DEEP Projects

Enhancing Remote Sensing Applications towards Exascale with the DEEP-EST Modular Supercomputer Architecture

Gabriele Cavallaro (1), Ernir Erlingsson (2), Morris Riedel (1,2), Helmut Neukirchen (2)

(1) Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
(2) School of Engineering and Natural Sciences, University of Iceland, Iceland

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

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

Outline

- The Dynamical Exascale Entry Platform (DEEP) Project
  - Serie of projects co-funded by the European Union
  - Project Applications
  - Modular Supercomputing Architecture
- Big Data Analytics for Earth Science
  - Highly Parallel DBSCAN
  - Parallel Support Vector Machines
  - Deep Neural Networks
- Conclusions

The DEEP Projects

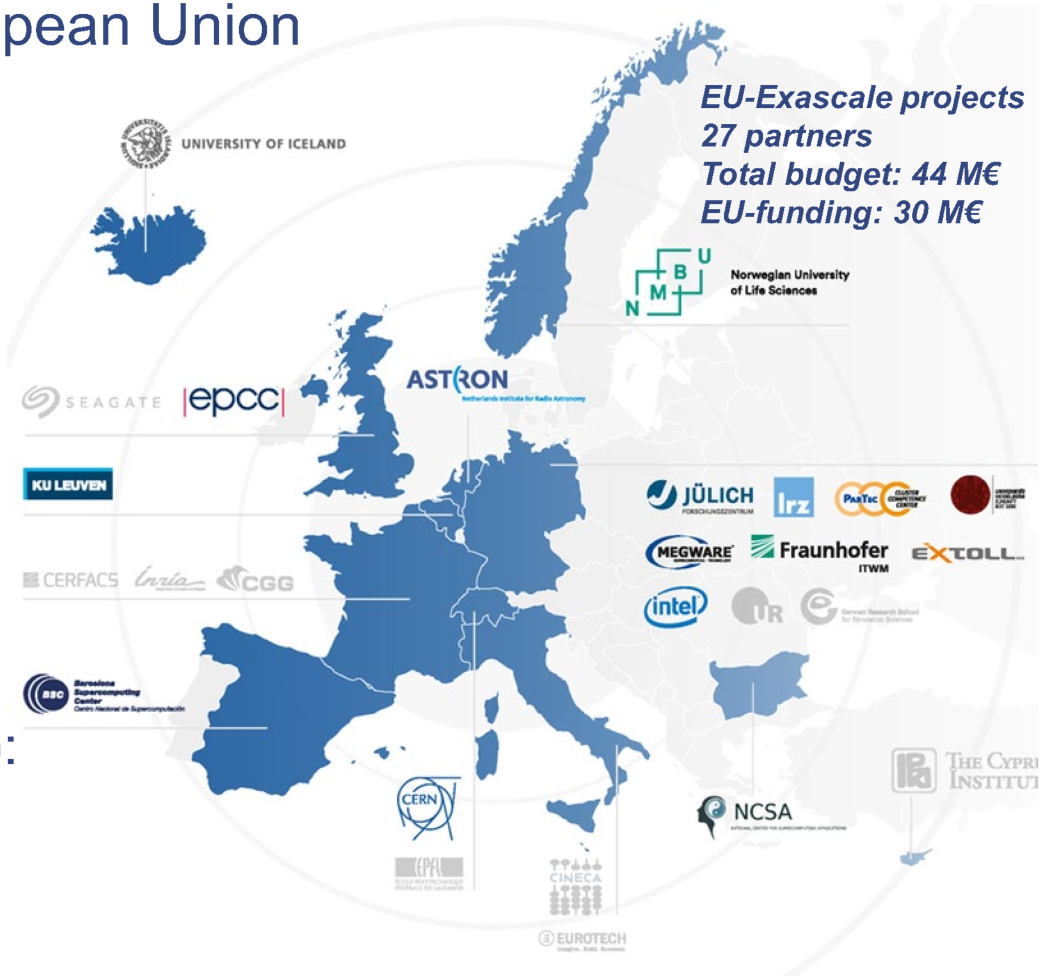
- Research & innovation projects co-funded by the European Union

- Paving the road towards Exascale computing

  – DEEP (2012–2015): Cluster/Booster concept

  – DEEP-ER (Extended Reach - 2013–2017):
  scalable I/O and resiliency

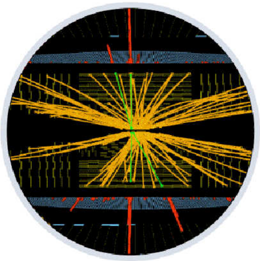
  – **DEEP-EST (Extreme Scale Technologies - 2017–2020)**:
  generalized modular supercomputing architecture,
  support for data analytics and machine learning
    - *One of only two Exascale project series funded*



[1] The DEEP projects

DEEP-EST Project Applications

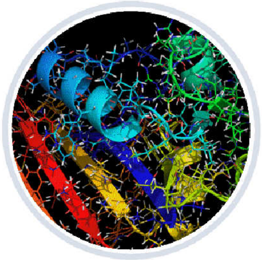
- <u>**University of Iceland – Earth Science**</u>
- CERN – High Energy Physics
- NEST – Brain Simulation
- ASTRON – Radio Astronomy
- KU Leuven – Space Weather
- GROMACS – Molecular Dynamics



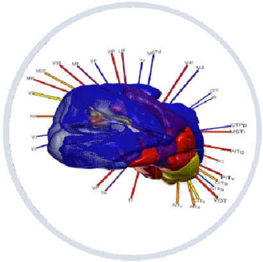
High Energy Physics

Earth Science

Space Weather



Molecular Dynamics



Neuroscience

Radio Astronomy

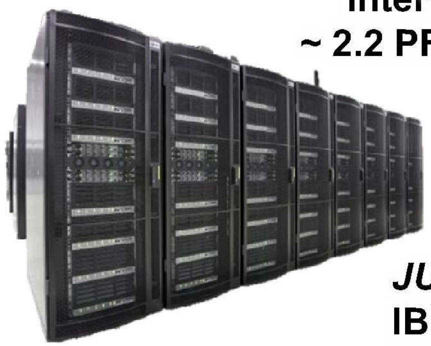
- Hardware design driven by software needs

HPC – One Size does NOT fit All

HPC systems come in two very different flavours

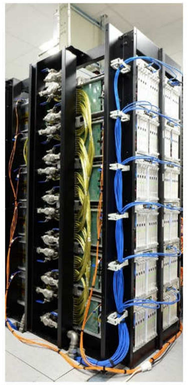
- General purpose Clusters with
  - *High flexibility & reliable performance*
  - *Preferred by many applications since "good enough" performance is easy to achieve*
  - *Relatively high power consumption*
- Dedicated, highly scalable, massively parallel systems
  - *Highest degree of parallelism, specialized fabrics*
  - *Few (highly parallelizable) codes can fully exploit them*
  - *Highly energy efficient*

The DEEP projects combine the two flavours with the Modular Supercomputing Architecture (MSA)



JSC JURECA Cluster
Intel® Xeon® Haswell
~ 2.2 PFlop/s in 1.3 MW

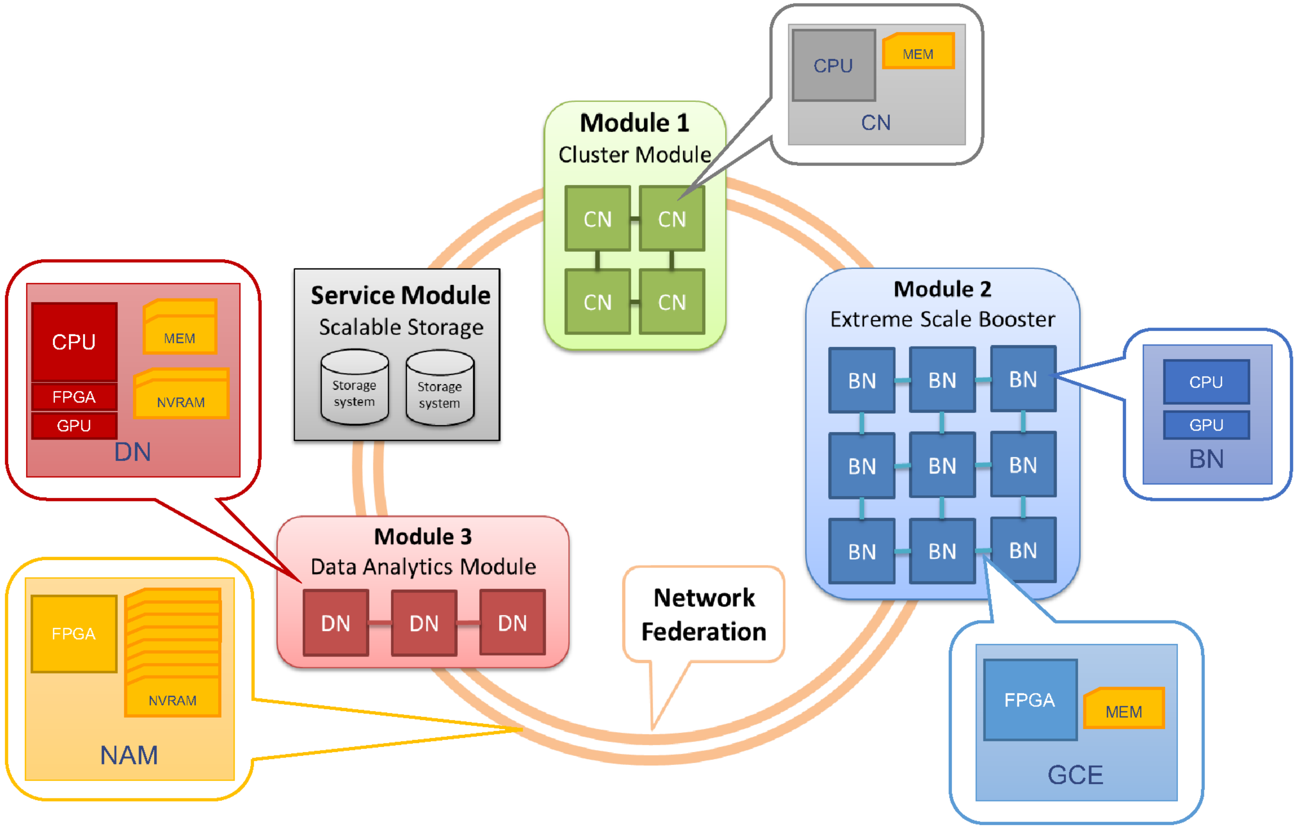
JUQUEEN
IBM Blue Gene/Q
5.9 PFlop/s in 2.3 MW



DEEP Prototype
Intel Xeon + Intel Xeon Phi
0.6 PFlop/s in 0.150 MW

Modular Supercomputing Architecture (MSA)

- DEEP-EST prototype includes three compute modules
  - CLUSTER Module
  - Extreme Scale Booster (ESB)
  - Data Analytics Module (DAM)
- Storage module handles workflow data
- Local storage and the network attached memory (NAM) for hot application data
- Global communication engine (GCE) accelerates MPI collective operations
- Fast network federation binds all parts together



Highly Parallel DBSCAN

- Unsupervised learning
  - Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm

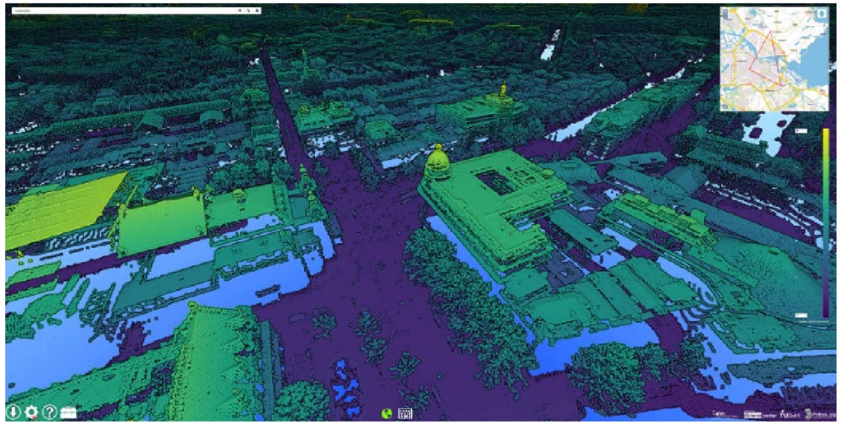
[2] DBSCAN

- HPDBSCAN: parallel implementation of DBSCAN for HPC [MPI, OpenMP, HDF5 I/O]

[3] HPDBSCAN

- Point-cloud data acquired with 3D laser scanner on board of ground or airborne platforms
  - Scans of cities, landmarks, or nature
  - Plans on using AHN2 (Actueel Hoogtebestand Nederland – Version 2)
    - *10 points per 1 m^2*

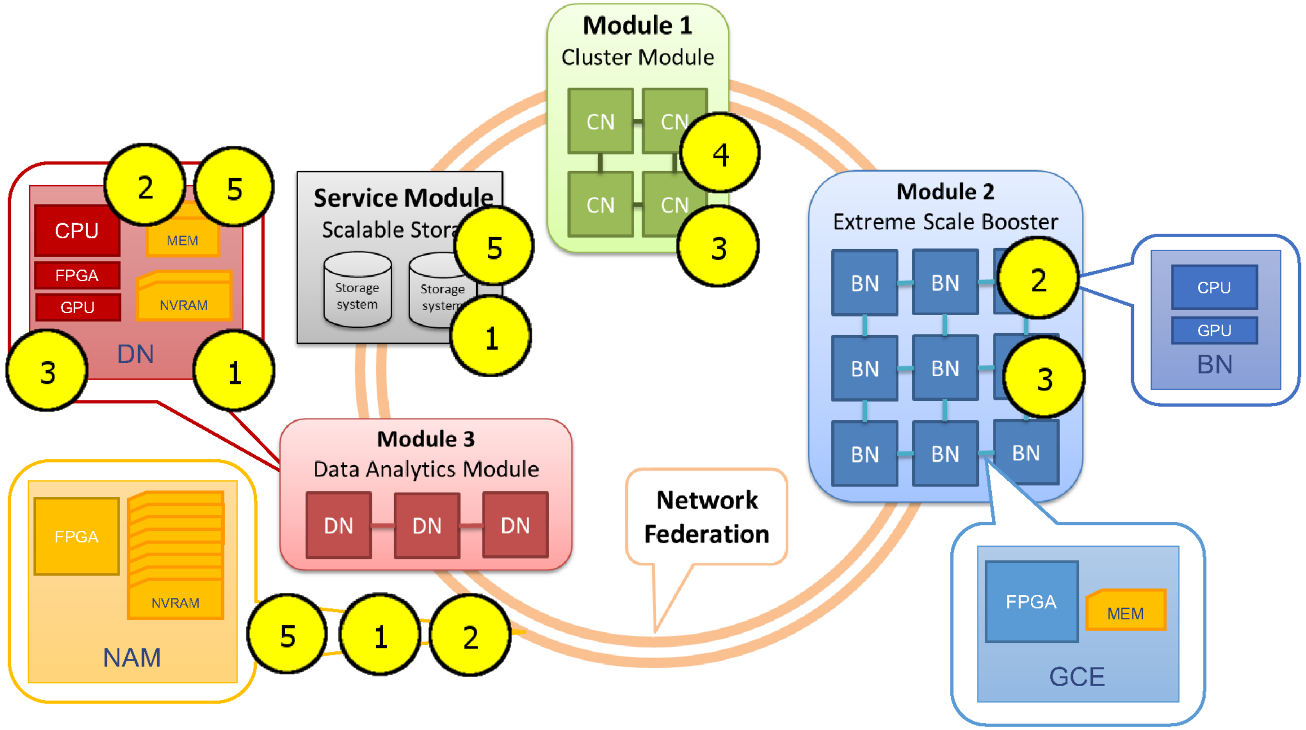
Name	Points	LAS files	Disk size [GB]	Area [km²]	Description
20M	20,165,862	1	0.4	1.25	TU Delft campus
210M	210,631,597	16	4.0	11.25	Major part of Delft city
2201M	2,201,135,689	153	42.0	125	City of Delft and surroundings
23090M	23,090,482,455	1,492	440.4	2,000	Major part of Zuid-Holland province
639478M	639,478,217,460	60,185	11,644.4	40,000	The Netherlands



[4] AHN2 (open data in NL)

Highly Parallel DBSCAN - Mapping

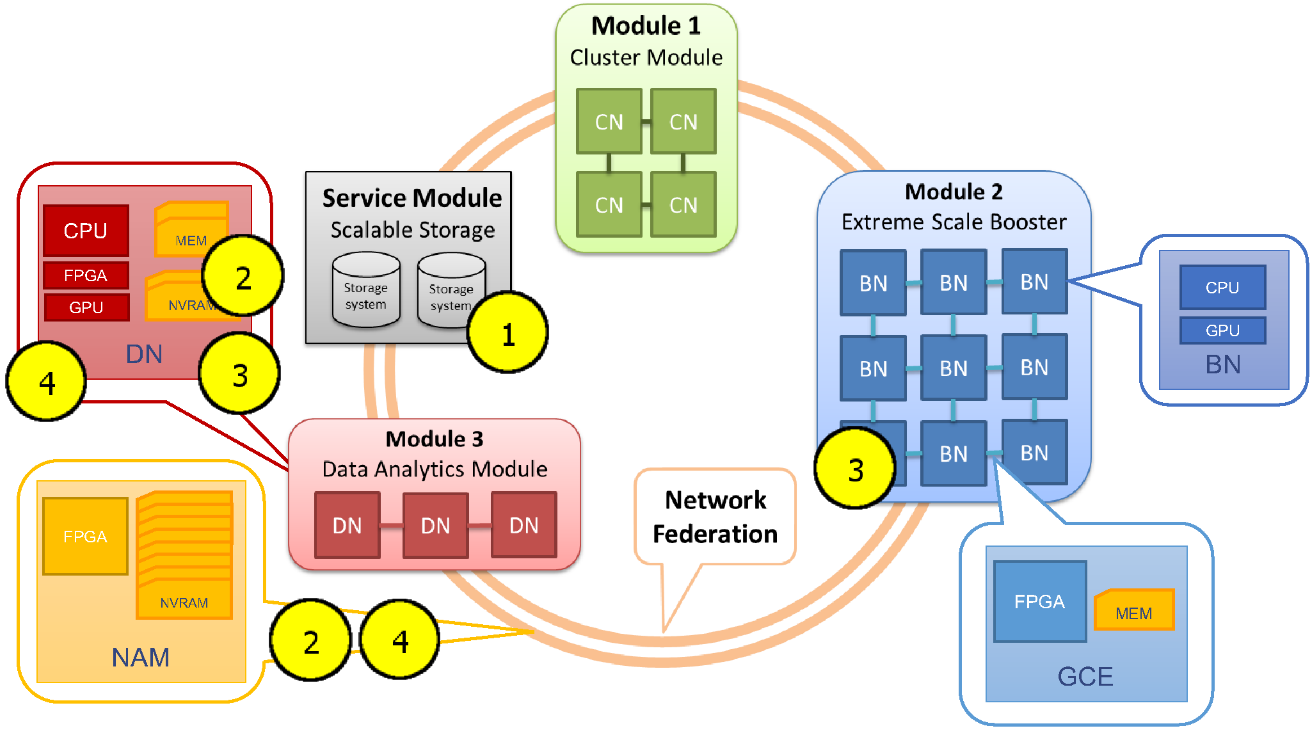
Clustering and Indexing



1. Load the dataset with parallel I/O using HDF5 in the Scalable Storage Service Module (SSSM) or the CLUSTER module. Larger size datasets can be firstly loaded into the persistent memory DIMMs (DAM), or the NAM
2. The indexing through sorting and cost heuristic small computing elements takes advantage of the possible GPGPU or FPGA accelerators in the DAM, NAM or ESB modules
3. Local clustering is performed on one of the three possible hardware mappings: CLUSTER module, ESB module with GPGPU accelerators, or DAM module with FPGA/GPGPU accelerators
4. Merging the locally computed clusters is performed on the CLUSTER module
5. Cluster IDs and noise IDs are written to a HDF5 file, or optionally directly into the DAM's NVRAM or the NAM for further analysis

Highly Parallel DBSCAN - Mapping

HPDBSCAN – Level of Detail (LoD) study



1. Load the dataset with parallel I/O using HDF5 in the SSSM
2. After the clustering (see previous slide), or step 1 above, the data resides in the NAM or the DAM's NVRAM
3. LoD study of selected point-cloud data is performed on the accelerators either in the ESB or DAM modules
4. The different data set results of the various modifications are placed either in the NAM or the DAM´s NVRAM
5. The modified data can optionally be re-clustered for further study

Parallel Support Vector Machines (PiSvM)

- Supervised learning
  - Classification with Support Vector Machines (SVMs) [LIBSVM I/O]

[5] Support-Vector Networks

- PiSvM: parallel implemention of SVMs for HPC [MPI, HDF5 I/O]
  - Fork of πSvM, which in turn is derived from LIBSVM [open source library - C/C++]

[6] PiSvM

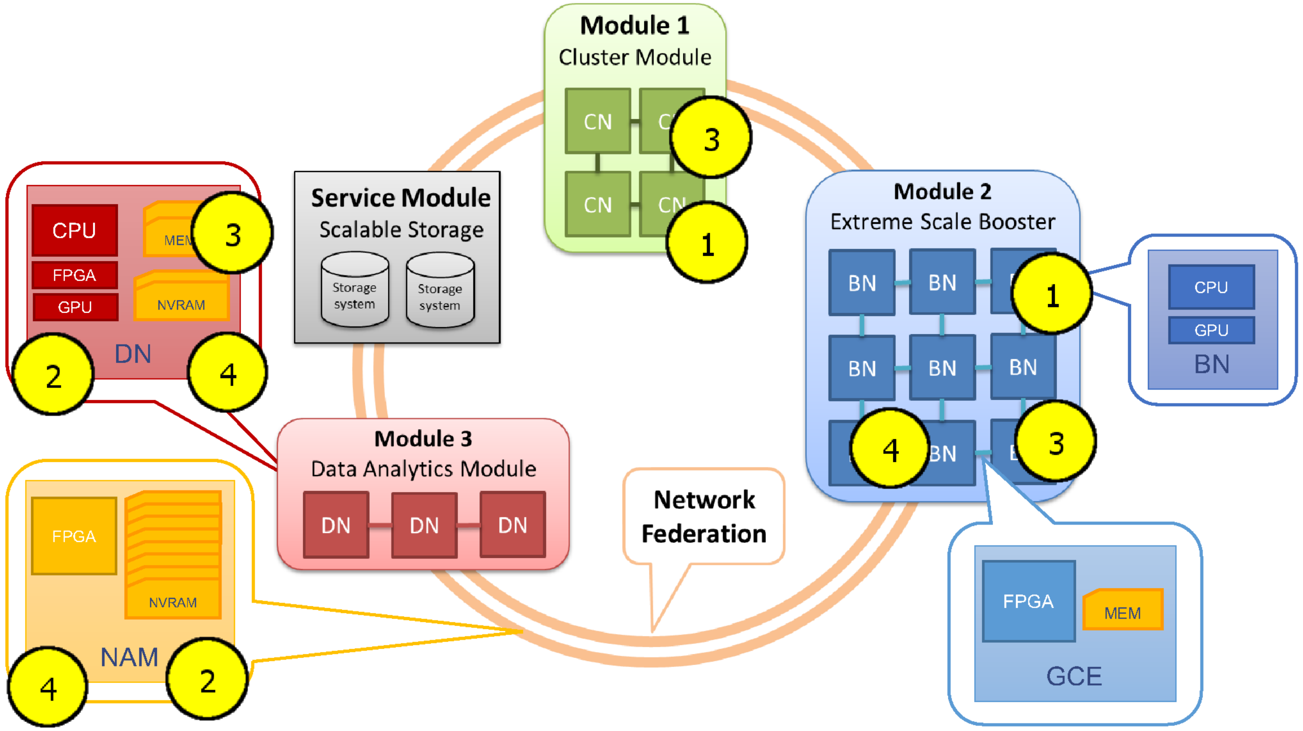
[7] LIBSVM

- Optical data acquired with sensors on board of satellite platforms
  - Large-scale production of land cover maps
  - Based on time series data (e.g., Sentinel-2)

[8] Copernicus

PiSvM - Mapping

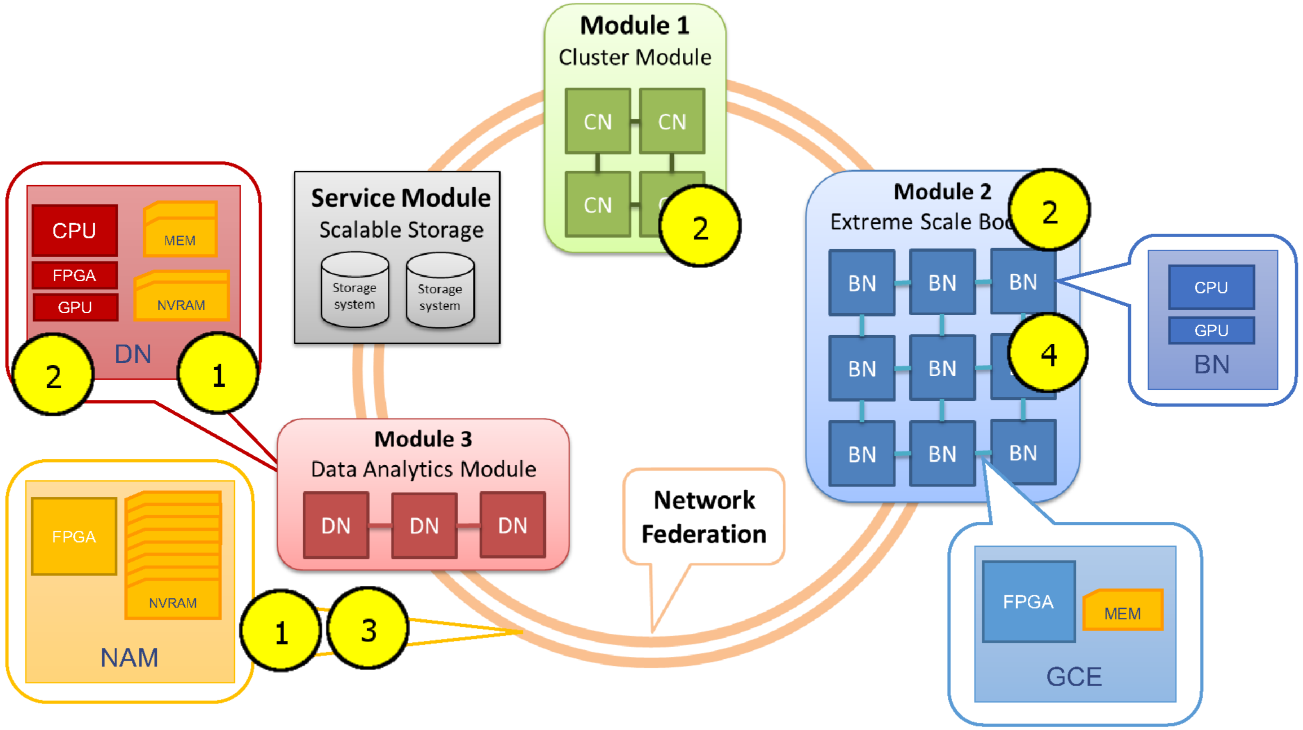
Cross-validation



1. The need for validation is established
2. Validation requires either a separate validation data set or the training data set with cross-validation. The selected data set is placed in the NAM or the DAM´s NVRAM for fast access
3. N-fold cross-validation is performed. This operation is “nicely parallel” and is divided among multiple nodes in the ESB, CLUSTER or DAM modules
4. The best parameters are determined and stored in the NAM
5. Optionally, the best parameters in the NAM can be given as input to the training/prediction workflow of the next slide

PiSvM - Mapping

Training and Prediction



1. I/O access performance is optimized by placing the data sets in the NAM or the DAM´s NVRAM
2. The training takes place in one of the three modules (all possibilities are studied and compared): the CLUSTER module with the most powerful CPUs, the ESB module using major GPU offloading and the DAM using a combination of accelerators and CPUs
3. The trained model is stored in the NAM for fast I/O
4. The prediction is a “nicely parallel” operation which therefore can take advantage of the massive parallelism offered by the ESB module
5. If the accuracy of the model is too low, the workflow can revert back to step (2)

Deep Neural Networks

- Supervised learning:
  – Classification with Convolutional Neural Networks (CNNs)

[8] TensorFlow

[9] Keras

- TensorFlow with the Keras extension

[8] Copernicus

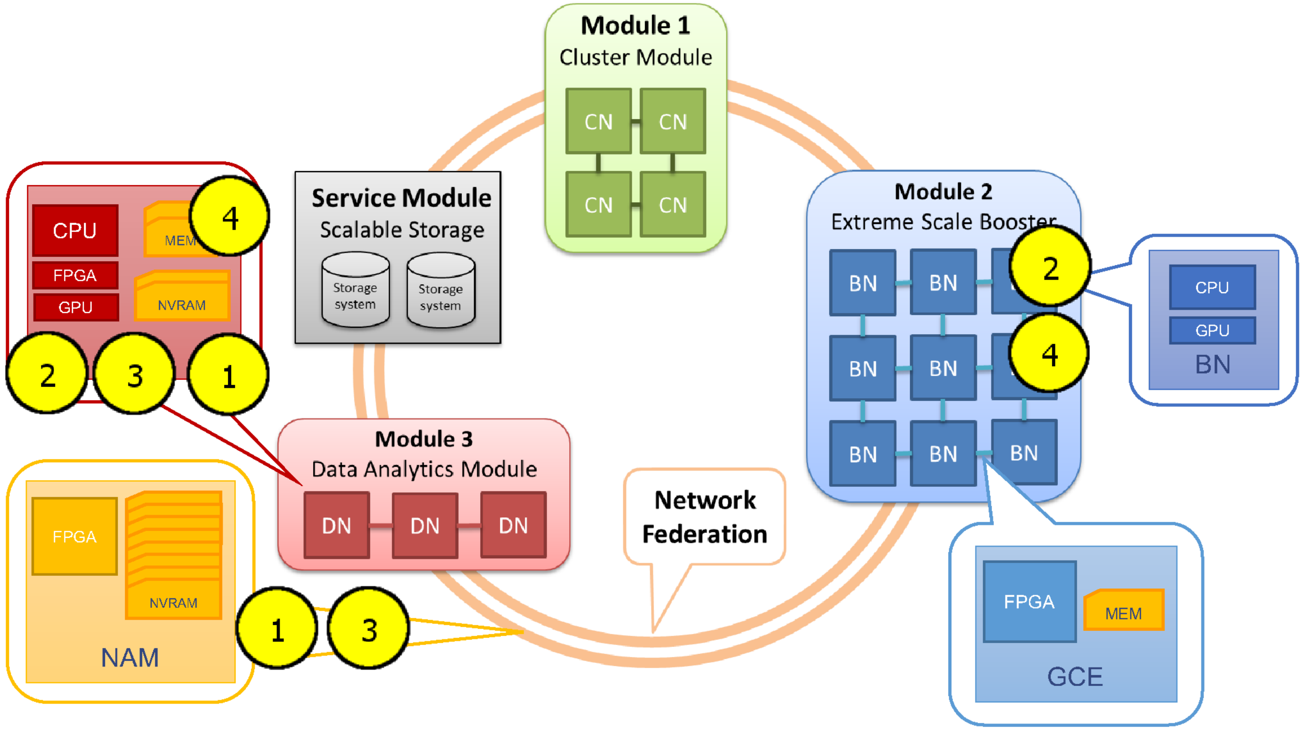
- Optical data acquired with sensors on board of satellite platforms
  – Large-scale production of land cover maps
  – Based on time series data (e.g., Sentinel-2)

- RGB images acquired with surveillance cameras
  – Sense the visibility reduction (fog detection)
  – Millions of RGB images from the Dutch freeways

[11] Fog detection

Deep Neural Networks - Mapping

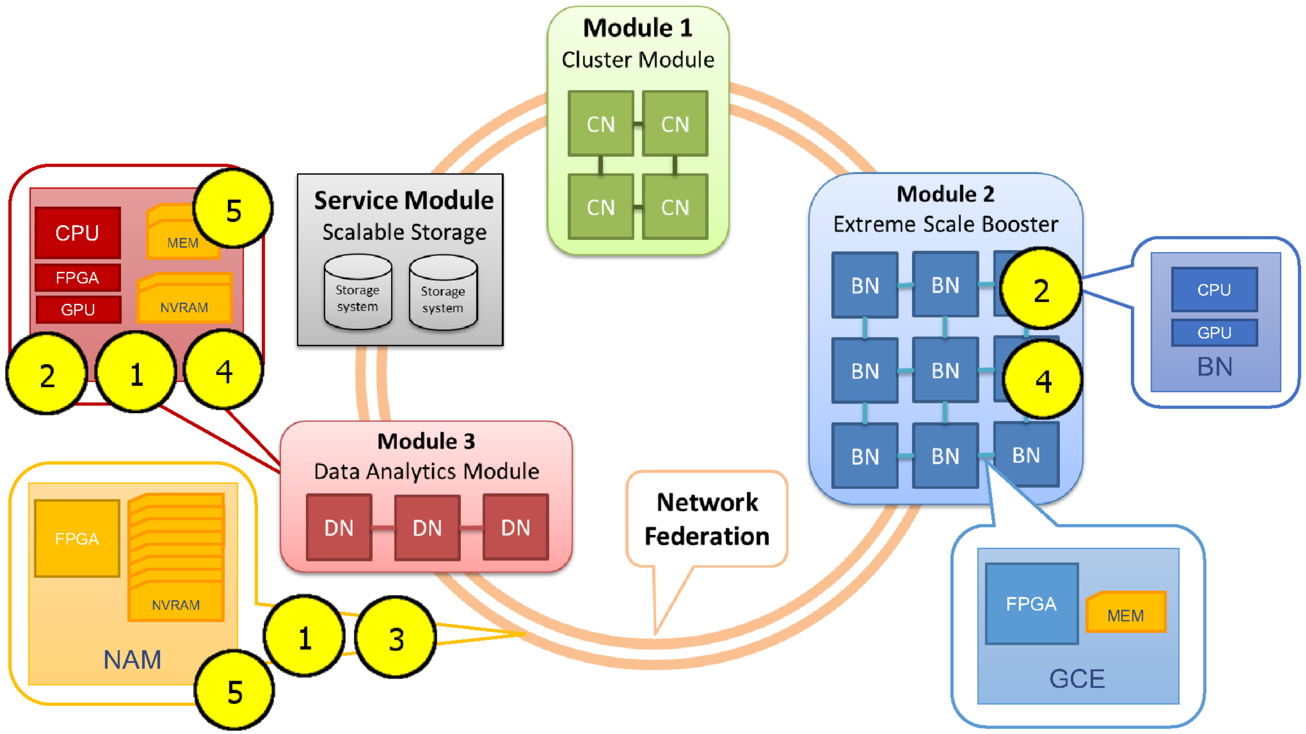
Convolutional Neural Networks



1. The training and test datasets are used multiple times and are therefore stored either in the NAM or the DAM´s NVRAM
2. The Training with CNNs and TensorFlow takes advantage of the GPGPUs in the DA and/or the ESB modules
3. Trained models are stored in the NAM, or the DAM module's NVRAM
4. The model evaluation takes advantage of the GPGPUs in the DAM or the ESB modules

Deep Neural Networks - Mapping

Transfer Learning



1. Trained models, which are stored in the NAM or the DAM´s NVRAM respectively, are re-used for training using transfer learning
2. Multiple CNNs are trained in parallel using the GPGPUs in the DAM and/or ESB modules
3. Newly trained models are stored in the NAM for fast I/O during inference
4. Models are evaluated with TensorFlow using the GPGPUs in the DAM and/or ESB modules
5. Results are written to the NAM and its FPGA is utilized for swift comparisons

Conclusions

- DEEP-EST project (2017-2020)
  - Co-design: hardware, software and applications
- Modular Supercomputers Architecture
  - Road to Exascale computing via efficiently interconnected accelerators
- Big Earth Observation data analytics
  - Algorithms that scale on innovative hardware infrastructures

The Team

Ernir Erlingsson
ere29@hi.is

Gabriele Cavallaro
g.cavallaro@fz-juelich.de

Morris Riedel
m.riedel@fz-juelich.de

Helmut Neukirchen
helmut@hi.is

Bibliography

- [1] Deep Projects
  Online: http://www.deep-projects.eu/
- [2] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” in Proceedings of the SIGKDD, 1996, pp. 226–231.
- [3] M. Gotz, C. Bodenstein, and M. Riedel, “HPDBSCAN: Highly Parallel DBSCAN,” in Proceedings of the Workshop on Machine Learning in High Performance Computing Environments (MLHPC ’15), 2015.
- [4] Realistic Benchmarks for Point Cloud Data Management Systems
  Online: http://www.pointclouds.nl/docs/JIGC_2015_keynote_pointclouds.pdf
- [5] C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
- [6] PiSvM Code Repository
  Online: http://pisvm.sourceforge.net/
- [7] LIBSVM -- A Library for Support Vector Machines
  Online: https://www.csie.ntu.edu.tw/~cjlin/libsvm/
- [8] Copernicus: European Union's Earth Observation Programme
  Online: http://www.copernicus.eu/
- [9] TensorFlow: An Open Source Machine Learning Framework for Everyone
  Online: https://www.tensorflow.org/
- [10] Keras: The Python Deep Learning library
  Online: https://keras.io/
- [11] Fog Detection from Camera Images
  Online: https://datalab.knmi.nl/en/case/fog-detection-from-camera-images/

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).