

# hpcscan version 1.1

## Performance benchmarks on Shaheen II (KAUST)

December 2020

- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements

- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements

# Introduction

All tests in single precision

Best performance is reported over 10 tries for each cases

Grids are 3D

- 1 Introduction
- 2 **Shaheen II (KAUST)**
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements

## Machine Shaheen II / Cray XC40

- Computing nodes Intel Haswell 2.3 Ghz dual socket (16 cores / socket)
- RAM 128 GB with Peak memory BW 136.5 GB/s
- Peak performance Single Prec. 2.36 TFLOP/s / Double Prec. 1.18 TFLOP/s
- Interconnect Cray Aries with Dragonfly topology
  - 60 GB/s optical links between groups
  - 8.5 GB/s copper links between chassis
  - 3.5 GB/s backplane within a chassis
  - 5 GB/s PCIe from node to Aries router



- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory**
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements

# Test Case Memory - Description

## Benchmark objective

Measure GByte/s and GPoint/s for

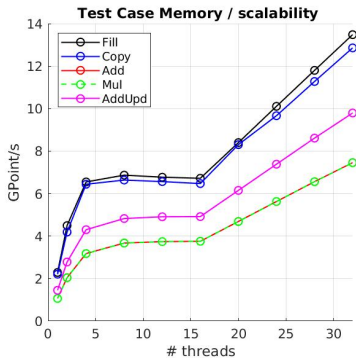
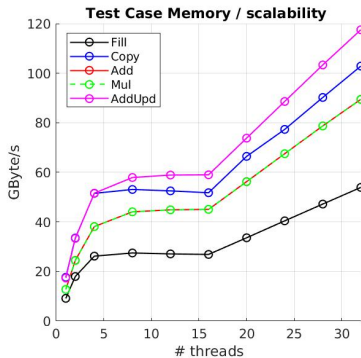
- Fill grid ( $W = \text{coef}$ )
- Copy grid ( $W = U$ )
- Add grids ( $W = U + V$ )
- Multiply grids ( $W = U * V$ )
- Add and update grids ( $W = W + U$ )

## Benchmark configuration

- Scalability on 1 node with 1 to 32 threads
- Baseline kernel
- Grid size 4 GB (1000 × 1000 × 1000 points)
- Reproduce results with `./script/testCase_Memory/hpcscanMemory.sh`
- Elapsed time about 4 minutes



# Test Case Memory - Results <sup>1</sup>



<sup>1</sup>Updated Dec 22, 2020

- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid**
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements

# Test Case Grid - Description

## Benchmark objective

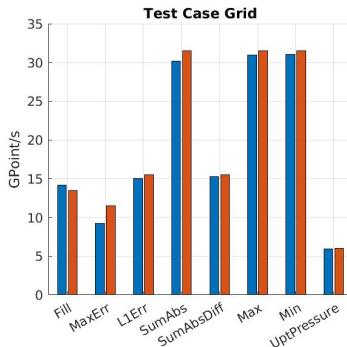
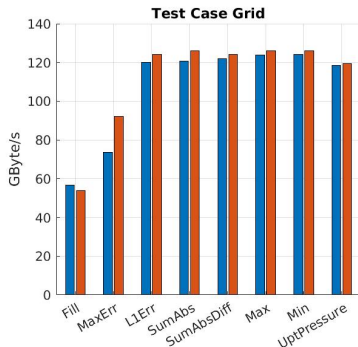
Measure GByte/s and GPoint/s for

- Fill grid ( $U = \text{coef}$ )
- Max. diff ( $U-V$ )
- L1 norm between  $U$  and  $V$
- Sum Abs( $U$ ) & Sum Abs( $U-V$ )
- Get max. & Get min. grid  $U$
- Update pressure (used in propagator)
- Boundary condition (free surface at all edges)

## Benchmark configuration

- 1 node with 32 threads
- Baseline kernel
- 2 grid sizes
  - Small size 500 MB ( $500 \times 500 \times 500$  points)
  - Medium size 4 GB ( $1000 \times 1000 \times 1000$  points)
- Reproduce results with `./script/testCase_Grid/hpcscanGrid.sh`
- Elapsed time less than 1 minute

# Test Case Grid - Results <sup>2</sup>



Blue small grid / Red medium grid

ApplyBoundaryCondition performs at 713/846 GBytes (89/105 Gpoint/s)

<sup>2</sup>Updated Dec 23, 2020

- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm**
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements

## Benchmark objective

Measure GByte/s and GPoint/s for

MPI point to point communication

- Send with MPI\_Send from proc X to proc 0 (Half-duplex BW)
- Send and receive with MPI\_Sendrecv between proc X and proc 0 (Full-duplex BW)

MPI collective communication

- Exchange of halos used in FD kernel with MPI\_Sendrecv
- Domain decomposition with  $N_1 \times N_2 \times N_3$  subdomains

## Benchmark configuration

- 8 nodes with 32 threads
- Baseline kernel
- Grid size 4 GB (1000 × 1000 × 1000 points)
- Subdomain decomposition: 1x4x2 / 1x2x4 & 2x2x2
- Reproduce results with `./script/testCase_Comm/hpcscanComm.sh`
- Elapsed time less than 1 minute

Table: Bandwidth GB/s

MPI#1	MPI#2	Send	Sendrecv	Halo exch.	Comm. size	Subdomains
0	1	8.5	15.3	-	47 MB	-
0	2	8.3	15.3	-	47 MB	-
0	3	8.6	15.3	-	47 MB	-
0	4	8.5	15.3	-	47 MB	-
0	5	8.2	15.3	-	47 MB	-
0	6	8.5	15.3	-	47 MB	-
0	7	8.6	15.3	-	47 MB	-
All	All	-	-	5.0	128 MB	1 4 2
All	All	-	-	5.1	128 MB	1 2 4
All	All	-	-	2.0	96 MB	2 2 2

<sup>3</sup> Updated Sep 19, 2020

- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2**
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements



# Test Case FD\_D2 - Description

## Benchmark objective

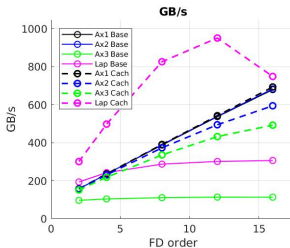
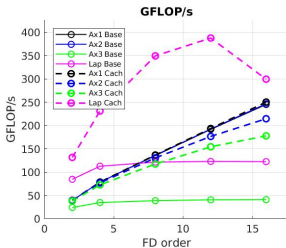
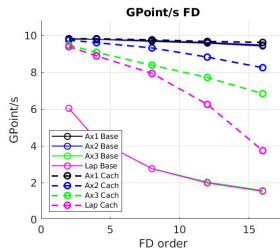
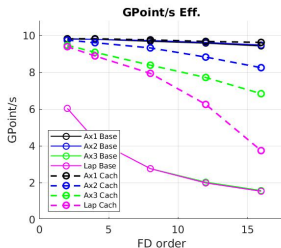
Measure GByte/s, GPoint/s & Gflop/s for

- Computation of second order derivatives with finite-difference stencil
- Directional derivatives
  - Axis 1,  $W = \partial_{x1}^2(U)$
  - Axis 2,  $W = \partial_{x2}^2(U)$
  - Axis 3,  $W = \partial_{x3}^2(U)$
- Laplacian  $W = \Delta(U)$
- Stencil orders 2, 4, 8, 12 & 16

## Benchmark configuration

- 1 node with 32 threads
- 2 test modes
  - Baseline
  - CacheBlk
- Grid size 4 GB (1000 × 1000 × 1000 points)
- Reproduce results with `./script/testCase_FD_D2/hpcscanFD_D2.sh`
- Elapsed time about 2 minutes

# Test Case FD\_D2 - Results <sup>4</sup>



- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa**
- 8 Summary
- 9 Acknowledgements

# Test Case FD\_D2 - Description

## Benchmark objective

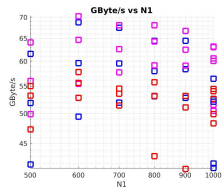
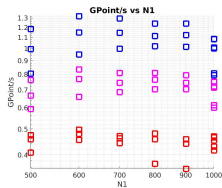
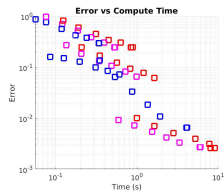
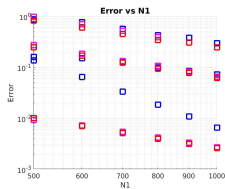
Measure GByte/s, GPoint/s & Gflop/s for

- Acoustic wave propagator
- Stencil orders 2, 4, 8, 12 & 16
- Time step 1, 0.5 & 0.1 of stability time step

## Benchmark configuration

- 1 node with 32 threads
- Test mode CachBlk
- 2 propagator implementations
  - Ac2Standard
  - Ac2SplitComp
- Grid size from  $500^3$  (500 MB) to  $1000^3$  (4 GB)
- Reproduce results with  
`./script/testCase.PropaParamAnalysis/hpcscanPropaParamAnalysis.sh`
- Elapsed time about XX minutes

# Test Case Propa - Results <sup>5</sup>



<sup>5</sup> Updated Nov 5, 2020

# Test Case FD\_D2 - Description

## Benchmark objective

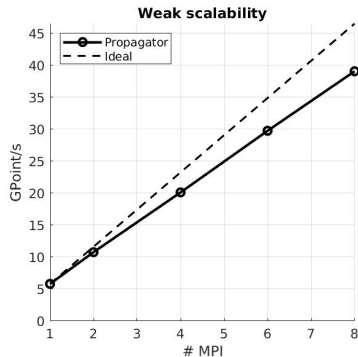
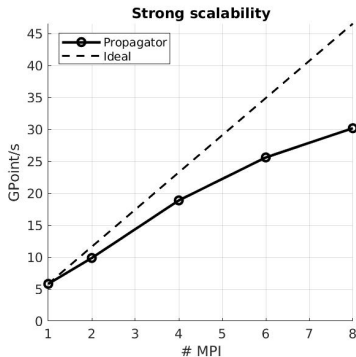
Measure GByte/s, GPoint/s & Gflop/s for

- Acoustic wave propagator
- Stencil order 8
- Strong and weak scalability

## Benchmark configuration

- From 1 node to 8 nodes with 32 threads/node
- Test mode CachBlk
- Propagator implementation Ac2Standard
- Strong scalability: Grid size  $1000^3$  (4 GB)
- Weak scalability: Grid size from  $1000 \times 1000 \times 1000$  (4 GB) to  $1000 \times 4000 \times 2000$  (32 GB)
- Reproduce results with  
`./script/testCase_strongWeakScalability/hpcscanPropaStrongWeakScalability.sh`
- Elapsed time about 7 minutes

# Test Case Propa - Results <sup>6</sup>



<sup>6</sup> Updated Dec 26, 2020

- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary**
- 9 Acknowledgements



## Test Case Memory

- Measured memory BW between 91 to 122 GB/s (67-90 % of peak BW)
- Low BW 59 GB/s for Fill (43 % of peak BW)
- Multiply (= imaging condition) performs at 7.6 Gpoint/s

## Test Case Grid

- L1 Err., Get Min & Max: 125 GB/s close to peak BW (92 % Peak Mem. BW)
- Low perf for Fill: 54-58 GB/s (40-43 % Peak Mem. BW)
- Max Err. 72-91 GB/s (53-67 % Peak Mem. BW)
- Pressure update 6 GPoint/s (120 GB/s, 88 % Peak Mem. BW)

## Test Case Comm

- TO DO

## Test Case FD\_D2

- Large benefit of cache blocking
- Significant effect of grid dimension and index (very bad performance for n3 without cache blocking)
- Min BW 50 GFLOP/s ( $\partial_{x3}^2$  O2) = 2 % peak BW [apparent Mem. BW 150 GB/s]
- Max BW 370 GFLOP/s ( $\Delta$  O8) = 16 % peak BW [apparent Mem. BW 900 GB/s]
- Apparent Mem. BW 150-900 GB/s (110-660 % Peak Mem. BW) = shows data in-cache effect
- Typical stencils of interest for geophysical applications
  - $\Delta$  O4 BW = 8-10 GPoint/s
  - $\Delta$  O8 BW = 7-9 GPoint/s
  - $\Delta$  O12 BW = 3-5 GPoint/s
- Parallel efficiency with 8 nodes 55 to 86 % (depends on workload on Shaheen)

## Test Case Propa

- TO DO

# Content

- 1 Introduction
- 2 Shaheen II (KAUST)
- 3 Test Case Memory
- 4 Test Case Grid
- 5 Test Case Comm
- 6 Test Case FD\_D2
- 7 Test Case Propa
- 8 Summary
- 9 Acknowledgements**

# Acknowledgements

- KAUST ECRC and KSL for access and support on Shaheen II