Plan for today

• Some directions things are headed
• Trends
  • ubiquitous computing
  • Internet-scale computing
  • Data-intensive supercomputing
• New technologies
  • Flash-based disk drives
  • Flash replacement: Phase change memory (PCM)
  • Multi-threaded multi-core processors
  • Better tools for handling complexity
• Exam review
Ubiquitous computing

• IDEA: Computers all over the place
  • but, they’re just part of the environment, not really in your face
• Lots of older projects started this trend
  • Aura (CMU), Oxygen (MIT), etc...
• Lots of interesting challenges
  • low power and power management
  • wireless networking
  • real-time services and real-time computing
  • computation and data migration
  • addressing information overload
  • security and privacy!
Internet-wide computing

• Very wide-area computing and information sharing
  • Video conferencing and other forms of communication
  • Grid computing
  • Peer-to-peer distributed lookup and storage systems
  • Edge computing
  • Other forms of remote information access

• Lots of projects

• Lots of interesting challenges
  • resource discovery and selection
  • person location
  • data and/or computation survival and migration
  • cost/benefit models
  • security, privacy, and fighting denial of service attacks
Data Intensive Supercomputing

- Large scale computer centred around data
  - Collecting, maintaining, indexing, computing
  - [https://www.cs.cmu.edu/~bryant/pubdir/cmu-cs-07-128.pdf](https://www.cs.cmu.edu/~bryant/pubdir/cmu-cs-07-128.pdf)

- Think Google-style computing
  - Millions of processors in local-area clusters
  - Commodity parts, reliability depends on redundancy and sw management
  - Partitioned workload

- Lots of examples of problems that fit this model

- Big challenge is moving data
  - 1 TB is cheap to store, hard to move
  - E.g. Average Seagate drive, 125 MB/s → 2.2 hours for 1 TB
  => Fedex > Internet
New technologies

- Changing technology requires changes in system mgmt.
- Moore’s Law
  - faster CPUs, more memory, more storage, more bandwidth
- Shared memory parallel computing becomes mainstream
  - Renewed interest in efficient synchronization, parallel scheduling, distributed shared memory
- Better wireless technologies, smaller devices
  - Ubiquitous computing vision becoming realistic
- Changes in storage technology
  - Cheap, fast non-volatile storage simplifies many file system problems
Better Tools

- Project AURA (CMU Ubicomp) catchphrase:
  - The most precious resource in a computer system is no longer its processor, memory, disk or network. Rather, it is a resource not subject to Moore's law: User Attention.

- Complexity of systems continues to grow, human ability to handle complexity does not
  - Need better languages to specify and check concurrent programs (HPCS languages: X10, FORTRESS, Chapel)
  - Automatic extraction of rules from code, and verification that rules are followed (Engler et al.: metaccompilation)
  - Automatic diagnosis and recovery from errors (autonomous computing)
  - Reduction to less complex system (virtual machines, containers)
  - More automation and tools to handle complexity in general
EOT (aka ^D)
Exam review
The Final

• When and where:

http://www.artsci.utoronto.ca/current/exams/dec17

| CSC469H1F | WED 13 DEC | AM 9:00 - 12:00 | GB 405 |

• Same building as the lectures, different room though...

• www.osm.utoronto.ca/map/

• Morning: 9AM!

• Please be there 10-15 minutes in advance. Exams start on the hour, *not* 10 minutes past!
Final Mechanics

• Final exam is cumulative, all topics are covered
  • Slightly more emphasis on topics after midterm!
• Quick answer questions, and longer questions that involve more thought, and/or working through solving a problem, will cover:
  • Pre-midterm: Systems Design (first lecture), OS Structure & Virtualization, Amdahl’s law, Non-blocking synchronization (except TM)
  • After: OS Scalability & Multiprocessor Scheduling, Virt. Mem. Mgmt, DSM, Logical Clocks, Distributed Agreement, Fault Tolerance, Reliable Storage
• Based mostly on lecture and tutorial material, plus readings
• Closed book, one 8.5”x11” double-sided sheet of notes allowed
  • Just one!
  • Must include the exam date, time, and room in the upper right corner!
  • May be prepared digitally if you like
    • Magnifying glass is not an allowed exam aid! :)
• BEWARE: do not just dump text blurbs from your aid sheet on the exam!
  • What is the question actually asking?
• No calculators or other aids
Overview

- Systems Design (first lecture)
- OS Structure and Virtualization
- Amdahl’s law
- Non-blocking synchronization
- OS Scalability & Multiprocessor Scheduling
- Virt. Mem. Mgmt
- DSM
- Logical Clocks
- Distributed Agreement
- Fault Tolerance
- Reliable High-Performance Storage
Systems Design

- Problems in complex systems
  - Emergent properties, Propagation of Effects, Incommensurate Scaling, Trade-offs
- What are sources of complexity?
- How do we cope with it?
- Techniques for dealing with complexity
  - Modularity, abstraction, layering and hierarchy
- The end-to-end argument
- Butler Lampson’s “Hints for computer system design”
  - In terms of: Functionality, Speed, Fault tolerance
OS structure

- Layered systems, Open systems, Monolithic kernels, Microkernels, Virtual Machine Monitors
  - What is the goal of each of them?
  - How does each work at a high-level?
  - Advantages, limitations/drawbacks?
  - Compare them, see if they fundamentally have commonalities (you must be able to operate with the concepts that you learn, right?)
Amdahl’s law

• What does this law tell you?
• What is the main idea?
• How can this be applied in practice?
• This is a very important law, you WILL run into it in your career!
  • This is not a threat, but a fact.. :)
Non-blocking synchronization

- Cost of locking
- Types of locks
- Problems with locks (deadlock, priority inversion, convoying, expensive even when uncontended etc.)
- NBS – what’s the main idea?
  - How do you use atomic operations to implement non-blocking operations?
  - ABA problem
- RCU – what’s the main idea, how does it work?
  - Grace period, Quiescent states..
OS Scalability + Multiprocessor Scheduling

• Why is scaling an existing OS hard?
  • Problems ..
  • Solutions ..

• Multiprocessor scheduling
  • Basic strategies, Load balancing, Processor affinity
  • How do each of these affect our scheduling decisions?
  • Parallel Job Scheduling: Why does it matter? How does it work?
  • Scheduler-awareness, Space sharing, Backfilling, Reservations
  • Time-sharing, Gang scheduling, etc.
  • Other forms of scheduler awareness
Virtual Memory Management

• Make sure you know the refresher material we discussed in class

• TLBs – exploiting locality
  • TLB coverage, trends, motivation for superpages..

• Superpages
  • What was the rationale behind this?
  • Hardware constraints, alignment, etc.
  • The Superpage problem -- issues: Allocation, Promotion, Demotion, Fragmentation
  • Relocation-based vs. reservation-based allocation
  • Navarro’s design: how does it approach the 4 issues above?
Virtual Memory Management

• Placement Policy
  • What are the reasons why this matters?
• Cache conflicts
• NUMA multiprocessors
• Energy savings
• How does placement policy correlate with each of these?
Distributed Shared Memory

- Features, distributed IPC
- Central Server DSM
- DSM Page Fault Handling
- Atomic Page Update problem
- Replication
  - MRSW, MRMW
- Consistency Model
  - Sequential Consistency
  - Relaxed Consistency
  - Release Consistency, Eager vs. Lazy
Time Clocks, Event Ordering

- Time in distributed systems – problems?
- Physical clocks vs Logical clocks
- The “happens before” relationship
- Lamport Clocks
- Vector Clocks
- How does each work? What’s the difference? When can one be used but not another?
Distributed Agreement

• Properties of Distributed Algorithms
• Timing Model (synchronous, asynchronous)
• Failure Model (Fail-stop, Byzantine)
• Distributed Consensus
• Synchronous Fail-Stop Consensus, FloodSet algorithm
• Byzantine Generals problem
  • How many nodes do we need to tolerate f failures? Rounds?
  • Make sure to revise the algorithms in detail (Oral vs Signed):
    • OM(f)
    • SM(f)
Fault Tolerance

• Avoiding, Masking, Recovering from faults
• Replicated State Machines
• Message order
• Broadcast taxonomy
  • Reliable Broadcast algorithm
  • How do we transform it into FIFO Broadcast, Causal Broadcast?
• Voting and Quorum
  • What’s the idea behind this?
  • What are the conditions we need to avoid conflicts?
  • What should R and W be? Implications of the decision?
Reliable, High-Performance Storage

• FFS vs Log-structured file system
  • How did LFS come about? What is it trying to address?
  • How does LFS work? What are the challenges?
  • How does LFS locate inodes?
• Free space management
• Crash recovery

• Reliability
  • Journaling
  • Soft Updates
  • Redundancy (RAID)
General tips

• Make sure you go over each topic and that you are at the very least able to answer the previous questions
• Course slides, tutorial slides, readings
• Test your knowledge with previous exams
  • Public on the UofT Libraries website – old exams repository
  • Course coverage may have changed since, but not substantially
  • Do not count on just these for prepping though!
• Use the discussion board! (not just to ask, also to answer, this way you test your own knowledge too!)
Preview:

• Question paper + a standard answer booklet

• Must write your answers in the booklet, NOT in the question paper!

• The question paper will not be marked!
More General Tips

• Do not panic! Take a deep breath, you’ve got this!
  • This is your chance to show us what you’ve learned
  • We WANT to give you the credit that you’ve earned

• Read carefully!
  • What is the question asking?
  • Don’t confuse things
  • If there’s anything unclear, please ask
  • Try not to overthink things, we’re not trying to “get you” :)

• Keep track of your time
  • Some questions take more time than others
  • Do not spend too much time on a question if you are stuck – might want to revisit it later
Concluding remarks

- **Congratulations** on surviving CSC 469 .. so far! :)  
  It’s a tough course, but I hope you found it worthwhile and that it will help you with future job prospects

- Remember **course evaluations** expire soon – please make sure to fill them out!

- Good luck with the final exam!
  - Double-check Triple-check the exam schedule carefully and arrive in advance

- Thank you for a great class, it’s been a real pleasure teaching you this term!
The end..