ECE566
Enterprise Storage Architecture

Spring 2025

Course Overview and Introduction

Tyler Bletsch
Duke University

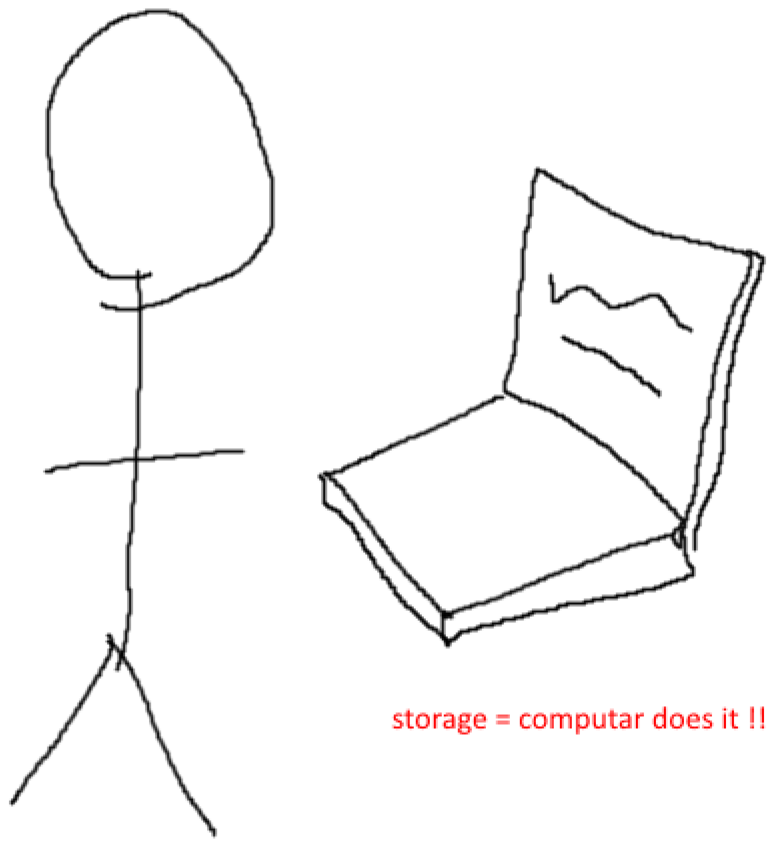
Slides include material from Vince Freeh (NCSU)

Instructor

- Professor: Tyler Bletsch
  - Office: Wilkinson 103
  - Email: Tyler.Bletsch@duke.edu
  - Office Hours: See course site

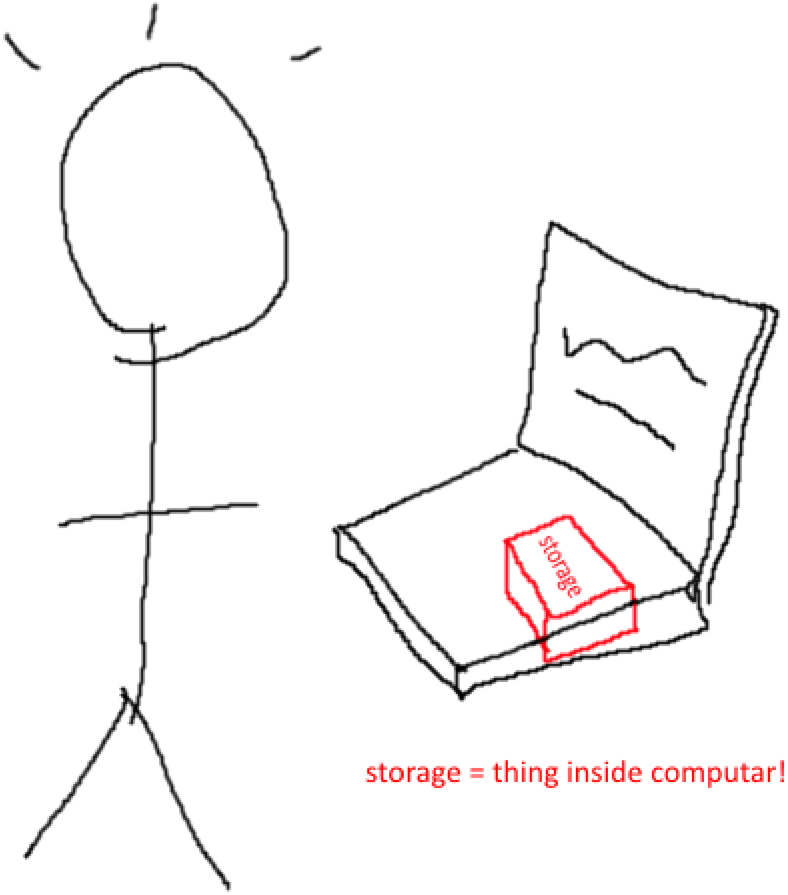
MOTIVATION

Average person's view of storage



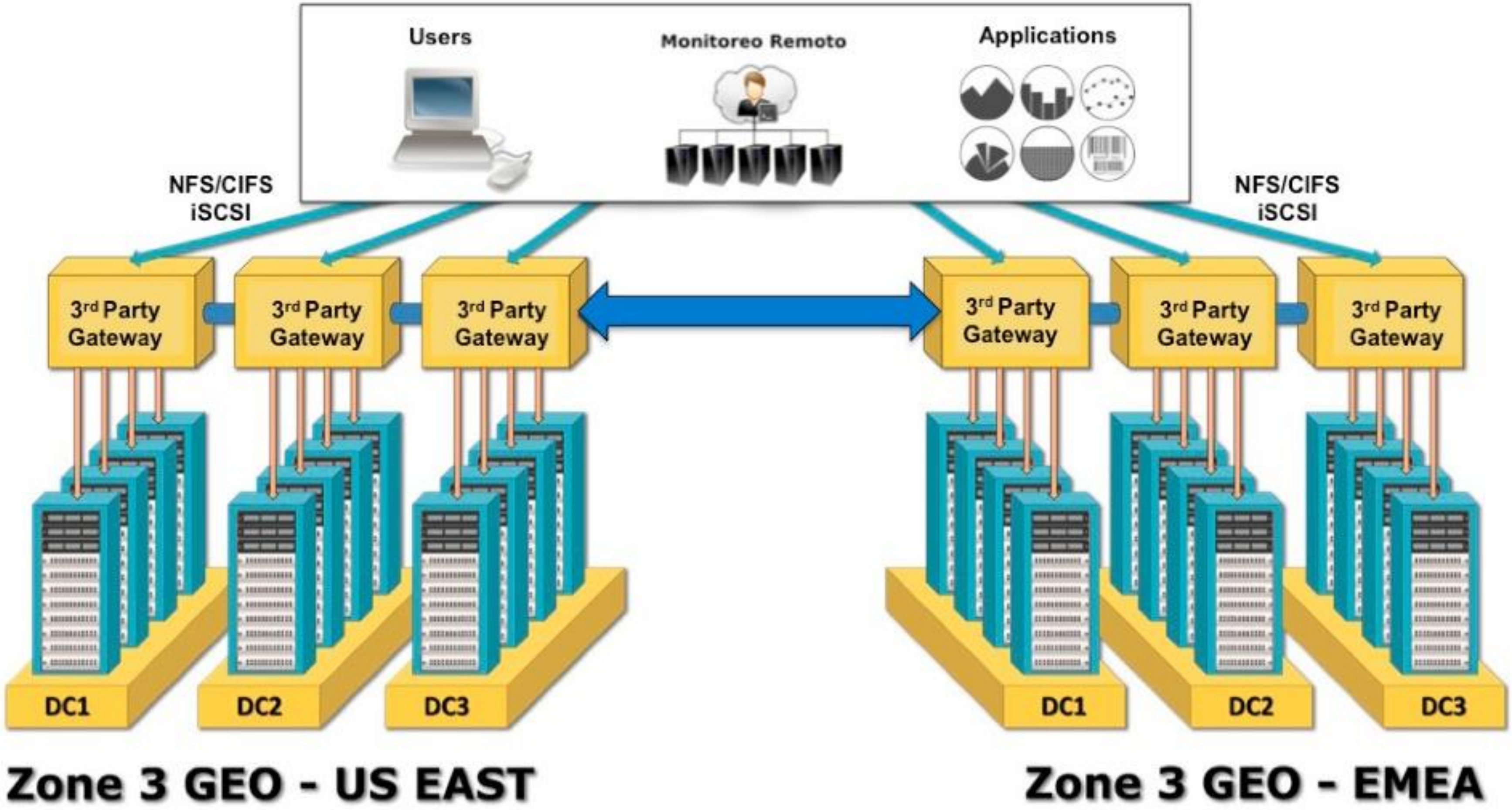
storage = computar does it !!

Average engineer’s view of storage



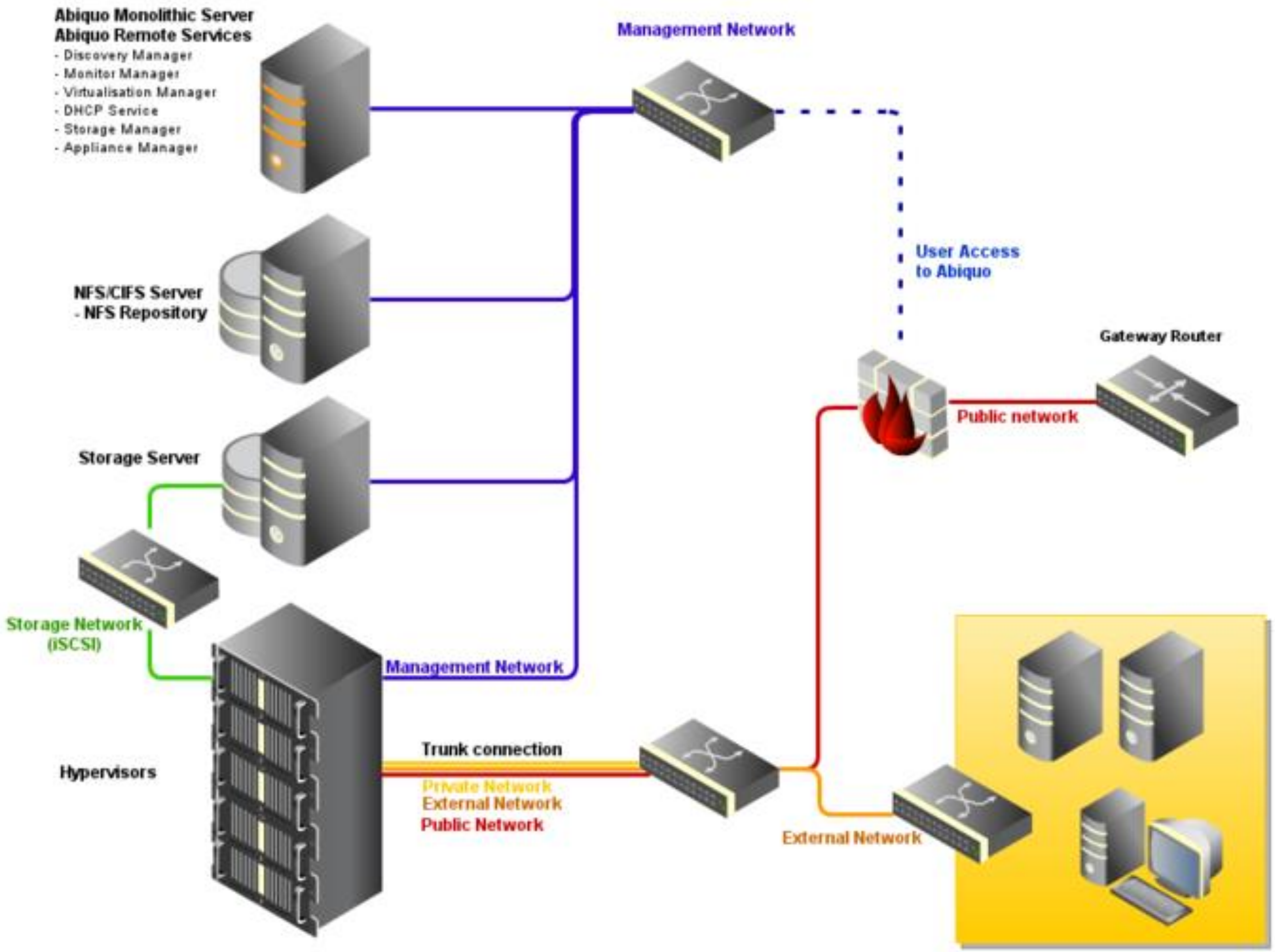
storage = thing inside computar!!

A few enterprise storage architectures (1)



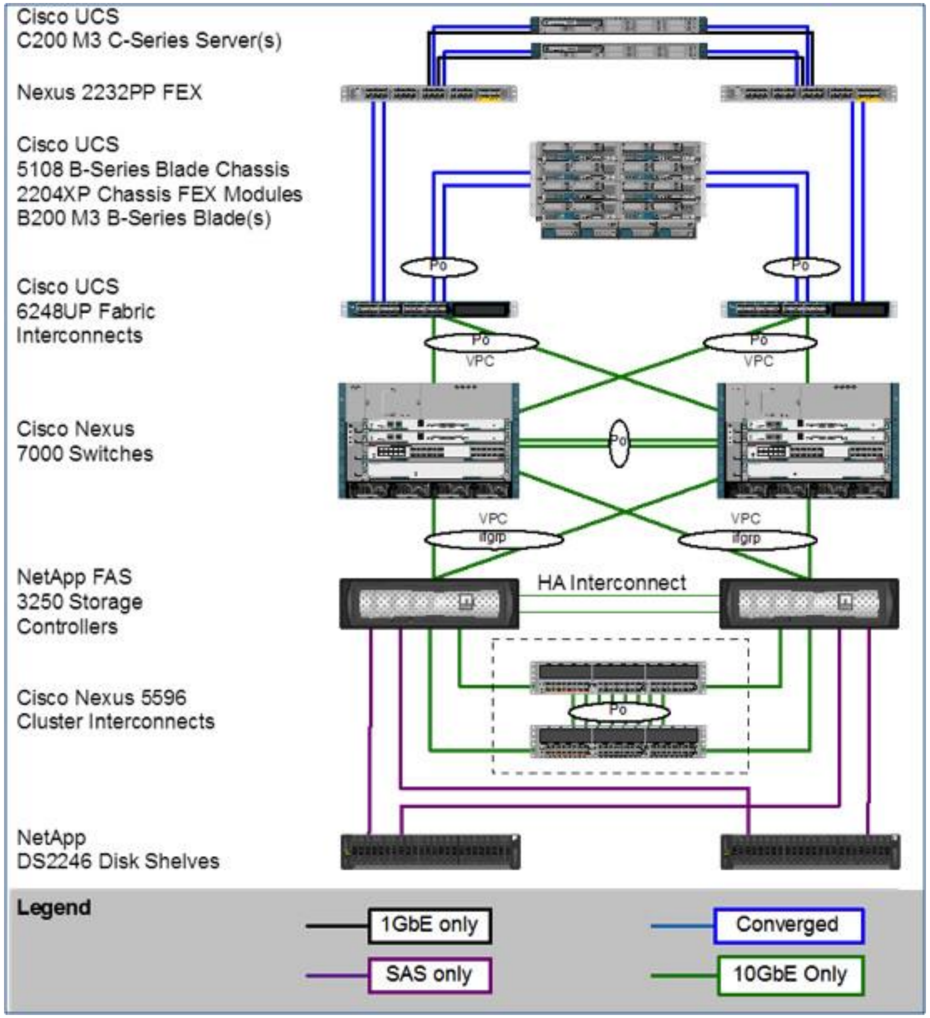
- From: http://www.storagenewsletter.com/rubriques/software/massively-scalable-himalaya-architecture-by-amplidata/

A few enterprise storage architectures (2)



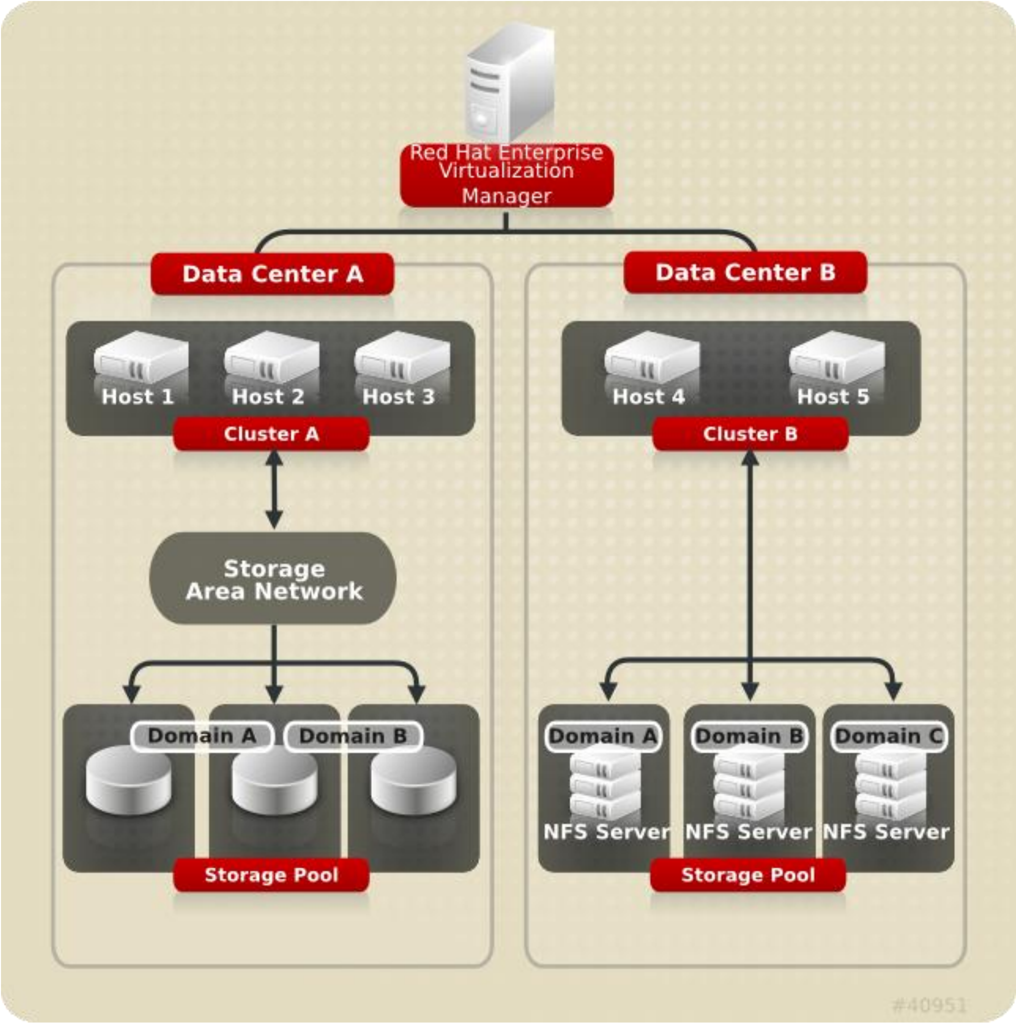
- From: http://wiki.abiquo.com/display/ABI20/Monolithic+Architecture

A few enterprise storage architectures (3)



- From: http://community.netapp.com/t5/Tech-OnTap-Articles/FlexPod-Innovation-and-Evolution/ta-p/85156

A few enterprise storage architectures (4)



- From: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Virtualization/3.0/html/Technical_Reference_Guide/chap-Technical_Reference_Guide-Storage_Architecture.html

Why do all this? What problems are we solving?

- **Cost**: Is it cheap enough?
- **Capacity**: Can it hold enough?
- **Performance**: Is it fast enough?
- **Accessibility**: Can the data be accessed by everyone who needs it?
- **Security**: Is data protected from unauthorized access?
- **Reliability**: Is the downtime probability low enough?
- **Integrity**: Is data protected from hardware failures, disasters, and malicious attacks?
- **Compliance**: Do I keep data long enough safely?
- **Accountability**: Can I track all changes?
- **Space efficiency**: How much floor space do I need?
- **Power efficiency**: How many watts do I burn?

Why do all this? What problems are we solving?

- **Cost**: Is it cheap enough?
- **Capacity**: Can it hold enough?
- **Performance**: Is it fast enough?
- **Accessibility**: Can the data be accessed by everyone who needs it?
- **Security**: Is data protected from unauthorized access?
- **Reliability**: Is the downtime probability low enough?
- **Integrity**: Is data protected from hardware failures, disasters, and malicious attacks?
- **Compliance**: Do I keep data long enough safely?
- **Accountability**: Can I track all changes?
- **Space efficiency**: How much floor space do I need?
- **Power efficiency**: How many watts do I burn?

Color code: how well can a simple drive in a laptop let you control these variables?

Online course Info

- Course Web Page: static info

➡ **https://people.duke.edu/~tkb13/courses/ece566-2025sp/**

  - Syllabus, schedule, slides, assignments, rules/policies, prof/TA info, office hour info
  - Links to useful resources

- **Ed forum**: questions/answers
  - Post all of your questions here
  - Questions must be "public" unless good reason otherwise
  - **No code** in public posts!
- **GradeScope**: Submit annotated PDFs for grading
- **Canvas**: just assignment submission and gradebook

Where to get info

- This info is fairly industry-connected, no great textbook
  - Semi-exception: “Evolution of the Storage Brain” by Larry Freeman (not a required text)

- Course material will come from lectures and supplementary readings
  - See course site for resources

- Additional independent research on your part will likely be necessary!

Grading Breakdown

Project: 40%

Category	%
Project proposal draft	2%
Project proposal	3%
Project milestone 1	5%
Project milestone 2	5%
Project final report	5%
Project final presentation	5%
Project final demo	15%
Homeworks/programs/labs	40%
Midterm exam	10%
Final exam	10%

HOMEWORKS, LABS, AND PROGRAMS

Lab Motivation: What is a computer?

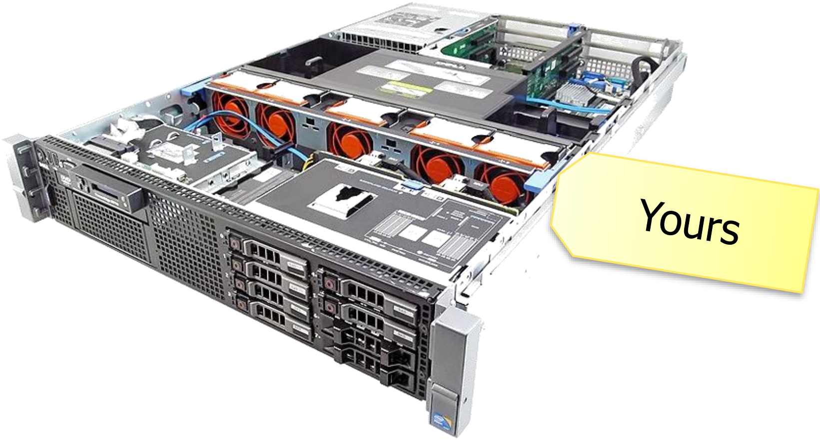
- Computers are:
  - ~~Abstract theoretical math engines that float around on the internet?~~

  - PHYSICAL OBJECTS
  - MADE OF MATERIALS
  - IN THE REAL WORLD
  - AND YOU CAN TOUCH THEM
  - AND PUT THEM PLACES
  - WITH YOUR ARMS/LEGS/FINGERS/BODY
  - AND LIKE A SCREWDRIVER OR WHATEVER!!!!!!!!

Result: this course is HANDS ON

- Historically, the most popular assignments have been the realistic, hands-on ones. So I've added a *lot* of hands-on experience to the course.

- Each student group will be assigned a physical storage server which is upstairs in Hudson 214
- **Lab 0** will have you prepare and deploy this server.
- **Labs 1+** will have you do realistic storage tasks on it.



Labs vs Homeworks

Labs

- Group work
- Hands-on
- Usually on your server
- Submitted via GradeScope (and Sakai for code)
- Can discuss concepts with other groups, but not answers

Homeworks

- Individual work
- Pen-and-paper questions
- Submitted via GradeScope (and Sakai for code)
- Can discuss concepts with others, but not answers

Also: a few "Program" assignments

- Programs will involve writing system code using BUSE (Block device in USErspace) and FUSE (Filesystem in USErspace)

- Assignments "Program 0" (BUSE) and "Program 1" (FUSE) are <u>individual</u>
  - Prepare each individual for the late-semester group project

Late penalties

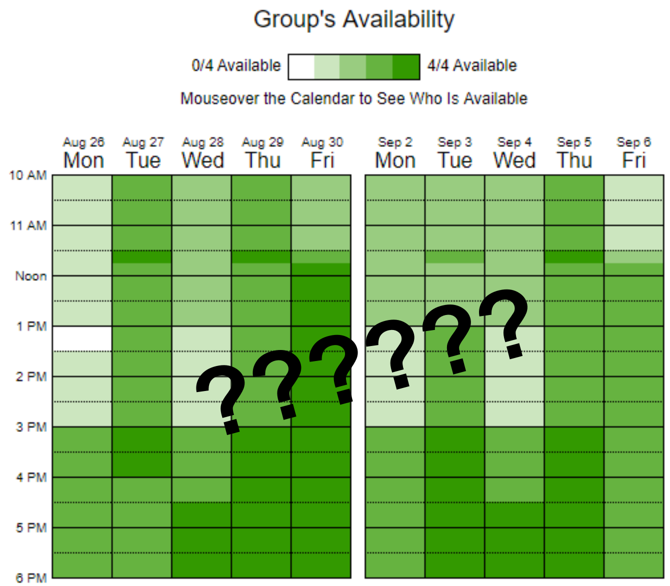
- Late homework/lab/program incur penalties as follows:
  - Submission is 0-24 hours late: total score is multiplied by 0.9
  - Submission is 24-48 hours late: total score is multiplied by 0.8
  - Submission is more than 48 hours late: total score is multiplied by the Planck constant (in J·s)

- NOTE: If you feel in advance that you may need an extension, contact the instructor.

Class lab sessions to kickstart homework

- We're going to schedule a few **class-wide lab sessions** so everyone can start to work on their server with instructor support
  - Why not a separate lab section? We don't need every week...

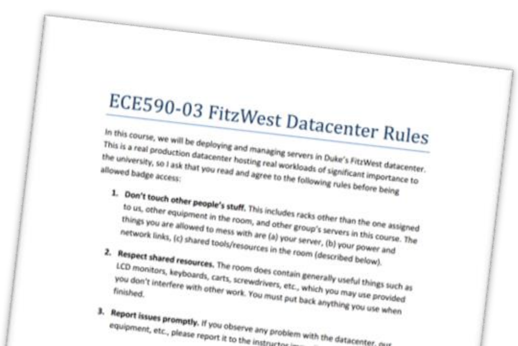
- Be sure to respond to the **scheduling survey** that I sent; deadline is tomorrow!



FitzWest Datacenter

- You will eventually deploy your server in a real datacenter: the FitzWest server room in the CIEMAS basement
  - This means you'll have **badge access** to a real datacenter

- Datacenter rules (you need to sign this to get access):
  1. **Don't touch other people's stuff.** Includes other racks, other equipment, and other group's servers in this course. You can touch your server, its cables, and shared tools.
  2. **Respect shared resources.** The room has LCD monitors, keyboards, carts, screwdrivers, etc., which you can use. You must not interfere with IT operations and you must put stuff away when done.
  3. **Report issues promptly.** Tell me if anything's wrong.



ECE590-03 FitzWest Datacenter Rules

In this course, we will be deploying and managing servers in Duke's FitzWest datacenter. This is a real production datacenter hosting real workloads of significant importance to the university, so I ask that you read and agree to the following rules before being allowed badge access:

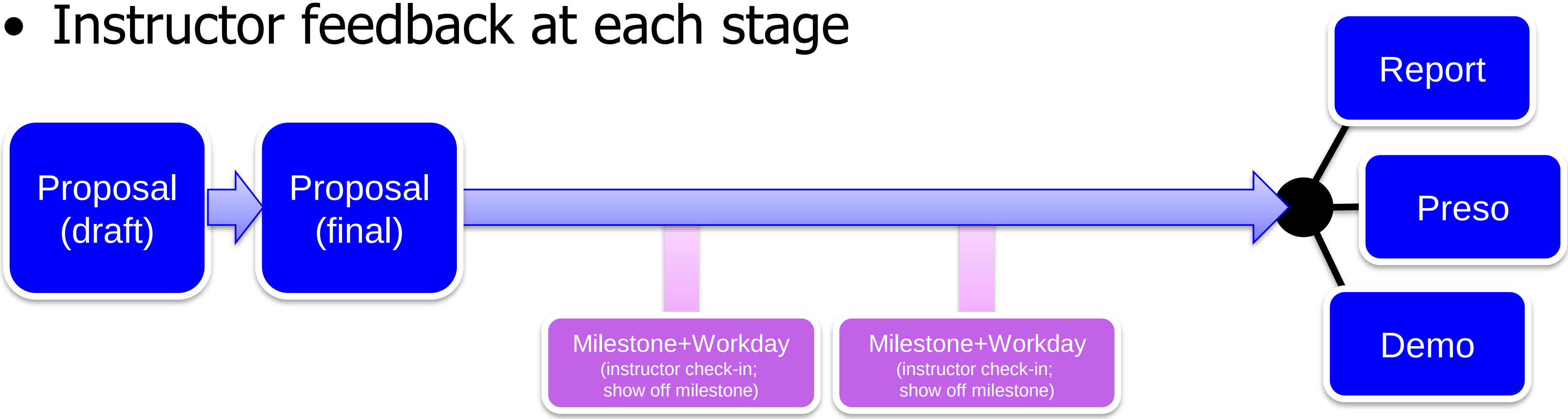
1. **Don't touch other people's stuff.** This includes racks other than the one assigned to us, other equipment in the room, and other group's servers in this course. The things you are allowed to mess with are (a) your server, (b) your power and network links, (c) shared tools/resources in the room (described below).
2. **Respect shared resources.** The room does contain generally useful things such as LCD monitors, keyboards, carts, screwdrivers, etc., which you may use provided you don't interfere with other work. You must put back anything you use when finished.
3. **Report issues promptly.** If you observe any problem with the datacenter, our equipment, etc., please report it to the instructor

< Sign online to gain access

THE COURSE PROJECT

The course project

- Half-semester effort in some area of storage
- Several choices (plus choose-your-own)
- Instructor feedback at each stage



- Any stage can result in a need for **resubmission** (grade withheld pending a second attempt).

- **See course site project page for details**

The Project

- **Draft proposal**: Say what you’re going to do and how.
  - Write-up plus 60-minute meeting scheduled out of class.
  - Must include weekly schedule and two milestones (demoed on workdays)!
  - Get feedback
- **Final proposal**: Incorporate feedback from above.
- **Milestones and workdays:** Evidence of progress that you demo to me directly; also includes free time to collaborate.
- **Final report**: Describe and evaluate your work (max 8 pages).
- **Final presentation**: Demo your work and explain the implementation process to the class (15 min).
- **Final demo**: Defend your project to the instructor.
  - 90+ minute meeting scheduled out of class.

- ***Read course page for details!***

But what *is* the project?

- Generally starts with your BUSE or FUSE program
- Create a system with feature(s) that improve one or more of:
  - **Availability/recoverability**
  - **Network-accessibility**
  - **Storage efficiency**
  - **Performance**
  - **Security**
- Alternately, you may propose a **wildcard project** (custom goal, may or may not use BUSE/FUSE at all)

Example projects

- **Availability/recoverability**
  - RAID at the filesystem level
  - Mirroring to second system (or cloud?)
- **Network-accessibility**
  - Make a network filesystem
  - Store to cloud service
- **Storage efficiency**
  - Filesystem deduplication
  - Filesystem compression
- **Performance**
  - Minimal-seek on-disk data structures
  - Caching with read-ahead
  - Hybrid SSD+HDD filesystem
- **Security**
  - Access control list support
  - Per-user at-rest file encryption

Wildcard projects

- Special purpose file system (e.g. MP3 transcoding)
- Custom block device instead of file system:
  - Custom RAID
  - Custom SAN
  - Block-level encryption
  - Block-level compression
  - Block-level deduplication

Project conclusion

Be thinking about possible projects as we go!

We'll revisit project selection closer to the proposal…

POLICIES

Grade Appeals

- All regrade requests must be in writing
  - Email the TA who graded the question (we'll indicate who graded what)

- After speaking with the TA, if you still have concerns, contact the instructor

- All regrade requests must be submitted no later than 1 week after the assignment was returned to you.

Academic Misconduct

- Academic Misconduct
  - Refer to Duke Community Standard
  - Labs are group-based – everyone works on it
  - Common examples of cheating:
    - Running out of time and using someone else's output
    - Borrowing code from someone who took course before
    - Using solutions found on the Web
    - Having a friend help you to debug your program
- I will not tolerate any academic misconduct!
  - Software for detecting cheating is very, very good … and I use it
- "But I didn't know that was cheating" is not a valid excuse

Our Responsibilities

- The instructor and TA will…
  - Provide lectures/recitations at the stated times
  - Set clear policies on grading
  - Provide timely feedback on assignments
  - Be available out of class to provide reasonable assistance
  - Respond to comments or complaints about the instruction provided

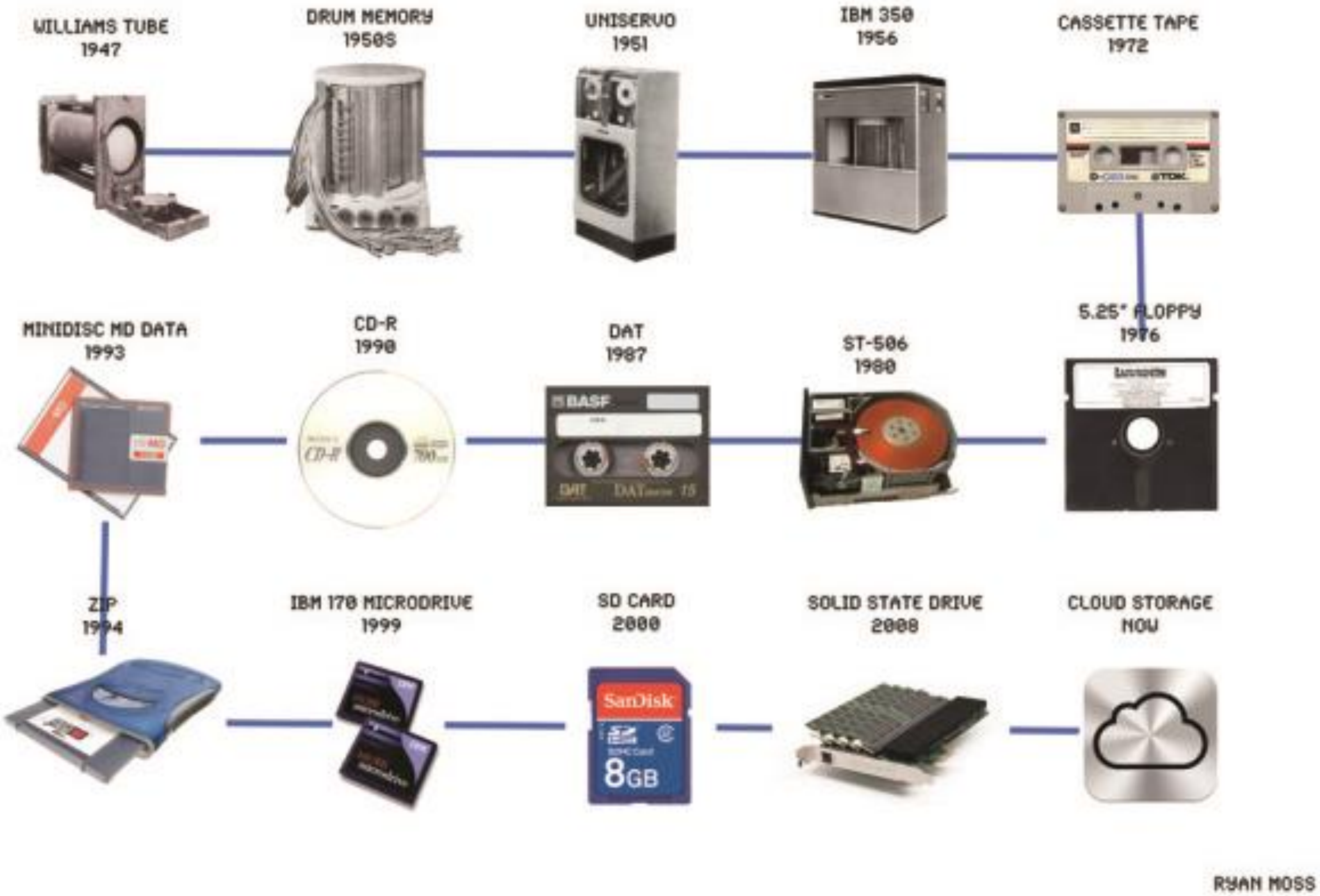
- Students are expected to…
  - Receive lectures/recitations at the stated times
  - Turn in assignments on time
  - Seek out of class assistance in a timely manner if needed
  - Provide frank comments about the instruction or grading as soon as possible if there are issues
  - Assist each other *within the bounds of academic integrity*

Course summary

- We have **hard disks** and **solid-state drives (SSDs)**
- We can use **RAID** to combine performance and capacity while masking effects of drive failure
- The concept of files and directories comes from **File Systems**, a rich field of study.
- We can provide virtual disks to users over **Storage Area Network (SAN)** protocols
- We can provide file access to users using **Network-Attached Storage (NAS)** protocols
- We can provide **storage as a service (SaaS)** via cloud-type protocols.
- Storage efficiency can be improved with **data deduplication** and **compression**.
- We need to preserve **business continuity**:
  avoid downtime and lost data through **backups** and **high availability**
- Storage arrays are deployed based on **workload sizing**.
- Storage is often folded into a complete hardware/software stack: **converged architecture**.
- Storage systems are large enough that **management/monitoring** is its own challenge.
- Storage architects need to understand **basic finance** and **legal/compliance issues**

INTRODUCTION TO STORAGE DEVICES

Basic storage device history



- From https://aaronlimmv.wordpress.com/2013/05/02/types-of-storage-and-basic-advantages-and-disadvantages/

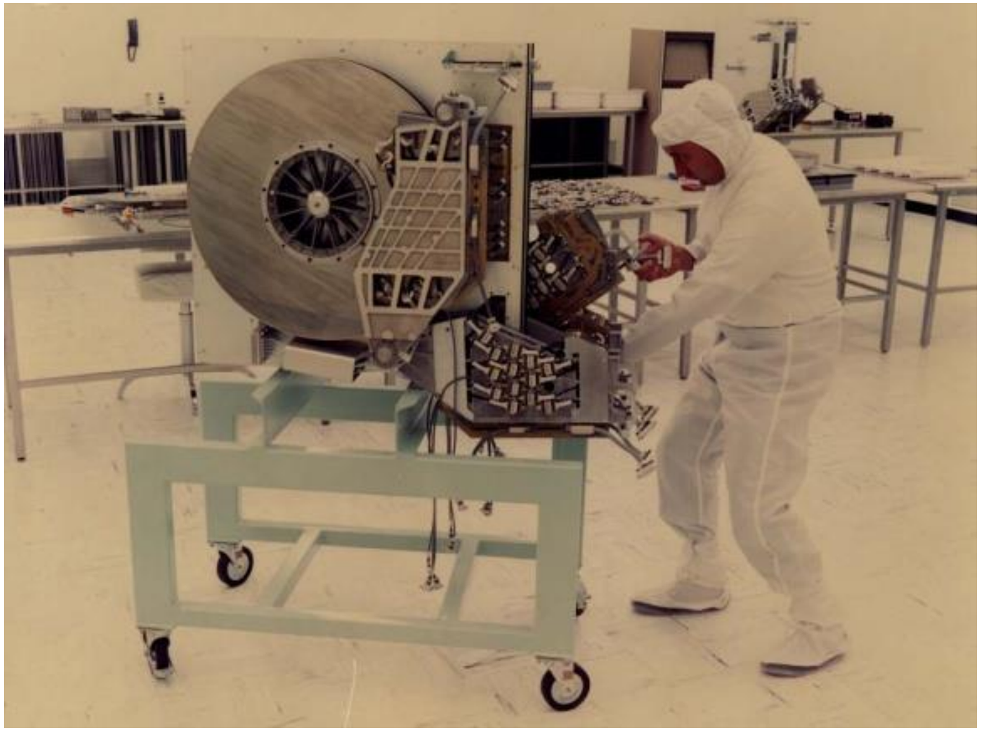
The ancient model of large enterprise storage

- DASD: Direct Access Storage Device
  - Starting with the IBM 350 in 1956
  - Your One Big Computer accesses your One Big Drive
  - Evolution: make the One Big Drive bigger and more reliable
  - Result: The One Big Drive became more and more expensive and critical
  - **Problem?**

An IBM 350 drive (5 MB) being loaded into a PanAm jet, circa 1956.

DASD problem: single point of failure

- The DASD was a single point of failure with *all* your data
  - Better treat it gently…



Man with amazing fashion sense moves a 250MB disk, circa 1979.

Key trend: consumerizaton

- A common evolution in IT:
  - Businesses use a fancy expensive "Enterprise Thing".
  - Normal people get a cheaper version, "Consumer Thing". It's cheap and good enough.
  - Consumer Thing gets better and better every year because:
    - There are more consumers than businesses (bigger market)
    - There are more vendors for consumers than for businesses (more competition)
    - The margins are thinner for consumer goods (more cut-throat competition)
  - A Smart Person finds a way to use the Consumer Thing for business.
  - Industry experts call the Smart Person dumb and say that no real business could ever use the Consumer Thing.
  - The Smart Person is immensely successful, and all businesses use the Consumer Thing.
  - Industry experts pretend they knew all along.

Consumerization in servers

- Big business use mainframe computers

- Everyone else uses microcomputers

- Microcomputers beat mainframes

- We start calling them “servers”

Piled up in a museum

- Mainframes almost entirely gone

Consumerization in storage

- Big business use DASDs

- Everyone else eventually gets small hard disks (SCSI)

- Disk arrays invented using "**JBOD**" and eventually "**RAID**"

- Storage companies based on disk arrays gain traction

Piled up in a museum

- DASDs are entirely gone

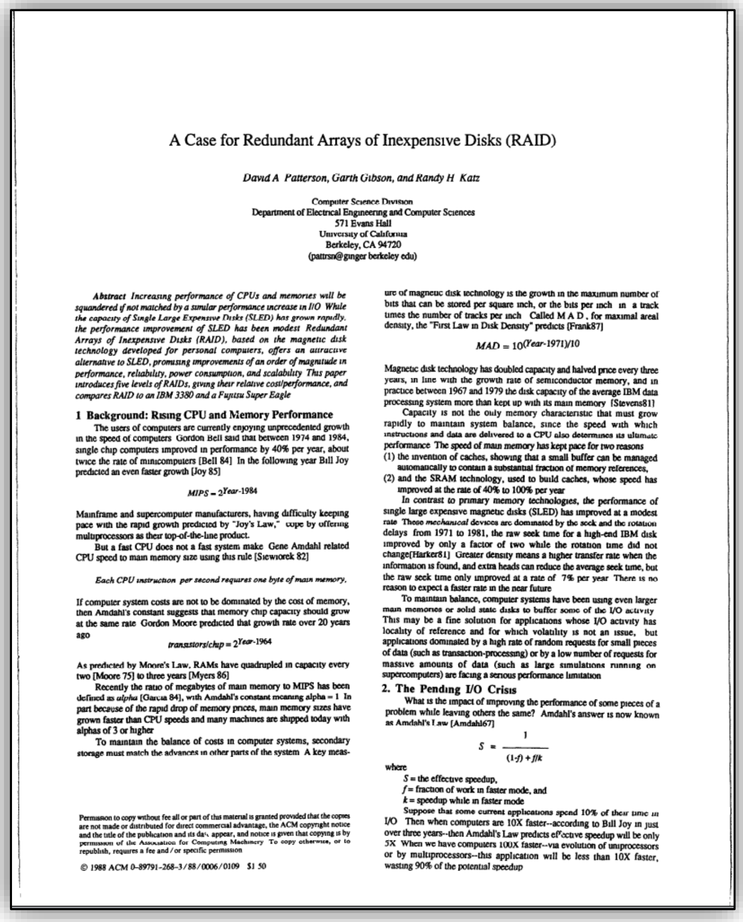
Disk arrays

- **JBOD**: Just a Bunch Of Disks
  - Multiple physical disks in an external cabinet
  - Array is connected to one server only.
  - Provides higher storage capacity with increased number of drives.
  - Effect on performance?
  - Effect on reliability?

- Can we do better?

Disk arrays

- **RAID**: Redundant Array of Inexpensive Disks
  - Academic paper from 1988
  - Revolutionized storage
  - Will discuss in depth later
  - Combine disks in such a way that:
    - Performance is additive
    - Capacity is additive
    - Drive failures can occur without data loss
  - Still directly attached to one server



A Case for Redundant Arrays of Inexpensive Disks (RAID)

David A Patterson, Garth Gibson, and Randy H Katz

Computer Science Division
Department of Electrical Engineering and Computer Sciences
571 Evans Hall
University of California
Berkeley, CA 94720
(pattrsn@ginger berkeley edu)

Abstract *Increasing performance of CPUs and memories will be squandered if not matched by a similar performance increase in I/O While the capacity of Single Large Expensive Disks (SLED) has grown rapidly, the performance improvement of SLED has been modest Redundant Arrays of Inexpensive Disks (RAID), based on the magnetic disk technology developed for personal computers, offers an attractive alternative to SLED, promising improvements of an order of magnitude in performance, reliability, power consumption, and scalability This paper introduces five levels of RAIDs, giving their relative cost/performance, and compares RAID to an IBM 3380 and a Fujitsu Super Eagle*

1 Background: Rising CPU and Memory Performance

The users of computers are currently enjoying unprecedented growth in the speed of computers Gordon Bell said that between 1974 and 1984, single chip computers improved in performance by 40% per year, about twice the rate of minicomputers [Bell 84] In the following year Bill Joy predicted an even faster growth [Joy 85]

$$MIPS = 2^{Year-1984}$$

Mainframe and supercomputer manufacturers, having difficulty keeping pace with the rapid growth predicted by "Joy's Law," cope by offering multiprocessors as their top-of-the-line product.

But a fast CPU does not a fast system make Gene Amdahl related CPU speed to main memory size using this rule [Siewiorek 82]

Each CPU instruction per second requires one byte of main memory,

If computer system costs are not to be dominated by the cost of memory, then Amdahl's constant suggests that memory chip capacity should grow at the same rate Gordon Moore predicted that growth rate over 20 years ago

$$transistors/chip = 2^{Year-1964}$$

As predicted by Moore's Law, RAMs have quadrupled in capacity every two [Moore 75] to three years [Myers 86]

Recently the ratio of megabytes of main memory to MIPS has been defined as *alpha* [Garcia 84], with Amdahl's constant meaning alpha = 1 In part because of the rapid drop of memory prices, main memory sizes have grown faster than CPU speeds and many machines are shipped today with alphas of 3 or higher

To maintain the balance of costs in computer systems, secondary storage must match the advances in other parts of the system A key measure of magnetic disk technology is the growth in the maximum number of bits that can be stored per square inch, or the bits per inch in a track times the number of tracks per inch Called M A D , for maximal areal density, the "First Law in Disk Density" predicts [Frank87]

$$MAD = 10^{(Year-1971)/10}$$

Magnetic disk technology has doubled capacity and halved price every three years, in line with the growth rate of semiconductor memory, and in practice between 1967 and 1979 the disk capacity of the average IBM data processing system more than kept up with its main memory [Stevens81]

Capacity is not the only memory characteristic that must grow rapidly to maintain system balance, since the speed with which instructions and data are delivered to a CPU also determines its ultimate performance The speed of main memory has kept pace for two reasons

(1) the invention of caches, showing that a small buffer can be managed automatically to contain a substantial fraction of memory references,

(2) and the SRAM technology, used to build caches, whose speed has improved at the rate of 40% to 100% per year

In contrast to primary memory technologies, the performance of single large expensive magnetic disks (SLED) has improved at a modest rate These *mechanical* devices are dominated by the seek and the rotation delays from 1971 to 1981, the raw seek time for a high-end IBM disk improved by only a factor of two while the rotation time did not change[Harker81] Greater density means a higher transfer rate when the information is found, and extra heads can reduce the average seek time, but the raw seek time only improved at a rate of 7% per year There is no reason to expect a faster rate in the near future

To maintain balance, computer systems have been using even larger main memories or solid state disks to buffer some of the I/O activity This may be a fine solution for applications whose I/O activity has locality of reference and for which volatility is not an issue, but applications dominated by a high rate of random requests for small pieces of data (such as transaction-processing) or by a low number of requests for massive amounts of data (such as large simulations running on supercomputers) are facing a serious performance limitation

2. The Pending I/O Crisis

What is the impact of improving the performance of some pieces of a problem while leaving others the same? Amdahl's answer is now known as Amdahl's Law [Amdahl67]

$$S = \frac{1}{(1-f) + f/k}$$

where

S = the effective speedup,
f = fraction of work in faster mode, and
k = speedup while in faster mode

Suppose that some current applications spend 10% of their time in I/O Then when computers are 10X faster--according to Bill Joy in just over three years--then Amdahl's Law predicts effective speedup will be only 5X When we have computers 100X faster--via evolution of uniprocessors or by multiprocessors--this application will be less than 10X faster, wasting 90% of the potential speedup

Next step: intelligent arrays

- Server acts as host for storage, provides access to other servers
  - Dedicated hardware for RAID
  - Optimized for IO performance
  - High speed cache
  - Can add various special features at this layer: access controls, multiple protocols, data compression and deduplication, etc.

Method of Attachment

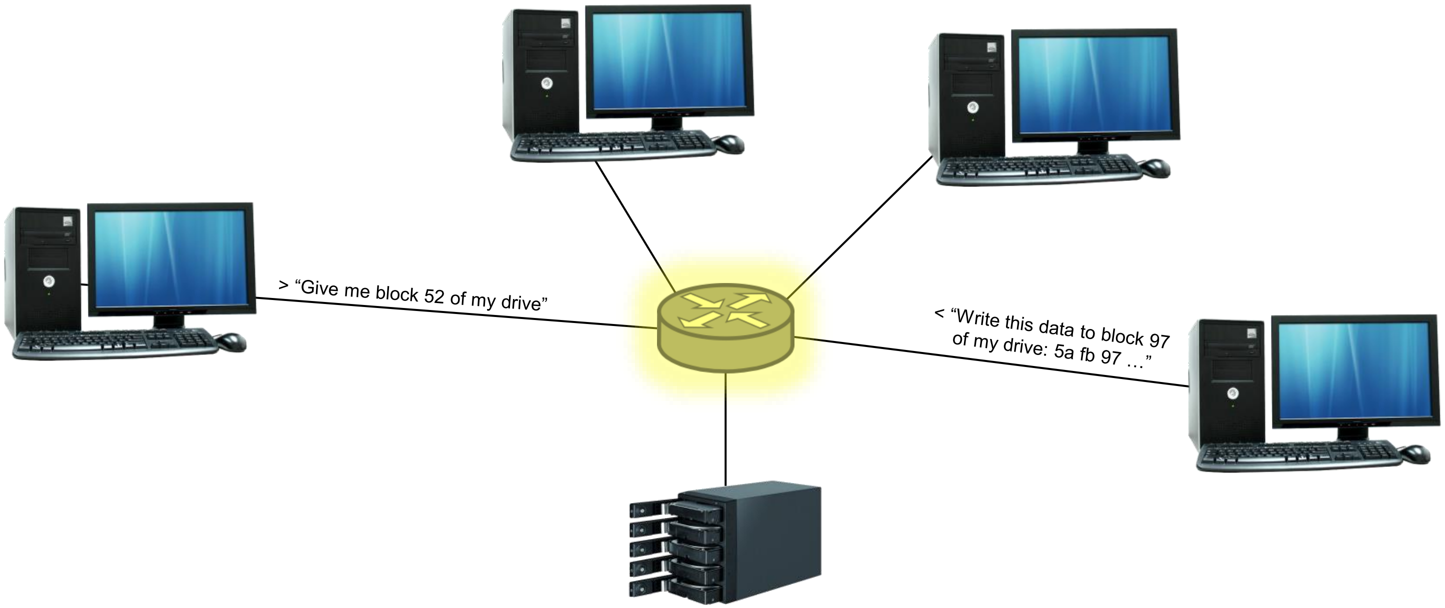
- How to connect storage array to other systems?
  - DAS: Direct Attached Storage
    - One client, one storage server
  - SAN: Storage Area Network
    - Storage system divides storage into “virtual block devices”
    - Clients make “read block”/”write block” requests just like to a hard drive, but they go to the storage server
  - NAS: Network-Attached Storage
    - Storage system runs a file system to create abstraction of files/directories
    - Clients make open/close/read/write requests just like to the OS’s local file system

DAS: Direct Attached Storage

- One-to-one connection
- Historically: connect via SCSI ("Small Computer Systems Interface")
  - Even though actual SCSI cables/drives/systems are gone, the software protocol is still *everywhere* in storage. We'll see it again very soon*.
- Modern:
  - USB: External drives, very fast as of USB 3.0
  - SATA (or if it's external, e-SATA): The protocol modern consumer drives use
  - SAS (Serial Attached SCSI): The protocol modern enterprise drives use
    * see, I told you.
  - NVMe (put SSD right on PCIe bus): for card-type SSDs 

SAN: Storage Area Network (1)

- Split the aggregated storage into virtual drives called Logical Units (LUNs)
- Clients make read/write requests for blocks of “their” drive(s)
- Storage server translates request for block 50 of client 2 to actual block 4000
  (which in turn is block 1000 of disk 3 of the RAID array)

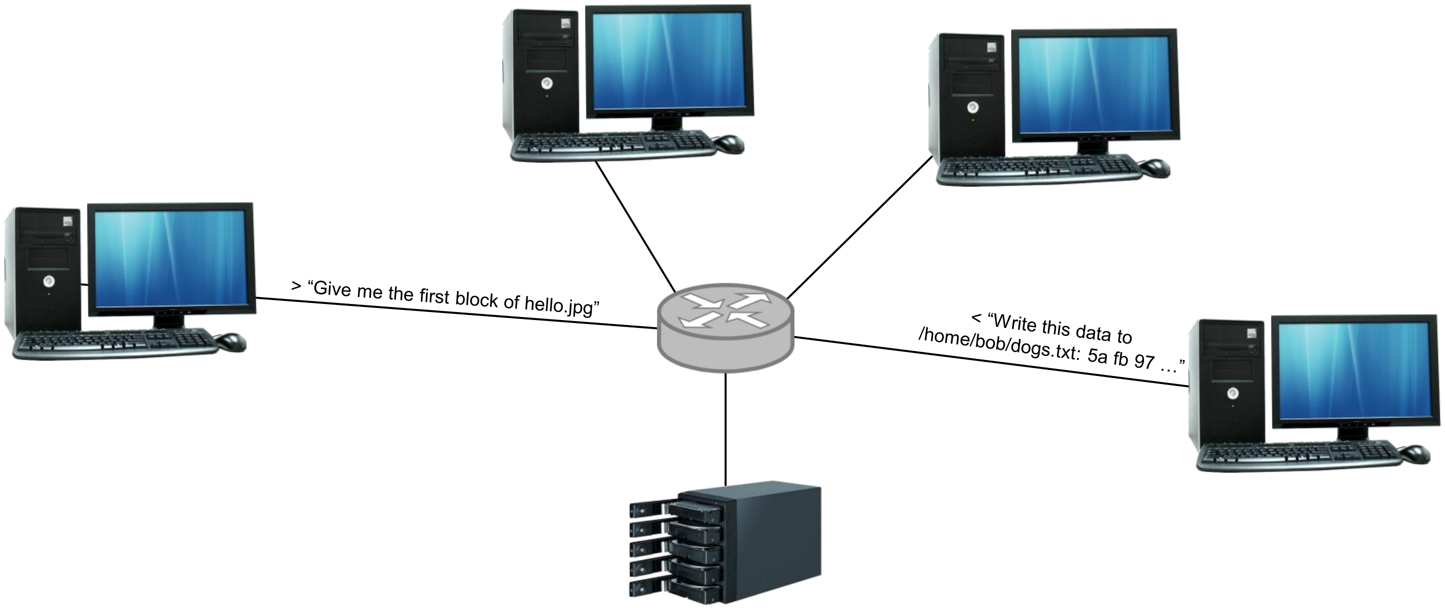


SAN: Storage Area Network (2)

- Traditional protocol: Fibre Channel (FC)
  - A special physical network just for storage
  - Totally unlike Ethernet in almost every way
  - Still popular with very conservative enterprises
  - Actual traffic is SCSI frames
  - Clients and servers have special cards: a Host Bus Adapter (HBA) for FC
- Modern protocols:
  - iSCSI:
    - SCSI inside of an IP frame, usually inside of an Ethernet frame
      (but it's IP, so it could be inside a bongo drum frame)
    - No special switch or cards needed (though iSCSI HBAs do technically exist)
  - NVMe-OF ("Non-Volatile Memory *Over Fabric*")
    - Very new protocol, uses the data protocol from NVMe rather than SCSI
    - Can run over various physical networks (Ethernet, Infiniband, even Fibre Channel!)

NAS: Network-Attached Storage (1)

- Put a file system on the storage server so it has the concept of files and directories
- Clients make open/close/read/write requests for files on the remote file system



NAS: Network-Attached Storage (2)

- No special network or cards – works on normal IP/Ethernet
- Network File System (NFS):
  - Common for UNIX-style systems, invented by Sun in 1984
  - Literally just turns the system calls open/close/read/write/etc into “remote procedure calls” (RPCs)
  - Many revisions, we’re up to NFS v4 now
- Server Message Block (SMB) also known as Common Internet File System (CIFS)
  - Microsoft Windows standard for network file sharing, developed around 1990
  - Really badly named
  - Many revisions, we’re up to SMB 3.1.1 now
  - Native on Windows, supported on Linux with Samba (client and server)

How to tell NAS and SAN apart

NAS = File

SAN = Block

System constraints

- What is a **tradeoff**?
- Constraints:
  - Cost
  - Physical environment
  - Maintenance & support
  - Compliance (regulatory/legal)
  - HW & SW infrastructure
  - Interoperability/compatibility

Management activities

- **Provisioning**: allocate storage based on requirements:
  - **Capacity** – capacity planning
  - **Performance** – workload profiling
  - **Security** – access rule creation, encryption policy
  - **Reliability** – type of redundancy, backup policy
  - **Other** – archival duration, regulatory compliance, etc.
- **Monitoring**: ensure proper functioning over time
  - **Capacity**: watch usage over time, identify workloads at risk of running out
  - **Performance**: collect metrics at storage layer and/or application layer, compare to requirement, alert on violation/deviation, add resources as needed
  - **Security**: verify access control rules, deploy intrusion/anomaly detection, ensure at-rest and in-flight encryption is used where appropriate
  - **Reliability**: receive alerts when failures occur at any layer, continually ensure that availability and backup policies remain satisfied
- **Archival/destruction**: retire data properly

The data lifecycle

From: http://www.spirion.com/us/solutions/data-lifecycle-management

Questions?