Taking Omid to the Clouds:
Fast, Scalable Transactions for Real-Time Cloud Analytics

O. Shacham • Y. Gottesman • A. Bergman • E. Bortnikov • E. Hillel • I. Keidar
Yahoo Research

Omid

Transactional API over **NoSQL** key value
Client Library + Runtime Service
Open source **– Apache incubator**
Implemented over **Apache HBase**
Snapshot Isolation consistency
Highly Available
Originally optimized for **throughput**

Omid Design Choices

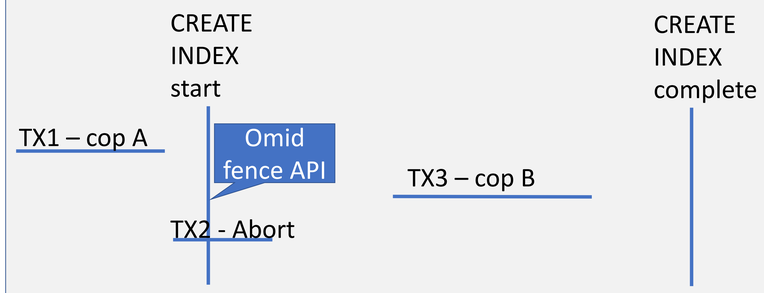
System	Validation	Commit entry writes	Multi tenancy
Perculator (2PC)	D	D	no
CockroachDB	D	D	yes
Omid	C	C	yes
Omid LL	C	D	yes

C – centralized, D – distributed

Phoenix Integration

Secondary index creation

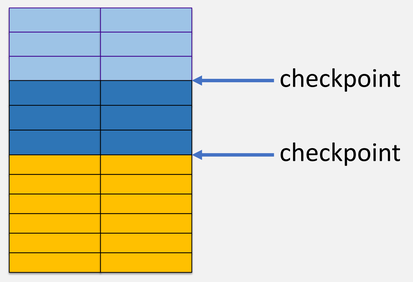
When a new index is created, two coprocessors are installed:
A. Populates new index with history
B. Maintains index by augmenting write to table with a write to index



Beyond snapshot isolation - SIX

```
INSERT INTO T
    SELECT ID+10 FROM T;

INSERT INTO T
    SELECT ID+100 FROM T;
```



Taking Omid To The Cloud

Optimized for low latency – Omid LL

Redesign Omid to eliminate key bottleneck
Distribute commit table updates among clients.

Novel Fast Path API – Omid FP

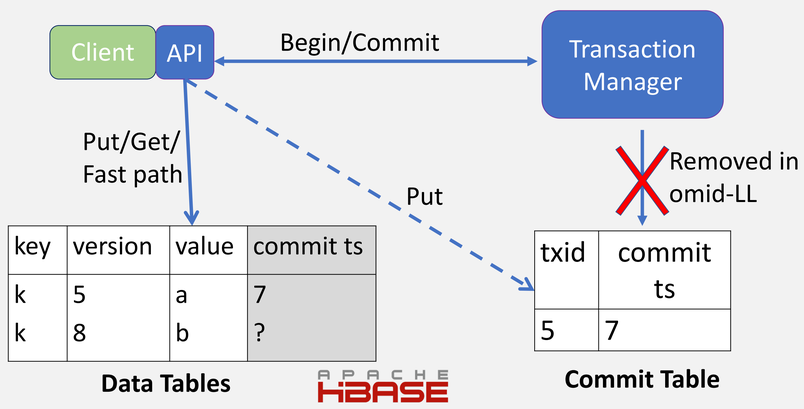
Many workloads have single key transactions
Eliminate transaction overhead for them

Direct access to region server without access to TM
API:
- **brc**(key) • **bwc**(key,value) • **br**(key) + **wc**(key,val)

Apache Phoenix Integration

Integrate Omid as the transactional layer in Phoenix SQL engine

Omid Architecture



Performance