

Geração de traços de simulação para instruções de processamento em memória

Aline S. Cordeiro -
alinesantanacordeiro@gmail.com

Prof. Dr. Marco A. Z. Alves

1.

Processamento em memória Introdução

Motivação

- ▶ Novas arquiteturas de Processamento em Memória (PIM);
- ▶ Permite o processamento de dados no mesmo chip que a memória DRAM.

Motivação

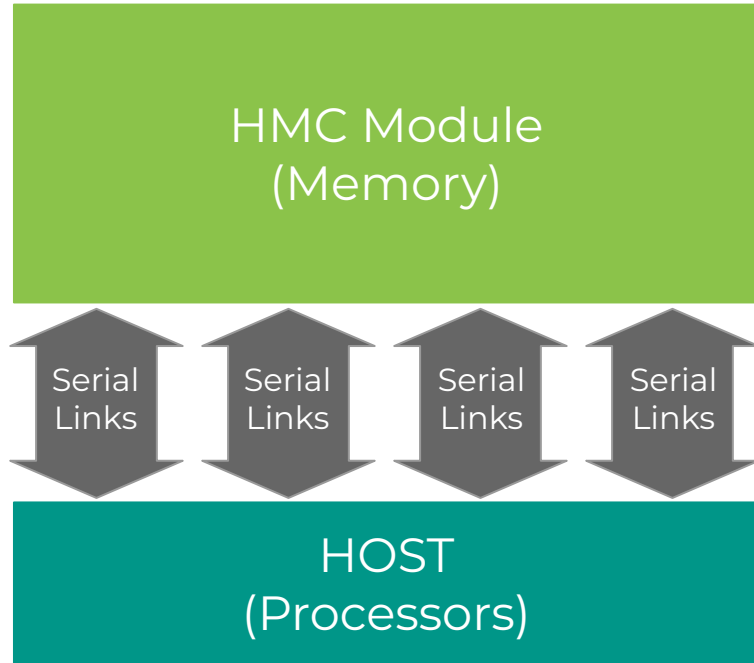
- ▶ Processamento mais rápido e menor consumo de energia devido à menor transferência de dados;
- ▶ Dois projetos principais de PIM (HMC/HBM) já estão sendo comercializados:
 - ▶ Instruções são executadas próximo à DRAM.



2.

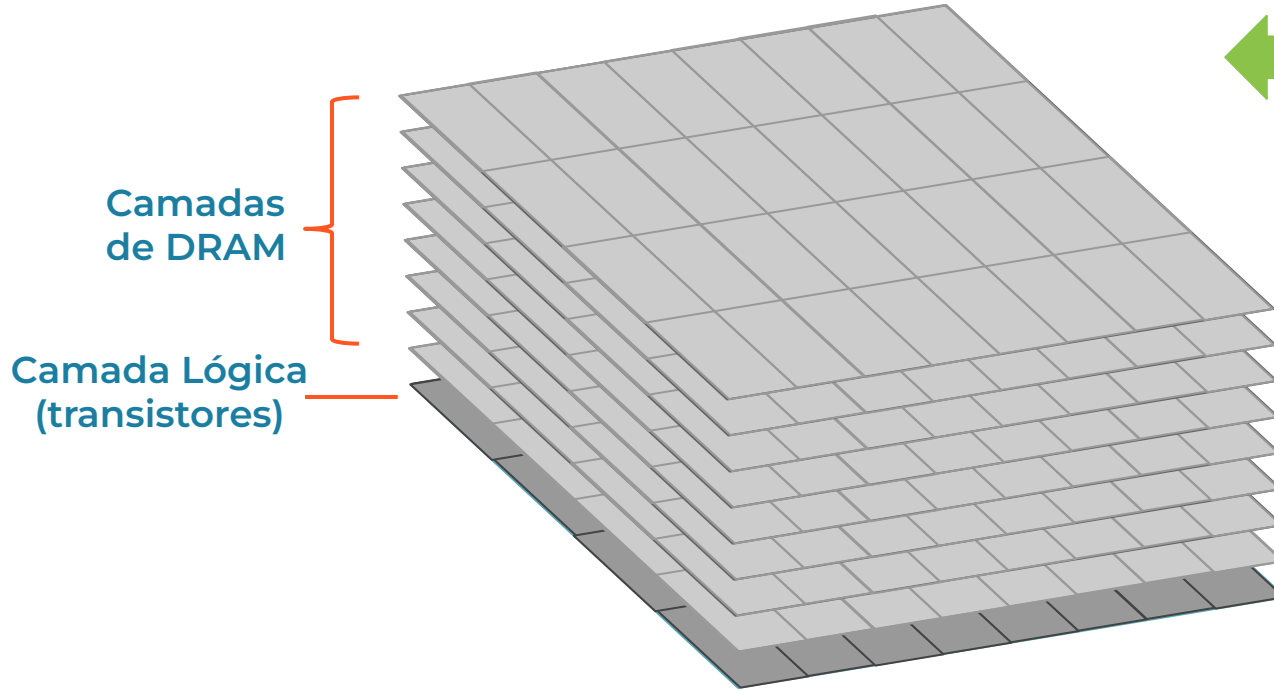
**Hybrid Memory
Cube**

Arquitetura do HMC

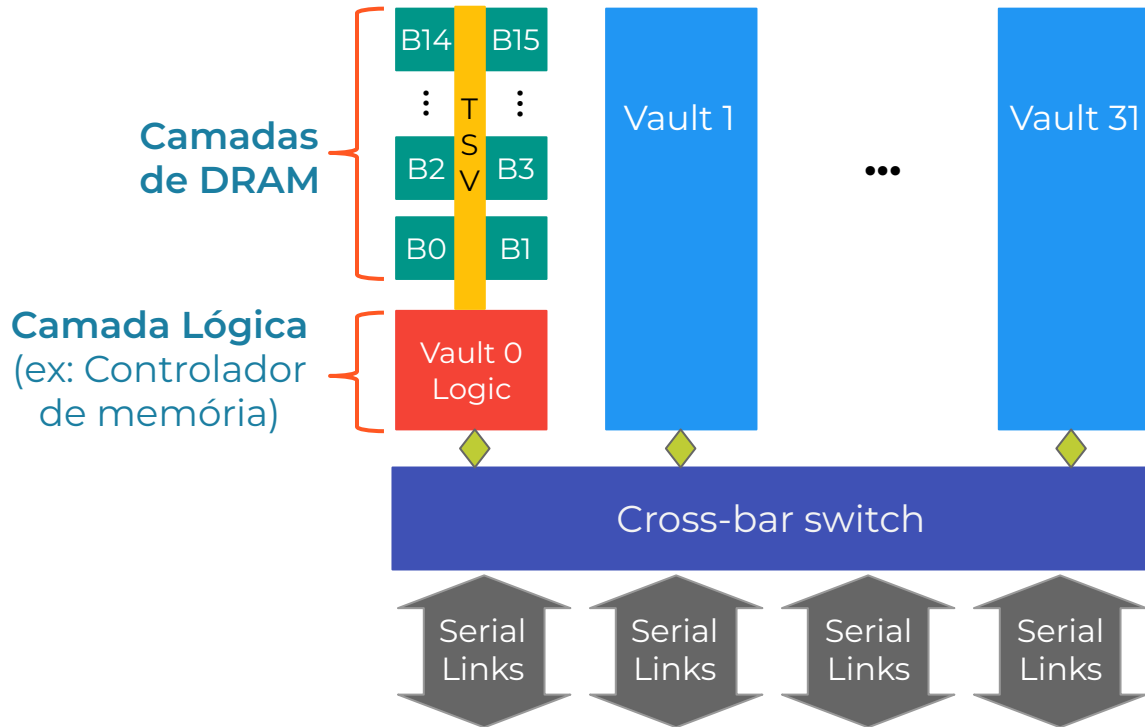


Arquitetura do HMC

HMC Module
(Memory)



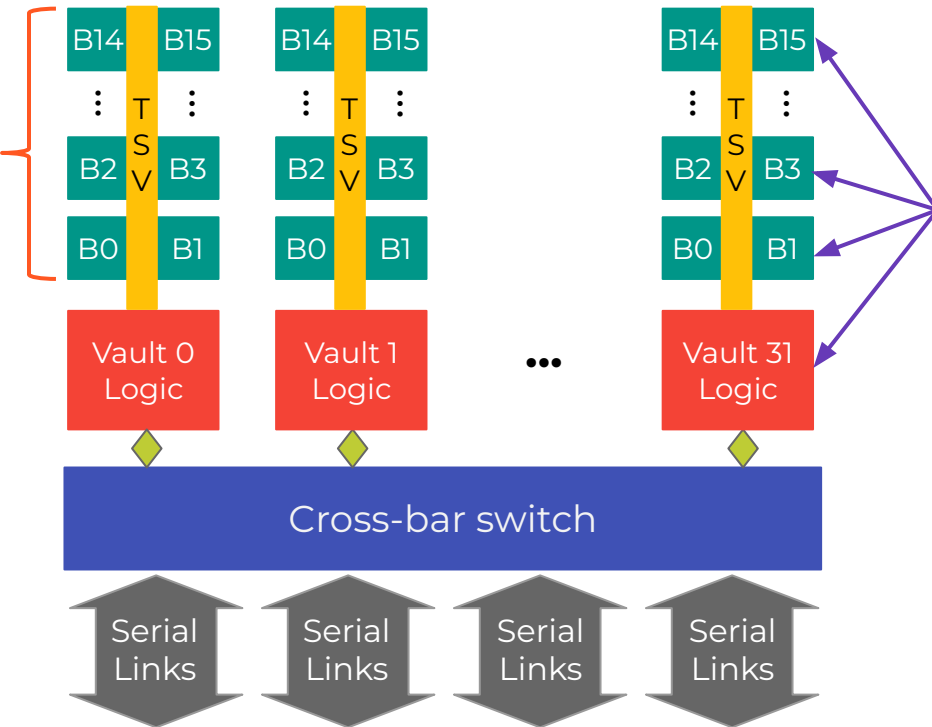
Arquitetura do HMC



Camadas de DRAM e lógica divididas em: **32 partições lógicas**

Arquitetura do HMC

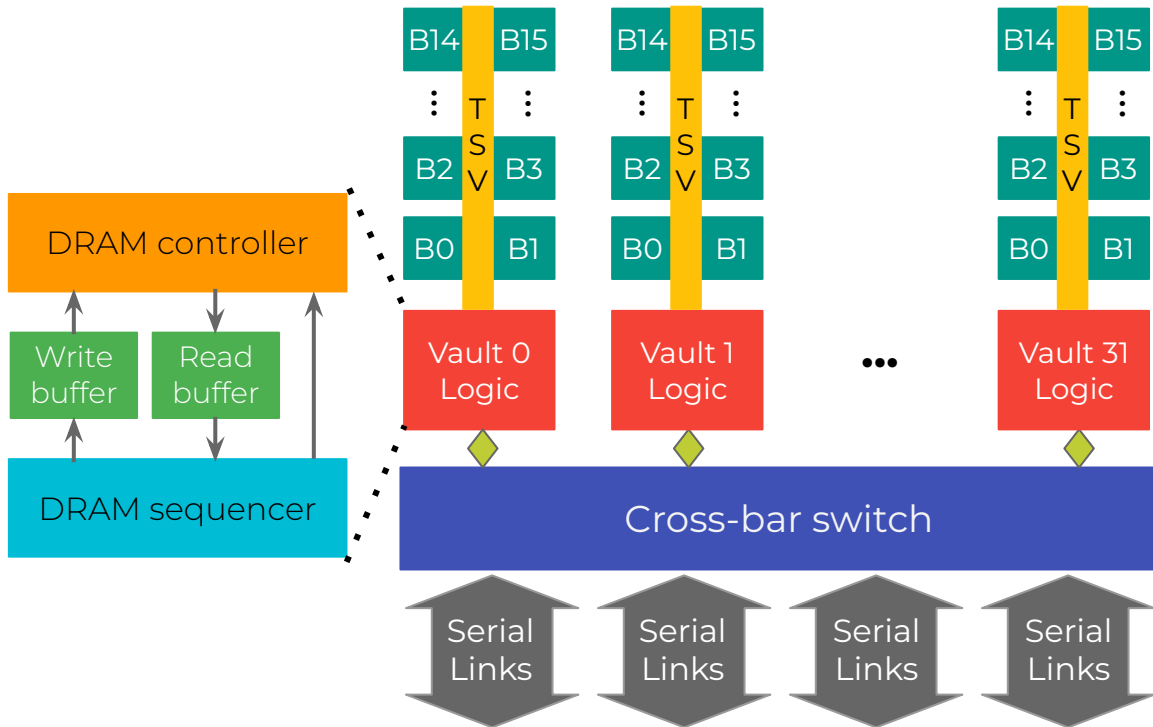
Múltiplos bancos independentes por vault



3D-Stacked Interconectado por TSVs

Camadas empilhadas em formato 3D utilizando **TSVs**

Arquitetura do HMC

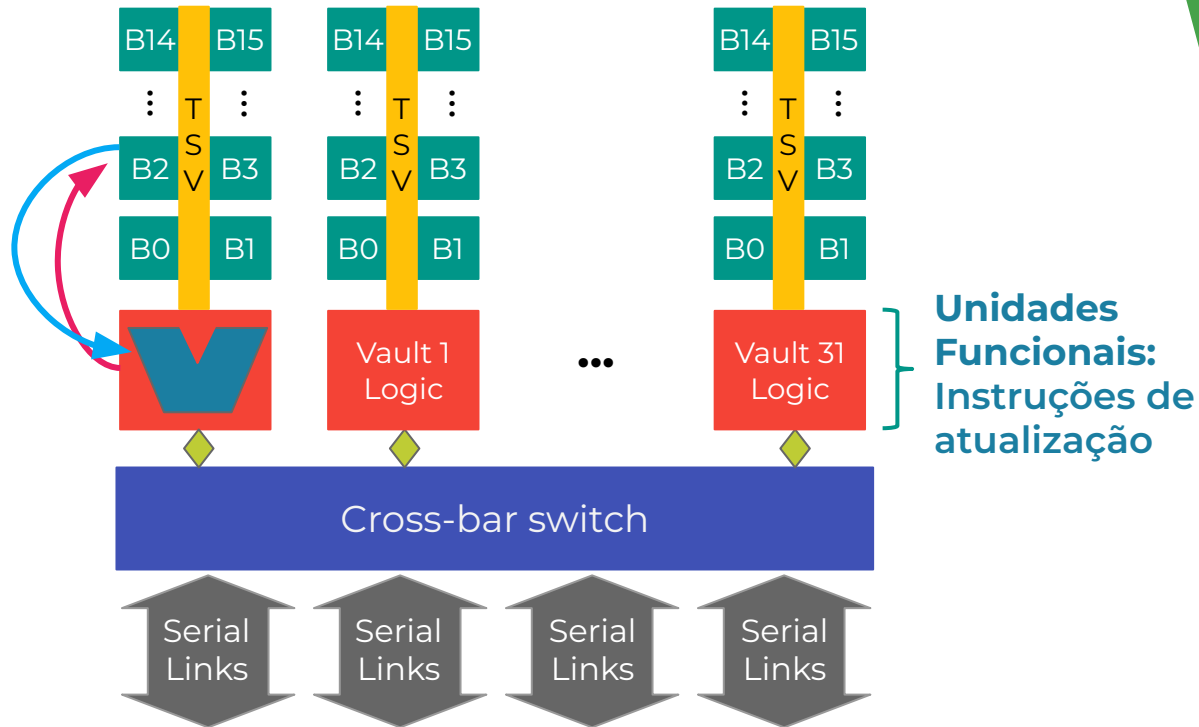


Camada Lógica
com **controlador
de memória
DRAM**, que
envia sinais
(ex: CAS, RAS)

Comparação

- ▶ HMC tem largura de banda 25% maior que HBM2:
 - ▷ HMC atinge 320 Gb/s;
 - ▷ HBM2 atinge 256 Gb/s;
- ▶ HMC utiliza um conjunto de instruções simples e bem documentado.

Arquitetura do HMC



Camada Lógica
suporta
**instruções
atômicas**
(ex: **ADDi**,
SWP)

Conjunto de instruções HMC

- ▶ Novas arquiteturas de Processamento em Memória (PIM) estão surgindo;
- ▶ HMC utiliza um conjunto de instruções simples e bem documentado;
- ▶ **Precisamos de uma forma de avaliar e prototipar novos conjuntos de instruções.**

Como avaliar novos conjuntos de instruções

- ▶ Comprando um HMC:
 - ▷ Precisamos do hardware e compilador;
 - ▷ Nenhuma ISA nova suportada;
- ▶ Prototipar usando um FPGA ou ASIC:
 - ▷ Consome muito tempo, caro e propenso a erros;

Como avaliar novos conjuntos de instruções

- ▶ Modelagem analítica:
 - ▷ Precisão baixa para sistemas complexos;
- ▶ **Simulação**
 - ▷ **Parece a melhor opção!**

3.

Simulação: Visão Geral


Simuladores de arquiteturas

- ▶ Full system:
 - ▶ Executa o binário + SO;
 - ▶ Precisamos de um compilador confiável ou escrever o binário manualmente;
- ▶ Trace driven:
 - ▶ Precisa do traço de simulação;
 - ▶ Traço contém o comportamento dinâmico;
 - ▶ Como escrever os traços? **Manualmente?**

Architectural Simulators

- ▶ Full system:
 - ▶ Executa o binário + SO;
 - ▶ Precisamos de um compilador confiável ou escrever o binário manualmente;
- ▶ Trace driven:
 - ▶ Precisa do traço de simulação;
 - ▶ Traço contém o comportamento dinâmico;
 - ▶ Como escrever os traços? **Manualmente?**

Consome
muito
tempo e
pode gerar
muitos
erros!!!



Nossa **proposta**

Gerar os traços de simulação automaticamente

- ▶ Escrevendo código de alto nível (C ou C++);
- ▶ Usando a biblioteca Intrinsic-HMC;
- ▶ Compilando e executando na arquitetura x86;

Nossa **proposta**

Gerar os traços de simulação automaticamente

- ▶ Escrevendo código de alto nível (C ou C++);
- ▶ Usando a biblioteca Intrinsic-HMC;
- ▶ Compilando e executando na arquitetura x86;

Nosso mecanismo converte funções x86 da biblioteca em instruções HMC.

4.

Simuladores PIM

Visão geral dos
simuladores
atuais

Gem5 + SMC Simulation Environment

- ▶ Smart Memory Cube é um módulo inserido no simulador Gem5:
 - ▷ Melhora a capacidade da camada lógica do HMC;
 - ▷ Configuração do estado-da-arte;

Precisa do binário para ser executado
Não alcança as especificações do HMC

CasHMC

- ▶ Simulador com precisão de ciclos para HMC;
- ▶ A implementação segue exatamente a arquitetura especificada do HMC;

Modela o HMC sem a capacidade de PIM

CLAPPS

- ▶ Precise Cycle Parallel PIM Simulator;
- ▶ Providencia um modelo mais preciso para arquiteturas PIM;

Não tem forma eficiente de providenciar workloads de entrada para HMC

Precisa de um simulador de processador para obter resultados realísticos

SiNUCA

- ▶ Simulador dirigido a traços;
- ▶ Com precisão de ciclos;
- ▶ Baseado na arquitetura x86;
- ▶ Facilmente extensível;
- ▶ Modela o HMC e sua capacidade PIM;

Precisa de traços de simulação

Resumo

- ▶ Todos os simuladores precisam do binário ou do traço de simulação;
- ▶ SiNUCA tem uma ferramenta de geração de traços:
 - ▷ É usado pelo grupo;
 - ▷ Mais conveniente;

SiNUCA foi utilizado neste trabalho

Código C/C++



Linkagem e
compilação



Código
binário
executável



Traços de
Simulação

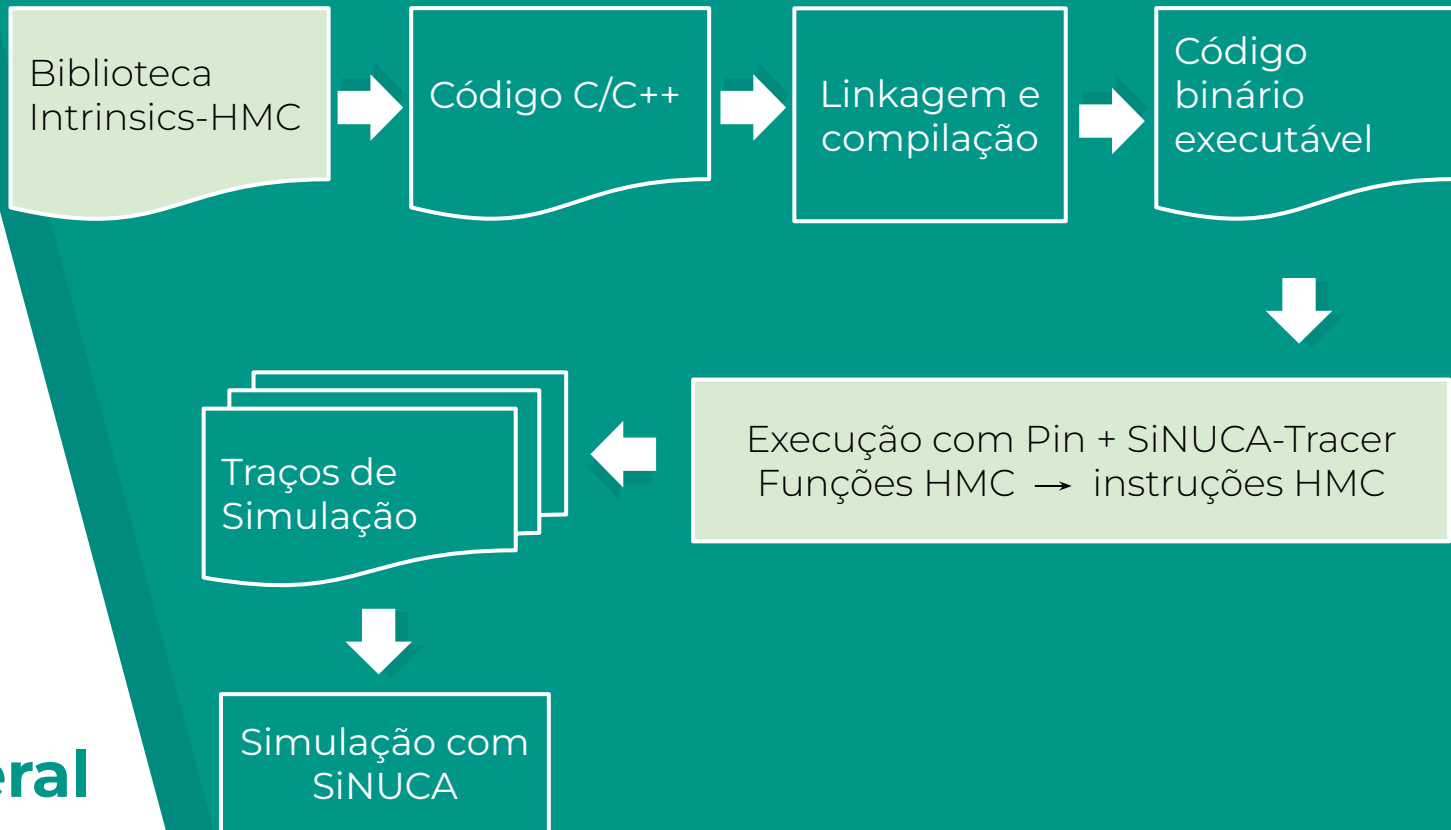


Execução com Pin + SiNUCA-Tracer

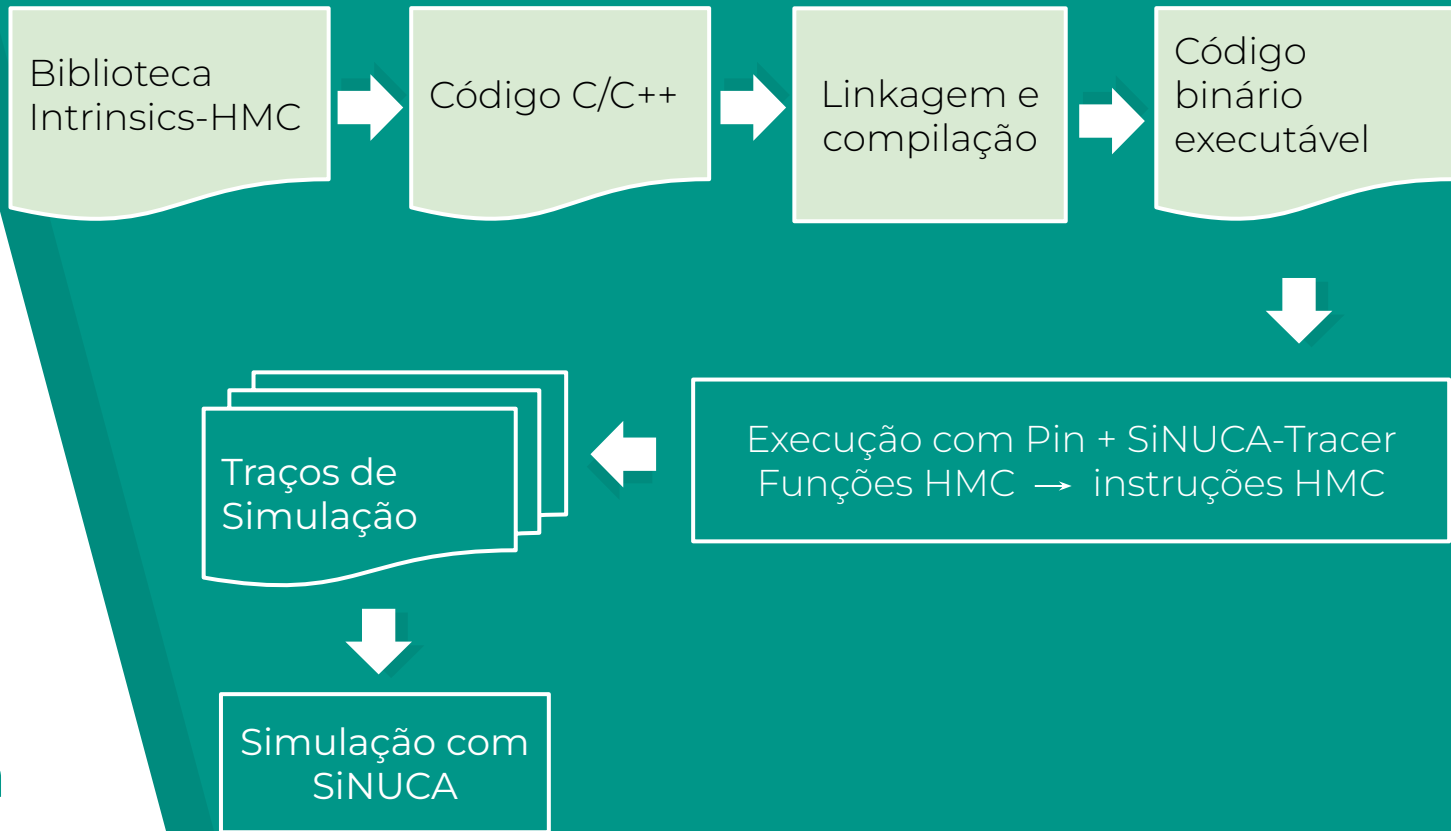


Simulação com
SiNUCA

SiNUCA
antes das
modificações



Visão geral da proposta



**Primeira
etapa**

Intrinsics da Intel

- ▶ Código em alto nível (C/C++);
- ▶ Funções aritméticas, matemáticas, binárias, etc;
- ▶ Otimização:
 - ▷ Instruções vetorizadas (SIMD);
 - ▷ SSE, AVX, AVX2

Especificação das operações HMC

- ▶ Operações tradicionais de leitura/escrita;
- ▶ Aritméticas;
- ▶ Binárias;
- ▶ Booleanas;
- ▶ Comparação.

Especificação das operações HMC

- ▶ Operações tradicionais de leitura/escrita;
- ▶ Aritméticas;
- ▶ Binárias;
- ▶ Booleanas;
- ▶ Comparação.

Ex.:

hmc_nor (ender_memória) \$imediato

Endereço de memória
simples (atualização)

Imediato simples

Intrinsics-HMC

- ▶ Escrita na linguagem C++;
- ▶ Reproduz o comportamento das instruções HMC;
- ▶ Tipo de dados padronizados;
- ▶ Inspiradas nas intrinsics da Intel;

Chamada de função

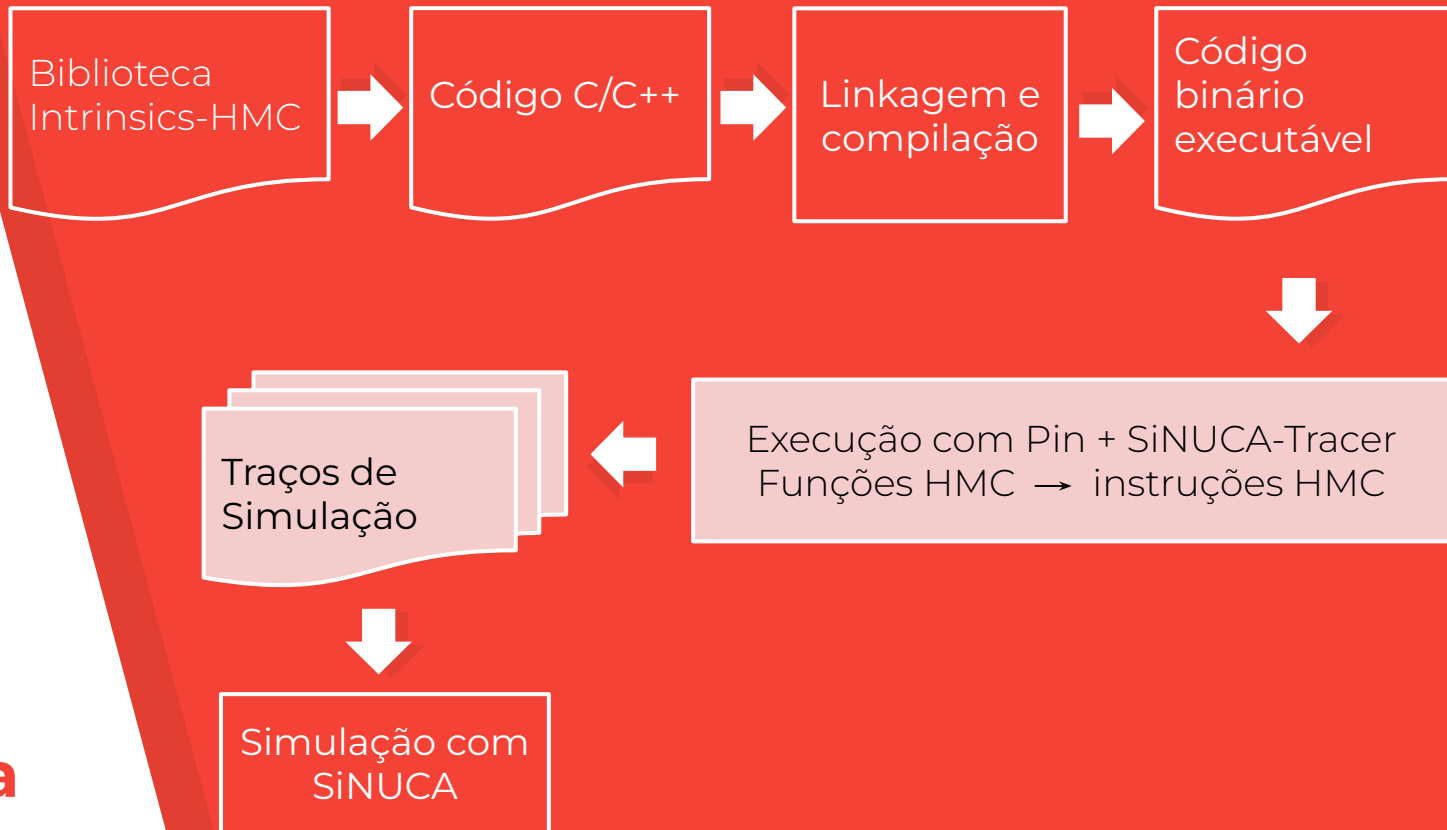
Intrinsics-HMC

```
#include "../hmc.hpp"
```

```
int main(int argc, char *argv[]){  
    uint128_t mem_ret;  
    mem_ret = _hmc128_nor_s(&mem_op, imm_op);  
}
```

Código fonte Intrinsics-HMC

```
__h128lll1 _hmc128_nor_s  
  (__h128lll1 *mem_op, __h128lll1 imm_op){  
    __h128lll1 r = *mem_op;  
    *mem_op = ~(mem_op | imm_op);  
    return r;  
}
```



Segunda etapa

5.

SiNUCA-Tracer & Pin (intel)

Gerador de traços &
instrumentador
binário

Pin

- ▶ Ferramenta de instrumentação e análise da Intel;
- ▶ Provê rotinas para análise:
 - ▷ Pin tools -> **SiNUCA-Tracer**;
 - ▷ C/C++;
 - ▷ Usa código binário.

Instrumentação Pin

- ▶ Por imagem;
- ▶ Por seção;
- ▶ Por rotina;
- ▶ Por bloco básico;
- ▶ Por instrução.

SiNUCA-Tracer

- ▶ Gera os traços para o SiNUCA
- ▶ Converte o conjunto de instruções x86 para o conjunto de instruções do SiNUCA;

Código assembly com **Intrinsics-HMC**

Chamada de
função:
Bloco de
instruções x86

```
1.  lw
2.  add
3.  or
4.  sw
5.  add
6.  add
7.  or
8.  CALL hmc_function ...
9.  nop
10. lw
11. beq
```

```
1.  add
2.  add
3.  and
4.  sw
5.  lw
6.  lw
7.  RET hmc_function ...
```

Intercepta geração de código x86

```
1.  lw
2.  add
3.  or
4.  sw
5.  add
6.  add
7.  or
8.  CALL hmc_function ...
9.  nop
10. lw
11. beq
```

```
1.  add
2.  add
3.  and
4.  sw
5.  lw
6.  lw
7.  RET hmc_function ...
```

Impede o SiNUCA-Tracer de gerar código x86 durante a chamada de função HMC

Inserir código assembly do HMC

```
1.  lw
2.  add
3.  or
4.  sw
5.  add
6.  add
7.  or
8.  CALL hmc_function ...
9.  nop
10. lw
11. beq
```

```
1.  add
2.  add
3.  and
4.  sw
5.  lw
6.  lw
7.  RET hmc_function ...
```

hmc_atomic_instruction

Dependencia entre registradores x86

```
1.  lw
2.  add
3.  or
4.  sw
5.  add
6.  add
7.  or
8.  CALL hmc_function ...
9.  nop
10. lw
11. beq
```

Registadores de entrada (leitura)

```
1.  add
2.  add
3.  and
4.  sw
5.  lw
6.  lw
7.  RET hmc_function ...
```

Registadores de saída (escrita)

Dependencia entre registradores x86

```
1.  lw
2.  add
3.  or
4.  sw
5.  add
6.  add
7.  or
8.  hmc_instruction
   read/write regs.
9.  nop
10. lw
11. beq
```

Registadores de entrada (leitura)

Mantém todas as dependências entre registradores x86 de entrada/saída nas instruções HMC

Registadores de saída (escrita)

6.

Resultados

Aplicações

- ▶ Algoritmos de banco de dados:
 - ▷ Join;
 - ▷ Select Scan;

Comportamento de streaming de dados
Adequado para explorar capacidade PIM

Join - Laços encadeados

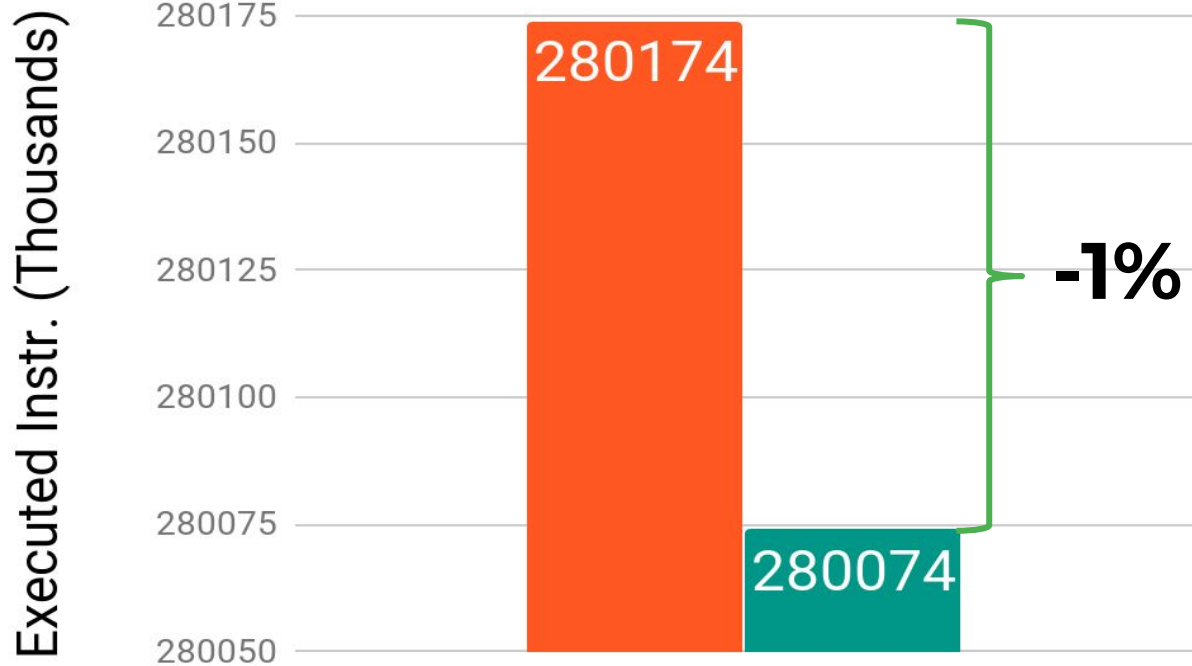
- ▶ Cruzamento de duas tabelas procurando por valores iguais;
- ▶ Usa 2 laços encadeados.

Select Scan - Busca em vetor

- ▶ Dado um valor de entrada, seleciona todos os valores iguais em uma tabela;
- ▶ Como uma busca em vetor;

Resultados - Instruções executadas

Selection Scan



■ x86
■ HMC

Espera-se
redução do
número de
instruções
executadas

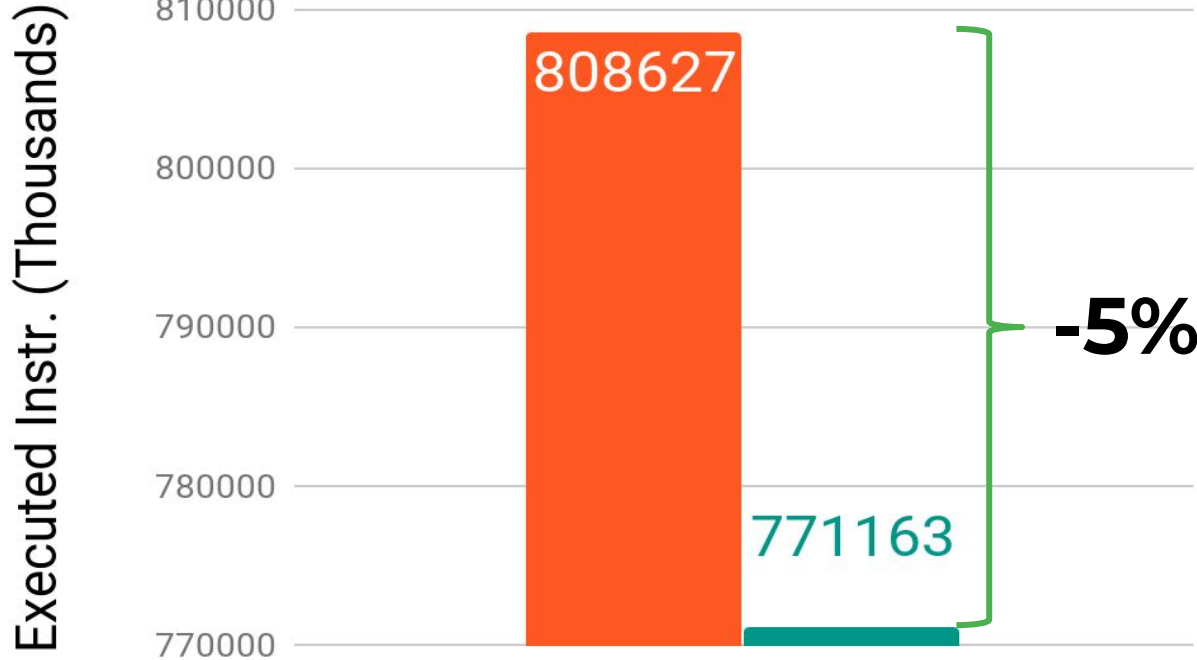
1 chamada de
função =
1 Instrução

Espera-se
redução do
número de
instruções
executadas

1 chamada de
função =
1 Instrução

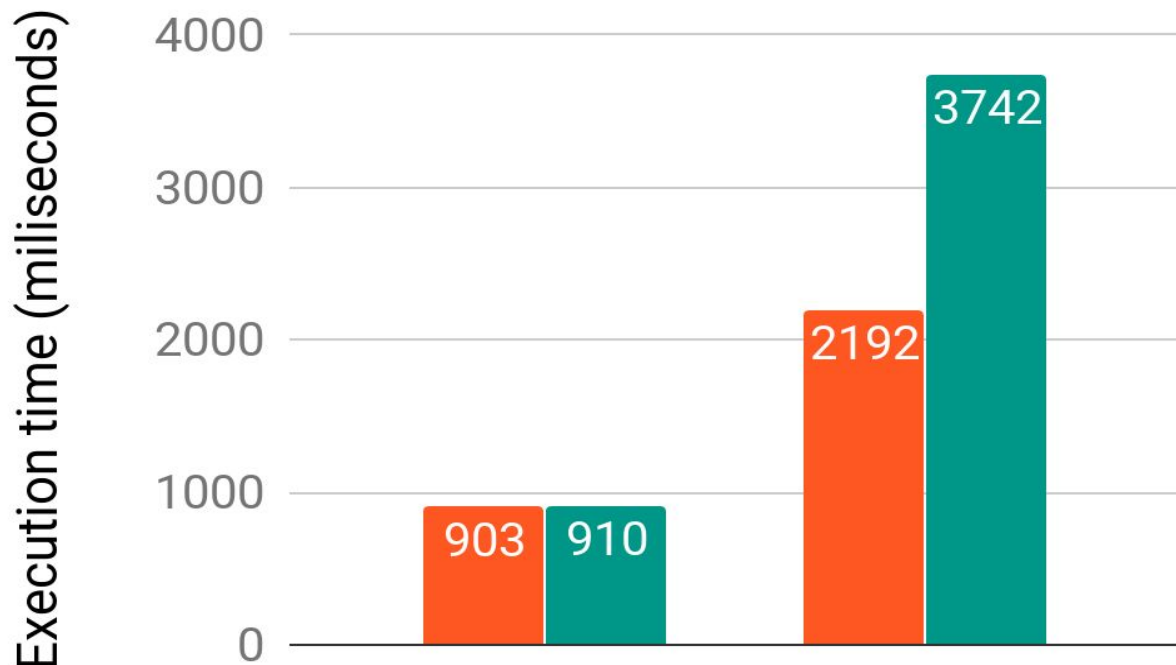
Resultados - Instruções Executadas

Join



Resultados - Tempo de execução

Selection Scan / Join



■ x86
■ HMC

Aumento do tempo de execução

Conjunto de instruções do HMC ainda é limitado



7.

Conclusão

Conclusão

- ▶ Podemos gerar traços de simulação automaticamente;
- ▶ Fácil de extender a biblioteca `intrinsic` para suportar novos conjuntos de instruções;

Conclusion

- ▶ Permite avaliações de novas arquiteturas PIM de forma rápida;
- ▶ Agora, é possível a avaliação de aplicações complexas com PIM.

Geração de traços de simulação para instruções de processamento em memória

<https://github.com/AlineS/intrinsics-hmc>

Obrigada!

Aline S. Cordeiro -
alinesantanacordeiro@gmail.com

Prof. Dr. Marco A. Z. Alves