Software Transactional Memory, For Reals

Brett Hall

www.wyatt.com

CppCon 2014

Just a research toy?

Just a research toy?

We've been using it for over three years in software that is shipping (Dynamics)

You might have some Questions:

- How did it go?

You might have some Questions:

- How did it go?
  - TL;DR: Great, but **YMMV!**

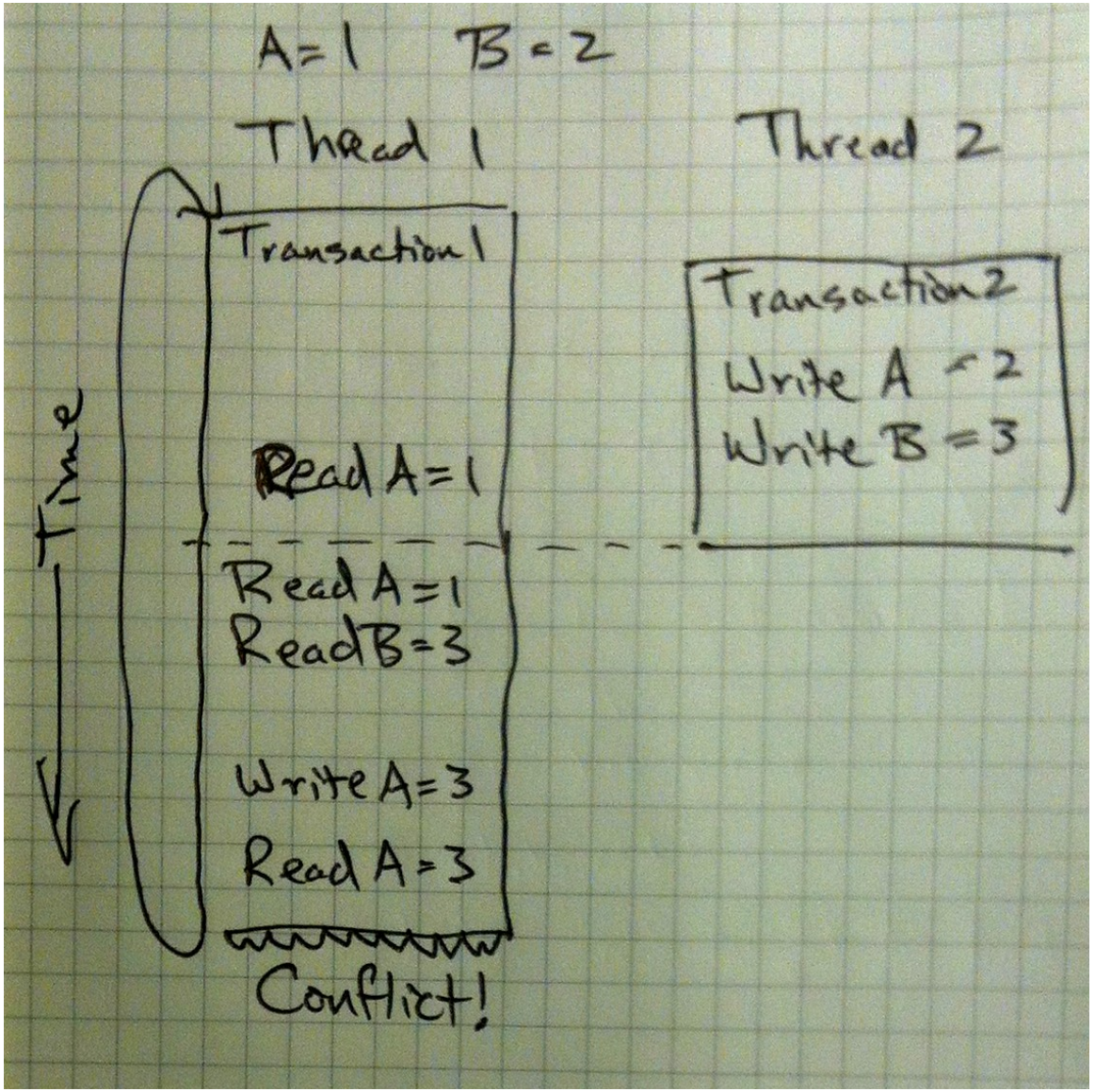
You might have some Questions:

- How did it go?
  - TL;DR: Great, but **YMMV!**
- Why?
  - Are (or were) you insane?

You might have some Questions:

- How did it go?
  - TL;DR: Great, but **YMMV!**
- Why?
  - Are (or were) you insane?
- What?

What?



Jargon: Unbounded, Explicit, Weakly Atomic, Indirected (i.e. not “in-place”)

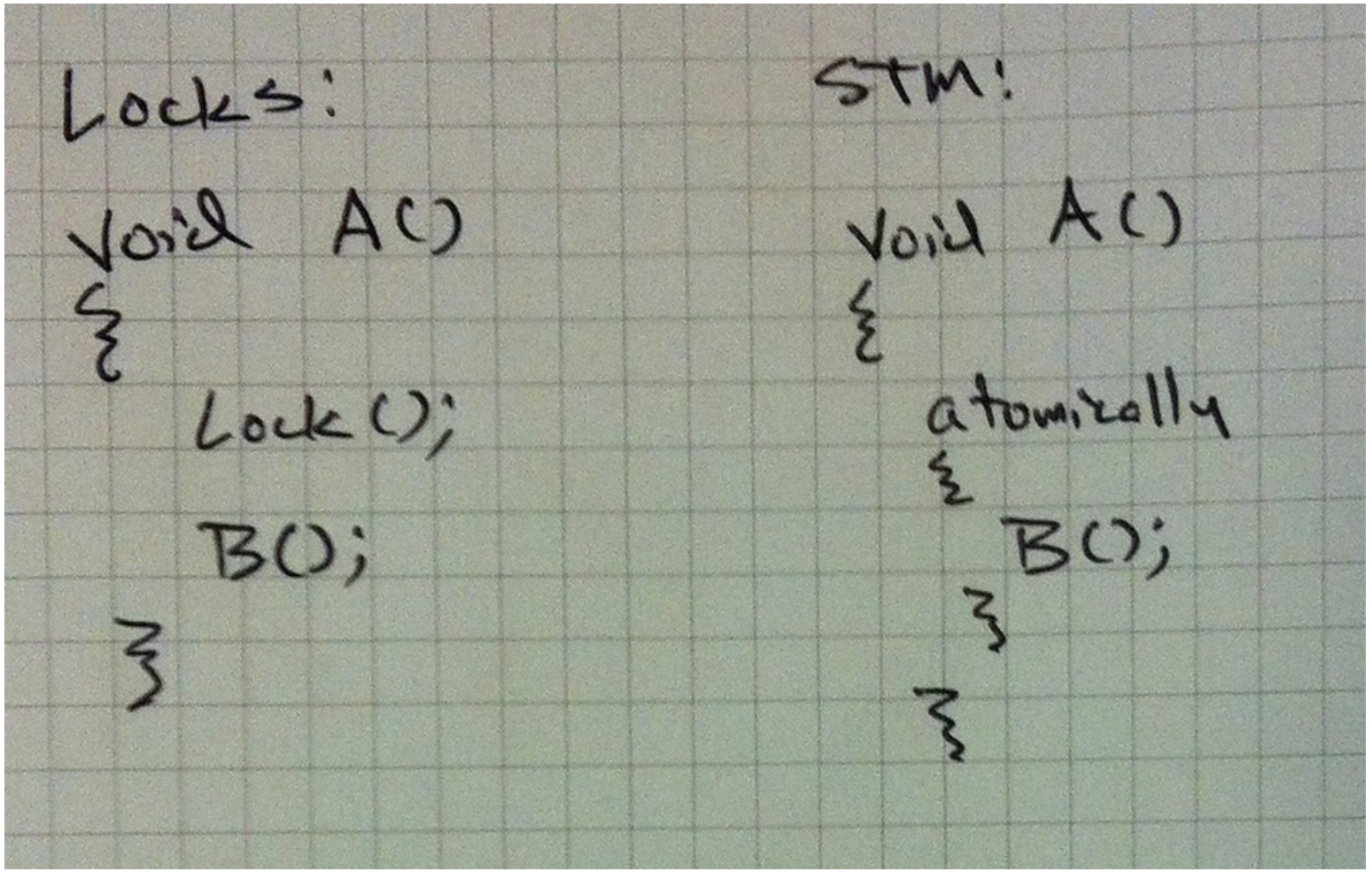
Why (use STM)?

Why (use STM)?

Composibility

Why (use STM)?

Composibility:



Why did we use STM?

Why did we use STM?

Photo credit: Daleus, Curmudgeon-at-Large (Flickr)

How did it go?

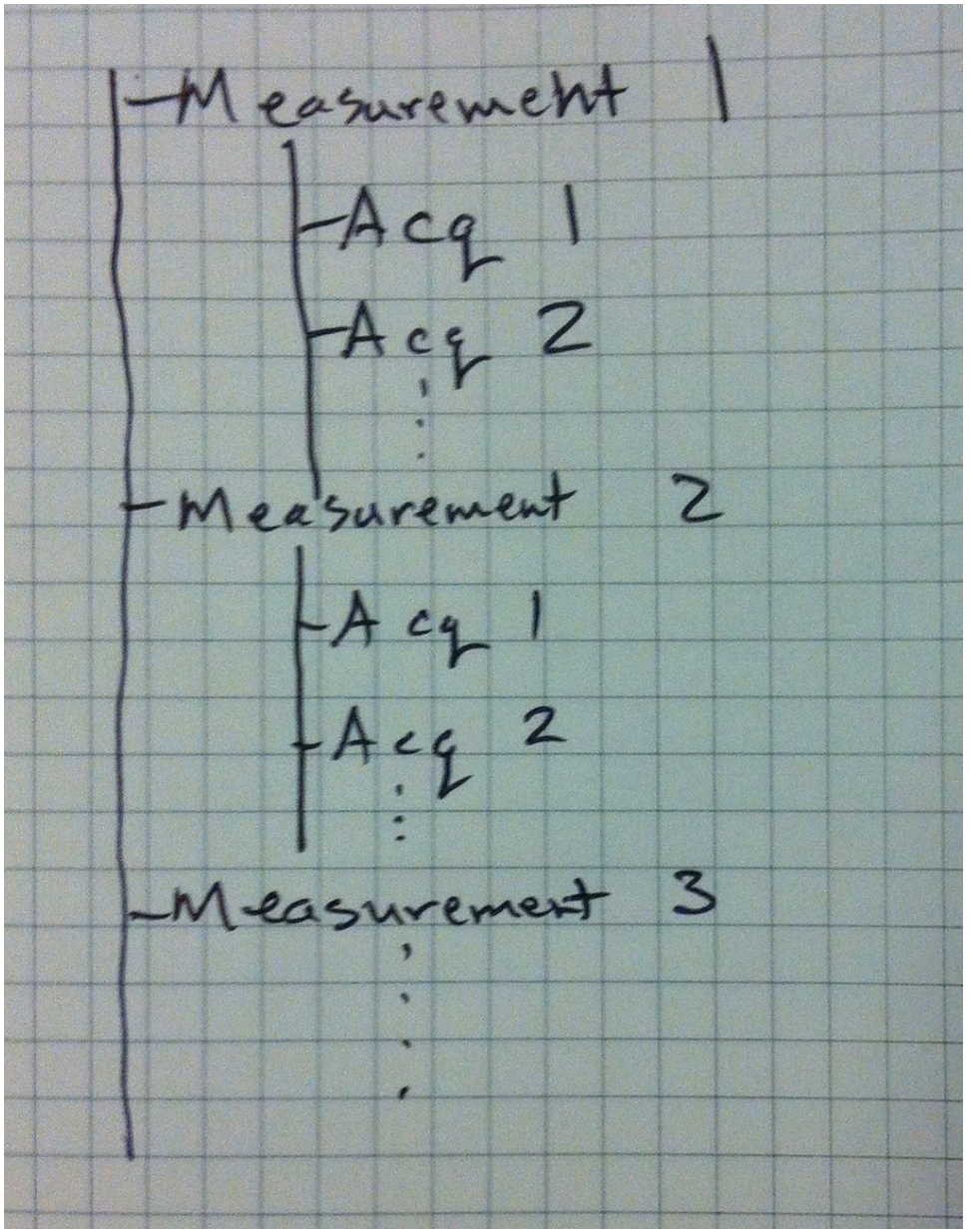
Great, but YMMV

How did it go?

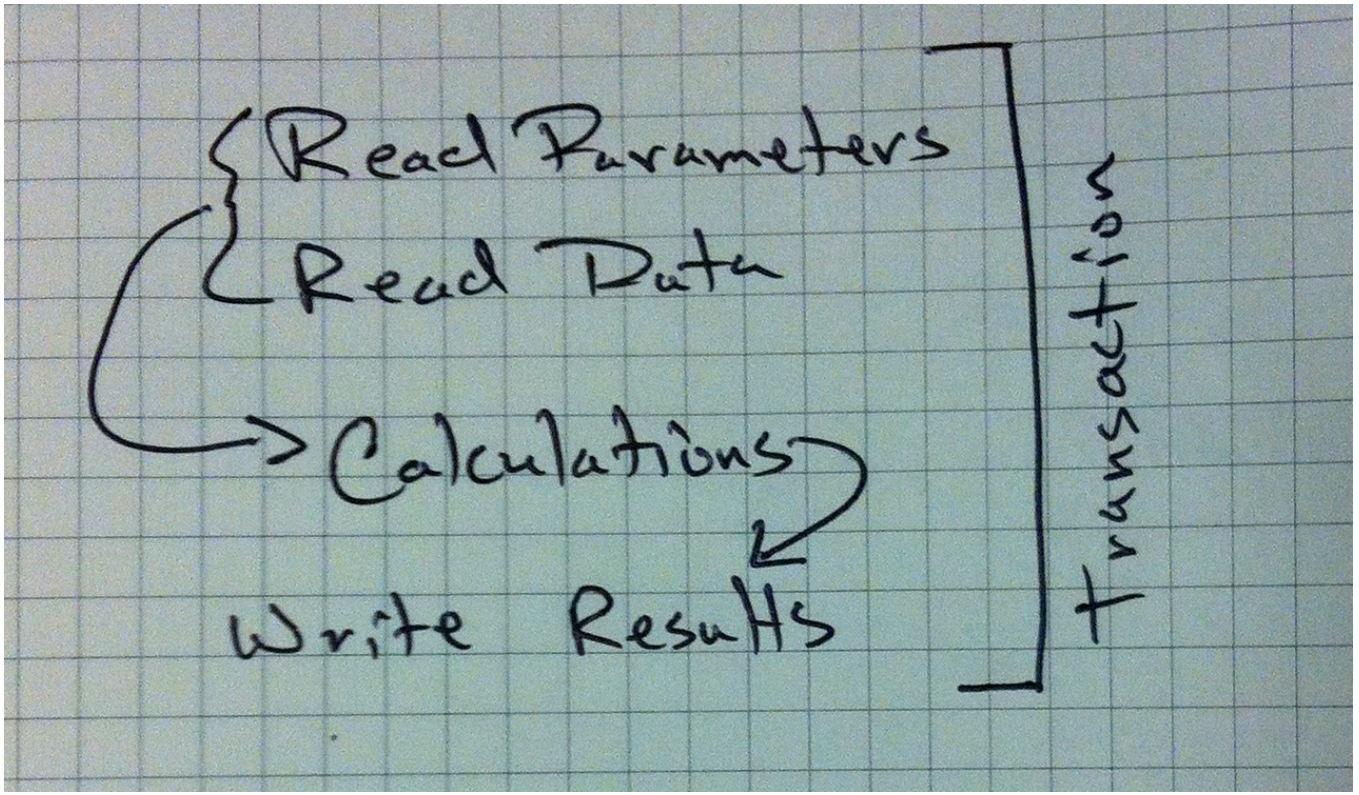
Great, but YMMV

Our application appears to be in the *sweet spot* for STM

Embarrassingly Parallel Structure



Privatized Calculations

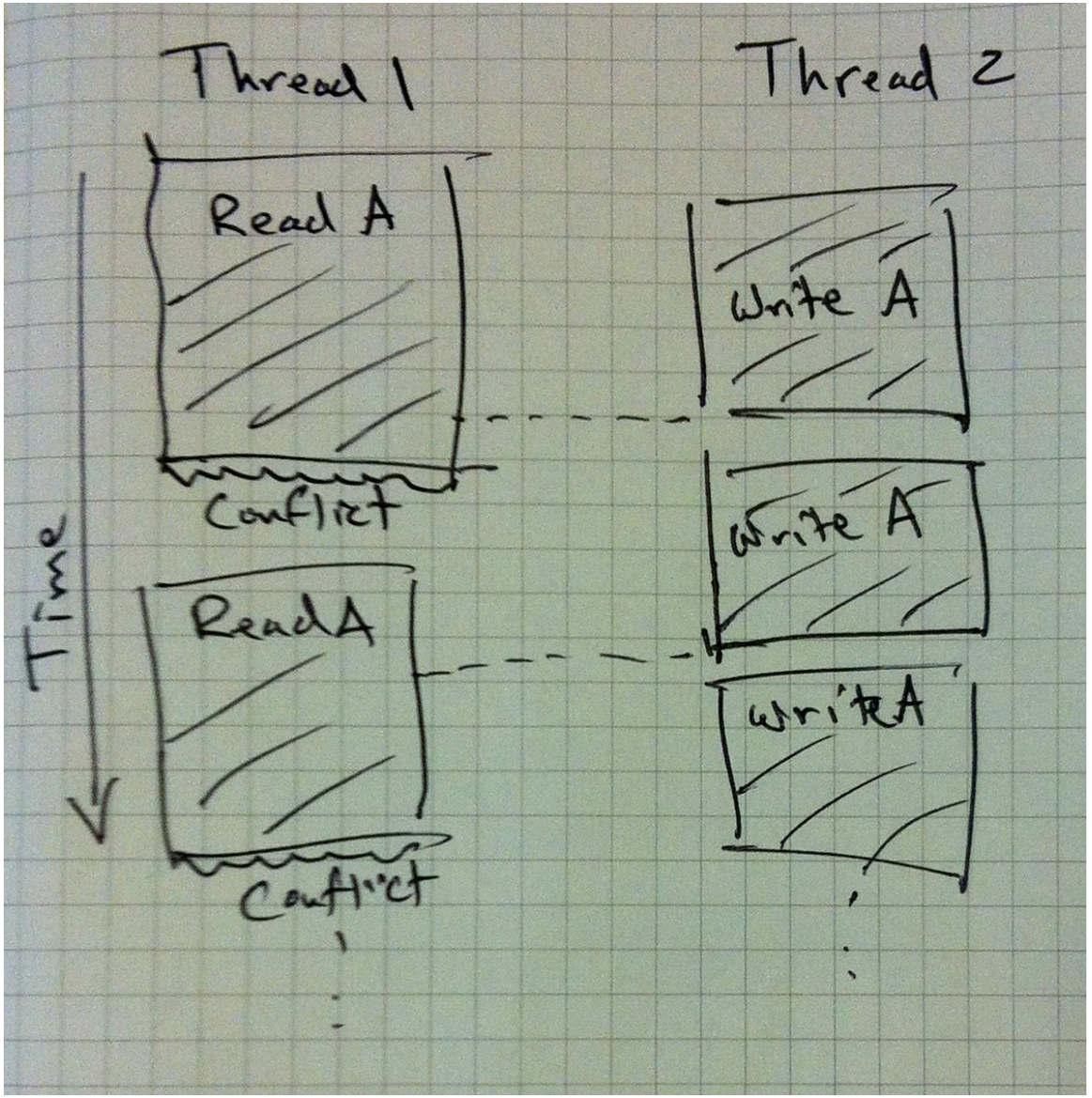


The “Great” Part

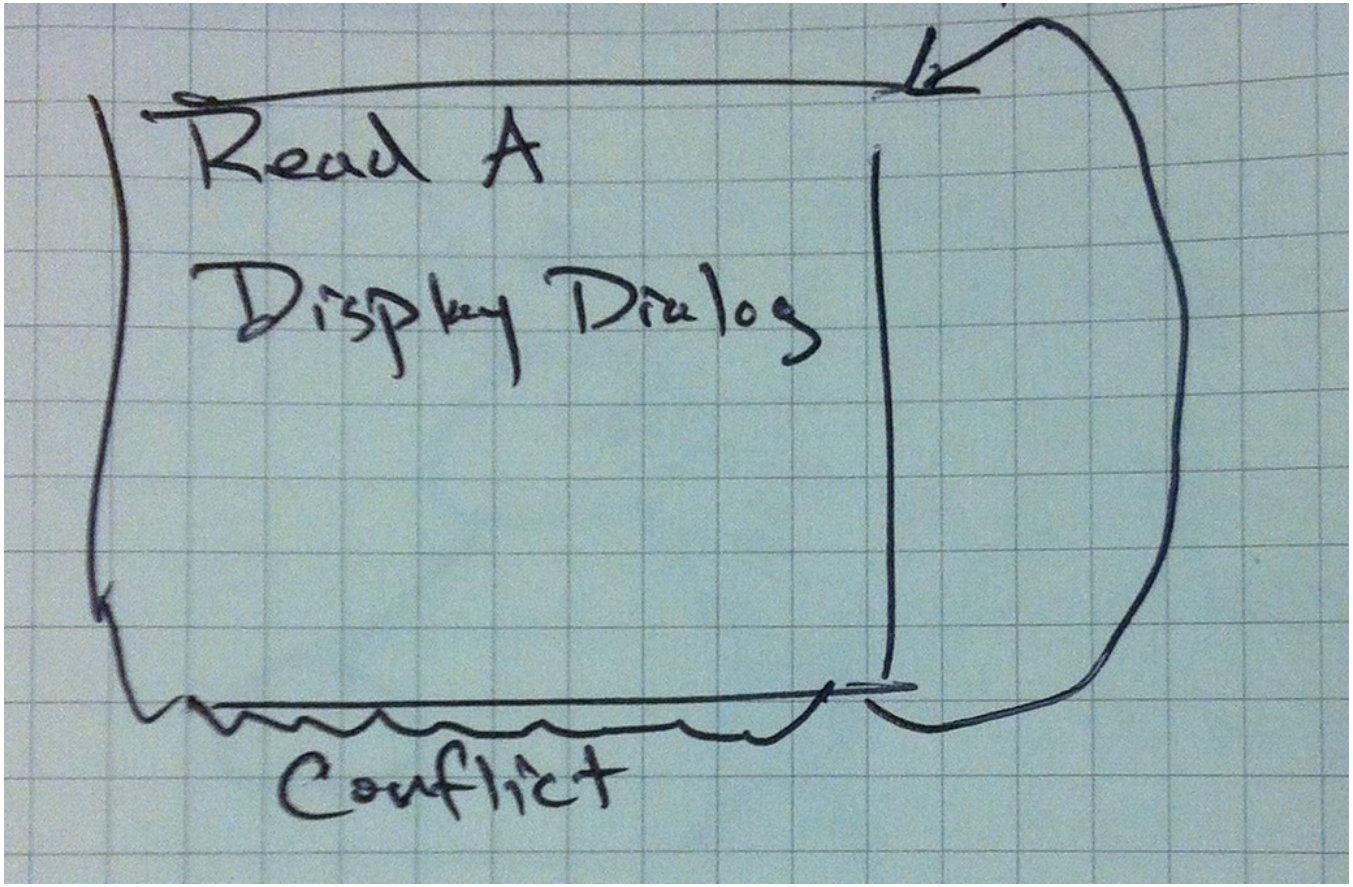
- Easy learning curve
- Much easier to reason about code

Problems I: Starvation

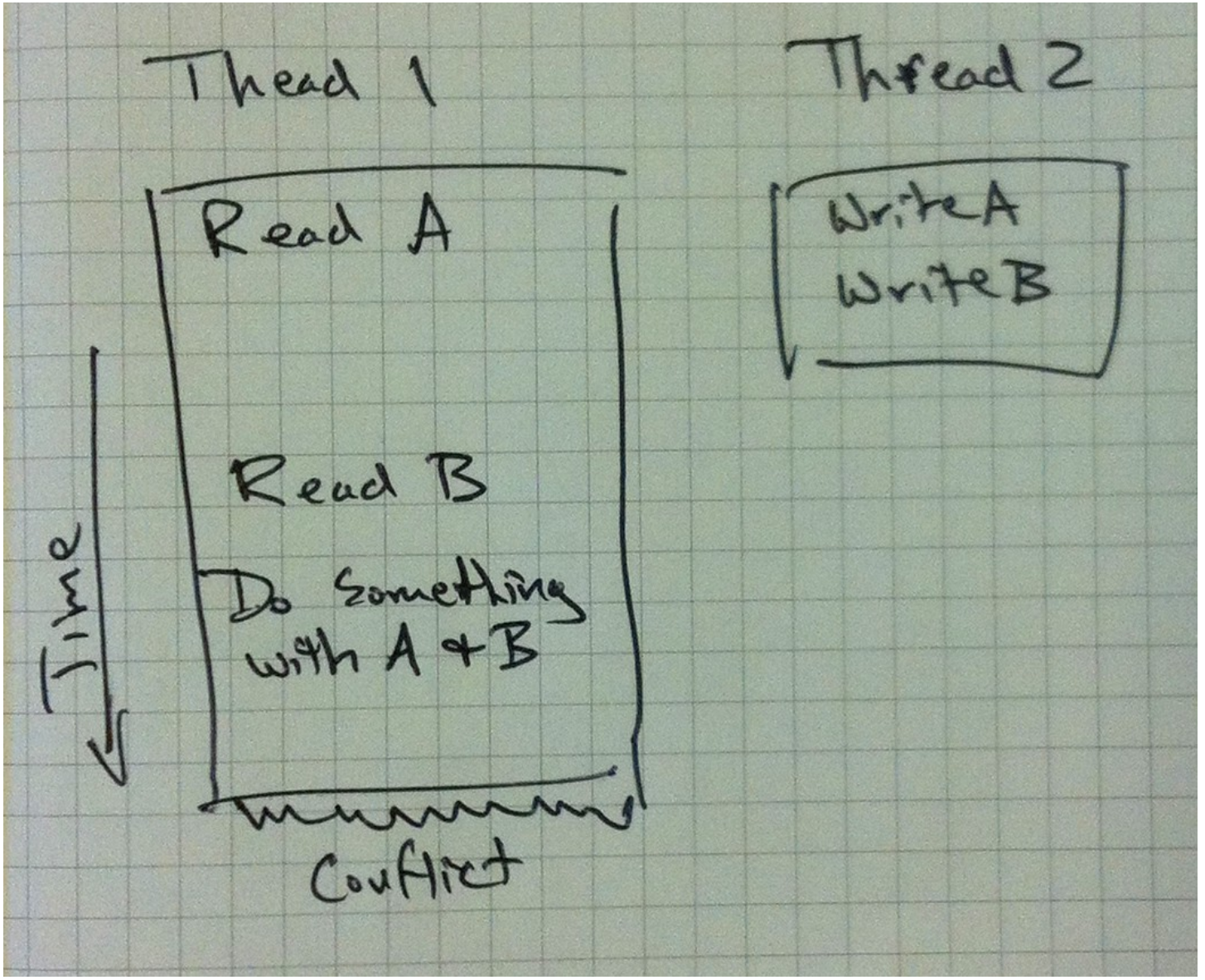
Aka “Live-lock”



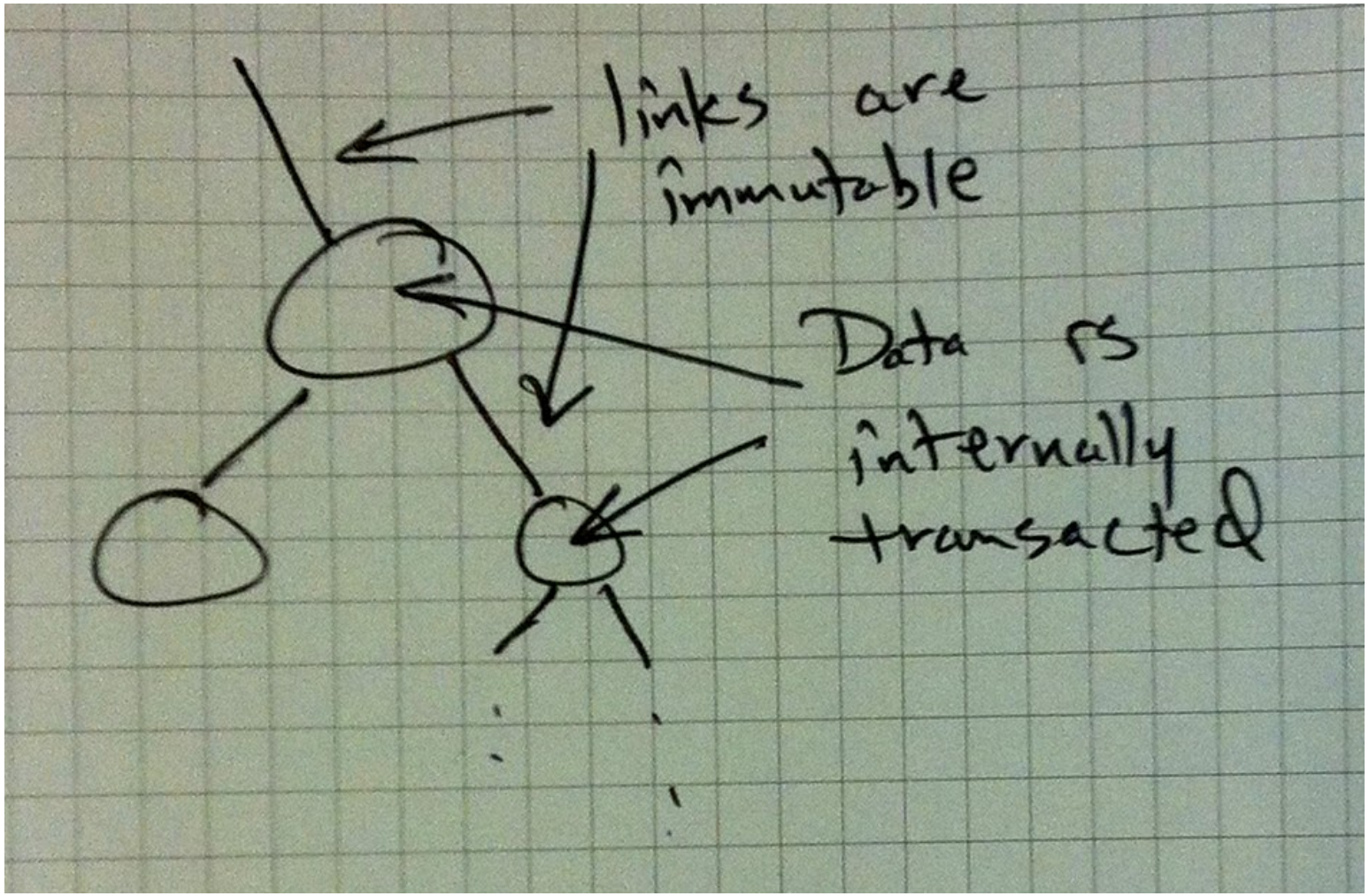
Problems II: Side-effects where they shouldn't be



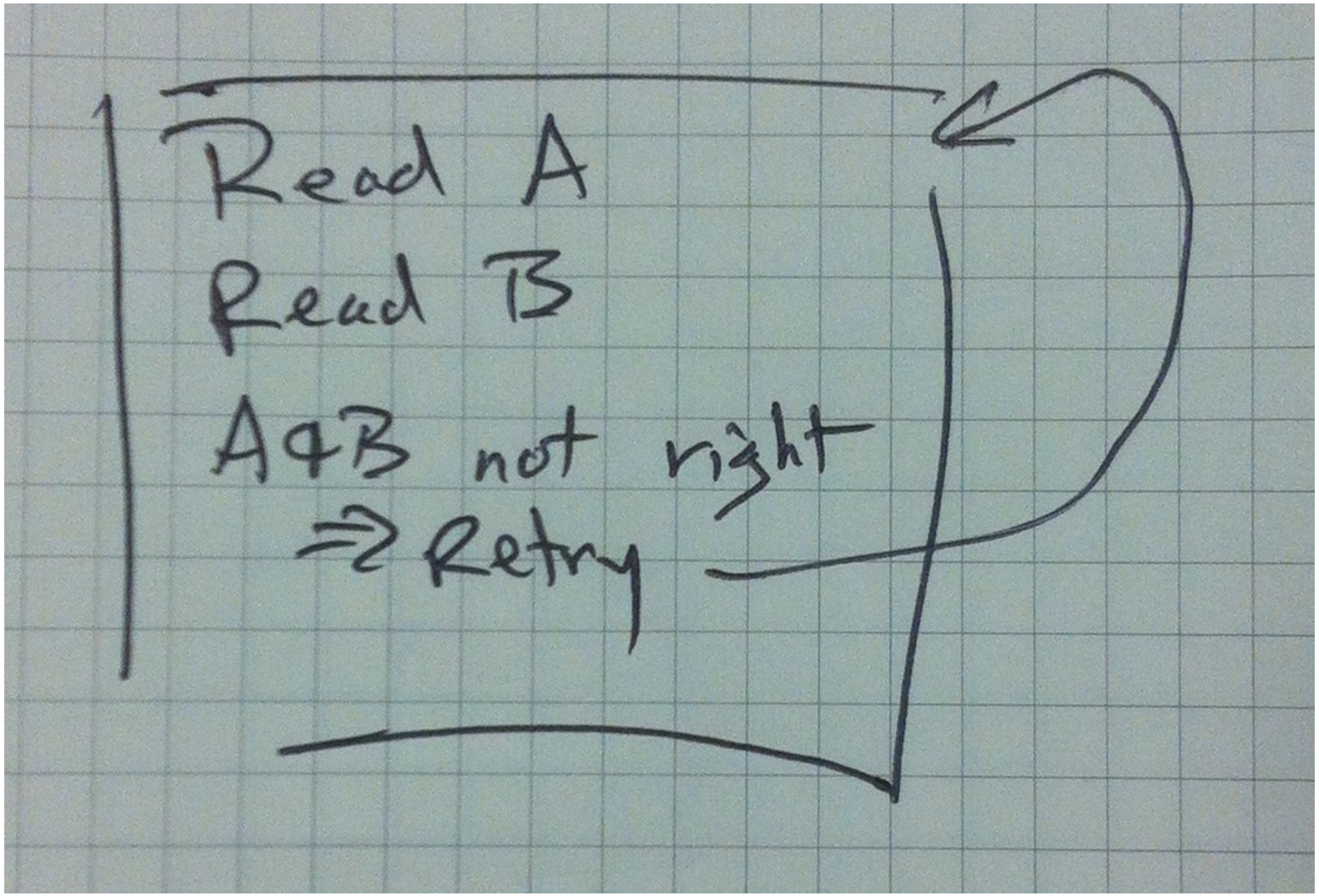
Problems III: Inconsistent Reads



Immutability & Internal Transacting



Icing on the Cake: Retry



“Works fine on my machine”

Did we really need it?

More detail:

backwardsincompatibilities.wordpress.com

Open Content Session

Thursday 8:30pm

Somewhere in the Meydenbauer Center

Want to work with STM?

www.wyatt.com

(we're hiring)