SAMC: Sematic-Aware Model Checking for Fast Discovery of Deep Bugs in Cloud Systems

Tanakorn Leesatapornwongsa, Mingzhe Hao,
Pallavi Joshi*, Jeffrey F. Lukman†,
and Haryadi S. Gunawi

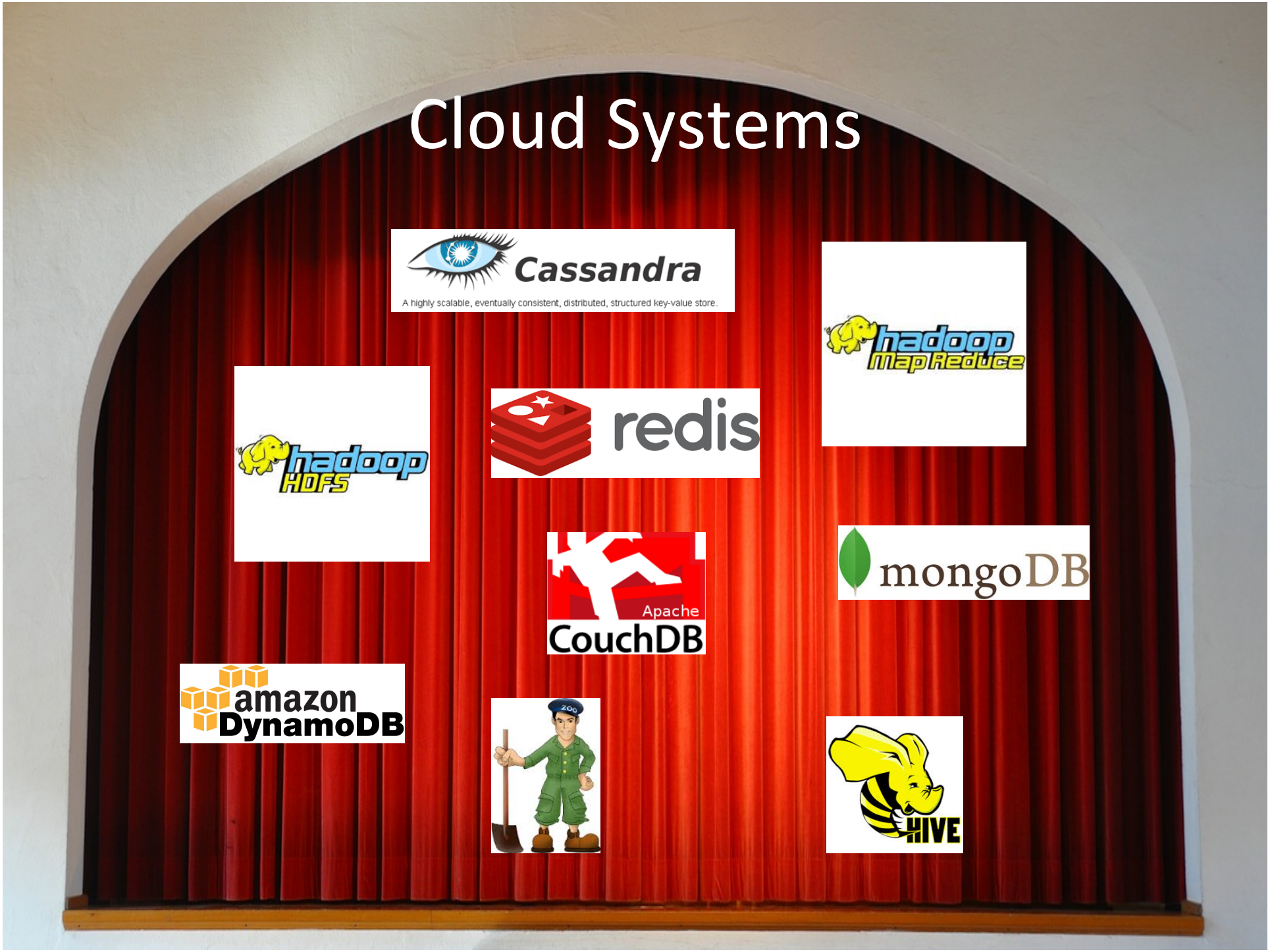
THE UNIVERSITY OF CHICAGO

† SURYA UNIVERSITY

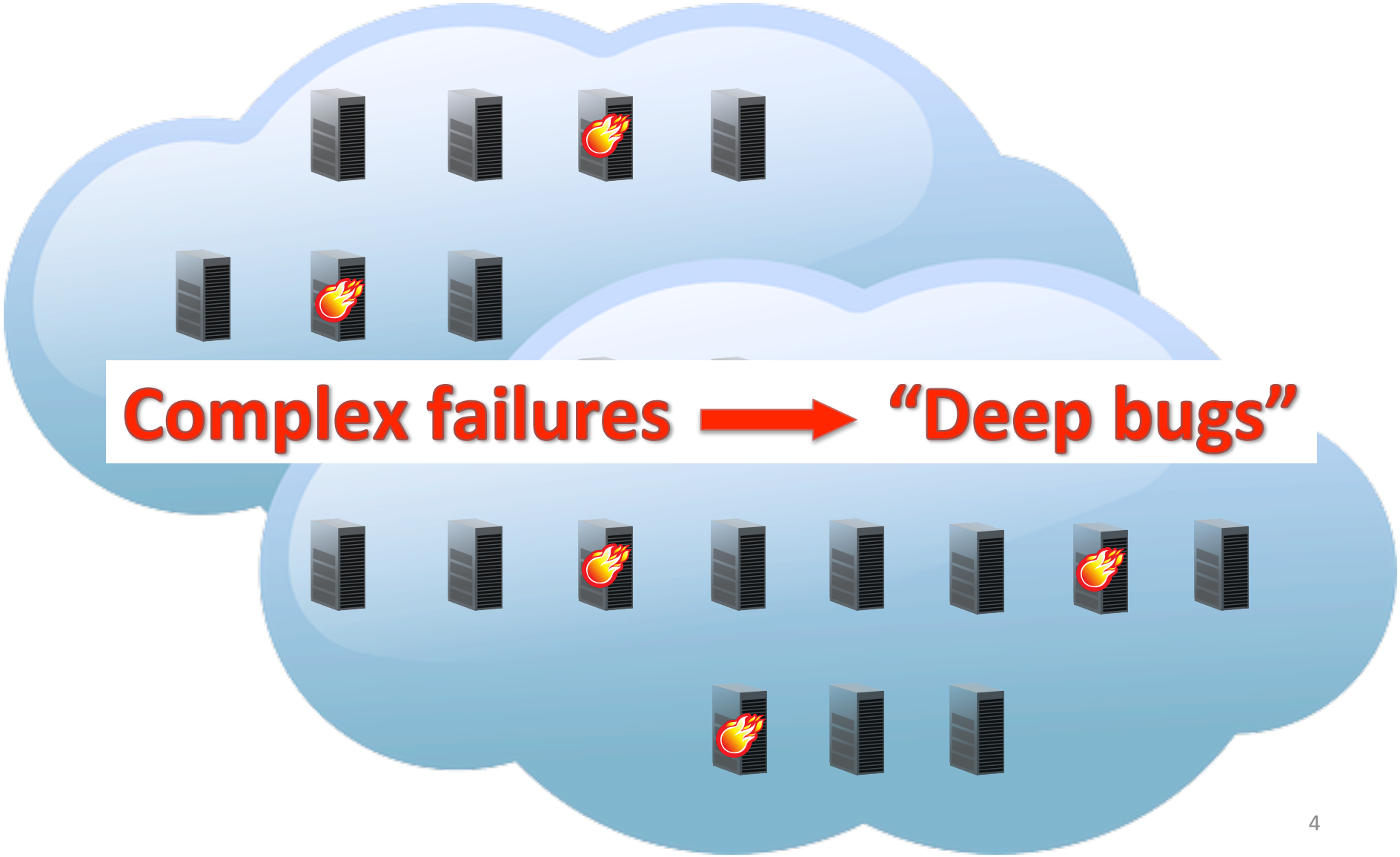
* NEC Laboratories America

Internet Services

Cloud Systems



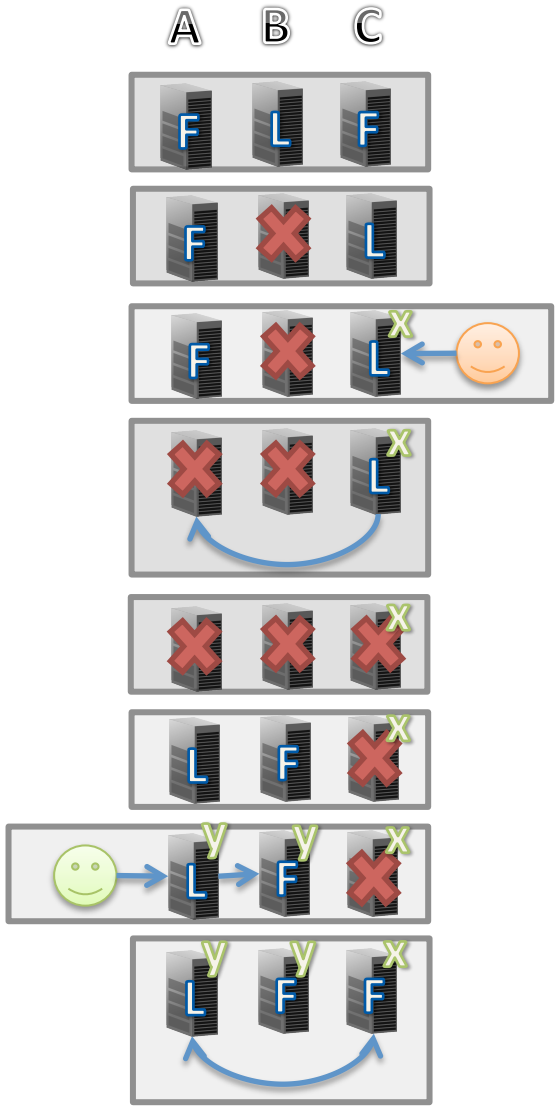
Reliability



Deep Bug Example

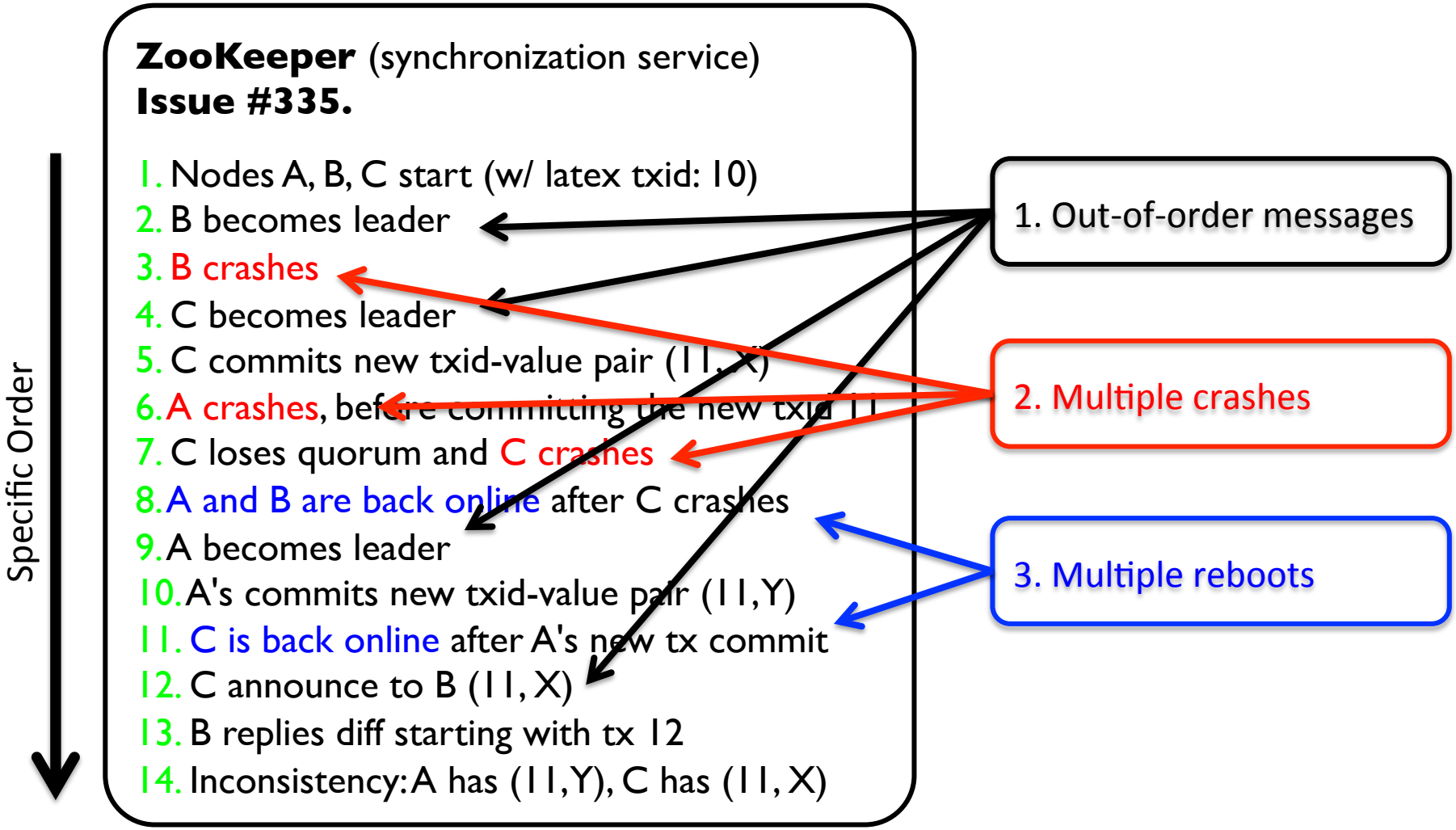
ZooKeeper (synchronization service)
Issue #335.

1. Nodes A, B, C start (w/ latex txid: 10)
2. B becomes leader
3. B crashes
4. C becomes leader
5. C commits new txid-value pair (11, X)
6. A crashes, before committing the new txid 11
7. C loses quorum and C crashes
8. A and B are back online after C crashes
9. A becomes leader
10. A's commits new txid-value pair (11,Y)
11. C is back online after A's new tx commit
12. C announce to B (11, X)
13. B replies diff starting with tx 12
14. Inconsistency: A has (11,Y), C has (11, X)



PERMANENT INCONSISTENT REPLICA

Deep Bug Characteristics



1 2 3 HAPPEN IN ANY ORDER

Study of Deep Bugs

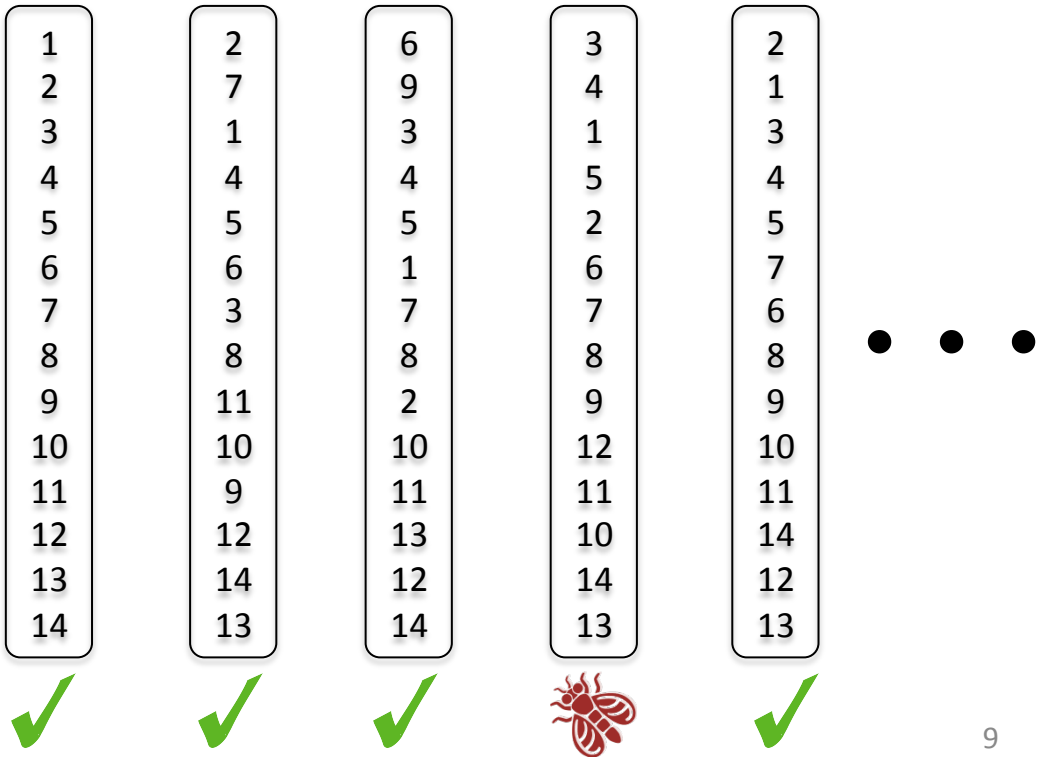
How do we catch deep bugs in distributed systems?

How to Catch Deep Bugs

- Distributed system model checker
  - Re-ordering all non-deterministic events
  - Find which specific orderings lead to bugs

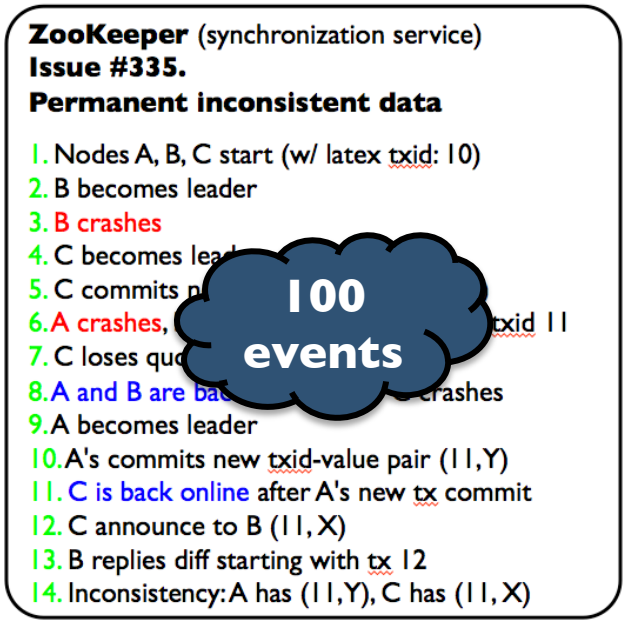
ZooKeeper (synchronization service)
Issue #335.
Permanent inconsistent data

1. Nodes A, B, C start (w/ latex txid: 10)
2. B becomes leader
3. B crashes
4. C becomes leader
5. C commits new txid-value pair (11, X)
6. A crashes, before committing the new txid 11
7. C loses quorum and C crashes
8. A and B are back online after C crashes
9. A becomes leader
10. A's commits new txid-value pair (11,Y)
11. C is back online after A's new tx commit
12. C announce to B (11, X)
13. B replies diff starting with tx 12
14. Inconsistency: A has (11,Y), C has (11, X)



What's Wrong with Existing Model Checkers?

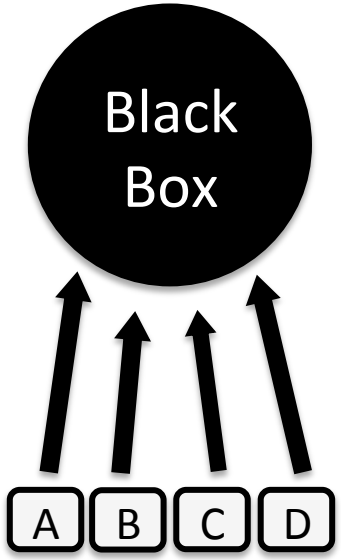
- Last 7 years
  - MaceMC [NSDI'07], Modist [NSDI'09], dBug [SSV'10], Demeter [SOSP'13], etc.
- **BUT**
  - Too many events
  - Multiple crashes and reboots
    - Create more messages
  - No model checker incorporate multiple crashes and reboots
  - **Cannot find deep bugs!**



How do we catch deep bugs
REALLY FAST?

Black-Box Approach

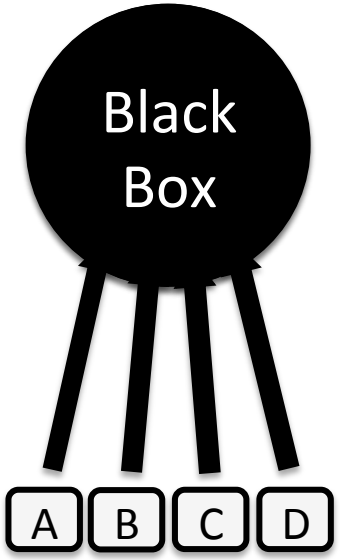
- Existing model checkers are so slow
- They treat target systems as black boxes
  - A large number of event orderings are generated



Black Box Model Checker
ABCD
ABDC
ACBD
ACDB
ADBC
. . .
(24 total)

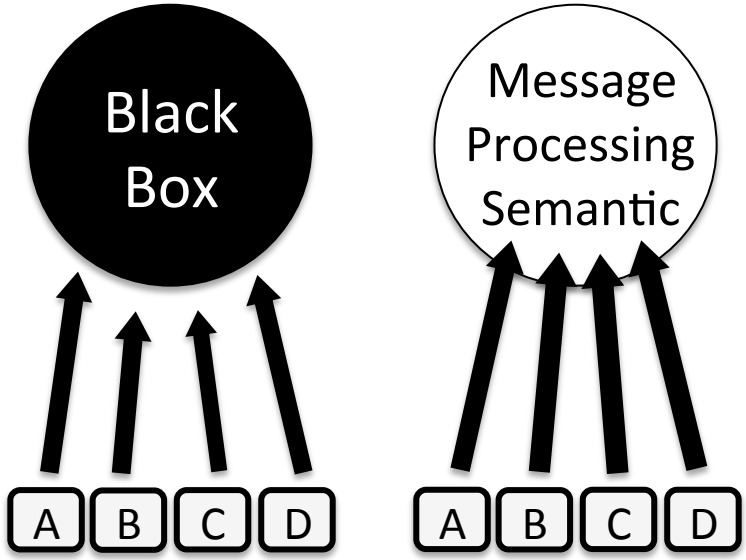
Semantic Knowledge

- How can we make model checker fast?
  - Exploit semantic knowledge
- Semantic-aware model checker (SAMC)



Black Box vs. SAMC

Black Box Model Checker
ABCD
ABDC
ACBD
ACDB
ADBC
ADCB
BACD
BADC
BCAD
BCDA
BDAC
. . .

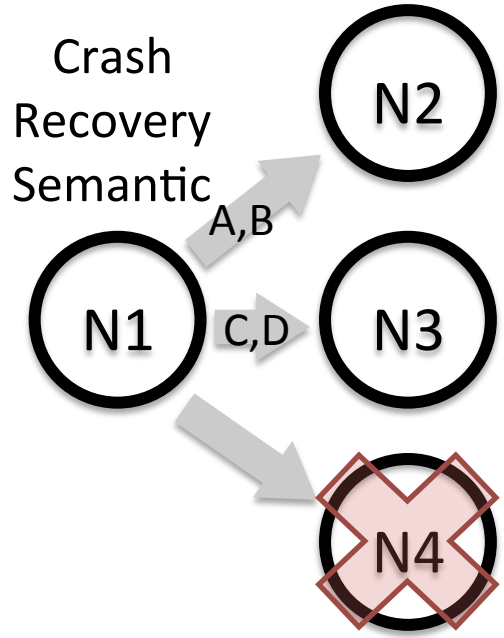


SAMC with message processing semantic
ABCD
~~ABDC~~
~~ACBD~~
~~ACDB~~
~~ADBC~~
~~ADCB~~
BACD
~~BADC~~
~~BCAD~~
~~BCDA~~
. . .

Unnecessary Re-orderings

Lead to the same state

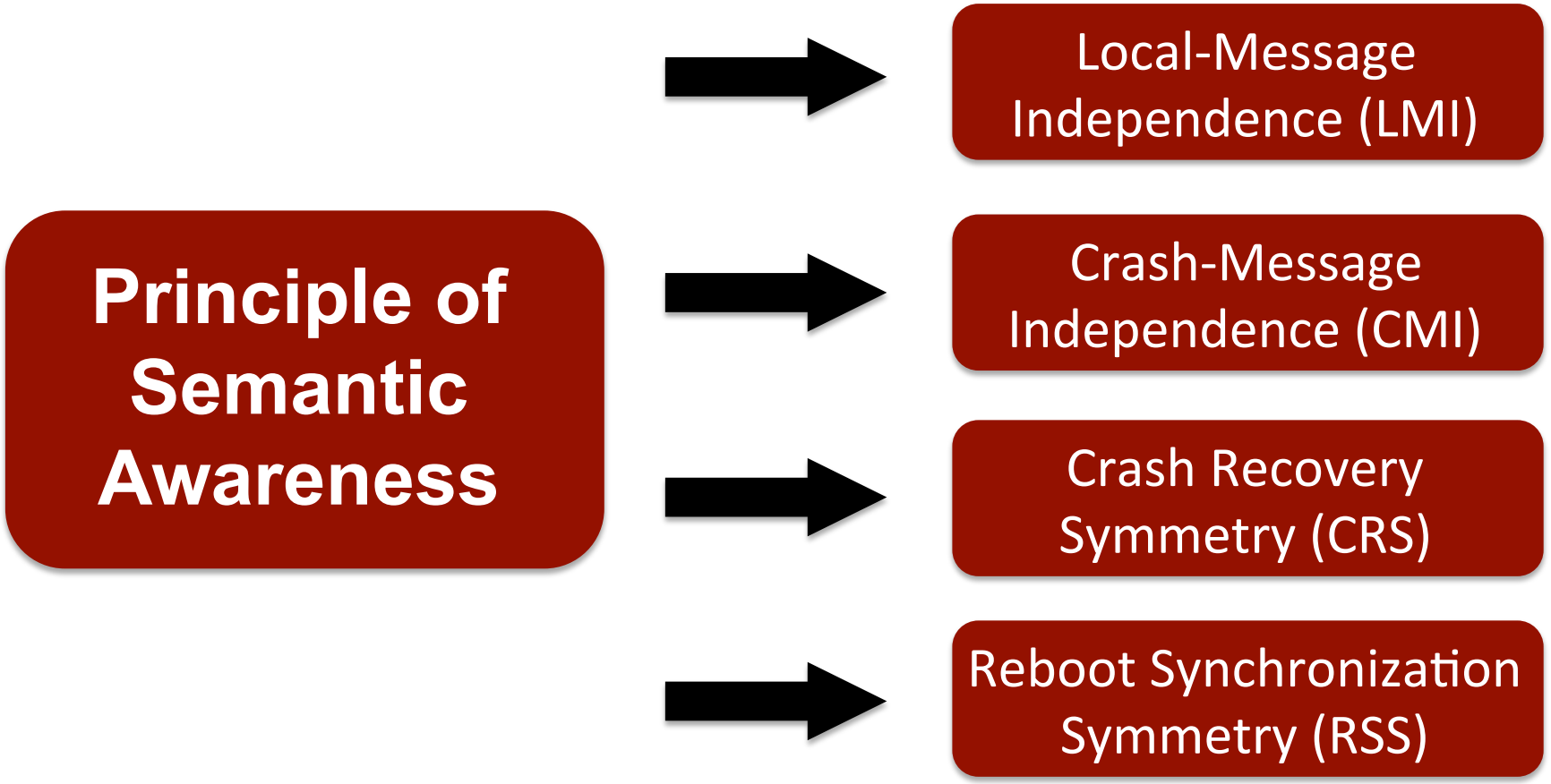
SAMC with Crashes



Black Box Model checker	SAMC with crash recovery semantic
ABCDX	ABCDX
ABCXD	~~ABCXD~~
ABXCD	~~ABXCD~~
AXBCD	~~AXBCD~~
XABCD	~~XABCD~~
ABDCX	~~ABDCX~~
ABDXC	~~ABDXC~~
. . .	. . .

Unnecessary Re-orderings

Generic Reduction Algorithms



SAMC Implementation and Integration

- SAMC implementation
  - 10,000 LOC from scratch
- Apply SAMC to 3 cloud systems
  - 7 protocols
  - 10 versions

Cloud systems	Protocol
Cassandra	Gossiper
	Hinted handoff
	Read/write
Hadoop 2.0	Cluster management
	Speculative execution
ZooKeeper	Atomic broadcast
	Leader election

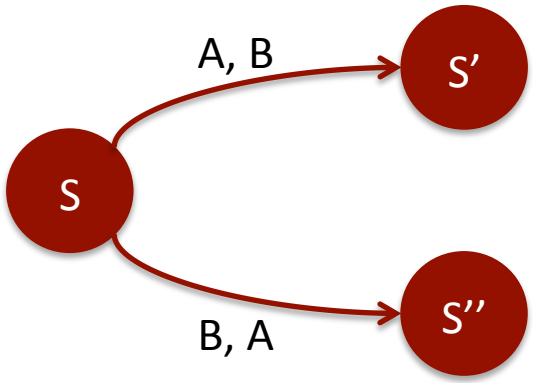
Result

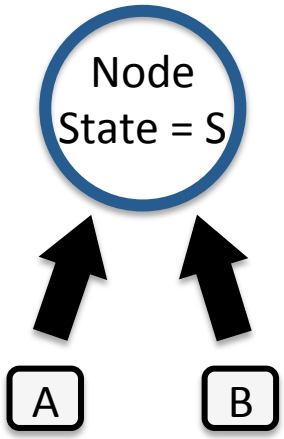
- Reproduced 12 old bugs
  - Compare to state-of-the-art techniques
    - Dynamic Partial Order Reduction (DPOR)
    - Random-DPOR
  - Find bugs **2x** to **340x** faster
    - **49x** on average
- Found 2 new bugs
  - Submit them to developers

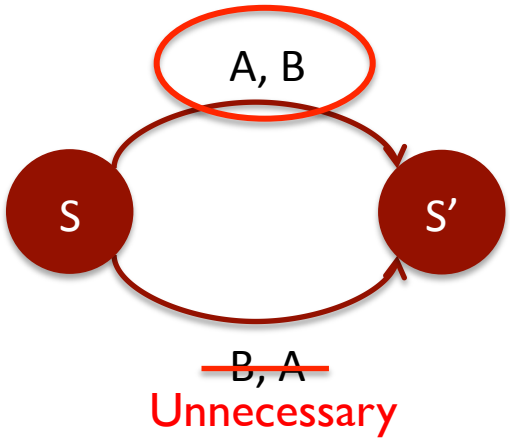
Outline

- **Intuition**
- SAMC
  - Local-Message Independence
  - Crash-Message Independence
  - Crash Recovery Symmetry
  - Reboot Synchronization Symmetry
- Evaluation

Dependency vs. Independency







A, B = Dependent

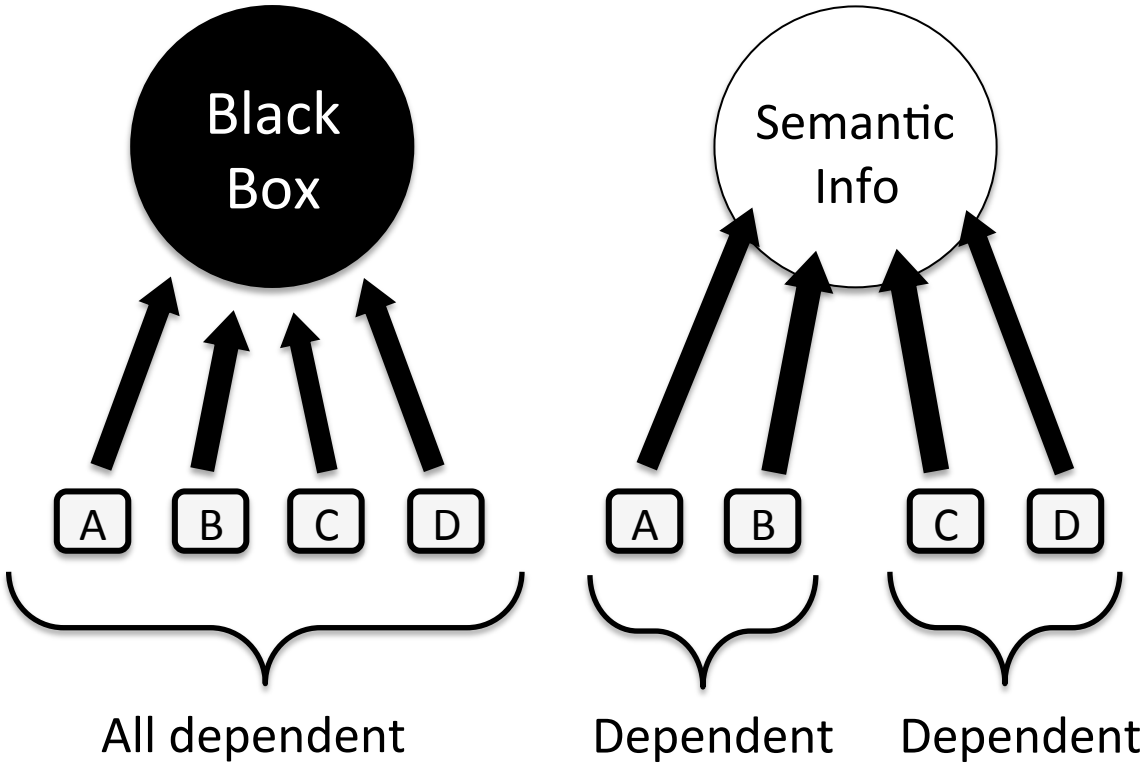
A, B = Independent

INDEPENDENT = NO REORDERING

2X SPEED-UP

Black Box vs. SAMC

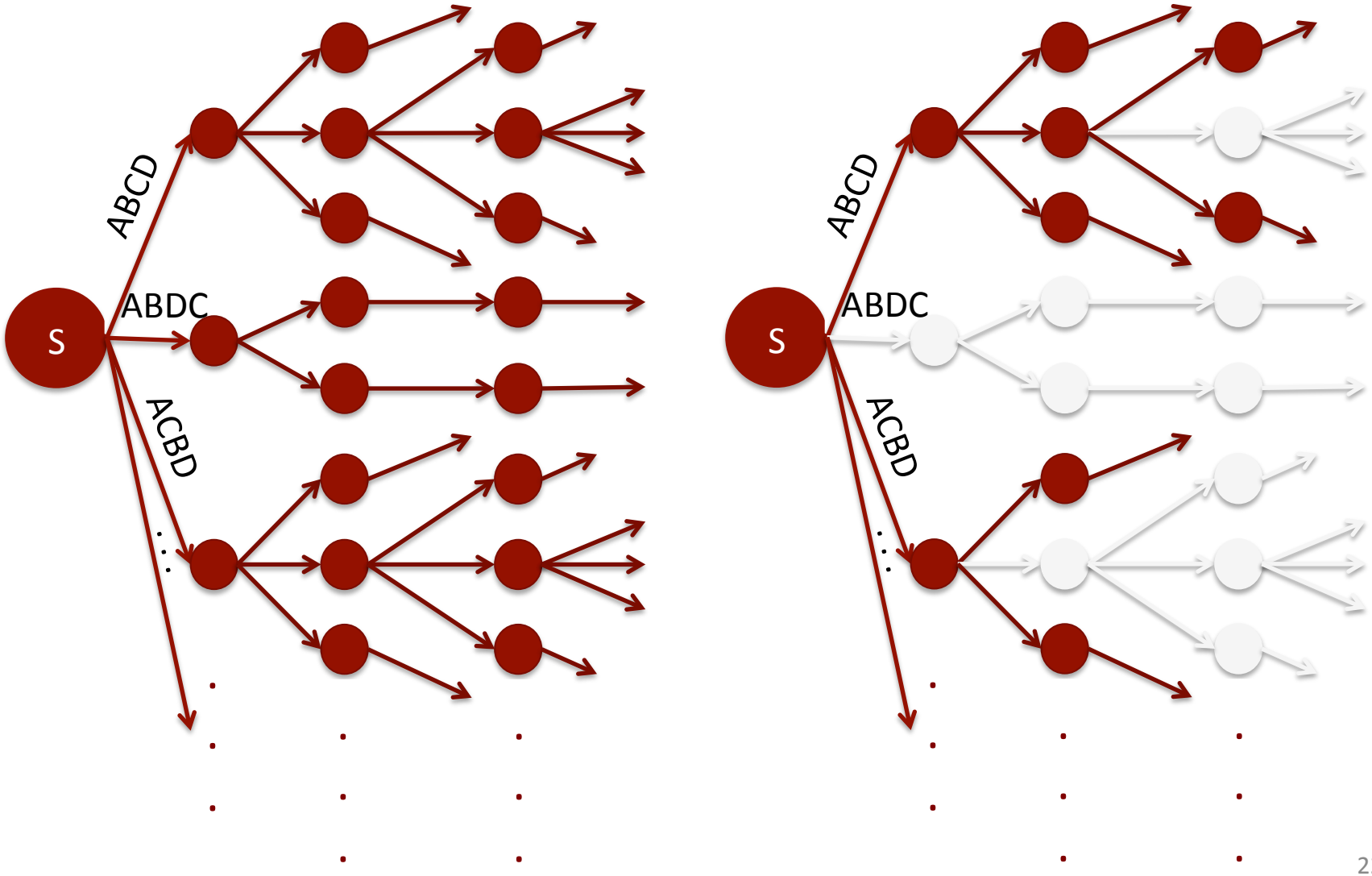
Black Box Model Checker
ABCD
ABDC
ACBD
ACDB
ADBC
ADCB
BACD
BADC
BCAD
BCDA
. . .
4!=24 orderings



SAMC
ABCD
ABDC
BACD
BADC

6X SPEED-UP

Reduction Speed-up

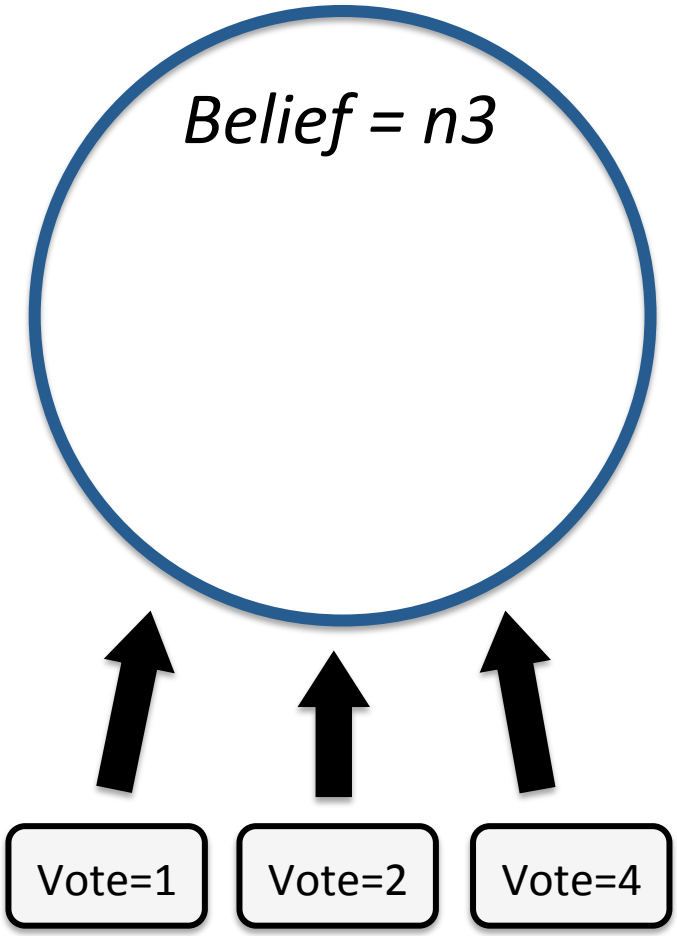


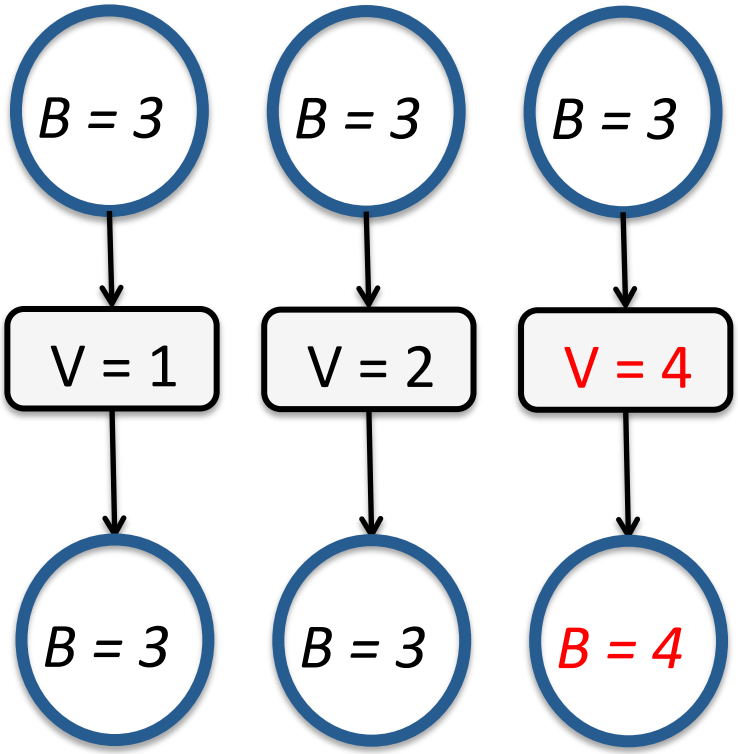
How to Declare Message Independency?

Q: Which concurrent messages are independent ?

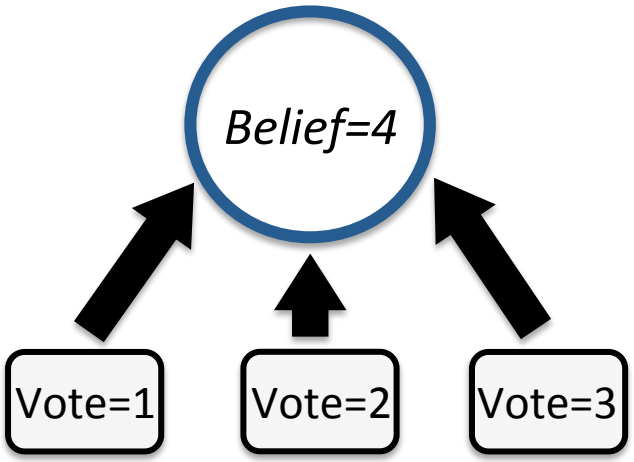
A: Use message processing semantic

Message Processing Semantic in Simplified Leader Election

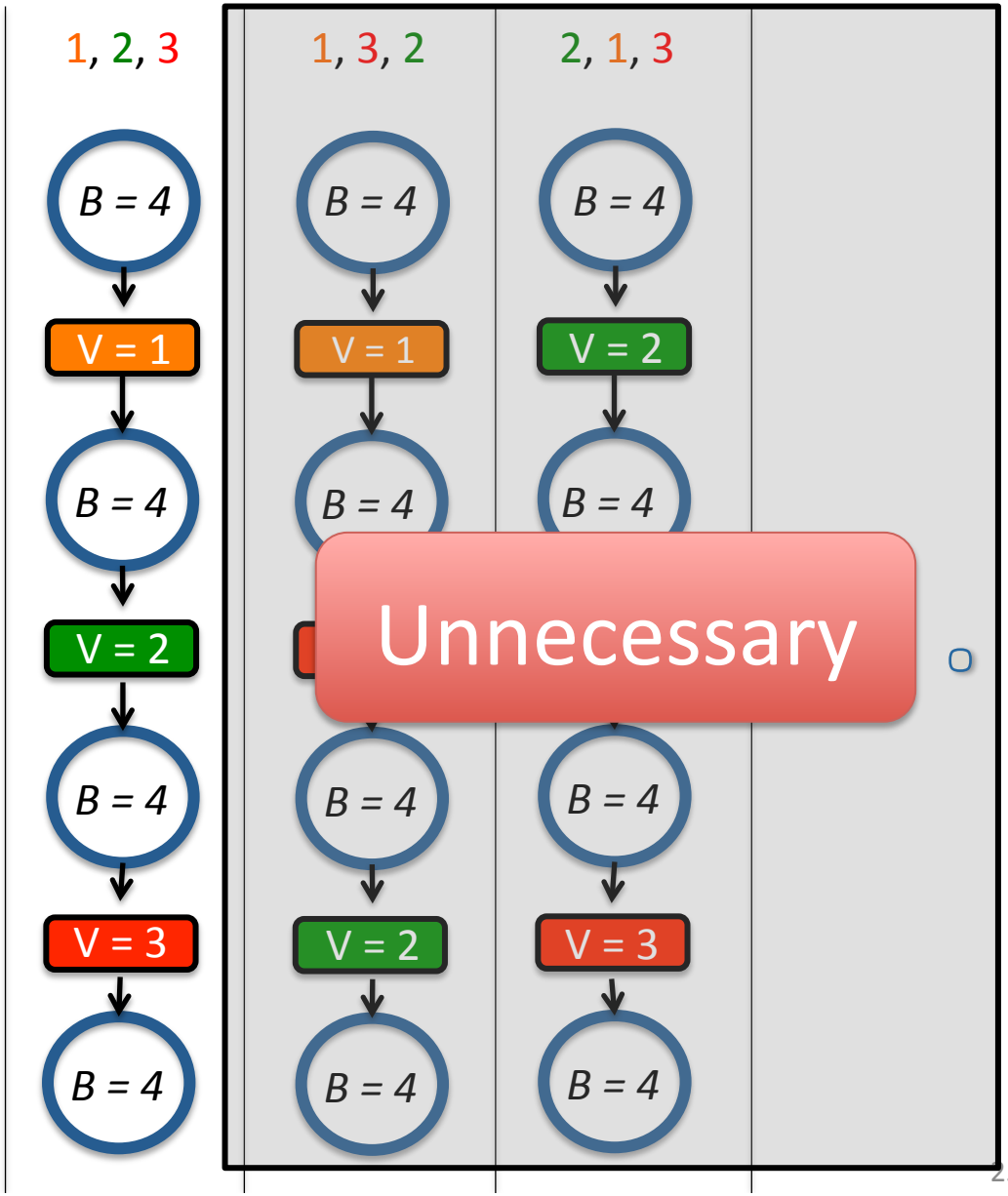




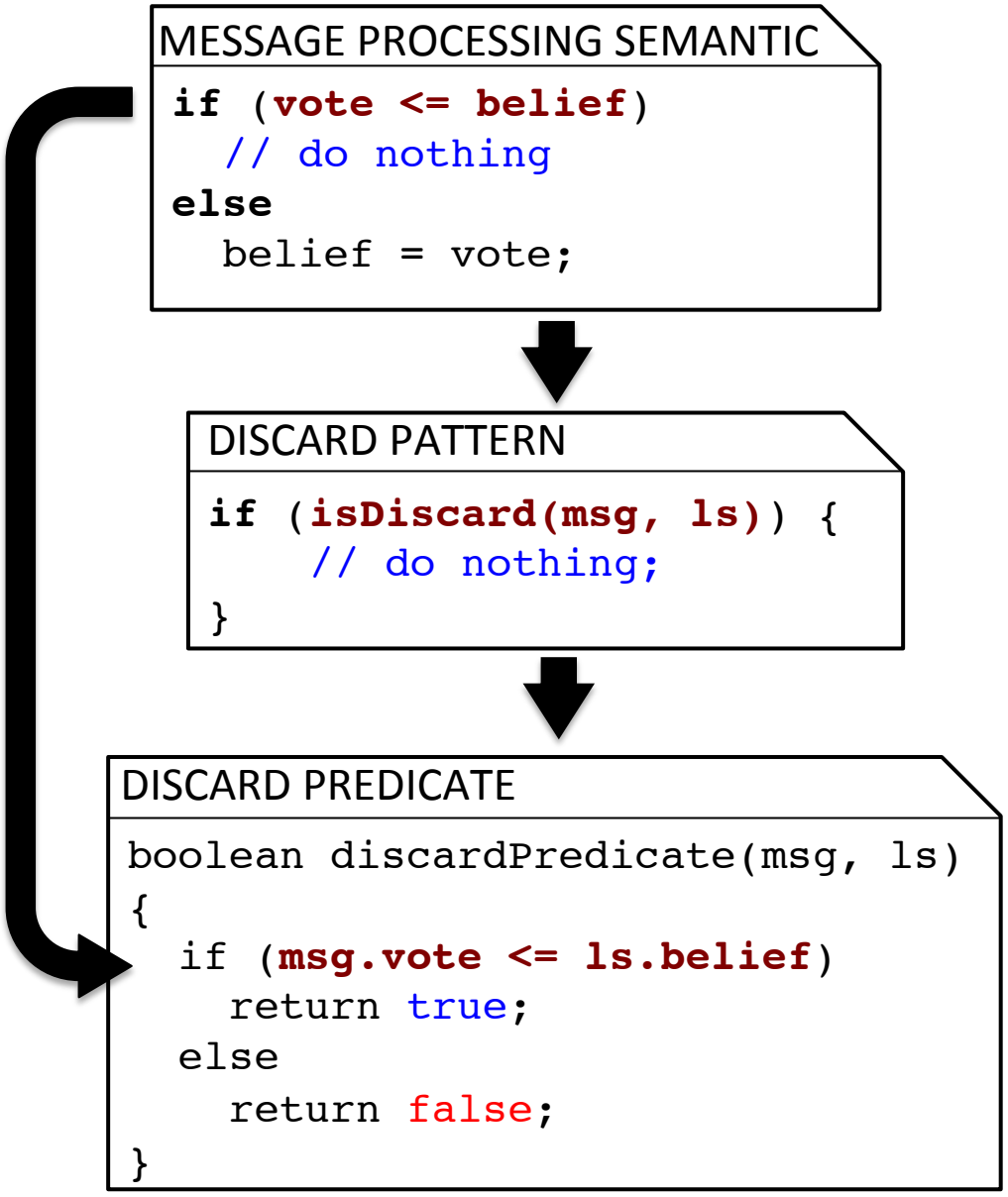
Removing Re-ordering via Message Processing Semantic



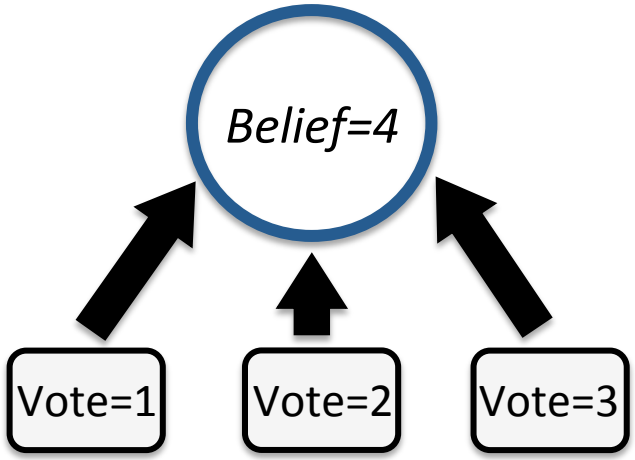

```
if (vote <= belief)
  // do nothing
else
  belief = vote;
```



Formalizing the Intuition



Formalizing the Intuition



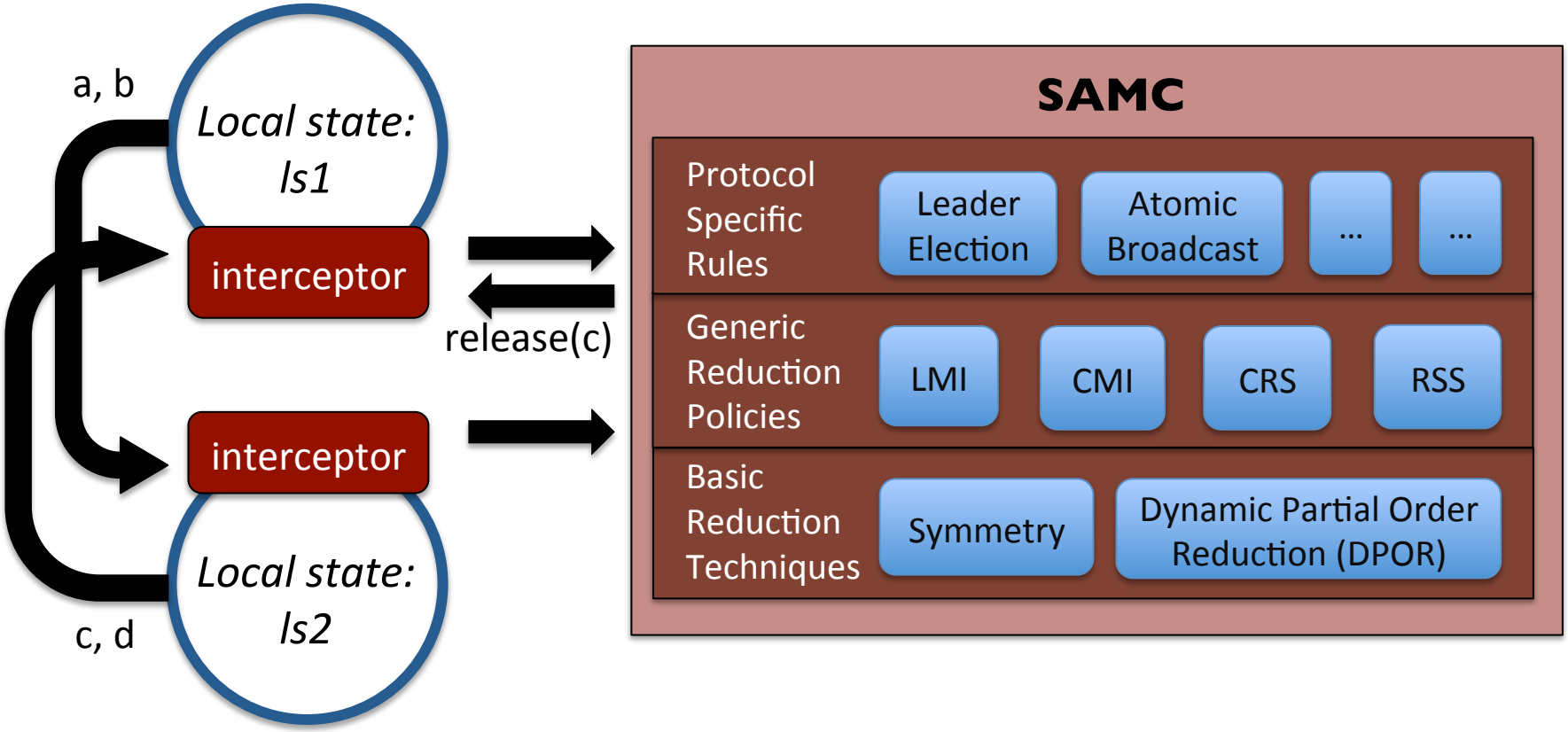
vote	belief	discardPredicate
1	4	*true*
2	4	*true*
3	4	*true*

m_x	m_y	discard(m_x)	discard(m_y)	Independent
1	2	true	true	✔
1	3	true	true	✔
2	3	true	true	✔

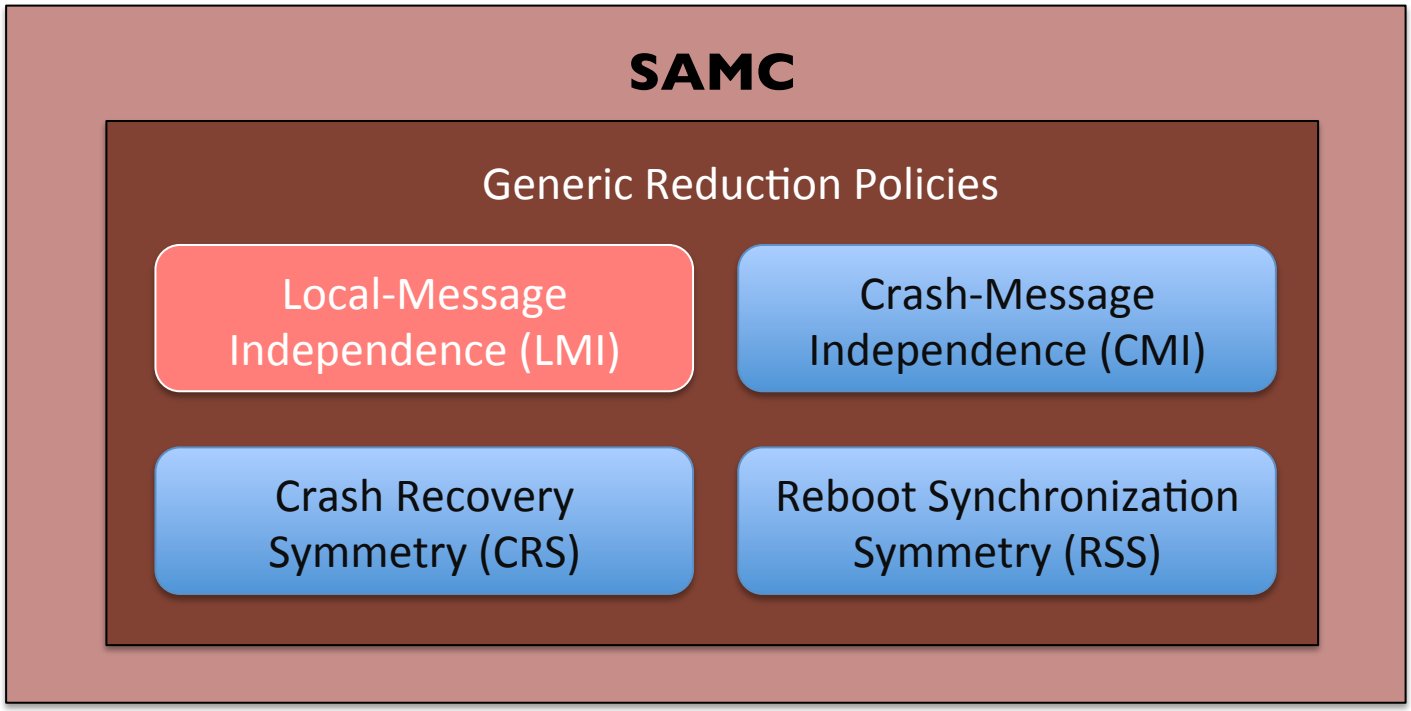
Outline

- Intuition
- **SAMC**
  - Local-Message Independence
  - Crash-Message Independence
  - Crash Recovery Symmetry
  - Reboot Synchronization Symmetry
- Evaluation

SAMC Architecture



Local-Message Independence

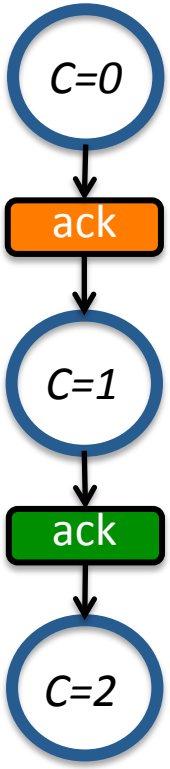


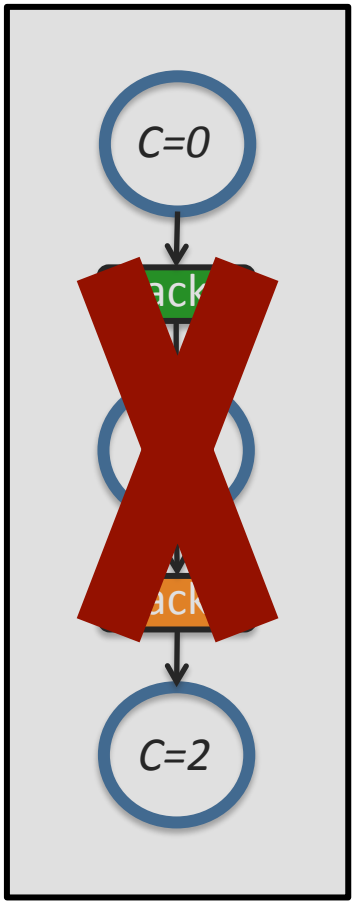
Local-Message Independence

- Discard pattern
- Increment pattern

```
if (msg.type == ack) {
  node.ackCount++;
}
```

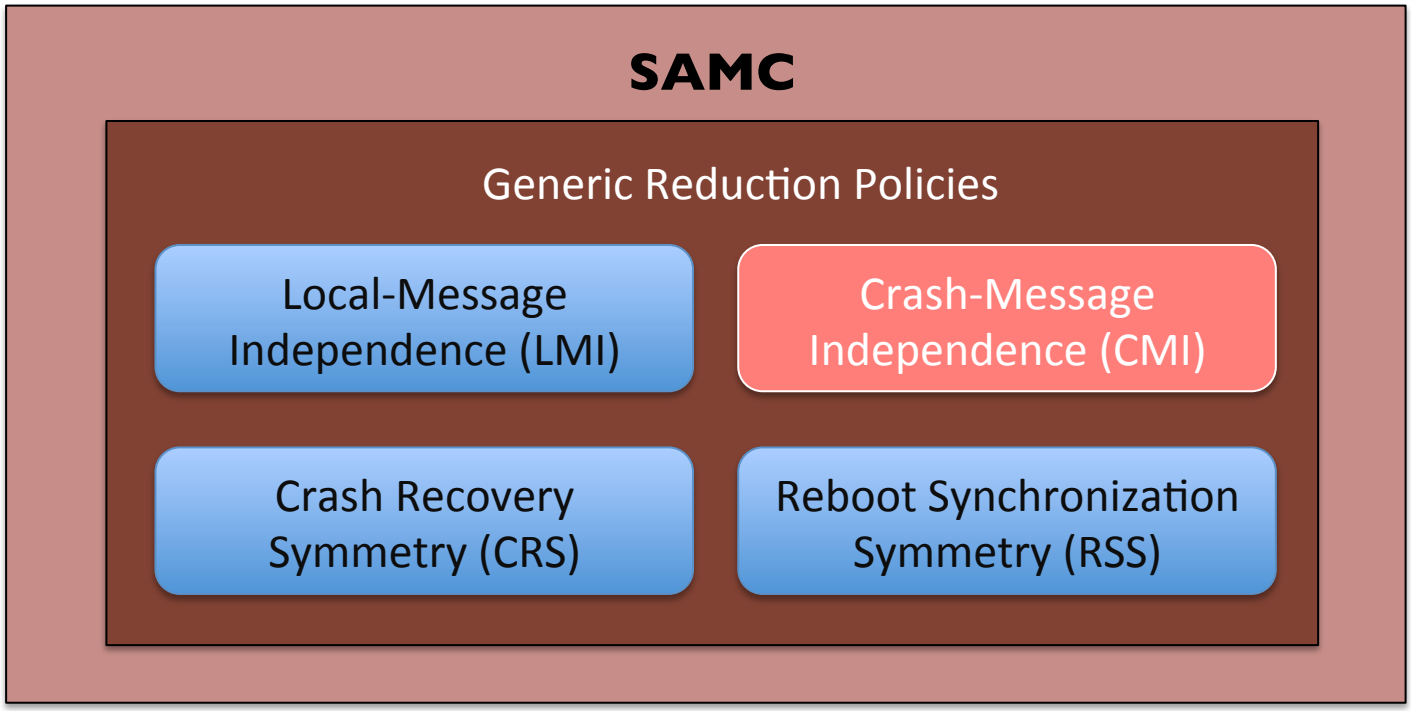
```
boolean incrementPredicate(msg, ls)
{
  if (msg.type == ack)
    return true;
  else
    return false;
}
```





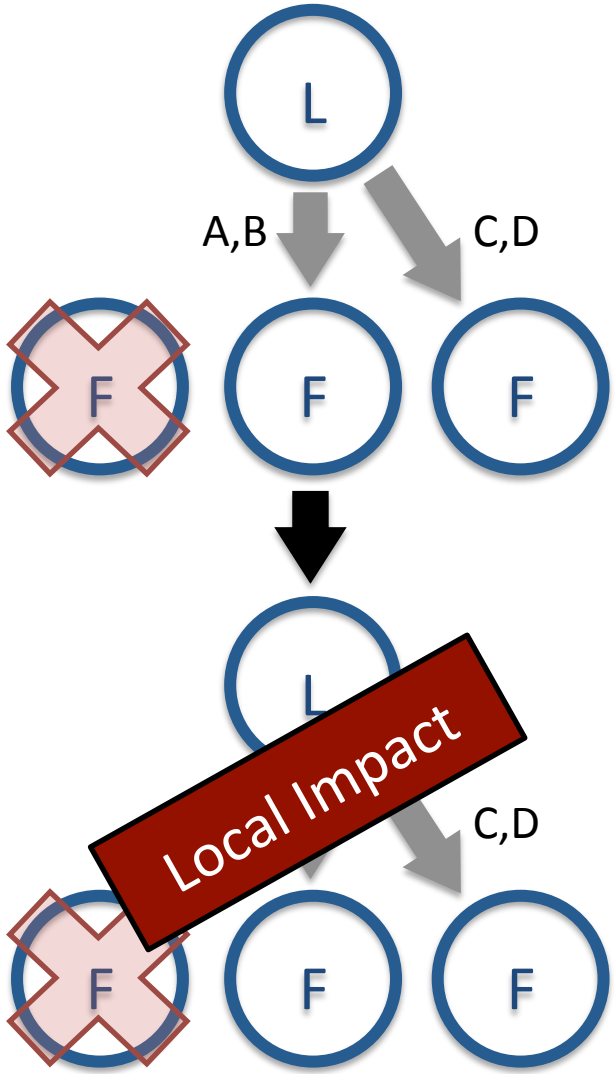
- Constant pattern

Crash-Message Independence



Crash-Message Independence

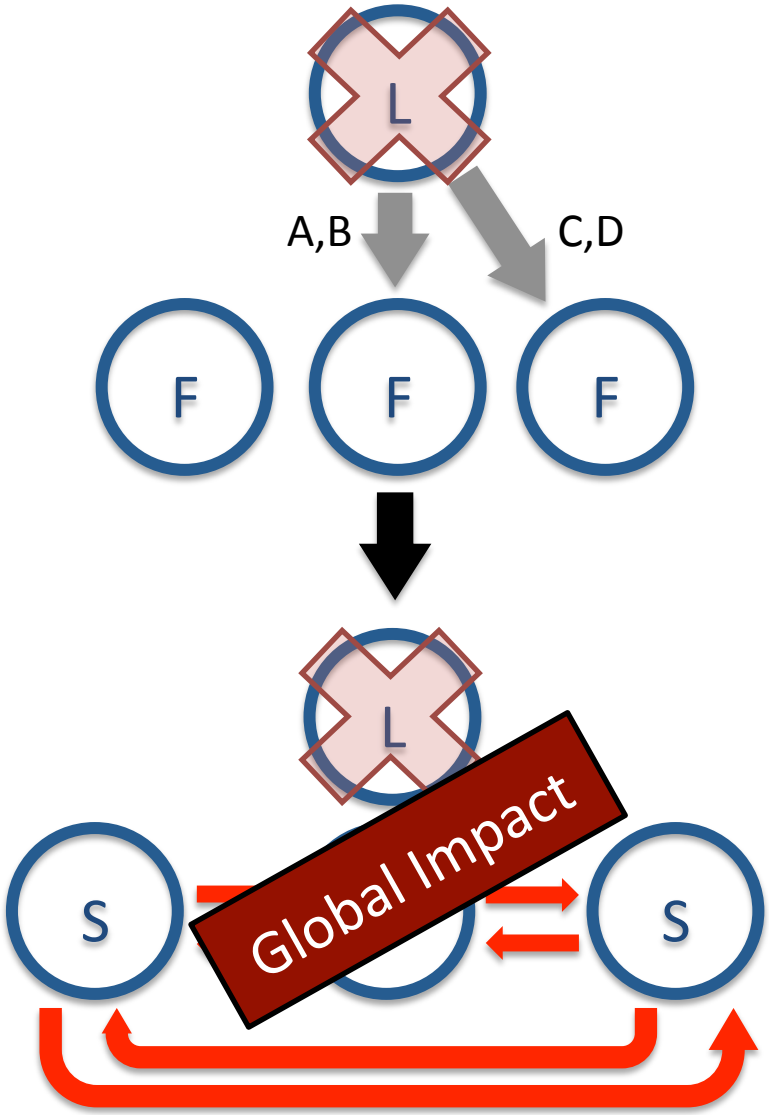
Black Box
ABCDX
~~ABCXD~~
~~ABXCD~~
~~AXBCD~~
~~XABCD~~
~~ABDCX~~
…




```
void handleCrash() {
  if (X == follower && isQuorum())
    followerCount--;
    // No new messages
}
```

```
boolean localImpact(X, ls) {
  if (X == follower && isQuorum())
    return true;
  else
    return false;
}
```

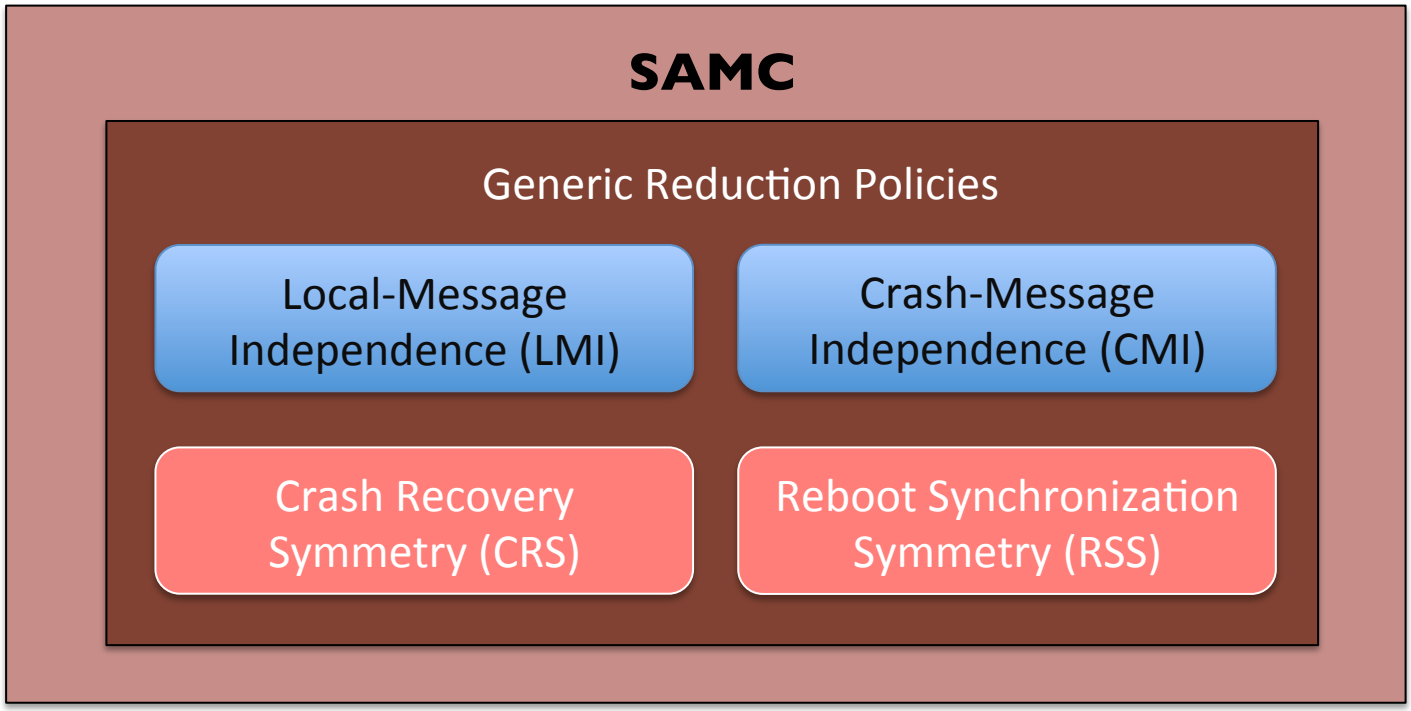
Crash-Message Independence




```
void handleCrash() {
  if (X == leader || !isQuorum())
    electLeader()
    // New messages created
}
```

```
boolean globalImpact(X, ls) {
  if (X == leader || !isQuorum())
    return true;
  else
    return false;
}
```

Crash Recovery Symmetry
Reboot Synchronization Symmetry



Outline

- Intuition
- SAMC
  - Local-Message Independence
  - Crash-Message Independence
  - Crash Recovery Symmetry
  - Reboot Synchronization Symmetry
- **Evaluation**

Evaluation

- SAMC implementation
  - 10,000 LOC from scratch
- Apply SAMC to 3 cloud systems
  - 7 protocols
  - 10 versions

Cloud systems	Protocol
Cassandra	Gossiper
	Hinted handoff
	Read/write
Hadoop 2.0	Cluster management
	Speculative execution
ZooKeeper	Atomic broadcast
	Leader election

Protocol-Specific Rules

(e.g. ZooKeeper Leader Election)

- Guide SAMC to remove re-orderings
- 35 LOC on average per protocol

Local-Message Independence (LMI)

```
bool pd : !newVote(m, s);

bool pm : newVote(m, s);

bool newVote(m, s) :
 if (m.ep > s.ep)
   ret 1;
 else if (m.ep == s.ep)
  if (m.tx > s.tx)
   ret 1;
  else if (m.tx == s.tx &&
           m.lid > s.lid)
   ret 1;

 ret 0;
```

Crash-Message Independence (CMI)

```
bool pg (s, X) :
 if (s.rl == F && X.rl == L)
  ret 1;
 if (s.rl == L && X.rl == F
     && !quorumAfterX(s)
  ret 1;
 if (s.rl == S && X.rl == S)
  ret 1;

bool pl (s, X) :
 if (s.rl == L && X.rl == F
     && quorumAfterX(s))
  ret 1;


bool quorumAfterX(s) :
  ret ((s.fol-1) >=
        s.all/2);
```

Crash Recovery Symmetry (CRS)

```
bool pr1(s,C):
 if (s.rl == L && C.rl == F
     && quorumAfterX(s))
  ret 1;
rals1:{rl,fol,all};

bool pr2(s,C):
 if (s.rl == L && C.rl == F
     && !quorumAfterX(s))
 ret 1;
rals2: {rl,fol,lid,ep,tx,clk}

bool pr3(s,C):
 if (s.rl == F && c.rl == L)
  ret 1;
rals3: {rl,fol,lid,ep,tx,clk}

bool pr4:
 if (s.rl == S)
  ret 1;
rals4: {rl,lid,ep,tx,clk}
```

Catching Old Bugs

A table shows number of executions to reach the bugs and speedup

Issue#	SAMC	Black-Box DPOR	Random	Random DPOR
ZooKeeper-335				
ZooKeeper-790				
ZooKeeper-975				
ZooKeeper-1075				
ZooKeeper-1419				
ZooKeeper-1492				
ZooKeeper-1653				
MapReduce-4748				
MapReduce-5489				
MapReduce-5505				
Cassandra-3395				
Cassandra-3626				

Catching Old Bugs

A table shows number of executions to reach the bugs and speedup

Issue#	SAMC	Black-Box DPOR		Random		Random DPOR	
	#exe	#exe	speedup	#exe	speedup	#exe	speedup
ZooKeeper-335	117	**5000+**	43+	1057	9	**5000+**	43+
ZooKeeper-790	7	14	2	225	32	82	12
ZooKeeper-975	53	967	18	71	1	163	3
ZooKeeper-1075	16	1081	68	86	5	250	16
ZooKeeper-1419	100	924	9	2514	25	987	10
ZooKeeper-1492	576	**5000+**	9+	**5000+**	9+	**5000+**	9+
ZooKeeper-1653	11	945	86	3756	341	3462	315
MapReduce-4748	4	22	6	6	2	6	2
MapReduce-5489	53	**5000+**	94+	**5000+**	94+	**5000+**	94+
MapReduce-5505	40	1212	30	**5000+**	125+	1210	30
Cassandra-3395	104	2552	25	191	2	550	5
Cassandra-3626	96	**5000+**	52+	**5000+**	52+	**5000+**	52

Reduction Ratio

- ZooKeeper leader election protocol
- Run black-box DPOR
- After each execution, find that this execution is executed by SAMC or not
- Count DPOR's executions that are executed by SAMC too

#Crash	#Reboot	Reduction Ratio				
		ALL	LMI	CMI	CRS	RSS
1	1	37X	18X	5X	4X	3X
2	2	63X	35X	6X	5X	5X
3	3	103X	37X	9X	9X	14X

Conclusion

- Deep bugs live in the cloud
- Model checker needs to incorporate complex failure to reach deep bugs
  - State space explosion
- Semantic-aware model checking
  - LMI, CMI, CRS, RSS
- Bring future research questions
  - What other semantic knowledge is useful?
  - How to extract them from the code automatically?

Thank You

Questions?

THE UNIVERSITY OF CHICAGO

http://ucare.cs.uchicago.edu