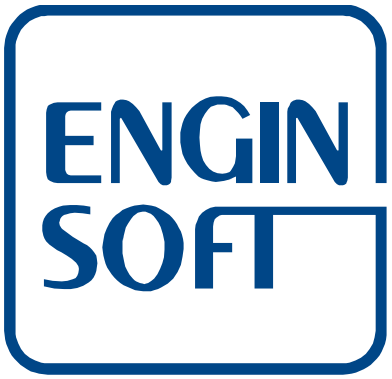


Key partner in Design Process Innovation

Scalability & performances of general purpose fluid dynamics solver on low power cluster: new perspectives on combined CUDA-ARM architecture

Gino Perna

Jun 24 th 2015 - Ecole Polytechnique, Palaiseau

Agenda

- Enginsoft
- ARM64 experience
- Experience in CFD
- CFD code running on ARM64+CUDA


Qui sommes nous
Intégrateur dans le domaine de la simulation
Des ingénieurs passionés
Groupe de 100 experts

Agenda

- Enginsoft
- ARM64 experience
- Experience in CFD
- CFD code running on ARM64+CUDA

EnginSoft en quelques mots

- Fondée en 1984 à partir de structures en place depuis 1973
- Leader européen en simulation numérique (+ de 100 ingénieurs, docteurs)
- Filiales/partenariats à travers toute l'Europe et les USA
- Consulting, distribution de logiciels scientifiques
- Développement de nouvelles technologies logicielles (ModeFRONTIER, CharLES…)
- Participation dans des projets industriels de recherche (France et Europe, H2020)
- Chiffre d'affaires 2012 16,4 Millions d'euros
- Effectif 125 personnes

EnginSoft - Activités principales

DISTRIBUTION DE LOGICIELS

Logiciels
Plus de 1500 licences de logiciels de simulation vendues en Europe

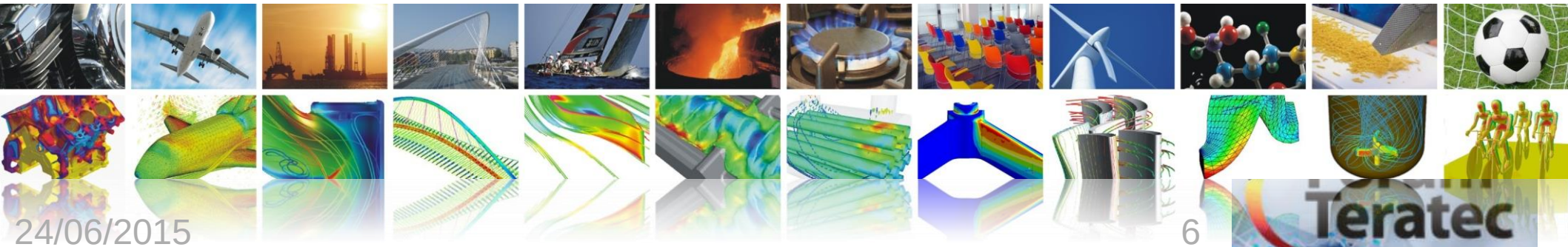
Formations
Plus de 130 formations par an. Consortium pour la formation supérieure technique et un portail dédié à la formation en ligne

Transfert de Compétences

CONSULTING

Etudes
Plus de 3000 études d'ingénierie réalisées depuis 1984, par plus de 100 ingénieurs experts.

Recherche & Développement
Actif dans plus de 30 projets de recherche avec un cofinancement public



Consulting

SIMULATION MECANIQUE

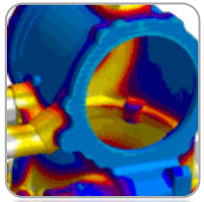
40 Ingénieurs

- Statique
- Dynamique
- Thermomécanique
- Vibratoire
- Sismique
- Acoustique
- Cinématique
- Tolérancement

SIMULATION FLUIDE

30 Ingénieurs

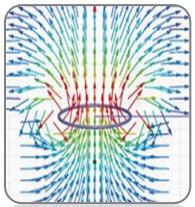
- Incompressible
- Compressible
- Système 1D
- Multi Phase
- Transfert
- Thermique
- Combustion
- Accrétion de glace
- Fluide Structure



SIMULATION DES PROCEDES

15 Ingénieurs

- Forgeage
- Fonderie
- Laminage
- Fraisage
- Tournage
- Perçage
- Traitement thermique
- Traitement de surface
- Injection plastique



ELECTRO-MAGNETISME

5 Ingénieurs

- Faible fréquence
- Haute fréquence
- Compatibilité Electromagnétique

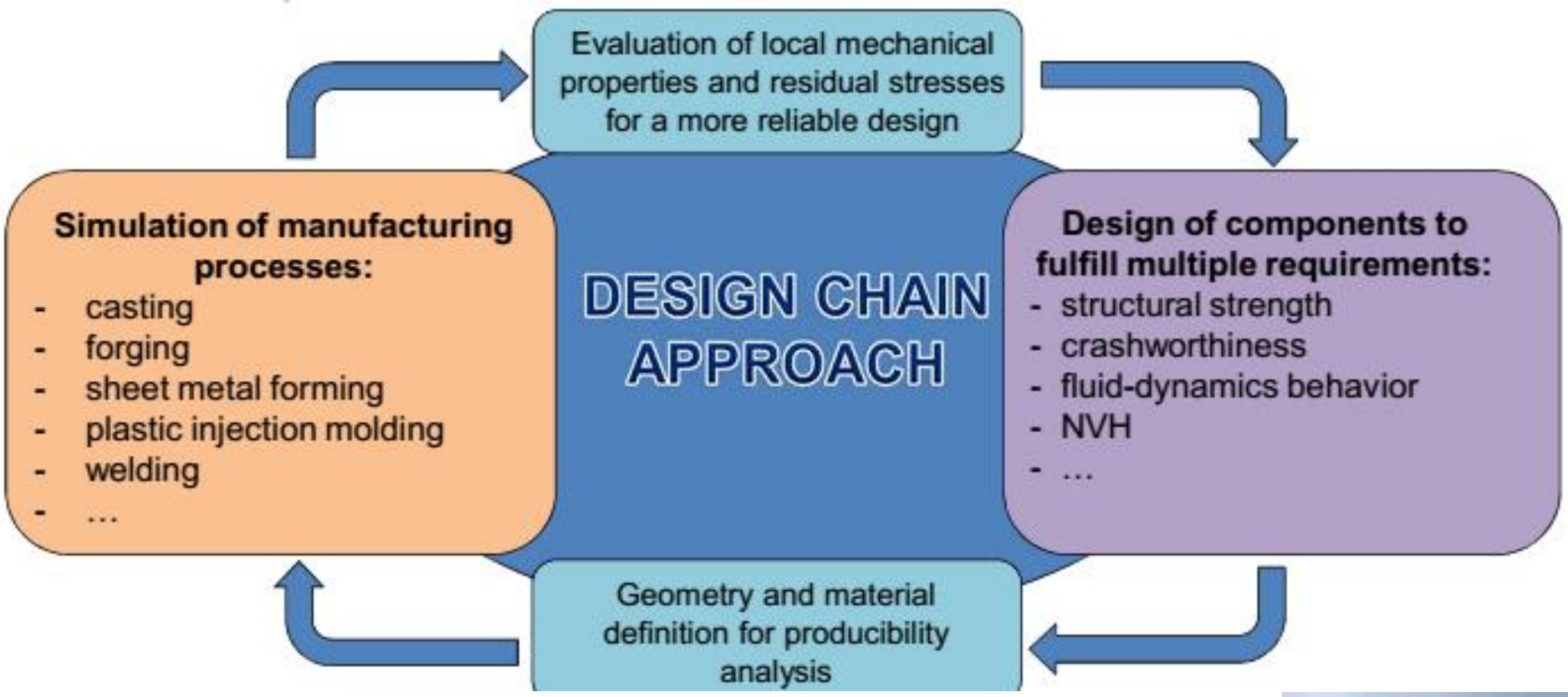
OPTIMISATION MULTI-OBJECTIF

10 Ingénieurs

- Intégration de processus
- Plan d'Expérience
- Analyse de sensibilité
- Méta Modèle
- Optimisation Multi Objectif
- Analyse de données

Approche par chaine de conception

Méthodologie innovatrice pour lier la fabrication, la conception et le développement



EnginSoft - Quelques références

- **Reconnu comme une entreprise d'ingénierie de haute qualité**

Our approach to ARM processors

ARM64-CUDA

- Paste experience with ARM32
- December/January 2015: tried to perform some benchmarks on ARM64 board. No way to get a working system till February.
- January 2015: First working system (could be switched on and stay on for days without crashes and have some performances) but lack of good I/O connections

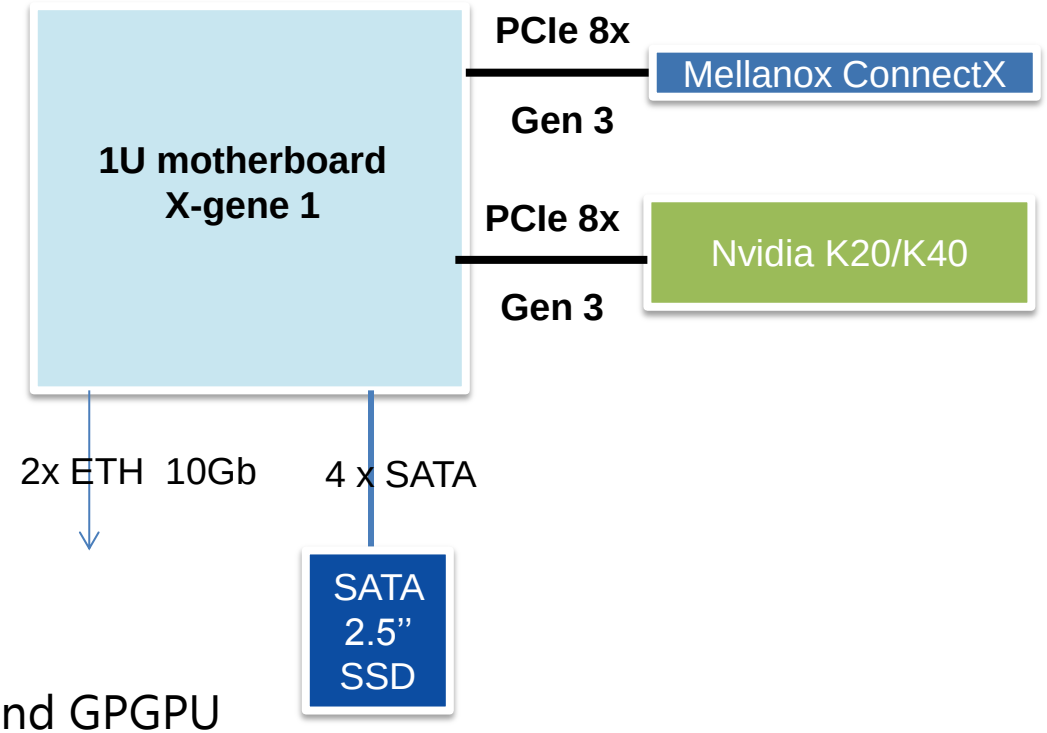
ARM64-CUDA

- May 2015: Thank to the hard work of Alessandro DeFilippo (E4 Computer engineering) and the ARM producer (APM mod X-gene1) :
  - network layer
  - (Alessandro DeFilippo) Infiniband RDMA working driver
  - NVIDIA K20 working with excellent performances
  - No support for pyCUDA, etc...
- We could start investigating at end of May.

- NVIDIA K40 still not working due to a problem in the initialization phase with the dimension of a buffer
- Hard … to get support for GPU and IB drivers

The ARKA RK003 Server (1)

FEATURES
Form Factor: 2U
CPU 1x APM X-Gene 8 cores
GPU up to 2x NVIDIA Kepler® K20, K40, K80
Memory Up to 128 GB RAM DDR3
Network 2x 10 GbE SFP+, 1x Infiniband FDR QSFP
Storage 4x SATA 3.0
Expansion slots 2x PCI-E 3.0 x8 (in x16)



ARM64 and GPGPU

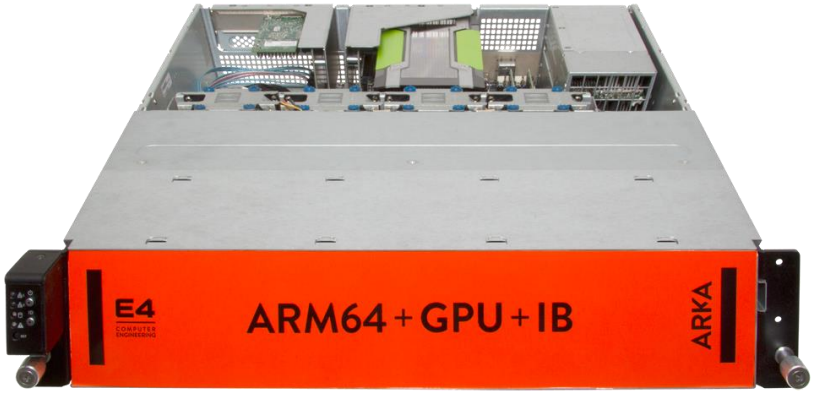
THE ENVIRONMENT

OS:
Ubuntu derivative for ARM 14.04 LTS

DEVELOPMENT TOOL:
NVIDA CUDA 6.5 (compilers, libraries, SDK)
MPI 2.0 Libraries
GNU compilers
Scalable Heterogeneous Computing (SHOC) Benchmark Suite

CLUSTER, MONITORING, MANAGEMENT TOOLS (optional):
Resource/Queue manager
Monitoring tools for cluster
Parallel shell for cluster-wide commands
Bare-metal restore

THE ENVIRONMENT



SOME PERFORMANCES

```
Device 0: 'Tesla K20m'
Device selection not specified: defaulting to device #0.
Using size class: 1

--- Starting Benchmarks ---
Running benchmark BusSpeedDownload
    result for bspeed_download:                3.1592 GB/sec
Running benchmark BusSpeedReadback
    result for bspeed_readback:                3.3575 GB/sec
Running benchmark MaxFlops
    result for maxspflops:                  3095.8700 GFLOPS
    result for maxdpflops:                  1165.9700 GFLOPS
Running benchmark DeviceMemory
    result for gmem_readbw:                  147.9020 GB/s
    result for gmem_writebw:                 139.9250 GB/s
```

SOME PERFORMANCES

```
Device 0: 'Tesla K20m'
Device selection not specified: defaulting to device #0.
Using size class: 1

--- Starting Benchmarks ---
Running benchmark BusSpeedDownload
    result for bspeed_download:                 3.1592 GB/sec
Running benchmark BusSpeedReadback
    result for bspeed_readback:                 3.3575 GB/sec
Running benchmark MaxFlops
    result for maxspflops:                   3095.8700 GFLOPS
    result for maxdpflops:                   1165.9700 GFLOPS
Running benchmark DeviceMemory
    result for gmem_readbw:                   147.9020 GB/s
    result for gmem_writebw:                  139.9250 GB/s
```

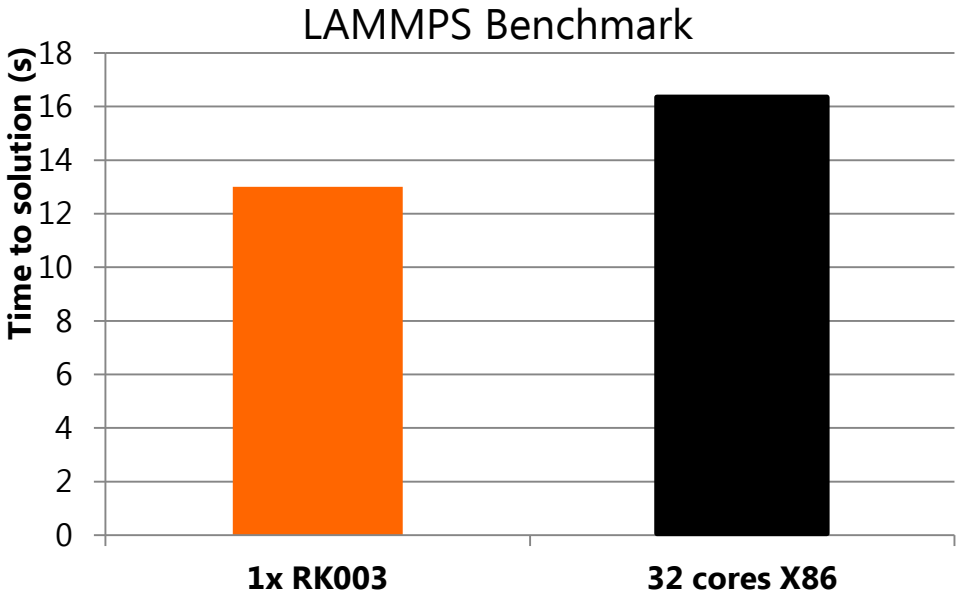
SOME PERFORMANCES

```
root@node3:~# mpirun --pernode -H node3,node4 --mca btl self,openib -np 2 /opt/osu/libexec/osu-micro-benchmarks/mpi/pt2pt/osu_latency
# OSU MPI Latency Test v4.4.1
# Size          Latency (us)
0                       1.82
1                       1.95
2                       1.95
4                       1.97
8                       1.99
16                      1.99
32                      2.00
64                      2.19
128                     3.47
256                     3.71
512                     4.00
1024                    4.52
2048                    5.45
4096                    6.39
8192                    8.51
16384                  10.47
32768                  13.68
65536                  20.36
131072                 33.68
262144                 60.37
524288                113.63
1048576               220.18
2097152               433.43
4194304               860.13
```

SOME PERFORMANCES

```
root@node3:~# mpirun --pernode -H node3,node4 --mca btl self,openib -np 2 /opt/osu/libexec/osu-micro-benchmarks/mpi/pt2pt/osu_mbw_mr
# OSU MPI Multiple Bandwidth / Message Rate Test v4.4.1
# [ pairs: 1 ] [ window size: 64 ]
# Size                  MB/s        Messages/s
1                       0.92         919677.46
2                       1.86         931841.07
4                       3.68         919677.46
8                       7.39         923919.10
16                     15.09         943368.32
32                     28.51         890865.05
64                     57.06         891634.41
128                   108.22         845466.00
256                   209.27         817478.62
512                   361.96         706948.61
1024                  629.07         614325.01
2048                 1088.73         531607.99
4096                 1659.77         405216.18
8192                 2387.15         291400.75
16384                3468.08         211674.85
32768                4278.98         130584.22
65536                4712.17          71902.03
131072               4827.55          36831.26
262144               4872.51          18587.14
524288               4896.03           9338.43
1048576              4883.04           4656.83
2097152              4913.11           2342.75
4194304              4917.44           1172.41
```

SOME PERFORMANCES



SETUP

COMPILER

-Intel compilers on E5-2650v2

-MKL

-GCC on ARM

-FFTW 3.2.2

HARDWARE

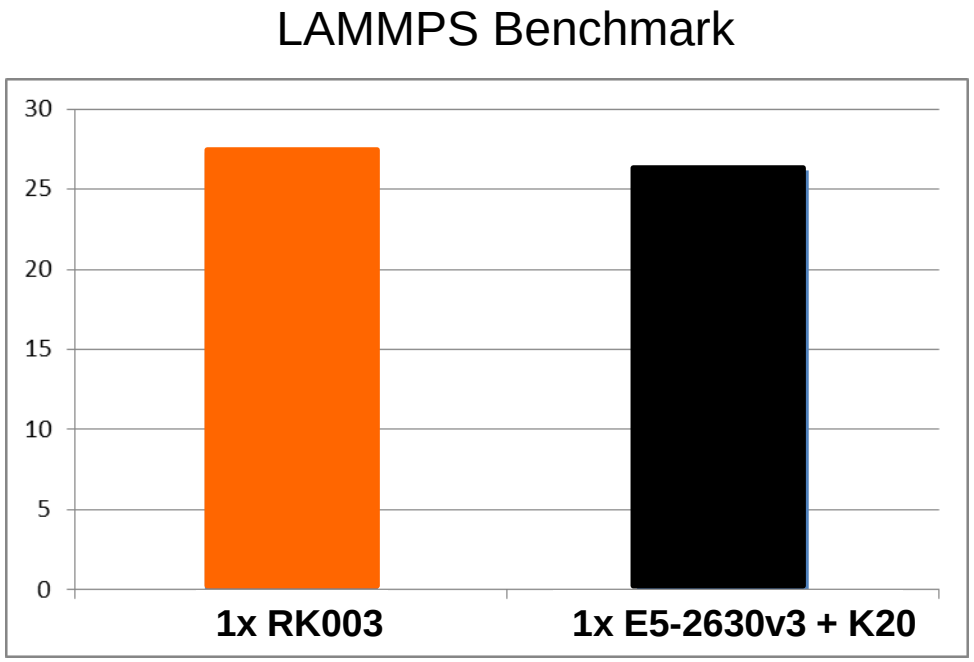
-Dual E5-2650v2, Infiniband QDR

-ARM X-gene1+ 1 GPU Nvidia K20m

INPUT FILES

-WdV 1M particle

SOME PERFORMANCES



SETUP

COMPILER GNU

Cuda 6.5

Cublas and cufft

HARDWARE

-Dual E5-2630v3 + 1 GPU Nvidia K20m

-ARM X-gene1+ 1 GPU Nvidia K20m

INPUT FILES

-WdV 1M particle

- 1000 steps

Parallel Performances

There where some numerical benchmarks we want to run:

Parallel Benchmarks (NPB3) (https://en.wikipedia.org/wiki/NAS_Parallel_Benchmarks)
Parboil benchmarks (http://impact.crhc.illinois.edu/Parboil/parboil.aspx)
HPCG: High Performance Conjugate Gradient Benchmark

CFD codes

Parboil benchmarks

histo

omp_base/2	omp_base/4	omp_base/8	cuda_base_default	cuda_default
		3303,5042	112,1839	22,6475

lbm

omp_base/2	omp_base/4	omp_base/8	cuda_base_default	cuda_default
883,3539	511,0997	381,7769	99,4660	
1,0000	1,7283	2,3138	8,8810	

mri-gridding

omp_base/2	omp_base/4	omp_base/8	cuda_base_default	cuda_default
70,8452	80,7177	98,6143	57,9537	59,6754
1,0000	0,8777	0,7184	1,2224	1,1872

NAS Parallel benchmarks

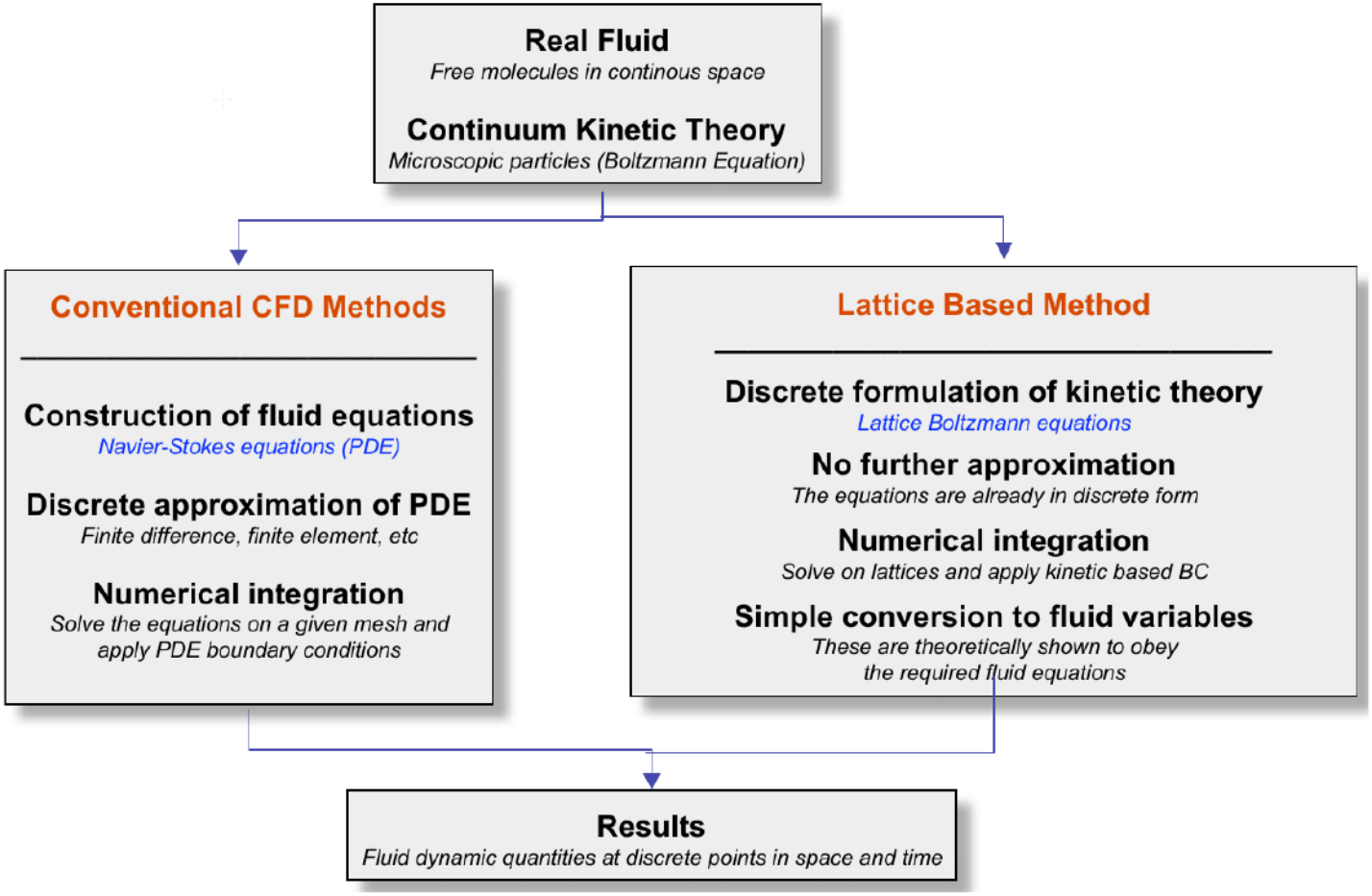
Mop/s total	B/2	B/4	B/8	C/2	C/4	C/8
BT-MZ MPI	2154.63	2150.28	2139.69	2166.28	2160.56	2151.38
LU-MZ MPI	1803.01	192.8	178.49	1479.25	444.78	4692.42
SP-MZ MPI	1415.40	1395.76	1373.63	1360.97	1354.54	1329.94

Time in seconds	B/2	B/4	B/8	C/2	C/4	C/8
BT-MZ MPI	279.03	279.59	280.98	1120.35	1123.31	1128.10
LU-MZ MPI	248.98	2327.85	2515.05	1298.30	4317.92	409.28
SP-MZ MPI	214.32	217.34	220.84	899.91	904.18	920.91

CFD Codes tests

The choice was driven to some previous experiences in deploying and delivering complex CPU GPU cluster at PRINCE Politecnico di Bari (60 nodes with 2xE5-2680v3) where among commercial codes we had to do benchmarks with SAILFISH CFD a Lattice Boltzmann CPU/GPU oriented program with CUDA/OpenCL capabilities

CFD solvers - Theory



CFD solvers - Theory

Macroscopic scale: continuum, velocity ($\vec{v}$), pressure (p),
Navier-Stokes equation:

$$\rho\left(\frac{\partial\vec{v}}{\partial t}+\vec{v}\cdot\nabla\vec{v}\right)=-\nabla p+\mu\nabla^2\vec{v}+\vec{f}$$

CFD solvers - Theory

Mesoscopic scale: particle ensemble, the lattice Boltzmann method.

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial \vec{x}} \cdot \frac{\vec{p}}{m} + \frac{\partial f}{\partial \vec{p}} \cdot \vec{F} = \left.\frac{\partial f}{\partial t}\right|_{\text{coll}}$$

Lattice Boltzmann - Basic

Discrete, regular, Cartesian grid (i is a node index).

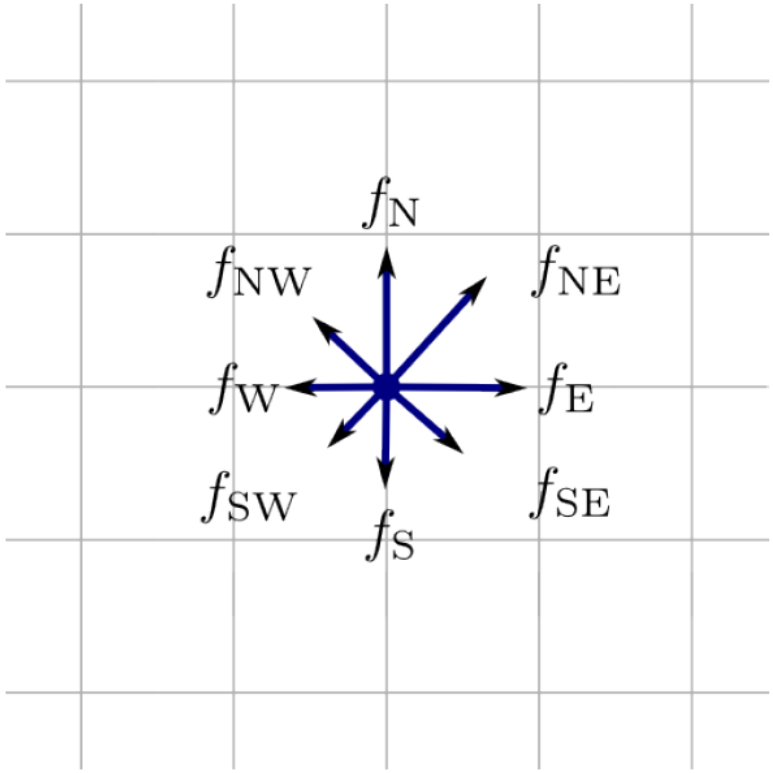
Mass fractions: f_α:

$f_C, f_E, f_W, f_S, f_N, f_{NE}, f_{NW}, f_{SE}, f_{SW}$

Macroscopic quantities:

$$\rho_i = \sum_\alpha f_\alpha(\vec{x}_i, t)$$

$$\rho_i \vec{v}_i = \sum_\alpha \vec{c}_\alpha f_\alpha(\vec{x}_i, t)$$



Forum Teratec

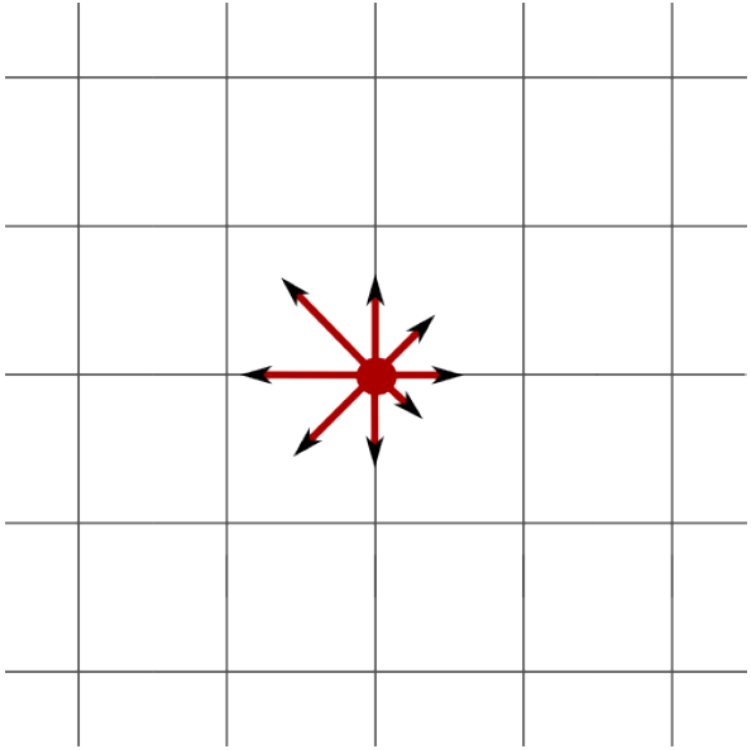
Lattice Boltzmann – The algorithm

1. Collision:

$$f_\alpha^\star(\vec{x}_i, t) = f_\alpha(\vec{x}_i, t) - \frac{f_\alpha(\vec{x}_i, t) - f_\alpha^{(\mathrm{eq})}(\rho_i, \vec{v}_i)}{\tau}$$

2. Streaming:

$$f_\alpha(\vec{x}_i + \vec{c}_\alpha, t+1) = f_\alpha^\star(\vec{x}_i, t)$$



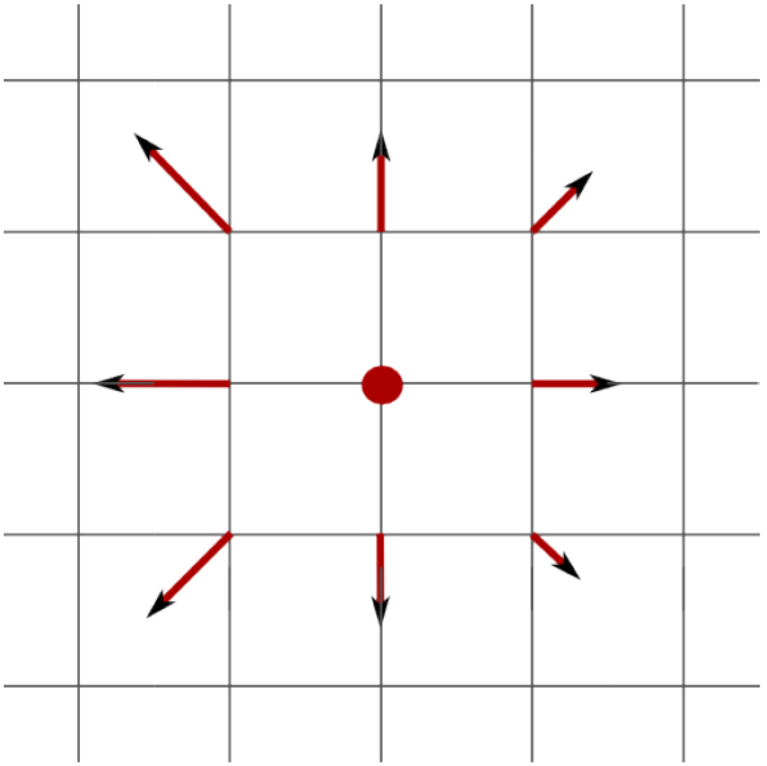
Lattice Boltzmann – The algorithm

1. Collision:

$$f_\alpha^\star(\vec{x}_i, t) = f_\alpha(\vec{x}_i, t) - \frac{f_\alpha(\vec{x}_i, t) - f_\alpha^{(\mathrm{eq})}(\rho_i, \vec{v}_i)}{\tau}$$

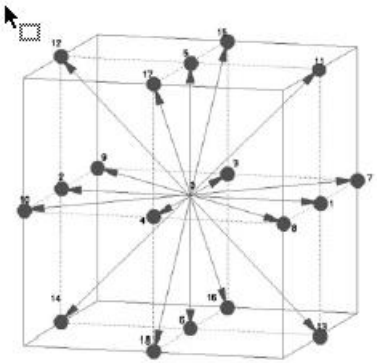
2. Streaming:

$$f_\alpha(\vec{x}_i + \vec{c}_\alpha, t+1) = f_\alpha^\star(\vec{x}_i, t)$$



Lattice Boltzmann – The algorithm

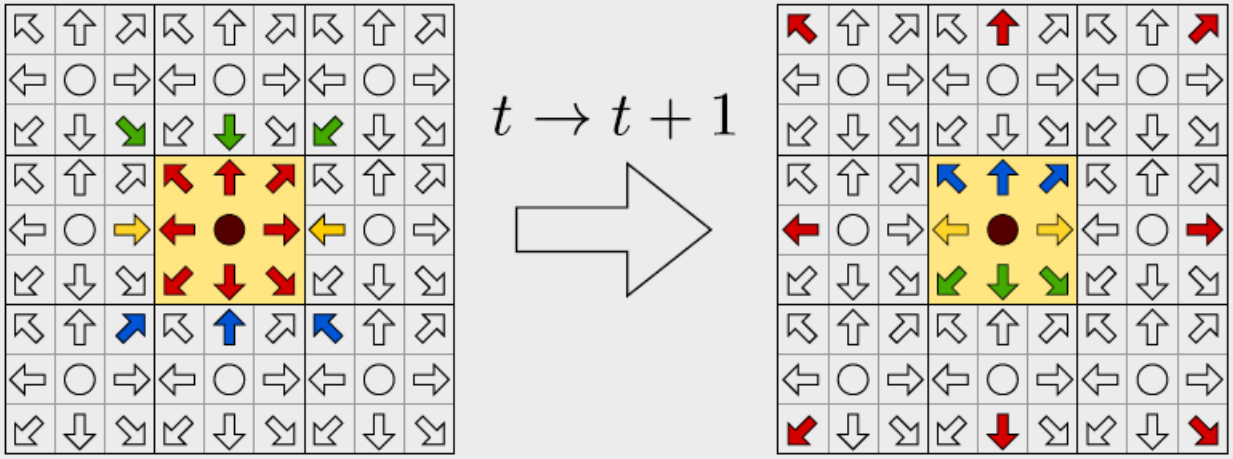
3D lattice



D3Q19

Lattice Boltzmann – WHY?

- Applicable for low Mach number flows.
- Good for flows in complex domains (e.g. porous materials).
- **Extremely well parallelizable (nearest-neighbor interactions).**
- **Easy to implement.**



The CFD code tested

SAILFISH-CFD

- GPU-based implementation of the lattice Boltzmann method.
- Open source (LGPL v3).
- Implemented using Python and CUDA C / OpenCL.
- GPU part implemented compiling on the fly the kernel for the specific problem solved

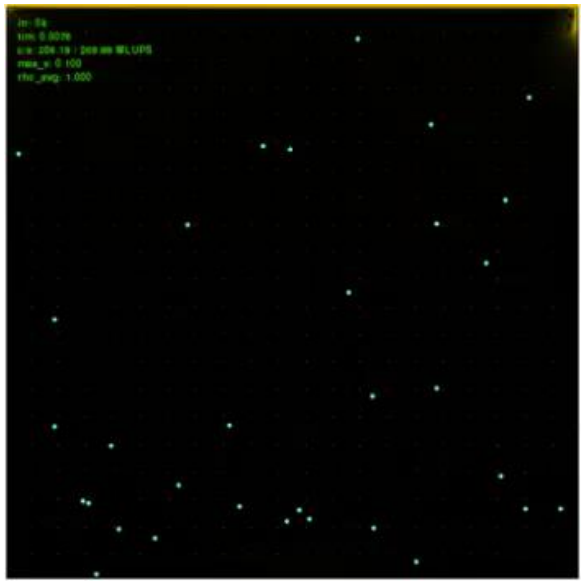
SAILFISH-CFD

Despite the new system was up and running there where some troubles trying to deploying this CFD software mainly due to:

- Problems with GCC compiler: the compiler had to be patched by the processor producer in order to build correct codes-
- Problems with python modules and packages: We had to recompile a couple of modules and libraries in order to get the full environment to work (pyCUDA)
- Troubles with NVIDIA CUDA ARM64 (yes compiler,& drivers) no support for outside SDK installation
- No OpenCL support

SAILFISH-CFD

At last we got it working!

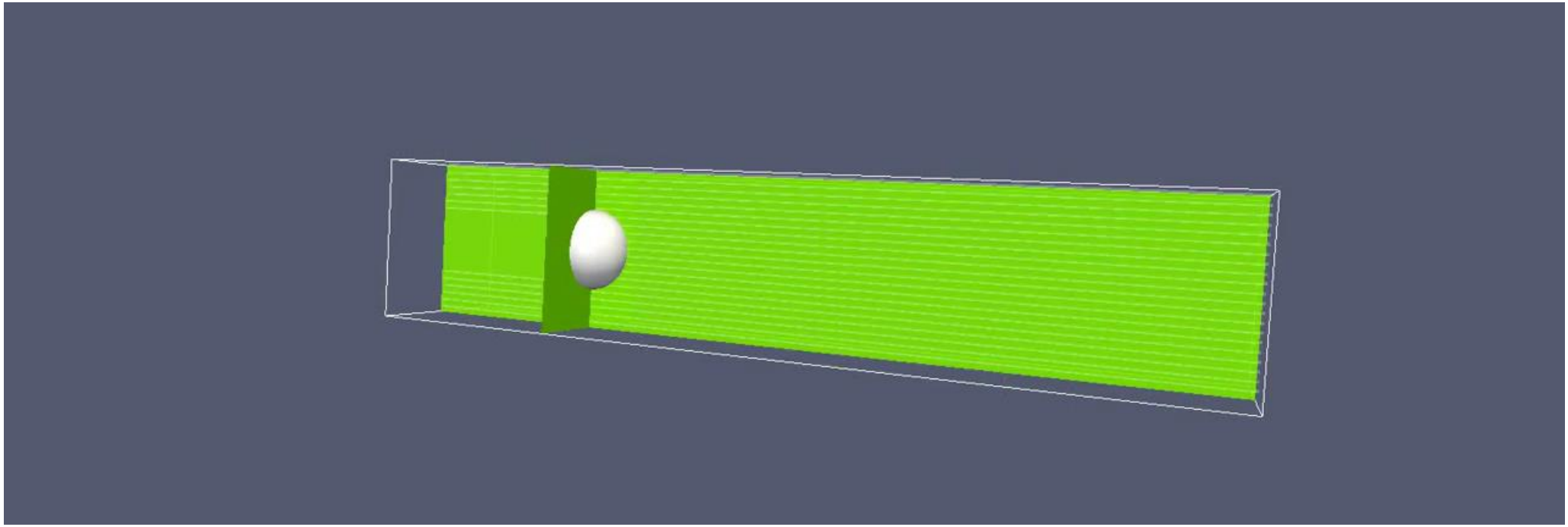


SAILFISH-CFD

Performances:
Extensive regression tests and examples where run with different options in order to identify bottlenecks

SAILFISH-CFD

On 3D problems



SAILFISH-CFD – 3D comparison

```
gino.perna@node3:/tmp/gino/sailfish-cfd$ ./ginobench/s_3d.py
[  1916  INFO Master/node3] Machine master starting with PID 18852 at 2015-06-23 03:20:07 UTC
[  1925  INFO Master/node3] Sailfish version: 3c4df9af3125ae2f4b2d57c29cf92dcea12b3528
[  1925  INFO Master/node3] Handling subdomains: [0]
[  1926  INFO Master/node3] Subdomain -> GPU map: {0: 0}
[  1926  INFO Master/node3] Selected backend: cuda
[  3174  INFO Subdomain/0] Initializing subdomain.
[  3176  INFO Subdomain/0] Required memory:
[  3176  INFO Subdomain/0] . distributions: 1010 MiB
[  3177  INFO Subdomain/0] . fields: 100 MiB
[ 14404  INFO Subdomain/0] Starting simulation.
[ 16616  INFO Subdomain/0] iteration:2000  speed:443.86 MLUPS
[ 17712  INFO Subdomain/0] iteration:3000  speed:438.71 MLUPS
[ 18808  INFO Subdomain/0] iteration:4000  speed:444.24 MLUPS
[ 19904  INFO Subdomain/0] iteration:5000  speed:444.24 MLUPS
[ 21000  INFO Subdomain/0] iteration:6000  speed:444.22 MLUPS
```

model name : APM X-Gene Mustang board
cpu MHz : 2400.00
fpu : yes

Features : fp asimd
CPU implementer : 0x50
CPU architecture: AArch64

```
user@user-System-Product-Name:~/sailfish-cfd$  ./ginobench/s_3d.py
[   612  INFO Master/user-System-Product-Name] Machine master starting with PID 2100 at 2015-
06-23 03:23:07 UTC
[   618  INFO Master/user-System-Product-Name] Sailfish version:
3c4df9af3125ae2f4b2d57c29cf92dcea12b3528
[   618  INFO Master/user-System-Product-Name] Handling subdomains: [0]
[   618  INFO Master/user-System-Product-Name] Subdomain -> GPU map: {0: 0}
[   618  INFO Master/user-System-Product-Name] Selected backend: cuda
[  1679  INFO Subdomain/0] Initializing subdomain.
[  1680  INFO Subdomain/0] Required memory:
[  1680  INFO Subdomain/0] . distributions: 1010 MiB
[  1681  INFO Subdomain/0] . fields: 100 MiB
[  5714  INFO Subdomain/0] Starting simulation.
[  7740  INFO Subdomain/0] iteration:2000  speed:483.07 MLUPS
[  8747  INFO Subdomain/0] iteration:3000  speed:479.49 MLUPS
[  9755  INFO Subdomain/0] iteration:4000  speed:483.14 MLUPS
[ 10763  INFO Subdomain/0] iteration:5000  speed:483.09 MLUPS
[ 11771  INFO Subdomain/0] iteration:6000  speed:482.97 MLUPS
[ 12779  INFO Subdomain/0] iteration:7000  speed:483.10 MLUPS
```

Intel(R) Core(TM) i5-2400 CPU @ 3.10GHz
stepping : 7

SAILFISH-CFD – overall

```
time python regtest/subdomains/2d_propagation.py
............
----------------------------------------------------------------------
Ran 12 tests in 125.015s
OK
194.84user 16.57system 2:07.20elapsed 166%CPU (0avgtext+0avgdata 142780maxresident)k
0inputs+513832outputs (0major+2190207minor)pagefaults 0swaps
time python regtest/subdomains/2d_ldc.py
..
----------------------------------------------------------------------
Ran 2 tests in 36.496s
OK
68.33user 5.16system 0:38.66elapsed 190%CPU (0avgtext+0avgdata 143256maxresident)k
0inputs+109096outputs (0major+665775minor)pagefaults 0swaps
time python regtest/subdomains/2d_cylinder.py
....
----------------------------------------------------------------------
Ran 4 tests in 67.593s
OK
95.48user 7.89system 1:09.77elapsed 148%CPU (0avgtext+0avgdata 146520maxresident)k
0inputs+169624outputs (0major+994431minor)pagefaults 0swaps
time python regtest/subdomains/3d_propagation.py
...........
----------------------------------------------------------------------
Ran 11 tests in 154.726s
OK
228.67user 17.85system 2:36.91elapsed 157%CPU (0avgtext+0avgdata 198964maxresident)k
0inputs+2326672outputs (0major+1930527minor)pagefaults 0swaps
time python regtest/subdomains/3d_ldc.py
.
----------------------------------------------------------------------
Ran 1 test in 48.749s
OK
183.44user 8.13system 0:50.92elapsed 376%CPU (0avgtext+0avgdata 162788maxresident)k
8inputs+195288outputs (0major+764999minor)pagefaults 0swaps
time python regtest/subdomains/binary_pbc.py
.....
----------------------------------------------------------------------
Ran 5 tests in 161.309s
OK
129.79user 10.05system 2:43.48elapsed 85%CPU (0avgtext+0avgdata 156156maxresident)k
0inputs+312312outputs (0major+939946minor)pagefaults 0swaps
time python regtest/subdomains/2d_binary.py
:
```

```
python regtest/subdomains/2d_propagation.py
............
----------------------------------------------------------------------
Ran 12 tests in 62.869s

OK
python regtest/subdomains/2d_ldc.py
..
----------------------------------------------------------------------
Ran 2 tests in 17.910s

OK
python regtest/subdomains/2d_cylinder.py
....
----------------------------------------------------------------------
Ran 4 tests in 31.407s

OK
python regtest/subdomains/3d_propagation.py
...........
----------------------------------------------------------------------
Ran 11 tests in 77.027s

OK
python regtest/subdomains/3d_ldc.py
.
----------------------------------------------------------------------
Ran 1 test in 21.525s

OK
python regtest/subdomains/binary_pbc.py
.....
----------------------------------------------------------------------
Ran 5 tests in 77.317s

OK
python regtest/subdomains/2d_binary.py
......
----------------------------------------------------------------------
Ran 6 tests in 51.511s
```

CFD on arm64 – next steps

Test on multiGPU on the same board

Must wait for NVIDIA to release firmware patch for K40

Should wait official IB drivers

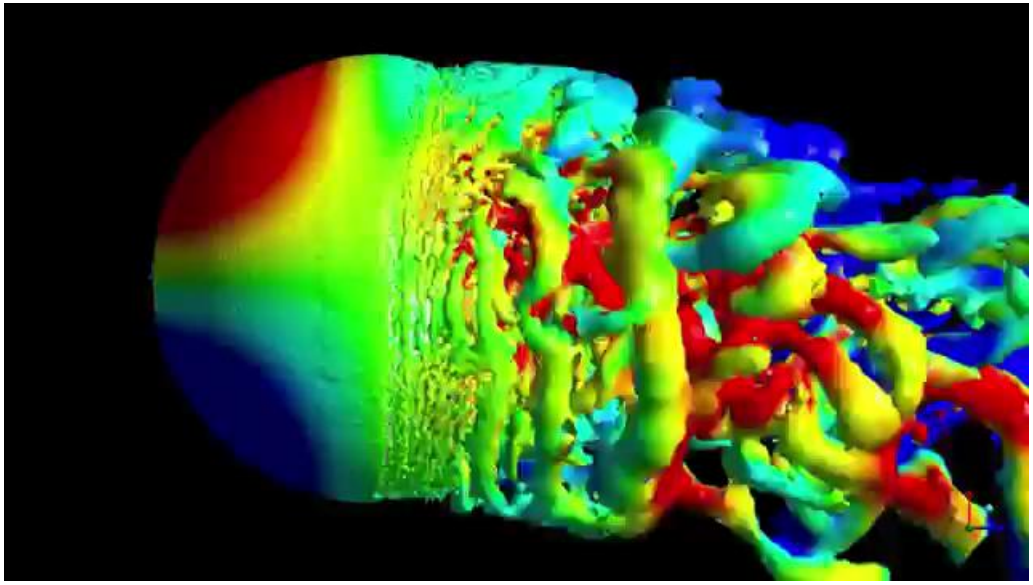
OpenCL tests when ready

EUROTECH
Imagine. Build. Succeed.

Forum Teratec

CFD on arm64 – next steps

Run on multiGPU - multi node ARM64 cluster
Profile the code to search bottleneck



Questions?
Thank you for your attention

g.perna@enginsoft.it