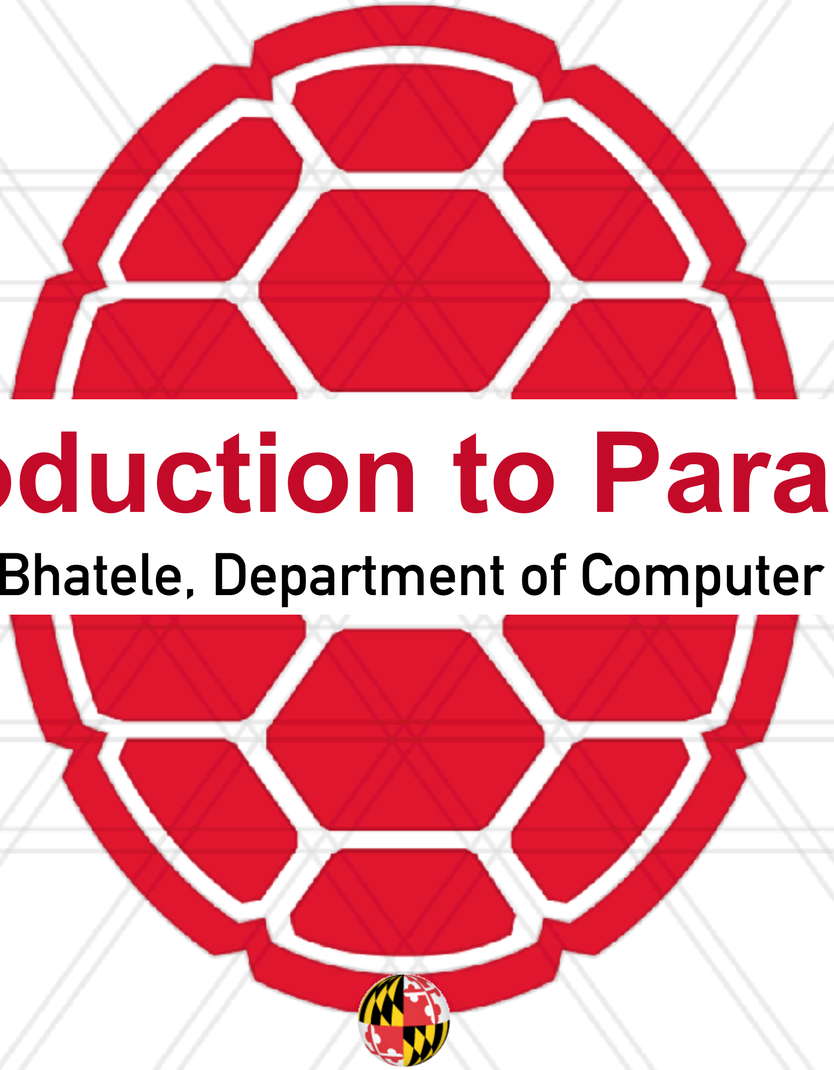
High Performance Computing Systems (CMSC714)



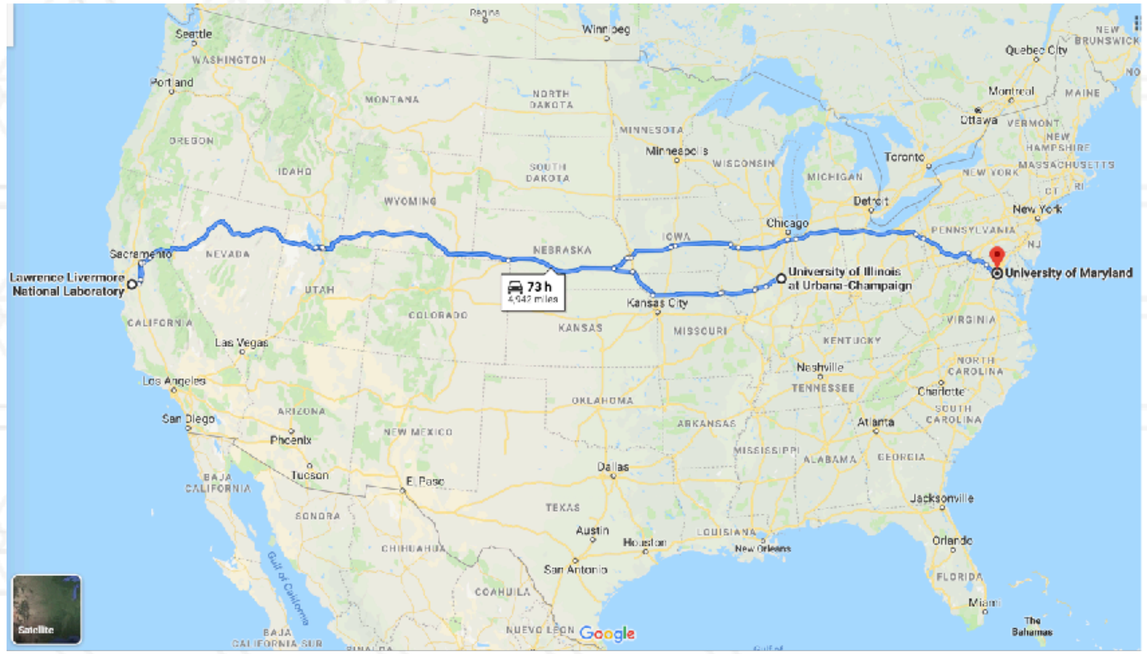
Lecture 1: Introduction to Parallel Computing

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

About the instructor

- Ph.D. from the University of Illinois
- Spent eight years at Lawrence Livermore National Laboratory
- Started at the University of Maryland in 2019



This course is

- A seminar-style course on the history and recent advances in parallel computing
  - Will cover programming models, architectures, tools, systems issues, algorithms and applications
- A qualifying course for MS/PhD: Computer Systems
- Work expected:
  - Three programming assignments
  - Midterm exam: due on April 20
  - Group project (3 students per group)
  - Classroom participation (present papers and send summaries)

Synchronous online lectures

If attending class synchronously is going to cause you unusual difficulty or hardship (distant time zone or poor internet connectivity, etc.), please schedule a time in the first two weeks of the semester for us to meet and develop an alternative individualized participation plan.

Excused absence

Any student who needs to be excused for an absence from a single lecture, due to a medically necessitated absence shall make a reasonable attempt to inform the instructor of his/her illness prior to the class. Upon returning to the class, present the instructor with a self-signed note attesting to the date of their illness. Each note must contain an acknowledgment by the student that the information provided is true and correct. Providing false information to University officials is prohibited under Part 9(i) of the Code of Student Conduct (V-1.00(B) University of Maryland Code of Student Conduct) and may result in disciplinary action.

Self-documentation may not be used for Major Scheduled Grading Events (midterm exams, and final project presentations) and it may only be used for two class meetings during the semester. Any student who needs to be excused for a prolonged absence (two or more consecutive class meetings), or for a Major Scheduled Grading Event, must provide written documentation of the illness from the Health Center or from an outside health care provider. This documentation must verify dates of treatment and indicate the timeframe that the student was unable to meet academic responsibilities. In addition, it must contain the name and phone number of the medical service provider to be used if verification is needed. No diagnostic information will ever be requested.

Course topics

- Introduction to parallel computing (1 week)
- Programming models (2 weeks)
- Parallel architectures and networks (2 weeks)
- Performance analysis, tools and modeling (2 weeks)
- Systems issues (2 weeks)
- Parallel algorithms and simulation codes (3 weeks)

Tools we will use for the class

- Syllabus, lecture slides, assignment descriptions on course website:
  - http://www.cs.umd.edu/class/spring2021/cmsc714
- Class on zoom (meeting details sent by email and on ELMS)
- Video recordings on Panopto
- Assignment submissions on ELMS
- Midterm submission via Gradescope

Deepthought2 accounts

- Joy will email your login/password for deepthought2
- Helpful resources:
  - https://www.glue.umd.edu/hpcc/help/usage.html
  - https://hpcbootcamp.readthedocs.io
  - http://www.cs.umd.edu/class/spring2021/cmsc714/deepthought2.shtml

Introductions

- Name
- MS or PhD / Major
- Area of research
- Why this course?
- Something interesting/ unique about yourself

What is parallel computing?

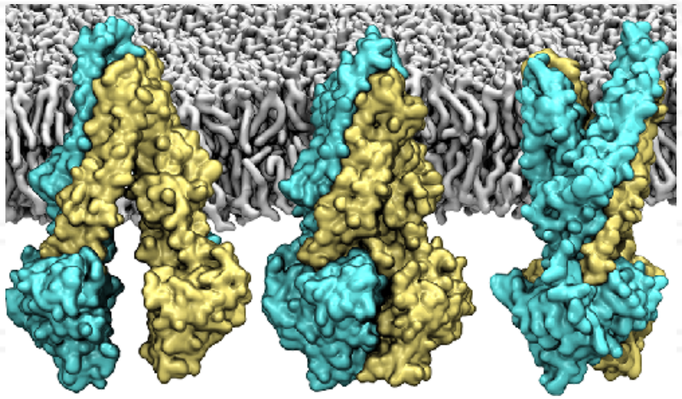
- Serial or sequential computing: doing a task in sequence on a single processor
- Parallel computing: breaking up a task into sub-tasks and doing them in parallel (concurrently) on a set of processors (often connected by a network)
- Some tasks do not need any communication: embarrassingly parallel

What is parallel computing?

- Does it include:
  - Grid computing
  - Distributed computing
  - Cloud computing
- Does it include:
  - Superscalar processors
  - Vector processors
  - Accelerators (GPUs, FPGAs)

The need for high performance computing

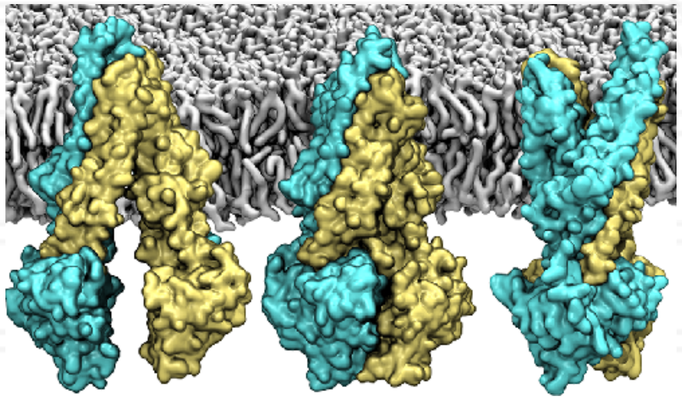
Drug discovery



https://www.nature.com/articles/nature21414

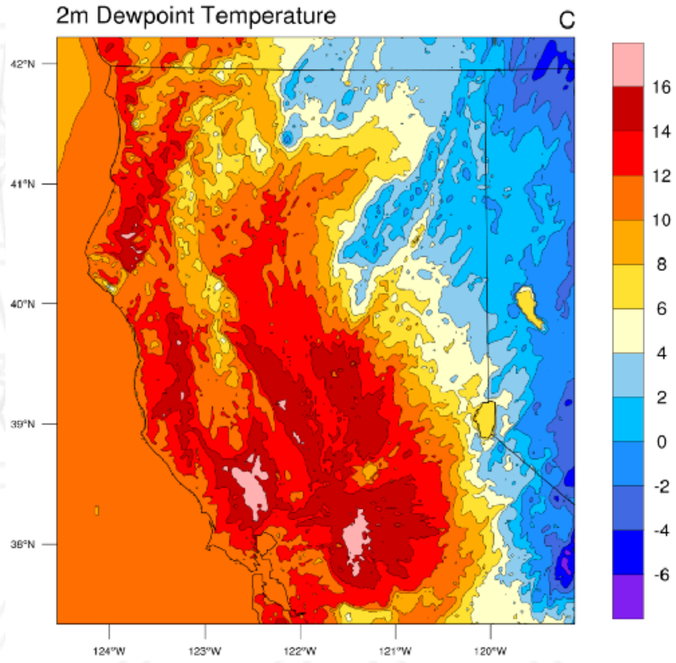
The need for high performance computing

Drug discovery



https://www.nature.com/articles/nature21414

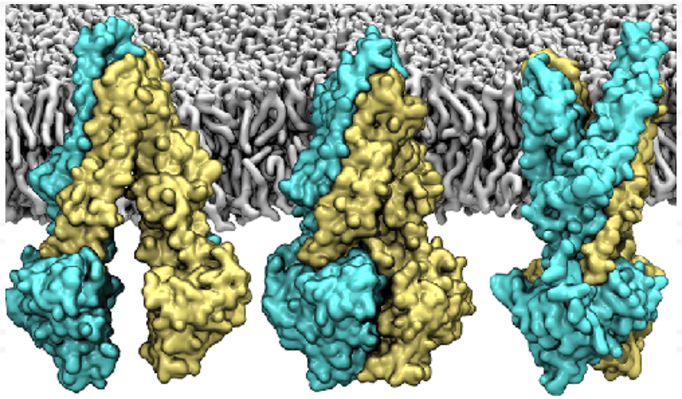
Weather forecasting



https://www.ncl.ucar.edu/Applications/wrf.shtml

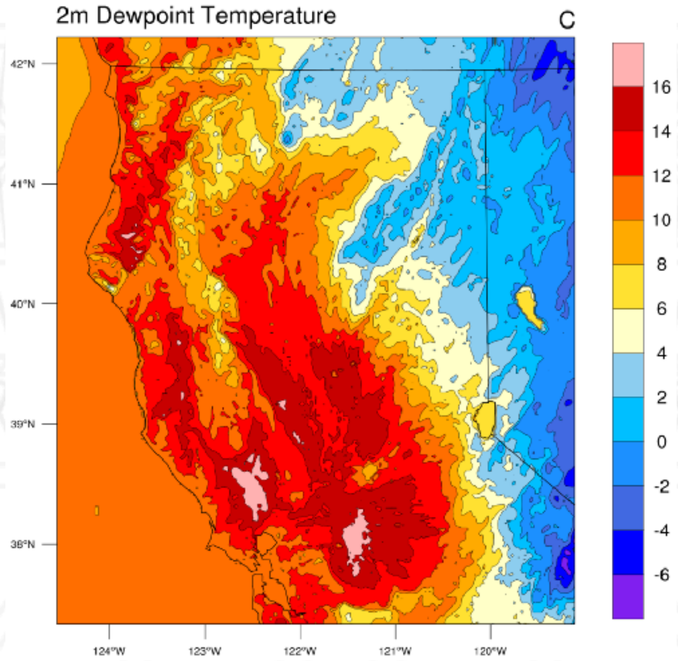
The need for high performance computing

Drug discovery



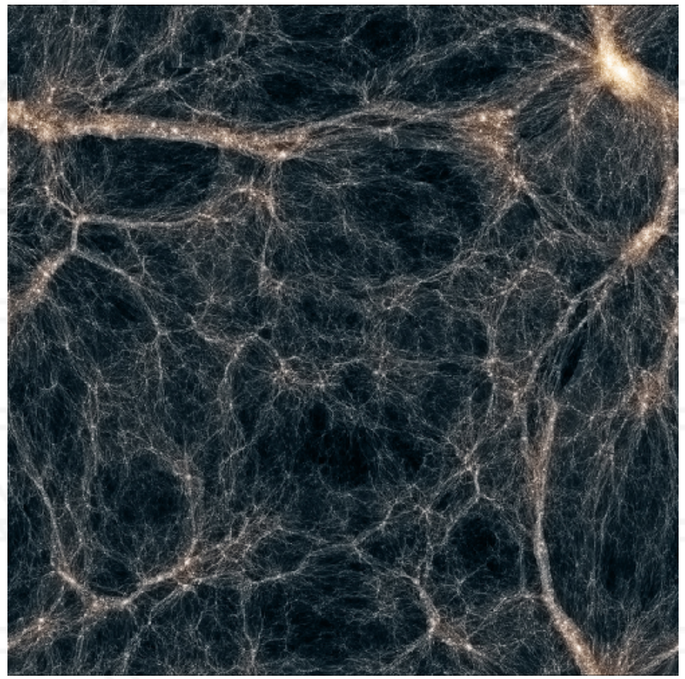
https://www.nature.com/articles/nature21414

Weather forecasting



https://www.ncl.ucar.edu/Applications/wrf.shtml

Study of the universe



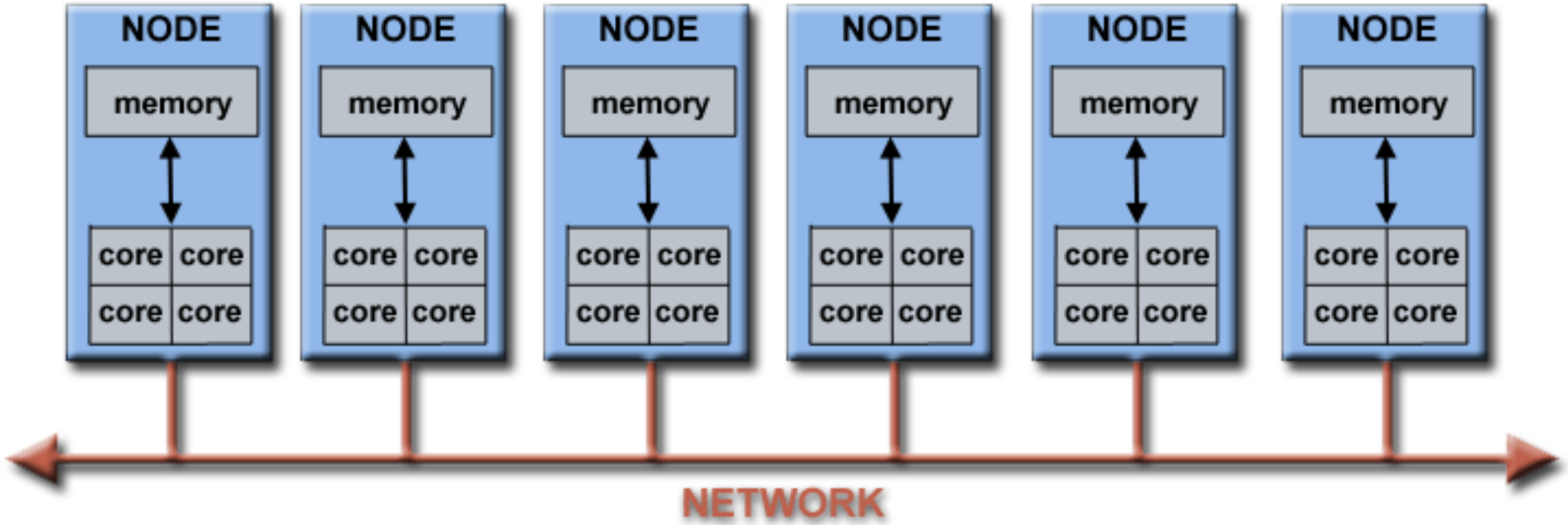
https://www.nas.nasa.gov/SC14/demos/demo27.html

Why do we need parallelism?

- Make some science simulations feasible in the lifetime of humans
  - Either due to speed or memory requirements
- Provide answers in realtime or near realtime

Parallel architecture

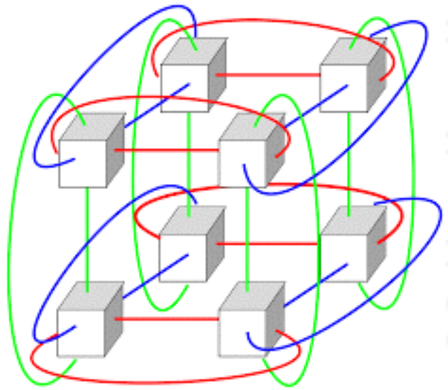
- A set of nodes or processing elements connected by a network.



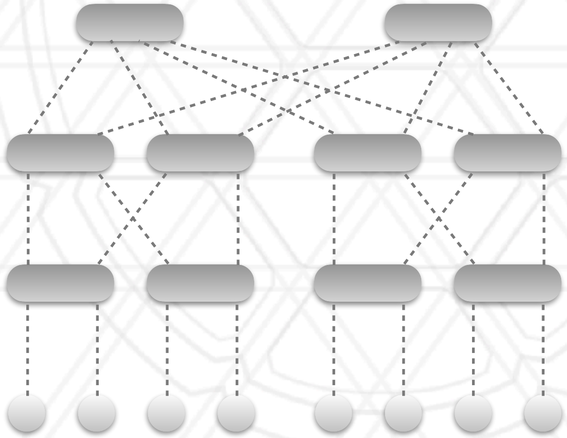
https://computing.llnl.gov/tutorials/parallel_comp

Interconnection networks

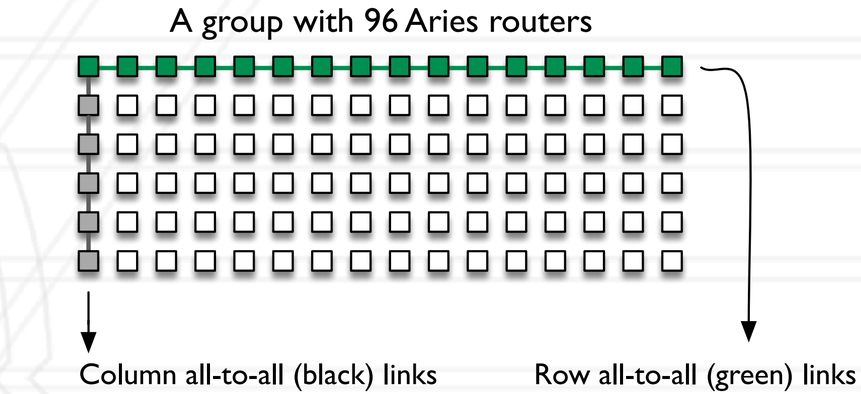
- Different topologies for connecting nodes together
- Used in the past: torus, hypercube
- More popular currently: fat-tree, dragonfly

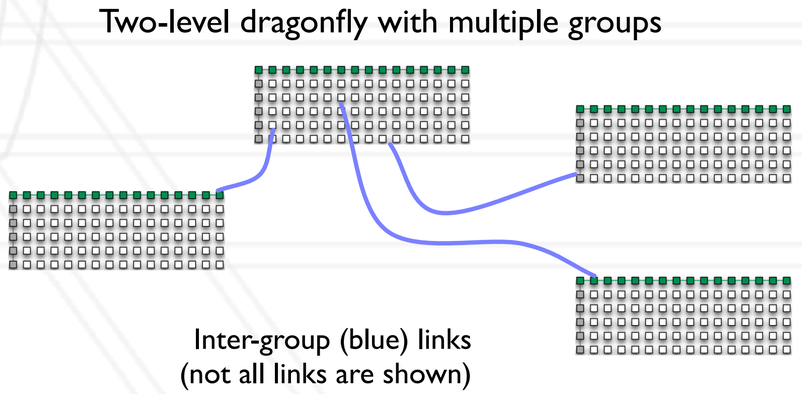


Torus



Fat-tree

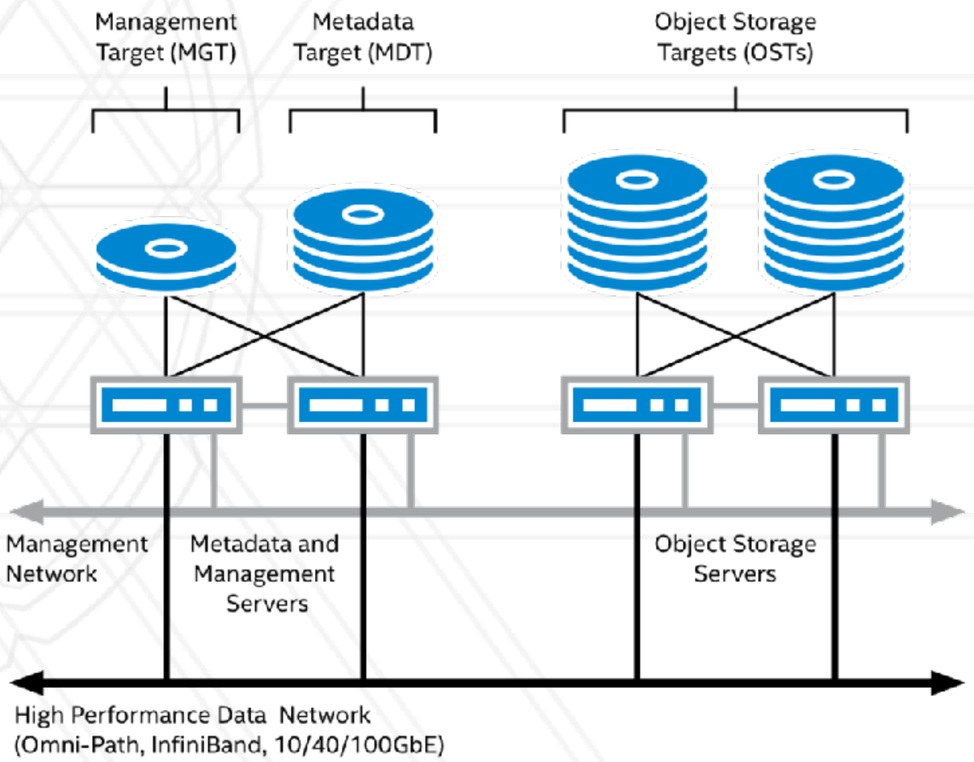




Dragonfly

I/O sub-system / Parallel file system

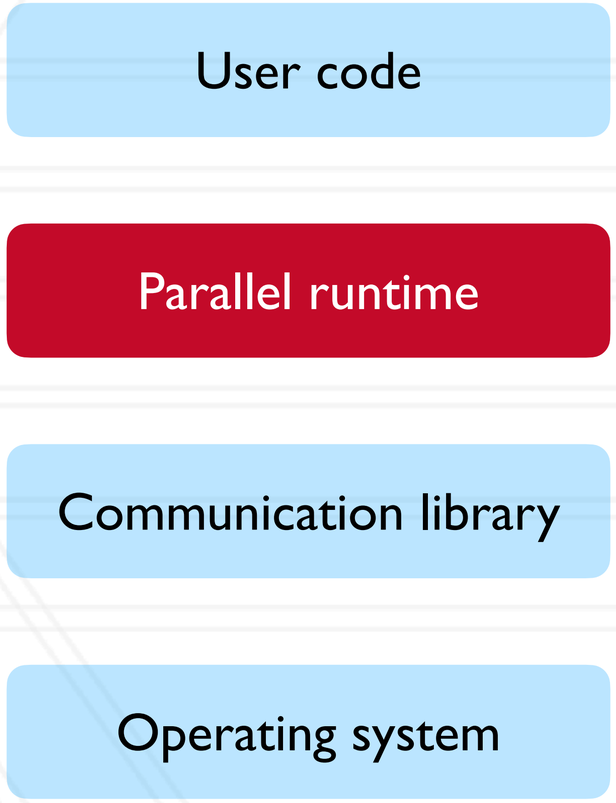
- Home directories and scratch space typically on a parallel file system
- Mounted on all login and compute nodes



http://wiki.lustre.org/Introduction_to_Lustre

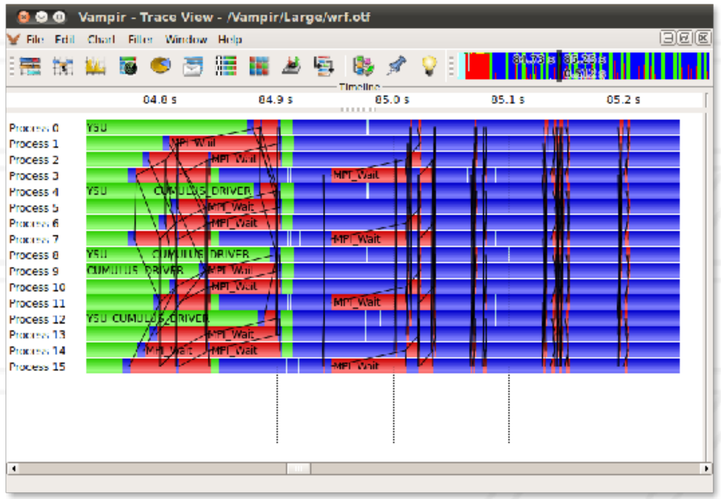
System software: models and runtimes

- Parallel programming model
  - Parallelism is achieved by making calls to a library and the execution model depends on the library used.
- Parallel runtime [system]:
  - Implements the parallel execution model
- Shared memory/address-space
  - Pthreads, OpenMP
- Distributed memory
  - MPI, Charm

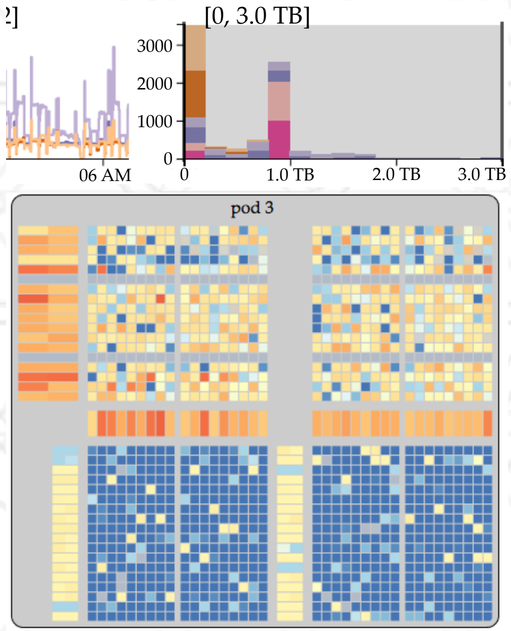


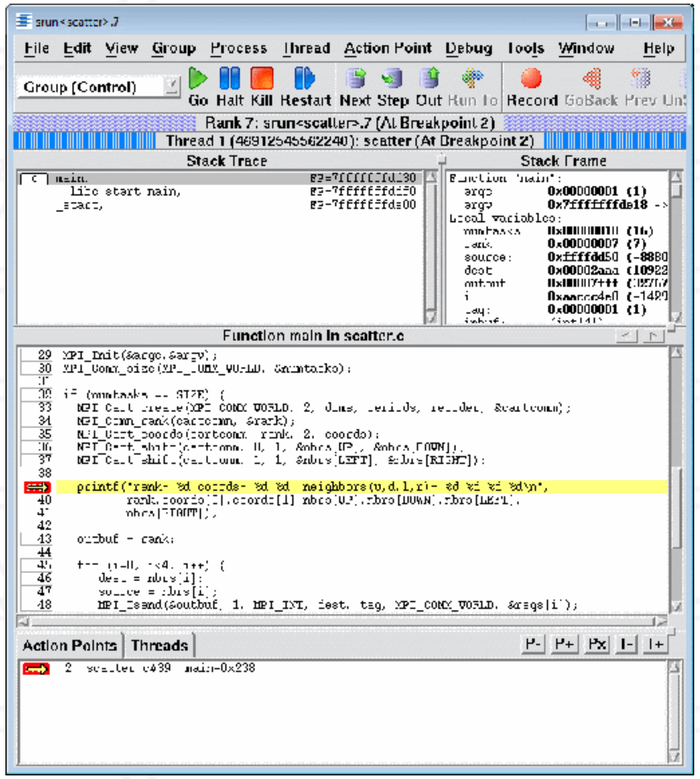
Performance and debugging tools

- Debugging parallel programs is challenging
- Performance analysis and tuning is critical but hard



https://vampir.eu/tutorial/manual/performance_data_visualization





https://computing.llnl.gov/tutorials/totalview/

Systems issues

- Operating system noise
- Network congestion
  - Congestion-avoiding routing
  - Parallel I/O
- Job scheduling:
  - Interference from other jobs

Parallel algorithms and applications

- Parallel Matrix Multiplication
- Parallel Sorting
- N-body calculations
- Discrete event simulations

Parallel algorithms and applications

- Parallel Matrix Multiplication
- Parallel Sorting
- N-body calculations
- Discrete event simulations
- Molecular dynamics
- Computational cosmology
- Weather and climate modeling
- Discrete-event simulation

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