RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

FRANCE 2030
PROGRAMME DE RECHERCHE
NUMÉRIQUE POUR L'EXASCALE

ExaDoST - Work Package 3
Exascale ML-based Analytics

WP Leaders:
Thomas Moreau (Inria Saclay)
Bruno Raffin (Inria Grenoble)

Breakout session summary

WP3 – Exascale ML-based Analytics - Objectives

Process the data **automatically** as they are produced

- Identify the algorithms which are relevant for such use cases

  ⇒ Data-integration / Anomaly detection / Unsupervised learning

- Identify/benchmark core ML building components to use these algorithms

  ⇒ Distributed / non i.i.d. learning

- Develop software bricks required to unlock / scale these use-cases

  ⇒ distributed learning paradigms and ensemble runs

  ⇒ in-situ workflows and benchmarks

  ⇒ Distributed computing stack in python (e.g. for scikit-learn)

Coddex: simulating crystals

- Simulation for crystal plasticity
- PDI/Damaris equipped for interface with external libraries
- GPUs are not used by the code so possibility to use them in-situ

Dyablo: astrophysics simulation

- Multi-physics simulation
- Improvement over RAMSES, in particular with AMR
- Young code

AI use-cases

Dyablo: no clear use-cases for now.

- **Event detection**:
  - *Coddex*: mostly based on physics models
  - These models could be used in ML algo?
- **Anomaly detection**:
  - *Coddex*: clear need for anomaly detection, in particular “non-physical” fields
  - Need to find annotations to be able to validate the models
- **Simulation-based inference**:
  - *Coddex*: very interested by such application (Targeting SBI sprint jan.)
  - Interest in learning from smaller scale simulations and generalize on larger ones

A common issue: input of AI systems

- **Coddex** - regular grid data (tensors)
- **Dyablo** - AMR (Oct-tree which is refined as the simulation goes)
    - A postdoc is working on developing a data format for easy

⇒ Finding efficient ways to input data to AI-models would be interesting

- Convolutional layers are an option
- Discussions on sampling-based representations, which are independent of the data format (provided one can sample efficiently)

SKA: astrophysics acquisition

- Acquire uvw visibility (Fourier freq), not on a grid, then reconstruct image
- Large stream of data (1Tb/s), not preserved on the observatory
- Limited computational power (peak 2MW)

Astronomers reluctant to use AI for image reconstruction

Simulator of SKA:

  - Python library: Oskar

AI use-cases: SKA

- Identifying events and anomalies, monitoring the pipeline
  - Classification of the constant sources (image or fourier domain)
- Categorizing transient element, which can be artefact (fourier domain)
  - Possible to generate some data
- Image reconstruction: potentially for dynamic imaging

Work plan: Find representative instances of SKA's workflows

- Inference:
  - Run an AI model on a large image (image reconstruction)
  - Run an AI model on a stream of many images
- Training:
  - Train a model on online data (compression of uvw plan)

Gysela: 5D gyrocinetic code

- Data: 5D regular mesh with 3D coords + 2D velocities
  - Medium resolution: 1024x1024x64 x 128x 64
  - Fields
    - 5D (Vlasov equation):
      - Ions
      - Electrons
      - Impurities
    - 3D (Poisson equation)

The amount of compute hours needed for one run and the amount of data generated is a major challenge.

AI use-cases: Gysela

- Anomaly detection
  - to stop the simulation early
  - Small case that would still show anomalies: 128x265x32x16x8
  - Anomaly detection can likely be done independently per process
- Deep surrogate (full or partial) of Gysela:
  - Physics informed NN (pure or augmented with simulation data)
- **Compression:**

  Incremental iPCA
  - Prototypes on the way (WP3, WP2)
  - Discussion on how to learn the model
    - Can we use a number of early timesteps ?
    - Can the model be trained on existing runs ? data access ? or make smaller runs ?
  - For the movement Tokam2D probably enough for testing if iPCA is relevant