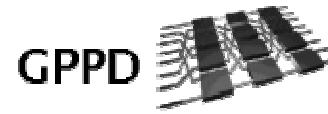


# Um breve estudo da adição de um quarto nível de cache em processadores multicore

**Carlos Eduardo Benevides Bezerra**

Orientador: Prof. Dr. Cláudio F. R. Geyer

**Porto Alegre, 4 de junho de 2009**



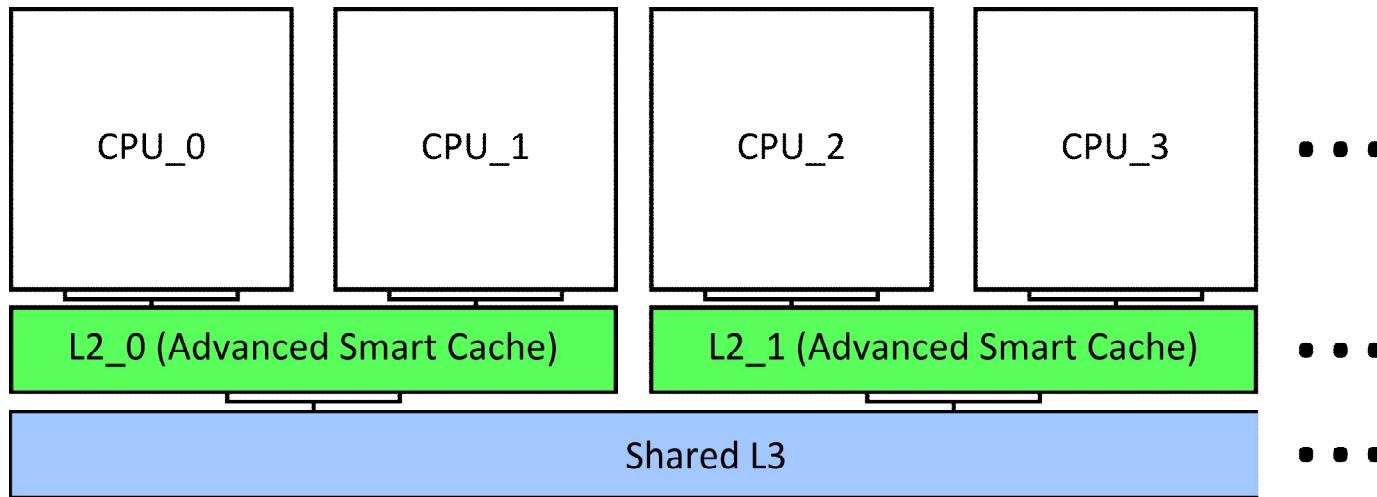
# Sumário

- **Contexto**
- **Motivação**
- **Proposta**
- **Modelagem**
- **Simulação**
- **Resultados**
- **Conclusões**

# Contexto

- Memórias cache aceleram a recuperação de dados da memória
- Aplicações cada vez mais pesadas, com maior volume de dados
- Maior memória principal para comportar os dados destas aplicações
- Processadores com maior freqüência (GHz)
- Maior número de cores (1, 2, 3, 4, 6, 8 ... Tera-Scale)
- As caches devem aumentar seu tamanho proporcionalmente

# Intel Dunnington (Xeon multicore)



- Dunnington (Xeon) é uma arquitetura multicore
- Cada memória L2 é compartilhada por um par de cores
  - Disponibiliza dados para ambos os cores (mais espaço)
  - Pode economizar banda do barramento
  - Menor esforço com manutenção de coerência
  - Utiliza “Advanced Smart Cache”
  - Porém, cache maior implica tempo de acesso maior
- Uma memória cache L3 é compartilhada por todos os cores

# Motivação

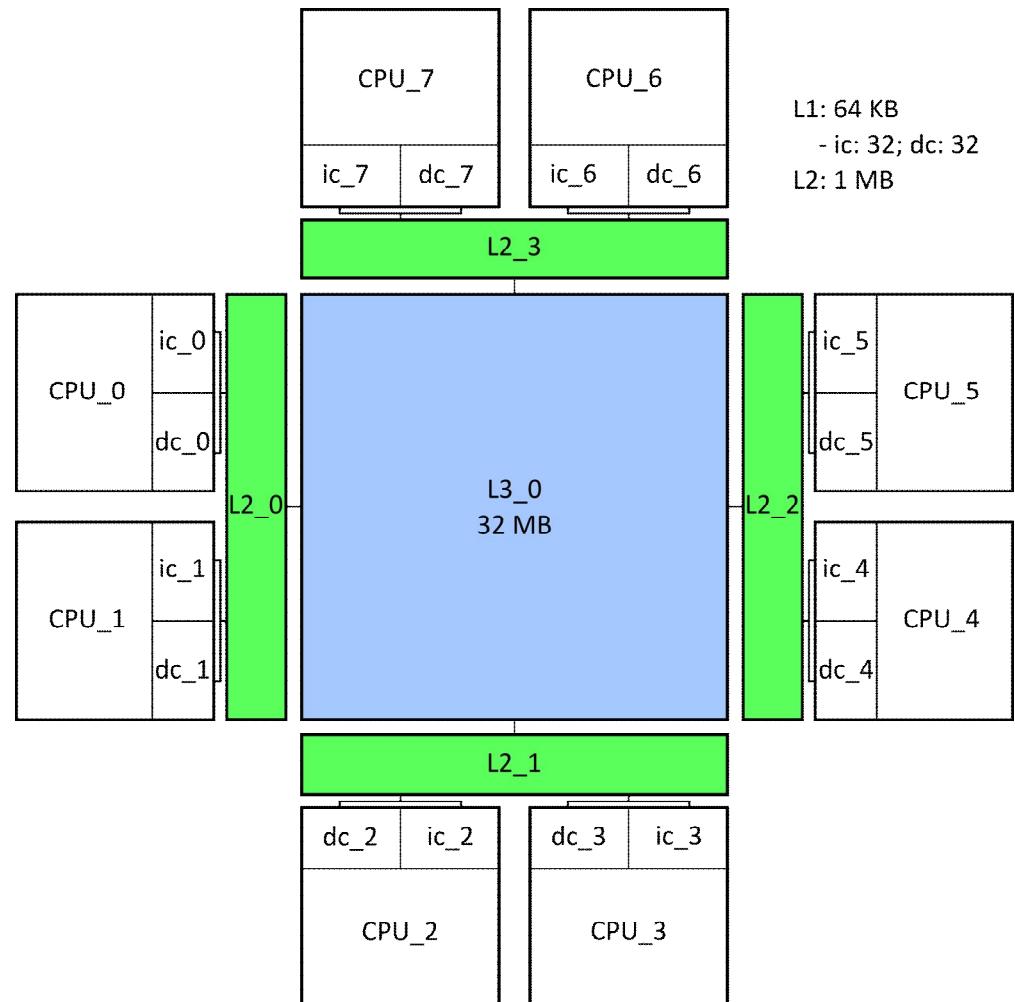
- Pede-se mais memória cache, porém:
  - Memória maior implica maior tempo de acesso
  - Perdas devido à contenção pelo acesso à cache, quando compartilhada por muitos cores
  - Adicionar memória cache custa caro
- Podem ser adicionadas memórias mais distantes do núcleo
  - São acessadas com menor freqüência
  - O problema da contenção é reduzido
  - O tempo de acesso maior gera um menor impacto sobre o desempenho
  - Ao invés de aumentar o tamanho da cache  $L<X>$ , pode ser adicionado uma cache  $L<X+1>$

# Proposta

- Baseando-se na arquitetura Dunnington, pretende-se investigar:
  - Benefício de se utilizar uma memória cache L3 grande (32 MB)
  - Benefício de se dividir a L3 em dois níveis (L3 e L4: 8 + 8 + 16)
  - Ganho ao entrelaçar a hierarquia de cache
    - Diminuir a maior profundidade mínima de cache que qualquer par de cores precisa buscar para compartilhar memória

# Modelagem – L123

- Arquitetura L123
  - Modelada sobre o SunFire
  - 3 níveis de memória cache
  - L2 compartilhada por cada par de cores
  - L3 compartilhada por todos
- Parâmetros
  - Memória principal: 1 GB
  - L1:
    - Instruções: 32 KB
    - Dados: 32 KB
    - Latência: 2 ciclos
  - L2:
    - 1 MB
    - Latência: 5 ciclos
  - L3
    - 32 MB
    - Latência: 30 ciclos



# Modelagem – L1234

- Arquitetura L1234

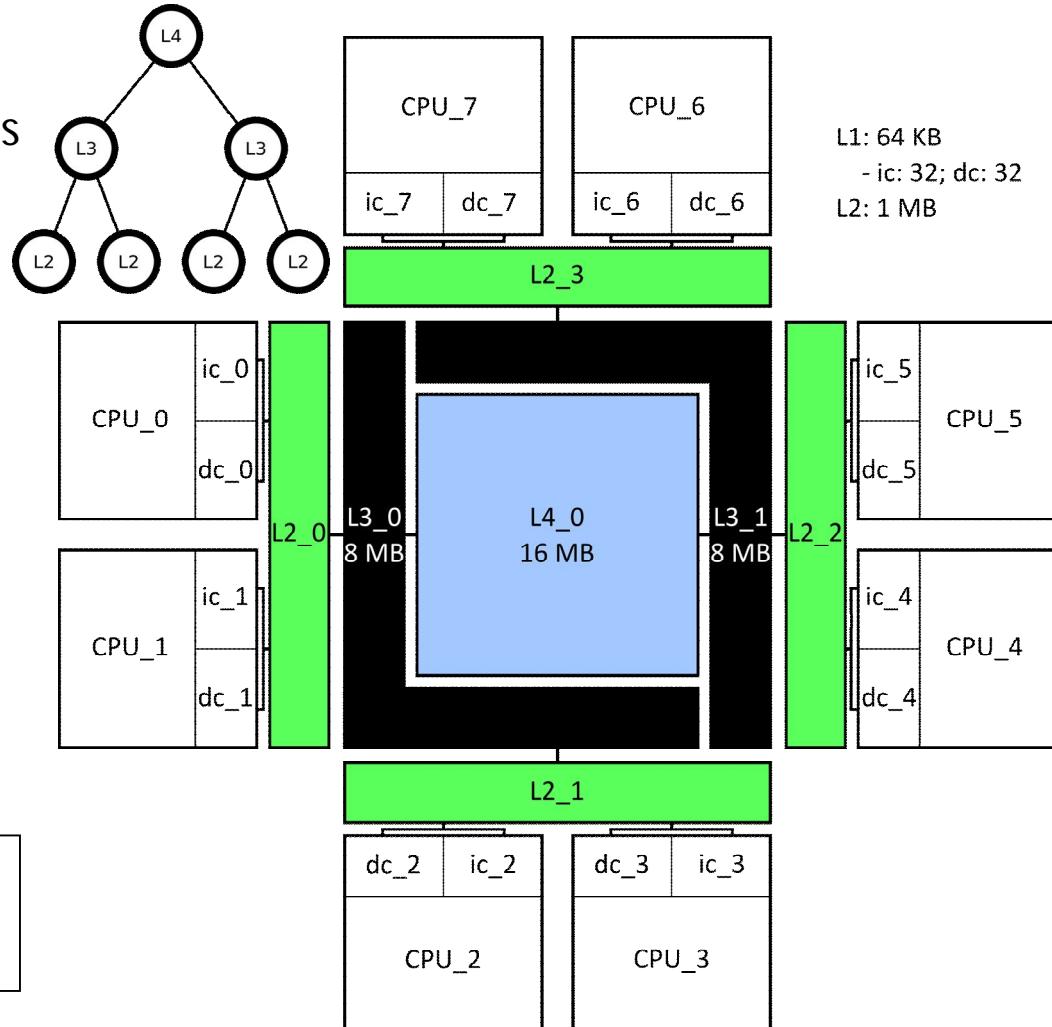
- 4 níveis de memória cache
- L2 compartilhada por cada par de cores
- L3 compartilhada por cada 4 cores (a cada 2 caches L2)
- L4 compartilhada por todos

- Parâmetros

- Memória principal: 1 GB
- L1:
  - Instruções: 32 KB
  - Dados: 32 KB
  - Latência: 2 ciclos
- L2:
  - 1 MB
  - Latência: 5 ciclos
- L3
  - 8 MB
  - Latência: 15 ciclos
- L4
  - 16 MB
  - Latência: 30 ciclos

**Simular:**  
Tamanho menor  
Menor contenção

**Tamanho menor**  
**Maior distância dos cores**



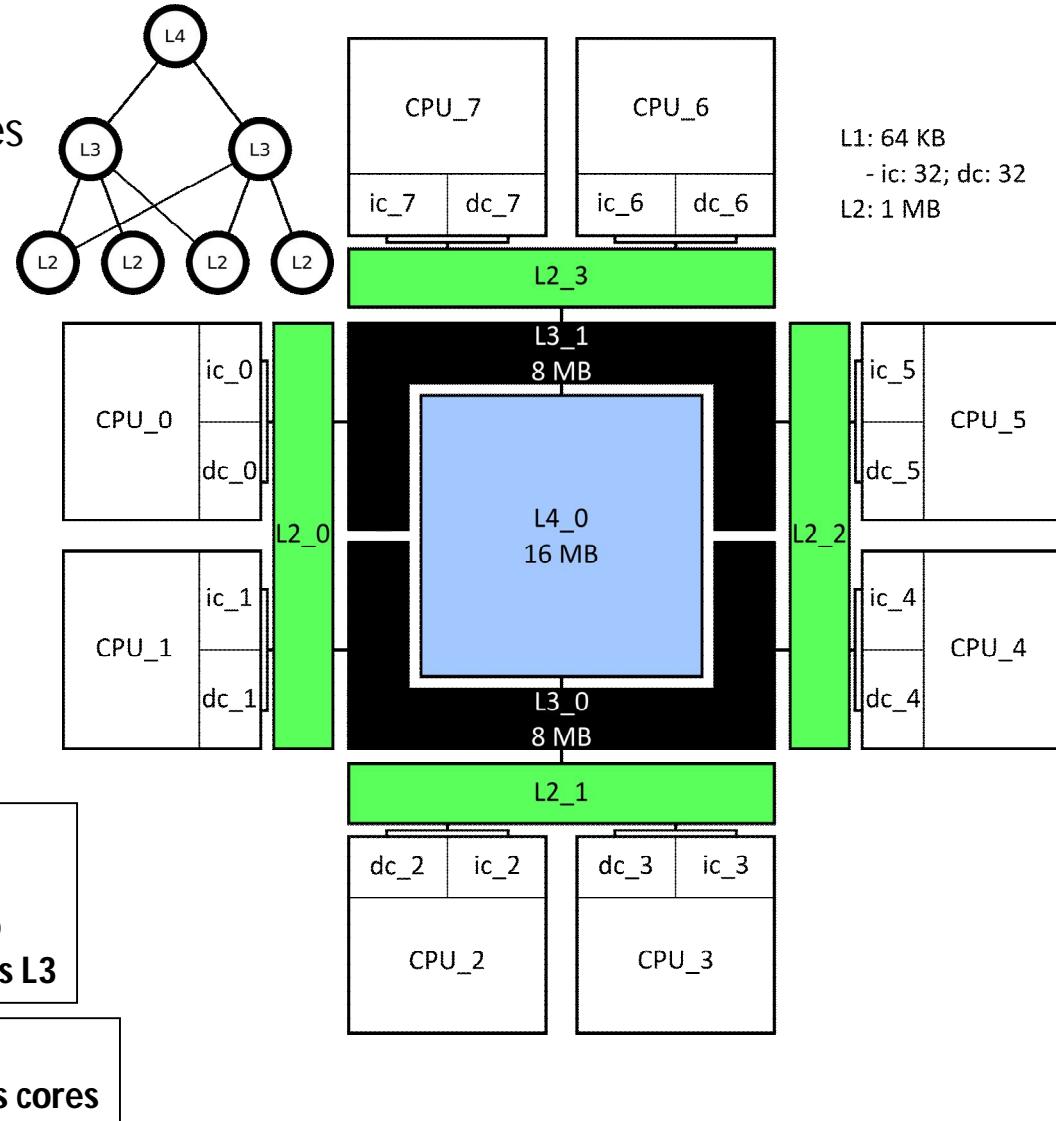
# Modelagem – InterleavedL4

- Arquitetura **InterleavedL4**

- 4 níveis de memória cache
- L2 compartilhada por cada par de cores
- L3 compartilhada por cada 4 cores
- L4 compartilhada por todos

- Parâmetros

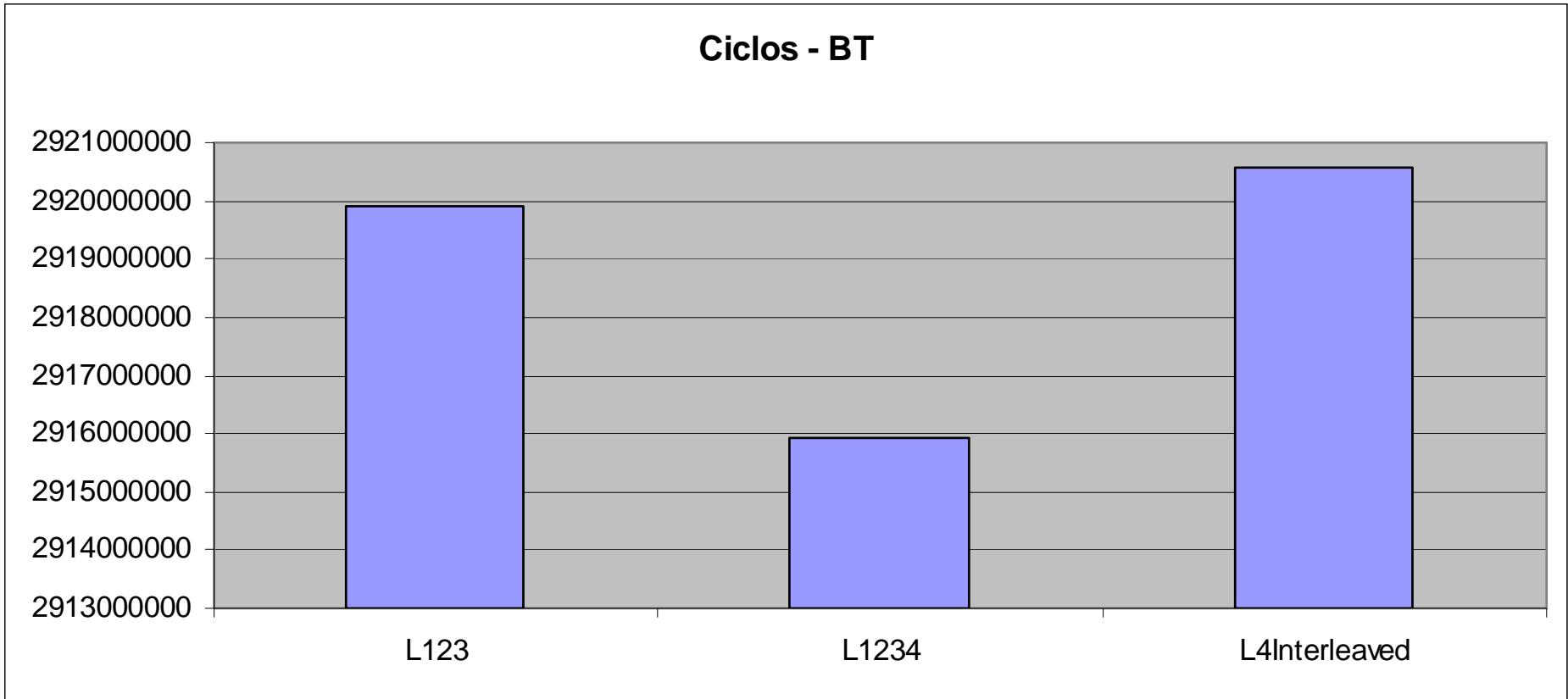
- Memória principal: 1 GB
- L1:
  - Instruções: 32 KB
  - Dados: 32 KB
  - Latência: 2 ciclos
- L2:
  - 1 MB
  - Latência: 5 ciclos
- L3
  - 8 MB
  - Latência: 20 ciclos
- L4
  - 16 MB
  - Latência: 30 ciclos



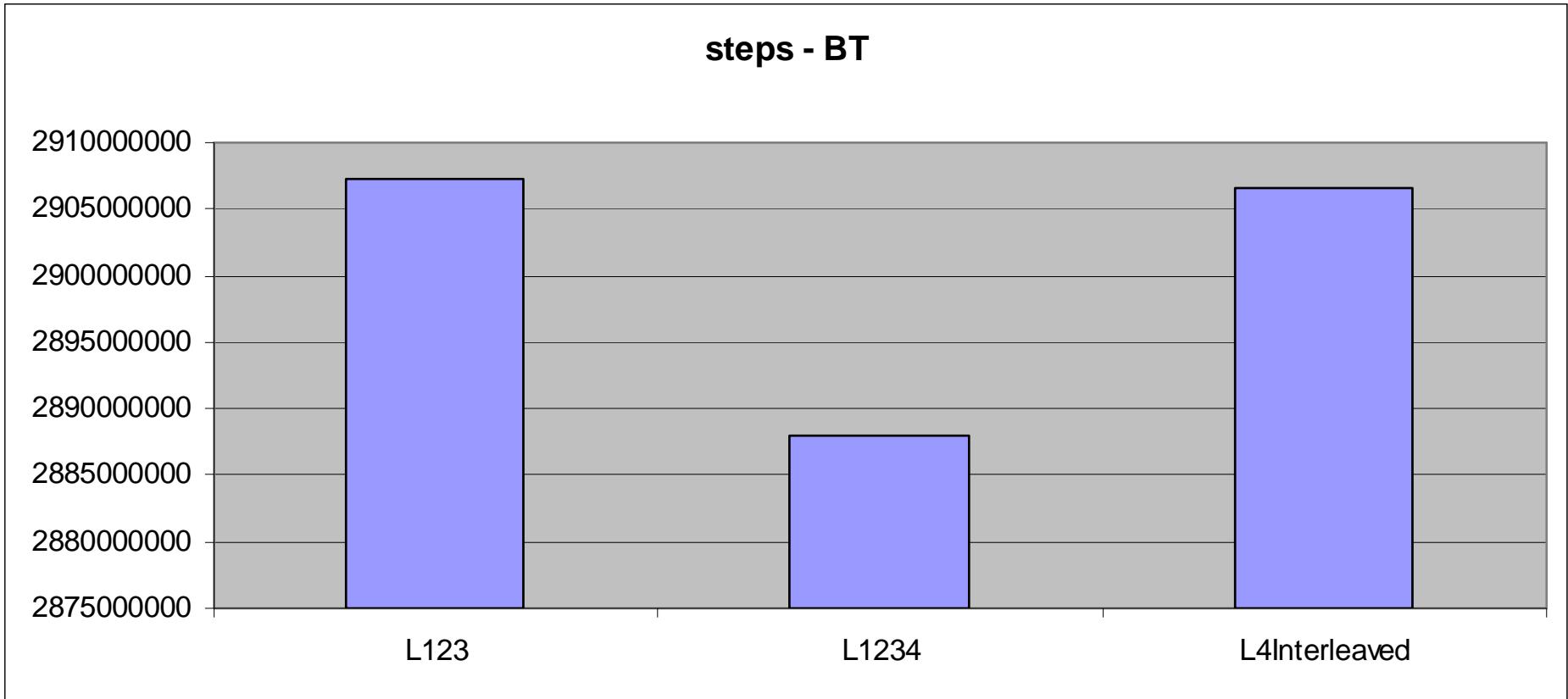
# Resultados

- Observações:
  - Repetições 3 ( $3\text{reps} * \sim3\text{horas} * 3\text{arqs} * 2\text{programas} = \sim54\text{ horas, ou 18 simulações}$ )
  - Execuções em seqüência (não a partir do mesmo ponto)
  - Desvios padrões do número de ciclos: < 1'000 (0,0001%)
  - Desvios padrões do número de steps: < 1'000'000 (0,1%)

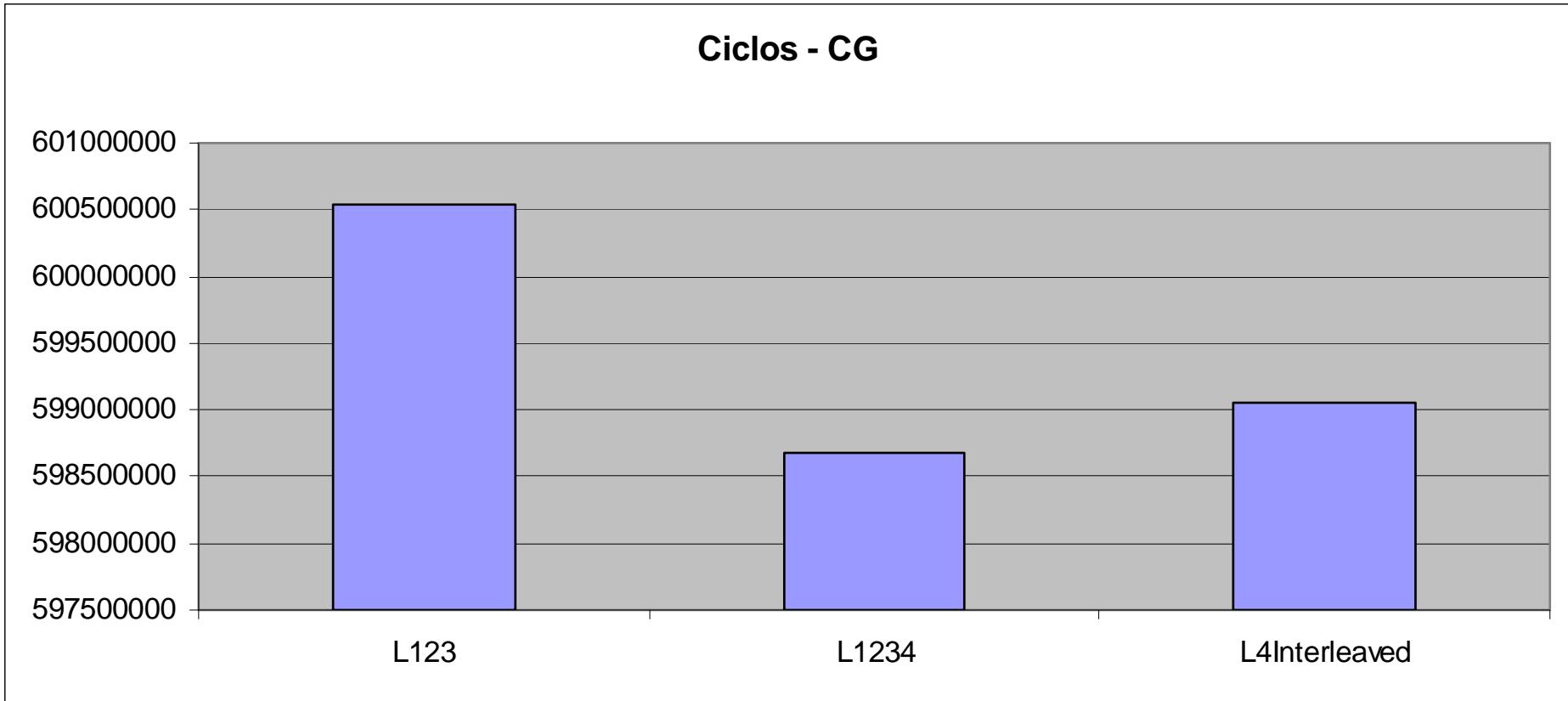
# Resultados



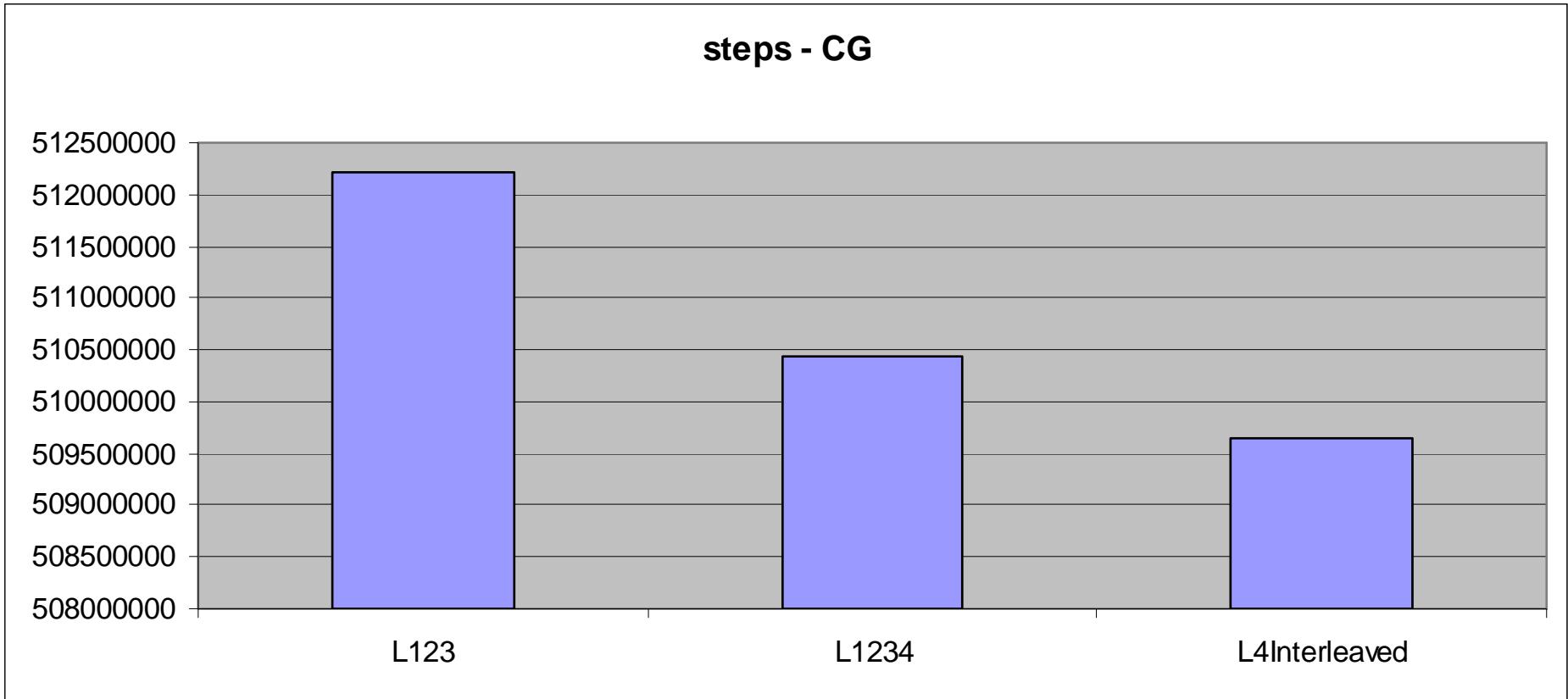
# Resultados



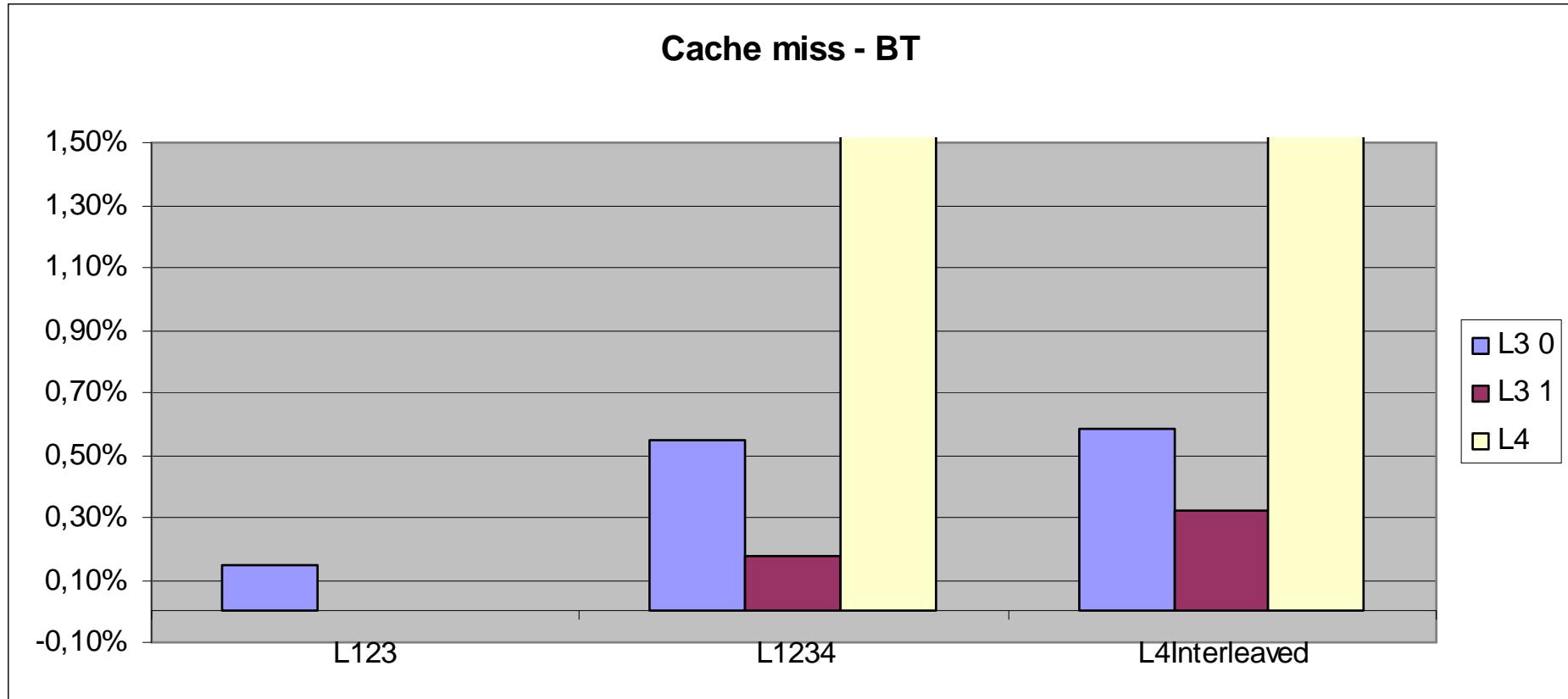
# Resultados



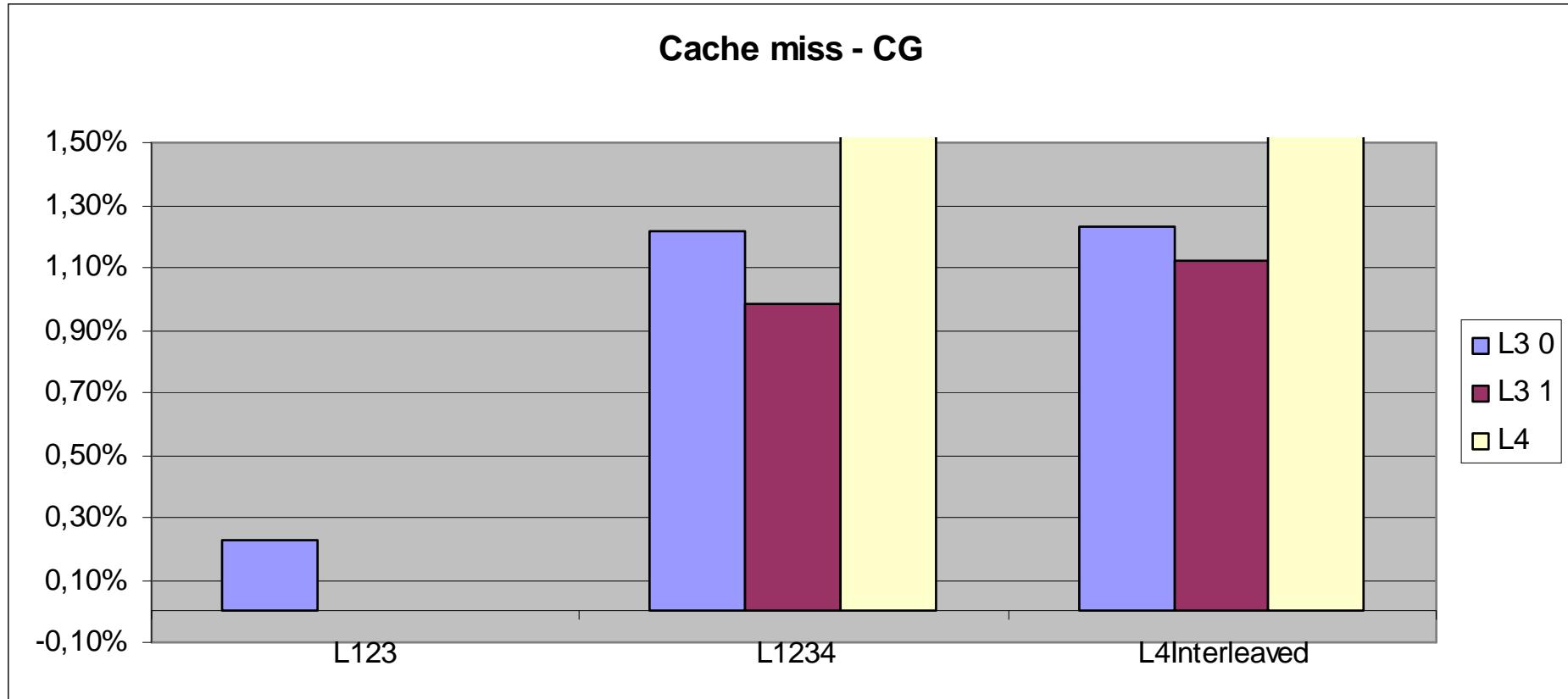
# Resultados



# Resultados



# Resultados



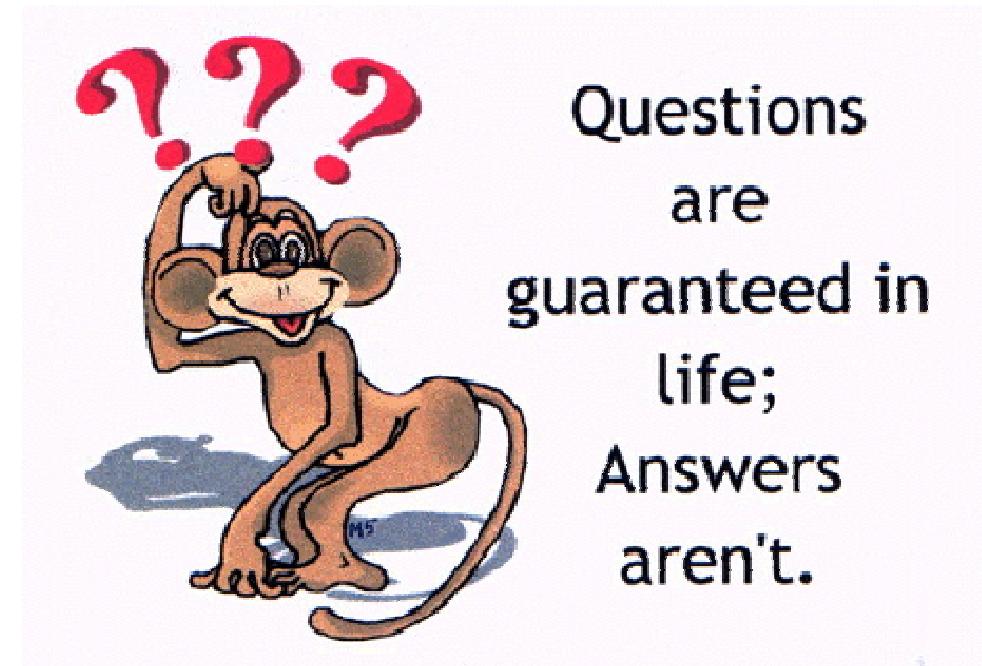
# Conclusões

- Ao se diminuir o tempo de acesso à L3, mesmo não sendo compartilhada e com espaço menor, o tempo de execução diminuiu
- A taxa de cache miss das L3, apesar de ser maior, foi compensada pelo tempo de acesso menor
- Provavelmente, o ganho se deve muito mais à redução do tempo de acesso às L3
- O maior tempo de execução da L4Interleaved provavelmente deve-se ao maior tempo de acesso às L3
- A cache L4 teve cache miss em torno de 30% a 70%. Uma explicação seria o compartilhamento de 16 MB entre os 8 cores (contra 32 MB de L3 na arquitetura L123)
- **Trabalhos futuros:**
  - Executar a mesma simulação da L123, com L3 de 16 MB, e medir a taxa de cache miss
  - Utilizar uma L4 de 32 MB

# Dúvidas/perguntas

# Obrigado!

## Perguntas?



Questions  
are  
guaranteed in  
life;  
Answers  
aren't.