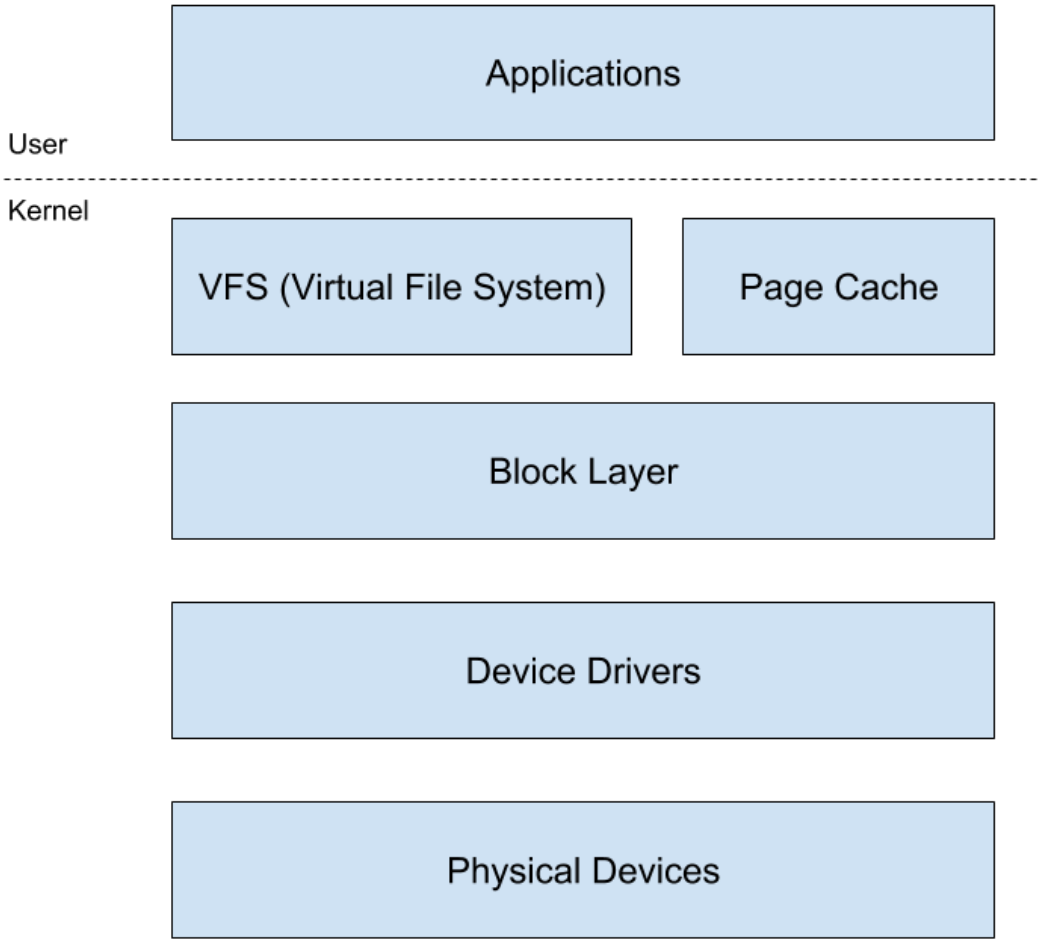
CSCI 333 Storage Systems :: Meeting 00 :: Course Overview

Learning Objectives

- Holistic understanding of storage landscape
- Start to think about interesting aspects of the course

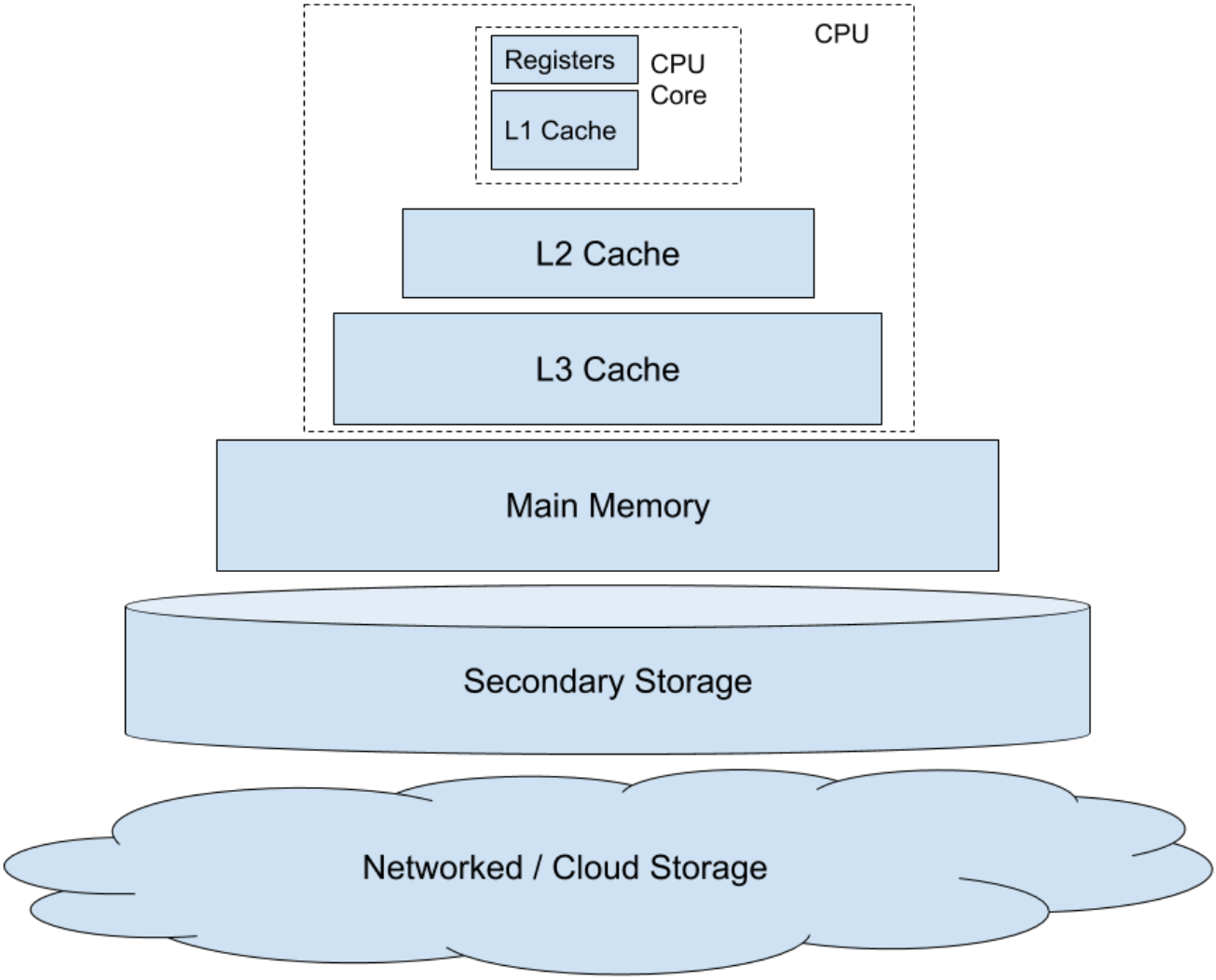
Overview of the software storage stack (Unix/Linux)



Questions

1. What are the interfaces at the layer boundaries?
2. What are the requirements of each layer (e.g., flexible allocation sizes, strict ordering of writes, recovery if power failure, persistence, encryption, etc.)?
3. What privileges and capabilities make implementing these details easy or hard (user vs. root permissions, can hardware be changed after manufacture, can interfaces be changed after adoption)?
4. What are the costs of changing details at each layer (rewrite one app, rewrite all apps, change manufacturing processes)?

Overview of the Memory Hierarchy



Questions

1. What are (rough) sizes at each level?
2. What are (rough) speeds at each level (latency and/or bandwidth)?
3. What are the costs ($/GiB) at each level?

- Are there different choices available to systems designers/users at any level (e.g., HDD vs. SSD vs. Tape vs. CD-ROM), and why would you choose one choice over another? How do those choices influence system designs?

4. What is the unit of transfer at each level (byte, word, page, large object), and what influences the sizes of the data transfer units at each level?
5. Think about the software storage stack diagram from above. How do the different levels of the cache hierarchy play into the design of each layer of the software storage stack (persistence, transfer size, interface, etc.)?

- Reminder from CS237: Cache hierarchy purpose and assumptions (locality locality locality...). The goal of the memory hierarchy is to present users with the performance profile of fast expensive memory but with the cost and capacity of slow cheap memory.

Course Overview

The course can be roughly divided into two "phases". In the first phase (pre-spring break), we will explore key chapters from the **Persistence** section of the OS textbook. We will start at the lowest levels (devices) and work our way up through applications (file systems). In this first phase, we will build a foundation that we expand upon during the second phase of the course (post spring break), where we will explore advanced research topics by reading papers.

Throughout the course, we will develop both language and frameworks for evaluating storage designs. We will use performance models to algorithmically analyze the performance of systems in their best and worst cases. We will also create and run experiments to empirically tease out the implications of system designs. Hopefully, our experiments will validate our performance models; if not, we will figure out why, and improve our models, our experiments, our systems, or all of the above.

Storage is a topic that we often take for granted because "our storage systems work". We care about correctness, but we also care about performance, usability, security, space efficiency, features, cost, and more.