WTF are CRDTs?

Adventures in distributed systems theory

Ben Tyler - YAPC::EU 2015

Follow along!

https://github.com/kanatohodets/crdt-talk

Disclaimer!

- Zero practical experience
- It looked cool
- So I decided to try to talk about it
- Sorry, no code slides

Overview

- Why this talk?
- AP systems
- The trouble with ^ (hint: missing ‘C’)
- Dealing with data conflicts
- Conflict free/convergent data types
- Data types: G-Counter, OR-Set
- ???

Why?

- Peer ~~pressure~~ learning group.
- Scaling reads is "easy."
- High availability is "easy."
- Horizontally scaling writes in a useful way? Really hard.

Scaling writes means…

- Multiple "masters" – many nodes that can accept writes (see: sharding).
- No 'stop the world' synchronization (that is, asynchronous replication).
- Say goodbye to SQL transactions

Scaling writes means…

(Distributed systems theory edition)

- Masterless
- Eventually consistent
- AP-ish

Aside: WAT? CAP?

- Consistency, availability, partition tolerance….

- Works like a single copy (not ACID!)
- Available for updates
- Readable under netsplit

AP ...ish?

- Consistency, availability, partition tolerance…
- Pick 2
- But more like 0.7/0.8/0.5
- Riak, Cassandra: "NoSQL" distributed key-value stores.

Cassandra

What flavor of distsys?

AP


THE
GOOD
THE
BAD
AND THE
WOOKIEE

AP: The Good

- A(vailability)
- P(artition tolerance)
- (potentially) Linear write scalability

AP: The Bad

- Eventually consistent (I just PUT it there, where'd it go?)
- Network overhead (maybe)

AP: The Wookiee

Data conflicts


MERGE CONFLICT

Resolving data conflicts

- Last write wins (bet the farm on ntpd)
- Causality tracking (logical clocks, version vectors/vector clocks, interval tree clocks)
- Give up and ask the client (siblings)

Last Write Wins

Time, LWW, and the farm

- "All objects with a lower timestamp will be silently deleted until GC removes the tombstone record–which means that a rogue client or node can cause the destruction of every write to a record for *days to weeks* afterwards"
  - https://aphyr.com/posts/299-the-trouble-with-timestamps

Causality tracking?

- Attempt to establish 'causality' – who has the more up to date data?

- Can't resolve every conflict (create a 'total ordering' – only partial).

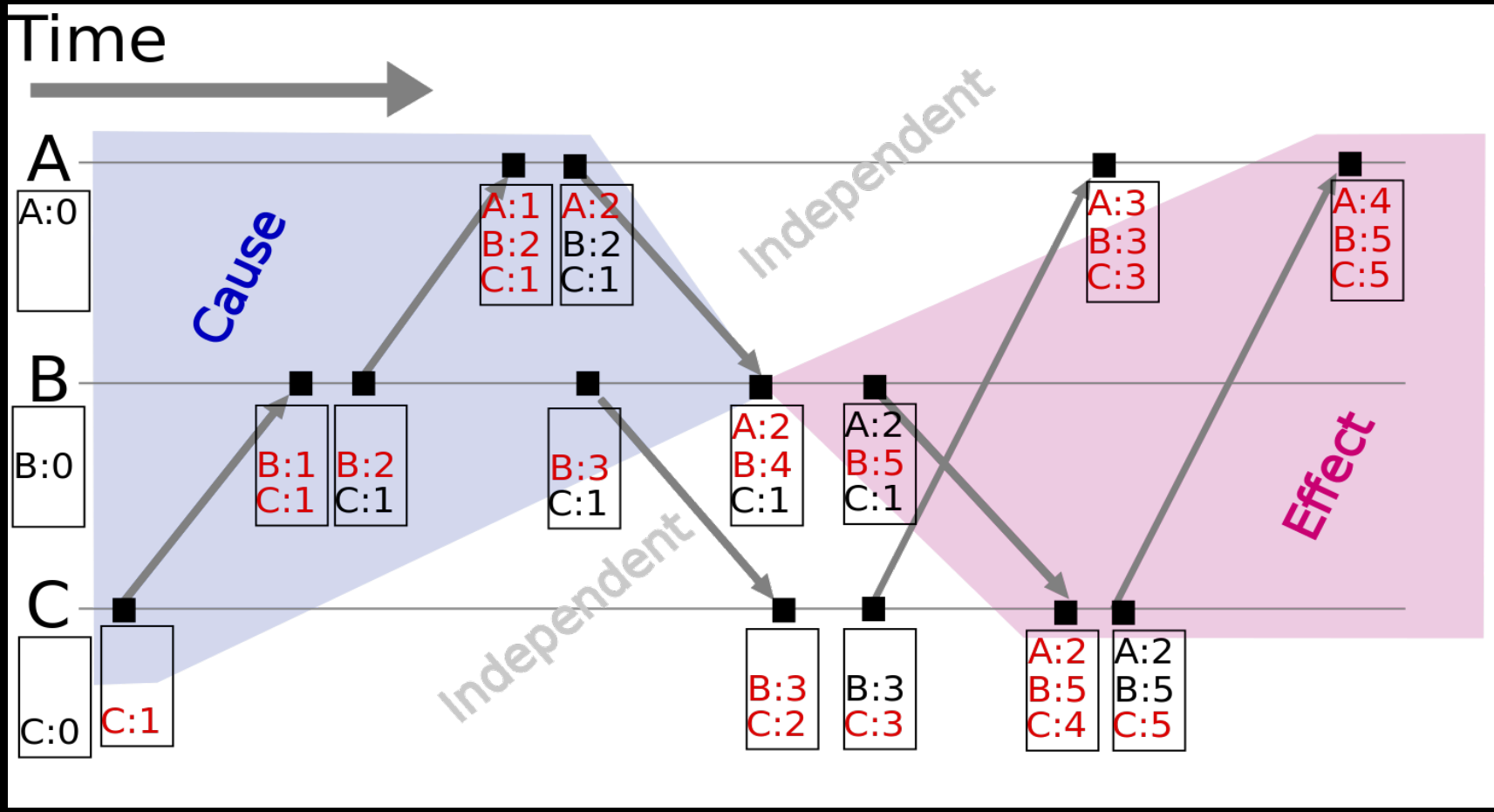
Causality tracking?

Vector clocks

- Every node has a counter. It increments whenever something happens.
- "Something" == receive message or "internal event"
- "Internal event" == client PUT something
- Any message includes all the clocks a node knows about

Causality tracking?

Vector clocks



Give up and ask the client

- Merge, then tell database the new state

- "In all such systems, we find developers spend a significant fraction of their time building extremely complex and error-prone mechanisms to cope with eventual consistency and handle data that may be out of date."
  - The GOOG (F1 paper)


EVERYTHING IS
AWFUL!
memegenerator.net

Can’t someone do that for me?

Merging data is hard, let’s not do it

Enter CRDTs

CRDTs

Commutative/convergent/conflict-free

Replicated

Data

Types

CRDTs

Commutative/convergent/conflict-free

Replicated

Data

Types

CRDTs

Commutative/convergent/conflict-free

Replicated

Data

Types

CRDTs

Commutative/convergent/conflict-free

Replicated

Data

Types

operation based: SBR

CRDTs

Commutative/convergent/conflict-free

Replicated

Data

Types

state based: RBR

(with a monotonic/idempotent merge)

Aside: WAT

Monotonic:

- "only moves in one direction"
- e.g. entirely increasing, or entirely decreasing

Idempotent

- "just run it again, don't worry"
- e.g. add an item to a set

CRDTs

Commutative/convergent/conflict-free

Replicated

Data

Types

the whole gang

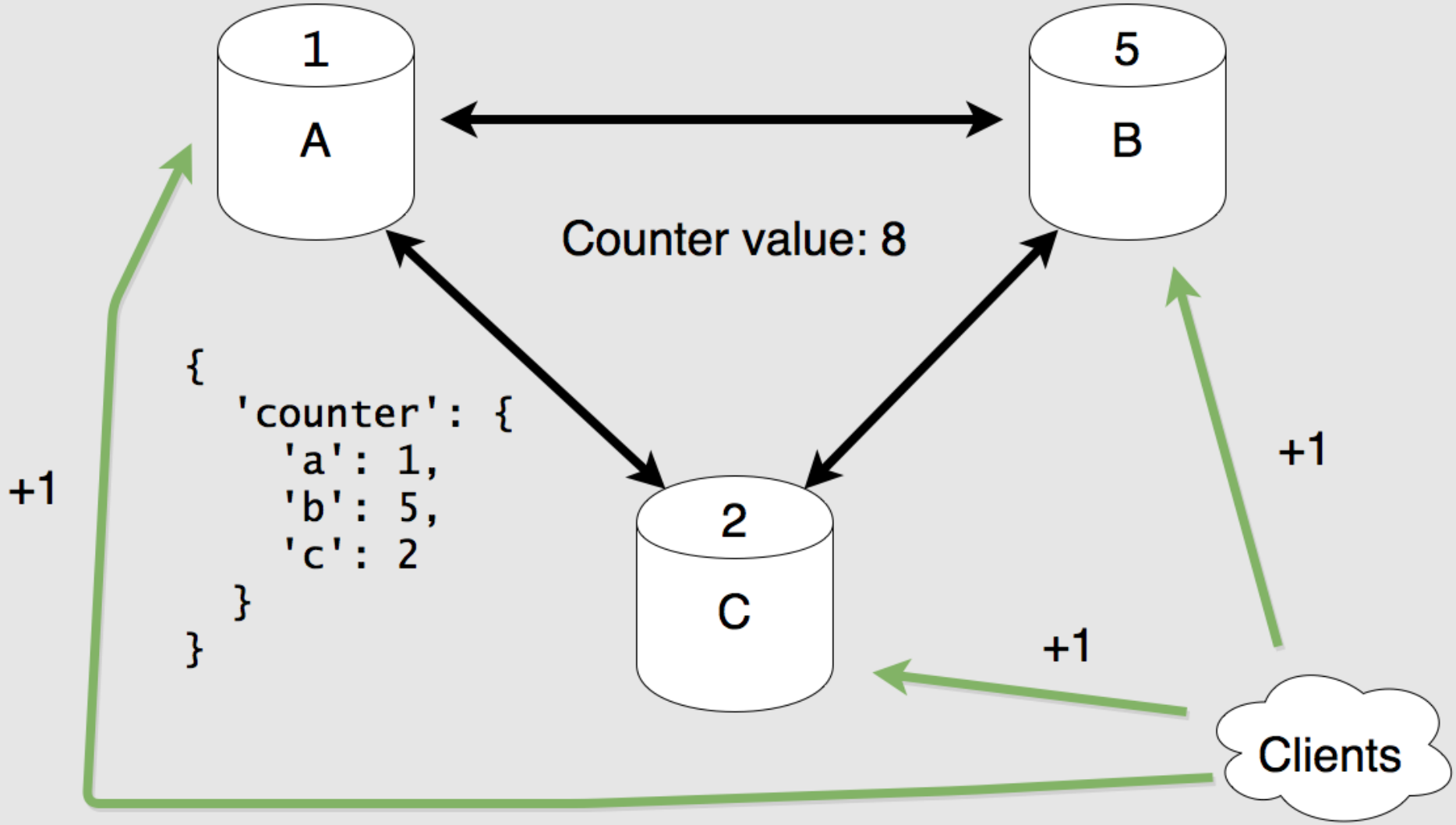
(they're theoretically equivalent)

Like what?

Grow-only counter

- Client says:
  - ‘yo, increment!’
- Track how many increments/replica: total is sum of all replica counts
- Trivially commutative operation
- …not all that interesting
- (or I lack imagination)
- (I totally lack imagination: Youtube view count)

G-Counter cluster



What about negatives?

- Two grow-only counters: increments and decrements.
- Value is increments – decrements.
- If your building blocks are all CRDTs, the result is also a CRDT.

Non-negative counter?

- E.g. gold in the bank in a game
- …global invariant! Noooooo!
- Requires synchronization

Okay, show me something I can use…

Observed-Removed Set

- Set of insertions, each with unique tag
- Set of deletions, each with unique tag
- On conflict, add > delete
- Merge: union of insertion, union of tombstones

Observed-Removed Set

```
{
  'type': 'or-set',
  'e': [
    ['foo', [1]],
    ['bar', [1], [1]],
    ['baz', [1, 2], [2, 3]]
  ]
}
```

Observed-Removed Set

```
{
  'type': 'or-set',
  'e': [
    ['foo', [1]], // foo exists
    ['bar', [1], [1]],
    ['baz', [1, 2], [2, 3]]
  ]
}
```

Observed-Removed Set

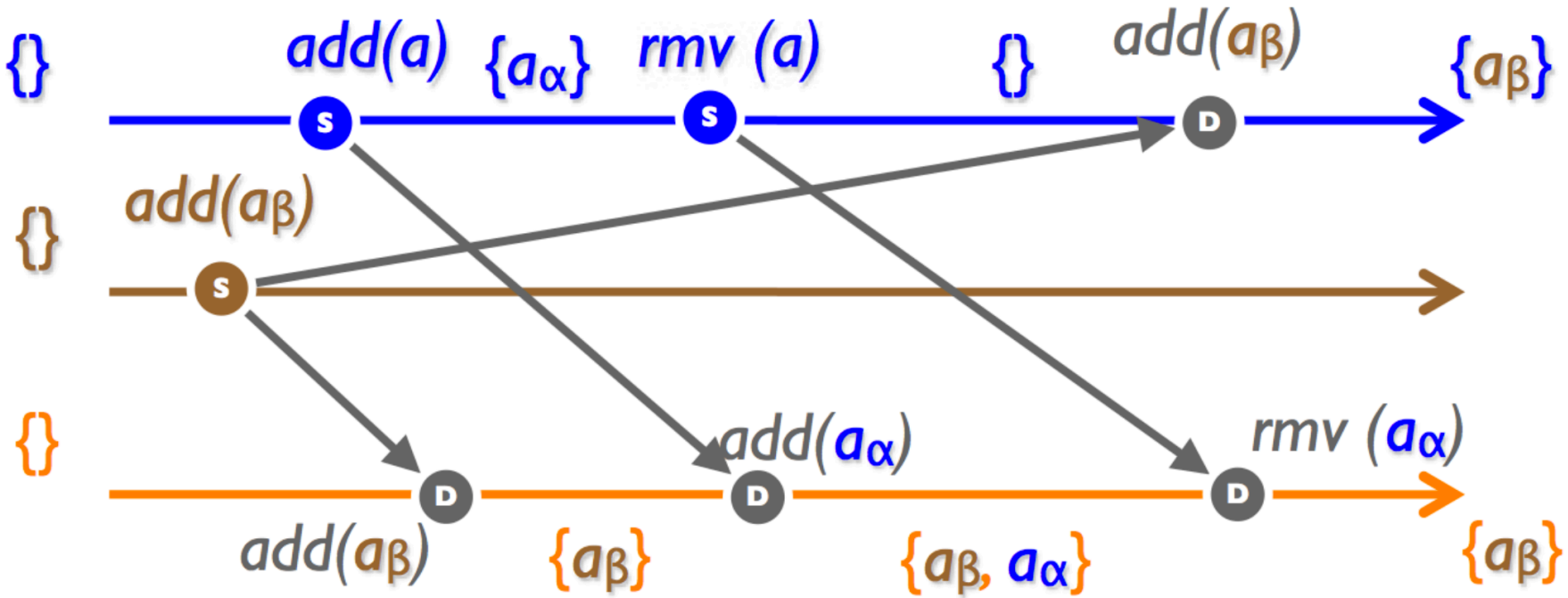
```
{
  'type': 'or-set',
  'e': [
    ['foo', [1]],
    ['bar', [1], [1]], // no bar
    ['baz', [1, 2], [2, 3]]
  ]
}
```

Observed-Removed Set

```
{
  'type': 'or-set',
  'e': [
    ['foo', [1]],
    ['bar', [1], [1]],
    ['baz', [1, 2], [2, 3]]
  ] //baz exists
}
```

Observed-Removed Set

Observed-Removed Set



Caveats

- Still eventual consistency (but stronger)
- Not all client operations will be respected; applications need to be aware.
- Garbage collection can be tricky (potential for unbounded garbage growth)

In the real world…

- Riak data types (Counters, Flags, Sets, Maps, etc.)
- Soundcloud stream: LWW element set
  - https://github.com/soundcloud/roshi
- League of Legends chat: friend list
  - Friends in general
  - Online/offline friends

Bonus! DAG

- Directed Acyclic Graph (e.g. tree)
- Global invariant??
- Locally enforced: only add edges in existing directions

Sources & Papers

- Marc Shapiro, Nuno Preguica, Carlos Baquero, Marek Zawirski. A comprehensive study of Convergent and Commutative Replicated Data Types. [Research Report] RR-7506, Inria — Centre Paris-Rocquencourt. 2011, pp.50.
  - https://hal.inria.fr/inria-00555588
- Lots of papers
  - http://christophermeiklejohn.com/crdt/2014/07/22/readings-in-crdts.html
- http://research.microsoft.com/apps/video/dl.aspx?id=153540
- https://github.com/aphyr/meangirls
- https://aphyr.com/tags/Distributed-Systems
- http://blog.acolyer.org/
- Riak docs:
  - http://docs.basho.com/riak/latest/theory/concepts/context/
  - http://docs.basho.com/riak/latest/theory/concepts/crdts/

The End