RIT

RIT Research Computing: Theory and Applications

COGS 621

2025 - 09 - 16

Prepared by: Viet Nguyen

Objectives

- Code basic bash programs and use Linux naturally
- Use any scheduling system that has SLURM-backend HPC cluster
- Code bash and python scripts to scale job submissions
- Use RIT’s research computing interactive platform

Agenda

1. (10 mins) Overview of Linux, Bash, and SSH
   - Linux and bash
   - SSH

2. (15 mins) Distributed Computing Overview
   - Concurrency and parallelism
   - Distributed systems

3. (25 mins) RIT Research Computing Theory
   - The grant
   - Filling forms and accessibility
   - RIT's Scheduled Processing On Research Computing (SPORC)
   - SLURM tutorial
   - Demo multi-trial job submission script
   - Demo sharded data parallel job submission script

4. (10 mins) Tips and Tricks
   - RIT RC's ondemand and related tools
   - How queuing system works
   - Conda – the powerful python package management system
   - File transfer using FileZilla and SFTP

Part 1 – Overview of Linux, Bash, and SSH

Overview of Linux, Bash, and SSH

Linux and Bash

- For windows, you might want to test out Linux with Windows Subsystem for Linux (WSL)
- **[Demo]** Linux file system
  - Use forward slash to separate folder instead of backward slash like in Windows
- **[Demo]** Instead of using the mouse to click, find, and open them, you type commands into the **terminal** to tell the computer what to do:
  - clear: clear the terminal
  - pwd: print work directory
  - ls: list files in current work directory
  - cd: change directory
  - rm: remove
  - ~: home directory
  - /: root directory
  - ./: current directory (in relative path)
  - ../: parent directory (in relative path)
  - Absolute path

Overview of Linux, Bash, and SSH

Linux and Bash

[Demo] Some other frequently used commands:

- cat <file>: view the file
- mkdir -p <folder_name>: create folder
- rmdir <folder>: remove folder (has to be empty)
- head -n 3 <file>: view the first three lines of the file
- tail -n 3 <file>: view the last three lines of the file
- touch <file>: create file
- cp <source> <destination>: copy a file from source to destination. -r: recursive
- mv <source> <destination>: move a file from source to destination. -r: recursive
- rm <file>: remove a file. rm -rf <file>: force and recursive. **DON'T EVER DO rm -rf /**
- grep <pattern> <file>: searching the file for a pattern. E.g., grep "*2022-08-19*" dataset1_results.csv. Pattern uses regular expression
- man <command>: print the command manual
- du -h . --max-depth=1: list all items and size of the current folder, no recursive

Overview of Linux, Bash, and SSH

Linux and Bash

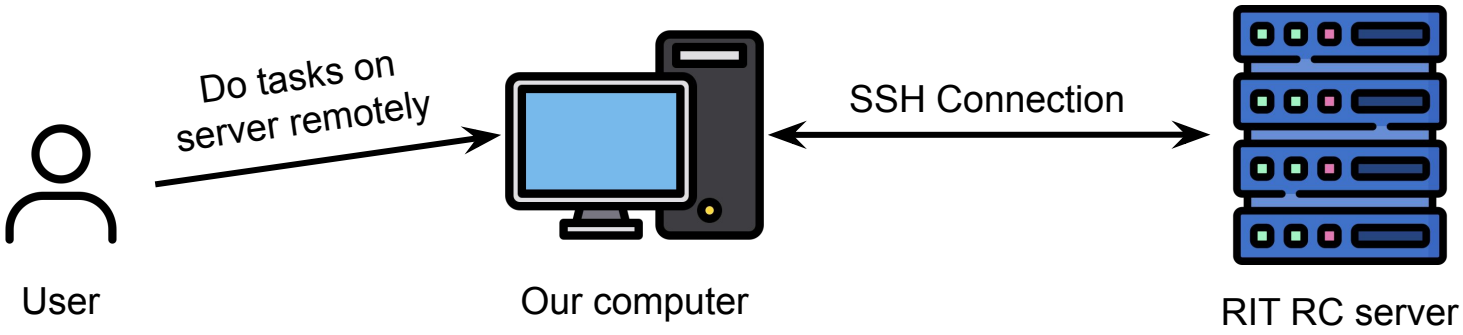
- **[Demo]** Using VSCode for convenience
  - Download and setup: https://code.visualstudio.com/download

- **[Demo]** Writing a bash script
  - Variables and comments
  - Loops and conditionals
  - Operators

Overview of Linux, Bash, and SSH

SSH

- SSH (Secure Shell Scripting):
  - Securely access remote servers using a public key and private key
  - Public key (like a lock): everyone can see
  - Private key (like the key in the holder's hand): only a holder having the key can unlock
  - First time connecting, the keys are automatically generated
  - Allows us to access the RIT Research Computing cluster server
- Connecting to the server
  - **[Demo]** VSCode (sporcsubmit.rc.rit.edu)
  - Any other SSH tool: MobaXterm, OpenSSH, etc.



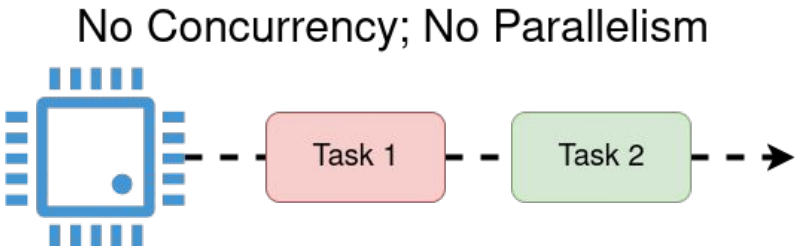
Part 2 – Distributed Computing Overview

Distributed Computing Overview

Concurrency and Parallelism

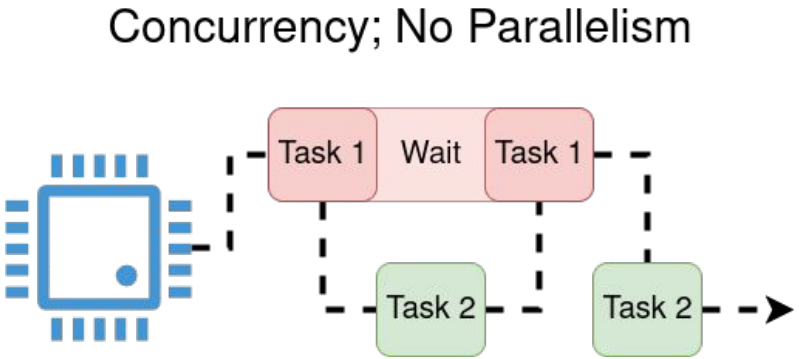
[Demo] No Concurrency; No Parallelism

- Task 1 -> task 2 -> …
- Pros: simple
- Cons: slow, monotone, not parallelizable



[Demo] Concurrency; No Parallelism

- In waiting for a task, do another task, then come back
- Pros: faster than no concurrency
- Cons: some tasks is too long, blocking the execution of other tasks

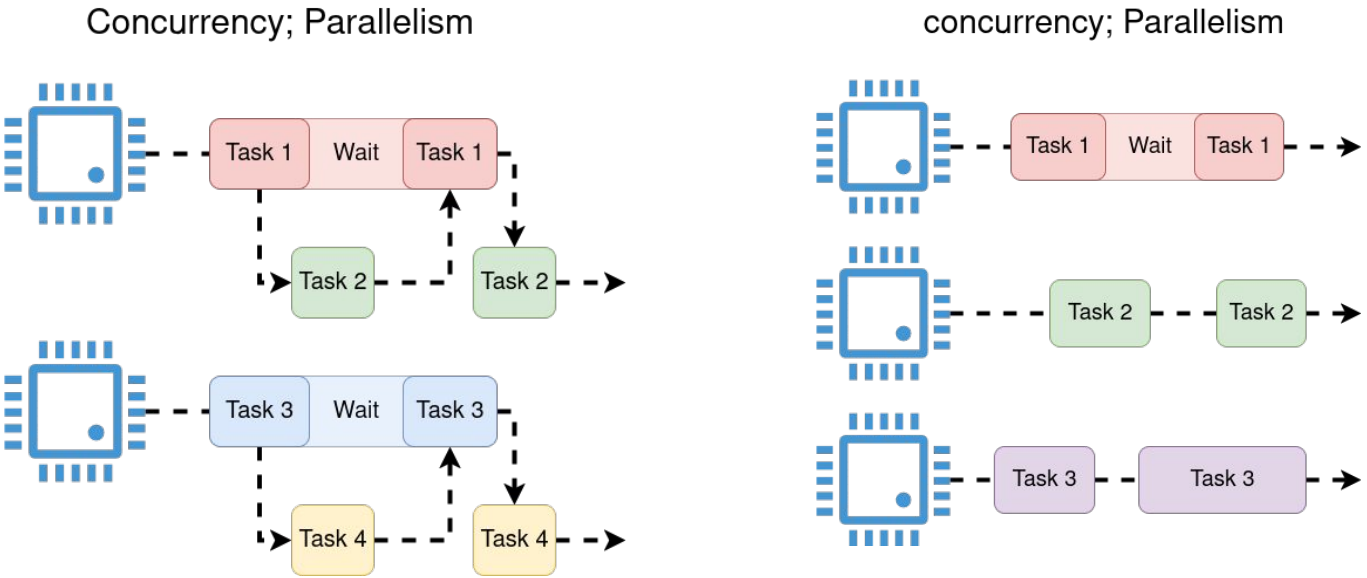


Distributed Computing Overview

Concurrency and Parallelism

[Demo] Concurrency; Parallelism

- Multiple CPU/GPU. Each handle a particular task
- Pros: Parallelize tasks that can be done separately

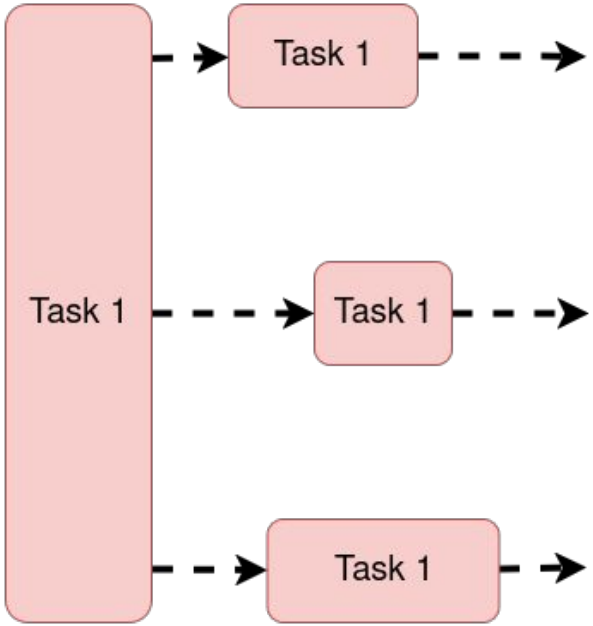


Distributed Computing Overview

Sharding

[Demo] *Sharding*

- The process of breaking a task down to multiple sub-tasks and aggregate them when computations are done
- Technique for **splitting** large datasets or workloads into smaller, more manageable **pieces (“shards”)** that can be **distributed across multiple machines or processes**. Each shard holds only a subset of the total data, and together, all shards represent the complete dataset.

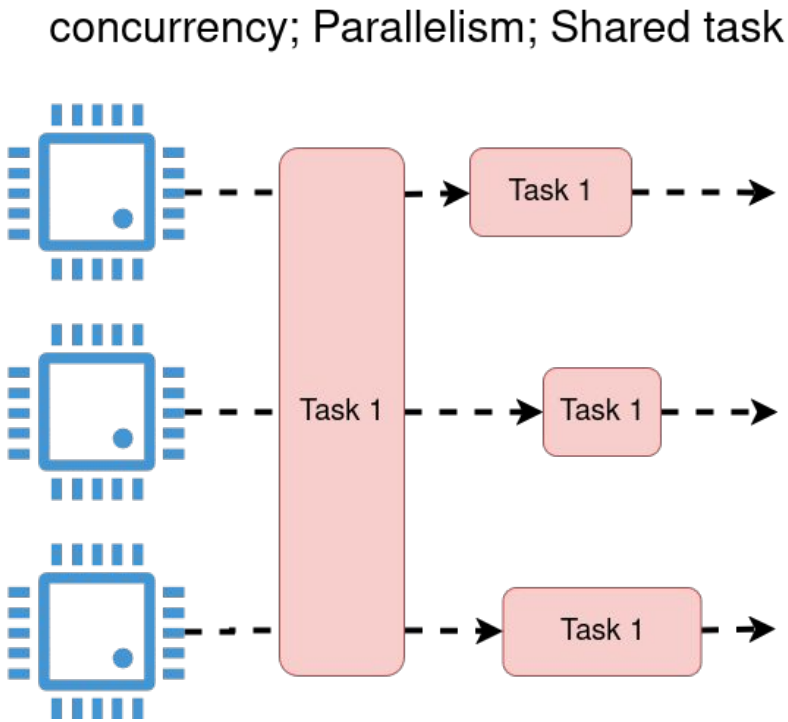


Distributed Computing Overview

Concurrency and Parallelism

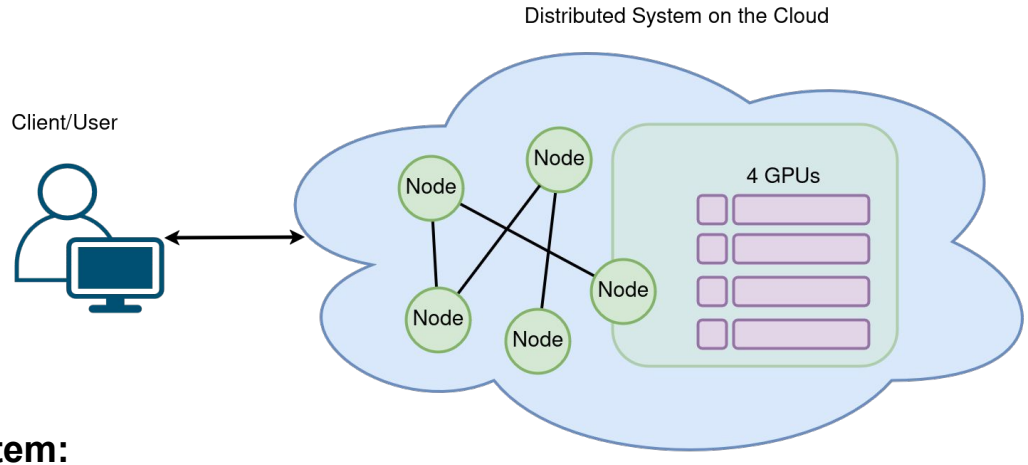
Concurrency; Parallelism; Shared Task(s)

- Each task is splitted into different subtasks
- Each subtask is solved using a dedicated device
- Results are then aggregated from all devices
- Concurrency can be performed within each device
- Pros: has concurrency, has parallelism, allow multiple devices to solve multiple tasks while minimizing idle time



Distributed Computing Overview

Distributed System



Distributed system:

- Definition: a collection of independent hardware and software called nodes (connected together in a network system).
  - No shared clock (each node has independent processors)
  - No shared memory (each node has independent memory)
  - Processes are autonomous and executed concurrently
- Purpose: coordinate and communicate to solve a particular problem/task.
- RIT's SPORC: each node will have from 0-4 GPUs. Most of them has 4 GPUs.