Kick-off Webinar

Friday 15th December 2023

HACKATHON HPC Codes industriels

Teams registration

September - October 2023

Kick-off webinar

15 december 2023 : 17h00 – 18h30

Getting to know the platform

Remote and free access

Hackathon

From monday january 22th 9h00 to Monday january 29th 9h00

Approximately 48 hours spread out as you wish

Awards ceremony at the Teratec Forum on 30 May 2024

17:00 : Welcome by Daniel Verwaerde, **Teratec** chairman

17h10 : Presentation of the codes and industrial issues :

- 17:10 : Code Telemac: Boris BASIC , **EDF**
- 17h25 : Riemann Zeta function, Patrick Demichel, **CGG**

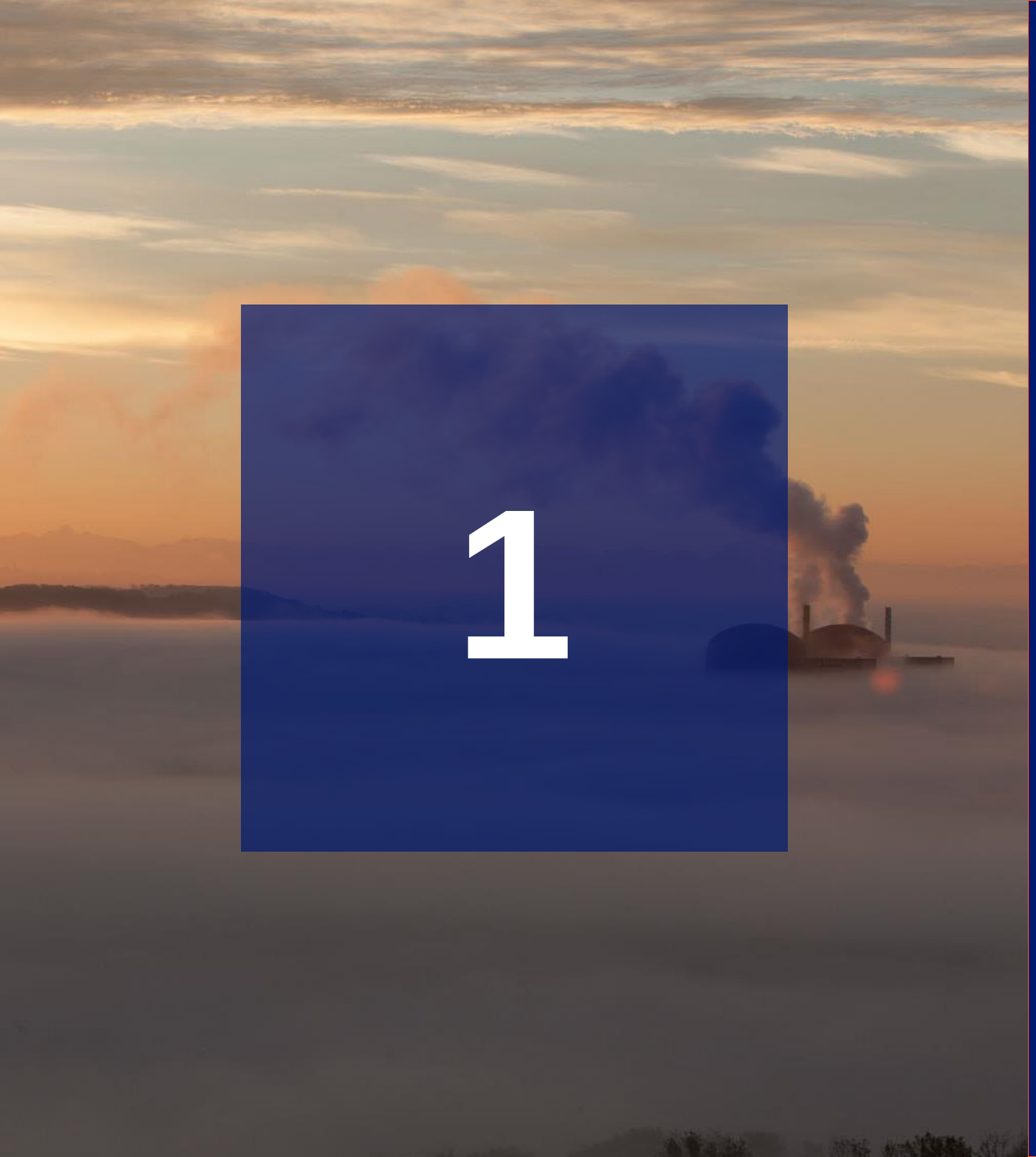
17h40 : Presentation of the platform and support :

- Conrad Hillairet**, ARM** and representing Gilles Tourpe from **AWS**
- Benjamin Depardon**, UCIT**
- Marcin Krzysztofik**, Linaro Forge**

18h15 : Questions / Answers

The TELEMAC system

Content



1

Introduction

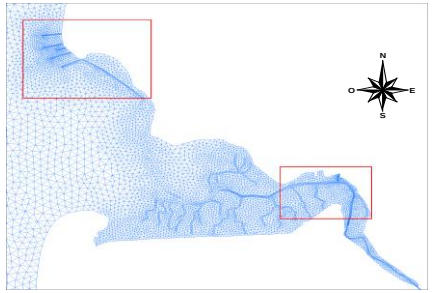
The TELEMAC system

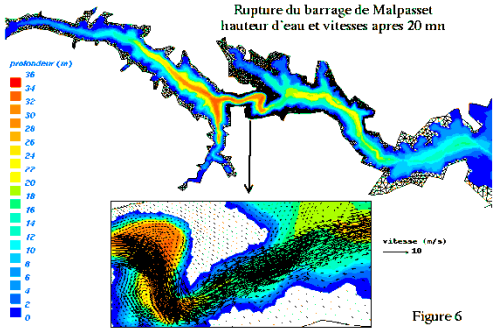
Main characteristics

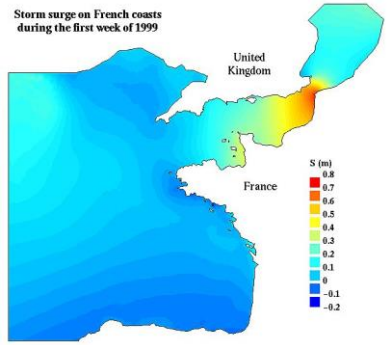
- Developed since 1987 at EDF R&D / LNHE
- Free and open source
- FORTRAN 90, Python 3
- API in FORTRAN and Python (TelApy)
- Based on unstructured grids
- Documentation and validation

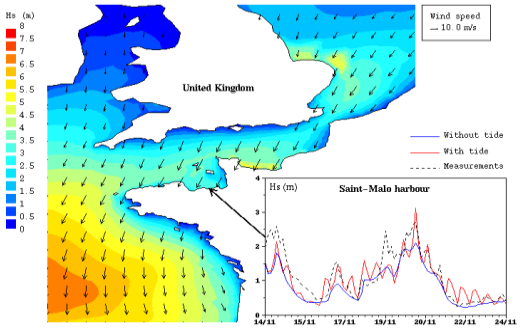
Key features

- Finite Elements or Finite Volumes, Implicit schemes
- Parallelism with domain decomposition
- Dry zones
- Code can be changed on the fly (user subroutines)

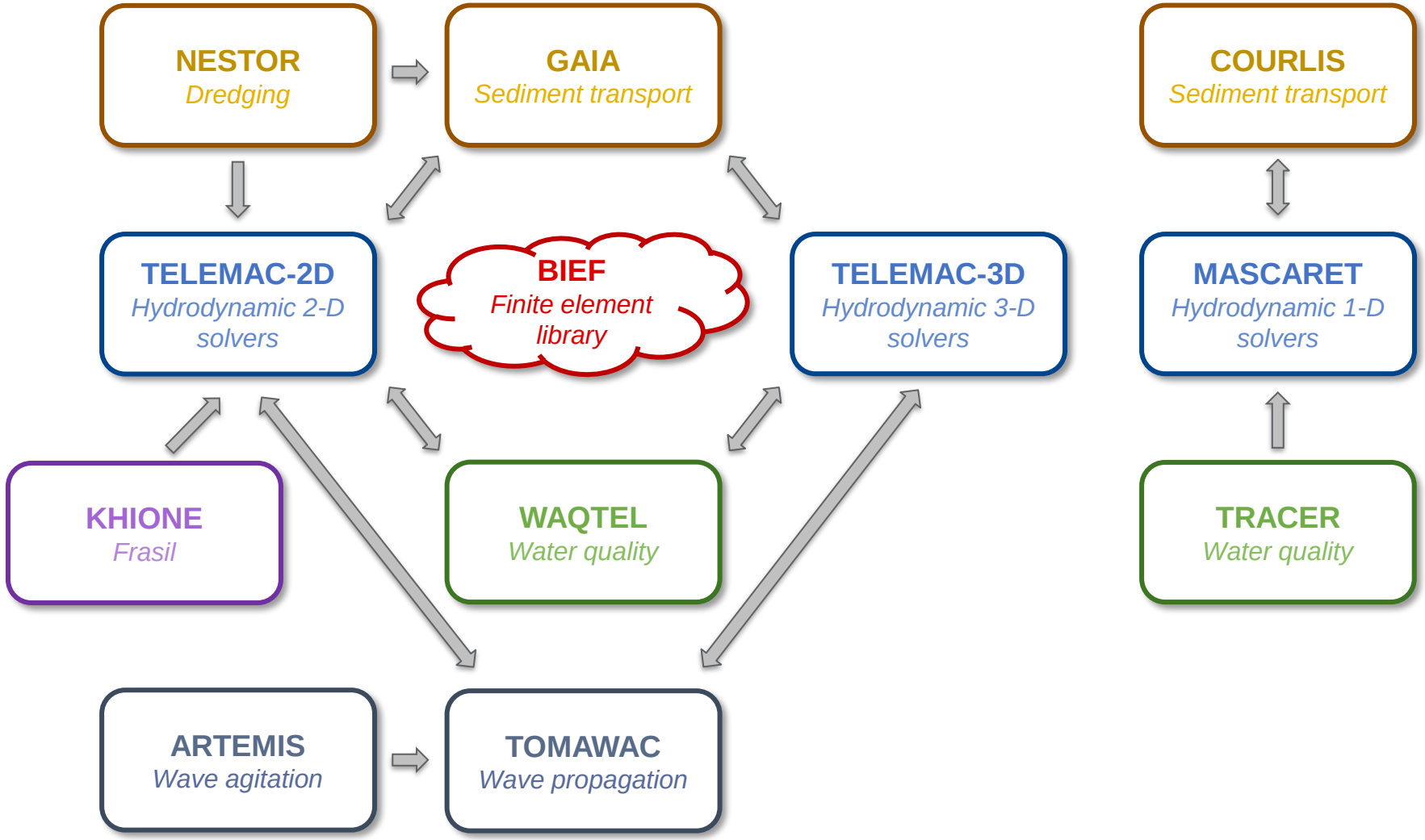








Overview of TELEMAC modules



Managed by an international consortium

- ✓ **EDF**
- ✓ HR Wallingford (UK)
- ✓ BAW (Germany)
- ✓ ARTELIA (France)
- ✓ CEREMA (France)
- ✓ IMDC (Belgium)
- ✓ Daresbury Laboratories (UK)
- ✓ CERFACS (France)
- ✓ Ecole des Ponts ParisTech (France)

executive

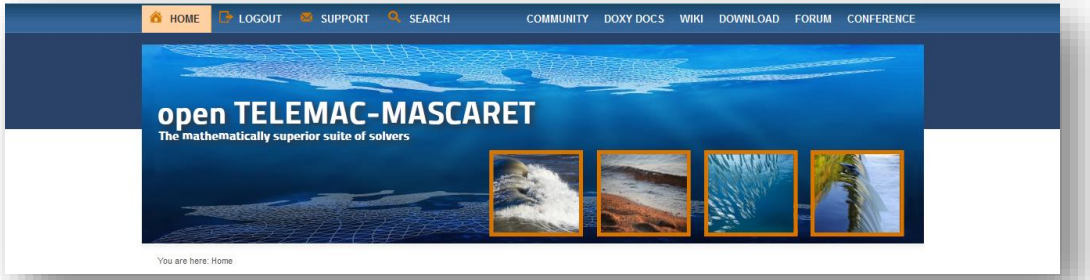
scientific

- ✓ Regular hackathons at EDF improve the code
- ✓ Yearly TELEMAC Users Conference (TUC), usually in October
- ✓ Yearly Scientific Committee (SciCo), around April or May

- ✓ **EDF remains the leader on these codes**

Used by a strong community

- ✓ Official website at www.opentelemac.org
  - ✓ A Wiki: wiki.opentelemac.org/doku.php
  - ✓ A Forum: opentelemac.org/index.php/kunena



- ✓ From 300 commercial licenses in 2009 to almost 2,500 active users in 2019 (12k total)

- ✓ In the last 10 years, additions of 5 modules, 3-way coupling, a Python environment, a Fortran and Python API

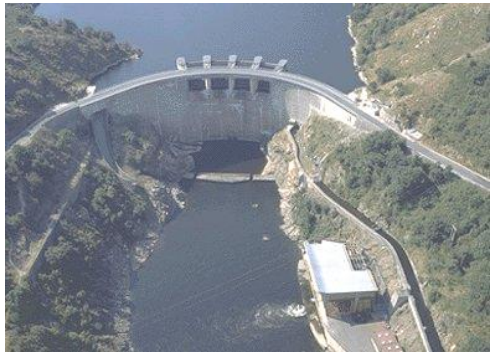
2

The TELEMAC system and its applications

Hydrodynamics with TELEMAC-2D

What is it ?

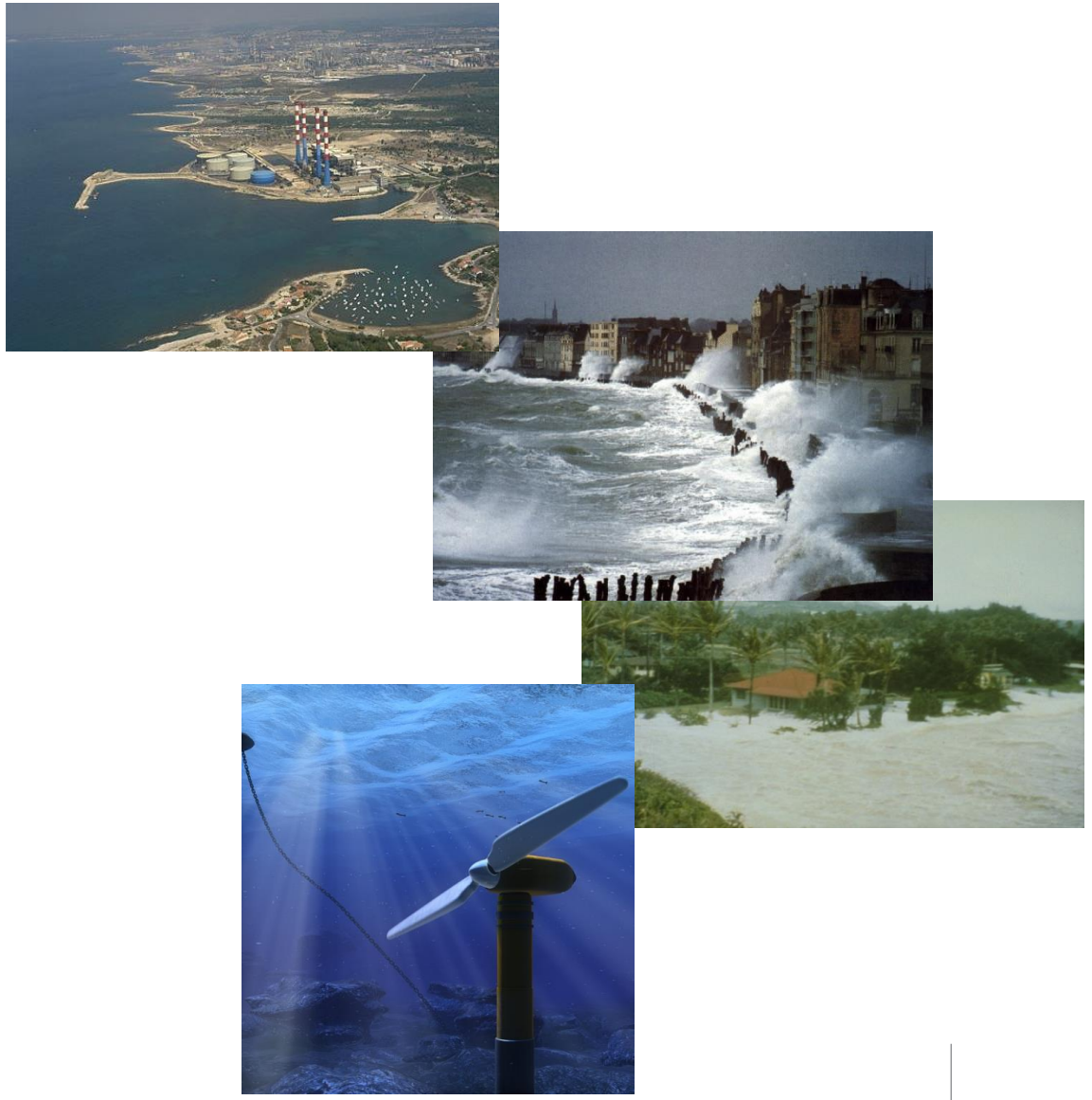
- Shallow water equations (Saint-Venant)
- Boussinesq equations
- Meshes of triangles
- Dry zones
- Turbulence models
- Tracers (temperature, pollutants, etc.)
- Weirs
- Culverts
- Bridges
- Sources and sinks
- Open boundary conditions



TELEMAC-2D: Tidal prediction and storm surge simulation

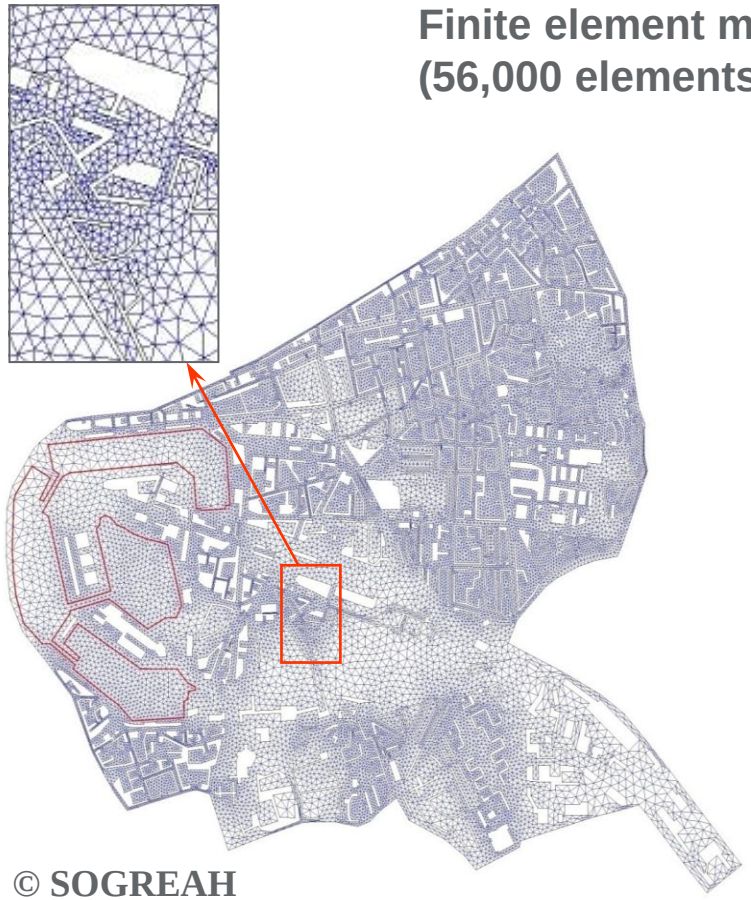
Main objectives

- Navigation safety
- Harbour design and coastal defence
- Flooding in estuaries and coasts
- Designing marine current turbines
- Pollutant advection
- etc.

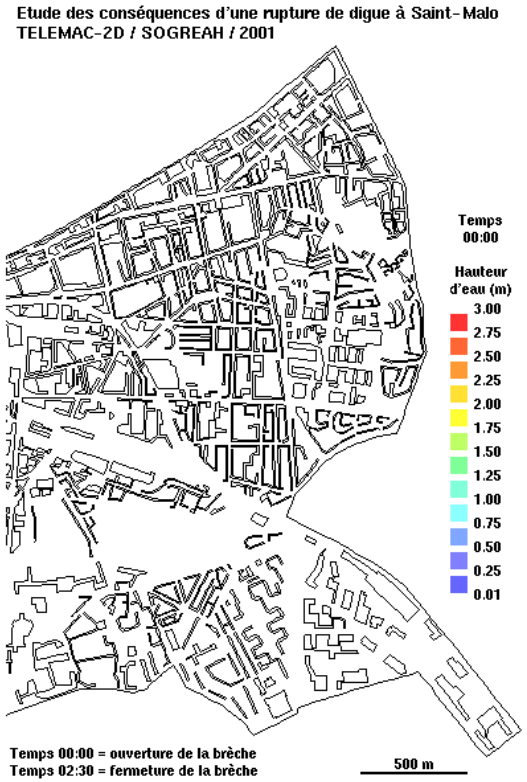


TELEMAC-2D: “Would-be” flooding at Saint-Malo

Finite element mesh
(56,000 elements)

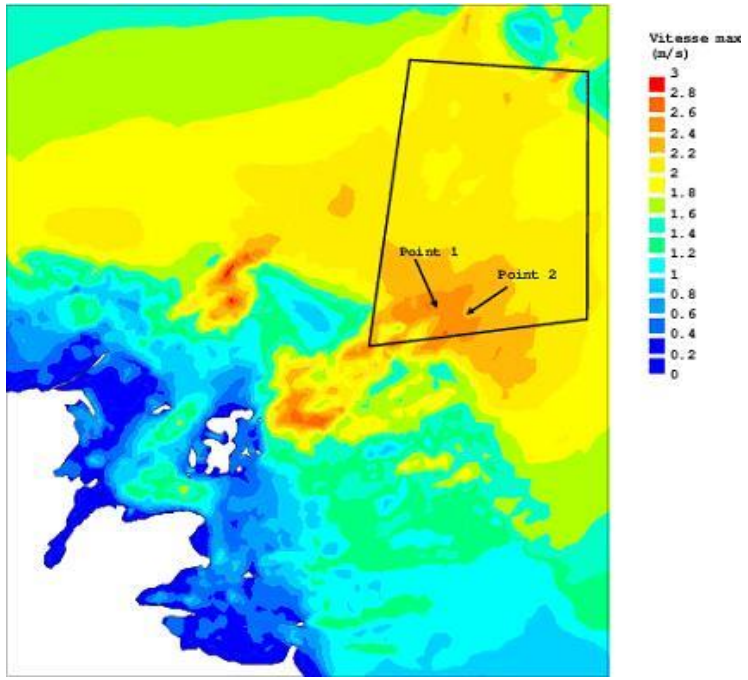


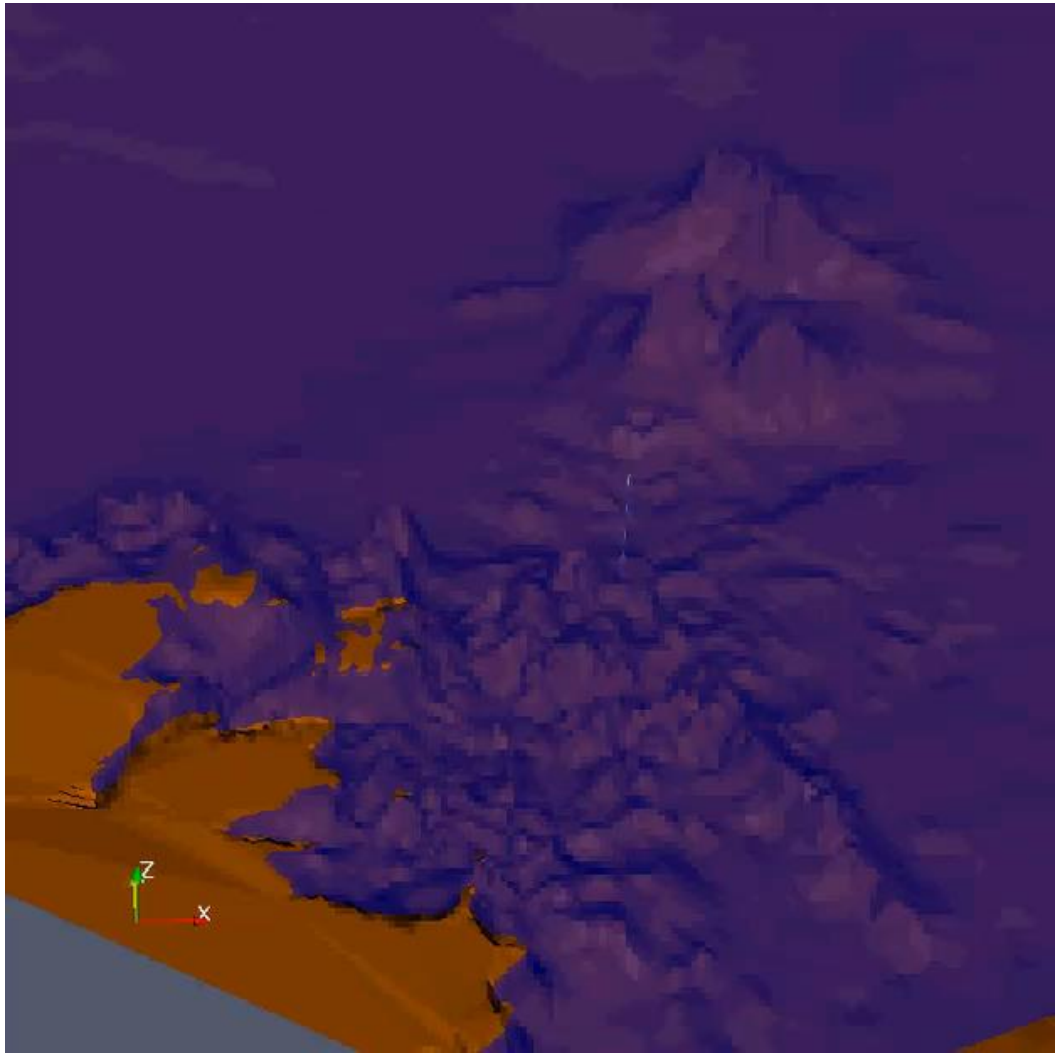
© SOGREAH



TELEMAC-2D: Modelling marine turbines

Real life test in Paimpol-Bréhat

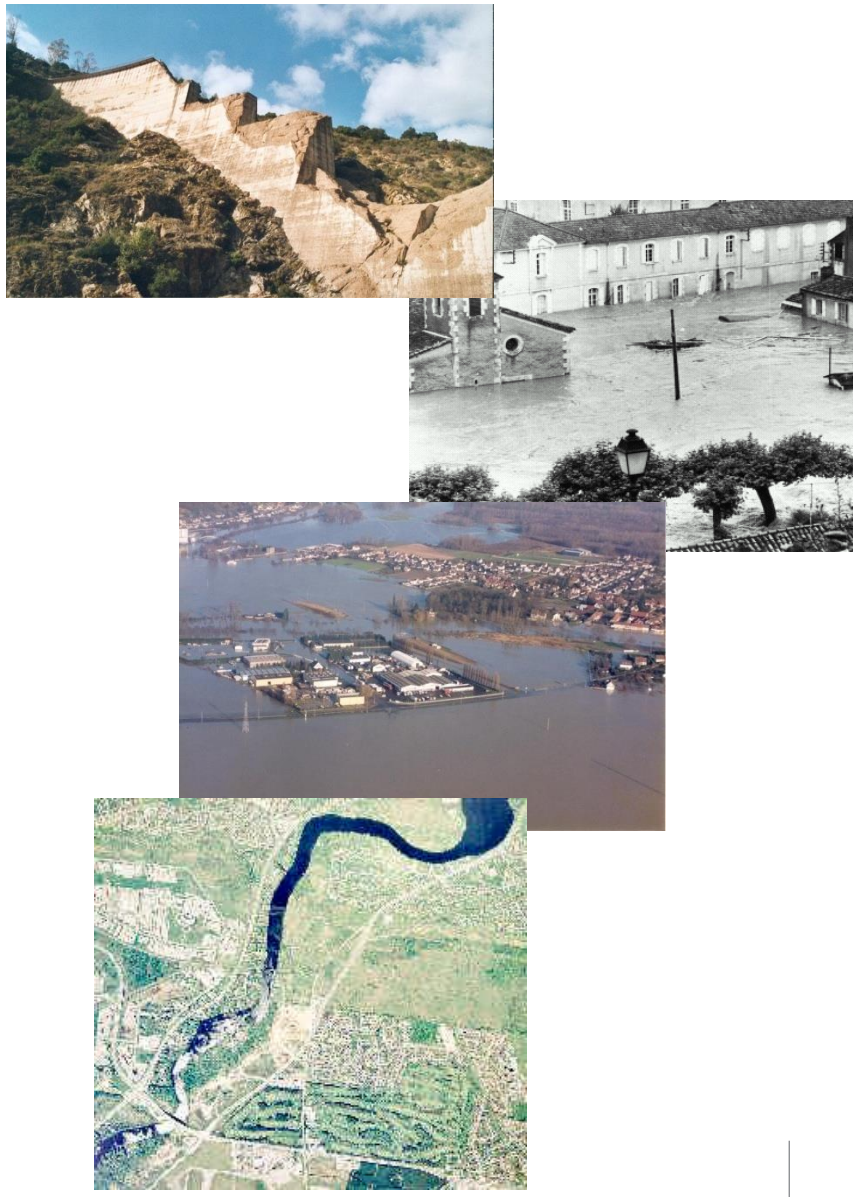




TELEMAC-2D: Dam break and river flood modelling

Main objectives

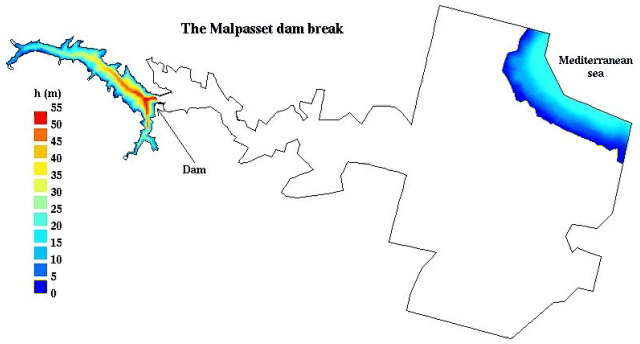
- Conception of dams and river waterworks
- Forecasting for population safety
- Protection of industrial areas
- Damage estimation
- River basin management
- etc.

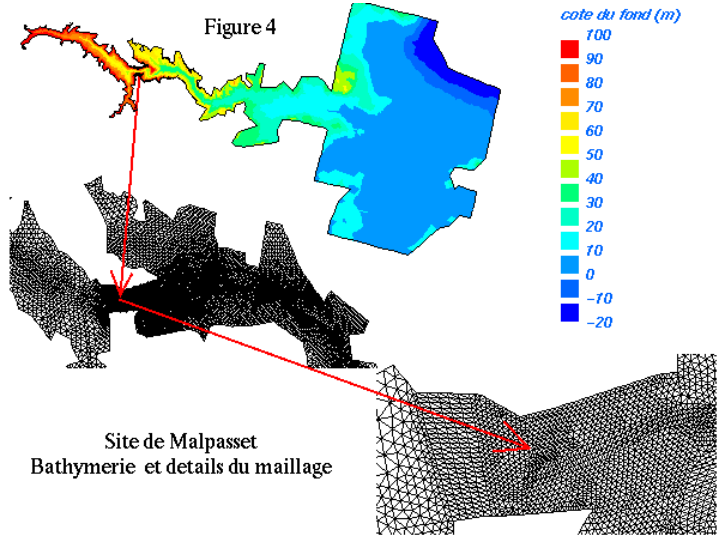


TELEMAC-2D: Malpasset dam break

The Malpasset dam

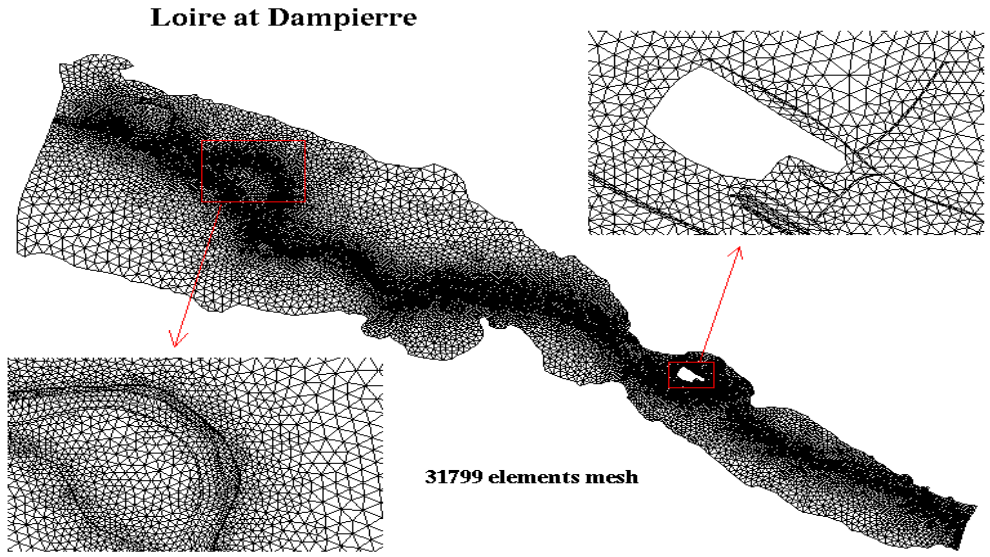
- 48 million m3
- December 2, 1959
- 433 dead

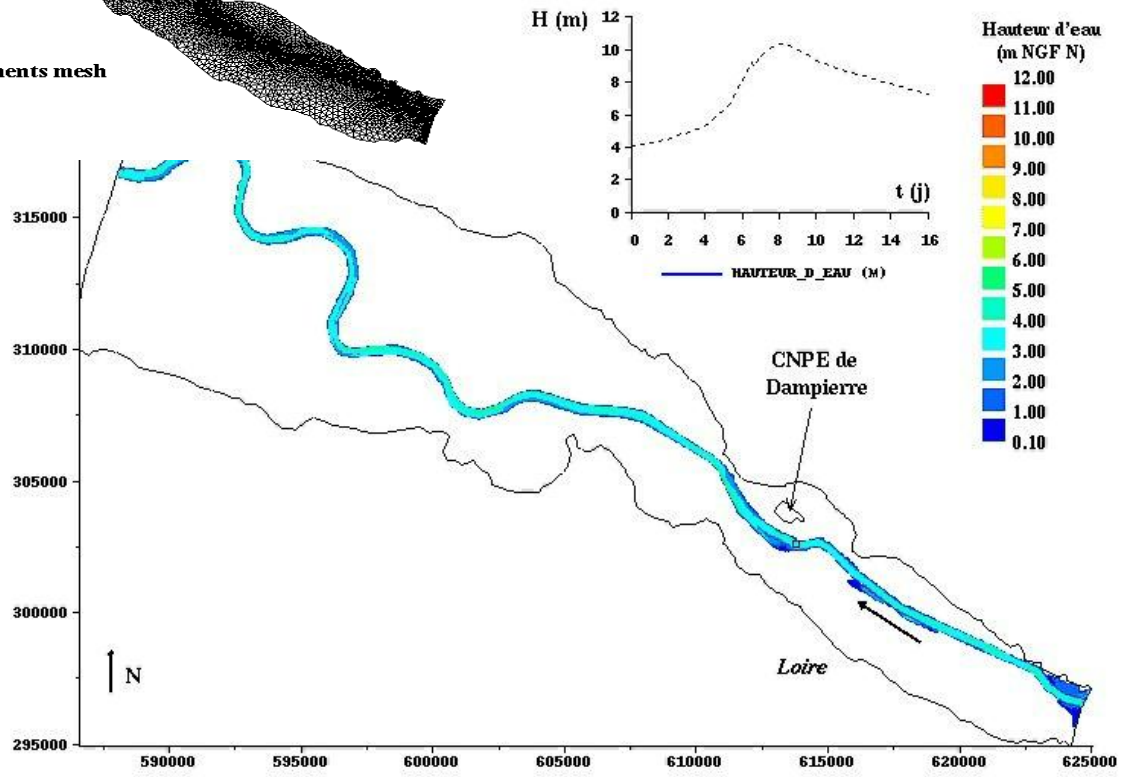




Dam for irrigation, not EDF property

TELEMAC-2D: Flood in river Loire near Dampierre

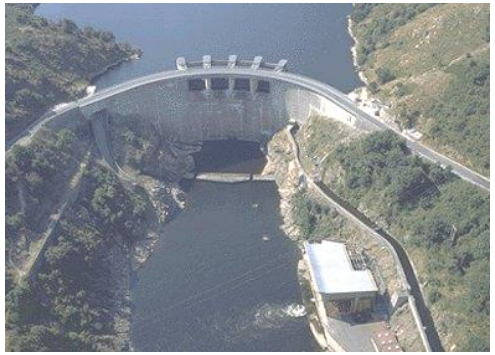


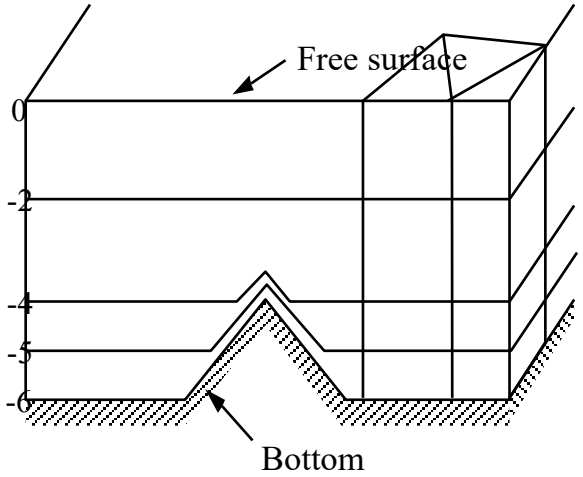


Hydrodynamics with TELEMAC-3D

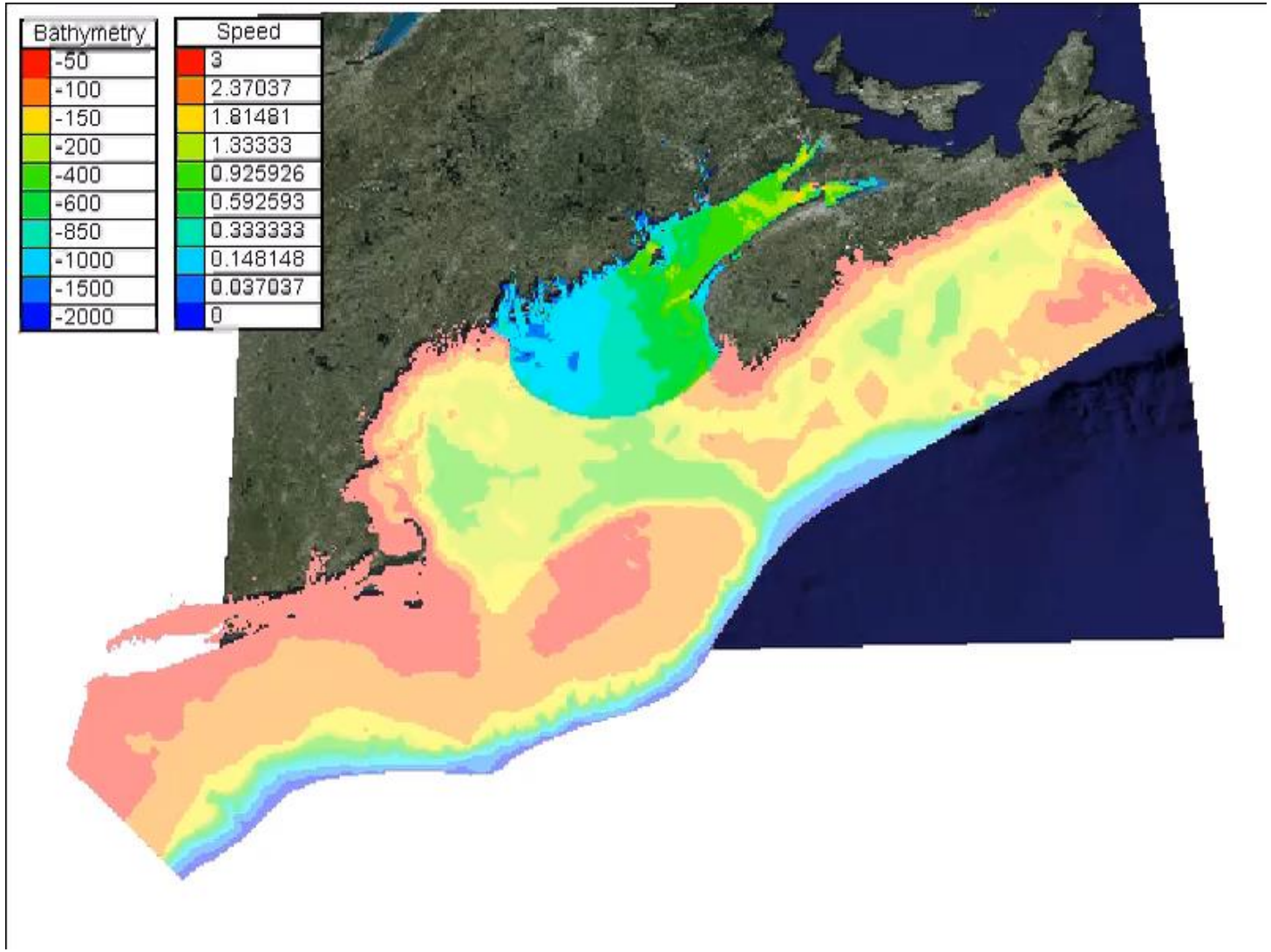
What is it ?

- Navier-Stokes equations
- Meshes of prisms (superimposed 2D meshes)
- Non hydrostatic 3D with free surface
- Dry zones, turbulence models
- Tracers (temperature, pollutants, sediment)

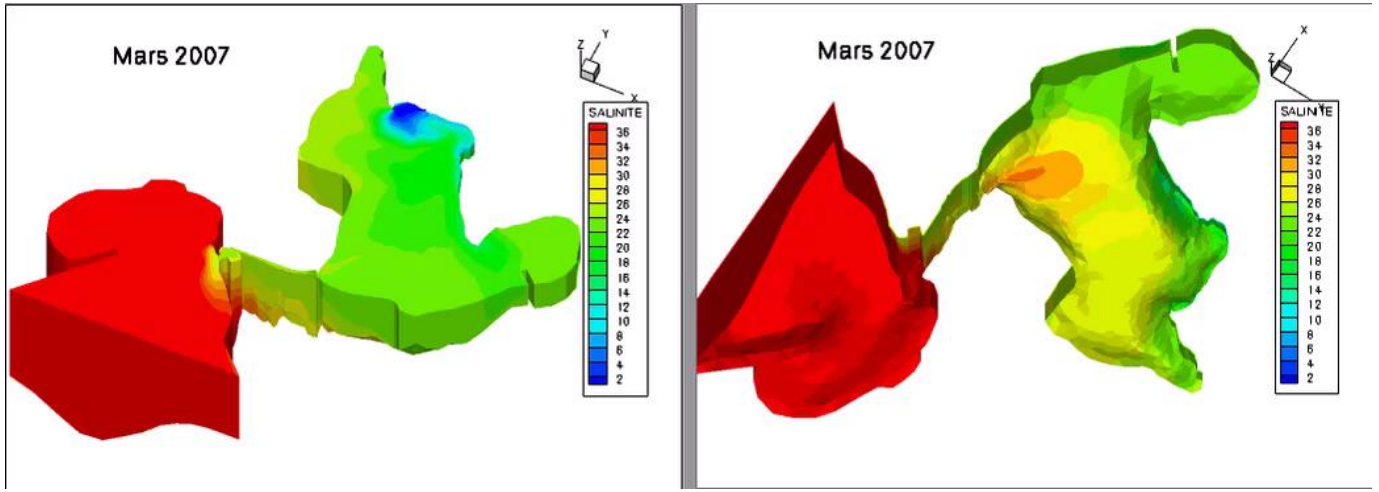




TELEMAC-3D: Tidal currents in Fundy Bay in Canada (CHC)

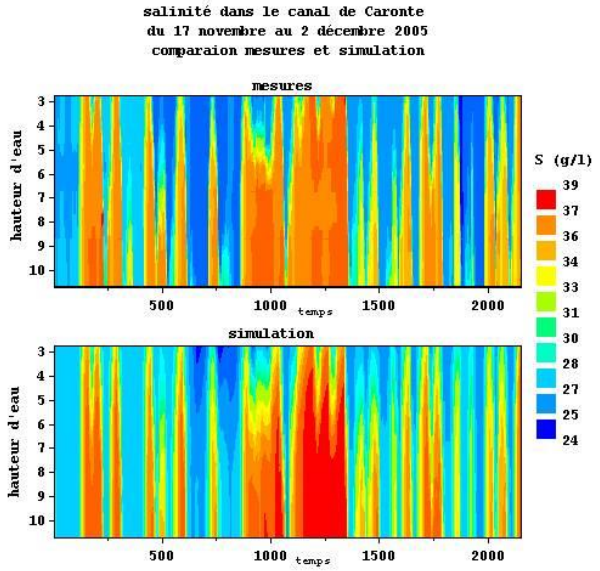


TELEMAC-3D: evolution of salinity in the Berre lagoon



*** Influence of releases of Saint Chamas power plant salinity and discharges in the Caronte canal**

*** Designing scenarii for releases**



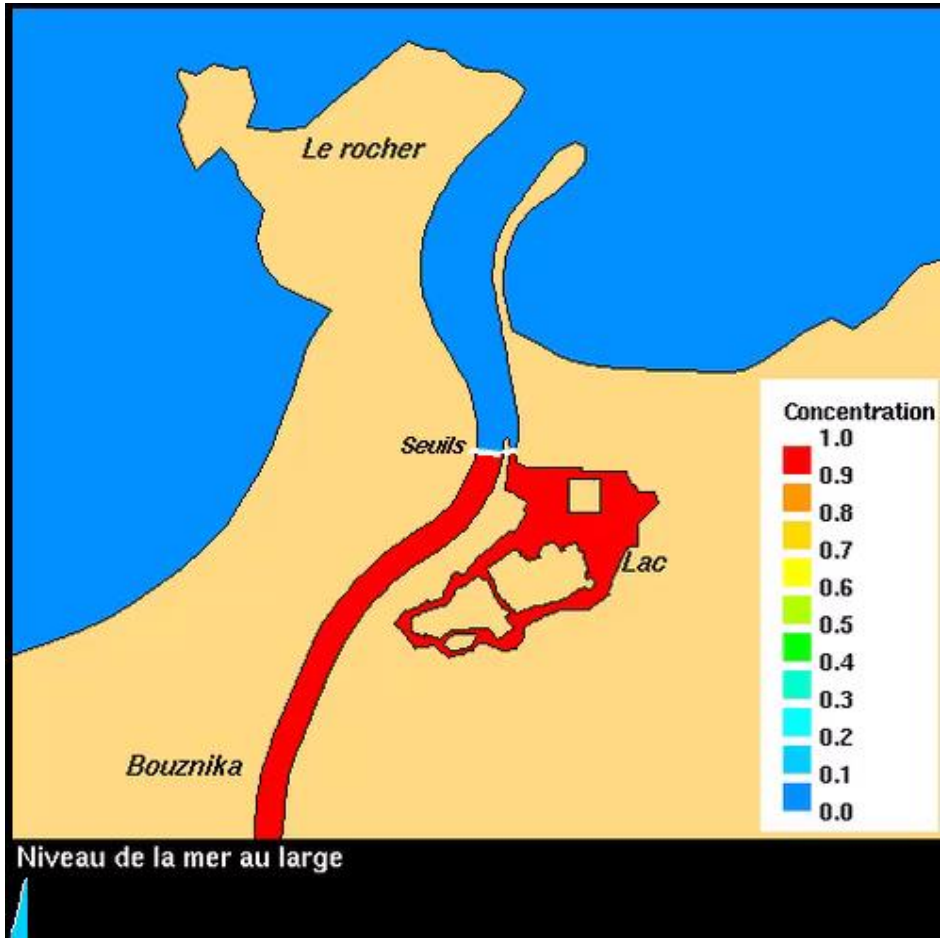
Salinity in Caronte canal between 17 November and 2 December 2005

TELEMAC-3D: pollutant dilution in the Bouznika bay

Goal of the study

To estimate the characteristic time scale (i.e. number of tidal cycles) required for a natural evacuation of a pollutant in a coastal basin.

- **Dirty water**
- **Clean water**

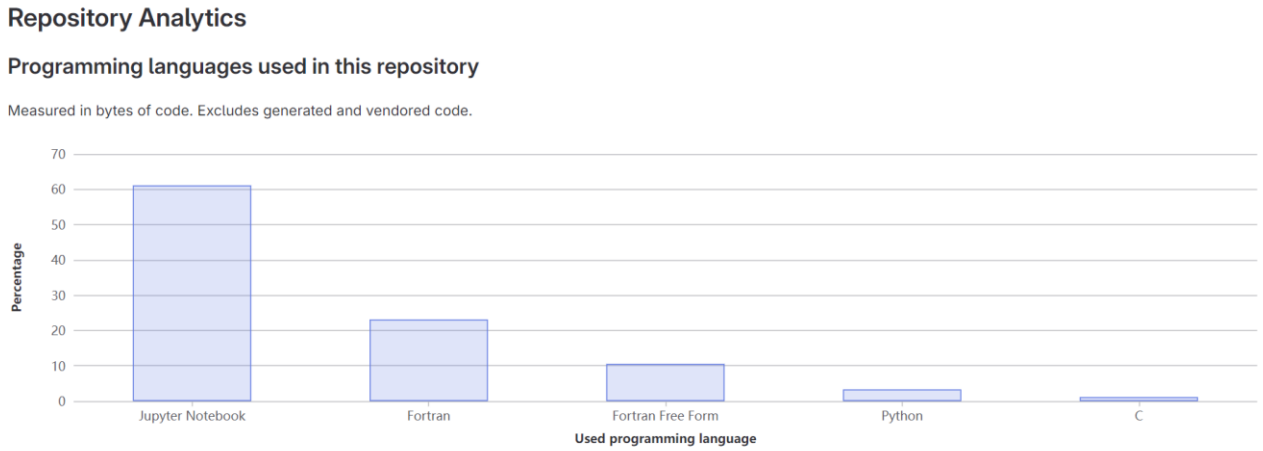


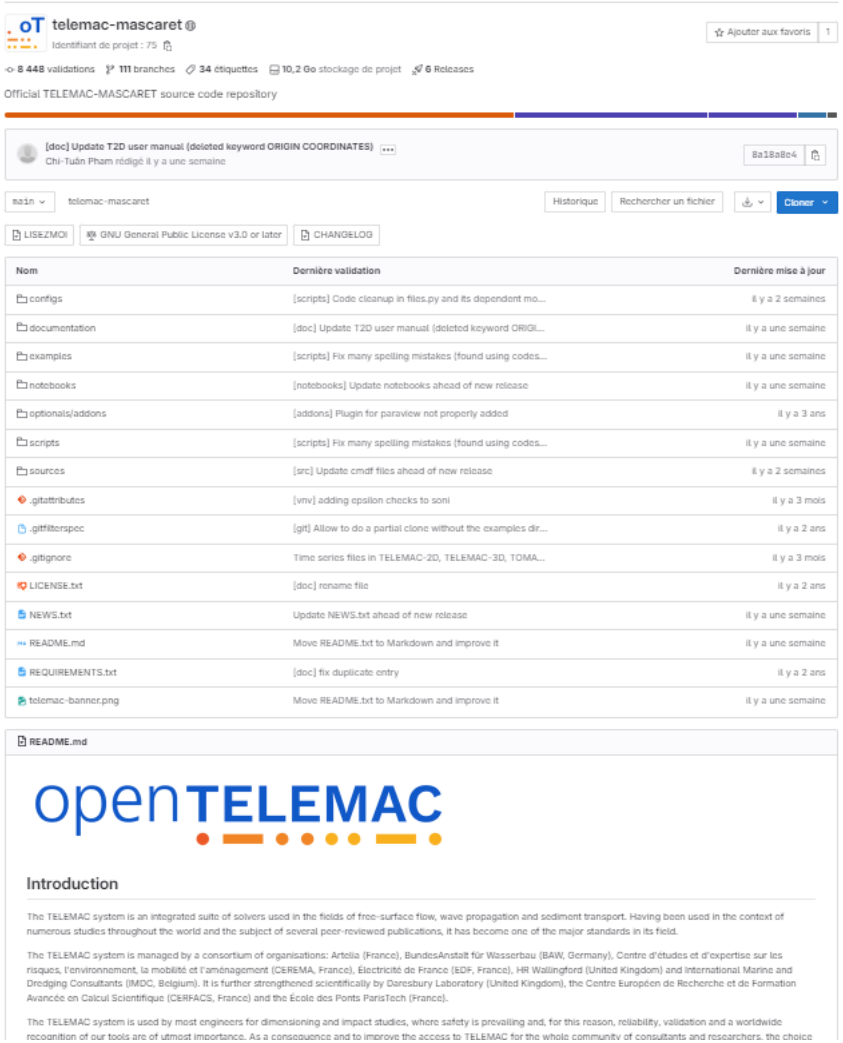
3

Source code management

Git repository

https://gitlab.pam-retd.fr/otm/telemac-mascaret





Commit statistics for main Aug 04 - Dec 04

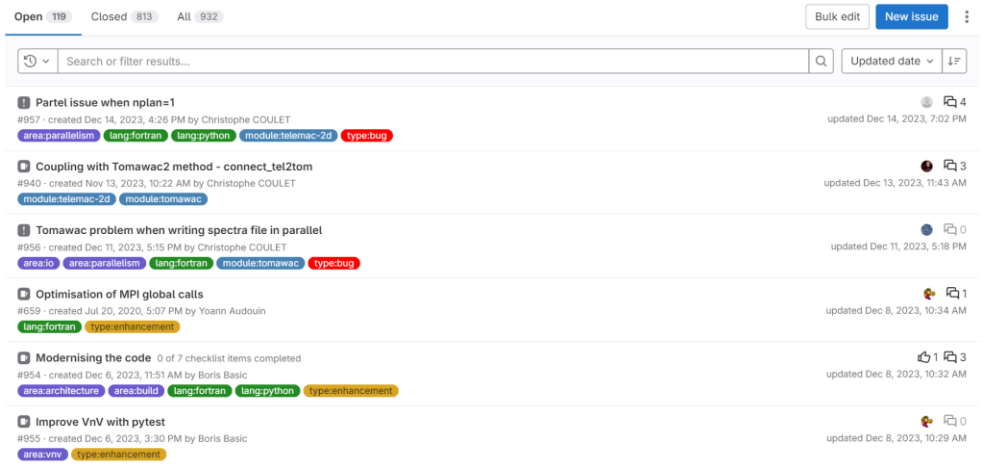
Excluding merge commits. Limited to 2,000 commits.

- Total: **2000 commits**
- Average per day: **1.6 commits**
- Authors: **31**

GitLab: Issue and Merge Requests

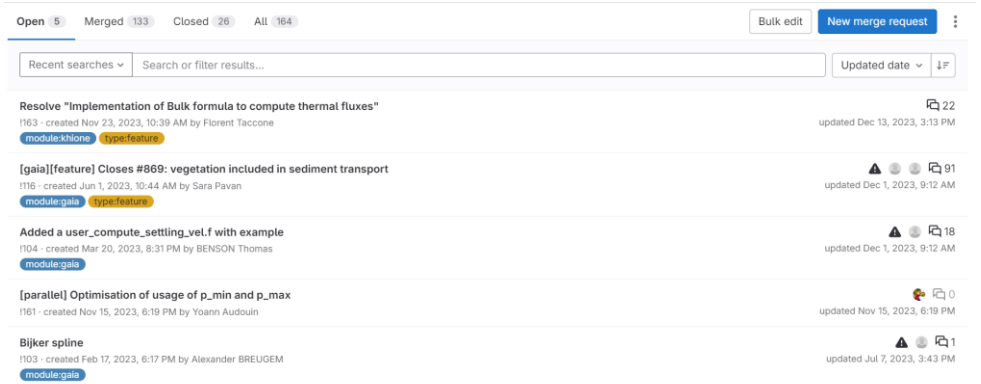
Issue

- Bug report
- Request for a new feature
- Suggestion of an improvement



Merge Request

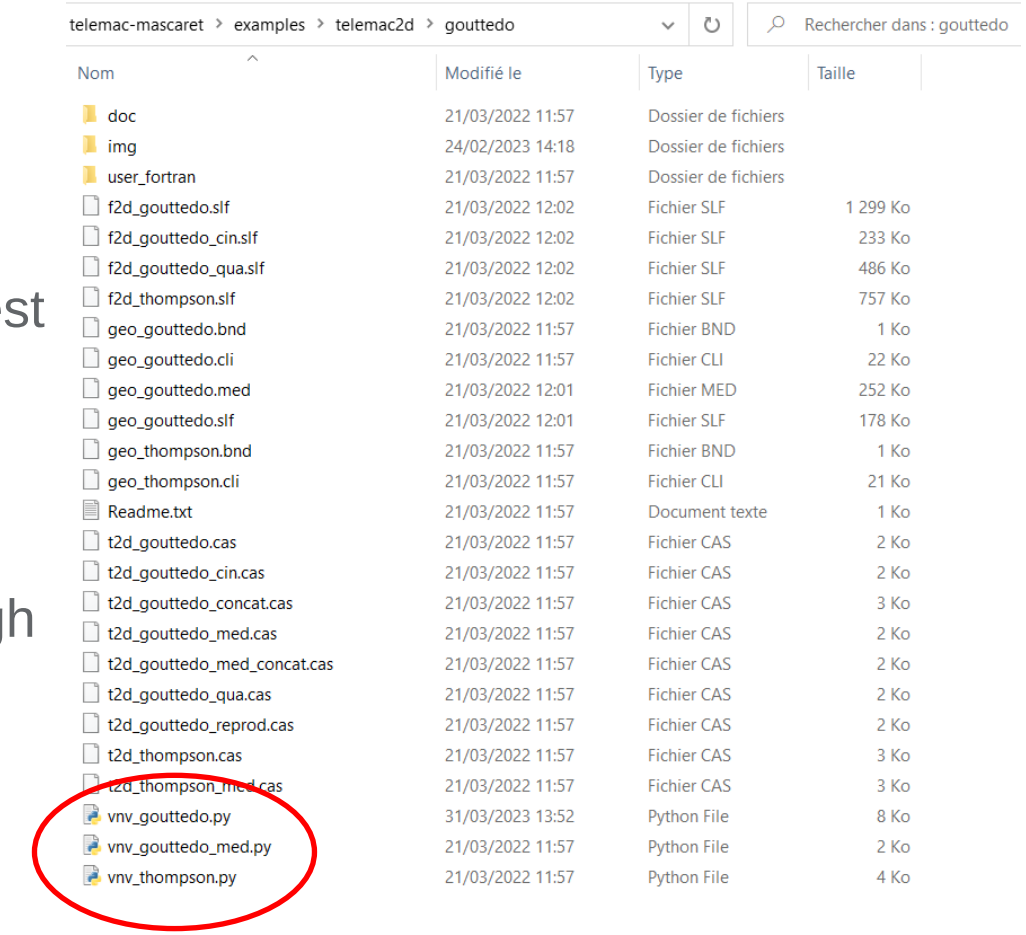
- Request the merge of your changes to the codebase into the main developmenet branch



Validation and Verification (VnV)

A set of acceptance and integration tests

- Available since TELEMAC V8P1 (04/12/2019)
  - Replaced a previous XML-based integration test system
- Python-based
- Provides test cases for all TELEMAC modules
  - All TELEMAC examples are launchable through VnV scripts
- Documented in TELEMAC Developer's Guide
  - Chapter 14: Validation



Validation and Verification: Jenkins

S	M	Nom du projet ↓	Dernier succès	Dernier échec	Dernière durée	Fav
✓	☀	Opentelemac_main_2-run-test_CRONOS_gnu.debug	5 j 21 h #152	19 j #149	10 h	☆
✓	☀	Opentelemac_main_2-run-test_CRONOS_gnu.dyn	18 h #611	27 j #592	4 h 38 mn	☆
✓	☁	Opentelemac_main_2-run-test_CRONOS_intel.debug	11 j #174	13 j #173	16 h	☆
✓	☀	Opentelemac_main_2-run-test_CRONOS_intel.dyn	22 h #657	6 j 21 h #652	3 h 46 mn	☆
✓	☀	Opentelemac_main_2-run-test_CRONOS_nag.debug	5 j 14 h #40	1 mo. 3 j #36	19 h	☆
✓	☀	Opentelemac_main_2-run-test_CRONOS_nag.dyn	14 h #175	4 j 14 h #170	3 h 49 mn	☆
✓	☀	Opentelemac_main_2-run-test_GAIA_gnu.debug	5 j 4 h #191	5 j 21 h #190	9 h 46 mn	☆
✓	☀	Opentelemac_main_2-run-test_GAIA_gnu.dyn	18 h #785	6 j 18 h #780	3 h 41 mn	☆
✓	☀	Opentelemac_main_2-run-test_GAIA_intel.debug	6 j 13 h #195	27 j #191	16 h	☆
✓	☀	Opentelemac_main_2-run-test_GAIA_intel.dyn	21 h #794	20 j #779	3 h 15 mn	☆
✓	☁	Opentelemac_main_2-run-test_GAIA_nag.debug	8 mo. 0 j #4	8 mo. 3 j #3	1 j 0 h	☆
✓	☁	Opentelemac_main_2-run-test_GAIA_nag.dyn	8 mo. 7 j #6	8 mo. 8 j #5	4 h 1 mn	☆
✓	☀	Opentelemac_main_2-run-test_mascaret.win	7 mo. 28 j #644	s. o.	32 mn	☆
✓	☁	Opentelemac_main_2-run-test_S10_gfortran.debug	6 j 20 h #129	10 j #127	4 j 3 h	☆
✓	☀	Opentelemac_main_2-run-test_S10_gfortran.dyn	21 h #525	6 j 21 h #521	7 h 23 mn	☆
✗	☁	Opentelemac_main_2-run-test_S10_nag.debug	1 mo. 4 j #83	10 j #85	14 j	☆


Thank you!

CGG – TERATEC HPC HACKATON2 - ARM

Riemann Siegel Algorithm Tuning

cgg.com

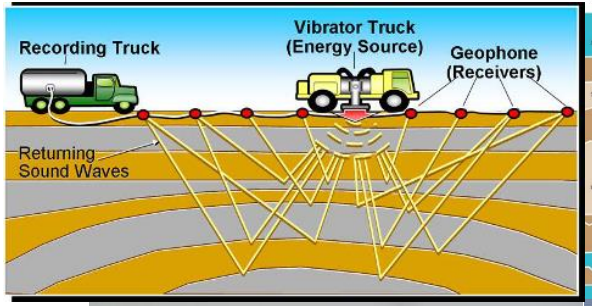
CGG

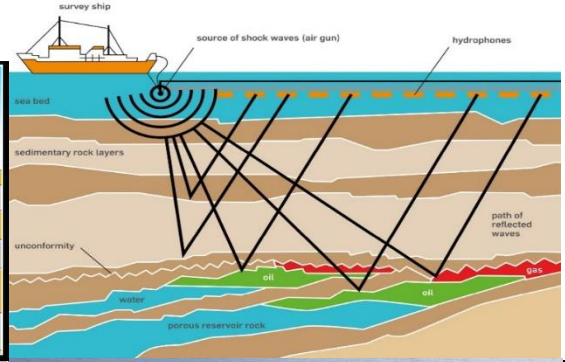
Agenda

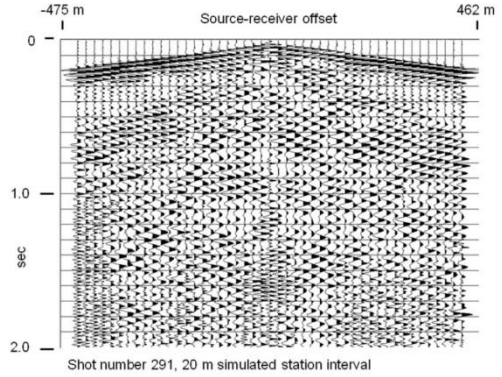
- CGG mission
- Riemann Siegel Conjecture
- The Hackathon problem

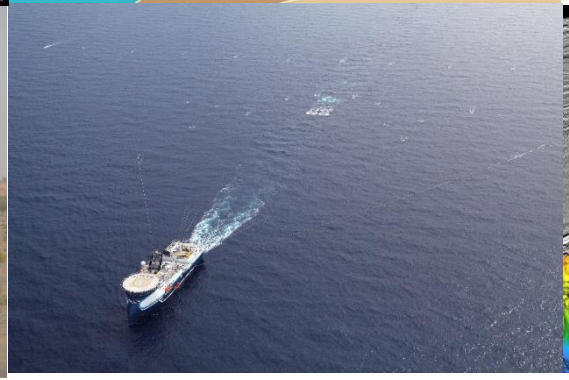
Core business: Geophysics

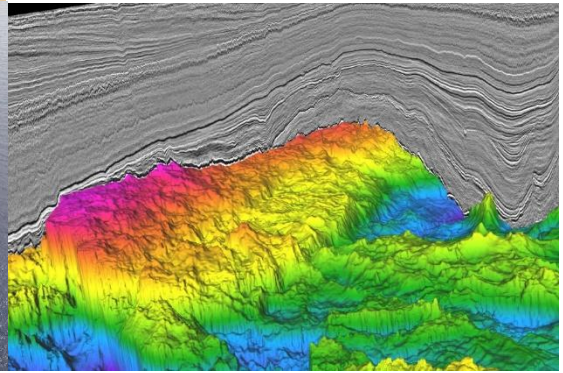
Land and Marine acquisition: seismic surveys







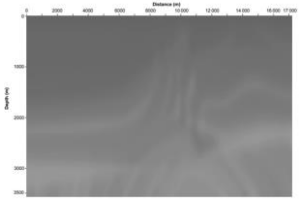




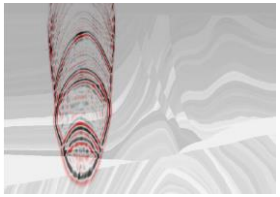
What is seismic imaging ? Why the HPC is so critical

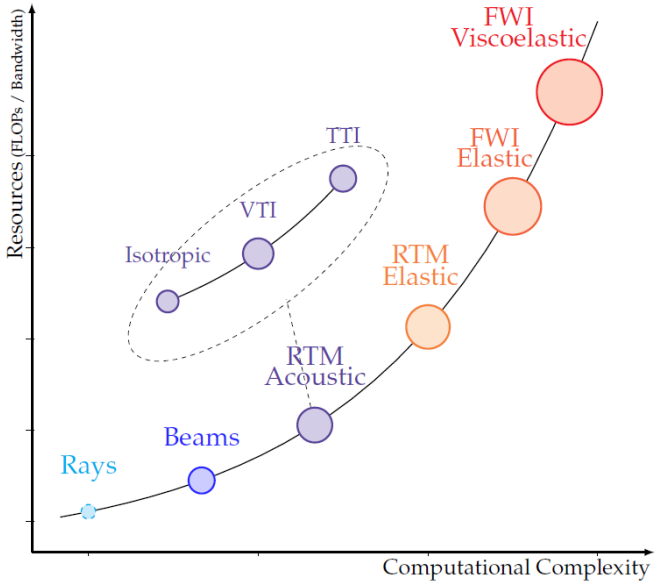
Inversion algorithms RTM,FWI…

Start with initial velocity/density model

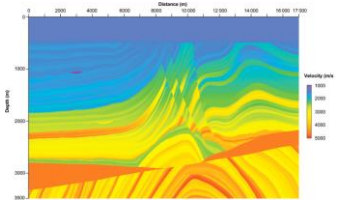


Propagate waves and record data « numerically »



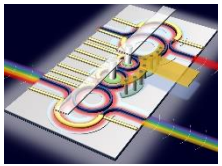


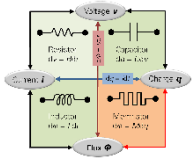
Compute differences between data acquisition (surveys) and numerical data
+ Compute gradient + update velocity/density model



R&D Main Challenges for Extreme IT

- End of cheap transistors
- End of cheap energy + sustainability
- Exponential complexity
  - Compute, Storage, Network, Security, Software,…
- Cambrian Explosion of disruptive technologies
  - Photonic, NVM, CXL, UCIe, …

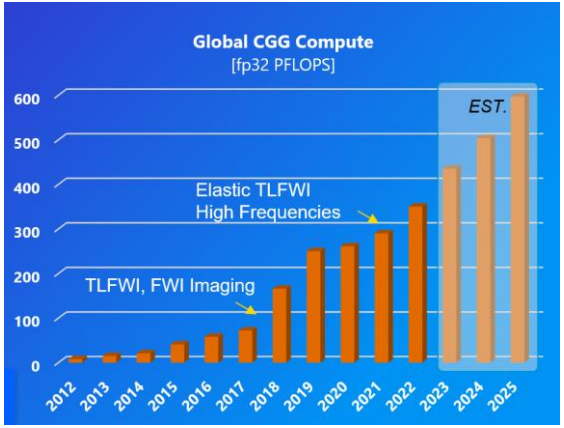


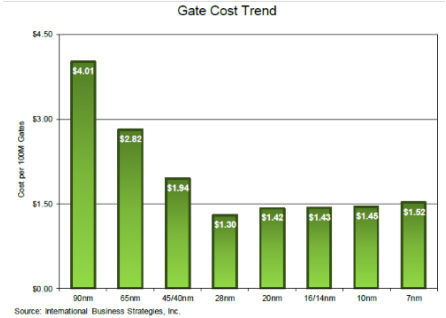


UCIe Universal Chiplet Interconnect Express

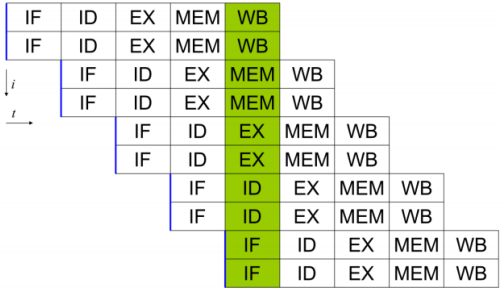
CXL Compute Express Link

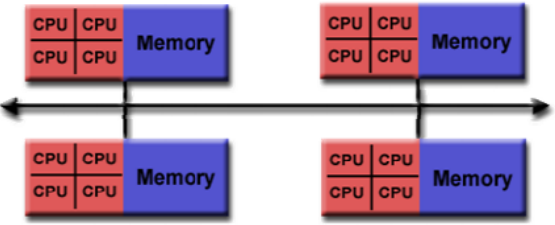
  - INTEL, AMD, NVIDIA, ARM, RISC-V, many others and chiplets variants
  - Pervasive IA, but still in its infancy in HPC
- Extreme Competition “shorter life time of innovation”
- Lack of Experts but always on the edge of innovation





Many Parallelisms to exploit == very hard problem





Software

- Instructions
- Pipeline
- supercalar
- Banks/busses
- Core/thread
- Sockets
- Nodes
- jobs
- IO
- kernel

Hardware

http://tinyurl.com/zntc4cp **Justin Hsia**

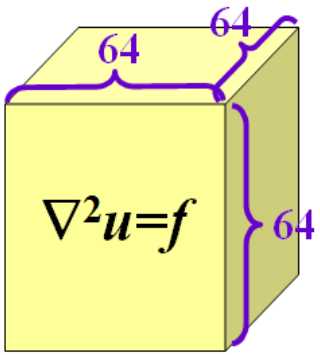
Better algorithms

Algorithmic efficiency is more critical than hardware architecture improvements at extreme scale

The new architecture offers a wide range of opportunities to make breaktrough transformations

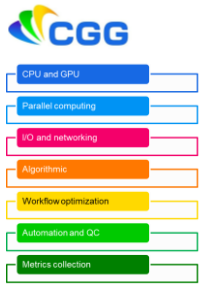
- Exemple : Poisson's equation on a cube of size $N=n^3$

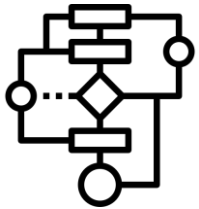
Year	*Method*	*Reference*	*Storage*	*Flops*
1947	GE (banded)	Von Neumann & Goldstine	n^5	n^7
1950	Optimal SOR	Young	n^3	$n^4 \log n$
1971	CG	Reid	n^3	$n^{3.5} \log n$
1984	Full MG	Brandt	n^3	n^3



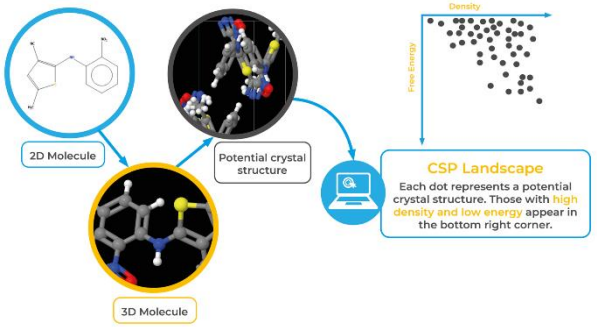
www.siam.org/about/science/**keyes**.ppt

New business: CGG now helps others customers to make breakthrough on extreme compute intensive challenges





40x * 50x = from months to hours



Extreme-scale AI and HPC computations require achieving significant leaps in performance, which can only be realized through the collaboration of experts from various fields.

Our partnership with CGG has proven to be a game-changer, thanks to the substantial enhancements we've made to our codes. This marks just the initial stages of our exciting journey.

Peter Doyle : CEO and cofounder of biosimulytics

The Riemann Conjecture

The Riemann Conjecture

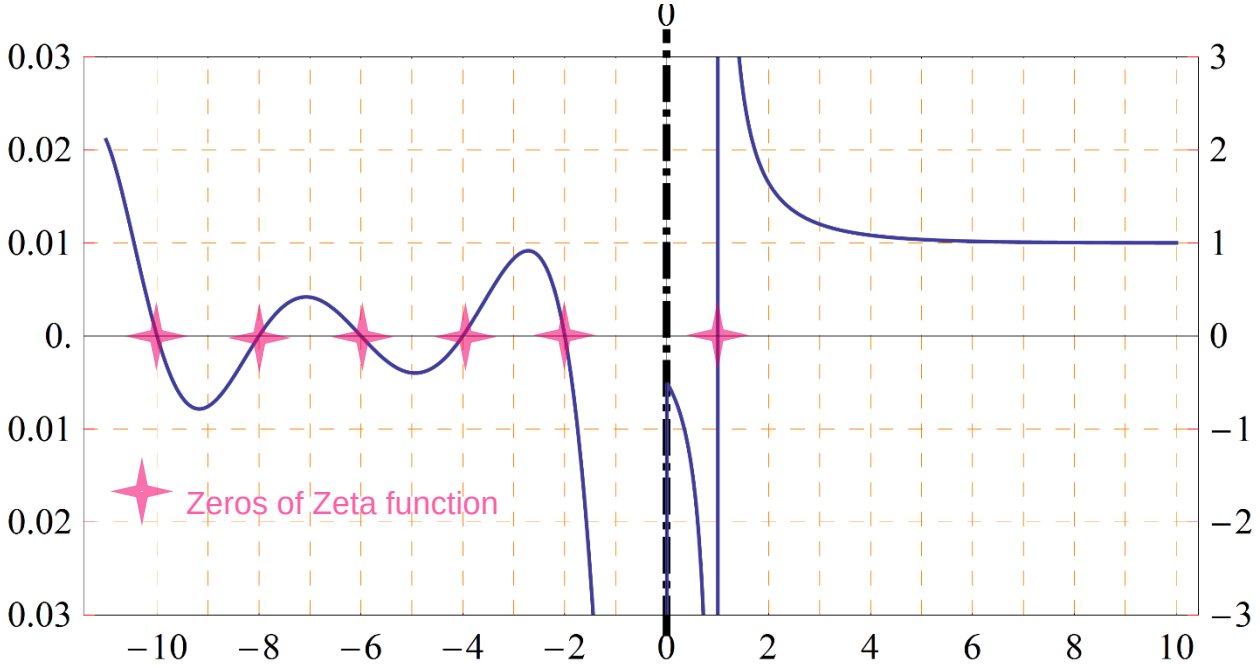
- In mathematics, the Riemann hypothesis is a conjecture formulated in 1859 by the German mathematician Bernhard Riemann. It states that the non-trivial zeros of the Riemann zeta function all have a real part of 1/2.
- Its proof would enhance the understanding of the distribution of prime numbers.
- This conjecture stands as one of the most significant unresolved problems in mathematics in the early 21st century: it is one of the twenty-three famous Hilbert problems proposed in 1900, one of the seven Millennium Prize Problems, and one of Smale's eighteen problems.
- Like the other six Millennium Prize Problems, the precise statement of the conjecture to be proven is accompanied by a detailed description providing numerous insights into the problem's history, significance, and the current state of research on it.

Bernhard Riemann 1826-1866

(Riemann — Wikipédia)

The Zeta Function in N

$$\zeta(n) = \sum_{k=1}^{\infty} \frac{1}{k^n} = 1 + \frac{1}{2^n} + \frac{1}{3^n} + \frac{1}{4^n} + \ldots$$



Zeta_function (wikimedia.org)

Special values:

$$\zeta(2) = \frac{\pi^2}{6}$$ Euler

zeta(2) : Wolfram MathWorld

$$\zeta(3) = 1.202056903\ldots$$

Apéry's constant - Wikipedia

$$\zeta(4) = 1 + \frac{1}{2^4} + \frac{1}{3^4} + \cdots = \frac{\pi^4}{90}$$

$$\zeta(6) = 1 + \frac{1}{2^6} + \frac{1}{3^6} + \cdots = \frac{\pi^6}{945}$$

$$\zeta(8) = 1 + \frac{1}{2^8} + \frac{1}{3^8} + \cdots = \frac{\pi^8}{9450}$$

$$\zeta(10) = 1 + \frac{1}{2^{10}} + \frac{1}{3^{10}} + \cdots = \frac{\pi^{10}}{93555}$$

$$\zeta(12) = 1 + \frac{1}{2^{12}} + \frac{1}{3^{12}} + \cdots = \frac{691\pi^{12}}{638512875}$$

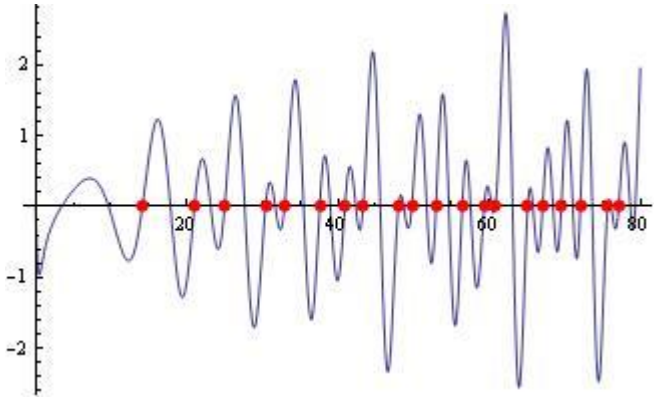
$$\zeta(14) = 1 + \frac{1}{2^{14}} + \frac{1}{3^{14}} + \cdots = \frac{2\pi^{14}}{18243225}.$$

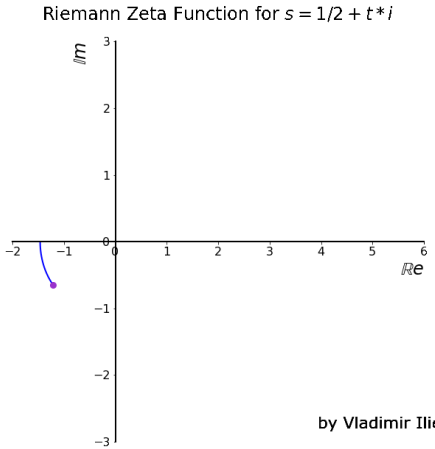
https://tinyurl.com/y6c48n3a

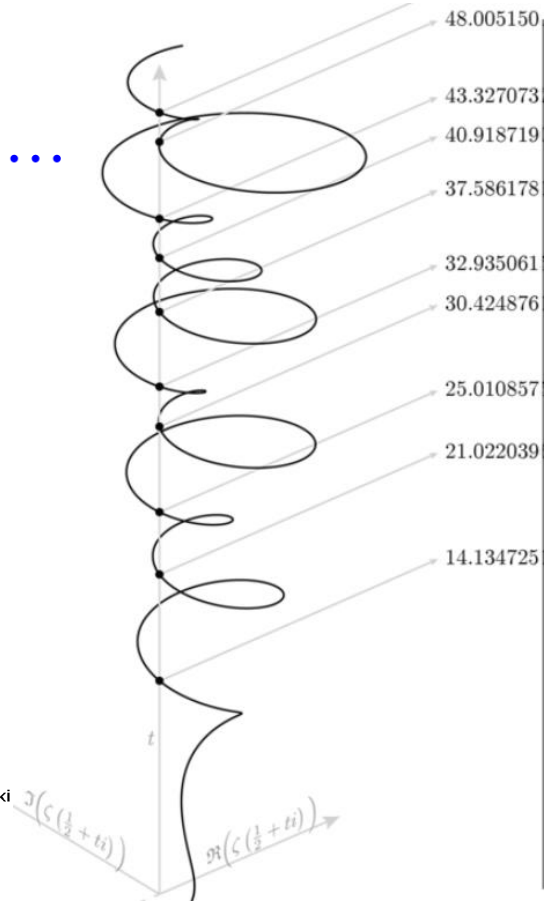
Riemann Extend the Zeta Function to the Complex Plane

$$\zeta(s) = \sum_{k=1}^{\infty} \frac{1}{k^s} = 1 + \frac{1}{2^s} + \frac{1}{3^s} + \frac{1}{4^s} + \dots$$

Riemann hypothesis that all non trivial zeros lay on the critical line

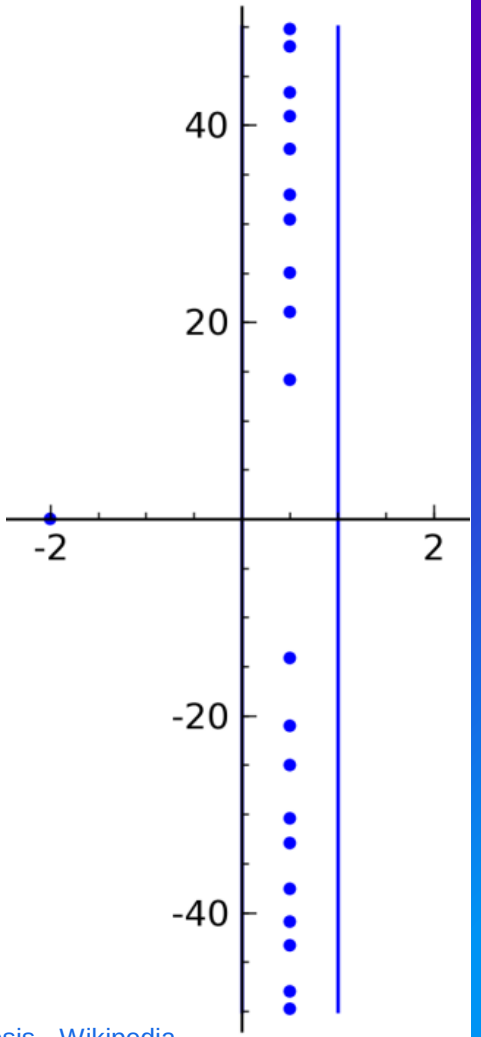






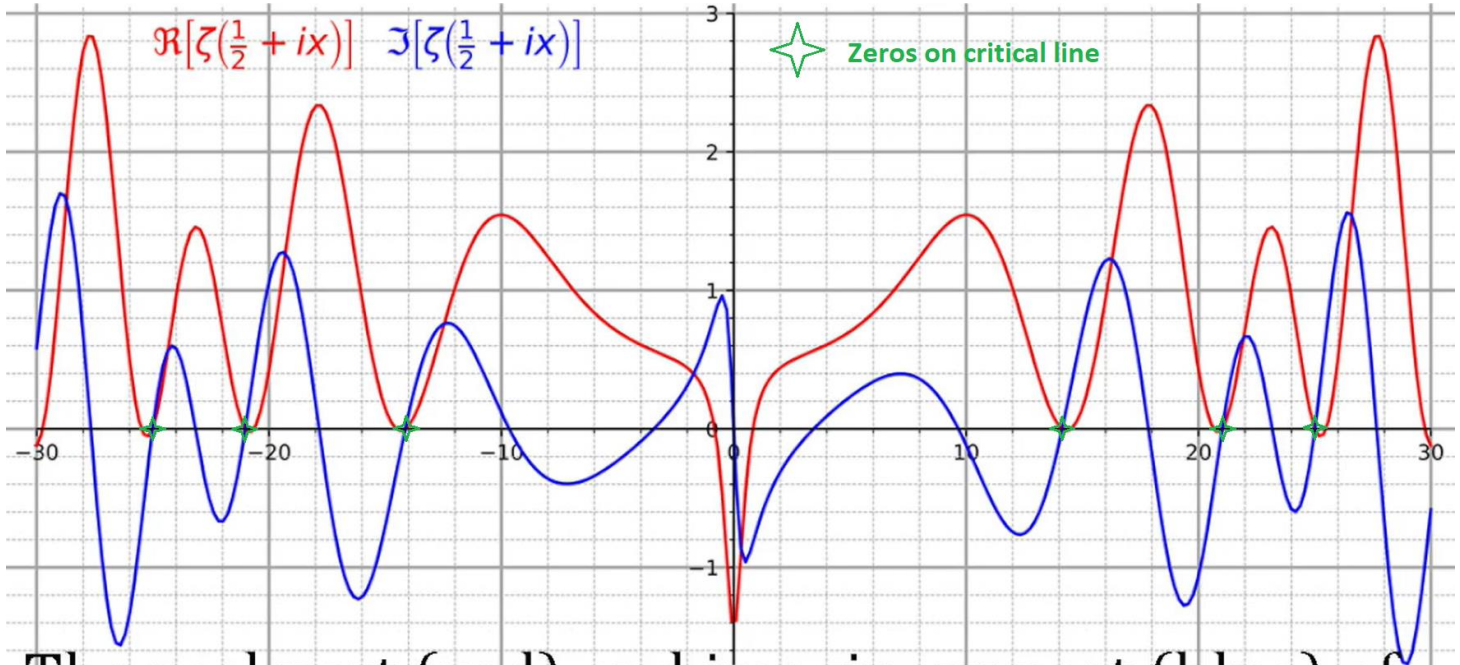
The Riemann Hypothesis

- the **Riemann hypothesis** is the conjecture that the Riemann zeta function has its zeros only at the negative even integers and complex numbers with real part 1/2.
- Riemann's original motivation for studying the zeta function and its zeros was their occurrence in his explicit formula for the number of primes $\pi(x)$ less than or equal to a given number x, which he published in his 1859 paper "On the Number of Primes Less Than a Given Magnitude".
- Riemann developed a very efficient unpublished method to compute “manually” the 3 first zeros; reported by Siegel in 1932; now called the Riemann–Siegel formula
- The competition to calculate the zeros, discover a counterexample, or provide a proof for the conjecture has begun.



Riemann hypothesis - Wikipedia

The Zeta Function on the critical line

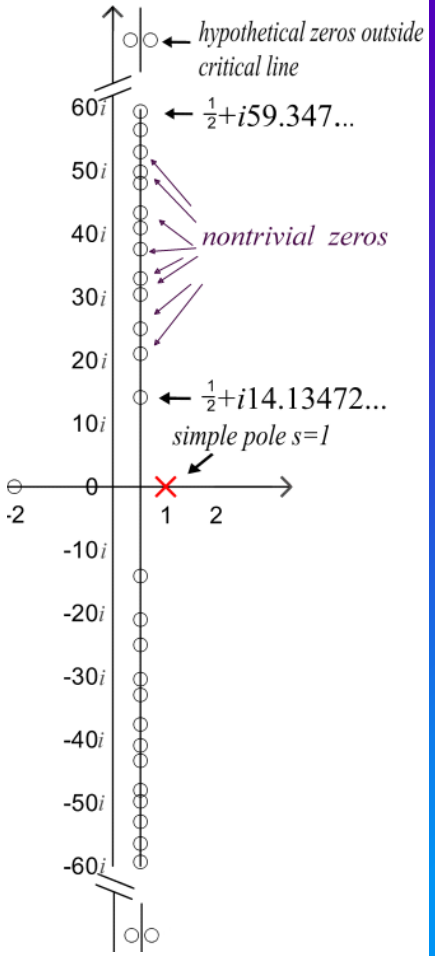


The real part (red) and imaginary part (blue) of the Riemann zeta function along the critical line.

Riemann hypothesis

The Race to compute lot of zeros

Year	Number of zeros	Author
1859?	3	B. Riemann used the Riemann–Siegel formula (unpublished, but reported in Siegel 1932).
1903	15	J. P. Gram (1903) used Euler–Maclaurin formula and discovered Gram's law. He showed that all 10 zeros with imaginary part at most 50 range lie on the critical line with real part 1/2 by computing the sum of the inverse 10th powers of the roots he found.
1914	79 ($\gamma_n \leq 200$)	R. J. Backlund (1914) introduced a better method of checking all the zeros up to that point are on the line, by studying the argument $S(T)$ of the zeta function.
1925	138 ($\gamma_n \leq 300$)	J. I. Hutchinson (1925) found the first failure of Gram's law, at the Gram point g_{126}.
1935	195	E. C. Titchmarsh (1935) used the recently rediscovered Riemann–Siegel formula, which is much faster than Euler–Maclaurin summation. It takes about $O(T^{3/2+\varepsilon})$ steps to check zeros with imaginary part less than T, while the Euler–Maclaurin method takes about $O(T^{2+\varepsilon})$ steps.
1936	1041	E. C. Titchmarsh (1936) and L. J. Comrie were the last to find zeros by hand.
1953	1104	A. M. Turing (1953) found a more efficient way to check that all zeros up to some point are accounted for by the zeros on the line, by checking that Z has the correct sign at several consecutive Gram points and using the fact that $S(T)$ has average value 0. This requires almost no extra work because the sign of Z at Gram points is already known from finding the zeros, and is still the usual method used. This was the first use of a digital computer to calculate the zeros.
1956	15 000	D. H. Lehmer (1956) discovered a few cases where the zeta function has zeros that are "only just" on the line: two zeros of the zeta function are so close together that it is unusually difficult to find a sign change between them. This is called "Lehmer's phenomenon", and first occurs at the zeros with imaginary parts 7005.063 and 7005.101, which differ by only .04 while the average gap between other zeros near this point is about 1.
1956	25 000	D. H. Lehmer



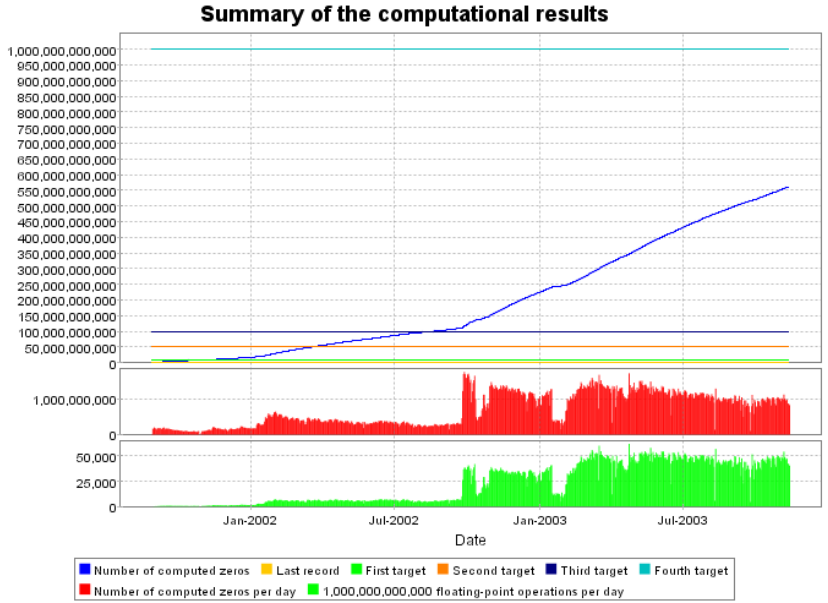
Riemann hypothesis

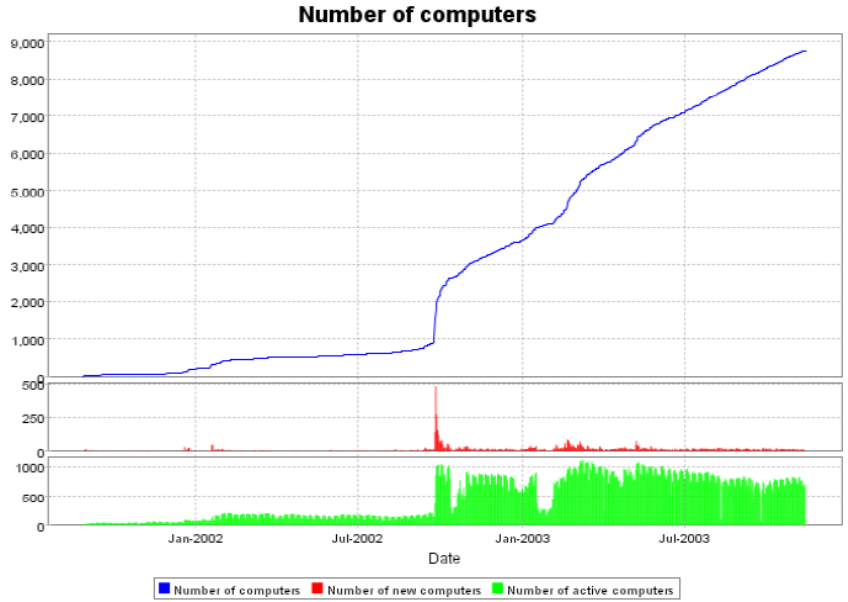
The Race in modern time “big computers”

1958	35 337	N. A. Meller
1966	250 000	R. S. Lehman
1968	3 500 000	Rosser, Yohe & Schoenfeld (1969) stated Rosser's rule (described below).
1977	40 000 000	R. P. Brent
1979	81 000 001	R. P. Brent
1982	200 000 001	R. P. Brent, J. van de Lune, H. J. J. te Riele, D. T. Winter
1983	300 000 001	J. van de Lune, H. J. J. te Riele
1986	1 500 000 001	van de Lune, te Riele & Winter (1986) gave some statistical data about the zeros and give several graphs of Z at places where it has unusual behavior.
1987	A few of large (~10^{12}) height	A. M. Odlyzko (1987) computed smaller numbers of zeros of much larger height, around 10^{12}, to high precision to check Montgomery's pair correlation conjecture.
1992	A few of large (~10^{20}) height	A. M. Odlyzko (1992) computed a 175 million zeros of heights around 10^{20} and a few more of heights around 2×10^{20}, and gave an extensive discussion of the results.
1998	10000 of large (~10^{21}) height	A. M. Odlyzko (1998) computed some zeros of height about 10^{21}
2001	10 000 000 000	J. van de Lune (unpublished)
2004	~900 000 000 000[30]	S. Wedeniwski (ZetaGrid distributed computing)
2004	10 000 000 000 000 and a few of large (up to ~10^{24}) heights	X. Gourdon (2004) and Patrick Demichel used the Odlyzko–Schönhage algorithm. They also checked two billion zeros around heights 10^{13}, 10^{14}, ..., 10^{24}.
2020	12 363 153 437 138 up to height 3 000 175 332 800	Platt & Trudgian (2021). They also verified the work of Gourdon (2004) and others.

Riemann hypothesis

The Zetagrid project





- Highly tuned versions in assembly for multiple architectures
- The project reached on 11000 computers 935.7 billion nontrivial zeros of the Riemann zeta function in 1146 days.

[Math Games: Ten Trillion Zeta Zeros (mathpuzzle.com)]

THE RIEMANN-SIEGEL FORMULA AND LARGE SCALE COMPUTATIONS OF THE RIEMANN ZETA FUNCTION 1/2

The Riemann-Siegel formula for computing $\zeta(1/2+it)$, $t \in \mathbb{R}_+$

Set $N = \left\lfloor (t/2\pi)^{1/2} \right\rfloor$ (the integer part of $(t/2\pi)^{1/2}$), $p = (t/2\pi)^{1/2} - N$. Then

$$Z(t) = 2\sum_{n=1}^{N} n^{-1/2} \cos\left[\vartheta(t) - t\log n\right] + R$$

where

$$\vartheta(t) = \Im\left[\log\Pi\left(\frac{it}{2} - \frac{3}{4}\right)\right] - \frac{t}{2}\log\pi$$

and

$$R \approx (-1)^{N-1}\left(\frac{t}{2\pi}\right)^{-1/4}\left[C_0 + C_1\left(\frac{t}{2\pi}\right)^{-1/2} + C_2\left(\frac{t}{2\pi}\right)^{-2/2} + C_3\left(\frac{t}{2\pi}\right)^{-3/2} + C_4\left(\frac{t}{2\pi}\right)^{-4/2}\right]$$

THE RIEMANN-SIEGEL FORMULA AND LARGE SCALE COMPUTATIONS OF THE RIEMANN ZETA FUNCTION 2/2

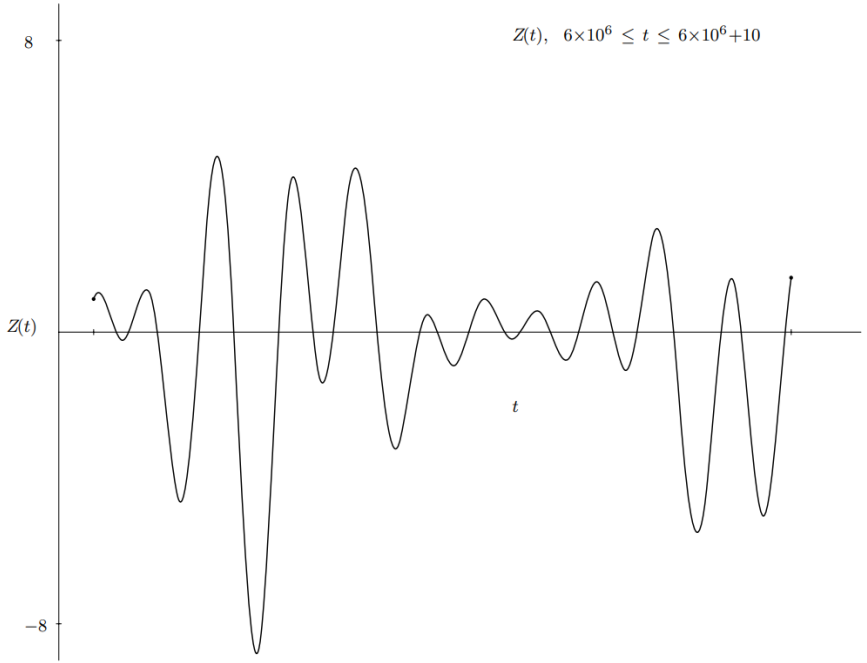
$$C_0 = \Psi(p) = \frac{\cos\left[2\pi\left(p^2 - p - 1/16\right)\right]}{\cos\left(2\pi p\right)},$$

$$C_1 = -\frac{1}{96\pi^2}\Psi^{(3)}(p),$$

$$C_2 = \frac{1}{18\,432\pi^4}\Psi^{(6)}(p) + \frac{1}{64\pi^2}\Psi^{(2)}(p),$$

$$C_3 = -\frac{1}{5\,308\,416\pi^6}\Psi^{(9)}(p) - \frac{1}{3840\pi^4}\Psi^{(5)}(p) - \frac{1}{64\pi^2}\Psi^{(1)}(p),$$

$$C_4 = \frac{1}{2\,038\,431\,744\pi^8}\Psi^{(12)}(p) + \frac{11}{5\,898\,240\pi^6}\Psi^{(8)}(p) + \frac{19}{24\,576\pi^4}\Psi^{(4)}(p) + \frac{1}{128\pi^2}\Psi(p)$$



The Hackathon

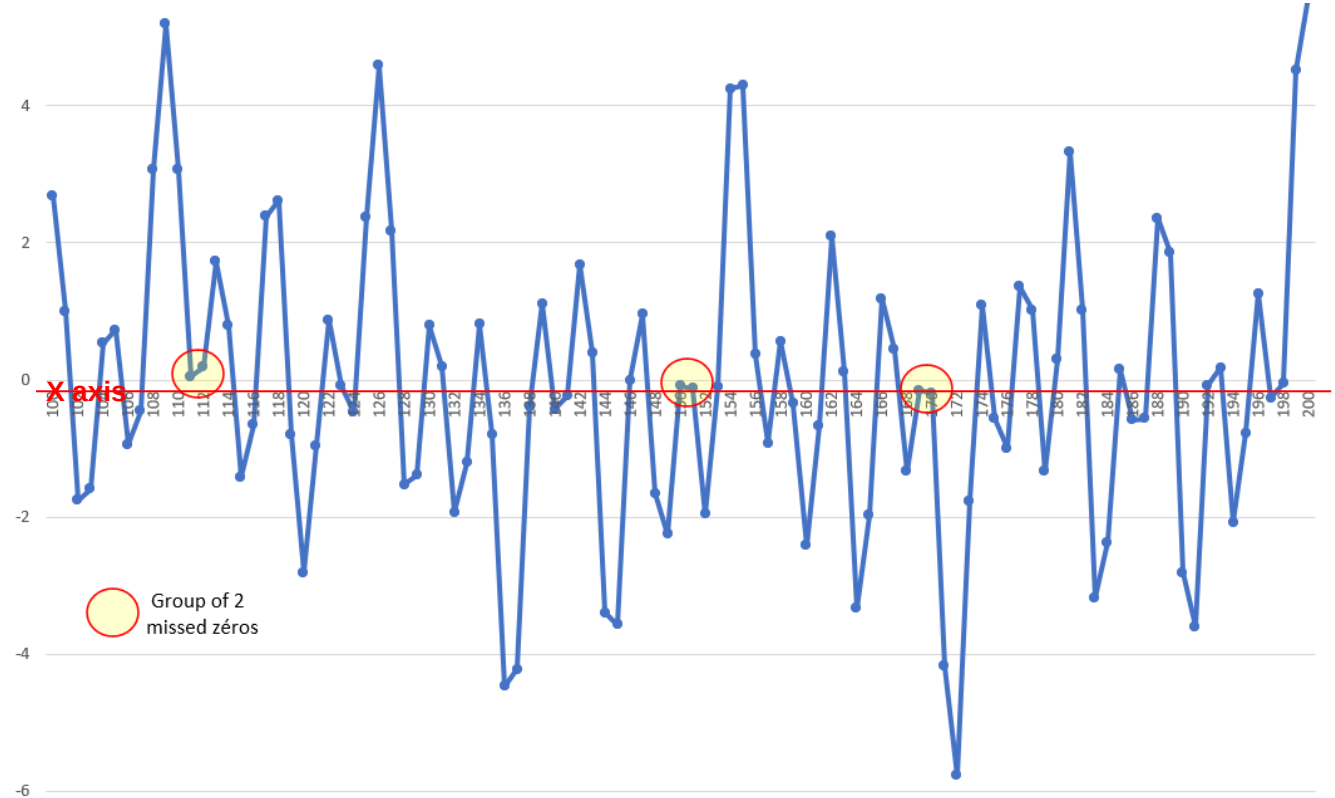
- The objective is to analyze and tune an application written by a mathematician
- The code is serial and poorly tuned
- We are confronted frequently to this situation; it is extremely difficult to be an expert in one domain of sciences and be an expert in tuning of HPC codes
- It’s not a big deal in most situations, but when the execution time increase exponentially with one dimension of the problem, the progress of the scientist stop rapidly
- Then most companies give the R&D code to another team “specialized” in code tuning/transformation and expert in the HPC hardware. This will be your future job
- At the end the code can be profoundly transformed, the performance will be orders of magnitude better

- Our small reference code is really poorly tuned and serial ; then present multiple opportunities for significant improvements

The reference code

```
compile g++ RiemannSiegel.cpp -O -o RiemannSiegel
-------------------------------------------------------------------------
./RiemannSiegel 10 10000 100
I found 10142 Zeros in 3.459 seconds     # OK
-------------------------------------------------------------------------
./RiemannSiegel 10 10000 10
I found 10142 Zeros in 0.376 seconds     # OK
-------------------------------------------------------------------------
./RiemannSiegel 10 100000 10
I found 137931 Zeros in 6.934 seconds    # INCORRECT sampling too small
-------------------------------------------------------------------------
./RiemannSiegel 10 100000 100
I found 138069 Zeros in 56.035 seconds   # OK
-------------------------------------------------------------------------
RiemannSiegel 10 1000000     should find : 1747146     zeros
RiemannSiegel 10 10000000    should find : 21136125    zeros
RiemannSiegel 10 100000000   should find : 248888025   zeros
RiemannSiegel 10 1000000000  should find : 2846548032  zeros
RiemannSiegel 10 10000000000 should find : 32130158315 zeros
```

Effect of sampling with large step Z(t) STEP=1.0



workarounds:

- Increase the sampling

  or

- Detect the missed zeros and calculate a middle point

  or

- Your ideas

All sinusoids SHOULD traverse the X axis, finding an exception would mean that the Riemann Hypothesis is wrong. This has been verified up to 10^13. In others words, if you approach the X axis like in the red circles but do not traverse the X axis; you have missed 2 zeros.

The Hackathon for Riemann Hypothesis

- The objective is to count the largest number of zeros possible with limited compute resource
- This is not the optimal way to compute the zeros; but this exact method was used by the Zetagrid project; then a very good exercise for a Hackathon
- Notice we do not compute the exact value of the zeros; but sample certain regions until we have found the exact “known” count of zeros. The RiemannSiegel function gives at least 6/7 digits of precision in the range of interest, this is far sufficient for the task if you select correctly the points to be evaluated

The Hackathon for Riemann Hypothesis

- You are free to optimize until you break the logic; the code is very sensitive; any logic error will miss some zeros. You should run regularly the validation function provided of some zeros to verify you have at least 4/5 digits correct
- There are many opportunities to optimize; this is an exercise where we want to see some “creativity”; in your carrier you will unfrequently be exposed to the exact same problems; then experience count but creativity is more fundamental
- The problem is trickier than it appears; the strange behavior of the Riemann Zeta function reserves you some difficulties.

The Hackathon for Riemann Hypothesis

- The evaluation will be done on performance “ranking” on zeros computed in the capped interval of time ; but also on strategy and methodology used
- Then document regularly your ideas; this will give points
- We expect 1 or 2 pages describing your work
- Document also what did not work

Thank you

Good luck and have fun

arm

Teratec Hackathon

Conrad Hillairet – Staff HPC Engineer

15th December 2023

Arm Technology is Defining the Future of Computing

A semiconductor design and software platform company

250+ Billion

Arm-based chips shipped to date.

29.2 Billion

Arm-based chips shipped in FY 2021.

650+

Active licensees, growing by 50+ every year.

The global leader in the development of licensable compute technology

R&D excellence for semiconductor companies and large OEMs.

Arm's energy-efficient processor designs and software platforms enable advanced computing

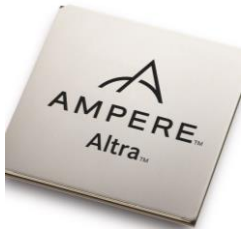
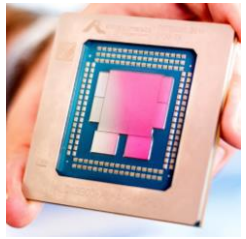
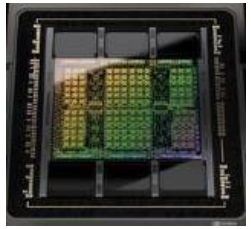
Our technologies securely power products from the sensor to the smartphone and the supercomputer.

Arm delivers the foundational building blocks for trust in the digital world

Arm provides enhanced system-level security technologies such as Arm TrustZone and Arm Confidential Compute Architecture (CCA).


arm NEOVERSE
The Cloud to Edge Infrastructure Foundation
for a World of 1 Trillion Intelligent Devices

Arm Industry Firsts in HPC

AWS	FUJITSU	AMPERE	AWS	NVIDIA	SIPEARL
					
FIRST ARM CPU FOR THE CLOUD	FIRST 1TB/S MEM BW	FIRST >100 CORES PER CPU	FIRST DDR5 PCIE GEN5.0	FIRST LPDDR5X MEM	FIRST European HPC Microprocessor

IP licensing brings **flexibility** – our partners use the flexibility to design their products for their chosen market

Although continuously differentiating and innovating – the standardization of Arm ISA and **standards** ensures compatibility with the software ecosystem

arm

AWS Graviton3

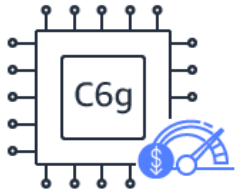
Hardware based on Arm technologies

Graviton2 Processor

Frequency
2.5 GHz

Peak Flops
1280 GFlops

Peak Memory B/W
204 GB/s



Many core
architecture
64 cores

Non-NUMA

7 nm

Arm Neoverse N1

Graviton3 Processor

Frequency
2.6 GHz

Peak Flops
2662 GFlops

Peak Memory B/W
307 GB/s

Many core
architecture
64 cores

Non-NUMA

Energy
efficiency
5 nm

Arm Neoverse V1

Compilers & Tools

I have no experience with Arm

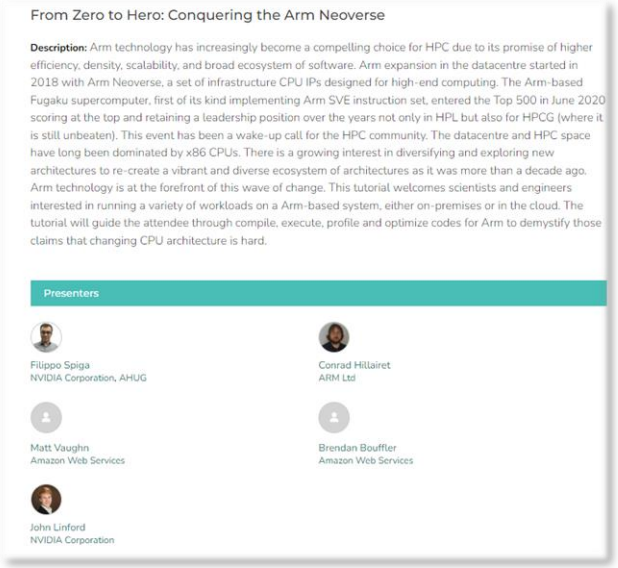
Should I be worried ?

Overthinking:
The art of creating problems that weren't even there.

Most applications compile on Arm architecture with little to no modification

- All major Linux distributions support Arm, with extensive library of common packages (AArch64).
- Upgrading to newer library / software versions might be needed.
- GNU Compiler Collection (GCC) and LLVM are fully supported.
- Commercial compilers for Arm are available.

SC23 Denver, CO | i am hpc.


From Zero to Hero: Conquering the Arm Neoverse

Description: Arm technology has increasingly become a compelling choice for HPC due to its promise of higher efficiency, density, scalability, and broad ecosystem of software. Arm expansion in the datacentre started in 2018 with Arm Neoverse, a set of infrastructure CPU IPs designed for high-end computing. The Arm-based Fugaku supercomputer, first of its kind implementing Arm SVE instruction set, entered the Top 500 in June 2020 scoring at the top and retaining a leadership position over the years not only in HPL but also for HPCG (where it is still unbeaten). This event has been a wake-up call for the HPC community. The datacentre and HPC space have long been dominated by x86 CPUs. There is a growing interest in diversifying and exploring new architectures to re-create a vibrant and diverse ecosystem of architectures as it was more than a decade ago. Arm technology is at the forefront of this wave of change. This tutorial welcomes scientists and engineers interested in running a variety of workloads on a Arm-based system, either on-premises or in the cloud. The tutorial will guide the attendee through compile, execute, profile and optimize codes for Arm to demystify those claims that changing CPU architecture is hard.

Presenters

Filippo Spiga
NVIDIA Corporation, AHUG

Conrad Hillairet
ARM Ltd

Matt Vaughn
Amazon Web Services

Brendan Bouffler
Amazon Web Services

John Linford
NVIDIA Corporation

https://github.com/arm-hpc-user-group/sc23-tutorial-neoverse

arm

arm COMPILER

Arm provided C/C++/Fortran compiler with best-in-class performance

Compilers tuned for Scientific Computing and HPC

Latest features and performance optimizations

Freely available

Tuned for Scientific Computing and HPC workloads

- Processor-specific optimizations for Neoverse-based platforms
- Optimal shared-memory parallelism using latest Arm-optimized OpenMP runtime

Linux user-space compiler with latest features

- C++ 17 and Fortran 2003 language support with OpenMP 4.5
- Support for Armv8-A and SVE architecture extension
- Based on LLVM and Flang, leading open-source compiler projects

Freely available on leading Linux distributions

Documentation

- Fortran : https://developer.arm.com/documentation/101380/2210
- C/C++ : https://developer.arm.com/documentation/101458/2210

arm PERFORMANCE LIBRARIES

Optimized BLAS, LAPACK and FFT for HPC applications

Freely available

Best-in-class performance

Validated with
NAG test suite

Arm provided 64-bit Armv8-A math libraries

- Optimized BLAS, LAPACK, FFT and math.h routines
- FFTW compatible interface for FFT routines
- Sparse linear algebra and batched BLAS support

Best-in-class serial and parallel performance

- Generic Armv8-A optimizations by Arm
- Tuned for Arm Neoverse family of processors

Validated by Arm Engineers

- Validated with NAG's test suite, a de-facto standard
- Community supported

Documentation

- https://developer.arm.com/documentation/102620/0100

Documentation

Have a look to the manuals

- Get started on Arm Guide https://developer.arm.com/documentation/102841/0100
- Arm Fortran Compiler https://developer.arm.com/documentation/101380/2210
- Arm C/C++ Compiler https://developer.arm.com/documentation/101458/2210
- Arm Performance Libraries https://developer.arm.com/documentation/102620/0100

GNU compilers are a solid option

With Arm being significant contributor to upstream GNU projects

- GNU compilers are first class Arm compilers
  - Arm is one of the largest contributors to GCC
  - Focus on enablement and performance
  - Key for Arm to succeed in Cloud/Data center segment
- GNU toolchain ships with Arm Allinea Studio
  - Best effort support
  - Bug fixes and performance improvements in upcoming GNU releases
- GCC 11.2.0

MPI

Parallel Programming

- Out-of-the-box support since 3.1.2 (currently 4.1.4)
- Provided by default on AWS EC2 c7g instances
- Upstream contributions
- Used inhouse
- Active development from Arm and Arm partners

Debugging with Linaro DDT

Analyze memory usage

Visualize data structures

Switch between MPI ranks and OpenMP threads

Linaro FORGE

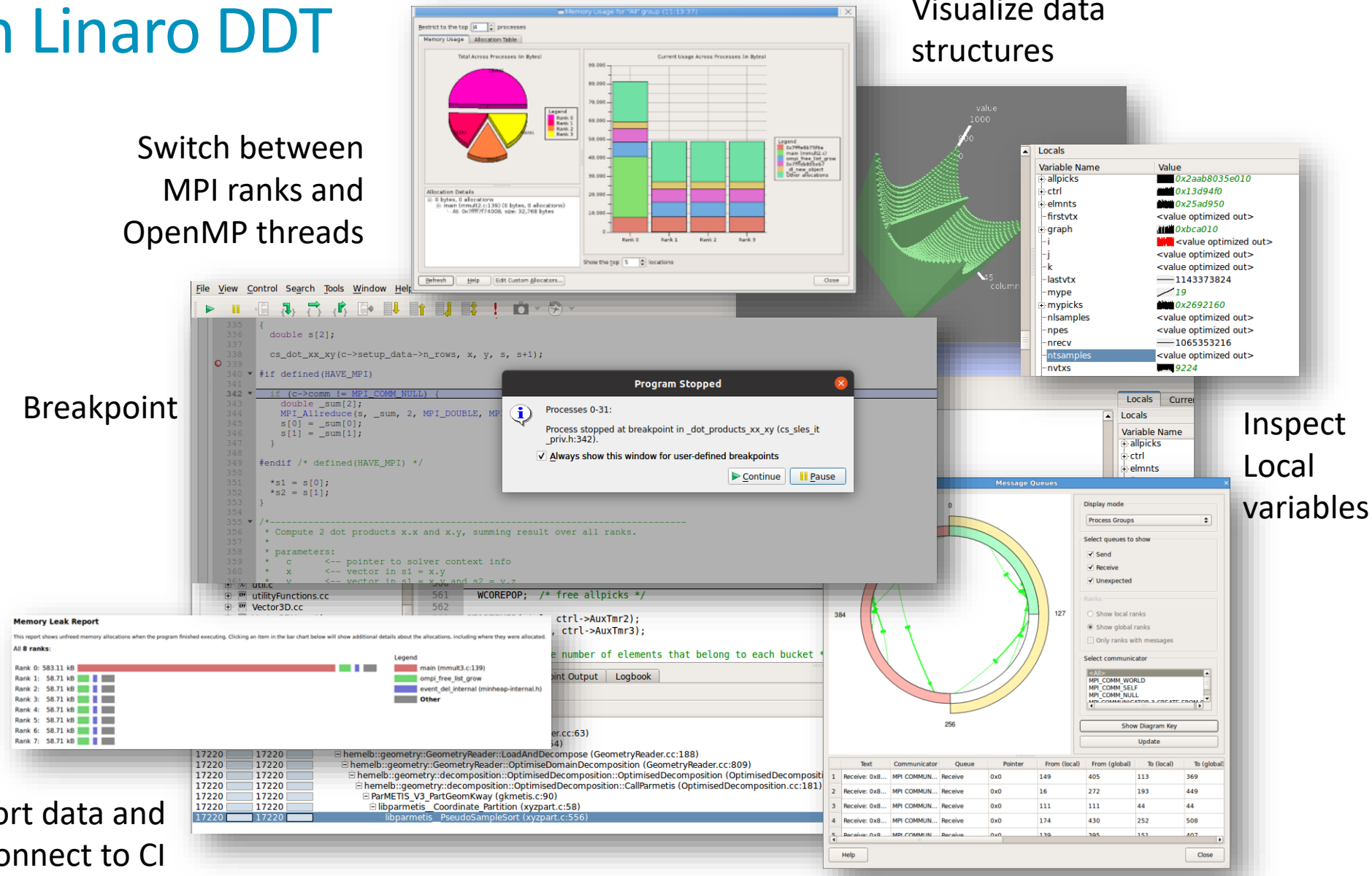
Breakpoint

Inspect Local variables

Export data and connect to CI system

Backtraces & Callstacks

Display pending communications

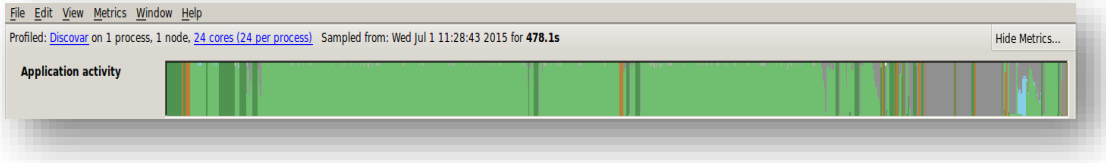


arm

Profiling with Linaro MAP

Linaro FORGE

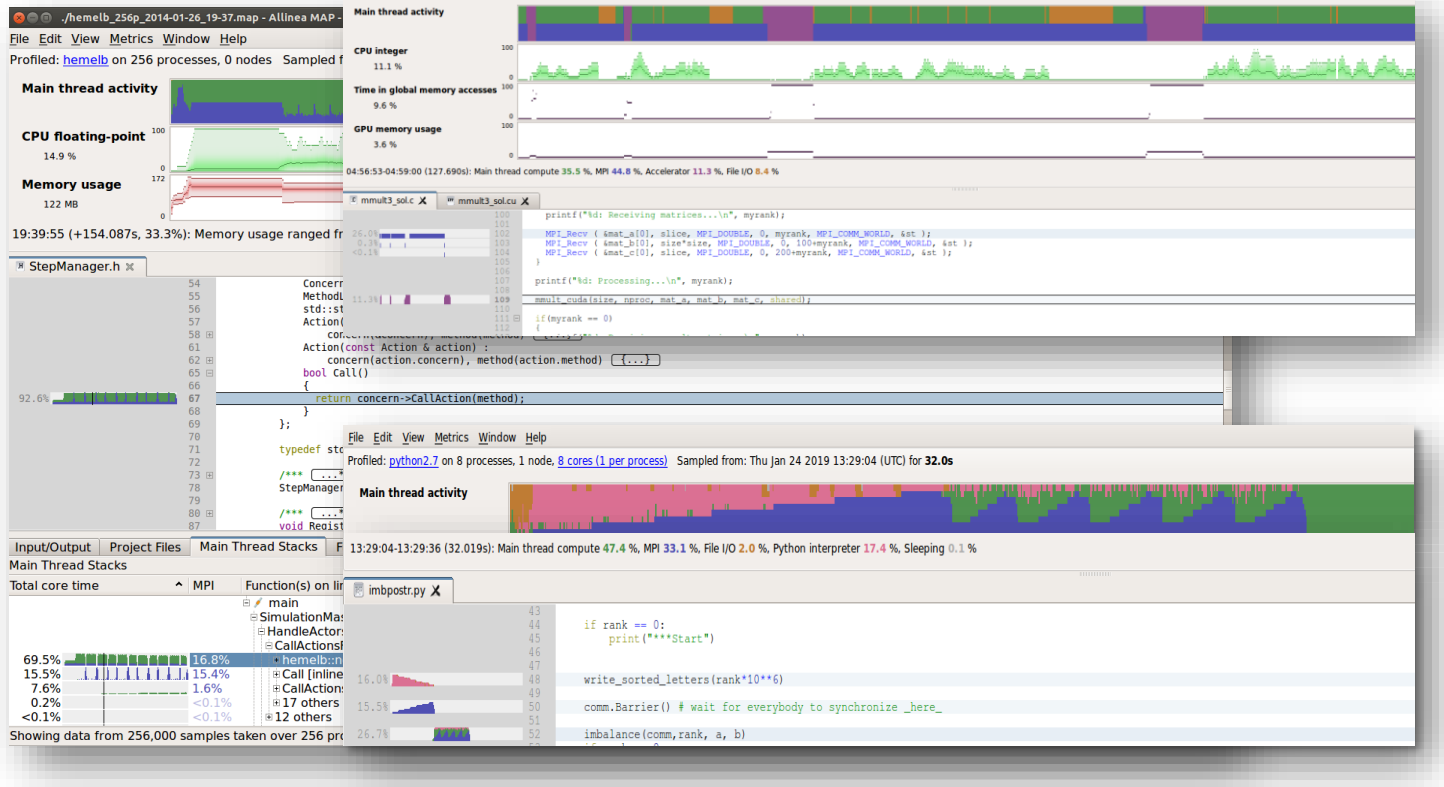
Inspect OpenMP activity



Understand MPI/CPU/IO operations thanks to timelines and metrics

Analyze GPU efficiency

Investigate annotated source code and stack view



Identify bottlenecks

Profile Python-based workloads

arm

Documentation

Have a look to the manuals

- Get started with Linaro MAP

  https://docs.linaroforge.com/23.0/html/forge/map/get_started_map/index.html

- Get started with Linaro DDT

  https://docs.linaroforge.com/23.0/html/forge/ddt/get_started_ddt/index.html

Support

Struggling with something during the Hackathon ?
Shout ! You are not alone.

Support

How can I get some help during the event ?

- Via Slack
  1. Join the AHUG Slack Workspace
     - You may receive an invitation prior to the event
     - Link available here https://a-hug.org/contact/
  2. Join the **teratec-hackathon-hpc** slack channel
     - Send a private message to Conrad Hillairet
  3. Ask your questions:
     - In the slack channel
     - Using private message to Conrad Hillairet or Kévin Tuil
- Email
  - conrad.hillairet@arm.com
  - kevtuil@amazon.fr

arm

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה

Teratec Hackathon

Benjamin Depardon

15/12/2023

one thing is sure…

Hybrid is the future of HPC

The question is

HOW TO GET THERE?

Your key partner for Hybrid HPC

OPTIMIZE

OKA Suite

WORKCLOUD

SELECT

RATIONALIZE

NICE
an aws company

aws partner network

FLEXIBILIZE

HPC Cloud Journey

Consulting

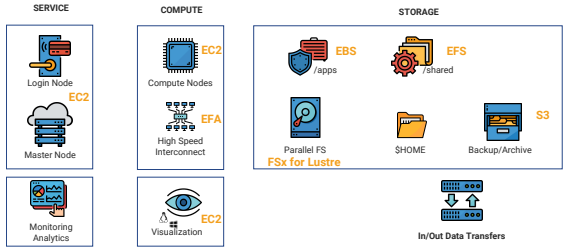
OKA Suite

Identify HPC Workloads and their On-Premises Behaviour

WORKCLOUD

Understand the appropriate Move to Cloud Strategy

Build



HPC Reference Architecture

CCME

Prototype / Test / Benchmark Integration & Move-to-Cloud

Run

CCME

CCME Support & Professional Services

Storengy Moves HPC to AWS, Runs Geoscientific Simulations 2.5 Times Faster

2021

Storengy, a subsidiary of the ENGIE Group, is a leading supplier of natural gas. The company offers gas storage, geothermal solutions, carbon-free energy production, and storage technologies to enterprises worldwide. To ensure its products are properly stored, Storengy uses high-tech simulators to evaluate underground gas storage, a process that requires extensive use of high-performance computing (HPC) workloads. The company also uses HPC technology to run natural gas discovery and exploration jobs.

For many years, Storengy ran its HPC workloads in an on-premises IT environment, but it struggled to manage an increase in jobs. “Our HPC environment was not designed to scale easily. We had to do larger simulations in a very short time as our business grew, and we lacked the ability to support the gas exploration workloads,” says Jean-Frederic Thebault, engineer at Storengy.

Storengy also sought to accelerate the deployment of HPC clusters for its engineers. “It typically took weeks or sometimes months to provision server clusters for a new project,” says Thebault. “We wanted our engineers spending their time on research, not provisioning.”

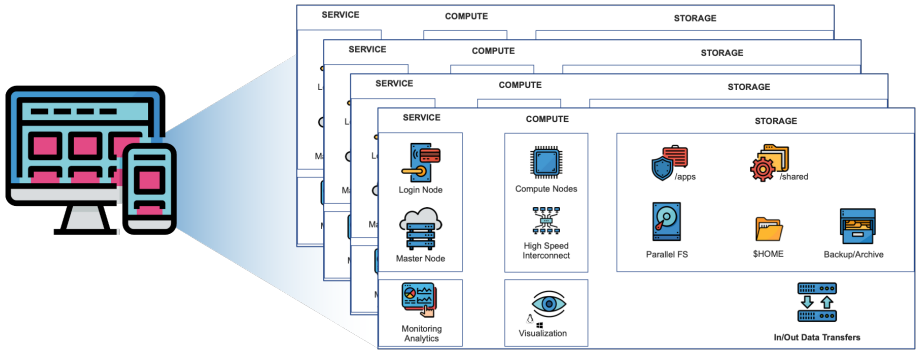
> **Using the CCME tool on AWS, we can deploy HPC resources in 30 minutes, compared to the weeks or months it would take to procure servers and provision compute in our on-premises environment.”**
>
> **Jean-Frederic Thebault**
> *Engineer, Storengy*

What is CCME?

Solution to transparently **build customizable HPC clusters in AWS**

CCME

Match workloads' needs: adapt the type and number of **compute resources**, provision a **high-performance network** and **file system**, automatically **scale in and out…**



adaptability simplicity
data management
heterogeneity publish services
remote visualization
standard jobs scheduler accessibility
web interface user management security
cost management

Cluster-in-a-Box: For An easy AWS Cloud Journey

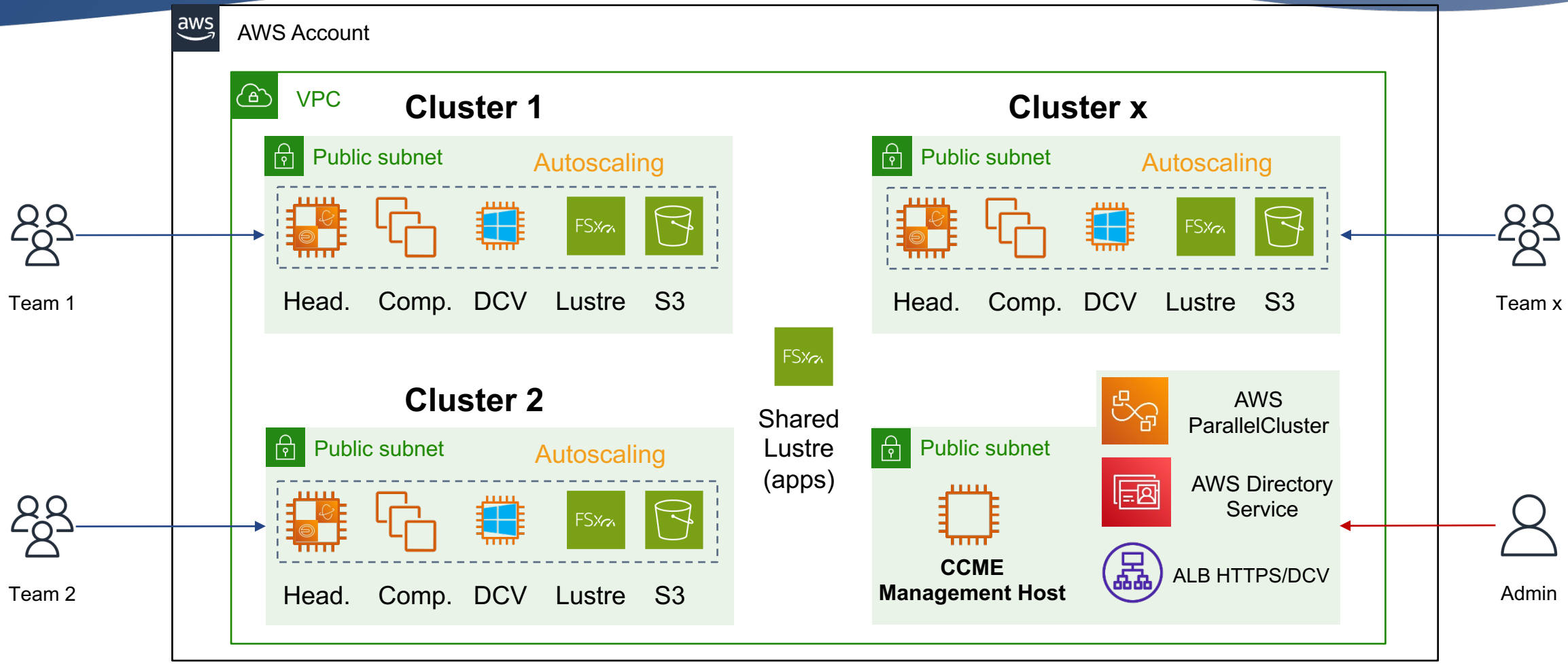
UCit manages it for You

Your HPC Cluster of CHOICE on AWS…

Cost & Budget under Control

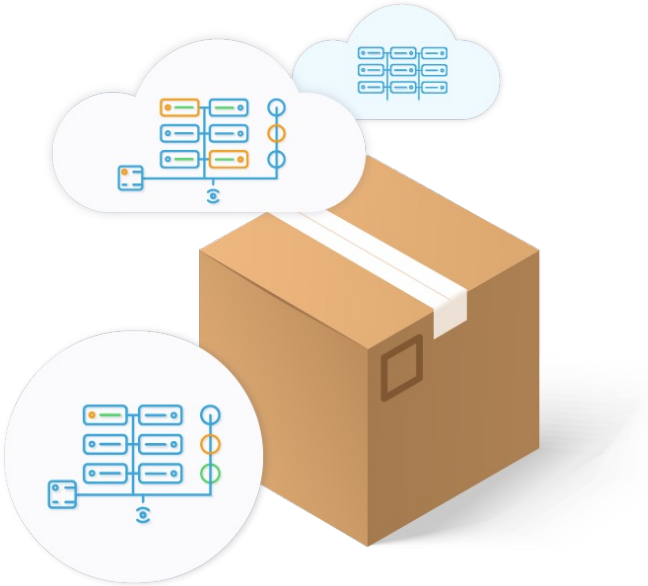
Simple Access through SSH or a Web Portal

Hackathon: Architecture



Cluster available for each team

- Each cluster is fully isolated from the others (network, accounts…)
- Job scheduler: SLURM
- Headnode : c7g.4xlarge
- Compute: partition
  - **hpc7g.16xlarge (64 vCPUs, 128 GiB) – limit 4 instances**
  - ➢ **256 vCPUs available**
- Storage:
  - Shared NFS – 500GiB (/home)
  - **FSx for Lustre – 1.2TB (/fsx)**
  - Shared FSx for Lustre with source codes and tools (Read Only)
- Remote access
  - **SSH** connection to frontend node through login/password
  - **Web portal** EnginFrame + **DCV** remote desktop on a Windows EC2 instance (g4dn.xlarge)
  - **AWS Console**



The Hackathon Box

Learn more at

www.ucit.fr

benjamin.depardon@ucit.fr

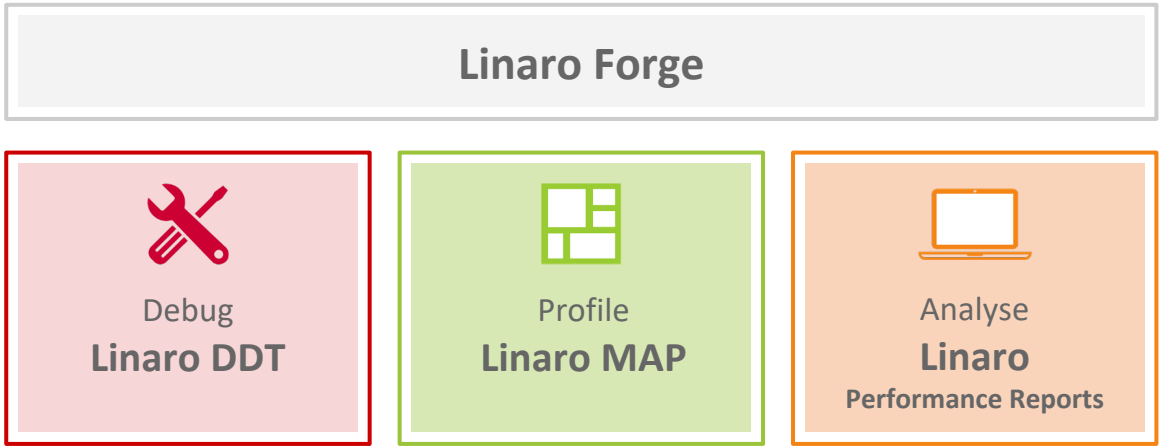
Linaro Forge

Teratec HPC Hackathon 22-29/01/2024
Tools presentation, 15/12/2023

Marcin Krzysztofik
Director, Sales
Linaro Forge
Marcin.Krzysztofik@linaro.org
+44 7803 519 131

Linaro Forge: a highly-scalable toolkit for debugging and profiling

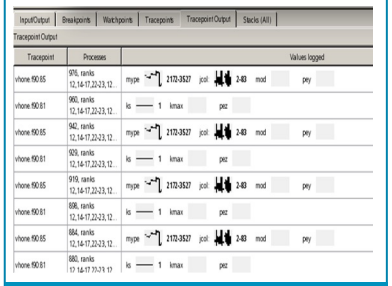
- Widely used, best in class commercially supported tools for HPC
- Fully supported by Linaro on all mainstream HPC hardware (CPUs & GPUs)
- Easy to use and intuitive interface



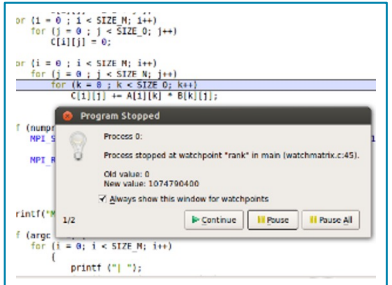
Performance Engineering for any architecture, at any scale

Linaro

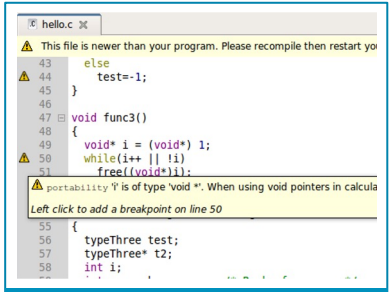
Linaro DDT Debugger Highlights



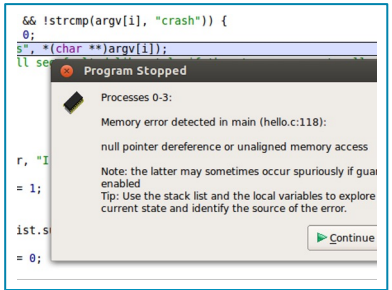
The scalable print alternative



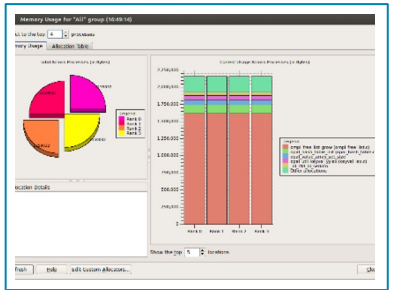
Stop on variable change



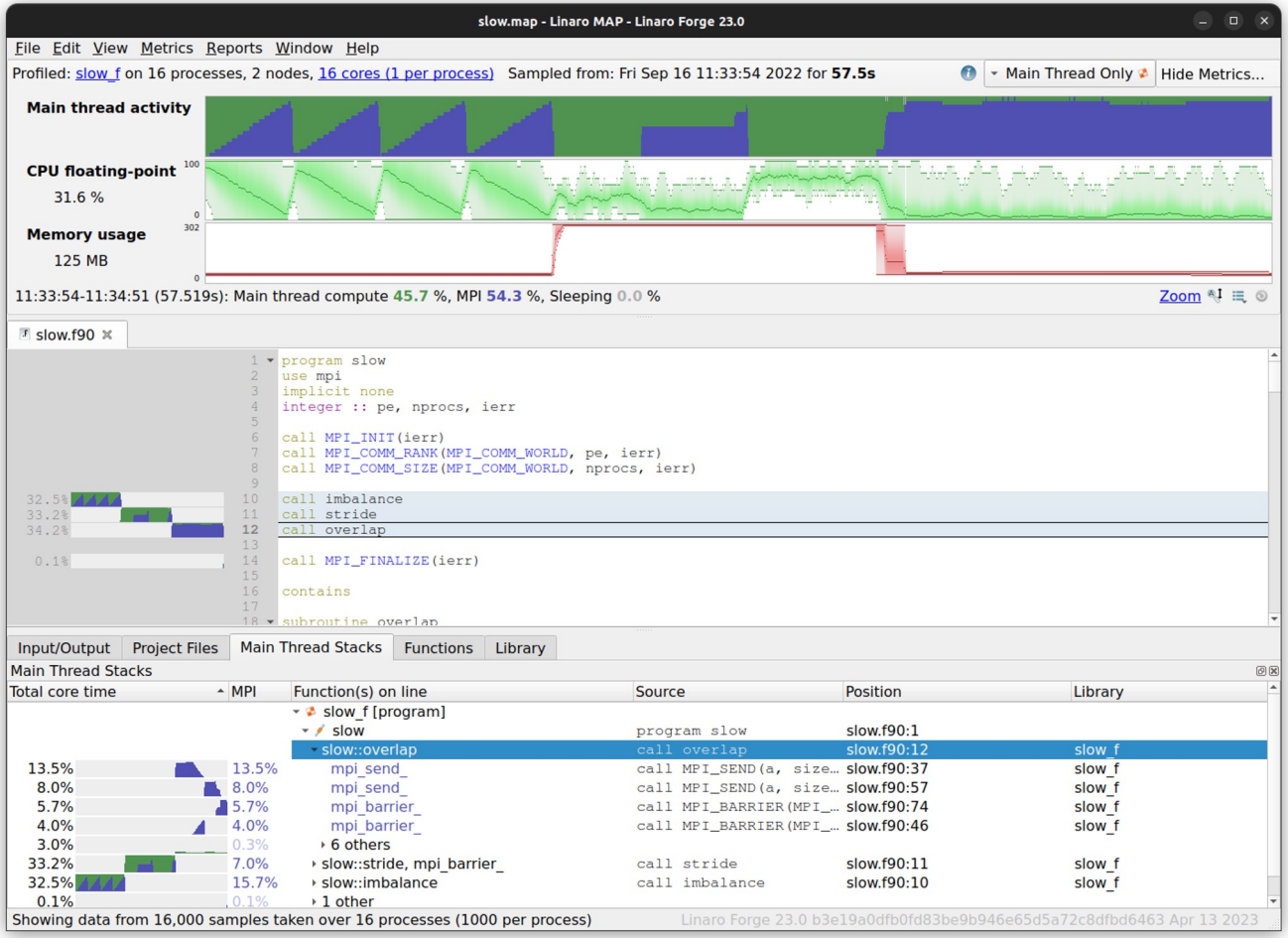
Static analysis warnings on code errors



Detect read/write beyond array bounds



Detect stale memory allocations



Linaro MAP

Sampling based profiler

- Built on same framework as DDT
- Parallel support for MPI, OpenMP, CUDA

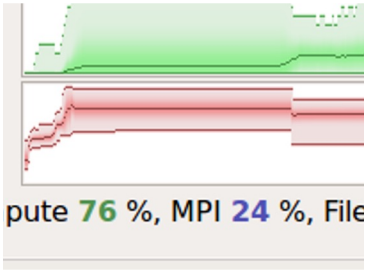
Designed for ‘hot-spot’ analysis

- Stack traces
- Augmented with performance metrics

Adaptive sampling rate

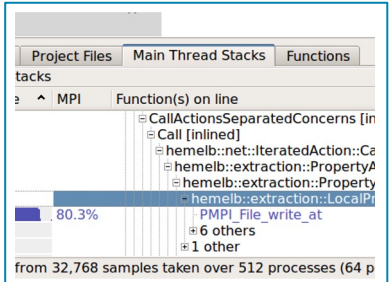
- 1,000 samples per process
- Low overhead, scalable and small file size

Linaro MAP Source Code Profiler Highlights

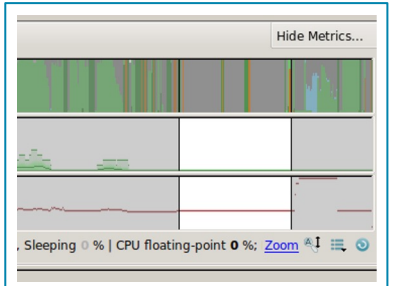


Find the peak memory use

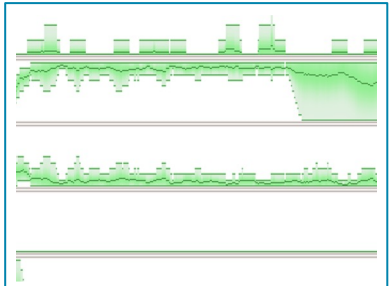
Fix an MPI imbalance



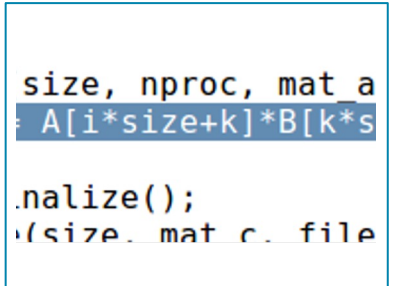
Remove I/O bottleneck



Make sure OpenMP regions make sense



Improve memory access

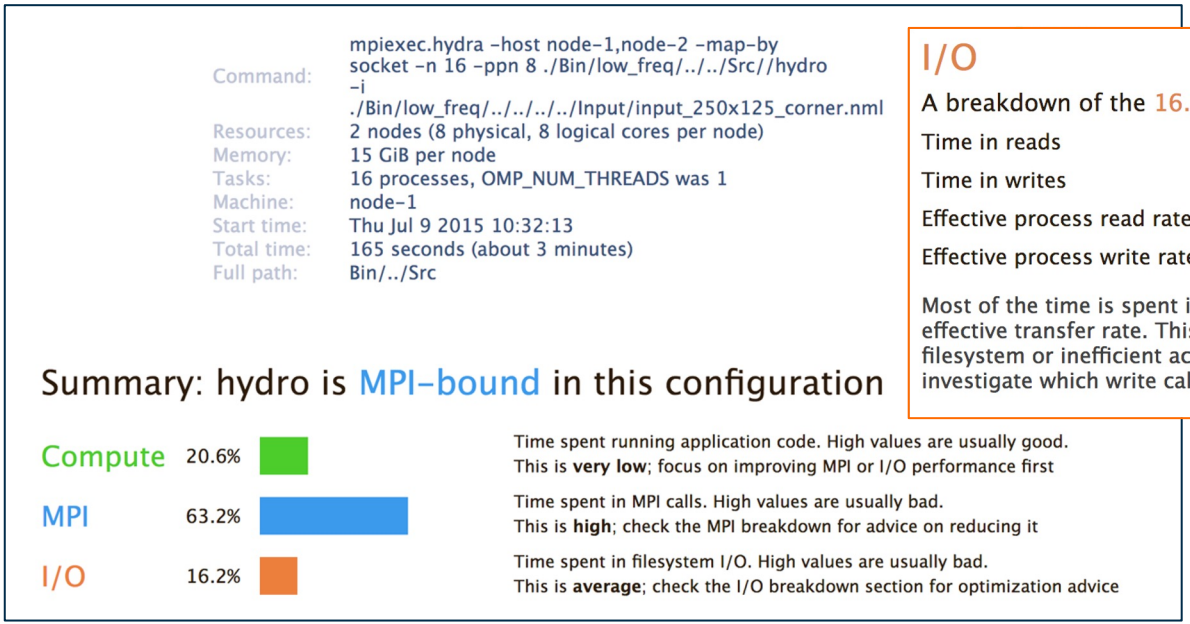


Restructure for vectorization

Linaro

Linaro Performance Reports

A high-level view of application performance with “plain English” insights



Command:	mpiexec.hydra -host node-1,node-2 -map-by socket -n 16 -ppn 8 ./Bin/low_freq/../../Src//hydro -i ./Bin/low_freq/../../../../Input/input_250x125_corner.nml
Resources:	2 nodes (8 physical, 8 logical cores per node)
Memory:	15 GiB per node
Tasks:	16 processes, OMP_NUM_THREADS was 1
Machine:	node-1
Start time:	Thu Jul 9 2015 10:32:13
Total time:	165 seconds (about 3 minutes)
Full path:	Bin/../Src

Summary: hydro is MPI-bound in this configuration

Compute	20.6%	Time spent running application code. High values are usually good. This is **very low**; focus on improving MPI or I/O performance first
MPI	63.2%	Time spent in MPI calls. High values are usually bad. This is **high**; check the MPI breakdown for advice on reducing it
I/O	16.2%	Time spent in filesystem I/O. High values are usually bad. This is **average**; check the I/O breakdown section for optimization advice

I/O

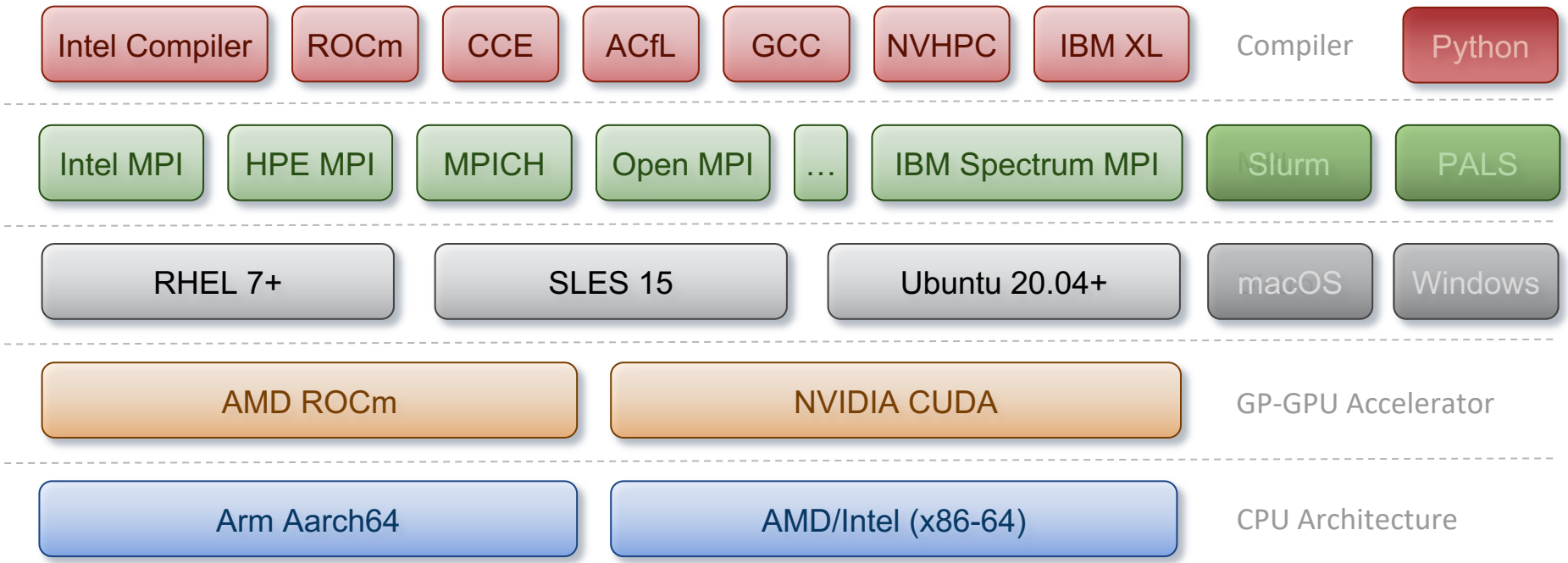
A breakdown of the 16.2% I/O time:

Time in reads	0.0%
Time in writes	100.0%
Effective process read rate	0.00 bytes/s
Effective process write rate	1.38 MB/s

Most of the time is spent in write operations with a very low effective transfer rate. This may be caused by contention for the filesystem or inefficient access patterns. Use an I/O profiler to investigate which write calls are affected.

Linaro

Supported Platforms



Thank you