A. Arpaci-Dusseau **Department of Computer Science**
CS739: Distributed Systems **University of Wisconsin, Madison**

Dynamo: Amazon's Highly Available Key-value Store - SOSP'07

1 Introduction

- How is the Amazon platform different from a peer-to-peer system? Where does that impact their design?

2 Background

- What properties does Dynamo strive to provide?
- Where does Dynamo use application-specific knowledge?
- What is a *service level agreement (SLA)*? What are the advantages of having one?

3 System Architecture

- What interface does Dynamo support? How is this different from CFS and how does that impact the design?
- How is the partitioning algorithm of key-objects to nodes done in Dynamo compared to CFS? (Remember to consider the description in Section 6.2 as well...)
- *Replication* is performed in both systems to help with data availability. Any differences here? Does Dynamo need to perform caching?
- Since Dynamo allows objects to be updated from multiple clients, and because clients may not have read the most recent version of an object, Dynamo has to worry about data consistency. Go through the example of how *vector clocks* are used to determine if two updates are in conflict or not.
- Read and write operations involve the first N healthy nodes in the preference list, skipping over those that are down or inaccessible. Why does Dynamo allow the number of read (R) and written (W) replicas to be configured? What properties need to hold between R and W for eventual consistency? What are some common configurations?
- Why does Dynamo provide a "sloppy quorum"? What can go wrong with a sloppy quorum?

- Merkle trees are very useful for summarizing hierarchical state. We'll see them more later (SUNDR).
- Can you think of other differences between CFS and Dynamo?

4 Experiences and Lessons Learned

- Any other interesting ways to improve performance at the cost of durability?
- Experiments: Demonstration that the system is working.

5 Conclusions

- Great job incorporating many different techniques from other distributed systems: consistent hashing, vector clocks, quorums, Merkle trees, gossip-based membership. Their contribution is the *combination* of all of those techniques into a working system and the demonstration that the system really operates as expected.