



Paralelismo em Nível de *Thread* (*Thread Level Parallelism* - TLP)

Eliana Alves Moreira (RA 120437)

Gerson Nunho Carriel (RA 123789)

Julián Esteban Gutiérrez Posada (RA 134097)

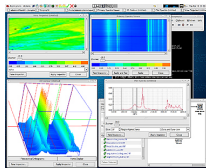
MO401 - Arquitetura de Computadores I - Prof. Paulo Centoducatte
IC/UNICAMP

11 de junho de 2012

Paralelismo^[1]



- O paralelismo tem sido bem utilizado na *computação científica* de alto desempenho
 - Simulações de clima
 - Desenvolvimento de medicamentos
 - Simulações de colisão pela indústria automotiva
 - Aplicações em computação gráfica
 - ...



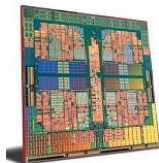
<http://www.nrc-cnrc.gc.ca>

- As pesquisas sobre alto desempenho incluíram **novos desenvolvimentos** em tecnologias paralelas de hardware e software, bem como **programação paralela**

Fabricantes de processadores^[1]



- Impossível **umentar a velocidade** indefinidamente em um **único núcleo**
- Começaram a produzir, por **razões físicas**, *chips* com várias unidades de computação (núcleos ou *cores*) chamados **multicore** [2]

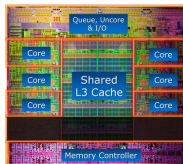


<http://www.dehne.carleton.ca/research/multi-core>

Os processadores multicore - Razões Físicas ^[1]e^[2]



- O **aumento do número de transistores** em um *chip* implica no **aumento do consumo de energia** e da produção de **calor**



<http://www.tomshardware.co.uk/core-i7-3960x-x79-sandy-bridge-e,review-32319.html>

- O **tempo de acesso de memória** não poderia ser reduzido no mesmo ritmo da taxa do **período de clock do processador**

Os processadores multicore - Razões Físicas II^[1]e^[1]



- A **velocidade de transferência** de sinal é **afetada** pelo aumento do **comprimento do fio do processador interno** para transferir o controle de dados entre as unidades funcionais do processador
- A **largura de banda** entre CPU e memória principal está **limitada** pelo o **número de pinos** que, por sua vez, está **limitado** pelo **tamanho físico do chip** do processador



<http://www.memory4less.com/images/products/lmg0922/BX80615E74870-lg.jpg>

Lei de Moore^[1]



- Observação feita por Gordon Moore em 1965
- O **número de transistores** de um chip de processador típico **dobra** a cada **18-24 meses**



<http://forwardthinking.pcmag.com/show-reports/297768>

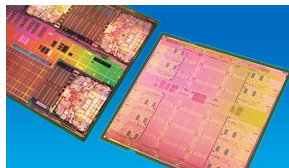
(**válido por mais de 40 anos**)

- Sempre acompanhado por um aumento na velocidade de clock

Multicore^[1]



- Os **processadores** de última geração vêm apresentando **incrementos nos números de núcleos (Lei de Moore)**:
 - **2009**: dois núcleos (*dual-core*) e quatro núcleos (*quad-core*)
 - **2010 - 2012**: oito núcleos (*oct-core*)
 - **2011 - 2012**: 10 a 16 núcleos em apenas um *chip*
(Intel Xeon processador E7, AMD Opteron™ 6282 SE)



<http://www.businessworld.in/businessworld/businessworld/content/Intel-Inside-India.html>

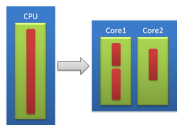
- Em **2015** um típico chip de processador deverá consistir de dezenas e até **centenas de núcleos** dedicados para fins específicos [2]

Paralelismo automático^{[1]e[3]}



Os **usuários** estão interessados em:

- Se **beneficiar** do aumento de **desempenho** fornecido pelos processadores **multicore**
- **Executar** uma **única aplicação** utilizando os vários **núcleos**



<http://www.devx.com/SpecialReports/Article/40965/1954>

Observações:

- Não é possível extrair suficiente **paralelismo automaticamente** em muitos programas sequenciais
 - A **porção de código paralelizável** de cada programa é variável
- A **aplicação** deveria ser **adequadamente escrita**

Evolução do paralelismo^[5]



- **Paralelismo em nível de bits**

O tamanho da palavra usada pelos processadores para operações aumentou gradualmente de 4 *bits* a 64 *bits*

- **Paralelismo por pipelining**

A ideia de *pipelining* em nível de instrução é uma sobreposição da execução de múltiplas instruções

- **Paralelismo por múltiplas unidades funcionais**

Muitos processadores são de emissão múltipla e podem trabalhar em paralelo, tais como processadores superescalares e processadores VLIW [4]

- **Paralelismo em processo ou em nível de thread (TLP) em processadores multicore**

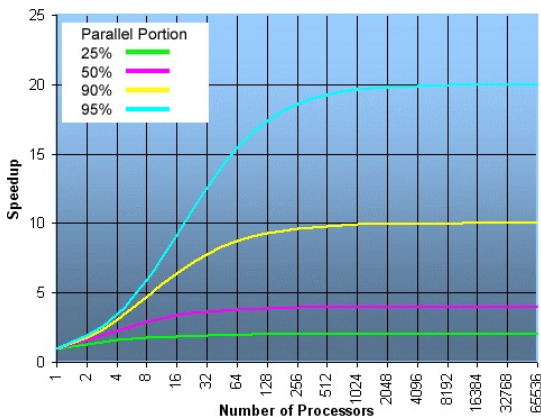
Cada um dos núcleos obtém fluxo de controle separado

Multithread



- Um único *thread*
Geralmente **não fornece suficiente paralelismo** em nível de instrução para usar todas as unidades funcionais dos processadores superescalares
- *Multithreading*
Threads compartilhando, de forma intercalada, as **mesmas unidades funcionais** de um processador, onde o processador duplica o estado independente de cada *thread*[4]
- Paralelismo em nível de *Thread* (TLP)
Envolve o **controle de múltiplos threads** do processador

Limites da Programação Paralela¹: Lei de Amdahl^[6]



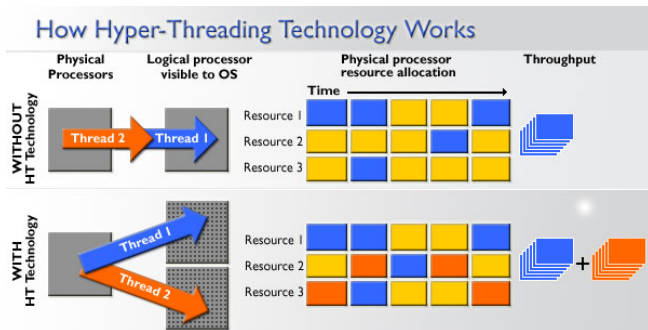
https://computing.llnl.gov/tutorials/parallel_comp/

¹Limites que estão relacionados com a **porção de código paralelizável**

Hyperthreading



- Uma implementação proprietária para o SMT é denominada “**Hyperthreading**” (HT) para os processadores Intel®



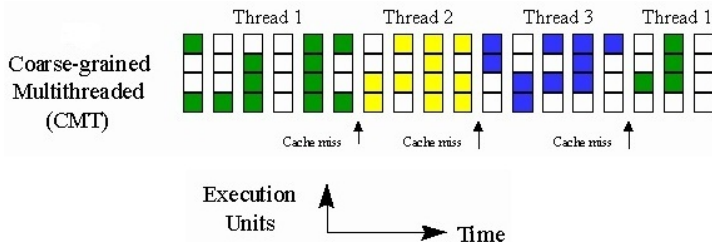
<http://www.techfrill.com/132/hyper-threading-technology-introduction/>

- Permite que um processador físico seja dividido em mais **processadores lógicos** que compartilham unidades funcionais

Multithreads com granularidade grossa (CMT)^[7]



- A **troca** entre *threads* ocorre pelo fato de o *thread* que está em execução utiliza uma operação de **longa latência**



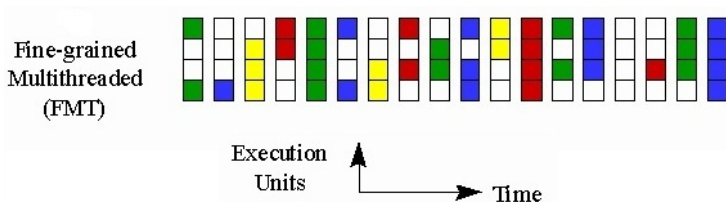
<http://www.realworldtech.com/page.cfm?ArticleID=RWT122600000000>

- O **desempenho** de um *thread* individual é **menos afetado**
- É **difícil** fazer um **balanceamento** eficiente de carga

Multithreads com granularidade fina (FMT)^[7]



- A **troca** entre *thread* ocorre em curtos espaços de tempo (cada **ciclo de clock**)



<http://www.realworldtech.com/page.cfm?ArticleID=RWT122600000000>

- Mais **fácil** fazer um **balanceamento** eficiente de carga
- **Atrasa a execução** de *threads* de forma individual

Programação Paralela: um complemento - I

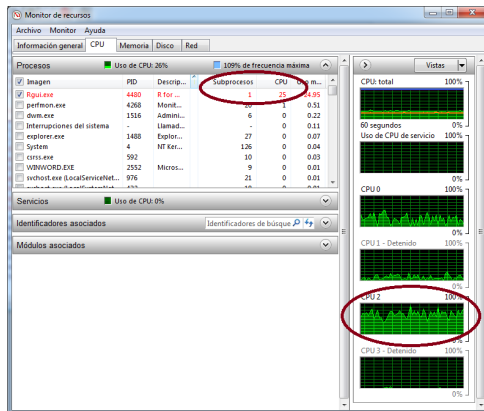


Figura : Ferramenta "R" - Uma execução

Programação Paralela: um complemento - II

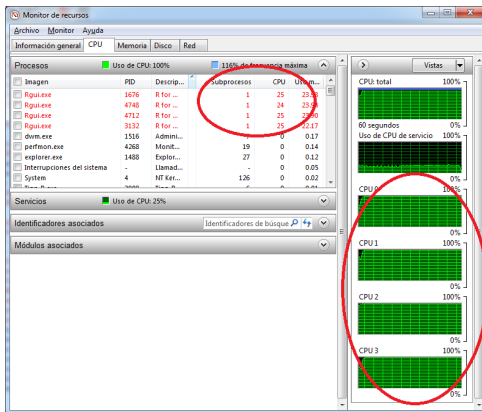
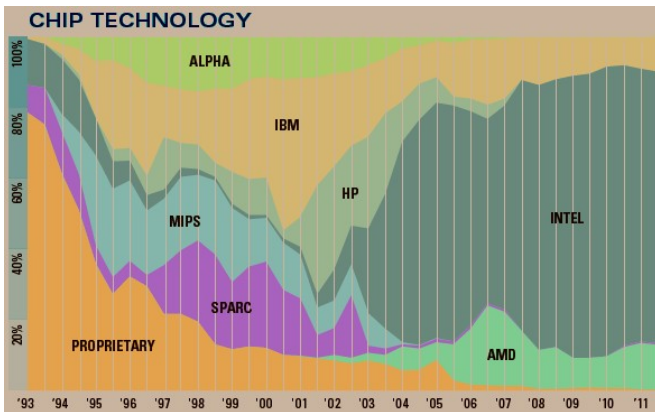


Figura : Ferramenta “R” - Quatro execuções simultâneas

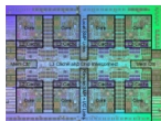
Tecnologias de *Chips*



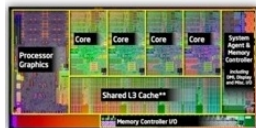
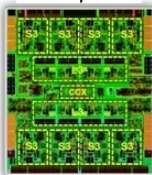
Processadores Atuais



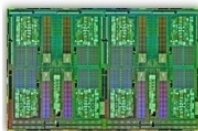
IBM Power 7



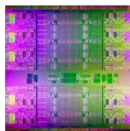
Oracle Sparc T4



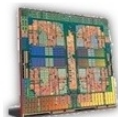
Intel Core i7



AMD Opteron



Intel Xeon E7



AMD Phenon

<http://www.techeta.com/processors/>
<http://www.theregister.co.uk/2011/04/05/intel\xe7\lauch/>
<http://www.theregister.co.uk/2011/08/22/oracle\sparc\t4\hot\chips/>
<http://www.grandblancnewcomers.org/post/amd-opteron-6200-with-16-core-processors>
<http://techgage.com/article/intels\core\i7-980x\extreme\edition\-\ready\for\sick\scores\2>

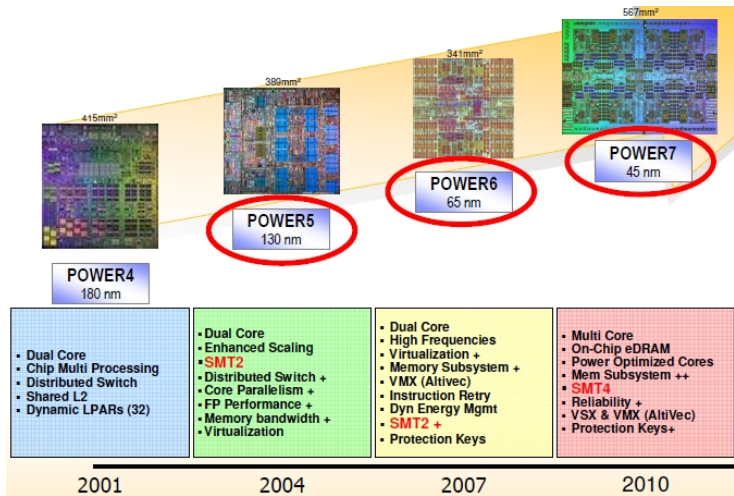
Exemplos de processadores



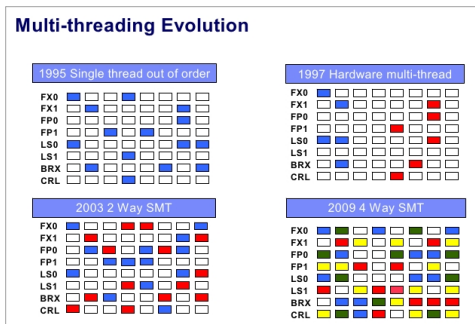
PROCESSADOR	DATA	CORES	THREAD
IBM Power 7[8]	2010	8	32
Intel® Core™ i7 i7-3960X[9]	2011	6	12
Intel® Xeon® E7 E7-8870[10]	2011	10	20
Oracle SPARC T4 [11]	2010	8	64
AMD Phenom™ II X6 Black 1100T[12]	2011	6	6*
AMD Opteron™ 6282 SE[13]	2011	16	16*

(*) AMD Escalabilidade Núcleo verdade[14]
(não suportam a tecnologia *Hyperthread*)[15]

Família de Processadores Power [8]e[16]



Evolução de SMT em IBM [17]



O Power 7 implementa uma execução fora de ordem **out-of-order (OoO)**

Unidades funcionais do Power 7:

duas de ponto fixo
 duas de *load/store*
 quatro de ponto flutuante (*double*)

uma de operações vetoriais
 uma de registrador condicional
 uma de *branch*
 uma de ponto flutuante decimal (*single*)

IBM Power 7: SMT4^[18] - I



Suporta:

- 4 *threads* simultâneas (**4-way SMT**)
- 8 núcleos
- Totalizando 32 *threads* por processador

Cada **núcleo** pode ser **reconfigurado** manualmente ou automaticamente para **fornecer todos** seus recursos para **menos threads** ou para **um único thread**

O Sistema Operacional **pode organizar** os *threads* para executarem **sozinhos** no *core* [18]

O **SMT4** provê uma **melhora média** no *throughput* de 1.5X - 2X vezes mais do que o fornecido quando um *thread* executa sozinho no *core* [19][18]

IBM Power 7: SMT4^[19] - II



Suporta:

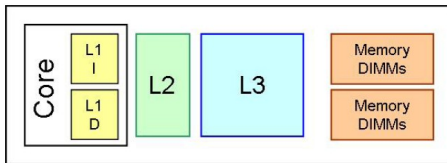
- **CUoD** (Capacity Upgrade on Demand):
Configuração da **quantidade de núcleos** que realmente se quer utilizar [19]
- Granularidade fina (**FMT**)
- No melhor dos casos, apresenta uma taxa de **cinco instruções por ciclo** (*IPC*):
 - Dois pipelines para execução de **instruções** com acesso a armazenamento
 - Dois pipelines para execução de **instruções aritméticas**
 - Um pipeline para **instruções de desvio**

IBM Power 7: Caches^[20]



O compartilhamento de informações das *threads* pode ser feito através de caches

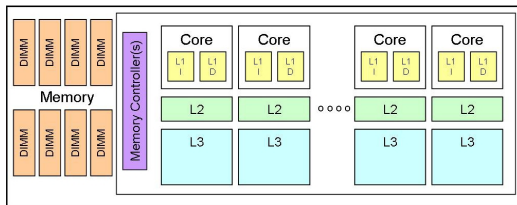
- Suporta um novo tipo de gerenciamento de cache: “*lateral cast-out*”²



- **Cache L1 (32 KB):** velocidade de frequência do processador
- **Cache L2 (256 KB):** poucos ciclos para o acesso
- **Cache L3 (4 MB):** mais de 50 ciclos

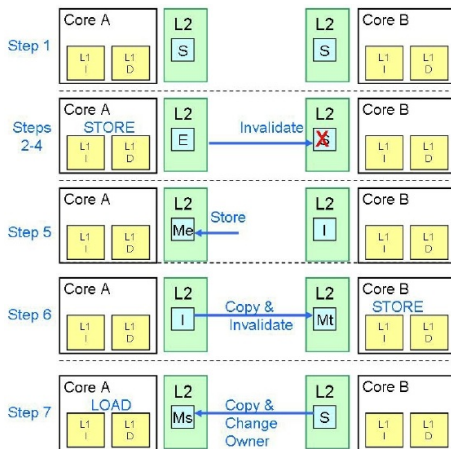
²Permite armazenar dados mais antigos dentro da cache de outras *cores* do mesmo *chip*, de modo a manter os dados na *cache* do *chip* por um pouco mais de tempo [20]

IBM Power 7: Coerência de Cache^[20]



- Os *cores* podem compartilhar dados entre suas caches L2
- É necessário ter mecanismos para garantir a coerência entre os dados
- Todos os *cores* recebem uma cópia dos mesmos dados e estes existem também na memória física

IBM Power 7: Cache Coherence^[20]



Conclusões



- Processadores *multicore* são o futuro dos processadores por várias **razões físicas**
- Há **limites reais** sobre o aumento do desempenho através paralelismo
- A **programação paralela** é um aspecto importante da computação de alto desempenho e vai se tornar a **tendência** das técnicas de desenvolvimento de software
- A **experiência** da investigação em paralelização tem mostrado que, para muitos programas sequenciais **não é possível extrair** o paralelismo **automaticamente**
- Nem todas as aplicações obtêm a mesma **proporção** de melhora no desempenho: depende da quantidade de **código paralelizável**
- TLP utiliza os **recursos** da máquina que estariam **ociosos** durante a execução de somente uma *thread*
- **Não** existe uma **única forma** de implementar TLP

Referências Bibliográficas I



- [1] [Gudula Runger Rauber Thomas.](#)
Parallel Programming: for Multicore and Cluster Systems.
Springer-Verlag Berlin Heidelberg, 3ra edition, 2010.
- [2] [David J. Kuck.](#)
Platform 2015 software-enabling innovation in parallelism for the next decade.
Technology@Intel Magazine, 2005.
- [3] [Tianzhou Chen, Xingsheng Tang, Jianliang Ma, Lihan Ju, Guanjun Jiang, and Qingsong Shi.](#)
Single thread program parallelism with dataflow abstracting thread.
ICA3PP'10. Springer-Verlag, Berlin, Heidelberg, 2010.
- [4] [David A. Patterson John L. Hennessy.](#)
Computer Architecture: A Quantitative Approach.
ELSEVIER, Morgan Kaufmann Publishers, 4ra edition, 2007.
- [5] [David Culler, J.P. Singh, and Anoop Gupta.](#)
Parallel Computer Architecture: A Hardware/Software Approach.
Morgan Kaufmann, 1st edition, 1998.
- [6] [Lawrence Livermore National Laboratory Blaise Barney.](#)
Introduction to parallel computing.
https://computing.llnl.gov/tutorials/parallel_comp/, 2012.
[Online; acessada 18-Mai-2012].
- [7] [Lin Gao.](#)
Multithreading.
<http://www.cse.unsw.edu.au/cs9244/06/seminars/05-gaol.pdf>, 2012.
[Online; acessada 18-Mai-2012].

Referências Bibliográficas II



- [8] **IBM Corporation.**
Power7 processors: The beat goes on.
<http://www.ibm.com/developerworks/wikis/download/attachments/104533501/POWER7+-+The+Beat+Goes+On.pdf>, 2011.
[Online; acessada 12-Mai-2012].
- [9] **Intel Corporation.**
Intel® core™ i7-3960x processor extreme edition.
<http://www.intel.com/content/www/us/en/processor-comparison/processor-specifications.html?proc=63696>, 2012.

[Online; acessada 12-Mai-2012].
- [10] **Intel Corporation.**
Intel® xeon® processor e7-8870.
<http://ark.intel.com/products/53580>, 2011.
[Online; acessada 12-Mai-2012].
- [11] **ORACLE Corporation.**
Sparc t4 processor.
<http://www.oracle.com/us/products/servers-storage/servers/sparc-enterprise/t-series/sparc-t4-processor-ds-497205.pdf>, 2010.

[Online; acessada 12-Mai-2012].
- [12] **AMD Corporation.**
Amd phenom™ ii x6 black 1100t.
<http://products.amd.com/en-us/DesktopCPUDetail.aspx?id=726>, 2010.
[Online; acessada 12-Mai-2012].

Referências Bibliográficas III



- [13] **AMD Corporation.**
Amd opteron™ 6282 se processor.
<http://products.amd.com/en-us/OpteronCPUDetail.aspx?id=756>, 2011.
[Online; acessada 12-Mai-2012].
- [14] **HP Hewlett-Packard Development Company.**
Amd opteron™ and intel® xeon® x86 processors in industry-standard servers.
<http://h20000.www2.hp.com/bc/docs/support/SupportManual/C02731435/C02731435.pdf>, 2012.
[Online; acessada 20-Mai-2012].
- [15] **AMD Corporation.**
Intel hyper-threading vs. amd true core scalability.
<http://blogs.amd.com/work/2010/03/30/intel-hyper-threading-vs-amd-true-core-scalability/>, 2010.
[Online; acessada 12-Mai-2012].
- [16] **IBM Corporation.**
Advances in simultaneous multithreading testcase generation methods.
http://www.research.ibm.com/haifa/conferences/hvc2010/present/Advances_in_Simultaneous_Multithreading_Testcase_Generation_Methods.pdf, 2011.
[Online; acessada 30-Mai-2012].
- [17] **Carlo Costantini Volker Haug Cesar Diniz Maciel John T Schmidt An Ding Chen, James Cruickshank.**
IBM Power 795: Technical Overview and Introduction.
ibm.com/redbooks, 1ra edition, 2010.
- [18] **IBM Corporation.**
What's this multi-core computing really?
<http://www-03.ibm.com/systems/resources/pwrsysperf/WhatIsMulticoreP7.pdf>, 2011.
[Online; acessada 30-Mai-2012].

Referências Bibliográficas IV



- [19] [IBM Corporation](#).
Simultaneous multi-threading on power7 processors.
http://www-03.ibm.com/systems/resources/pwrsysperf_SMT4OnP7.pdf, 2011.
[Online; acessada 30-Mai-2012].
- [20] [IBM Corporation](#).
Under the hood: Of power7 processor caches.
http://www-03.ibm.com/systems/resources/systems_power_software_i_perfmgmt_underthehood.pdf, 2011.
[Online; acessada 30-Mai-2012].



Obrigado(a) pela atenção!!



<http://padegaindia.in/wp-content/uploads/2012/01/interview-150x150.jpg>