Distributed Deep Learning by Horovod

Jozsef Kovacs, SZTAKI
jozsef.kovacs@sztaki.hu
NEANIAS: Services, Infrastructures, Communities & Business

NEANIAS - Novel EOSC Services for Emerging Atmosphere, Underwater & Space Challenges
Call: INFRAEOSC-02-2019
Project ID: 863448
Duration: 36 months
Finish: 31.10.2022

› Address community-specific needs for underwater, atmosphere and space research sectors
› Onboard communities to the Open Science, EOSC and interdisciplinary research era
› Nurture new business opportunities
› Power-up EOSC
Neanias project structure – Core services

WP6 services
  › C1 Open-science lifecycle support services
  › C2 EOSC, RI and cloud integration enabling service
  › C3 Artificial Intelligence service
  › C4 Visualisation services
AI Gateway and its integrated components

› Basic core services
› Development and training of ML models
› Model serving of ML models
Horovod

› Open-source distributed deep learning framework from Uber
› Supports TensorFlow, Keras, PyTorch, Apache MXNet and Spark
› Provide an easy-to-use framework for distributed training
  – Execute on hundreds of GPUs with just a few lines of additional code
  – Great scaling efficiency
› Data parallel execution
  – Training data is divided into subsets
  – Train the same replicated model on each node
  – Model parameters are synchronised between the workers
› Ring-Allreduce strategy
  – Horovod utilizes Message-Passing Interface (MPI)
  – Each node communicates with two of its peers 2*(N-1) times
  – NVIDIA NCCL 2.0 for intra-node communication
Workflow Overview
Monitoring with Prometheus Grafana stack
Integration into NEANIAS AI-Gateway

Server Options

- ML model development using Tensorflow/Keras
  Environment for ML model development supported by Tensorflow and Keras Python ML libraries

- Distributed training of ML models using Horovod
  Environment for Distributed Deep Learning by Horovod. IMPORTANT: You need to request a personal cluster before choosing this environment at eosc-horovod@sztaki.hu

- Serving ML models using BentoML
  Environment for establishing a service by BentoML with a ML model behind

- ADAM API
  Environment for using ADAM API

- ASTRO ML
  Environment for using MRCNN

- TIRAMISU
  Environment for using Tiramisu modelling

Options for mounting remote storage

Start
Intro page

ai-gateway.neanias.eu
Al-Gateway Environments

Server Options

- ML model development using Tensorflow/Keras
  Environment for ML model development supported by Tensorflow and Keras Python ML libraries

- Distributed training of ML models using Horovod
  Environment for Distributed Deep Learning by Horovod. IMPORTANT: You need to request a personal cluster before choosing this environment at eosc-horovod@sztaki.hu

- Serving ML models using BentoML
  Environment for establishing a service by BentoML with a ML model behind

- ADAM API
  Environment for using ADAM API

- ASTRO ML
  Environment for using MRONN

- TIRAMISU
  Environment for using Tiramisu modeling

Options for mounting remote storage

Start
Starting the single-user server using SSHSpawner
JupyterLab Environment
File Browser - /horovod directory
Cluster Summary

*************** Horovod Cluster Specification ***************
---
- harovod-master | IP: 192.168.0.181
- CPU
  - Number of GPUs: 1
  - GRID V1000K-8C, 1532 MB
- CPU
  - Number of CPUs: 4
  - Intel Xeon Processor (Cascade Lake)
- Memory (GB)
  - total used free shared buff/cache available
    15 0 0 0 0 0 0
- Storage
  - Disk /dev/sda: 32 GB, 34359738368 bytes, 67108864 sectors
  - Disk /dev/sdb: 224 GB, 24951016976 bytes, 669762048 sectors
- harovod-worker-1 | IP: 192.168.0.174
- GPU
  - Number of GPUs: 1
  - GRID V1000K-8C, 1532 MB
- CPU
  - Number of CPUs: 4
  - Intel Xeon Processor (Cascade Lake)
- Memory (GB)
  - total used free shared buff/cache available
    15 0 0 0 0 0 0
- Storage
  - Disk /dev/sda: 32 GB, 34359738368 bytes, 67108864 sectors
Host list for training

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>192.168.0.181 slots=1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>192.168.0.174 slots=1</td>
<td></td>
</tr>
</tbody>
</table>
Example training script

```python
# Copyright 2019 Uber Technologies, Inc. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
#-----------------------------------------------------
# Imports
import sys
import tensorflow as tf
import horovod
import horovod.tensorflow as hvd

def main():
    # Horovod: initialize Horovod.
    hvd.init()
    # Horovod: pin GPU to be used to process local rank (one GPU per process)
    gpus = tf.config.experimental.list_physical_devices('GPU')
    for gpu in gpus:
        tf.config.experimental.set_memory_growth(gpu, True)
    if gpus:
        tf.config.experimental.set_visible_devices(gpus[hvd.local_rank()], 'GPU')
    (mli_images, mli_labels) = 
    tf.keras.datasets.mnist.load_data(path='mnist-16384' % hvd.rank())
    dataset = tf.data.Dataset.from_tensor_slices(
        (tf.cast(mli_images[..., tf.newaxis] / 255.0, tf.float32),
         tf.cast(mli_labels, tf.int64)))
    dataset = dataset.repeat().shuffle(10000).batch(128)
    mli_model = tf.keras.Sequential([
        tf.keras.layers.Conv2D(32, [3, 3], activation='relu'),
        tf.keras.layers.Conv2D(64, [3, 3], activation='relu'),
    ])
```
Onboarding to the EOSC Marketplace

Distributed Deep Learning by Horovod

Providing researchers a reliable platform designed for performing distributed deep learning operations with great scaling efficiency.

The Distributed Deep Learning by Horovod service aims to provide the infrastructure, resources and libraries to its users in order to perform effective distributed training of deep neural networks.

Horovod is a distributed training framework with the main goal of enabling the simple and effective distribution of deep learning operations. While requiring just a few lines of additional code (compared to sequential version), Horovod enables training to be performed across possibly hundreds of CPUs, with great efficiency.
Access modes

Limited-time demo
- Gain short term access to a demo cluster
- Hosted on ELKH Cloud
- 4 GPU enabled nodes

Request deployment on EOSC resources
- Long term access
- Exact period length and node count is up to negotiation

Self-hosted
- User manual
- Technical consultation
Thank you for your attention!

József Kovács, SZTAKI
jozsef.kovacs@sztaki.hu

Follow us:

http://www.neanias.eu
https://twitter.com/Neanias_eu
https://www.facebook.com/neanias.eu/
https://www.linkedin.com/groups/13786081/

Novel EOSC Services for Emerging Atmosphere, Underwater & Space Challenges