ShuttleDB: Database-Aware Elasticity in the Cloud

Sean Barker†‡, Yun Chi‡, Hakan Hacigumus‡, Prashant Shenoy†, and Emmanuel Cecchet†

University of Massachusetts Amherst†
NEC Laboratories America‡
Elastic Cloud Platforms

- Cloud platforms for scalable, high-performance applications and services

- Key property: **elasticity**
  - Increased system utilization
  - Decreased infrastructure cost
  - Handle dynamic workloads

![Diagram showing dedicated and elastic load shifting](image-url)
Database Clouds

- Cloud employs **virtual machines (VMs)**
  - Physical server partitioned into VMs
  - VMs contain database tenants

- “Large” tenants
  - Dedicated VMs

- “Small” tenants
  - Collocated within a VM
Database Cloud Elasticity

- **Option 1:** virtual machine-based
  - Transfer entire VM to larger server
  - Off-the-shelf, black-box approach
  - Designed for LAN migrations
  - Inefficient for small (collocated) tenants!

- **Option 2:** database-level elasticity
  - Operate directly on DB tenants
  - Not readily available off-the-shelf
  - Potential to address VM limitations

- Can migrate or replicate
ShuttleDB Problem Statement

- Which elasticity mechanisms are best for DB clouds?

- How can we exploit different elasticity mechanisms in a database cloud to efficiently meet the needs of diverse tenants?
Combining Elasticity Approaches

- Different approaches best in different scenarios!
  - Dedicated or collocated tenant?
  - LAN or WAN scaling?

- Scale-up (migration) or scale-out (replication)

<table>
<thead>
<tr>
<th></th>
<th>Dedicated Tenant (large)</th>
<th>Collocated Tenant (small)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LAN scale-up</strong></td>
<td>VM migrate</td>
<td>DB migrate</td>
</tr>
<tr>
<td><strong>LAN scale-out</strong></td>
<td>DB or VM replicate</td>
<td>–</td>
</tr>
<tr>
<td><strong>WAN scale-up</strong></td>
<td>DB migrate</td>
<td>DB migrate</td>
</tr>
<tr>
<td><strong>WAN scale-out</strong></td>
<td>DB replicate</td>
<td>–</td>
</tr>
</tbody>
</table>
Database-Aware Elasticity

- DB-level elasticity not out-of-the-box

- ShuttleDB implements a database-agnostic migration layer

**Objectives**

- Migrate *individual* tenants
- LAN or WAN capable
- “Live” migration
- Minimize data overhead
Database Live Migration Protocol

- Three-phase protocol

- (1) Copy database blocks
  - Continue handling workload

- (2) Update changed blocks
  - Multiple rounds

- (3) Switch workload to target
  - Minimal (<1s) downtime
Delta Migrations

- Transfer only DB delta when possible

- Outgoing migration
  - Save local snapshot

- Returning migration
  - Compare local snapshots
  - Transfer (small) delta
Automated Elasticity

- Automatically scale tenant capacity

- **When**: track latency, utilization
  - ARIMA forecast

- **Which**: cost-benefit
  - Tenant size vs workload

- **Where**: under-utilized machine
  - Prefer LAN over WAN

- **How**: small/large, LAN/WAN
  - VM or DB migration as appropriate
Outline

- Motivation
- Cloud database architecture
- Live migration mechanisms
- Automated elasticity
- Prototype evaluation
- Conclusions
Prototype Implementation

- Implemented on top of MySQL (DB) and Xen (VM)
  - Modified Yahoo Cloud Serving Benchmark for evaluation
Evaluation: Large Tenants

- **Performance over a LAN**
  - Baseline VM migration, VM (+DB), and DB-aware migration
- **Large (20 GB) tenant in dedicated VM**

- **VM migration suitable for dedicated tenants**
Evaluation: Small Tenants

- Twenty small (1 GB) tenants
  - Migrate single tenant

- DB-aware migration important for small tenants
Scaling up to a remote datacenter (Amazon EC2)

- Later delta migration to scale back

ShuttleDB efficiently scales across a WAN

87% reduction w/ delta migration

Only 1s downtime
Two spare servers, two workload spikes
  • Scale-up (DB migration) followed by scale-out (VM replication)

Migration, replication combined to increase elasticity
Related Work

- Live migration of virtual machines
  - LAN-based techniques [Clark - NSDI 05, Bradford - VEE 07]
  - WAN extensions [Wood - SIGPLAN 11]

- Database migration systems
  - Shared-nothing systems [Elmore - SIGMOD 11]
  - NAS-based systems [Das - PVLDB 11]

- Cloud databases and multitenancy
  - Provisioning using cost models [Cecchet - VEE 11]
  - Workload-aware multitenant provisioning [Curino - SIGMOD 11]
Conclusions

- Elasticity as a key property of cloud applications
  - Unique challenges in elastic cloud databases

- ShuttleDB combines virtual machine techniques with database-level migration for flexible elasticity
  - Scale-up and scale-out
  - Usable over LAN or WAN
  - Automated scaling based on system performance

- Prototype on an off-the-shelf DBMS
  - Minimal (<2s) downtime during scaling
  - Rapid response to workload changes
Questions?