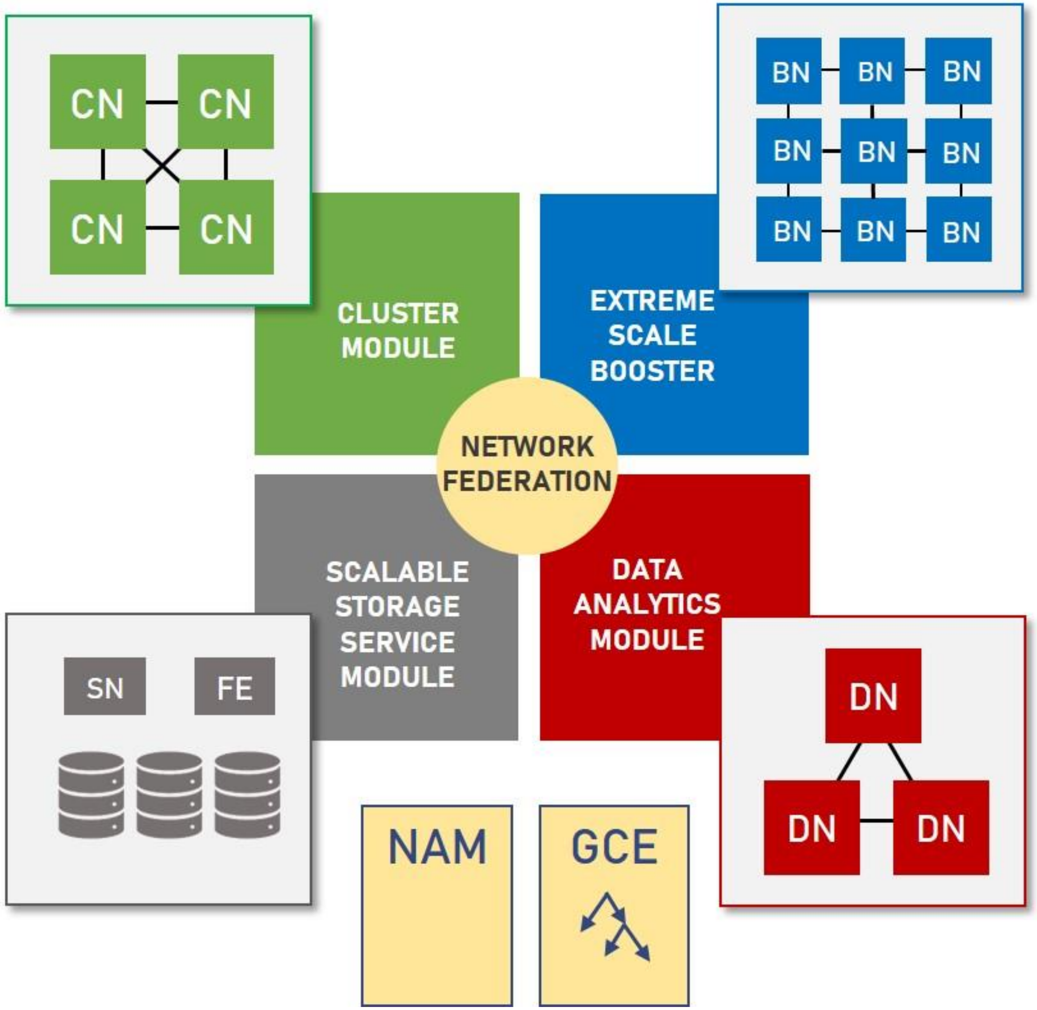
Exploiting Modular HPC in the context of DEEP-EST and ATTRACT projects

Viktor Khristenko (CERN)

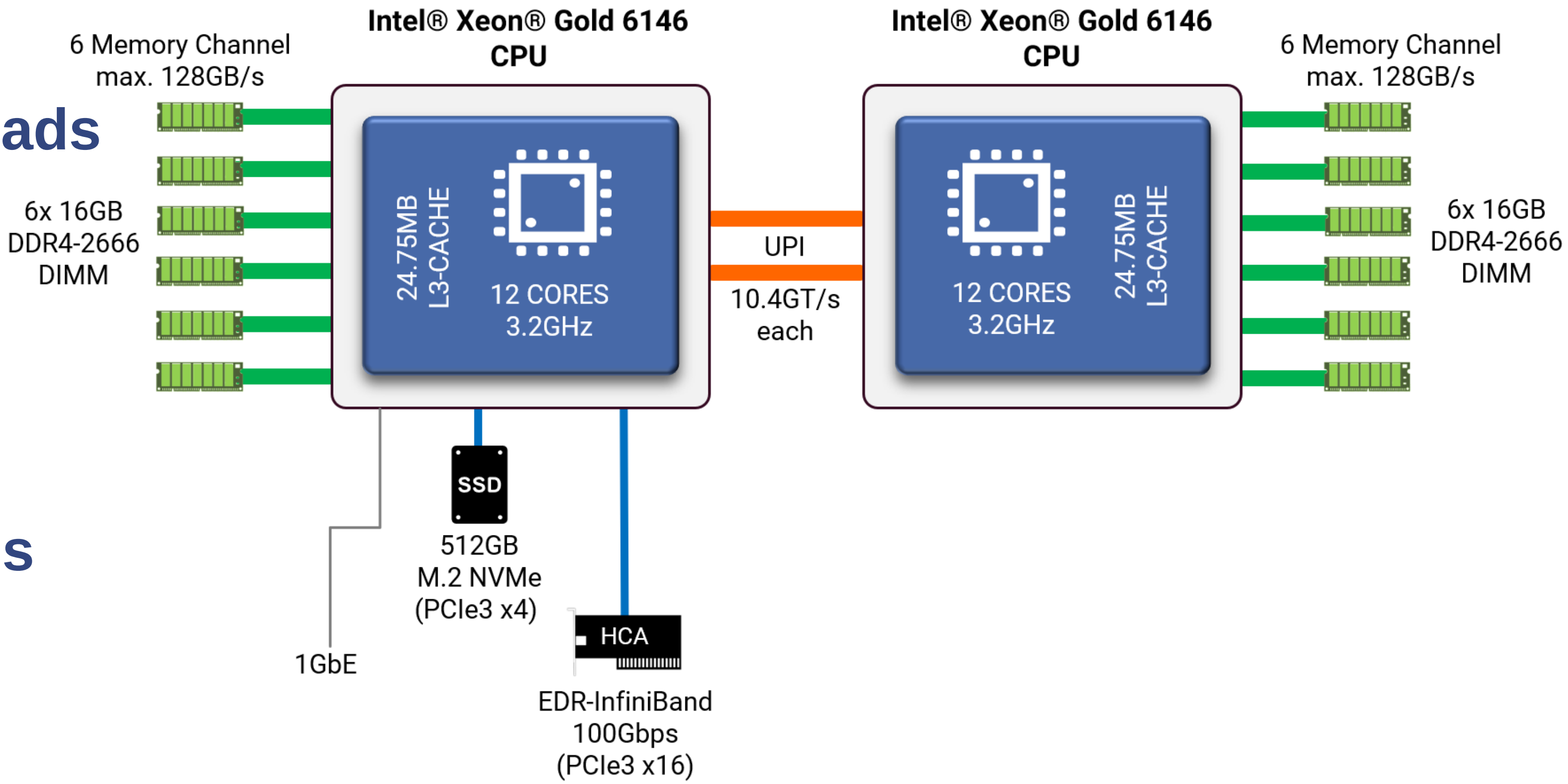
DEEP-EST Modular Supercomputer

- Prototype for the Modular Heterogeneous HPC system
- Convergence of HPC and HPDA worlds
- Variety of hardware to enable wide range of applications
- Software Hardware co-design driven by 6 applications



Cluster Module

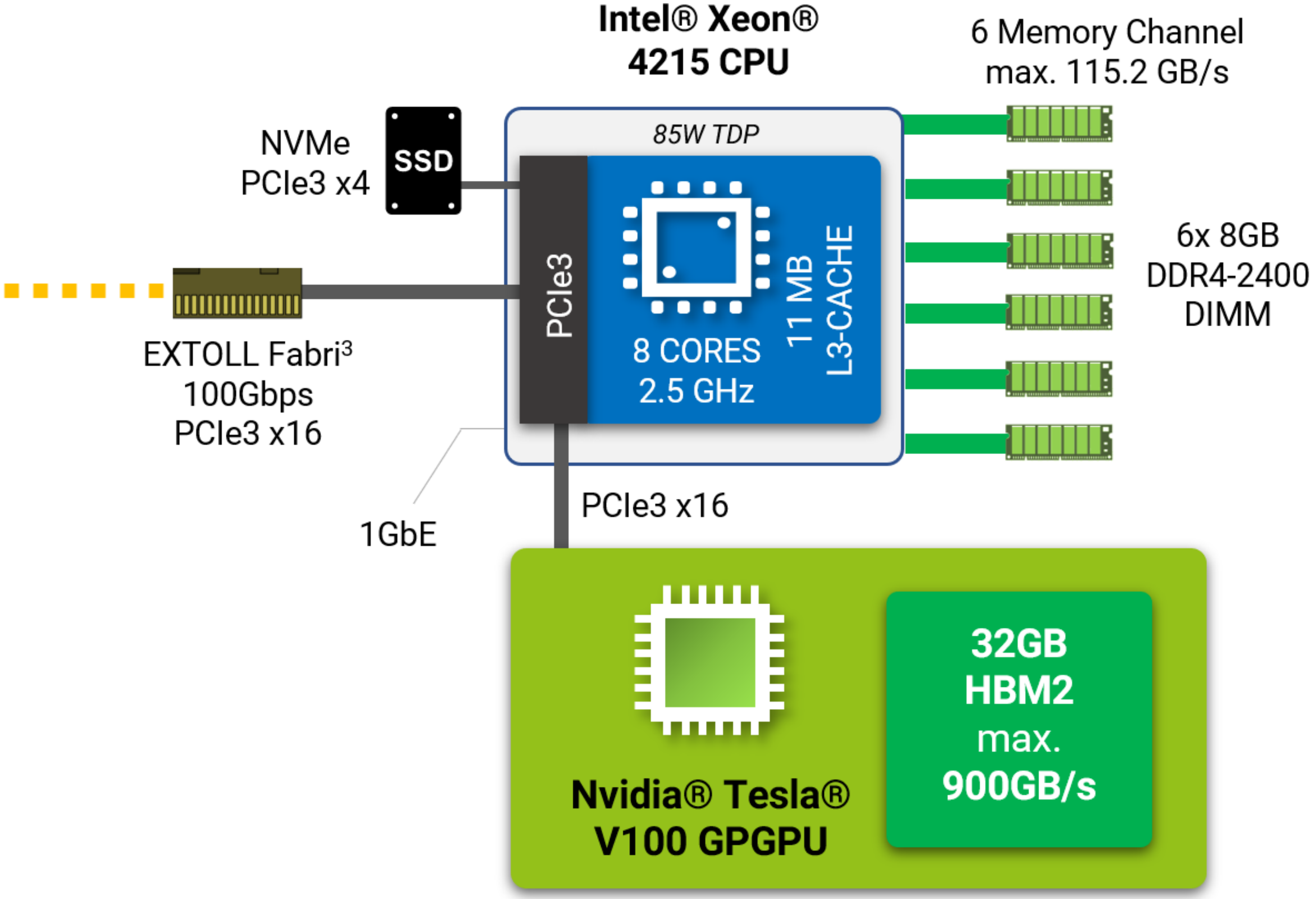
- Overall 50 nodes
- Aimed at CPU-bound workloads
- To/from ESB
  - Infiniband/Extoll Bridge
- To/from DAM
  - Inifiband/Ethernet 40Gbps Bridge



6 Memory Channel max. 128GB/s
Intel® Xeon® Gold 6146 CPU
Intel® Xeon® Gold 6146 CPU
6 Memory Channel max. 128GB/s
6x 16GB DDR4-2666 DIMM
24.75MB L3-CACHE
12 CORES 3.2GHz
UPI
10.4GT/s each
12 CORES 3.2GHz
24.75MB L3-CACHE
6x 16GB DDR4-2666 DIMM
SSD
512GB M.2 NVMe (PCIe3 x4)
HCA
1GbE
EDR-InfiniBand 100Gbps (PCIe3 x16)

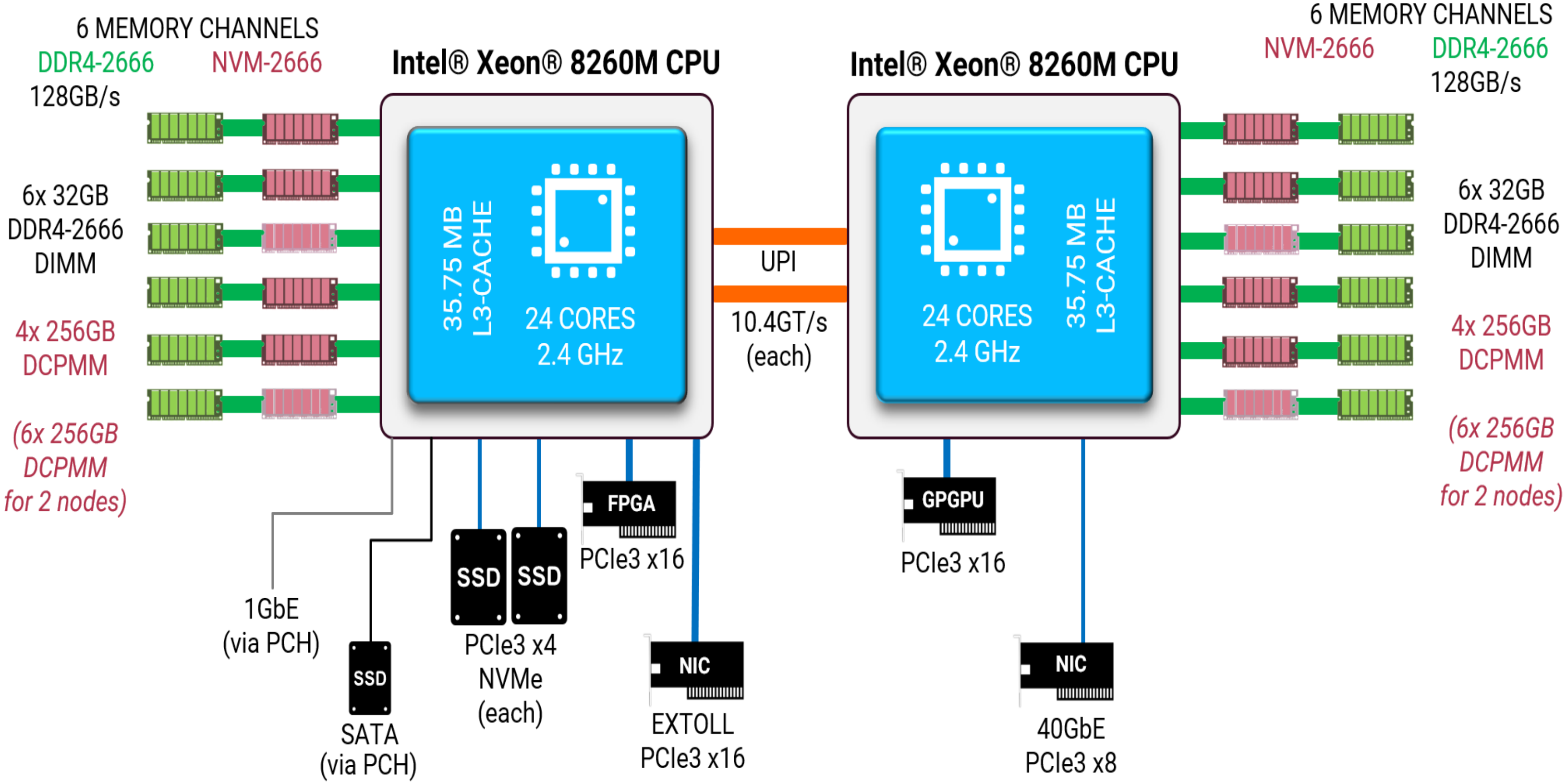
Extreme Scale Booster

- Overall 75 nodes
- GPU-based, Nvidia V100
- Extoll Network Fabric
- From/to CM
    - Infiniband – Extoll Bridge



Data Analytics Module

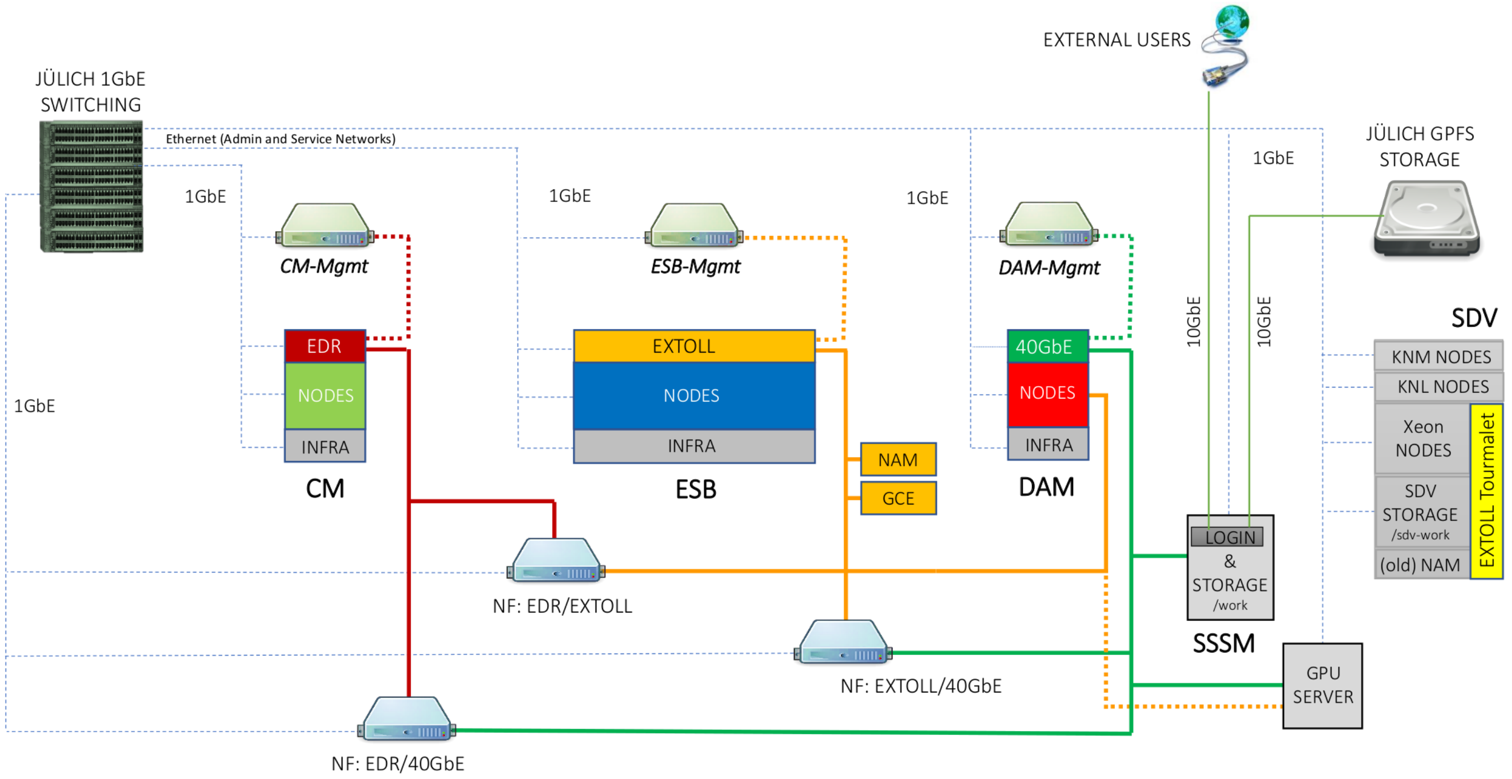
- 16 nodes
- 2 accelerators per node
  - 1 Nvidia V100
  - 1 Intel Stratix 10
- 2-3TBs Intel Optane Memory + 384GB DDR4



Network Federation + Auxiliary

- Multiple fabrics
  - 100Gbps Infiniband
  - 100Gbps Extoll
  - 40 Gbps Ethernet
  - Bridges
- Network Attached Memory NAM
  - Extoll's FPGA based solution
  - 128GBs DDR4
  - TB(s) SSDs
  - See ATTRACT slides
- Global Collective Engine GCE
  - Extoll's FPGA based solution
  - Accelerate MPI-collective operations

DEEP-EST Prototype – Schematic Network Overview



Racks Assembly Movie

Installation of DEEP-EST Prototype
(Cluster Module)

DEEP-EST Early Access Programme

- Apply here, https://www.deep-projects.eu/access.html

DEEP-EST: Heterogenous data processing

- Heterogenous Execution for CMSSW
  - Concentrating on HCAL / ECAL Local Energy Reconstruction

Current Calorimeters take 15-20% RECO time

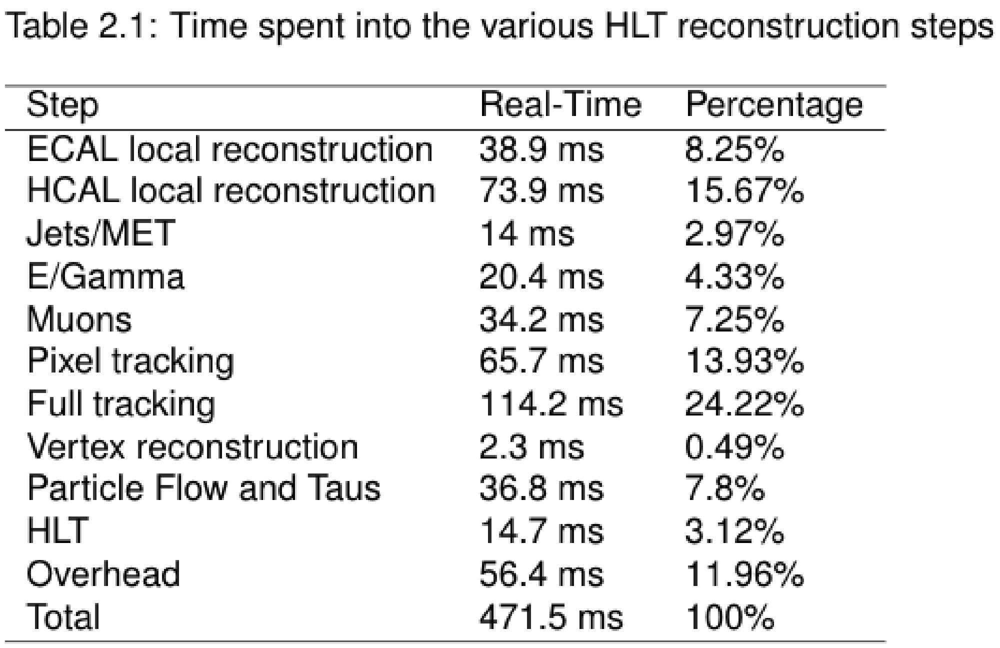
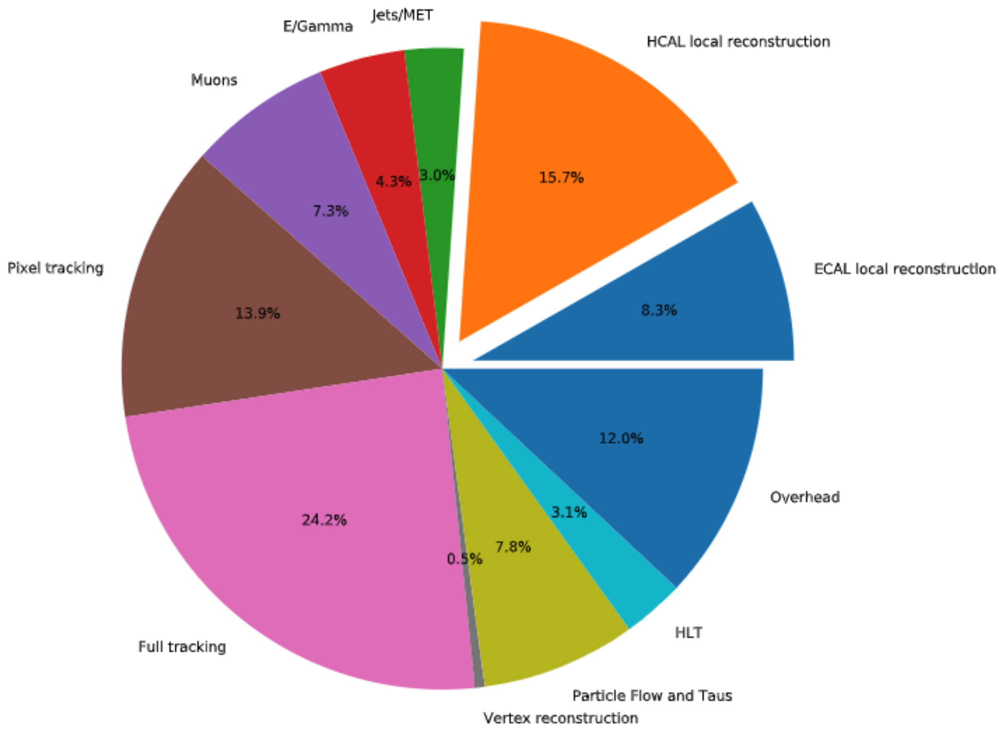


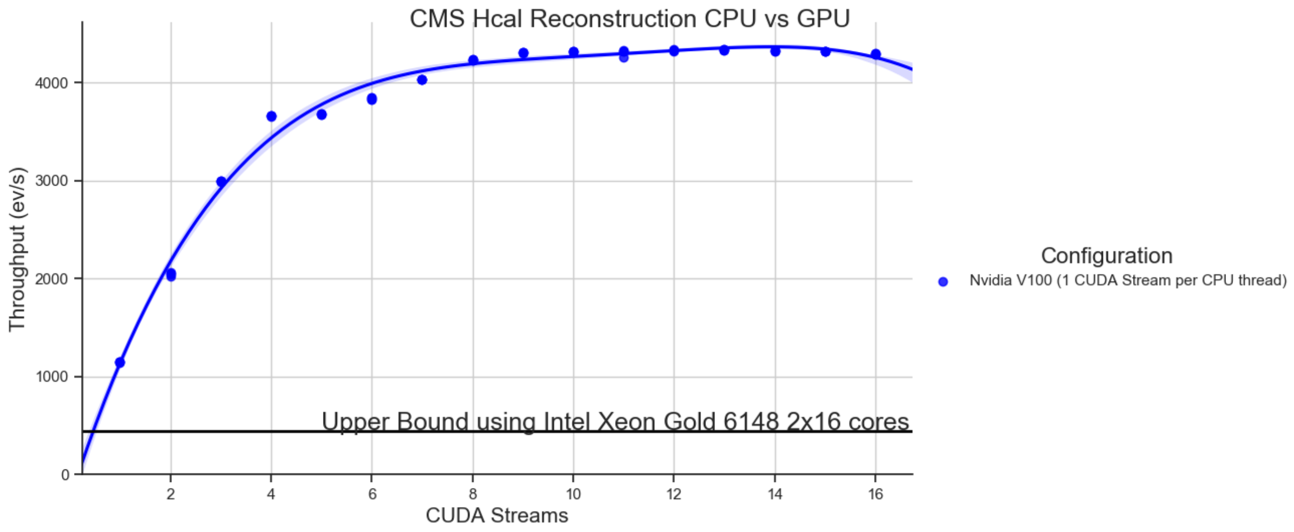
Table 2.1: Time spent into the various HLT reconstruction steps

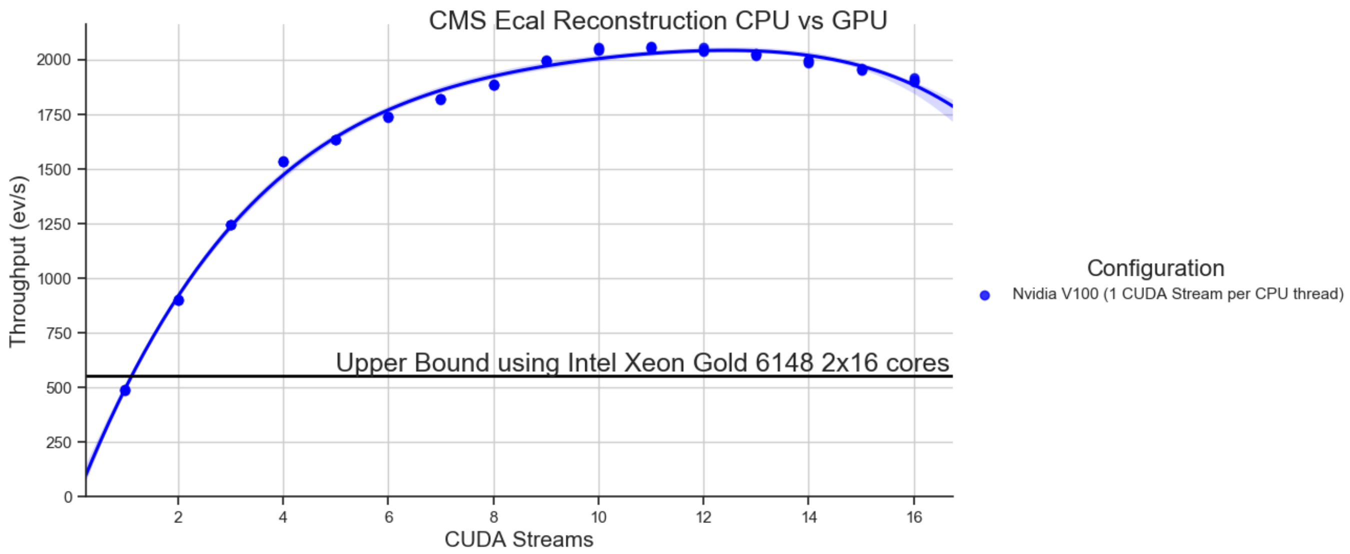
Step	Real-Time	Percentage
ECAL local reconstruction	38.9 ms	8.25%
HCAL local reconstruction	73.9 ms	15.67%
Jets/MET	14 ms	2.97%
E/Gamma	20.4 ms	4.33%
Muons	34.2 ms	7.25%
Pixel tracking	65.7 ms	13.93%
Full tracking	114.2 ms	24.22%
Vertex reconstruction	2.3 ms	0.49%
Particle Flow and Taus	36.8 ms	7.8%
HLT	14.7 ms	3.12%
Overhead	56.4 ms	11.96%
Total	471.5 ms	100%



Results

- http://opendata.cern.ch/record/12303
- 20K events. Replicate twice
- @flatiron
- exclusive allocation
- Nvidia V100
- Intel Xeon Gold 6148





DEEP

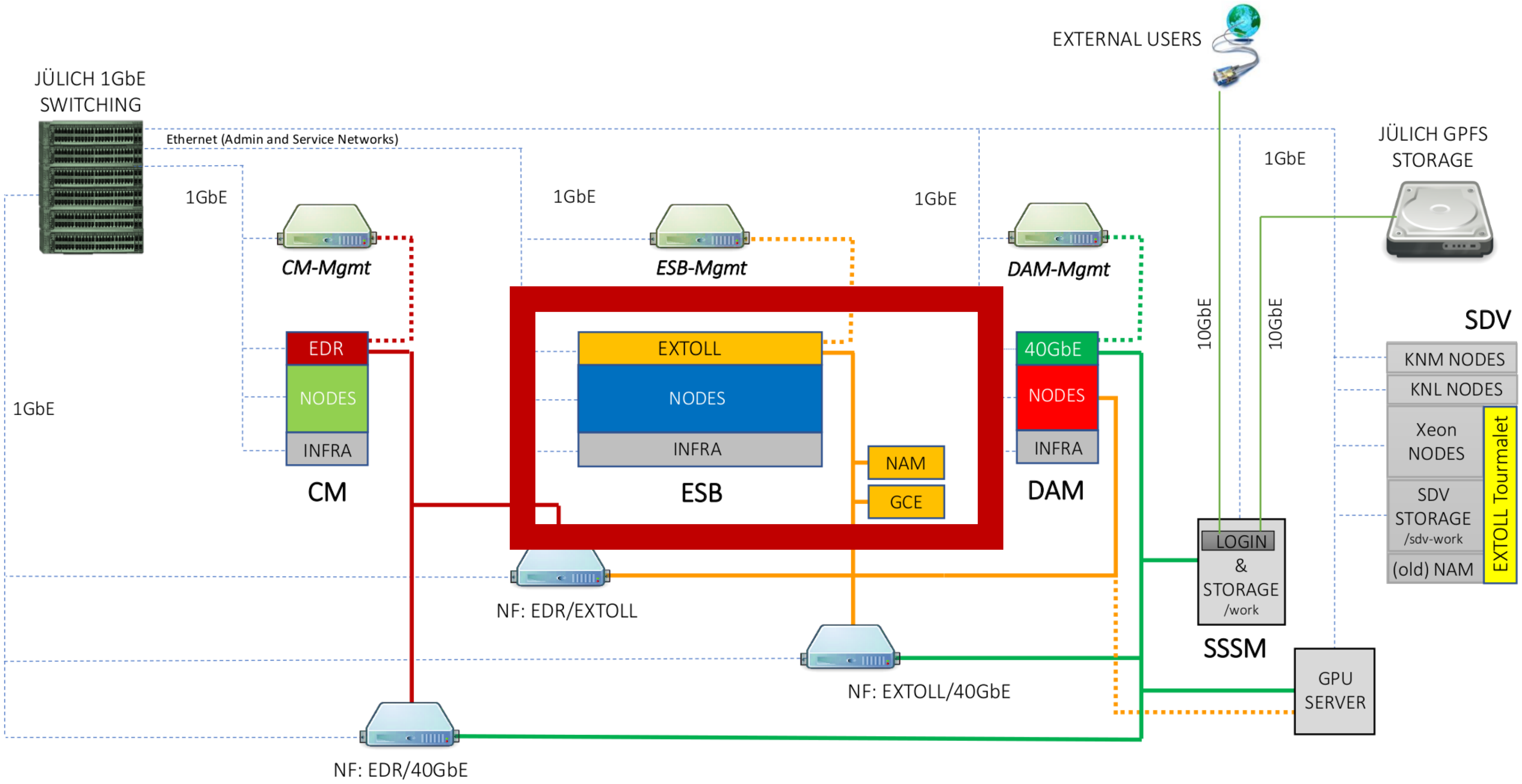
Projects

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).

ATTRACT HIOS

Remember the NAM

DEEP-EST Prototype – Schematic Network Overview



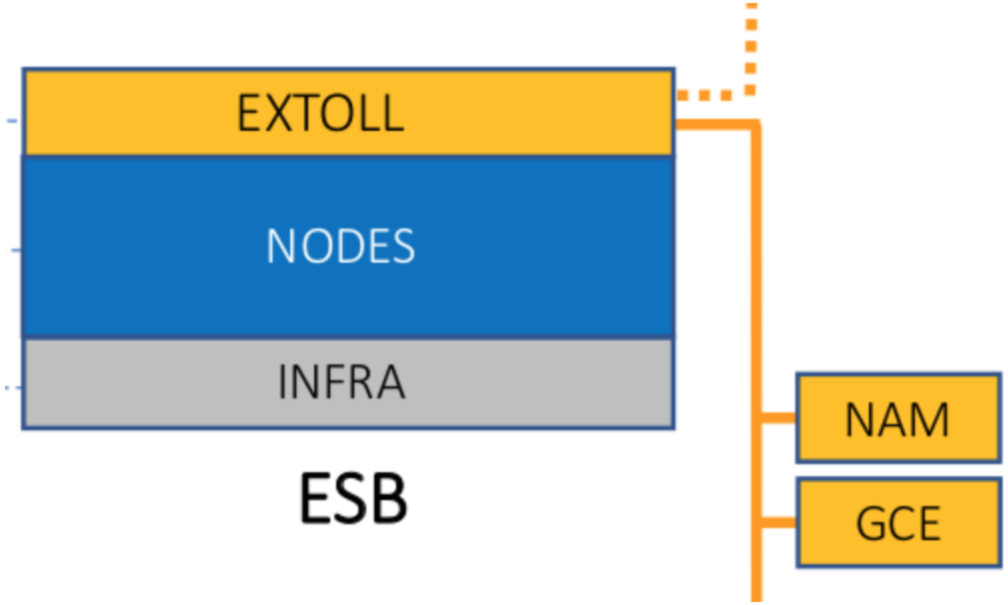
Extoll Network Attached Memory

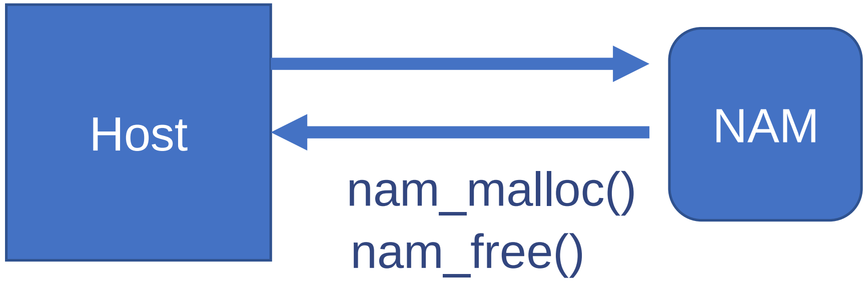
- FPGA-based solution to provide
  - Anther layer In Memory Hierarchy
  - Persistent / shared

- Basic Functionality
  - Allocate/Free/put/get
  - RMA over Extoll

- Connectivity
  - Extoll’s links
  - QSFPs for Ethernet (unutilized)

- Carries
  - 128 GBs DDR4
  - Several TBs SSDs



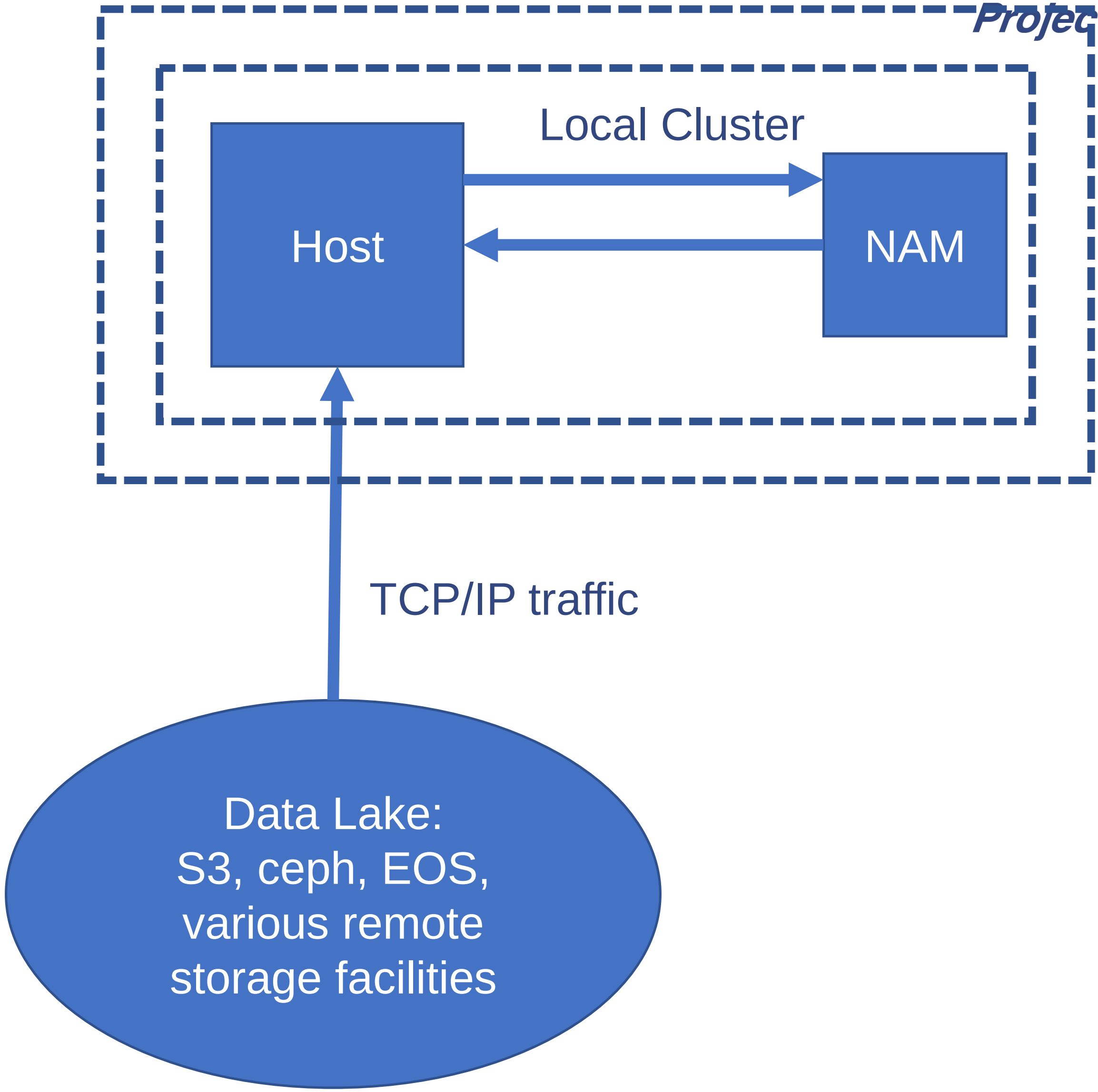


Current usage of NAM

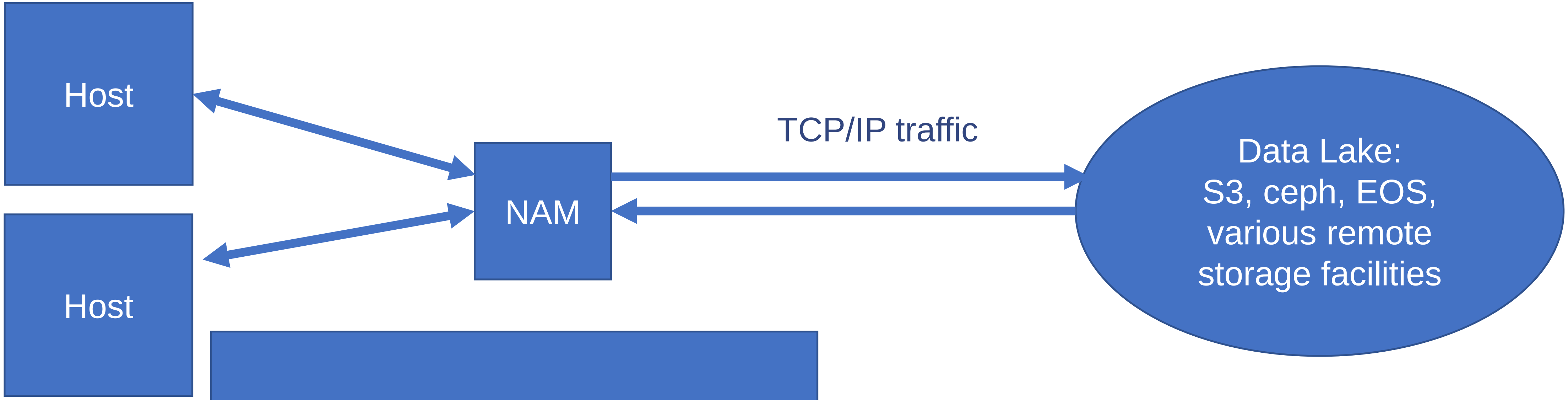
- NAM contains
  - Extoll link impl
  - Memory controller impl
  - Can be used with e.g. MPI

What do we do now on each compute host:

1) **Requesting (read/write) buffers/arrays of data**
2) **Compression/decompression**
3) **Ser/Deser**
4) **Compute/Offload to GPU/etc…**



Foreseen usage of NAM – ATTRACT HIOS



HIOS POC:

- Utilize Ethernet links
- UDP/IP payload only
  - May be instead do SOCs?
- Simple working RX/TX implementations
- WIP

What we could do differently:

1) Read data from EOS to NAM
2) Code/decode on the NAM
3) Compress/decompress on NAM
   1) Change the algorithm to be robust
4) [already there!]Use MPI to transfer from NAM to device mem just once.
   1) Dev = cpu mem/gpu mem/etc…