

Coordination Avoidance in Database Systems

Peter Bailis et al.

Presented by Chenggang Wu

Motivation

- Goal: Concurrency & Consistency
- Classic Strategy: Coordination
 - ACID transactions
 - Consensus algorithms
- Effect: “Illusion”
 - One copy of state
 - Serial access to state
- Problem: Coordination is expensive!

Motivation

- Coordination penalizes:
 - Scalability
 - Computing resource \neq Capacity
 - Performance
 - Speed of light limitation
 - Availability
 - Network partition
 - Server Failure

Coordination-Free Execution

- Benefits:
 - Scalability
 - Infinite scale-out
 - Performance
 - Coordination latency is gone
 - Availability
 - Failures and partitions are tolerable

Challenges

- Composing divergent states
 - Eventually need to agree on something
- States remain consistent after composition
 - Consistent = Application-level Correctness
 - Correctness is maintained by invariants

Solutions

- Reconcile by “merging”
 - Union, addition...
 - Bloom^L, CRDTs
- Invariant confluence test (ICT)

ICT

- Can invariants be violated by merging?
 - Yes: Coordination is required
 - No: Coordination can be avoided
- Result depends on:
 - Transactions
 - Invariants

ICT Example

Invariant	Operation	ICT
Balance > 100	Deposit	Yes
Balance > 100	Withdraw	No
Balance < 100	Deposit	No
Balance < 100	Withdraw	Yes

Practicality

Invariant	Operation	<i>I-C?</i>	#	Informal Invariant Description	Type	Txns	<i>I-C</i>
Attribute Equality	Any	Yes	1	YTD wh sales = sum(YTD district sales)	MV	P	Yes
Attribute Inequality	Any	Yes					
Uniqueness	Choose specific value	No	2	Per-district order IDs are sequential	S _{ID} +FK	N, D	No
Uniqueness	Choose some value	Yes	3	New order IDs are sequentially assigned	S _{ID}	N, D	No
AUTO_INCREMENT	Insert	No	4	Per-district, item order count = roll-up	MV	N	Yes
Foreign Key	Insert	Yes	5	Order carrier is set iff order is pending	FK	N, D	Yes
Foreign Key	Delete	No	6	Per-order item count = line item roll-up	MV	N	Yes
Foreign Key	Cascading Delete	Yes	7	Delivery date set iff carrier ID set	FK	D	Yes
Secondary Indexing	Update	Yes	8	YTD wh = sum(historical wh)	MV	D	Yes
Materialized Views	Update	Yes	9	YTD district = sum(historical district)	MV	P	Yes
>	Increment [Counter]	Yes	10	Customer balance matches expenditures	MV	P, D	Yes
<	Increment [Counter]	No					
>	Decrement [Counter]	No	11	Orders reference New-Orders table	FK	N	Yes
<	Decrement [Counter]	Yes					
[NOT] CONTAINS	Any [Set, List, Map]	Yes	12	Per-customer balance = cust. expenditures	MV	P, D	Yes
SIZE=	Mutation [Set, List, Map]	No					

Common SQL and ADT invariants

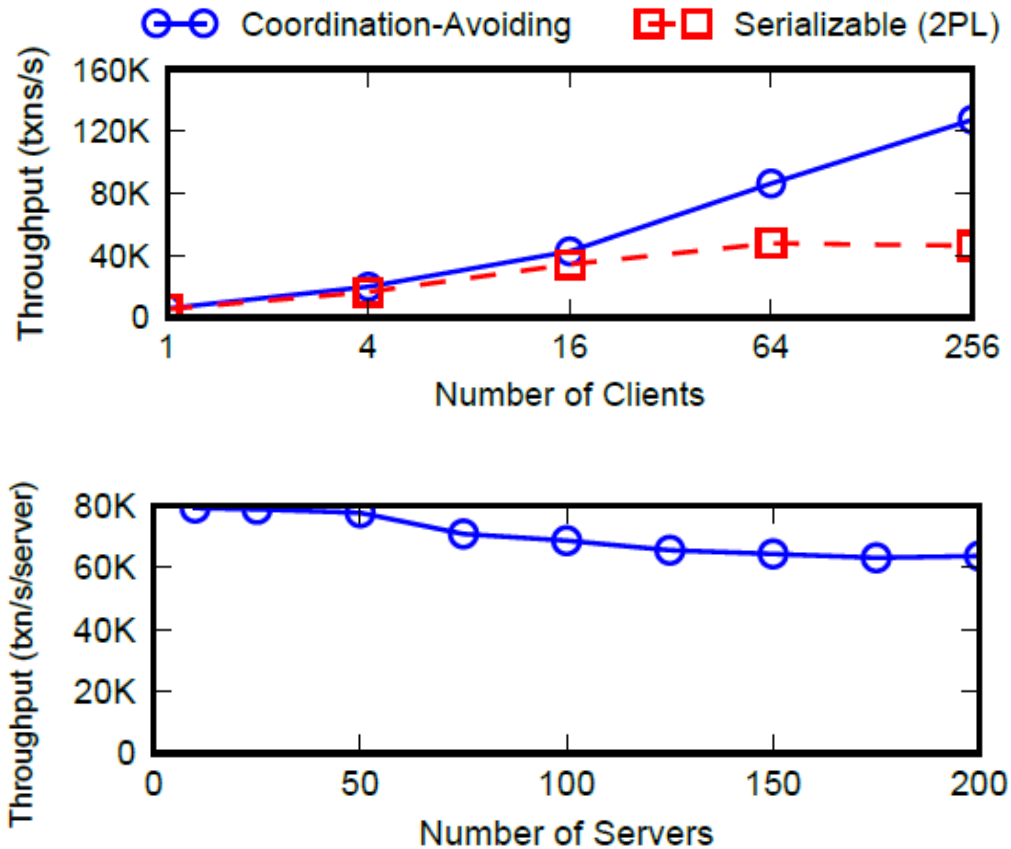
TPC-C Consistency Conditions

Implementation

- RAMP Transaction (SIGMOD 2014)
 - Coordination-free
 - See all updates, or none
 - Sufficient to ensure foreign key & MV invariants

Experimental Evaluation

- TPC-C Benchmark



Discussion

- Trade-off
 - Consistency vs Coordination
 - Transparency vs Developer's effort
 - Invariant specification
 - ICT evaluation
- Eventually Consistent
 - Read current state?