جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

Introduction to parallel I/O

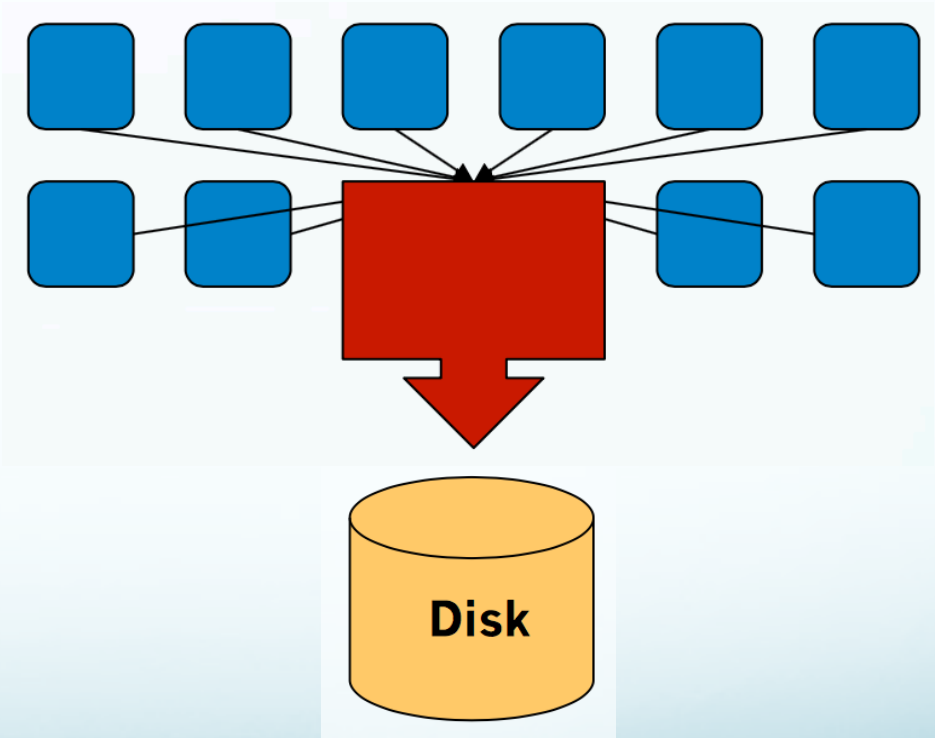
- I/O can create bottlenecks
  - I/O components are much slower than the compute parts of a supercomputer
  - If the bandwidth is saturated, larger scale of execution can not improve the I/O performance
- Parallel I/O is needed to
  - Do more science than waiting files to be read/written
  - No waste of resources
  - Not stressing the file system, thus affecting other users

I/O Performance

- There is no magic solution
- I/O performance depends on the pattern
- Of course a bottleneck can occur from any part of an application
- Increasing computation and decreasing I/O is a good solution but not always possible

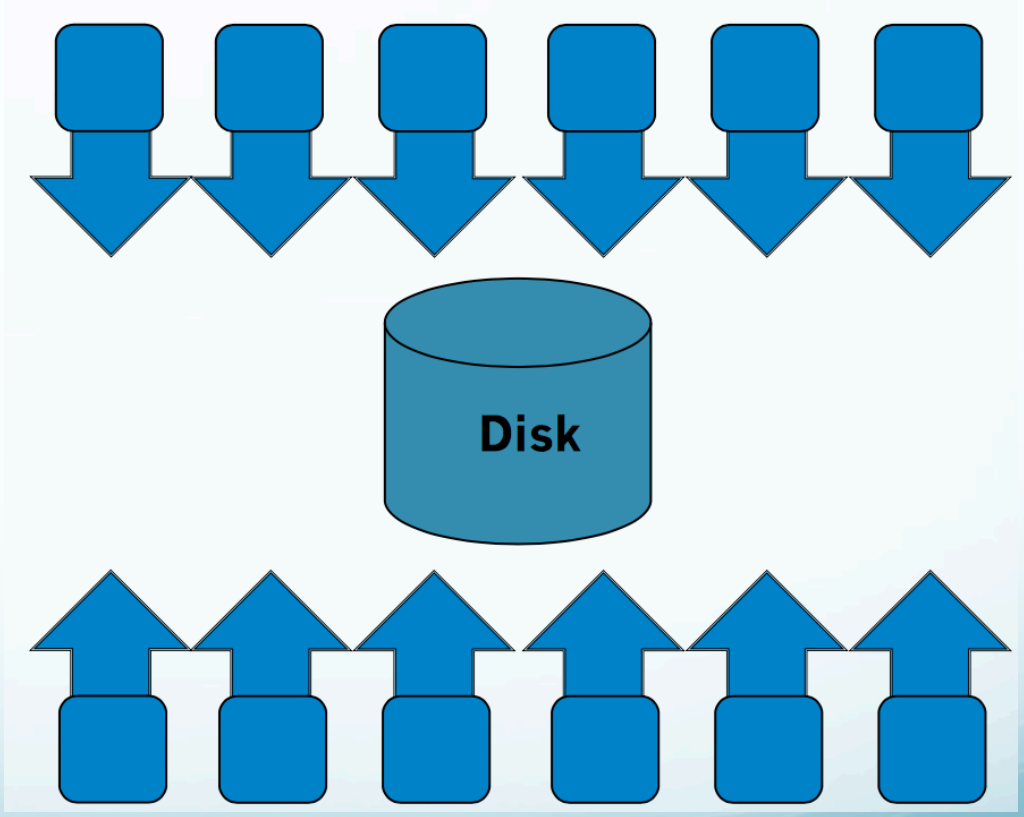
Serial I/O

- Only one process performs I/O (default option for WRF)
  - Data Aggregation or Duplication
  - Limited by single I/O process
- Simple solution but does not scale
- Time increases with amount of data and also with number of processes



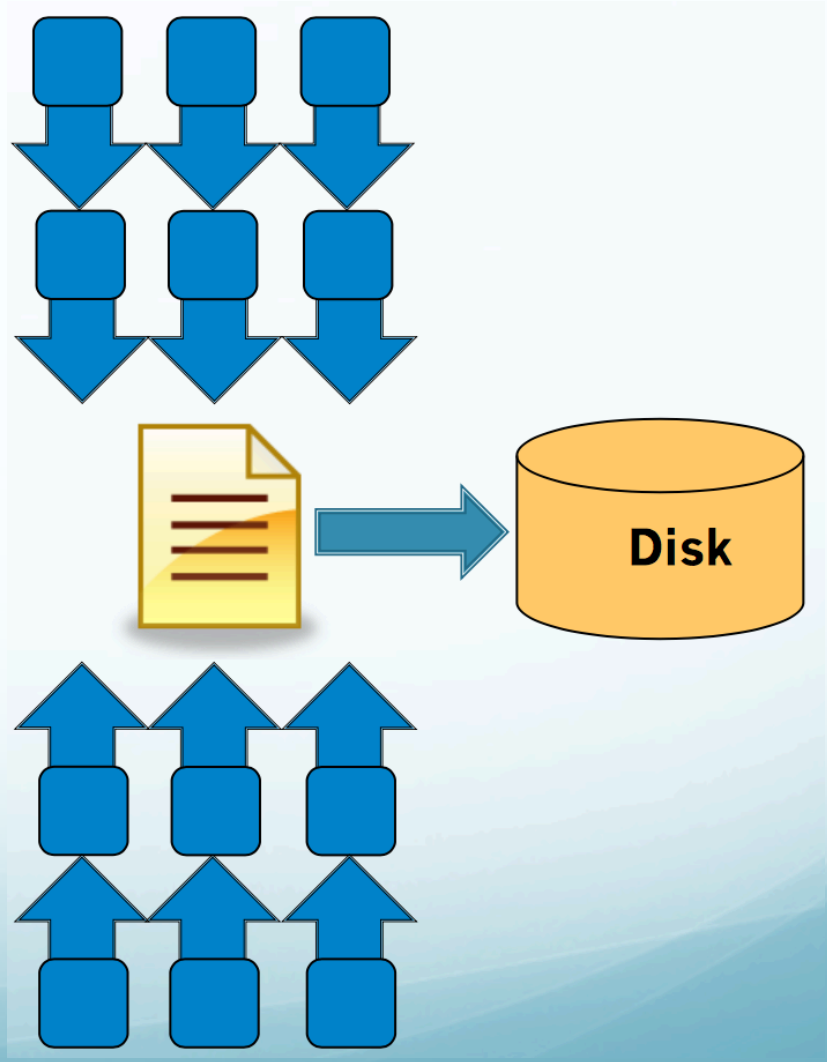
Parallel I/O: File-per-Process

- All processes read/write their own separate file
  - The number of the files can be limited by file system
  - Significant contention can be observed



Parallel I/O: Shared File

- Shared File
  - One file is accessed from all the processes
  - The performance depends on the data layout
  - Large number of processes can cause contention



جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Pattern Combinations

- Subset of processes perform I/O
  - **Aggregation** of a group of processes data
  - I/O process may access independent files
  - Group of processes perform parallel I/O to a shared file