Greg Ganger

Associate Professor, ECE & CS
Director, Parallel Data Lab
http://www.ece.cmu.edu/~ganger/
ganger@ece.cmu.edu

Courses:

15-712: Adv. OS & Distributed Systems
18-746: Storage Systems

Parallel Data Lab

Academia's premiere storage systems research center

MAJOR PROJECTS

- Self-* Storage
- Survivable Storage
- Self-Securing Devices
- Semantic File Systems
- Fates Database Storage
- Expressive Storage Interfaces

Self-* Storage

Self-managing, self-organizing, self-healing, self-tuning, self-configuring, ...

- architect with a clean slate
  - integrate self-admin into design
    - rather than after-the-fact tools
  - automate, automate, automate
    - people specify goals
    - system figures out mechanisms
- and, we are actually building it
  - target: 1 PB for CMU researchers
  - goal: research in administration and survivability

18-200 Talk

Software Systems at CMU, and my main sub-area of Storage Systems

Plan for today

- Talking software systems
  - some examples and courses
- My sub-area: storage systems
  - what it's all about
  - example problems
  - my big research activity
- A few random thoughts
  - for what they're worth…

Example software systems

- Operating systems
  - Microsoft Windows, linux, etc.
- Distributed games
  - Quake, Xpilot, etc.
- Distributed storage systems
  - the Web (WWW), AFS, etc.
- Data processing systems
  - Quicken, Matlab, etc.
- … and many, many, many, others

These things are COMPLEX

- What do operating systems do?
  - Provide abstractions of system resources
    - processes, files, sockets, malloc, etc…
  - Isolate application writers from details and each other
- Why are they complex?
  - many objectives
    - performance, reliability, ease of use, security, maintainability, …
  - desire for high utilization of resources
  - generality: support all applications well
- Also, the problem keeps changing
  - technology advances and new applications
- … and combining multiple systems (the soul of distributed systems) complicates everything further

What makes software systems tough and interesting

- Reality
  - simplifying assumptions often don’t hold up
    - people are rarely rational, arrivals are rarely Gaussian, environments are rarely clean, failures are rarely independent, etc…
  - poorly done systems can be incredibly expensive
    - billions of dollars and even life&death
- Rapid changes in technology and applications
  - technology advances change the rules
  - new applications change the requirements
- Most generally: Complexity
  - coping with complexity is what almost all of it boils down to

Real examples of disasters (from J. Saltzer)

- Tax system modernization
  - U.S. Internal Revenue Service
    - tried to replace 27 aging systems
  - Started 1989, scrapped 1997, spent: $4B
  - Causes: complexity
    - all-or-nothing massive upgrade
- Advanced automation system
  - U.S. Federal Aviation Administration
    - tried to replace 1972 Air Route Traffic Control System
  - Started 1982, scrapped 1994, spent: $6B
  - Causes: complexity
    - changing specs, grandiose expectations, congressional meddling

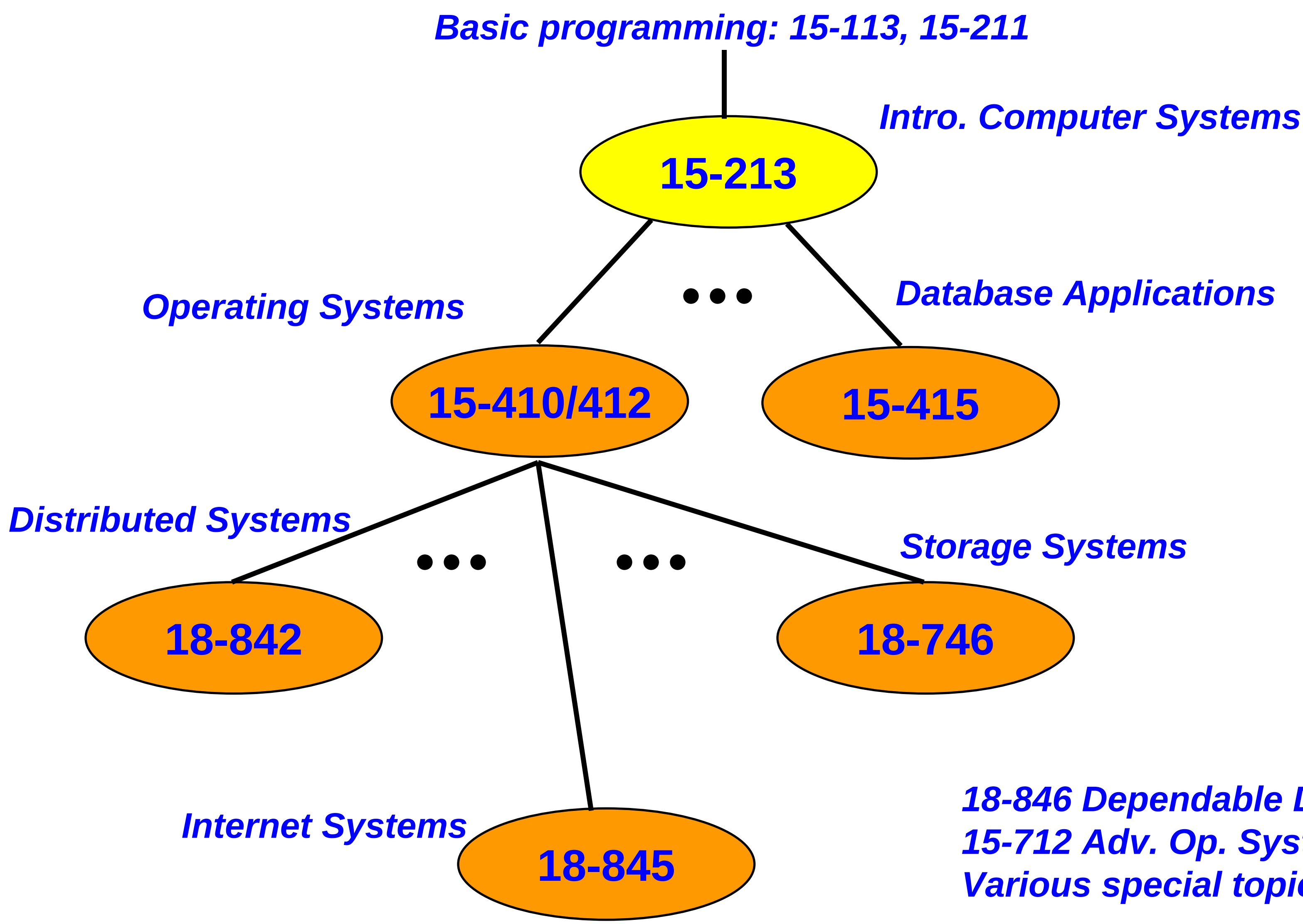
More examples of disasters (from J. Saltzer)

- Ambulance dispatching
  - London ambulance service
    - automate dispatching and routing
  - Started 1991, scrapped 1992, cost: 20 lives lost (and $2.5M)
    - shut down after 2 days of operation
  - Causes: complexity and poor management
    - unrealistic schedule (5 months), overambitious objectives, unidentifiable project manager, low bidder had no experience, no testing/overlap with old system, users not consulted during design
- Automated DMV
  - California department of motor vehicles
  - Started 1987, scrapped 1994, spent: $44M
  - Causes: complexity and blame shedding
    - underestimated cost by 3X, slower than 1965 system, governor fired whistlerblower, DMV blames Tandem, Tandem blames DMV

Rapid pace of our field

- Technology is a major driver
  - Technology eliminates some problems and creates new ones (and enables new applications) over time
  - Incommensurate scaling makes things interesting
  - This means that one has to be on top of technology characteristics and trends
- New application requirements are another major driver
  - Changes the rules (assumptions), often forcing redesign
  - Example: video conferencing vs. best-effort networking
  - Example: mobile computing vs. file system caching
- Systems are complicated and consist of many parts
  - To do top-quality work, you must know about them all!
  - … and their interactions too.

Software Systems Courses



18-846 Dependable Dist. Middleware
15-712 Adv. Op. Systems and Dist.
Various special topics courses

What are "storage systems"?

- Software, interconnects, and devices that collectively provide application writers with an infrastructure that retains persistent data and provides it on request
  - "storage" is where the data lives
  - "systems" are organized collections of components
- Storage systems are at the heart of the information age!
  - the foundation of modern computing
    - the Internet is a network, but the web is a storage system
  - digital libraries, bank accounts, e-commerce transactions, medial records, myriad databases, e-mail, & all your files…
  - EMC, Veritas, Google, NetApp, Oracle, IBM, HP, Sun, …

What makes storage systems so cool?

1. Combines so many topic areas:
   - hardware meets OS meets networking meets embedded systems meets distributed systems meets security…
   - so, it's just plain cool!

What makes storage systems so cool?

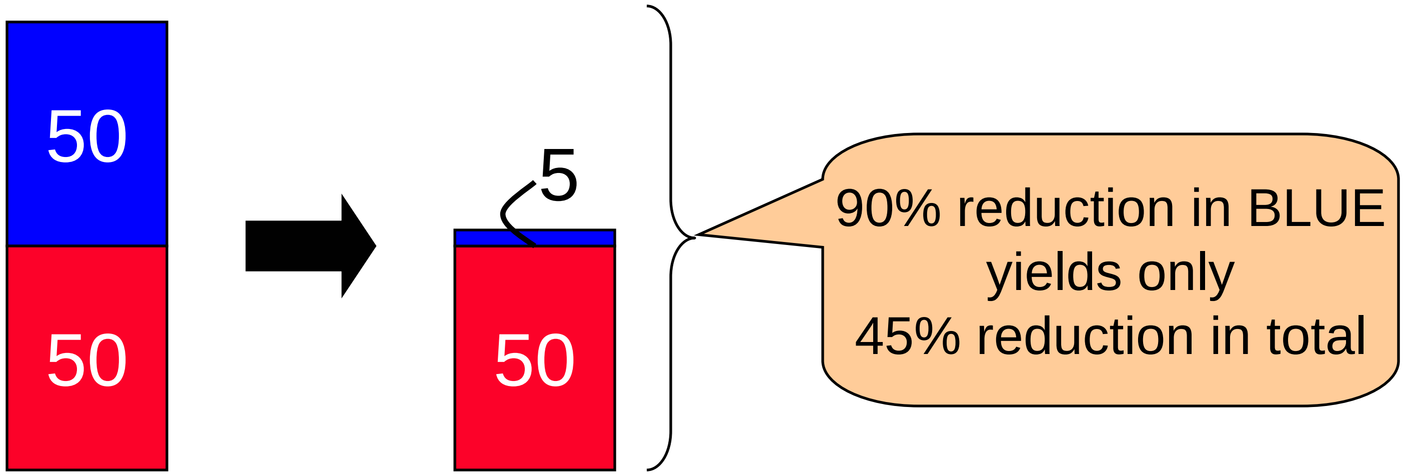
1. Combines so many topic areas
2. This is where great jobs are!
   - Designers and implementers still needed
     - not just testing
   - Continuing growth area for the future
     - The Internet is a network, but the web is a storage system
     - Strong existing companies: EMC, Veritas, NetApp, IBM, HP, …
     - and still support for start-ups: Panasas, Spinnaker, and many others

What makes storage systems so cool?

1. Combines so many topic areas
2. Great careers
3. Still so much room to contribute:
   - performance actually matters here
     - in fact, it dominates system performance in many cases
   - storage management wide open

Amdahl's Law

- Speedup limited to fraction improved
  - obvious, but fundamental, observation



- Consequence for storage systems
  - this has been going on for years!

Major goals/topics of 18-746

- Deep, end-to-end understanding of storage systems
  - key applications and technology trends
  - major components and fitting them together
  - storage management
- Major topics
  - disk operation, firmware, and disk arrays
  - storage interconnects and networking
  - file systems and database back-ends
  - distributed file systems and content delivery
  - virtualization and server farms
  - storage management, data protection, disaster recovery
  - history, alternate technologies, and trends

Some key "storage systems" challenges

- Mitigating the performance limitations of disks
  - often the bottleneck to overall system performance
- Understanding and integrating new technologies
  - how will systems change when…
- Helping users find desired info
  - as the web and growing disks enable info overload
- Robustness in the face of failures and malicious attacks
  - you understand, unless you've had your head in the sand
- Simplifying and automating storage administration
  - 4-8 dollars for every $1 of hardware


Diet Coke