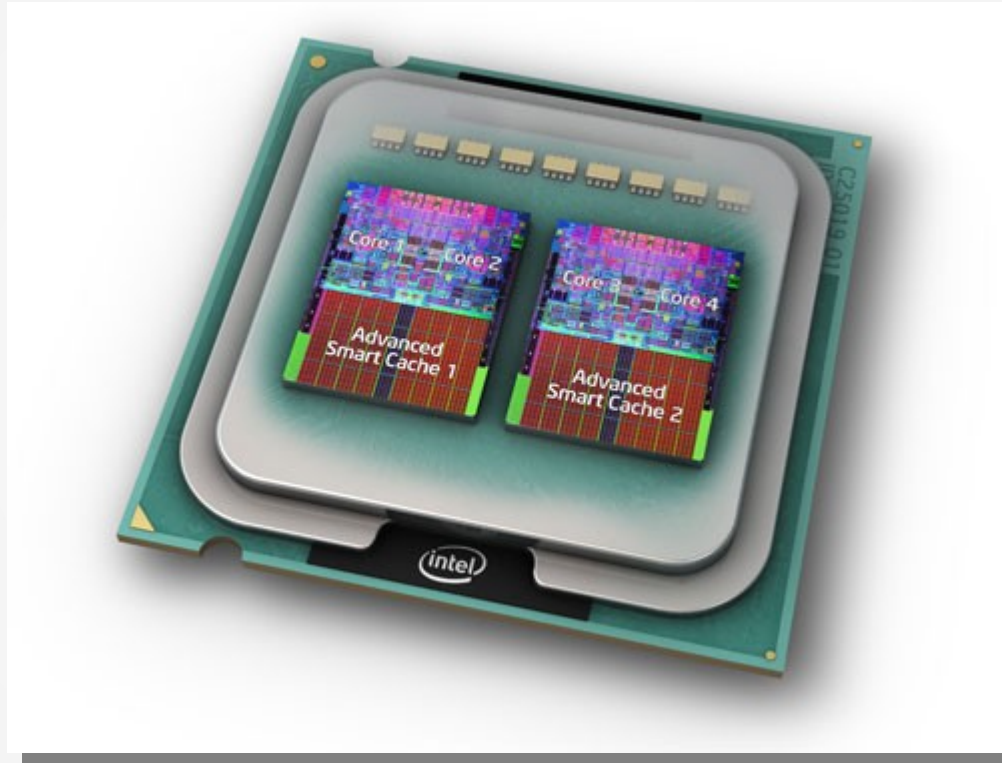


Arquitetura de Computadores



Multicore

Iago Alves da Silva

Mestrando – ECOMP / UPE

Tópicos

- Conceitos Necessários
- Novos Conceitos
- Visões
- Bibliotecas
- Na prática
- Na prática (continuação)
- Na prática (continuação)
- Resultados (sem otimização por compilador)
- Resultado (com otimização do compilador)
- Referências

Conceitos Necessários

- Cache
 - L1
 - L2
 - L3
- Mapeamento de memória
 - Direto: mais simples
 - Associativo: sobrescrita > procurar próximo slot
- Thread
- MIMD – multiple instructions multiple data

Novos Conceitos

- L1
 - L1i – mapeamento direto
 - L1d – associativo
- Mutex – **MUT**ual **EX**clusion
- CPU x Núcleo x Thread x Soquete
 - Soquete
 - Núcleo(s)
 - Thread(s)
 - CPU(s)
- SMT – simultaneous multithreading

Visões

Chuck Moore [...] suggested computers should be like cellphones, using a ***variety of specialty cores*** to run modular software scheduled by a high-level applications programming interface.

[...] Atsushi Hasegawa, a senior chief engineer at Renesas, generally agreed. He suggested the cellphone's use of ***many specialty cores*** working in concert is a good model for future multi-core designs.

[...] Anant Agarwal, founder and chief executive of startup Tiler, took the opposing view. He said multi-core chips need to be ***homogeneous collections of general-purpose*** cores to keep the software model simple.

Fonte: CPU designers debate multi-core future:

http://www.eetimes.com/document.asp?doc_id=1167932

Bibliotecas

- OpenMP – C/C++ e Fortran
 - Padroniza os últimos 20 anos de arquitetura SMP (shared-memory parallel)
- `<thread>` e `<pthread>` – C/C++11 (ou mais recente)

Na prática

- 0.sh
 - Como o S.O. vê os núcleos
 - Resumo dos núcleos
 - Topologia do computador (soquete, cache, cores etc)
- 1
 - Threads executando em paralelo
- 2
 - Exibindo thread executando em processadores de forma aleatória
 - Execução com restrição, por S.O., de núcleo (.sh)
- 3
 - Identificação de thread
- 4
 - Execução de threads com restrição, por código, de núcleo

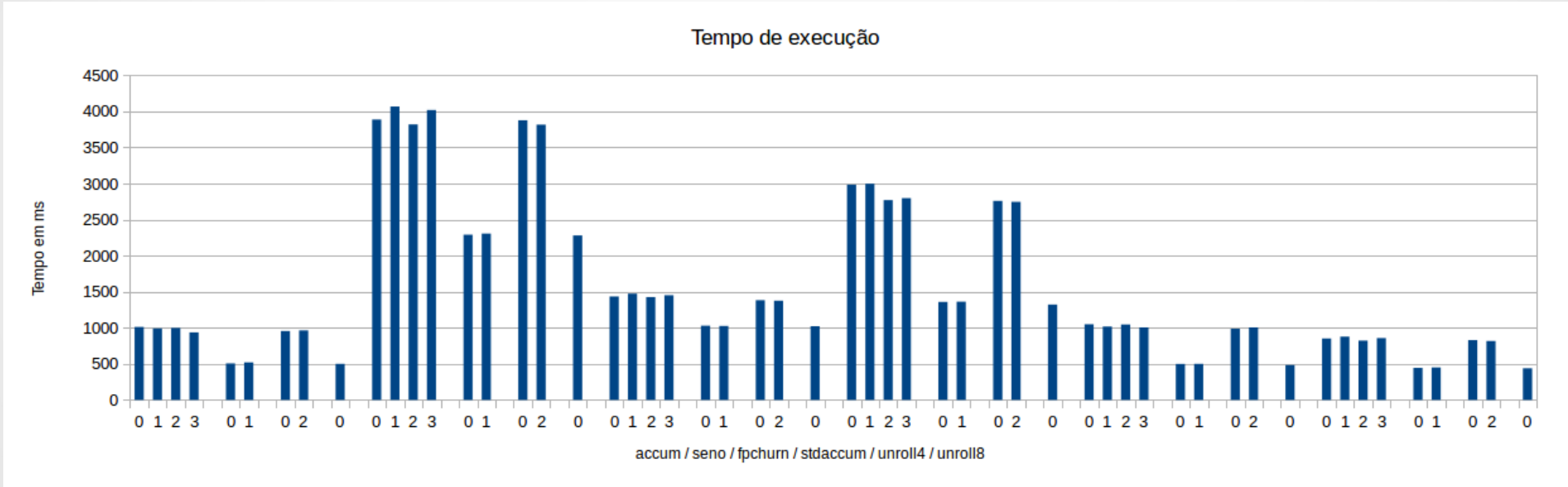
Na prática (continuação)

- 5
 - Fpchurn: cálculos de ponto flutuante (seno, exponencial)
 - Seno
 - Acumulador: soma simples
 - Acumulador: soma usando biblioteca C/C++
 - UnrollAcum4
 - UnrollAcum8

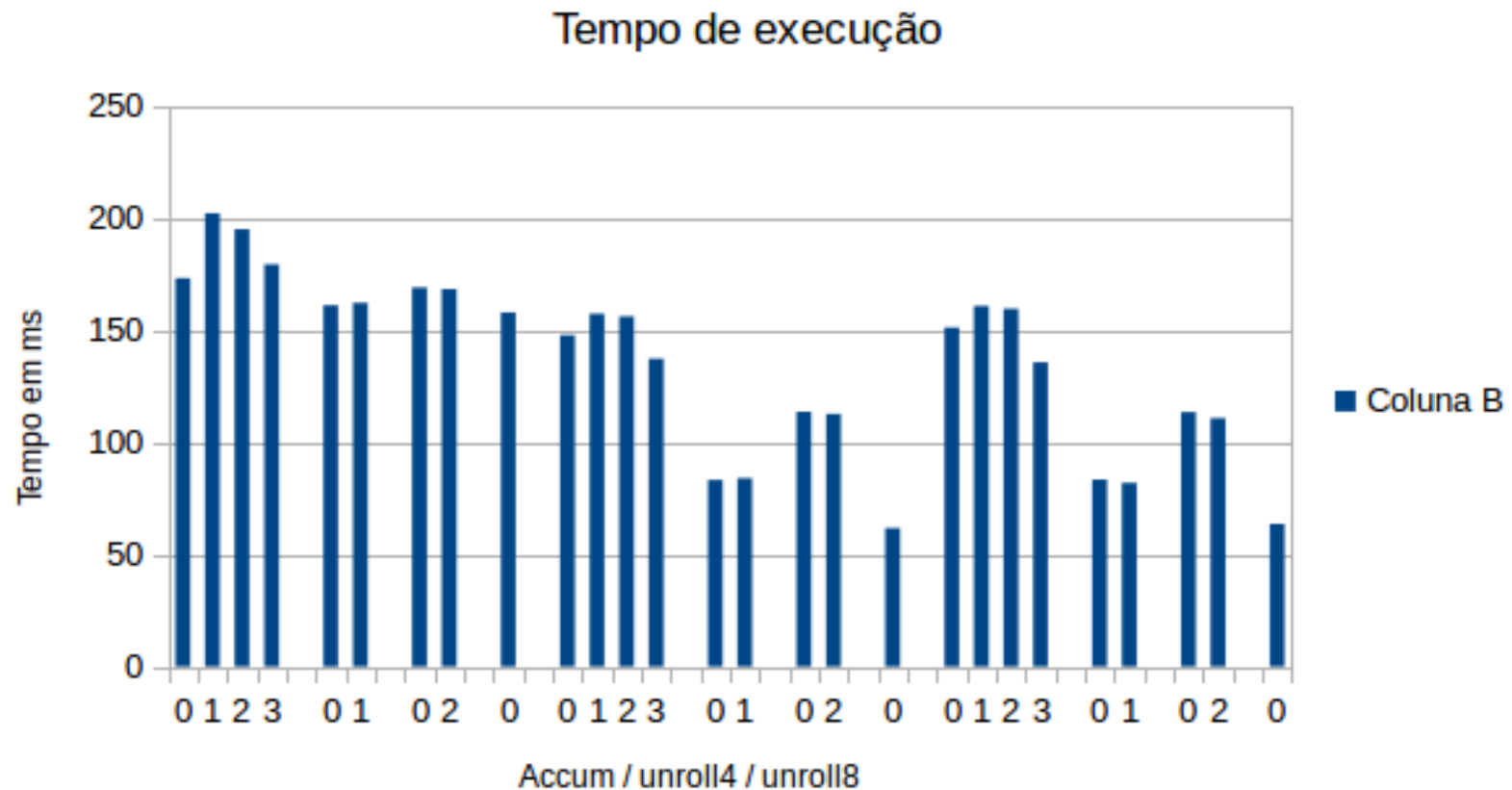
Na prática (continuação)

- 4 CPUs (concorrência em núcleo)
- 2 CPUs (sem concorrência em núcleo)
- 2 CPUs (concorrência em núcleo)
- 1 CPU (sem concorrência em núcleo)

dor)



Resultado (com otimização do compilador)



Referências

- <https://www.cs.cmu.edu/~fp/courses/15213-s06/lectures/27-multicore.pdf>
- http://www.eetimes.com/document.asp?doc_id=1167932
- * <http://www.ti.com/lit/an/sprab27b/sprab27b.pdf>
- http://www-usr.inf.ufsm.br/~reis/publicacoes/wsl2007_gomp.pdf
- *
<http://eli.thegreenplace.net/2016/c11-threads-affinity-and-hyperthreading/>