

Introdução ao NAS Parallel Benchmarks

Performance Relativa de Kernels Sequenciais, em ambiente de Memória Partilhada e ambiente de Memória Distribuída

Filipe Costa Oliveira

UCE: Engenharia de Sistemas de Computação
Mestrado Integrado em Engenharia Informática
Departamento de Informática

Universidade do Minho, 2016

Introdução – Contextualização das Benchmarks

As "NAS Parallel Benchmarks" englobam 5 kernels (EP, MG, CG, FT, IS) e 3 aplicações que simulam dinâmica de fluídos (LU,SP,BT). Temos por interesse os 5 kernels:

- **EP**: implicitamente embaraçosamente paralelo. (espectável obtermos os melhores resultados de performance neste kernel)
- **MG**: implica uma elevada comunicação para a resolução do algoritmo.
- **CG**: testa computação e comunicação não estruturada, sendo portanto expectável uma fraca performance deste kernel quando em comparação com o **EP**.
- **FT**: excluído do caso de estudo em detrimento do **MG**.
- **IS**: testa tanto a capacidade de computação de um sistema em termos de operações sobre inteiros, assim como a performance de comunicação do mesmo.

Caracterização do Hardware do ambiente de testes e dimensão das diferentes classes de dados

Ambiente de Clustering Search¹ @ Universidade do Minho.

- grande porção dos nós de computação com configurações de hardware relativamente homogéneas ² **(28 dos 54 nós disponíveis)**
- inclusão de nós do tipo 662, 652, 641, e 431 **(abrangendo 33 dos 54 nós disponíveis)**.
 - preservam características entre eles fundamentais para a possibilidade de comparação (p.e. suporte da rede Myrinet 10Gbps).
 - englobam como requerido mais do que uma classe de arquitectura existente no Search6.

Table: Dimensão do dataset para as diferentes Classes e Benchmarks

Bench.	data type	S	A	B	C
EP	double	128 MB	2 GB	8 GB	32 GB
MG	double	256 KB	128 MB	128 MB	1024 MB
CG	double	15MB	1,46 GB	41,91 GB	167,64 GB
IS	integer	256 KB	32 MB	128 MB	512 MB

¹Services and Advanced Research Computing with HTC/HPC clusters

²p.e. mesma família de processadores - Ivy Bridge

Influência de:

- Diferentes ferramentas de compilação (**GCC compiler suite** e o **Intel compilers suite**):
 - icc versão 13.0.1 (gcc version 4.4.6 compatibility)
 - gcc versão 4.4.6
 - gcc versão 4.9.0 (versão default no nosso ambiente de clustering)
 - **sem otimização e flags de compilação -O2 e -O3**
- Diferentes configurações de ferramentas de comunicação:
 - Gigabit Ethernet
 - Myrinet 10Gbps
- Kernels Sequenciais(SEQ), em ambiente de Memória Partilhada(OMP) e ambiente de Memória Distribuída(MPI)

Benchmarking em ambiente sequencial – NPB SEQ

- Kernels compilados com o compilador gcc versão 4.9.0 com flag de compilação -O3 obtiveram os melhores resultados;
- Máximo de Milhões de FP Operations alcançado para o kernel SEQ - MG, classe de dados C, para compilador gcc v4.9.0 com flag de compilação -O3.
- Nós do tipo compute-662 obtiveram os valores mais baixos de FP Operations por segundo (figura 1).
- Compilador icc tem melhor performance nos kernels IS e EP (sem dependência de dados).
- Nós do tipo compute-651 e compute-431 obtiveram os melhores valores relativamente ao tempo de computação. (figura 2).

Benchmarking em ambiente sequencial – Análise FP Operations/s

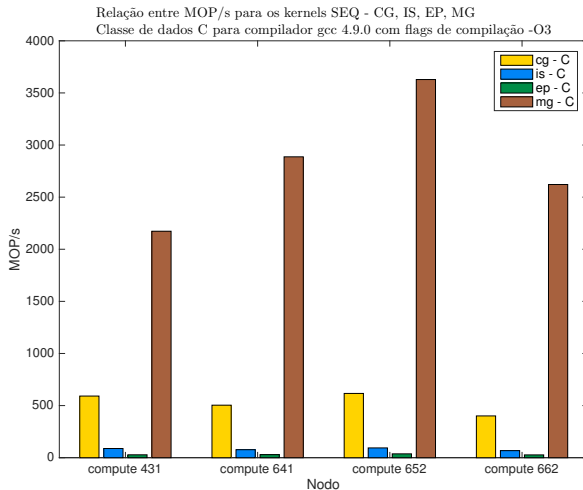


Figure: Milhões de FP Operations alcançado para os kernels SEQ - CG, IS, EP, e MG, classe de dados C, para compilador gcc 4.9.0 com flag de compilação -O3

Benchmarking em ambiente sequencial – Análise T. Total Solução

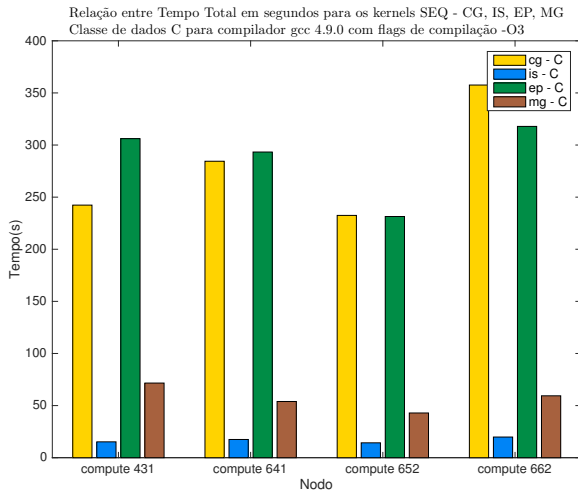


Figure: Milhões de FP Operations alcançado para os kernels SEQ - CG, IS, EP, e MG, classe de dados C, para compilador gcc 4.9.0 com flag de compilação -O3

Benchmarking em Amb. de Memória Partilhada – NPB OMP

- Criados casos de teste que superassem o número de threads possíveis de correrem concorrente por tipo de CPU.
- Tempo total p/solução apresenta uma redução para todo os incrementos no número de threads presente na solução até ao número de threads openMP superar o número de threads disponíveis por CPU.
- Nós do tipo compute-662 obtiveram os melhores valores relativamente ao tempo total para solução (maior #threads).
- Nós do tipo compute-662 obtiveram a maior relação de ganho vs SEQ. (maior #threads e fraco resultado SEQ).
 - kernel EP apresenta melhor resultado (aprox 32)
 - kernel MG apresenta melhor resultado (aprox 7)
- Estudada influência de diferentes flags de compilação (-O2 / -O3) e diferentes versões de compilador (gcc 4.4.6 / gcc.4.9.0) para as máquinas com melhor resultado OMP (nós compute-662).

Benchmarking Amb. Mem. Partilhada - Relação de Ganho - EP

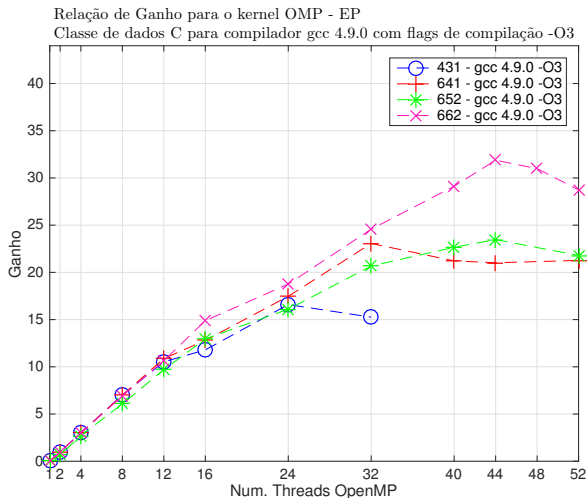


Figure: Relação de ganho para o kernel OMP - EP, classe de dados C para compilador gcc 4.9.0 com flag de compilação -O3

Benchmarking Amb. Mem. Partilhada - Relação de Ganho - CG

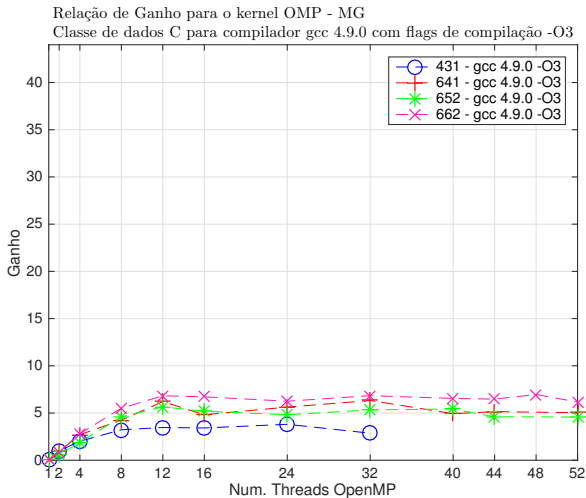


Figure: Relação de ganho para o kernel OMP - CG, classe de dados C para compilador gcc 4.9.0 com flag de compilação -O3

- Criados casos de teste que envolveram 2 e 4 máquinas distintas do mesmo tipo.
 - 2 máquinas envolvidas na solução teremos em teste 8 e 16 processos MPI.
 - 4 máquinas envolvidas na solução teremos em teste 32 processos MPI.
 - (aconselhado #processos comunicantes via Myrinet 10Gbps não ultrapassasse os 8).
- Testadas diferentes formas de comunicação (Gigabit Ethernet e Myrinet 10Gbps).
- Limitado o ambiente de teste em memória distribuída aos nós do tipo compute-641 e compute-431.
- Todos os kernels em estudo a adição de nós de computação e número de processos MPI envolvidos para solução **diminui o tempo de computação** quando a comunicação entre processos é feita via **Myrinet 10Gbps**. Para os casos de comunicação via **Gigabit Ethernet** o tempo de solução **piora**.

Relação entre tempo total solução - SEQ vs OMP vs MPI

Relação entre Tempo Total em segundos para os kernels SEQ/OMP/MPI - CG, IS, EP, MG, Nós 641
Classe de dados C para compiladores gcc 4.9.0 e icc 13.0.1 com flag de compilação -O3
Para comunicação Gigabit Ethernet e Myrinet 10Gbps

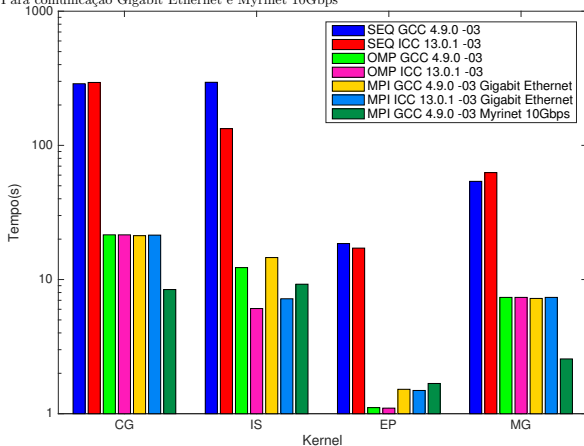


Figure: Relação entre tempo total em segundos para a solução para os kernels SEQ/OMP/MPI - CG, IS, EP, MG, nós 641, classe de dados C para compiladores gcc 4.9.0 e icc 13.0.1 com flag de compilação -O3, para comunicação Gigabit Ethernet e Myrinet 10Gbps

- Kernels EP e IS a melhor solução: ambiente de memória partilhada com paralelismo via threads openMP.
- Kernels CG e MG a melhor solução: ambiente de memória distribuída com comunicação via Myrinet 10Gbps.
- A solução em ambiente de memória distribuída é sempre melhor com comunicação via Myrinet 10Gbps
 - latência de comunicação: comunicação via Gigabit Ethernet ronda valores entre os 10 e 100 μ s.
 - latência de comunicação: via Myrinet 10Gbps ronda valores entre os 1 e 10 μ s.
 - Analisemos essa suposição teórica na figura 6.

Benchmarking Amb. Mem. Partilhada - Relação de Ganho - ETH vs MX

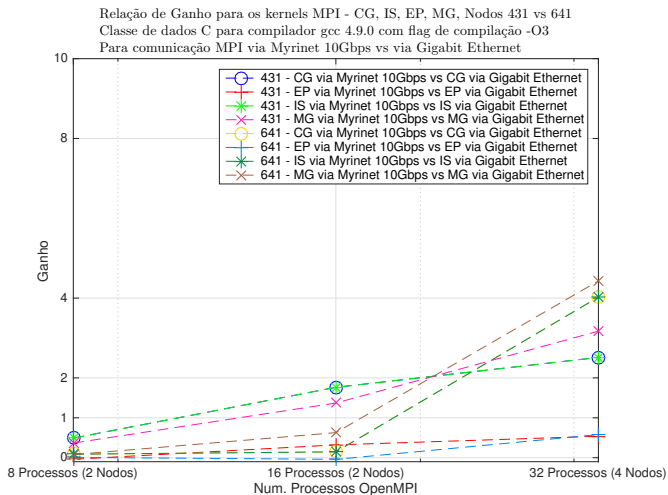


Figure: Relação de ganho para os kernels MPI - CG, IS, EP, MG, nós 431 e 641, classe de dados C para compilador gcc 4.9.0 com flag de compilação -O3, para comunicação Gigabit Ethernet vs Myrinet 10Gbps

Relação de ganho vs SEQ – kernels OMP e MPI

Relação de Ganho para os kernels OMP/MPI - CG, IS, EP, MG, Nós 641

Classe de dados C para compiladores gcc 4.9.0 e icc 13.0.1 com flag de compilação -O3

Para comunicação MPI via Gigabit Ethernet e Myrinet 10Gbps

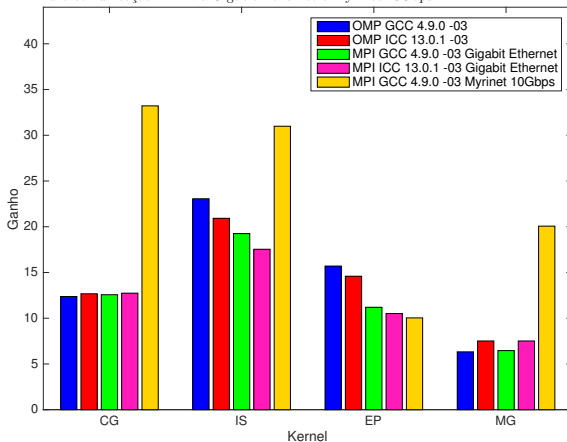


Figure: Relação de ganho para os kernels OMP/MPI - CG, IS, EP, MG, nós 641, classe de dados C para compilador gcc 4.9.0 e icc 13.0.1 com flag de compilação -O3, para comunicação(apenas kernels MPI) Gigabit Ethernet vs Myrinet 10Gbps

Relação de ganho vs SEQ – kernels OMP e MPI – conclusões

Melhores soluções do conjunto de testes realizados são:

- **CG** - Kernel em memória distribuída com comunicação via Myrinet 10Gbps (4 nós 32 Processos), compilador gcc 4.9.0 flag -O3;
- **IS** - Kernel em memória partilhada com n° de processos openMP igual ao número de threads disponível, compilador gcc 4.9.0 flag -O3;
- **EP** - Kernel em memória partilhada com n° de processos openMP igual ao número de threads disponível, compilador gcc 4.9.0 ou compilador icc 13.0.1 flag -O3;
- **MG** - Kernel em memória distribuída com comunicação via Myrinet 10Gbps (4 nós 32 Processos), compilador gcc 4.9.0 flag -O3;

Análise de propriedades para monitorização de execução do mesmo kernel para diferentes implementações SEQ, OMP e MPI

- As **melhores soluções** consomem quase a totalidade da capacidade de computação dos nós.
 - o oposto, no nosso caso de estudo, implica que existe uma melhor solução alternativa.
- Kernels com melhor solução implementada em **ambiente de memória partilhada** apresentam um maior número de **interrupções do sistema e context swap** (EP e IS).
- Kernels que impliquem uma elevada comunicação para a resolução do algoritmo apresentam a melhor solução em **ambiente de memória distribuída** (CG e MG).
- Nenhum dos kernels recorre intensivamente a leitura e escrita em disco por grandes períodos, centrando os mesmos ora no início ou término do mesmo.
- Resultados do teste de performance mantêm-se inalterados independentemente da sequência de "n-testes" realizados.

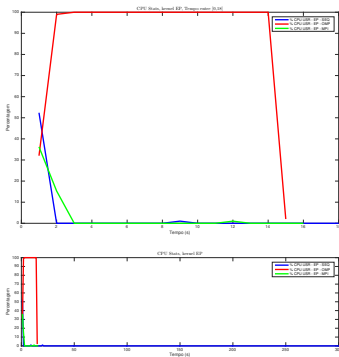


Figure: Relação de % de tempo de CPU para os kernels SEQ/OMP/MPI - EP, nós 641, classe de dados C para compilador gcc 4.9.0 com flag de compilação -O3, para comunicação Myrinet 10Gbps (memória partilhada)

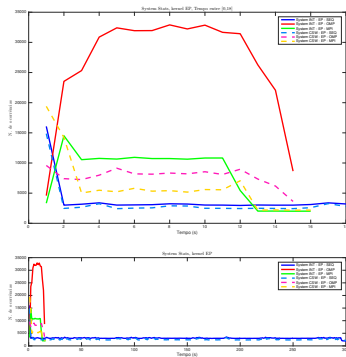


Figure: Relação de número de interrupções de processo e número de "context swap" para os kernels SEQ/OMP/MPI - EP, nós 641, classe de dados C para compilador gcc 4.9.0 com flag de compilação -O3, para comunicação Myrinet 10Gbps (memória partilhada)

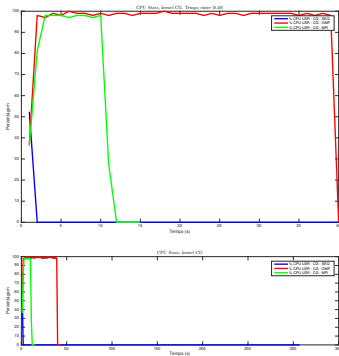


Figure: Relação de % de tempo de CPU para os kernels SEQ/OMP/MPI - CG, nós 641, classe de dados C para compilador gcc 4.9.0 com flag de compilação -O3, para comunicação Myrinet 10Gbps (memória distribuída)

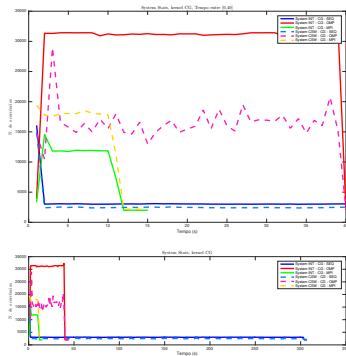


Figure: Relação de número de interrupções de processo e número de "context swap" para os kernels SEQ/OMP/MPI - CG, nós 641, classe de dados C para compilador gcc 4.9.0 com flag de compilação -O3, para comunicação Myrinet 10Gbps (memória distribuída)

Introdução ao NAS Parallel Benchmarks

Performance Relativa de Kernels Sequenciais, em ambiente de Memória Partilhada e ambiente de Memória Distribuída

Filipe Costa Oliveira

UCE: Engenharia de Sistemas de Computação
Mestrado Integrado em Engenharia Informática
Departamento de Informática

Universidade do Minho, 2016