D-Tunes: Self Tuning Datastores for Geo-distributed Interactive Applications
Shankaranarayanan P N, Ashiwan Sivakumar, Sanjay Rao, Mohit Tawarmalani
Purdue University

Motivation and Challenges
- Online interactive applications
- Low latency – data close to users (e.g. < 100ms)
- High availability – DC or server failures, network partitions
- Strong consistency – All reads see the latest write
- Geo-distributed datastores (e.g. Spanner, Cassandra)

Novel aspects of our model:
- Many parameters – location, # of replicas, quorum sizes
- Judiciously tradeoff consistency, latency and availability
- Heterogeneity across data items (e.g. location of access)
- Scale of the data – millions of users (e.g. Twitter)

D-Tunes design
- Application constraints e.g. read latency < 50ms, N > 3
- Configuration decisions e.g. N = 3 (2 in USE 1 in USW)

Motivating example – real world Twitter trace
- Real world application traces – Twitter, Wikipedia, Gowalla
- across 8 regions, 21 Availability Zones world-wide
- Real world application traces – Twitter, Wikipedia, Gowalla

Experimental validation on Amazon EC2
- Cassandra cluster on Amazon EC2
- Across 8 regions, 21 Availability Zones world-wide
- Real world application traces – Twitter, Wikipedia, Gowalla

Large scale experiments – trace driven simulation
- Real world application traces
  - Twitter – 5 year trace, 3 million users
  - Wikipedia – 3 year trace, 4 million+ wiki articles
  - Gowalla – 2 year trace, 0.2 million users

- Across 8 regions, 21 Availability Zones world-wide
- Real world application traces – Twitter, Wikipedia, Gowalla

Basic Availability model
- Guarantee availability under failures
- Variable performance during failures

Failure resilient model
- Guarantee availability
- Good performance even during failures, congestion events

Acknowledgements
This work was supported in part by NSF grants 0953622 and 1162333