High Performance Computing Systems (CMSC714)

Lecture 20: Parallel I/O

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Summary of last lecture

- Task mapping can be used to optimize the placement of MPI processes within a job allocation
- Can reduce inter-node communication volume and optimize it
- Heuristic-based approaches
- Metrics: hop-count, hop-bytes

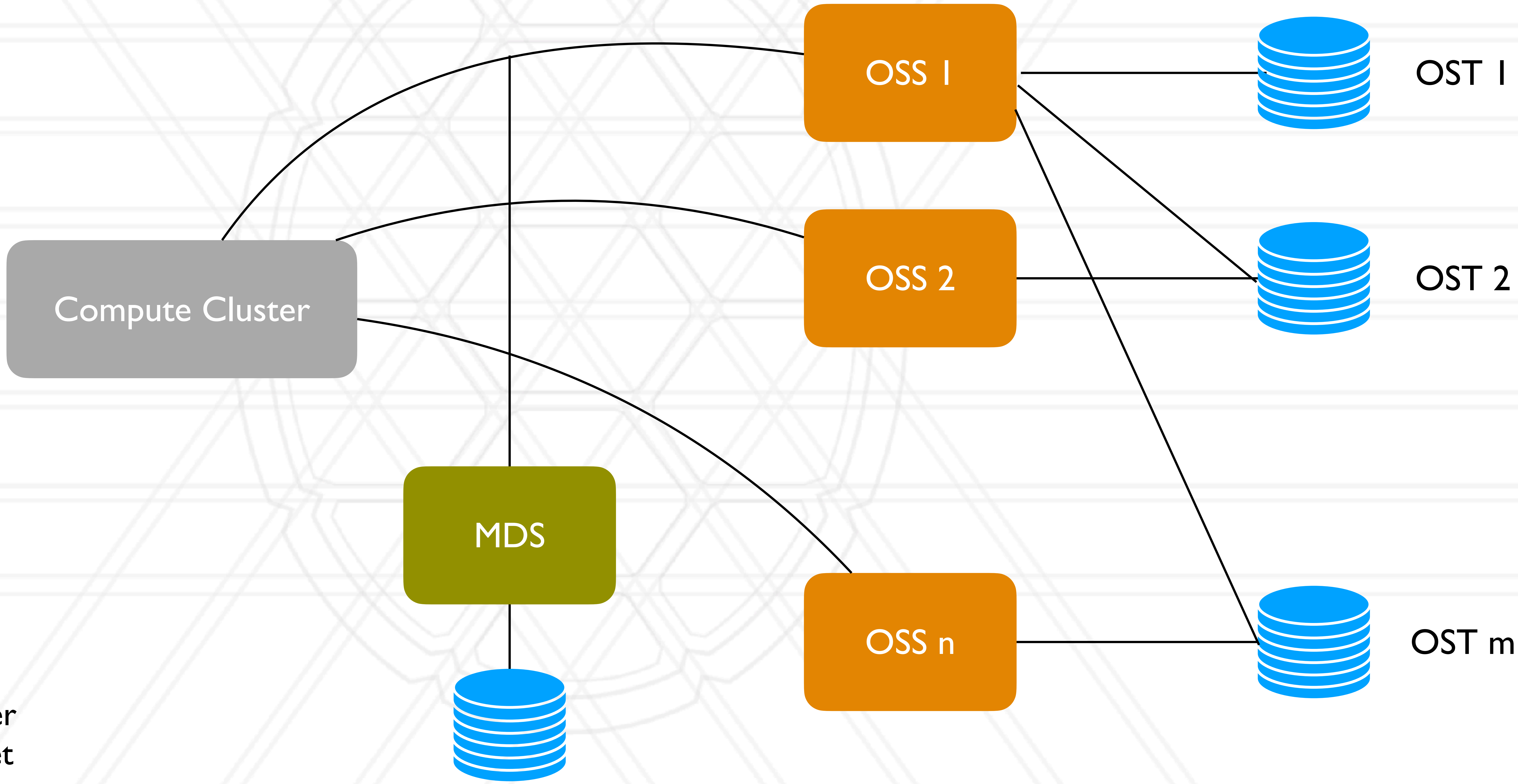
When do parallel programs perform I/O?

- Reading input datasets
- Writing numerical output
- Writing checkpoints

Non-parallel I/O

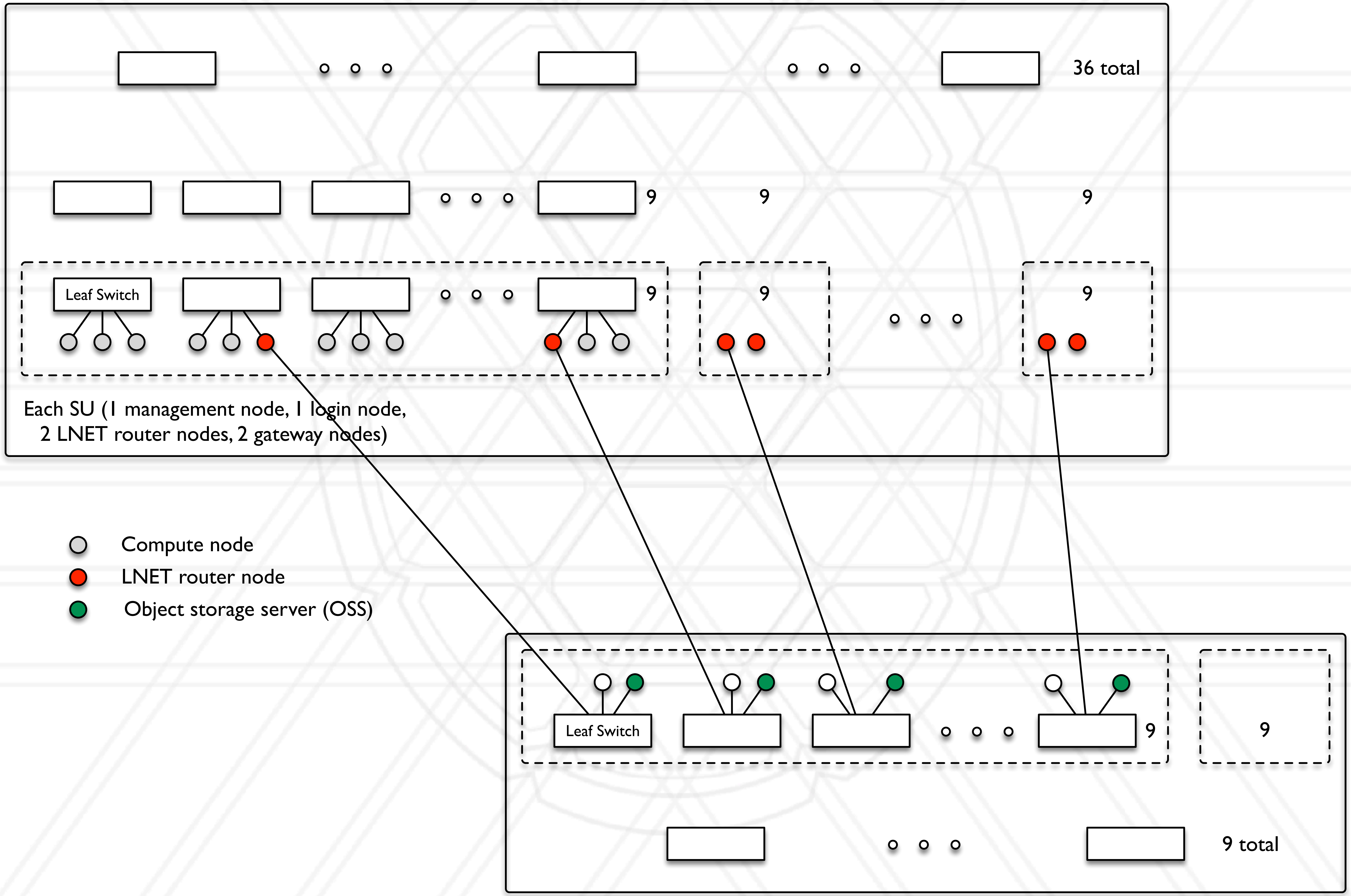
- Designated process does I/O
- All processes send data to/receive data from that one process
- Not scalable

Parallel filesystem



MDS = Metadata Server
MDT = Metadata Target
OSS = Object Storage Server
OST = Object Storage Target

Links between cluster and filesystem



Different parallel filesystems

- Lustre: open-source (lustre.org)
- GPFS: General Parallel File System from IBM, now called Spectrum Scale
- PVFS: Parallel Virtual File System

Tape drive (archive) and burst buffers

- Store copy of data on magnetic tapes for archival
- Burst buffers: fast, intermediate storage between compute nodes and the parallel filesystem
- Two designs:
  - Node-local burst buffer
  - Remote (shared) burst buffer

I/O libraries

- High-level libraries: HDF5, NetCDF
- Middleware: MPI-IO
- Low-level: POSIX IO

Different I/O patterns

- One process reading/writing all the data
- Multiple processes reading/writing data from/to shared file
- Multiple processes reading/writing data from/to different files
- Different performance depending upon number of readers/writers, file sizes, filesystem etc.

Questions?

UNIVERSITY OF MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu