The User Experience in IBiSCo Data Center

L. Carracciuolo (CNR), P. R. Hegde (KTH), G. Sabella (UniNa, INFN), A. Tortora (UniNa)

gianluca.sabella@unina.it

Parallel IBiSCo activities: Consolidation and Experimentation

- Consolidation of IBiSCo UniNa resources:
  - Implementation of services to enhance user experience.
  - Creation of a Wiki page and collaborative computing platform.
  - Establishment of contacts to communicate with the resource Admin team.
- Experimental activity for dynamic resource provisioning:
  - Provide flexible and scalable resources to users.
  - Utilization of container and Kubernetes infrastructures to support the data analysis.
  - Transition of IBiSCo activities to ICSC project resources.

IBiSCo HPC/HTC resources access and use

Steps to use resources:

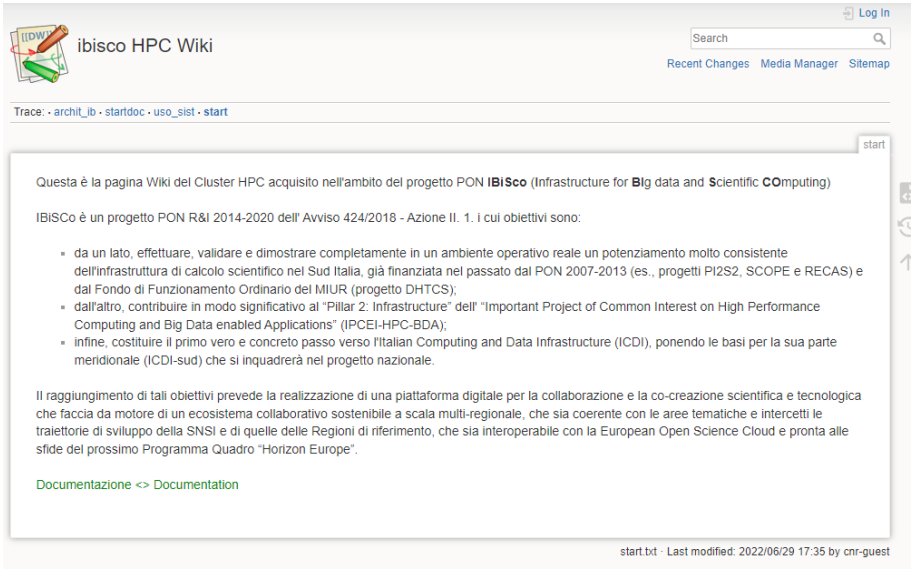
1. User account creation
   - By sending a request to system admins
2. SSH protocol to access resources
3. SLURM tool to run or terminate jobs, submit script, etc. …
   - Srun
   - Sbatch
   - Scancel
   - …

SLURM - Simple Linux Utility for Resource Management

- *Flexible job scheduling*: it enables users to schedule jobs
- *Dynamic resource management*: it allocates resources to jobs as needed, optimizing resource utilization
- *Scalability and reliability*: it efficiently manages clusters of varying sizes
- *Customizable configurations*: it offers extensive configuration options
- *High-Performance Computing* (*HPC*): it is widely used in scientific and engineering applications that require extensive computational resources
- *Batch processing*: it is suitable for batch processing tasks, executing multiple jobs sequentially or in parallel.

The IBiSCo Wiki Page

SPIN ISASI
Consiglio Nazionale delle Ricerche
Università degli Studi di Napoli Federico II
INFN Istituto Nazionale di Fisica Nucleare



Cluster architecture

Account registration

Job submission

https://ibiscohpc-wiki.scope.unina.it/

On-site experimentation: High throughput data analysis

Goal: to provide the users with an infrastructure that represents a tradeoff between deployment speed-flexibility, resource efficiency and service performance

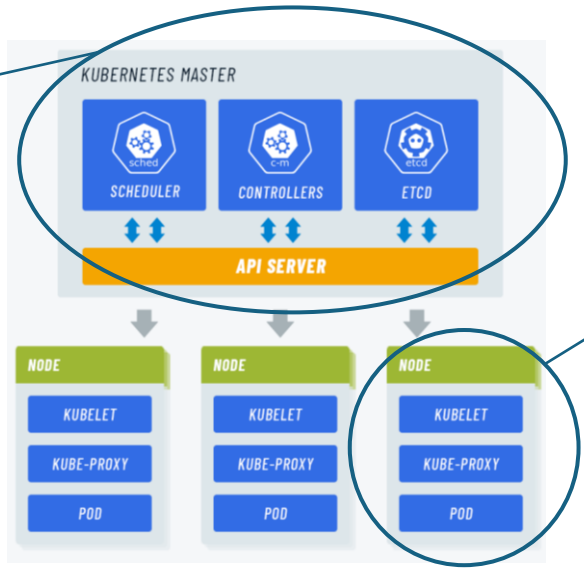
Solution being tested: the use of container technology (via Docker 20.10) that runs the applications and the Kubernetes tool for orchestration

Kubernetes

Local testbed infrastructure provides 6 nodes, orchestrated via Kubernetes (1.26.3):

1 node plays master role

Master node



Worker node

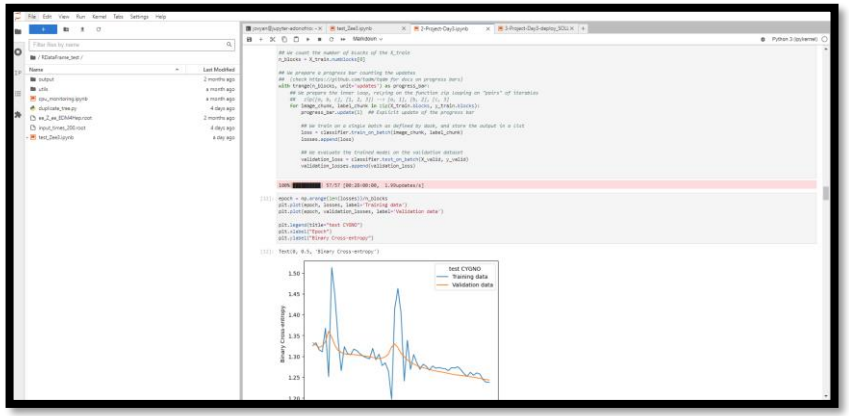
5 nodes play worker role

The **Master node** coordinates and manages the entire Kubernetes infrastructure. It includes the API Server, the Controller Manager, the Scheduler, and etcd.

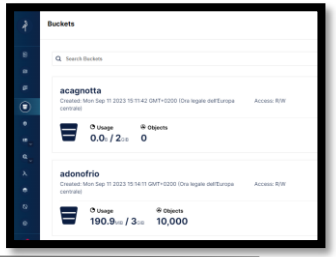
The **Worker nodes** represent the nodes of the cluster where containers are executed. Each one hosts the Kubelet, Kube-proxy and the containers, providing the actual computational capacity of the cluster.

Tools for the high throughput data analysis

Through the Kubernetes infrastructure created, we set up a prototype to respond to previous requests, offering the user computing resources. The tools used were:



- ✓ **Jupyter Hub/Lab**: interactive, web-based development environment capable of managing multiple accesses
- ✓ **DASK:** Open-source and flexible Python library for parallel computing that can use the most popular batch systems (SLURM, HTCondor, etc. …) or is provided with a native scheduler if necessary (our choice)



- ✓ **S3:** Object Storage designed to be deployed anywhere: public or private cloud, bare metal infrastructure, orchestrated environments and edge infrastructures.

User experience

Support to supply users with the tools they needed (ROOT, editors, etc. …)

The infrastructure was tested using several realistic high energy physics workflows:

- The overall execution time is the metric chosen to evaluate and compare the performance of the standard "serial" model and the parallel approach to data analysis, based on the new infrastructure
- Preliminary feasibility studies show that the execution time improves significantly exploiting the DASK distributed approach on the analysis platform

User experience

Support to supply users with the tools they needed (ROOT, editors, etc. …)

The infrastructure was tested using several realistic high energy physics workflows:

- The overall execution time is the metric chosen to evaluate and compare the performance of the standard "serial" model and the parallel approach to data analysis, based on the new infrastructure
- Preliminary feasibility studies show that the execution time improves significantly exploiting the DASK distributed approach on the analysis platform

More here

Conclusions and Outlooks

In conclusion, our goal is to provide IT services to users in a rapid and efficient manner, while ensuring scalability and adaptability to meet evolving needs

Our ongoing efforts to enhance and optimize the user experience through experimentation with new technologies are also aimed at facilitating the transition from IBiSCo to ICSC.

Thank you for your attention