DEEP Projects

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

HELMHOLTZ
RESEARCH FOR GRAND CHALLENGES

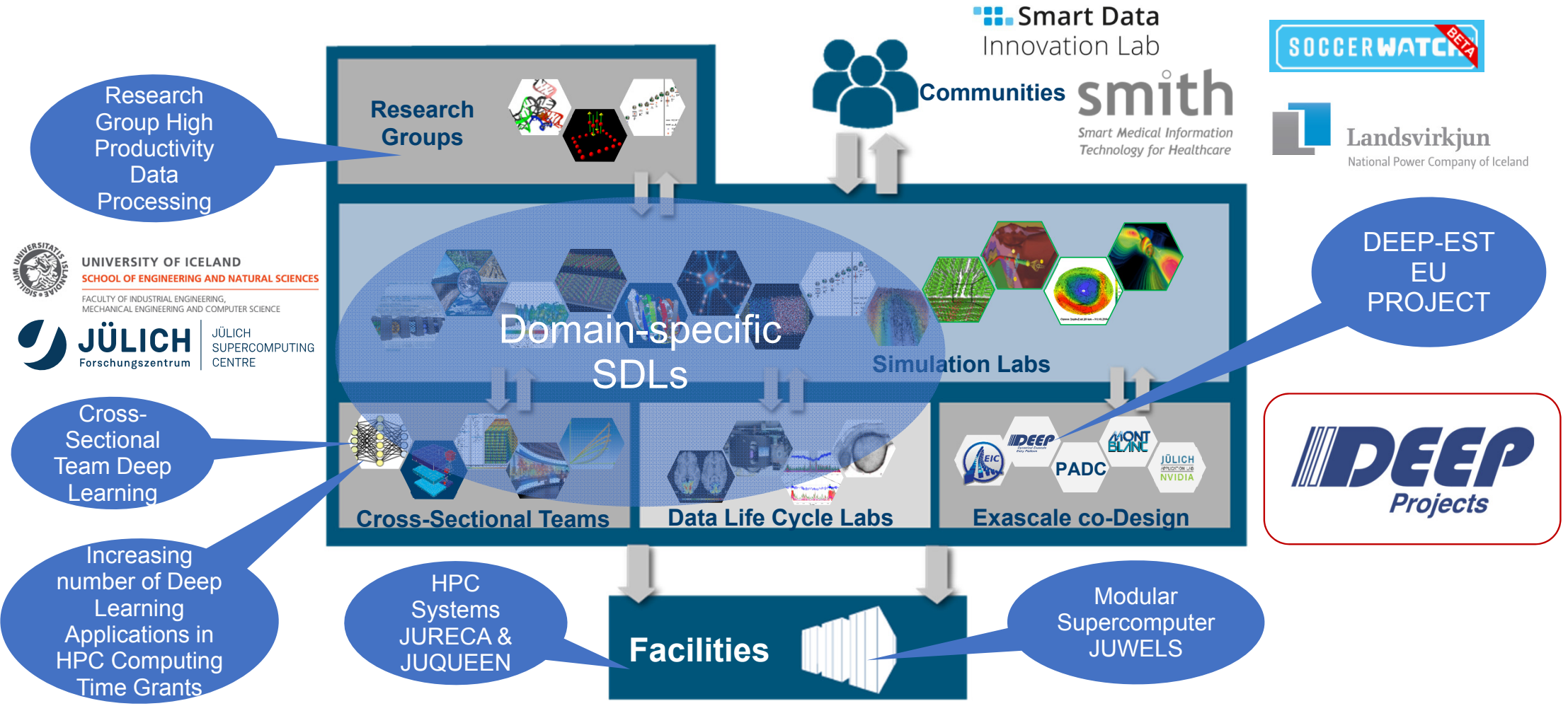
Modular Supercomputing Design supporting Machine Learning Applications

E. Erlingsson+, G. Cavallaro, A. Galonska*, M. Riedel+*, H. Neukirchen+*

**Juelich Supercomputing Centre, Forschungszentrum Juelich, Juelich, Germany*

+School of Natural Sciences & Engineering, University of Iceland, Reykjavik, Iceland

Supercomputing & Deep/Machine Learning



Big Data Analytics for Earth Science

UoI´s three applications

- **H**ighly **P**arallel **DBSCAN** (HPDBSCAN)
  - Clustering algorithm tailored for HPC and point-cloud datasets
    - *MPI, OpenMP, HDF5*
- **P**arallel **S**upport **V**ector **M**achines (PiSvM)
  - Supervised Learning algorithm
    - *MPI, HDF5**
- **D**eep **N**eural **N**etworks (DNNs)
  - Supervised and Unsupervised* Learning algorithms
    - *Tensorflow with the Keras extension*

* Work in progress

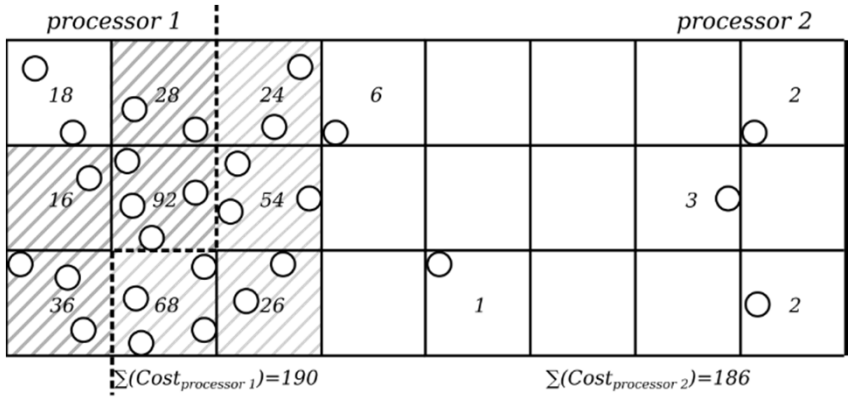
Application Analysis

HPDBSCAN – The Algorithm

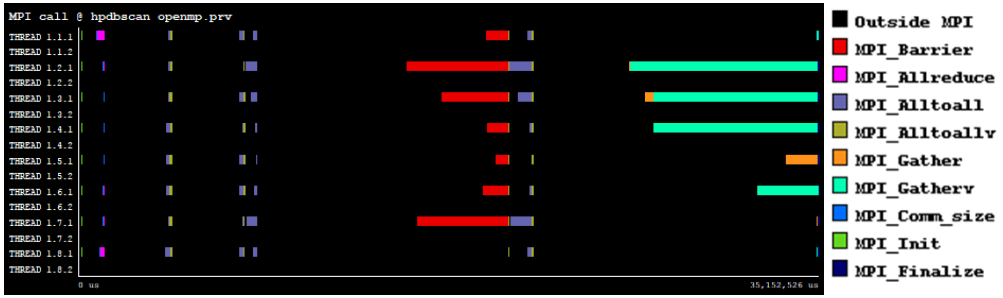
- Data-space hypergrid overlay with load balancing.
- Exhibits good scaling properties
- Fast parallel I/O (HDF5)

- Is limited by memory and the MPI collectives
- No resilience

The Challenge: Adapt HPDBSCAN for Big Data in the terabyte range with added resilience.



The hypergrid overlay and processor load-balancing



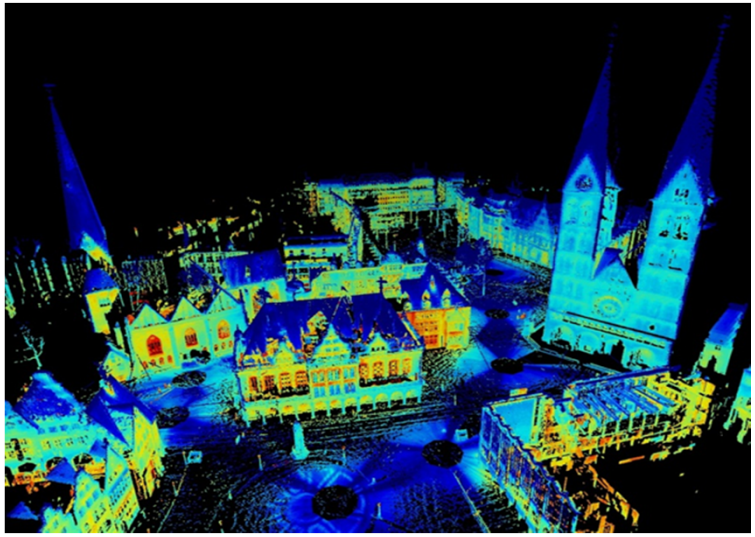
HPDBSCAN´s use of MPI collectives

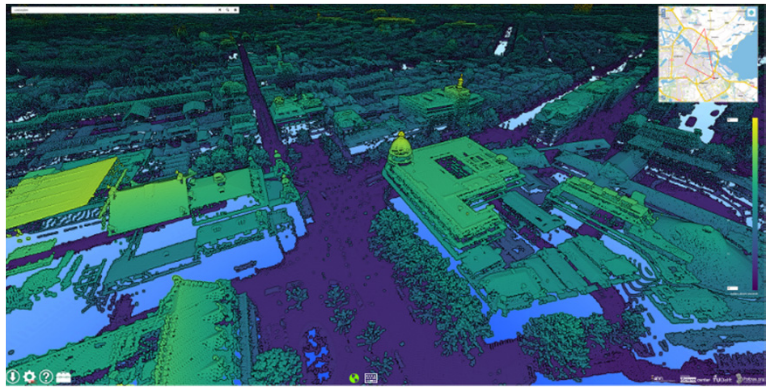
Application Analysis

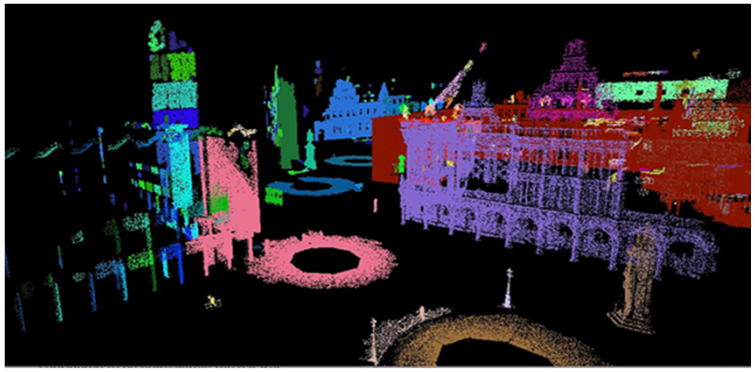
HPDBSCAN – The Datasets

Uses point-clouds acquired via 3D laser scans of cities, landmarks, or nature, using ground or airborne sensors.

- Inner city of Bremen (~2GB, too small for exascale)
- The national LiDAR dataset of the Netherlands
  - AHN-2: 640.000.000.000 points, 1.6 TB (compressed)

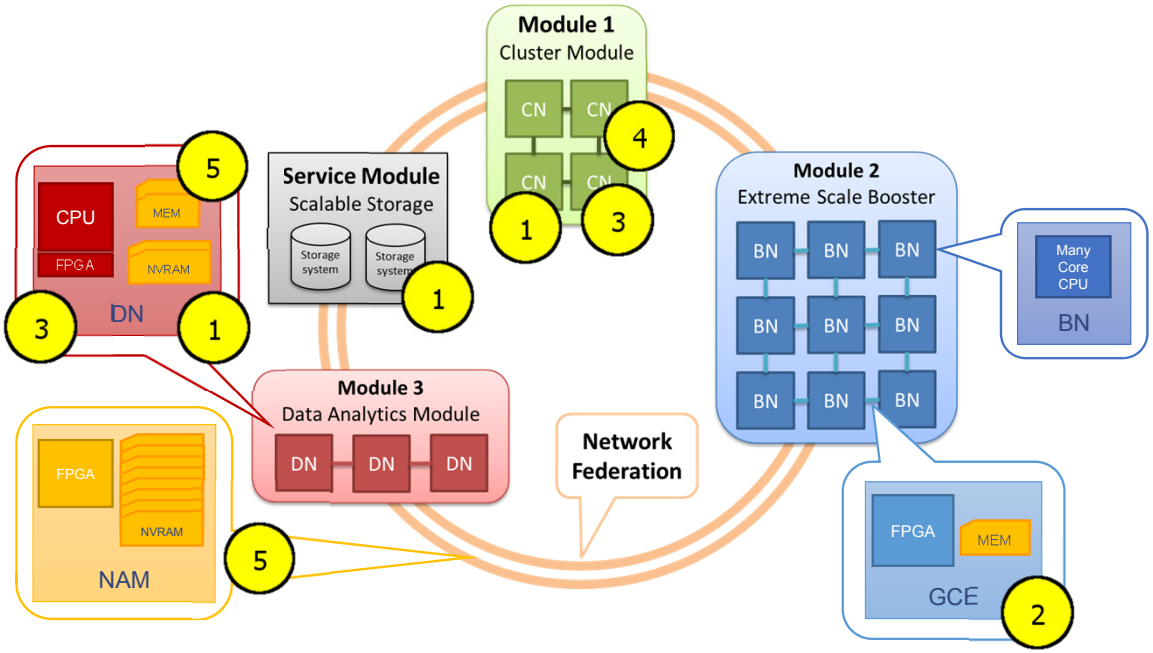






Application Analysis

HPDBSCAN – Clustering and Indexing



(1) The point cloud dataset is loaded with parallel I/O using HDF5 in the DEEP-EST Scalable Storage Service Module (SSSM) or the DEEP-EST CLUSTER module. **For Big datasets we might also consider loading the data first into the persistent memory DIMMs.**

(2) The indexing through sorting and cost heuristic small computing elements takes advantage of the FPGA in the DEEP-EST Global Collective Engine (GCE) module in combination with MPI collectives to distribute the points equally among the DEEP-EST CLUSTER

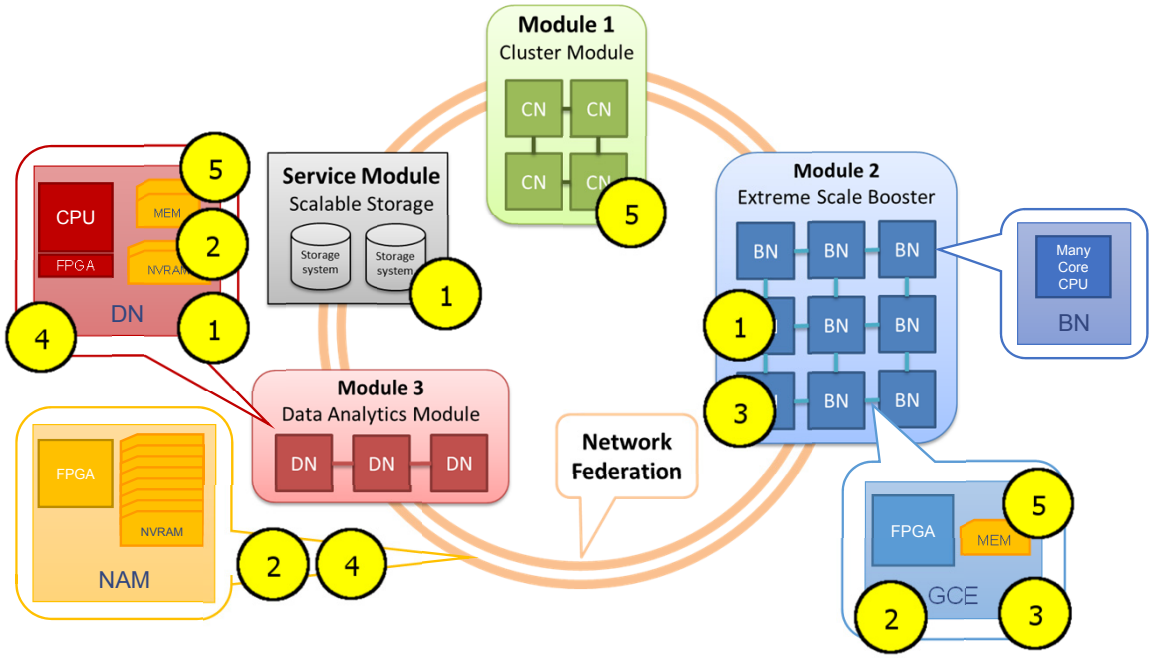
(3) Clustering with HPDBSCAN is performed on the DEEP-EST CLUSTER locally (OpenMP) for shared memory elements and the load provided by (2). **Optionally, when using NVRAM, we perform this step in the DAM.**

(4) Merging the different computed clusters on chunk edges according to specific rules using halos across nodes is performed on the
DEEP-EST CLUSTER globally (MPI)

(5) Cluster ID and noise ID are written to the HDF5 file but it can also be written to the Network Attached Memory (NAM) for further study, e.g. level of detail (LoD). **For Big Data we might also continue to use the NVRAM for this purpose.**

Application Analysis

HPDBSCAN – LoD and cLoI studies



(1) The point cloud dataset is loaded with parallel I/O using HDF5 in the SSSM and the ESB, **or the DAM**

(2) After clustering following Workload A, the data reside in the DEEP-EST NAM or **DAM NVRAM** as a fixed number of levels of importance (w.r.t. detail/scale)

(3) Selected point cloud LoD studies require continuous levels of importance using importance values of a point regarded as an added dimension to space and time using n-D space filling curves or tree structures whereby the latter may take advantage of MPI collectives using the DEEP-EST GCE enabling small computing modifications to the original clustered datasets in combinations with ESB

(4) The different data set results of the various modifications towards continuous levels of importance (cLoI) can be placed in the DEEP-EST NAM, **or DAM NVRAM** in order to take advantage of a variety of (semi-) continuous and spatio-temporal representations (e.g. zoom-in/out) for scientific studies.

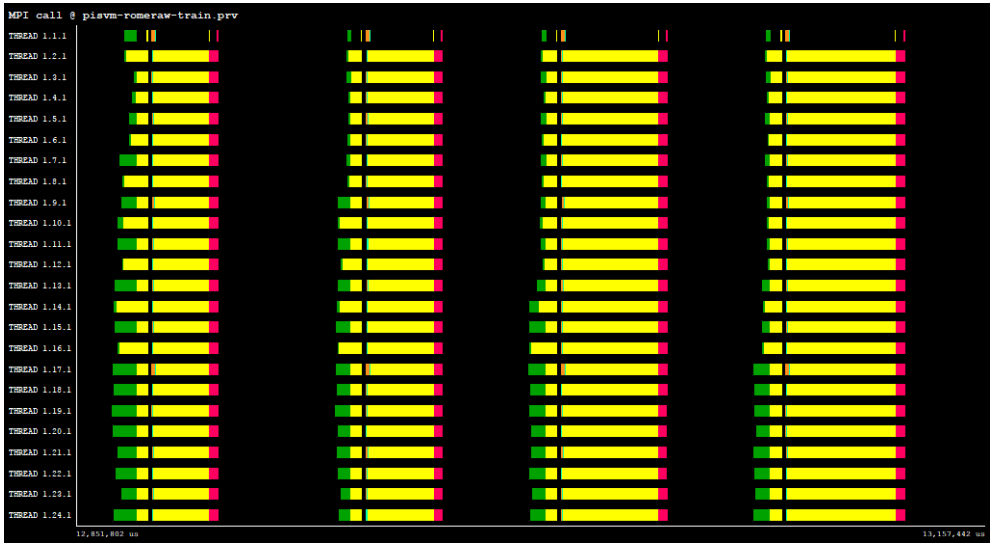
(5) Based on the CLoI data available in the DEEP-EST NAM, the DEEP-EST CM **or DAM** could use this information to re-cluster the data but on a modified dataset and/or using different parameters.

Application Analysis

PiSvM – The Algorithm

- MPI with room for improvement
- Slow Sequential I/O, no HDF5 (yet!)
- Slow training, no cascades (yet!)
- No resilience

The Challenge: Improve PiSvM´s code for performance and add resiliency.



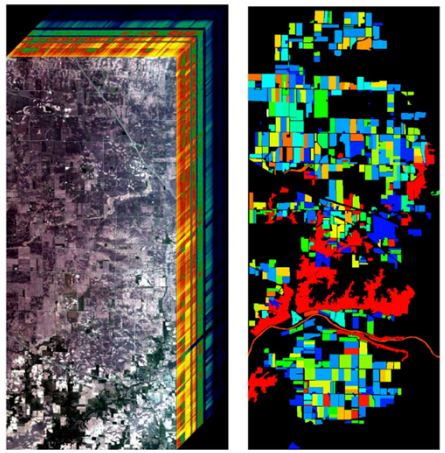
PiSvM´s MPI calls, note the load-balancing issue when comparing the first line with the rest.

Application Analysis

PiSvM & DNN – The Datasets

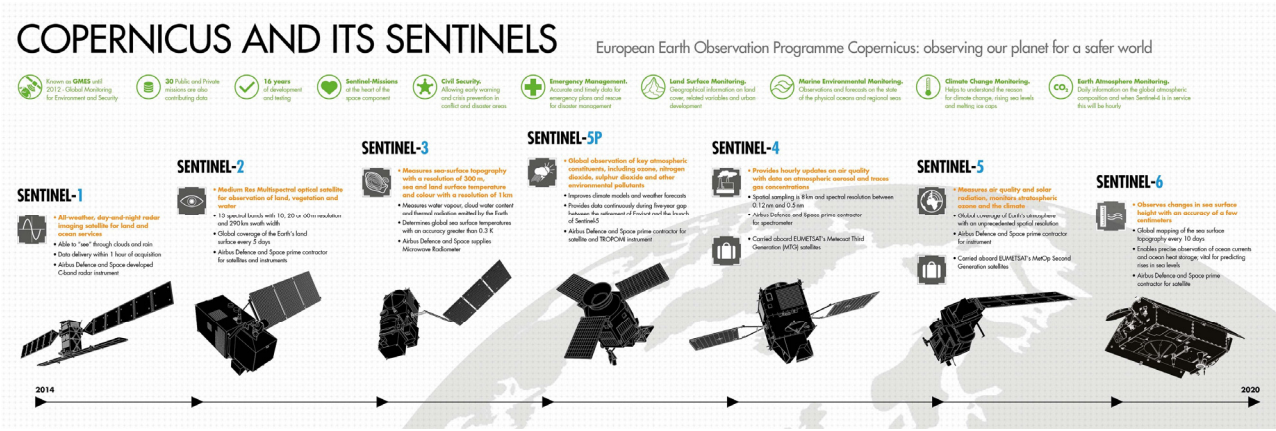
Uses hyperspectral or multispectral labelled remote-sensing datasets, mostly acquired through satellites, for supervised learning.

- The Indian Pines dataset from North-western Indiana USA gathered by AVIRIS sensor and consisting of 224 spectral reflectance bands.



The Indian Pines dataset

- The Copernicus earth observation programme (EC and ESA)

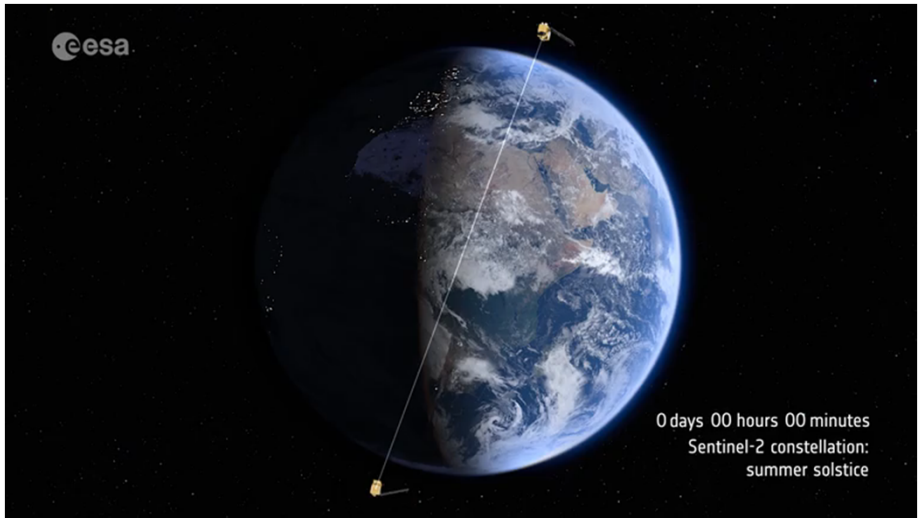


http://www.copernicus.eu/

Application Analysis

Sentinel 2 Mission

- Twin polar-orbiting satellites, phased at 180° to each other with a temporal resolution of 5 days at the equator in cloud free conditions.
- Provides images for agriculture, forests, land-use change and land-cover change, mapping biophysical variables, and monitoring of coastal and inland waters.
- Data is unlabelled but can be merged with labelled data from other sources, e.g. the Corine programme, an inventory on land cover in 44 classes.

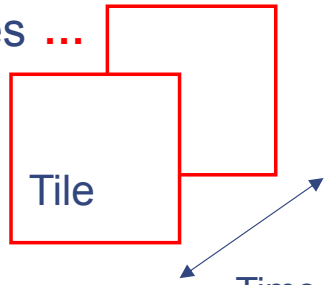


~23 TB data stored per day

Application Analysis

Germany – 1 year of Sentinel 2 Data

- Sentinel-2 tile gridding is based on the NATO Military Grid Reference System where each tile covers an area of 100 km x 100 km (excluding overlapping edges of 9.8 km).
- Germany can be covered with 56 tiles ...

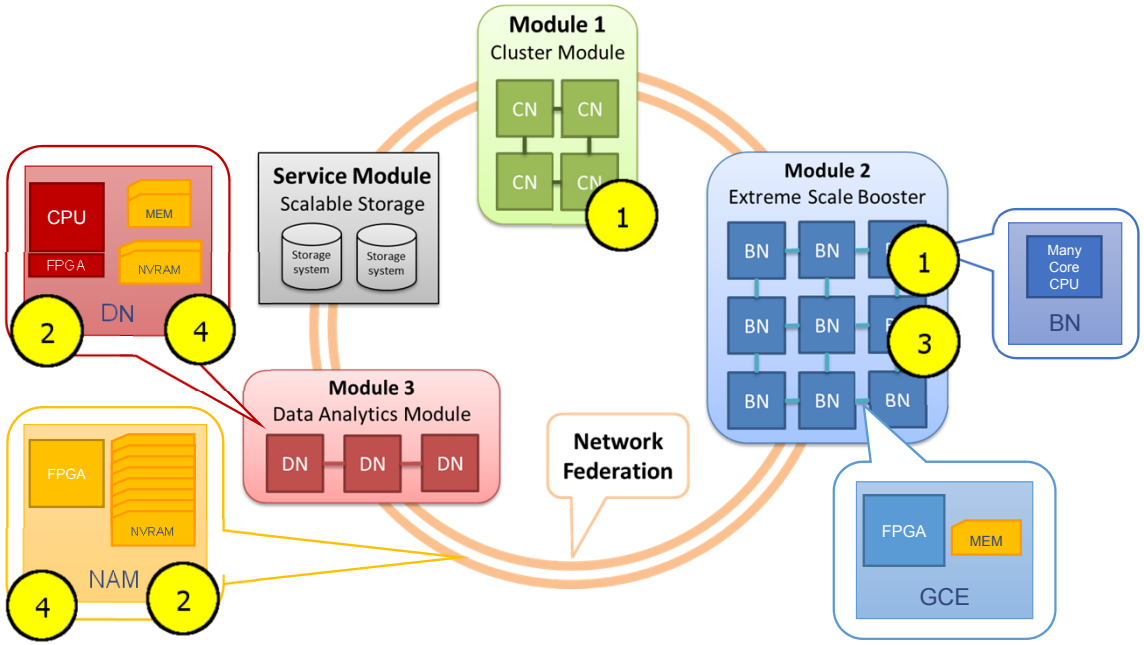


Time serie of tile images for 1 year
(365 days / 5 days = 73 acquisitions)

- Data size to be processed:
  73 acq. * 56 tiles * 700Mb = 2.72 TB

Application Analysis

PiSvM – Cross-validation



(1) Initial experiments performed with training and testing shows that a parameter space search is required in order to perform model selection (i.e. validation)

(2) Validation requires a validation dataset or again the training dataset when using cross-validation (low bias) but instead of reading the training data from a file again and again it can be placed in the DEEP-EST NAM **or the DAM´s NVRAM**

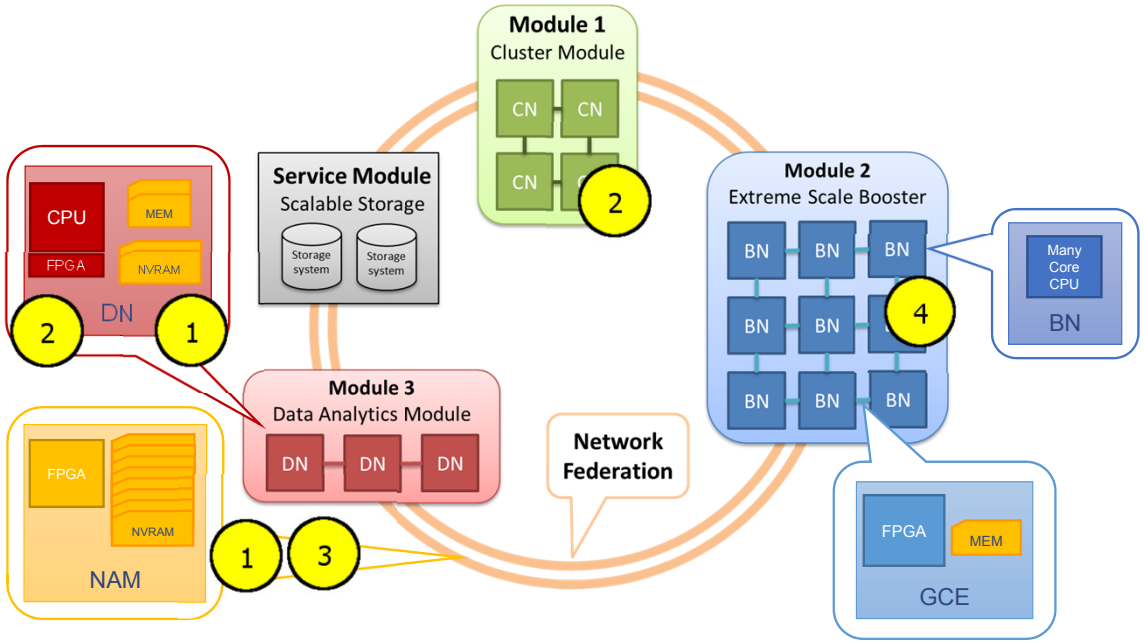
(3) n-fold cross-validation over a grid of parameters (kernel, cost) performs an estimate of the out-of-sample performance and performs n-times independent training process on a “folded” subset of the dataset (use of training data in folds). n-fold Cross-Validation (e.g. 10-fold often used) with piSVM is partly computational intensive whereby each fold can be nicely parallelized without requiring a good interconnection and thus can take advantage of the DEEP-EST CLUSTER module (use of training data in folds) whereby results of each fold per parameter can be put in the DEEP-EST NAM module **or the DAM´s NVRAM**

(4) The best parameters w.r.t. MAXIMUM accuracy in all the folds across all the parameter spaces can be computed using the DEEP-EST NAM **or DAM** module (FPGA computing maximum)

(5) The best parameter set that resides in the DEEP-EST NAM is given as input to the training/test pipeline (see PiSvM Tranining)

Application Analysis

PiSvM – Training



(1) The training dataset and testing dataset of the remote sensing application is used many times in the process and make sense to put into the DEEP-EST Network Attached Memory (NAM) module **or the NVRAM in the DAM.**

(2) Training with piSVM in order to generate a model requires powerful CPUs with good interconnection for the inherent optimization process and thus can take advantage of the DEEP-EST CLUSTER module (use of training dataset, requires piSVM parameters for kernel and cost). **Optionally, the DAM can be used in the case when its NVRAM is employed**.

(3) Instead of dropping the trained SVM model (i.e. file with support vectors) to disk it makes sense to put this model into the DEEP-EST NAM module

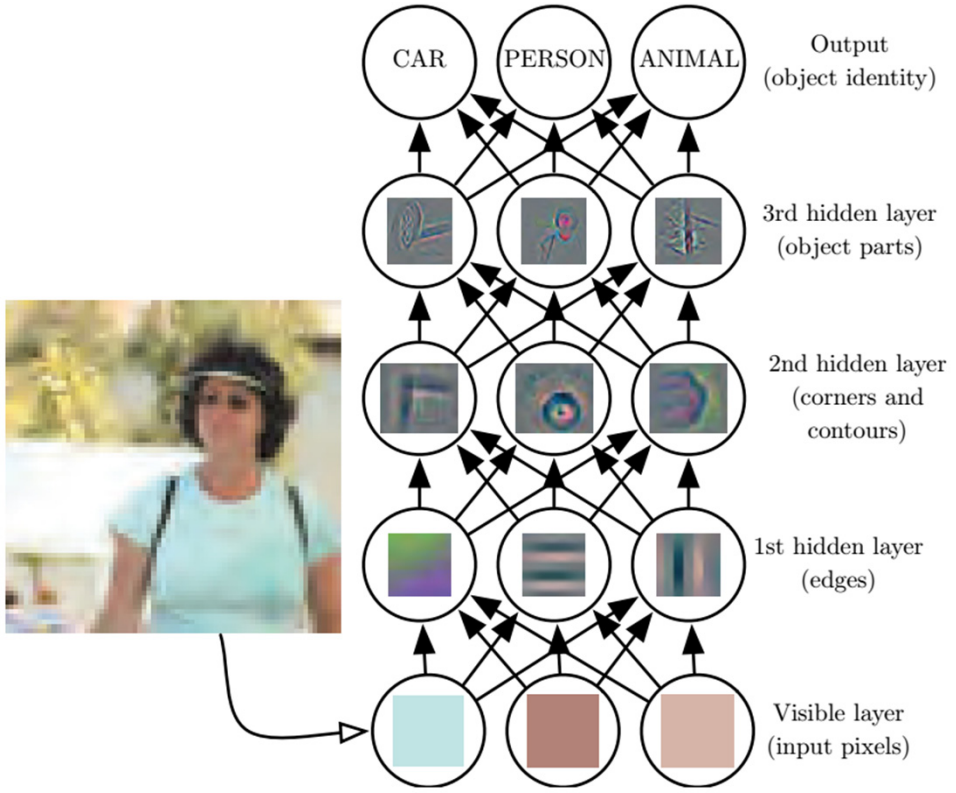
(4) Testing with piSVM in order to evaluate the model accuracy requires not powerful CPUs and not a good interconnection but scales perfectly (i.e. nicely parallel) and thus can take advantage of the BOOSTER module (use of testing dataset & model file residing in NAM)

(5) If accuracy too low back to (2) to change parameters

Application Analysis

Deep Neural Networks – The Algorithm

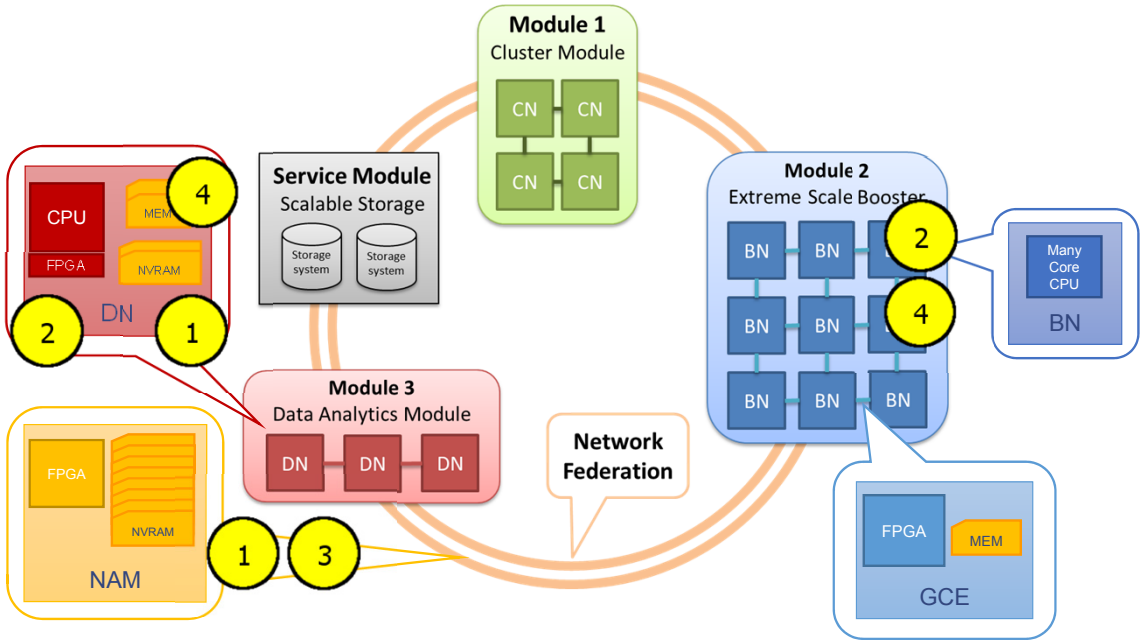
- Intrinsic feature engineering
- Supervised and unsupervised learning
- Tensorflow with the Keras extension
- Uses 3D CNNs for multi/hyper-spectral data



Ian Goodfellow, Yoshua Bengio, and Aaron Courville 'Deep Learning'

Application Analysis

DNNs – Convolutional Neural Networks



(1) The training dataset and testing dataset of the remote sensing application is used many times in the process and make sense to put into the DEEP-EST Network Attached Memory (NAM) module **or the DAM´s persistent memory**

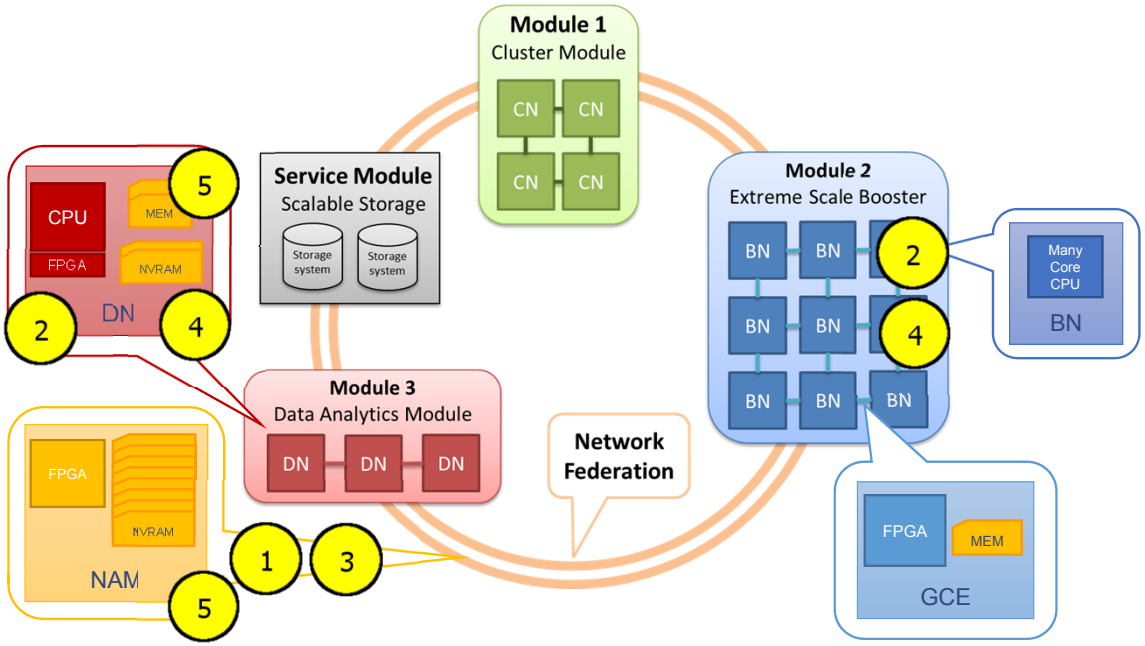
(2) Training with CNNs in Tensorflow works best with **GPGPUs and therefore the DAM module should be employed for the inherent optimization process based on Stochastic Gradient Descent (SGD). Optionally, the** many-core CPUs MPI collective available in the DEEP-EST Global Collective Environment (GCE) and can take advantage of the DEEP-EST BOOSTER module (use of training dataset, requires CNN architectural design parameters).

(3) Trained models of selected architectural CNN setups need to be compared and thus can be put in the DEEP-EST NAM module

(4) Testing with Tensorflow in order to evaluate the model **we use the GPGPUs, however,** inference works also quite well for many-core architectures, scales perfectly (i.e. nicely parallel), and thus can take advantage of the BOOSTER module (use of testing dataset & CNN models residing in NAM)

Application Analysis

DNNs – Transfer Learning



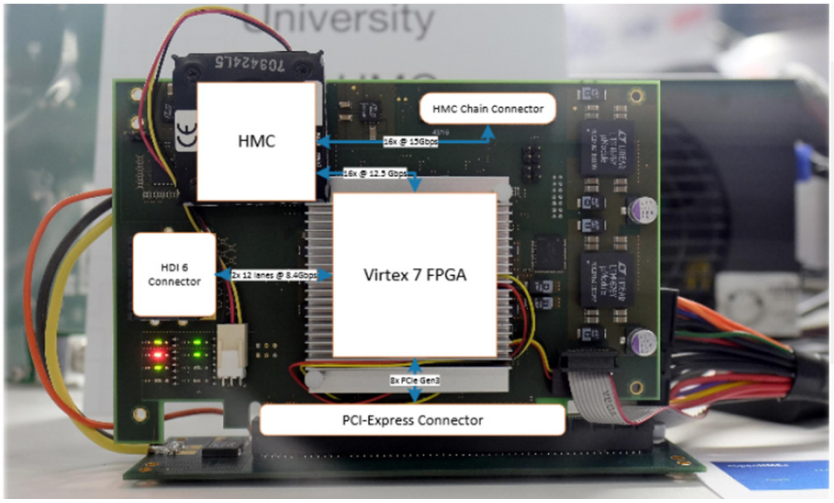
(1) Trained Models are stored in the DEEP-EST NAM module to be re-used for different unsupervised deep learning CNN training processes, possibly utilizing its FPGA for model pre-processing, e.g. prepare a trained model for transfer learning.
(2) Based on the pre-trained features, multiple new CNNs are trained using either the GPGPUs in the DAM or the CPUs in the BOOSTER module, possibly using the FPGAs in the former case for input data pre-processing.
(3) Trained models of selected architectural CNN setups that have been used with pre-trained features need to be compared and can be put in the DEEP-EST NAM module for fast accessibility.
(4) Testing with Tensorflow in order to evaluate the model accuracy should be done using the DAM's GPGPUs but it should also work well for many-core architectures as it scales perfectly (i.e. nicely parallel). Therefore, it can also take advantage of the BOOSTER module (use of testing dataset residing in memory & CNN models residing in NAM)
(5) Testing results cen be written to the DEEP-EST NAM using the FPGA in it to compute the best obtained accuracy for all the different setups. **Optionally the DAM can also be used for this purpose.**
(6) If accuracy is too low consider to move back to step (1) in order to change the pre-trained network or step (2) in order to create a better CNN architectural setup based on (another set of) pre-trained features

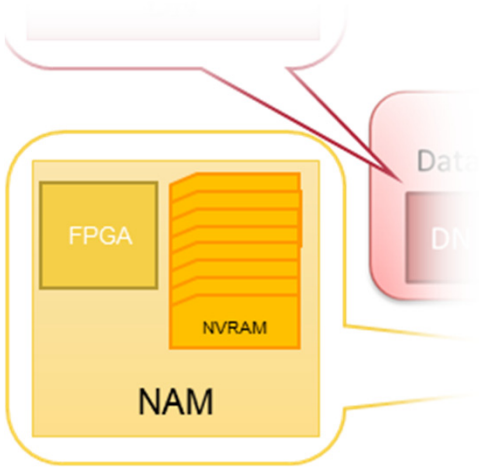
NAM

- Managed set libNAM up on DEEP

  https://gitlab.version.fz-juelich.de/galonska1/libNAM

- Started integrating the NAM into PiSvM workflow, using the extended MPI interface.





Conclusions

- The Modular Supercomputing Architecture (MSA) provides benefits for machine learning and deep learning applications
- Modular approach fits machine learning processes like training, testing, cross-validation, sampling, or averaging over models
- HPDBSCAN leverages MSA for unsupervised clustering
- piSVM leverages MSA for supervised classification
- Deep Learning models using Tensorflow and Keras are able to take advantage of the MSA concept too
- Memory will be a key for running data-intensive applications
- Innovative hardware like Network Attached Memory (NAM) or Non volatile RAM / persistent DIMMs enable new memory layouts
- Juelich Supercomputing Centre implements the MSA in its roadmap

DEEP Projects

The DEEP projects have received funding from the European Union's Seventh Framework Programme (FP7) for research, technological development and demonstration and the Horion2020 (H2020) funding framework under grant agreement no. FP7-ICT-287530 (DEEP), FP7-ICT-610476 (DEEP-ER) and H2020-FETHPC-754304 (DEEP-EST).