code

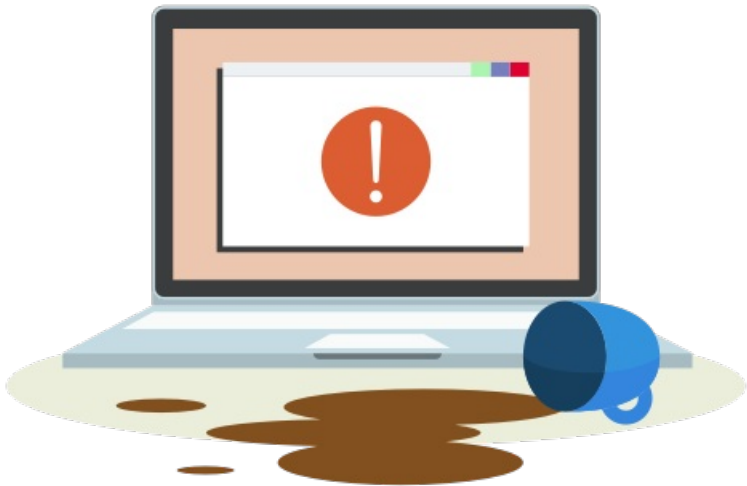
Weeks of debugging can save you hours of TLA+

Markus A. Kuppe
Principal Engineer – RiSE@MSR

develop & learn

Microsoft

Wanted to work on my slides...



Blocking Queue

Wait: {}

Run: {p1, p2, p3, p4, c1, c2, c3}

Sched: p1

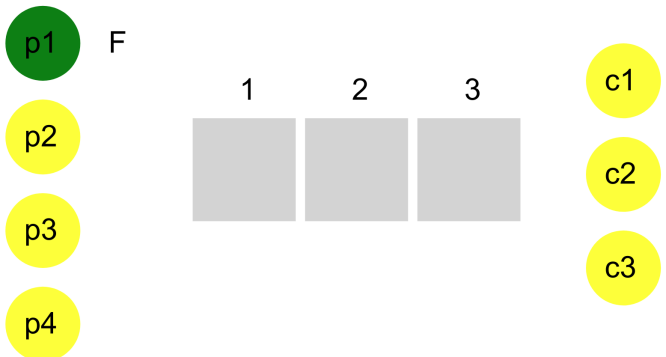


Figure: 1/47

Blocking Queue

Wait: {}

Run: {p1, p2, p3, p4, c1, c2, c3}

Sched: p1

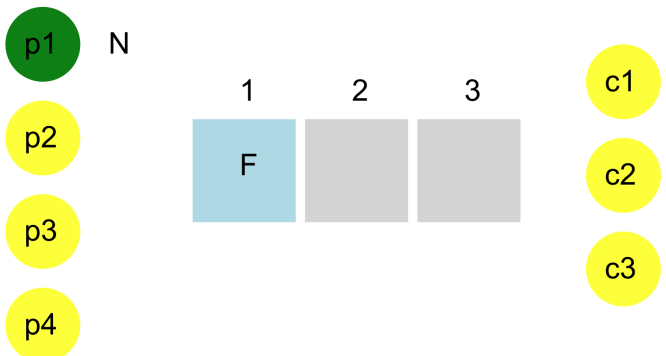


Figure: 2/47

Blocking Queue

Wait: {}

Run: {p1, p2, p3, p4, c1, c2, c3}

Sched: p1

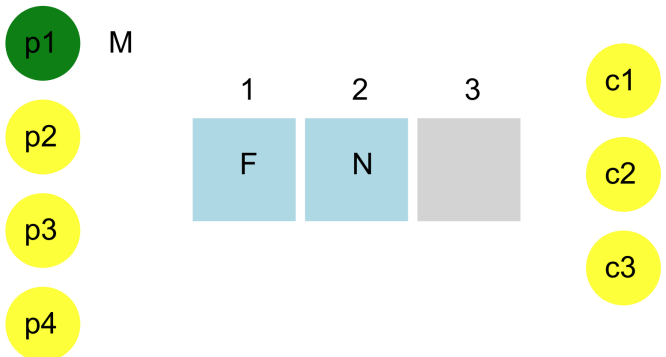


Figure: 3/47

Blocking Queue

Wait: {p1, p2, p3, p4}

Run: {c1, c2, c3}

Sched: c1

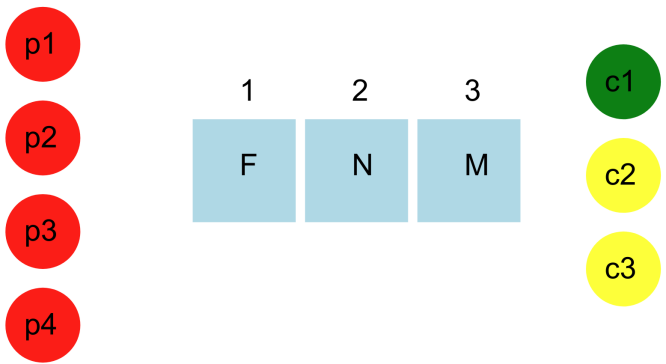


Figure: 17/47

Blocking Queue

Wait: {p2, p3, p4}

Run: {p1, c1, c2, c3}

Sched: c1

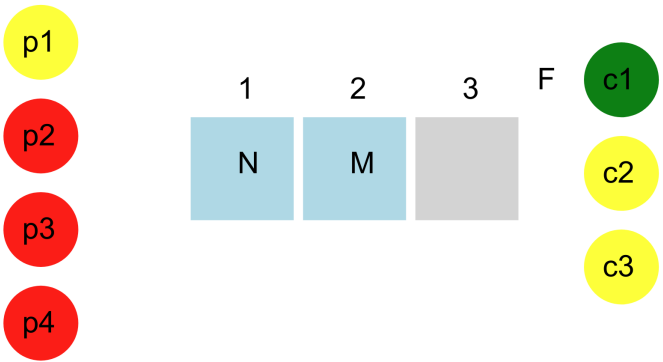


Figure: 18/47

Blocking Queue

Wait: {p1, p2, p3, p4, c1, c2, c3}

Run: {}

Sched: p1

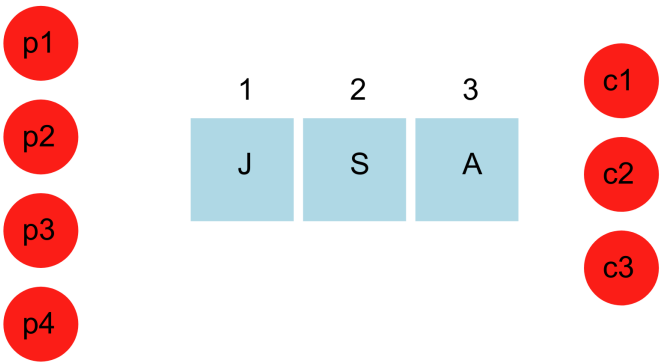


Figure: Deadlock!!!

TLA$^+$ 30.000ft above

TLA$^+$ is a specification language to design, document, and verify reactive systems.

Figure: TLA+ creator

TLA$^+$ 30.000ft above

TLA$^+$ is a specification language to design, document, and verify reactive systems.

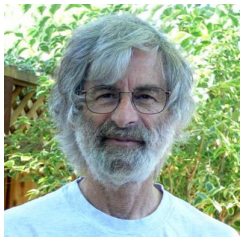


Figure: Leslie Lamport

Specify Large Systems

- DynamoDB: scalable high-performance "no SQL" data store with cross datacenter replication and strong consistency guarantees
- First informal proofs and excessive (fault-injecting) testing

Specify Large Systems

- DynamoDB: scalable high-performance "no SQL" data store with cross datacenter replication and strong consistency guarantees
- First informal proofs and excessive (fault-injecting) testing

- TLC found very subtle bug: shortest error trace 35 steps
- "Using TLA$^+$ in place of traditional proof writing would thus likely have improved time to market, in addition to achieving greater confidence in the system's correctness." [Newcombe, 2014, Newcombe et al., 2015]

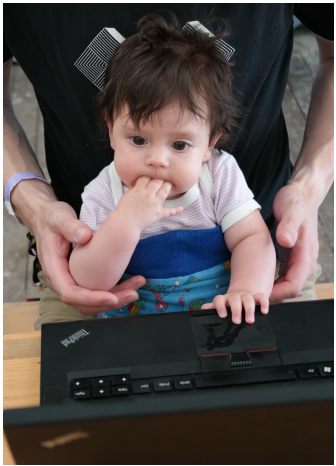
Specify Large Systems

- DynamoDB: scalable high-performance "no SQL" data store with cross datacenter replication and strong consistency guarantees
- First informal proofs and excessive (fault-injecting) testing

- TLC found very subtle bug: shortest error trace 35 steps
- "Using TLA$^+$ in place of traditional proof writing would thus likely have improved time to market, in addition to achieving greater confidence in the system's correctness." [Newcombe, 2014, Newcombe et al., 2015]

Everybody Happy?

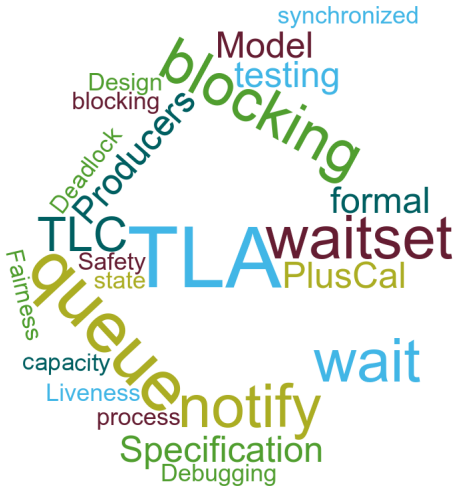
Everybody Happy?



What Next?

- Provide Feedback on CODE!
  - http://aka.ms/CODE-Feedback
- Learn TLA+:
  - http://aka.ms/TLAbq
  - http://aka.ms/TLA
  - http://aka.ms/TLAclass
    - Sign up for a hands-on class
    - Raffle: Handsigned by LL copy of Specifying Systems book

Q&A


synchronized
Model
testing
blocking
Design
blocking
Producers
Deadlock
formal
TLC
TLA
waitset
Safety
state
PlusCal
Fairness
queue
wait
capacity
Liveness
process
notify
Specification
Debugging

Bibliography I

Chris Newcombe. Why Amazon Chose TLA+. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Alfred Kobsa, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Demetri Terzopoulos, Doug Tygar, Gerhard Weikum, Yamine Ait Ameur, and Klaus-Dieter Schewe, editors, Abstract State Machines, Alloy, B, TLA, VDM, and Z, volume 8477, pages 25–39. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-662-43651-6 978-3-662-43652-3. URL http://link.springer.com/10.1007/978-3-662-43652-3_3.

Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, and Michael Deardeuff. How Amazon Web Services Uses Formal Methods. Communications of the ACM, 58(4):66–73, March 2015. ISSN 00010782. doi: 10.1145/2699417. URL http://dl.acm.org/citation.cfm?doid=2749359.2699417.

"Writing is nature's way of letting you know how sloppy your thinking is." Guindon

- First release of the Xbox 360
- MSR intern spec'ed IBM's memory coherence protocol
- Writing the spec revealed a subtle bug
- IBM acknowledge the bug only after several weeks
- Chips would have deadlocked after $\sim$ 4 hours of use
- Xbox Christmas launch would have been missed

Ratio BufCapacity, Consumers, and Producers

Deadlock iff:

$$2K < |Consumers \cup Producers|$$