GenRocket

SOLUTION SPOTLIGHT

API PERFORMANCE TESTING AT SCALE

Background

A global financial services company annually needed to conduct performance testing and synthetic data generation for an internal API. The scenario involved assessing the functionality and scalability of an API where transaction records were stored in a large-scale Postgres database. The database held a staggering 7 billion records, but the sensitive financial data and personally identifiable information (PII) contained in these records made typical production data masking untenable due to potential security risks. Creating an ad hoc script to mask the data was also deemed too time-consuming.

GenRocket's Synthetic Test Data Automation platform was chosen to efficiently generate the large volume of synthetic test data required, while ensuring data security and referential integrity.

Challenges

The financial services company faced several challenges in this performance testing initiative:

- Generating millions of synthetic test data records to adequately stress-test the API
- Ensuring a wide variety of conditional data for comprehensive testing
- Provisioning data that could not be pulled from production environments due to PII concerns
- Enabling flexibility to modify data attributes and logic for future testing needs
- Avoiding the time-consuming process of manual test data creation or complex scripting

GenRocket Solution

GenRocket addressed these challenges through its powerful test data generation capabilities:

1. **Domain Modeling with Custom Generation Logic**
   - Test Data Cases were designed using the GenRocket platform to specify the data generation rules for various attributes such as transaction dates, IDs, company names, and compliance ratings.
   - GenRocket's flexible platform allowed for precise definition of data for testing business rules and data variations for both positive and negative test scenarios.

2. **High-Speed Data Generation with Scenario Thread Engine**
   - To expedite the creation of the 20 million user records, GenRocket's Scenario Thread Engine was leveraged to perform high speed parallel data generation.
   - Four concurrent threads were used to generate data across four scenarios, with the output organized into six-month intervals per the client's requirements.
   - This multi-threaded approach dramatically reduced the time needed to create the test data.

3. **Streamlined Execution and Database Integration**
   - The entire data generation process was executed with a single command, greatly simplifying the client's workflow.
   - Integration with the client's Postgres database was seamlessly handled through a GenRocket properties configuration file specifying the database connection details.
   - This enabled the direct insertion of the generated test data into the database, establishing an end-to-end testing process.

Benefits and Achievements

By implementing GenRocket, the financial company realized significant benefits:

- **Massive Scale and Speed**
  GenRocket generated test data for 20 million users across four tables, totaling 1.2 billion attribute values. The same process that previously took 2 days to generate 1 million users was reduced to just 13 hours for 20 million users—a 74x speed improvement.
- **Comprehensive Test Coverage**
  The synthetic data provided by GenRocket enabled thorough testing of the API across a wide range of scenarios and edge cases, ensuring robustness and reliability.
- **Seamless Database Integration**
  Direct insertion of test data into the Postgres database was made possible through GenRocket's flexible configuration options, establishing an efficient end-to-end testing process.
- **Time and Effort Savings**
  GenRocket's Test Data Automation platform dramatically reduced the time and manual effort required compared to traditional data provisioning approaches, allowing the testing team to focus on higher-value tasks.

Conclusion

GenRocket proved to be instrumental in enabling the global financial services company to conduct extensive performance testing of their internal API. The platform's ability to rapidly generate huge volumes of synthetic test data while maintaining referential integrity and complex data relationships showcased its versatility and effectiveness.

By leveraging GenRocket, the company significantly reduced their test data generation time, achieved comprehensive test coverage, and established an efficient end-to-end testing process integrated with their database. The successful execution of this testing initiative demonstrated GenRocket's value in supporting the performance and scalability needs of enterprise-scale applications.

REQUEST A DEMO