LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

Escola Supercomputador SDUMONT

Introdução E/S Paralela no SDUMONT
Atividades

André Ramos Carneiro (andrerc@lncc.br)
Bruno Alves Fagundes (brunoaf@lncc.br)

BT-IO Benchmark

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- BT-IO (https://www.nas.nasa.gov/publications/npb.html):
- BT - Block Tri-diagonal solver with test of different parallel I/O techniques
- Modo: Full mpi-io → Operações coletivas N-1 (possui epio: file-per-proc N-N)
- Executado com a classe:
  - B: Timestep 0.0003 | Intervalo de escrita 5s | Tamanho do arquivo 1,7 GB
  - C: Timestep 0.0001 | Intervalo de escrita 5s | Tamanho do arquivo 6,8 GB
- Número de processos
  - 1
  - 36 (precisa ser uma raiz quadrada)
- Na saída gera um resumo sobre o desempenho:

```
BTIO -- statistics:
   I/O timing in seconds    :           0.88
   I/O timing percentage    :          23.21
   Total data written (MB)  :        1697.93
   I/O data rate  (MB/sec)  :        1923.96
 BT Benchmark Completed.
...
 Time in seconds =                     3.80
```

Preparação

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- Conectar no SDumont com o ssh
  - `ssh username@login.sdumont.lncc.br`
- Utilizaremos o OpenMPI 4.0.1
  - `module load openmpi/gnu/4.0.1`
- Acessar o diretório de trabalho através da variável de ambiente `$SCRATCH`
  - `cd $SCRATCH`
- Download do pacote:
  - http://www.lncc.br/~andrerc/intro_ES_SD-2021.tar.gz
  - http://www.lncc.br/~andrerc/intro_ES_SD-2021.zip
  - http://www.cenapad-rj.lncc.br/tutoriais/materiais-hpc/semana-sdumont
- Entrar no diretório `intro_ES_SD`
- Diretório padrão do minicurso para os exercícios:
  - `$SCRATCH/intro_ES_SD`

Atividade 1

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- Executar o BT-IO B serial com a configuração padrão de stripes (count = 1 e size = 1MiB)

1) Acessar o diretório das atividades deste minicurso
   - `cd $SCRATCH/intro_ES_SD`
2) Criar o diretório "`exec_dir_1`" e listar as informações sobre o stripe do diretório
   - `mkdir exec_dir_1`
   - `lfs getstripe exec_dir_1`

     ```
     exec_dir_1
     stripe_count: 1 stripe_size: 1048576 pattern: 0 stripe_offset: -1
     ```
3) Entrar no diretório "`exec_dir_1`"
   - `cd exec_dir_1`
4) Submeter o script `run_btio-B.sh`:
   - `sbatch ../run_btio-B.sh`
5) Listar as informações sobre o stripe do arquivo de saída
   - `lfs getstripe btio.full.out`
6) Lembre-se de ao final remover o arquivo de saída gerado pelo BT-IO
   - `rm btio.full.out`

Atividade 2

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- Executar o BT-IO B serial em um diretório com o número de stripes igual a 10

1) Acessar o diretório das atividades deste minicurso
   - `cd $SCRATCH/intro_ES_SD`
2) Criar o diretório "`exec_dir_2`" para a execução da segunda atividade
   - `mkdir exec_dir_2`
3) Alterar o número de stripes do diretório criado no passo anterior para 10
   - `lfs setstripe --stripe-count 10 exec_dir_2`
4) Listar as informações do stripe para esse diretório
   - `lfs getstripe exec_dir_2/`

   ```
   exec_dir_2/
   stripe_count: 10 stripe_size: 1048576 pattern: raid0 stripe_offset: -1
   ```
5) Entrar nesse diretório
   - `cd exec_dir_2`
6) Submeter o script de execução utilizado na atividade anterior
   - `sbatch ../run_btio-B.sh`

Atividade 2

7) Listar as informações sobre o stripe do arquivo de saída. Houve diferença em relação ao exercício anterior?

```
lfs getstripe exec_dir_2/btio.full.out
exec_dir_2/btio.full.out
lmm_stripe_count:  10
lmm_stripe_size:   1048576
lmm_pattern:       raid0
lmm_layout_gen:    0
lmm_stripe_offset: 0
    obdidx       objid          objid          group
         0         1279521461     0x4c43f2b5                0
         8         1287150163     0x4cb85a53                0
         2         1279745967     0x4c475faf                0
         9         1276804253     0x4c1a7c9d                0
         1         1255860164     0x4adae7c4                0
         7         1288210886     0x4cc889c6                0
         5         1255968956     0x4adc90bc                0
         3         1279465720     0x4c4318f8                0
         6         1283337802     0x4c7e2e4a                0
         4         1277880817     0x4c2ae9f1                0
```

Atividade 2

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

8) Comparar o resultado dessa execução com a da atividade anterior. Houve alteração?
   - `cd $SCRATCH/intro_ES_SD`
   - `diff -y -W 200 exec_dir_1/slurm-<JOBID>.out exec_dir_2/slurm-<JOBID>.out`

9) Lembre-se de ao final remover o arquivo de saída gerado pelo BT-IO
   - `rm exec_dir_2/btio.full.out`

Atividade 2 - RESPOSTA

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- Executar o BT-IO B serial em um diretório com o número de stripes igual a 10

Atividade 1 - serial + 1 stripe	Atividade 2 - serial + 10 stripe
BTIO -- statistics: I/O timing in seconds : 2.83 I/O timing percentage : 0.86 Total data written (MB) : 1697.93 **I/O data rate (MB/sec) : 600.48** BT Benchmark Completed. Class = B Size = 102x 102x 102 Iterations = 200 Time in seconds = 328.07	BTIO -- statistics: I/O timing in seconds : 2.39 I/O timing percentage : 0.73 Total data written (MB) : 1697.93 **I/O data rate (MB/sec) : 709.16** BT Benchmark Completed. Class = B Size = 102x 102x 102 Iterations = 200 Time in seconds = 327.58

- É esperado um aumento de no throughput!

Atividade 3

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

Implementar coleta de estatísticas com o darshan

- Alterar o arquivo run_btio-C.sh
- Carregar o módulo para a versão do openmpi utilizada:
  `module load darshan/3.2.1_openmpi_gnu_4.0.1`
- Definir a variável DARSHAN_LOGPATH com o diretório onde serão armazenados os arquivos de log do darshan:
  `export DARSHAN_LOGPATH=$SLURM_SUBMIT_DIR`

Atividade 3

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

```
#!/bin/bash
#SBATCH --nodes=2                       # here the number of nodes
#SBATCH --ntasks=36                     # here total number of mpi tasks
#SBATCH --ntasks-per-node=18            # here ppn = number of process per nodes
#SBATCH -p treinamento                  # target partition
#SBATCH -J NPB-BTIO-C-36                # job name
#SBATCH --exclusive                     # to have exclusvie use of your nodes

echo $SLURM_JOB_NODELIST
cd $SLURM_SUBMIT_DIR

echo ""
echo "***********"
echo ""

####################################
#          COMPILER                #
####################################
module load openmpi/gnu/4.0.1

module load darshan/3.2.1_openmpi_gnu_4.0.1
export DARSHAN_LOGPATH=$SLURM_SUBMIT_DIR

#Configure BTIO executable
EXEC=/scratch/app/NPB3.3.1-MZ/bin/bt.C.36.mpi_io_full.ompi40+gnu

#Start the benchmark
srun $EXEC
```

Atividade 3

- Executar o BT-IO C com 2 nós e 18 processos por nó (total de 36 processos) em um diretório com a configuração padrão de stripes.

1) Acessar o diretório das atividades deste minicurso
   - `cd $SCRATCH/intro_ES_SD`
2) Criar o diretório "`exec_dir_3`" para a execução da segunda atividade
3) Listar as informações do stripe para esse diretório
   - `mkdir exec_dir_3`

     `lfs getstripe exec_dir_3`

     `exec_dir_1`

     `stripe_count: 1 stripe_size: 1048576 pattern: 0 stripe_offset: -1`
4) Entrar nesse diretório
   - `cd exec_dir_3`
5) Submeter o script `run_btio-C.sh`:
   - `sbatch ../run_btio-C.sh`

Atividade 3

6) Listar as informações sobre o stripe do arquivo de saída.
   - `lfs getstripe btio.full.out`

```
$ lfs getstripe btio.full.out
btio.full.out
lmm_stripe_count:  1
lmm_stripe_size:   1048576
lmm_pattern:   raid0
lmm_layout_gen:0
lmm_stripe_offset: 7
    obdidx       objid       objid       group
         7      1288257040     0x4cc93e10                  0
```

7) Lembre-se de ao final remover o arquivo de saída gerado pelo BT-IO
   - `rm btio.full.out`

Atividade 4

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- Executar o BT-IO C com 2 nós e 18 processos por nó (total de 36 processos) em um diretório com o número de stripes igual a 2

1) Acessar o diretório das atividades deste minicurso
   - `cd $SCRATCH/intro_ES_SD`
2) Criar o diretório "`exec_dir_4`" para a execução da segunda atividade e alterar o número de stripes para 2
   - `mkdir exec_dir_4`
   - `lfs setstripe -c 2 exec_dir_4`
3) Listar as informações do stripe para esse diretório
   - `lfs getstripe exec_dir_4/`

```
exec_dir_4/
stripe_count:  2 stripe_size:   1048576 pattern:     raid0
stripe_offset: -1
```

4) Entrar nesse diretório
   - `cd exec_dir_4`

Atividade 4

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

5) Submeter o script `run_btio-C.sh`:
   - `sbatch ../run_btio-C.sh`
6) Listar as informações sobre o stripe do arquivo de saída.
   - `lfs getstripe btio.full.out`

```
btio.full.out
lmm_stripe_count:  2
lmm_stripe_size:   1048576
lmm_pattern:       raid0
lmm_layout_gen:0
lmm_stripe_offset: 5
    obdidx       objid       objid       group
         5      1256019014     0x4add5446                  0
         3      1279518804     0x4c43e854                  0
```

7) Comparar o resultado dessa execução com a da atividade anterior. Houve alteração?
   - `cd $SCRATCH/intro_ES_SD`
   - `diff -y -W 200 exec_dir_3/slurm-<JOBID>.out exec_dir_4/slurm-<JOBID>.out`
8) Lembre-se de ao final remover o arquivo de saída gerado pelo BT-IO
   - `rm exec_dir_4/btio.full.out`

Atividade 4 - RESPOSTA

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- Executar o BT-IO C com 2 nós e 18 processos por nó (total de 36 processos) em um diretório com o número de stripes igual a 2

Atividade 3 - paralelo + 1 stripe	Atividade 4 - paralelo + 2 stripe
BTIO -- statistics: **I/O timing in seconds : 15.71** I/O timing percentage : 25.67 Total data written (MB) : 6802.44 **I/O data rate (MB/sec) : 432.89** BT Benchmark Completed. Class = C Size = 162x 162x 162 Iterations = 200 Time in seconds = 61.22	BTIO -- statistics: **I/O timing in seconds : 8.91** I/O timing percentage : 16.39 Total data written (MB) : 6802.44 **I/O data rate (MB/sec) : 763.61** BT Benchmark Completed. Class = C Size = 162x 162x 162 Iterations = 200 Time in seconds = 54.36

- Aumento de mais de 300 MiB/s no thoughput e redução quase que pela metade no tempo de I/O

Atividade 5

- Executar o BT-IO C com 4 nós e 9 processos por nó (total de 36 processos) em um diretório com o número de stripes igual a 4

1) Acessar o diretório das atividades deste minicurso
   - `cd $SCRATCH/intro_ES_SD`
2) Editar o script `run_btio-C.sh` para utilizar 4 nós e 9 processos/nó
   - `#SBATCH --nodes=4`
   - `#SBATCH --ntasks-per-node=9`
3) Criar o diretório "`exec_dir_5`" para a execução da atividade e alterar o número de stripes para 4
   - `mkdir exec_dir_5`
   - `lfs setstripe -c 4 exec_dir_5`
4) Listar as informações do stripe para esse diretório
   - `lfs getstripe exec_dir_5`

```
exec_dir_5
stripe_count: 4 stripe_size: 1048576  pattern: raid0 stripe_offset: -1
```

5) Entrar no diretório
   - `cd exec_dir_5`

Atividade 5

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

6) Submeter o script `run_btio-C.sh`:
   - `sbatch ../run_btio-C.sh`
7) Listar as informações sobre o stripe do arquivo de saída.
   - `lfs getstripe btio.full.out`

```
btio.full.out
lmm_stripe_count:  4
lmm_stripe_size:   1048576
lmm_pattern:   raid0
lmm_layout_gen:0
lmm_stripe_offset: 6
    obdidx       objid       objid       group
         6      1283403114     0x4c7f2d6a                 0
         4      1277948841     0x4c2bf3a9                 0
         0      1279589210     0x4c44fb5a                 0
         8      1287218261     0x4cb96455                 0
```

8) Comparar o resultado dessa execução com as das atividades 3 e 4. Houve alteração?
9) Lembre-se de ao final remover o arquivo de saída gerado pelo BT-IO
   - `rm exec_dir_5/btio.full.out`

Atividade 5

- Para sumarizar as estatísticas do darshan em um arquivo pdf siga os passos abaixo:

10) Carregar o módulo
   - `module load darshan/darshan/3.2.1_openmpi_gnu_4.0.1`

11) Executar o script darshan-job-summary.pl
   - `cd $SCRATCH/intro_ES_SD`
   - `darshan-job-summary.pl exec_dir_3/arquivo-log.darshan`
   - `darshan-job-summary.pl exec_dir_4/arquivo-log.darshan`
   - `darshan-job-summary.pl exec_dir_5/arquivo-log.darshan`

Atividade 5 - RESPOSTA

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

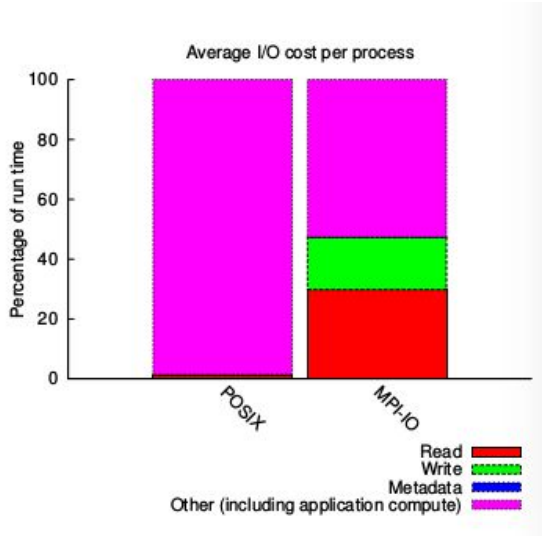
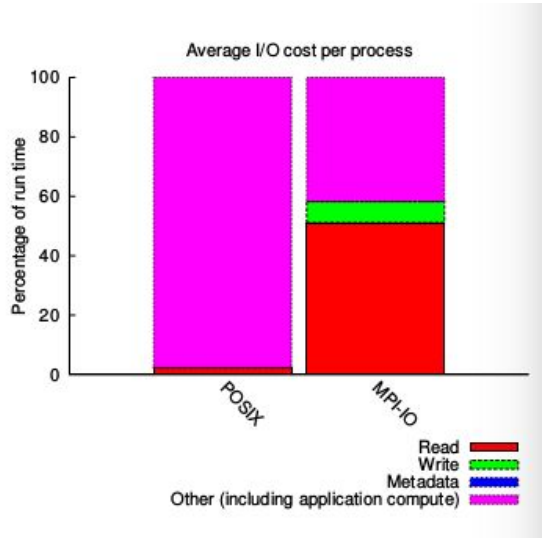
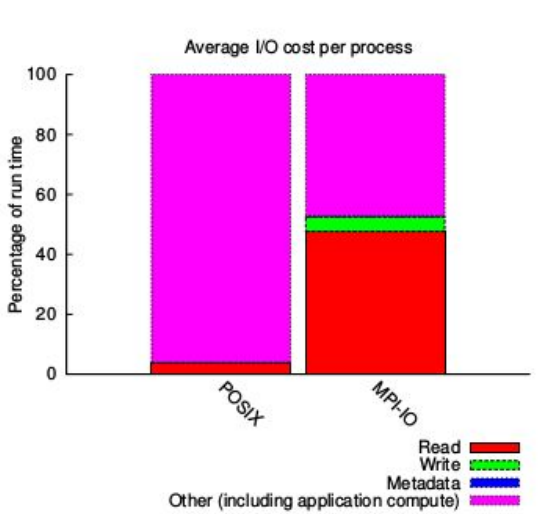
- Comparativo dos resultados

Atividade 3 - paralelo + 1 stripe	Atividade 4 - paralelo + 2 stripe	Atividade 5 - paralelo + 4 stripe
BTIO -- statistics: **I/O timing in seconds : 15.71** I/O timing percentage : 25.67 **I/O data rate (MB/sec):432.89** Time in seconds = 61.22	BTIO -- statistics: **I/O timing in seconds : 8.91** I/O timing percentage : 16.39 **I/O data rate (MB/sec):763.61** Time in seconds = 54.36	BTIO -- statistics: **I/O timing in seconds : 4.82** I/O timing percentage : 10.10 **I/O data rate (MB/sec): 1410** Time in seconds = 47.97

- Duplicou o throughput e reduziu o tempo gasto com I/O pela metade

Atividade 5 - RESPOSTA

- Nos gráficos gerados pelo darshan é possível observar uma redução no tempo gasto com as operações de escrita utilizando o MPI-IO

Atividade 3 - paralelo + 1 stripe	Atividade 4 - paralelo + 2 stripe	Atividade 5 - paralelo + 4 stripe
		

IOR Benchmark

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- IOR (https://github.com/hpc/ior):
- Utilizado para testar vários padrões de acesso e simular o comportamento de I/O
- Configurado para bloco de transferência de **1 MiB** e **100 segmentos**
- Gerando arquivo de 4,69 GiB
- Na saída gera um resumo sobre o desempenho:

```
Summary of all tests:
Operation   Max(MiB)   Min(MiB)  Mean(MiB) … segcnt   blksiz    xsize aggs(MiB)   API
write         828.84     828.84     828.84 …    100  1048576  1048576    4800.0 MPIIO
read          702.25     702.25     702.25 …    100  1048576  1048576    4800.0 MPIIO
```

Atividade 6

- Executar o IOR com 2 nós e 24 processos por nó (total de 48 processos) alterando as *hints* do **ROMIO**

1) Acessar o diretório das atividades deste minicurso
   - `cd $SCRATCH/intro_ES_SD`
2) Criar o diretório "`exec_dir_6`" para a execução da atividade
   - `mkdir exec_dir_6`
3) Alterar o número de stripes do diretório criado no passo anterior para 4
   - `lfs setstripe -c 4 exec_dir_6`
4) Entrar no diretório criado
   - `cd exec_dir_6`
5) Submeter o script `run_ior_romio.sh`
   - `sbatch ../run_ior_romio.sh`

Atividade 6

LNCC
www.lncc.br
Laboratório Nacional de Computação Científica

- Executar o IOR com 2 nós e 24 processos por nó (total de 48 processos) alterando as *hints* do **ROMIO**

6) Agora, vamos editar o arquivo de hints `$SCRATCH/intro_ES_SD/hints.txt` para:
   - aumentar o número de agregadores por host para **2**

     ```
     cb_nodes 2
     ```
   - aumentar o tamanho do buffer intermediário do **CB** para **50MiB**

     ```
     cb_buffer_size 52428800
     ```

7) Editar o script `run_ior_romio.sh` para utilizar o arquivo de hints através da variável de ambiente e exibir todas as hints utilizadas na execução

   ```
   export ROMIO_HINTS=../hints.txt
   export ROMIO_PRINT_HINTS=1
   ```

8) Submeter o script `run_ior_romio.sh`
   - `sbatch ../run_ior_romio.sh`

9) Comparar o resultado dessa execução com a do passo 5. Houve diferença?

Atividade 6

- script run_ior_romio.sh

```
#Configure to use the ROMIO MPI-IO implementation
export OMPI_MCA_io=romio321

#Configure the environment variable to use the file hints
export ROMIO_HINTS=../hints.txt

#Configure the environment variable to show the used hints
export ROMIO_PRINT_HINTS=1
```

- Arquivo hints.txt

```
cb_buffer_size 52428800
cb_nodes 2
```

Atividade 6 - RESPOSTA

- Executar o IOR com 2 nós e 24 processos por nó (total de 48 processos) alterando as *hints* do **ROMIO**

Hints padrão	HInts modificada
`Max Write: 828 MiB/a` `Max Read: 631 MiB/a`	`Max Write: 1327 MiB/a` `Max Read: 841 MiB/a`

- Aumento de mais de 500 MiB/s no *thoughput* de escrita e 200 MiB/s para leitura

Dúvidas?