VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

Optimizations for CPUs, GPUs and numerical stability

Georg Zitzlsberger georg.zitzlsberger@vsb.cz

29-11-2022

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS

Agenda

CPU Optimizations

GPU Optimizations

Numerical Stability

Summary

CPU Optimizations

- Intel offers own version of scikit-learn via Intel Distribution for Python
- It builds on Intel's performance libraries:
  - Intel Data Analytics Acceleration Library (Intel DAAL):
    Replaces some algorithms/tools from scikit-learn
  - Intel Math Kernel Library (Intel MKL):
    Backend to NumPy and SciPy
- Intel DAAL substitutions are turned off by default, enable with:

```
import daal4py.sklearn
daal4py.sklearn.patch_sklearn()
```

. . . or

```
$ USE_DAAL4PY_SKLEARN=YES python ...
```

- Don't use toy sets to test speedups; there is some overhead involved

GPU Optimizations

- From the scikit-learn FAQ:
  Q: Will you add GPU support?
  A: No, or at least not in the near future. ...

- There are two alternatives:
  - H2O4GPU from H2Oai:
    - Inherits *scikit-learn* tools/algorithms
    - Adds (NVIDIA) GPU support to some; falls back to CPU if not available
    - Builds on existing GPU solvers
  - cuML from NVIDIA:
    - Part of NVIDIA's RAPIDS Open GPU Data Science toolkit
    - Run ML tasks on GPUs w/o programming in CUDA
    - API matches the one of scikit-learn

H2O4GPU

Advantages:

- Selected algorithms can significantly benefit from higher GPU throughput:
  - GLM: Lasso, Ridge Regression, Logistic Regression, Elastic Net Regulariation
  - KMeans
  - Gradient Boosting Machine (GBM) via XGBoost
  - Singular Value Decomposition (SVD) + Truncated Singular Value Decomposition
  - Principal Components Analysis (PCA)
- Transparently falls back to scikit-learn implementation (Intel DAAL)
- Supports multiple GPUs
- Offers more tools than scikit-learn

Disadvantages:

- Data transfers between host and GPU are limiting the speedup (or even slow down)
- Lacks behind scikit-learn and has smaller community

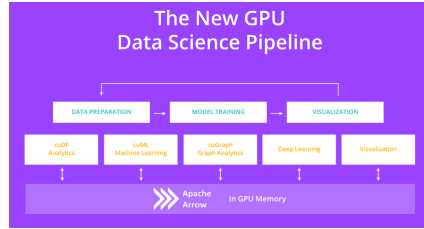
H2O4GPU Example

```
import numpy as np
from sklearn.decomposition import TruncatedSVD as tsvd
from h2o4gpu.solvers import TruncatedSVDH2O

#Set up matrix (m x n)
np.random.seed(1234)
m=200000; n=1000
X = np.random.rand(m, n)
k=10 # number of components

# scikit-learn
model = tsvd(n_components=k, algorithm="arpack")
model.fit(X)

# Same with H2O4GPU
h2o4gpu_model = TruncatedSVDH2O(n_components=k)
h2o4gpu_model.fit(X)
```

cuML

Advantages:

- Supports multiple GPUs and multiple nodes using Dask
- Offers more tools than scikit-learn
- Part of a Apache Arrow Data Science Pipeline:



(Image: NVIDIA)

Disadvantages:

- Data transfers between host and GPU are limiting the speedup (or even slow down)
- Vendor specific and not official part of scikit-learn

cuML Example

```
import cudf
import cupy
import matplotlib.pyplot as plt
from cuml.cluster import KMeans as cuKMeans
from cuml.datasets import make_blobs
from sklearn.cluster import KMeans as skKMeans
from sklearn.metrics import adjusted_rand_score

n_samples = 100000
n_features = 2
n_clusters = 5
random_state = 0

device_data, device_labels = make_blobs(n_samples=n_samples, n_features=n_features,
                                        centers=n_clusters, random_state=random_state,
                                        cluster_std=0.1)
device_data = cudf.DataFrame(device_data)
device_labels = cudf.Series(device_labels)

# Copy dataset from GPU memory to host memory.
# This is done to later compare CPU and GPU results.
host_data = device_data.to_pandas()
host_labels = device_labels.to_pandas()

kmeans_sk = skKMeans(init="k-means++", n_clusters=n_clusters,
                     n_jobs=-1, random_state=random_state)
kmeans_sk.fit(host_data)

kmeans_cuml = cuKMeans(init="k-means||", n_clusters=n_clusters,
                       oversampling_factor=40, random_state=random_state)
kmeans_cuml.fit(device_data)
```

Numerical Stability

- Optimizations can cause changes in numerical results:
  - Change of FP operation order
  - Different operations (e.g. use of FMA)
  - Rounding modes
  - ...

  Example: $(a + b) + c \neq a + (b + c)$

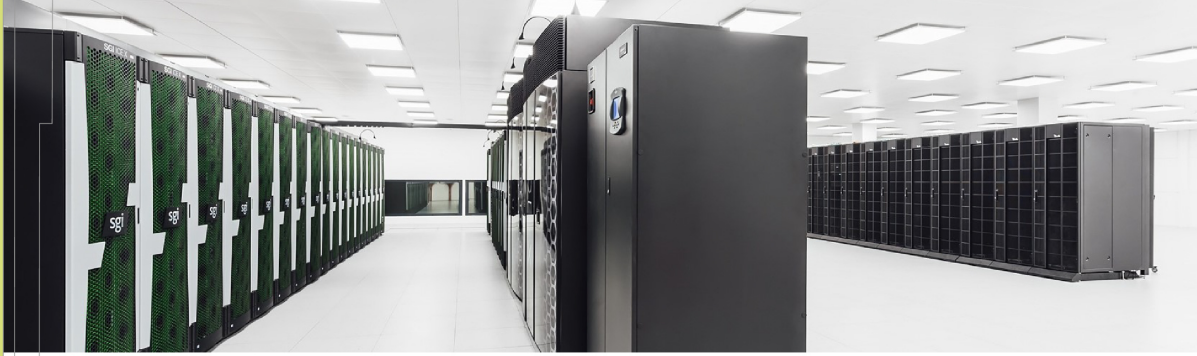
  $2^{-63} + 1 + -1 = 2^{-63}$: mathematical result
  $(2^{-63} + 1) + -1 \approx 0$: correct IEEE result
  $2^{-63} + (1 + -1) \approx 2^{-63}$: correct IEEE result

- The scikit-learn community did not pick Intel's version as default
- If you like (Intel) optimizations, you need to enable them explicitly
- Selectively use H2O4GPU:
  Uses Intel DAAL as CPU fallback before switching to scikit-learn!
- Even if results are (bitwise) the same on one system, another system might execute different code paths!

Summary

- Some algorithms/tools from scikit-learn exist in optimized versions for CPUs and GPUs
- They require sufficient workload/minimized data transfer to benefit
- H2O4GPU, cuML and Intel DAAL offer even more algorithms but with their own API
- **Be aware:**
  They are different implementations and can lead to different results[1]!

[1] If bitwise reproducibility is required.



VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER

IT4Innovations National Supercomputing Center

VŠB – Technical University of Ostrava
Studentská 6231/1B
708 00 Ostrava-Poruba, Czech Republic
www.it4i.cz

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS