SSC0951 – Desenvolvimento de Código Otimizado

2ª aula – Profiling

Profa. Sarita Mazzini Bruschi

sarita@icmc.usp.br

Conteúdo basedo em:

Derrube Todos os Recordes de Ganho de Desempenho Otimizando seu Código

Sherlon Almeida da Silva, Matheus S. Serpa, Claudio Schepke

Minicurso ERAD-RS 2017

Como obter desempenho de uma aplicação

- Lembrando que (primeira aula):
  - otimização do código == melhorar o desempenho do código

- O que é necessário para melhorar o desempenho?
  - Conhecer profundamente a arquitetura
  - Utilizar boas práticas de programação

Relembrando alguns conceitos

- ILP: Instruction Level Paralellism
- IPC: Instruction Per Clock
- CPI: Clock Per Instruction
- Pipeline
  - Parada do pipeline (*Stalls*)
  - Predição de desvio
  - Dependência de dados
- Cache

Profiling (Perfilador)

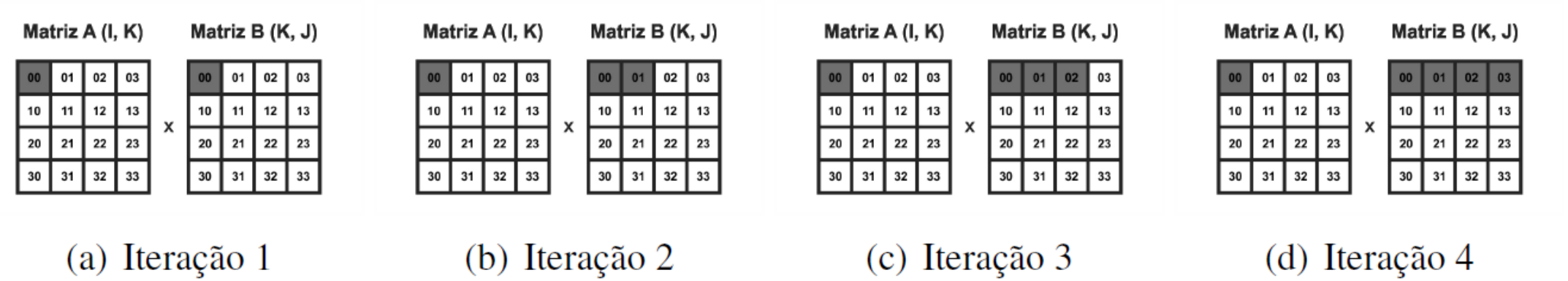
- Profiling: aquisição de informações sobre o comportamento de um programa, especialmente no uso dos recursos computacionais (memória, CPU, etc.)
- Dois tipos:
  - Instrumentação: compilador insere instruções no código binário para que informações sejam coletadas
  - Amostragem: o programa é interrompido de tempos em tempos para coleta das informações
- Ferramentas:
  - Gprof (vinculada ao gcc)
  - PERF
  - Intel PCM (*Performance Counter Monitor*)

Análise de Desempenho

- A coleta das métricas com uma única execução não tem validade estatística
- É necessário que o programa seja executado várias vezes para a coleta das métricas e então o cálculo da média e do intervalo de confiança

Loop Interchange

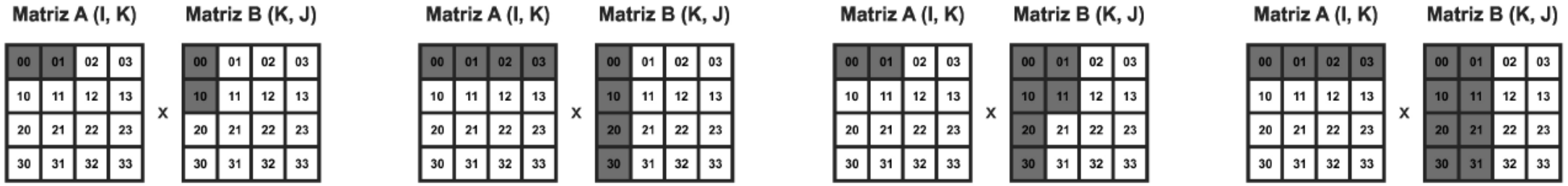
```
1 for(i = 0; i < SIZE; i++)
2         for(k = 0; k < SIZE; k++)
3                 for(j = 0; j < SIZE; j++){
4                         C(i, j) += A(i, k) * B(k, j);
5 }
```



(a) Iteração 1 (b) Iteração 2 (c) Iteração 3 (d) Iteração 4

Loop Unrolling

```
1 for(i = 0; i < SIZE; i++)
2         for(j = 0; j < SIZE; j++)
3                 for(k = 0; k < SIZE; k+=2){
4                         C(i, j) += A(i, k) * B(k, j);
5                         C(i, j) += A(i, k+1) * B(k+1, j);
6 }
```



(a) Iteração 1 (b) Iteração 2 (c) Iteração 3 (d) Iteração 4

Multiplicação de matrizes

- Fazer o código de multiplicação de matrizes, separando por funções
- Executar o perf para obter as seguintes métricas:
  - L1-dcache-load
  - L1-dcache-load-misses
- Alterar o código utilizando a técnica Loop Interchange
  - Repetir a avaliação de desempenho
- Alterar o código utilizando a técnica Loop Unrolling
  - Repetir a avaliação de desempenho

Onde conseguir informações sobre a ferramenta *perf*

- http://www.brendangregg.com/perf.html
  - Seção 2 (One-Liners), counting events