

The Design and Structure of a Cluster Supercomputer

Henry Neeman, University of Oklahoma

Director, OU Supercomputing Center for Education & Research (OSCER)

Associate Professor, Gallogly College of Engineering

Adjunct Associate Professor, School of Computer Science

Virtual Residency Introductory Workshop 2023

Tuesday June 27 2023





Outline

- What is Supercomputing?
- What is a Cluster Supercomputer?
- Cluster Supercomputer Components
 - Nodes
 - Interconnection Network
 - Storage
 - Software
 - Facility
- Why Cluster Supercomputers
- The Desert Islands Analogy
- Distributed Parallelism





WARNING!

WARNING! Don't use the word "cluster" by itself in front of users!
Instead, say "cluster supercomputer" or "HPC cluster."
(Or the name of your institution's cluster supercomputer.)

Why?

- The word cluster is **VERY OVERLOADED**.
 - In some research disciplines (or sub-disciplines), the word "cluster" has a completely unrelated meaning that is very familiar to that (sub-)discipline's researchers.
- In a conversation with a researcher, you can't expect the researcher to know which definition of "cluster" you're using.
 - Make yourself easy to understand, not hard!



What is Supercomputing?





What is Supercomputing?

Supercomputing is the **biggest, fastest computing right this minute.**

Likewise, a **supercomputer** is one of the biggest, fastest computers right this minute.

So, the definition of supercomputing is **constantly changing.**

Rule of Thumb: A supercomputer is typically at least 100 times as powerful as a desktop or laptop PC.

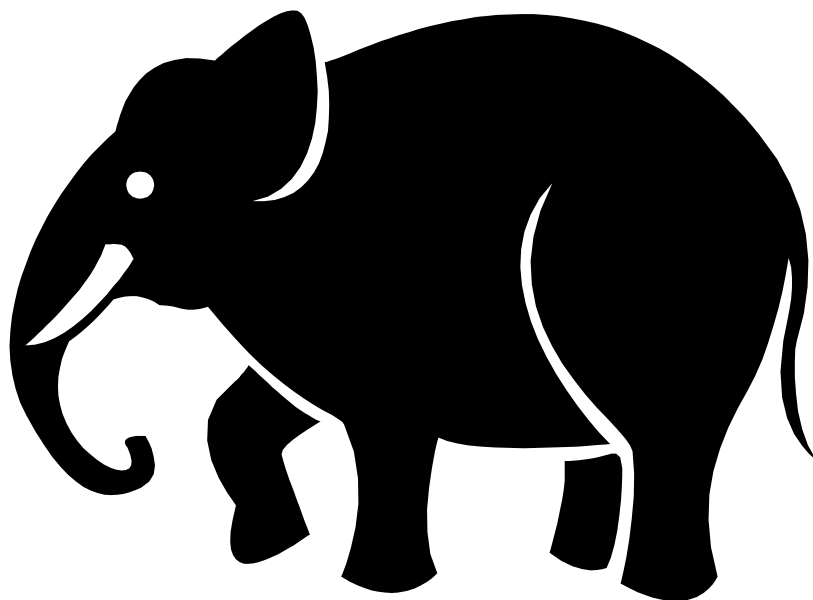
Jargon: Supercomputing is also known as **High Performance Computing (HPC)** or **Cyberinfrastructure (CI, National Science Foundation term).**



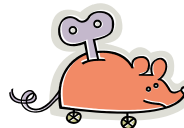
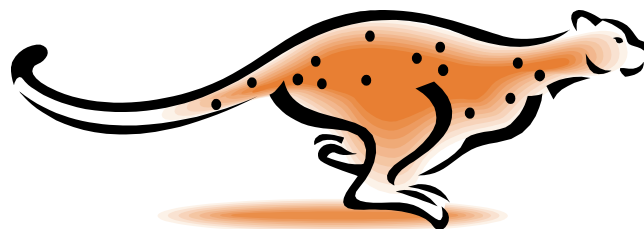


What is Supercomputing About?

Size



Speed



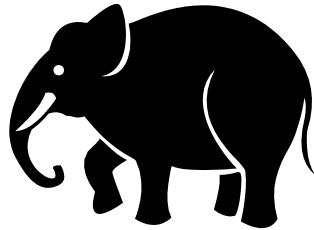
Laptop



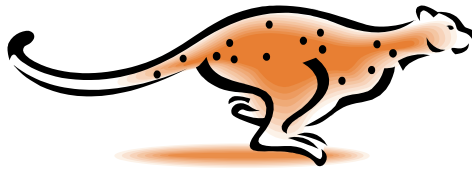


What is Supercomputing About?

- **Size:** Many computational experiments that are interesting to scientists and engineers **can't fit on a PC** – usually because they need more than, say, 8 GB of RAM, or more than, say, a TB of disk.



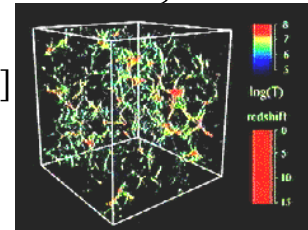
- **Speed:** Many problems that are interesting to scientists and engineers would take a very very long time to run on a PC: months or even years. But a problem that would take **a month on a PC** might take only **an hour on a supercomputer.**





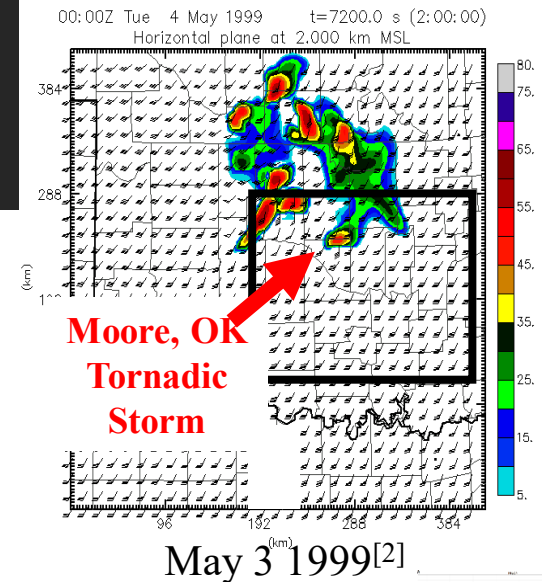
What Is HPC Used For?

- Simulation of physical phenomena, such as
 - Weather forecasting
 - Galaxy formation
 - Oil reservoir management



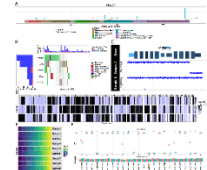
[1]

- Data mining: finding needles of information in a haystack of data, such as
 - Machine Learning (typical behavior)
 - Gene sequencing (specific patterns)
 - Signal processing



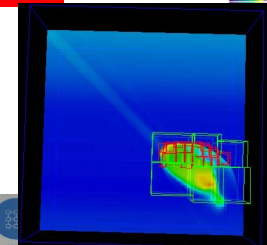
May 3 1999[2]

- Scientific Visualization: turning a vast sea of data into pictures that a researcher can understand



[3a]

[3]





Supercomputing Issues

- The tyranny of the *storage hierarchy*
- *Parallelism*: doing multiple things at the same time



Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023

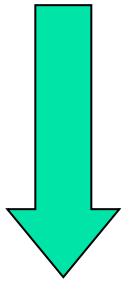




The Storage Hierarchy



Fast, expensive, few



Slow, cheap, a lot

- Registers
- Cache memory
- Main memory (RAM)
- Hard disk
- Removable media (CD, DVD etc)
- Internet



[5]



Speed, Price, Size

Medium	Speed (MB/sec)	Size (MB)	Media Type	Can write to it?	Portable?	Popular?	Drive cost (\$)	Media cost (\$/MB)
Cache	51,200	768	L1/L2/L3	Y	N	Req'd		\$10.5000000
RAM	29,000	33,554,432	DDR4	Y	N	Req'd		\$0.0050000
SSD NVMe	3100	30,720,000	Solid	Y	N	Y		\$0.0007000
USB 3 Flash	625	2,000,000	Solid	Y	Y	Y		\$0.0001000
Hard Disk	150	20,000,000	Mag	Y	N	Y		\$0.0000190
Mag tape	400	18,000,000	Mag	Y	Y	N	\$4751	\$0.0000045

* Maximum among models commonly available for PCs

Note: All numbers are approximate as of Aug 2021 (amazon.com, bestbuy.com, cendyne.com, creativelabs.com, dell.com, ebay.com, floppydisk.com, nextag.com, pcworld.com, rakuten.com, sony.com, toshiba.com, walmart.com, wikipedia.org).

Tape drive is LTO-8, tape cartridge is LTO-7 Type M.



Cluster Supercomputer

Virtual Residency Workshop, Tue June 27 2023

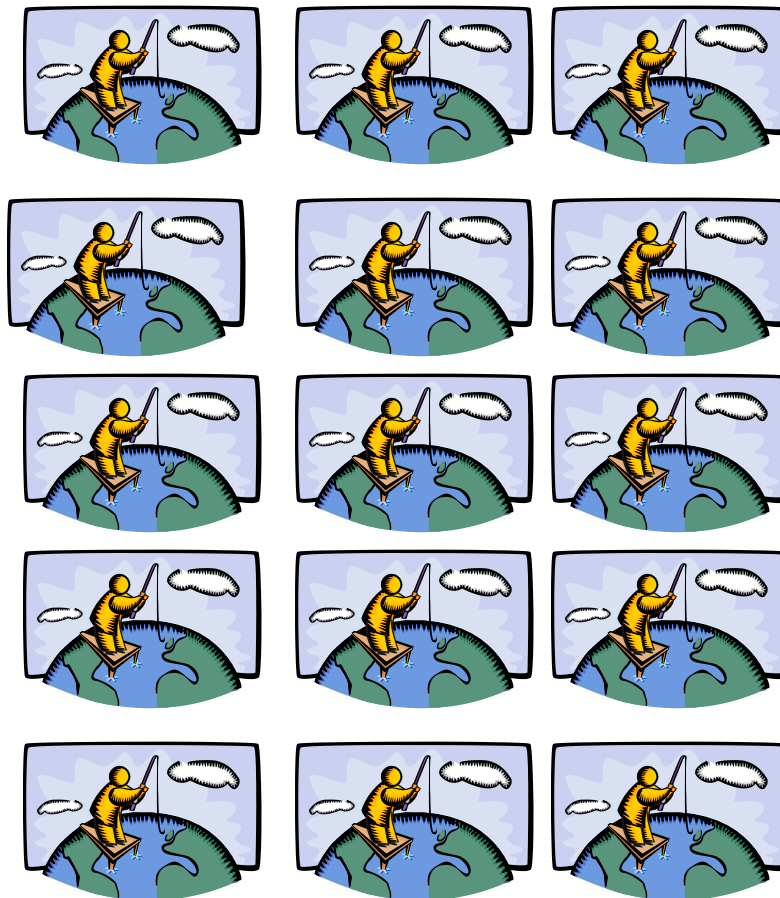




Parallelism

Parallelism means doing multiple things at the same time: you can get more work done in the same time.

Less fish ...



More fish!



What is a Cluster Supercomputer?





What is a Cluster Supercomputer?

“... [W]hat a ship is ... It's not just a keel and hull and a deck and sails – that's what a ship **needs**. But what a ship is ... is freedom.”

– Captain Jack Sparrow

“Pirates of the Caribbean”



http://lh3.ggpht.com/_6hgSmco4R9M/SfpFA3057zI/AAAAAAAAACsg/G-AGCgLrQOk/s1600-h/pirates%5B5%5D.jpg



Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023





What is a Cluster Supercomputer?

A cluster supercomputer **needs**:

- many **small computers**, called **nodes**;
- an **interconnection network** (or **interconnect** for short) for the nodes to communicate with each other;
- **storage** for users' data (and system administration data);
- **software** that allows the nodes to communicate via the interconnect, with each other and with the storage.

A cluster supercomputer **is**:

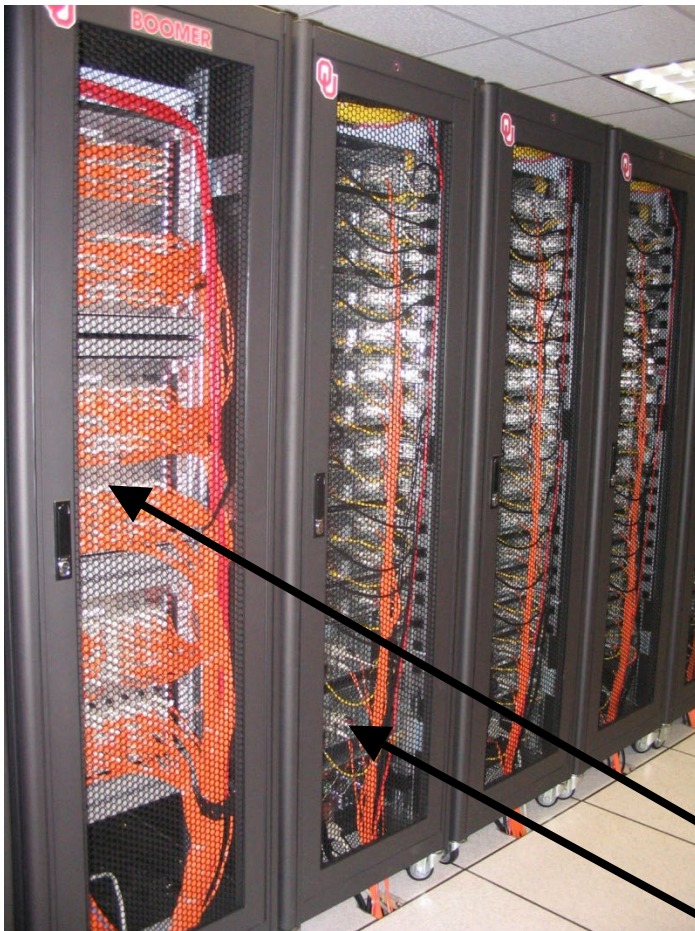
all of these pieces working together

as if they're one big computer ... a **super** computer.

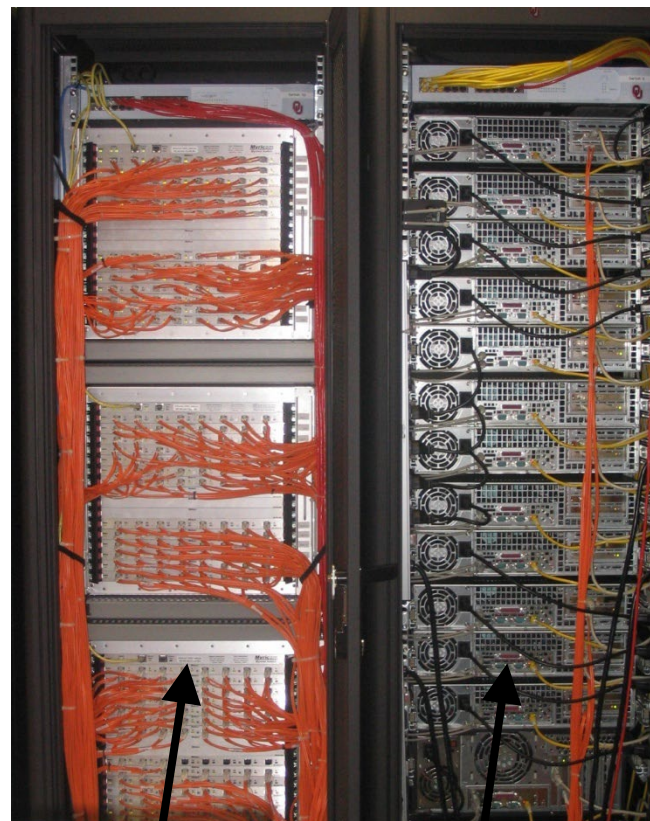




An Actual Cluster Supercomputer



Boomer, in service at OU 2002-5.



Interconnect

Nodes



Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023





What Does a Cluster Supercomputer Cost?

How much money do you have?



Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023





How Much Should You Spend on HPC?

- How important is HPC to your institution's research portfolio?
- How much computing/data-intensive research does your institution have?
 - Don't buy a \$10M supercomputer for a \$1M research portfolio – you'll probably get fired!
 - This rule applies to the total cost of machines and personnel.
 - This should take into account not only your institution's current research portfolio, but also your institution's aspirations.
 - A good range is, your institution's total research computing budget should be between a fraction of a percent to a few percent of your institution's annual research expenditure.
 - If you don't know your institution's annual research expenditure, find out!
<https://ncesdata.nsf.gov/profiles/site?method=rankingBySource&ds=herd>



Cluster Supercomputer Components





Cluster Supercomputer Components

- **Nodes**: Computers (typically server computers) that provide various kinds of functionality (typically, the overwhelming majority are for user number crunching).
- **Interconnect**: Network(s) connecting the nodes to each other, to the storage, to the users, and to the system administrators.
- **Storage**
- **Software**





Cluster Supercomputer Considerations

- Your budget and funding cycle
- Your facility: Space, power, cooling, network (world-facing)
- Your team (labor)
- Your users' workloads
- What else?





Nodes



Cluster Supercomputer Nodes

A **node** is an independent computer (typically a server computer, though in principle it could be a desktop PC or even a cell phone).

- **Compute nodes**: Used for users' number crunching.
- **Non-compute nodes**: Used for purposes other than users' number crunching (for example, system administration purposes).





Compute Nodes

Compute nodes: Optimized for heavy number crunching by users.

- **Regular compute nodes**: CPU-based number crunching with modest to medium RAM per node.
 - On most cluster supercomputers, these are the most common.
 - Goal: Make them as cheap as possible, to maximize how many.
- **Accelerator nodes**: Compute nodes that have accelerators (for example, GPUs) for number crunching (for example, machine learning, traditional floating point).
 - These cost much more than regular compute nodes, so usually few.
- **Large RAM nodes**: Compute nodes for number crunching that needs lots of RAM per node.
 - These cost much more than regular compute nodes, so usually few.





Non-Compute Nodes

Non-Compute Nodes

- Login nodes: Edit source code/scripts/small input files, compile, submit batch jobs, etc.
- Storage nodes: For example, servers full of disk drives (if that's how your cluster implements storage).
- Support/management nodes: Run various administrative tasks:
 - Batch scheduler
 - Disk *imaging*: putting the appropriate contents onto a drive (for example, the operating system)
 - User database
 - ...



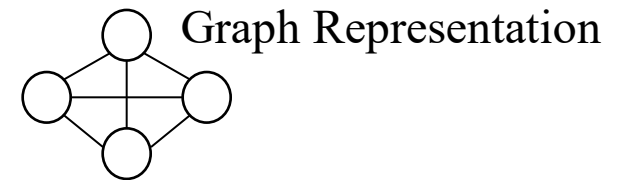
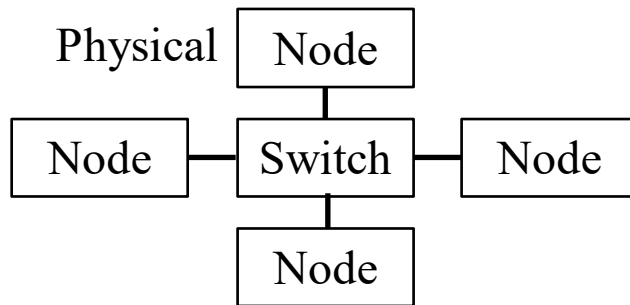


Why Are They Called “Nodes”?

Where did the term “node” come from?

From graph theory: A cluster supercomputer can be represented as a **graph** – a collection of **vertices**, or **nodes**, connected by **edges**.

The vertices are computers, and the edges are network connections.



The people who invented cluster supercomputers were very interested in interconnect network **topologies** (layouts), which can be represented as graphs (in this sense of the word).



Compute nodes

- Regular compute nodes
 - Processor type (Intel x86, AMD x86, ARM, POWER, etc)
 - How many cores per CPU chip?
 - What CPU family/generation?
 - RAM (amount, speed, number of memory channels to fill)
 - Memory bandwidth is maximized if the number of RAM DIMM sticks is a multiple of the number of memory channels times the number of CPU sockets (places to put CPU chips).
 - Network card/cable: Ethernet, Infiniband/Omnipath, proprietary
 - As cheap as possible and still make users happy – buy many!
- Special compute nodes
 - Accelerator nodes (GPUs etc)
 - Large RAM nodes

Choices depend on your users’ needs, your budget, and the price/performance “sweet spot.”





Accelerators (GPUs etc)

Accelerators are devices that can do (certain kinds of) number crunching much faster than CPUs:

- Graphics Processing Units (GPUs)
 - Very popular for Machine Learning calculations
- Field Programmable Gate Arrays (FPGAs)
- Special purpose accelerators

Accelerators typically require special ways of coding.

For some kinds of use (for example, Machine Learning), there are libraries of functions that researchers can build wrapper codes around, so they don't have to learn a special coding approach.

For other kinds of use (for example, traditional general-purpose floating point number crunching), that might not be practical.





Non-Compute: Login & Management Nodes

■ Login nodes

- Need one or a few, enough to handle the number of your users who you expect to be logged in at the same time (and what they'll do).
- Monitor to ensure users aren't running heavy number crunching.
- If you have multiple login nodes, you can have high availability.
 - Log in to a single login node name, but that puts you on one of multiple physical login nodes (via ***round robin*** or whatever).

■ Management nodes (also known as support nodes)

- Where you run all the cluster-wide administrative services.
- Lots of variation in how this is done.
 - How many depends on the size of cluster and the services needed.

■ Storage nodes: Full of disk drives, accessible by everything else.

- Small clusters often have a single server for both login and management (and maybe also storage).





Condominium Supercomputing

A *condominium supercomputing* model allows users to buy their own compute nodes and have them incorporated into the centrally-owned cluster supercomputer.

- The analogy is a condominium apartment building, where the condominium company owns the building, but each individual apartment is owned by its occupant.
- There are many charging models for condominium supercomputing, involving a wide variety of pricing.
 - The owner gets to decide who gets to run on their condominium nodes.
TO
 - The owner gets high priority to run on their condominium nodes (or maybe everywhere).



Interconnection Network





Cluster Supercomputer Interconnect #1

What is a cluster supercomputer's interconnect network **for**?

1. **Node-to-node** communication (for example, MPI parallelism)
2. **Node-to-storage** communication (for example, disk I/O)
3. **System administrator-to-hardware** communication
(sysadmin-to-node, sysadmin-to-storage, sysadmin-to-switch:
management of nodes, storage and networks)

These could be:

- all be on one network, **OR**
 - each on their own network, **OR**
 - a mix,
- depending on your budget and needs.





Interconnect Technology Options

Typical interconnect candidate technologies:

- **Ethernet** (dedicated and internal, not world-facing)
 - Common in almost every context (including cluster supercomputers)
 - Reliable (because very well tested under zillions of circumstances)
 - Relatively high latency
 - ~10 microseconds per message on a dedicated internal network
 - MSRP: 768 nodes @ 25 GE fully subscribed: ~\$1.5M
- **Infiniband/Omnipath** (dedicated and internal, not world-facing)
 - Very rare outside of HPC and a few other application areas
 - Low latency (~1 microsecond) => much faster MPI runs!
 - Much less well tested than Ethernet, because much less popular
 - MSRP: 768 nodes @ HDR100 4:1 oversubscribed: ~\$2.1M (+35%)
- **Proprietary networks**: Typically for a specific vendor's hardware and typically low latency.





Interconnect Considerations

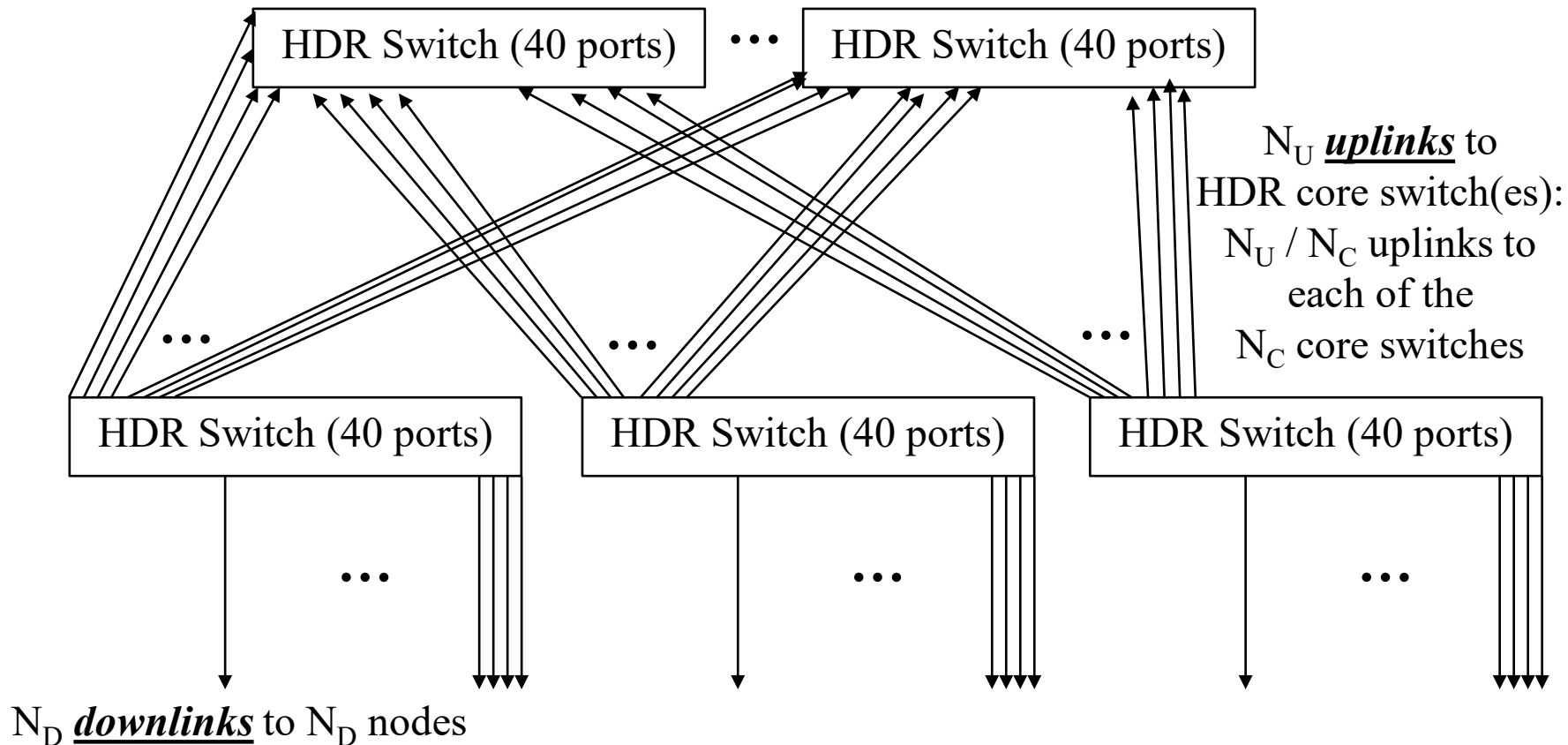
- **Ports per node**: Do your compute nodes need multiple ports each, or just one port? (Usually just one, unless you have \$\$\$.)
- **Bandwidth per node**: How much bandwidth do your users need to the compute nodes, high bandwidth or only medium bandwidth?
 - This depends on whether you expect to have lots of large messages, versus mostly small messages, especially for MPI.
 - 1 microsecond @ 25 Gbps => ~3 KB before bandwidth matters at all
 - Spinning hard disk (HDD) bandwidth typically is modest; SSD bandwidth typically is fast.
 - 1000 HDDs => 150 GB/sec (0.15 GB/sec/HDD)
=> 48 compute nodes max @ 25 Gbps
 - 1000 SSDs => 500 - 3000 GB/sec (0.5 - 3 GB/sec/SSD)
 - **Oversubscription rate**: Derived from the network hardware's bandwidth, combined with the amount of bandwidth needed.
 - e.g., HDR100 4:1 oversubscribed = 25 Gbps when fully saturated





Common Interconnect Topology

For each node, there are N_C possible paths that a message can take to any other node.

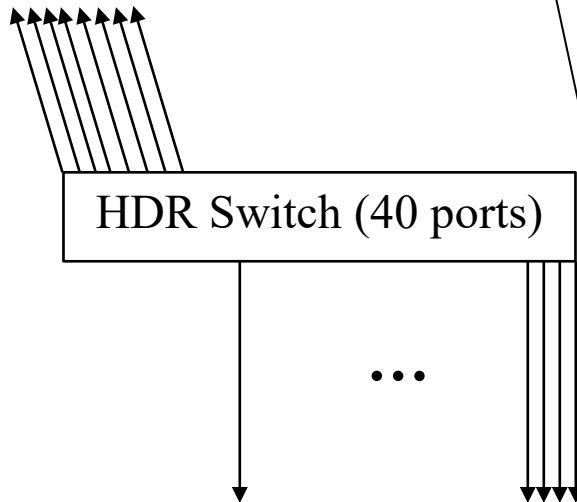




Interconnect Oversubscription

4:1 Oversubscribed

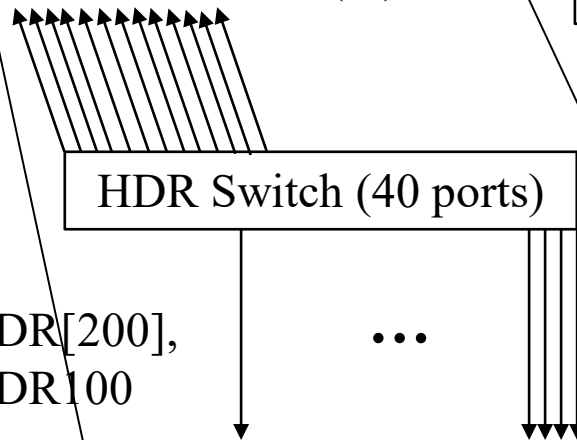
8 Uplinks to HDR core switch(es)



32 Downlinks to 32 nodes @ HDR[200],
32 Downlinks to 64 nodes @ HDR100

2:1 Oversubscribed

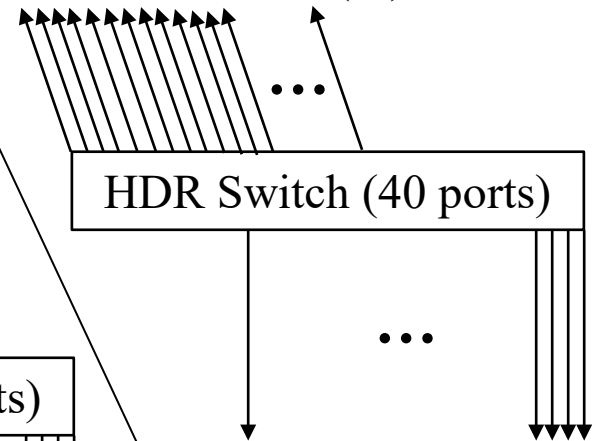
13 Uplinks to HDR core switch(es)



26 Downlinks to nodes

Fully Subscribed

20 Uplinks to HDR core switch(es)



20 Downlinks to nodes





Oversubscription Factors Can Be Weird

- The oversubscription rate might be controlled by weird factors that wouldn't be obvious to people who aren't in the CI business.
 - For example, for Infiniband, copper cables are **MUCH CHEAPER** than fiber cables:
 - HDR splitter passive copper 2 m: \$303 MSRP
 - HDR splitter active copper 3 m: \$1497 MSRP (5.0 times passive)
 - HDR splitter active fiber 3 m: \$2509 MSRP (8.3 times passive)
- https://store.nvidia.com/en-us/networking/store/?page=4&limit=9&locale=en-us&category=INTERCONNECT&technology=InfiniBand&max_speed=HDR
- So, it's **MUCH CHEAPER** to have all your nodes within 2 m of the nearest HDR top-of-rack “leaf” switches.
 - For standard racks, 64 1U compute nodes within 2 m cable run of an HDR switch is plausible, so 4:1 oversubscribed is achievable – barely.





Storage



Storage Priorities

For storage, which of these are your top priorities?

- **Reliability**: If your storage is down, your supercomputer is down.
- **Capacity**: How much storage do your users need?
 - Actual answer: Unlimited!
 - Practical answer: As much as you can afford.
- **Speed**
 - High bandwidth for sequential I/O.
 - Low latency/high IOPS (I/O Operations per Second) rates for random I/O.
- **Price** (including price per TB)

“Big, fast and cheap – pick any two!”





Cluster Storage Hardware

Storage can be one or more of the following:

- Commodity hardware running free storage software
 - A node full of disk drives
 - A disk enclosure directly attached to a node
(***JBOD***: Just a Bunch Of Disks)
- Inexpensive chunks of storage
 - Might operate independently
 - Might be aggregated together
- Proprietary storage system (usually big, fast, reliable and pricey)





Cluster Storage Software

Storage software can be one or more of the following:

- Network File System (NFS): Slow but reliable
 - Well-tested in a broad variety of circumstances
- Free, possibly open source filesystem
 - Examples: Ceph, Lustre, BeeGFS, OrangeFS, etc
- Proprietary filesystem (only runs on one vendor's hardware, in some cases)
 - Examples: Storage Scale/GPFS (IBM), WekaFS (Weka), OneFS (Isilon/EMC/Dell), PanFS (Panasas), **MANY OTHERS**
- Hybrid: Proprietary “appliance” running open source software
 - Examples: DataDirect Networks (Lustre), Omnibond (OrangeFS)





Multiple Storage Filesystems

- **Home**: (very) small, simple (e.g., NFS), dedicated, persistent
 - Scripts, software, small input files – **NOT** bulk datasets.
 - Typically **IS** backed up regularly.
 - Nightly **incremental**: copy all files that are new or changed.
 - Occasional **full dump**: copy all files (weekly, monthly, etc).
- **Scratch**: big, fast, shared, temporary
 - Typically **NOT** backed up.
 - Do you need a parallel filesystem?
 - Do you have sufficient staff and/or 3rd party support (might be pricey)?
 - How big? (Depends on budget, workload and purge policy.)
 - How to prioritize size versus speed versus reliability versus cost?
- **Work/project**: dedicated, persistent storage – **MAYBE**
 - Might be backed up, or might not – It's pricey to back up multiple PB!
- **Archive**: Where's the long-term parking lot for users' big datasets?





Backup, Archive, Mirror

- **Backup**: Nightly “incremental” of all files that are new or changed in the past 24 hours; occasional “full dump” of all files.
 - Destination files of the same filename can exist in multiple versions.
 - Deleting a file from the source **EVENTUALLY** (not immediately) deletes the same file (all versions) from the destination.
- **Archive**: “Write once, read seldom.” On some archives, files have to be uploaded to the archive by the file owner intentionally.
 - Files of the same filename exist in a single (most recent) version only.
 - Deleting a file from the source has zero effect on the destination copy; deleting the destination copy is a manual task by the file owner (and has zero effect on the source copy, if it still exists).
- **Mirror**: Identical source and destination (within a small time lag).
 - One version of the file in each place.
 - Deleting one deletes the other (within a small time lag).





Storage Causes the Most Outages #1

On cluster supercomputers, the number one cause of unscheduled outages is storage problems.

- Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) unscheduled outages Sep 1 2022 - June 22 2023:
 - Storage: 6 of 11 (2 Stampede, 1 Delta, 3 Anvil)
 - ACCESS ticketing system: 2
 - Coolant outage: 1
 - Multifactor Authentication: 1
 - Network: 1

https://operations.access-ci.org/infrastructure_news





Storage Causes the Most Outages #2

On cluster supercomputers, the number one cause of unscheduled outages is storage problems.

- Purdue Rosen Ctr for Advanced Computing unscheduled outages Apr 1 2022 - March 31 2023:
 - Storage: 11 of 20
 - Power/Cooling: 5
 - Remote desktop: 1
 - Open OnDemand: 1
 - Gateway for interactive jobs: 1
 - Data ingress/egress: 1

https://www.rcac.purdue.edu/news/outages-and-maintenance?state=all&order_dir=desc





Multiple Storage Hardware Systems? #1

- What's your budget for storage?
- Where should you put all of your filesystems?
 - On a single physical storage hardware system for all?
 - On a separate physical storage hardware systems for each?
 - Somewhere in between?
- Different users can have different usage patterns:
 - Mostly sequential reads (e.g., some bioscience codes)
 - Sequential => small numbers of large I/O transactions
 - Mostly sequential writes (e.g., transport codes like weather, CFD)
 - Mostly random reads (e.g., machine learning)
 - Mostly random writes (e.g., codes with poorly designed I/O)
- It's very difficult to optimize a single physical storage system for all of these usage patterns.





Multiple Storage Hardware Systems? #2

- What's your budget for storage?
- How high availability do you need?
 - **RAID**: Redundant Array of Independent (or Inexpensive) Disks: Multiple disk drives pretending to be a single big, fast disk drive
 - **RAID 0: Striping**: Any failed drive loses all your data forever: bigger and faster than a single drive, but **FAR LESS RELIABLE**.
 - **RAID 1: Mirroring** (2 identical drives): One failed drive is invisible to users, **BUT** if both drives fail, RAID1 loses all your data forever.
 - **RAID10**: RAID1 + RAID0: Highly resilient but **EXPENSIVE!** (double)
 - **RAID 5**: 1 drive's worth of redundancy: losing 2 drives at the same time loses data and access – a failed drive takes ~1 day of rebuild time per TB of capacity per drive, so losing 2 drives is high probability – **RISKY!**
 - **RAID 6**: 2 drives' worth of redundancy: losing 3 drives at the same time loses data and access – losing 3 drives is **lower** probability.
 - For RAID5 and RAID6, redundancy means extra data to rebuild from (via complicated math).





Multiple Storage Hardware Systems? #3

- What's your budget for storage?
- How high availability do you need (and can you afford)?
 - **ZFS** (originally Zettabyte File System)
 - **RAID-Z**: Like RAID5 (1 drive's worth of redundancy)
 - **RAID-Z2**: Like RAID6 (2 drives' worth of redundancy)
 - **RAID-Z3**: 3 drives' worth of redundancy
 - **Erasure Coding**: Like RAID, but:
 - you can have as many drives' worth of redundancy as you want;
 - rebuild times are much faster (minutes or hours instead of days or weeks).
 - Example: Ceph
 - On Ceph, erasure coding runs half as fast as replication.





Multiple Storage Hardware Systems? #4

- What's your budget for storage?
- If all your filesystems are on the same physical storage hardware system, then a failure of that system leaves your supercomputer completely dead and useless, until you can restart that filesystem.
 - Restart time might depend on when the failure happened (e.g., Xmas).
- If every filesystem is on a separate physical storage system, then a failure on one of them leaves the rest running.
 - If you lose the home filesystem, then all users are dead, so the supercomputer is totally dead anyway.
 - If you lose the scratch filesystem, most users are dead, so the supercomputer is mostly dead anyway.
 - If your supercomputer budget is small, this might be okay – a limited amount of research is affected by such downtimes.





Multiple Storage Hardware Systems? #5

- What's your budget for storage?
- What if you split each of your storage filesystems into multiple separate physical storage hardware subsystems?
 - N_h home storage subsystems, each with $\sim 1/N_h$ of the users.
 - N_s scratch storage subsystems, each with $\sim 1/N_s$ of the users.
 - Or, some scratch subsystems optimized for sequential I/O (bandwidth) and others for random I/O (IOPS = I/O OPerations per Second).
 - If a filesystem fails, the other $\sim (N-1)/N$ users can still run!
- What about redundancy at the physical system level?
 - If you're doing servers full of disk drives, or JBODs attached to servers, this can be relatively affordable.
 - You can have 2 servers full of disk drives mirror each other, but this doubles the cost.
 - How much redundancy do you need, to ensure that a failure doesn't lead to excessive downtime?





Software



Cluster Software

- User applications (science research software)
- Operating System (usually Linux/Unix)
- Batch scheduler
 - Free: e.g., Slurm, OpenPBS, HTCondor
 - Proprietary: e.g., LSF, LoadLeveler, Univa Grid Engine, PBSpro, Moab
- Compilers
- User database
- Provisioning/Disk imaging
- ... and so much more!





Why Batch? #1

- **Q:** Why do we use batch computing on a cluster supercomputer, instead of running interactively like on a normal computer?
- **A:** The same answer as for everything else: **MONEY**.
 - Specifically, a cluster supercomputer is typically much more expensive than a single-user computer – so it needs to be fully subscribed all day every day.
 - Meaning, jobs have to run even when their owners are asleep (and many jobs have long duration, so that'll happen anyway).
 - In most cases, there are many users of the supercomputer, because supercomputers typically are **popular** (if not, they get shut down).
 - For each user, their deadline is the only deadline in the world that matters.





Why Batch? #2

Scenario #1: NO BATCH SYSTEM

- Suppose a user wants to run a job on a supercomputer.
- If there's no batch system, then they'll run that job, interactively, on whichever compute node(s) they want.
- But so will lots of other users, at the same time.
- So the compute nodes will get very slowed down (**thrashing** among the many processes) – maybe the nodes will even crash!
 - Infiniband can be especially sensitive to this.
- In that case, all of those users will be angry.
- Those users will send hate mail to the HPC center director.
 - The director can't help, because there's no batch system!
- So then the users will send hate mail to the director's boss.
- Eventually, the boss will fire the director: too much hate mail!





Why Batch? #3

Scenario #2: Batch by Hand

- Each user e-mails the director a description of how they want to run: runtime duration, number of nodes/cores, amount of RAM, etc.
- The director uses a calendar to determine which job runs on which nodes, and when.
- But, each user's deadline is the only deadline that matters (to them).
- So, they'll ask the director to bump them up in the schedule.
- The director can't do that for everyone all the time, but everyone all the time wants the director to do it – for themselves.
- So, all the users are angry at the director again!
- The boss gets tons of hate mail about the director!
- The director gets fired!





Why Batch? #4

- We need a way to manage all these user requests so that no user gets angry, so that the director won't get fired.
- Who is the only “person” who can do the scheduling without everyone getting angry with them? **SOFTWARE!**
- So, the best solution is a piece of software that does the scheduling for everyone, as fairly as possible.
 - We call this a *batch scheduler*.





How Many Users to Need Batch?

- Q: How many users are needed on a supercomputer in order to need a batch scheduler?
- A: **TWO**.
 - This applies to condominium queues too!





Facility



Facility Considerations

- **Space**: How many Rack Units (RU or just U) do you need?
 - 1U = 2 feet wide by 3 to 4 feet deep by 1.75 inches high
- **Power**: How many KW do you need?
 - You need to get, from the manufacturer, the wattage for each configuration of node, network switch and storage subsystem.
- **Cooling**: How many tons/BTU/KW do you need?
 - 1 **ton** of cooling is the amount of cooling needed to keep 1 ton of dry ice from melting.
 - 1 ton of cooling = 12,000 British Thermal Units (BTUs)
 - 1 ton of cooling = 3.52 KW of cooling <https://www.rapidtables.com/convert/power/ton-to-kw.html>
- **Network**: How much world-facing connectivity (and redundancy) does your supercomputer need?
- **Other**: Fire suppression, weight limit, sizes of hallways and doors, loading dock, elevator, uninterruptible power supply/generator



Why Cluster Supercomputers?





Moore's Law

In 1965, Gordon Moore was an engineer at Fairchild Semiconductor.

He noticed that the number of transistors that could be squeezed onto a chip was doubling about every 2 years.

It turns out that computer speed, and storage capacity, is roughly proportional to the number of transistors per unit area.

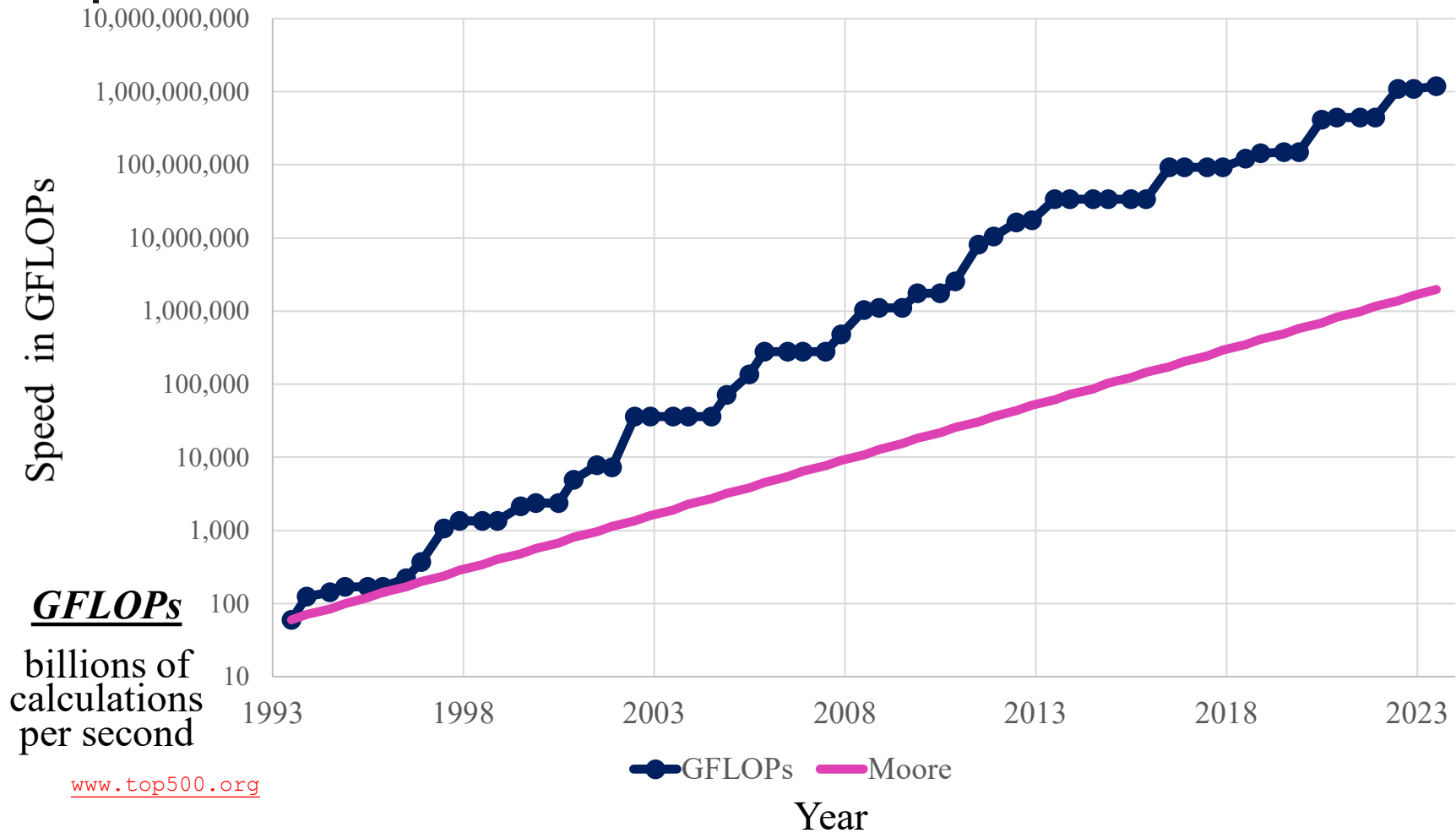
Moore wrote a paper about this concept, which became known as “*Moore's Law.*”

(Originally, he predicted a doubling every year, but not long after, he revised that to every other year.)

G. Moore, 1965: “Cramming more components onto integrated circuits.” *Electronics*, 38 (8), 114-117.



Fastest Supercomputer vs. Moore



www.top500.org

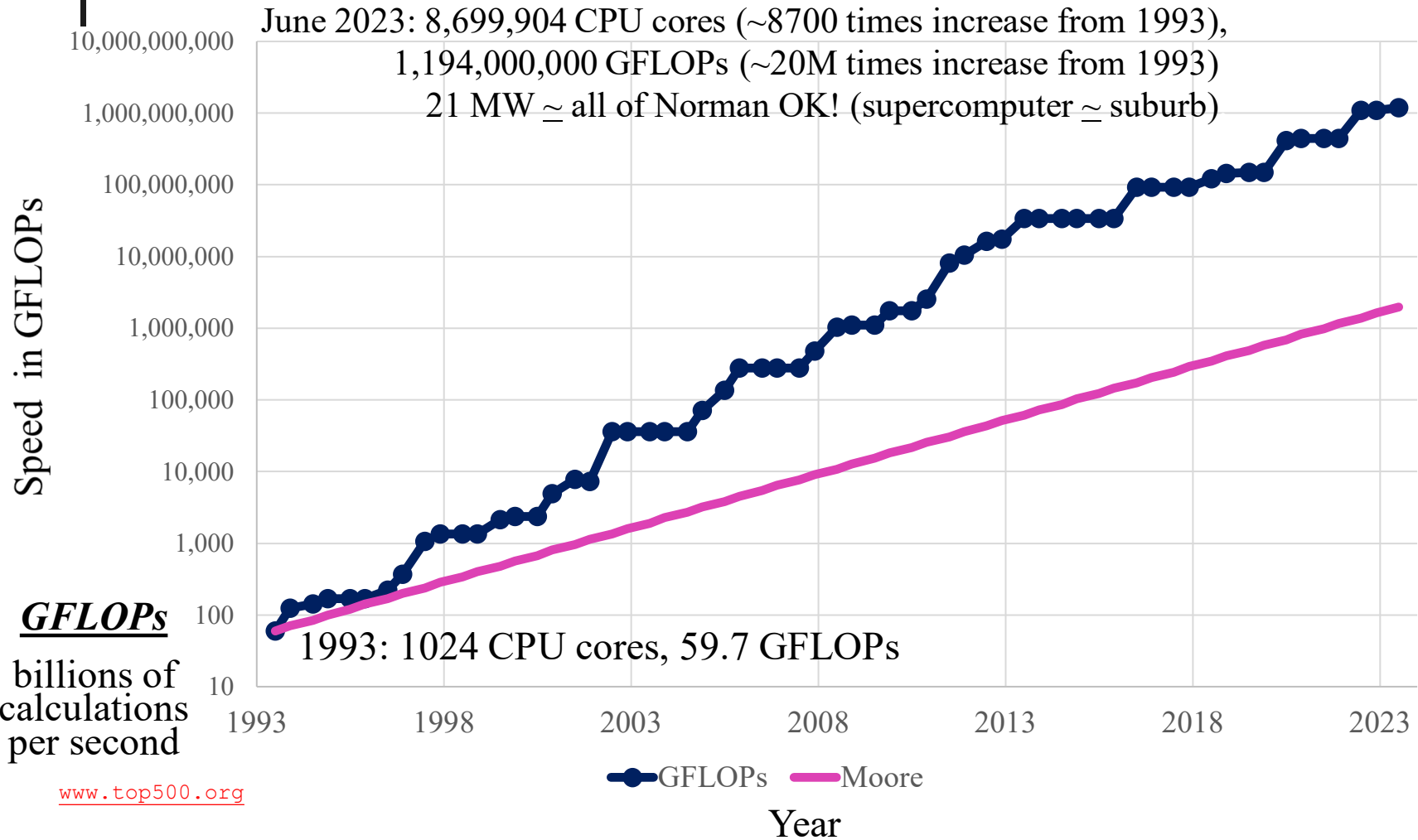


Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023





Fastest Supercomputer vs. Moore



www.top500.org



Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023





Moore: Uncanny!

- Nov 1971: Intel 4004 – 2300 transistors
- March 2010: Intel Nehalem Beckton – 2.3 billion transistors
- Factor of 1,000,000 improvement in 38 1/3 years
- $2^{(38.33 \text{ years} / 1.9232455)} = 1,000,000$

So, transistor density has doubled every 23 months:

UNCANNILY ACCURATE PREDICTION!



Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023





Is Moore's Law Dead?

- Moore's Law isn't dead yet:
 - CPU chip speed improved from 53 GFLOPs per CPU chip on Intel Nehalem W5590 in 2009^[13] to 3200 GFLOPs per CPU chip on Intel Ice Lake 8380 in 2021^[14], a doubling period of 24 months.
 - Memory bandwidth went from, for example, ~36 GB/sec on dual Intel Nehalem W5580 in 2009^[15,16] to ~766 GB/sec in late 2022 on dual AMD EPYC Genoa^[17], a doubling period of 35 months.





The History of 1 TFLOPs

1 TFLOPs: 1 trillion calculations per second

1997: Room



ASCI RED
Sandia National Lab
https://micro.medium.com/max/1400/1*Svvv35e0-FO_hifchK722g.webp

2002: Row



boomer.oscer.ou.edu
In service 2002-5: 11 racks

2012: Card



AMD
FirePro
W9000

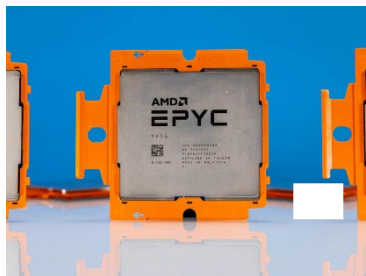


NVIDIA
Kepler K20



Intel MIC
Xeon PHI

2022: ~8 cores



<https://www.servethehome.com/wp-content/uploads/2022/11/AMD-EPYC-9654-romeo-CPU-1.jpg>

2017: Chip



AMD Naples



Intel Skylake



INFORMATION TECHNOLOGY
The UNIVERSITY of OKLAHOMA

Cluster Supercomputer

Virtual Residency Workshop, Tue June 27 2023



Multicore

- A ***multicore*** CPU is a chip with multiple, independent “brains,” known as ***cores***.
- These multiple cores can run completely separate programs, or they can cooperate together to work simultaneously in parallel on different parts of the same program.
- All of the cores share the same connection to memory – and the same ***bandwidth*** (memory speed).

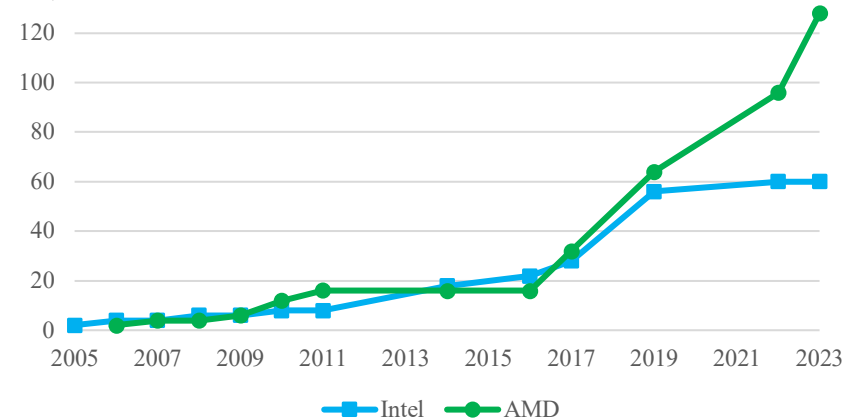




Multicore History (x86)

- Single core: November 1971 (Intel 4004)
- 2-core: October 2005 (Intel), March 2006 (AMD)
- 4-core: June 2006 (Intel), Sep 2007 (AMD)
- 6-core: Sep 2008 (Intel), June 2009 (AMD)
- 8-core (Intel & AMD): March 2010
- 12-core (AMD only): March 2010
- 16-core: Nov 2011 (AMD only)
- 18-core: Sep 2014 (Intel only)
- 22-core: March 2016 (Intel only)
- 28-core: July 2017 (Intel only)
- 32-core: June 2017 (AMD only)
- 56-core: Apr 2019 (Intel only)
- 64-core: Aug 2019 (AMD only)
- 96-core: Nov 2022 (AMD only)
- 128-core: June 2023 (AMD only)

CPU Cores per Chip



Note that this is only for x86 – other CPU families (for example, POWER) introduced multicore earlier.

- <http://www.intel.com/pressroom/kits/quickreffam.htm> (dual core, quad core)
- <http://ark.intel.com/products/family/34348/Intel-Xeon-Processor-7000-Sequence#@Server> (hex core)
- <http://ark.intel.com/ProductCollection.aspx?familyID=594&MarketSegment=SRV> (oct core)
- [http://en.wikipedia.org/wiki/Intel_Nehalem_\(microarchitecture\)](http://en.wikipedia.org/wiki/Intel_Nehalem_(microarchitecture)) (oct core)
- http://en.wikipedia.org/wiki/AMD_Opteron (12-core)
- [https://en.wikipedia.org/wiki/Broadwell_\(microarchitecture\)](https://en.wikipedia.org/wiki/Broadwell_(microarchitecture)) (22-core)
- [https://en.wikipedia.org/wiki/Skylake_\(microarchitecture\)](https://en.wikipedia.org/wiki/Skylake_(microarchitecture)) (28-core)
- [https://en.wikipedia.org/wiki/Cascade_Lake_\(microarchitecture\)](https://en.wikipedia.org/wiki/Cascade_Lake_(microarchitecture)) (56-core)
- <https://en.wikipedia.org/wiki/Epyc> (128-core, 96-core, 64-core, 32-core)





Why Multicore? #1

- In the golden olden days (through about 2005), the way to speed up a CPU was to increase its “clock speed.”
 - Every CPU has a little crystal inside it that vibrates at a fixed frequency (for example, 1 GHz = 1 billion vibrations per second).
 - Each operation (add, subtract, multiply, divide, etc) requires a specific number of ticks of that clock (“clock cycles”) to complete.
- But, the power density (watts per square centimeter) of a CPU chip is **proportional to the square** of the clock speed.
- So, continuing to increase the clock speed would have been, quite literally, a dead end, because by now such CPU chips would have already reached the power density of the sun.





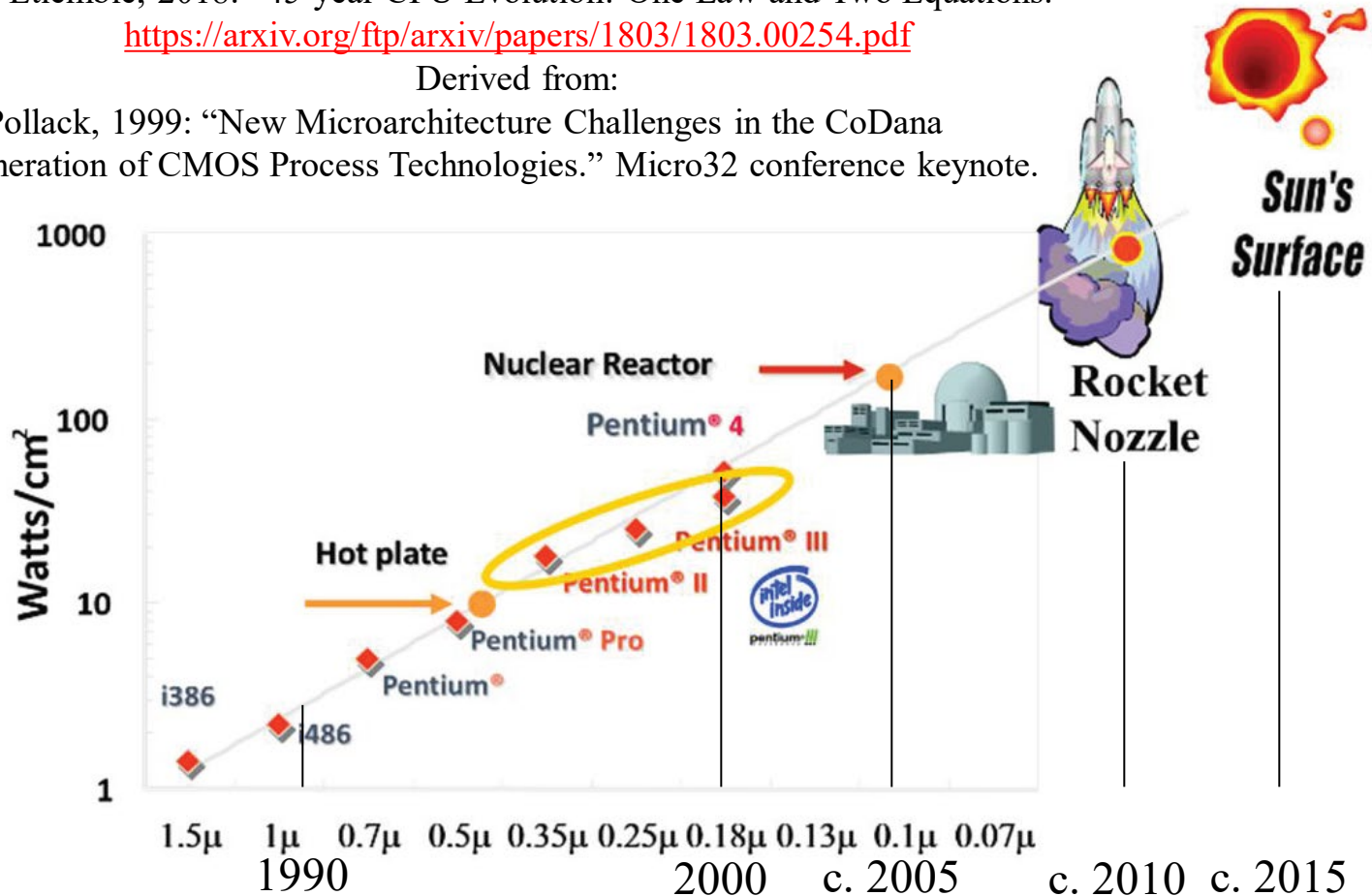
Why Multicore? #2

D. Etiemble, 2018: "45-year CPU Evolution: One Law and Two Equations."

<https://arxiv.org/ftp/arxiv/papers/1803/1803.00254.pdf>

Derived from:

F. Pollack, 1999: "New Microarchitecture Challenges in the CoDana Generation of CMOS Process Technologies." Micro32 conference keynote.



If we had kept increasing clock speed



Cluster Supercomputer
Virtual Residency Workshop, Tue June 27 2023





Why Cluster Supercomputers? #1

- Once upon a time (1970s through 1990s), most supercomputers were ***big iron***: proprietary, boutique hardware that was specifically designed to be huge and fast.
 - At that time, you could build a chip factory reasonably affordably.
- Nowadays, the cost of building a chip factory is huge: \$5B+!
 - Just to break even, you've got to sell zillions of CPUs from that factory!
 - So there's no way for a boutique CPU flavor to be cost effective.
 - "To make a small fortune in supercomputing, start with a large fortune."
- Cluster supercomputers work great with Commodity Off-The-Shelf (COTS) parts; big iron doesn't.
 - Big iron can't compete financially with cluster supercomputers!





Why Cluster Supercomputers? #2

- Supercomputing works best with lots of parallelism.
- It's cheaper to buy lots of commodity computers and hook them together with an interconnect network than to design and build a big iron machine that uses proprietary chips.
- So the most cost effective approach to supercomputing is clusters.





Why Cluster Supercomputers? #3

- Research application software is usually written in a few standard languages (Fortran, C, C++, Python)
 - Sometimes in “vanilla” ways (which usually compile and run okay).
 - Sometimes in non-vanilla ways (which can be much harder to compile and run).
- Most of these codes have been ported successfully to x86 CPUs (Intel, AMD), and compile and run well on x86.
- Many of these codes haven’t been ported to other CPU families (e.g., ARM, POWER).
- Probably most of them will port okay, but not all of them – and it can take a **lot** of labor to find out.
- Many institutional HPC centers support hundreds of research codes, and national HPC centers support thousands of them.



The Desert Islands Analogy





How Does a Cluster Supercomputer Work?

- It's easy to imagine how a job runs if it only needs a single core, or even an entire node.
- But what if the job needs **multiple nodes** running in parallel, for example to run a much bigger problem that could fit on just one node?





An Island House

- Imagine you're on an island in a little house.
- Inside the house is a desk.
- On the desk is:
 - a phone;
 - a pencil;
 - a calculator;
 - a piece of paper with instructions
 - a piece of paper with numbers (data).



DATA

1. 27.3
2. -491.41
3. 24
4. -1e-05
5. 141.41
6. 0
7. 4167
8. 94.14
9. -518.481
- ...

Instructions: What to Do

```
...
Add the number in slot 27 to the number in slot 239,
and put the result in slot 71.
if the number in slot 71 is equal to the number in slot 118 then
  Call 555-0127 and leave a voicemail containing the number in slot 962.
else
  Call your voicemail box and collect a voicemail from 555-0063,
  and put that number in slot 715.
...
```





Instructions

The instructions are split into two kinds:

- Arithmetic/Logical – for example:
 - Add the number in slot 27 to the number in slot 239, and put the result in slot 71.
 - Compare the number in slot 71 to the number in slot 118, to see whether they are equal.
- Communication – for example:
 - Call 555-0127 and leave a voicemail containing the number in slot 962.
 - Call your voicemail box and collect a voicemail from 555-0063, and put that number in slot 715.





Is There Anybody Out There?

If you're in a house on an island,
you **aren't specifically aware** of anyone else.

Especially, you don't know whether anyone else is working on
the same problem as you are, and you don't know who's at
the other end of the phone line.

All you know is what to do with the voicemails you get, and
what phone numbers to send voicemails to.





Someone Might Be Out There

Now suppose that Dana is on another island somewhere, in the same kind of house, with the same kind of equipment. Suppose that Dana has the same list of instructions as you, but a different set of numbers (both data and phone numbers). Like you, Dana doesn't know whether there's anyone else working on the same problem.





Even More People Out There

Now suppose that Pat and Lee are also in houses on islands.

Suppose that each of the four has the exact same list of instructions, but different lists of numbers.

And suppose that the phone numbers that people call are each others': that is, your instructions have you call Dana, Pat and Lee, Dana's calls Pat, Lee and you, and so on.

Then you might all be working together on the same problem.





All Data Are Private

Notice that you can't see Dana's or Pat's or Lee's numbers, nor can they see yours or each other's.

Thus, everyone's numbers are private: there's no way for anyone to share numbers, except by leaving them in voicemails.





Long Distance Calls: 2 Costs

When you make a long distance phone call, you typically have to pay two costs:

- **Connection charge**: the **fixed** cost of connecting your phone to someone else's, even if you're only connected for a second
- **Per-minute charge**: the cost per minute of talking, once you're connected

If the connection charge is large, then you want to make as few calls as possible.

See:

<http://www.youtube.com/watch?v=8k1UOEYIQRo>



Distributed Parallelism





Like Desert Islands

Distributed parallelism is very much like the Desert Islands analogy:

- processes are independent of each other.
- All data are private.
- Processes communicate by passing messages (like voicemails).
- The cost of passing a message is split into:
 - latency (connection time)
 - bandwidth (time per byte)





Latency vs Bandwidth on Schooner

In 2018, a benchmark of the Infiniband interconnect on the University of Oklahoma's Linux cluster revealed:

- **Latency** – the time for the first bit to show up at the destination – is ~ 1.26 microseconds;
- **Bandwidth** – the speed of the subsequent bits – is ~ 37.2 Gigabits per second (~ 0.027 nanosec per bit).

Thus, on OU's cluster Infiniband:

- the first bit of a message shows up in ~ 1260 nanosec;
- the last bit of a message shows up in ~ 0.027 nanosec.

So latency is $\sim 47,000$ times worse than bandwidth!





Latency vs Bandwidth on Schooner

In 2018, a benchmark of the Infiniband interconnect on the University of Oklahoma's Linux cluster revealed:

- **Latency** – the time for the first bit to show up at the destination – is ~ 1.26 microseconds;
- **Bandwidth** – the speed of the subsequent bits – is ~ 37.2 Gigabits per second (~ 0.027 nanosec per bit).

Thus, on OU's cluster Infiniband:

- the first bit of a message shows up in ~ 1260 nanosec;
- the last bit of a message shows up in ~ 0.027 nanosec.

So latency is **$\sim 47,000$ times worse** than bandwidth!

That's like having a long distance service that charges:

- \$470 to make a call at all, regardless of duration;
- 1¢ per minute – after the **first 33 days** on the call.



**Thanks for your
attention!**



Questions?

www.oscer.ou.edu