

【VLDB2011勉強会】

Session 15 : Distributed Systems

担当：榎 美紀(日本IBM)

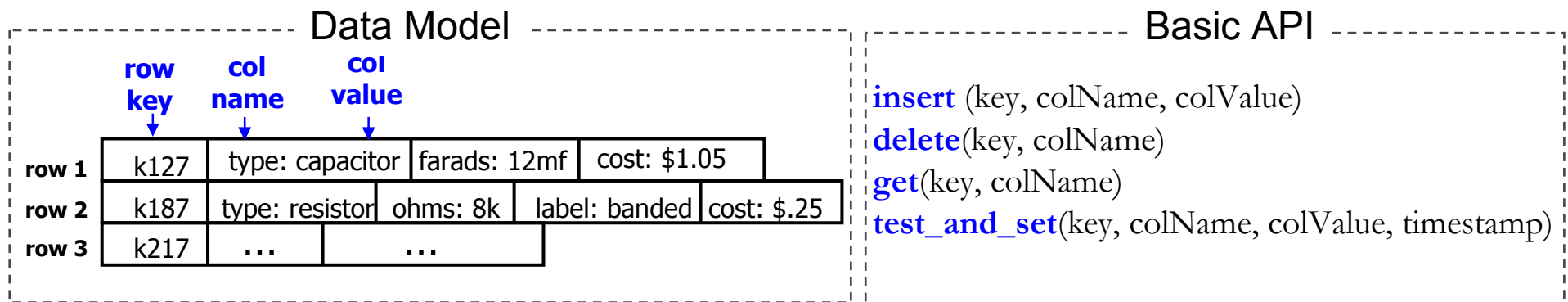
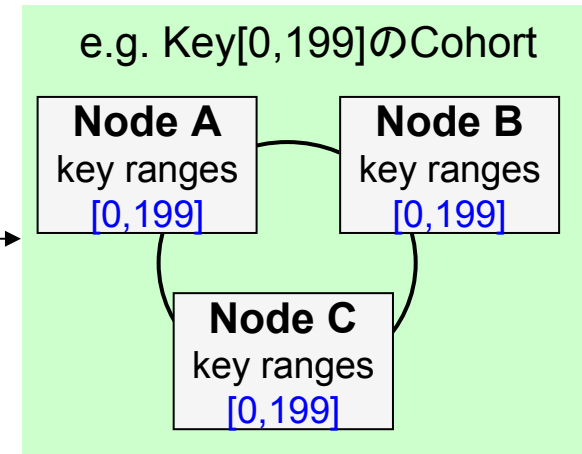
Using Paxos to Build a Scalable, Consistent, and Highly Available Datastore

Jun Rao, Eugene Shekita, Sandeep Tata (IBM Almaden Research Center)



▶ Spinnaker

- ▶ Key-range partitioning
- ▶ Chained declustering
- ▶ The replicas of every partition form a cohort →
- ▶ Multi-Paxos executed within each cohort
- ▶ Timeline consistency (オブジェクト中心一貫性)



Paxos-basedな同期プロトコルについてフォーカスした論文



Multi-Paxos Replication Protocol

- ▶ 分散合意 : Paxos
 - ▶ Quorum1: Propose -> Promise
 - ▶ Quorum2: Accept -> OK

- ▶ Multi-Paxos
 - ▶ Leaderを予め決めておくことにより, 上記Quorum2から開始

Zookeeper

Make implementing a system that uses many instances of consensus much simpler than previously possible

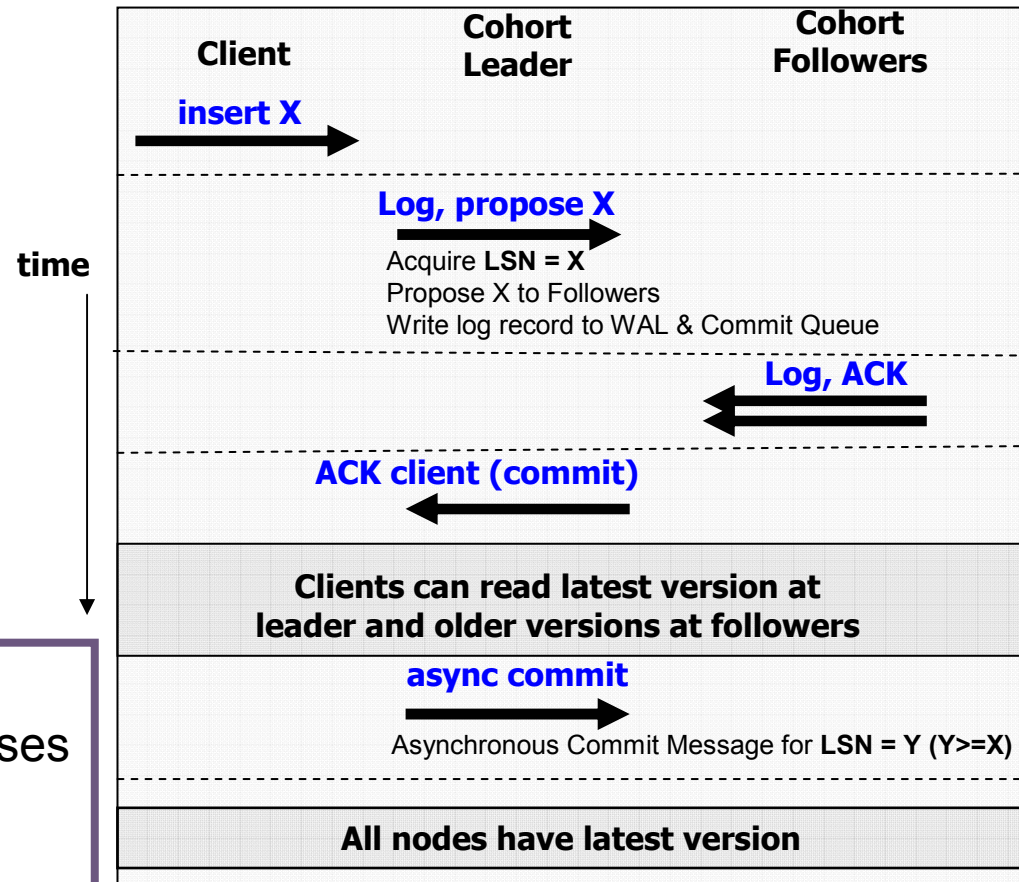


Figure 4: The replication protocol.



類似製品との比較

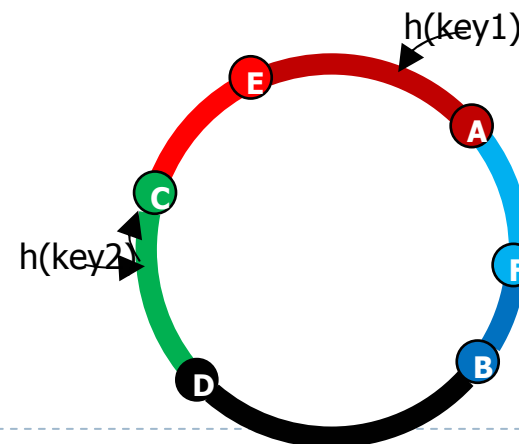
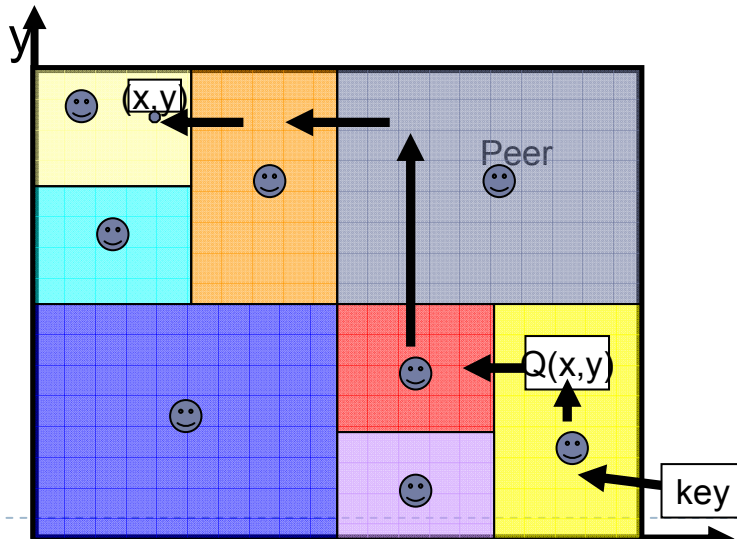
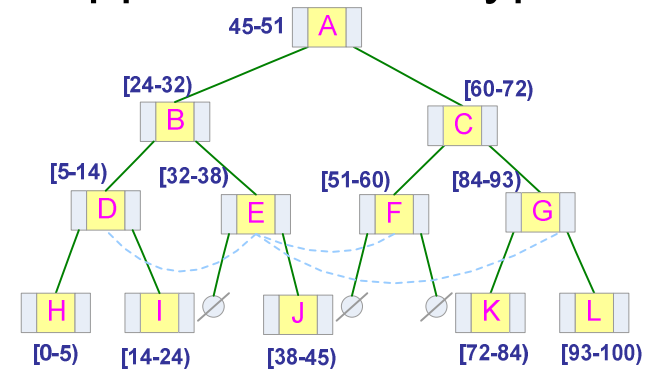
- ▶ vs BigTable (Google)
 - ▶ Strong consistency
 - ▶ LogのGFS(Google File System)への書き込みが遅い
- ▶ vs Dynamo (Amazon)
 - ▶ Eventual consistency
 - ▶ Vector timestampにより, conflictを解消する
- ▶ vs PNUTS (Yahoo)
 - ▶ Timeline consistency
 - ▶ Cross-datacenter向け

**Spinnaker can be used for replication with *good* performance
10% slower writes, faster reads compared to Cassandra**

A Framework for Supporting DBMS-like Indexes in the Cloud

Gang Chen, Hoang Tam Vo, Sai Wu, Beng Chin Ooi, M. Tamer Özsu

- ▶ クラウド上にScalableでDBMS-likeな分散Indexを生成
- ▶ Different overlays are required to support different types of indexes
 - ▶ BATON for B-tree
 - ▶ CAN for R-tree
 - ▶ Chord for Hash

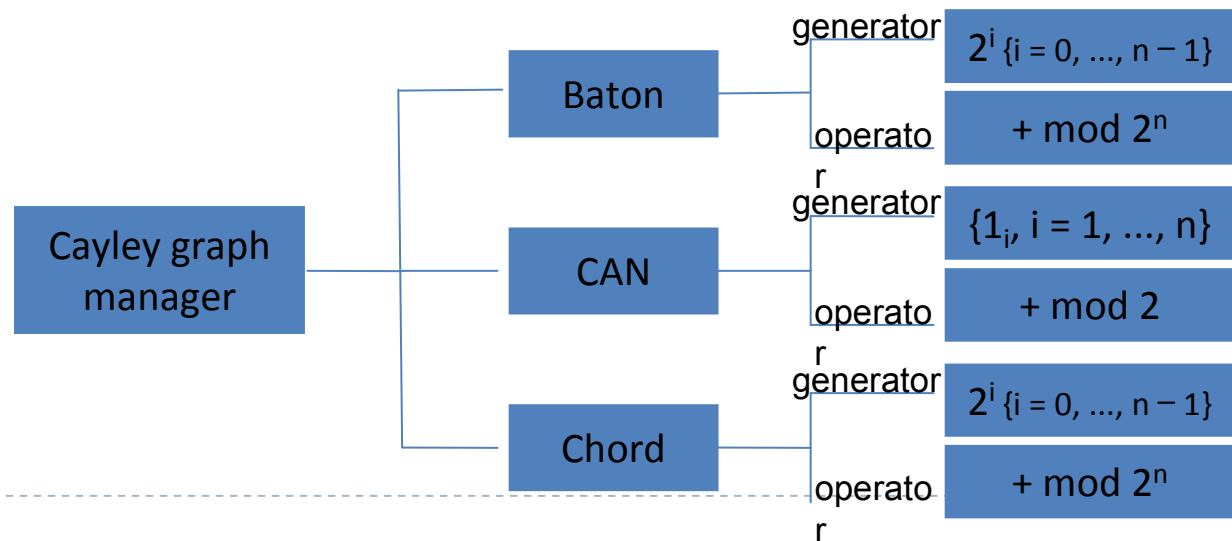


Overlay Mapping

- ▶ Two interfaces for mapping a specific type of overlay to Cayley graph
 - ▶ Generating set
 - ▶ Operator

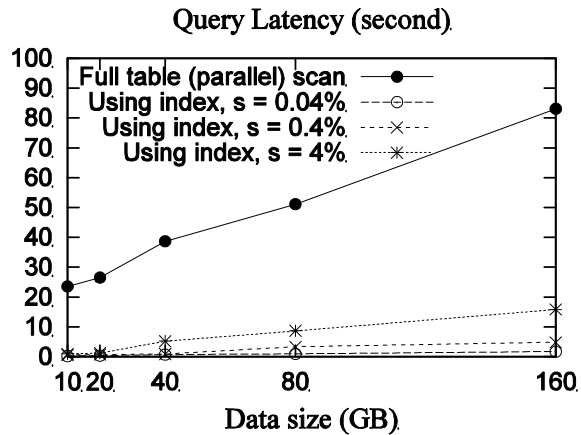
DEFINITION 1. *Cayley Graph* : A Cayley graph $\mathcal{G} = (S, G, \oplus)$, where S is an element set, G is a generator set and \oplus is a binary operator, is a graph such that

1. $\forall e \in S$, there is a vertex in \mathcal{G} corresponding to e .
2. $\oplus : (S \times G) \rightarrow S$.
3. $\forall e \in S$ and $\forall g \in G$, $e \oplus g$ is an element in S and there is an edge from e to $e \oplus g$ in \mathcal{G} .
4. There is no loop in \mathcal{G} , namely $\forall e \in S, \forall g \in G \rightarrow e \oplus g \neq e$.

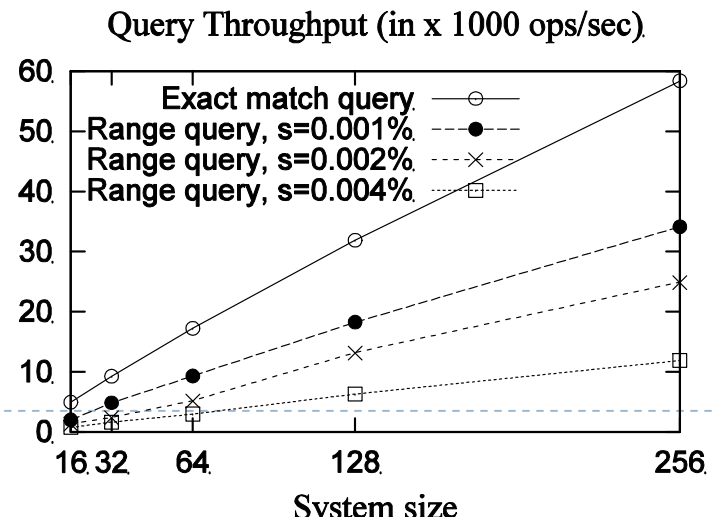
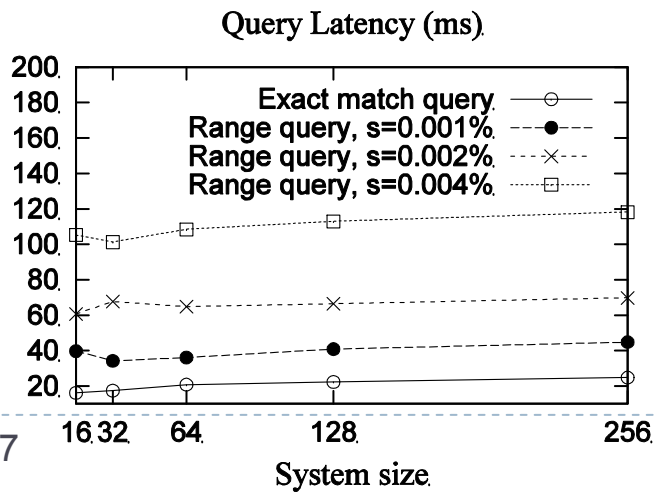


Performance and Scalability

Index plan vs. full table (parallel) scan



Scalability on EC2



Serializable Snapshot Isolation for Replicated Databases in High-Update Scenarios

Hyungsoo Jung, Hyuck Han, Alan Fekete, Uwe Röhm

(The University of Sydney, Seoul National University)

▶ New algorithm for 1-copy Serializable

- ▶ 1 SR over SI replicas
- ▶ Update anywhere-anytime-anyway transactional replication
- ▶ Strong Consistency
- ▶ Optimized for update heavy

Snapshot Isolation [Berenson et al., SIGMOD'95]

各トランザクションは開始時にsnapshotを取得し、開始以前にcommitされたデータのみ読める

DBMSはReadロックを回避するため複数のバージョンを保持しており、トランザクションCommit時に競合(write-write)が検出された場合はトランザクションをロールバック

Replicated Snapshot DBでSerializationを保証したい

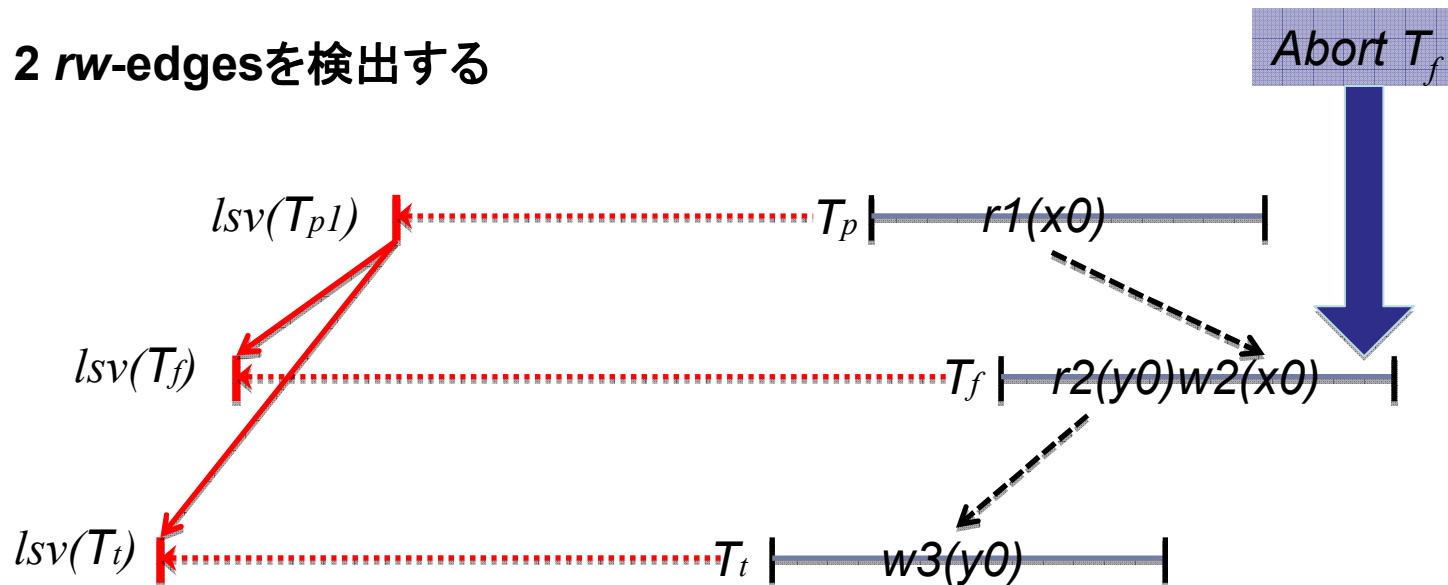
Descending Structure

【Definition 3.1】

Three transactions T_p , T_f and T_t with the following relationships:

1. $T_p \xrightarrow{rw} T_f$ and $T_f \xrightarrow{rw} T_t$
2. $lsv(T_f) \preceq lsv(T_p)$ && $lsv(T_t) \preceq lsv(T_p)$

2 *rw*-edgesを検出する

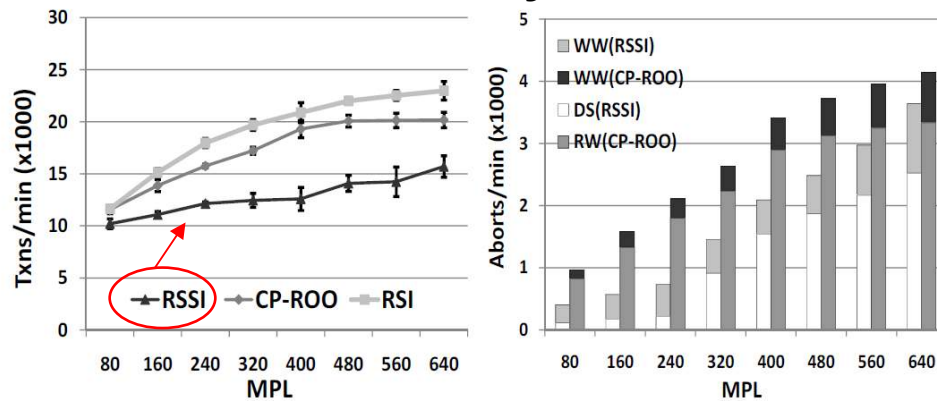


著者スライド図より抜粋

Experiments

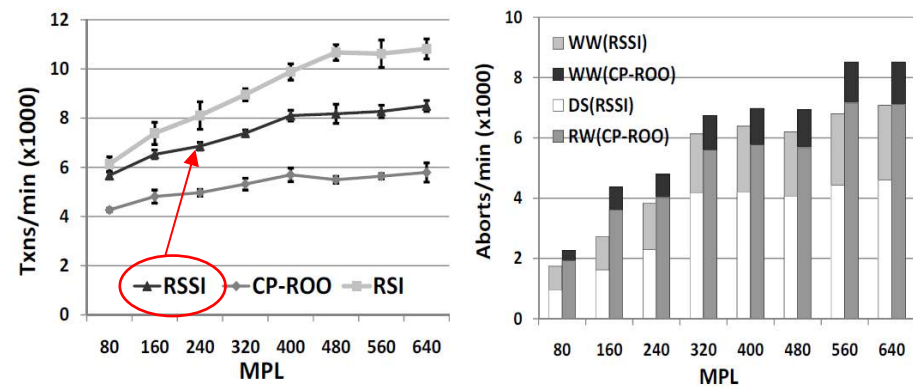
	Bornea et al. (CP-ROO)	This Work (RSSI)
Architecture	Middleware	Kernel
Readset Extraction	SQL parsing	Kernel interception
Certification	<i>ww</i> -conflict 1 <i>rw</i> -edge	<i>ww</i> -conflict 2 <i>rw</i> -edges
Optimized for	Read mostly	Update heavy

Read mostly Scenario



(d) 75%RO-25%W

Write mostly Scenario



(b) 25%RO-75%W