COKA is dead, long live COKA!

(Computing On Kepler Architectures)

Laura Cappelli, Enrico Calore, Andrea Miola, Concezio Bozzi, Sebastiano Fabio Schifano

INFN and University of Ferrara, Italy

29/05/2025

The COKA Project

HPC @ INFN Ferrara

Long-lasting experience in computational physics:

- Several HPC systems have been designed, implemented and operated to solve specific problems (e.g., APE, Janus).
- In the last 15 years, due to **higher costs in producing custom ASIC** processors, we have focused on:
  - technology tracking;
  - benchmarking;
  - exploitation to speed-up scientific simulations;

  of **off-the-shelf hardware**, spanning from ARM many-core CPUs, to GPUs and FPGAs as compute accelerators.

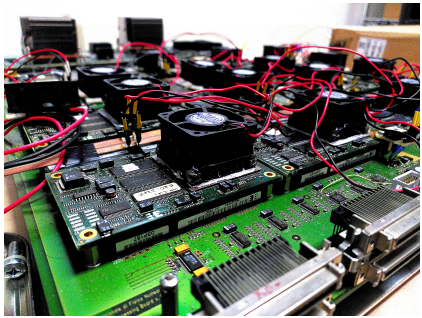
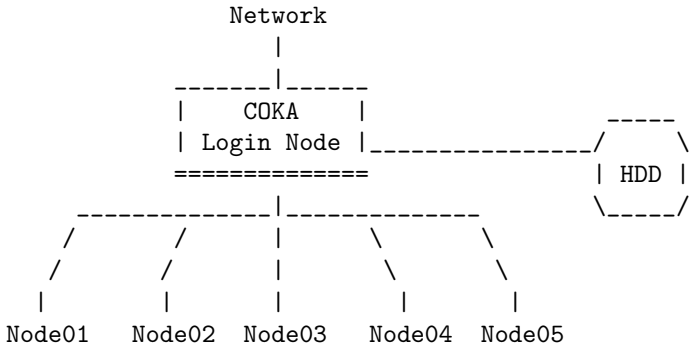


Figure: One board of APEnext

COKA Cluster

The **COKA Cluster Project** started in 2014:

- Funded by University and INFN of Ferrara for an on-premises HPC system;
- The front-end hosts the storage and 5 compute nodes with GPUs;
- Additional heterogeneous compute nodes were attached to benchmark novel accelerators;



The First Cluster Architecture

Each of the first 5 nodes is equipped with:

No.	Device	Model	Architecture
2×	CPU	Intel Xeon E5 2630	(Haswell)
8×	2xGPU	NVIDIA Tesla K80	(Kepler)
2×	IB NIC	Mellanox ConnectX-3	(56Gb/s FDR)

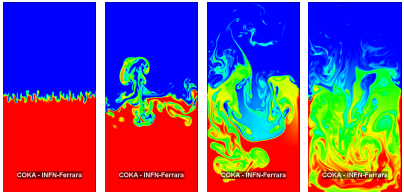
Double-precision computing performance: ≈ 100 TFLOPs.

The most powerful cluster installed in the premises of an Italian University... at the time!

Applications

COKA is extensively used for:

- **Theoretical physics simulations**, e.g., Lattice-Boltzmann Models or Lattice QCD Simulations [1], [2], [3], [4];
- **Experimental data processing and analysis**, e.g., cosmological data analysis concerning CMB [5], [6];
- **Technology tracking** of hardware accelerators [7], [8], [9];
- **AI workloads**, e.g., medical imaging [10] or QML for HEP applications [11];



Since the commissioning of COKA, the SLURM scheduler has accounted for:
179 users, more than **2.1M of Jobs** completed; **1.99M CPU Core/h** and **0.72M GPU/h**.

What next?

COKA is dead, long live COKA!

COKA outlived its original mission and it has been continuously operational since almost ten years.

The original system is now being decommissioned and replaced with:

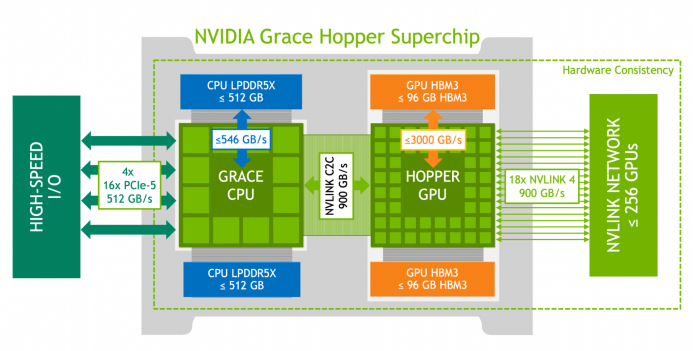
- **New hardware**:
  - A new front-end
    - Dell PowerEdge R7525 2×CPU AMD7413, 512GB RAM, 16×2.4TB disks;
  - New compute nodes (see next slide);
  - New network devices:
    - 2× 36-port Mellanox Infiniband EDR switches
    - 1× 10Gbit/s Ethernet
- A completely **refactored software ecosystem**.

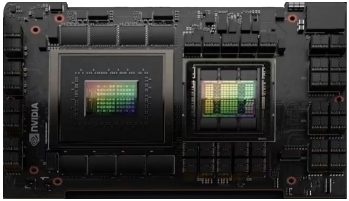
The New Compute Nodes

New compute nodes:

- 6× Server with NVIDIA **GraceHopper** Superchip (72-core Arm CPU + Hopper GPU);
  - 1× First prototype (EU MESEO Project)
  - 2× Cosmo CMB analysis (RELiCS)
  - 3× Quantum computing (Quantum PNR)
- 4× IBM Power AC922 System with 2× IBM **POWER9** 16-core CPUs;
- 1× Gigabyte G493-SB1 with 2× Intel Xeon Gold 6548Y+ 32-core CPUs, to **host up to 10 FHFL dual-slot PCIe card**;

NVIDIA Grace Hopper Superchip





Cluster Setup

Cluster setup

Work has already been done on the following topic:

- **OS** definition and installation;
  - Main services configuration;
- **Network** configuration:
  - Network definition;
  - Firewall;
  - DHCP and DNS;
- **File System** configuration;
- **Slurm** Workload manager configuration;
- Policy definition about **AuthN and AuthZ**;

We have to address several other topics (i.e., installing application on different architectures, automatization of the process, ...)

OS: RedHat based vs Debian based

The **old COKA** front-end still uses CentOS7;

Diskless compute nodes boot CentOS7 using PXE/TFTP from the front-end.

The **new COKA** uses Ubuntu 24.04;

An architecture-dependent version of the OS is installed on each node.

ubuntu

We have evaluated that all the applications we need are available as deb packages.

Services containerization

In the old cluster all the services are installed on the the front-end:

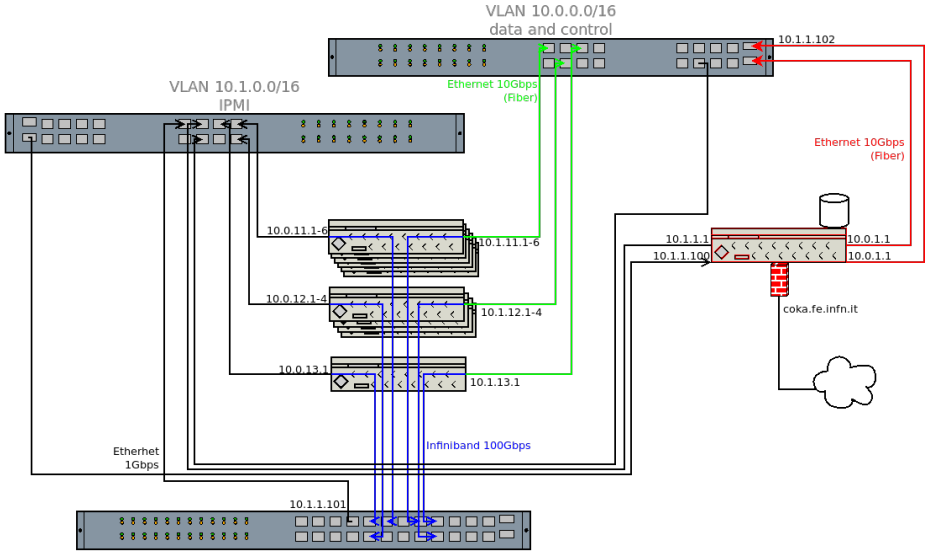
- We experienced **issues in the services update** (Slurm);
- We could benefit from containerizing some services;

Docker could help us with the following:

- Fast updates and rollbacks;
- Improving testing and development;
- Easily deploying the environment across multiple systems to improve portability on heterogeneous architectures;
- Improving **automation** of service installation, i.e., using **Docker Compose**;

So far, we have containerized the **DHCP** and **DNS** services, as well as **Slurm**.

COKA Networks Architecture



Firewall

The cluster is protected by **two firewalls**:

- The INFN Section's firewall, i.e., both the front-ends are inside the INFN Section's LAN;
- The second firewall installed on the front-end only accepts connections from the INFN network or from known IP addresses.

We use **iptables** on both the front-ends:

- We evaluated the use of *nftables*, the successor of iptables, but...
- ... Docker only uses and supports iptables;
- We might use *Uncomplicated FireWall* (UFW), an Ubuntu program designed to easily write firewall rules.

DHCP & DNS

Previously, on the **OLD cluster**:

- We didn't use the DNS: the nodes were a limited number...
- We configured *isc-dhcp-server*, but this software is no longer supported;

On the **NEW cluster**:

- As the number of nodes increases, a DNS could be useful;
- *isc-kea*, the successor of *isc-dhcp-server*, is probably overcomplicated;

We have created a Docker Image with **dnsmasq**, a lightweight DNS and DHCP server designed for small-scale networks:

- Easy to install and configure;
- Thanks to the Docker container, we can easily migrate to another server;

Storage and File System

Slot 0-1:

- 2× 2.4TB HHD in RAID 1
- Used to store the OS

Slot 2-15:

- 14× 2.4TB HHD in RAID 6
- Used to store home directories

Slot 23:

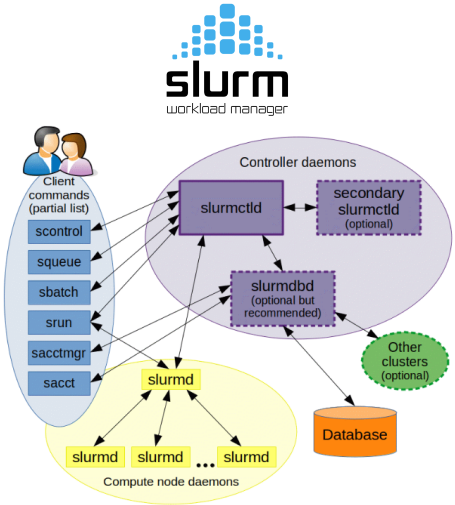
- 1× 2.4TB SSD
- Not allocated

Front-end and compute nodes share home directories via **NFS**.

Experimented some issues due to the different architectures (i.e., using conda environment)

Slurm

- We use **Slurm** for cluster management and job scheduling;
- To simplify future updates, we encapsulated almost all the Slurm components inside Docker containers:
  - One container for **slurmctld**;
  - One container for **slurmdb**;
  - One container for the database (**MariaDB**);
- **Slurmd** is installed directly on the front-end and on the compute nodes;
- To share the configuration files between all the component we use a Docker NFS volume.



Conclusion and future works

Conclusion and future works

The installation has just started, but we have successfully configured:

- The front-end with a basic setup and some compute nodes;
- The network and its services;
- Slurm (we've already run the first job on the GPU!)

We are still in the experimental phase, and the new cluster is not yet in production.

Future work:

- Consolidate the presented services;
- Implement authentication and authorization using the Ferrara section DB of users;
- Automate package installation on different hardware (i.e., using Ansible);

We are going to share our work in a repository on Baltig...

References

[1] E. Calore et al., *Optimization of lattice Boltzmann simulations on heterogeneous computers*, International Journal of High Performance Computing Applications, 33(1), pp. 124–139, 2019, doi: *10.1177/1094342017703771*

[2] Bonati, et al., *Portable multi-node LQCD Monte Carlo simulations using OpenACC*, International Journal of Modern Physics C, 29(1), 2018, doi: *10.1142/S0129183118500109*

[3] E. Calore, et al., *Massively parallel lattice-Boltzmann codes on large GPU clusters*, Parallel Computing, vol. 58, pp. 1–24, 2016, doi: *10.1016/j.parco.2016.08.005*

[4] E. Calore et al., *Performance and portability of accelerated lattice Boltzmann applications with OpenACC*, Concurrency and Computation: Practice and Experience, 28(12), pp. 3485–3502, 2016, doi: *10.1002/cpe.3862*

[5] G. Zagatti et al., *A halo model approach to describe clustering and emission of the two main star-forming galaxy populations for cosmic infrared background studies*, Astronomy & Astrophysics, vol. 692, pp. A190, 2024, doi: *10.1051/0004-6361/202451424*

[6] P. Campeti et al., *From few to many maps: A fast map-level emulator for extreme augmentation of CMB systematics datasets*, accepted in Astronomy & Astrophysics, 2025, doi: *10.48550/arXiv.2503.11643*

[7] S.F. Schifano, et al., High throughput edit distance computation on FPGA-based accelerators using HLS, in Future Generation Computer Systems, vol. 164, 2025, doi: *10.1016/j.future.2024.107591*

[8] E. Calore et al., *FER: A Benchmark for the Roofline Analysis of FPGA Based HPC Accelerators*, in IEEE Access, vol. 10, pp. 94220–94234 (2022), doi: *10.1109/ACCESS.2022.3203566*

[9] E. Calore et al., *ThunderX2 Performance and Energy-Efficiency for HPC Workloads*, in Computation, vol. 8(1):20, doi: *10.3390/computation8010020*

[10] G. Minghini et al., *An HPC Pipeline for Calcium Quantification of Aortic Root From Contrast-Enhanced CCT Scans*, in IEEE Access, vol. 11, pp. 101309–101319, 2023, doi: *10.1109/ACCESS.2023.3315734*

[11] M. Argenton et al., *Charged particle tracking with quantum graph neural networks*, Proceedings of Science (ICHEP 2024), 2024, doi: *10.22323/1.476.0997*

Thanks for Your Attention