1) How is Warehouse Scale Computer (WSC) different from a server computer? (2 points)

1) \*\*Difference between Warehouse Scale Computer (WSC) and a Server Computer\*\*:

- Warehouse Scale Computers (WSCs) are distinct from traditional server computers primarily in their massive scale. WSCs are designed to power large internet services by integrating thousands of computing nodes along with their networking and storage subsystems, power distribution, and cooling systems within a facility that resembles a large warehouse. In contrast, server computers, including both pizza-box servers and high-end multiprocessors, are designed for smaller-scale operations. WSCs represent a shift from the traditional computing model, where a single program runs on a single machine, to a model where an internet service, consisting of multiple interacting programs, is deployed across a vast hardware platform.

Location:

This information is discussed in the introduction and early sections of the document, specifically around the definition and description of WSCs, which starts on page 1 and continues to page 6. Especially, see sections **1.1 "Warehouse-Scale Computers"** and **1.3 "Not Just a Collection of Servers"** for distinctions between WSCs and traditional server computers​​.

2) What are the workloads that run on WSC? ( 3 points )

2) \*\*Workloads on WSC\*\*:

- Workloads on WSCs are predominantly internet services that require massive data processing and storage capabilities. These include but are not limited to web search, email services, social networking, video streaming, and cloud storage. The defining feature of these workloads is their requirement for handling large-scale data and request processing, which is made possible by the parallel processing capabilities and extensive storage infrastructure of WSCs.

Location:

Workloads that run on WSCs, such as web search, video streaming, and machine learning, are detailed in **Chapter 2 "Workloads and Software Infrastructure,"** starting on page 17. This chapter outlines the diverse range of applications and services powered by WSCs​​.

3) The book chapter makes a point that software development for data center based online services is quite different from classical software development for desktops or servers? Explain this point. ( 3 points )

3) \*\*Software Development for Data Center-Based Online Services vs. Classical Software Development\*\*:

- Software development for data center-based online services differs from classical desktop or server software development in several key aspects:

- \*\*Parallelism\*\*: Online services exhibit a significant amount of parallelism at both data and request levels. Managing and efficiently harnessing this parallelism is a major focus in the development for data centers.

- \*\*Workload Churn\*\*: The deployment cycles for internet services are much shorter, with rapid updates and iterations. This contrasts with the longer development cycles for desktop software, making the environment conducive to rapid innovation.

- \*\*Platform Homogeneity\*\*: Data centers offer a more homogeneous computing environment compared to the diverse hardware and software configurations found in desktop computing. This simplifies aspects like scheduling, load balancing, and maintenance.

Location:

The differences in software development approaches between data center-based services and classical desktop or server environments are explored in **Chapter 2,** particularly in the sections discussing platform-level software, cluster-level infrastructure software, and application-level software. Key insights can be found throughout Chapter 2 but are summarized and contrasted specifically in section **2.6 "WSC Software Tradeoffs"**​​.

4) Based on the book chapter, describe the problem associated with tail latency and what is the solution described in the book chapter for tail tolerance? (2 points)

4) \*\*Tail Latency and Its Mitigation\*\*:

- \*\*Problem with Tail Latency\*\*: Tail latency refers to the latency experienced by the slowest requests in a system. In large-scale online services, a small fraction of latency outliers can significantly impact service responsiveness, especially as the scale of the system increases.

- \*\*Solution for Tail Tolerance\*\*: Techniques to mitigate tail latency often leverage redundancy and resource replication, which are also used for fault tolerance. These approaches enable systems to still deliver low tail latency by providing alternative resources to handle requests when outliers occur. The use of such techniques is predicted to become increasingly valuable as online services continue to grow in scale.

Location:

The problem of tail latency and the solutions for tail tolerance are discussed in the context of performance and availability considerations for internet services. These are detailed in Chapter 2, within the discussions on software infrastructure and specifically in the part of 2.6.4 "Tail-Tolerance". This section delves into strategies to manage the latency of the slowest requests, crucial for maintaining service performance​​.