Capturing bugs in extreme stress testing:
Improving software quality in SAP HANA with Undo

Greg Law, Undo
Stefan Baeuerle, SAP SE
June 15th, 2018
The challenge:
Stress Testing for SAP HANA
Problem Setup

- **SAP HANA as an enterprise-class, in-memory database management system**
  - OLTP and OLAP, relational and noSQL functionality in a single system
  - Complex codebase

- **Very strict quality and governance processes**
  - Sophisticated continuous integration platform
  - Large functional and performance test harness (see Rehmann@RDSS 2014)

- **„Regular“ tests plus highly parallel, multi-user stress tests (PMUT)**
  - Arbitrary database operations (DML, DDL, etc) in parallel
  - High amount of stress for system resources
  - Complements other tests with explorative/non-deterministic testing
  - Similar approaches with other systems („chaos monkey“)
Problem Statement

- **Good**: PMUT captures sporadic problems not detected by other tests
  - Improves software quality by finding bugs early
- **Bad**: Non-deterministic nature makes reasoning very hard
  - PMUT workload is highly parallel and (pseudo-) randomized
- **Ugly**: Developers spend days/weeks to resolve underlying root cause
  - Challenge: Problem reproduction is extremely time consuming
  - Fixing often is trivial afterwards
- **Goal**: Avoid overhead for scenario/problem reproduction
The solution:
Live Recorder from Undo
CCTV for program execution

- Record program’s execution
- Replay at any time
- Freeze-frame
- Single-step backwards
- Single-step forwards

Find out why the program made the decisions it did.
int i; i = 90;
for (i = 0; i < cache_size; ++i)
{
    if (cache[i].number == number)
    {
        /* Cache hit. */
        return cache[i].sqroot;
    }
}

/* Cache miss. Find correct result and populate a few cache entries. */
int sqroot = 0; sqroot = 0;
int number2; number2 = 1;
for (number2 = number1 - 1; number2 < number1 + 1; ++number2)
{
    int sqroot2 = (int) (sqrt(number2)); sqroot2 = -2147483648
    i = (int) (1.0 * cache_size * rand() / (RAND_MAX + 1.0));
    cache[i].number = number2;
    cache[i].sqroot = sqrt2;
    if (number2 == number)
    {
        /* This is our return value. */
        sqroot = sqrt2;
    }
}
return sqroot;
The solution

- SAP uses Live Recorder from Undo to record multi-user stress test (PMUT) runs
- When a failure occurs the recording is kept and handed over to developers to diagnose
- Turns the sporadic problem into a 100% reproducible
- SAP developers use Undo’s interactive reversible debugger – UndoDB – on the recording to diagnose the root cause of the problem
Some in-production defects captured in test using Live Recorder and diagnosed using UndoDB

- A number of sporadic memory leaks and memory corruption defects
- Several issues in the networking code, including the incorrect flushing of a receive buffer and sporadically releasing channels in cases of timeout, resulted in queries incorrectly aborting
- Incorrect parallel access to a shared data-structure which resulted in very subtle sporadic problems which were hard to reproduce
- A race condition in SAP HANA’s transaction management cache with the potential of incorrectly reusing cached session data
- Very sporadic race condition in SAP HANA’s asynchronous garbage collection for in memory table structures during table unloads under heavy system load
Wider perspective

- This isn’t just about test
- This isn’t just about SAP HANA
- Existing approaches are not good enough
- Recording program execution is a new approach
- Getting a recording of your test run enables new ways of investigating how your code is behaving
Thank you.

See us at booth 6 during the coffee break or after the workshops

Contact information:

**Stefan Bäuerle**
Chief Development Architect
SAP HANA Platform & Database
Stefan.Baeuerle@sap.com

**Greg Law**
CEO
Undo
greg@undo.io