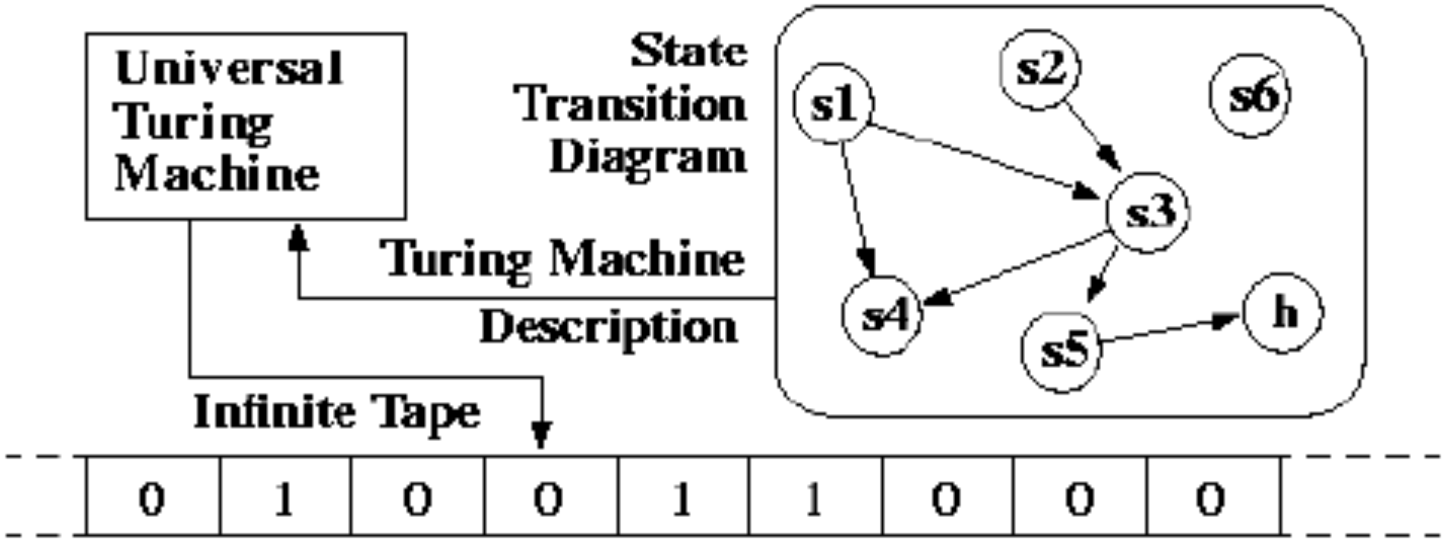
A brief history supercomputing

Lecture 1

The ideal computer



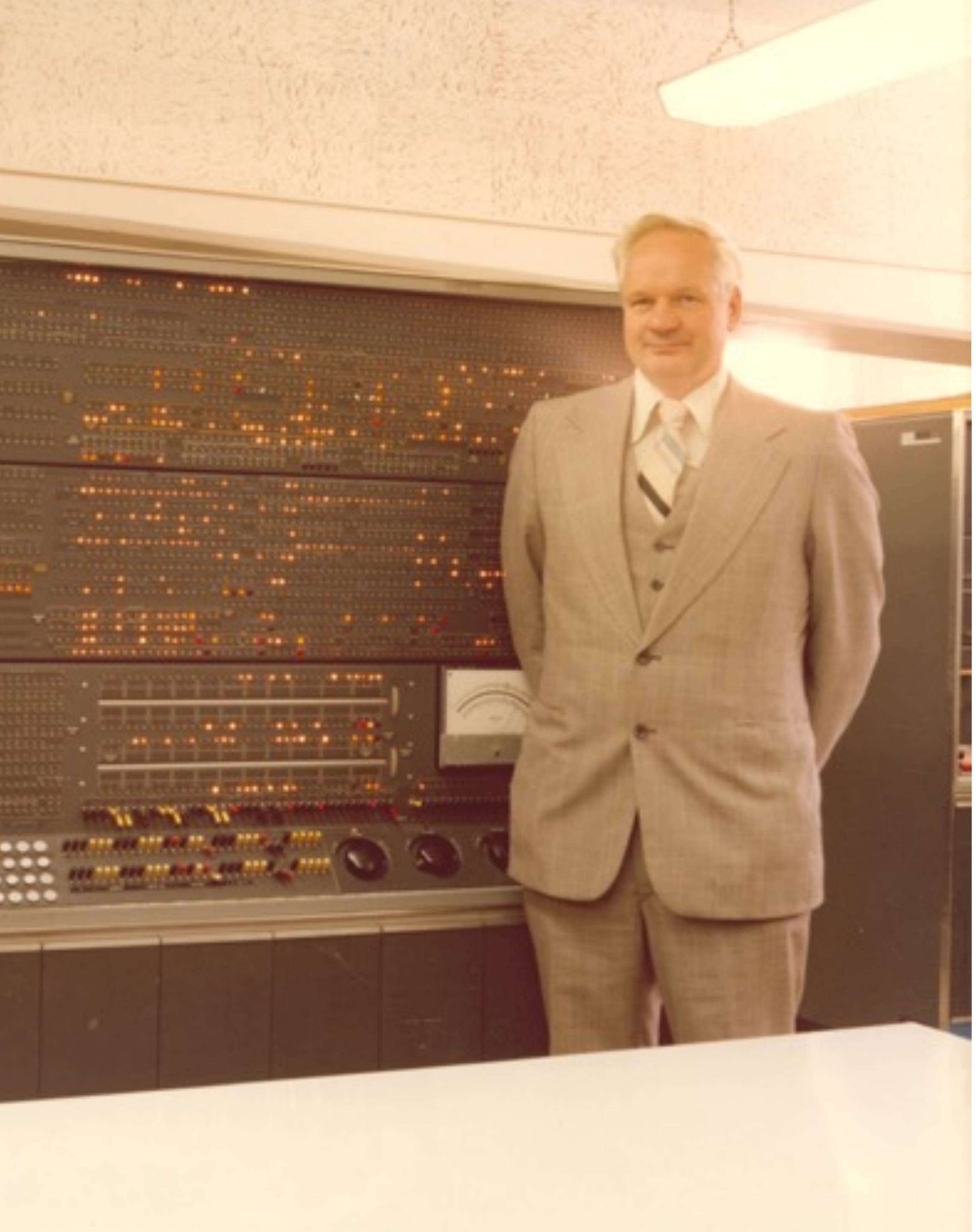
- Has an infinitely fast clock
- Executes one program instruction at a time
- Instructions and data are in an infinitely large storage device
- Access time to storage is always the same

This is the model for most runtime complexity analysis

Supercomputing vs Distributed Computing

- Supercomputers: These are large, expensive systems—usually housed in a single room—in which multiple processors are connected by a fast local network.

- Distributed Computing: These are systems in which processors are not necessarily located in close proximity to one another—and can even be housed on different continents—but which are connected via the Internet or other networks. (Think cloud computing)



Supercomputer

- Supercomputers are the highest performing computers of the day
- They are intended to solve "Grand Challenge" problems.
- In 1956 the IBM Stretch was built to be 100x faster than standard computers of the day. It cost $76,190,000 (today's dollars)
- Clock Speed: 1Mhz (10^6 cycles per second)
- **A clock cycle is when the CPUs transistors update. It's a fundamental limit on computational speed.**
- The definition of a supercomputer is always relative to all the desktops and laptops of the time.
- Ran at Los Alamos Labs simulating nuclear reactions

FORTRAN

- John Bakus at IBM directed the team that wrote FORTRAN (the formula translation language) in 1956.
- FORTRAN was the first widely used high-level language.
- Before FORTRAN programmers would write machine language programs directly.
- There were other high level languages at the time. But the machine language programs they produced were often 10x slower than human machine language.
- FORTRAN made sure their machine language was just as fast as machine code written by programmers.
- FORTRAN is intended for scientists (not programmers) to write fast code
- C on the other hand is intended for programmers not domain scientists

Backus J. The history of Fortran I, II, and III. ACM Sigplan Notices. 1978 Aug 1;13(8):165-80.

FORTRAN

- By 2010 software applications had been developed for nearly every scientific and engineering discipline.
- Some of the biggest drivers are:
  - Nuclear simulations (bombs)
  - Climate modelling (global warming)
  - Finance (automated stock trading)
  - Geology (oil)
  - Protein structure (drug discovery)
  - Molecular structure (designing new materials)
  - Artificial Intelligence (understanding you)
  - Computational Fluid Dynamics (aircraft, spakcecraft, cars, blast furnaces)
  - Bioinformatics (genetics)

Government Funding

- Supercomputing has been driven by government funding through the national labs and national supercomputing centers.
- Government agencies set the grand challenges that drive innovation in supercomputing
- National labs work closely with supercomputing vendors from the design stage.
- The US Advanced Research Project Agency (ARPA) was tasked with creating revolutionary new technologies after Sputnik (to win the technology cold war)
- ARPA established: 1) supercomputing centers, 2) computer science departments in universities, and 3) developed a national computer network.

ASCI Red (1996) – Sandia National Labs

First computer to pass 1 Teraflop

(10^{12} floating point operations/second) on the LINPACK benchmark.

(ASCI: Accelerated Strategic Computing Initiative)

Designed to simulate nuclear weapons after the Nuclear Test Ban Treaty (1992)

RAM: 1212 gigabytes

850kw

Industry

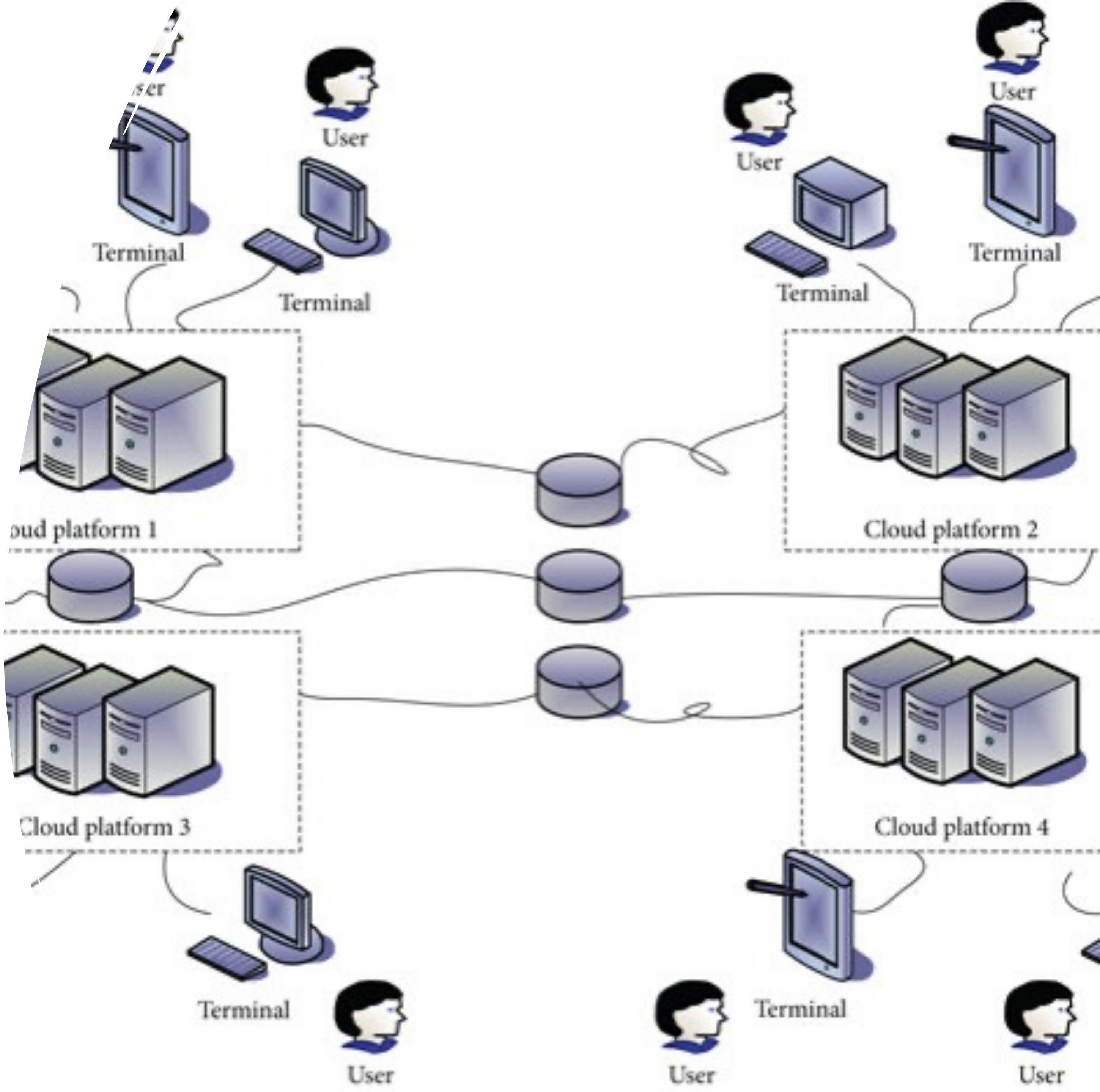
- The non-government sources of supercomputing demand are

Examples

1) Pharmaceuticals ()
2) Oil and Gas (computational fluid dynamics of oil given huge amounts of seismic data)
3) Computational Finance (high frequency trading) – 60-75% of trading

Distributed Computing Example

- When NASA's Jet Propulsion Laboratory needed to analyse images taken by the Mars Rovers, then the Amazon cloud was a good fit. Slow connections between monocomputers don't matter to much because the image processing is largely independent.
- Each computer processes one image or set of images.



Supercomputers

- Weather prediction is a classic example.
- To predict the weather even a few hours in the future requires processing enormous amounts of data gathered from millions of locations each recording temperature, wind, humidity, solar radiation, etc.
- The computation has to be spread over many computers but very fast communications are needed to they can access the data they need.
- Weather models run continually so they need new data all the time.

Evolution of Supercomputers

- Increasing processor speed (1960 - now)
- Processing vectors (1970s - now)
- Additional processors for single memory systems (monocomputer)
- Fast networks for distributed computation over many monocomputers (compute clusters or multicomputers) (1990s - now)
- GPUs (2000s - now)

- Supercomputing has evolved from a primarily hardware challenge (1960s-1990s – Cray Era) to creating software that runs on compute clusters (1990s-today – Multicomputer Era)
- (GPU accelerators recapitulate this pattern but over the past 10 years)

Discussion (5 mins)

Nuclear simulations involve modeling particles in regions of space and how they interact with their neighbours.

Are nuclear simulations better run on a distributed computing system or a supercomputer and why?

Four Eras. - Vector Operatons

- The first thirty years of supercomputing was dominated by companies like Cray Research (Seymore Cray), Control Data Corp (CDC) and International Business Machines (IBM) (American companies).
- This included the beginning of the Vector operation “era”
- Vector operations increase CPU speed by applying the same instruction to more than one variable during the clock cycle.

CPU Vector Operations (Advanced Vector Extensions [AVX] and Single Instruction Multiple Data Streaming Extentions [SSE])

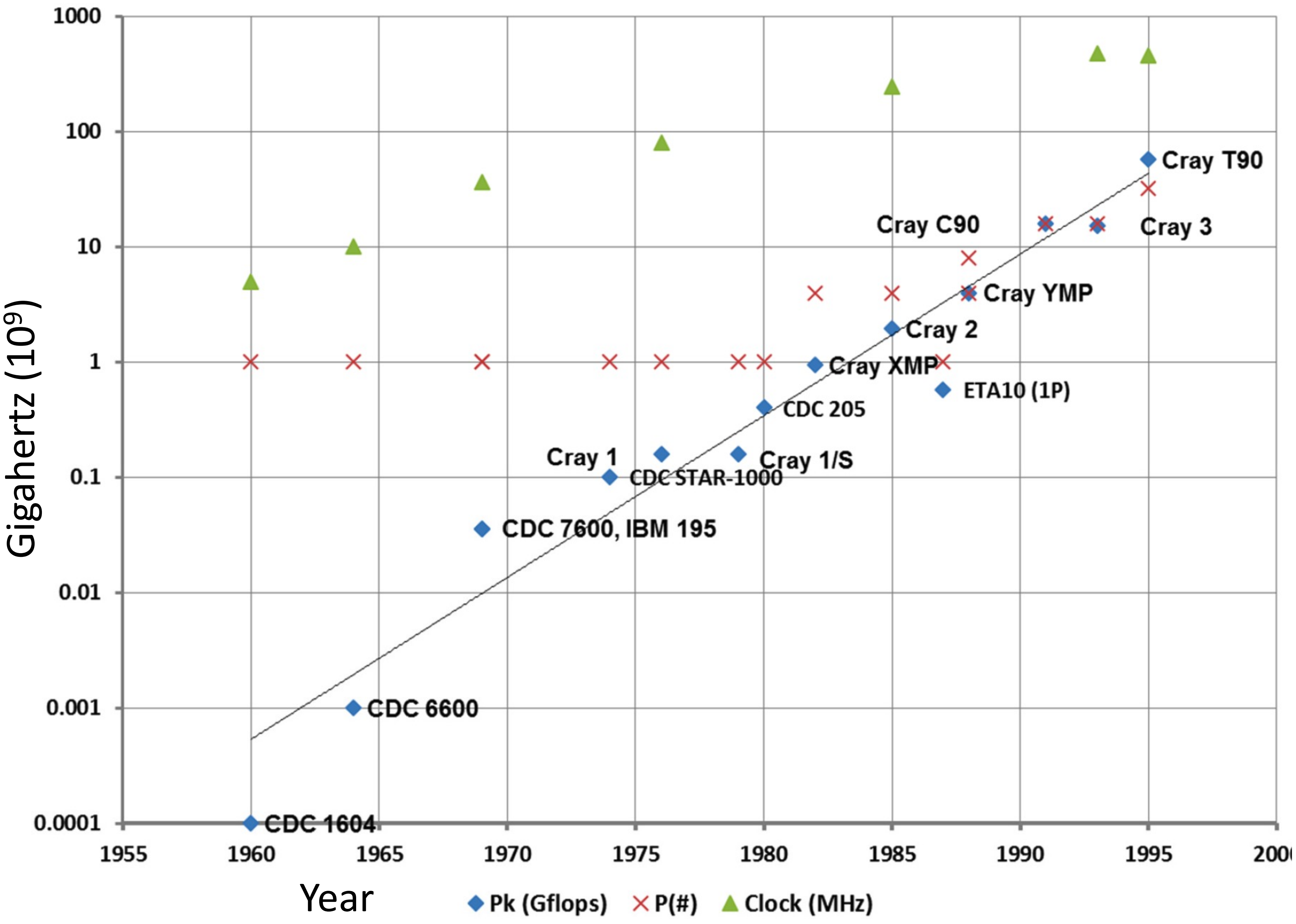
```
ssh vanilla@hopper.alliance.unm.edu
lscpu
lscpu | grep avx

ssh xena
lscpu   | grep avx

ssh wheeler
lscpu | grep avx
lscpu | grep sse
```

Cray, CDC, and IBM Supercomputer clock Speeds

1995 was the last serious Cray supercomputer.

Era of increasing clock speeds.

The clock speed race stopped dominating speed

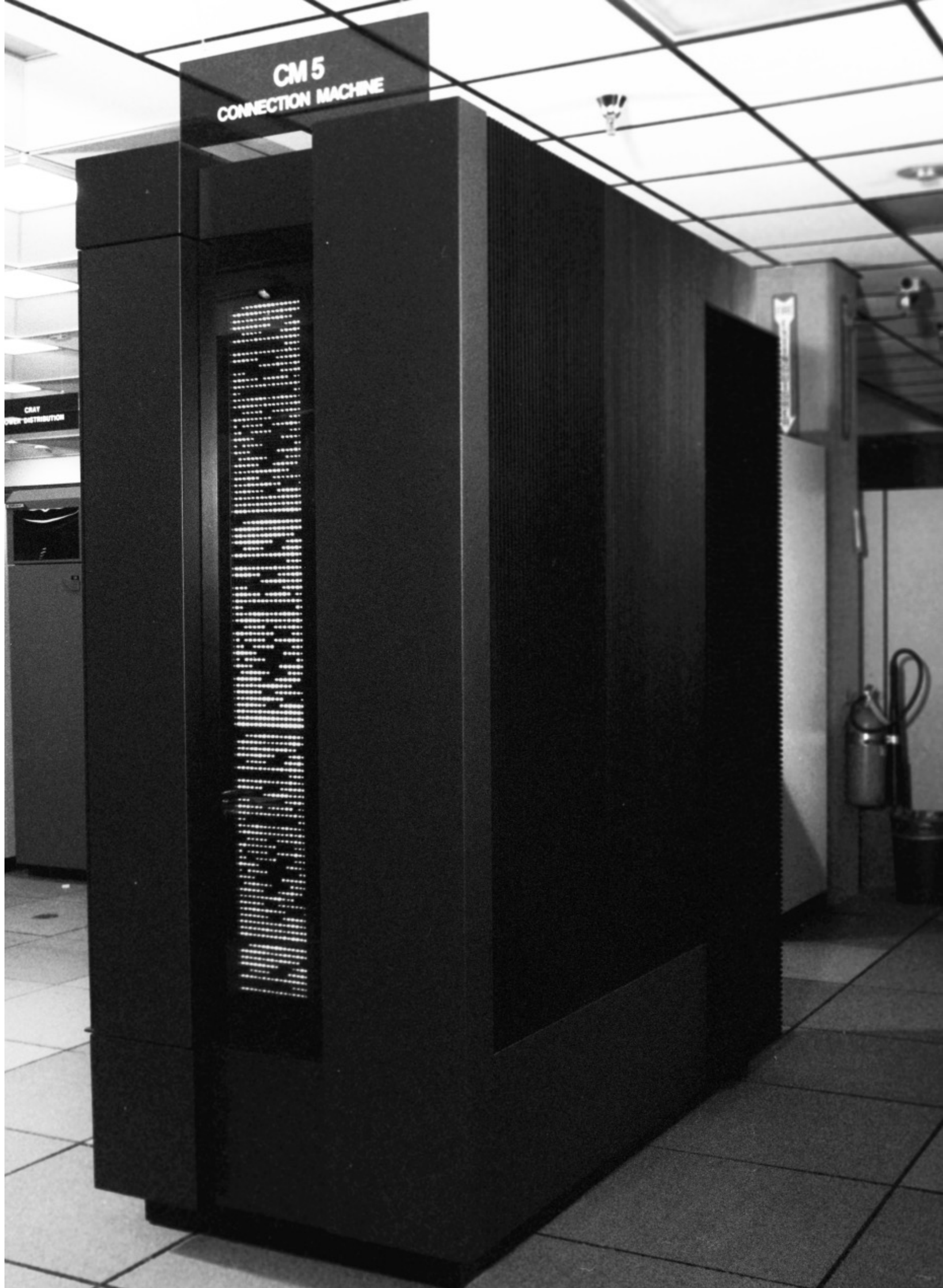


End of the Monocomputer Era

- In 1993 Thinking Machines built their 32 node multicomputer (called connection machines at the time)
- Ran on UNIX
- Achieved a LINPACK benchmark of 60 gigaflops.

- The best Cray monocomputer that year had 16 CPUs and a LINPACK score of 16 gigaflops.

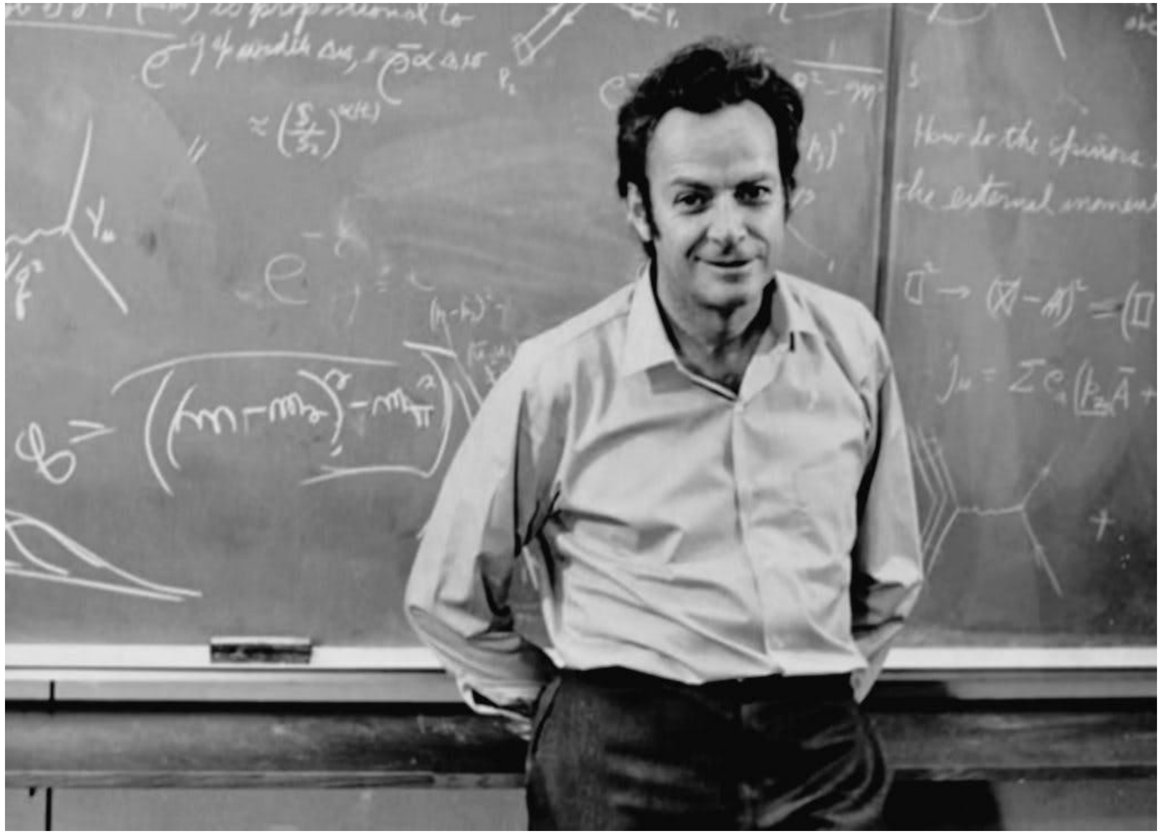
- Seymore Cray: "If you were plowing a field, which would you rather use? Two strong oxen or 1024 chickens?"



Thinking Machines Corp

Connectionist Machines

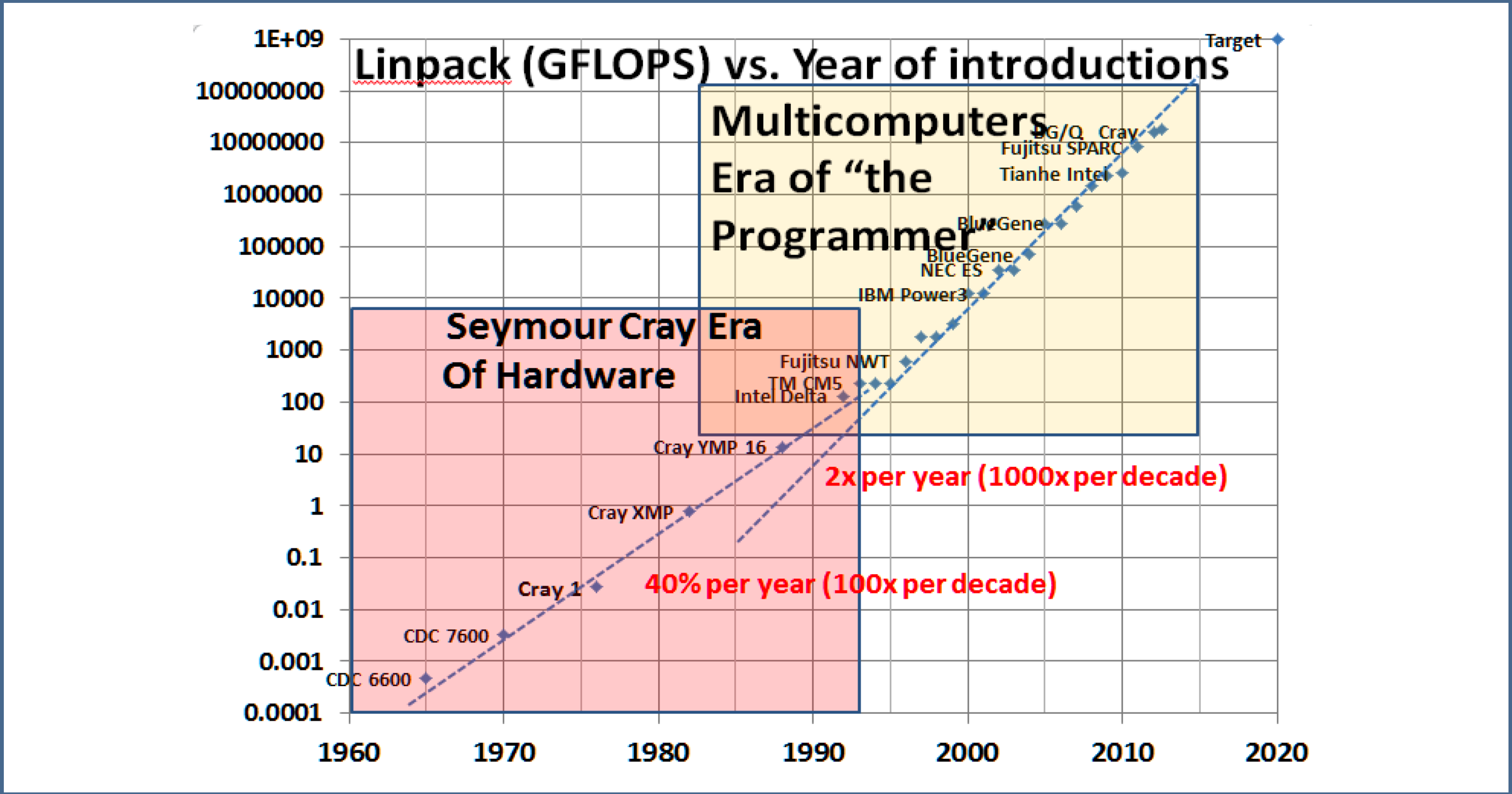
- Founded by MIT student Danny Hillis
- That's good. Now I don't have to explain to people that I work with a bunch of loonies. I can just tell them the name of the company.”
- Feynman was a quantum physicist interested in computation.
- One of the first examples of parallel computations was Feymans reorganization of the IBM computing units during the Manhatten project so they could process calculations in parallel.



Richard Feynman, Caltech

Clock Speed vs Cluster Size

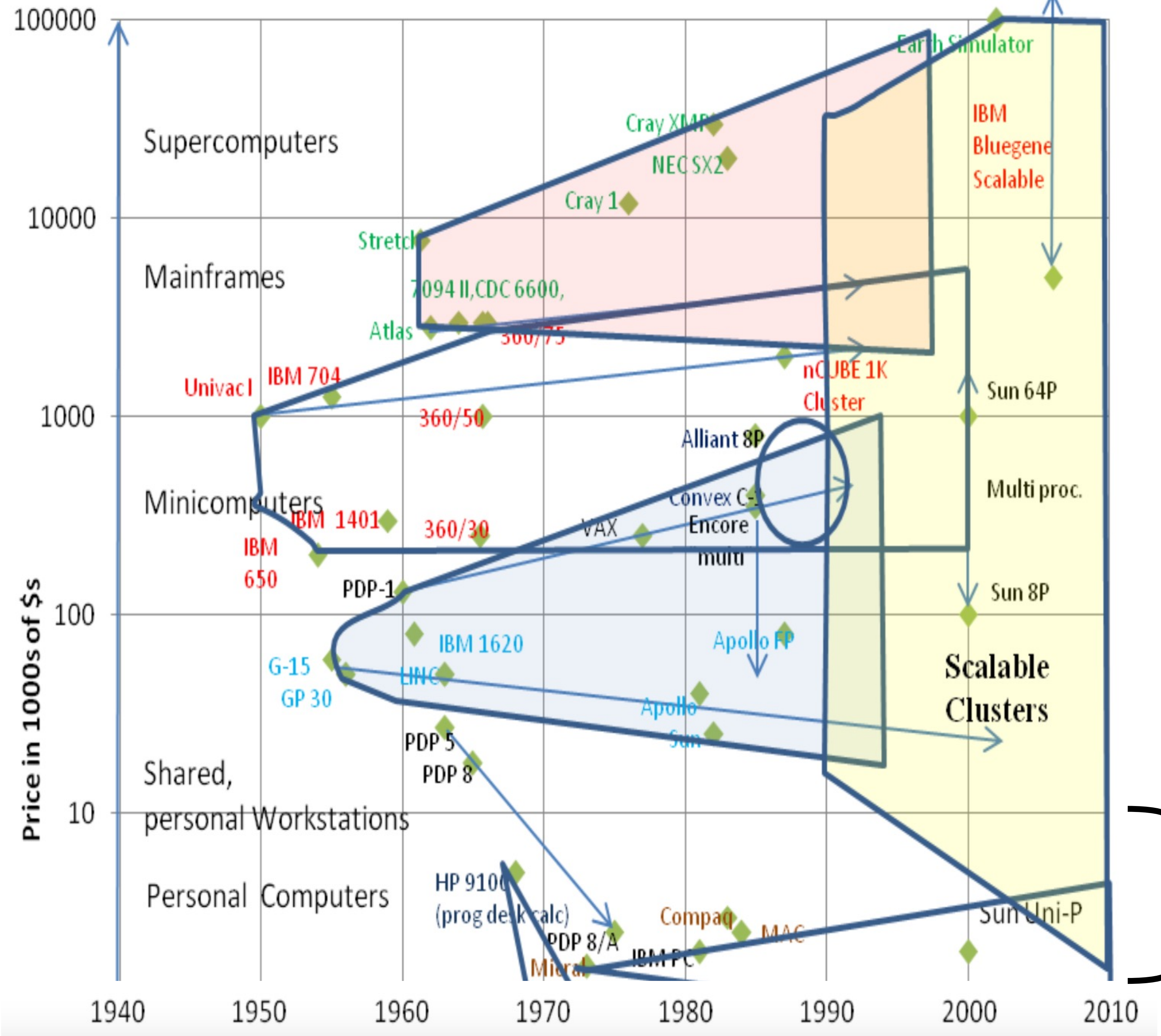
- Between 1983 (5Mhz, 64 node, Caltech Cosmic Cube) and 2013 the number of nodes in a single cluster reach 3 million.
- That's a 50,000 fold increase over 30 years.
- Clock speed increased from from 5 Mhz to 5 Ghz over the same period. 1000x increase.

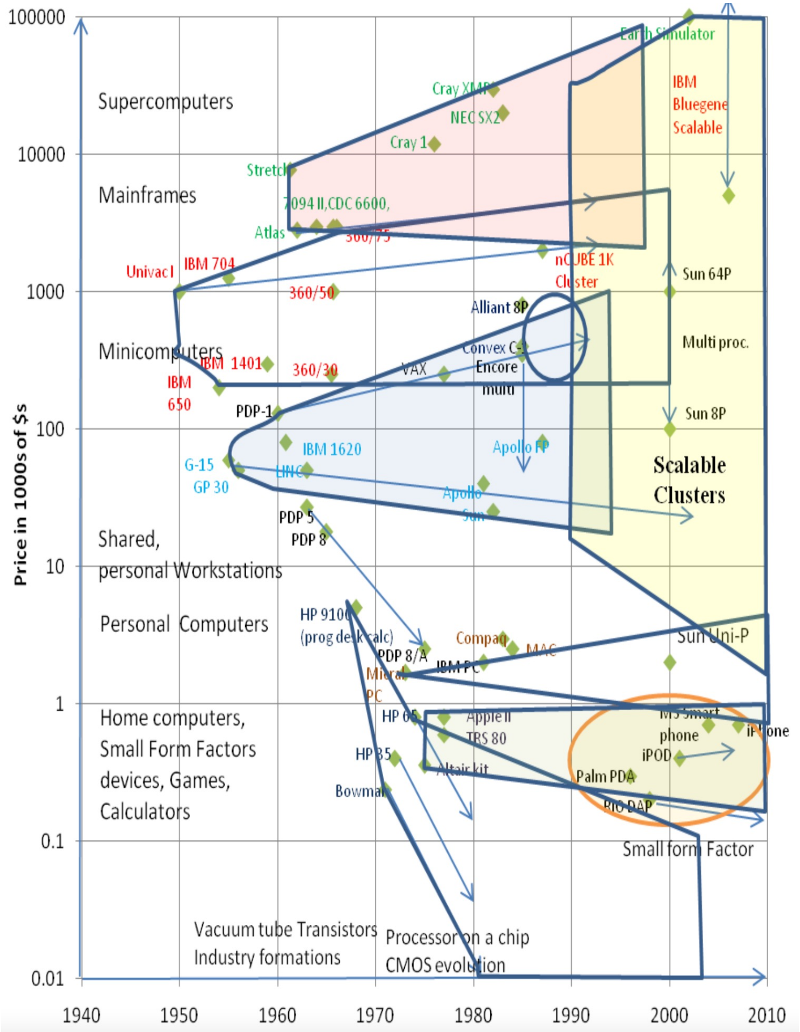


Gordon Bell, Supercomputers: The Amazing Race: DOI: 10.13140/2.1.3425.0562

Four Eras

- The first thirty years of supercomputing was dominated by companies like Cray Research (Seymore Cray), Control Data Corp (CDC) and International Business Machines (IBM) (American companies).
- During this time Japan entered the supercomputing business with Hitachi, Fujitsu, and NEC.
- From 1995 until the present scaling up (monocomputers) became less important than scaling out (multicomputers).
- This change was partly enabled by CMOS and BIOS standards that allowed many manufacturers to create compatible computers (the PC revolution).

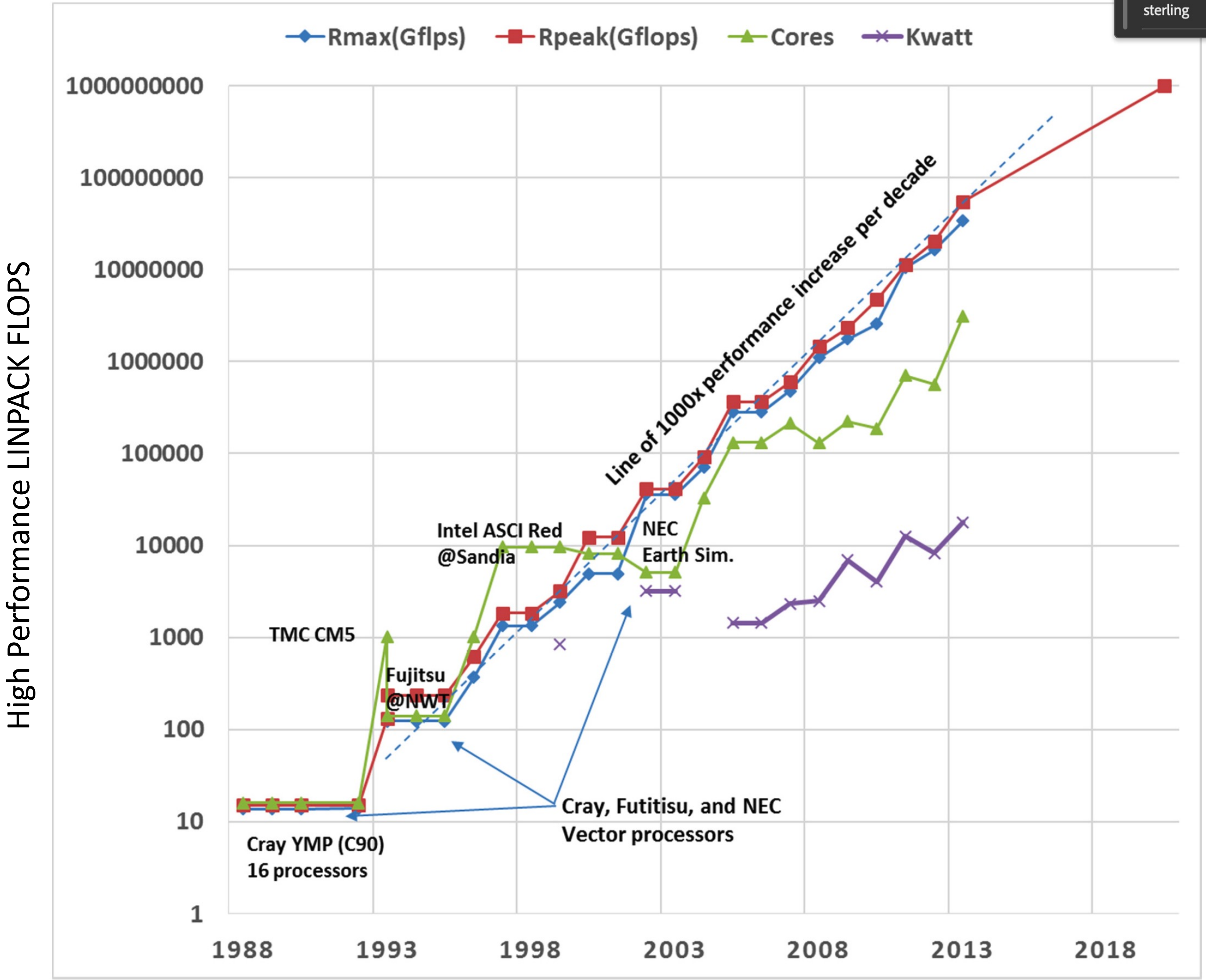




cost/performance
driven down

Gordon Bell, Bell's Law for the birth and death of computer classes: a theory of the computer's evolution

Increasing Cores drives super computing performance



Beowulf Clusters

- **Beowulf clusters are a collection of PCs running Linux networked together.**
- Version 1.0 of the Message Passing Interface (MPI) was developed in 1993.
- It set a standard for FORTRAN and C programs to communicate with each other over a regular ethernet network.
- In particular, it defined how many computers to talk to many other computers and have all the messages synchronized.
- Stirling and Beckers Beowulf system packaged MPI, ran on commodity CMOS machines (PCs), and democratised supercomputing.

Beowulf Clusters

- In 1994 Donald Becker and Tom Stirling, both at NASA, built a cluster using available PCs and networking hardware.
- 16 Intel 486DX PCs connected with 10 Mb/s Ethernet
- Achieved 1 GFLOP/s on $50,000 system
- Linux had been released in 1991.

- Beowulf: ’Because my heart is pure, I have the strength of a thousand men.’

The GPU Era

- In 2001 Nvidia introduced the GeForce 3 video card.
- This card allowed the pixel renderers to be programmed instead of being hardcoded.
- Over time Nvidia made their graphic processing units (GPUs) more programmable with additional programmable engines for vertex and geometry shading.
- These different engines for programming different parts of th graphics pipeline became too complex.
- Nvidia came up with a general model for programming pixel math on the GPU (2009 GeForce 8800)
- The language Nvidia invented to program these pixel operations was the Compute Unified Compute Architecture (CUDA)
- GPUs are part of the reason clusters are moving towards havi fewer but more powerful nodes.

GPUs for Computation

- GPUs proved to be ideal for matrix math.
- The GPU architecture applies the same operation to thousands of elements at the same time.
- This is a bit like the earlier CPU vectorization operations on steroids (512 vector operations in the latest Intel CPUs, The Nvidia H100 GPU has 14,592 CUDA cores)
- The H100 can execute almost 15,000 instructions per clock cycle.
- This doesn't work for all types of computations, if you can't apply the same operation to a vector of values the GPU will be slower than a CPU.

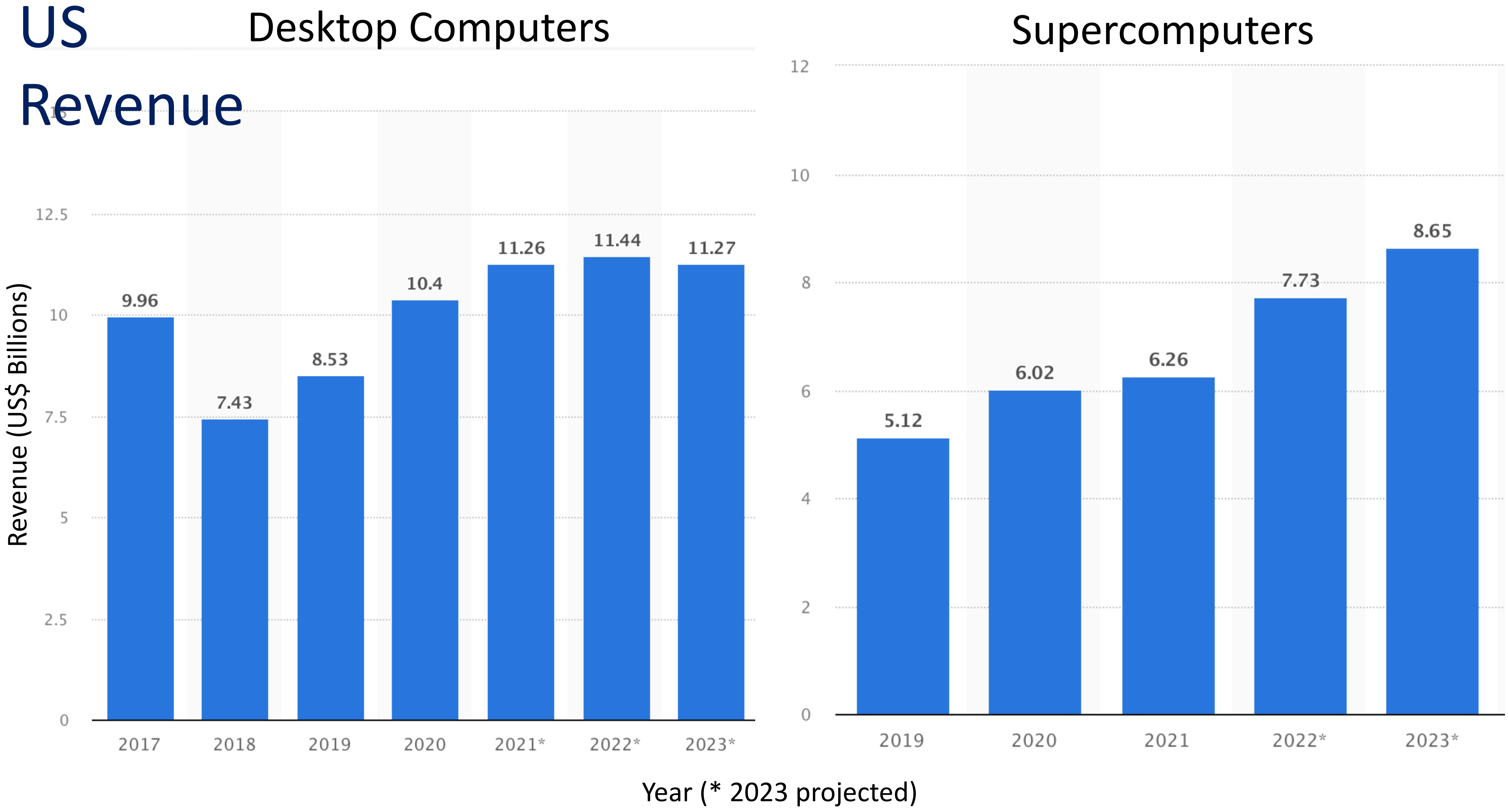
Scaling up

- NVIDIA DGX-2H ($400,000 each, 81k CUDA cores, 10240 tensor cores) https://www.nvidia.com/content/dam/en-zz/es_em/Solutions/Data-Center/dgx-2/nvidia-dgx-2h-datasheet.pdf

Scaling out

- Stampede 2
- https://www.tacc.utexas.edu/systems/stampede2
- $30,000,000, 285,000 CPUs

US Revenue



Desktop Computers
Revenue (US$ Billions)
15
12.5
10
7.5
5
2.5
0
9.96
7.43
8.53
10.4
11.26
11.44
11.27
2017
2018
2019
2020
2021*
2022*
2023*
Supercomputers
12
10
8
6
4
2
0
5.12
6.02
6.26
7.73
8.65
2019
2020
2021
2022*
2023*
Year (* 2023 projected)

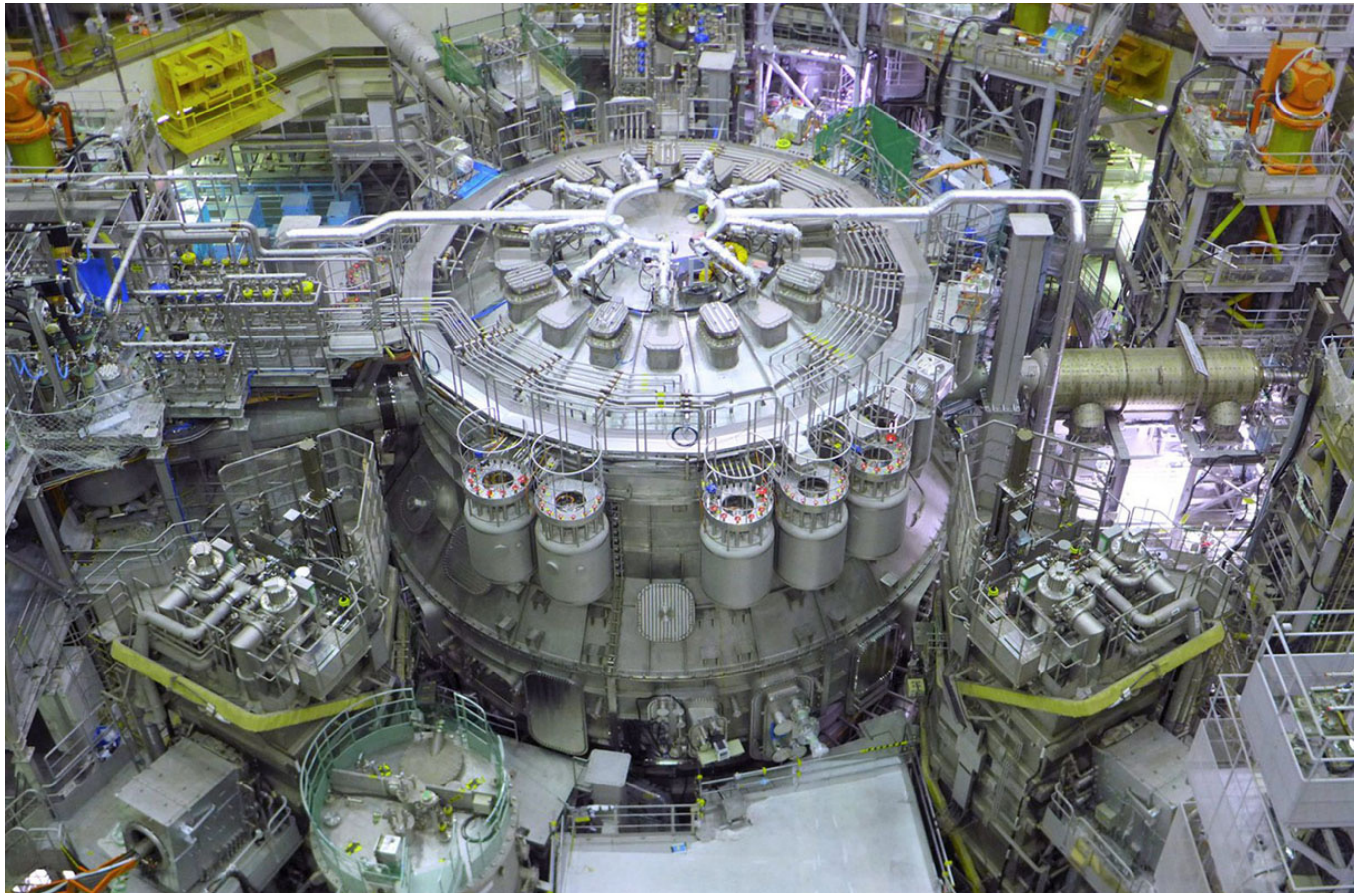
Frontier and Aurora

- Frontier and Aurora are the two fastest supercomputers in the world.

- Frontier is at Oak Ridge National Labs
  - 606,208 AMD Epyc Cores, 37,888 AMD Radeon MI250X GPUs (8,335,360 CUDA cores).
  - 22 Mwatt system
  - Frontier was the first exaflop system with 1.67×10^{18} floating point operations per second.
  - Cost $600 million
  - Built by HPE Cray and Advanced Micro-Devices (Hewlett Packard bought Cray in 2019)

Frontier and Aurora

- Aurora is still being deployed but is already the second fastest super computer in the world at 500 petaflops.
- When complete it will be a 2 exaflop supercomputer as measured by the High Performance LINPACK (HPL) Benchmark.
- Argonne National Labs
- Frontier is at Oak Ridge National Labs
    - 1.1 million Intel Saphire Rapids cores, 63,744 Intel GPUs (8,335,360 CUDA cores).
    - 60 Mwatt system
    - Cost $500 million
    - Built by HPE Cray and Intel



Frontier and Aurora

- Both will primarily be used for simulating nuclear fusion and fission reactors.
- They will also be used for everything from modeling virus evolution to windfarms.

Measuring Performance

- I've mentioned FLOPS a lot so far along with LINPACK.
- Supercomputers have historically been defined by the number of floating point operations they can perform per second.
- The standard benchmark for measuring those operations is High Performance LINPACK.
- LINPACK is a very commonly used linear algebra package.
- Jack Dongarra developed LINPACK and HPL while he was a CS graduate student here at UNM.

HPCG

- The Top500 list is based on LINPACK and HPL
- High Performance Conjugate Gradients (HPCG) benchmark is a newer benchmark
- HPCG is supposed to have access patterns that are more representative of other applications
- Benchmarks drive cluster development

What is Modern Supercomputing?

1 processor with 6 cores.
Can run 6 programs at a time.

1 processor 2 cores

What is Supercomputing?

1 processor with 6 cores.
Can run 6 programs at a time.

1 processor 2 cores

1 processor with 8 cores.
Can run 8 programs at a time.

CARC Systems

$200,000/yr power bill

Wheeler Cluster (John Wheeler)

608 processors with 2432 cores.
Can run 2432 programs at a time.

Xena Cluster
33 K40 GPUs

Hopper Cluster (Grace Hopper)
2336 cores
27 A100 GPUs

And others…

Computational Science

Massive computation allows us to simulate all kinds of things that we could never observe directly.

George Luger called this the "computational telescope".

Four Eras

- The first thirty years of supercomputing was dominated by companies like Cray Research (Seymore Cray), Control Data Corp (CDC) and International Business Machines (IBM) (American companies).
- During this time Japan entered the supercomputing business with Hitachi, Fujitsu, and NEC.
- Until 1995 (What happened in 1995?)
- From 1995 until the present scaling up (monocomputers) became less important than scaling out (multicomputers).
- GPU Era – 2000s to now

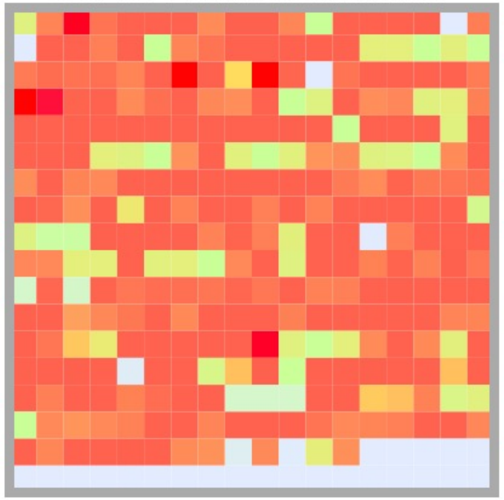
Overview of @ 2023-04-20 20:21

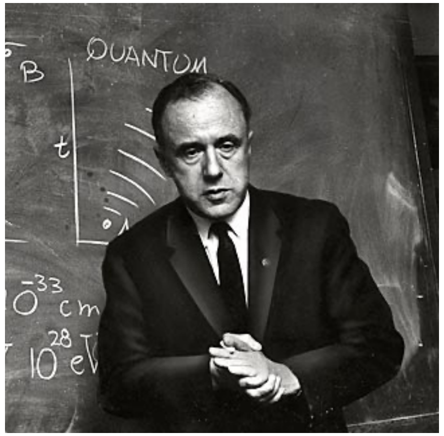
CPUs Total: **2440**
Hosts up: **305**
Hosts down: **0**

Current Load Avg (15, 5, 1m):
84%, 84%, 83%

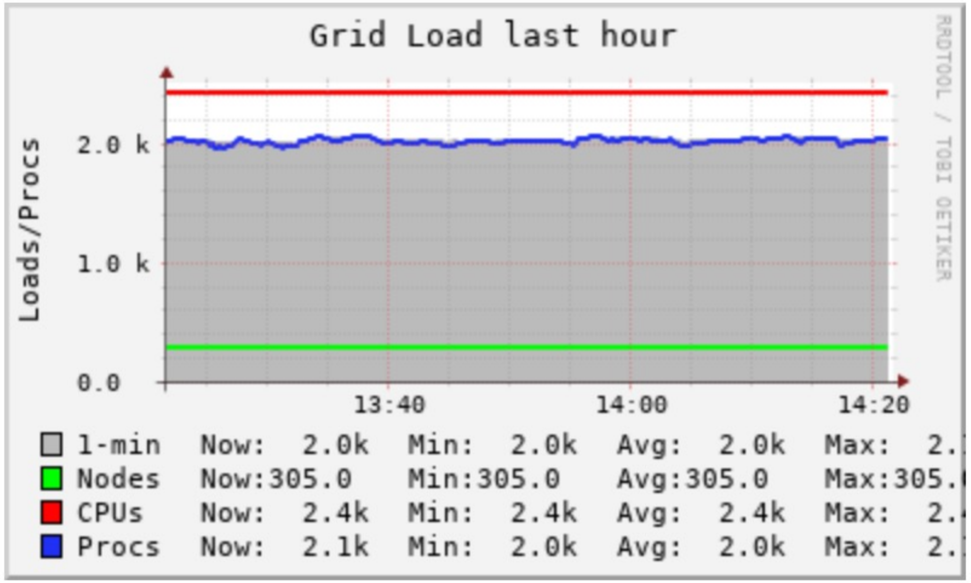
Avg Utilization (last hour):
83%

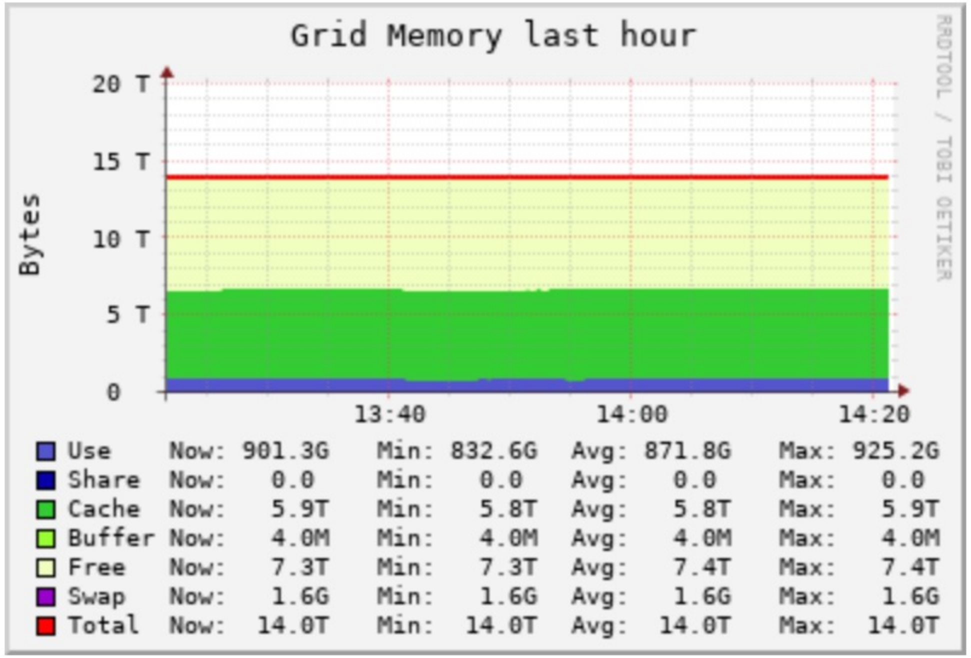
Server Load Distribution

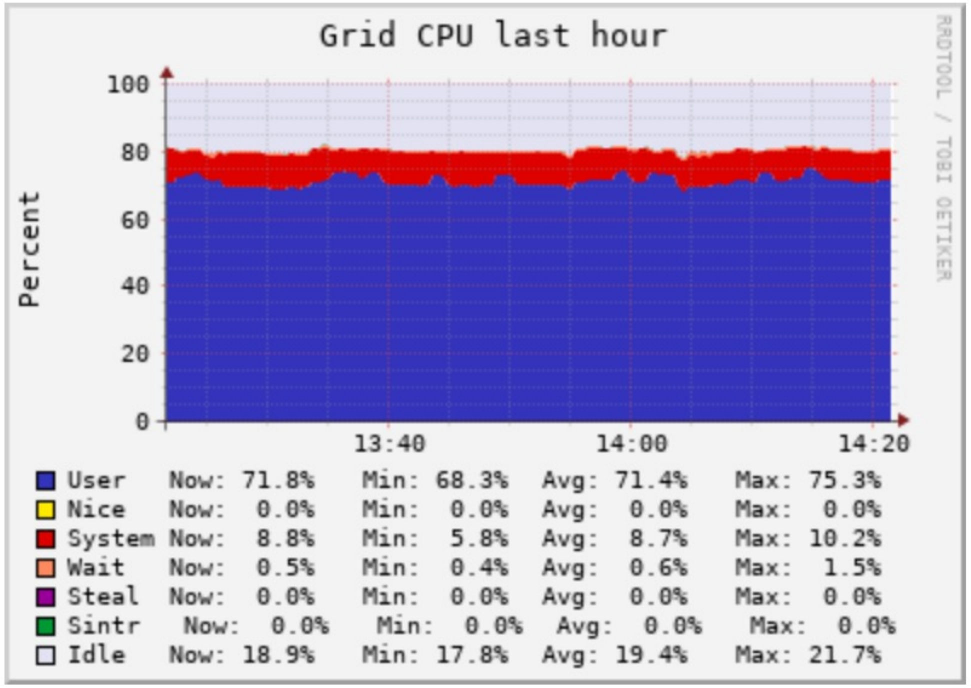


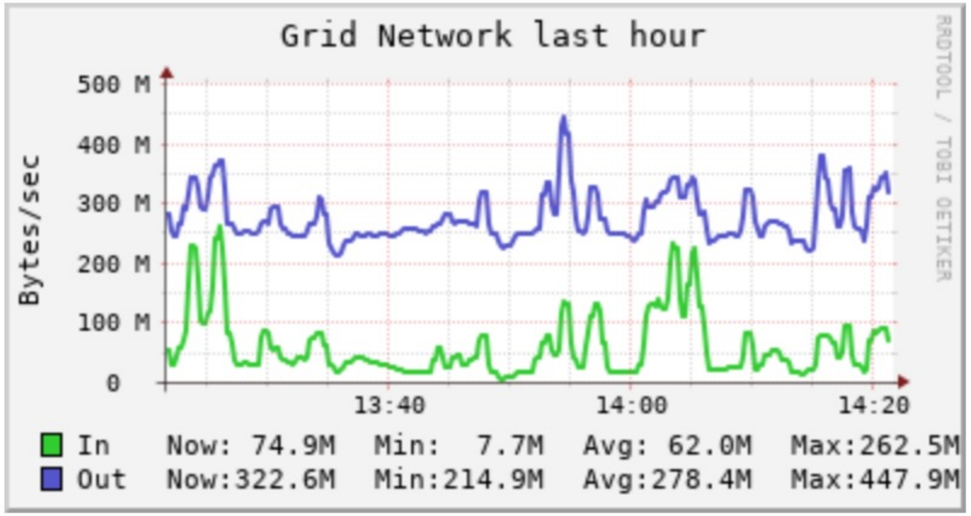


Wheeler







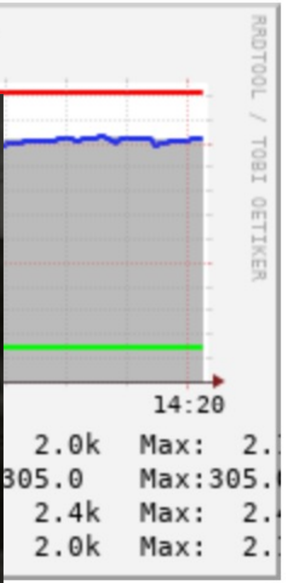


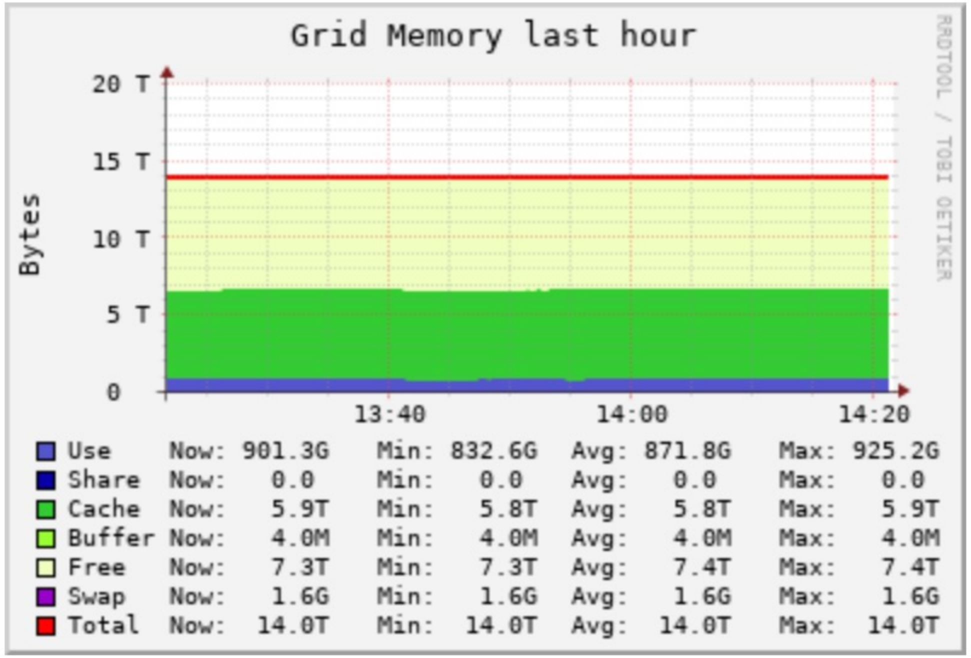
Overview of @ 2023-04-20 20:21

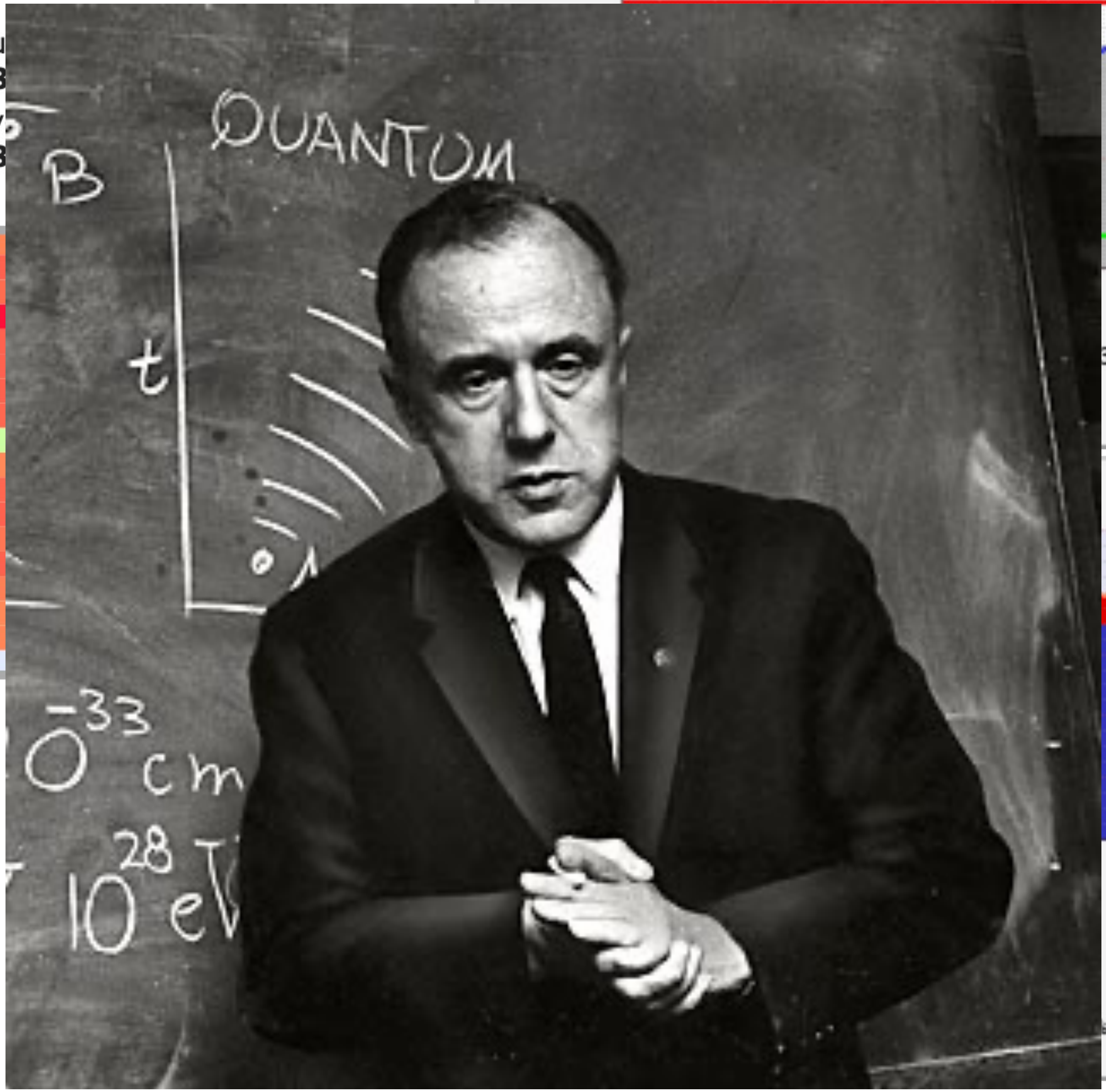
CPUs Total: **2440**
Hosts up: **305**
Hosts down: **0**

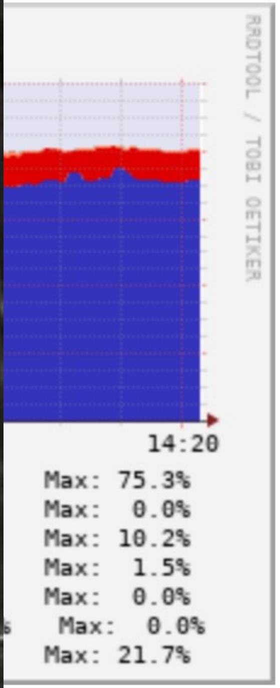
Cu
8
Av
8

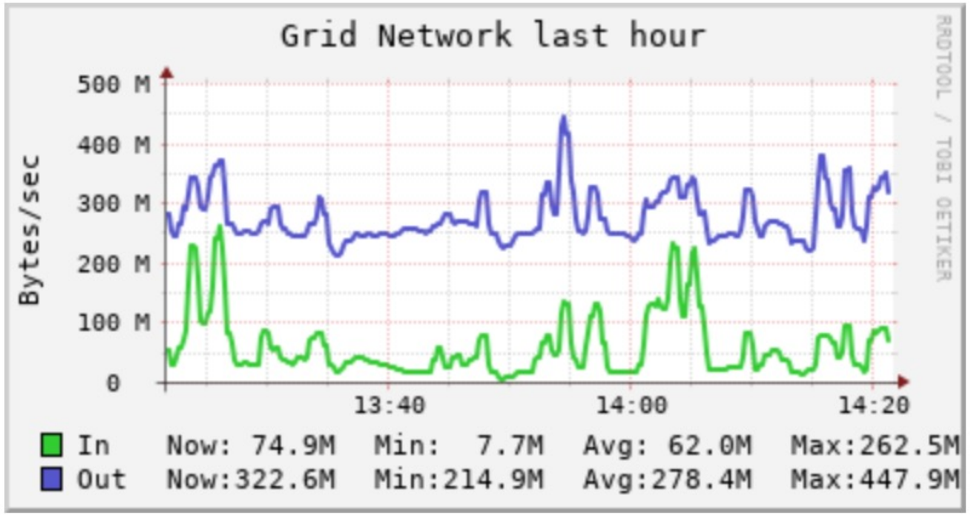
Wheeler











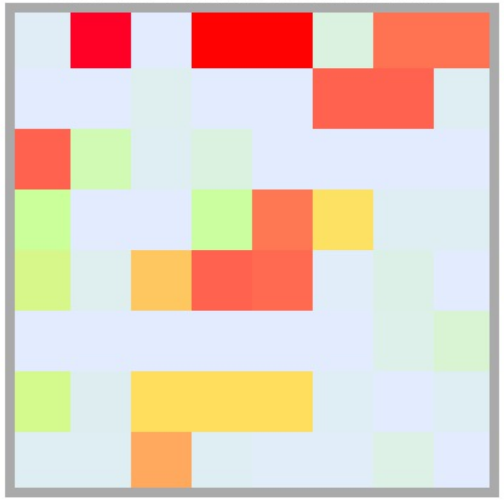
Overview of @ 2023-04-20 20:39

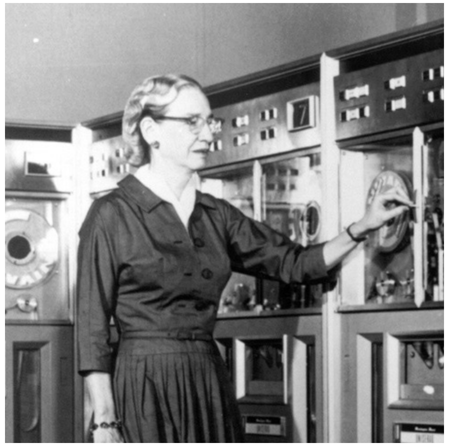
CPUs Total:	**2336**
Hosts up:	**63**
Hosts down:	**0**

Current Load Avg (15, 5, 1m):
40%, 37%, 35%

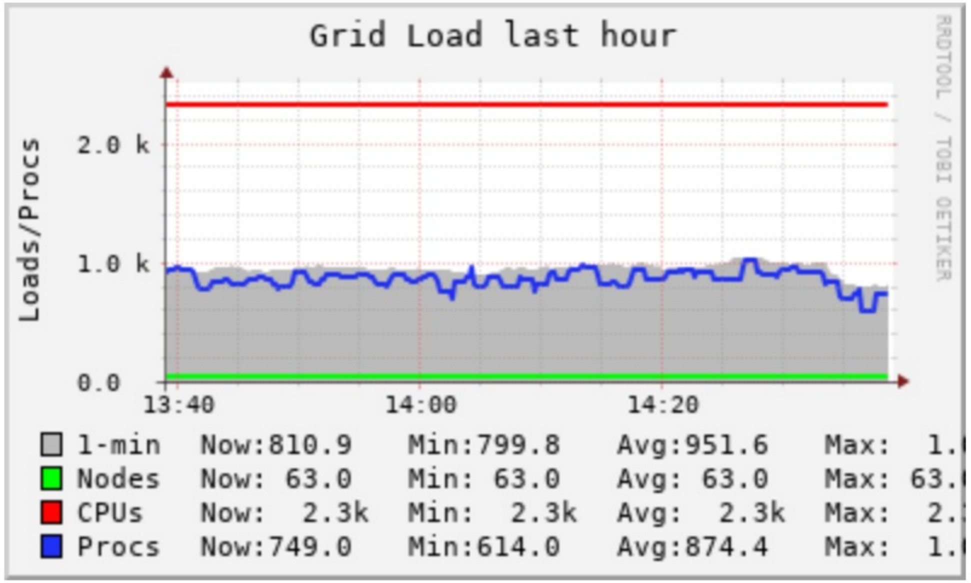
Avg Utilization (last hour):
41%

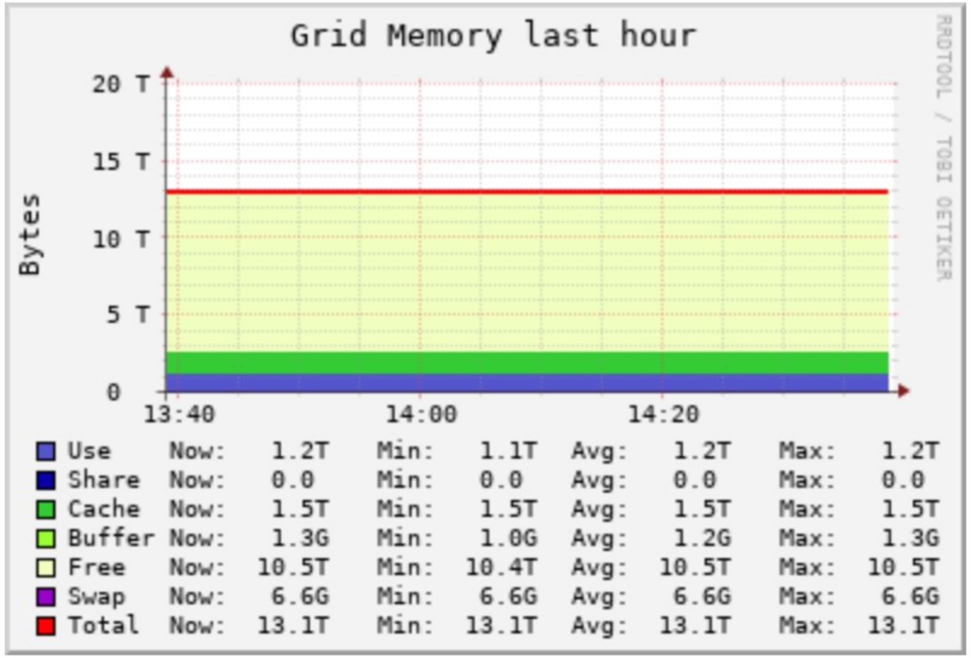
Server Load Distribution

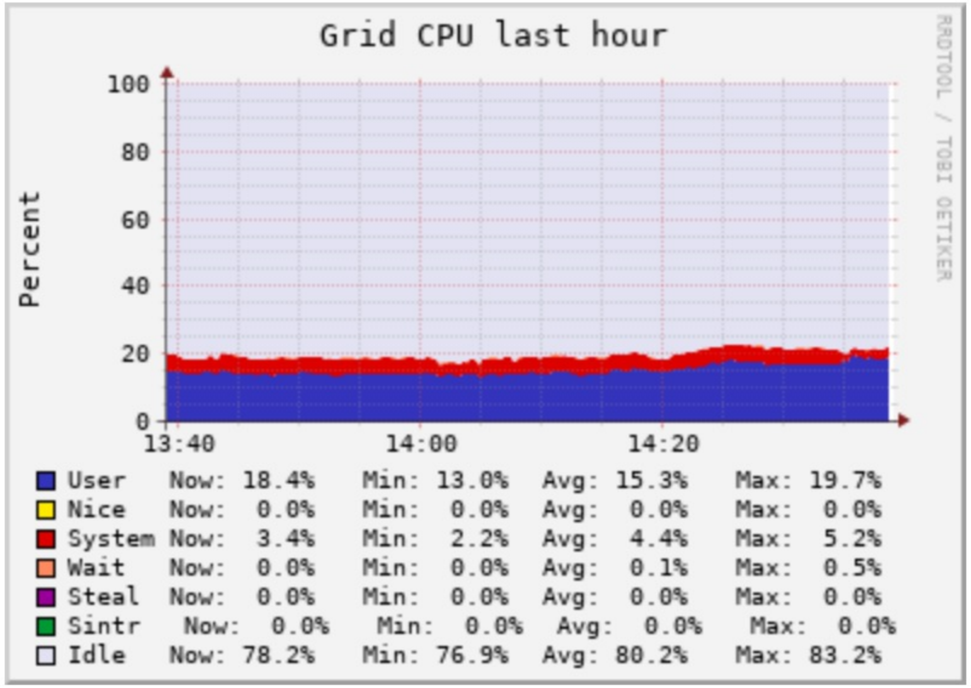


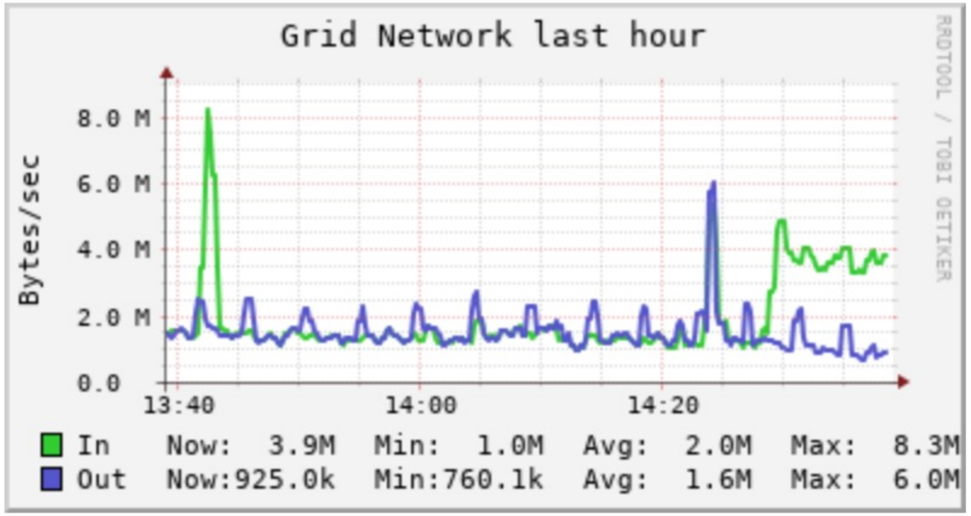


Hopper





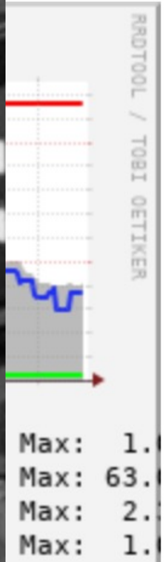


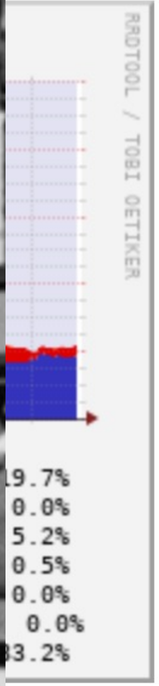


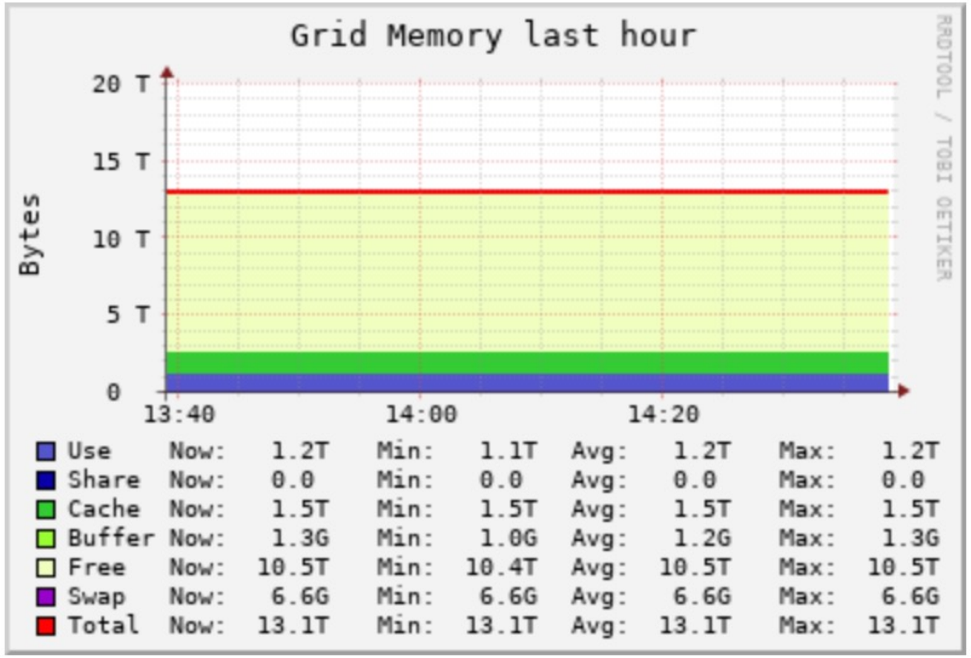
Overview of @ 2023-04-20 20:39

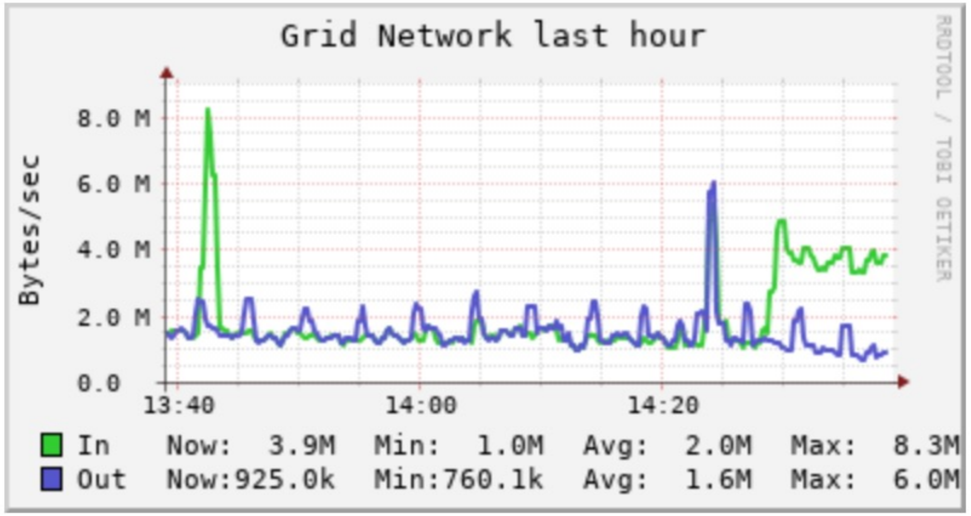
CPUs Total: 2336

Hopper







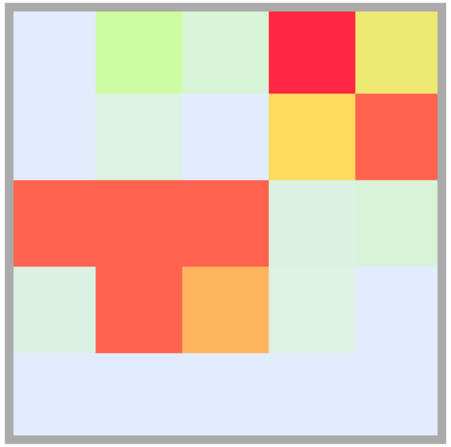


CPUs Total:	**Xena**	**416**
Hosts up:		**26**
Hosts down:		**0**

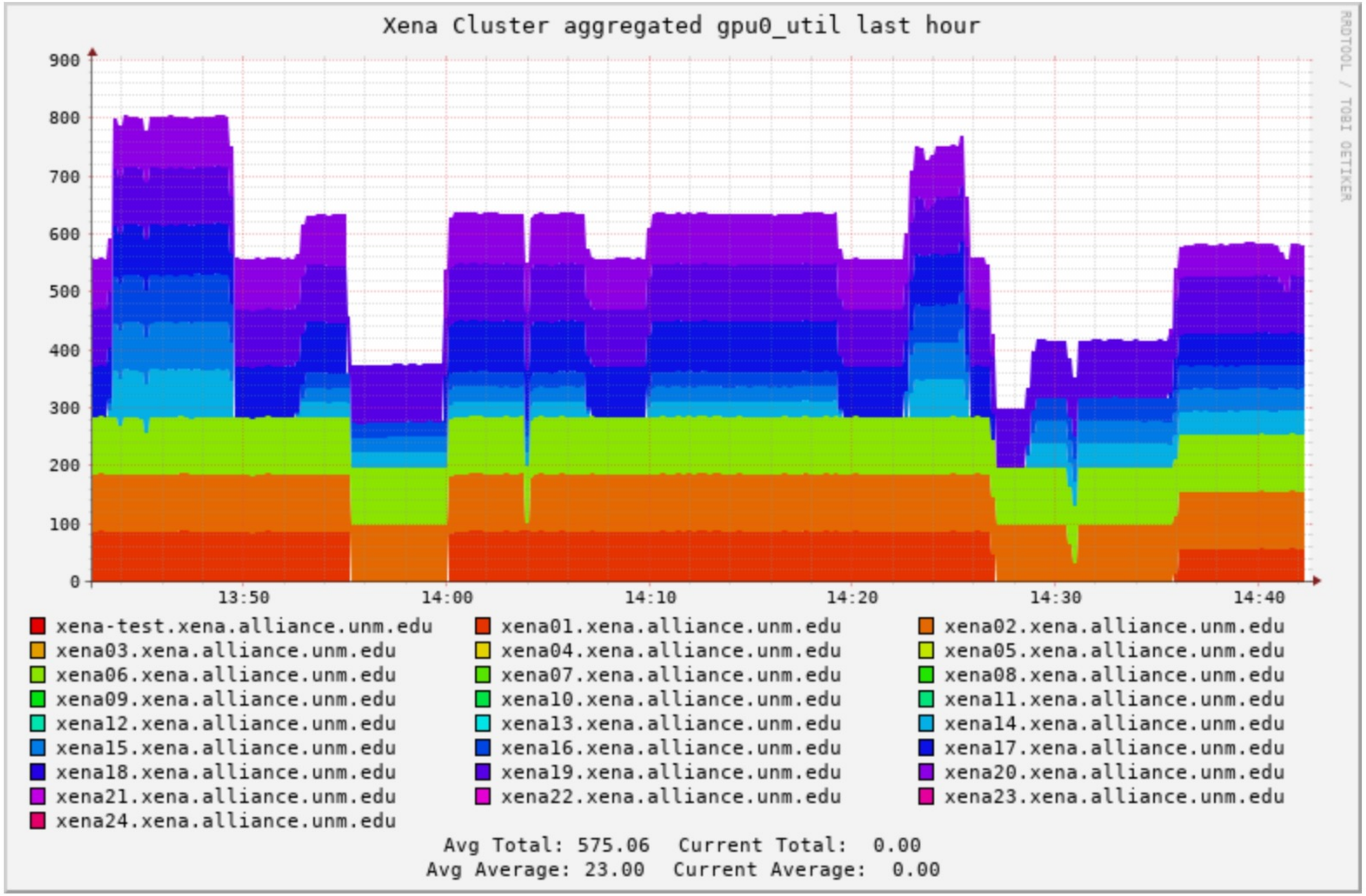
Current Load Avg (15, 5, 1m):
39%, 39%, 39%

Avg Utilization (last hour):
38%

Server Load Distribution



Stacked Graph - gpu0_util (%)



CPUs Total: **416**

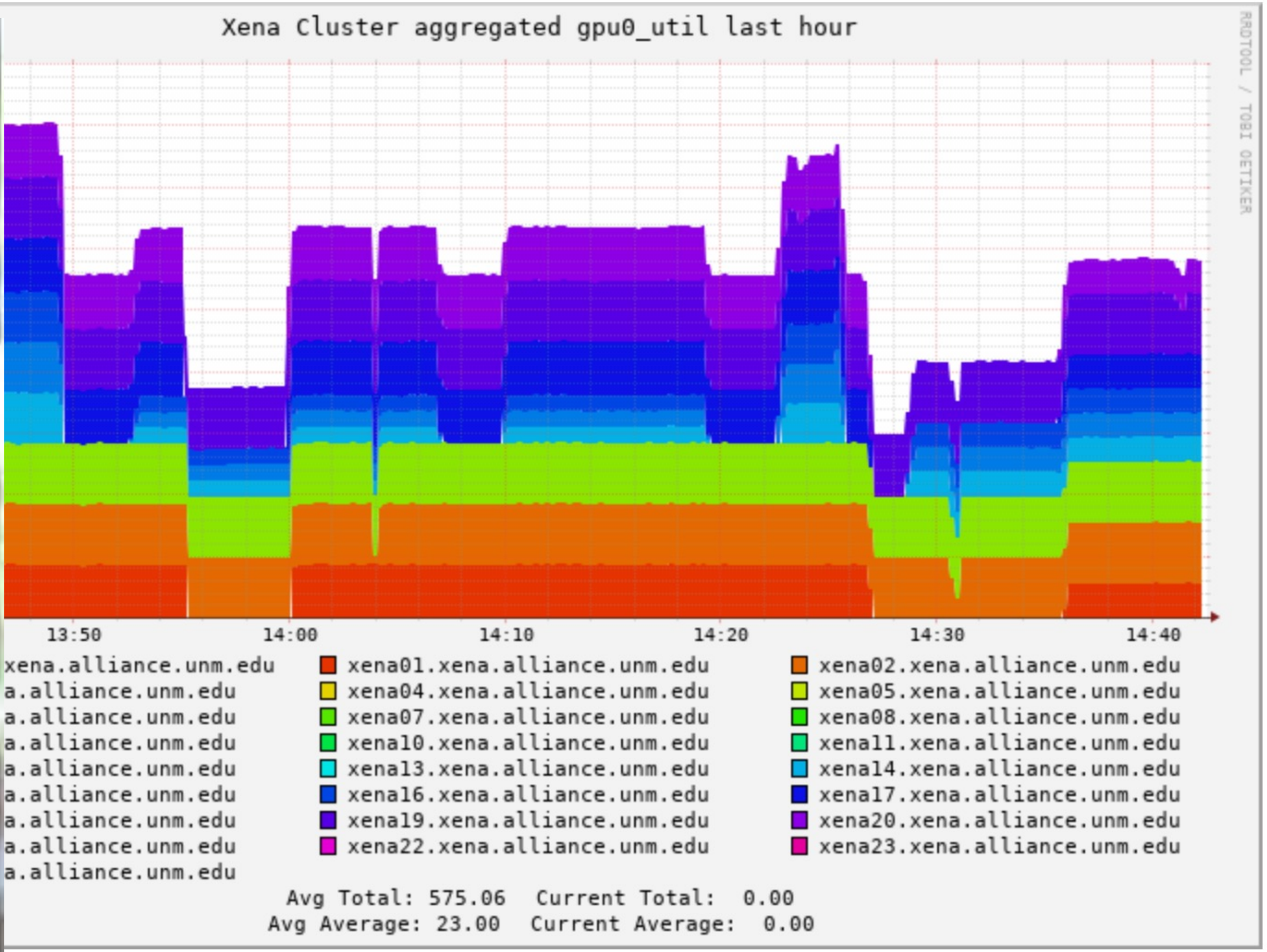
Hosts up: **26**

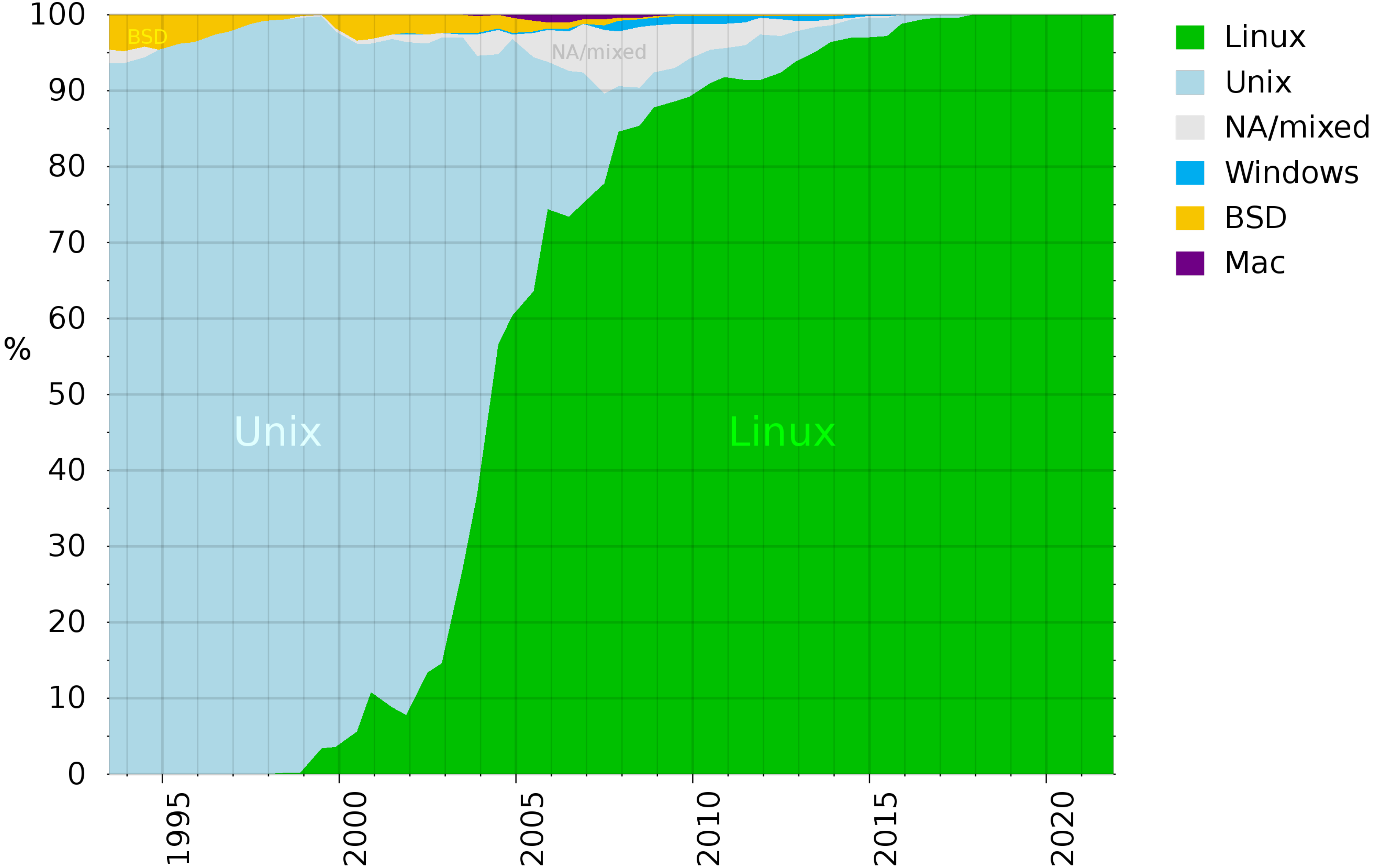
H

C

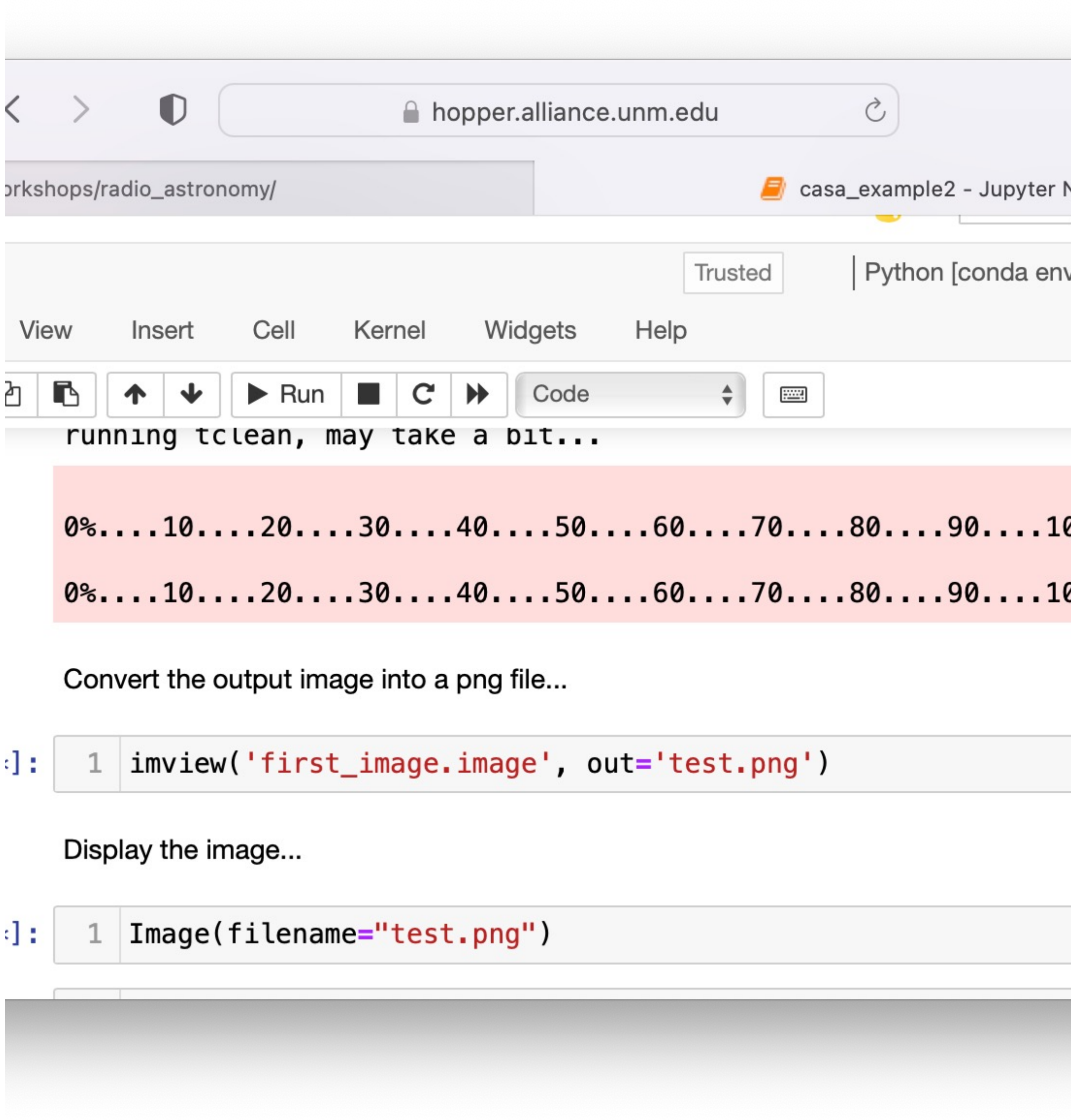
A

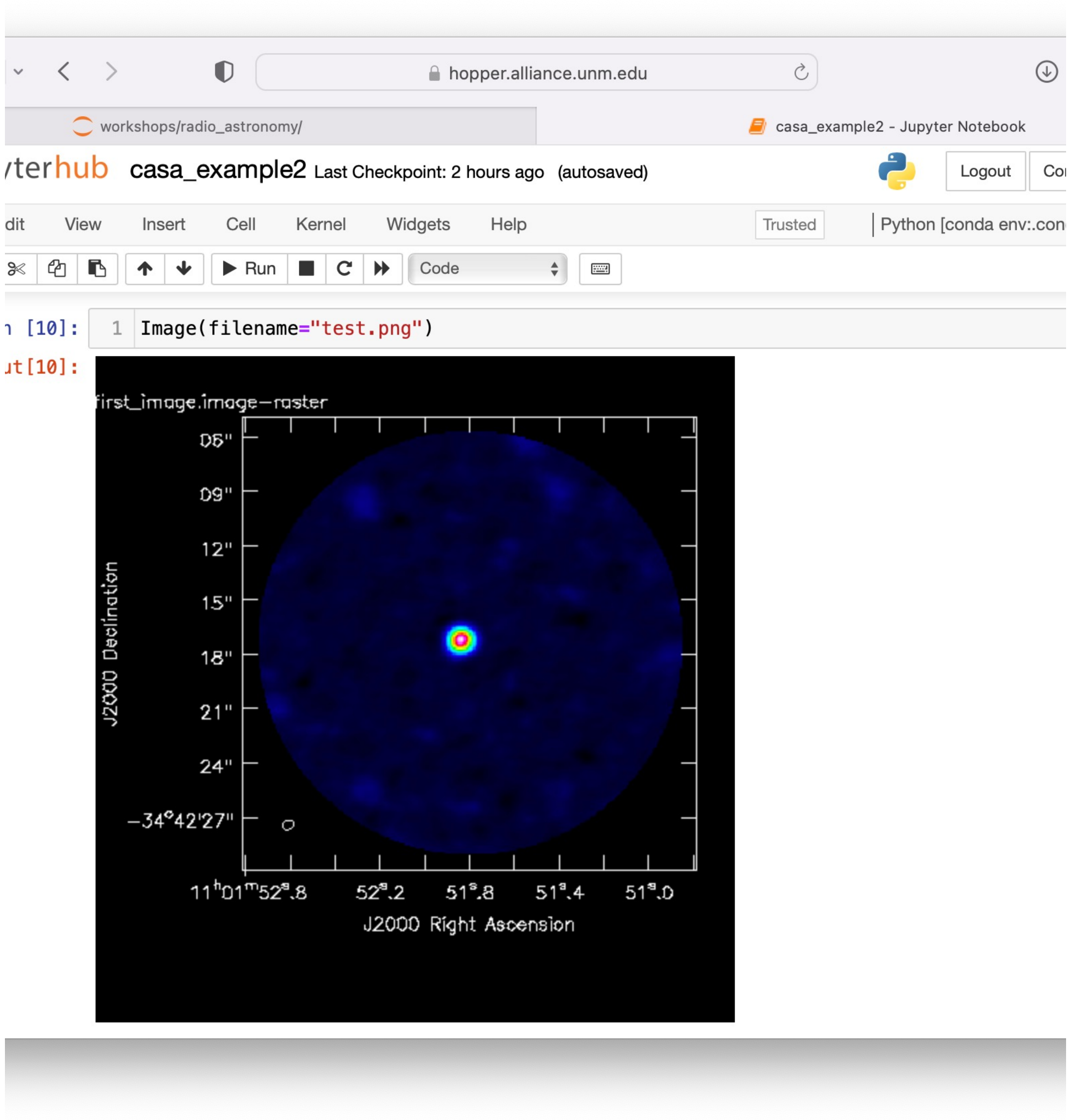
Stacked Graph - gpu0_util (%)



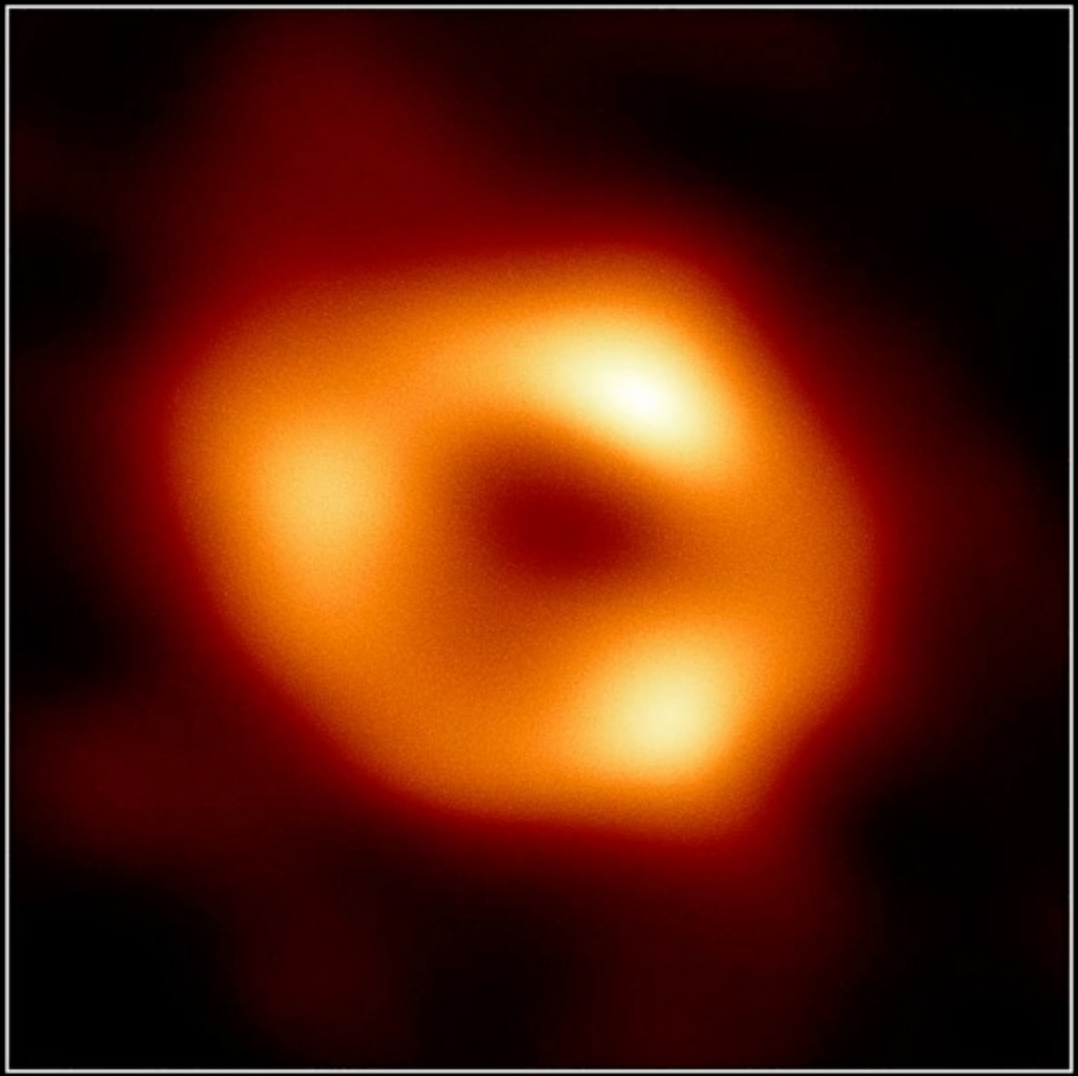

100
90
80
70
60
50
40
30
20
10
0
%
BSD
NA/mixed
Unix
Linux
1995
2000
2005
2010
2015
2020
Linux
Unix
NA/mixed
Windows
BSD
Mac

Astronomy


hopper.alliance.unm.edu
orkshops/radio_astronomy/
casa_example2 - Jupyter N
Trusted
Python [conda env
View Insert Cell Kernel Widgets Help
Run
Code
running tclean, may take a bit...
0%....10....20....30....40....50....60....70....80....90....10
0%....10....20....30....40....50....60....70....80....90....10
Convert the output image into a png file...
]: 1 imview('first_image.image', out='test.png')
Display the image...
]: 1 Image(filename="test.png")


hopper.alliance.unm.edu
workshops/radio_astronomy/
casa_example2 - Jupyter Notebook
yterhub casa_example2 Last Checkpoint: 2 hours ago (autosaved)
Logout
dit View Insert Cell Kernel Widgets Help
Trusted
Python [conda env:.con
Run
Code
n [10]: 1 Image(filename="test.png")
ut[10]:
first_image.image–raster
06"
09"
12"
15"
18"
21"
24"
−34°42'27"
J2000 Declination
11h01m52s.8
52s.2
51s.8
51s.4
51s.0
J2000 Right Ascension

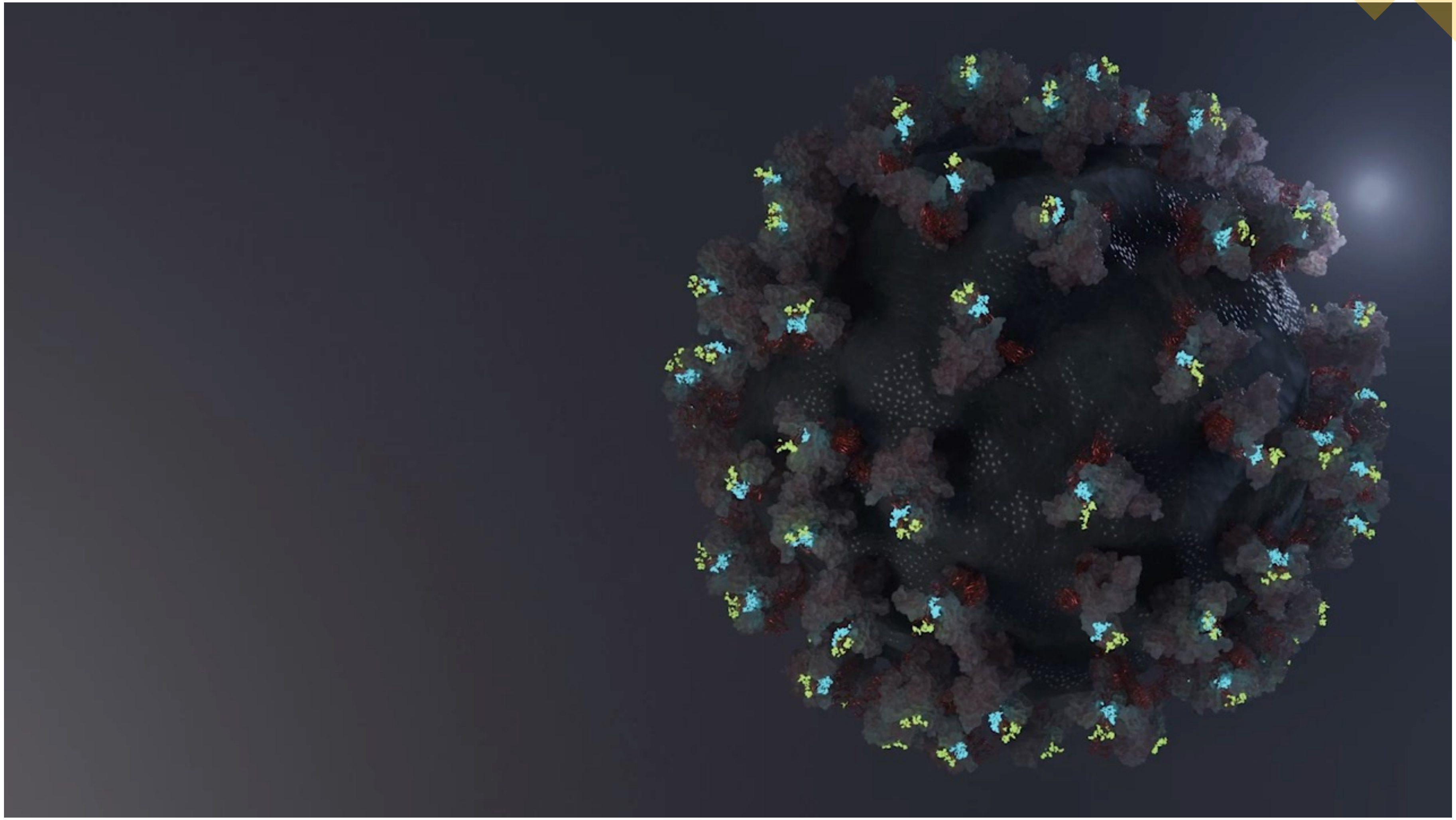
Texas Advanced Computing Center 448,448 cores (Frontiera Cluster)



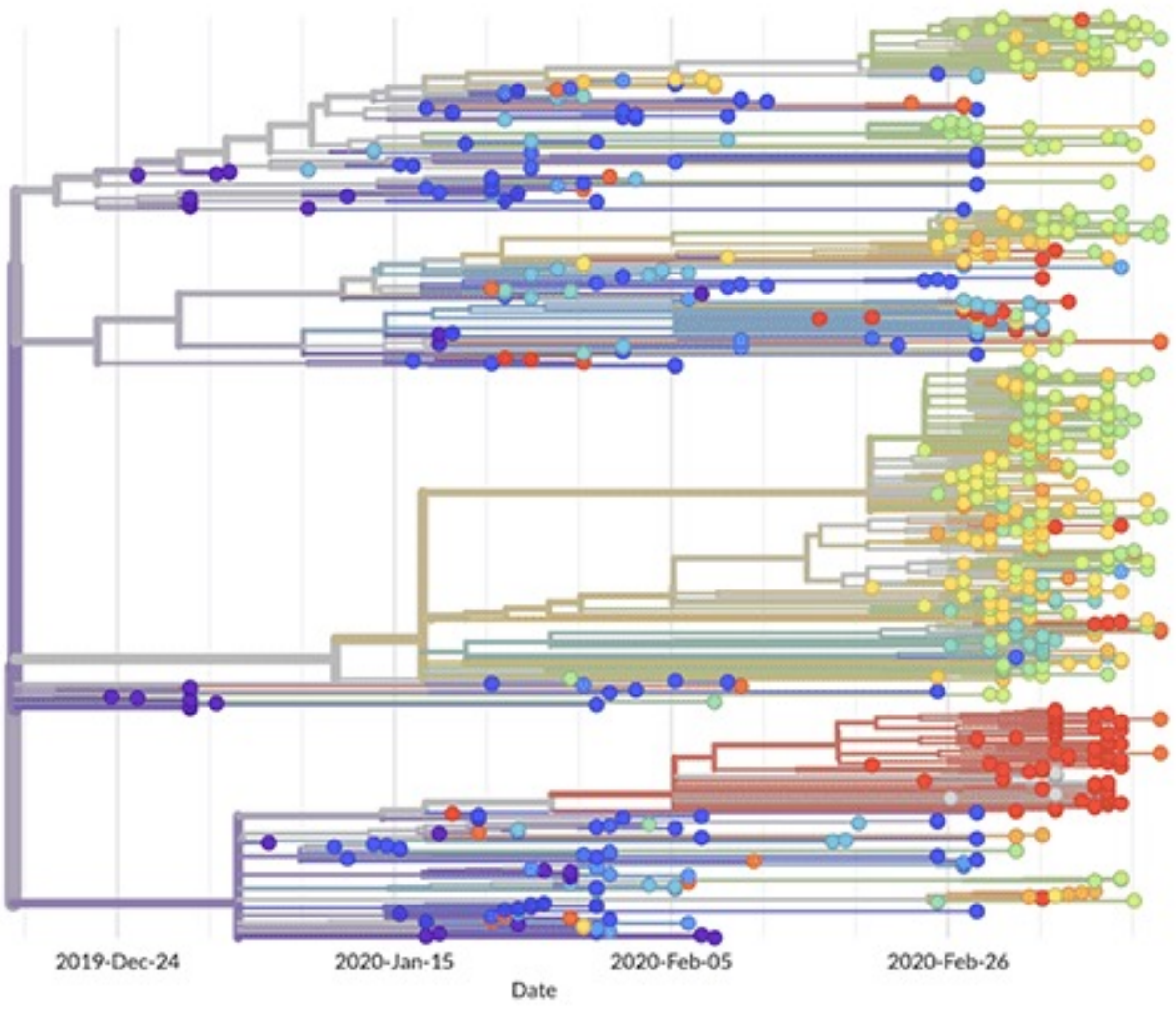
DiRAC High-Performance Computing facility

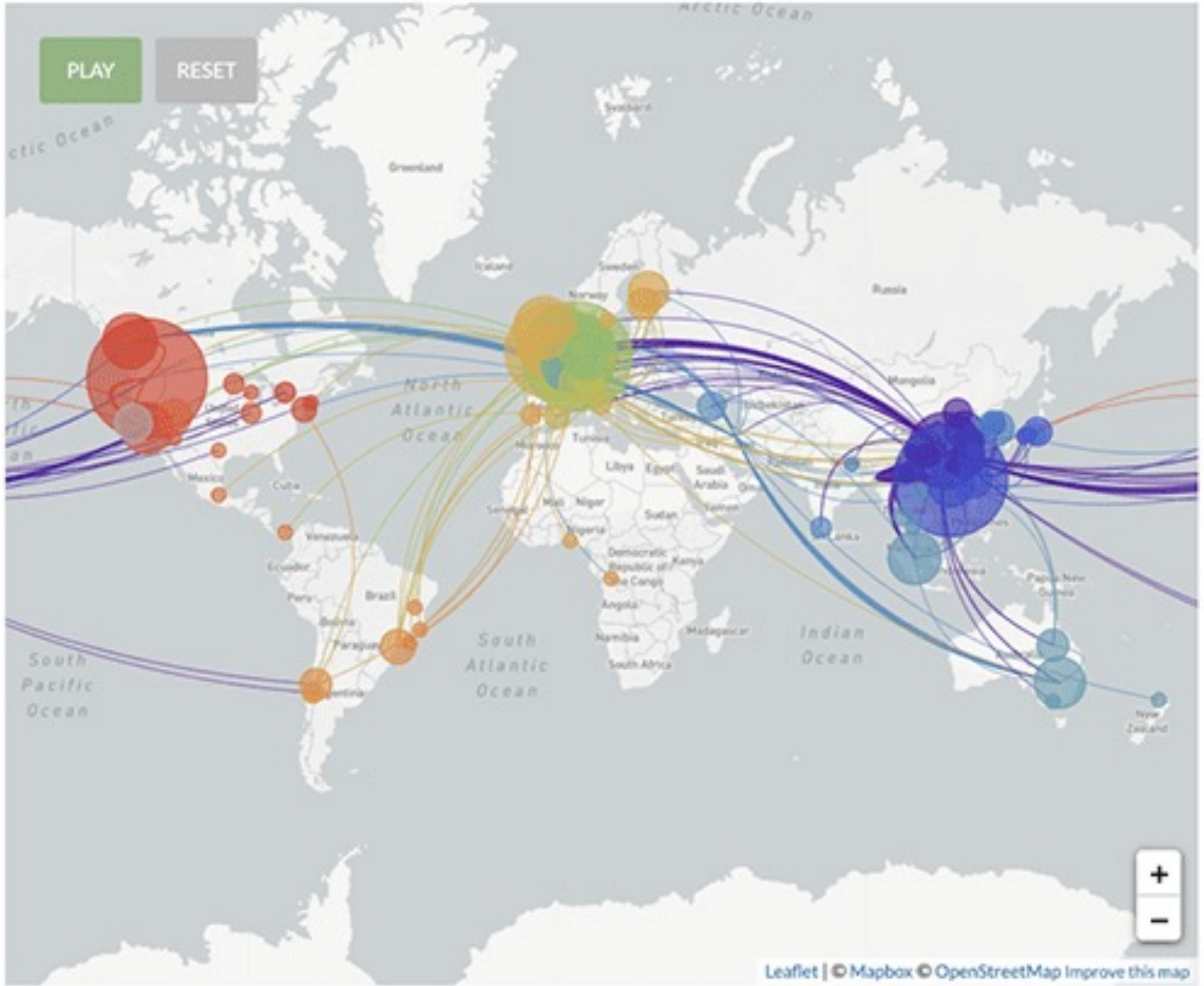
Biology and Chemistry – CoVID Virus Model

- VMD – Visual Molecular Dynamics
- This model shows how the CoVID spike works (It's the part in yellow)
- That's how CoVID gets into our cells
- This simulation used 200,000 GPUs

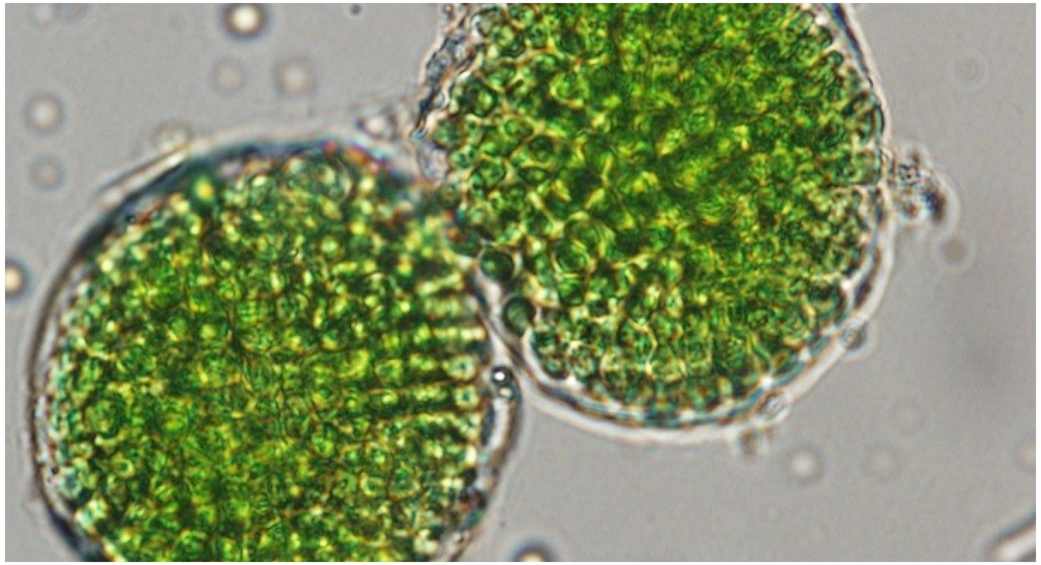


Biology and Chemistry – CoVID Evolution





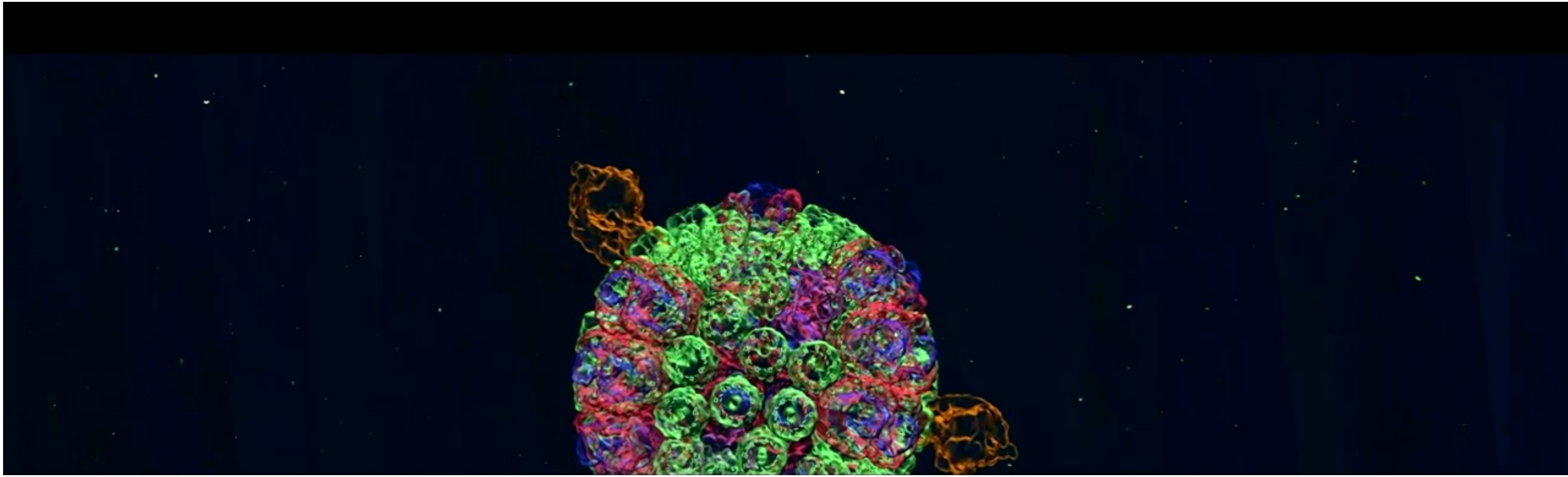
CoVID 19's family tree. CARC systems are responsible for sequencing CoVID variants for 4 states.



Molecule level model of a photosynthesizing organelle. The green objects are chloroplasts.

Specialized for low level light.

NASA was able to grow these bacteria under the same light conditions that you find around red dwarf stars. They are a lot more common than our own type of star.



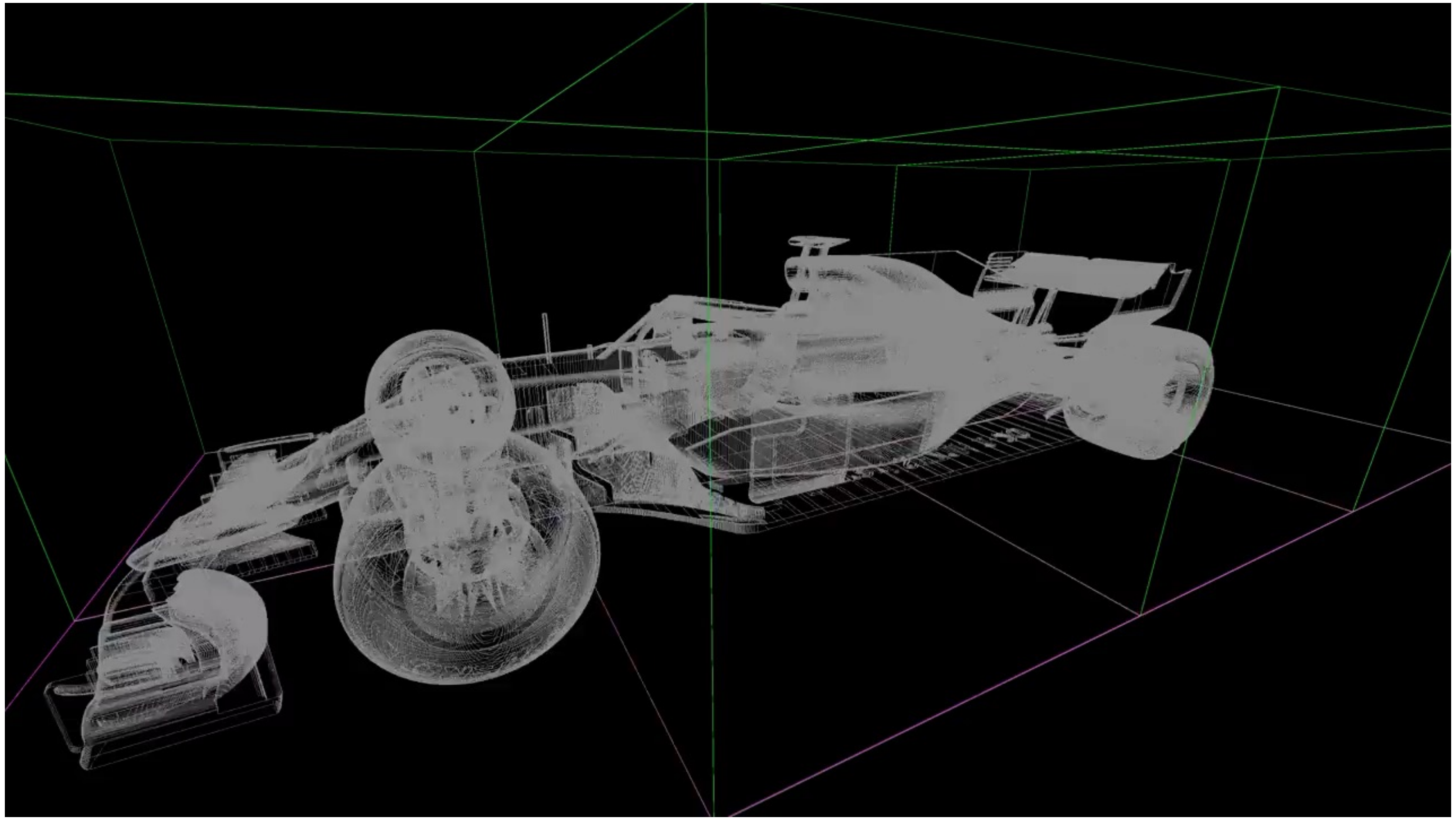
Artificial Intelligence

- Machine Learning is everywhere, needs lots of processing power which supercomputers provide.
- Cars (self-driving)
- Banking (money laundering detection)
- Medicine (cancer diagnosis, brain hemorrhage detection)
- Chihuahua vs muffins
- Volcano mapping

Engineering

Computational Fluid Dynamics

The UNM Racecar Team uses these tools
To design their cars.



Engineering – NASA Mars Lander Rockets

- Computational Fluid Dynamics

- Modelling the rockets that will allow a lander to deliver people safely to Mars.

- Run on the Oakridge National Labs Summit Cluster
- 202,752 cores