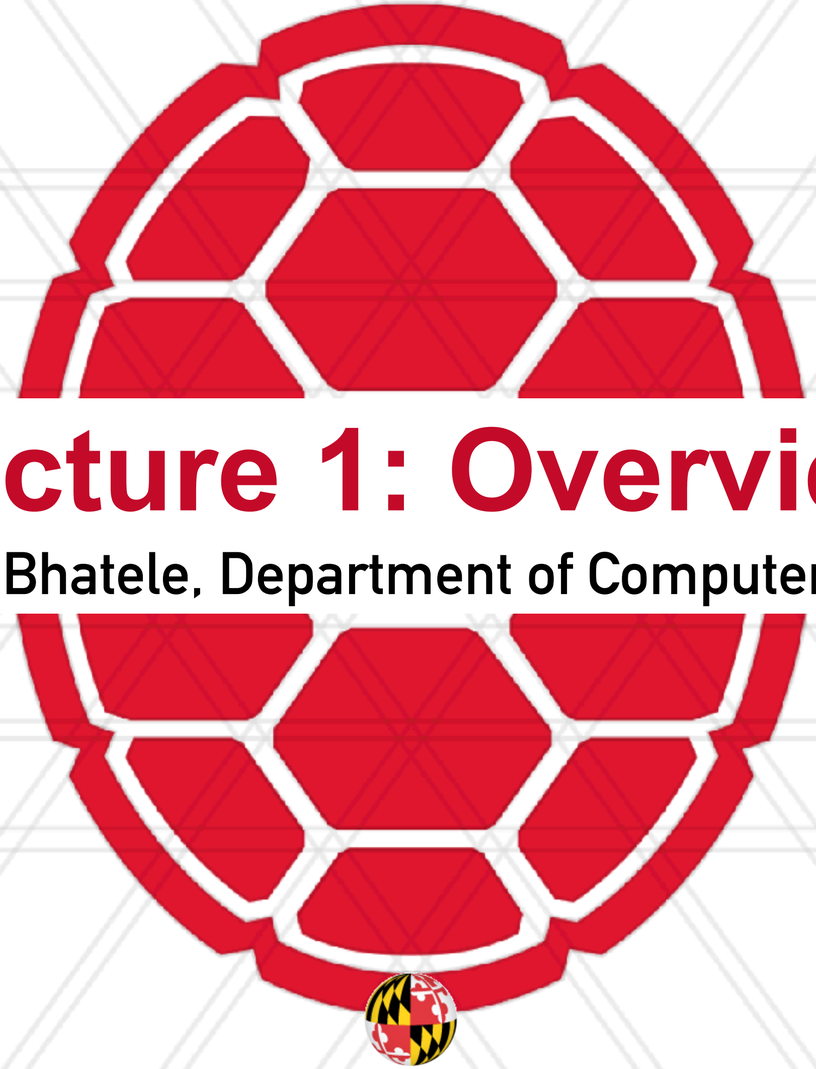
Introduction to Parallel Computing (CMSC498X / CMSC818X)



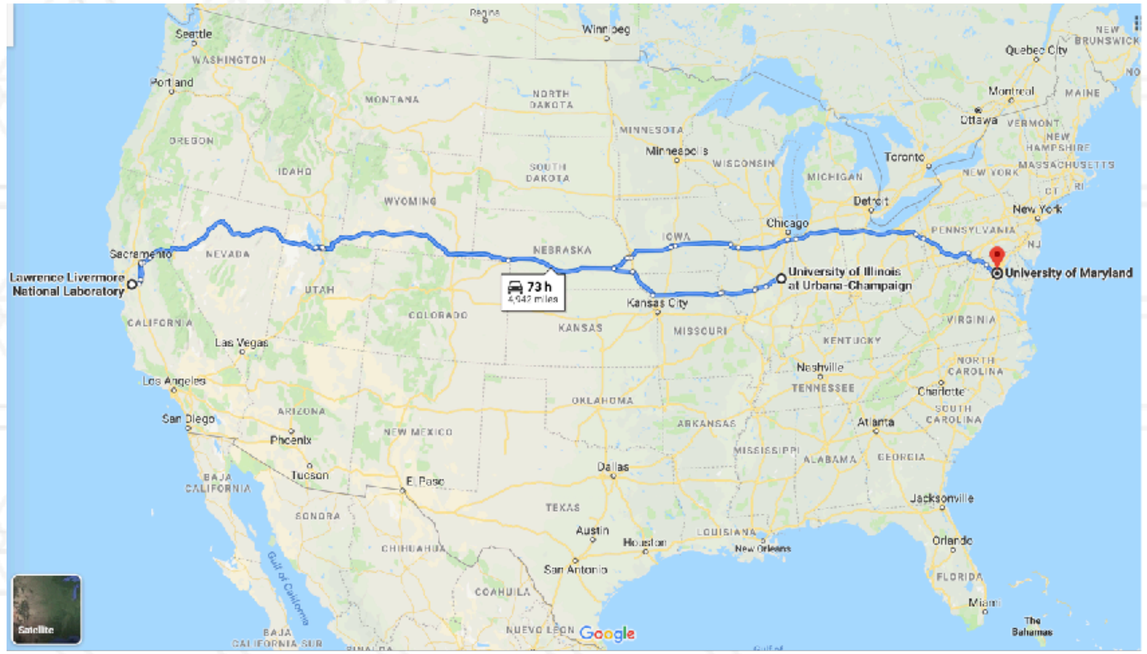
Lecture 1: Overview

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 last year



Introductions

- Name
- Undergraduate or graduate / Major
- Why this course?
- Something interesting/ unique about yourself
- Advice for online classes

This course is

- An introduction to parallel computing
- 498X: Upper Level CS Coursework / General Track / Area 1: Systems
- 818X: Qualifying course for MS/PhD: Computer Systems
- Work expected:
  - Three programming assignments
  - Four quizzes
  - Midterm exam: due on October 27
  - Final group project (3-4 students per group): due on December 14

Course topics

- Introduction to parallel computing (1 week)
- Distributed memory parallel programming (3 weeks)
- Shared memory parallel programming (1 week)
- Parallel algorithms (2 weeks)
- Performance analysis (2 weeks)
- Performance modeling and optimization (1 week)
- Load balancing (1 week)
- Network topologies and communication (1 week)
- Parallel simulation codes (2 weeks)

Tools we will use for the class

- Syllabus, lecture slides, assignment descriptions on course website:
  - http://www.cs.umd.edu/class/fall2020/cmsc498x
- Class on zoom (meeting details sent by email and on ELMS)
- Video recordings on Panopto
- Assignment submissions and quizzes on ELMS
- Midterm submission via Gradescope
- Discussions: Piazza or slack?

Deepthought2 accounts

- Shoken will email your login/password for deepthought2
- Helpful resources:
  - https://www.glue.umd.edu/hpcc/help/usage.html
  - https://hpcbootcamp.readthedocs.io

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.

What is parallel computing?

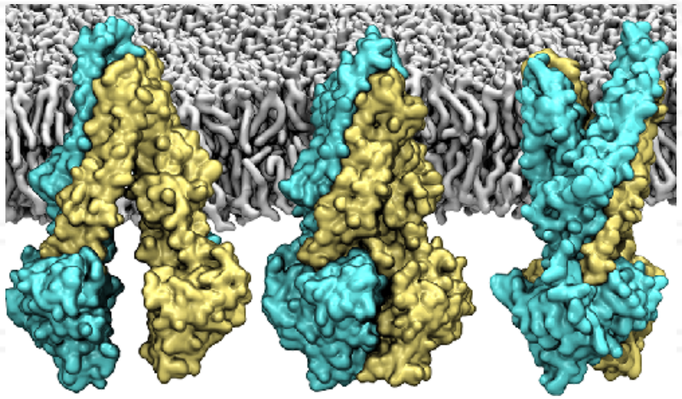
- Serial or sequential computing: doing a task in sequence on a single processor
- Parallel computing: breaking up a task into sub-tasks and doing them in parallel (concurrently) on a set of processors (often connected by a network)
- Some tasks do not need any communication: embarrassingly parallel

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 parallel computing or HPC

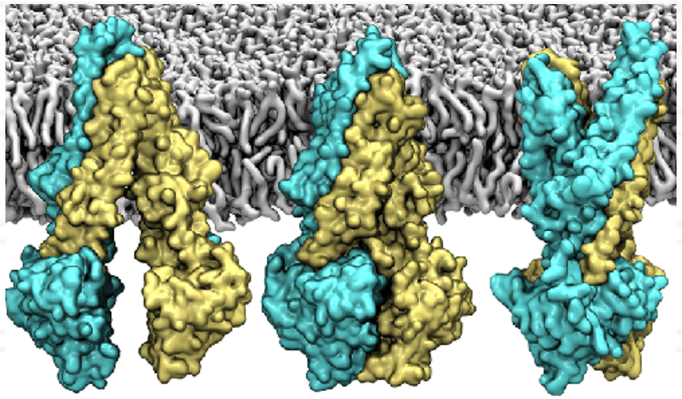
Drug discovery



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

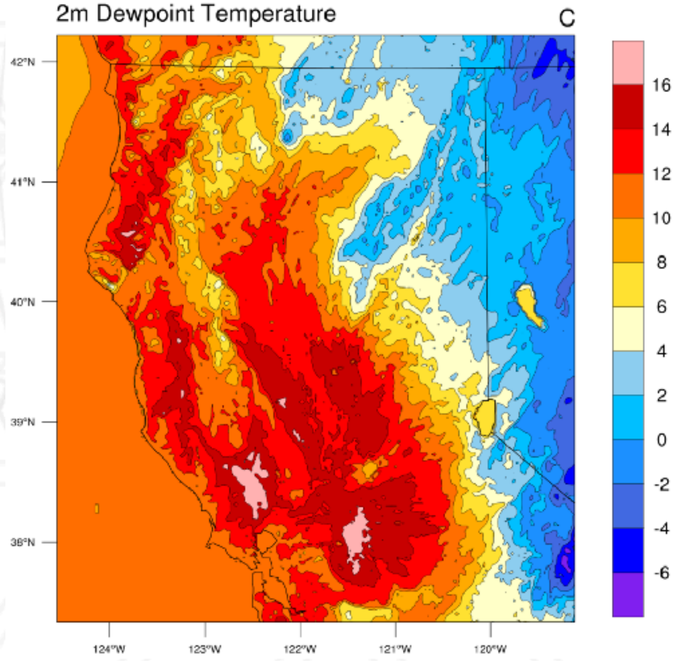
The need for parallel computing or HPC

Drug discovery



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

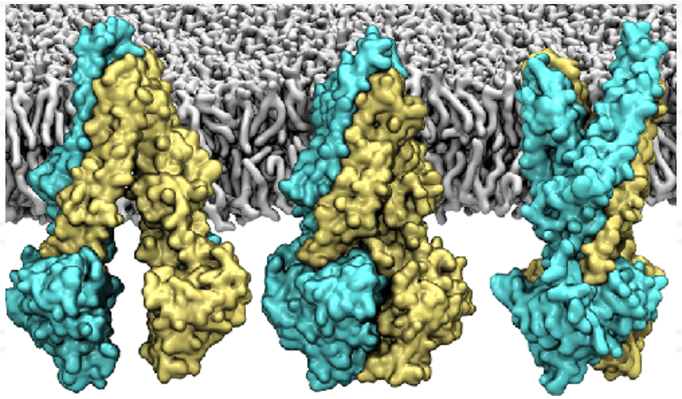
Weather forecasting



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

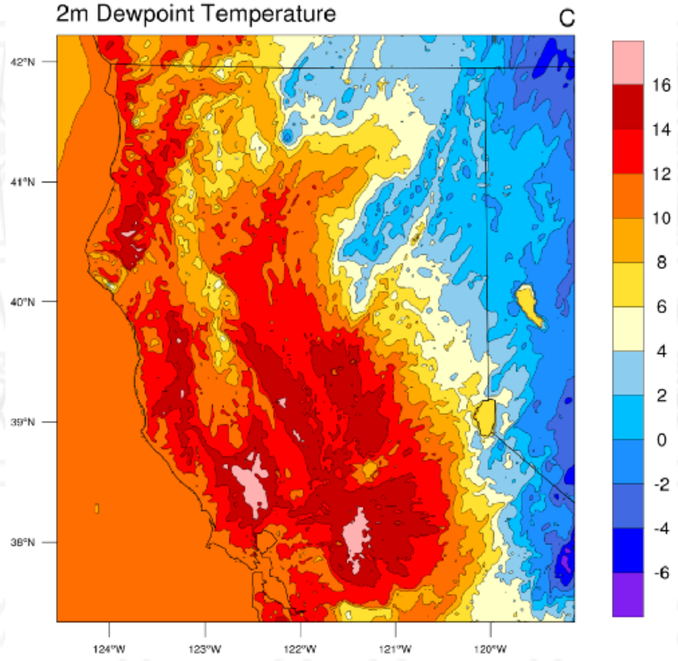
The need for parallel computing or HPC

Drug discovery



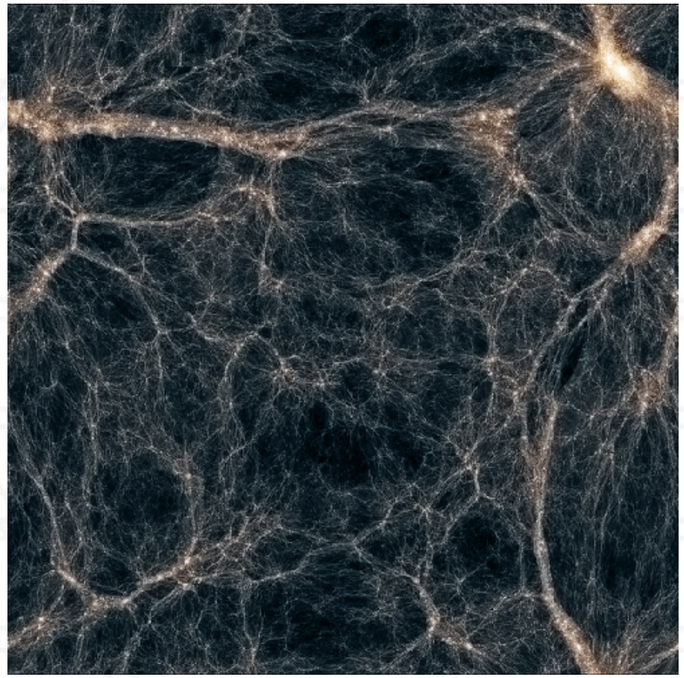
https://www.nature.com/articles/nature21414

Weather forecasting



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

Study of the universe



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

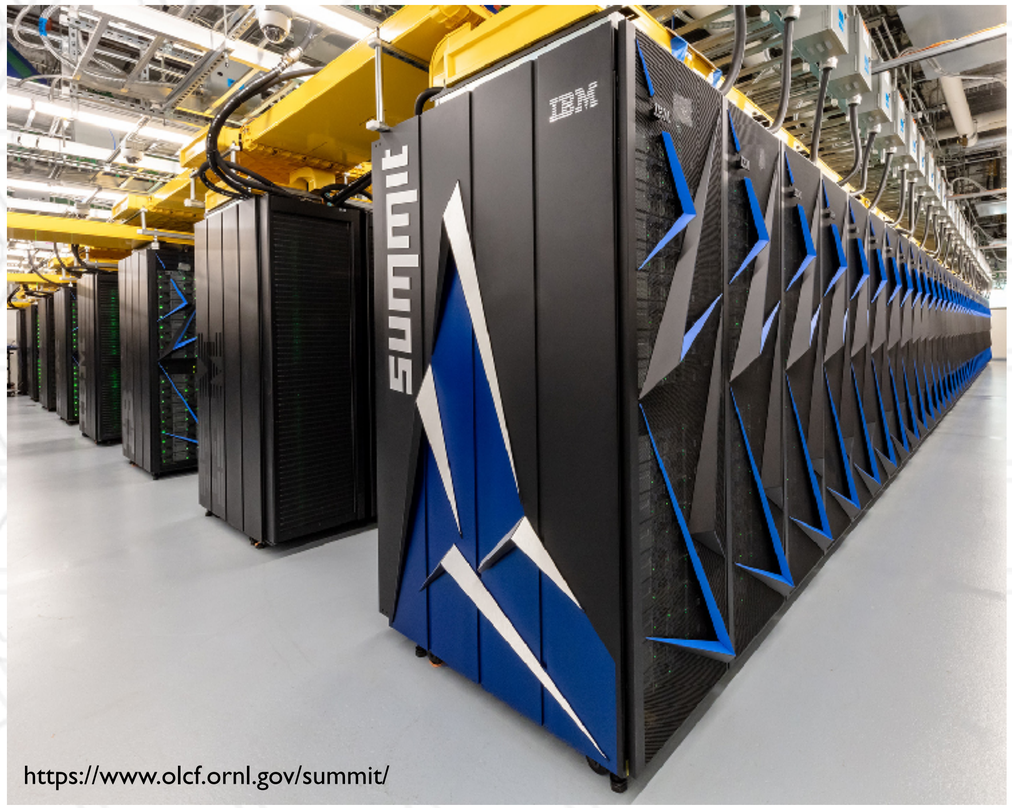
Why do we need parallelism?

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

Large supercomputers

- Top500 list: https://www.top500.org/lists/top500/2020/06/

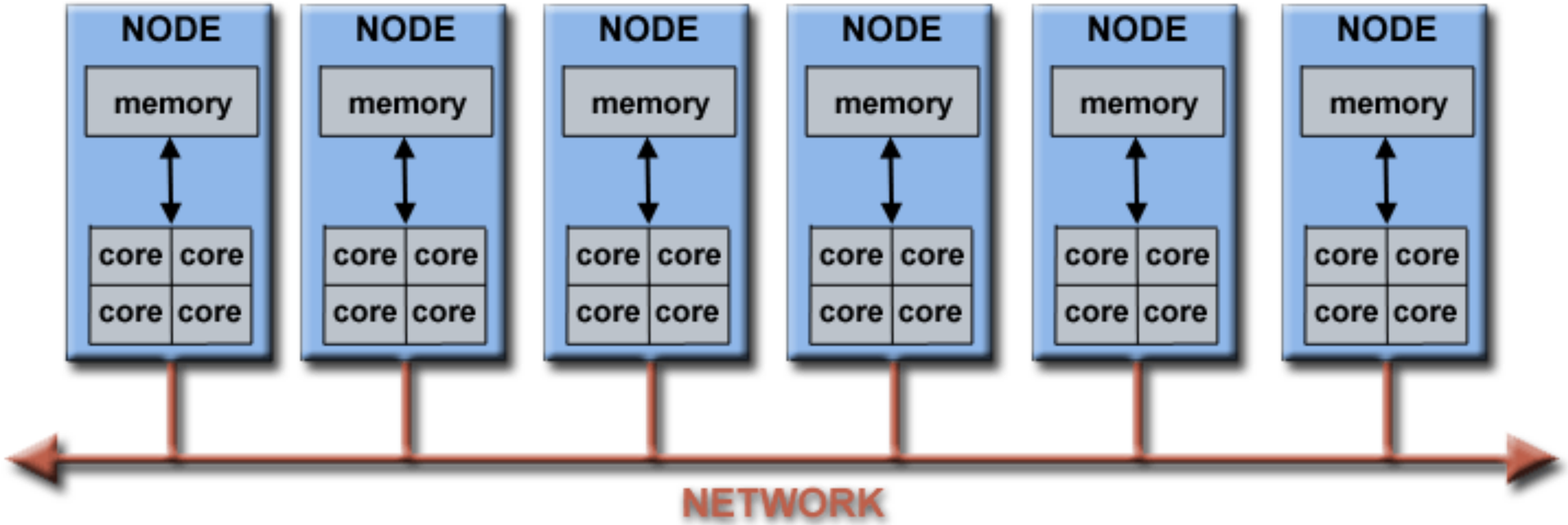
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,299,072	415,530.0	513,854.7	28,335
2	**Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM DOE/SC/Oak Ridge National Laboratory United States	2,414,592	148,600.0	200,794.9	10,096
3	**Sierra** - IBM Power System AC922, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband, IBM / NVIDIA / Mellanox DOE/NNSA/LLNL United States	1,572,480	94,640.0	125,712.0	7,438
4	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway, NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
5	**Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000, NUDT National Super Computer Center in Guangzhou China	4,981,760	61,444.5	100,678.7	18,482



https://www.olcf.ornl.gov/summit/

Parallel architecture

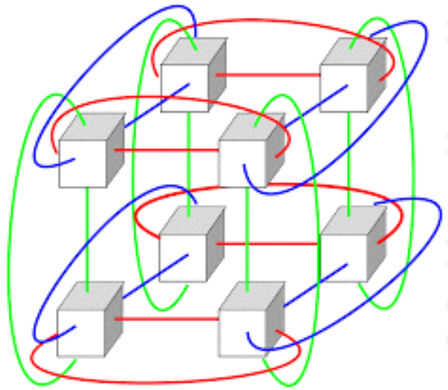
- A set of nodes or processing elements connected by a network.



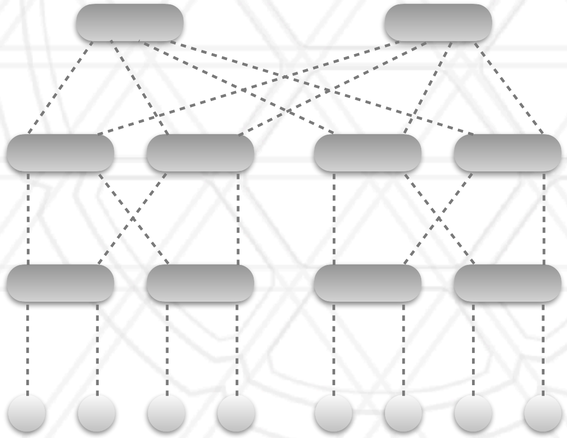
https://computing.llnl.gov/tutorials/parallel_comp

Interconnection networks

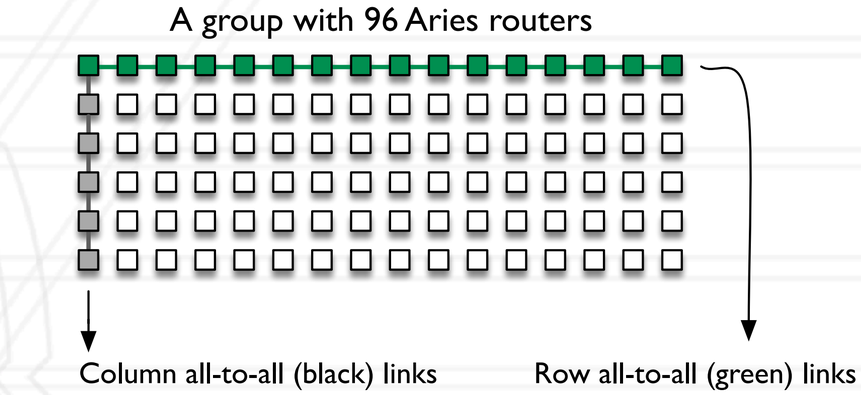
- Different topologies for connecting nodes together
- Used in the past: torus, hypercube
- More popular currently: fat-tree, dragonfly

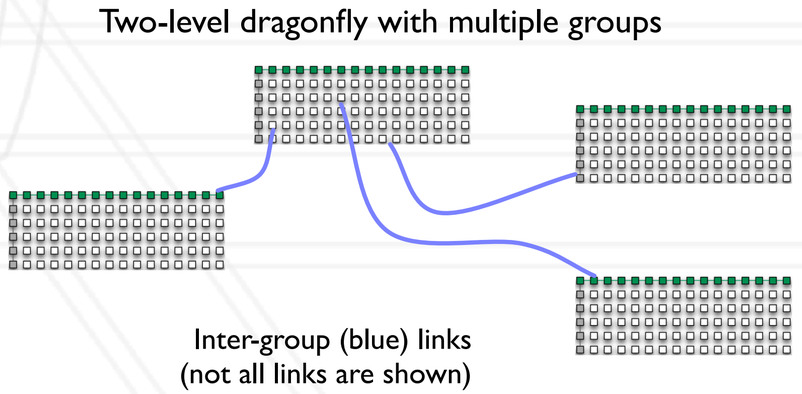


Torus



Fat-tree

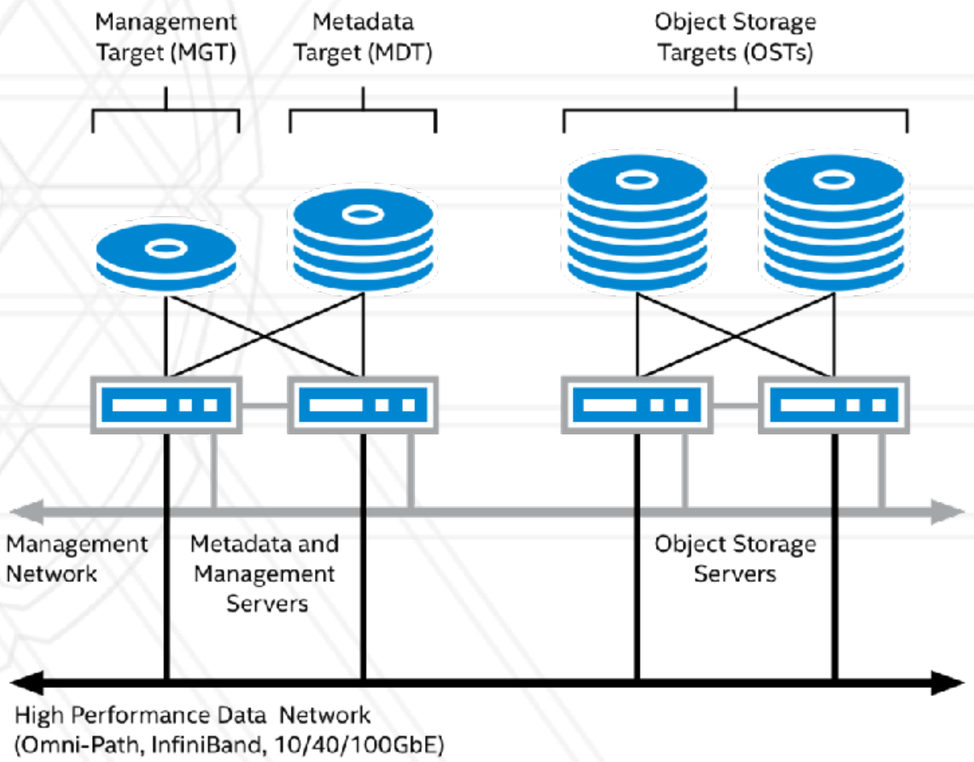




Dragonfly

I/O sub-system / Parallel file system

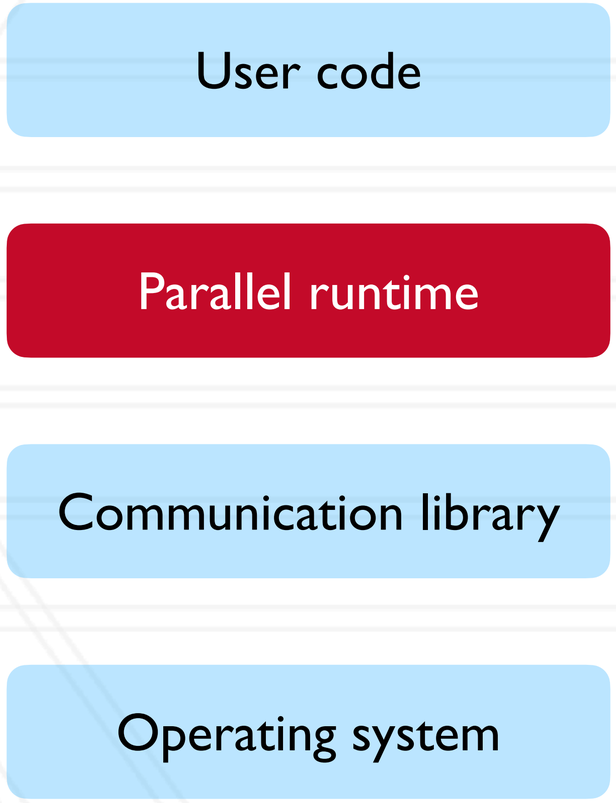
- Home directories and scratch space typically on a parallel file system
- Mounted on all login and compute nodes



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



UNIVERSITY OF
MARYLAND

Abhinav Bhatele

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

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