Introduction to Parallel R

October 20, 2026

Presented By: Jose Hernandez

Agenda

I. Intro to Parallel Computing
II. R & RStudio overview
III. Using the parallel library on RStudio
IV. Intro to submit scripts

Parallel Computing

Introduction to Parallel Computing

- Parallel Computing can be thought of like a class project. Suppose you are a teacher and you have a large project that you want to assign to the students.
  - You divide up the students into N groups.
  - Give each group a small part of the task.
  - Have them turn in results at the end and then you combine them to a final product.

Key Terminology in Parallel Computing

Core

- An individual CPU core on a given computer (most CPUs have multiple cores).

Node

- An individual computer. Often a Node will have many Cores.

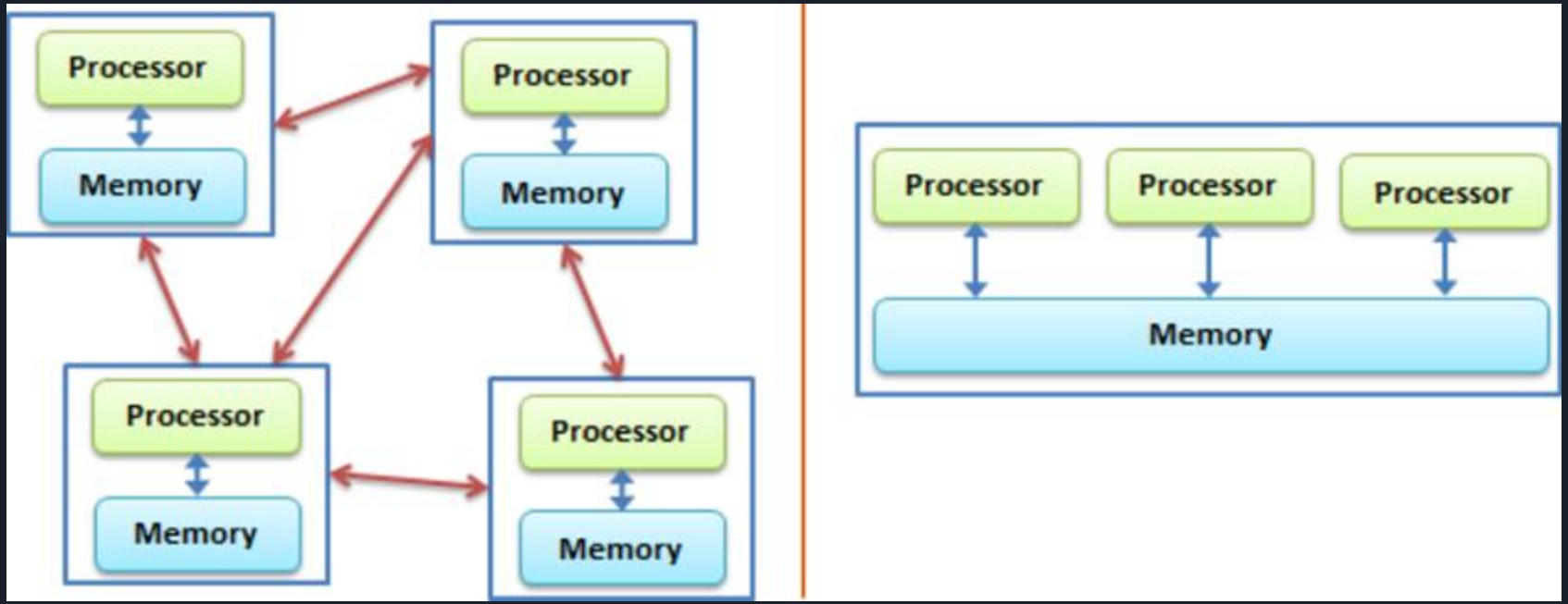








Introduction to Parallel Computing



The 'parallel' Package in R

Thankfully, once again R makes life a bit easier for us. R has a package called 'parallel' which gives us a really high-level framework to do parallel computations.



Parallel use in RStudio

Parallel use in Batch scripts

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```

Job Name

- *The job name parameter is simply an identifier for your job. This can be helpful for keeping track of what each job is doing.*

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```



Queue/SLURM Account

- *If this is not specified, it will default to the* ***genacc_q*** *SLURM Account.*

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```

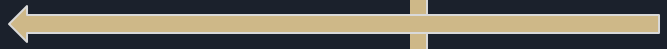
Number of Hours

- *Time requests are more flexible in batch jobs. SLURM's time allocation format is:*
- ***Days-Hours:Minutes:Seconds***
- *Example: 14-00:00:00 = 14 Days*
- *Example: 34:25:16 = 34 Hours, 25 Minutes and 16 Seconds.*
- *If this is not specified, it will default to the listed* ***Default Wall Time*** *listed in the documentation.*

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```



Number of Cores

- *This is the number of physical CPU processors you want to use.*
- *Most of the benefits of HPC come from parallel processing.*
- *Check your software's documentation to see if it supports this!*

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```

Number of Nodes

- *This is the number of full computers (think desktop/workstation towers for reference) that you want to use.*
- *This is for programs that support Distributed Computing.*
- *If this parameter is left out the job scheduler will decide this on its own.*

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```



Environment Modules

- *Open OnDemand Apps will often do this for you. These load software and applications into your terminal environment for you.*

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```



Command to Run your Program

- *In non-interactive jobs, you have to tell the computer what you want it to do in your job ahead of time.*

Command-Line Batch Jobs

```
#!/bin/bash
#SBATCH -J "MyJob"
#SBATCH -A genacc_q
#SBATCH -t 1:00:00
#SBATCH -n 4
#SBATCH -N 2


module load R
R CMD BATCH file.R
```

Translating the Job Script

- *Run it in the genacc_q SLURM Account*
- *Give me at most 1 hour to run my job*
- *I also need 4 cores for this.*
- *I also need all 2 cores to be localized onto 2 physical computer node (don't spread the cores over multiple nodes unless you can support distributed computing)*
- *Please run my code called file.R*

Batch submission

Exercise: Multi-Node R submission

Using Files already on your system, run these commands

```
vim multinode-R.sh
sbatch multinode-R.sh
```

This will return a job ID number (number will change each job)

```
Submitted batch job 123456
```

When the job is complete, three new files fill be present called:

- ***Slurm-123456.out*** *(again, the number will change from job to job)*
- ***multinode-R.Rout*** *(name will bne identical to the script submitted)*
- ***regression-results.rds*** *(results in a format that R can read later)*

Thank You!

Questions?