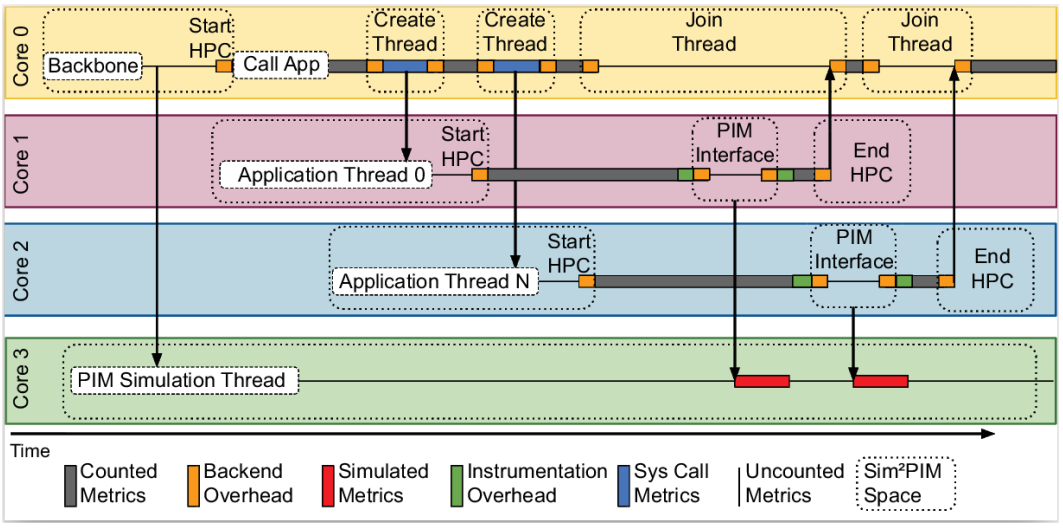

Figura 1 - Diagrama de Execução Aplicação Multithread.

Para validação das medições e comparação dos resultados foi necessário escolher e implementar *benchmarks* a serem rodados no simulador. As aplicações escolhidas foram baseadas no *PolyBench* [9]. Cada uma das aplicações foi executada no Sim2PIM para obtenção das medidas. Ademais, também foram extraídos indicadores utilizando a ferramenta Perf. A tabela 1 apresenta a comparação entre os valores obtidos pelo Sim2PIM e pelo Perf.

Benchmark - data size	Perf cycles		Sim²PIM cycles		cycles % increase		Perf Time (s)		Sim²PIM Time (s)	
	1T	4T - Average	1T	4T - Average	1T	4T - Average	1T	4T - Average	1T	4T - Average
vecsum - 32MB	1.25E+07	3.06E+06	1.23E+07	3.05E+06	-1.799	-0.541	0.0165	0.0083	0.037	0.035
gemm - 1,5MB	2.89E+07	9.02E+06	2.84E+07	8.77E+06	-1.577	-2.771	0.0173	0.0089	0.029	0.033
2mm - 750kB	1.57E+08	3.93E+07	1.57E+08	3.92E+07	-0.018	-0.255	0.0524	0.0219	0.039	0.046
covariance - 16MB	1.58E+09	4.30E+08	1.61E+09	4.23E+08	1.898	-1.734	0.7163	0.4740	1.041	0.5
Floyd-Warshall - 8MB	1.18E+10	3.93E+09	1.18E+10	3.88E+09	-0.018	-1.261	3.2186	1.1955	3.232	1.22
Nussinov - 8MB	2.86E+10	7.23E+09	2.88E+10	7.21E+09	0.597	-0.319	7.7568	1.9769	7.825	1.998

Tabela 1 - Ciclos Simulados vs Medidas do Perf.

Pode-se observar que a diferença percentual entre as medições se manteve baixa, com um pico de aproximadamente 2,7% na simulação do algorítmo gemm com 4 *threads*. Além disso, é importante ressaltar que o tempo de simulação foi próximo do tempo de execução utilizando a ferramenta Perf, comprovando o baixo *overhead* inserido pelo simulador. Assim, é possível perceber que a arquitetura implementada conseguiu satisfazer os objetivos de precisão e baixos *overheads*.

Referências

[1] H. A. D. Nguyen, J. Yu, M. A. Lebdeh, M. Taouil, S. Hamdioui, F. Catthoor, A classification of memory-centric computing, J. Emerg. Technol. Comput. Syst. 16 (2).

[2] J. Ahn, S. Hong, S. Yoo, O. Mutlu, K. Choi, A scalable processing-in-memory accelerator for parallel graph processing, in: Int. Symp. on Computer Architecture (ISCA), IEEE, 2015.

[3] S. Aga, S. Jeloka, A. Subramaniyan, S. Narayanasamy, D. Blaauw, R. Das, Compute caches, in: 2017 IEEE International Symposium on High Performance Computer Architecture (HPCA), 2017.

[4] P. C. Santos, G. F. Oliveira, D. G. Tomé, M. A. Alves, E. C. Almeida, L. Carro, Operand size reconfiguration for big data processing in memory, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2017.

[5] P. C. Santos, B. E. Forlin, L. Carro, Sim²pim: A fast method for simulating host independent pim agnostic designs, DATE ’21, 2021.

[6] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, D. A. Wood, The gem5 simulator, SIGARCH Comput. Archit. News.

[7] M. A. Z. Alves, C. Villavieja, M. Diener, F. B. Moreira, P. O. A. Navaux, Sinuca: A validated micro-architecture simulator, in: 2015, 17th Int. Conf. on High Performance Computing and Communications, 2015.

[8] Xu, X. Chen, Y. Wang, Y. Han, X. Qian, X. Li, Pimsim: A flexible and detailed processing-in-memory simulator, IEEE Computer Architecture Letters.

[9] L.-N. Pouchet, Polybench: The polyhedral benchmark suite, URL: http://www.cs.ucla.edu/pouchet/software/polybench.