Application and System Benchmarks

Practical: High-Performance Computing System Administration

Silin Zhao

Supervisor: Marcus Merz

February 24, 2023

Outline

1. Benchmark of HPC

2. Examples of UoB-HPC

3. Hand-on testing

4. Benchmark exploring

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Background

- Indecator of Performance
- Measurement
- Scientific community and Industry

Measurement

- measurement
  - computation power -> TOP500
  - IO performance -> IO500
  - Energy consumption -> Green500
  - network connection
  - memory bandwidth
  - ...

Playground

- Well scalable?
- Regression?
- GPU support?
- Synthetic?
- Open source?
- ...

Abstact[1]

Architecture

- clxq
- a64fx
- romeq
- sk56
- arm

Compiler

- arm-21.3 (arm)
- arm-21.0 (arm)
- cce-10.0 (x86)
- cce-sve-10.0 (x86)
- fcc-4.3(*)
- gcc-8.1(*)
- gcc-11.0(*)
- llvm-11.0(*)

Benchmarks

- bude
- cloverleaf
- CP2k
- minifmm
- NAME
- Neutral
- OpenFOAM
- OenSBLI
- SNAP system
- ...

[1] University of Bristol High Performance Computing Group, https://github.com/UoB-HPC

Implementation

- IO500, HPL, HPCG, Stream are already implemented in GWDG
- Self implementation in SCC
- `https://pad.gwdg.de/s/w5_TJ9Yrp`

IO500[2]

geting code and installing

```
git clone  https://github.com/IO500/io500
cd io500
./prepare.sh
./io500 --list > config-all.ini
sbatch myjob.sh
```

runing

```
#!/bin/bash
#SBATCH --job-name test_benchmark
#SBATCH -N 1
#SBATCH -p fat
#SBATCH -n 1
#SBATCH --time=1:00:00

module purge
module load openmpi

mpiexec -np 1 ./io500 config-all.ini
```

turning

- scc, datadir(BeeGFS cluster), transferSize, blockSize...

[2] https://io500.org/

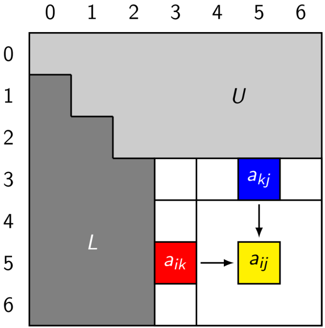
IO500

Result example

```
================================================================================
JobID = 14138671
User = hpctraining19, Account = all
Partition = fat, Nodelist = dsu002
================================================================================
IO500 version io500-isc22_v1 (standard)
[RESULT]       ior-easy-write        0.109581 GiB/s : time 336.524 seconds
ERROR INVALID (src/main.c:403) Runtime of phase (226.066966) is below stonewall tim
ERROR INVALID (src/main.c:409) Runtime is smaller than expected minimum runtime
[RESULT]    mdtest-easy-write        4.443261 kIOPS : time 226.067 seconds [INVALID
[      ]            timestamp        0.000000 kIOPS : time 0.000 seconds
[RESULT]       ior-hard-write        0.109578 GiB/s : time 336.483 seconds
[RESULT]    mdtest-hard-write        0.917004 kIOPS : time 301.056 seconds
[RESULT]                 find       74.936304 kIOPS : time 17.018 seconds
[RESULT]        ior-easy-read        0.095179 GiB/s : time 387.433 seconds
[RESULT]     mdtest-easy-stat        7.555278 kIOPS : time 133.385 seconds
[RESULT]        ior-hard-read        0.096507 GiB/s : time 382.050 seconds
[RESULT]     mdtest-hard-stat        7.567918 kIOPS : time 37.371 seconds
[RESULT]   mdtest-easy-delete        3.694371 kIOPS : time 272.073 seconds
[RESULT]     mdtest-hard-read        0.407664 kIOPS : time 675.925 seconds
[RESULT]   mdtest-hard-delete        3.808506 kIOPS : time 73.334 seconds
```

HPL (High-Performance Linpack)

- solves dense linear system (LU factorization)[3]
- double precision (64 bits)
- on distributed-memory system



[3] https://webspace.science.uu.nl/~bisse101/Book2/psc2_2.3.pdf

HPL

configuation

```
make arch=Make.MyHPL
OpenBLAS -> MPdir
OpenMPI  -> LAdir
```

/lib/bin/HPL.dat

```
6            device out (6=stdout,7=stderr,file)
4            # of problems sizes (N)
29 30 34 35  Ns
4            # of NBs
1 2 3 4      NBs
0            PMAP process mapping (0=Row-,1=Column-major)
3            # of process grids (P x Q)
2 1 4        Ps
2 4 1        Qs
16.0         threshold
3            # of panel fact
0 1 2        PFACTs (0=left, 1=Crout, 2=Right)
2            # of recursive stopping criterium
2 4          NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
3            # of recursive panel fact.
......
```

HPL

Results example

```
T/V    : Wall time / encoded variant.
N      : The order of the coefficient matrix A.
NB     : The partitioning blocking factor.
P      : The number of process rows.
Q      : The number of process columns.
Time   : Time in seconds to solve the linear system.
Gflops : Rate of execution for solving the linear system.

The following parameter values will be used:

N      :      29       30       34       35
NB     :       1        2        3        4
PMAP   : Row-major process mapping
P      :       2        1        4
Q      :       2        4        1
PFACT  :    Left    Crout    Right
NBMIN  :       2        4
NDIV   :       2
RFACT  :    Left    Crout    Right
BCAST  :   1ring
DEPTH  :       0
```

HPCG(High Performance Conjugate Gradients) [4]

- complement to HPL
- target to a widely used patterns between computational and data access
- get the code, configure, and make, executable with slurm
- generate two files
  - hpcg-timestamp.txt
  - HPCG-Benchmark-timestamp.txt

[4] `https://hpcg-benchmark.org/`

Stream[5]

Implement

- self descripted, but free to use and modification
- memory transfer rates for computational kernels
- get the code, configure, and make, executable with slurm

Results segmentation

```
......
-------------------------------------------------------------
Function    Best Rate MB/s  Avg time     Min time     Max time
Copy:           19109.5     0.011772     0.008373     0.014118
Scale:          12326.4     0.013455     0.012980     0.014593
Add:            16486.8     0.015539     0.014557     0.016992
Triad:          16389.1     0.015810     0.014644     0.019724
-------------------------------------------------------------
Solution Validates: avg error less than 1.000000e-13 on all three arrays
-------------------------------------------------------------
......
```

[5] https://github.com/jeffhammond/STREAM

Purpose

- Implementing to SCC cluster
- Playing around and turning
- Documenting for integration in GWDG

bude[6]

- Apache-2.0
- Biophycics, folding NDM-1 (New Delhi metallo-beta-lactamase 1) protein for energy evaluation
- C, C++...
- small, middle, large model
- dependence
  - OpenMP for CPUs
  - OpenMP target for GPUs
  - CUDA for GPUs
  - OpenCL for GPUs
  - OpenACC for GPUs
  - SYCL for CPUs and GPUs
  - Kokkos for CPUs and GPUs

[6] `https://github.com/UoB-HPC/miniBUDE`

Candidates

cloverleaf

- High energy physics, Fortan, GPL-3.0
- solves the compressible Euler equations on a Cartesian grid, using an explicit, second-order accurate method

CP2K

- GPL-2.0 licence, Fortan
- quantum chemistry and solid state physics software package that can perform atomistic simulations of solid state, liquid, molecular, periodic, material, crystal, and biological systems.

Candidates

OpenSBLI

- License: GPL-3.0, python
- Given Equation, generating code for finite difference methods as numerical modelling, auch as Computational Fluid Dynamics.

SNAP system

- License: BSD license, C++
- Network performance test with data on nodes and/or edges in a graph network, can be easy scaled.

Question and some comments?