Managing Data in The Cloud: The view from Santa Barbara

Divy Agrawal and **Amr El Abbadi**
Department of Computer Science
University of California at Santa Barbara

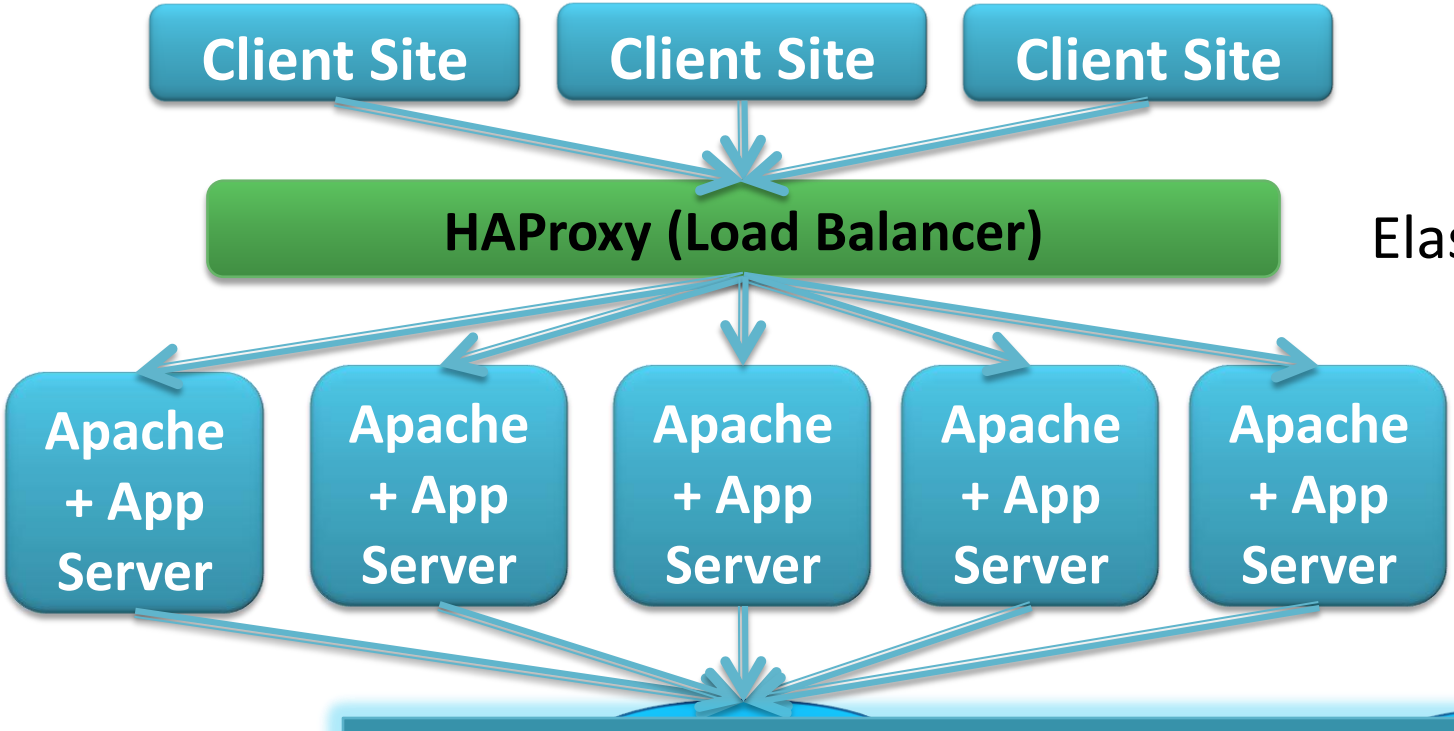
The UCSB Team (in summer migration to Seattle)

- **Sudipto Das** (Microsoft summer intern)
- Shyam Antony (Microsoft now)
- Aaron Elmore (Amazon summer intern)

The Big Picture

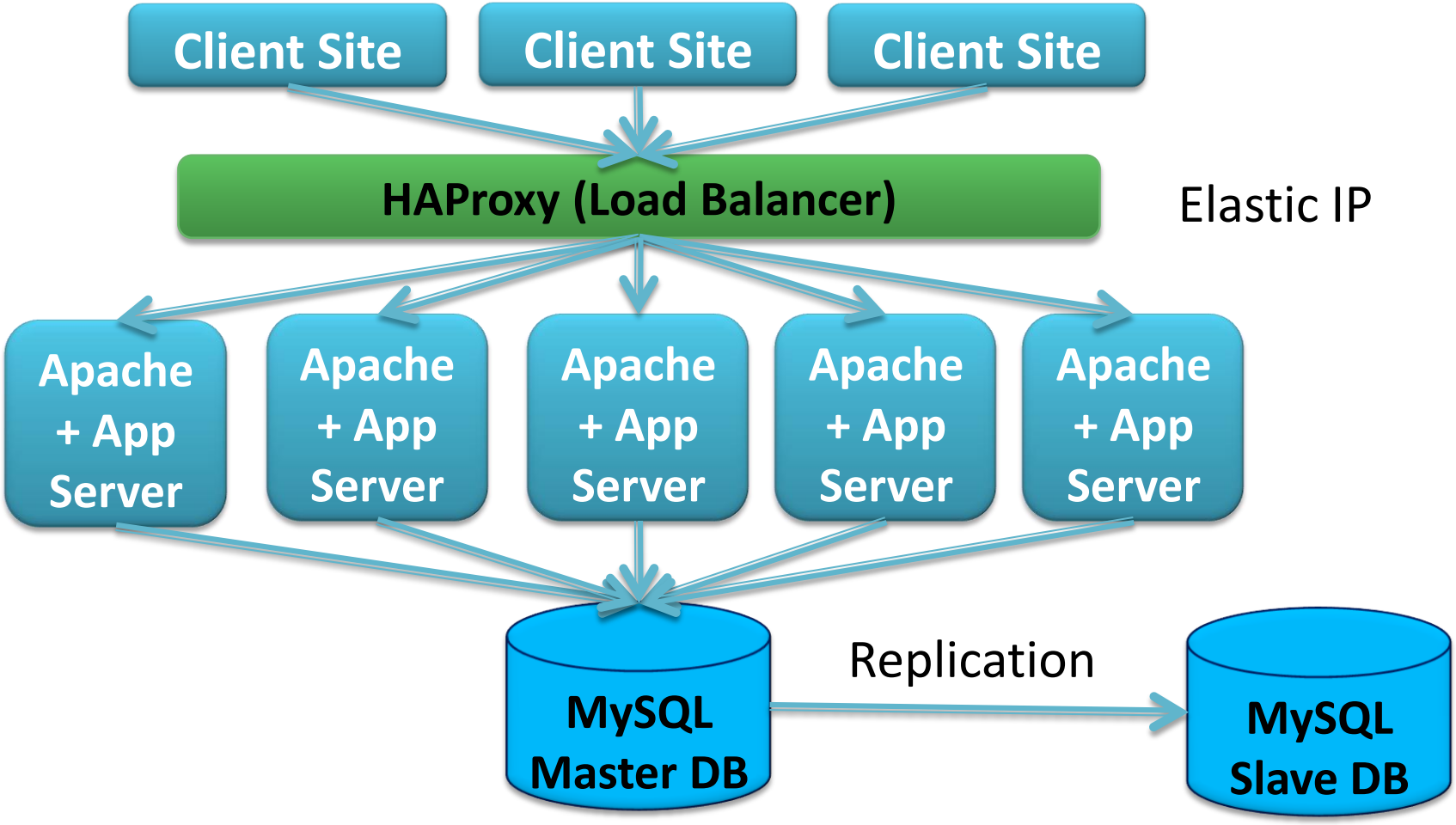
- Different from earlier attempts:
  - Distributed Computing
  - Distributed Databases
  - Grid Computing
- Cloud Computing is REAL
  - Organic growth: Yahoo!, Microsoft, Amazon, Google
  - Poised to be an integral aspect of the global computing Infrastructures

Scaling in the Cloud

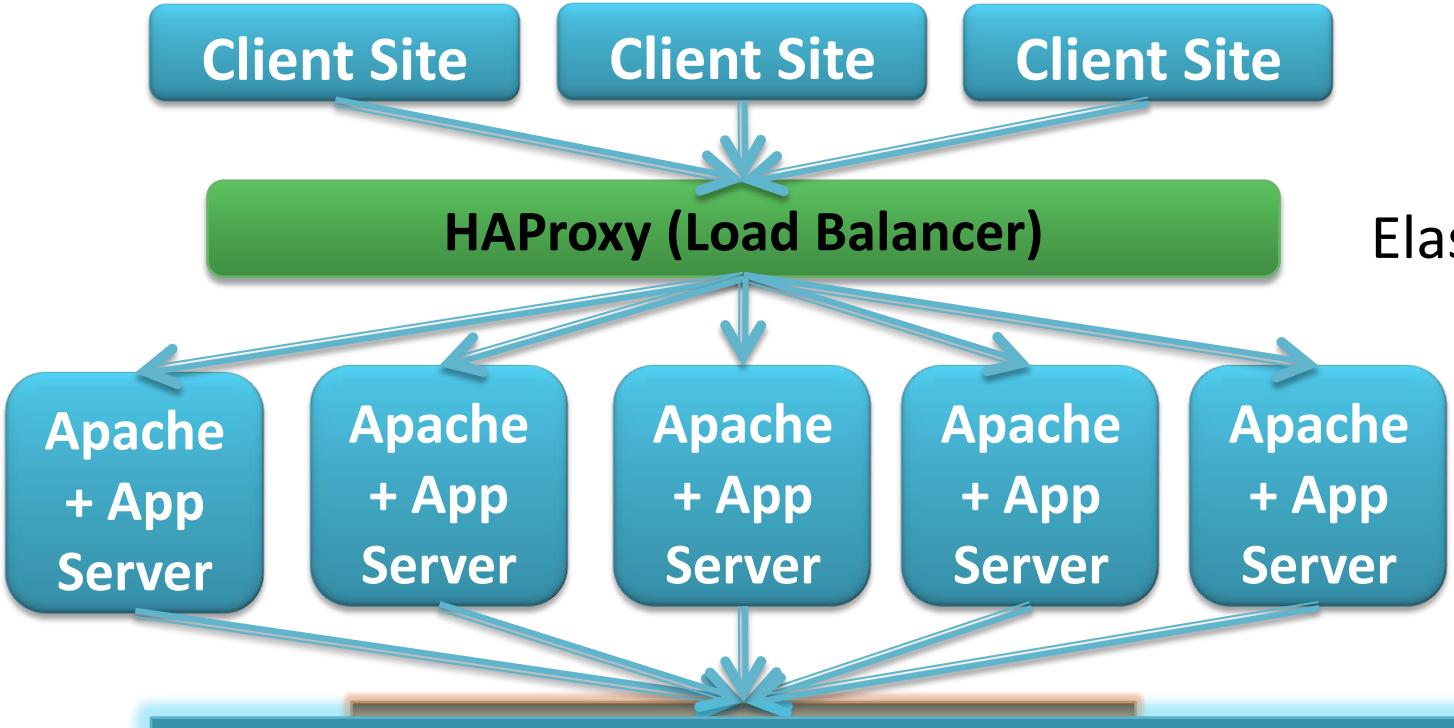


Database becomes the Scalability Bottleneck
Cannot leverage elasticity

Scaling in the Cloud



Scaling in the Cloud



Scalable and Elastic
But limited consistency and operational flexibility

Cloud Computing Desiderata

- Scalability
- Elasticity
- Fault tolerance
- Self Manageability
- Sacrifice consistency?
  - Can be done, but is it a foregone conclusion!!!

Design Principles

- Separate System and Application State
- Limit interactions to a single node
- Decouple Ownership from Data Storage
- Limited dist synchronization is practical

Two UCSB Approaches to Scalability

- **Data Fusion**
  - Enrich Key Value stores
  - GStore: Efficient Transactional Multi-key access [ACM SOCC'2010]
- **Data Fission**
  - Cloud enabled relational databases
  - ElasTraS: Elastic TranSactional Database [HotClouds2009;Tech. Report'2010]

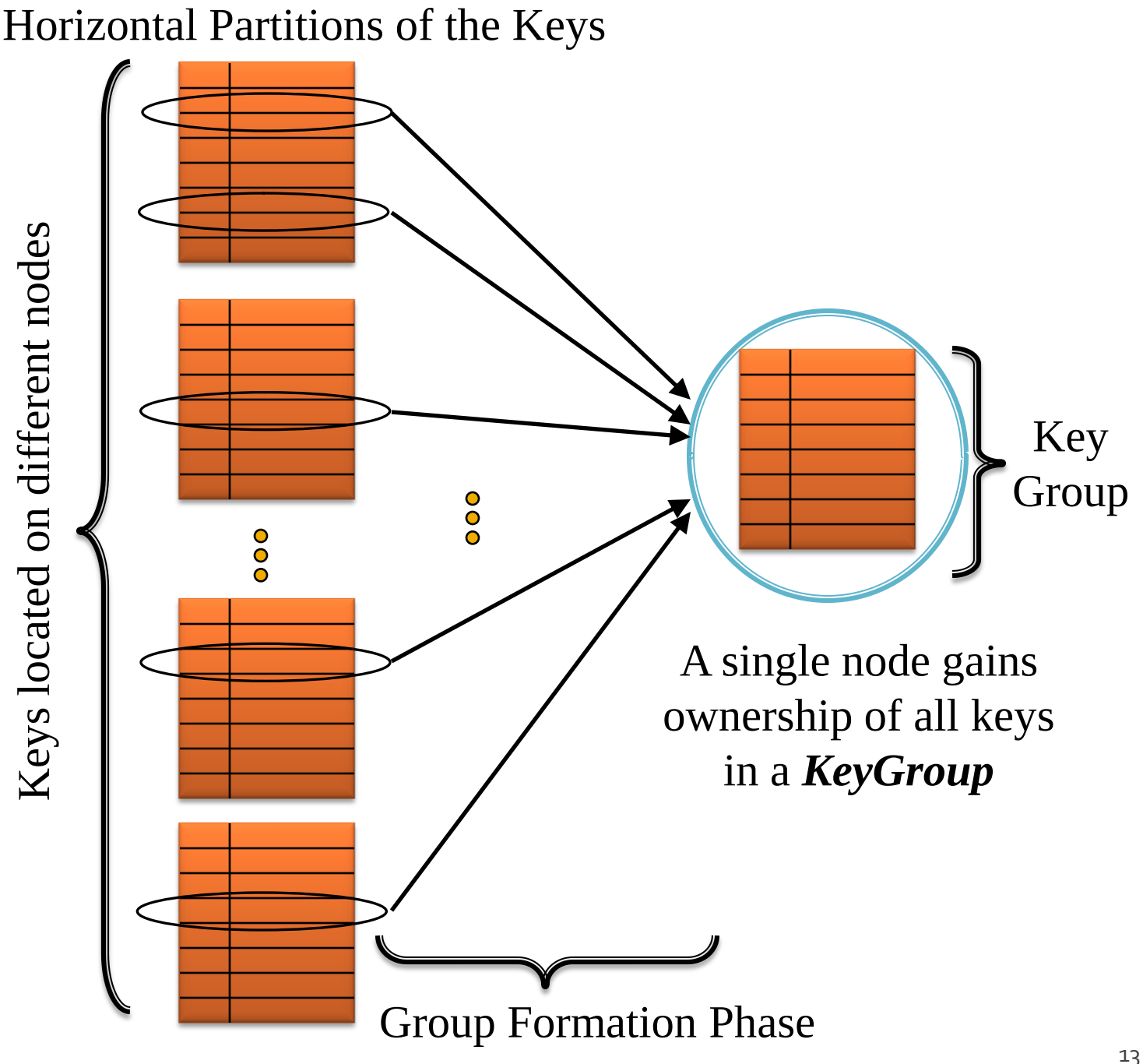
Data Fusion: GStore

Atomic Multi-key Access

- Many applications need multi-key accesses:
  - Online multi-player games
  - Collaborative applications
- Enrich functionality of the Key value stores

Key Group Abstraction

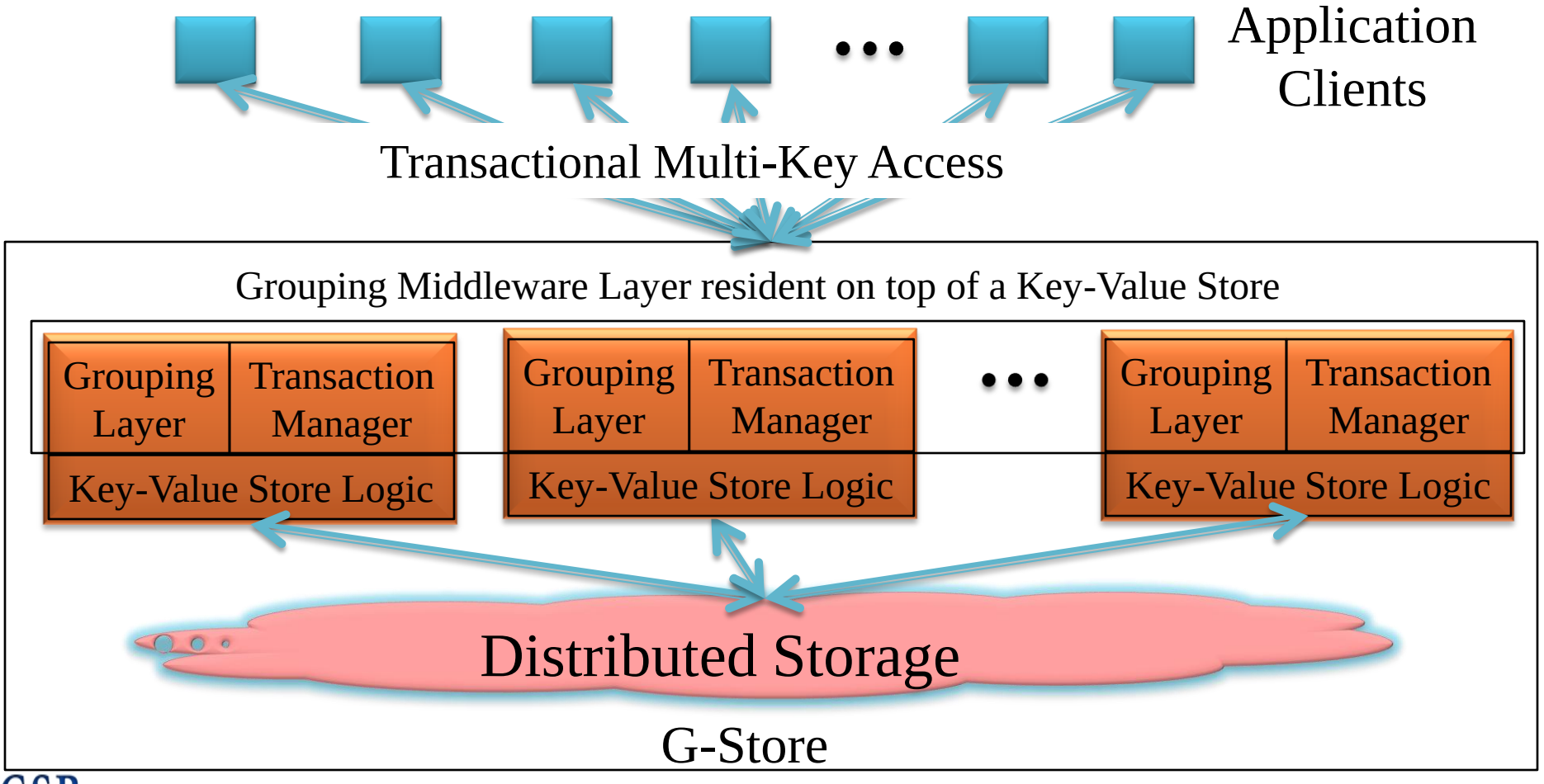
- Apps select any set of keys to form a group
- A granule of on-demand transactional access
- Data store provides transactional group access
- Non-overlapping groups


Horizontal Partitions of the Keys
Keys located on different nodes
Key Group
A single node gains ownership of all keys in a KeyGroup
Group Formation Phase

Key Grouping Protocol

- Conceptually akin to locking
- Allows collocation of ownership
- Transfer key ownership from followers to leader
- Guarantee safe transfer in the presence of system dynamics:
  - Dynamic migration of data and its control
  - Failures

Implementing GStore



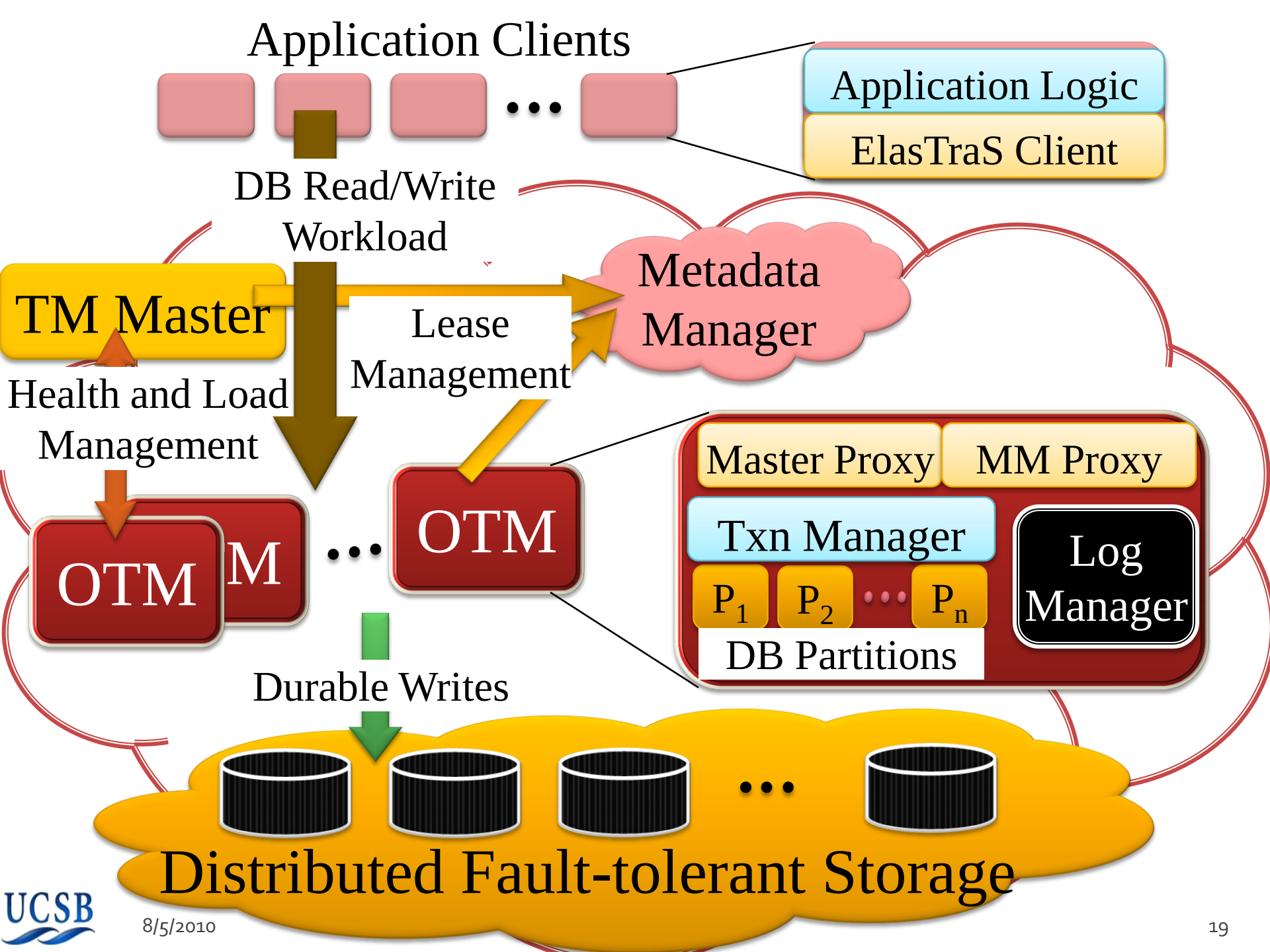
Data Fission: ElasTraS

Elastic Transaction Management

- Designed to make RDBMS cloud-friendly
- Database viewed as a collection of partitions
- Suitable for standard OLTP workloads:
  - Large single tenant database instance
    - Database partitioned at the schema level
  - Multi-tenant with large number of small databases
    - Each partition is a self contained database

Elastic Transaction Management

- Elastic to deal with workload changes
- Dynamic load balancing of partitions
- Automatic recovery from node failures
- Transactional access to database partitions


Application Clients
Application Logic
ElasTraS Client
DB Read/Write Workload
TM Master
Metadata Manager
Lease Management
Health and Load Management
OTM
OTM
OTM
Master Proxy
MM Proxy
Txn Manager
Log Manager
P_1
P_2
P_n
DB Partitions
Durable Writes
Distributed Fault-tolerant Storage

ElasTraS Prototype

- Performed in Amazon EC2
- Used TPC-C for evaluation
- Cluster size: 10 to 30 nodes
- Number of concurrent clients: 100 to 1800
- Data size: ~1T
- Each node: 8 cores, 7G RAM, 1.7T disk

- Max thruput: 0.2M TPC-C Xact/sec on 30 machines using HDFS & Zookeeper software.

Concluding Remarks

- Cloud Computing poses fundamental challenges to database researchers:
  - Scalability, Reliability and **Data Consistency**
  - Need to understand the new applications
  - **Live** Data migration is critical.
  - Challenging **multi**-node/center atomic operations
  - Clear characterization of **properties** & **guarantees**.