High Performance Computing Systems (CMSC714)

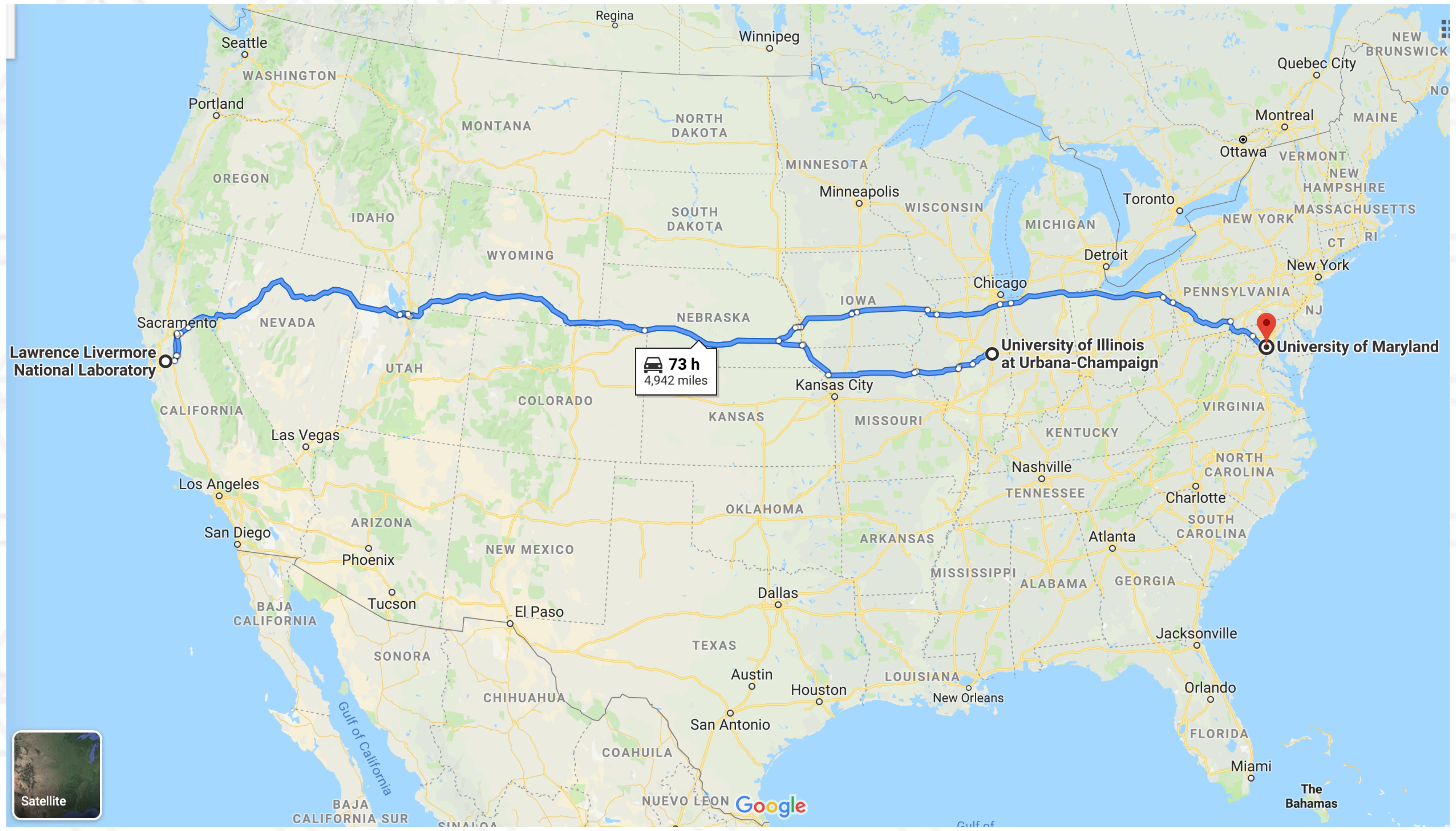
# Lecture 1: Introduction to Parallel Computing

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

# A little bit about me …

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



# Introductions

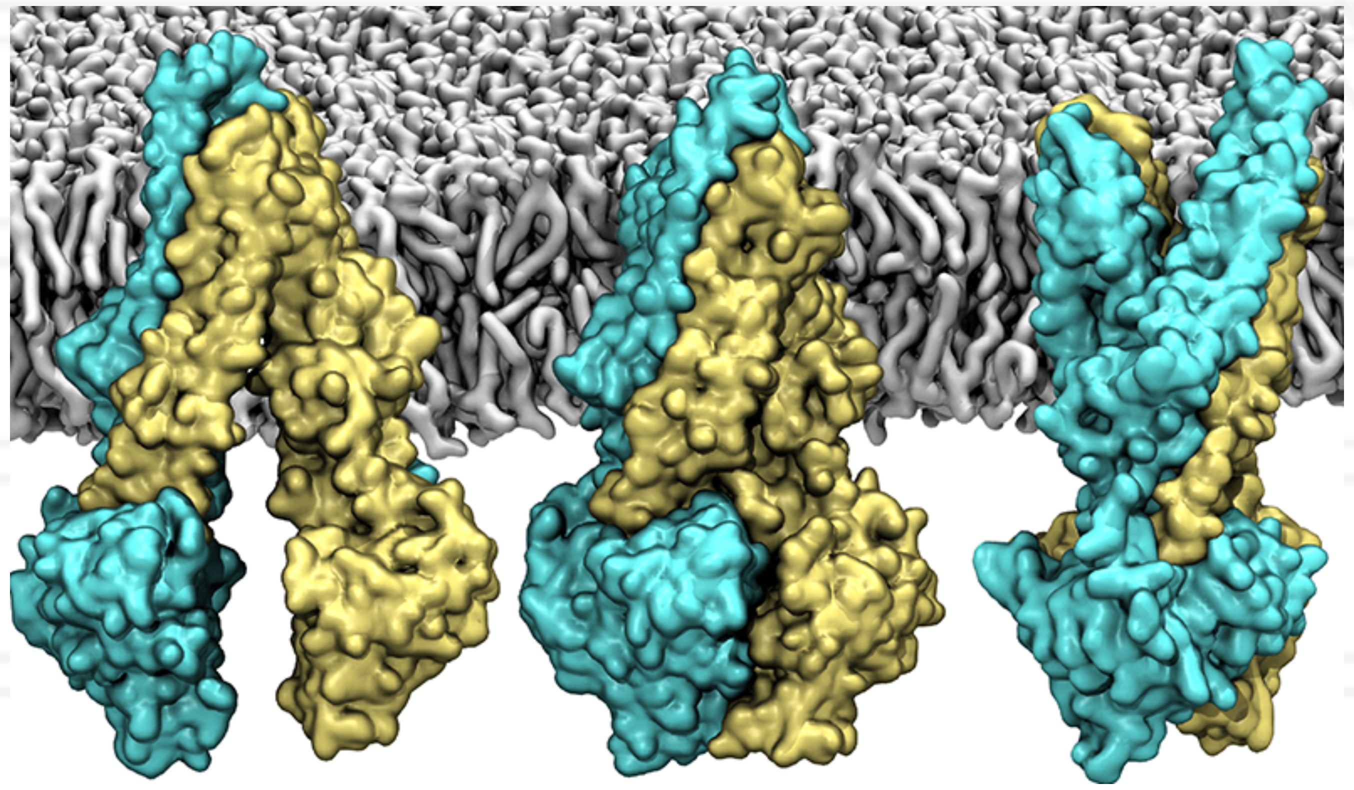
- Name
- MS or PhD / Department
- Area of research
- Why this course?
- Something interesting/ unique about yourself

# This course is

- An introduction to parallel computing
  - Will cover programming models, architectures, tools, systems issues, algorithms and applications
- A qualifying course for MS/PhD
- Work expected:
  - Two programming assignments
  - Mid-term
  - Group project (3 students per group)
  - Classroom participation

# The need for high performance computing

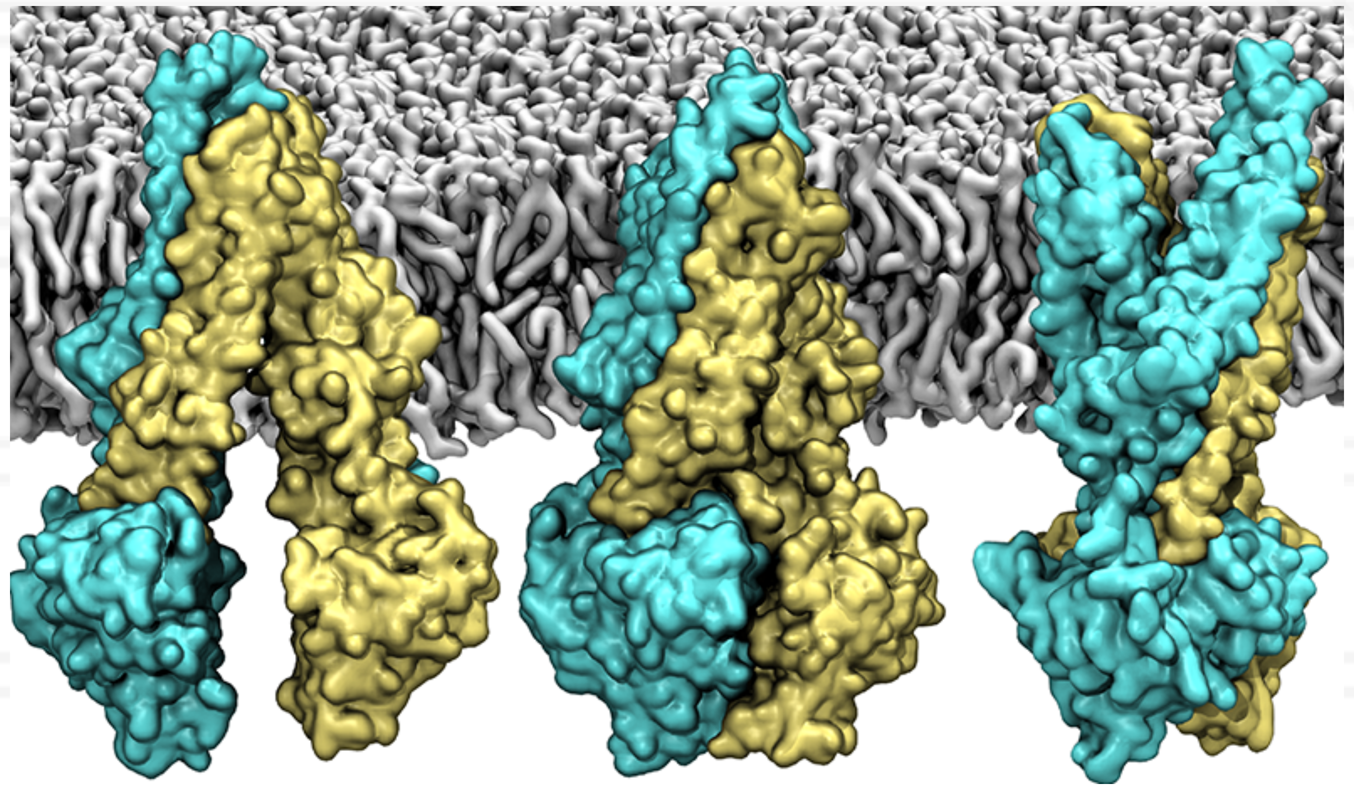
**Drug discovery**



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

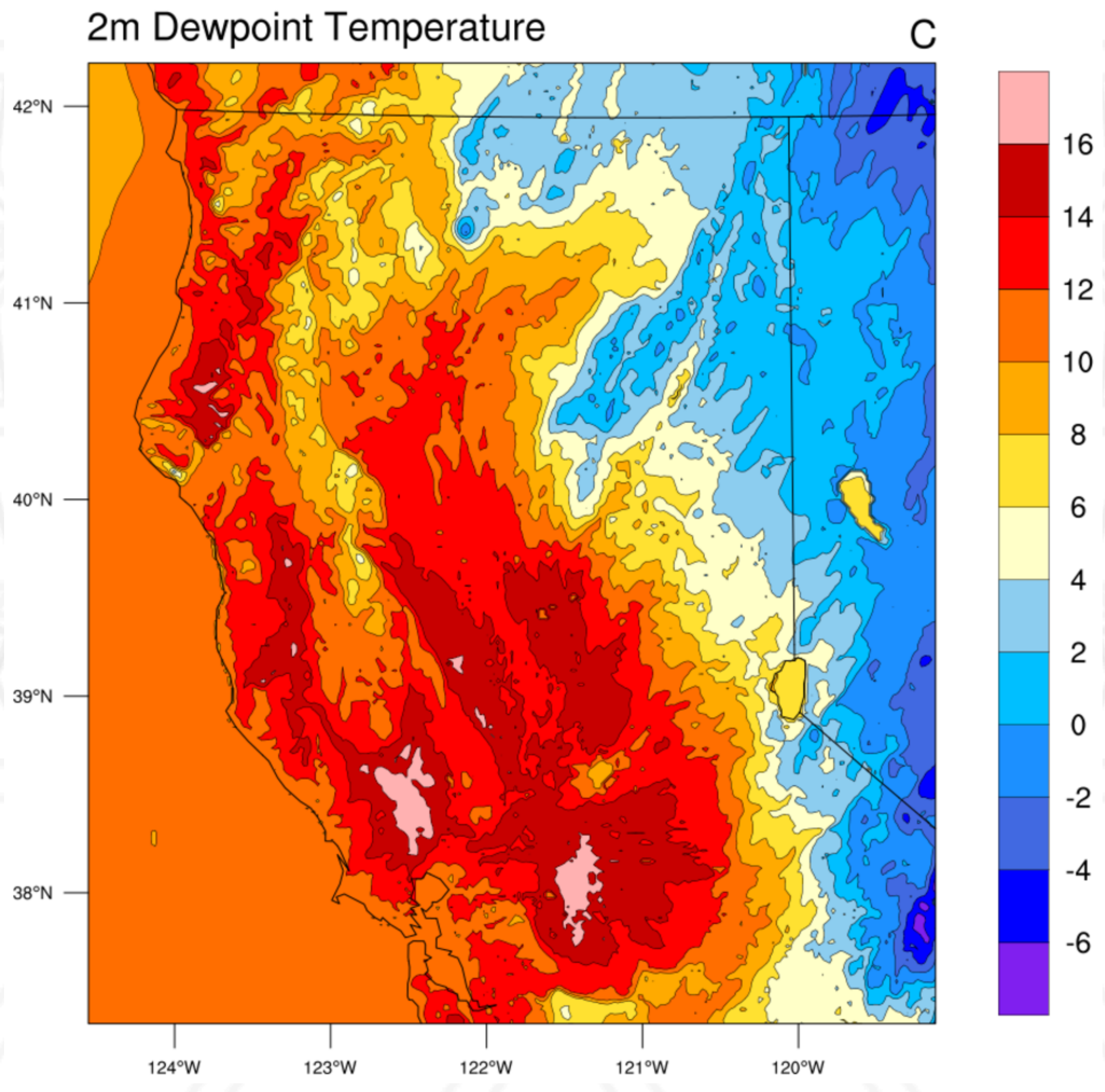
# The need for high performance computing

**Drug discovery**



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

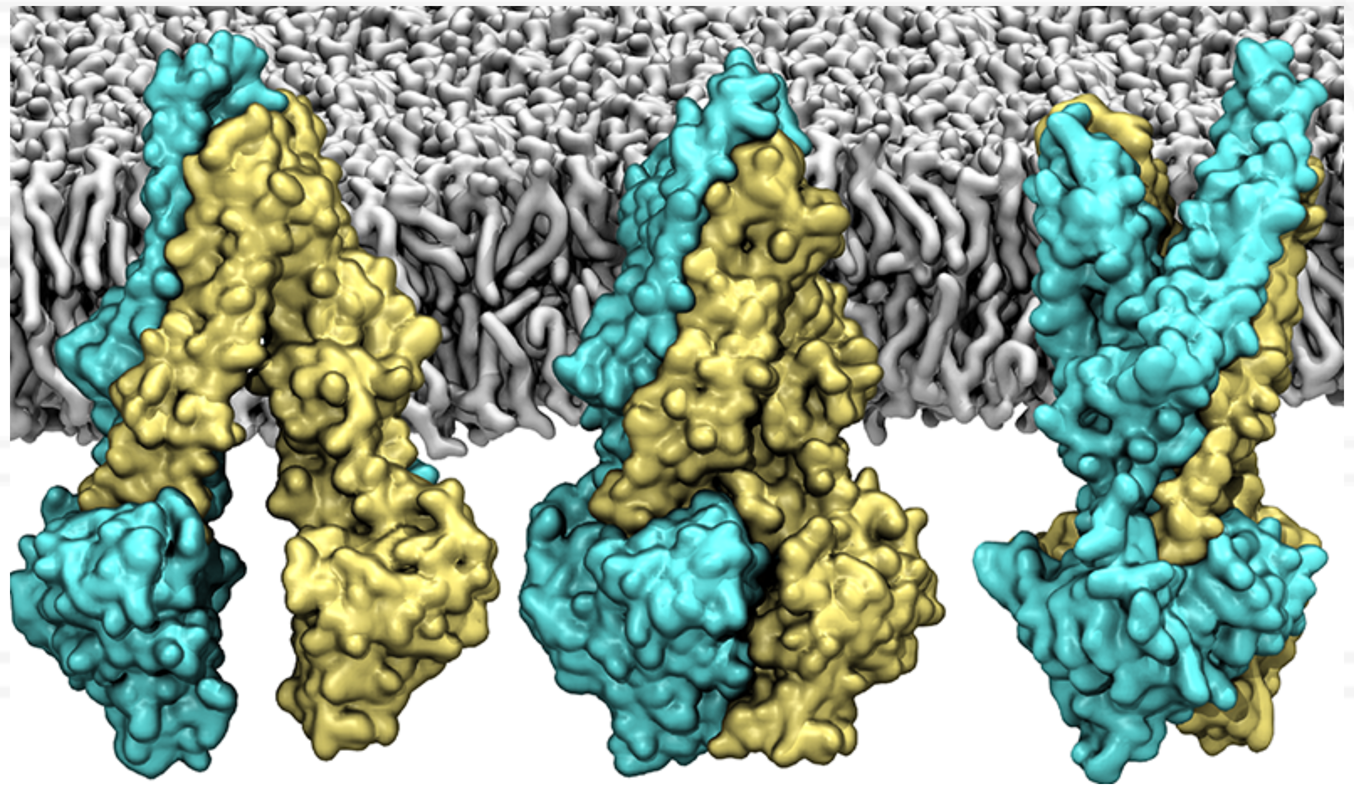
**Weather forecasting**



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

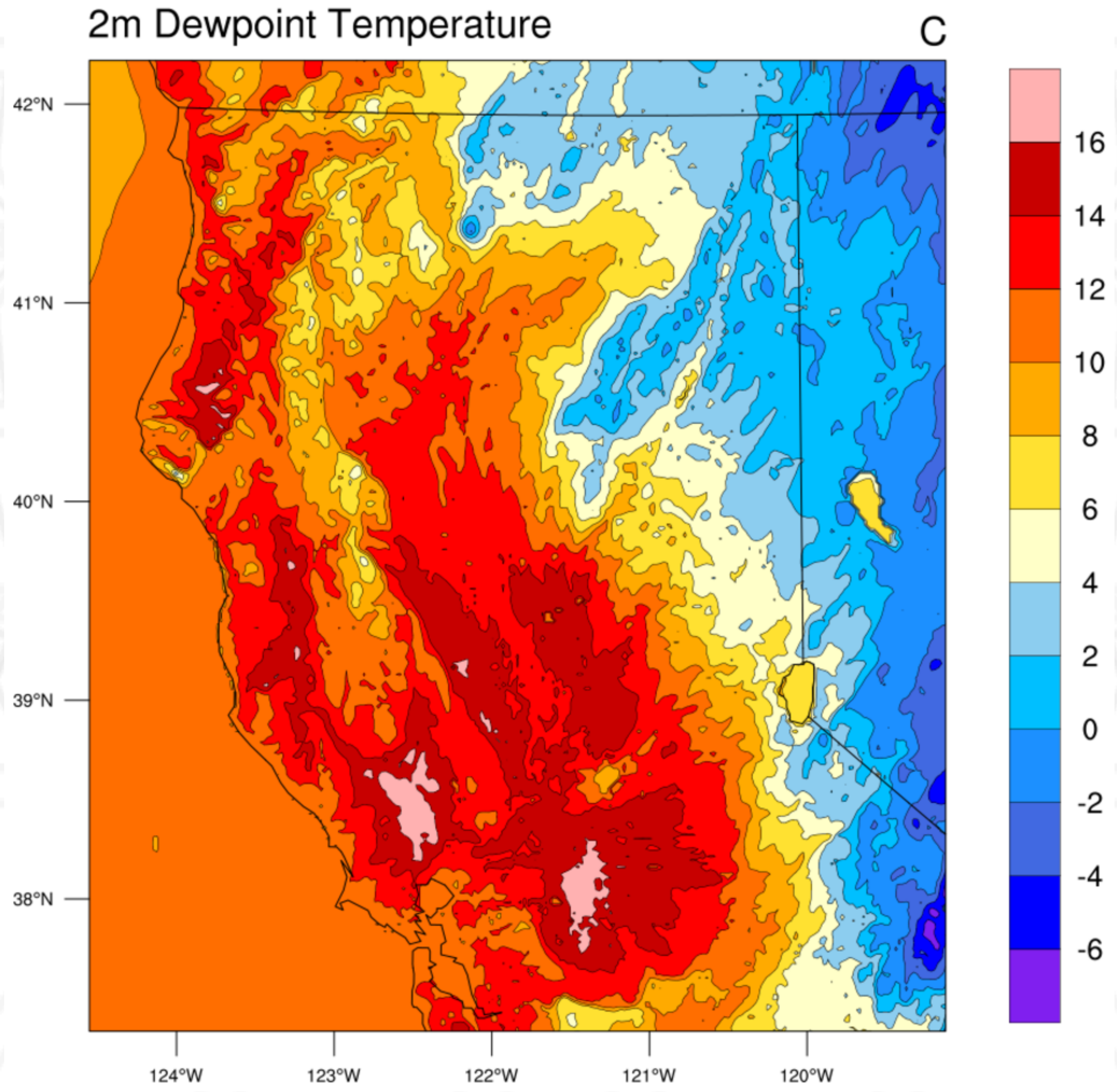
# The need for high performance computing

### Drug discovery



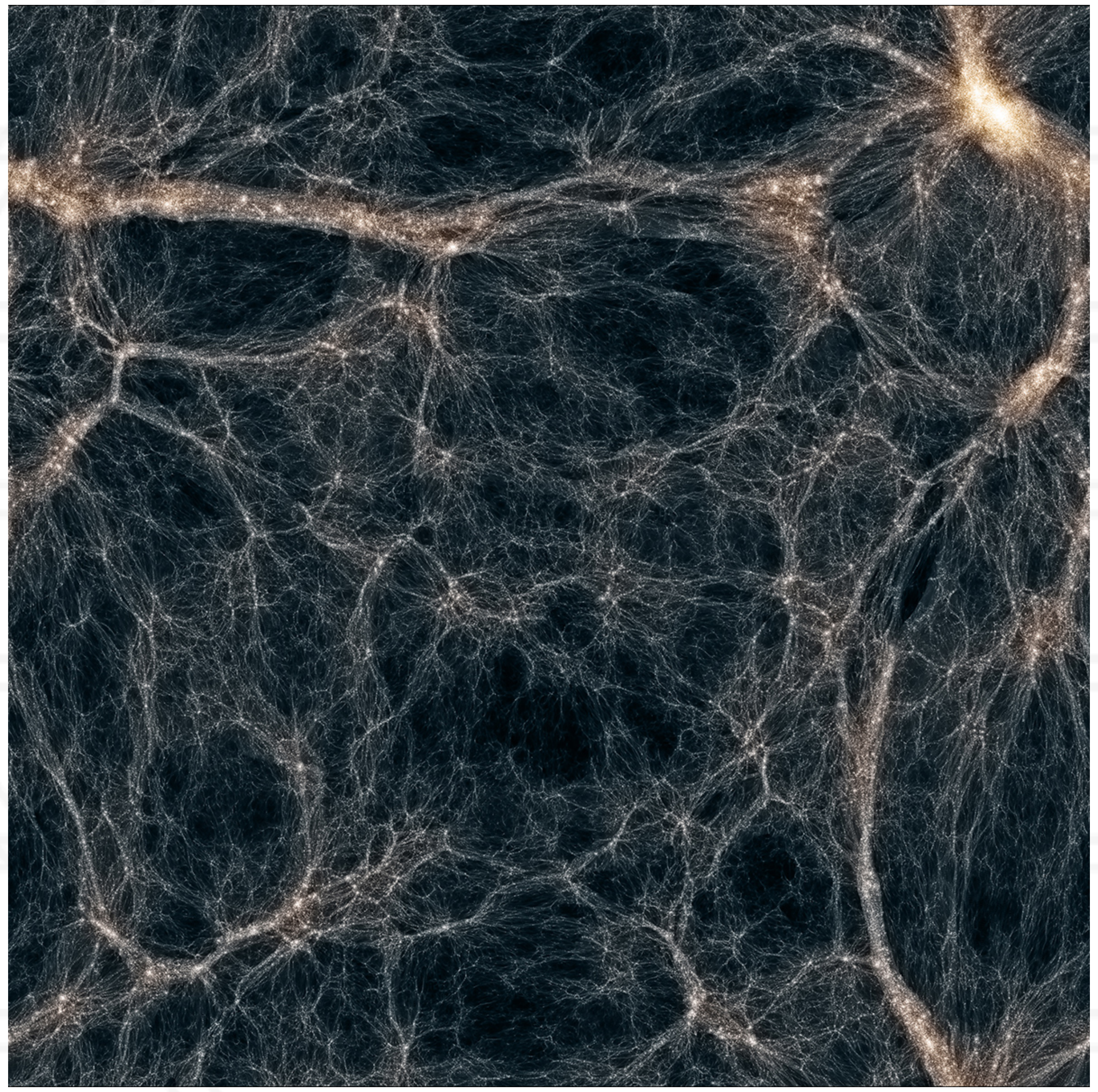
https://www.nature.com/articles/nature21414

### 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

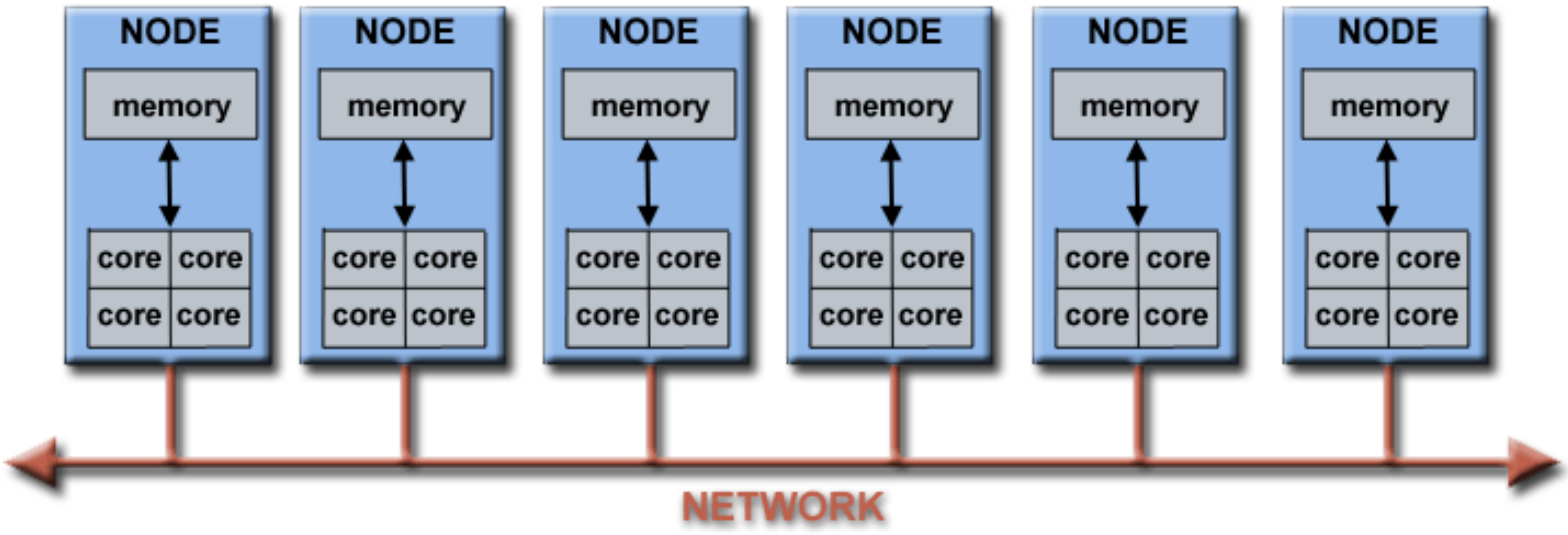
- Make some science simulations feasible in the lifetime of humans
  - Either due to speed or memory requirements
- Provide answers in realtime or near realtime

# What is parallel computing?

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

# Parallel Architecture

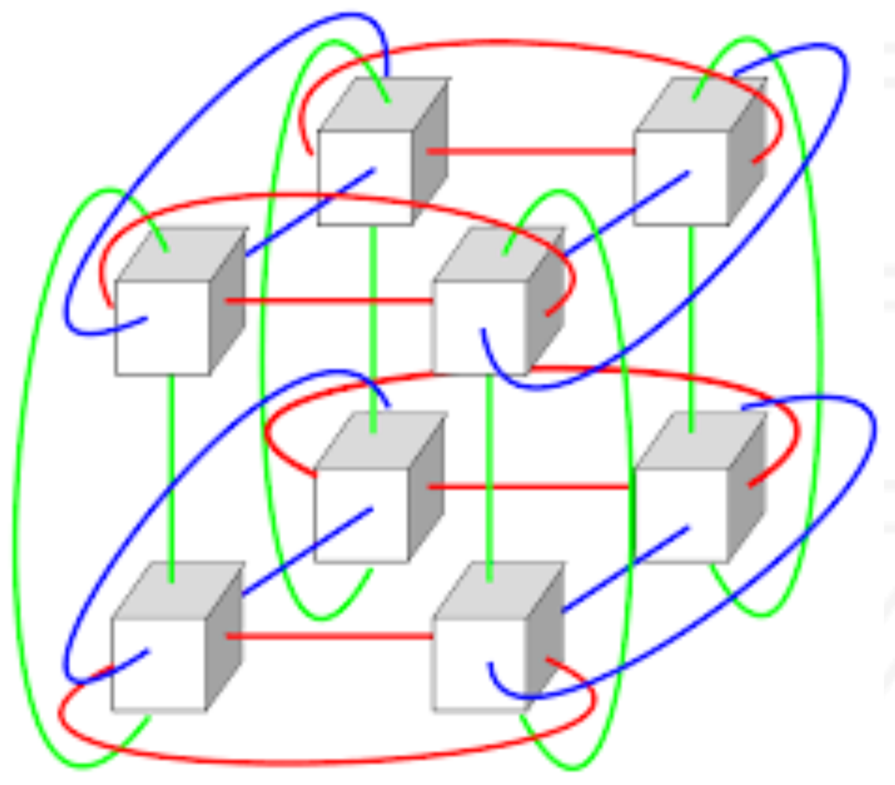
- A set of nodes or processing elements connected by a network.



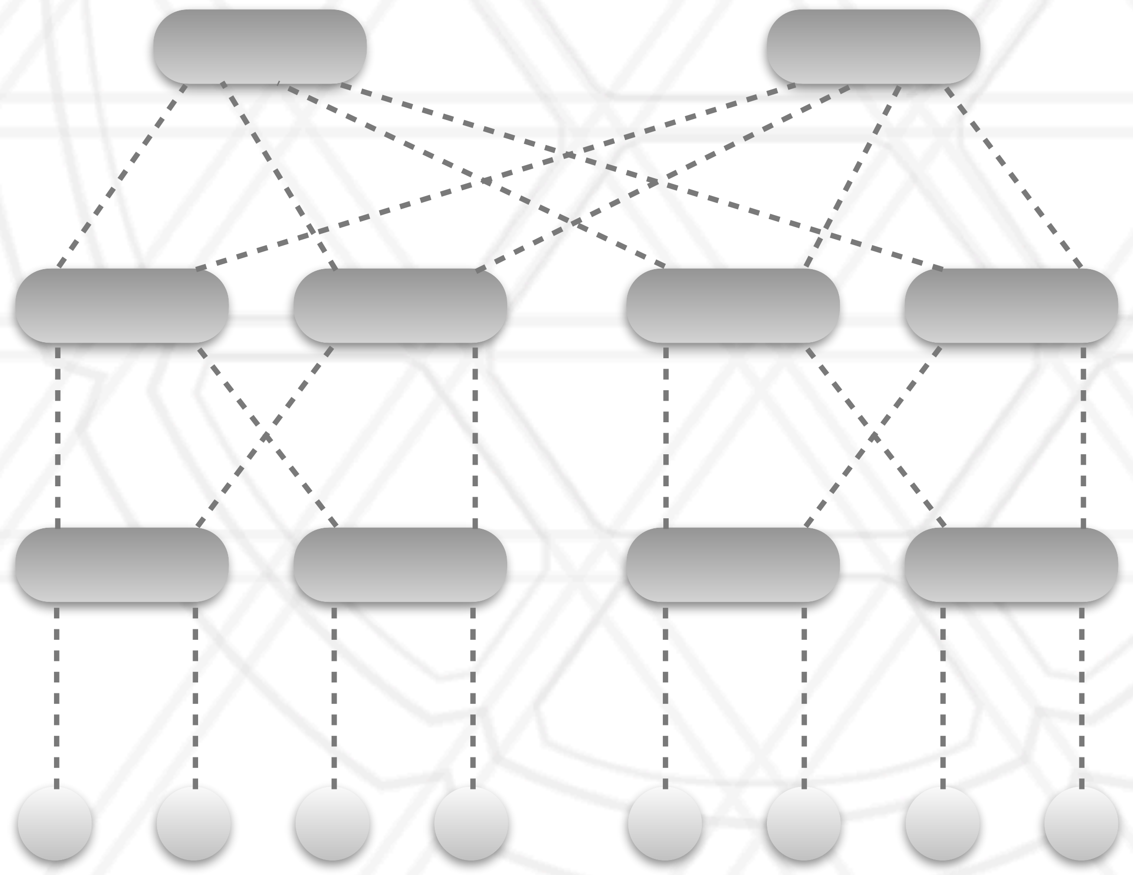
https://computing.llnl.gov/tutorials/parallel_comp

# Interconnection networks

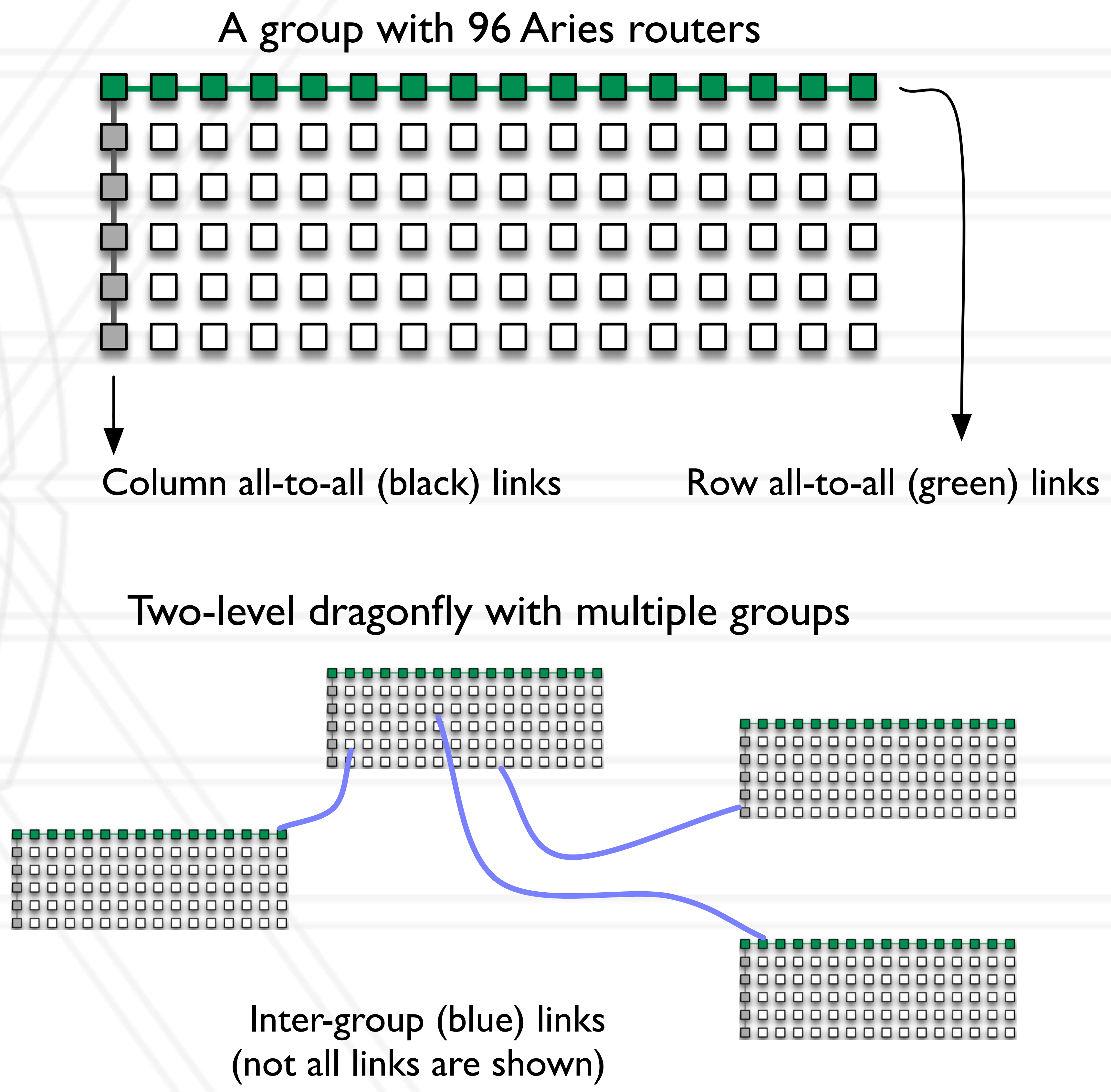
- Different topologies for connecting nodes together
- Used in the past: torus, hypercube
- More popular currently: fat-tree, dragonfly



Torus



Fat-tree



A group with 96 Aries routers
Column all-to-all (black) links
Row all-to-all (green) links
Two-level dragonfly with multiple groups
Inter-group (blue) links
(not all links are shown)

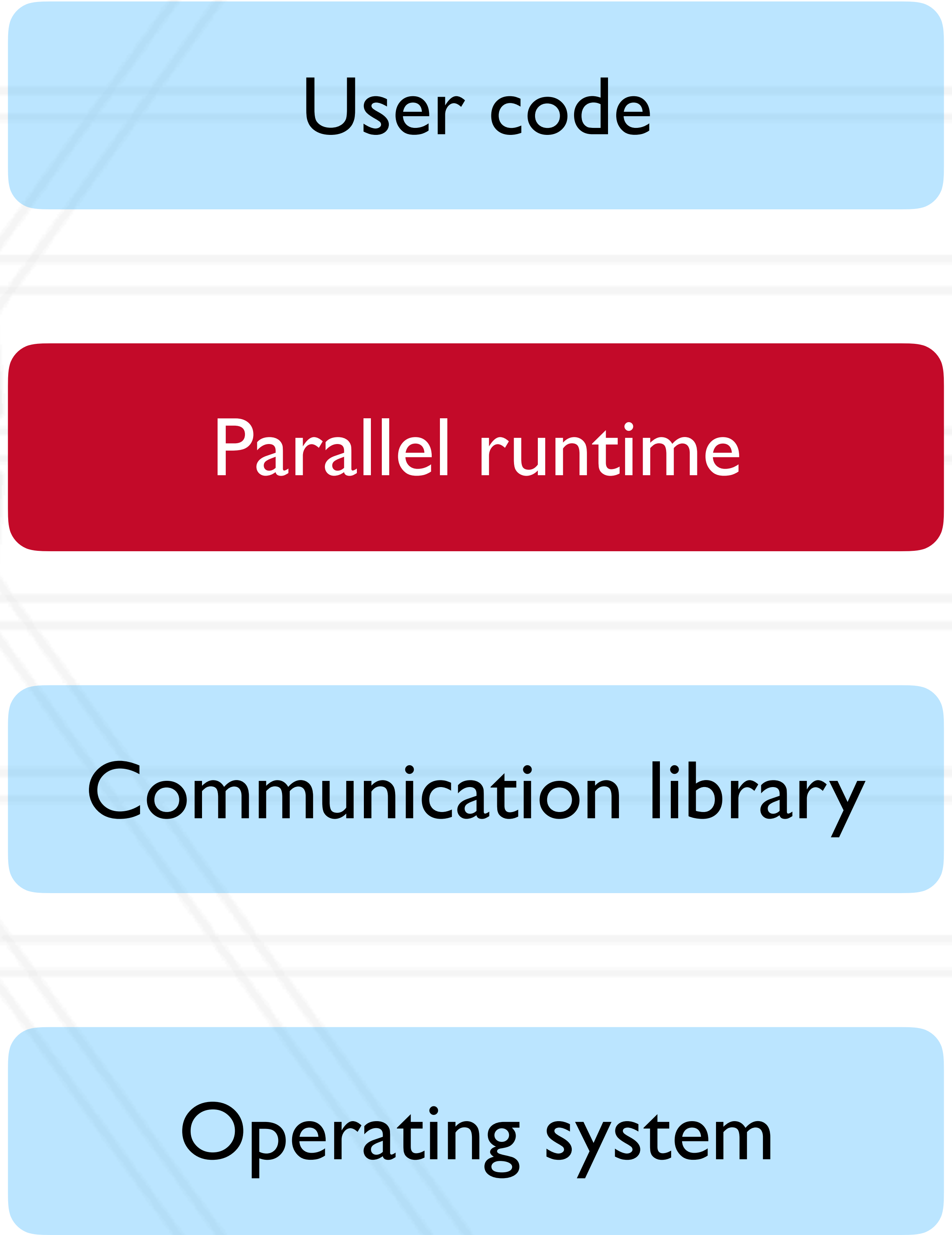
Dragonfly

# Memory and I/O sub-systems

- Similar issues for both memory and disks (storage):
  - Where is it located?
  - View to the programmer vs. reality
- Performance considerations: latency vs. throughput

# System software: Programming models

- Shared memory/ address-space
  - Explicit: Pthreads
  - Implicit: OpenMP
- Distributed memory
  - Explicit: MPI
  - Implicit: Task-based models (Charm++)



User code

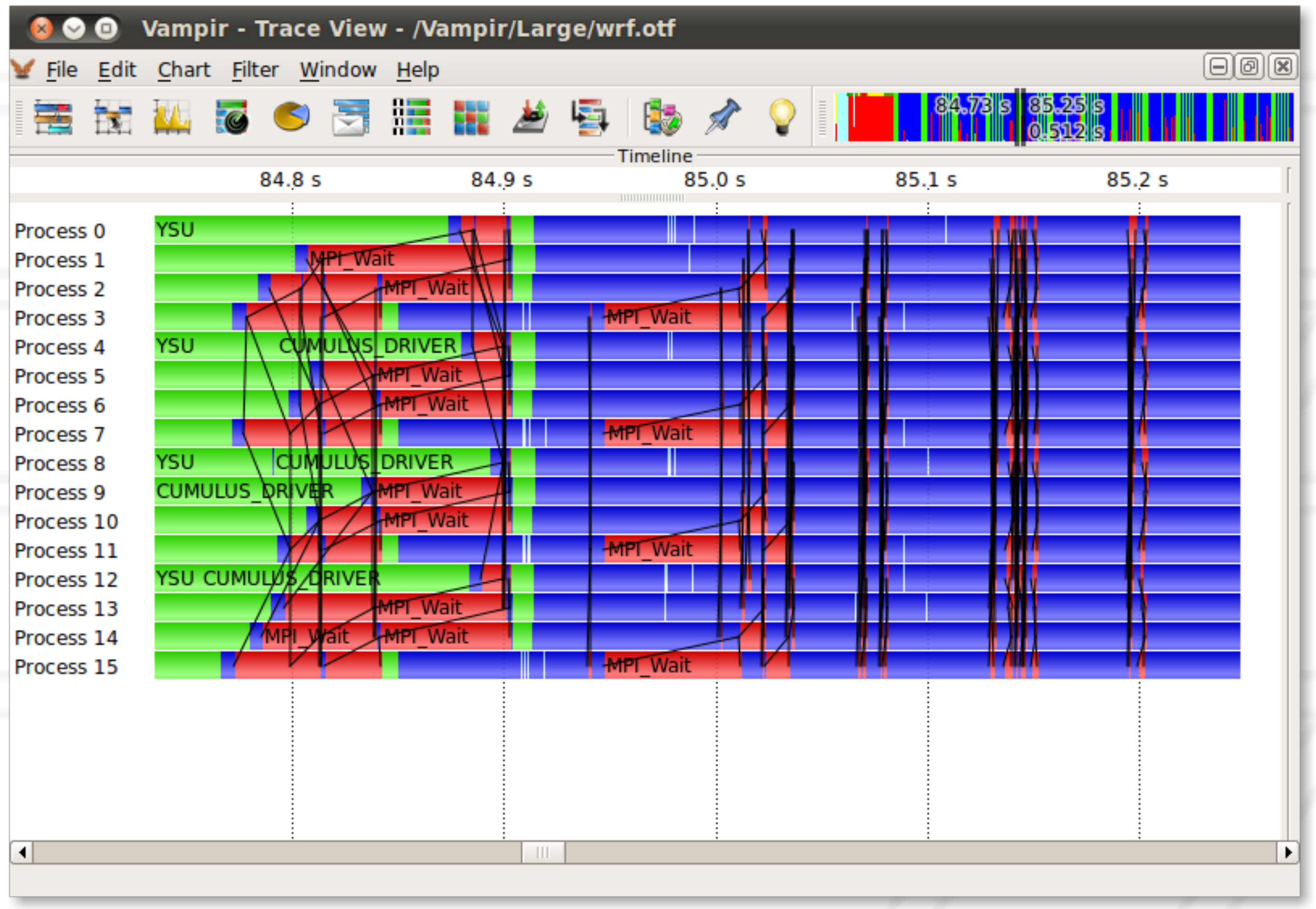
Parallel runtime

Communication library

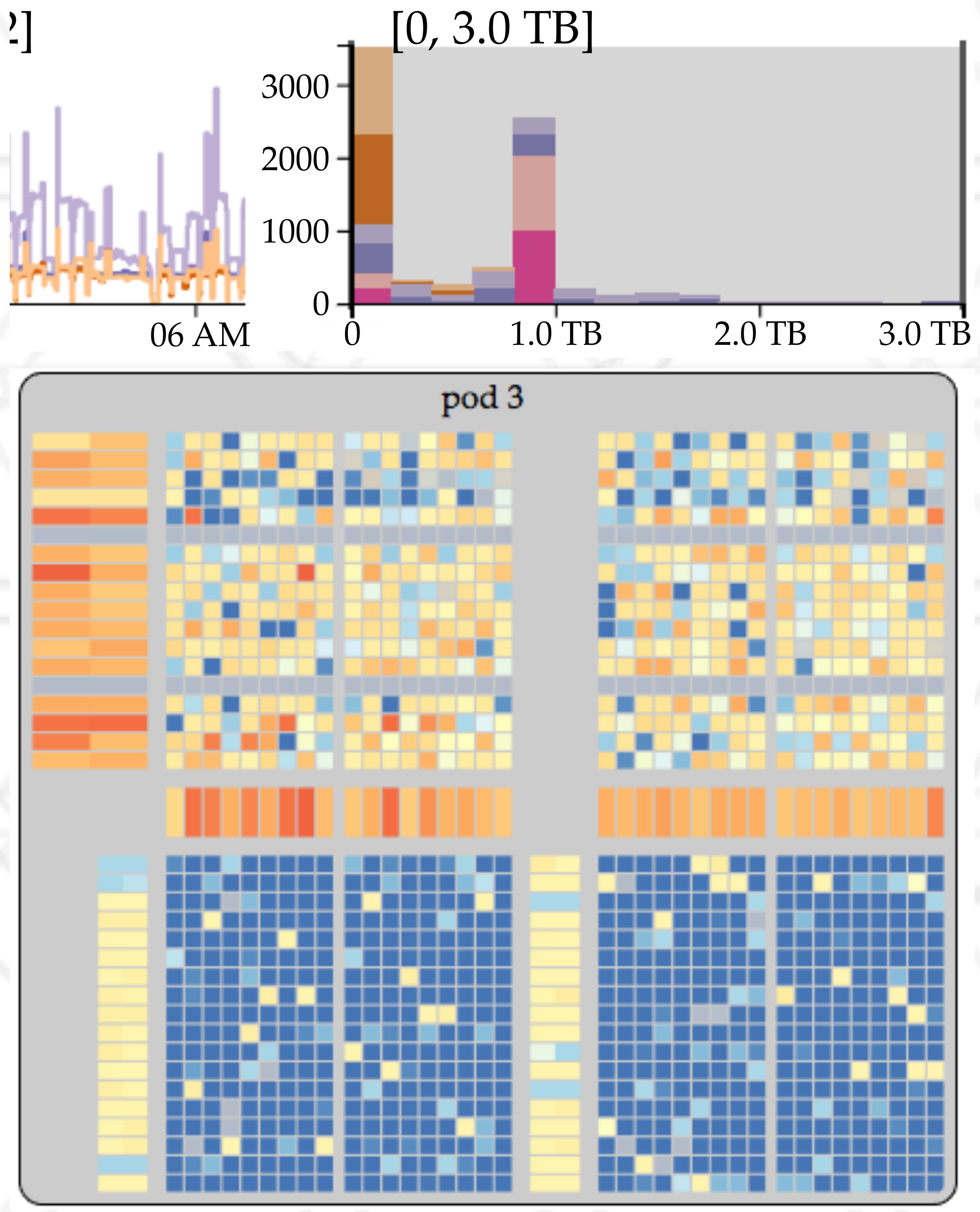
Operating system

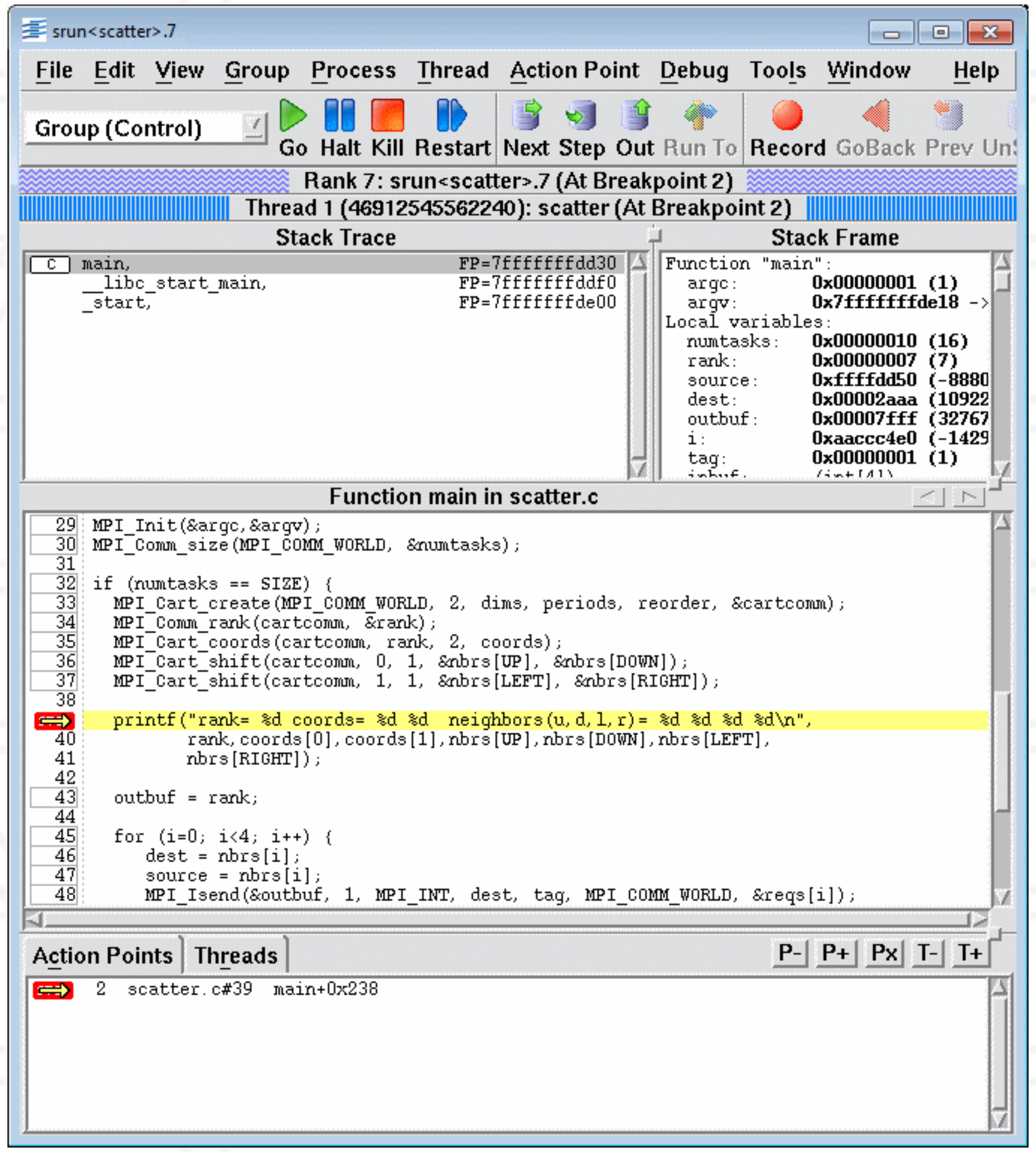
# Performance and debugging tools

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



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




srun<scatter>.7
File Edit View Group Process Thread Action Point Debug Tools Window Help
Group (Control)
Go Halt Kill Restart Next Step Out Run To Record GoBack Prev Un!
Stack Trace
Stack Frame
Function main in scatter.c
Action Points
Threads
P- P+ Px T- T+

Rank 7: srun<scatter>.7 (At Breakpoint 2)

Thread 1 (46912545562240): scatter (At Breakpoint 2)

```
C  main,                       FP=7fffffffdd30
   __libc_start_main,          FP=7fffffffddf0
   _start,                     FP=7fffffffde00
```

```
Function "main":
  argc:          0x00000001 (1)
  argv:          0x7fffffffde18 ->
Local variables:
  numtasks:      0x00000010 (16)
  rank:          0x00000007 (7)
  source:        0xffffdd50 (-8880
  dest:          0x00002aaa (10922
  outbuf:        0x00007fff (32767
  i:             0xaaccc4e0 (-1429
  tag:           0x00000001 (1)
```

```
MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD, &numtasks);

if (numtasks == SIZE) {
  MPI_Cart_create(MPI_COMM_WORLD, 2, dims, periods, reorder, &cartcomm);
  MPI_Comm_rank(cartcomm, &rank);
  MPI_Cart_coords(cartcomm, rank, 2, coords);
  MPI_Cart_shift(cartcomm, 0, 1, &nbrs[UP], &nbrs[DOWN]);
  MPI_Cart_shift(cartcomm, 1, 1, &nbrs[LEFT], &nbrs[RIGHT]);

  printf("rank= %d coords= %d %d  neighbors(u,d,l,r)= %d %d %d %d\n",
         rank,coords[0],coords[1],nbrs[UP],nbrs[DOWN],nbrs[LEFT],
         nbrs[RIGHT]);

  outbuf = rank;

  for (i=0; i<4; i++) {
     dest = nbrs[i];
     source = nbrs[i];
     MPI_Isend(&outbuf, 1, MPI_INT, dest, tag, MPI_COMM_WORLD, &reqs[i]);
```

```
2  scatter.c#39  main+0x238
```

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