Quantitative Analysis of Consistency in NoSQL Key-value Stores

Si Liu, Son Nguyen, Jatin Ganhotra, Muntasir Raihan Rahman
Indranil Gupta, and José Meseguer

September 2015
International Conference on Quantitative Evaluation of Systems (QEST), 2015

NoSQL Systems

- Growing quickly
  - $3.4B industry by 2018

- Apache Cassandra
  - Among top 10 most popular database engines in September 2015
  - Top 1 among all Key-value/NoSQL stores (by DB-Engines Ranking)

- Large scale Internet service companies rely heavily on Cassandra

  - e.g., IBM, eBay, Netflix, Facebook, Instagram, GitHub

Predicting Cassandra Performance...

- …Is Hard. Today's options:
  - **Deploy** on Real Cluster
    - Many man-hours
    - Non-repeatable experiments
  - **Prove theorems on paper**
    - Very hard to do for performance properties
  - **Simulations**
    - Large and unwieldy
    - Take time to run
    - Hard to change (original Cassandra is 345K lines of code)

Our Approach

1. Write formal model of Cassandra (in Maude language)
2. Use statistical model-checking to measure performance of Maude model
3. Validate results with real-life deployment
4. In future, use model to predict performance

– First step towards a long-term goal: a library of formal executable building blocks which can be mixed and matched to build NoSQL stores with desired consistency and availability trade-offs

How we Go about it

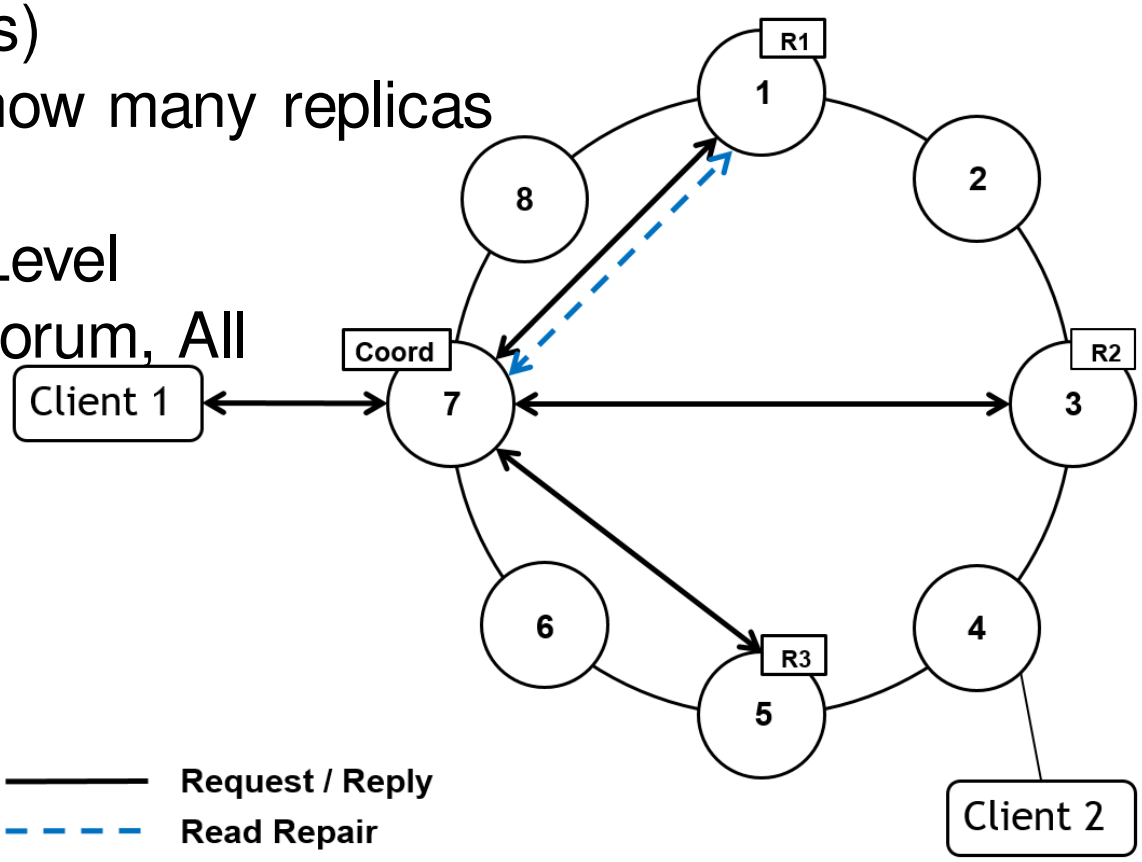
We use:

1. **Maude**
   - Modeling framework for distributed systems
   - Supports rewriting logic specification and programming
   - Efficiently executable

2. **PVeStA**
   - Statistical model checking tool
   - Runs Monte-Carlo simulations of model
   - Verifies a property up to a user-specified level of confidence

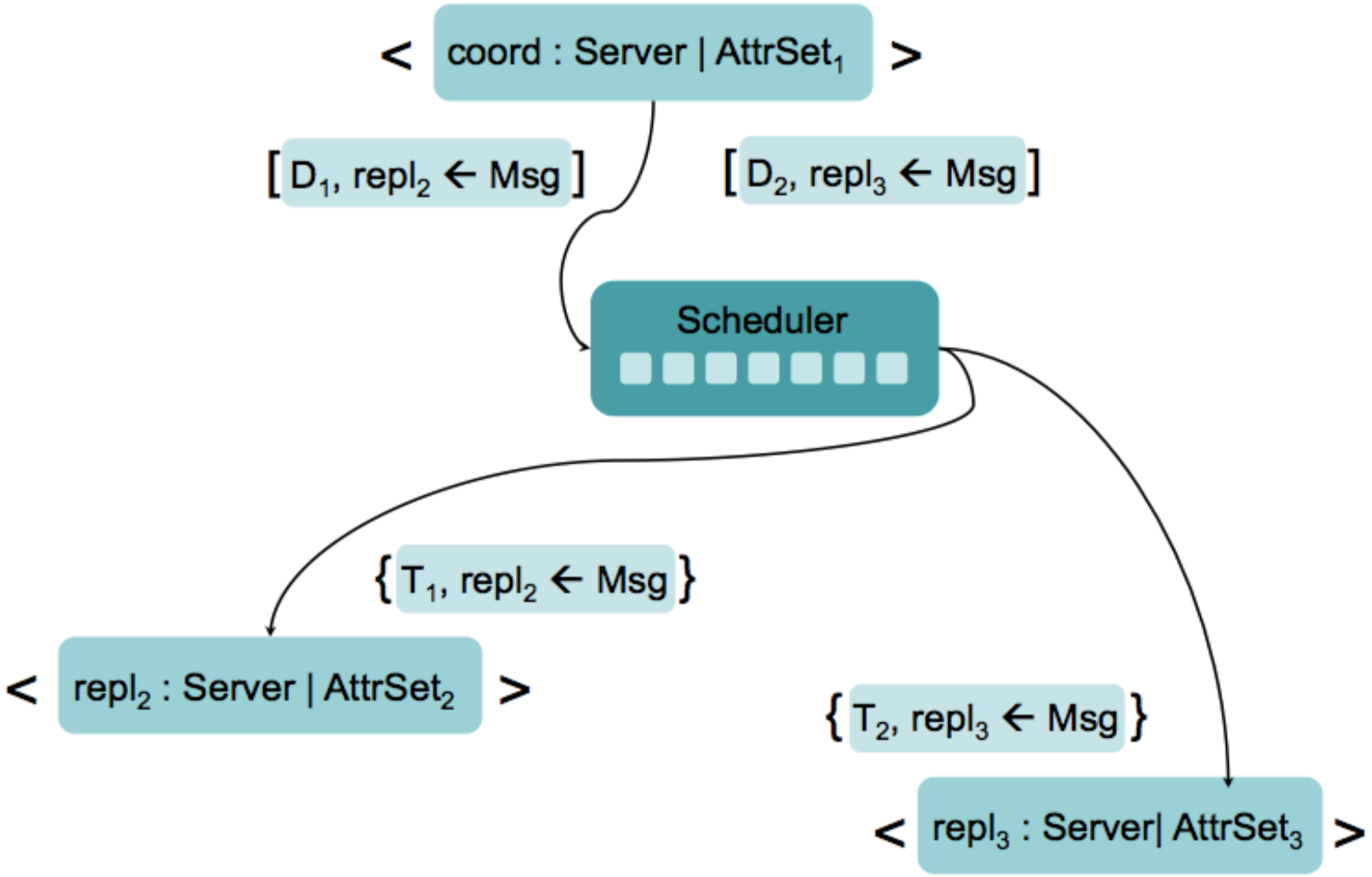
Apache Cassandra Overview

- Cassandra is deployed in data centers
- Each key-value pair replicated at multiple servers
- Clients can read/write key-value pairs
- Read/write goes from client to Coordinator, which forwards to replica(s)
- Client can specify how many replicas need to answer
  - Consistency Level
  - E.g., One, Quorum, All

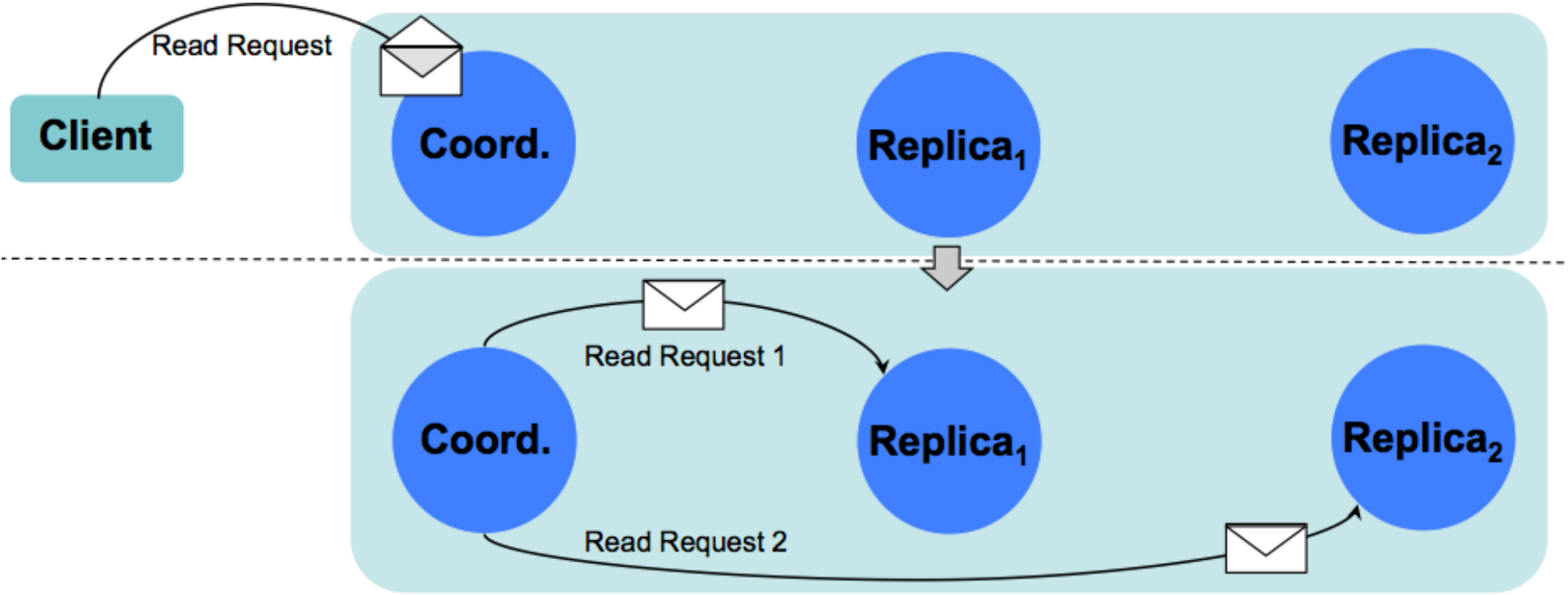


Cassandra Model in Maude: Reads & Writes

- The distributed state of Cassandra model is a “multiset” of Servers, Clients, Scheduler and Messages



Cassandra Model in Maude: Requests




```
crl [COORD-FORWARD-READ-REQUEST] :
   < S : Server | ring: R, buffer: B, ...>
   {T, S <- ReadRequestCS(ID,K,CL,A)}
 =>
   < S : Server | ring: R, buffer:
                  insert(ID,fac,CL,K,B),  ... > C
 if generate(ID,K,replicas(K,R,fac),S,A)  => C .
```

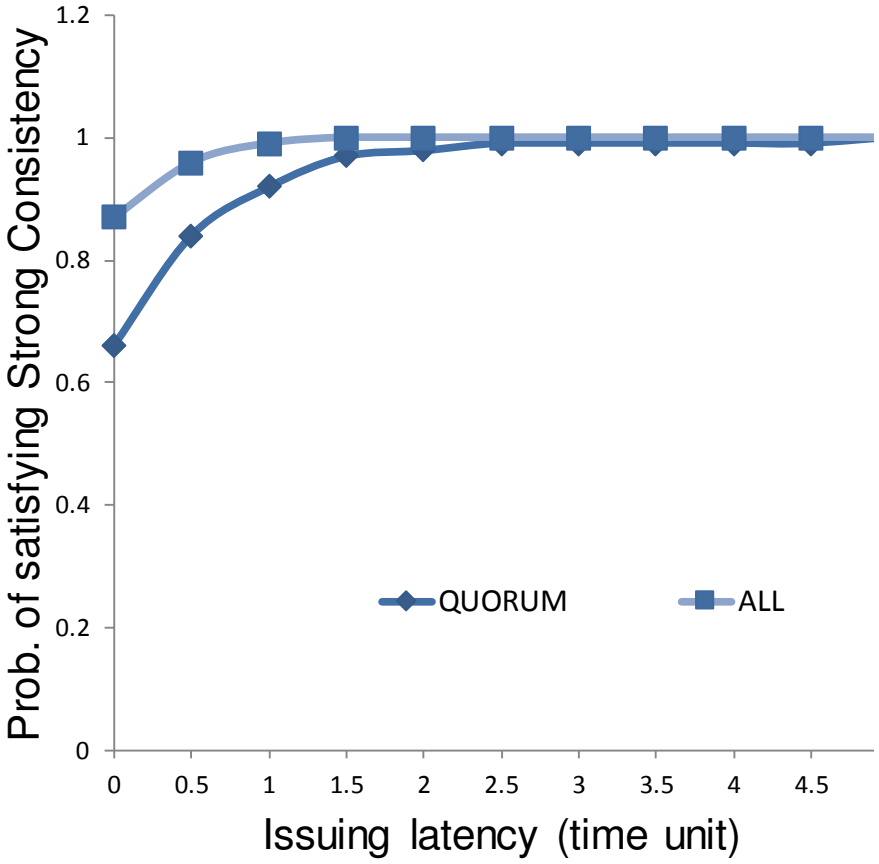
```
rl [GENERATE-READ-REQUEST] :
   generate(ID,K,(A',AD'),S,A)
 =>
   generate(ID,K,AD',S,A)
   [D, A' <- ReadRequestSS(ID,K,S,A)]
 with probability D := distr(...)
```

Validating Performance

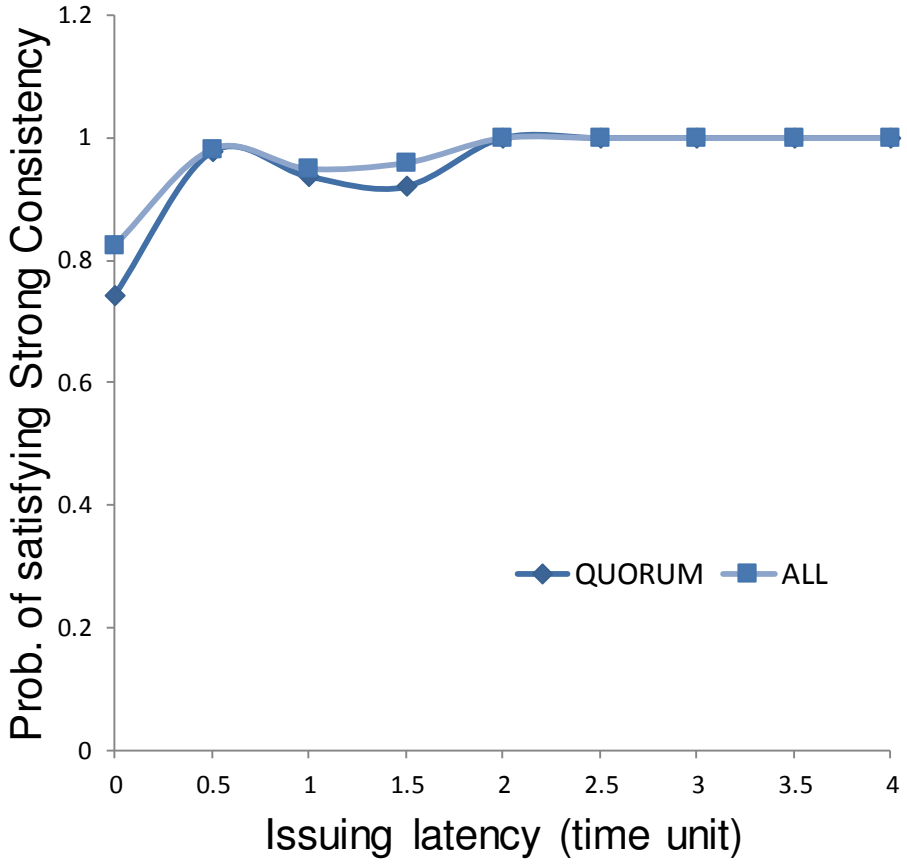
- We measure Cassandra's satisfaction of various consistency models
  - strong consistency (SC)
  - read your writes (RYW) consistency
- We answer two questions:

1. How well does our Cassandra model satisfy SC (and RYW)?
2. Do these results match reality?

Performance: Strong Consistency



Statistical Model Checker



Real-deployed cluster

- (X axis =) Issuing Latency = time difference between the given read request and the latest write request
- (Y axis =) Probability of a request satisfying that model
- **Conclusion: Statistical Model Checker is accurate in predicting low and high consistency behaviors**

Summary

- First formal and executable model of Cassandra
  - Captures all major design decisions
- Predicting Consistency behavior by using Statistical Model-Checking
- Statistical Model-checking matches reality (deployment numbers)
- Our work best to predict Cassandra Performance
  - Faster than simulations
  - Less work than full deployment
  - Repeatable

Ongoing and Future Work

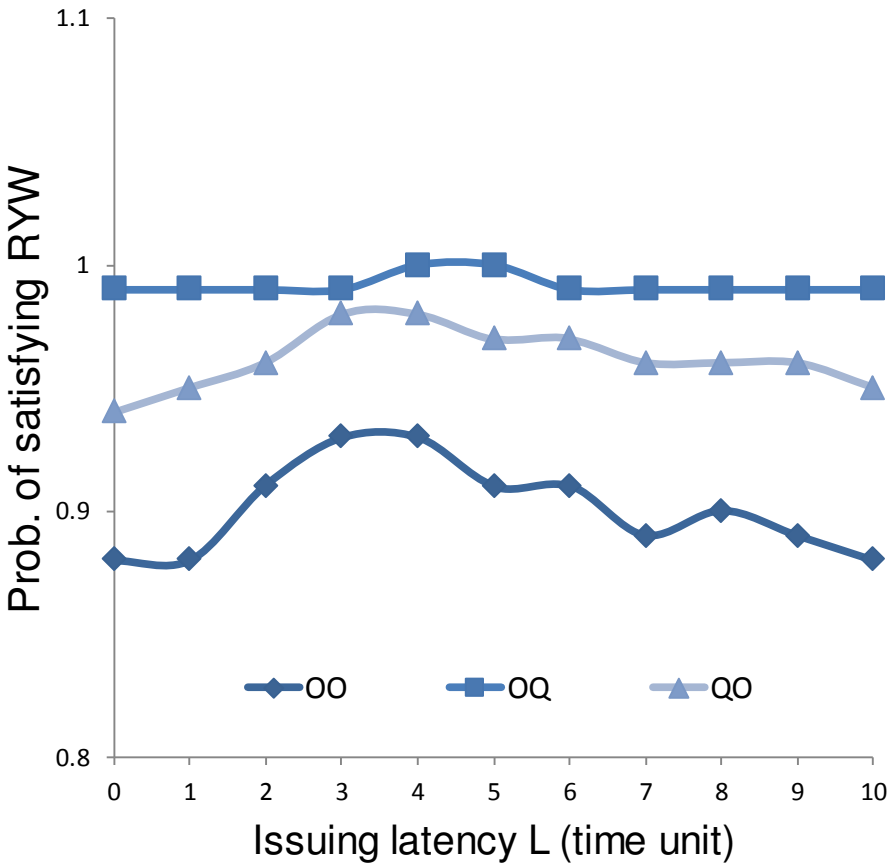
- Other consistency models
- Other performance metrics
  - Throughput, Latency
- Model other NoSQL systems
- Build a library of building blocks
  - Mix and match to generate any NoSQL system with desired consistency and availability trade-offs

Backup Slides

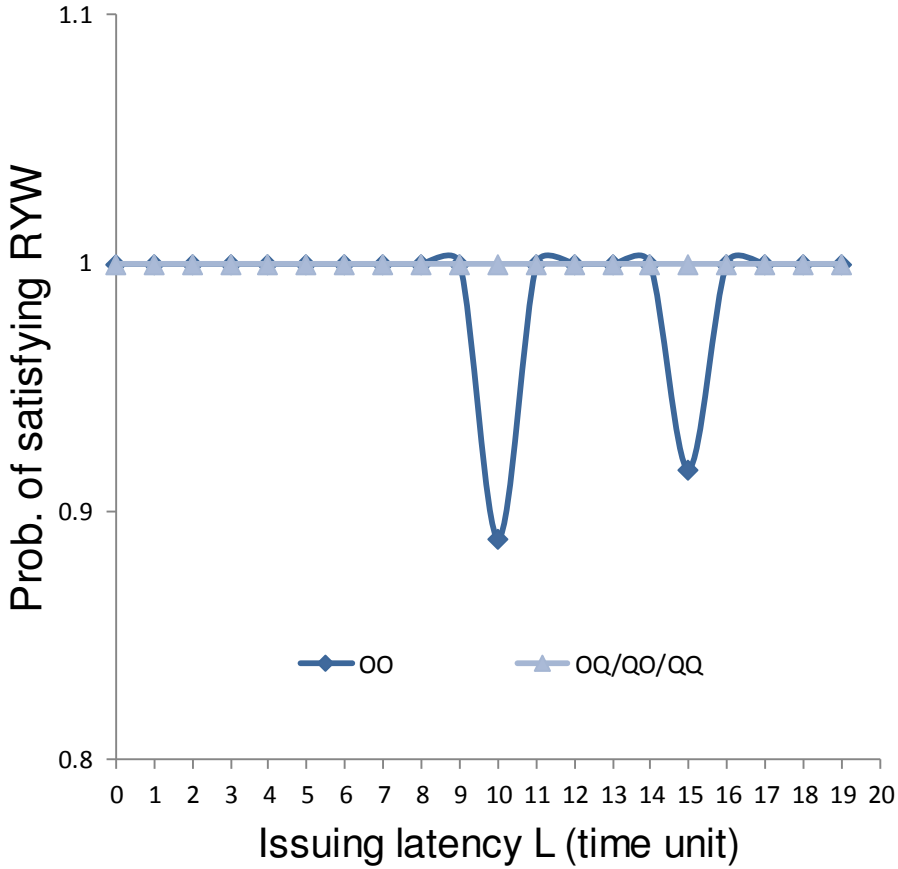
Rewriting Logic, Maude and PVeStA

- Rewriting logic is a semantic framework for modeling and analyzing distributed systems
- Maude is a high-performance language and system supporting rewriting logic specification and programming
- PVeStA is a statistical model checking tool that can verify a property up to a user-specified level of confidence by running Monte-Carlo simulations of the system model

Read Your Writes



Statistical Model Checker



Real-deployed cluster

- **Conclusion: Statistical Model Checker is reasonably accurate (to within 10-15%) in predicting consistency behaviors**