CSE 5449: Intermediate Studies in Scientific Data Management

Lecture 3: Intro to parallel computing and Software stack of storage and data management

Dr. Suren Byna

The Ohio State University

E-mail: byna.1@osu.edu

https://sbyna.github.io

01/17/2023

Summary of the last class

- Common data formats in science
  - **Homework** – Present a few data structures in the next class
- Brief intro to data storage hierarchy
  - Hardware
  - Software
- Class projects
  - **Homework** – Look at the project options and discuss in the next class

Send me an email if you have any questions regarding the homework or project topics

Today's class

- Data format – Student presentations (2 min each)
- Class projects – questions
- A (very) brief intro to parallel computing
- Parallel I/O software stack

Data formats – Student presentations

Class projects

1. File format comparison

- A comparison of various file formats in performing I/O operations on sequential and parallel storage systems
- Prior work
  - https://arxiv.org/pdf/2207.09503.pdf
- Deliverable: A short paper comparing performance using real scientific data

2. A retrospection of metadata standards in scientific data

- Numerous metadata standards are available
- Question: What's their readiness to be used for finding desired datasets and knowledge in massive amounts of data?
- Deliverable: A short paper with a survey of metadata standards and their usefulness / readiness for querying desired data.

Class projects

3. Performance tuning of High Energy Physics I/O benchmarks

- Question: What's the performance of a realistic use case from a high energy physics benchmark that's representative of the CMS and the ATLAS experiments (from the Large Hadron Collider data sets)
- Benchmark: https://github.com/Dr15Jones/root_serialization
- Deliverable: A short paper describing the current performance and improved performance by applying various tuning options

4. Study of parallel I/O problems and solutions/optimizations explored so far

- Questions
  - What was the parallel I/O problem?
  - How did the authors find a parallel I/O problem?
  - What was the solution?
  - How was the solution applied to fix the problem?
- Background: Various papers available in literature
- Deliverable: A short paper surveying I/O problems, solutions applied, and exposing research gaps (an advanced version of this is a cookbook for I/O performance)

Class projects

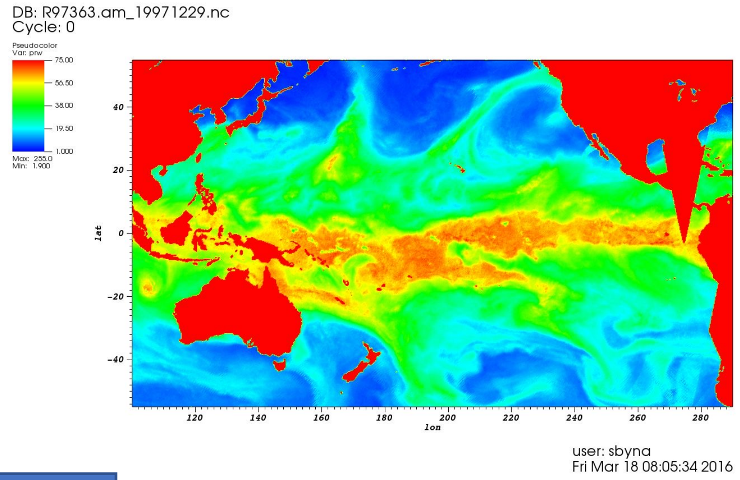
5. Performance comparison of sub-filing in HDF5 and PnetCDF

- Background: Sub-filing is an approach to split a very large file into smaller files. However, there are pros / cons with the approach on how the data is organized.
- Question
  - Which of the HDF5 and PnetCDF sub-filing approaches are best?
  - What better strategies for sub-filing are there?
- Deliverable: A short paper describing

- Before today's class, look at the project topics
  - Discuss your class project interests with me in the next class
  - Think about why are you interested in any of the project topics

Very brief intro to parallel computing

- First – Sequential computing



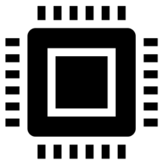
Problem (e.g., detect atmospheric rivers in 10,000 images)

Program Functions / Instructions

Read data | Verify data correctness | Image analysis – step 1 | Image analysis – step 2 | … | Image analysis – step N | Write result

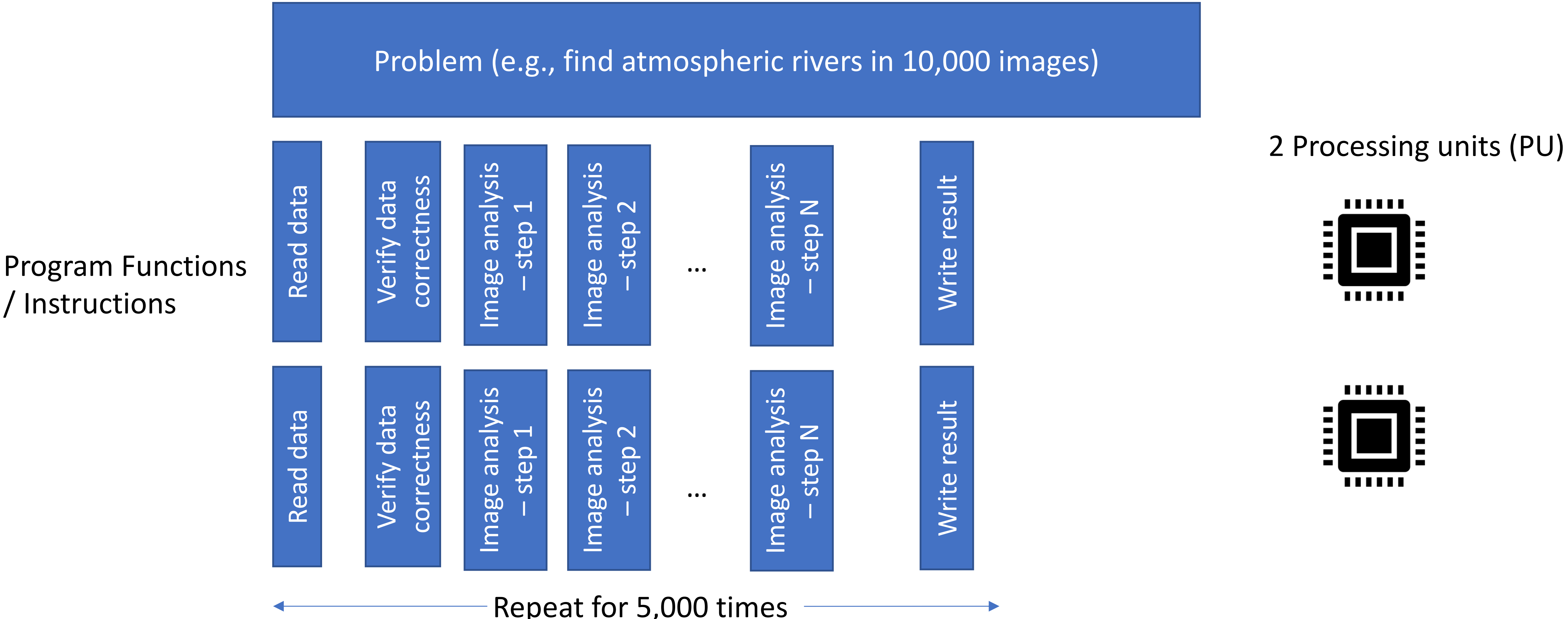
Repeat for 10,000 times

Processing unit (PU)



- Processing 1 image at a time (1 second)
- Total time: 10,000 seconds

Very brief intro to parallel computing – 2 way parallel

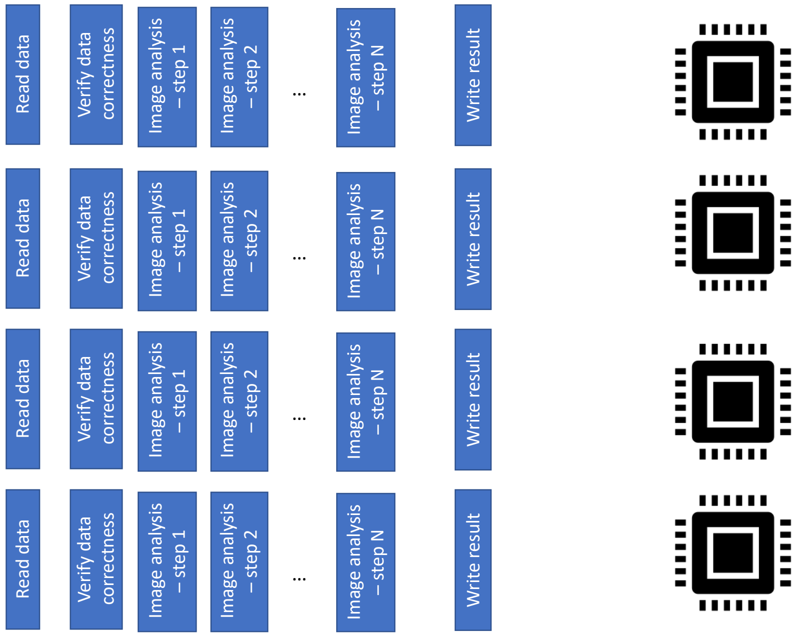


- Processing 1 image at a time (1 second)
- Total time: 10,000 seconds / 2 PU → 5,000 seconds (assuming all PUs are working independently)

Very brief intro to parallel computing – 4 way parallel

Problem (e.g., find atmospheric rivers in 10,000 images)

Program Functions / Instructions



4 Processing units (PU)

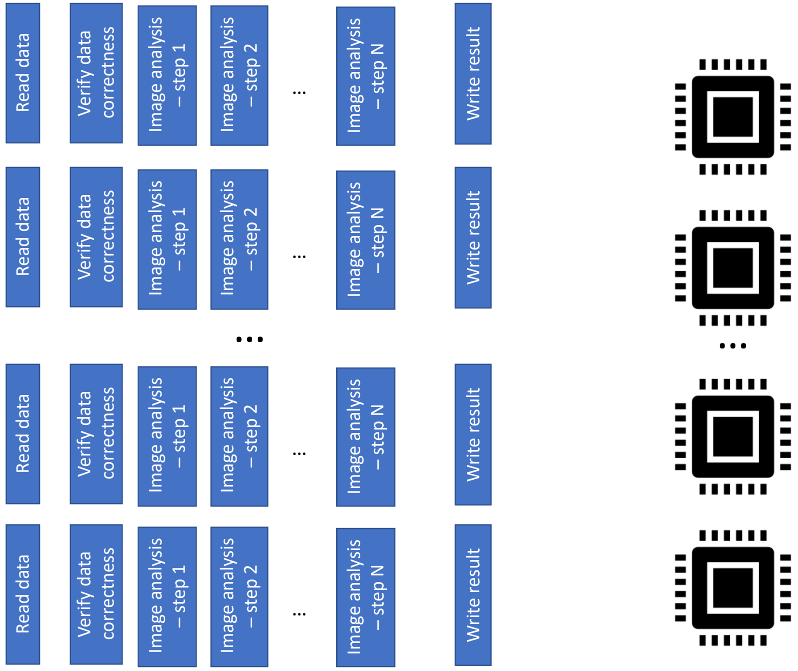
Repeat for 2,500 times

- Processing 1 image at a time (1 second)
- Total time: 10,000 seconds / 4 PUs → 2,500 seconds (assuming all PUs are working independently)

Very brief intro to parallel computing – 10,000 way parallel

Problem (e.g., find atmospheric rivers in 10,000 images)

Program Functions / Instructions



10,000 Processing units (PU)

- Processing 1 image at a time (1 second)
- Total time: 10,000 seconds / 10,000 PUs → 1 second (assuming all PUs are working independently)

Different types of parallelism – Flynn's taxonomy

- Problem – Data stream
- Work – Instruction stream

- Single
- Multiple

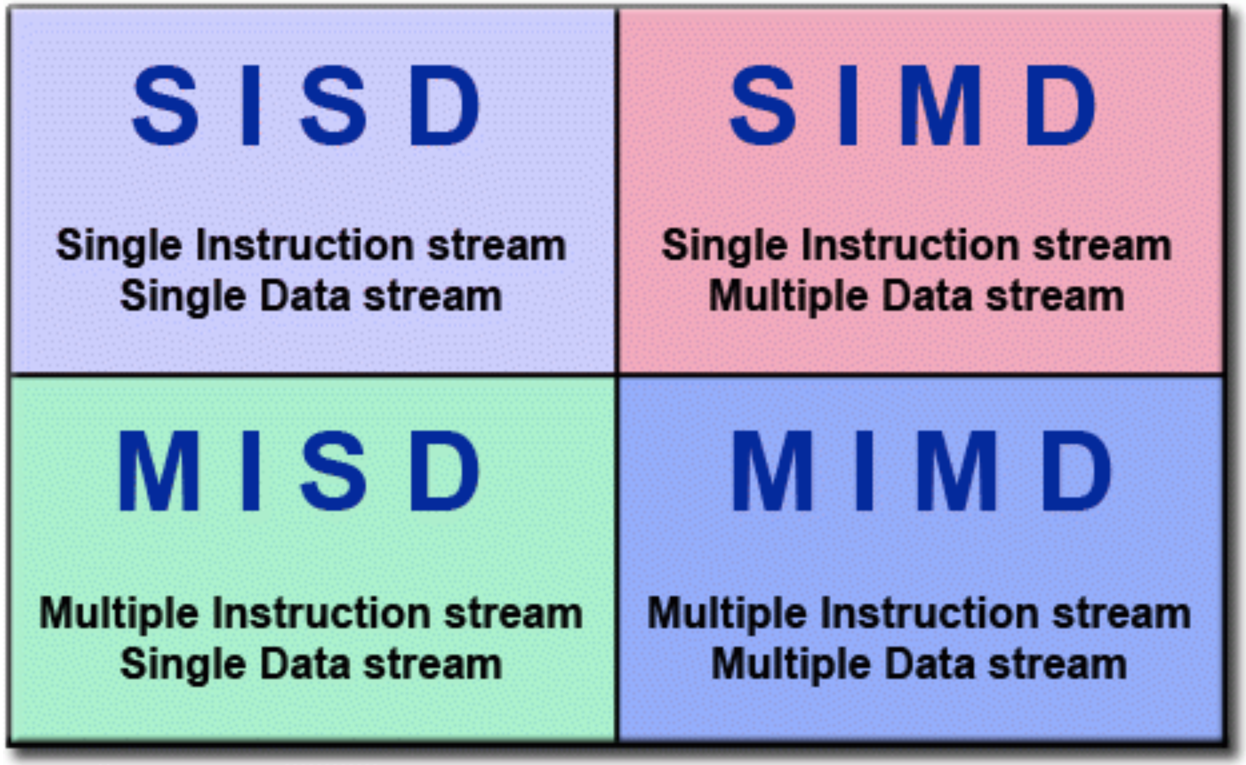


Image from LLNL parallel computing tutorial
https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial

Flynn's taxonomy

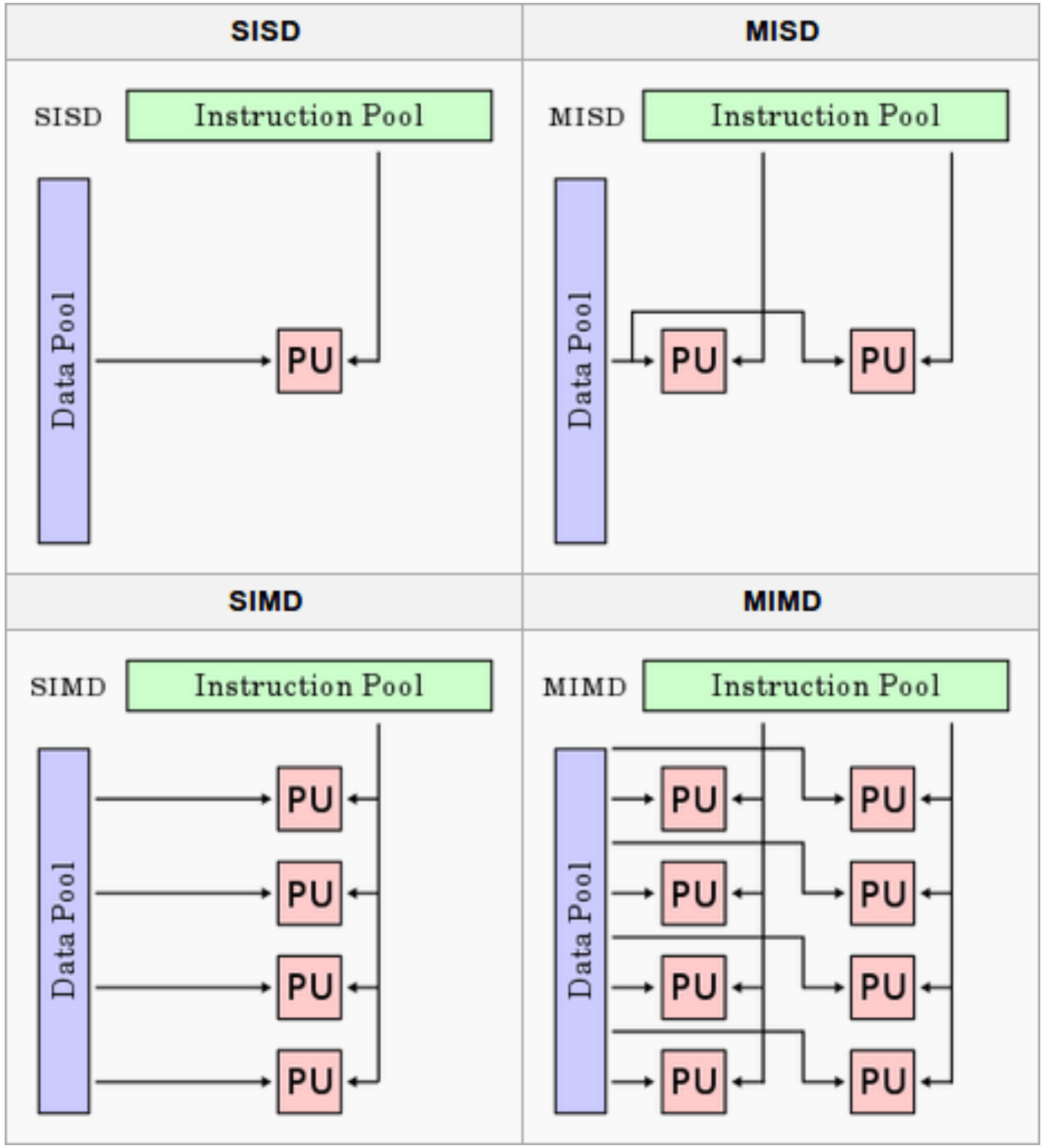
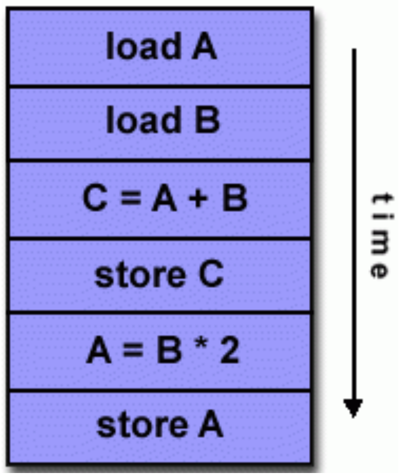
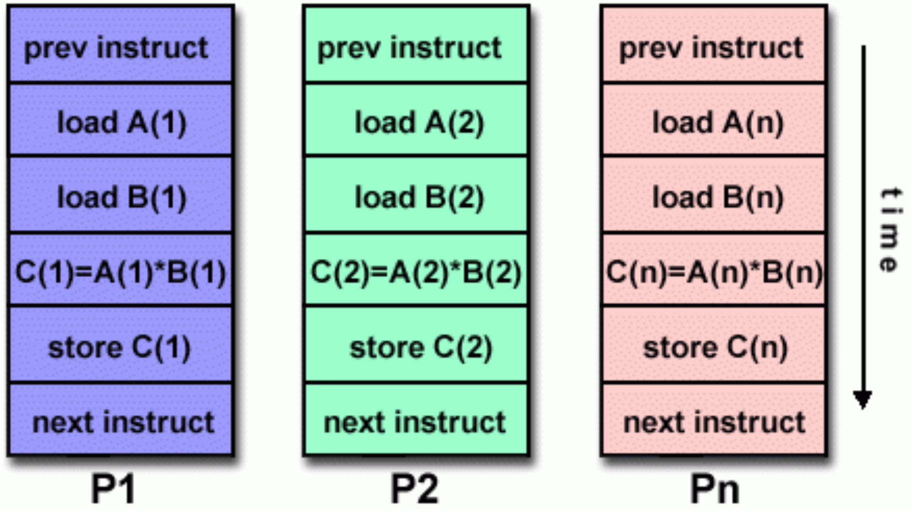


Image from https://www.thephysicsmill.com/2014/07/27/parallel-computing-primer/, originally from Wikipedia

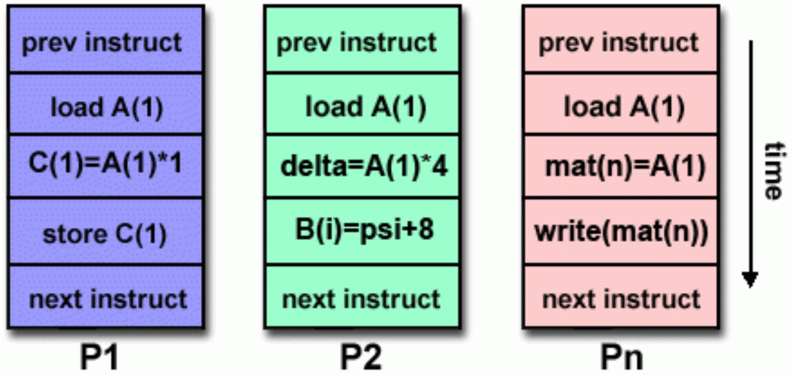
Flynn's taxonomy

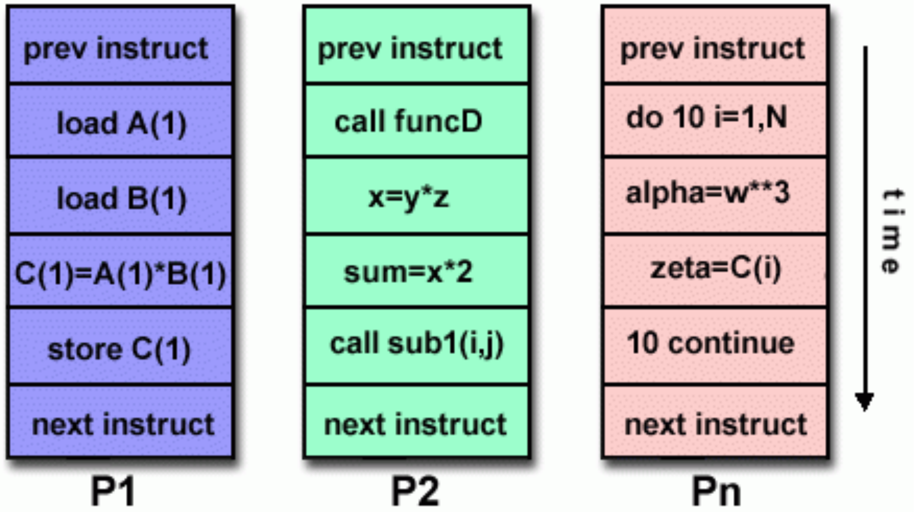
SISD





MISD





Images from LLNL parallel computing tutorial - https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial

Generalization of parallel computing

- Assume that we have **P** processing units
- Problem size is **N** (could be images or equations, or any work)

- Parallelization steps
  - Partition work across processes
  - Each process works on its problem

  - Write the output

Generalization of parallel computing – communication comes in

- Assume that we have **P** processing units
- Problem size is **N** (could be images or equations, or any work)

- Parallelization steps
  - Partition work across processes
  - Each process works on its problem
  - Communicate / synchronize with other processes
  - Write the output

A bit more on parallel computing

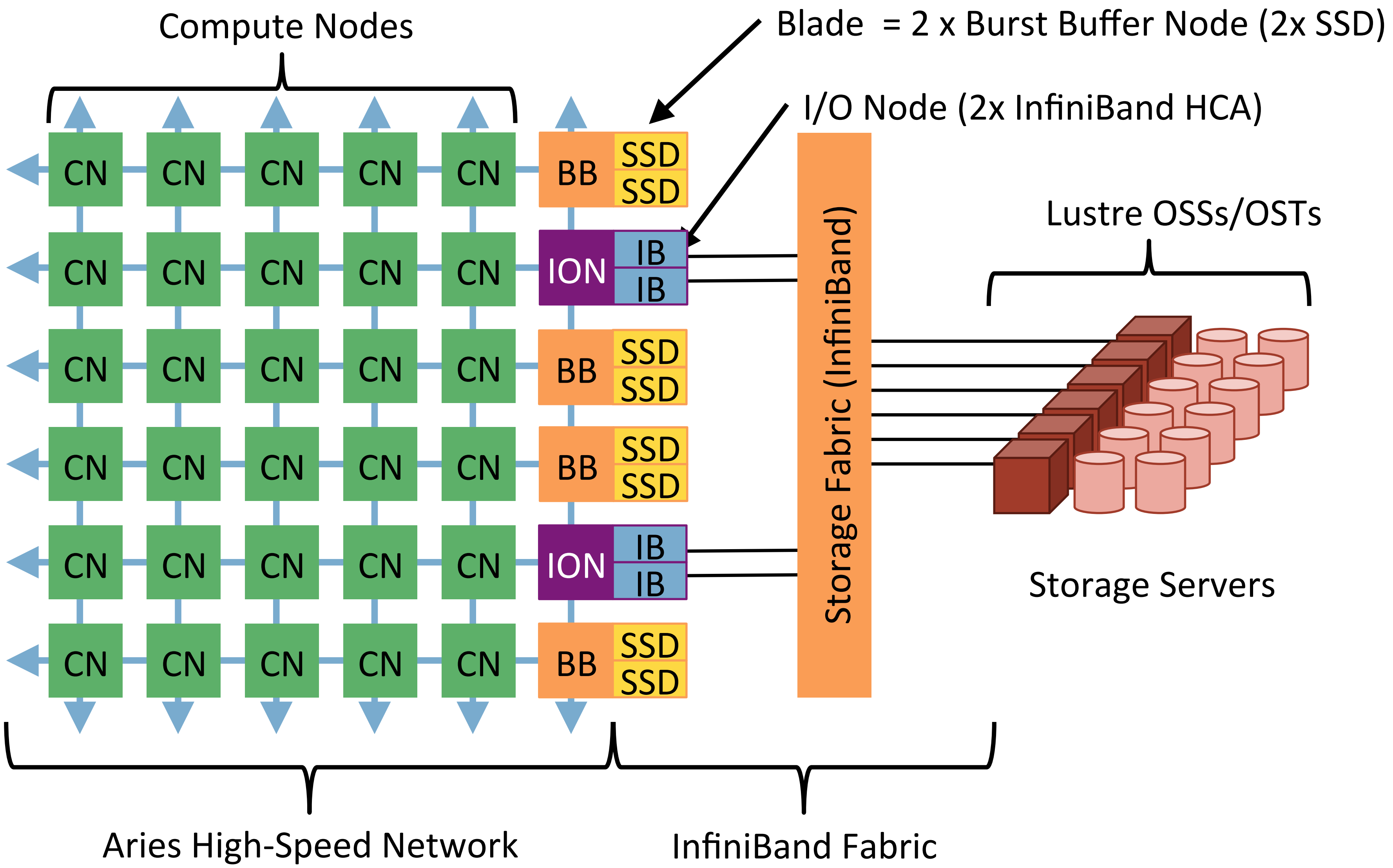
- Data parallel – Same instructions are performed simultaneously on different / multiple data items – Single Instruction, Multiple Data (SIMD)
- Task parallel – Different instructions on different data items – Multiple Instructions, Multiple Data (MIMD)
- Single Program, Multiple Data (SPMD) – synchronization among processes less frequently
- Message Passing Interface (MPI)
  - A standard for multiple processes in a parallel program to communicate and synchronize
  - MPI is for SPMD / MIMD parallelism
  - Will discuss MPI in one of the next week's classes

Further reading on parallel computing

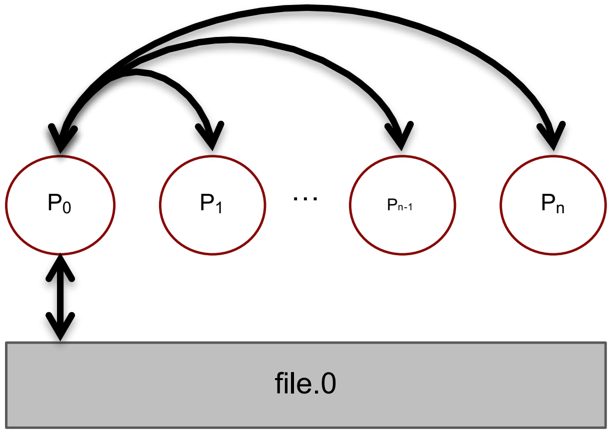
- A tutorial from Lawrence Livermore National Laboratory
  - https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial
- The physics mill
  - https://www.thephysicsmill.com/2014/07/27/parallel-computing-primer/
- YouTube

- Message Passing Interface (MPI)
  - https://hpc-tutorials.llnl.gov/mpi/
  - https://www.mcs.anl.gov/research/projects/mpi/
  - https://www.mcs.anl.gov/research/projects/mpi/tutorial/gropp/talk.html
  - https://mpitutorial.com/tutorials/

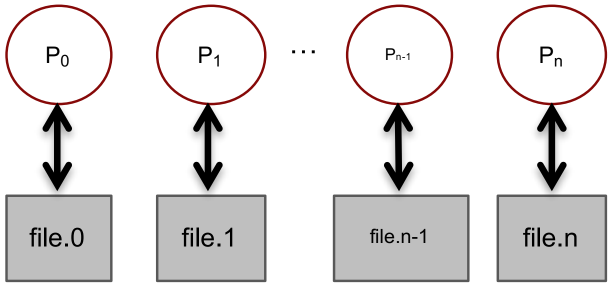
A typical supercomputer architecture



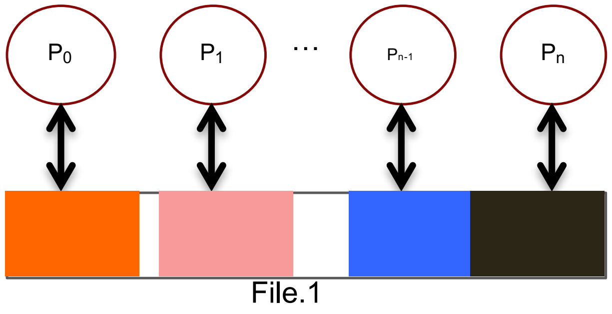
Parallel I/O



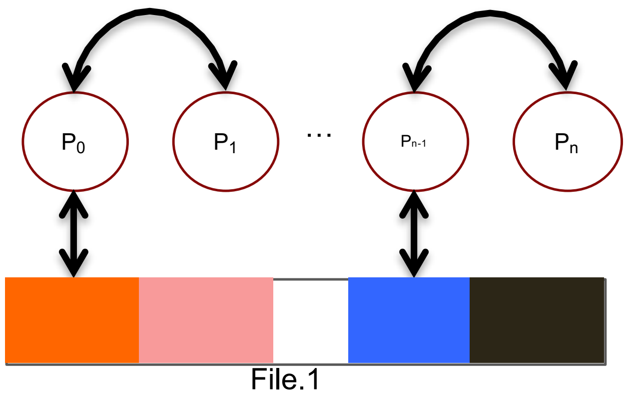
1 Writer/Reader, 1 File



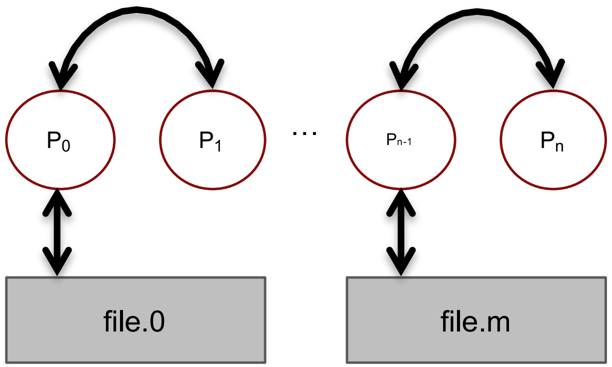
n Writers/Readers, n Files



n Writers/Readers, 1 File



M Writers/Readers, 1 File



M Writers/Readers, M Files

Summary of today's class

- Class projects
  - Homework:
    - Go through the projects and discuss if there are any questions / concerns
    - Select one project and let me know which one you would like to work on – Jan 26th
    - Provide an initial plan of execution – list tasks and timelines – Jan 26th
- What is parallel computing?
- High-level concept of parallel I/O

After the class, slides are uploaded to:
https://osu.instructure.com/courses/141406/files

Also available at: https://sbyna.github.io/teaching/5449-sdm.html

Next class

- Discussion of class projects you selected
- High-level I/O libraries
- HDF5