ShadowDB:
A Replicated Database on a Synthesized Consensus Core

Nicolas Schiper, Vincent Rahli, Robbert van Renesse,
Mark Bickford, Robert L. Constable

Cornell University
October 7th, 2012
ShadowDB is an experiment on building fault-tolerant software using synthesized code
ShadowDB – The Big Picture

Primary-backup replication protocol

- primary
- f backups

BDB

(reconfigure replicas)

Synthesized consensus service (2/3-consensus)

3f+1 machines

Evaluator in SML

Nuprl terms

Evaluator in OCaml

Nuprl terms

(Hand-written)
Outline

• The Replication Protocol

• Synthesis of Consensus in Nuprl

• Preliminary performance results

• Future work
The Replication Protocol – Normal Case

1. send T
2. execute T
3. (Tid, order, updates)
4. store msg.
5. (ACK, Tid, order)
6. commit T
7. answer
8. (commit, Tid, order)
9. commit T

(seq. no, group incarnation)
The Replication Protocol – Recovery

network partition

reconfigure

reconfigure

States diverge!
The Replication Protocol – Recovery

- Network partition
- Reconfigure
- Synthesized consensus service (2/3-consensus)
- Propose (group inc, seq, primary, backups)
- Decide (group inc, seq’, primary’, backups’)

- Reconfigure

- Reconfigure
Synthesis of Consensus in Nuprl

- Consensus properties:
  - **Agreement**: no two processes decide differently
  - **Validity**: the decision is one of the proposals
  - **Non-blocking**: there exists an execution in which all crash-free processes decide
Synthesis of Consensus in Nuprl

- Specification in English → Pseudo-code specification
- Nuprl specification → EventML specification
- Synthesized code → Correctess proofs
- Correctess proofs → Untrusted code

Diagram arrows indicate the flow of processes.
vals: values received in the current round
locs: locations from which we received a msg.

```plaintext
when receiveMsg(group inc, round, value, sender) {
    if (first msg from sender in round of group inc)
        if (|vals| = 2 * f + 1)
            if all values are equal to some x then decide x
            if a value x appears as a majority then retry with x
        else retry with first value received in round
}
```

```plaintext
let when_quorum si loc (((s, i), v), sender) (vals, locs)
    if si = (s, i) & !(deq member (op =) sender locs)
        then if length vals = 2 * f + 1
            then let (k, x) = possmaj valeq vals v in
                if k = 2 * f + 1
                    then { decided'send loc (s, x) }
                else { retry'send loc ((s, i+1), x) }
            else {};
        else {};
```
Preliminary Performance Results

- ShadowDB’s performance is good ($f = 1$)
  - read-only: 79% of single node throughput (>9K TPS)
  - updates: 45% of single node throughput (>500 TPS)

- Recovery time is “OK” for ShadowDB
  - Running an instance of consensus took ~3.5 seconds at time of submission
  - It currently takes ~0.6 second
  - This is too slow to synthesize bigger parts of ShadowDB
  - More optimizations coming up!
Future work

• Synthesize the ShadowDB replication protocol

• Diversity in time
  • We synthesized Paxos
  • BFT consensus
  • Change the participants of the consensus service

• Diversity in space
  • Diversify replica (e.g. data structures, sorting algorithms...)

• Evaluate diversity
Questions?

Thank you!