Generating Correct-by-Construction Distributed Implementations from Formal Maude Designs

Si Liu[1] Atul Sandur[2] José Meseguer[2]
Peter Ölveczky[3] Qi Wang[2]

(IFIP WG 1.3 meeting, Jan 19, 2022; work originally presented at NFM'20)

[1]ETH Zürich
[2]University of Illinois at Urbana-Champaign
[3]University of Oslo

Background: Maude (I)

Maude

- based on rewriting logic
- expressive, simple, and general
- applied to wide range of systems
  - cloud transaction systems
  - semantics of programming and modeling languages
  - electronic contracts
  - large distributed systems protocols
  - systems biology (Pathway Logic)
  - cryptographic protocols (Maude-NPA)
  - ...

Background: Maude (II)

Maude

- explicit-state simulation, reachability analysis, LTL model checking
- "symbolic" analysis:
  - narrowing
  - rewriting modulo SMT

Background: Maude (II)

Maude

- explicit-state simulation, reachability analysis, LTL model checking
- "symbolic" analysis:
  - narrowing
  - rewriting modulo SMT
- object-based specification for distributed systems
  - state multiset of objects and messages

Example: Token-Ring Mutex

Example

```
rl [executeInCS] :
   (msg token from O to O')
   < O' : Node | state : waiting >
 =>
   < O' : Node | state : executingInCS > .


rl [passOnToken] :
   (msg token from O to O')
   < O' : Node | state : outsideCS, nextNode : O'' >
 =>
   < O' : Node | >
   (msg token from O' to O'') .
```

Performance Estimation

Statistical model checking via (e.g.) PVeStA

Performance Estimation

Statistical model checking via (e.g.) PVeStA

- model-based performance "curves" / "comparisons" consistent with actual implementations

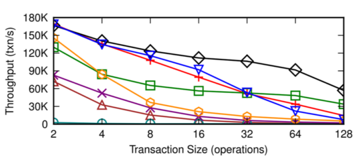
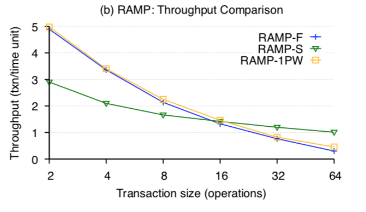


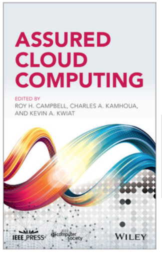
Fig. 3. RAMP-F (blue) vs RAMP-S (green): throughput under varying transaction size by the Java implementation-based evaluations in [16].



Background: Talk at IFIP Meeting 2019

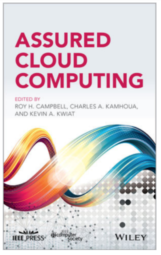
Maude for cloud-based transaction systems

- UIUC Center for Assured Cloud Computing



Background: Talk at IFIP Meeting 2019

Maude for cloud-based transaction systems



- UIUC Center for Assured Cloud Computing
- Correctness and performance estimation
  - Google Megastore (+ modified design)
  - Apache Cassandra (+ alternative design)
  - Apache Zookeeper
  - UC Berkeley's RAMP transactions (+ variations)
  - Walter, Jessy, P-Store, ...
  - ROLA (new design for new consistency model)

Today's Talk

So far

Maude specifications of correct designs with promising performance

Today's Talk

So far

Maude specifications of correct designs with promising performance

Today

Synthesize correct-by-construction distributed implementation of such Maude specification

Today's Talk

So far

Maude specifications of correct designs with promising performance

Today

Synthesize correct-by-construction distributed implementation of such Maude specification

- distributed (prototype?) implementation for free
- correct
- analyze with real workloads (e.g., YCSB)
- latency due to communication instead of computation (?)

Obtaining Distributed Maude Implementations

Goal

Synthesize distributed Maude implementation

Obtaining Distributed Maude Implementations

Goal

Synthesize distributed Maude implementation

External Objects in Maude

- Maude object can send/receive messages to/from external objects
- TCP/IP socket manager is one external object
- Maude instances on different machines can communicate
- Communication with outside world (e.g., YCSB)

Obtaining Distributed Maude Implementations

Goal

Synthesize distributed Maude implementation

External Objects in Maude

- Maude object can send/receive messages to/from external objects
- TCP/IP socket manager is one external object
- Maude instances on different machines can communicate
- Communication with outside world (e.g., YCSB)

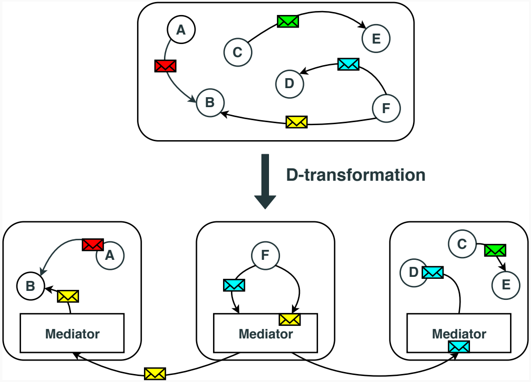
Questions

1. Proving socket-based implementation correct?
2. Is resulting implementation useful/efficient?

D Transformation

The D Transformation

- Middleware for communication
  - between Maude sessions
  - with external objects
- Implementation
  - mediator object added to each Maude session



Transformation $D : (M, init, di) \mapsto D(M, init, di)$

- Input:
  - object-oriented Maude module M defining actor system
  - initial state $init$
  - distribution information di

Transformation $D : (M, init, di) \mapsto D(M, init, di)$

- Input:
  - object-oriented Maude module M defining actor system
  - initial state $init$
  - distribution information di
- Output for each distributed Maude session (ip, i):
  - Maude program $D(M, init, di)$
  - initial state $init_{D_{di}}(ip, i)$

Correctness of D Transformation

- Verifying $(M, init, di) \mapsto D(M, init, di)$ requires verifying TCP/IP sockets and their implementation in Maude

Correctness of D Transformation

- Verifying $(M, init, di) \mapsto D(M, init, di)$ requires verifying TCP/IP sockets and their implementation in Maude
- We instead define abstract model of socket communication
- Abstract model $D_0(M, init, di)$ of distributed model $D(M, init, di)$

Correctness of D Transformation

- Verifying $(M, init, di) \mapsto D(M, init, di)$ requires verifying TCP/IP sockets and their implementation in Maude
- We instead define abstract model of socket communication
- Abstract model $D_0(M, init, di)$ of distributed model $D(M, init, di)$

Theorem

$\mathcal{K}(D_0(M, init, di), init_{D_0})$ and $\mathcal{K}(M, init)$ are stuttering bisimular

Implementation

Implementation

- D transformation
  - 300 LOC in Maude
- Deployment
  - Python-based prototype
  - automated deployment and run on distributed machines
- Foreign object
  - Yahoo! Cloud Serving Benchmark (YCSB)
    - open standard for performance evaluation of data stores

Case Studies

Case Studies: ROLA & NO_WAIT

- ROLA
  - distributed update atomic transactions
  - 850 LOC in Maude
  - No existing implementation
- NO_WAIT
  - lock-based distributed transactions providing serializability
  - optimized implementation at CMU & MIT
  - 12K LOC in C++ (vs 600 LOC in Maude)

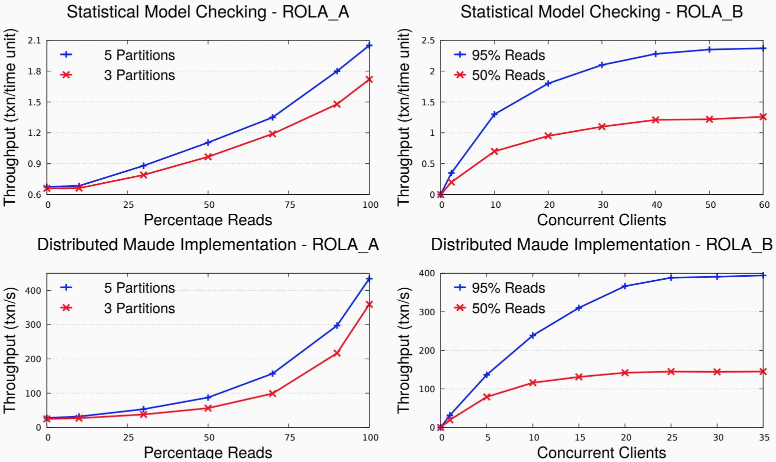
Performance Comparison

Model-based statistical model checking vs Implementation-based evaluation

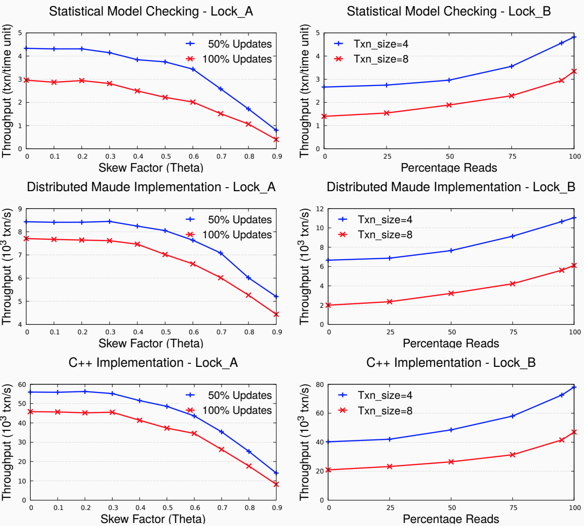
- Consistent trends
  - statistical model checking
  - distributed Maude implementation
  - conventional distributed implementation
- Actual performance values
  - distributed Maude implementation
  - conventional distributed implementation (C++)

Results for ROLA

- Similar trends for 2 sets of experiments



Results for NO_WAIT



- Similar trends
- Maude implementation (only?) 6 times slower than C++ implementation

Take Away

- Automated transformation of Maude specifications to correct-by-construction Maude distributed implementation
  - broad class of systems
  - performance analysis on real workloads (YCSB)
  - correctness proof uses simplified model of sockets

Take Away

- Automated transformation of Maude specifications to correct-by-construction Maude distributed implementation
  - broad class of systems
  - performance analysis on real workloads (YCSB)
  - correctness proof uses simplified model of sockets
- Specification and implementation in Maude
  - correctness proof easy

Take Away

- Automated transformation of Maude specifications to correct-by-construction Maude distributed implementation
  - broad class of systems
  - performance analysis on real workloads (YCSB)
  - correctness proof uses simplified model of sockets
- Specification and implementation in Maude
  - correctness proof easy
- One artifact for:
  - correctness checking
  - performance prediction
  - distributed implementation

Take Away

- Automated transformation of Maude specifications to correct-by-construction Maude distributed implementation
  - broad class of systems
  - performance analysis on real workloads (YCSB)
  - correctness proof uses simplified model of sockets
- Specification and implementation in Maude
  - correctness proof easy
- One artifact for:
  - correctness checking
  - performance prediction
  - distributed implementation
- Maude distributed implementation > 6 times slower than optimized C++ implementation by Stonebraker et al.
  - proof-of-concept prototype vs optimized C++ implementation
  - extra layer around foreign objects?
  - socket implementation in Maude?