

Stasis: Transactions Without SQL

Mark Gitter
Tintri, Inc.

Minnebar
May 21, 2010

Transactions

- Here is the boring example everybody gives:
 - Decrease balance in savings by \$100
 - Increase balance in checking by \$100
 - Ensure that either both occur, or neither, no matter when a failure occurs.

More Interesting Examples

- When a new connection is added to a social network, update the friend-of-a-friend list of everyone affected.
- Rename every occurrence of an attribute name in a set of XML documents.
- Delete a user from a game, and remove all game objects belonging to her.
- Ensure that shutoff commands are sent to both valves, or neither.

SQL transactions

- BEGIN TRANSACTION
- INSERT or UPDATE
- COMMIT TRANSACTION

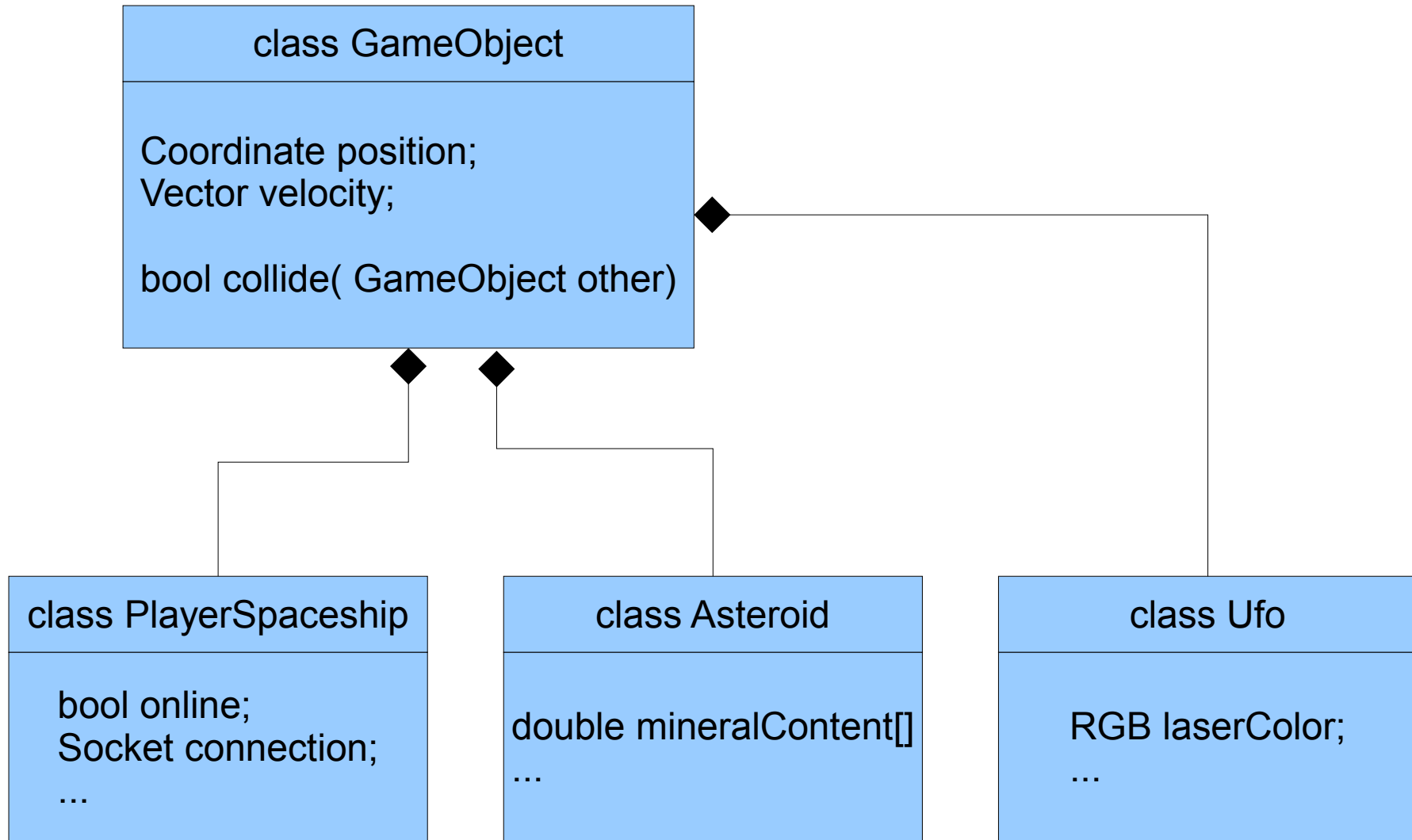
That works great...

Last name	First name	Customer number
Simpson	Homer	2134712341235
Flanders	Edward	1242103487123
Wiggum	Clancy	8709877123434

Account	Customer Number	Type	Balance
007-1123	2134712341235	Savings	\$20.05
008-2345	8709877123434	IRA	\$10,234.56

... if your data looks like this.

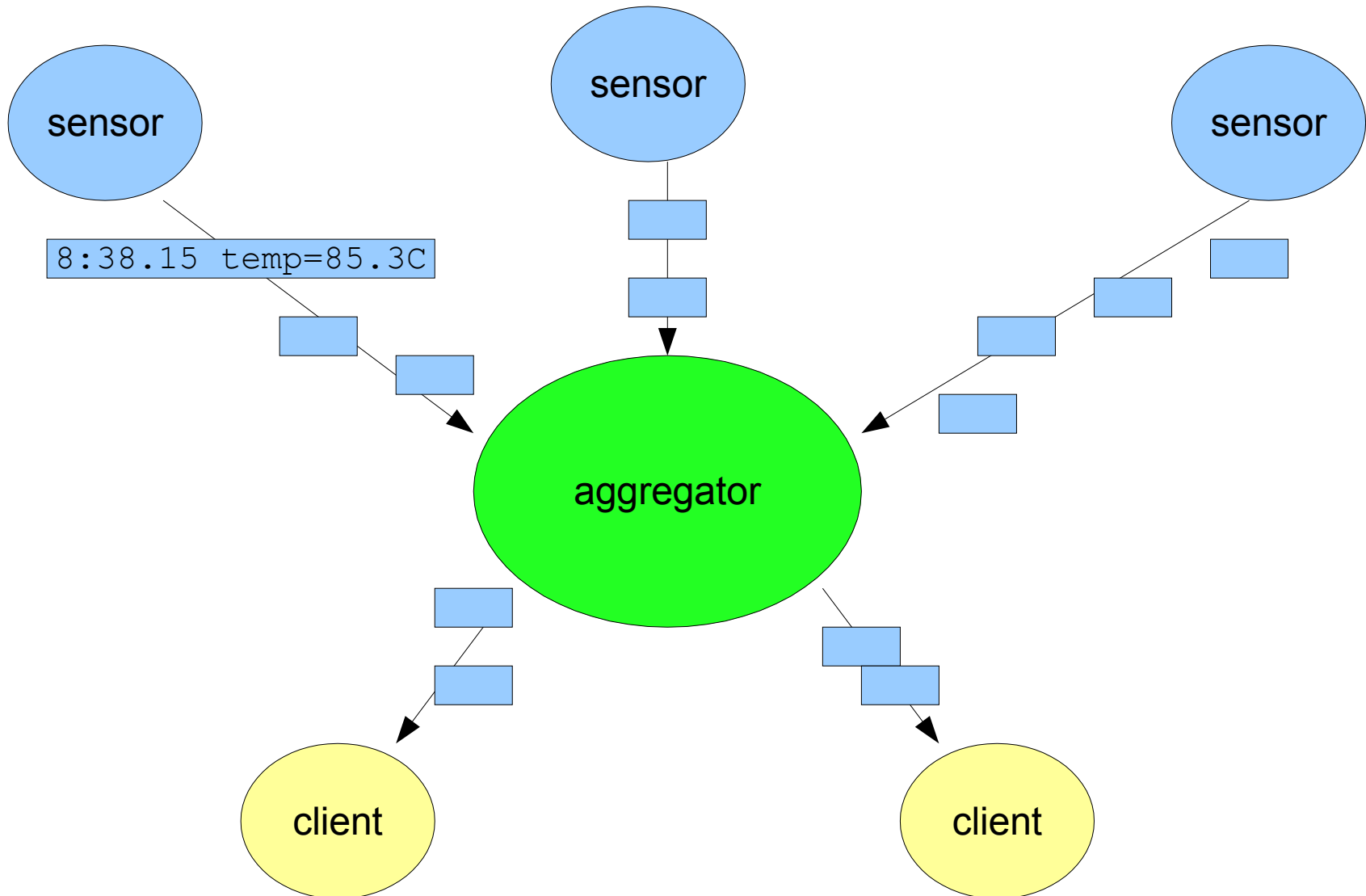
But what if it looks like this?



... or this?

```
<veryimportantform>
  <version>789</version>
  <lastmodified>Tuesday</lastmodified>
  <subsection label="Personal info">
    <applicantName last="Tuttle" .../>
  </subsection>
  <subsection label="Credits requested">
    <taxCredit type="Rice Straw"
tonnage="300" amount="$15"/>
  </subsection>
  <addendum>
    <silly-requirement required="yes"/>
  </addendum>
</veryimportantform>
```

... or this?



Transactional properties are still important...

- Atomic
- Consistent
- Isolated
- Durable

...but other properties differ.

Buffering

Memory
locality

Versioning

Atomic
updates

Class
hierarchy

Unpredictable
data lifetime

Semi-structured
data

Multi-threaded
updates

Stasis

- Separate transactional mechanisms from
 - data representation
 - locking
- Modular architecture lets us specify custom
 - operations
 - buffer management
 - transactional log
 - back-end storage
 - page types

Basic Stasis API

```
int xid = Tbegin();
```

```
Tcommit( xid );
```

Basic Stasis API

```
int xid = Tbegin();
```

```
recordid myObj = Talloc( 8 );
```

```
Tdealloc( xid, myObj );
```

```
Tcommit( xid );
```

Basic Stasis API

```
int xid = Tbegin();
```

```
recordid myObj = ...;
```

```
uint64_t value;
```

```
Tread( xid, myObj, &value );
```

```
value += n;
```

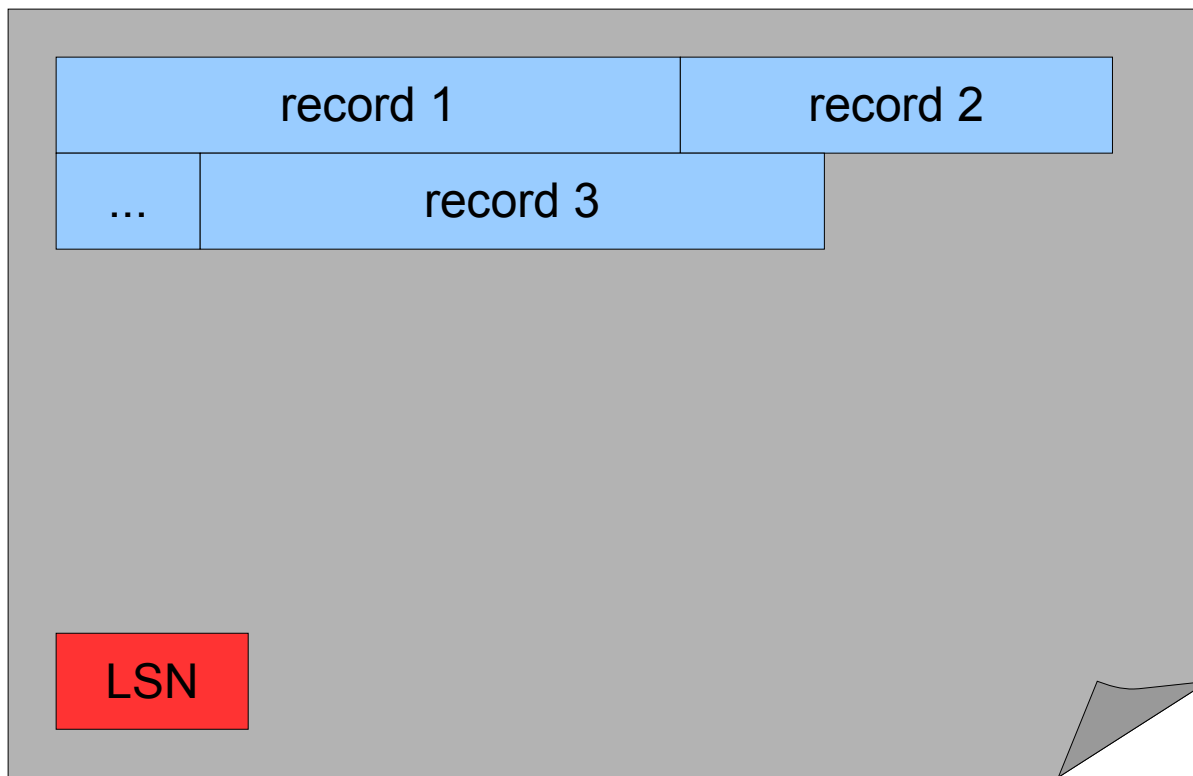
```
Tset( xid, myObj, &value );
```

```
Tcommit( xid );
```

Example #1

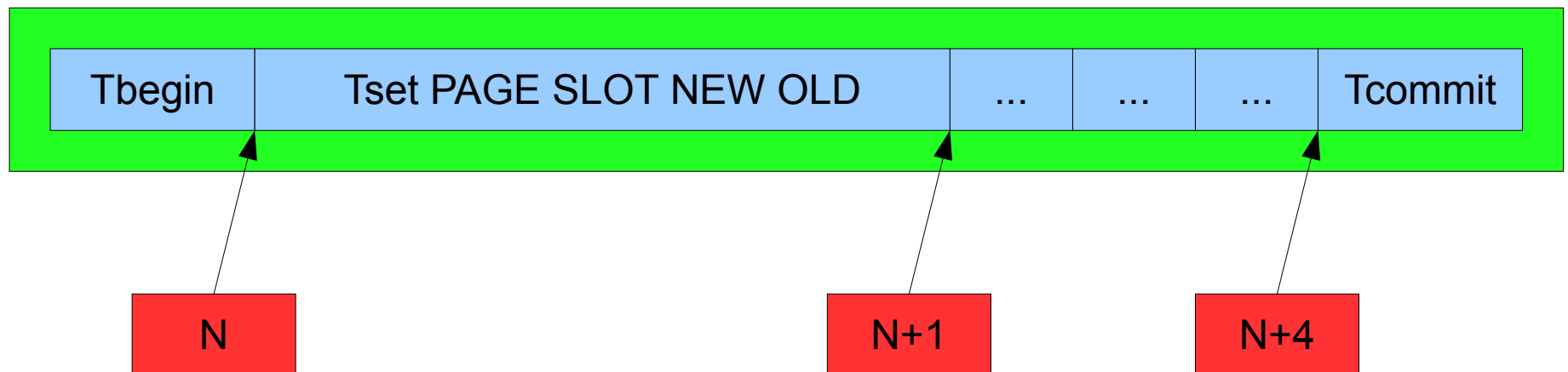
Stasis Internals

- recordid = (page, slot, size)

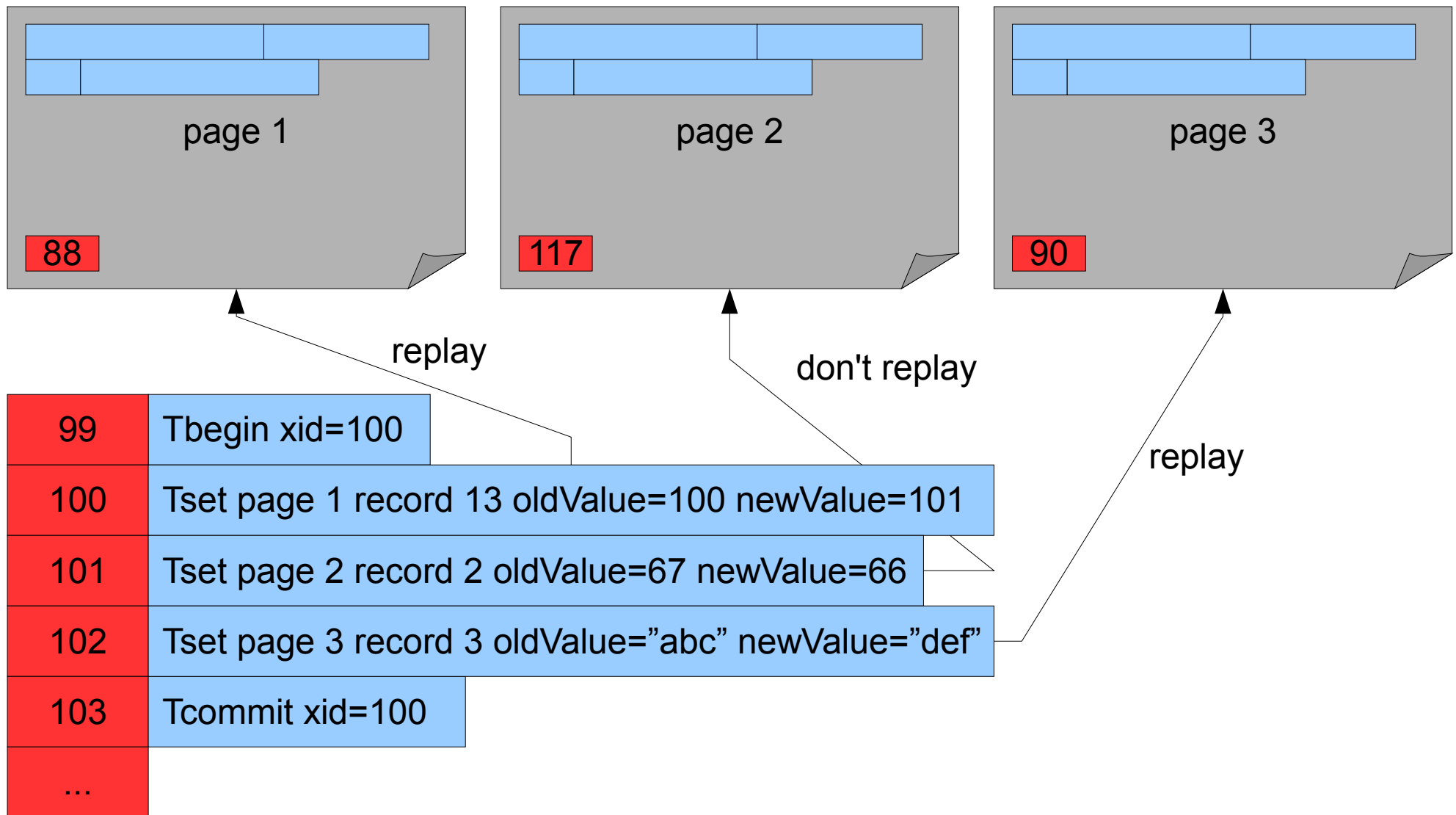


Stasis Internals

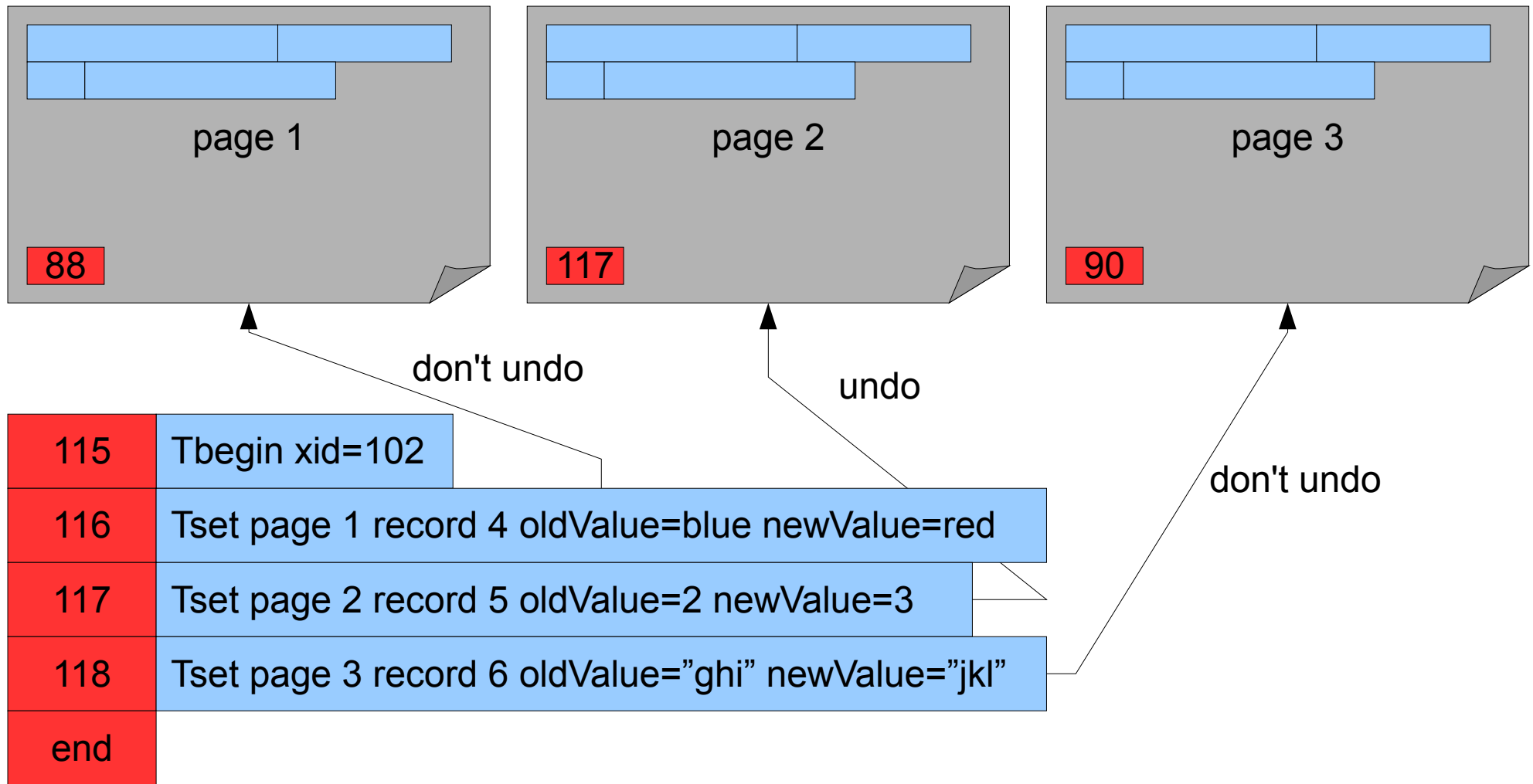
- Transaction log contains:
 - Transaction begin/commit/abort
 - Modifications:
 - Which transaction?
 - Which operation?
 - Which record?
 - New/old values



Recovering Committed Transactions



Rolling Back Incomplete Transactions



Custom Operations

- Stasis allows the programmer to specify new operation types
 - Usually more efficient than just Tset() everywhere!
- Programmer provides:
 - Custom log entry
 - “redo” function
 - “undo” function

Advanced Stasis API

```
int xid = Tbegin();
```

```
recordid myObj = ...;
```

```
MyLogEntry log;
```

```
log.slot = myObj.slot
```

```
log.newValue = 12345;
```

```
Tupdate( xid, myObj, &log,  
         sizeof( log ), MY_CUSTOM_OP );
```

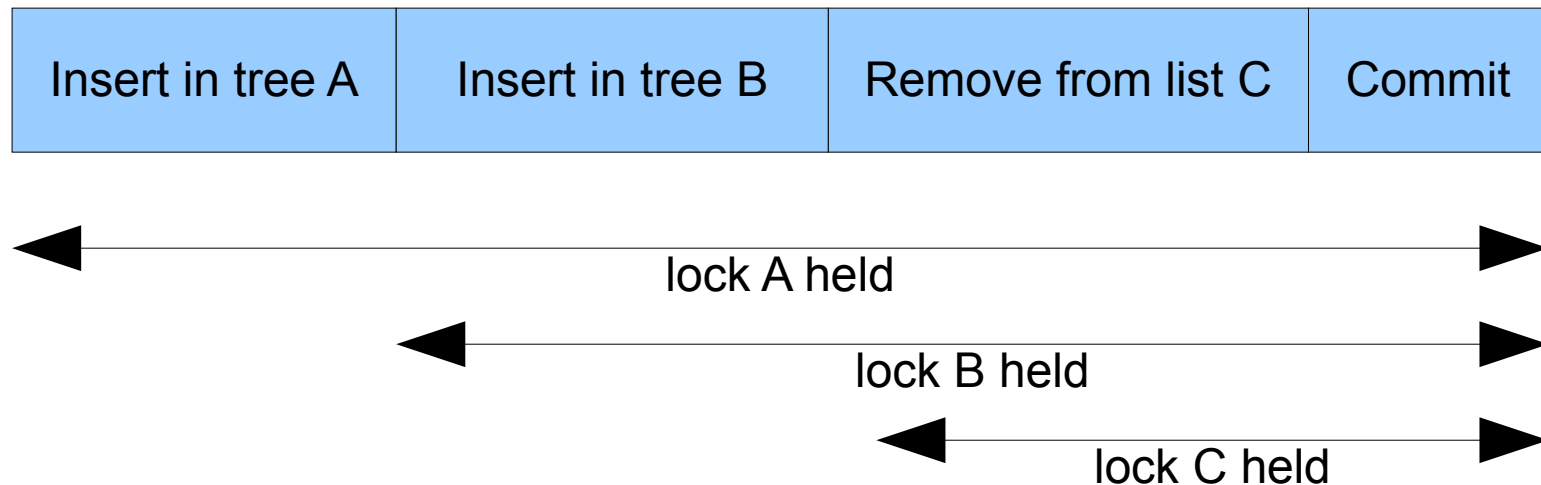
```
Tcommit( xid );
```

Example #2

(just Example #1 implemented more efficiently.)

A Bottleneck...

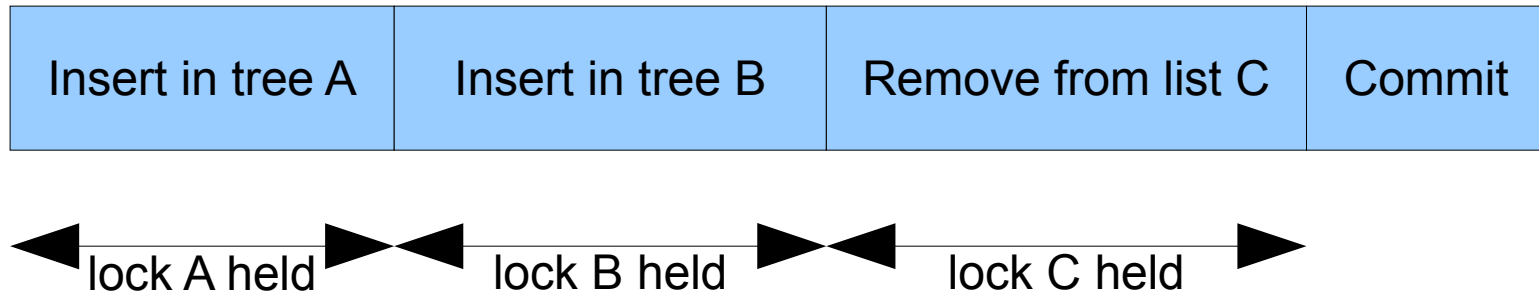
- In a multi-threaded system, you will usually be holding a lock until the end of a transaction.



- Stasis isn't providing the "I" in ACID (Isolation) unless you enforce it.

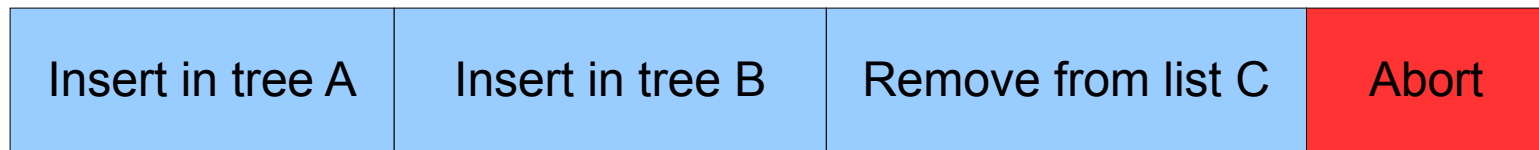
Nested Top Actions

- We can release the lock if we tell Stasis how to revert the logical operation!



Nested Top Actions

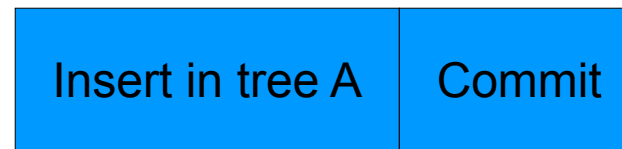
Transaction 1



↑
rebalances tree

↑
corrupts transaction
2's changes

Transaction 2



↑
modifies rebalanced
tree node

Advanced Stasis API

```
TreeInsertLog log;  
log.key = 5;  
log.value = "abcdef";
```

```
void *handle =  
    TbeginNestedTopAction  
        ( xid, TREE_INSERT,  
          &log, sizeof( log ) );
```

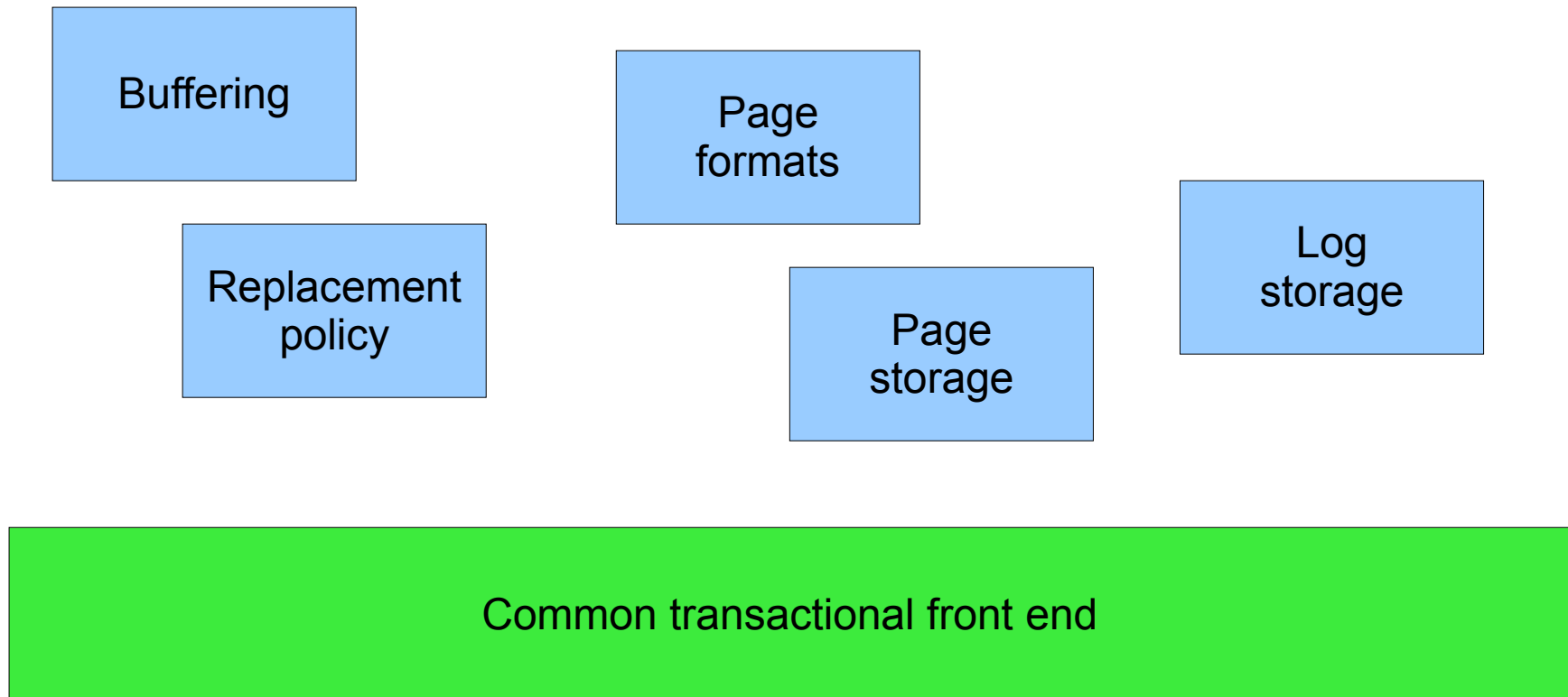
```
...
```

```
TendNestedTopAction( xid, handle );
```

More Stasis Features

- Transactional data structures:
 - linked list
 - array
 - B-tree
 - hash table
- Soft commits
- Read-in-place

Stasis Customization



Thanks for attending!

- Stasis:
 - <http://code.google.com/p/stasis>
 - <http://www.cs.berkeley.edu/~sears/stasis>
 - OSDI 2006 paper, or Rusty's thesis
- This presentation and example code:
 - <http://lowballgurus.com/minnebar/>
 - <http://markgritter.livejournal.com/>