Deploying Deep Learning Workloads on the Computing Continuum

Thomas Bouvier    Alexandru Costan    Gabriel Antoniu
first.last@inria.fr

KerData team © Inria, France

December 16, 2021
13th JLESC Workshop
Deep Learning (DL)

- Outperform human experts in many domains
- Build knowledge by **training** deep neural networks (DNNs) over large datasets
Deep learning (DL)

- Outperform human experts in many domains
- Build knowledge by *training* deep neural networks (DNNs) over large datasets

Computing Continuum (CC)

- Shift from centralized clouds towards *multi-tier processing elements* (PEs)
- Span over *clouds and fog* mini-clusters to *edge* devices
Training DNNs, the usual way

Distributing the DNN training is achieved by splitting different dimensions

Popular parallel strategies include:

- **Data parallelism**: distributing training examples among PEs
- **Model parallelism**: partitioning the DNN along its output/input channels
- **Layer parallelism**: partitioning the DNN along its depth
  - **Pipeline parallelism**: overlapping computation between one layer and the next layer
- **Hybrid parallelism**: a combination of the above strategies
Training DNNs usually implies that the computing environment is homogeneous.

The CC emerges in the context of the convergence of big data analytics, AI systems and HPC simulations.
Problem statement

Training DNNs usually implies that the computing environment is homogeneous.

The CC emerges in the context of the convergence of big data analytics, AI systems and HPC simulations.

How to perform **efficient DNN training** on the Computing Continuum?
Challenges of the Computing Continuum

The CC brings heterogeneity to the whole stack

**Infrastructure**

- **Network heterogeneity**: low-speed, high-latency wide-area networks connecting geo-distributed PEs
- **Compute heterogeneity**: CPUs vs. GPUs
Challenges of the Computing Continuum

The CC brings heterogeneity to the whole stack

Infrastructure

- **Network heterogeneity**: low-speed, high-latency wide-area networks connecting geo-distributed PEs
- **Compute heterogeneity**: CPUs vs. GPUs

Middleware

- Batch vs. **stream processing**: real-time events introduce extra uncertainty

Application

- **Parallel computing**: some limitations are inherent to the parallel strategies themselves
Deploying DL workloads on the CC requires to distinguish deterministic limitations from temporary bottlenecks.

**Deterministic limitations**
- Before the deployment of the application, some limitations are predictable.

**Temporary bottlenecks**
- At runtime, temporary slowdowns can be detected in real-time.
Deterministic limitations

Examples of deterministic limitations include:

- Shortcomings of the parallel strategies themselves
- Heterogeneity of the CC
  - Unbalanced compute resources of PEs
  - Slow network links
Deterministic limitations

Examples of deterministic limitations include:

- Shortcomings of the parallel strategies themselves
- Heterogeneity of the CC
  - Unbalanced compute resources of PEs
  - Slow network links

A static analysis can help predict and overcome these limitations:

- **Combination of parallel strategies**: hybrid parallelism
- **Operator placement**: estimate the performance of PEs to place operators across the continuum accordingly
Deterministic limitations

**Figure 1:** A **middleware layer** to distribute DNN training on the CC
Temporary bottlenecks

Examples of temporary bottlenecks include:

- Communication bottlenecks
- I/O bottlenecks
Temporary bottlenecks

Examples of temporary bottlenecks include:

- Communication bottlenecks
- I/O bottlenecks

**Real-time monitoring** can help detect and mitigate these bottlenecks:

- **Asynchronous communication methods**: skip iterations so that the effect of slower PEs is mitigated
- **Operator reconfiguration**: reorganize the operator placement across the continuum in case of a persistent bottleneck
Evaluation

Deploy experiments at scale on the Grid’5000 testbed

E2Clab as a deployment/performance analysis tool

Optimize accuracy, makespan and fairness objectives

Validate with synthetic benchmarks and real life applications: digital twins
A first goal would be to train ResNet models on CIFAR/ImageNet datasets in the context of the CC

**Continual learning**

- "Incremental deep learning"
- A natural approach on the CC, where sensors generate data over time
- Apply the devised techniques to continual learning training
Ongoing collaboration with Bogdan Nicolae from ANL on continual learning:

- DNN architecture search
- State replication for DNNs
- Data parallel continual learning

Contact: thomas.bouvier@inria.fr