How to Scale Out MySQL on EC2 or RDS

Victoria Dudin, Director R&D, ScaleBase

Boston AWS Meetup
August 11, 2014
Victoria Dudin

- Director of R&D, ScaleBase
- 15 years of product development experience
- Previously responsible for managing a 24/7 R&D team at Comverse
- Holds a MS in Computer Science from Harvard University
Agenda

• Helicopter drawing registration
• Scale out MySQL on EC2 and RDS for new and existing apps
• Case study
• Demo
Scale Out MySQL on EC2 and RDS for New and Existing Apps
Leverage Distributed Workloads on AWS

Distributed databases are a great fit for the cloud!

- Elastic computing
- Elastic storage
- Commodity hardware
- Shared nothing
- Only distributed databases are elastic!

Working with multiple less expensive machines is key to being cheap in the cloud
RDS/EC2 Scalability Challenges

- Increasing number of users and transactions
- More data
- Growing number of writes
- Insufficient capacity
- Declining throughput
- Performance inconsistencies
Dealing with Intensive Workloads: RDS Solution

Amazon says:

“Amazon RDS for MySQL – Enables you to scale out beyond the capacity of a single database deployment for read-heavy database workloads.”*

Options to improve RDS scalability and performance:

1. Larger RDS instances with more compute power and memory
2. Provisioned IOPS for faster and guaranteed I/O
3. Leverage read replicas for read-intensive applications
4. “You can implement partitioning, thereby spreading your data across multiple DB Instances”**

Partition Data Across Multiple RDS

Q:
Is there any documents that describe the partition DB across multiple RDS?

I need to use DB with more 1TB but exist a limitation during the create process, but I read in the any FAQ that you need to partition database, but I don't find any documents that describe it.

A:
Hi,

DB partitioning/sharding is not an official feature of Amazon RDS or MySQL, but a technique to scale out database by using multiple database instances. The appropriate way to split data depends on the characteristics of the application or data set. Therefore, there is no concrete and specific guidance.
Dealing with Intensive Workloads: RDS Solution

Shortcomings:

1. The largest instance has still limited compute power and memory
   - High-Memory Quadruple Extra Large DB Instance: 68 GB of memory, 26 ECUs (8 virtual cores with 3.25 ECUs each), 64-bit platform, High I/O Capacity, Provisioned IOPS Optimized: 1000Mbps

2. Provisioned IOPS are limited (and expensive)

3. You need to modify your application to utilize read replicas

4. RDS has no solution for scaling writes

Now what?

Amazon says: “scale out your data across multiple DB instances”
Data Distribution on AWS

10,000 tps

2,500 tps

2,500 tps

10,000 tps

2,500 tps

2,500 tps

2,500 tps
Example of a poorly chosen data distribution policy

A bad data distribution policy requires many transactions/queries to access/collect data from multiple databases.

- Data and concurrent transactions low on each MySQL instance
- More databases as data/concurrency grows.
Desired benefits with a well chosen distribution policy

A good data distribution policy ensures that every specific transaction/query is complete within a specific database.

- Data and concurrent transactions low on each MySQL instance
- More databases as data/concurrency grows.
Case Study: XillO
What Does The Tesseract Application Do?

• High volume video distribution on the YouTube platform
  – 2,000,000 business transactions per day (video shares)
  – Each business transaction (video share) comprised of multiple operations (back-end DB, middleware, front-end API)
  – 25,000,000 video shares per campaign (30 days)
  – Multiple campaigns running in parallel
  – Detailed analytics reporting & data reconciliation

• Highly targeted demographic delivery
  – Gender
  – Age
  – Location
  – Genre
  – Interests
  – User Profile Updates
XIII O Requirements: A Distributed Relational Database on AWS

- Single logical database
- Database becomes elastic
- Availability & fault tolerance
- Workload Mix
  - reads AND writes
- Developer and administrator friendly
  - No code changes
  - All MySQL tooling and ecosystem works.
Tesseract – One Campaign Pod

ScaleBase Software
  • Amazon EC2 (XL)

ScaleBase Configuration DB
  • Amazon RDS (S)

Database Shard Array
  • Sharded Tables
  • Amazon RDS (L)

Master Database
  • Non-Sharded Tables
  • Amazon RDS (L)
ScaleBase Delivered for Tesseract

- Achieve more throughput than possible with the largest single database machine
- No changes in the application
- Spin up in hours

- Versatile to match all 4 use cases; yet optimized for scale and throughput
- Changing mix of r/w

- Working with multiple less expensive database machines is key to being cheap in the cloud

- Distribution, elasticity, full control over distribution vectors; yet deliver cross-DB multiple-campaign analytics reconciliation
Demo
Demo Agenda

- DBT2 Benchmark
- Data Distribution Policy
- TPCC runs
DBT2 Benchmark

• We chose DBT2 because it is a well-known benchmark that is frequently used for evaluation of performance and scalability of MySQL DB environments.

• From hardware sizing perspective, approx. 20 warehouses and 12 to 15 connections should typically saturate a single Large RDS instance.

• Some results:

  **Results for 20 warehouses:**
  1 Shard 12 conns: 8.570 NOTPMs 24 conns: 9.924 NOTPMs
  2 Shards 12 conns: 10.137 NOTPMs 24 conns: 15.023 NOTPMs

  **Results for 40 warehouses:**
  1 Shard 12 conns: 8.785 NOTPMs 24 conns: 9.159 NOTPMs
  2 Shards 12 conns: 10.632 NOTPMs 24 conns: 16.084 NOTPMs
DBT2 Distribution Policy
TPCC Runs

- Single MySQL instance
- Additional Instances
About ScaleBase
A Distributed Database Platform
Built on MySQL - Optimized for the Cloud

“We welcome ScaleBase to AWS Marketplace with their Distributed Database Platform. Customers with growing application and MySQL database requirements can benefit from migrating to a scalable, virtualized deployment on the cloud.”

Terry Hanold
VP, Cloud Commerce, AWS
ScaleBase Benefits

- **Scale out** to an unlimited # of users, data and transactions
  - Dynamic online database rebalancing

- **Reduce time-to-market**
  - Deploys in minutes and requires no changes to the application

- **Built on MySQL/InnoDB 5x (maintain ACID)**
  - Same tools, know how, methodologies

- **Optimize for the cloud**
  - Geo-distribute your data to any private/public cloud

- **Simplified, centralized management**
  - Automatic analysis, migration, configuration and schema change

- **Ensure high availability**
  - Transparent failover while preserving data consistency (ACID)
ScaleBase Architecture
ScaleBase Editions

Free Edition
Scale Out in Minutes
Limited by Deployment Size

Startup Edition
Think Big from the Start
Free for Startups

Enterprise Edition
Deploy in Your Environment
Pay as you Scale

Also available on: 
amazon.com  

rackspace  

IBM Cloud / marketplace
ScaleBase Analysis Genie

• Determines the best way to scale out a single MySQL instance to a distributed database
• Creates the optimal distribution policy for your specific app by analyzing your schema and queries
• Presents you with efficiency scores and scale out guidelines on how to distribute your
• Combines relational integrity of MySQL with the scalability of a modern distributed, architecture
• Real time policy development and control
Questions and Answers

Victoria Dudin

www.scalebase.com
victoria.dudin@scalebase.com
Helicopter Drawing

ScaleBase Resources
scalebase.com
scalebase.com/resources
scalebase.com/blog

info@scalebase.com       617. 630.2800

Boston AWS Meetup
August 11, 2014
Appendix
The Rise of Distributed Databases

NoSQL
- gain distributed data capabilities
- throw out the relational model

ScaleBase converts the world’s most popular open source database (MySQL/InnoDB) into a distributed relational database system.