1 Introduction

All benchmarks, including job scripts, can be found at `https://gitlab.tudelft.nl/dhpc/benchmarks/`. Filesystem performance has not been benchmarked in this scope.

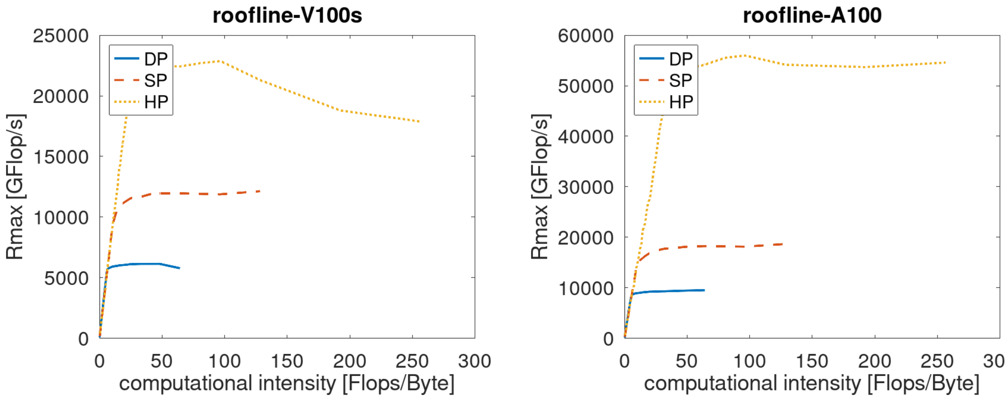
2 Dedicated benchmarks

2.1 Floating-point operations

Hardware	DP	SP	HP
Cascade Lake	3.9	7.7	15[1]
Sapphire Rapids	5.7	11.3	45[1]
V100s	5.8	12	18
A100	9.5	19	55

Table 1: Peak flop rate in TFlop/s (double/single/half precision) measured by liwkid-bench (CPU) and mixbench (GPU), respectively. CPU half precision has not been benchmarked.

[1] Estimated assuming AVX512 (phase 1) and AMX (phase 2)

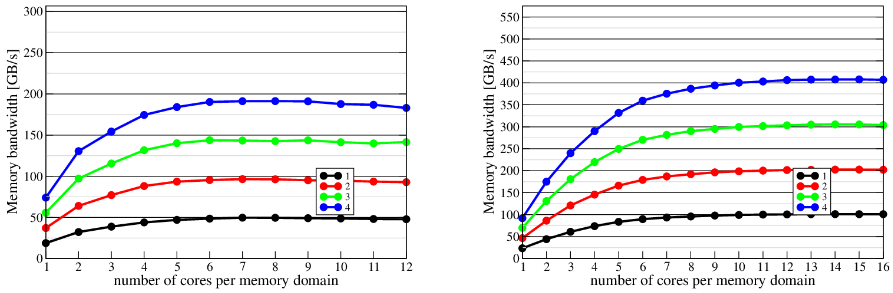


(a) Phase 1: V100s GPU

(b) Phase 2: A100 GPU

Figure 1: Roofline performance in double (DP), single (SP) and half precision (HP), measured using mixbench.

2.2 Memory bandwidth



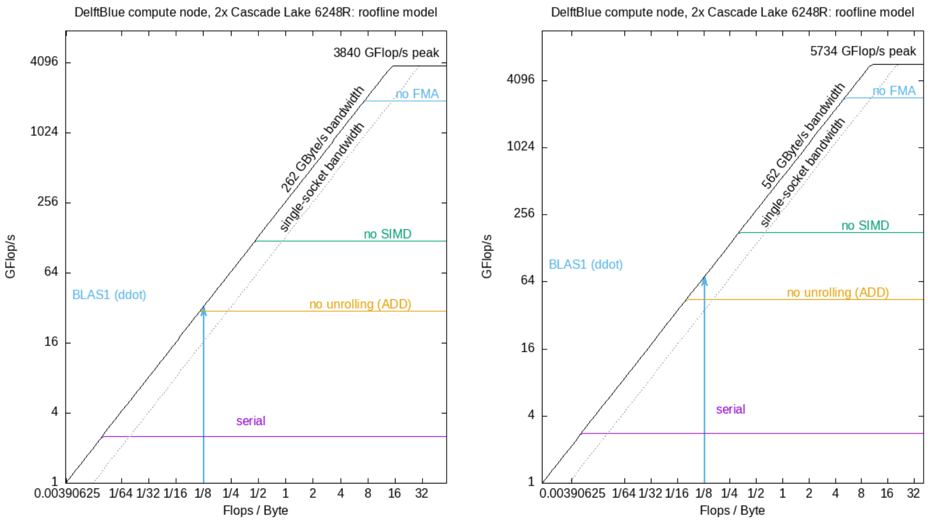
(a) Phase 1: 48-core "CascadeLake"

(b) Phase 2: 64-core "Sapphire Rapids"

Figure 2: Memory-bandwidth scaling across four NUMA domains (vector update $z = a \cdot x + y$).

For the GPU's, we measure approximately **910 GB/s (V100s)** and **1640 GB/s (A100)**, respectively from GPU RAM (global memory); Results obtained using mixbench.

2.3 Roofline plots for the CPU's



(a) Phase 1: 48-core "CascadeLake"

(b) Phase 2: 64-core "Sapphire Rapids"

Figure 3: Memory-bandwidth scaling across four NUMA domains (vector update $z = a \cdot x + y$).

2.4 MPI (InfiniBand) benchmark

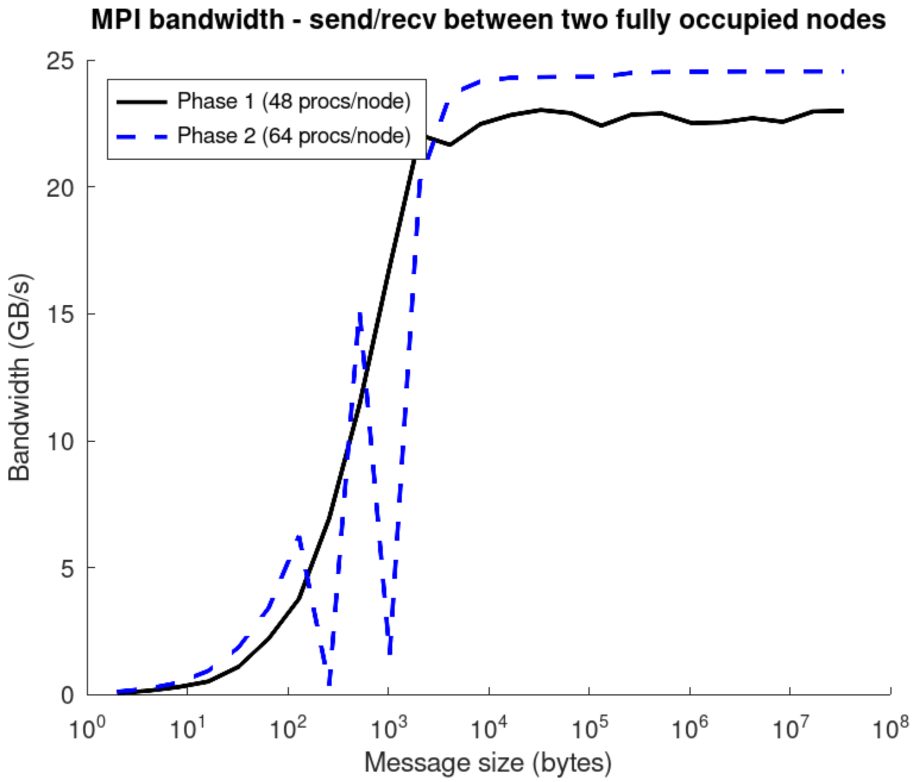


Figure 4: Two nodes, one MPI process per core, sending and receiving pairwise between the nodes.

3 Application benchmarks

The original benchmarks are described in Appendix **??**. The numbers in parenthesis for benchmarks 5 and 7. indicate the runtime on a quarter of a node rather than a fixed number of cores. This is a more meaningful metric of progress because the single-core performance is not increasing in recent hardware generations, and memory bandwidth can not be saturated with a small number of cores.

4 Application benchmarks

Benchmark	Phase 2 11/2023	Phase 1 10/2023	12/2021[1]	TU-Delft Reference time[2]
1_TurbulentPipeFlow	69.3	105	78.27s	143s
2_TurbulentBox	37.2	53.7	51.1s	79.6s
3_Radiation	7.45	14.4	12.94s	53.7s
4_LAMMPS	45.0	59.0	72s	152.4s
5_SquarePlate (2x 1/4 node)	297 (85.7)	466.2 (344)	412s	617s
6_ComputerVision	118	198	180s	303s
7_PacBio (1/4 node)	58.4 (29.9)	65.7 (40.8)	55.7s	105.6s

[1] measured by Fujitsu on the empty DelftBlue

[2] measured by users on various systems, e.g., Cartesius.

A Description of the TU Delft benchmarks

Benchmarks

for speedup assessment

The speedup of the HPC cluster to be provided will be compared against the speed of existing clusters on the basis of the following benchmarks provided by the TU Delft.

This archive contains 7 benchmarks from 3 different fields of research:

- **Computational Fluid Dynamics:**
  - **1_TurbulentPipeFlow**
    Incompressible Navier-Stokes solver (direct numerical simulation, DNS)
    256 cores
  - **2_TurbulentBox**
    Incompressible Navier-Stokes solver (direct numerical simulation, DNS)
    320 cores
  - **3_Radiation**
    GPGPU photon Monte Carlo radiative transfer solver
    128 cores & 16 GPUs

- **Molecular Dynamics:**
  - **4_LAMMPS**
    Molecular dynamics simulation
    4 cores & 2 GPUs
  - **5_SquarePlate**
    Linear elastic solver
    10 cores

- **Machine Learning:**
  - **6_ComputerVision**
    Image recognition using ImageNet database
    8 cores & 2 GPUs
  - **7_PacBio**
    Genome reading, find overlaps
    8 cores

Each benchmark directory contains a README document with instructions on how to install and run the benchmark, and how to determine the result.

1_TurbulentPipeFlow

Benchmark description

Brief description of the scientific application:

Navier-Stokes solver for turbulent pipe flows

What will be tested (the main bottlenecks):

□ *Single core performance (CPU)*
□ *GPU performance*
□ *Memory bandwidth and latency*
☒ *Shared memory parallelization (intra-node)*
☒ *Distributed memory parallelization (inter-node)*
□ *Filesystem bandwidth and latency*
□ *Other:*

2_TurbulentBox

Benchmark description

Brief description of scientific application:

Direct Numerical Simulation of a turbulent incompressible fluid

What will be tested (the main bottlenecks):

- □ *Single core performance (CPU)*
- □ *GPU performance*
- □ *Memory bandwidth and latency*
- □ *Shared memory parallelization (intra-node)*
- ☒ *Distributed memory parallelization (inter-node)*
- ☒ *Filesystem bandwidth and latency*
- □ *Other:*

3_Radiation

Test case description

Brief description of scientific application:

Monte Carlo radiative transfer solver coupled with Navier-Stokes solver (direct numerical simulation).

What will be tested (the main bottlenecks):

☐ *Single core performance (CPU)*
☒ *GPU performance*
☒ *Memory bandwidth and latency*
☒ *Shared memory parallelization (intra-node)*
☒ *Distributed memory parallelization (inter-node)*
☐ *Filesystem bandwidth and latency*
☐ *Other:*

4_LAMMPS

Test case description

Brief description of scientific application:

Running classical molecular dynamics (MD) simulations with the mainstream MD code lammps. The systems are large (~15 or 60 million atoms) and hence they are very suitable for exploiting gpu’s.

What will be tested (the main bottlenecks):

☐ *Single core performance (CPU)*
☒ *GPU performance*
☒ *Memory bandwidth and latency*
☐ *Shared memory parallelization (intra-node)*
☐ *Distributed memory parallelization (inter-node)*
☐ *Filesystem bandwidth and latency*
☐ *Other:*

5_SquarePlate

Test case description

Brief description of scientific application:

Mechanical analysis of a doubly-notched square plate subjected to a distributed vertical load. Material is linear-elastic and a parallel iterative solver equipped with an automatic domain decomposition scheme is used.

What will be tested (the main bottlenecks):

□ *Single core performance (CPU)*
□ *GPU performance*
□ *Memory bandwidth and latency*
☒ *Shared memory parallelization (intra-node)*
☒ *Distributed memory parallelization (inter-node)*
□ *Filesystem bandwidth and latency*
□ *Other:*

6_ComputerVision

Test case description

Brief description of scientific application:

One of the most common Computer Vision tasks is to perform object recognition on the ImageNet dataset. A large portion of active research concerns optimizations to popular state-of-the-art Deep Learning models such as the ResNet-50. Our test case is to train this model as fast as possible, i.e. with a time per mini-batch as low as possible.

What will be tested (the main bottlenecks):

☒ *Single core performance (CPU)*
☒ *GPU performance*
□ *Memory bandwidth and latency*
☒ *Shared memory parallelization (intra-node)*
□ *Distributed memory parallelization (inter-node)*
☒ *Filesystem bandwidth and latency*
□ *Other:*

7_PacBio

Test case description

Brief description of scientific application:

This is a de novo assembly test case based on simulated reads. First, a simulator is generating a random genome and then generating PacBio reads from it. After that a very fast tool is used to find pairwise overlaps between the reads. At last, a very fast tool is used to reconstruct the random genome based on pairwise overlaps.

What will be tested (the main bottlenecks):

☒ *Single core performance (CPU)*
□ *GPU performance*
□ *Memory bandwidth and latency*
☒ *Shared memory parallelization (intra-node)*
□ *Distributed memory parallelization (inter-node)*
☒ *Filesystem bandwidth and latency*
□ *Other:*