ISSN/2499-5347

M. ARTIOLI, S. BEOZZO

Department of Energy Efficiency Unit
Energy Efficiency Systems, Projects and Services Division
Bologna Research Centre

MINI BEOWULF CLUSTER FOR PERFORMANCE AND ENERGY TOOLS

Building a cost-effective test-bed for assessing software performance

RT/2025/15/ENEA

ENEA

ITALIAN NATIONAL AGENCY FOR NEW TECHNOLOGIES, ENERGY AND SUSTAINABLE ECONOMIC DEVELOPMENT

M. ARTIOLI, S. BEOZZO
Department of Energy Efficiency Unit
Energy Efficiency Systems, Projects and Services Division
Bologna Research Centre

MINI BEOWULF CLUSTER FOR PERFORMANCE AND ENERGY TOOLS

Building a cost-effective test-bed for assessing software performance

RT/2025/15/ENEA

ITALIAN NATIONAL AGENCY FOR NEW TECHNOLOGIES,
ENERGY AND SUSTAINABLE ECONOMIC DEVELOPMENT

MINI BEOWULF CLUSTER FOR PERFORMANCE AND ENERGY TOOLS

Building a cost-effective test-bed for assessing software performance

M. Artioli, S. Beozzo

Abstract
This work provides a summary, including images, of the development phases of a small distributed computer to be used as a testbed for energy performance measurements. The rationale for the specific design choices is also explained.

Keywords: *HPC, Cluster, Beowulf, performance, power.*

Riassunto
Questo lavoro riporta un riepilogo, anche con immagini, delle fasi di realizzazione di un piccolo calcolatore distribuito da utilizzare come banco di prova per misure di prestazione energetica. Vengono anche illustrate le motivazioni delle particolari scelte progettuali.

Parole chiave: HPC, Cluster, Beowulf, prestazioni, potenza.

Contents

List of figures

MINI BEOWULF CLUSTER FOR PERFORMANCE AND ENERGY TOOLS

BUILDING A COST-EFFECTIVE TEST-BED FOR ASSESSING SOFTWARE PERFORMANCE

1. MOTIVATION

1.1. Regular HPC cluster

An HPC (High-Performance Computing) cluster (Fig. 1) is a networked collection of computers (often called nodes) designed to work together to perform complex computations at high speed [1]. Its typical components and characteristics are:

a. Head Node (or Master Node):

- manages the cluster;
- handles job scheduling, resource allocation, and user interfaces;
- often runs job management software (like Slurm, LSF, PBS, etc.);

b. Compute Nodes:

- are the workhorses of the cluster;
- perform the actual computations;
- are usually identical in hardware and run tasks assigned by the head node.

c. Storage System:

- it's a shared file system accessible by all nodes (like NFS, Lustre, BeeGFS, etc.);
- stores input data, output results, and user files;

d. High-Speed Interconnect:

- it's a low-latency, high-bandwidth network (like InfiniBand, Omni-Path, etc.);
- ensures fast communication between nodes, crucial for parallel/distributed computing;

e. Job Scheduler / Resource Manager:

- it's a software suite that queues, schedules, and monitors jobs (see Head Node);
- allocates resources efficiently across users and tasks.

Typical features of a regular HPC cluster are mainly:

- Parallel/Distributed Processing: tasks are split across multiple nodes to reduce computation time;
- Batch Processing: users submit jobs to be run when resources are available;
- Environment Modules: for configuring and managing software versions and dependencies.

Use cases for a regular HPC cluster are many, but all have the common necessity to carry massive computations which are not possible within the resource constraints of conventional computers (like memory or disk space) or which would otherwise take too long to accomplish.

Very typical examples are:

- Scientific simulations (e.g., climate modeling, physics);
- Data analysis (e.g., genomics, machine learning);

- Engineering (e.g., CFD, FEA);
- Financial modeling.

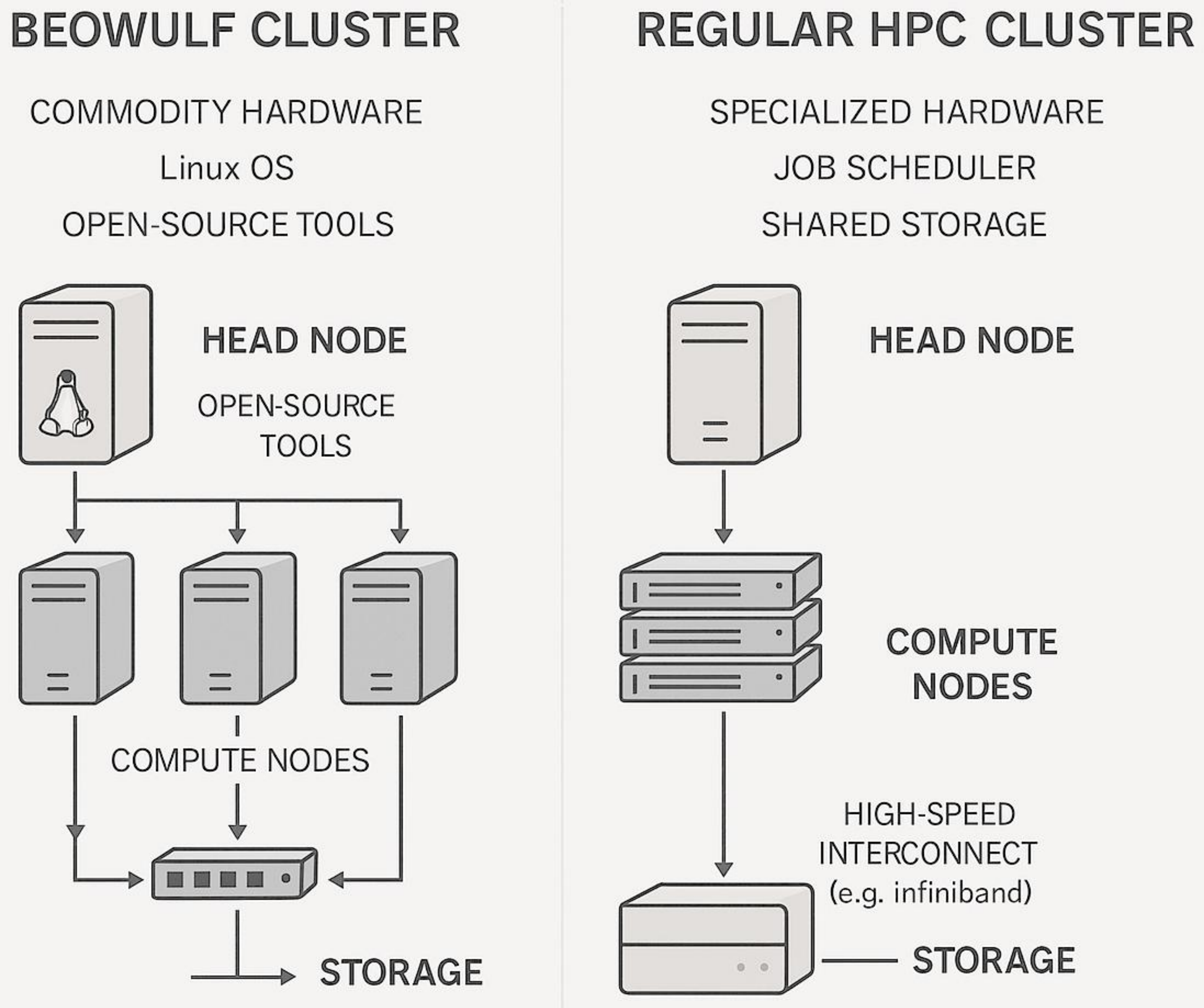

Fig. 1: Basic sketch of Beowulf and regular clusters.

1.2. Beowulf cluster

A *Beowulf cluster* [2] is a specific type of HPC cluster (Fig. 1) that is built using commodity hardware and open-source software, typically Linux. It's designed to be a cost-effective way to achieve parallel computing power by linking together standard computers.

Key Characteristics of a Beowulf Cluster are

a. Commodity Hardware:

- uses off-the-shelf PCs or servers;
- no need for specialized or proprietary hardware;

b. Linux-based operating systems:

- typically runs a common Linux distribution;
- leverages open-source tools and libraries;

c. Dedicated Network:

- nodes are connected via a private network (often Ethernet);
- also built with off-the-shelf components (like switches or consumer network cards) ;

d. No Specialized Management Software:

- unlike commercial HPC systems, Beowulf clusters often use simple scripts or lightweight schedulers;
- they're not meant for multi-users facility;

e. Scalability:

- easily scalable by adding more nodes;
- designed to be flexible and adaptable.

The concept was popularized in the mid-1990s by Thomas Sterling and Donald Becker at NASA. It was a revolutionary idea at the time because it allowed researchers to build powerful computing systems without the high cost of supercomputers. Its typical use cases are still

- Academic research;
- Small-scale scientific simulations;
- Educational environments;
- Prototyping parallel applications.

One of the motivating differences between *regular* and *Beowulf* clusters is that the former ones are far more complex infrastructure that normally needs to be administered by a dedicated team and follow given user policies. While the latter ones, though far less powerful, are easier to administer and in total control by the user (or small group of users).

1.3. Aims

The goal is to have a cost-effective experimental hardware infrastructure where modern operating systems and applications can run. Here, parallel applications can be prototyped, performance profilers can be tested in a totally controlled environment, avoiding collisions and unwanted interactions with other uses or users outside the experimentation. This is particularly true for assessing power/energy issues of software.

2. REALIZATION

2.1. The plot

The plot for building the Beowulf cluster initially was simply to interconnect a small set of identical desktop PC with a network switch. They are recycled, quite old but the main specs let them to be used for the purpose:

- CPU: Intel i7-2600 CPU @ 3.40GHz (4 physical cores)
- RAM: 8 GB
- Ethernet: 1 Gb/s

To easily deploy and change the experimental setup, a "Preboot Execution Environment" (PXE) boot configuration has been planned.

PXE boot is a method that allows a computer to boot up using a network interface, instead of a local storage device like a hard drive or SSD. In this way, all nodes of the cluster boot up with the same operating system image that don't need to be installed but is hosted on a network server.

2.2. Stripping down the plot

With PXE booting, the hard disks of the nodes could be removed, leaving only motherboard (MB) and power supply unit (PSU) in the PC case.

Furthermore, the peak power consumption of every motherboard is around 70W, so that (theoretically) a single 500W PSU could boot and run at least 6 MBs

Also the network interface and the video connection are integrated onto the MB; as a consequence, the hardware setup can be stripped down to 6 bare MBs without case and individual PSU, powered instead by a single PSU housed in the same enclosure.

Fig. 2:Metal enclosure with plastic rails on the internal bottom and top (hidden).

2.3. Practical set up

The enclosure is a metal box recycled from a dismissed telephony rack, which already has plastic rails for inserting boards (Fig. 2).

Asus P8H67-M-EVO

Asus P8H67-M-PRO

Asus P8H77-M

Fig. 3: Three almost identical types of motherboards.

The MBs are six, not exactly of the same model but of the same chipset and computational features; they differ mostly by connection options and layout (Fig. 3). The sizes of regular ATX form factor of the board don't fit to the height of the enclosure and even so the margins on the edges are too short for the depth of the rails.

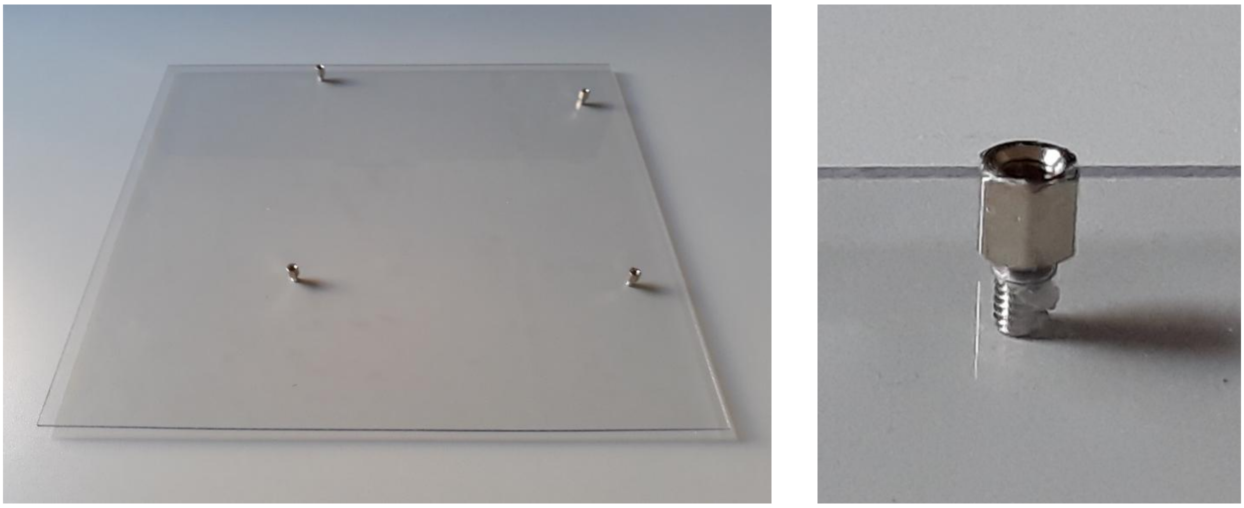

Fig. 4: Polycarbonate board with screw spacers.

To overcome this problem inexpensively, some polycarbonate boards have been cut out of the size fit for the enclosure and for the rails (Fig. 4). Polycarbonate, in fact, is easy to cut and drill, has good mechanical resistance and tolerates high temperatures. Four screw spacers will hold the MB in place (Fig. 5).

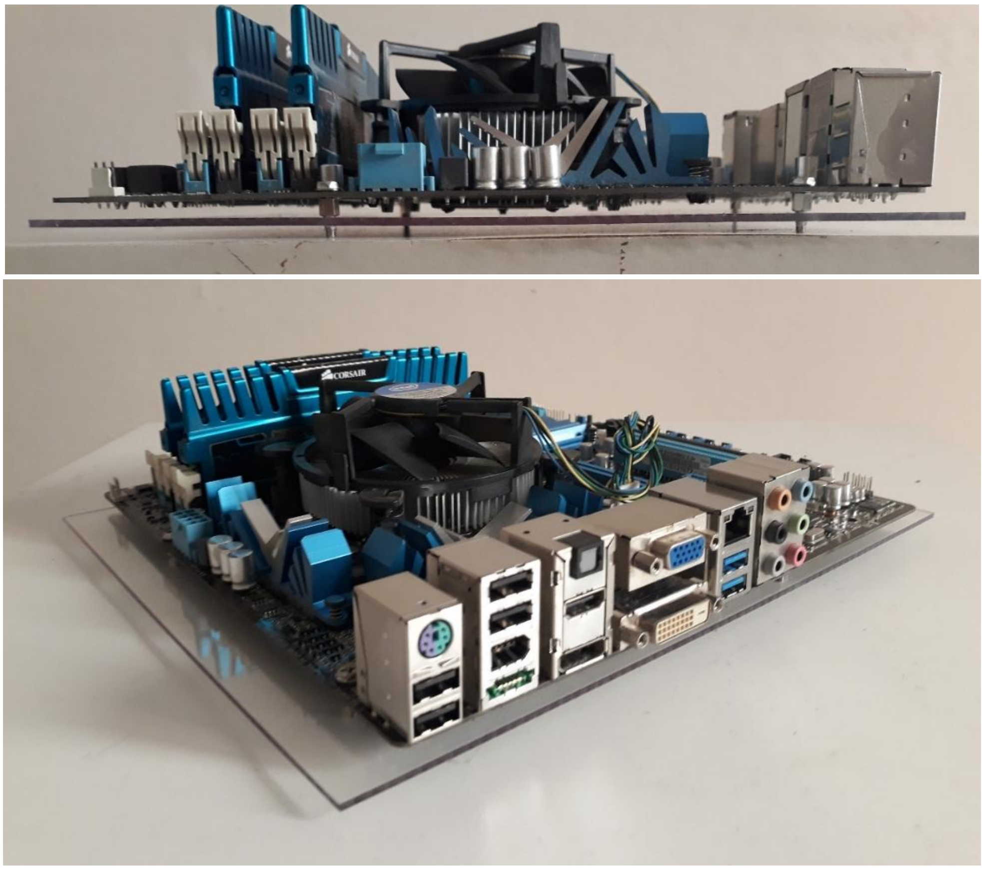

Fig. 5: Motherboard mounted on the polycarbonate slab.

With such mounting, the MB can be easily inserted into or removed from a slot of the enclosure (Fig. 6), leaving space for the fan and the cabling.

Fig. 6: Motherboard sliding into a slot.

Fig. 7: Regular power supply unit, with several connectors for additional power lines.

Power cabling has been done from a regular PSU (Fig. 7) with several connectors for additional 6-pin power lines required by each MB. The main 24-pin power line has been distributed with a series of Y splitters (Fig. 8 and 9). The ATX specifications [3] require that the PSU must wait for a *power-good* signal from the MB before switching on the power lines, but once the signal is received, it is then ignored.

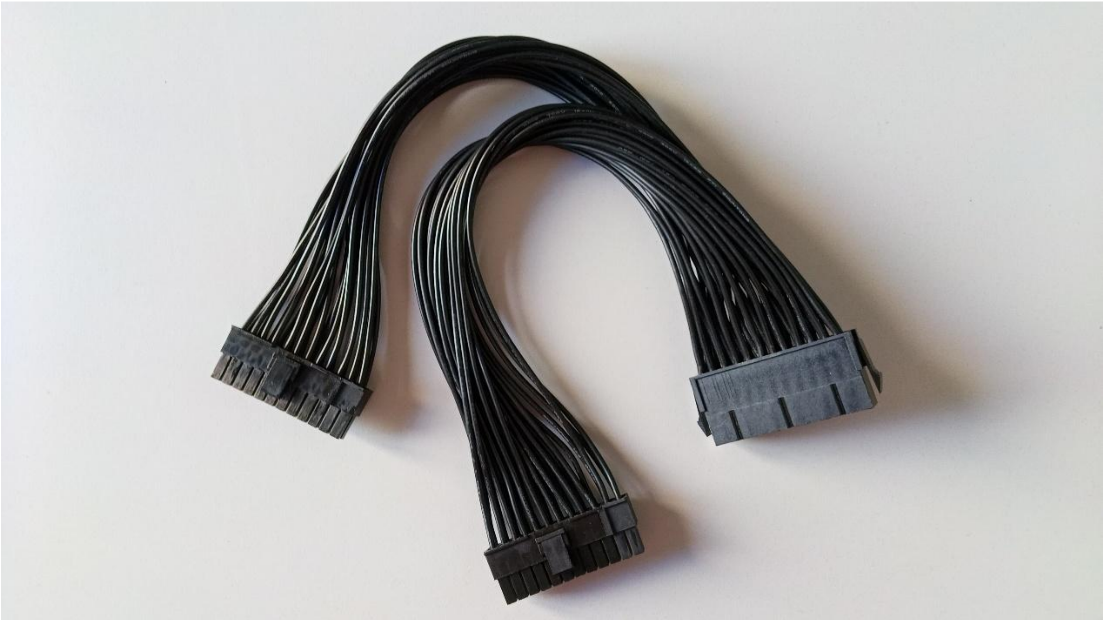

Fig. 8: Power chords.

Fig. 9: Power cabling.

The same mounting has been done for the dedicated 8-port 1 Gb/s network switch which is also powered by the same PSU with an adapter from 4-pin Molex to jack barrel (Fig. 10 and 11).

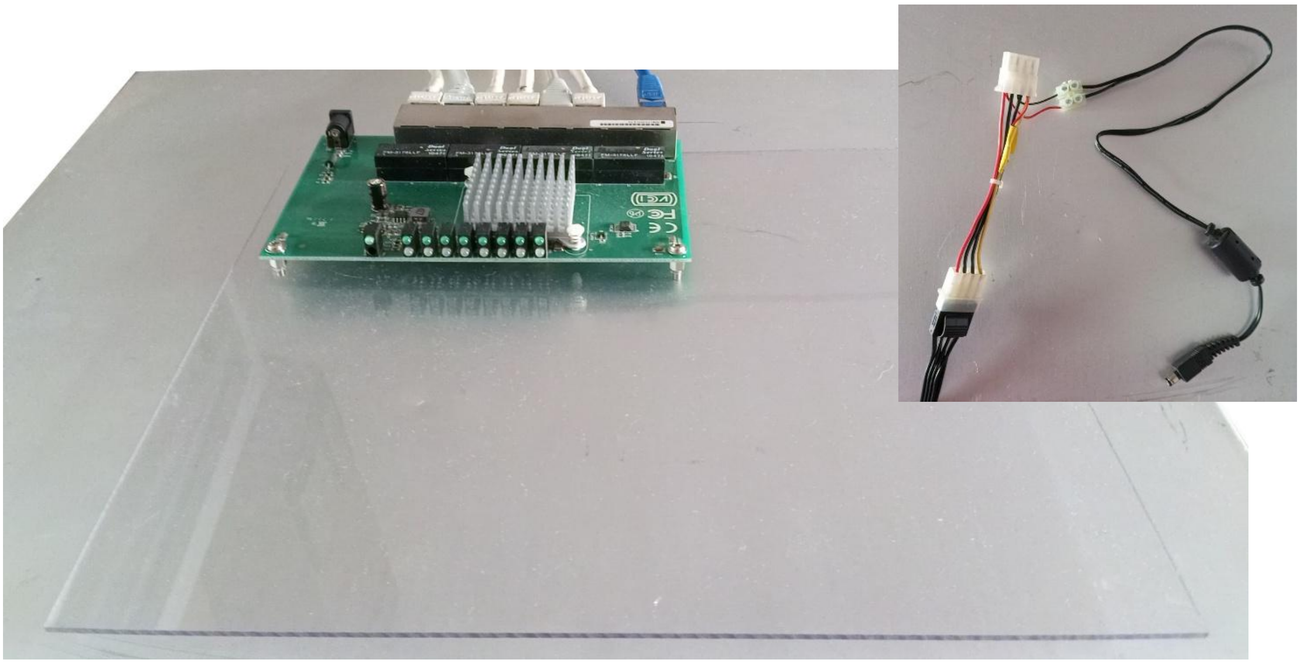

Fig. 10: Network switch and its power line adapter.

Fig. 11: Network switch mounting.

If multiple MBs are connected to the same PSU, no logic circuitry is needed to handle the power-good signaling because as soon as one of the MBs answers, the PSU powers on and keeps on.

3. RESULT

Once all the cabling is done, the result is compact and quite polished (Fig. 12).

Shorter network cables and a custom 24-pin power chord (instead of a groove of Y splitters) could improve the layout.

Fig. 12: Final assembly.

When powered on, as soon the first MB sends out the *power-good* signal, the PSU kicks in and all MBs are up and running, together with network switch.

The network switch is unmanaged and after power-on becomes operational after a delay shorter than the time needed by the MBs to complete the BIOS initialization. In this manner, the PXE procedure finds all the network stack ready for booting.

4. REFERENCES

[1] https://en.wikipedia.org/wiki/High-performance_computing
[2] https://beowulf.org/overview/faq.html
[3] https://en.wikipedia.org/wiki/ATX

5. AKNOWLEDGEMENT

This activity was carried by ENEA within the framework of the Electric System Research Programme (Three-Year Implementation Plan 2025-2027), Project 1.6 "Energy Efficiency of Industrial Products and Processes", Work Package 1 ("Research and solutions for the efficiency improvement and optimization of products, processes, industrial districts, and supply chains."), financed by the Ministry for the Environment and Energy Security. CUP: I53C24003340001.

ENEA
Servizio Promozione e Comunicazione
www.enea.it

Stampa: Laboratorio Tecnografico ENEA - Centro Ricerche Frascati
ottobre 2025