Impacto das Interfaces de Programação Paralela e do Grau de Paralelismo no Consumo Energético de uma Aplicação

Thayson R. Karlinski, Arthur F. Lorenzon, Antonio C. Beck, Márcia C. Cera.

unipampa
Universidade Federal do Pampa

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

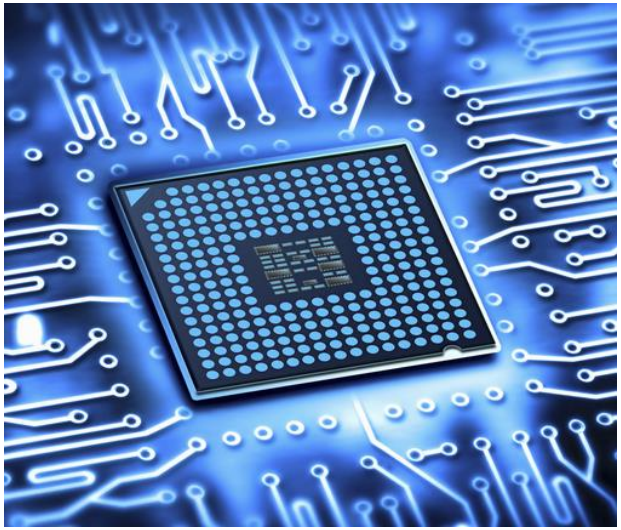
ROTEIRO

- Motivação e Objetivo
- Interfaces de Programação Paralela (IPP): OpenMP, Pthreads, MPI
- Aplicação Alvo: Cálculo do PI
- Metodologia de Trabalho
- Resultados
- Conclusão e Trabalhos futuros
- Referências

Motivação

- Processamento Paralelo
  - *Aumento de Desempenho*
  - *Impacto das Interfaces de Programação Paralela*
  - *Observando Ambiente de Memória Compartilhada*
  - *Acessos Concorrentes a memória*

- Consumo de Energia
  - Maior número de *Threads, Menor Tempo*
  - Acessar a Memória Interfere?



OBJETIVO

- Avaliar o desempenho e o consumo energético da paralelização
  - Variando IPP e Distribuições de Cargas de Trabalho
  - Variando número de *threads/*processos
  - Aplicação CPU-Bound

Interfaces de Programação Paralela (IPP)

- Interface de programação paralela (IPP)
  - OpenMP
  - Posix threads
  - Message Passing Interface (MPI)

IPP:	Cria-se:	Baseado Em:	Controle de Acessos a Dados
OpenMP	Threads	Diretivas	Busy-Waiting
Pthreads	Threads	Funções de Biblioteca	Mutex
MPI - 1	Processos	Funções de Biblioteca	Send/Receive

Aplicação: Cálculo do PI

- Algoritmo CPU-Bound
- Método de Leibnetz : $\pi = 4\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1} = 4\left(\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + - \cdots\right)$

Cód. Sequencial

```
1  #define NUM_PONTOS
2  long long int i;
3  float pi=0.0;
4  for(i=0;i<NUM_PONTOS;i++){
5      pi += 4.0/(4.0*i+1.0);
6      pi -= 4.0/(4.0*i+3.0);
7  }
```

Ferramenta de Coleta de Eventos

- *PAPI (Perfomance Application Programming Interface)*
  - Desenvolvida no *Innovative Computing Laboratory* (ICL)
  - Universidade do Tenessi
- Estrutura de contadores de Hardware
- Eventos utilizados:
  - `PAPI_TOT_INS` → Total de Instruções Executadas
  - `PAPI_CYC_INS` → Total de Ciclos de *Clock*
  - `PAPI_LD_INS` → Total de *Load* a *cache* L1
  - `PAPI_SR_INS` → Total de *Store* a *cache* L1
  - `PAPI_L1_ICM` → Total de *Misses* de Instruções a *cache* L1
  - `PAPI_L1_DCM` → Total de *Misses* de Dados a *cache* L1

Estimativa do consumo energético

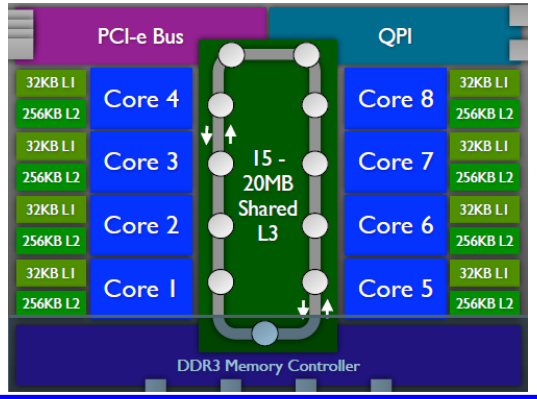
- $E_{total} = E_{instrução} + E_{memória} + E_{estática}$
- $E_{instrução}$ = Consumo total das instruções executadas
- $E_{memória}$ = Consumo de energia de acessos a memória
- $E_{estática}$ = Energia estática do processador e do sistema de memória

Ambiente e Metodologia de Execução

- Distribuição de cargas de trabalho
  - *Static*
  - *Guided*
  - *Nº Iterações / Nº Threads*
- Total de execuções: 16 vezes
  - Eliminou-se os 3 melhores e 3 piores

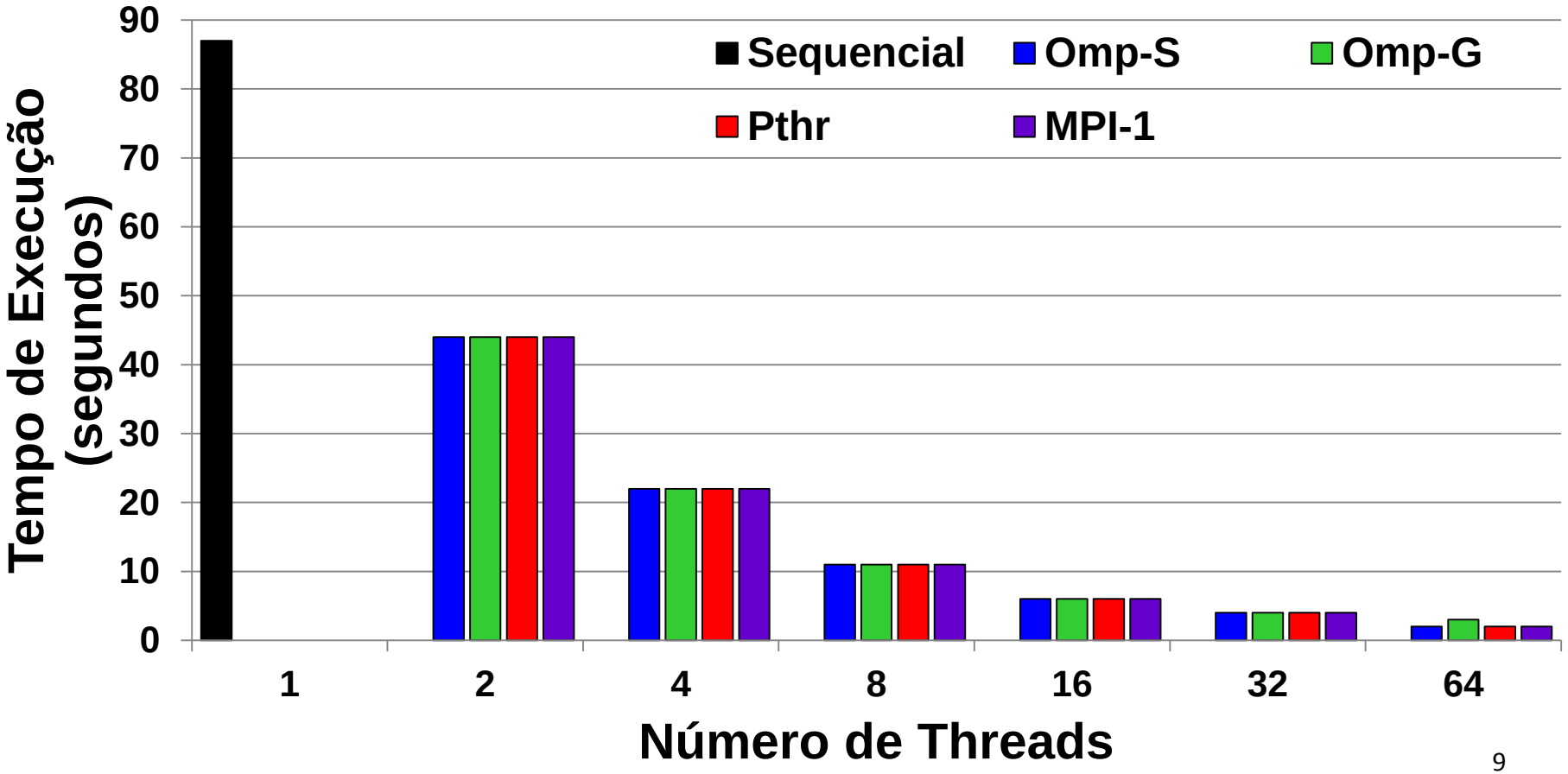
- 1 Multiprocessadores – 2 X Intel(R) Xeon(R) CPU E5-2650 2.80 GHz, 8 cores com Tecnologia Hyper-Threading.

- Número de *Threads*
  - 2 4 8 16 32 64



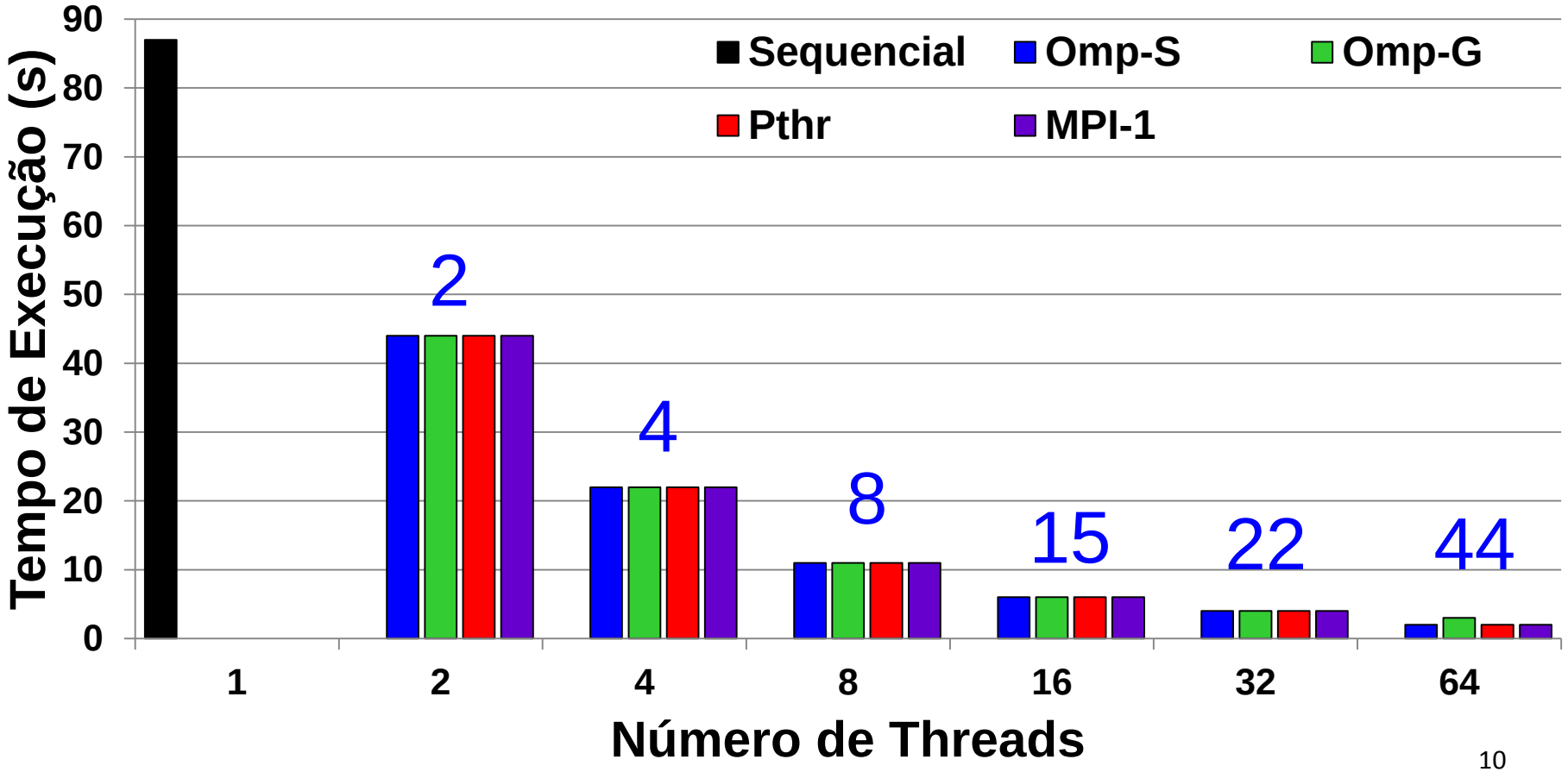
Resultados: Tempo de Execução

- Maior número de *threads*, menor tempo de execução
  - Distribuição da carga de trabalho entre as *threads*



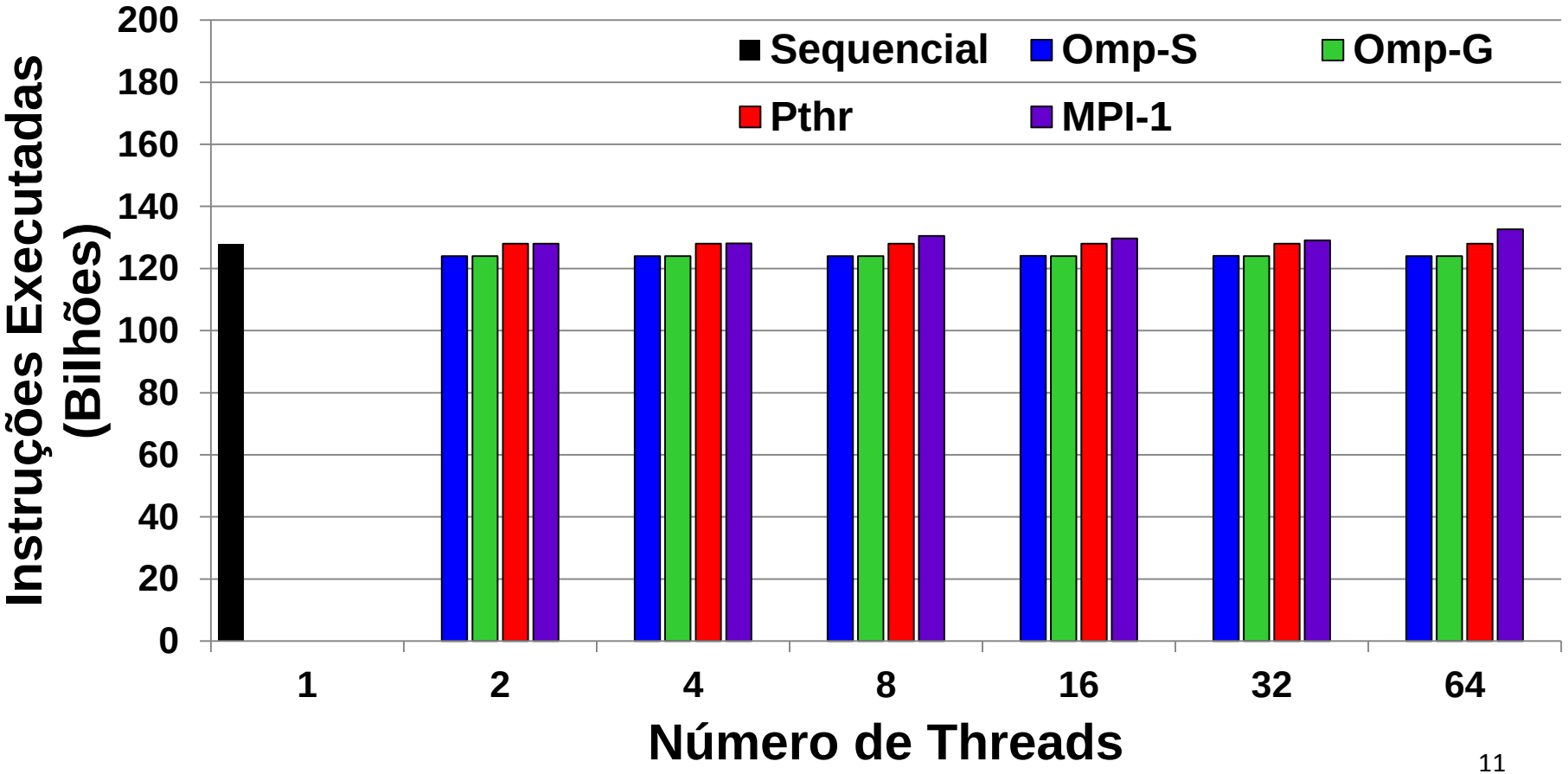
Resultados: Speedup

- até 16 threads, *speedup* similar ou próximo do ideal
  - Tipo de aplicação não possui troca de dados



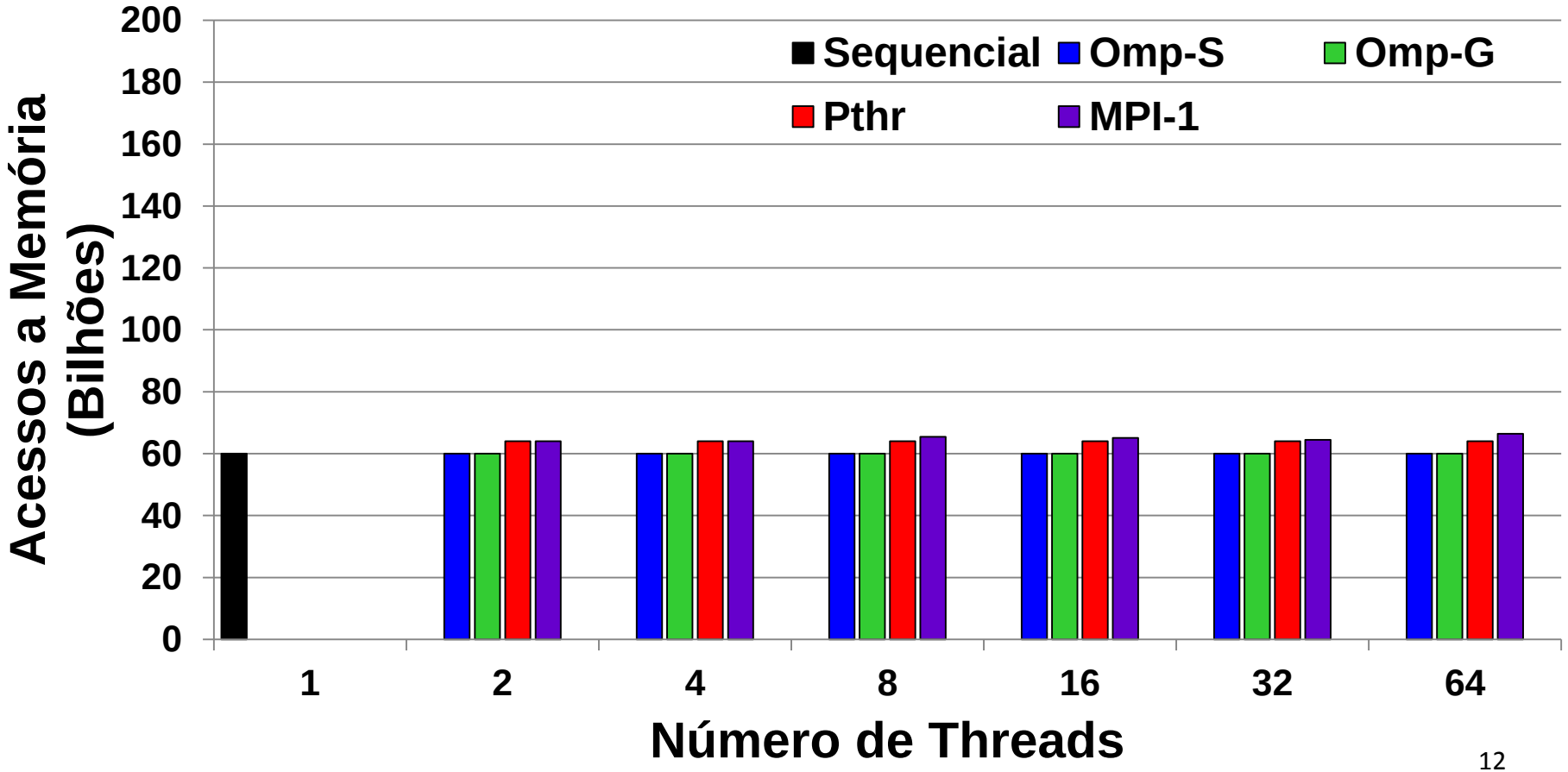
Resultados: Nº de Instruções Executadas

- Cada IPP tem sua particularidade de execução
  - OpenMP teve 2% menor número de instruções, devido otimização da *flag -fopenmp*



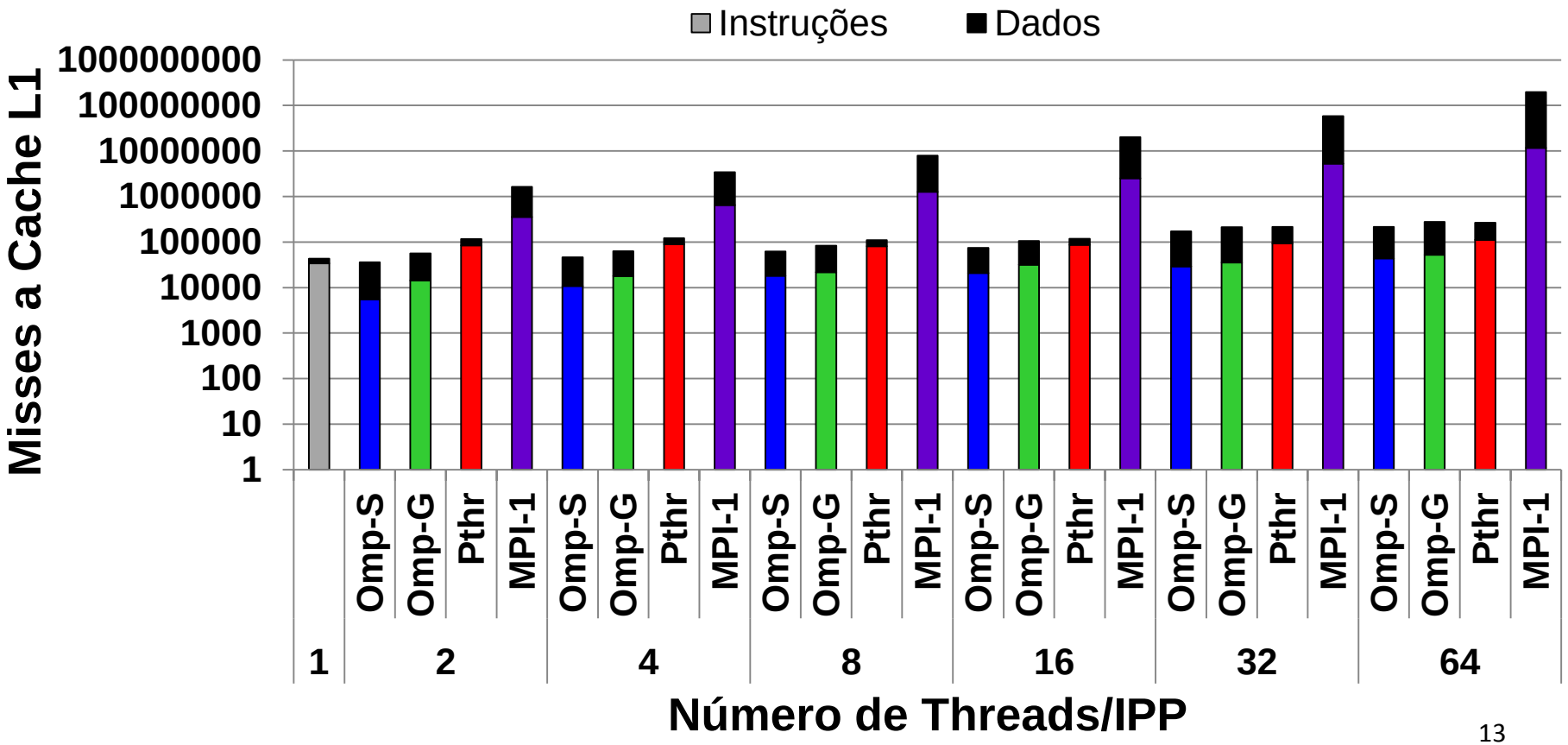
Resultados: Acessos a *cache* L1

- Gerenciamento de threads/processos
  - Tipo da aplicação



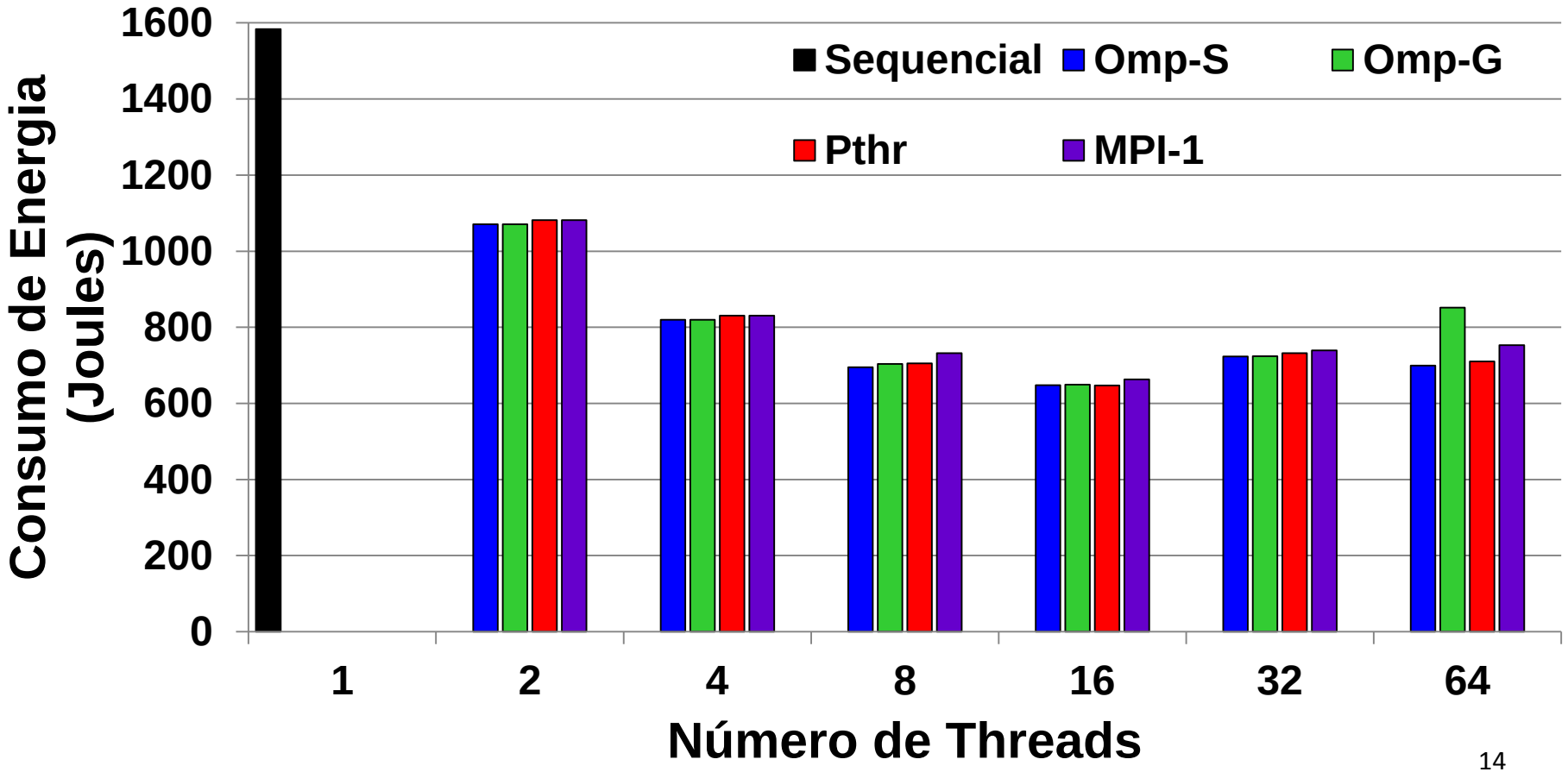
Resultados: Misses a *cache* L1

- *OpenMP* faz espera ativa (*Busy-Waiting)*
- MPI_Init desencadeia maior taxa de misses



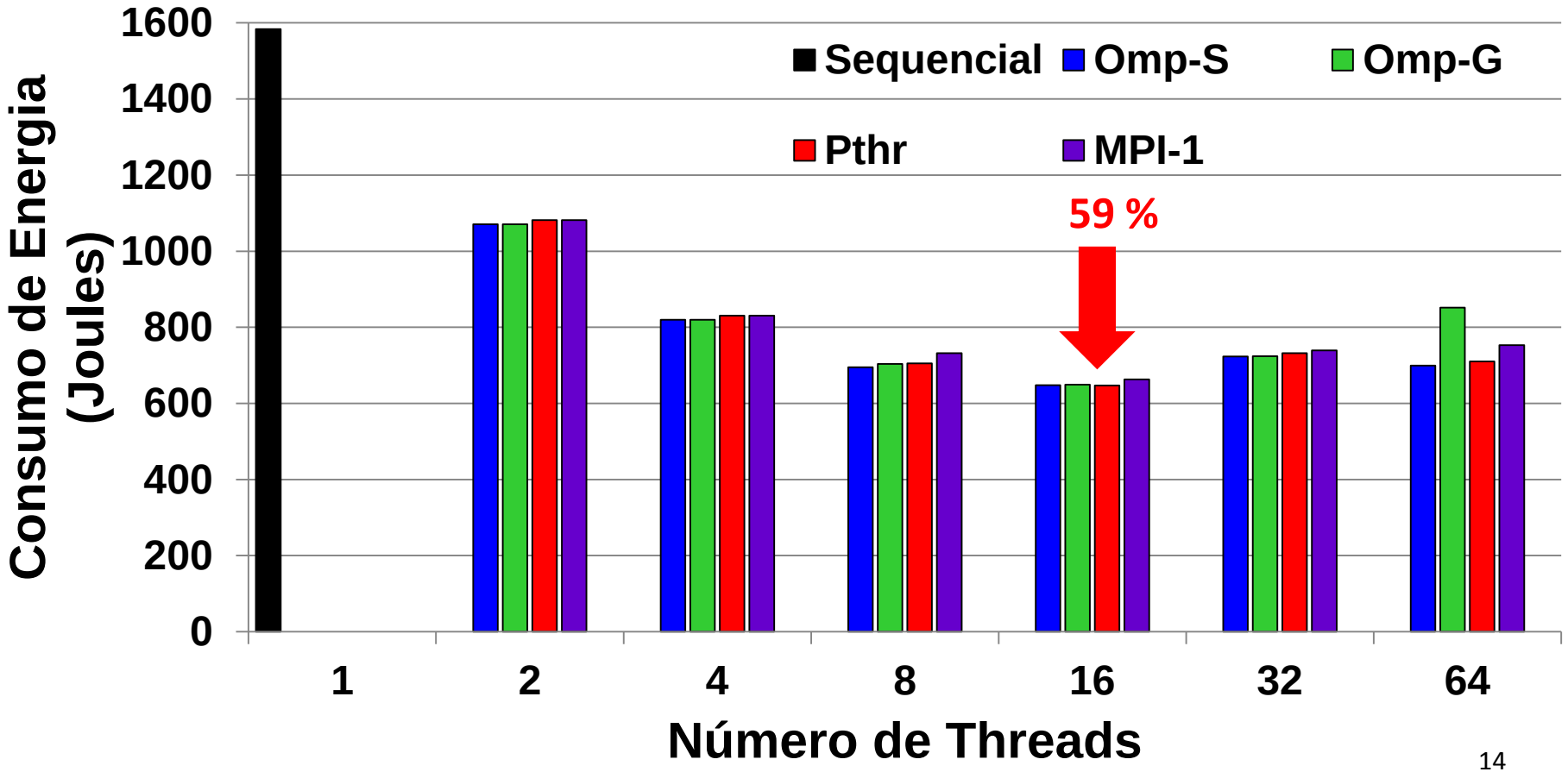
Resultados: Consumo de Energia

- Paralelização obteve menor consumo de energia.
- Melhor Resultado = Pthreads (59%)



Resultados: Consumo de Energia

- Paralelização obteve menor consumo de energia.
- Melhor Resultado = Pthreads (59%)



Conclusão

- A utilização de IPPs melhorou tempo de execução da aplicação

- Com a utilização de 16 threads, utilizando Pthreads, obteve-se um consumo de energia 59% menor em relação ao sequencial

- Utilizando 32 e 64 threads o tempo de execução foi menor, porem o consumo de energia aumentou.

- *Misses* da *cache* L1 aumentam de acordo com o número de *threads*
  - OpenMP e MPI tiveram um maior número de misses de dados em relação ao sequencial

Trabalhos Futuros

- Executar outros benchmarks do tipo CPU-Bound e IO-Bound

- Expandir o escopo das IPP, utilizando MPI-2

- Expandir o escopo de acessos a memória e misses com a coleta de dados da cache L2 e L3.

Referências

• Intel (2014). IntelR xeonR processor (e5-product family) datasheet – volume 2. http://www.intel.com/content/www/us/en/processors/xeon/xeon-e5-1600-2600-vol-2-datasheet.html.

•The Innovative Computing Laboratory (2014). Papi - overview. http://icl.cs.utk.edu/papi/overview/index.html.

• Karlinski, T. R., Lorenzon, A., F., A. C. B., and Cera, M. (2014). **Análise de desempenho e consumo energético de uma aplicação openmp**. *In Workshop de Iniciação Científica do WSCAD 2014*, São Jóse dos Campo/SP.

• Lorenzon, A., Cera, M., and Schneider Beck, A. (2014). **Performance and energy evaluation of dif. multi-threading interf. in embedded and general purpose syst. Journal of Signal Processing Systems**.

• Korthikanti, V. A. and Agha, G. (2010). **Towards optimizing energy costs of algorithms for shared memory architectures**. In auf der Heide, F. M. and Phillips, C. A., editors,SPAA, ACM.

Agradecimentos

- Universidade Federal do Pampa – Campus Alegrete.
  - Apoio no desenvolvimento das atividades de pesquisa.
  - Apoio Infraestrutura do Campus Alegrete – Grupo LEA.

- Bolsa de Iniciação a Pesquisa - Edital PBDA/UNIPAMPA 2014.

Análise de Desempenho e Consumo Energético de uma Aplicação OpenMP

Muito Obrigado!

Perguntas

Thayson R. Karlinski, Arthur F. Lorenzon, Antonio C. Beck, Márcia C. Cera.

thaysonr.karlinski@gmail.com, aflorenzon@inf.ufrgs.br, caco@inf.ufrgs.br, marciacera@unipampa.edu.br

unipampa

Universidade Federal do Pampa

UFRGS

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL