

1 Integration Strategy for Data-Intensive Applications

- **Leverage Cross-Disciplinary Learning:**
 - Focus on concepts that overlap with your current studies in operating systems and algorithms. For example, understanding memory management in operating systems can translate into understanding how distributed systems handle memory (e.g., caching and sharding).
- **Skim Key Topics Without Deep Diving:**
 - Use your Kleppmann book to skim chapters that relate directly to your core studies. For example, review sections on distributed databases without the need for full mastery, focusing instead on gaining high-level knowledge.
- **Engage with Practical Examples:**
 - Spend small amounts of time experimenting with lightweight tools that expose you to new concepts. For example, setting up a local environment for tools like Apache Kafka or MongoDB to understand their architecture.
- **Audio Learning and Podcasts:**
 - Continue using audiobooks, and consider adding podcasts related to distributed systems, AI, and data engineering. Passive learning keeps you informed without needing focused study time.
- **Tie Concepts to CLRS:**
 - Recognize that many data-intensive systems rely on algorithms you encounter in CLRS. As you study concepts in CLRS (e.g., graph theory or dynamic programming), try to relate them to real-world data systems such as distributed consensus protocols (e.g., Paxos or Raft).
- **Stay High-Level on Emerging Tech:**
 - Keep an awareness of technologies like blockchain and AI without diving too deeply. For now, understand basic principles like cryptographic hashing and distributed ledgers.