Snellius File Systems

Hands-on data management in high-performance computing

Xavier Álvarez-Farré
High-Performance Computing and Visualization, SURF, Science Park 140, 1098XG Amsterdam, The Netherlands

October 9, 2024

Introduction

Overview of HPC applications

HPC applications are software designed to exploit the power of supercomputers or high-performance clusters to solve complex problems, perform intensive computations, facilitate communications between parallel processes, and manage input/output operations on a high-performance file system.

Overview of HPC applications

HPC applications are software designed to exploit the power of supercomputers or high-performance clusters to solve complex problems, perform intensive computations, facilitate communications between parallel processes, and manage input/output operations on a high-performance file system.

Computations

Extensive numerical operations on a specific region of interest (*e.g.*, mesh elements, particles, or even words in a text document).

- Scale $\mathcal{O}(P)$ for P processors.

Overview of HPC applications

HPC applications are software designed to exploit the power of supercomputers or high-performance clusters to solve complex problems, perform intensive computations, facilitate communications between parallel processes, and manage input/output operations on a high-performance file system.

Computations

Extensive numerical operations on a specific region of interest (*e.g.*, mesh elements, particles, or even words in a text document).

- Scale $\mathcal{O}(P)$ for P processors.

Communications

Coordination and data exchanges between parallel processes to produce meaningful results.

- Scale $\mathcal{O}(\sqrt{P})$ for P processors.

Overview of HPC applications

HPC applications are software designed to exploit the power of supercomputers or high-performance clusters to solve complex problems, perform intensive computations, facilitate communications between parallel processes, and manage input/output operations on a high-performance file system.

Computations

Extensive numerical operations on a specific region of interest (*e.g.*, mesh elements, particles, or even words in a text document).

- Scale $\mathcal{O}(P)$ for P processors.

Communications

Coordination and data exchanges between parallel processes to produce meaningful results.

- Scale $\mathcal{O}(\sqrt{P})$ for P processors.

I/O operations

Interactions with the file system to read input data, write output results, and perform intermediate data storage.

- Usually have no scaling factor.

Overview of I/O operations

Input operations

- **Initialization data** such as geometric grids, genomic sequences, climate simulation data, or large text files.
    - Massive files that require a high level of precision.
    - Involve parallel input operations.

Overview of I/O operations

Input operations

- **Initialization data** such as geometric grids, genomic sequences, climate simulation data, or large text files.
    - Massive files that require a high level of precision.
    - Involve parallel input operations.
- **Job parameters** that provide instructions for guiding the behavior of the job and coordinating processes are typically stored in small files but require coordination between processes.

Overview of I/O operations

Input operations

- **Initialization data** such as geometric grids, genomic sequences, climate simulation data, or large text files.
  - Massive files that require a high level of precision.
  - Involve parallel input operations.
- **Job parameters** that provide instructions for guiding the behavior of the job and coordinating processes are typically stored in small files but require coordination between processes.

Output operations

- **Restart files** or checkpoints enable job continuation in the event of an interruption.
  - Large files that require full precision of simulation data.
  - Involve periodic parallel output operations.

Overview of I/O operations

Input operations

- **Initialization data** such as geometric grids, genomic sequences, climate simulation data, or large text files.
    - Massive files that require a high level of precision.
    - Involve parallel input operations.
- **Job parameters** that provide instructions for guiding the behavior of the job and coordinating processes are typically stored in small files but require coordination between processes.

Output operations

- **Restart files** or checkpoints enable job continuation in the event of an interruption.
    - Large files that require full precision of simulation data.
    - Involve periodic parallel output operations.
- **Analysis data** for visualization, post-processing and in-depth examination.
    - May be possible to use reduced precision or store a subset of the full-resolution data.
    - Involve periodic parallel output operations.

Overview of I/O operations

Input operations

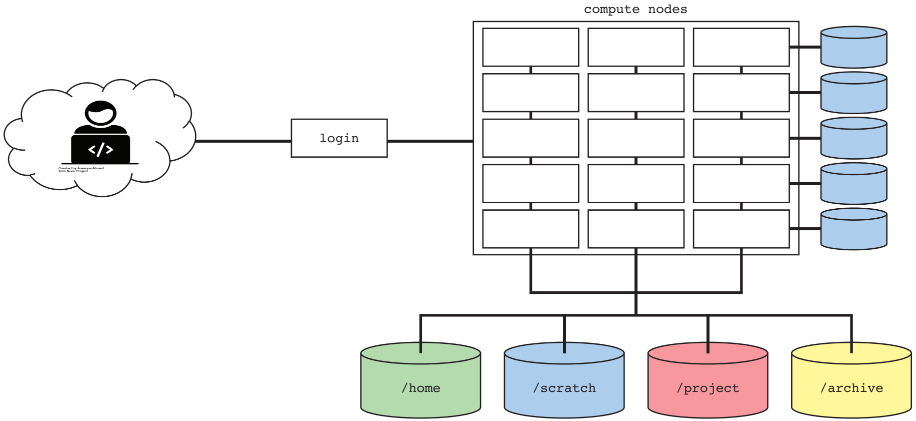
- **Initialization data** such as geometric grids, genomic sequences, climate simulation data, or large text files.
  - Massive files that require a high level of precision.
  - Involve parallel input operations.
- **Job parameters** that provide instructions for guiding the behavior of the job and coordinating processes are typically stored in small files but require coordination between processes.

Output operations

- **Restart files** or checkpoints enable job continuation in the event of an interruption.
  - Large files that require full precision of simulation data.
  - Involve periodic parallel output operations.
- **Analysis data** for visualization, post-processing and in-depth examination.
  - May be possible to use reduced precision or store a subset of the full-resolution data.
  - Involve periodic parallel output operations.
- **Job monitoring** data that provides insights into the application's performance and progress is typically generated by process 0 in a human-readable format, such as ASCII.

Overview of Snellius file-systems

Data management plays a crucial role in HPC applications, encompassing diverse tasks with varying natures and specific requirements. At the same time, HPC systems are equipped with distinct file systems to meet these specific needs.



Overview of Snellius file-systems

Data management plays a crucial role in HPC applications, encompassing diverse tasks with varying natures and specific requirements. At the same time, HPC systems are equipped with distinct file systems to meet these specific needs.

File-system	Speed	Size	Backed-up	Expiration
`/home`	Normal	200 GB	Yes	Permanent
`/scratch-shared`	Fast		No	6/14 days
`/scratch-local`	Fast	8 TB	No	6/14 days
`/scratch-node`	Fastest		No	End of job
`/project`	Fast	On demand	No	Permanent
`/archive`	Slow	On demand	Yes	Permanent

Hands-on

Objectives of the hands-on session

1. **Navigating file systems**:
   - Understand the structure of HPC file systems (home, project, scratch, archive)
   - Move between directories and check the current location
2. **File management**:
   - Create, view, and edit files using basic terminal commands
   - Move, rename, and copy files between different file systems
3. **Directory management**:
   - Create and delete directories
   - Navigate directories and manage file hierarchy
4. **File permissions**:
   - Check and modify file access permissions
5. **Disk usage monitoring**:
   - Monitor storage usage and available space across file systems

1. Navigating file systems

Objective: Learn how to move between directories in different file systems (home, project, scratch, archive) and understand the structure of the HPC environment.

Key commands:

- `pwd` - Display the current directory
- `ls` - List the contents of a directory
- `cd <path>` - Change directories

Tasks:

- Start in the `home` directory and use `pwd` to check your location.
- Use `cd` to navigate to `scratch` directory.
- Explore the contents of `scratch` directory with `ls`.
- Return to the `home` directory using `cd`.

Expected output:

- Comfortable navigating and exploring directories in the HPC file system.

2. File management

Objective: Learn how to create, view, move, and rename files across different file systems in an HPC environment.

Key commands:

- `touch <fname>` - Create a new empty file
- `nano <fname>` or `vim <filename>` - Edit a file
- `cat <fname>` - Display the contents of a file
- `cp <src> <tgt>` - Copy files between directories
- `mv <src> <tgt>` - Move or rename files

Tasks:

- Create a file in the `home` directory using `touch`.
- Edit the file with `nano` or `vi`, and view its contents with `cat`.
- Copy the file to the `scratch` directory using `cp`.
- Rename or move the file using `mv`.

Expected output:

- Understand how to manage files: creation, editing, copying, and renaming.

3. Directory management

Objective: Learn how to create, navigate, and remove directories within the HPC file system.

Key commands:

- `mkdir <dname>` - create a new directory
- `cd <dname>` - navigate into a directory
- `rmdir <dname>` - remove an empty directory
- `ls [dname]` - list directory contents

Tasks:

- Create a new directory in the home directory using `mkdir`.
- Navigate into the new directory with `cd`.
- List its contents with `ls`, then return to the home directory.
- Remove the directory using `rmdir`.

Expected output:

- Understand how to create, enter, and remove directories and manage directory structure effectively.

File permissions

Objective: Learn how to view and modify file access permissions to control who can read, write, or execute files.

Key commands:

- `ls -l` - view file permissions
- `chmod` - change file permissions

Explanation:

- Permissions are shown as `rwxrwxrwx` (for owner, group, others).
- `r` - read (4), `w` - write (2), `x` - execute (1).
- Numeric format: `chmod 755` means owner can `rwx`, group and others can `r-x`.

Tasks:

- Use `ls -l` to check the permissions of a file.
- Modify the file's permissions to allow read access for everyone using `chmod 644`.
- Verify the permission changes with `ls -l`.

Expected output:

- Understand how to check and modify file permissions in the HPC environment.

Disk usage monitoring

Objective: Learn how to monitor storage usage and check available space across different file systems.

Key commands:

- **`df -h`** - view disk space usage for all mounted file systems
- **`du -sh`** - check the size of a specific directory

Tasks:

- Use **`df -h`** to view overall disk usage and available space in the home, scratch, and project file systems.
- Check the size of a specific directory in your home using **`du -sh`**.
- Identify where you are using the most space.

Expected output:

- Understand how to monitor storage usage and manage disk space efficiently.