RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

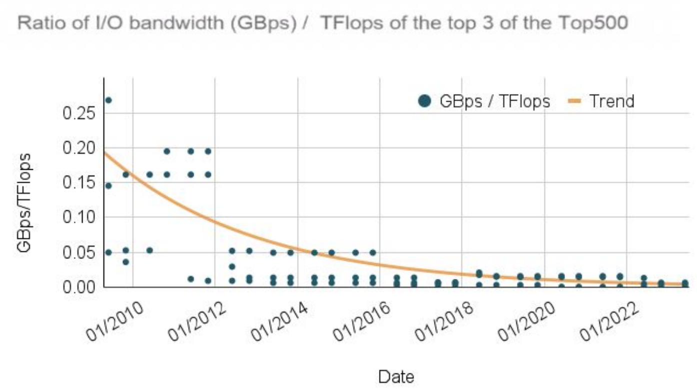
FRANCE 2030
PROGRAMME DE RECHERCHE
NUMÉRIQUE POUR L'EXASCALE

WP3 - ML-based data analytics

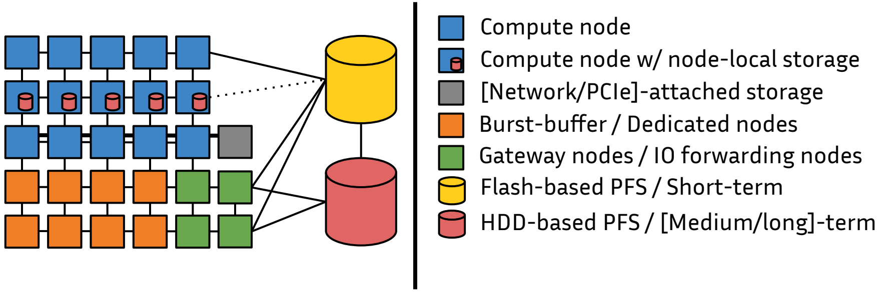
Thomas Moreau & Bruno Raffin

Data at exascale: a challenge in hardware

- Increasing **gap between compute and I/O** performance on large-scale systems
  - Ratio of I/O to computing power divided by ~10 over the last 10 years on the top 3 supercomputers
- … and data deluge!
  - At NERSC, **data volume x41** in 10 years



- New storage tiers and advanced architectures to try to mitigate this increasing bottleneck
  - More complex on-node memory layout
  - Emerging complex applications and workflows have to adapt



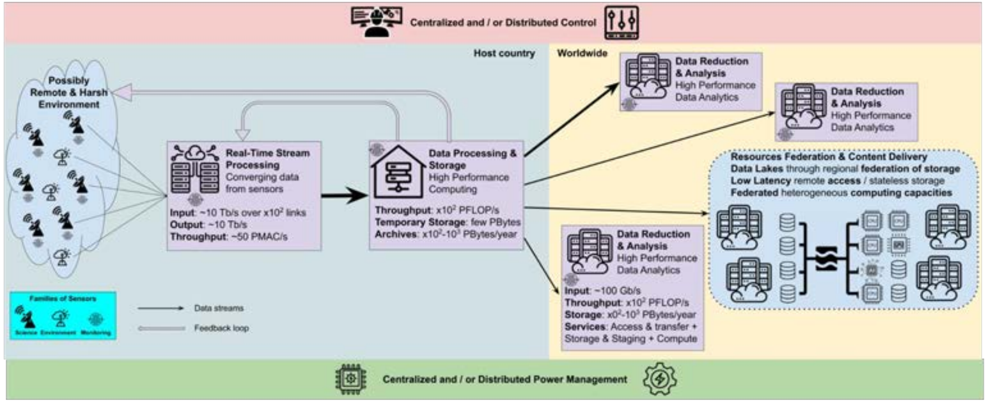
Trend in storage technologies available on extreme-scale systems

Data at exascale: a challenge in usages

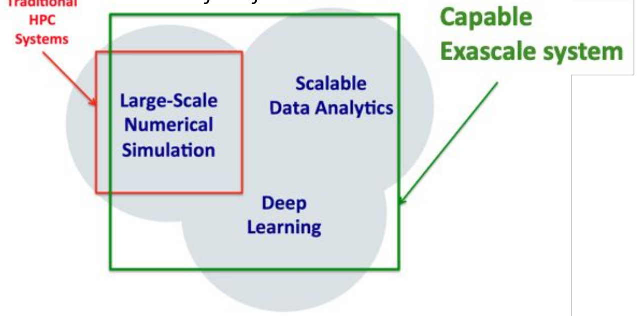
- HPC centers do not live in isolation anymore
  - Edge - Cloud - HPC continuum

- Emerging workloads are hybrid
  - High-performance simulation
  - High-performance data analytics
  - Machine learning and artificial intelligence

- Interaction with data from the outside world sensors
  - Large scientific instruments
  - …



SKA data workflow from sensors to HPC centers



RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité
FRANCE 2030
PROGRAMME DE RECHERCHE
NUMÉRIQUE POUR L'EXASCALE

cea cnrs Inria

Our ambition

Fill the gaps in the existing software stack designed by previous projects (e.g. ECP)

Take into account French & European specificities

Ensure French & European needs are taken into account in roadmaps

Approach:

- **Research** on data-oriented tools for HPC
- Transverse, **re-usable tools**
- Usable **in production** at exascale

⇒ ExaDoST will produce:

- **New approaches** to handle the data challenge at exascale
- Transverse **libraries & tools** that implement these approaches

Validated in illustrators at full scale

Fully application agnostic

Fully open-source

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

FRANCE 2030

PROGRAMME DE RECHERCHE
NUMÉRIQUE POUR L'EXASCALE

Work Packages in Exa-DoST

WP1: Exascale
I/O and storage

WP2: Exascale
In-situ data processing

WP3: Exascale
ML-based data analytics

WP4: Shared building blocks
& integrated illustrators

WP5: Management, dissemination and training

Identified applications

From discussion with Gysela / SKA / Coddex

Event detection and tracking

- Finding and tracking patterns
- Finding change points

Anomaly detection

- Modeling nominal data
- Finding model deviation

Data compression

- For storage/comm’
- For anomaly detection

Challenges

- *Data cannot be stored* -> need learning algorithms that can handle streams of data
- *Data is distributed* -> need models that work on subdomains
- *Labelling is costly* -> unsupervised learning/transfer learning
- *In situ* -> need to be fast enough and have limited auxiliary memory

Event detection

Codex

- Hot spot
- only few event per simulation/ few simulations
- Used for steering

Tokam2D/Gysela

- Burst of density
- many events per frame
- Trajectory are of interest
- Used for steering

SKA

- Fast radio bursts
- few events but many "frames"
- The trajectory of events is of interest

In Mesh data

In Nd-array that evolve through time

Distributed data

Single node data

Roadmap: adapt Computer Vision literature to physical signals

Data-driven compression

- Necessary to do compression to store/communicate the simulation result (big Nd-array)
- But compression can be adapted to specifically compute some diagnostic (statistics)

This is the interest of data-driven compression

Gysela/Tokam2d:

- Compression of the 3D information to have the best reconstruction?
- The compression model needs to run with the distributed data

Machine Learning Motifs

From a ML perspective

Learning from distributed data

Learn a model that makes local decision based on Nd-array data partitioned into sub-domains, by minimizing communication and auxiliary memory consumption.

Unsupervised event-tracking

In large Nd-array evolving in time, some patterns are repeating (spatially) and moving (time). We would like to identify them and track them automatically, if possible with low memory/latency.

Anomaly detection

Detect deviation of the simulation with normal behavior to be able to stop simulation before numerical instability.

The Large Scale Ensemble Motif

From one to many simulation runs (parameter-sweep) to sample the simulation behavior in the parameter space

A classical pattern for:

- Sensibility Analysis
- Data Assimilation
- Deep Surrogate Training
- Simulation Based Inference

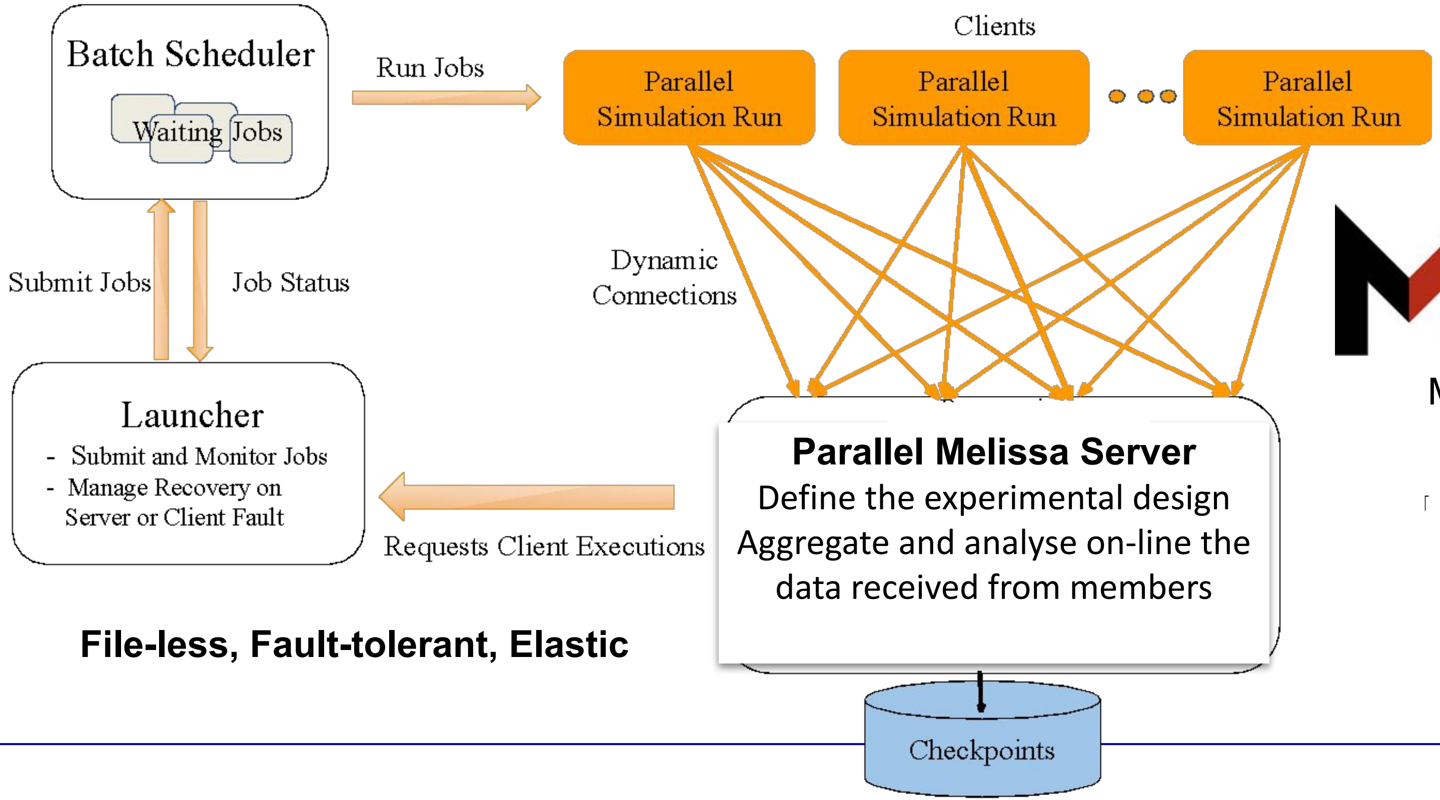
Exascale: Embarrassingly parallel but still not that easy:

- Beware of data aggregation and I/Os.
- Support for heterogeneity, resilience, elasticity, modularity are critical at large scale

Melissa: A framework for large scale ensemble runs and on-line data processing:

Open source, Free-BSD: https://gitlab.inria.fr/melissa

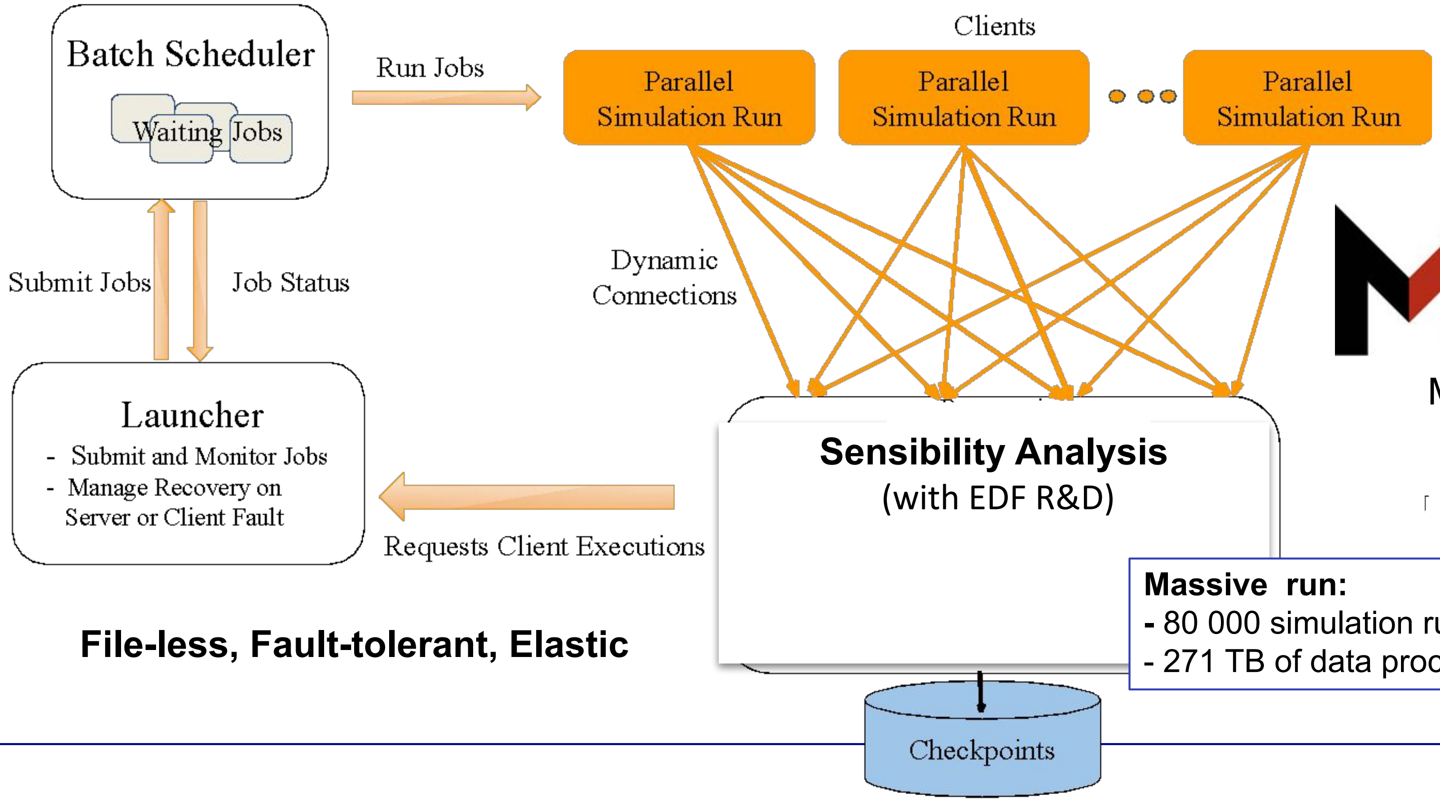
Melissa Architecture



Melissa

File-less, Fault-tolerant, Elastic

Melissa Architecture

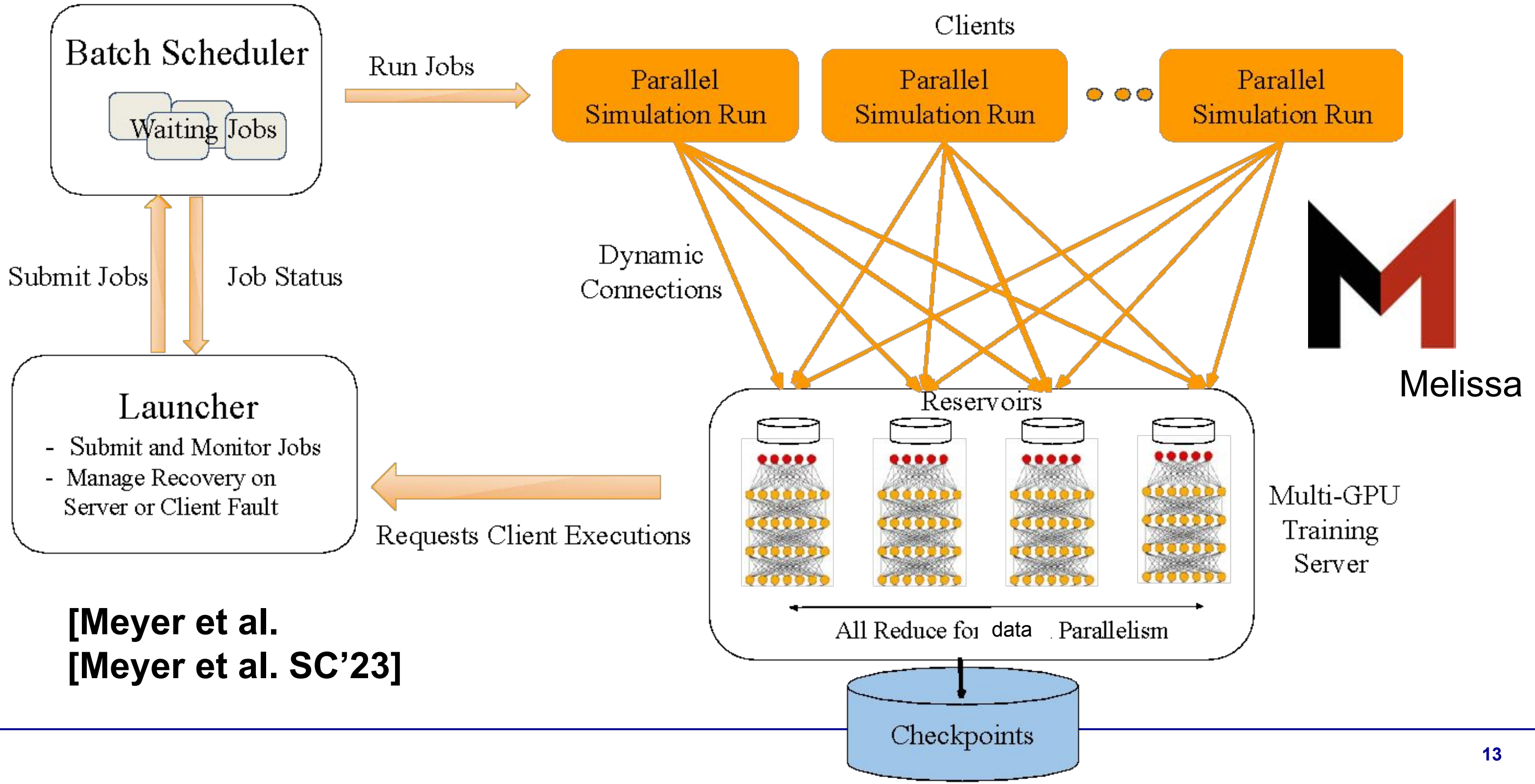


Melissa

File-less, Fault-tolerant, Elastic

Massive run:
- 80 000 simulation runs
- 271 TB of data processed online

Deep Surrogate Training



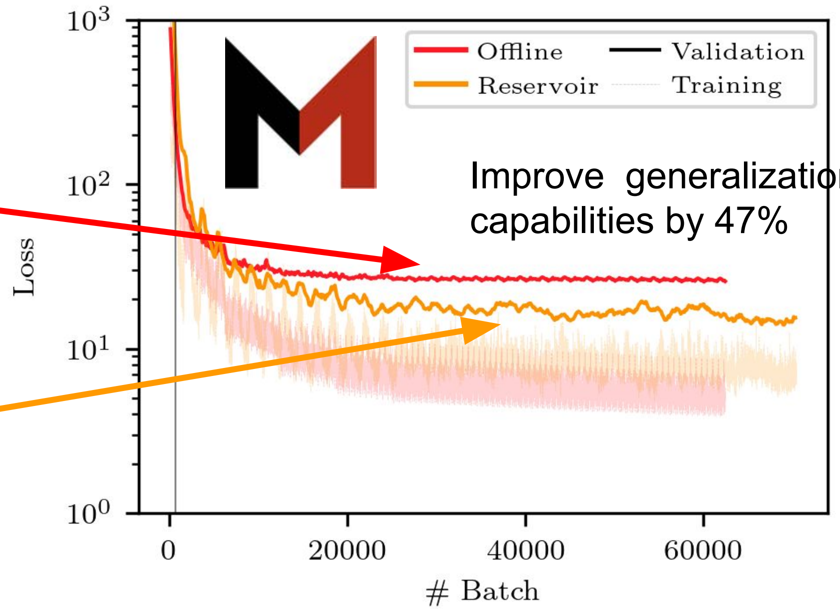
[Meyer et al.
[Meyer et al. SC'23]

Heat-PDE Surrogate Training

Solver: 2D Heat PDE on a 1000x1000 grid.

Offline training: 250 simulations, 100 epochs, 100 GB Dataset, **24.5h training on 4 GPUs**

Online training: 5 120 cores to run 20 000 simulations generating 8TB of data processed online with **4 GPUs in 1.9 h**



Improve generalization capabilities by 47%

Based on GENCI consolidated costs (1 kh/core CPU = 6 euros, 1 kh/GPU V100 = 360 euros, 1To SSD storage = 56 euros):

Offline data generation + initial training: **49.1 euros**, retraining 41.16 euros (but storing the 8TB would cost **448 euros**)

Online training: 63.8 euros

Ensemble Based Motifs

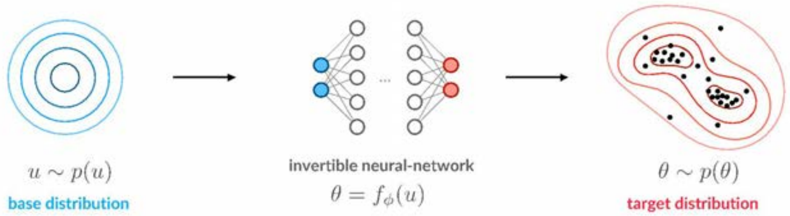
Direct problems:

- Sensibility Analysis
- Deep Surrogate Training (currently investigating active learning)

Inverse problems:

- Data Assimilation (Ensemble Kalman Filters, Particle Filters)
- Simulation Based Inference (SBI):

Ensemble to train an invertible stochastic NN (Normalizing Flow) to learn the posterior



$$p_\phi(\theta) = p\left(f_\phi^{-1}(\theta)\right)\ |\det \nabla f_\phi|^{-1}$$

HPC versus DL Software

HPC traditional programming stack:

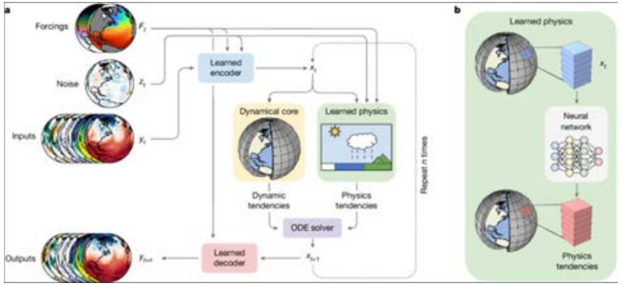
- Fortran / C
- MPI (message passing interface)
- OpenMP (for multicore programming)
- CUDA / OpenMP/ OpenCL / Sycl / Kokkos… for GPU programming

~~Deep Learning~~ Differentiable programming stack:

- Python
- Tensorflow / Pytorch / Jax (NUMPY+ Auto Diff)
- Transparent GPU support through advanced JIT optimizations
- MPI for parallel training on multiple GPUs

Attempts to use these tools for developing classical solvers (JAX-Fluids, JAX-CFD)

NeuralGCM [Kochkov et al. 2024]



RÉPUBLIQUE FRANÇAISE

Liberté
Égalité
Fraternité

FRANCE 2030

PROGRAMME DE RECHERCHE

NUMÉRIQUE POUR L'EXASCALE

Retrouvez toutes nos actualités

NumPEx