Accelerate Database Development and Testing on AWS

Vlad Vlasceanu, Principal Database Specialist Solutions Architect, AWS

8/21/2018
Webinar Agenda

- Overview of Amazon Aurora
- Agility enhancing features of Amazon Aurora
- Accelerating database development and testing with:
  - Aurora Serverless
  - Fast Database Cloning
  - Backtrack
  - Performance Insights
Overview of Amazon Aurora
Amazon Aurora

A relational database re-imaged for the cloud:

- **Speed** and **availability** of high-end commercial databases
- **Simplicity** and **cost-effectiveness** of open source databases
- Drop-in **compatibility** with MySQL and PostgreSQL
- Simple **pay as you go** pricing

**Delivered as a managed service**
“Intuit invests significantly to own and operate high-end commercial databases underpinning our business. Until now, there just wasn’t a real alternative to obtain the reliability and performance our customers need. Amazon Aurora is a game-changer for us: providing the performance and availability features that rival expensive on-premises databases and SANs at a significantly lower price point. The RDS management capabilities on top of Amazon Aurora will allow us to focus our resources and energy on what matters most – building great applications and delighting our customers.”

Troy Otillio, Director, Public Cloud, Intuit
Amazon Aurora: re-imaging the relational database

1. Scale-out, distributed architecture
   Purpose-built, log-structured and failure tolerant distributed storage system

2. Service-oriented architecture leveraging AWS services
   Integrated with AWS Lambda, Amazon S3, AWS IAM and Amazon CloudWatch

3. Automate administrative tasks – fully managed service
   You focus on schema design, query construction and optimization, we do the rest

4. Cloud-native capabilities simplify use
   Auto-scaling, reduced lag readers, continuous backups, storage grows with workload

Amazon Aurora is fast...
   5× faster than MySQL; 3× faster than PostgreSQL
<table>
<thead>
<tr>
<th>How did we achieve this?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DO LESS WORK</strong></td>
</tr>
<tr>
<td>Do fewer I/Os</td>
</tr>
<tr>
<td>Minimize network packets</td>
</tr>
<tr>
<td>Cache prior results</td>
</tr>
<tr>
<td>Offload the database engine</td>
</tr>
<tr>
<td><strong>BE MORE EFFICIENT</strong></td>
</tr>
<tr>
<td>Process asynchronously</td>
</tr>
<tr>
<td>Reduce latency path</td>
</tr>
<tr>
<td>Use lock-free data structures</td>
</tr>
<tr>
<td>Batch operations together</td>
</tr>
<tr>
<td><strong>NEW CAPABILITIES AND FEATURES</strong></td>
</tr>
<tr>
<td>Elasticity enhancements</td>
</tr>
<tr>
<td>Instant crash recovery</td>
</tr>
<tr>
<td>Zero downtime patching</td>
</tr>
<tr>
<td>Adaptive thread pooling</td>
</tr>
<tr>
<td>Fast DDL operations</td>
</tr>
<tr>
<td>Engine performance enhancements</td>
</tr>
<tr>
<td>Monitoring and management features</td>
</tr>
<tr>
<td>Fast database cloning, backtrack</td>
</tr>
</tbody>
</table>
Accelerating database development and testing

1. Cost optimized lower environments, on-demand accessible
   Ability to scale dev/test/staging database environments up or down based on usage

2. Fast access to replicas of production data sets at scale
   Data movement is slow with large data sets which impedes agility

3. Ability to quickly undo mistakes
   Mistakes are hard to fix, and take a long time to revert

4. Easily assess performance of workload or changes to it
   Impact of application and query patterns varies as workload scale increases
Accelerate development and testing with Amazon Aurora

Amazon Aurora Serverless

Fast Database Cloning

Backtrack

Performance Insights
Introducing Amazon Aurora Serverless
Why Amazon Aurora Serverless?

Because sizing a relational database is hard...

**PROVISION FOR PEAK**
Handles spikes in usage
... But costs more

**PROVISION FOR AVERAGE**
Costs less
...But may not meet peak demand

This choice affects development and testing patterns, too!
Amazon Aurora Serverless is different...

1. Simple provisioning
   No need to indicate storage or compute capacity needed

2. Starts up on demand, shuts down when not in use
   Data durably stored using the aurora purpose-built storage system, independent of DB engine

3. Seamlessly scale with no servers to manage
   No need to provision capacity, application connections are maintained

4. Pay only for what you use
   Billed per second for the database capacity used at that time

5. Drop-in compatible with existing workloads
   No application changes needed, available for MySQL-compatible workloads
Aurora Serverless use cases

Development and test databases:
- Easily provisioned
- Cost savings when DBs are not in use
- Simplify dev/test pipelines

Infrequently used applications (e.g. low-volume blog site)

Applications with variable load - peaks of activity that are hard to predict (e.g. news site)
Aurora Serverless architecture

- Application talks to MySQL compatible endpoint
- Fleet of routers manage queue, client connections and route DB traffic
- Instance handles database operations
- Data kept durable and highly available on Aurora storage volume
- When scaling thresholds are reached we scale the instance substituting with capacity from warm pool
- Scaling operations are transparent to applications
- You configure min. and max. capacity, whether and when to pause DB if there’s no activity
Aurora Serverless pricing

You pay for capacity your database consumes while it is active

Database capacity is measured in **Aurora Capacity Units** (ACUs)

Flat rate per second of ACU usage, with a minimum of 5 minutes of usage each time the database is activated

Storage and I/O are billed the same as with Aurora instances
Aurora Serverless demo
Fast Database Cloning
Get faster access to copies of your data so you can:

1. Test changes in pre-production on relevant data sets
2. Reorganize a database with minimal production impact
3. Save a point in time snapshot for data analysis without impacting production systems
Fast database cloning with Amazon Aurora

Clone database cluster without copying the data
- Available for both MySQL and PostgreSQL compatible versions of Amazon Aurora
- Creation of clone is near instantaneous
- Data copy happens only on write, when the original and cloned volume start to differ
- Operations on clone do not affect the source cluster
- Up to 15 clones from the same source
- Pay only for the data storage difference on the clone
Fast database cloning demo
Backtrack Databases
Easily undo changes to your data

1. Reduce risk of database changes at scale
2. Undo unintentional DML and DDL changes
3. Mitigate risk of malicious changes to your data
4. Avoid time consuming data restore from backups
How does database backtrack work?

- We keep periodic snapshots of each segment; we also preserve the logs.
- For backtrack, we identify the appropriate segment snapshots.
- Apply log streams to segment snapshots in parallel and asynchronously.
Going back in time with database backtrack

- Backtrack is not destructive
- You can backtrack multiple times to find the right point in time
- Pay for the volume of change records retained for the desired duration (up to 72 hours)
- Available for Aurora MySQL 5.6 compatible
Backtrack demo
Performance Insights
Monitor the performance of your queries

1. Establish a baseline of acceptable/desired query performance
2. Assess performance impact of workload changes
3. Troubleshoot poor performance, identify bottlenecks
4. Effective capacity planning
Performance Insights

- Easy and powerful dashboard showing load on your database
- Helps you identify source of bottlenecks: top SQL queries, wait statistics
- Adjustable time frame (hour, day, week, month)
- 7 days of performance data history free – perfect for dev & test
- up to 2 years of long term retention for production use cases
Performance Insights demo
Summary

Aurora’s cloud-native capabilities simplify database operations

Accelerate database development and testing using Aurora features such as:

- Aurora Serverless
- Fast database cloning
- Database backtrack
- Performance Insights
Thank you!

For additional details on Amazon Aurora visit: https://aws.amazon.com/rds/aurora/