Hackathon to Accelerate Scientific Applications Using GPUs

Aramco-KAUST Team 3

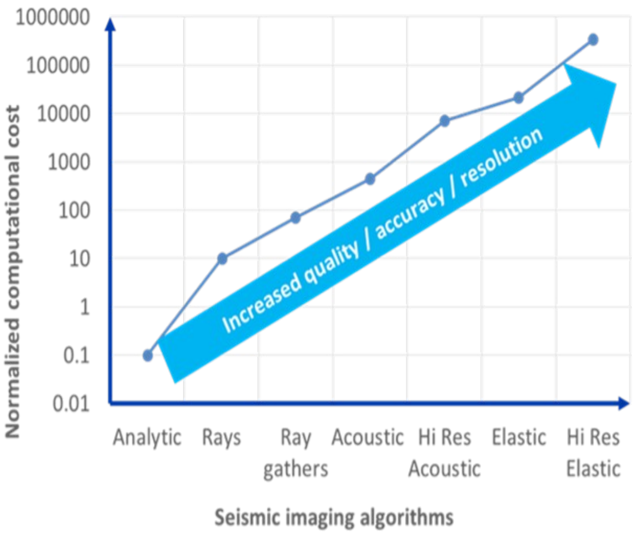
November 30, 2020 – December 2, 2020

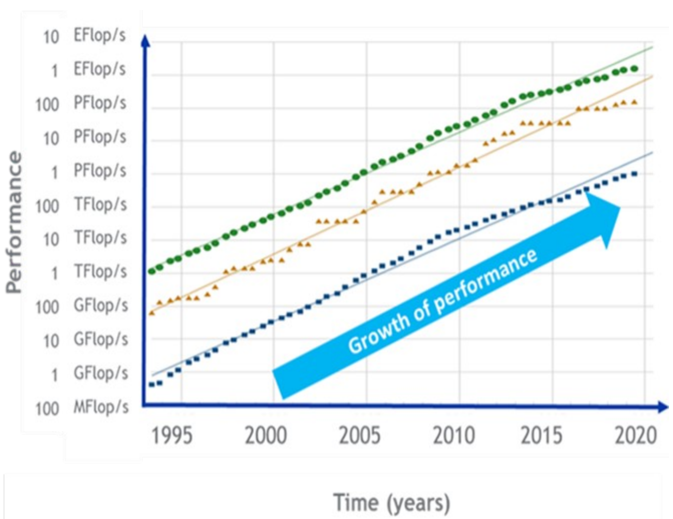
Vincent Etienne

Suha Kayum

Marcin Rogowski

HPC in Oil & Gas





HPL (dense matrix) ranking not representative of many scientific applications

	Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
CPU	1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442,010.0	[illegible]	[illegible]
GPU	2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
GPU	3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
CPU	4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
	5	**Selene** - NVIDIA DGX A100, AMD EPYC 7742 64C 2.25GHz, NVIDIA A100, Mellanox HDR Infiniband, Nvidia NVIDIA Corporation United States	555,520	63,460.0	79,215.0	2,646
	6	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482
	7	**JUWELS Booster Module** - Bull Sequana XH2000 , AMD EPYC 7402 24C 2.8GHz, NVIDIA A100, Mellanox HDR InfiniBand/ParTec ParaStation ClusterSuite, Atos Forschungszentrum Juelich (FZJ) Germany	449,280	44,120.0	70,980.0	1,764
	8	**HPC5** - PowerEdge C4140, Xeon Gold 6252 24C 2.1GHz, NVIDIA Tesla V100, Mellanox HDR Infiniband, Dell EMC Eni S.p.A. Italy	669,760	35,450.0	51,720.8	2,252
	9	**Frontera** - Dell C6420, Xeon Platinum 8280 28C 2.7GHz, Mellanox InfiniBand HDR, Dell EMC Texas Advanced Computing Center/Univ. of Texas United States	448,448	23,516.4	38,745.9	
	10	**Dammam-7** - Cray CS-Storm, Xeon Gold 6248 20C 2.5GHz, NVIDIA Tesla V100 SXM2, InfiniBand HDR 100, HPE Saudi Aramco Saudi Arabia	672,520	22,400.0	55,423.6	

Top ranked industrial systems are from O&G

TOP 500 The List.

Why does it matter?

Wide range of architectures
- CPU / GPU / FPGA / Other accelerators

Number of programming models

Versatile Benchmarking
- Take full benefits of hardware and programming models
- Avoid bias in existing codes

Solution
- A flexible benchmarking tool
- Assess characteristics of HPC platforms
- Representative of most scientific computing kernels
- Open to the scientific community and vendors
- Adapt, reuse, analyze, optimize

hpcscan - an HPC benchmarking tool

hpcscan is a C++ code for benchmarking HPC kernels with a focus on solving PDEs with the Finite Difference Method

- Simple code structure based on individual test cases
- Easy to add new test cases
- Hybrid MPI/OpenMP parallelism
- All configuration parameters on command line
- Single and double precision computation
- Compilation with standard Makefile
- No external libraries
- Built-in performance measurement and validation

Features

- General operations on grids
- Memory operations
- MPI communication
- FD computation
- Basic wave propagator

GitHub https://github.com/vetienne74/hpcscan

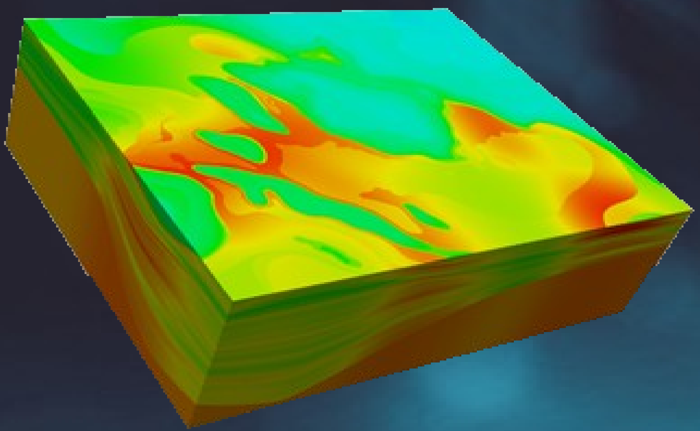
Main kernels in the testcase - Grid

- Fill grid (W = coeff.)
- Max. err. grid W
- L1 err. grid W
- Get min. grid W
- Get max. grid W
- Update pressure (used in propagator)

Representative kernels chosen for Scientific Computing

Main kernels in the testcase - Propagator

- Seismic wave propagator
- 2^{nd} order acoustic wave equation
- Time-domain Finite-difference
- Various FD Order in space
- 2^{nd} FD order in time
- Various grid size and time steps
- Total 18 configurations
- Comparison against analytical solution (Eigen mode)



Different Implementations

OpenACC

CUDA

OpenACC

- Ported FILL kernel to OpenACC
- OpenMP pragmas already present in initial code
  - Time spent in compiling and changing FILL function to openACC (NVIDIA & KAUST help)
  - 2X slower when compared to CUDA
  - Continued efforts on CUDA

```
        grid_3d = new Myfloat[npoint] ;
#pragma acc enter data copyin(this[0:1]) create(grid_3d[0:(n1*n2*n3)])
```

```
void Grid_GPU2::fill(Point_type pointType, Myfloat val)
{
        printDebug(FULL_DEBUG, "In Grid_GPU2::fill") ;

        Myint64 i1Start, i1End, i2Start, i2End, i3Start, i3End ;
        getGridIndex(pointType, &i1Start, &i1End, &i2Start, &i2End, &i3Start, &i3End) ;

#pragma acc parallel loop collapse(2) present(this) copyout(grid_3d[:(n1*n2*n3)])
        for (Myint64 i3 = i3Start; i3<= i3End; i3++)
        {
                for (Myint64 i2 = i2Start; i2<= i2End; i2++)
                {

                        for (Myint64 i1 = i1Start; i1<= i1End; i1++)
                        {
                                Myint64 ii = i1 + i2*n1 + i3*n2*n1 ;
                                grid_3d[ii] = val ;
                        }
```

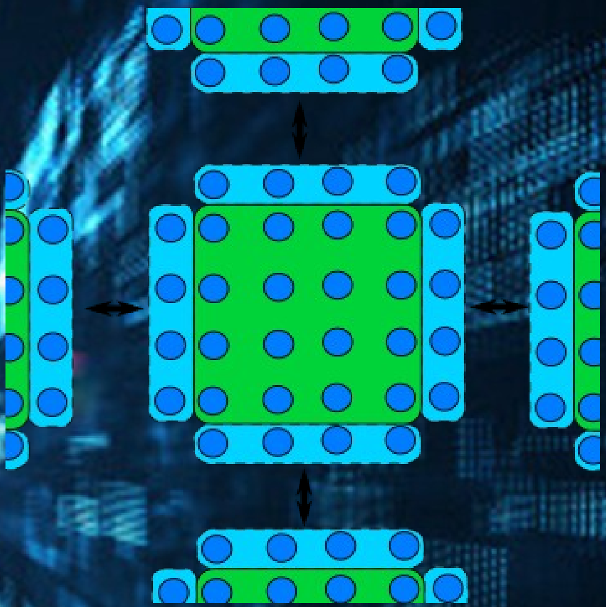
Different Implementations

OpenACC

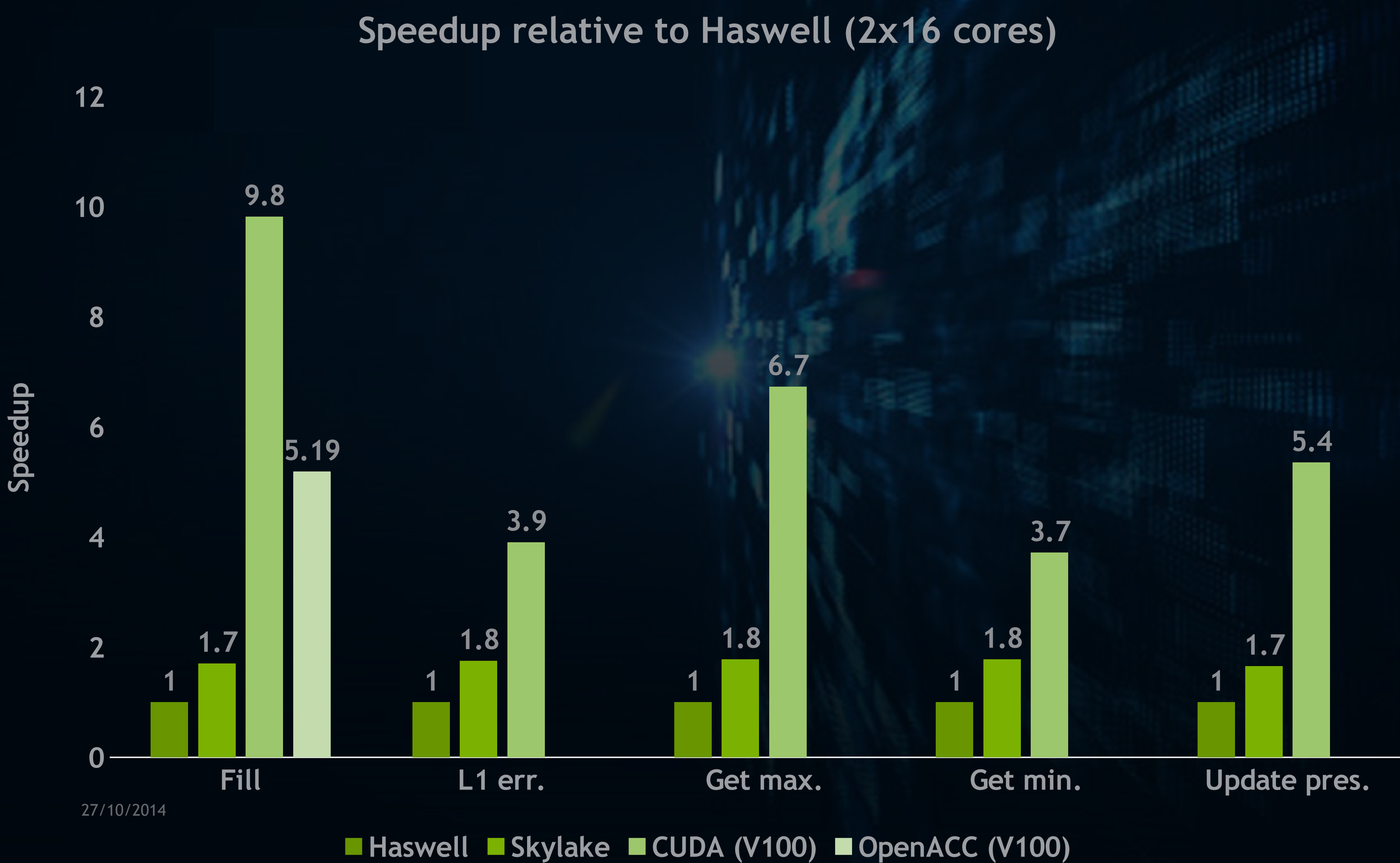
CUDA

CUDA

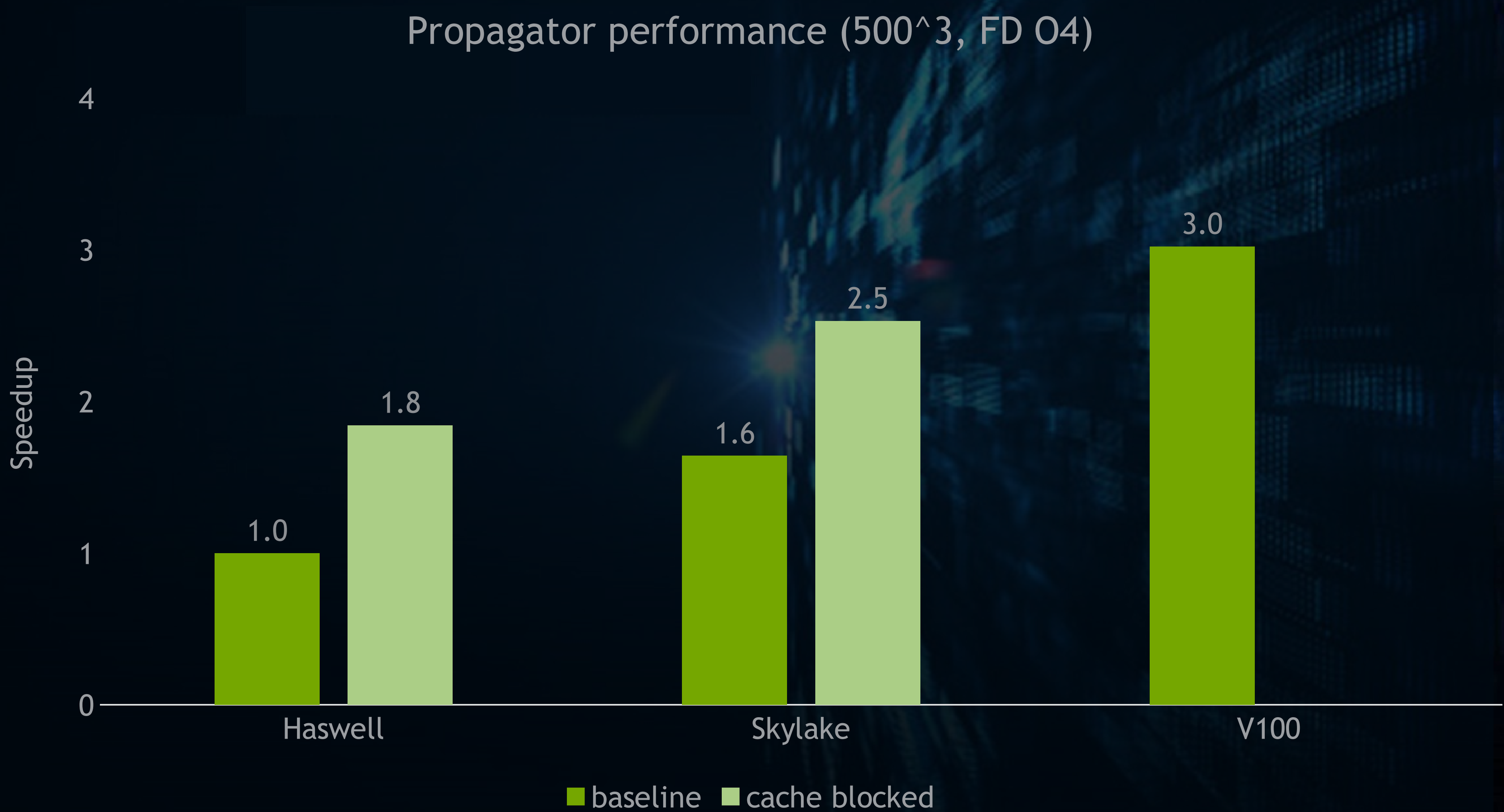
- Ported all major kernels to CUDA
  - Grid, Propagator
- Cell-type addressing the biggest challenge
  - “halo 1 of dim 3” (I,J,K)
  - threadIdx => i,j,k
- Grid operations perform well
- Stencil operations near memory bandwidth but cache utilization poor
  - 30-35% L1/L2 cache utilization according to nSight Compute



Results – CUDA – Grid operations



Results – CUDA – Stencil operations



* Best case scenario. Cache-blocked Skylake won in many other scenarios!

Future Work

Complete porting & tuning all kernels to CUDA and OpenACC
Profile & optimize CUDA kernels
Extend to multi-GPU, multi-node
Once fully validated, make it available to the community

Promote hpcscan and welcome contributions

With time, embedded kernels will benefit from and to the contributors

Acknowledgements

KAUST: KSL/ECRC staff, especially Bilel Hadri & Saber Feki
NVIDIA: Asma Farjallah & Rached Abdelkhalek

Hackathon to Accelerate Scientific Applications Using GPUs

Aramco-KAUST Team 3

November 30, 2020 – December 2, 2020

Vincent Etienne

Suha Kayum

Marcin Rogowski