Reliable and Sustainable Computations: A Brief Overview

Roman Iakymchuk[1]
joint work with
José Aliaga (UJI), Stef Graillat (Sorbonne),
and Pablo Oliveira (Paris-Saclay)

Umeå University
Sweden
riakymch@cs.umu.se

SIAM CSE
Amsterdam, The Netherlands
March 1st, 2023

Sustainable Computing with the Help of Tools, Mixed-Precision, and Optimistic Error Estimates

MS177		
09:45AM	Roman Iakymchuk	Reliable and Sustainable Computations: A Brief Overview
10:05AM	Marc Casas	Trade-Offs between Performance, Energy, and Accuracy of Non-Standard Computer Number Formats
10:25AM	Pablo De Oliveira Castro	Verificarlo: Tuning and Debugging Floating-Point Computations Through Stochastic Rounding
10:55AM	David Defour	Interflop: a Project for Interoperable Tools for Computing, Debugging, Validation and Optimization of Floating-Point Programs

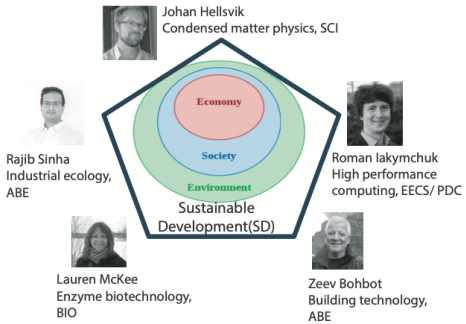
Outline

Sustainability is ...

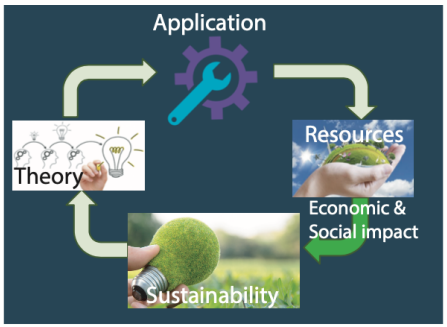
Incorporating Sustainable Development into the Design of a Student Project

Zeev Bohbot, Johan Hellsvik, Roman Iakymchuk, Lauren McKee, Rajib Sinha
KTH Royal Institute of Technology, Stockholm, Sweden

Who we are?



How is everything connected to Sustainability



KTH's goals for sustainability 2016-2020

KTH Environment Education
Basic knowledge about KTH's work with the environment and Sustainable Development (SD)

Education
Increase all employee's and student's knowledge
Integrate SD into all educational programs at all levels

Research
Increase research for SD
Increase the integration of SD in KTH's research base

How can we incorporate sustainable development in a student project?

Resources are needed to conduct tasks relating to the project. 'Resource Footprints are Good Proxies of Environmental Damage' (Steinmann et al 2017). Minimize the resource use of i) designing a project, ii) activities in the project, iii) component in the project, and iv) economic and social impacts

Sustainable development project framework
Study the current system (Systems approach)
Connect the sustainability issues (3 pillars of SD)
Apply IE perspective* to understand and solve the issues

*Waste as a resource, Industrial symbiosis, Closed loop/zero waste/circular economy, Material/energy flows, Life cycle thinking, Systems perspective

Energy-efficient architectures such as graphic processors (GPUs)

Sustainable algorithms

Some applications do not require 'full accuracy' answers: i) signal & video processing, ii) Monte Carlo simulation, iii) machine learning

Adaptive precision calculations
Mixed precision solvers with iterative refinement
Neural networks and deep learning

Application in Computer Science: High Performance Computing (HPC)

Reference

Steinmann, Z.J., Schipper, A.M., Hauck, M., Giljum, S., Wernet, G. and Huijbregts, M.A., 2017. Resource footprints are good proxies of environmental damage. Environmental science & technology, 51(11), pp.6360-6366.

Today's syllabus: resource awareness

X 30 =

- lagom – not too much, not too little, the right amount
- Sustainability is the **art of living well, within the ecological limits** of a finite planet. (Jackson, 2010)
- How can we incorporate sustainability into Scientific Computing, including HPC?

Sustainable HPC → Energy-efficient HPC

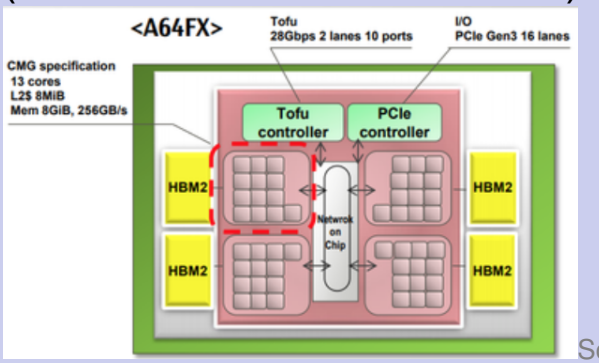
- **Energy-efficient architectures** such as graphic processors (GPUs) and FPGAs – Green HPC computing
- PDC@KTH extracts the produced heat to **warm up the main campus**
- CSCS at Switzerland proposes **'free cooling'** with the water from the lake of Lugano

Exascale computing and linear algebra

- Exascale computing is constrained by **power consumption**

→ Power-efficient hardware

- RIKEN's Fugaku w A64FX (FP64:FP32:FP16 = 1:2:4)
- EPI (ARM, FPGA, RISC-V)



Source: Fujitsu

- Linear algebra is known to be dominant by **double precision**

→ Sustainable algorithms

math **Mixed and adaptive precision computing**

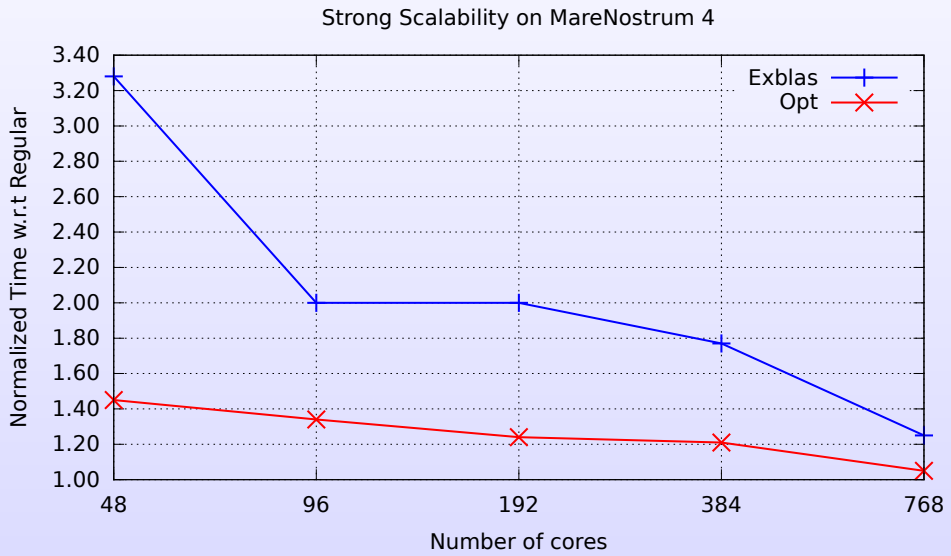
code **Communication hiding or avoiding**

Robustness of Algorithms

- **Robustness: accuracy and reproducibility**
- FP ops are non-associative : $(-1+1)+2^{-53} \neq -1+(1+2^{-53})$
- Non-reproducibility in PCG: `dot`, `axpy`, and `spmv`

→ Solution : ExBLAS (ParCo15, NRE15, JCAM, IJHPCA)

3D Poisson equation with 27 stencil points and $tol = 10^{-8}$



Iteration	Residual			
	MPFR	*Original* 1 proc	*Original* 48 procs	*Exblas & Opt*
0	0x1.19f179eb7f032p+49	0x1.19f179eb7f03**3**p+49	0x1.19f179eb7f03**3**p+49	0x1.19f179eb7f032p+49
2	0x1.f86089ece9f75p+38	0x1.f86089**f08810d**p+38	0x1.f86089e**d07a76**p+38	0x1.f86089ece9f75p+38
9	0x1.fc59a29d329ffp+28	0x1.fc59a29d**1b6a**p+28	0x1.fc59a29d**2e989**p+28	0x1.fc59a29d329ffp+28
10	0x1.74f5ccc211471p+22	0x1.74f5cc**b8203ad**p+22	0x1.74f5ccc**1fafef**p+22	0x1.74f5ccc211471p+22
...	...	...	...	...
40	0x1.7031058eb2e3ep-19	0x1.703105**aea0e8a**p-19	0x1.7031058e**8ff5a**p-19	0x1.7031058eb2e3ep-19
42	0x1.4828f76bd68afp-23	0x1.4828f**6fabbf2a**p-23	0x1.4828f76b**b9038**p-23	0x1.4828f76bd68afp-23
45	0x1.8646260a70678p-26	0x1.864626**01300d2**p-26	0x1.8646260a7**1301**p-26	0x1.8646260a70678p-26
47	0x1.13fa97e2419c7p-33	0x1.13fa9**8038c44e**p-33	0x1.13fa97e**54e903**p-33	0x1.13fa97e2419c7p-33

Table 3: Accuracy and reproducibility comparison on the intermediate and final residual against MPFR for a matrix with condition number of 10^{12}. The matrix is generated following the procedure from Section 5.1 with n=4,019,679 (159^3).

European Processor Initiative

VRP and STX

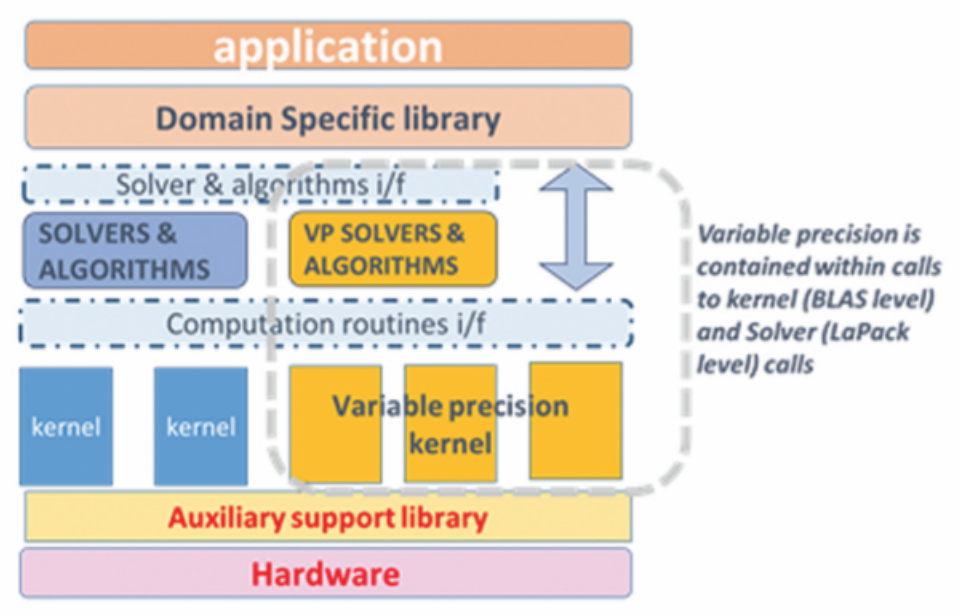
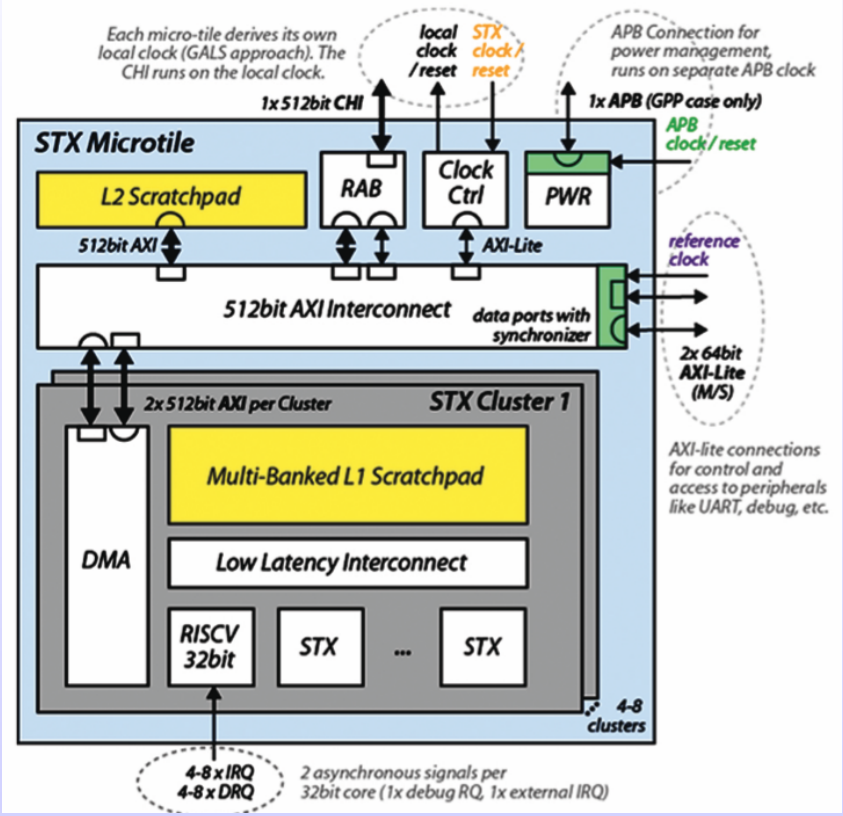


Fig: Layered programming model

- for large ill-conditioned systems
- "when the standard precision unit cannot reach the expected accuracy, the variable precision unit takes the relay"
- zero-copy from GPP to VRP

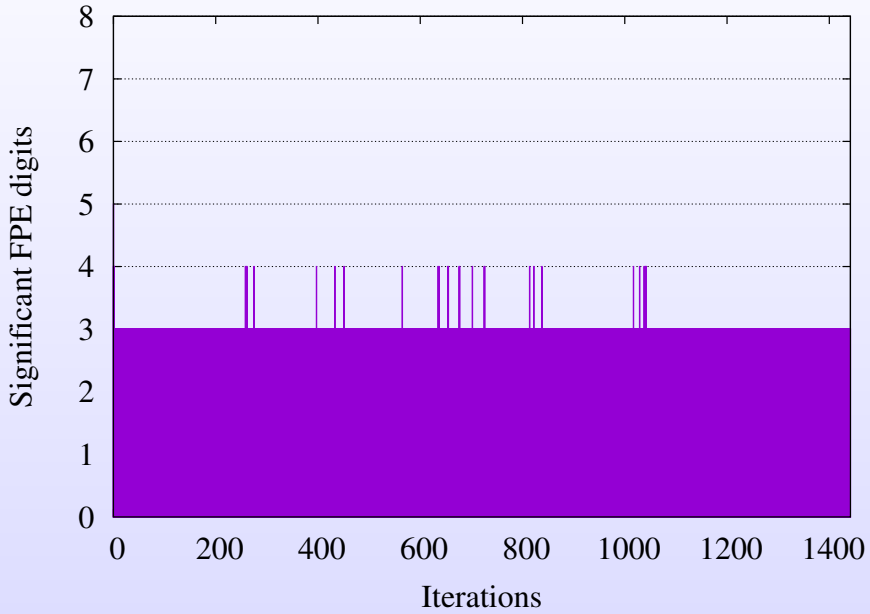


- Stencil/ tensor accelerator
- Energy efficiency
- Posit-based ML & DNN Acceleration

Source: EPI

Possibilities to exploit VRP

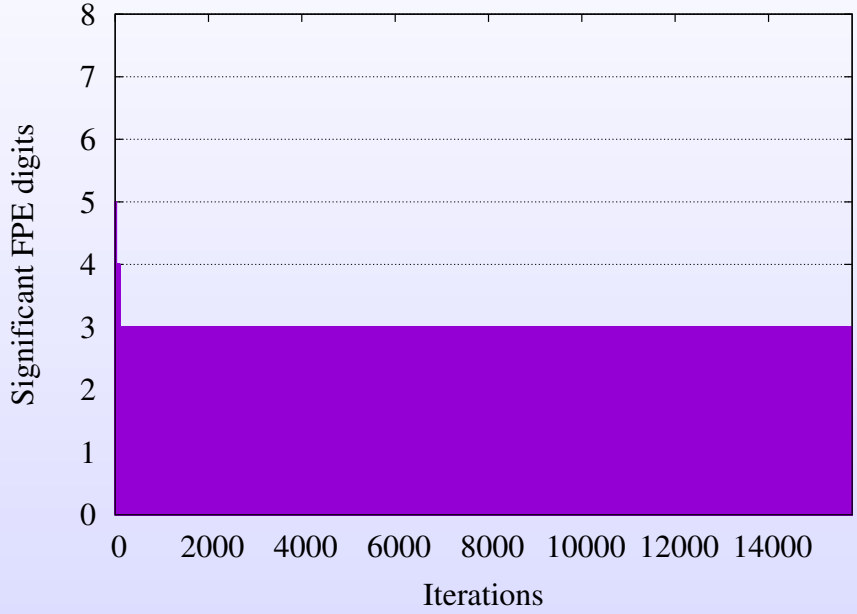
Required precision in PCG to keep every bit



msc01050 (Boeing)

$NNZ = 26,198$

$cond(A) = 9.0e+15$

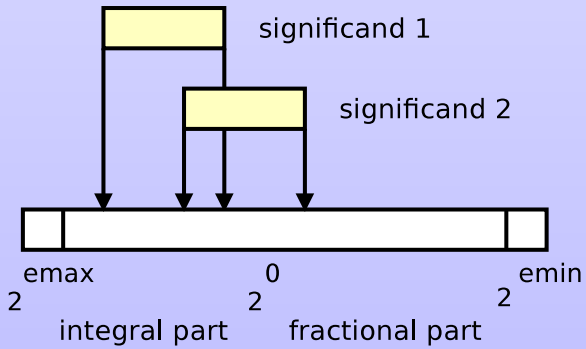


gyro_k (Oberwolfach)

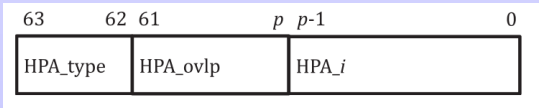
$NNZ = 1,021,159$

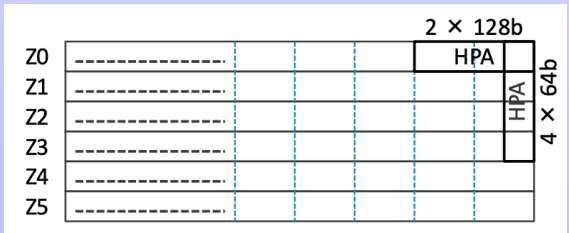
$cond(A) = 1.1e+09$

Kulisch accumulator



ARM High-Precision Anchored





Source: ARM. IEEE ToC18

Sustainable algorithms

Analysis w tools $\rightarrow$ Strategy $\rightarrow$ Revision of algorithms

Sustainable algorithms

Analysis w tools $\rightarrow$ Strategy $\rightarrow$ Revision of algorithms

Workplan

1. **Arithmetic tool** applied to **code** $\rightarrow$ **optimized binary**
2. iff $gain \geq 3\,\%$, apply algorithmic solutions
3. conduct probabilistic (aka optimistic) error analysis
   - error bound with constant $\sqrt{n}\mu$ with high probability
4. Implement on hardware with stochastic rounding support – randomly maps x to one of two bounds

Analysis with tools

- verificarlo – an automatic tool for debugging and assessing FP precision based on Monte Carlo Arithmetic
- `Vprec` is a backend to emulate variable FP representations

k	x_{k+1}	s_k^{10}	s_k^2
0	0.0690266447076745	0.11	0.37
1	0.1230846130203958	0.21	0.70
2	0.1985746566605835	0.43	1.43
3	0.2732703639721015	0.84	2.79
4	0.3119369815109966	1.79	5.95
5	0.3181822938100336	3.40	11.3
6	0.3183098350392471	6.79	22.6
7	0.3183098861837825	13.6	45.2
8	0.3183098861837907	15.6	51.8
9	0.3183098861837907	15.6	51.8

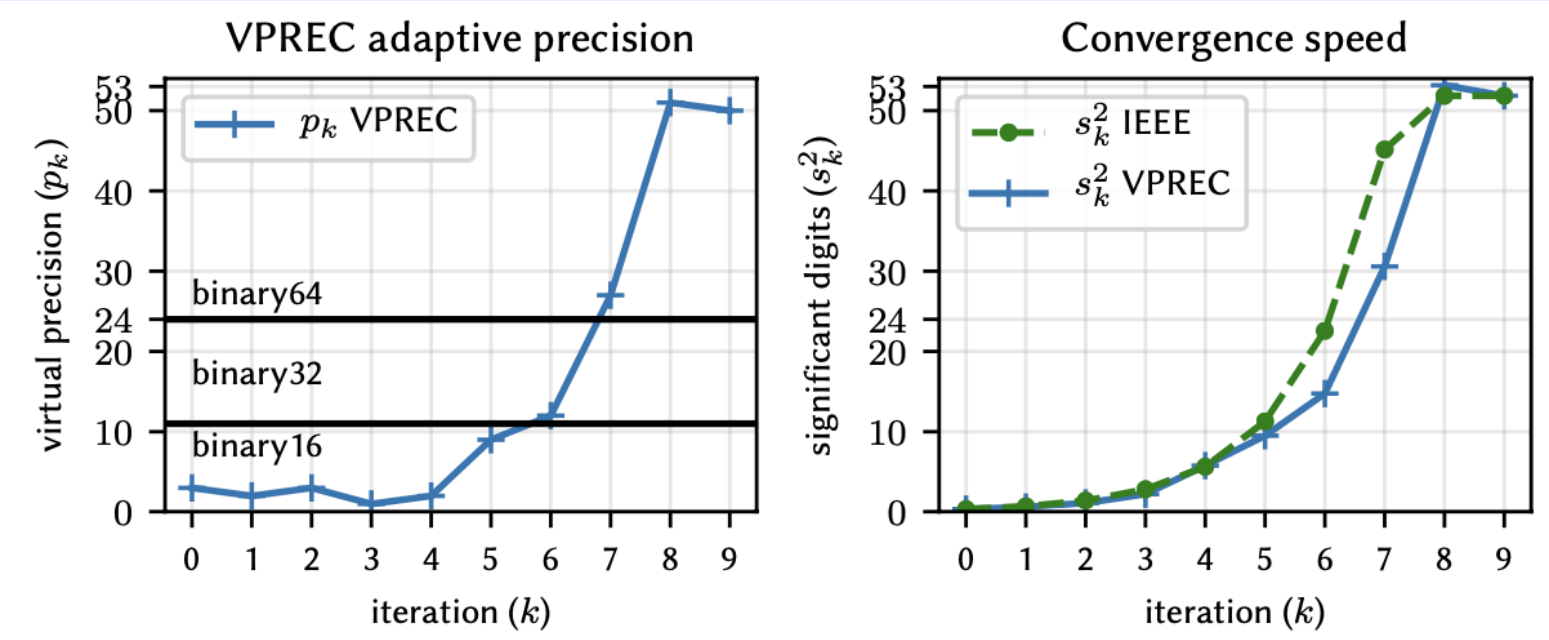
```
double newton(double x0) {
  double x_k, x_k1=x0, b=PI;
  do {
    x_k  = x_k1;
    x_k1 = x_k*(2-b*x_k);
  }while (fabs((x_k1-x_k)/x_k)
  >= 1e-15);
  return x_k1;
}
```

The Newton-Raphson method for inverse of π[a]

[a]Pablo Oliveira et al. *Automatic exploration of reduced floating-point representations in iterative methods.* Euro-Par 2019

Analysis with tools

- verificarlo – an automatic tool for debugging and assessing FP precision based on Monte Carlo Arithmetic
- `Vprec` is a backend to emulate variable FP representations

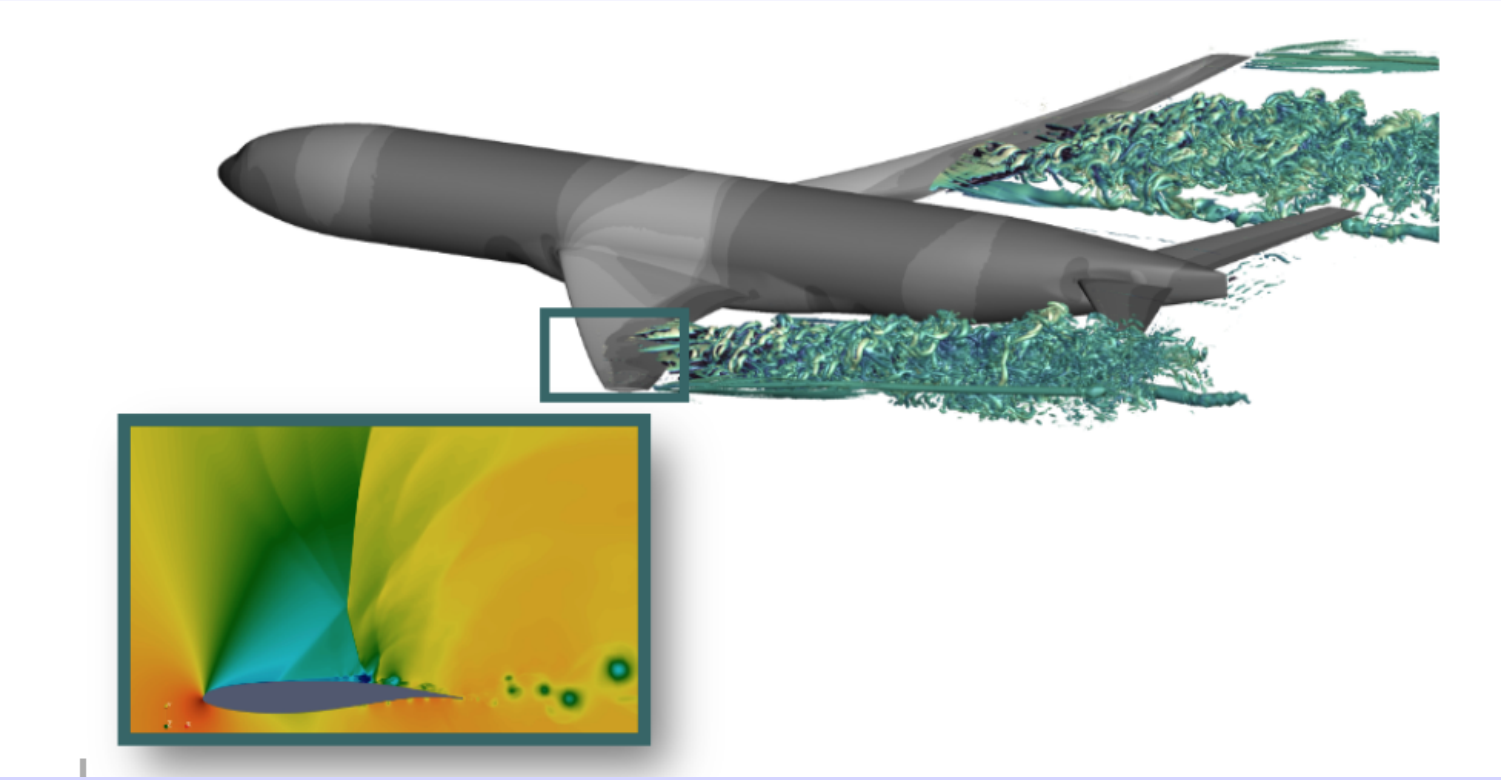


The Newton-Raphson method for inverse of π[a]

[a]Pablo Oliveira et al. *Automatic exploration of reduced floating-point representations in iterative methods.* Euro-Par 2019

EuroHPC JU CoE CEEC: 2023-26

Center of Excellence for Exascale CFD



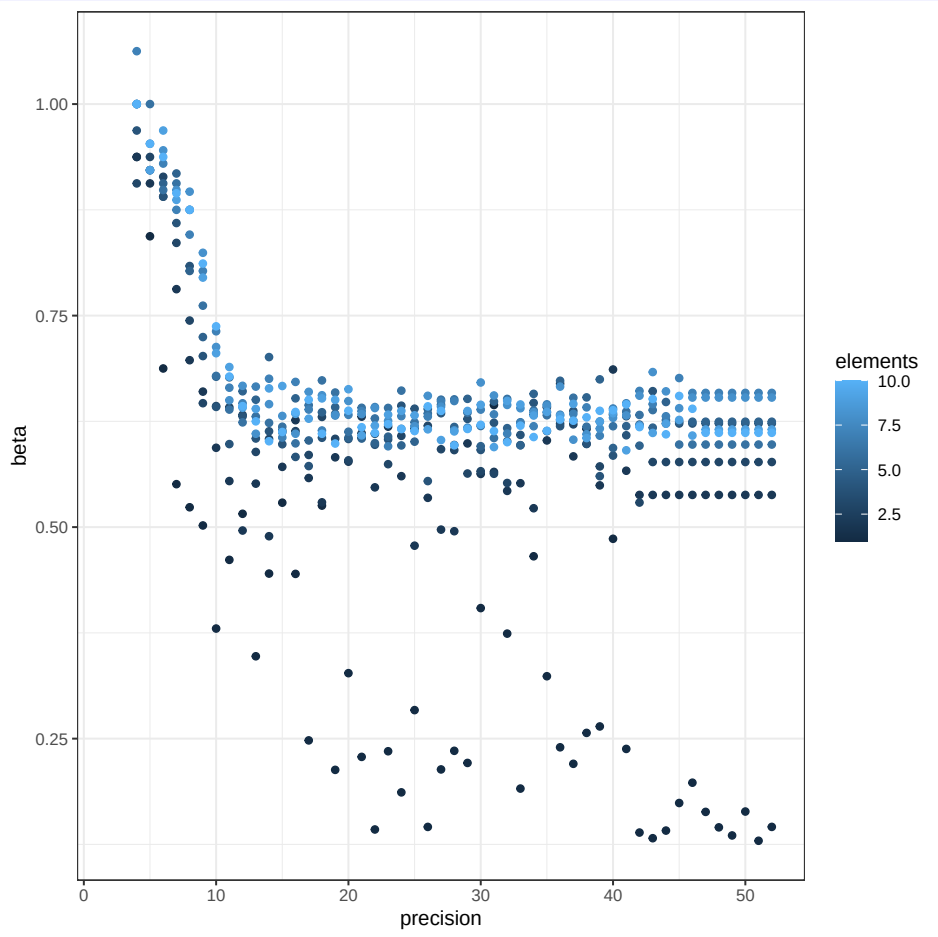
- **Sustainable algorithmic solutions with mixed-precision and tools**
- **WP lead** on Exascale algorithms: scalable, resilient & sustainable
- **Test cases**: Neko@KTH (NEK5000) and other consortium CFD codes

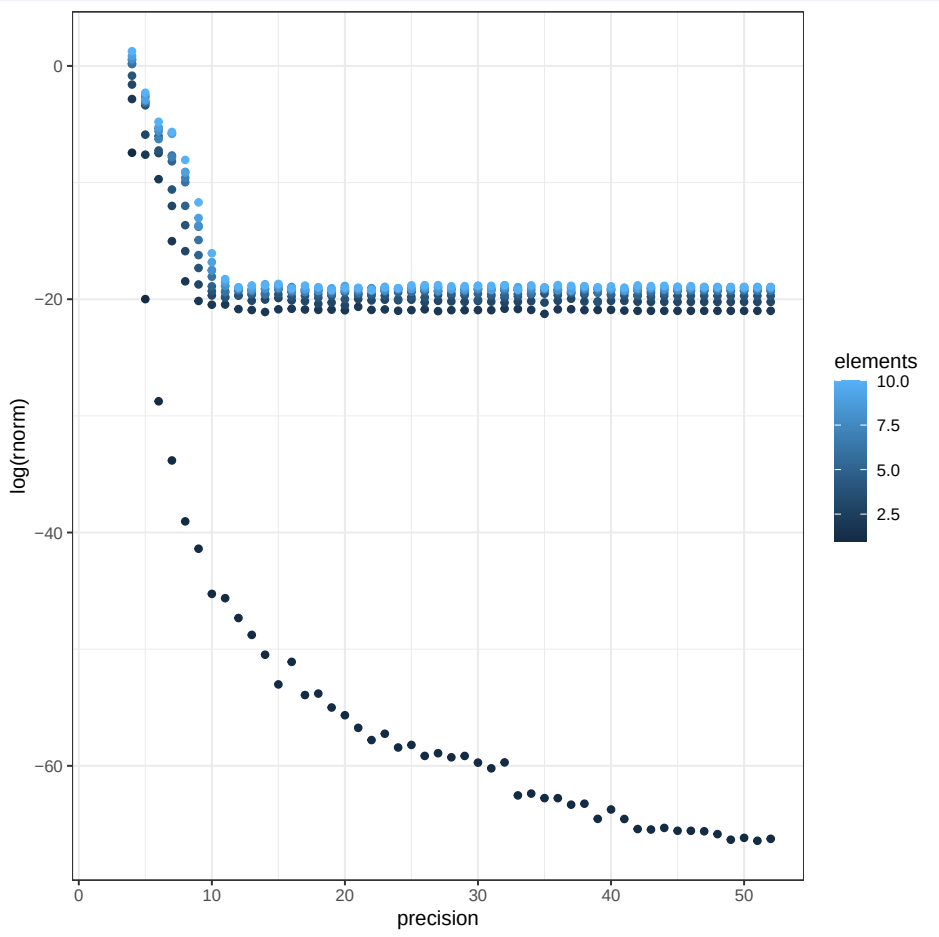
Nekbone test case

- **NEK5000** is a high order, incompressible Navier-Stokes solver based on the spectral element method
- **Nekbone** solves a Poisson equation using a Conjugate Gradient method with a simple or spectral element multigrid preconditioner
- Nekbone is written in F77 and C plus MPI
- `example1` was used w/o the multi-grid preconditioner

Nekbone w `Vprec`

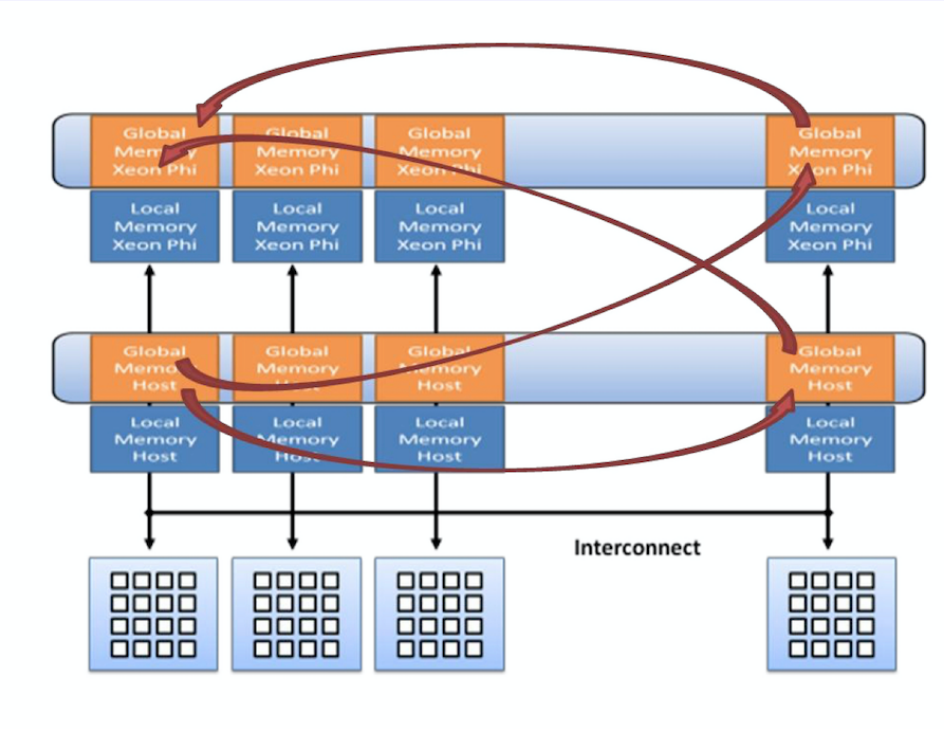
Preliminary results

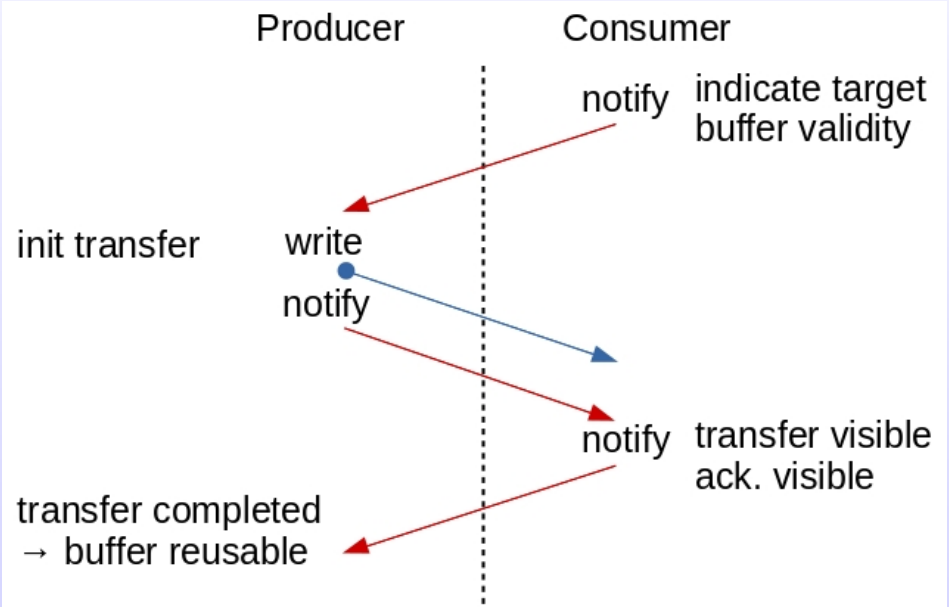




A Programming Model of Choice

GASPI: Global Address Space Programming Interface

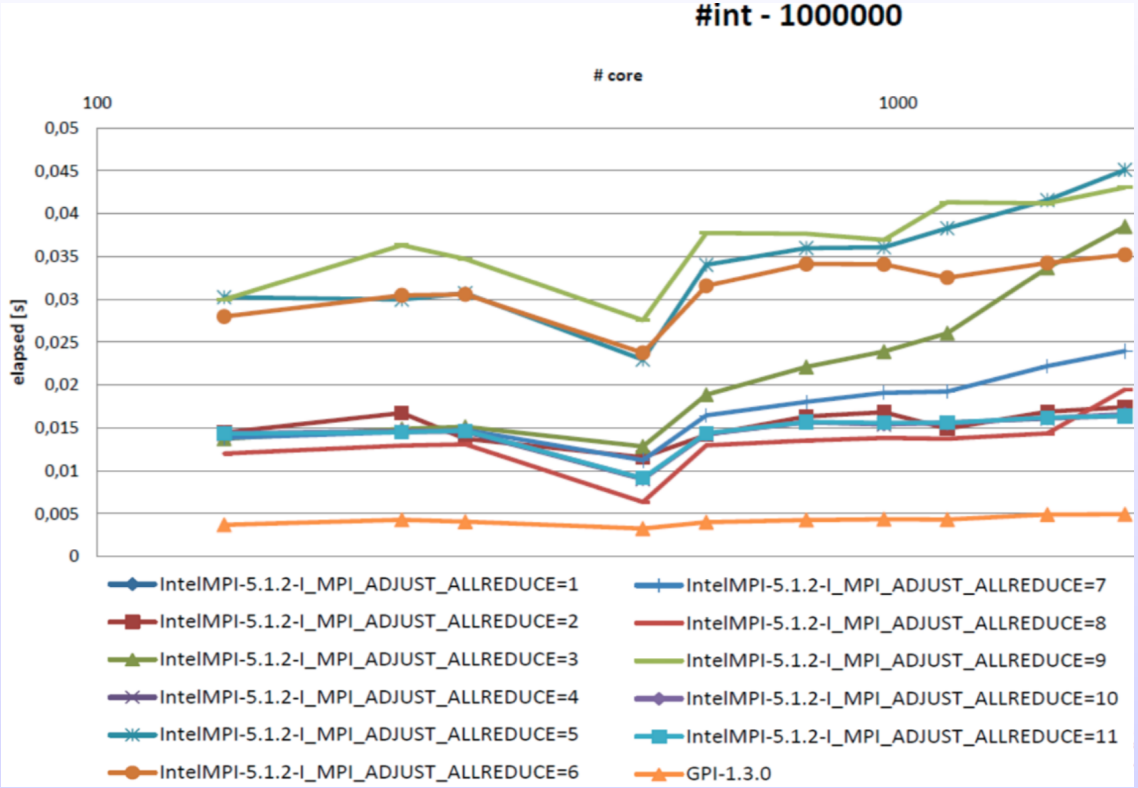




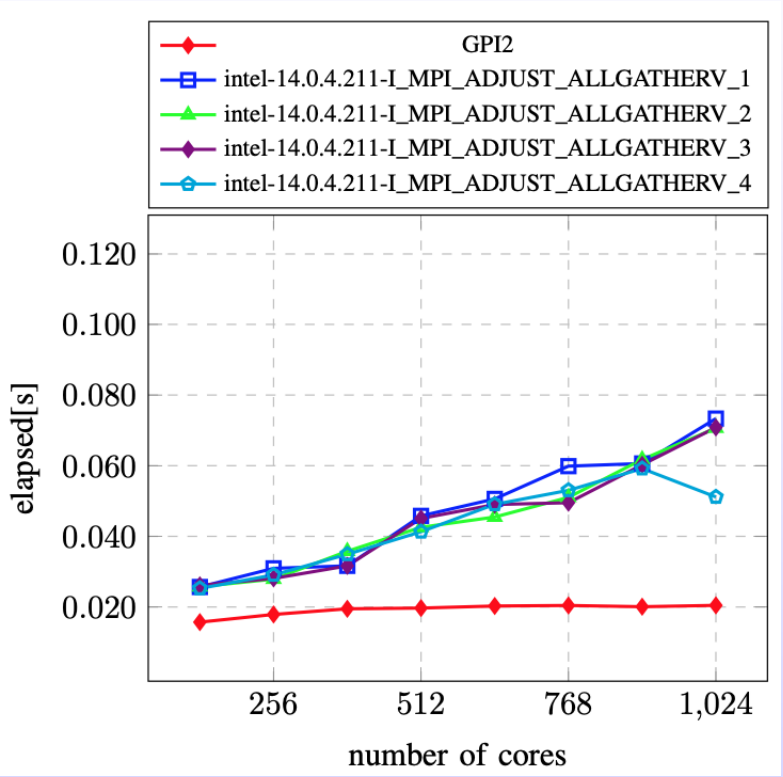
GASPI

- Asynchronous and one-sided communication
- Naturally overlaps communication and computation

Exploring shared memory w GASPI



Intel Ivy Bridge and an Infiniband FDR interconnect



Cray XC40 with Intel Xeon Broadwell and a Cray Aries interconnect

- Node-local ranks access **shared memory** (shared segment)
- Node-local ranks see all notifications node-wise (shared notifications)
- **Local collective operations are followed by global**

Thank you for your attention!

Sustainable Computing with the Help of Tools, Mixed-Precision, and Optimistic Error Estimates

MS177		
09:45AM	Roman Iakymchuk	Reliable and Sustainable Computations: A Brief Overview
10:05AM	Marc Casas	Trade-Offs between Performance, Energy, and Accuracy of Non-Standard Computer Number Formats
10:25AM	Pablo De Oliveira Castro	Verificarlo: Tuning and Debugging Floating-Point Computations Through Stochastic Rounding
10:55AM	David Defour	Interflop: a Project for Interoperable Tools for Computing, Debugging, Validation and Optimization of Floating-Point Programs