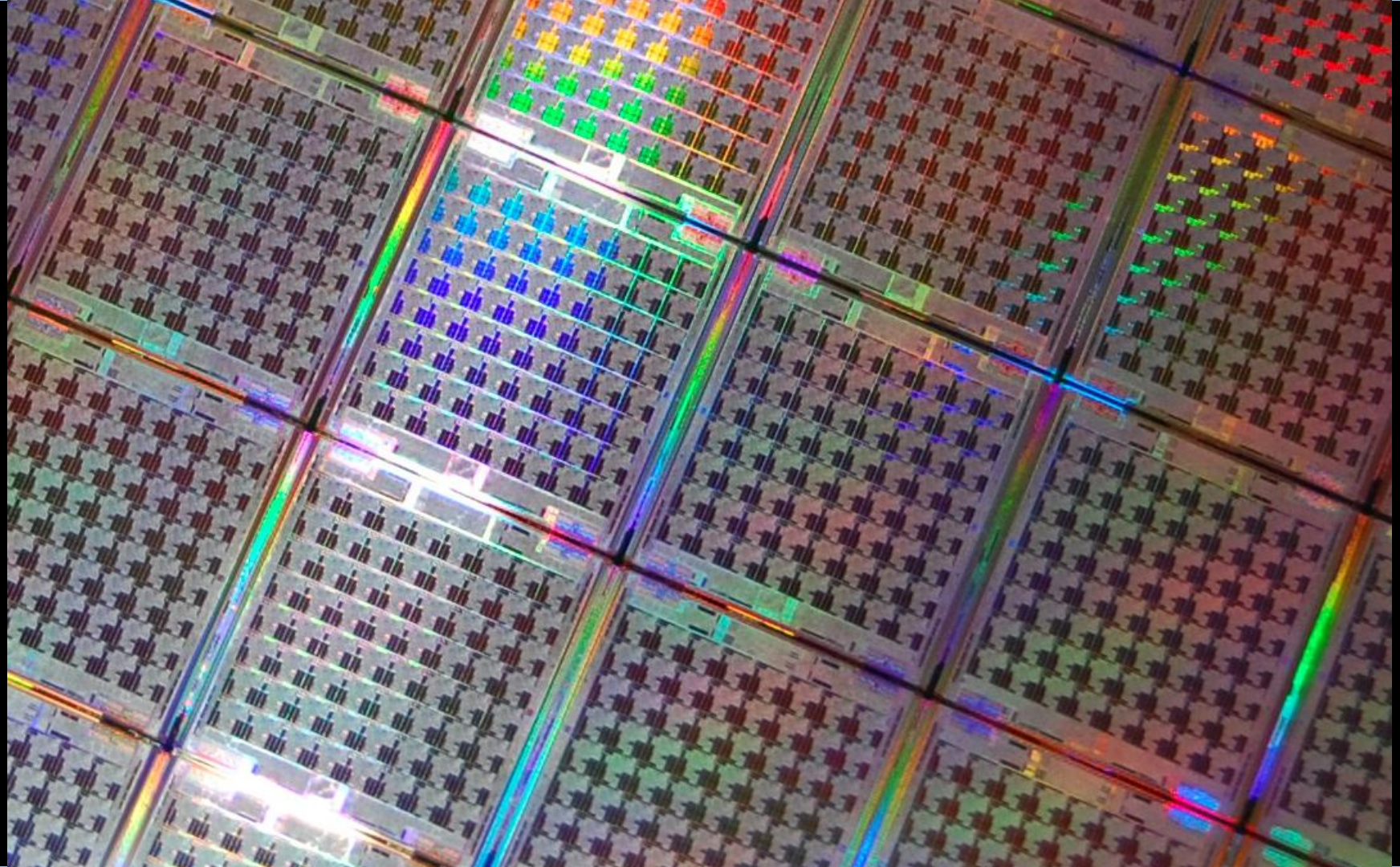
ANÁLISE DE DESEMPENHO E ESCOLHA DINÂMICA DE ESCALONAMENTO PARA SISTEMAS MULTICORE

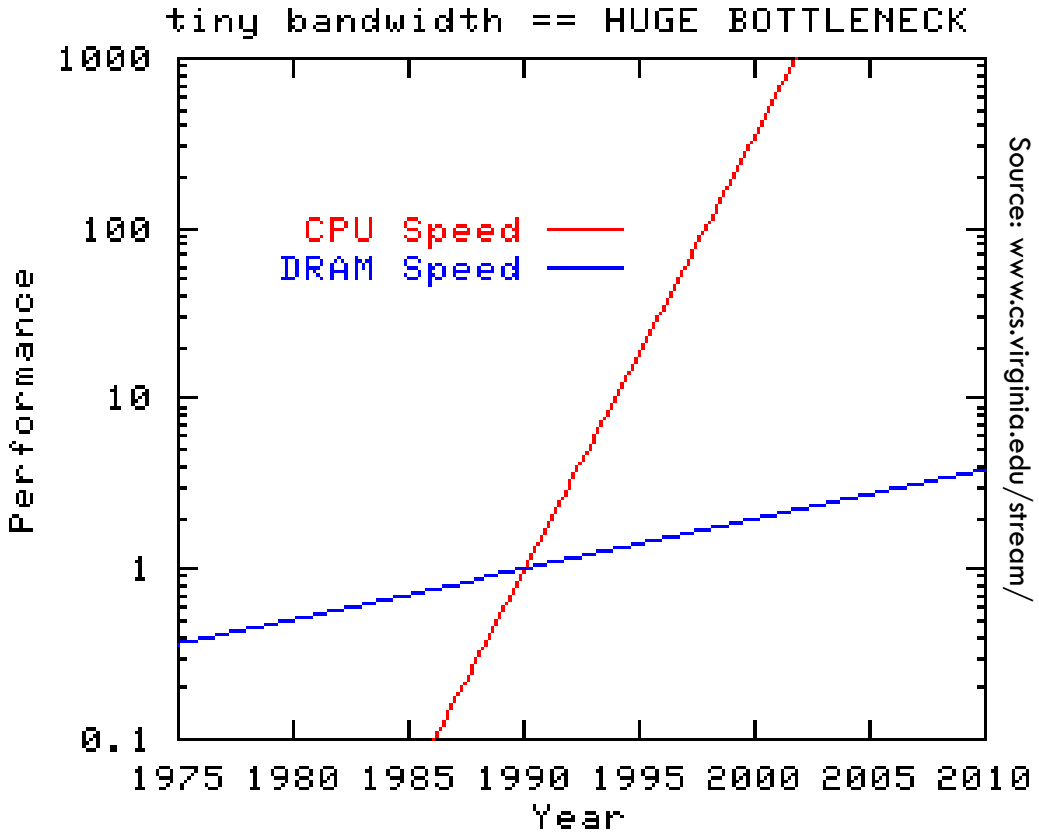
EMILIO FRANCESQUINI, ALFREDO GOLDMAN
UNIVERSIDADE DE SÃO PAULO
{EMILIO, GOLD}@IME.USP.BR

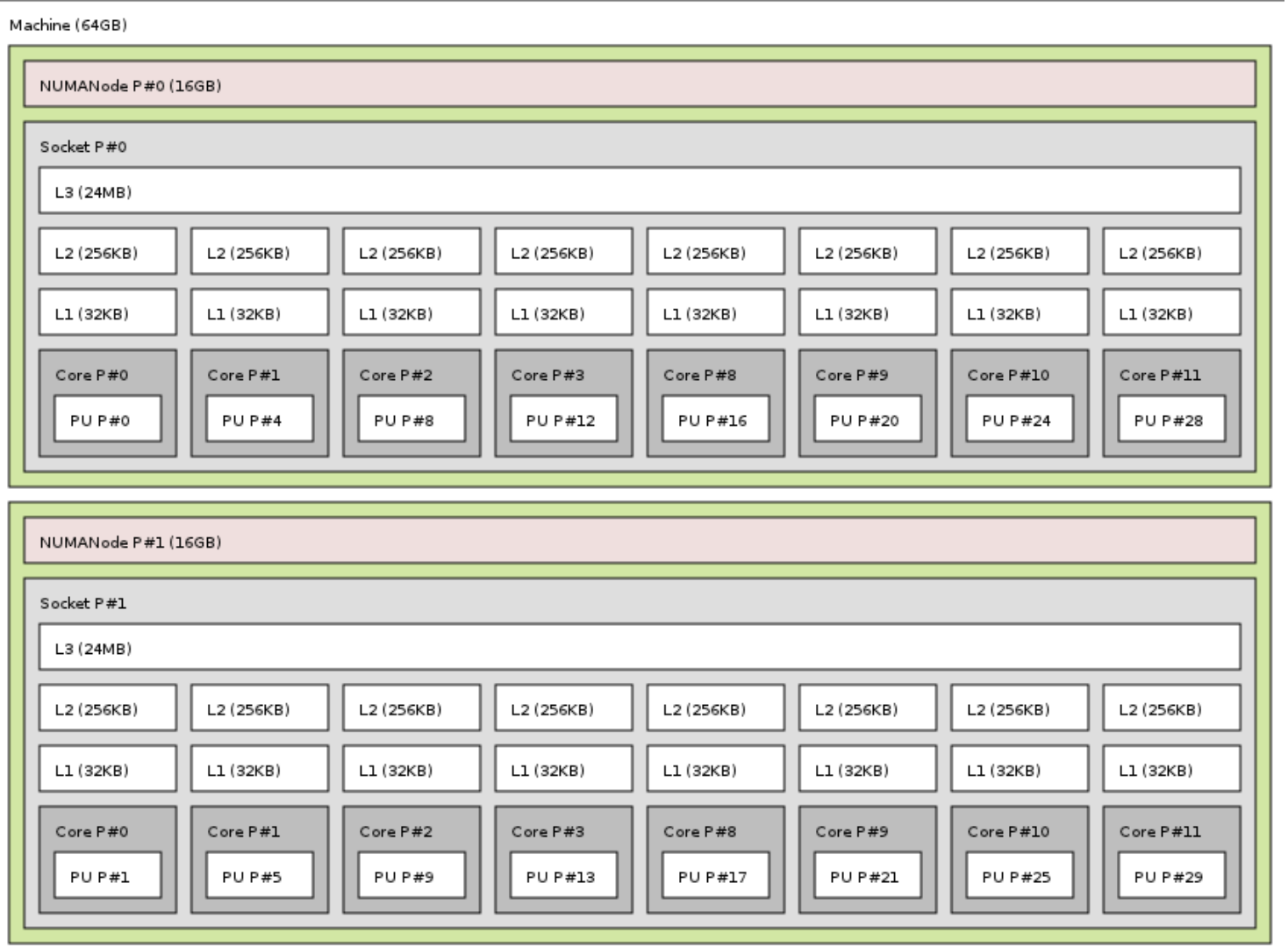
Julho de 2011



Problemas

- Power Wall
- Memory Wall
- ILP Wall





idrouille
-Intel(R) Xeon(R)
Beckton X7560
@ 2.27GHz

Possíveis ações

- Definição de afinidade processo ⇔ processador
- Colocação explícita de páginas de memória

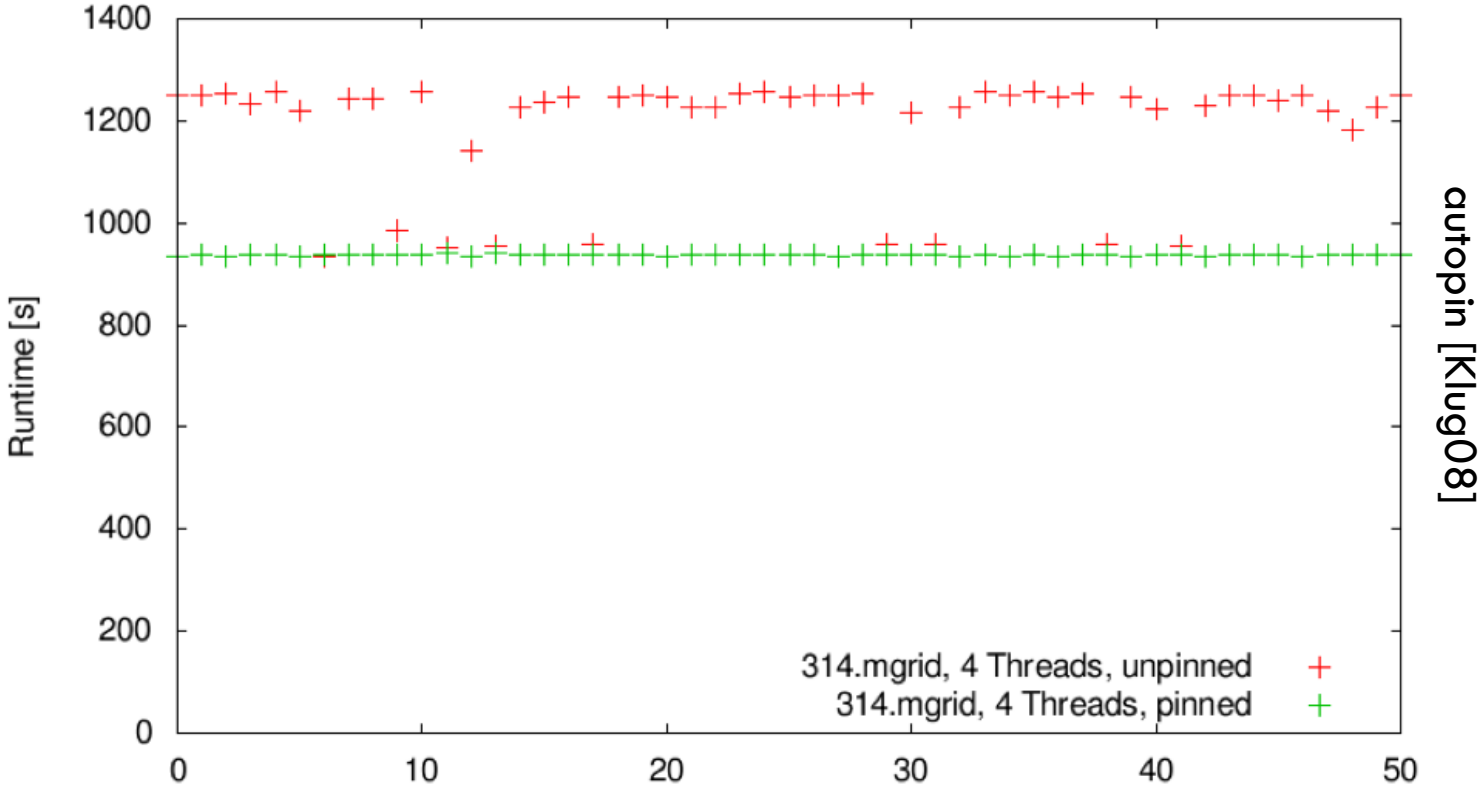
Fixação de processos

- Linpack em16 cores

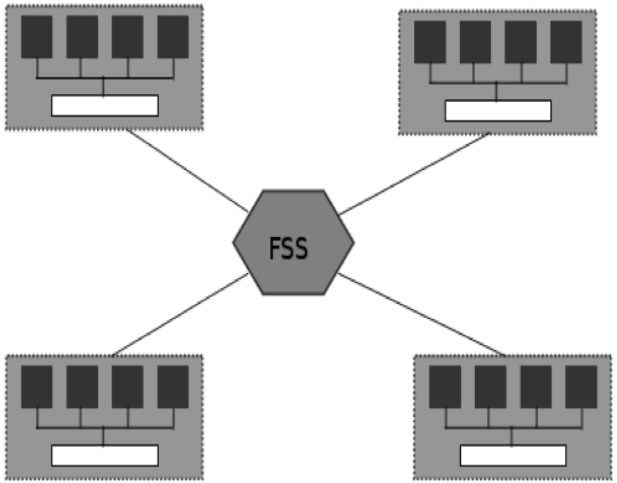
Configuration	CPUs	Time (sec)	% Speedup
No affinity control used	16	467.16	
taskset 0xf	16	481.83	-3.04%
taskset 0x1; 0x2; 0x4; 0x8	16	430.44	8.53%
–cpu_bind=map_cpu:0,1,2,3 -B 1:1	16	430.36	8.55%

[Balle2007]

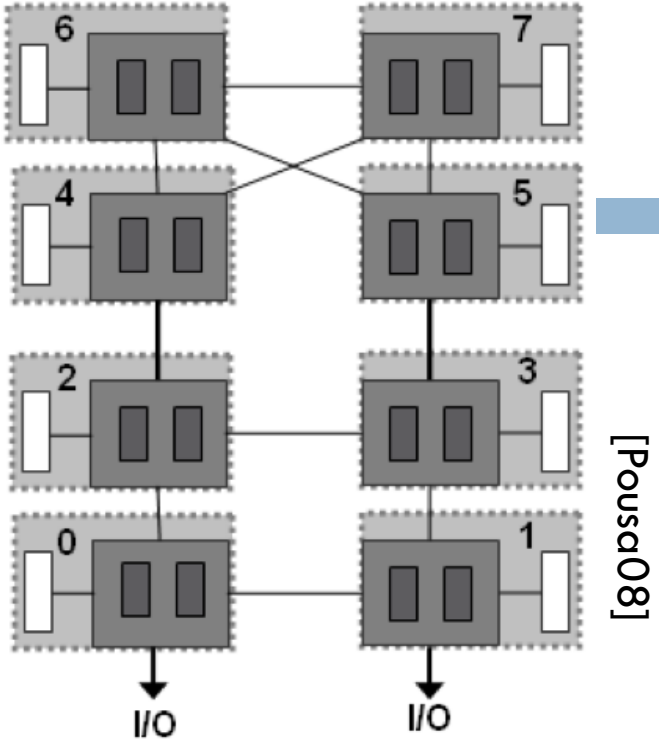
Fixação de processos



NUMA

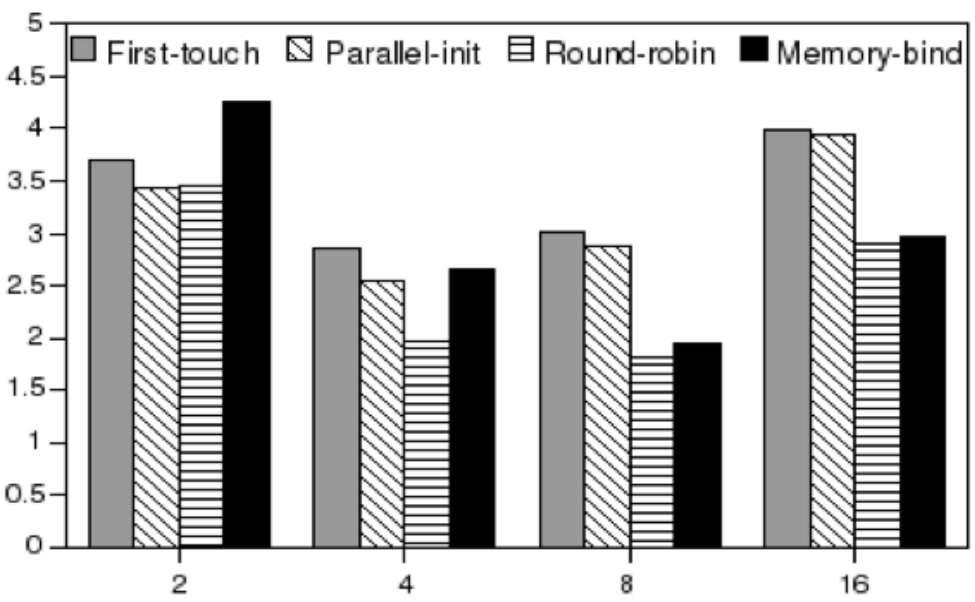


- Fator NUMA: 2 → 2.5
- 16 Itanium2 @1.6 GHz
- 64 GBytes RAM



- NUMA factor: 1.2 → 1.5
- 8 dual-core Opteron @ 2.2 GHz
- 32 Gbytes RAM

NAS Benchmark

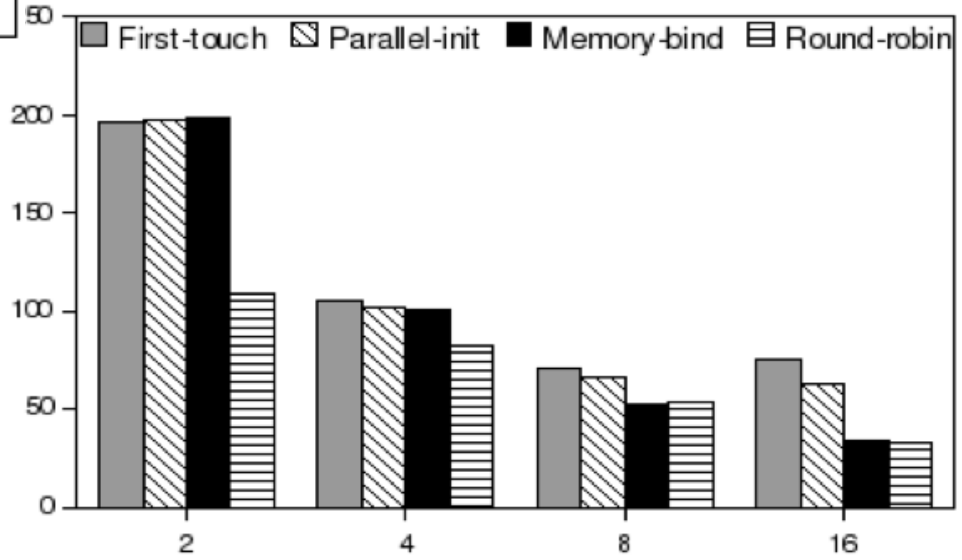


Opteron

- CG Kernel – CFD
- Alto uso de memória
- Padrão de acesso irregular

[Pousa08]

Itanium



Resumindo

- Ambientes heterogêneos
  - Alocações distintas levam a tempos de execução diferentes
    - Máquinas heterogêneas
      - Cada core pode possuir diferentes tempos de acesso à memória
    - Variações no tempo de comunicação entre cada um dos cores
      - Propriedades da interconexão são dinâmicas
        - Localização
        - Carga
        - Perfil da aplicação
        - ...
  - Tencionamos minimizar o tempo total de execução da aplicação

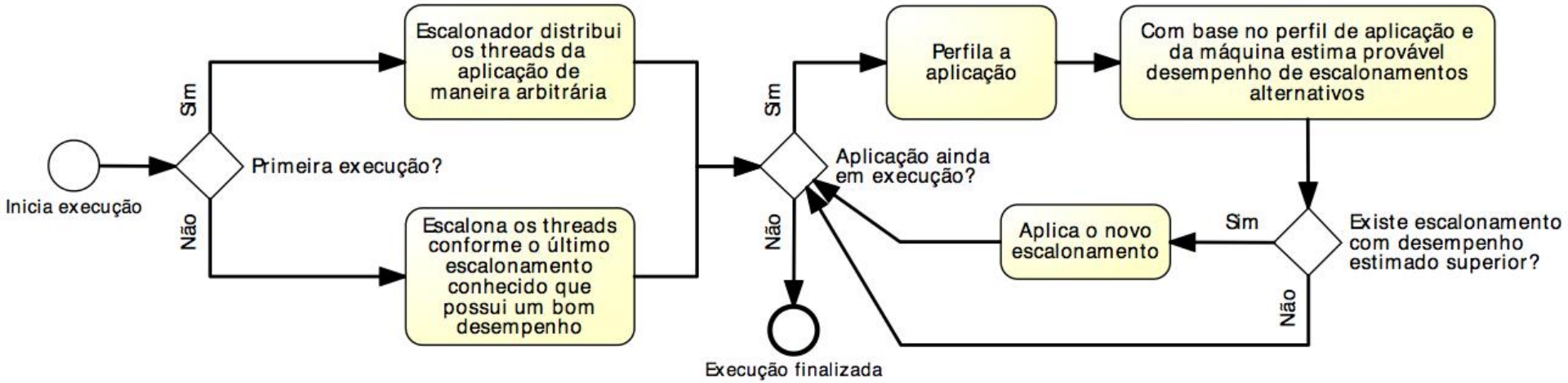
Problemas com as soluções atuais

- Falta de escalabilidade
  - Teste e escolha manual de alocações
    - Para cada aplicação
    - Para cada máquina
- Falta de adaptabilidade dinâmica
  - Ao ambiente
  - Ao comportamento da aplicação

Nossa proposta

- Perfilamento online e offline tanto das aplicações quanto do ambiente
- Escalonamento dinâmico de tarefas
- Colocação dinâmica de páginas de memória
- Proceder gradualmente para uma solução abrangente

Nossa proposta (cont.)

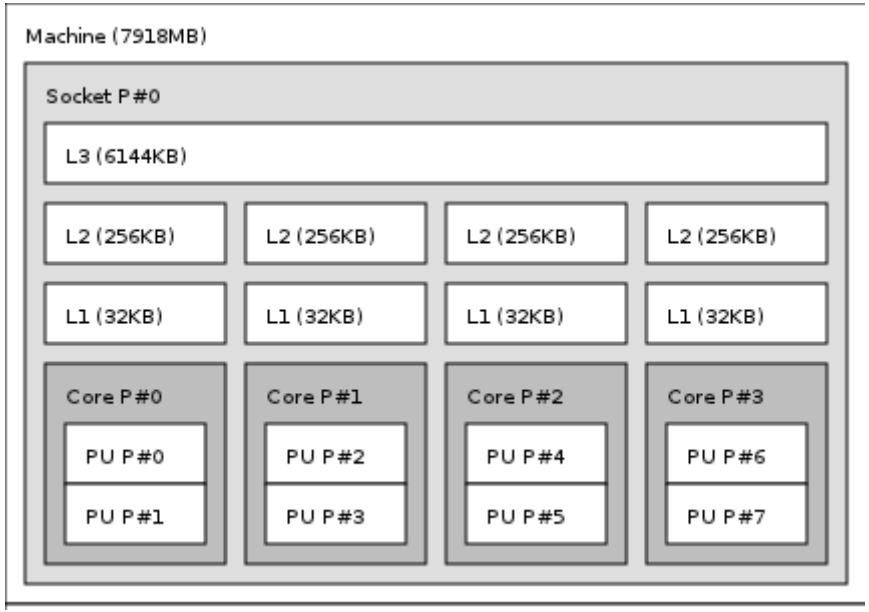


Perfilamento

PAPI

- Autopin [Klug08]
  - Retired instructions
- Exemplos
  - Level 1/2/3 data/instruction cache misses
  - Cache Line Invalidation (SMP)
  - Data/Instruction translation lookaside buffer misses
  - Integer/FP instructions executed, FLOPS

hwloc



Outras ferramentas úteis

- Netperf
- OProfile 

- STREAM
- NAS Parallel Benchmarks
- SPEC Benchmarks 

Desafios de pesquisa

- Escalonamento de processos leves
  - Relacionamento entre os processos
    - Aplicação de BubbleSched
  - Fixação de processos aos cores
  - Modelo de atores
- Construção de um escalonador dinâmico e automático

Referências

- [Balle2007] Enhancing an Open Source Resource Manager with Multi-Core/Multi-threaded Support, S. M. Balle and D. Palermo, Job Scheduling Strategies for Parallel Processing, 2007.
- [Pousa08]Christiane Pousa Ribeiro, Jean-Francois Mehaut, Alexandre Carissimi, Marcio Castro, and Luiz Gustavo Fernandes. Memory affinity for hierarchical shared memory multiprocessors. Computer Architecture and High Performance Computing, Symposium on, 0:59–66, 2009.
- [Klug08] Tobias Klug, Michael Ott, Josef Weidendorfer, and Carsten Trinitis: *"autopin - Automated Optimization of Thread-to-Core Pinning on Multicore Systems"* Transactions on High-Performance Embedded Architectures and Compilers, 3(4), 2008

Obrigado!

Perguntas