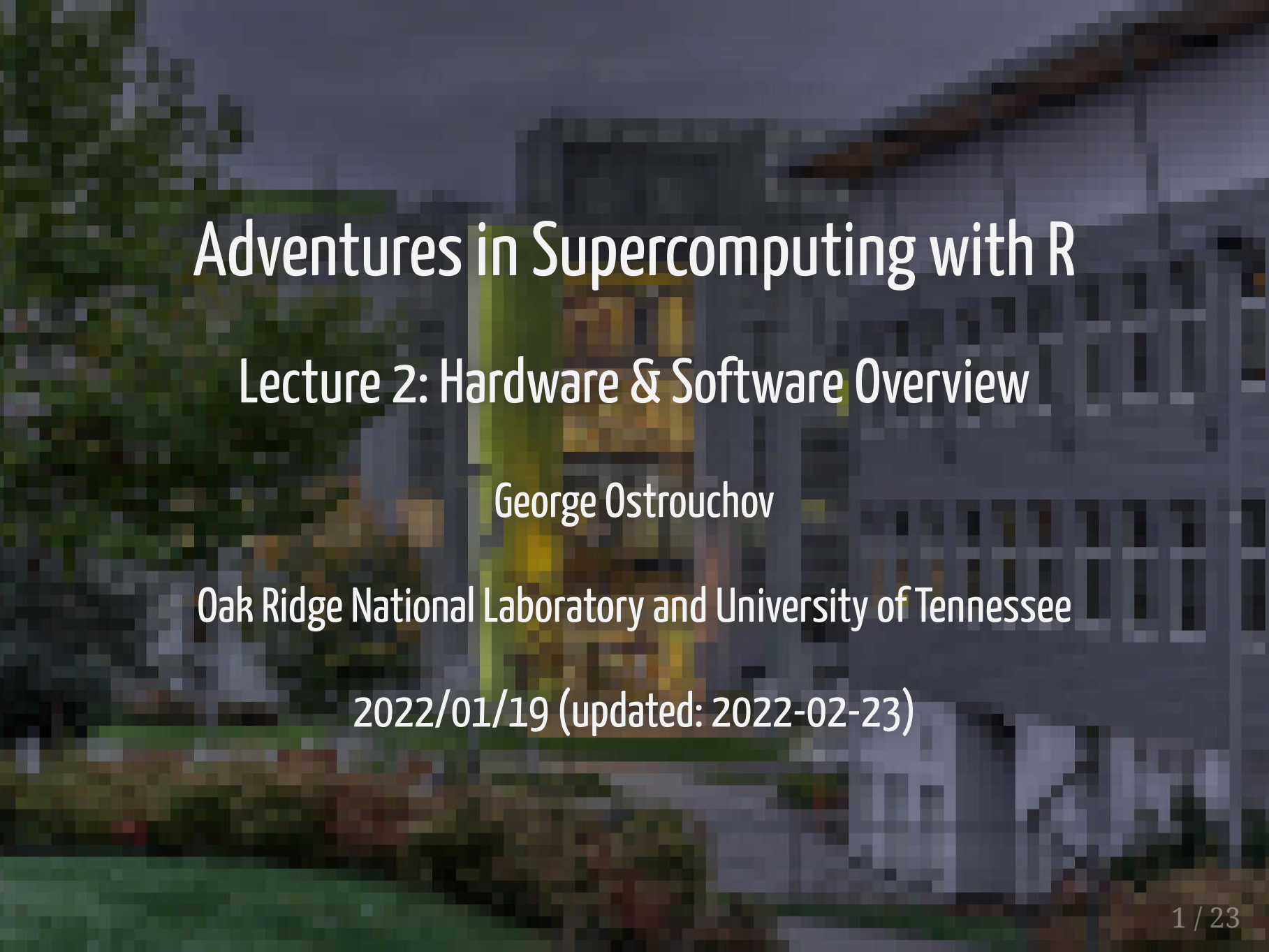


Adventures in Supercomputing with R

Lecture 2: Hardware & Software Overview

George Ostrouchov

Oak Ridge National Laboratory and University of Tennessee

2022/01/19 (updated: 2022-02-23)

Summary of Last Lecure

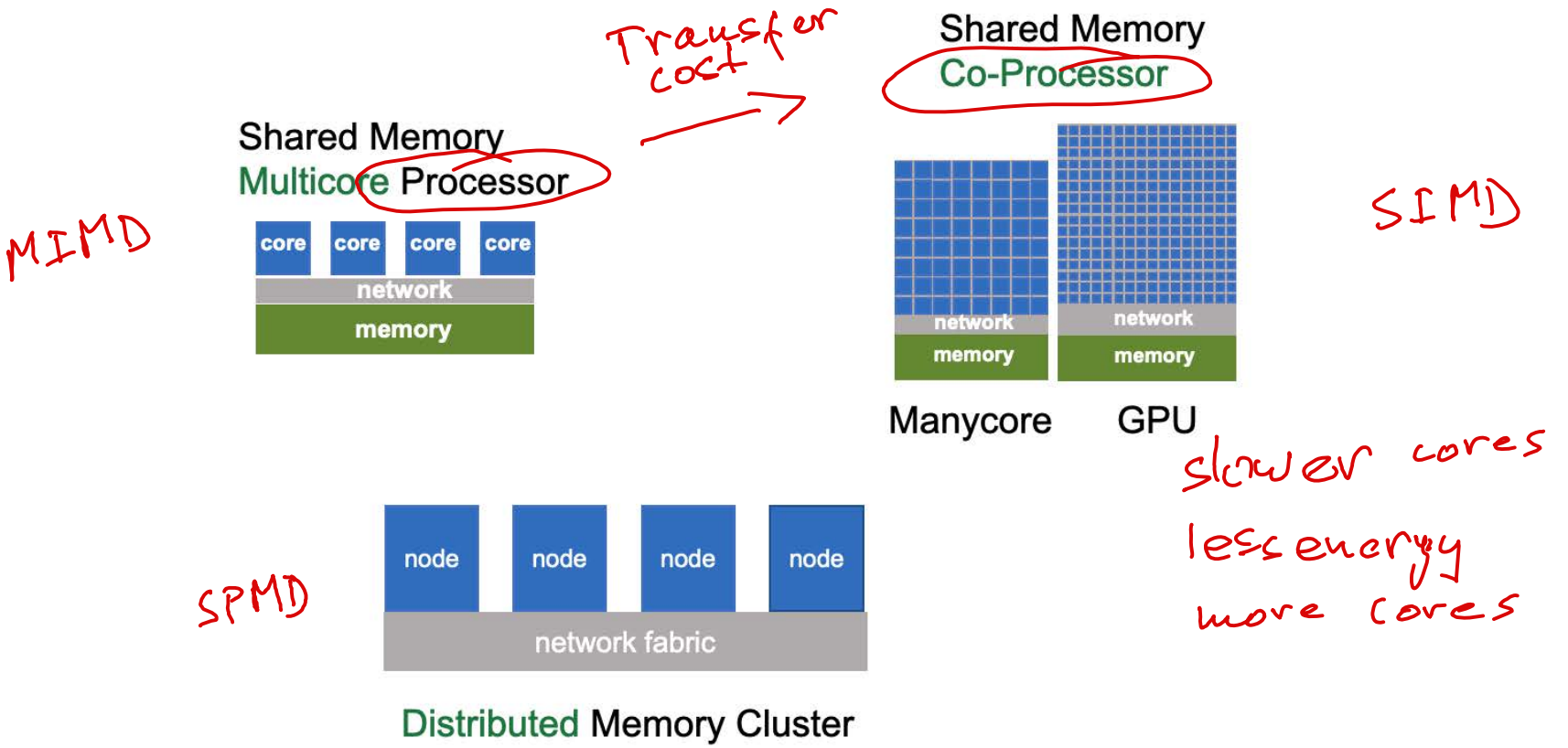
- Two parallel computing communities
  - Parallel numerical math (generating data) - supercomputing
  - Parallel databases (storing and serving data) - data centers
- Statistical computing left batch environment to be interactive
  - Developed S and then R
  - High-level, extensible, and interactive
  - Use numerical libraries
- Supercomputing takes R back to batch
  - High-level, extensible, but batch
  - Use scalable numerical libraries
- Workflow, accounts, tools
  - RStudio, git, GitHub, ssh, unix

PuTTY
WinSCP

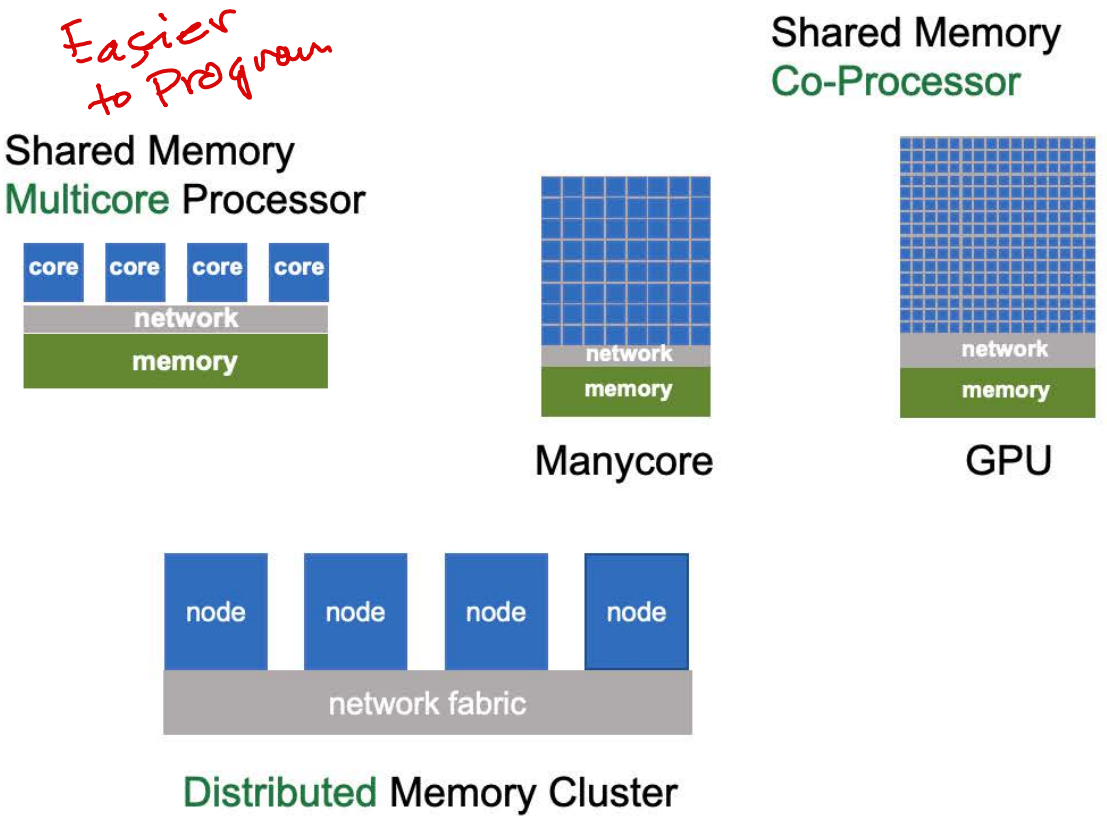
Parallel Hardware

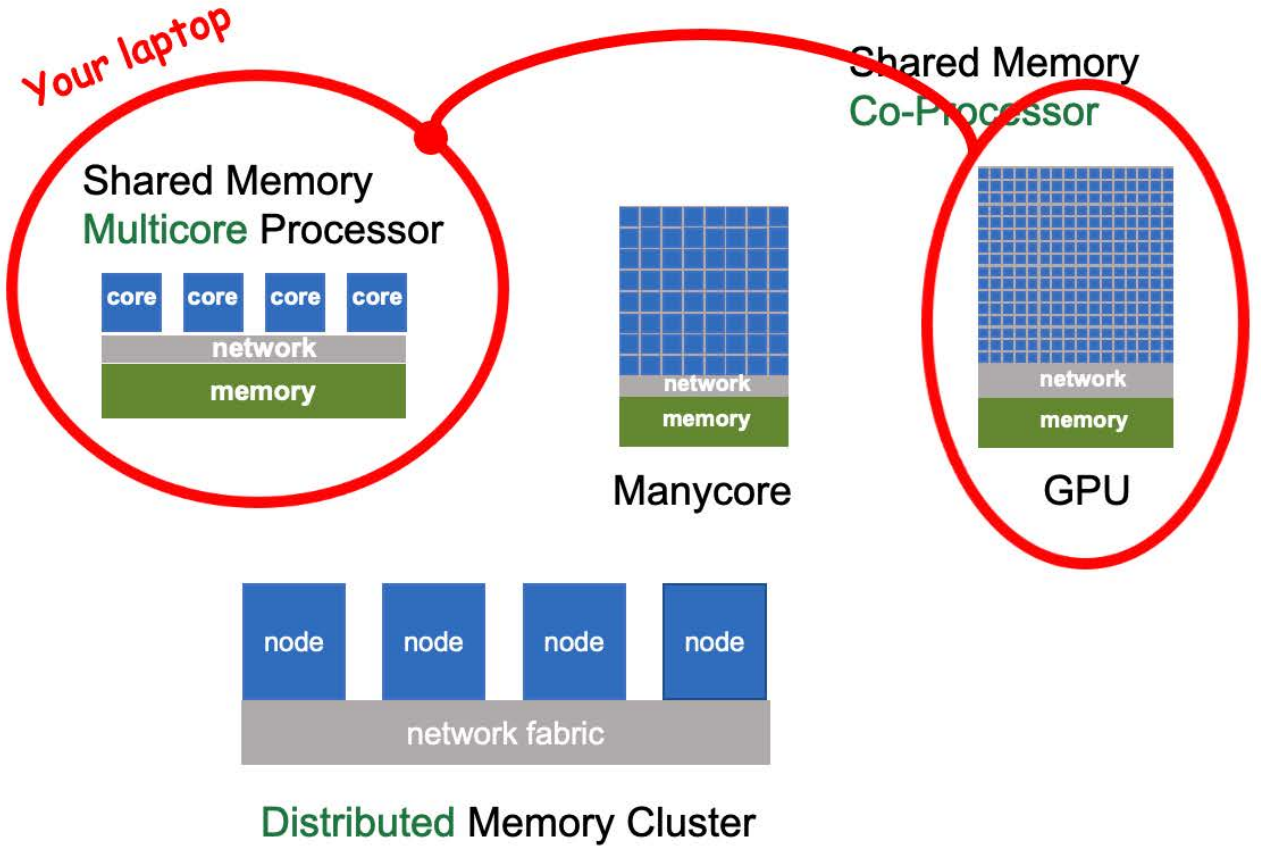
A high level look at what matters ...

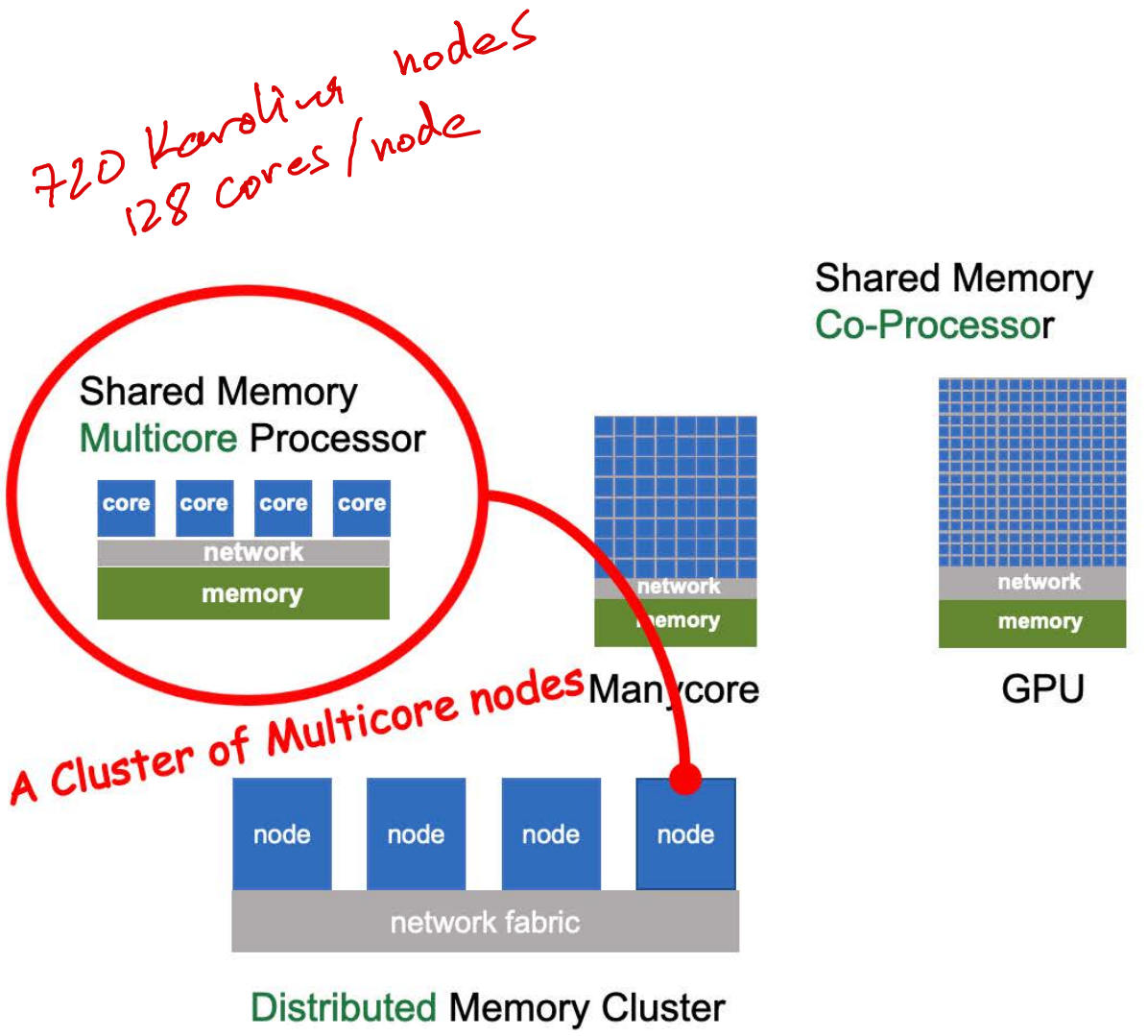
Three Basic Concepts in Hardware

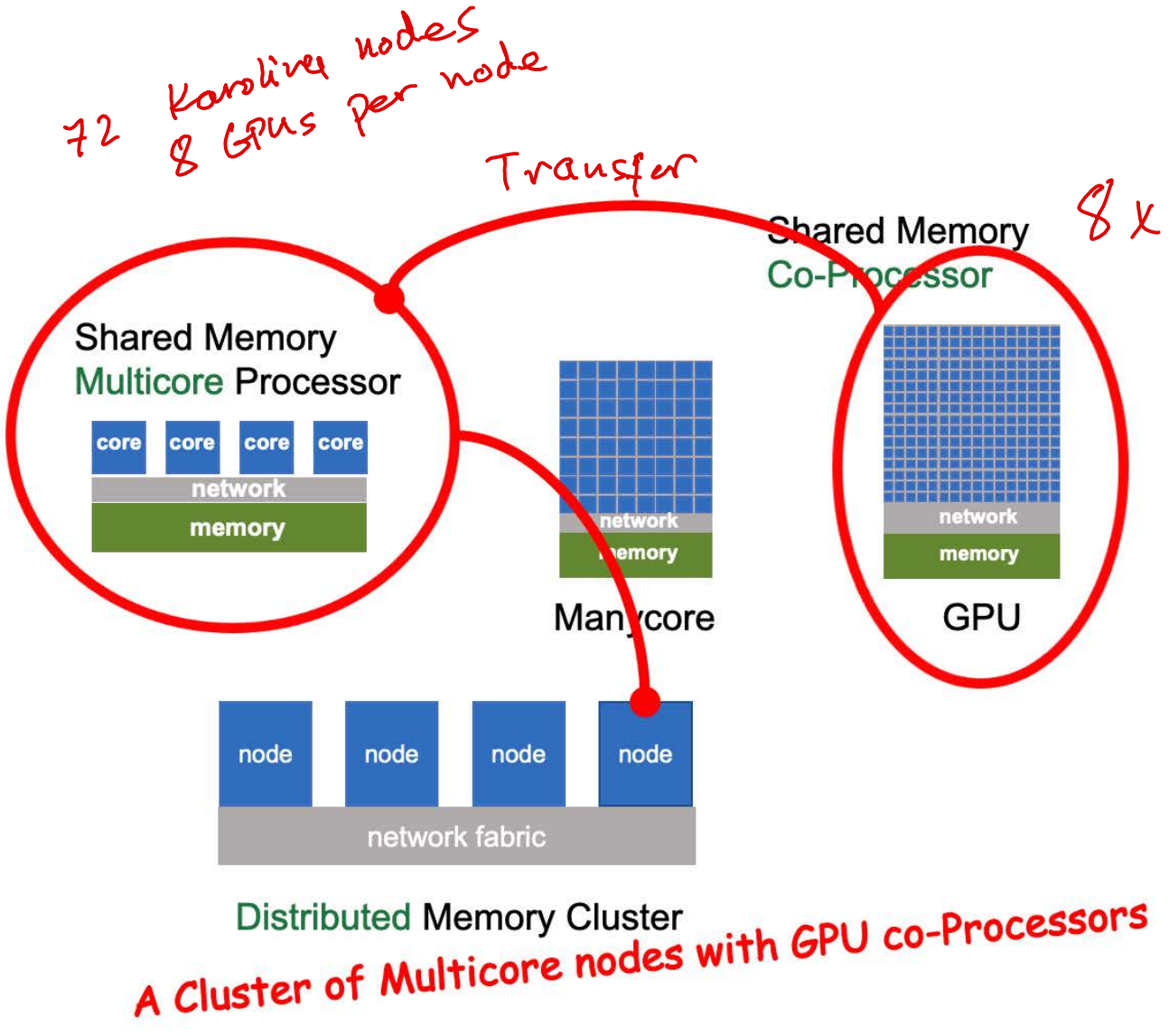


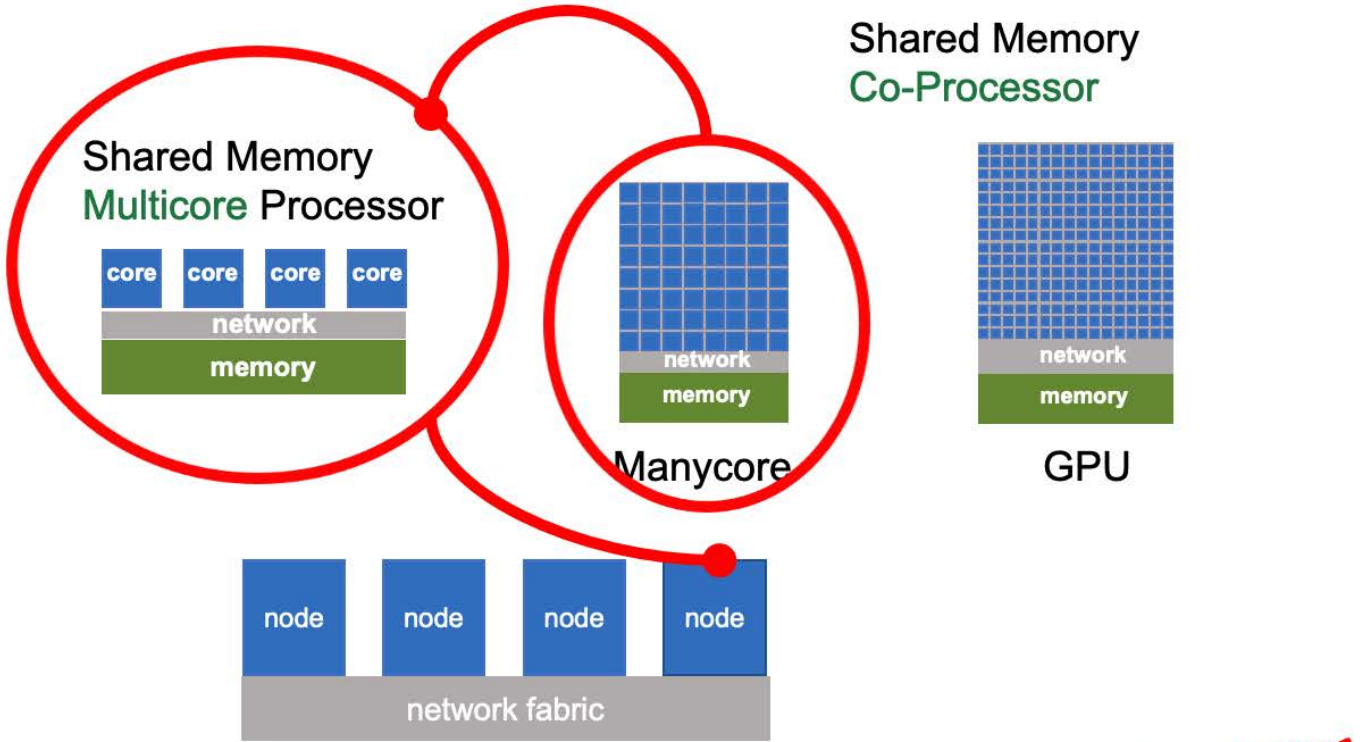
Three Basic Concepts in Hardware




Your laptop
Shared Memory
Multicore Processor
core
core
core
core
network
memory
Shared Memory
Co-Processor
network
memory
GPU
network
memory
Manycore
node
node
node
node
network fabric
Distributed Memory Cluster


720 Karolina nodes
128 cores / node
Shared Memory
Co-Processor
Shared Memory
Multicore Processor
core
core
core
core
network
memory
network
memory
Manycore
network
memory
GPU
A Cluster of Multicore nodes
node
node
node
node
network fabric
Distributed Memory Cluster


72 Kepler nodes
8 GPUs per node
Transfer
8 x
Shared Memory
Co-Processor
Shared Memory
Multicore Processor
core
core
core
core
network
memory
network
memory
Manycore
network
memory
GPU
node
node
node
node
network fabric
Distributed Memory Cluster
A Cluster of Multicore nodes with GPU co-Processors

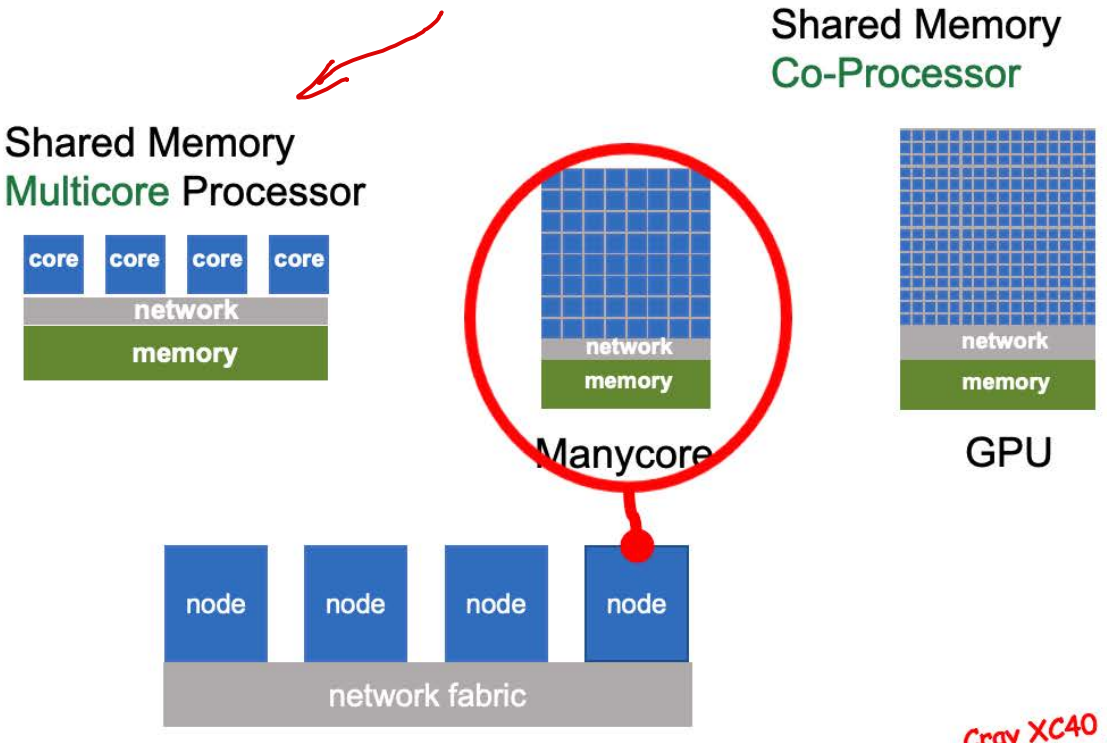


A Cluster of Multicore nodes with Manycore co-Processors

Best of both worlds. Future?

Can act like

Programmable like

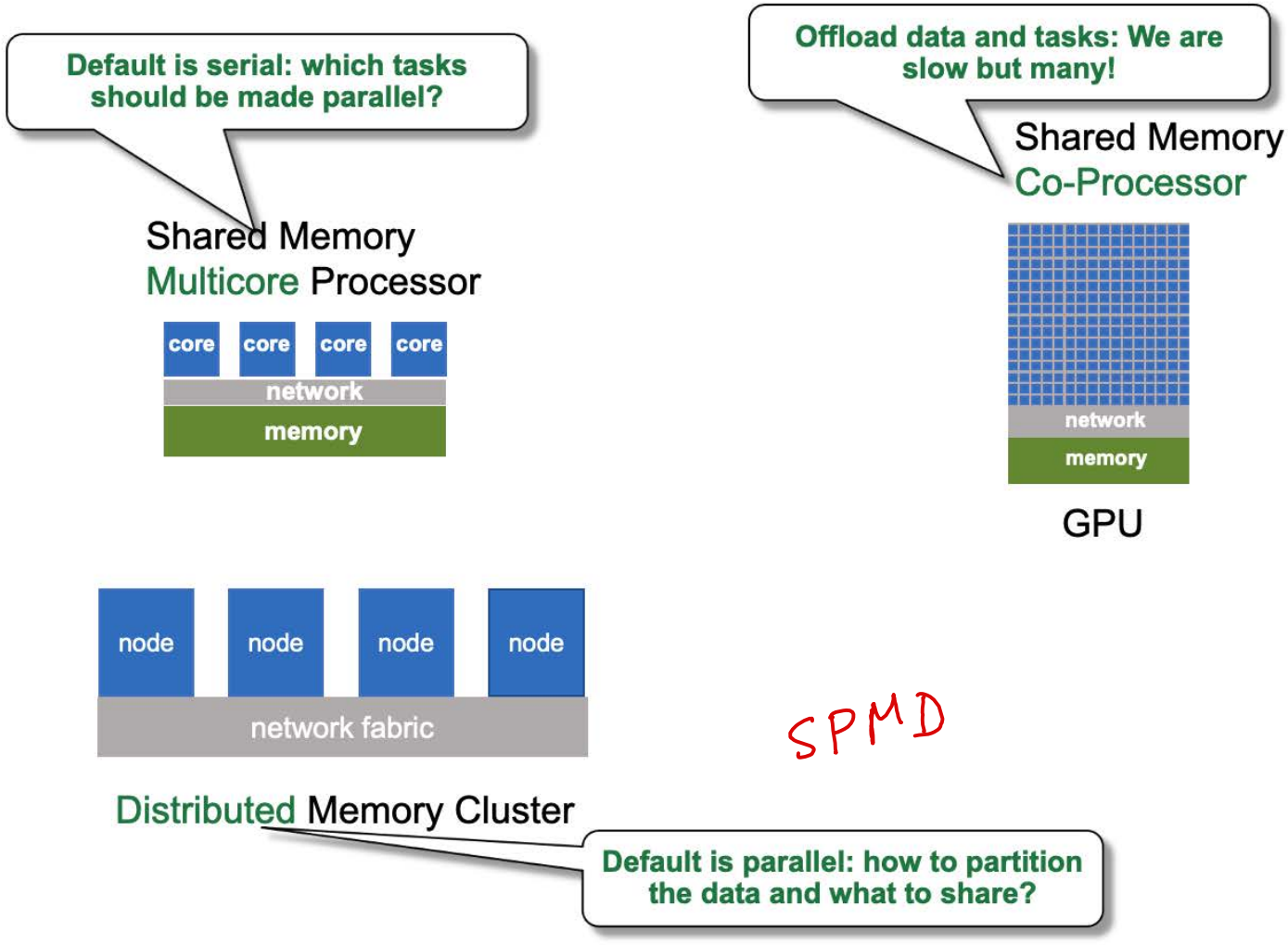


A Cluster of Manycore Nodes

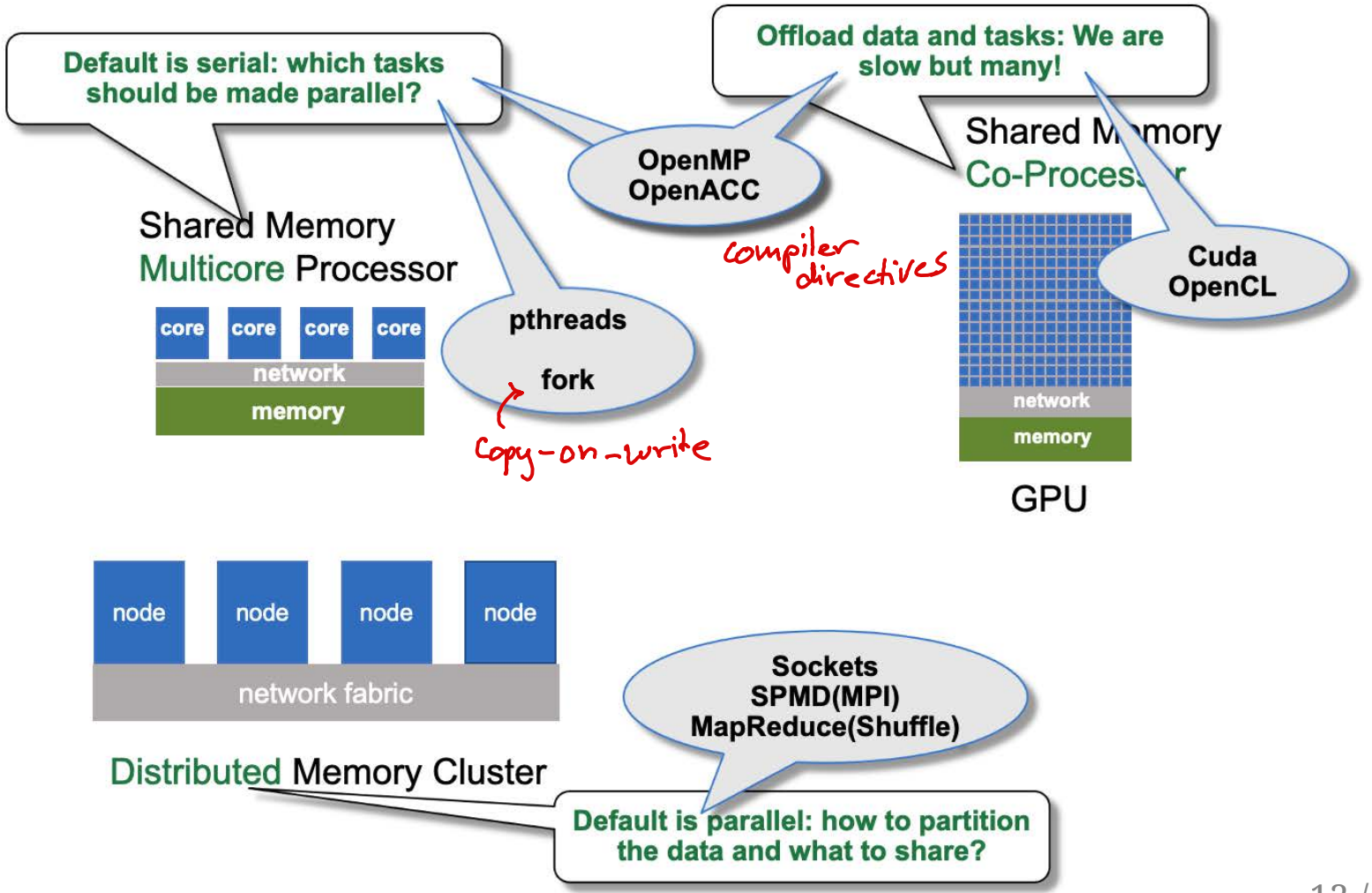
Cray XC40
2nd generation Manycore
Intel Xeon Phi "KNL"

Fugaku ARM #1

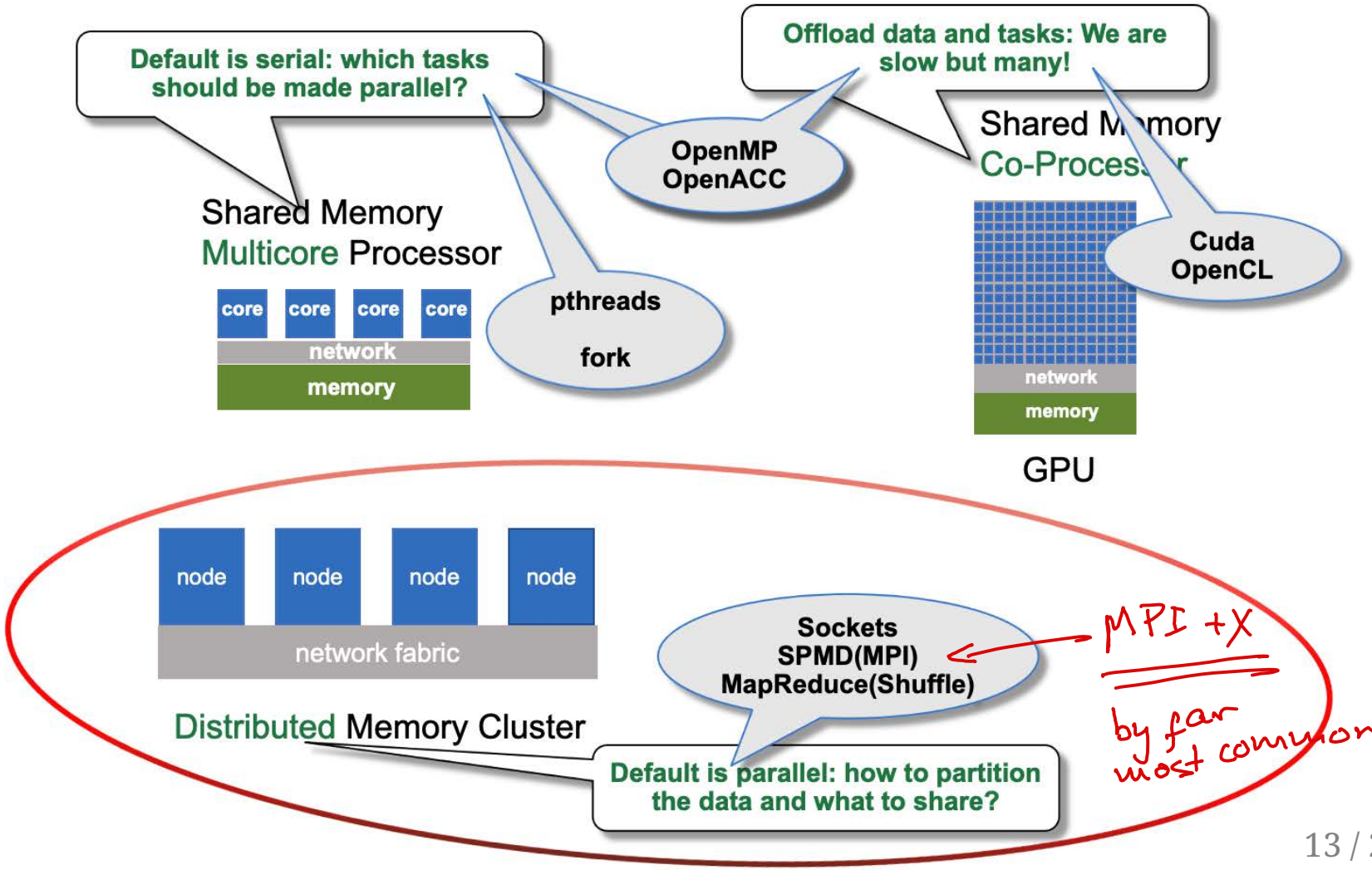
Native Programming Mindset



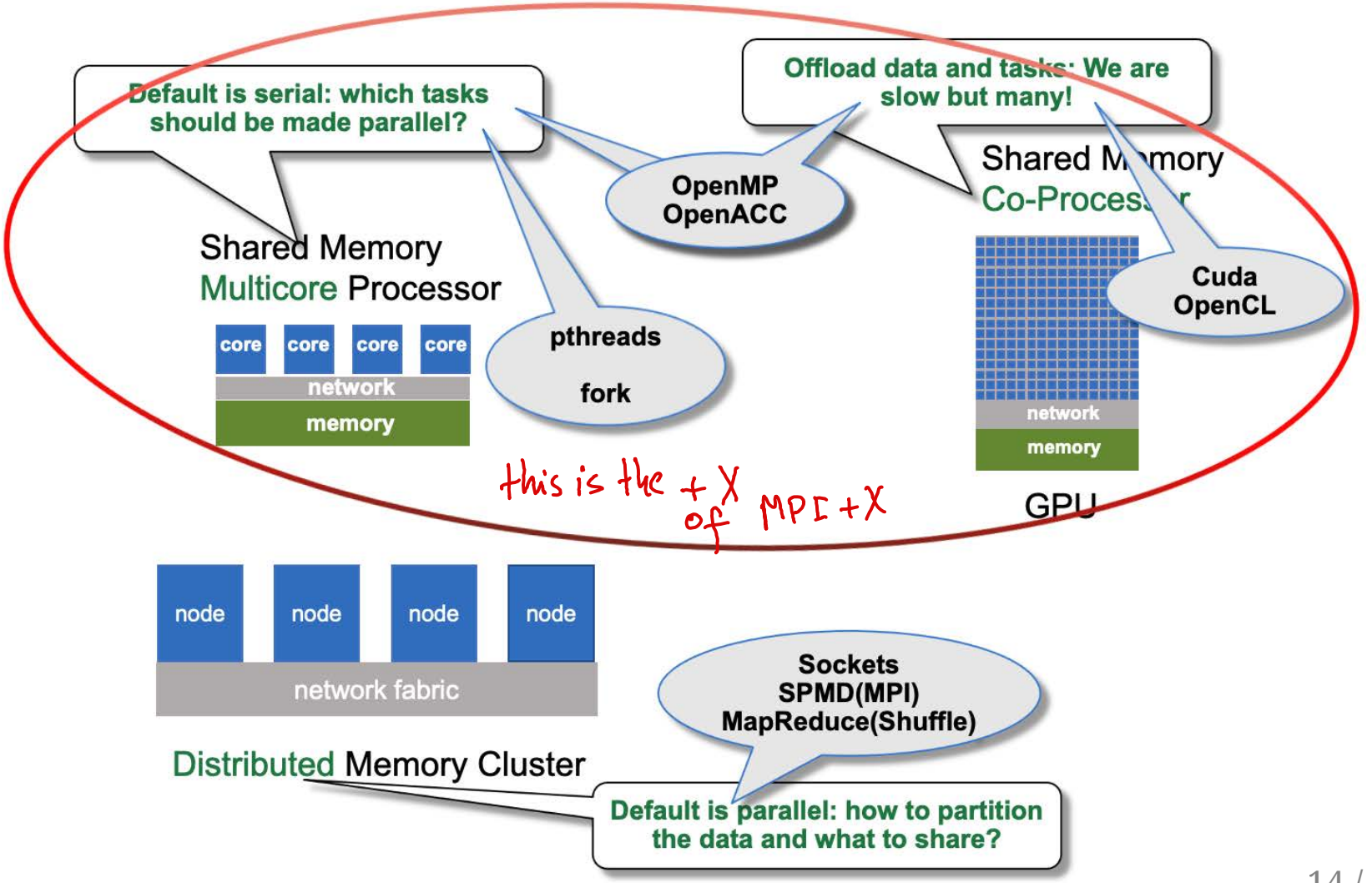
Native Programming Models and Tools



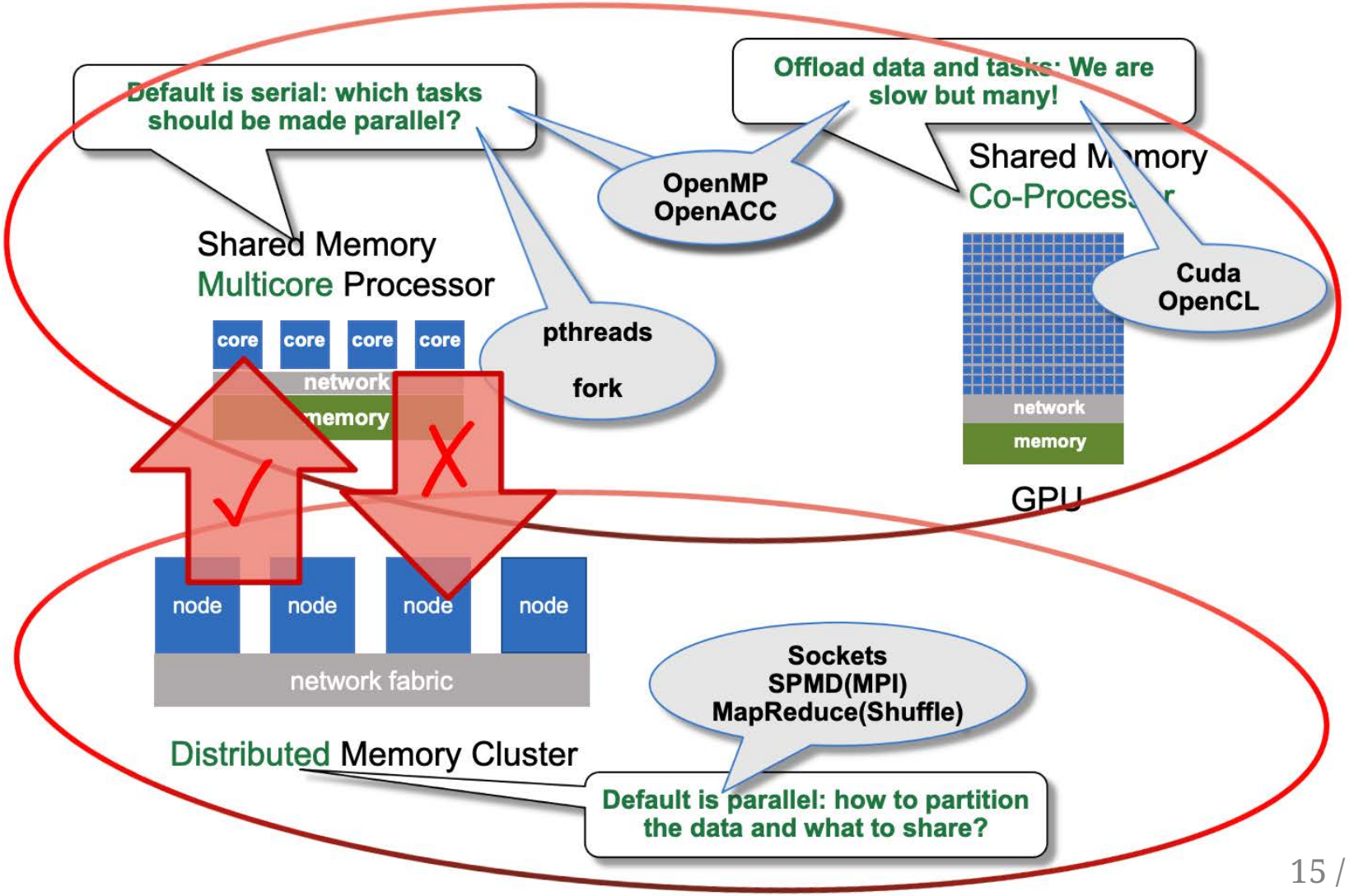
35+ Years of Parallel Computing Research



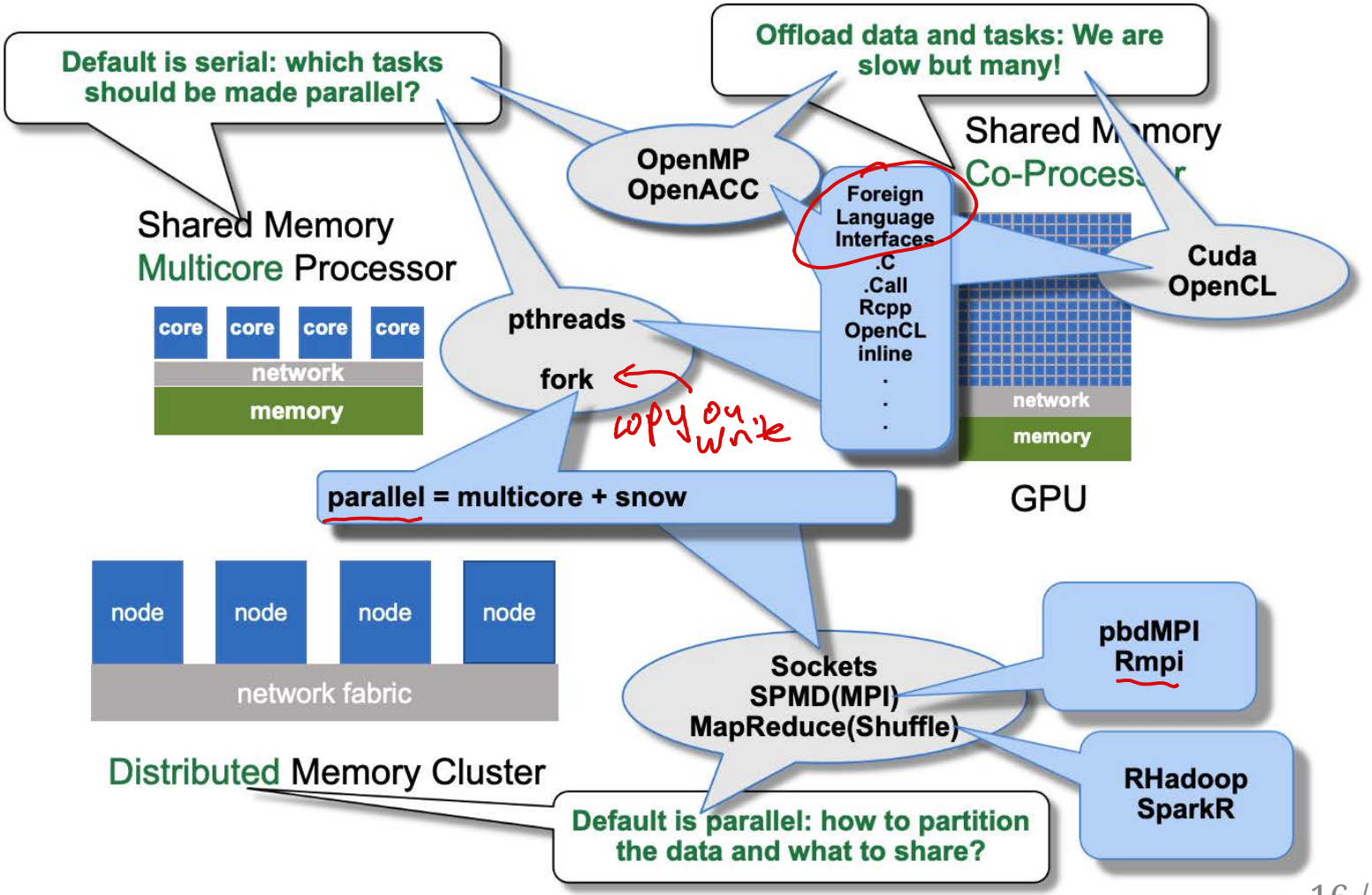
Last 15+ years of Advances



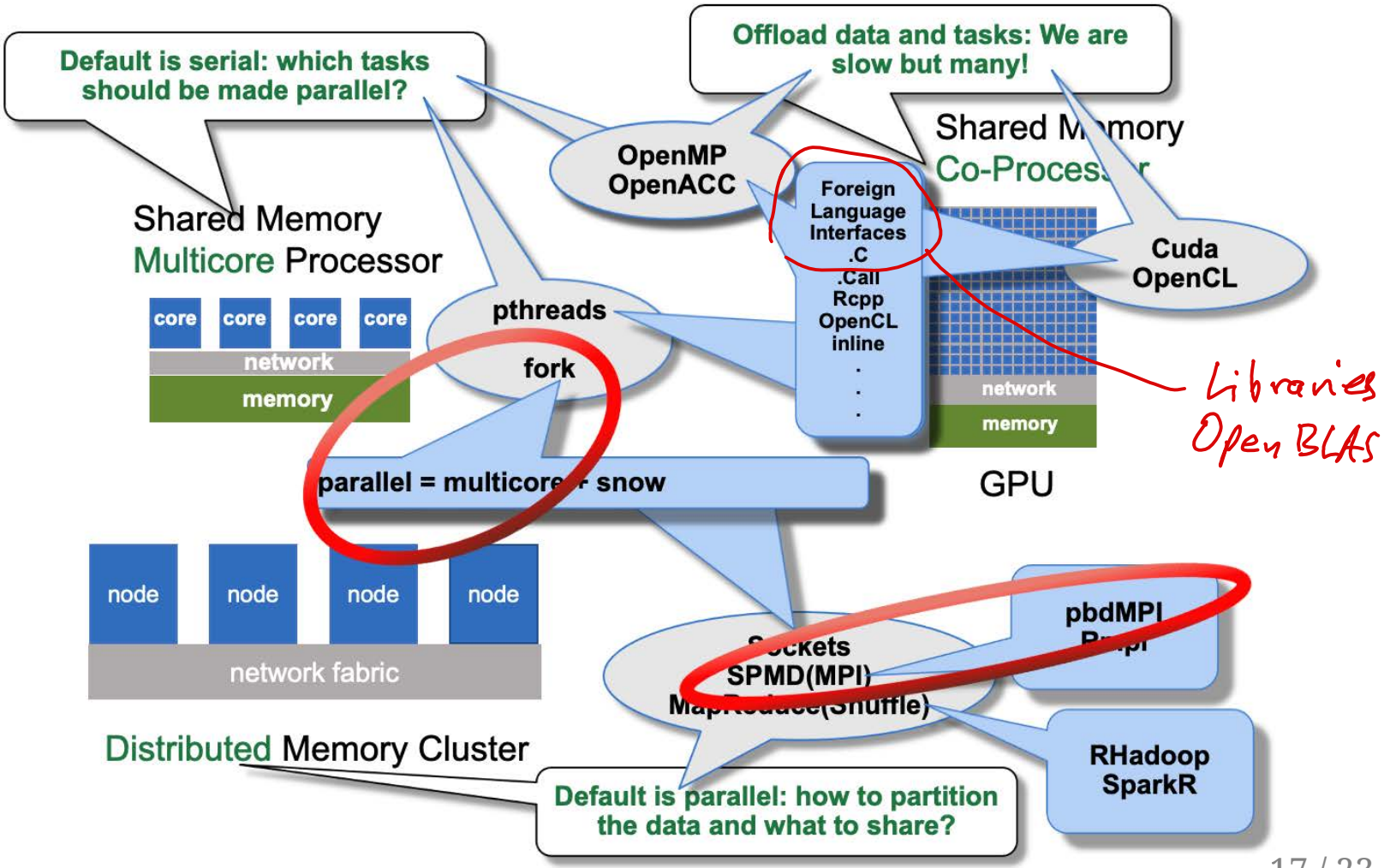
Distributed Programming Works in Shared Memory



R Interfaces to Low-Level Native Tools

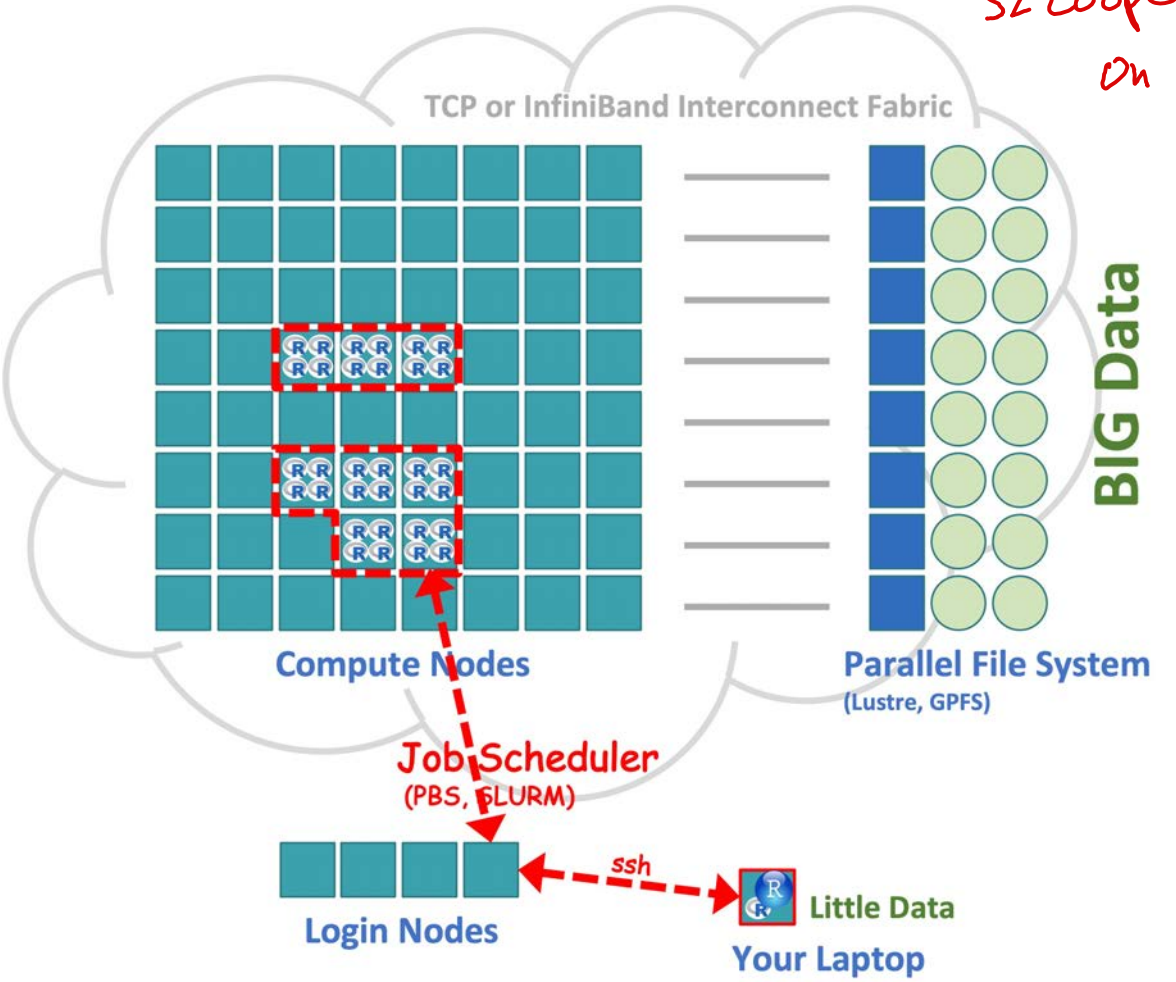


R Interfaces to Low-Level Native Tools

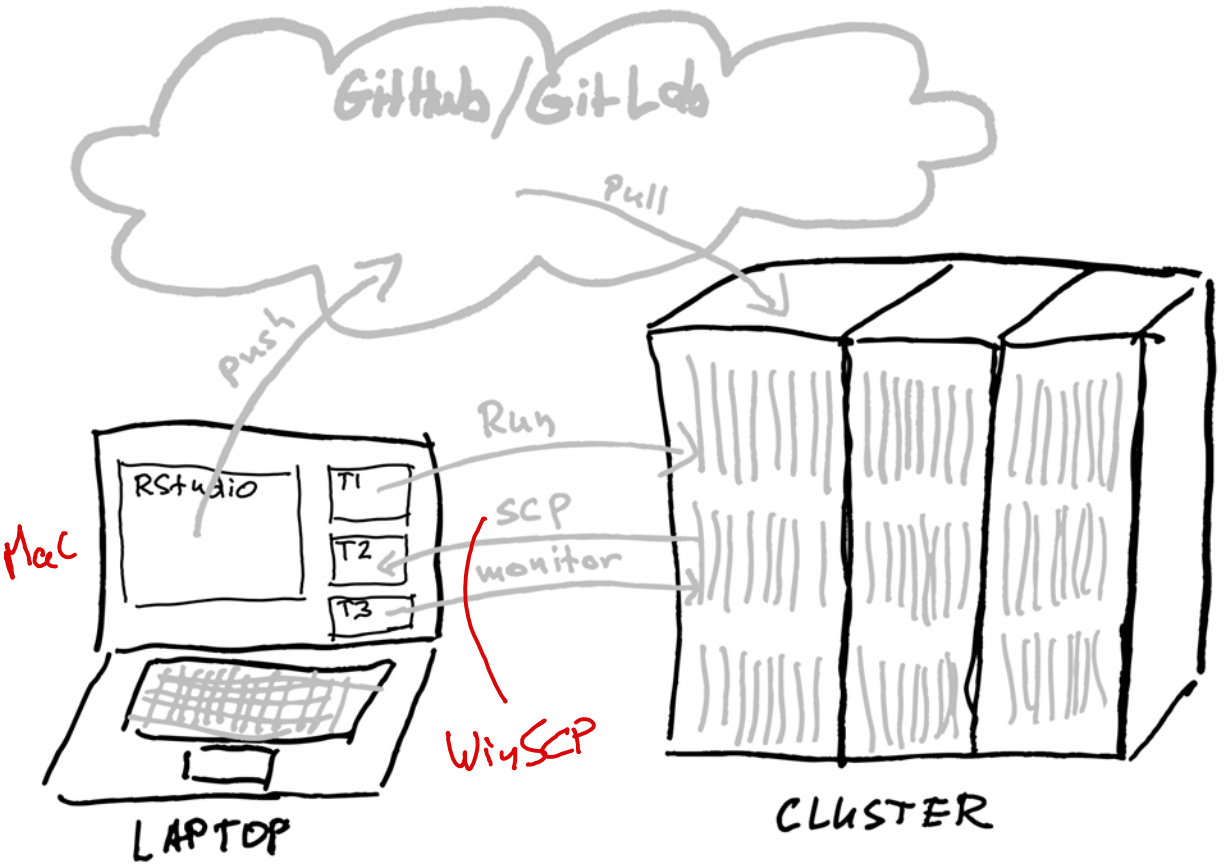


Running Distributed on a Cluster

32 cooperating R sessions on 8 nodes



Typical Workflow from Laptop to Cluster



Unix Concepts and Starting your Workflow

...

Some useful Unix concepts

- Linux is one of many descendants of original Unix. Mac OS X is another.
- Like all file systems, Unix files are organized as a tree.
- When in a terminal, you are talking to a *shell* program (*bash* is most common)
- Commands are looked up in directories listed in your PATH variable
- Spaces delimit commands and options
- > and >> redirect standard output to a file
- *command1* | *command2* pipes standard output1 to standard input2
- **$** means substitute variable value

echo $PATH

- There are many resources on the web to learn Linux basics.

Some useful Linux commands

- **pwd** Show curent directory
- **ls** List files in current directory
  - **ls** *-a* Include files that start with **.**
  - **ls** *-l* Long listing with *permissions*, *owners*, and *last change time*
- **cd** *dir_name* Change directory to dir_name
  - **cd** without dir_name or ~ as dir_name changes to home directory
- **mkdir** *dir_name* Creates directory dir_name
- **rmdir** *dir_name* Deletes directory (must be empty)
- **rm** *file_name* Deletes file_name
- **cat** *file_name* Displays content of entire file_name
- **less** *file_name* Displays content of file_name with paging
- **man** *command* Displays the manual page for *command* with paging
- **which** *command* Returns location of command
- **exit** Quit shell and logout

Demo follows

Slides created via the R package **xaringan** and converted to pdf with **xaringanBuilder**