The **Helmholtz Analytics Toolkit**

or:

![HeAT Logo]

D. Coquelin, C. Comito, M. Götz, B. Hagemeier, P. Knechtges, K. Krajsek, M. Siggel

Member of the Helmholtz Association

April 15, 2019
What is HeAT?

The Helmholtz Analytics Toolkit

- Data analytics framework for transparent distributed computation
- Build on top of PyTorch - written in Python
- Part of the Helmholtz Analytics Framework (HAF)
- Project start: May 2018
- Developed in the open:
  - https://github.com/helmholtz-analytics
  - https://pypi.org/project/heat
- Liberally licensed: MIT
- Designed for extreme data scales
HAF

Who is involved?

Earth System Modelling

Research with Photons

Aeronautics and Aerodynamics

Neuroscience

Structural Biology

Member of the Helmholtz Association
Where Do I Fit In?

Earth Systems Modeling:
- Terrestrial Systems Modeling Platform (TerrSysMP)
- Simulation Laboratories
  - SLTS / CSLTS
### HAF

**What are the goals?**

<table>
<thead>
<tr>
<th>Scientific Big Data Analytics:</th>
<th>• Develop and expand on Methodologies and tools for problems of the highest data and computational complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeAT:</td>
<td>• Foster data science in HAF&lt;br&gt;• Develop and exploit the Helmholtz Data Federation (HDF)</td>
</tr>
<tr>
<td>Use case driven co-design between -</td>
<td>• Domain scientists&lt;br&gt;• Data experts&lt;br&gt;• Infrastructure professionals</td>
</tr>
<tr>
<td>Create Data analysis Techniques -</td>
<td>• In a systematic manner&lt;br&gt;• Domain-specific as well as generalized and standardized</td>
</tr>
<tr>
<td>HAF Use Case Methodologies</td>
<td></td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><strong>Clustering</strong></td>
<td>K-means, mean shift clustering</td>
</tr>
<tr>
<td><strong>Uncertainty Quantification</strong></td>
<td>Ensemble methods</td>
</tr>
<tr>
<td><strong>Dimension Reduction</strong></td>
<td>Autoencoder, reduced order models</td>
</tr>
<tr>
<td><strong>Feature Learning</strong></td>
<td>Image descriptors, autoencoder</td>
</tr>
<tr>
<td><strong>Data assimilation</strong></td>
<td>Kalman filter, 4Dvar, particle filter/smoother</td>
</tr>
<tr>
<td><strong>Classification/Regression</strong></td>
<td>Random forest, CNN, SVM</td>
</tr>
<tr>
<td><strong>Modelling</strong></td>
<td>Fiber tractography, point processes</td>
</tr>
<tr>
<td><strong>Optimization techniques</strong></td>
<td>L-BFGS, simulated annealing</td>
</tr>
<tr>
<td><strong>Hyper-parameter Optimization</strong></td>
<td>Evidence framework, grid search</td>
</tr>
<tr>
<td><strong>Interpolation</strong></td>
<td>Radial basis function, kriging</td>
</tr>
<tr>
<td><strong>Data mining</strong></td>
<td>Frequent itemset mining</td>
</tr>
</tbody>
</table>
Why is HeAT needed?

Data scale + TSMP

- Extreme data scales in modern sciences
  - 50 GB of data generated daily by simulations in TSMP

- Extreme-scale data mandates data distribution across computing nodes
  - Pro: more computing power
  - Con: communication overhead
Why is HeAT Needed?

Why not use an existing data analysis framework?

<table>
<thead>
<tr>
<th>Framework</th>
<th>Spark</th>
<th>BigDL</th>
<th>PyTorch</th>
<th>MXNet</th>
<th>TensorFlow</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPI</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>GPU</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ML</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ND-Tensors</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Transparent</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Distributed</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Linear Algebra</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Why HeAT?

How does HeAT solve the problem?

- Split data into multiple PyTorch tensors
- Aim to keep the Numpy API
- Run on both CPU and GPU
- Designed for a distributed data environment
**PyTorch**

**Runs on:**
- CPU
- GPU

**Data structure:** ND-Tensor

**Operations**
- Elementwise operations
- Slicing
- Matrix operations
- Reduction
- **Automatic differentiation**

**JÜLICH**
Forschungszentrum

Member of the Helmholtz Association
HeAT

How does HeAT work?

- Runs on: CPU, GPU, or MPI
- Data structure: ND-Tensor
- Operations:
  - Elementwise operations
  - Slicing
  - Matrix operations
  - Reduction
  - Automatic differentiation
HeAT Basics

Example:

```python
import heat as ht

# construct a range tensor
>>> range_data = ht.arange(6, split=0)

rank#0  [0, 1]
rank#1  [2, 3]
rank#2  [4, 5]

>>> range_data.mean()
2.5

>>> range_data.argmax()
5
```
HeAT

Challenges

- Random number generator
  - PyTorch randn dependent on tensor size
- Matrix multiplication
- Distributed Eigenvalue solver
Summary

- HeAT developed along scientific use cases
- HeAT extends PyTorch for transparent distributed computation
- Pre-alpha phase, basic operations implemented
- Light weight data analysis methods will be build based on HeAT

https://github.com/helmholtz-analytics/heat/
Thank You