Elastic Ensemble Run
Data Processing with Melissa

Bruno Raffin,
DataMove
INRIA

November 2020
Ensemble Runs

• Run various instances of a given simulation code to sample its behavior in the parameter space

• A pattern common across different use-cases:
  – Sensibility Analysis
  – Data Assimilation
  – Hyperparameter search
  – Deep Surrogate training
  – Reinforcement Learning

• « Killer use case » for exascale computing ?
• A framework for large scale ensemble management run and on-lined ata processing:
  – Elastic
  – Fault tolerant
  – File free
  – Modular

https://melissa-sa.github.io/
Launcher:
  • Detect failing server with timeout
  • Detect zombies simulations with timeout

Batch Scheduler → Run Jobs

Heartbeat

Sim (i) fresh start
Sim (j) restart
Sim (k) zombie

NxM data distribution

Parallel Melissa Server:
  • Aggregate data received from clients
  • Periodic checkpoints

Job monitoring
Use Case 1: Sensibility Analysis

- **Server**: compute (iterative) statistics [Terraz et. Al SC’17]
- **Exp 1**:

  8000 simulation runs:
  - 512 cores each
  - 34,000 CPU.h

  **32 server nodes**:
  - 740 CPU.h (2.1%)

  Simulation runs 13% faster on average than when writing to disk

- **Exp 2**:
  - 80,000 simulation runs (24 cores each)
  - 271 TB of data processed on-line (and thus not saved to disk)
Use Case 2: Data Assimilation

Launcher
- Submit and monitor jobs
- Manage recovery on server or runner fault

Batch Scheduler
- Submit Jobs
- Job Status

Parallel Runner
- Run Jobs

Clients
- Propagated States
- Dynamic Connections
- Analyzed States

Parallel Server
1. Gather states from runners
2. Update states (PDAF DA engine)
3. Distribute analyzed states to runners

Checkpoint
Observations

- EnKF assimilation with Parflow sims.
- Up to 16384 members distributed on up to 400 runners running with 16,960 cores
- 2.9 TB of Data transfers per assimilation cycle

bruno.raffin@inria.fr
Use Case 3: Deep Surrogate Training

**Launcher:**
- Detect failing server with timeout
- Detect zombies simulations with timeout

**Control**
- Batch Scheduler
- Run Jobs
- Job monitoring
- Heartbeat
- Report progress

**Computations**
- Sim (i)
- Sim (j)
- Sim (k)
- NxM data distributions

Leverage the Jean-Zay Supercomputer

On-line Parallel learning (with Horovod)

bruno.raffin@inria.fr
Mesh: 1600 cells. Result after learning from 10 000 simulations
Summary

Ensemble runs: a key pattern for exascale but volumes of produced data are huge

**MELISSA**: an architecture for supporting large scale ensemble runs with on-line parallel data aggregation
- Elastic
- Fault tolerant
- File free
- Modular

[https://melissa-sa.github.io/](https://melissa-sa.github.io/)

On-going work:
- Training large deep surrogates
- Reinforcement learning

bruno.raffin@inria.fr