Adaptive multi-tier intelligent data manager for Exascale

admire-eurohpc.e

Scaling New Heights: Challenges and Opportunities in Dynamical Deep Learning for Remote Sensing on HPC Platforms

Rocco Sedona

Barcelona, 12/12/2023

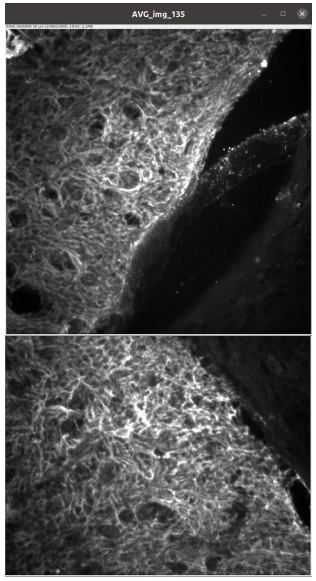
Grant Agreement number: 956748 — ADMIRE — H2020-JTI-EuroHPC-2019-1

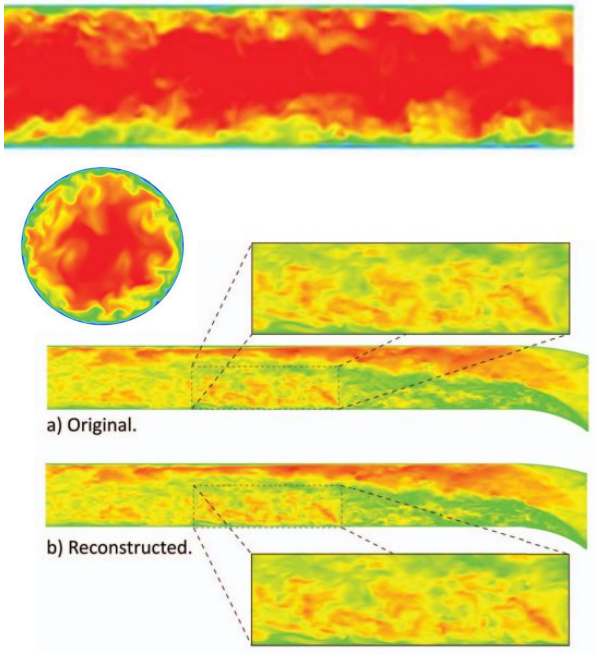
Outline

- A general overview on applications
- A remote sensing application
  - Land cover maps
  - Deep learning
    - Distributed deep learning with Horovod
    - Elastic Horovod
    - Challenges
- Conclusions

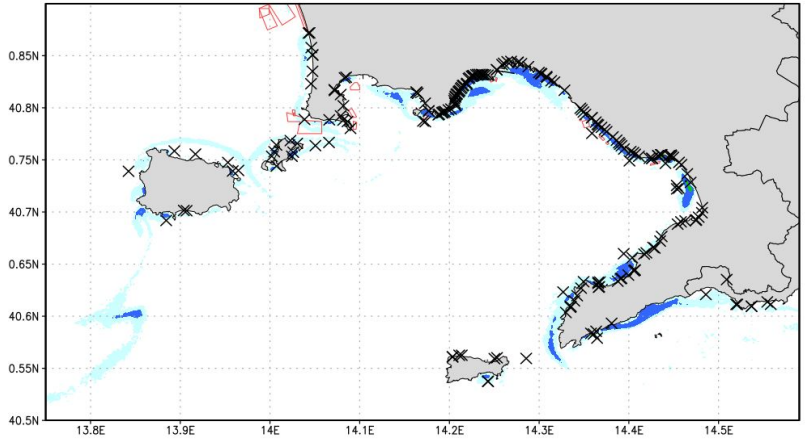
Overview

Life sciences





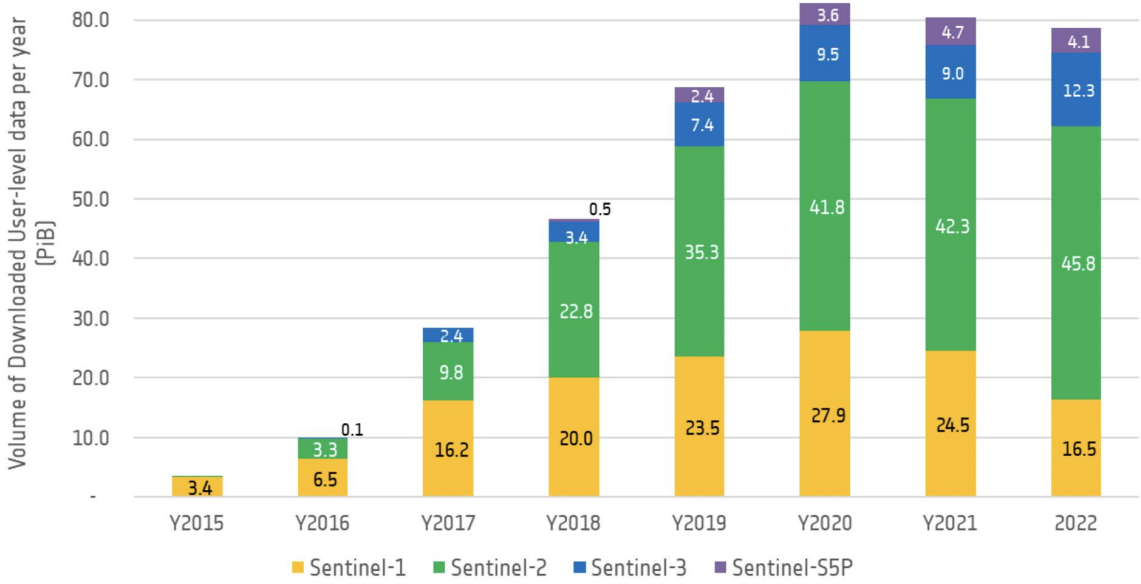
Nek5000 - computational fluid dynamics



Environmental application

Remote sensing in the era of exascale

- Many satellite missions continually acquire data.
  - Earth observation applications greatly benefit from it.
- Large availability of open-source data
- Copernicus published a total data volume of 6.64 PiB in 2022.



Downloaded data volume, from the Copernicus Sentinel Data Access Annual Report 2022

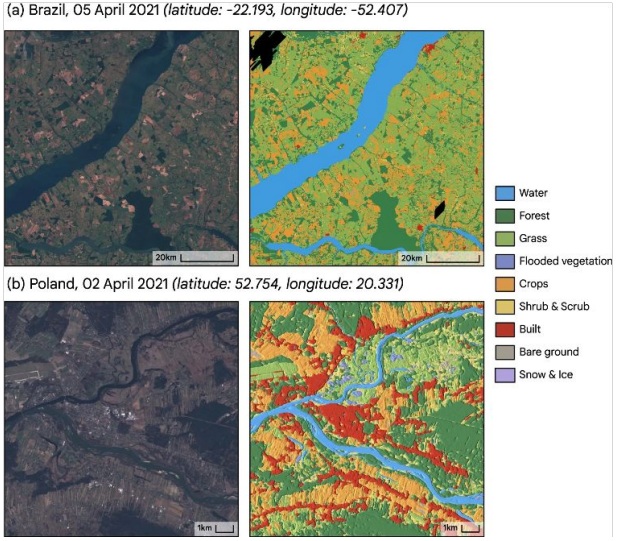
Remote sensing the era of exascale

- Larger datasets -> larger models for better accuracy.
- Deep learning is computationally intensive.

Growth of model size:
https://huggingface.co/blog/large-language-models

Land cover maps

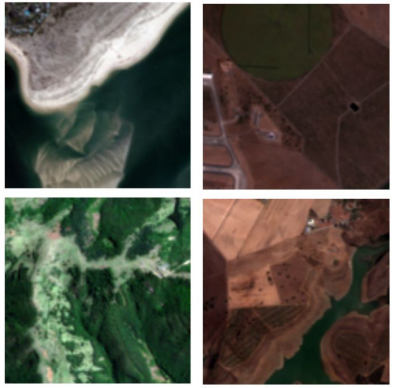
- Land Cover Maps:
  - Show spatial information on Earth's surface coverage types (e.g., forests, grasslands, croplands, lakes, wetlands).
  - Represent different physical classes of the Earth's surface.
- Dynamic Land Cover Maps:
  - Include transitions of land cover classes over time.
  - Capture and depict changes in land cover.
  - Can be frequently updated using satellite data.



Land cover maps from Dynamic World. Brown, C.F. *et al.* Dynamic World, Near real-time global 10 m land use land cover mapping. *Sci Data* **9**, 251 (2022).

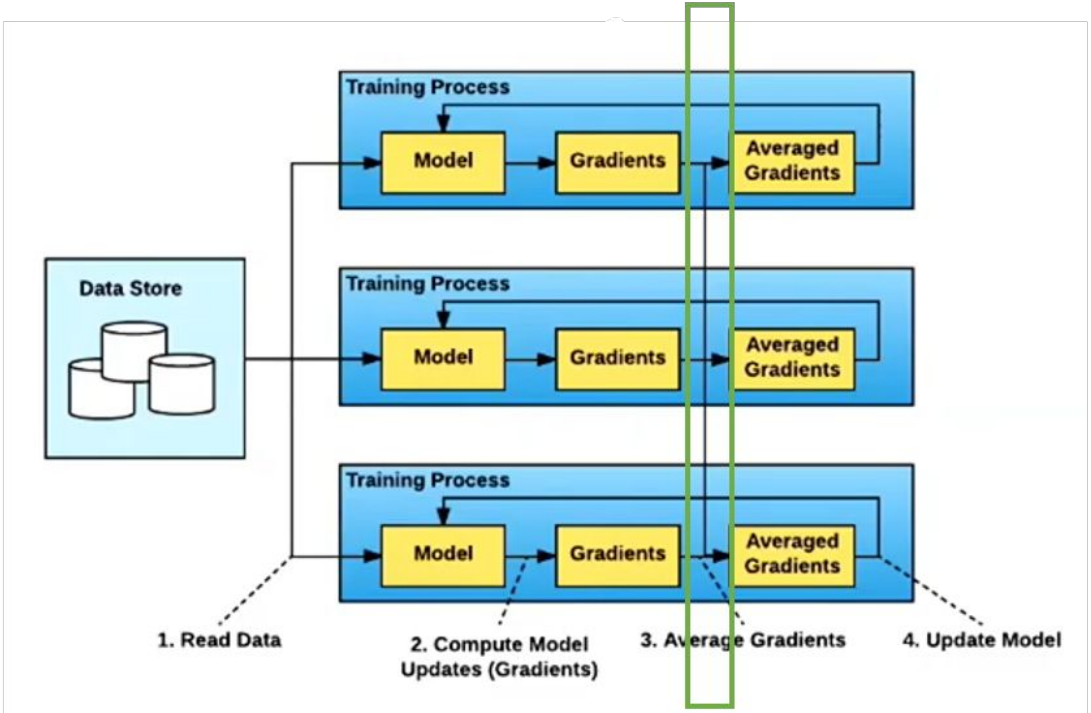
A remote sensing application

- Use case:
  - Land cover classification.
  - BigEarthNet:
    - Multispectral data acquired by Sentinel-2.
    - More than 500.000 patches.
    - Multi-class problem.
  - Convolutional Neural Network.
  - Scaling training with Horovod.



R. Sedona, G. Cavallaro, J.Jitsev, A. Strube, M. Riedel and J. A. Benediktsson, ”Remote Sensing Big Data Classification with High Performance Distributed Deep Learning”, journal of Remote Sensing (MDPI), 2019

Data parallelism



Data parallelism in deep learning:

- Parallel processing of data across devices: Involves distributing and processing different batches of data simultaneously across multiple devices.
- Model replication:
  - Replicating model across devices.

Ben-Nun, T., & Hoefler, T. (2018). Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis (Version 2). arXiv. https://doi.org/10.48550/ARXIV.1802.09941

Horovod

- Horovod is a framework for data parallelism.
  - Works on top of MPI, NCCL and Gloo.
  - On top of Tensorflow and PyTorch
- **Elastic Horovod:**
  - **Allows dynamical training with a changing number of resources.**
  - Gloo.

Sergeev, A., & Del Balso, M. (2018). Horovod: fast and easy distributed deep learning in TensorFlow (Version 3). arXiv. https://doi.org/10.48550/ARXIV.1802.05799

Malleability

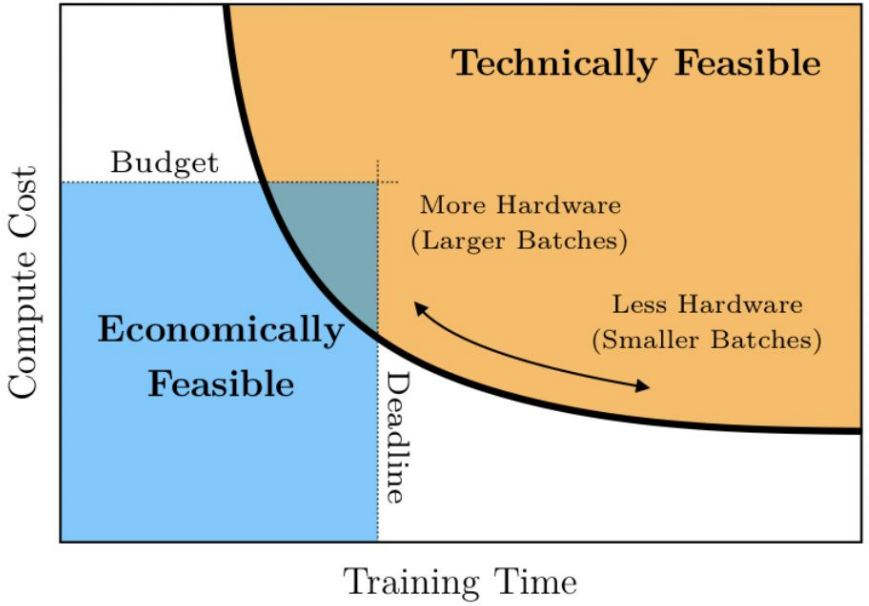
- Malleability is the ability of a computational job to be dynamically resized or reconfigured during its execution.
- Importance in HPC:
  - Optimizing Resource Utilization:
    - Allows for dynamic adjustment of compute and storage resources based on evolving application and system requirements.
  - Performance Advantages:
    - Offers obvious performance advantages over traditional static allocations in high-performance computing (HPC) environments.
  - Enhancing Adaptability:
    - Facilitates adaptability to changing workloads, optimizing the efficiency of resource usage.
  - Balancing Computation and Storage Performance:
    - Aims to balance computation and I/O performance through coordinated scaling.

Challenges

- Scientific challenges:
  - Up to which batch size can we scale?
  - When is it safe to increase / decrease job size?
- Technical challenge
  - How to dynamically shrink and expand resources during training?

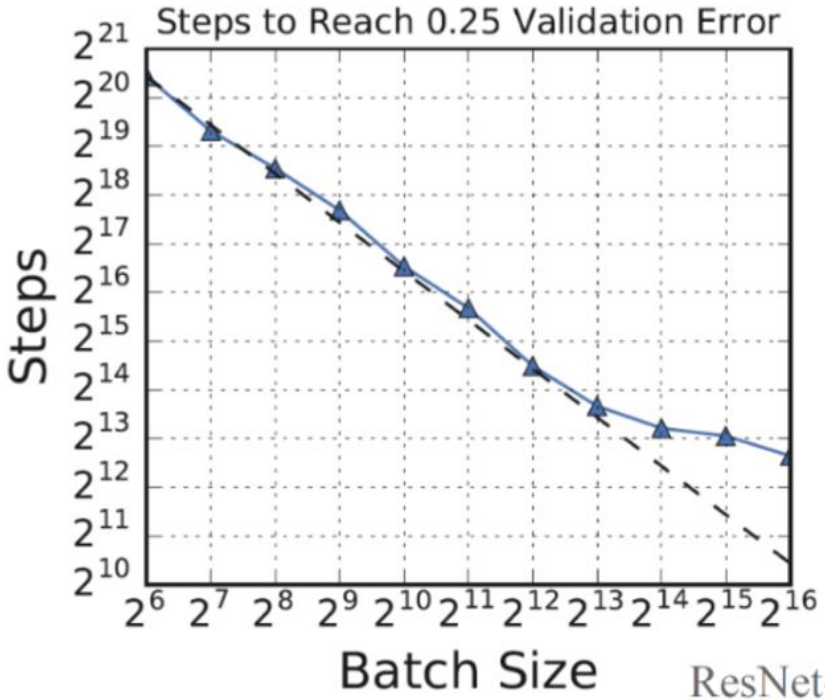
Trade-off

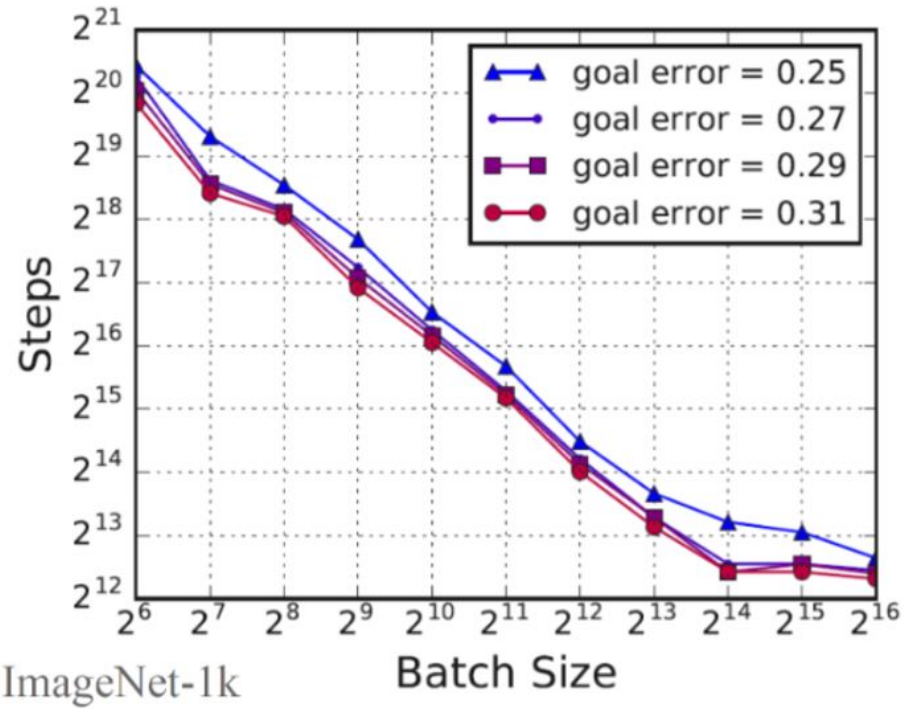
- Small batch size?
  Longer training
- Large batch size?
  Higher compute cost



McCandlish, S., Kaplan, J., Amodei, D., & Team, O.D. (2018). An Empirical Model of Large-Batch Training. *ArXiv, abs/1812.06162*.

Large batch sizes





ResNet-50, ImageNet-1k

Shallue, C. J., Lee, J., Antognini, J., Sohl-Dickstein, J., Frostig, R., & Dahl, G. E. (2018). Measuring the Effects of Data Parallelism on Neural Network Training. arXiv. https://doi.org/10.48550/ARXIV.1811.03600

When can we increase / decrease allocated resources?

- Typical scenario:
  - Application needs smaller batch size (less resources) at the initial phase of training*.
  - Batch size can be increased after a few epochs to speed up training.
  - Once an acceptable accuracy has been reached, training can be put in lower priority and be allocated to less resources.

* McCandlish, S., Kaplan, J., Amodei, D., & Team, O.D. (2018). An Empirical Model of Large-Batch Training. *ArXiv, abs/1812.06162*.

Attempts to dynamically allocate nodes

1. srun -N 2 horovodrun --verbose --log-level TRACE --gloo-timeout-seconds 120 -np 2 --min-np 2 --max-np 2 --network-interface "ibs1f1" --host-discovery-script ***path_to_discovery_script*** --slots 1 python ***./application***

```
#!/bin/bash
NODELIST=$(scontrol show hostnames "$SLURM_JOB_NODELIST")
printf '%s\n' "${NODELIST[@]}"
```

host-discovery-script.sh

Attempt to dynamically allocate nodes

2. Replacing host discovery script with list of nodes.

```
#!/bin/bash
printf "cascadelake-001\n"
printf "cascadelake-002\n"
```

3. Now modify the list of nodes

```
#!/bin/bash
printf "cascadelake-001\n"
printf "cascadelake-003\n"
```

Node changed

4. SLURM does not permit that!

<stderr>:Access denied by pam_slurm_adopt: you have no active jobs on this node

5. What is possible? Reducing the number of nodes using scontrol.

scontrol update jobid=slurm_job_id Numnodes=shrinked_number_of_nodes

Technical solutions

Expanding the training.

- Checkpointing:
  - Straightforward “start&stop” solution.
  - Longer time to convergence.

Preferred solution

- Allocate multiple jobs and use a controller to dynamically update the hostfile.
  - Elastic Horovod automatically shrinks or expands the application.

Conclusions

- A diverse set of applications in a constantly evolving scenario.
  - ADMIRE tools are validated on real use cases.
- Several technical and scientific challenges remain open.

¡Muchas Gracias! Moltes Gràcies! Thank You!

 por su atención y participación.

 per la vostra atenció i participació.

 por your attention and participation.

Contacto / Contacte / Contact: r.sedona@fz-juelich.de


“This is the end”*
“Apocalypse Now” by Francis Ford Coppola, 1979
* "The End" by the Doors, 1967