UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

AREA
SCIENCE PARK

New approaches in scientific computing: the design of a cloud-ready HPC infrastructure

Isac Pasianotto

Supervised by: dr. *Ruggero Lot*
Cosupervised by: prof. *Stefano Cozzini*

Graduation session: M.Sc. in *Data Science and Scientific Computing*

20th September 2024

University of Trieste

🤔 What is this thesis about?

Investigating the possibility to adopt a cloud-native approach based on container orchestration in a private HPC cluster to provide it in a SaaS fashion

What is *“Cloud Computing”*

“Cloud computing has been coined as an umbrella term to describe a category of sophisticated **on-demand computing services** *initially offered by commercial providers”*

– R. Buyya et al. Cloud Computing: Principles and Paradigms

What is *“Cloud Computing”*

*“Cloud computing has been coined as an umbrella term to describe a category of sophisticated **on-demand computing services** initially offered by commercial providers”*

– R. Buyya et al. Cloud Computing: Principles and Paradigms

☁️ ***for us:***

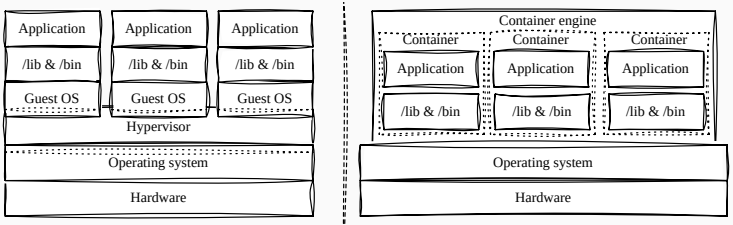
*Cloud Computing is the result of the **evolution** of the computing concept **driven by the technology** improvements and by users’ requirements.*

Virtualization

- Isolated environment with its own OS and resources
- **Overhead** for the hypervisor

Containerization

- Host's processes → they share the Kernel with the host
- Build over namespaces, cgroups, and capabilities



⚖️ Performance/Usability trade-off

Container orchestrator: simplify the interaction with the cluster for both the user and the administrator

- Automatic scaling and load balancing
- Availability, self healing and fault tolerance
- Standardized API → portability
- ...

⚖️ **Performance/Usability trade-off**

Container orchestrator: simplify the interaction with the cluster for both the user and the administrator

- Automatic scaling and load balancing
- Availability, self healing and fault tolerance
- Standardized API → portability
- . . .

☠️ *First attempts in this direction were catastrophic*

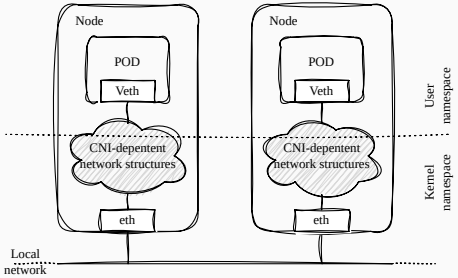
⏳ *Network latency rapidly became the bottleneck of the system*

📝 *Def.*

A CNI plugin is responsible for inserting a network interface into the container network namespace (e.g., one end of a virtual ethernet (veth) pair) and making any necessary changes on the host (e.g., attaching the other end of the veth into a bridge).

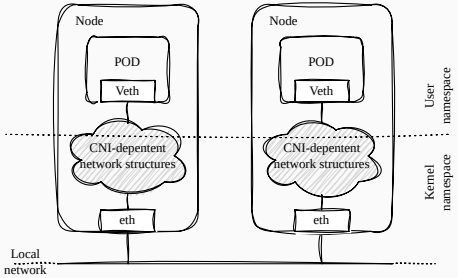
– Red Hat sysadmin's guide

🤯 *“And making any necessary changes on the host . . .”*



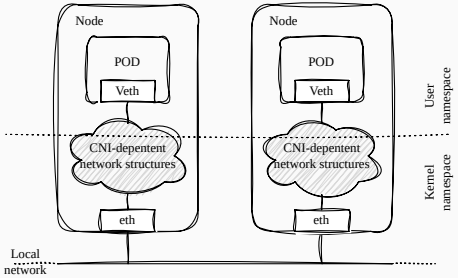
🤯 *“And making any necessary changes on the host . . .”*

➡️ *Calico*: BGP routing



Calico: BGP routing

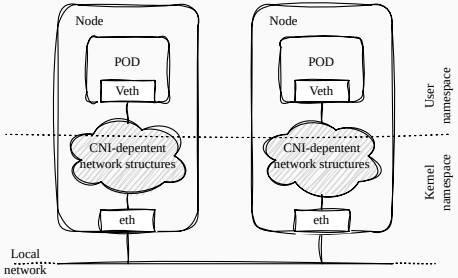
➡️ *Flannel*: VXLAN encapsulation



Calico: BGP routing

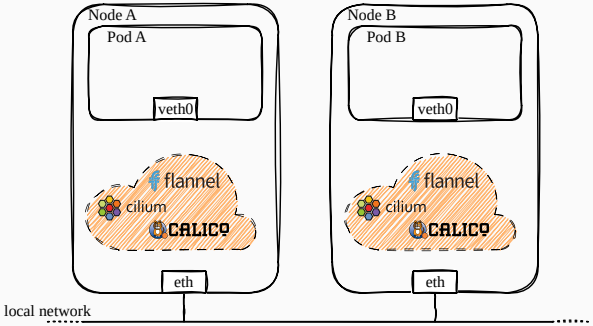
Flannel: VXLAN encapsulation

➡️ *Cilium*: eBPF (sandbox technology within the Linux kernel)



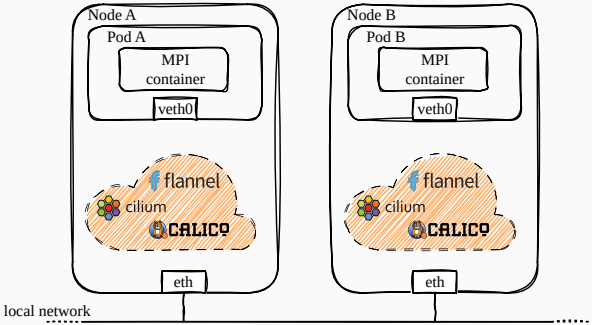
💡 *The idea:*

- Spawn 2 pods in 2 (different) nodes



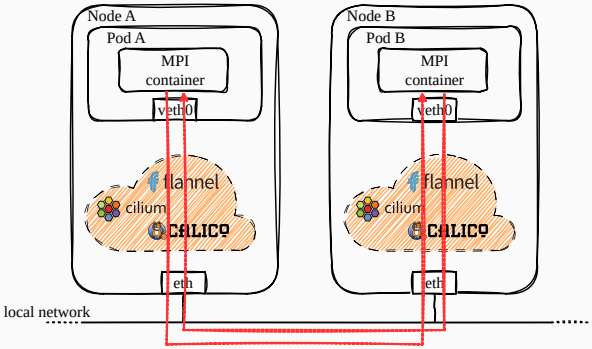
💡 *The idea:*

- Spawn 2 pods in 2 (different) nodes
- Each pod will have a container running an MPI process



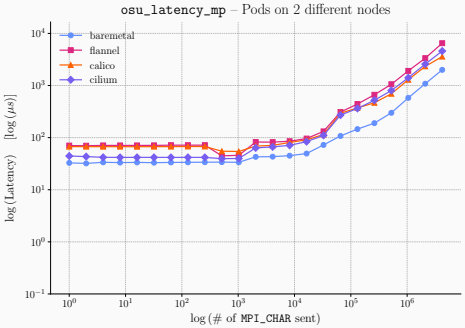
💡 *The idea:*

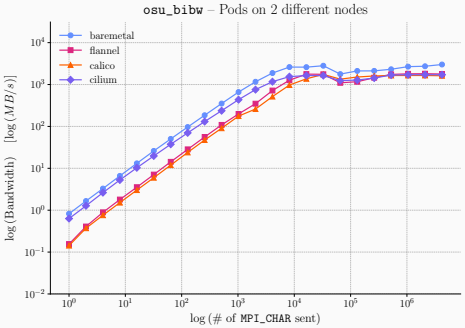
- Spawn 2 pods in 2 (different) nodes
- Each pod will have a container running an MPI process
- Perform ping-pong tests between the 2 pods 🏓



Results

🔎 Nodes with ***25 Gbps*** ethernet ports





$[\mu s]$	Bare metal	Calico	Flannel	Cilium
Min	32.7	67.5	70.4	44.4

$[MB/s]$	Bare metal	Calico	Flannel	Cilium
Max	3.016e3	1.62e3	1.78e3	1.71e3

🤌 What is Dask?

- Flexible library for parallel computing in Python
- Allow to *distribute the computation* **without managing the parallelism**

🤌 What is Dask?

- Flexible library for parallel computing in Python
- Allow to *distribute the computation* **without managing the parallelism**

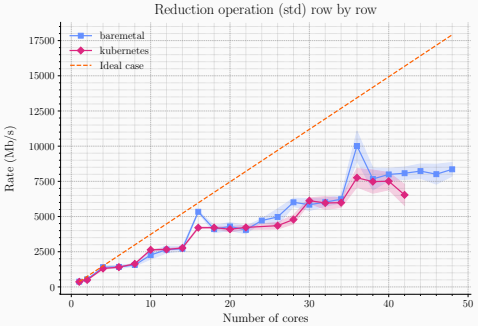
🤌 *What is Dask?*

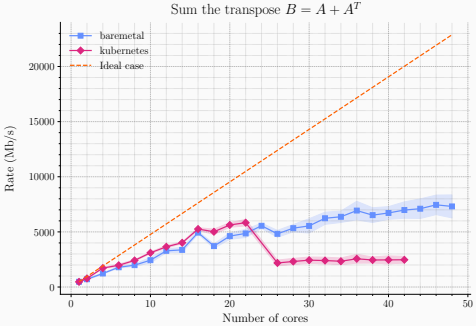
- Flexible library for parallel computing in Python
- Allow to *distribute the computation* **without managing the parallelism**

💡 *Our solution to develop a parallel code to solve most common problems in data science*!

The results

🔎 Nodes with *2* × *12 cores* each.





🤔 Can we use Kubernetes to deploy and manage HPC applications?

🤔 Can we use Kubernetes to deploy and manage HPC applications?

- 🥳 *Yes!*, but with *some limitation.*

📝 To sum up:

🤔 Can we use Kubernetes to deploy and manage HPC applications?

- 🥳 *Yes!*, but with *some limitation.*
- Times are **not** mature yet for a full migration
- ⚠️ In the **communication bounded** application, the performance gap is **not acceptable**
- In all the other cases, the overhead is negligible

🤔 Can we use Kubernetes to deploy and manage HPC applications?

- 🥳 *Yes!*, but with *some limitation.*
- Times are **not** mature yet for a **full migration**
- ⚠️ In the **communication bounded** application, the performance gap is **not acceptable**
- In all the other cases, the **overhead** is **negligible**
- **Kubernetes** comes with a lot of benefits that improve the user experience
- 🤞 The technology evolution of last years make us be **optimistic** about the future

Thank you for your attention