North by Northwest: Infrastructure Agnostic and Datastore Agnostic Live Migration of Private Cloud Platforms

Navraj Chohan, Anand Gupta, Chris Bunch, Sujay Sundaram, Chandra Krintz

UC Santa Barbara

AppScale

Outline

- Private PaaS
- AppScale
- Design
- Implementation
- Evaluation
- Conclusion

Private PaaS

- On site cloud technology
  - Elastic
  - Distributed
  - Fault tolerance & high availability
- Enables programmer productivity
  - Abstract away lower level details (IaaS/OS)
  - Focus on application development
- Offerings
  - AppScale, CloudFoundry, OpenShift

AppScale

AppScale

- Google App Engine (GAE) private PaaS
- Released open source in 2009
- Infrastructure agnostic
  - Packaged as a VM Image (Ubuntu 10.04)
  - Xen, KVM, EC2, Eucalyptus, OpenStack, etc.
- Datastore agnostic
  - Cassandra, Hypertable, HBase, etc.

AppScale

AppScale

- Datastore abstraction layer
- Transaction support within “entity groups”
  - Programmatically assign during runtime
  - Multi-row ACID semantics
- Lock management via ZooKeeper
- Limited query support (GQL)
  - No JOINs or queries that can do inserts

AppScale

Live Migration Motivation

- Goal: Update the PaaS system
- PaaS, IaaS, HW layers change over time
  - Software updates
  - New features
  - OS updates and patches
  - Hardware updates (HDD->SSD)
- Datastore updates and migrations
  - NoSQL stores
    - Performance and features
    - No porting tools
  - Eliminates DB “lock-in”

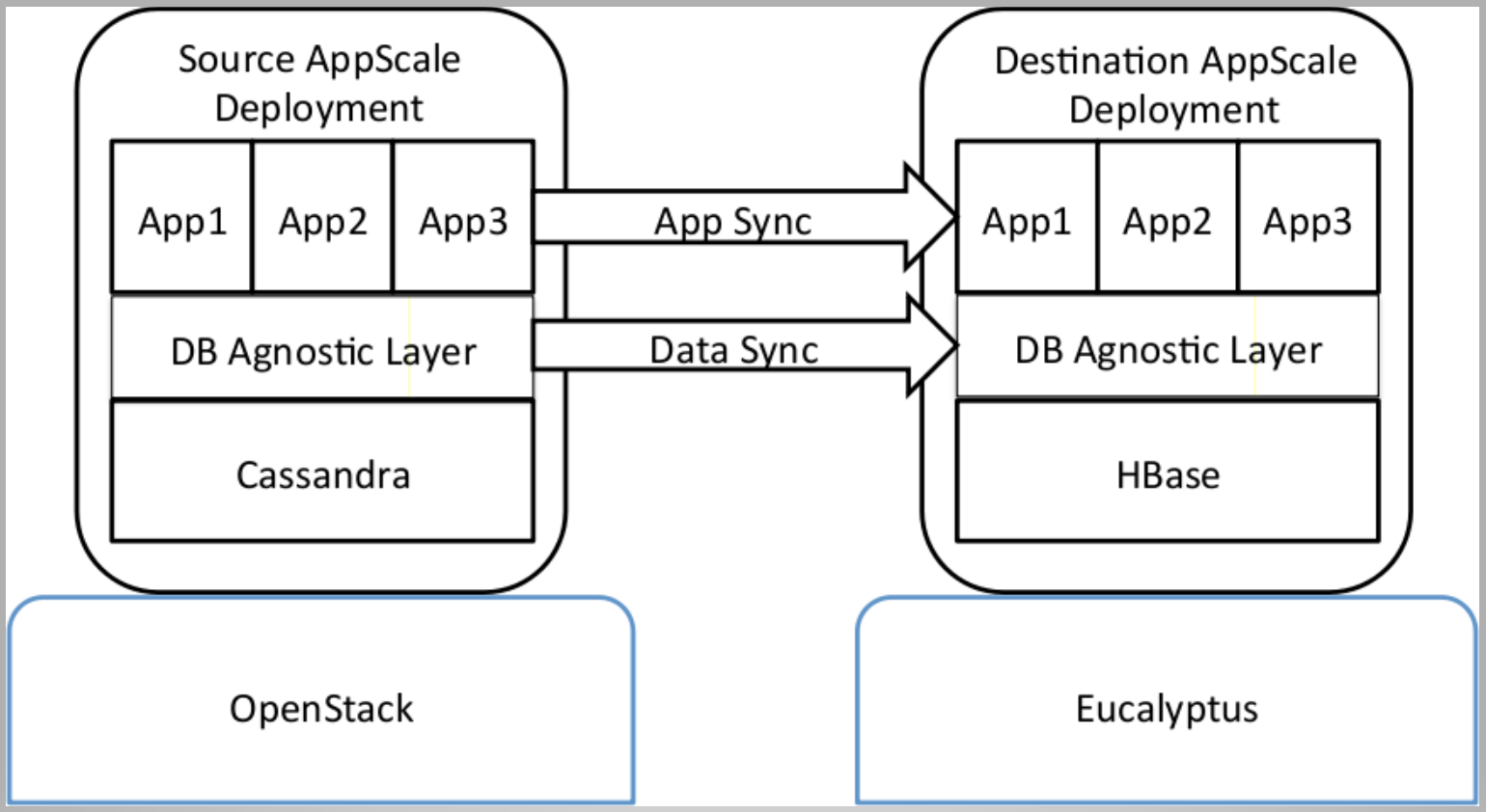
AppScale

Design

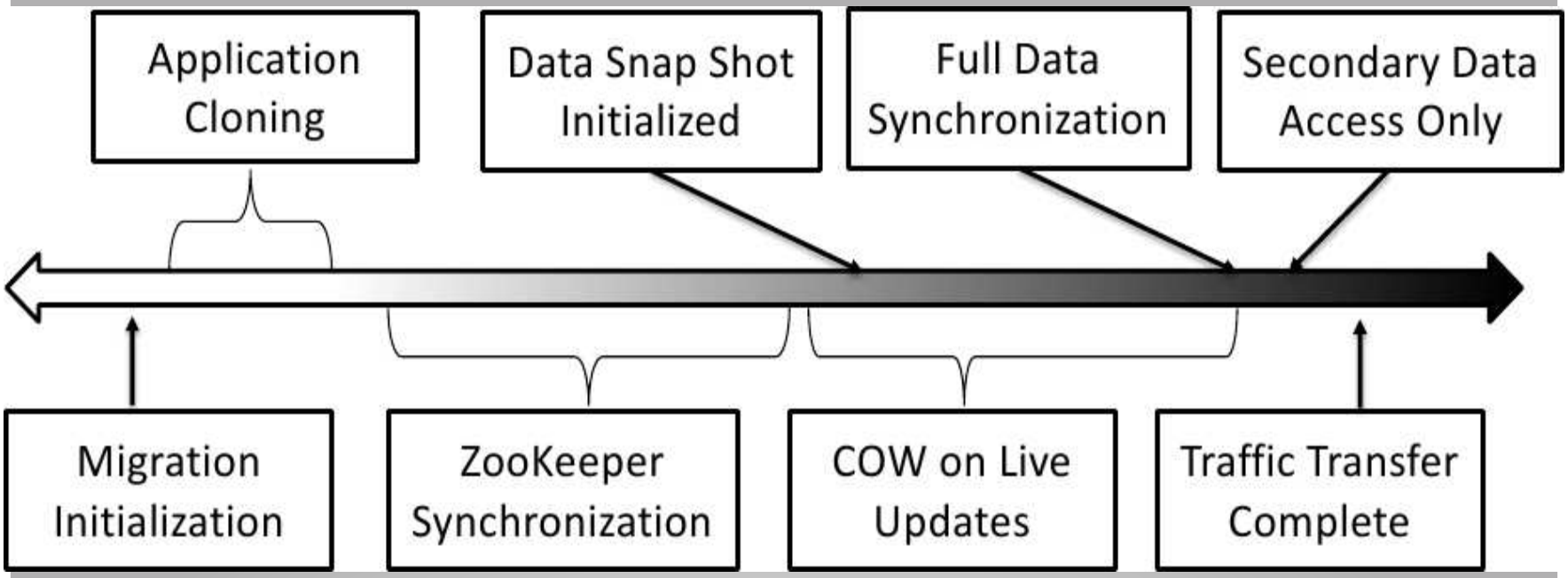
- Minimal-to-no downtime
- Minimal overhead
- Real system
  - Simple and effective
  - Rolling upgrades
- Backward compatibility for applications
- Must support transactions
- No data loss

AppScale

AppScale Migration

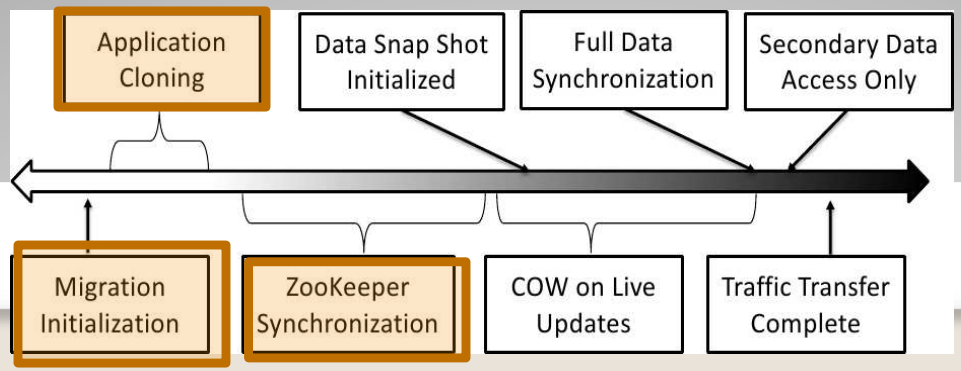


Implementation



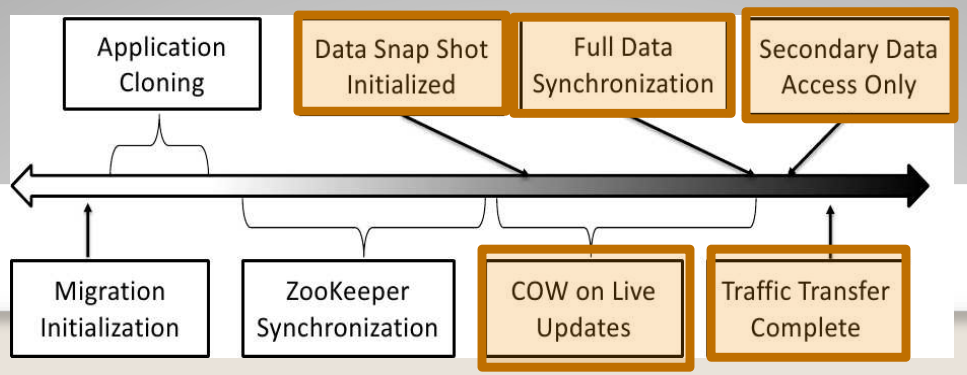
Migration Synchronization

- N1 initializes N2
  - Uploads existing applications
- Synchronize ZK data between N1 and N2 deployments
- Memcache Warm-up
  - Minimizes load on DB
  - Do a copy-on-write and copy-on-read for each key



Migration Synchronization

- Start with copy-on-write (COW) to N2
- Initialize a snapshot of the datastore
- Transfer and load the snapshot
  - Only copy over non-existing keys
  - ZooKeeper locks are required for transactional operations
- Go into data proxy mode
- Full handoff via DNS



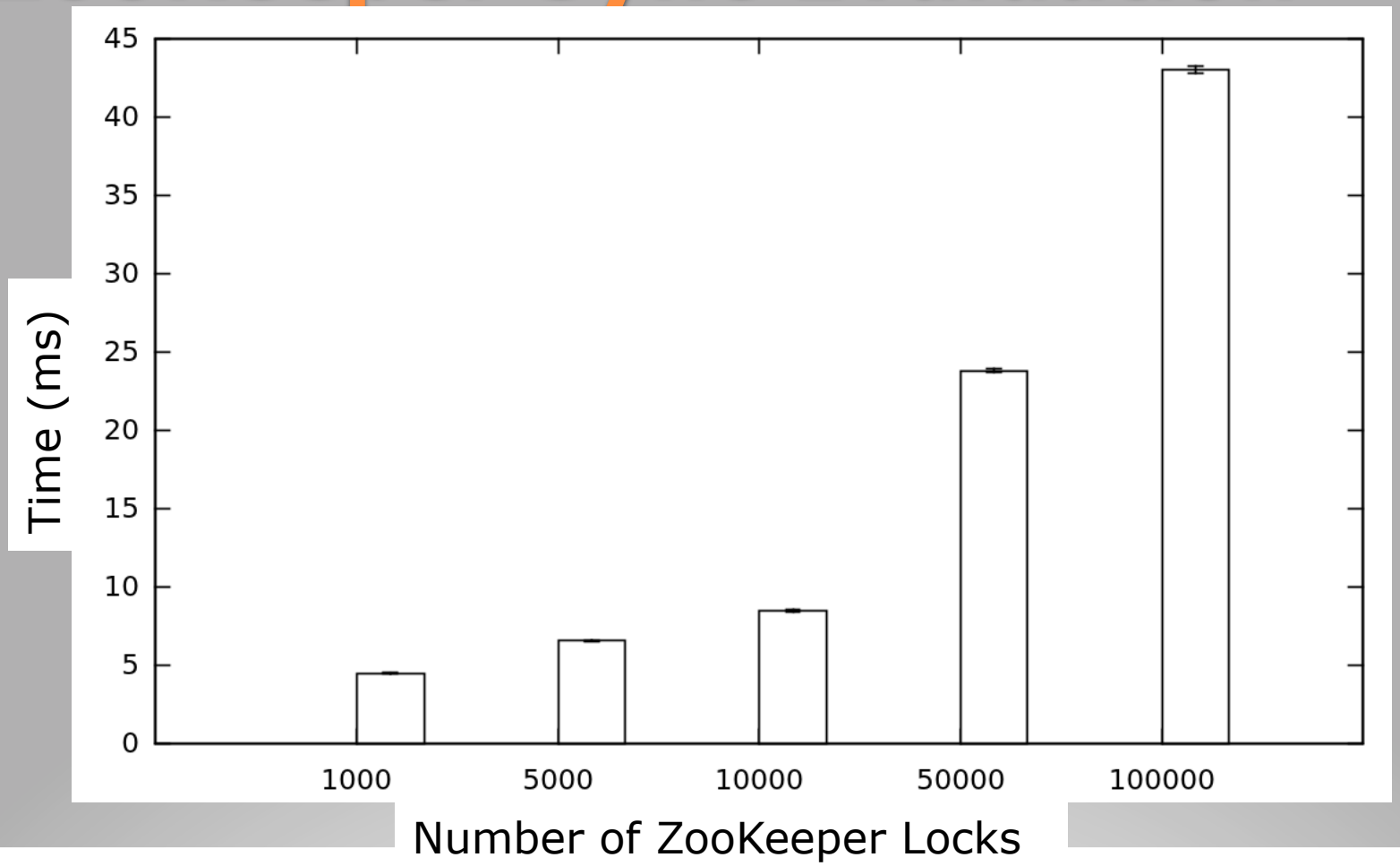
Results

Memcache Warming

- 5KB entities
- COW
  - .17ms of overhead
- COR
  - .85ms of overhead
  - Writes are 10.3 times longer
- <1% of overhead for a request

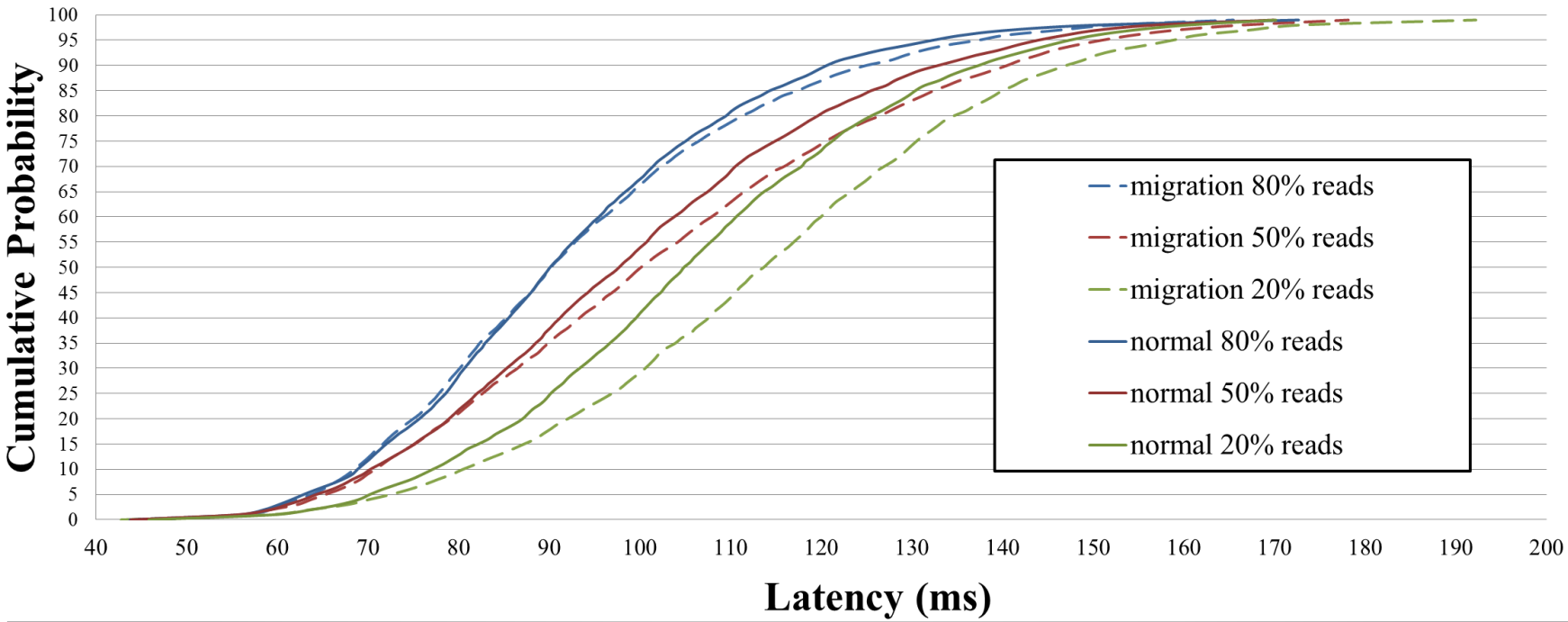
AppScale

ZooKeeper Sync Evaluation



AppScale

Datastore Sync Evaluation



AppScale

Data loading and Handover

- Entity loading from snapshot
  - 2.45-3.18ms for 100B-100KB entries
- Amazon Route 53 DNS Service
  - Dynamically updated via REST API
- Timed with Apache Benchmark tool
- Average switch over time of 46.4 seconds

AppScale

Related Work

- Private PaaS offerings
  - Red Hat's OpenShift
  - VMWare's CloudFoundry
- GAE Compatible
  - TyphoonAE
- Migration
  - VM Migration
  - Albatross by Das et al.

AppScale

Thank You

- Check out AppScale at:

appscale.cs.ucsb.edu

AppScale

AppScale