



Positional Update Handling in Column Stores

Sandor Heman, Marcin Zukowski, Niels Nes,
Lefteris Sidirourgos, Peter Boncz

Presented by Guozhang Wang

2013.2.4

DB Lunch

Row Store v.s. Column Store



- Row store
 - Unnecessary read I/O
 - Not good for OLAP



- Column store
 - Save read I/O
 - Enable compression / data ordering

What about updates?

- Requires n random I/Os for one tuple if ***update-in-place***
- Compression makes things worse
- Replication and data ordering makes thing even worse



Differential Updates

- Collect updates in a write store (WS)
 - How to handle updates efficiently?
- Periodically propagate to read store (RS)
 - How to propagate quickly?
- Access both RS and WS for reads
 - How to handle reads efficiently?

LSM-Tree for Indexing

- (2-3) tree or AVL tree for WS in RAM
 - Maintain fast search
- Hierarchy of trees
 - Keep merging efficient
- B+ tree for lower-level trees for seq. I/O



Array for Key-Value Store

- Sorted key/values in memory table



- Merging is simplified as disk flush
- Periodically compact tables on disk
- Bloom filter used to improve searching

VD for Column Stores



- Columns in RS sorted by sort keys (SK)
- ***Value-based delta (VD)*** entries in WS ordered by SKs of the RS columns
- Queries gets answered by scanning both the RS and the delta entries in WS



An Example of VD

SELECT *
FROM *inventory*

Read Store: *inventory*
Sort-Key (SK): [*store*, *prod*]

store	prod	new	qty
London	stool	N	10
London	table	N	20
Paris	rug	N	1
Paris	stool	N	5

Write Store: *INS*

store	prod	new	qty
Berlin	chair	Y	5
Berlin	cloth	Y	20

Write Store: *DEL*

store	prod
Paris	rug

An Example of VD

```
SELECT * FROM ins
UNION
(SELECT * FROM inventory
WHERE NOT EXISTS
(SELECT * FROM del
WHERE inventory.store =
del.store AND
inventory.prod = del.prod))
```

Read Store: *inventory*
Sort-Key (SK): [*store, prod*]

store	prod	new	qty
London	stool	N	10
London	table	N	20
Paris	rug	N	1
Paris	stool	N	5

Write Store: *INS*

store	prod	new	qty
Berlin	chair	Y	5
Berlin	cloth	Y	20

Write Store: *DEL*

store	prod
Paris	rug

An Example of VD

SELECT * FROM *ins*
UNION

(SELECT * FROM *inventory*
WHERE NOT EXISTS

(SELECT * FROM *del*
WHERE *inventory.store* =
del.store AND
inventory.prod = *del.prod*))

Read Store: *inventory*

Sort-Key (SK): [*store, prod*]

store	prod	new	qty
London	stool	N	10
London	table	N	20
Paris	rug	N	1
Paris	stool	N	5

Write Store: *INS*

store	prod	new	qty
Berlin	chair	Y	5
Berlin	cloth	Y	20

Write Store: *DEL*

store	prod
Paris	rug

An Example of VD

SELECT * FROM *ins*
UNION

(SELECT * FROM *inventory*
WHERE NOT EXISTS

(SELECT * FROM *del*
WHERE *inventory.store* =
del.store AND
inventory.prod = *del.prod*))

Read Store: *inventory*

Sort-Key (SK): [*store, prod*]

store	prod	new	qty
London	stool	N	10
London	table	N	20
Paris	rug	N	1
Paris	stool	N	5

Write Store: *INS*

store	prod	new	qty
Berlin	chair	Y	5
Berlin	cloth	Y	20

Write Store: *DEL*

store	prod
Paris	rug

An Example of VD

SELECT * FROM *ins*
UNION

(SELECT * FROM *inventory*
WHERE NOT EXISTS

(SELECT * FROM *del*
WHERE *inventory.store* =
***del.store* AND**
***inventory.prod* = *del.prod*))**

Read Store: *inventory*

Sort-Key (SK): [*store, prod*]

store	prod	new	qty
London	stool	N	10
London	table	N	20
Paris	rug	N	1
Paris	stool	N	5

Write Store: *INS*

store	prod	new	qty
Berlin	chair	Y	5
Berlin	cloth	Y	20

Write Store: *DEL*

store	prod
Paris	rug

Problems of VD

- Expensive I/O scan of SK columns in RS to merge with WS
 - No matter if the query needs SK columns
- Key-based union and diff operations are computationally intensive



The Idea:

- *Remember the **position** of an update in RS rather than its sorted key values*
- *Blindly apply updates according to positions when merging RS and WS*

Tuple Position in RS

- RID: RowID
 - Unique, changeable according to updates
- SID: StableID
 - Non-unique, fixed at tuple creation time
- $RID(t) = SID + \Delta(t)$
 - $\Delta(t) = \#inserts - \#deletes$ *before t*

An Example of RID/SID

<i>SID</i>	store	prod	new	qty	<i>RID</i>
0	London	chair	N	30	0
1	London	stool	N	10	1
2	London	table	N	20	2
3	Paris	rug	N	1	3
4	Paris	stool	N	5	4

INSERT INTO *inventory* VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory* VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory* VALUES('Berlin', 'chair', Y, 5)

An Example of RID/SID

<i>SID</i>	store	prod	new	qty	<i>RID</i>
0	Berlin	chair	Y	5	0
0	Berlin	cloth	Y	20	1
0	Berlin	table	Y	10	2
0	London	chair	N	30	3
1	London	stool	N	10	4
2	London	table	N	20	5
3	Paris	rug	N	1	6
4	Paris	stool	N	5	7

INSERT INTO *inventory* VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory* VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory* VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree for WS

- Keeps track of cumulative deltas (Δ) in a **counting B-Tree**
- Fast insertion of update entries and updating deltas accordingly
- Fast merging updates with underlying RS with the computed $RID = SID + \Delta$

An Example of PDT

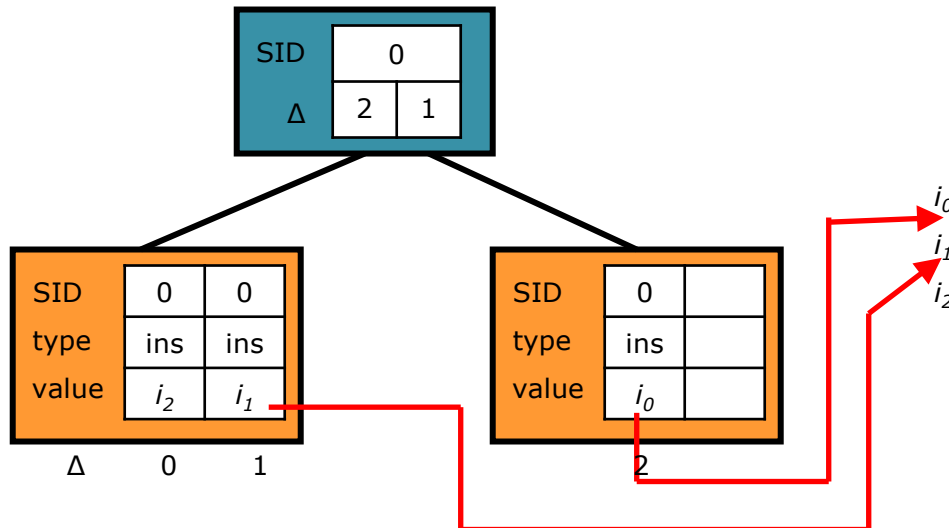
<i>SID</i>	store	prod	new	qty	<i>RID</i>
0	London	chair	N	30	0
1	London	stool	N	10	1
2	London	table	N	20	2
3	Paris	rug	N	1	3
4	Paris	stool	N	5	4

INSERT INTO *inventory*
VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory*
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory*
VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree



Insert Value Table

store	prod	new	qty
Berlin	table	Y	10
Berlin	cloth	Y	5
Berlin	chair	Y	20

An Example of PDT

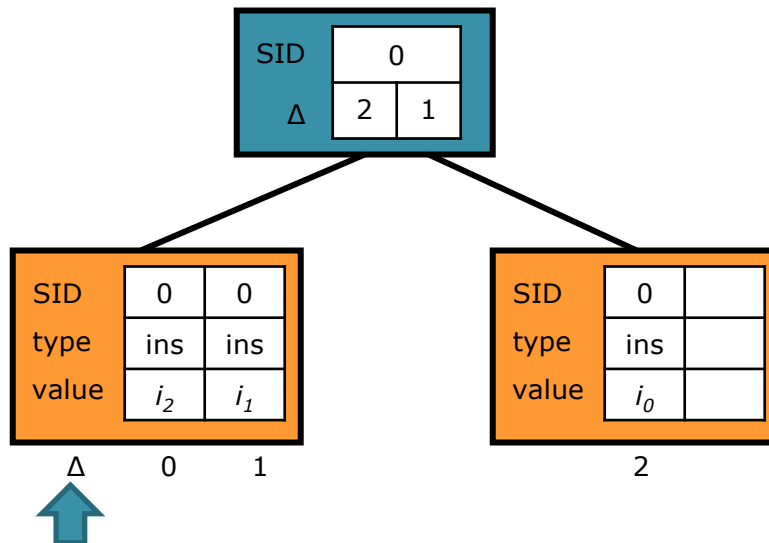
SID	store	prod	new	qty	RID
0	London	chair	N	30	0
1	London	stool	N	10	1
2	London	table	N	20	2
3	Paris	rug	N	1	3
4	Paris	stool	N	5	4

INSERT INTO *inventory*
VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory*
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory*
VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

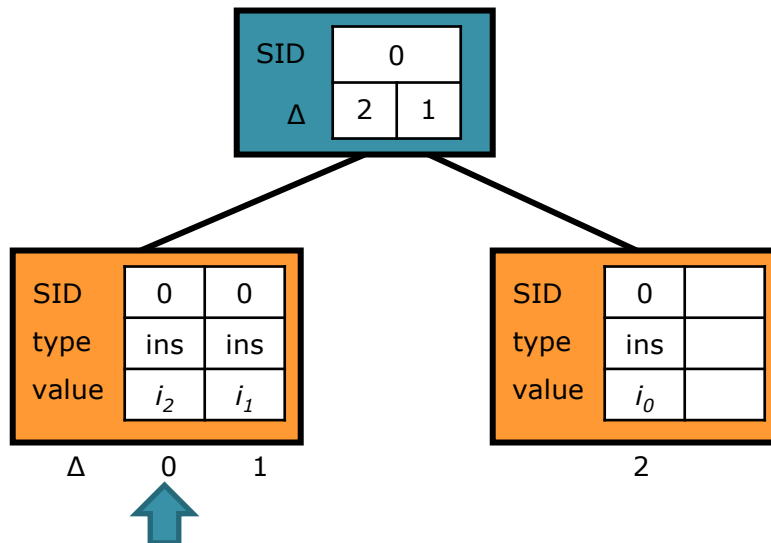
<i>SID</i>	store	prod	new	qty	<i>RID</i>
0	Berlin	cloth	Y	20	0
0	London	chair	N	30	1
1	London	stool	N	10	2
2	London	table	N	20	3
3	Paris	rug	N	1	4
4	Paris	stool	N	5	5

INSERT INTO *inventory*
VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory*
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory*
VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

<i>SID</i>	store	prod	new	qty	<i>RID</i>
0	Berlin	cloth	Y	20	0
0	Berlin	chair	Y	5	1
0	London	chair	N	30	2
1	London	stool	N	10	3
2	London	table	N	20	4
3	Paris	rug	N	1	5
4	Paris	stool	N	5	6

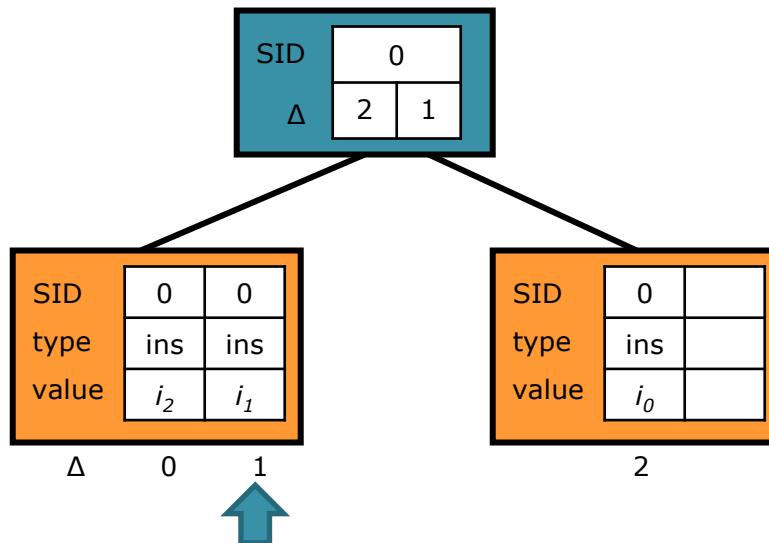
```

INSERT INTO inventory
VALUES('Berlin', 'table', Y, 10)

INSERT INTO inventory
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO inventory
VALUES('Berlin', 'chair', Y, 5)
    
```

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

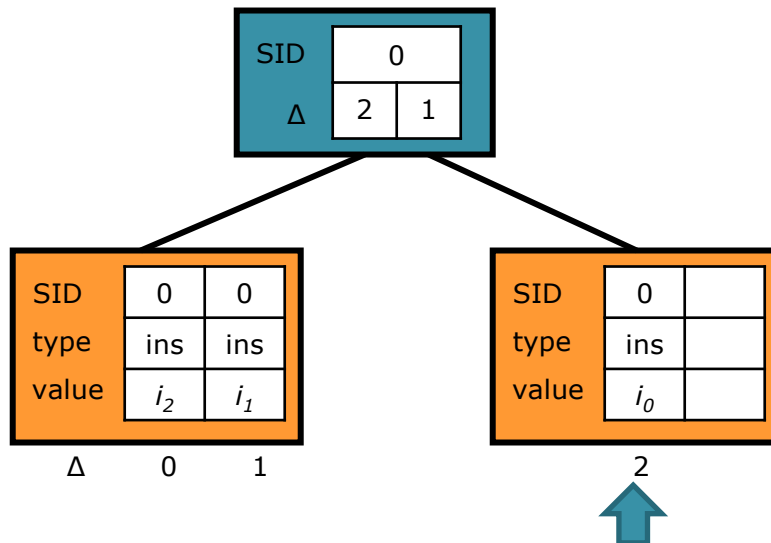
An Example of PDT

<i>SID</i>	store	prod	new	qty	<i>RID</i>
0	Berlin	cloth	Y	20	0
0	Berlin	chair	Y	5	1
0	Berlin	table	Y	10	2
0	London	chair	N	30	3
1	London	stool	N	10	4
2	London	table	N	20	5
3	Paris	rug	N	1	6
4	Paris	stool	N	5	7

```

INSERT INTO inventory
VALUES('Berlin', 'table', Y, 10)
INSERT INTO inventory
VALUES('Berlin', 'cloth', Y, 20)
INSERT INTO inventory
VALUES('Berlin', 'chair', Y, 5)
    
```

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

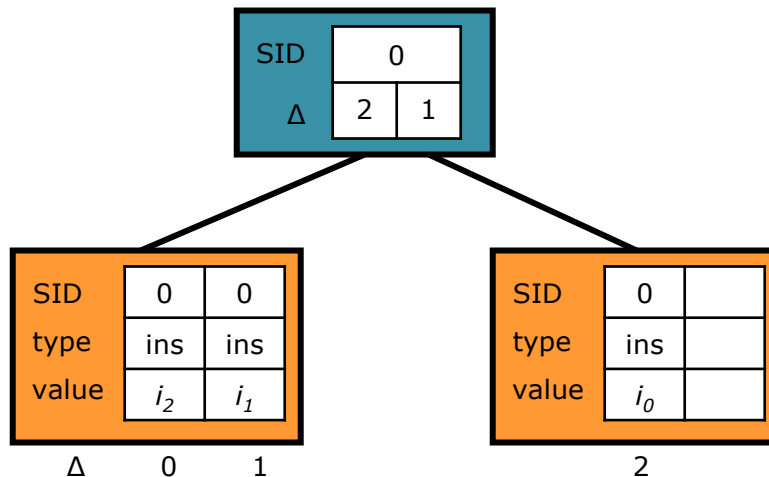
SID	store	prod	new	qty	RID
0	Berlin	cloth	Y	20	0
0	Berlin	chair	Y	5	1
0	Berlin	table	Y	10	2
→ 0	London	chair	N	30	3
1	London	stool	N	10	4
2	London	table	N	20	5
3	Paris	rug	N	1	6
4	Paris	stool	N	5	7

INSERT INTO *inventory*
VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory*
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory*
VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

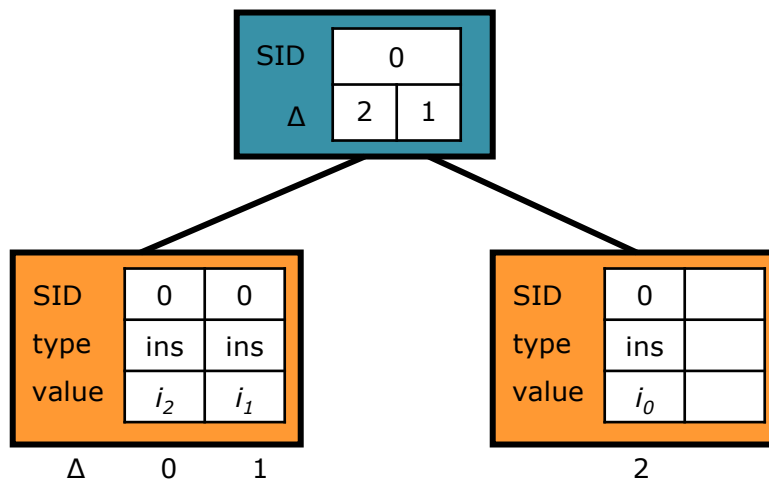
<i>SID</i>	store	prod	new	qty	<i>RID</i>
0	Berlin	cloth	Y	20	0
0	Berlin	chair	Y	5	1
0	Berlin	table	Y	10	2
0	London	chair	N	30	3
1	London	stool	N	10	4
2	London	table	N	20	5
3	Paris	rug	N	1	6
4	Paris	stool	N	5	7

INSERT INTO *inventory*
VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory*
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory*
VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

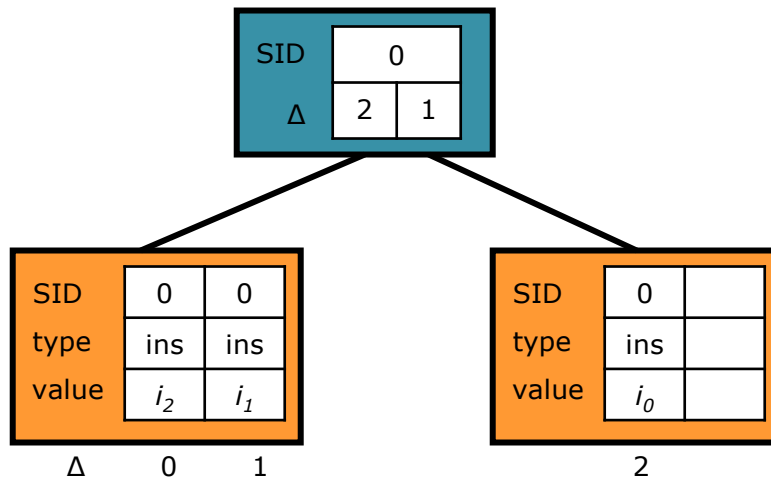
SID	store	prod	new	qty	RID
0	Berlin	cloth	Y	20	0
0	Berlin	chair	Y	5	1
0	Berlin	table	Y	10	2
0	London	chair	N	30	3
1	London	stool	N	10	4
2	London	table	N	20	5
3	Paris	rug	N	1	6
4	Paris	stool	N	5	7

INSERT INTO *inventory*
VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory*
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory*
VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

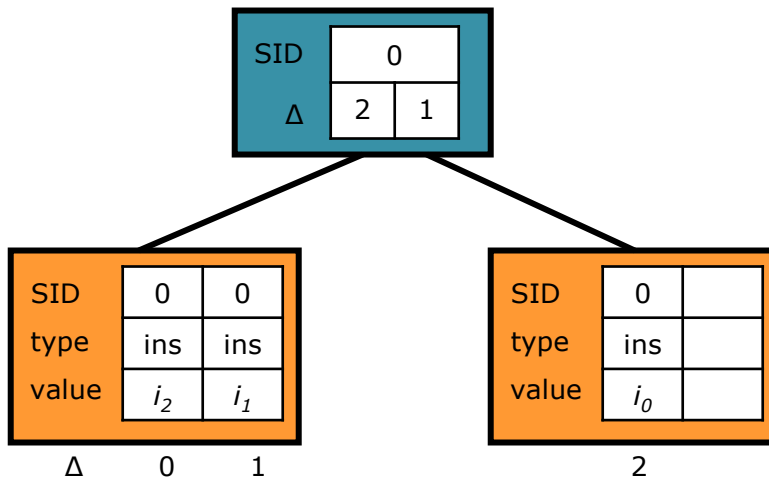
SID	store	prod	new	qty	RID
0	Berlin	cloth	Y	20	0
0	Berlin	chair	Y	5	1
0	Berlin	table	Y	10	2
0	London	chair	N	30	3
1	London	stool	N	10	4
2	London	table	N	20	5
3	Paris	rug	N	1	6
4	Paris	stool	N	5	7

INSERT INTO *inventory*
VALUES('Berlin', 'table', Y, 10)

INSERT INTO *inventory*
VALUES('Berlin', 'cloth', Y, 20)

INSERT INTO *inventory*
VALUES('Berlin', 'chair', Y, 5)

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

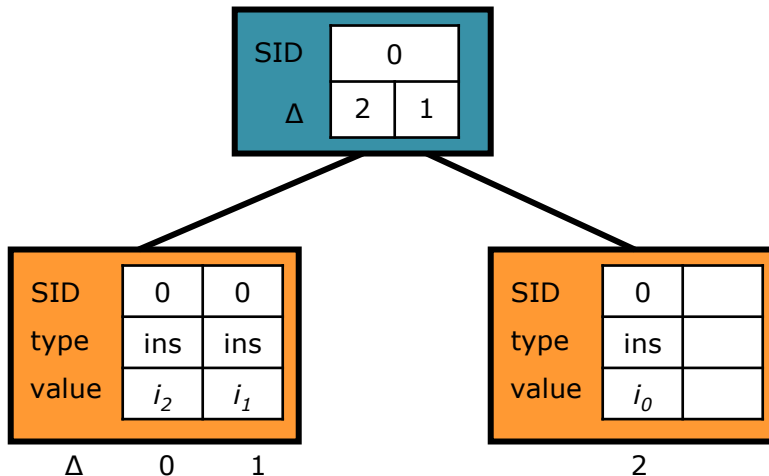
An Example of PDT

SID	store	prod	new	qty	RID
0	Berlin	cloth	Y	20	0
0	Berlin	chair	Y	5	1
0	Berlin	table	Y	10	2
0	London	chair	N	30	3
1	London	stool	N	10	4
2	London	table	N	20	5
3	Paris	rug	N	1	6
4	Paris	stool	N	5	7

```

INSERT INTO inventory
VALUES('Berlin', 'table', Y, 10)
INSERT INTO inventory
VALUES('Berlin', 'cloth', Y, 20)
INSERT INTO inventory
VALUES('Berlin', 'chair', Y, 5)
    
```

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

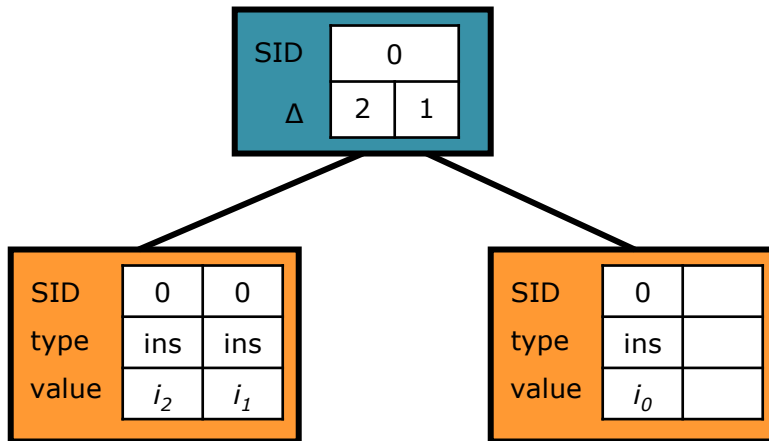
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

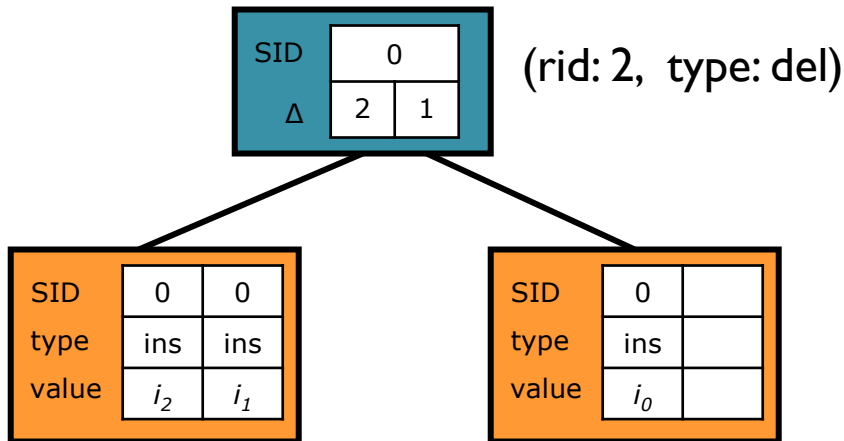
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

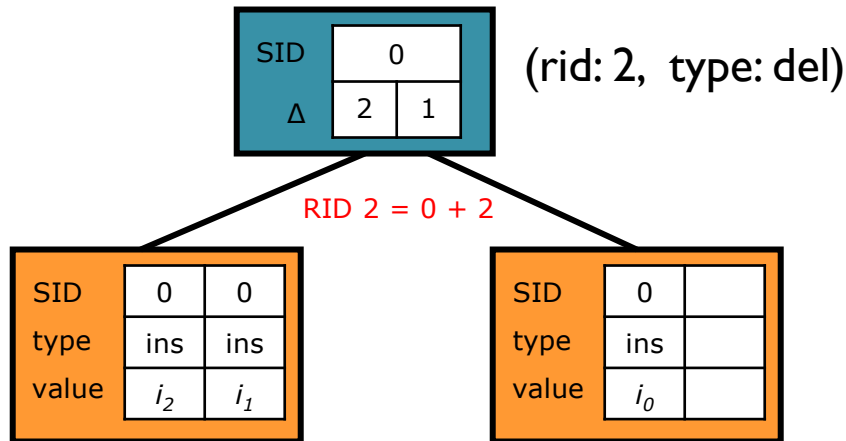
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

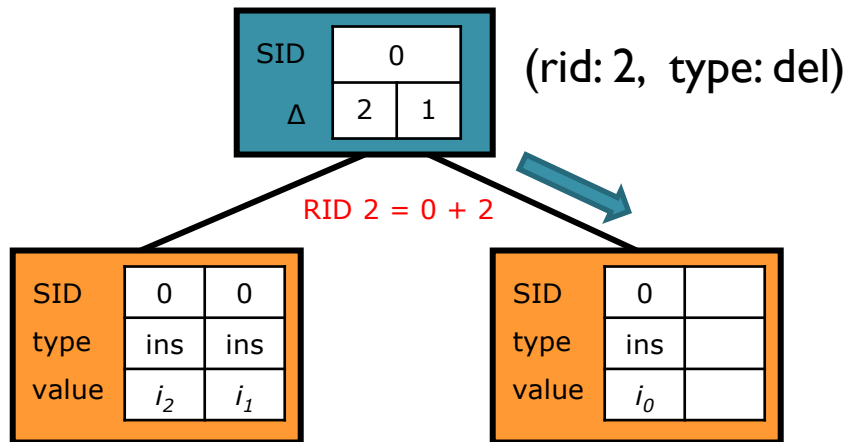
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Berlin	table	Y	10
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

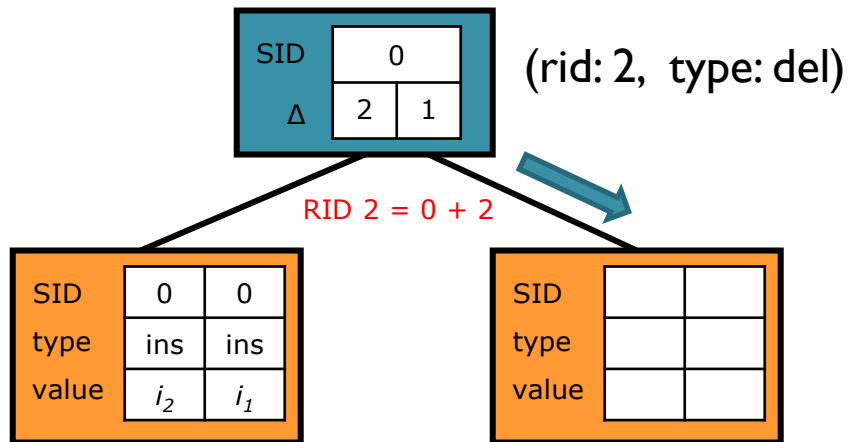
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	cloth	Y	5
i_2	Berlin	chair	Y	20

An Example of PDT

<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

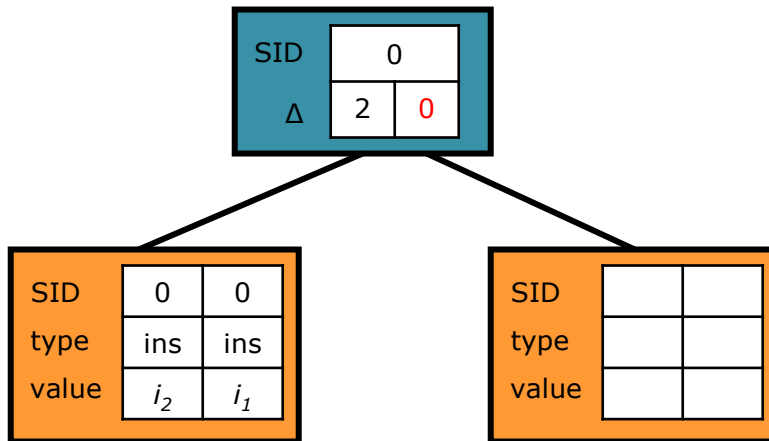
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

An Example of PDT

<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

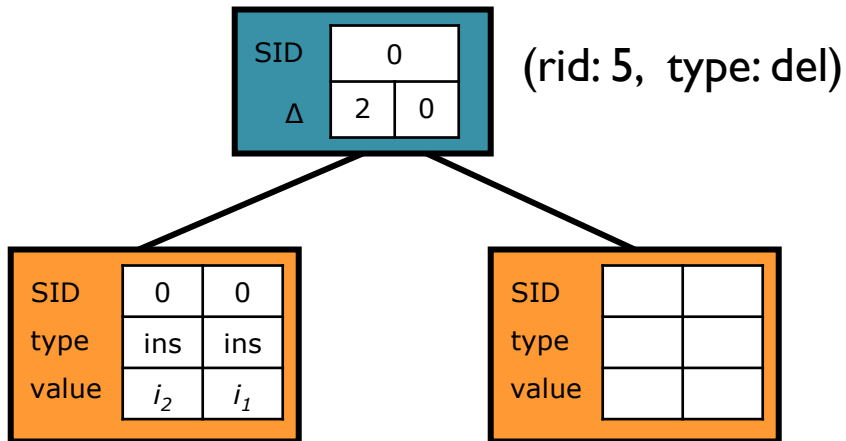
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

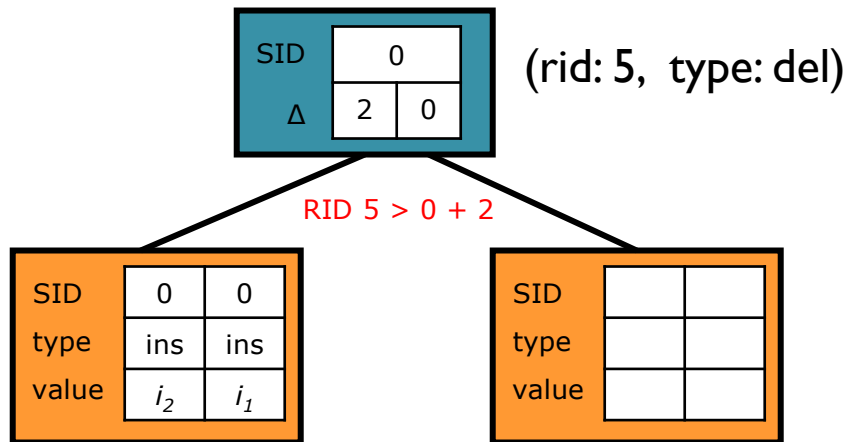
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

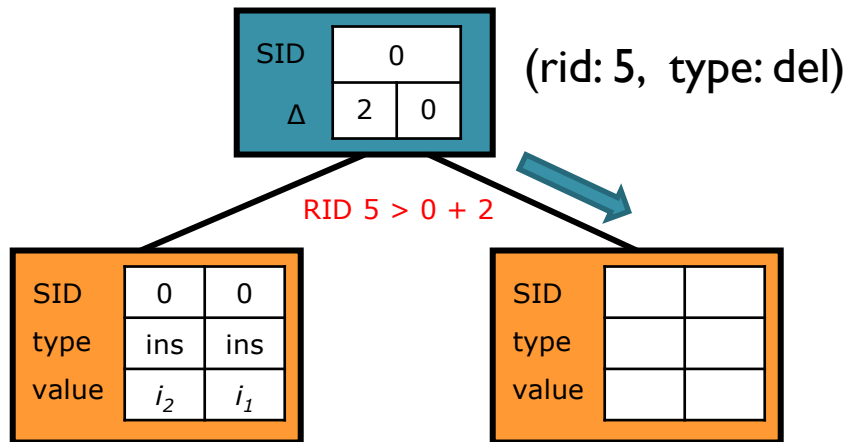
RID

0
1
2
3
4
5
6
7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

An Example of PDT

SID	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

RID

0

1

2

3

4

5

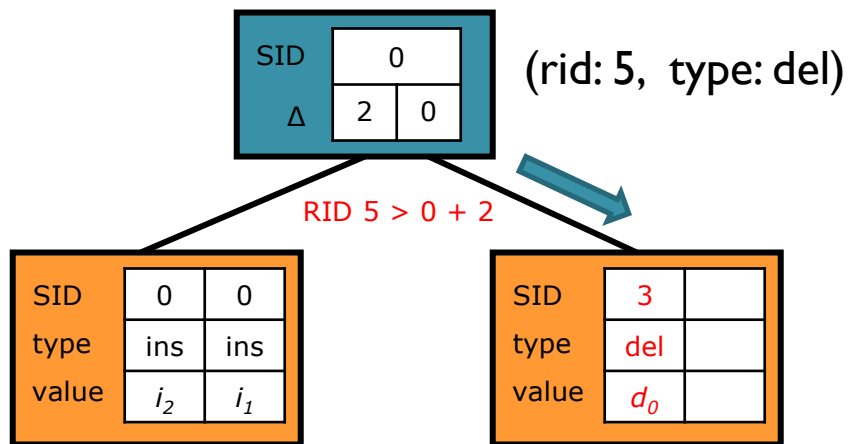
6

7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



$$SID(t) = RID - \Delta(t)$$

Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

Delete Value Table

	store	prod
d_0	Paris	rug

An Example of PDT

<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	Berlin	table	Y	10
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
3	Paris	rug	N	1
4	Paris	stool	N	5

RID

0

1

2

3

4

5

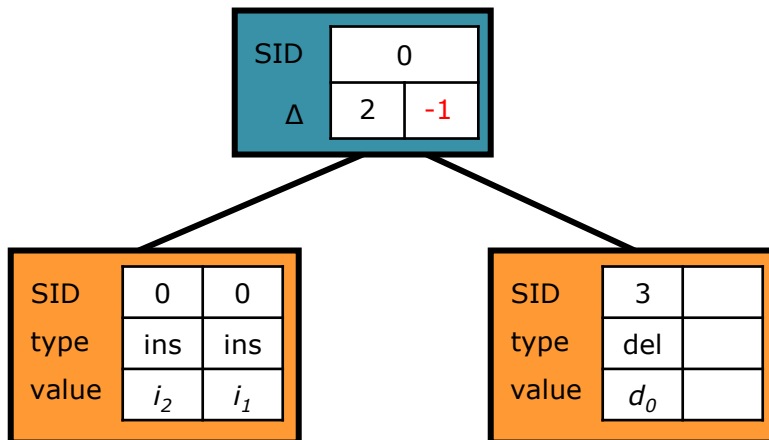
6

7

DELETE FROM *inventory*
WHERE store = 'Berlin'
AND prod = 'table'

DELETE FROM *inventory*
WHERE store = 'Paris'
AND prod = 'rug'

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

Delete Value Table

	store	prod
d_0	Paris	rug

An Example of PDT

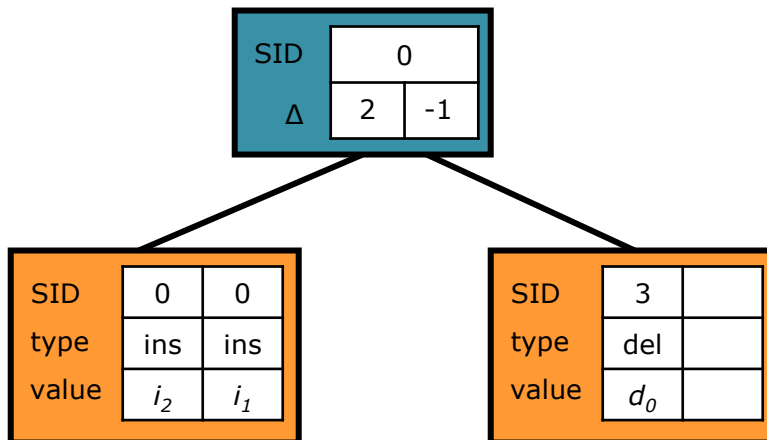
<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
4	Paris	stool	N	5

RID
0
1
2
3
4
5

INSERT INTO *inventory*
VALUES('Paris', 'rack', Y, 4)



Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

Delete Value Table

	store	prod
d_0	Paris	rug

An Example of PDT

<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
4	Paris	stool	N	5

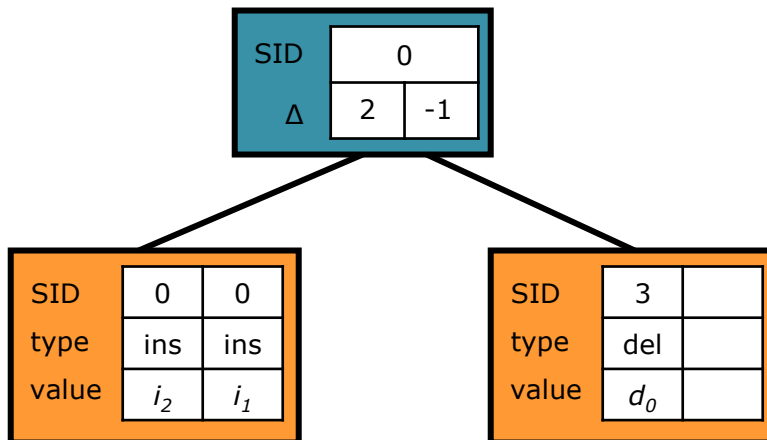
RID

0
1
2
3
4
5

INSERT INTO *inventory*
VALUES('Paris', 'rug', Y, 4)

Insert before RID = 5

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

Delete Value Table

	store	prod
d_0	Paris	rug

An Example of PDT

<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
4	Paris	stool	N	5

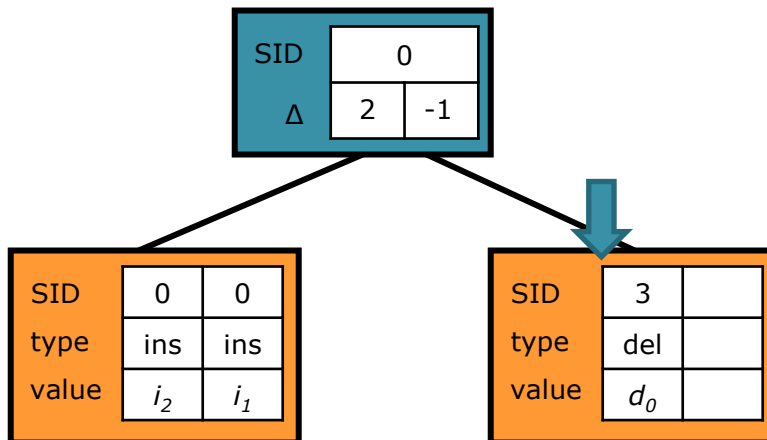
RID

0
1
2
3
4
5

INSERT INTO *inventory*
VALUES('Paris', 'rack', Y, 4)

Insert before RID = 5,
which is at SID = 3

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

Delete Value Table

	store	prod
d_0	Paris	rug

An Example of PDT

<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
4	Paris	stool	N	5

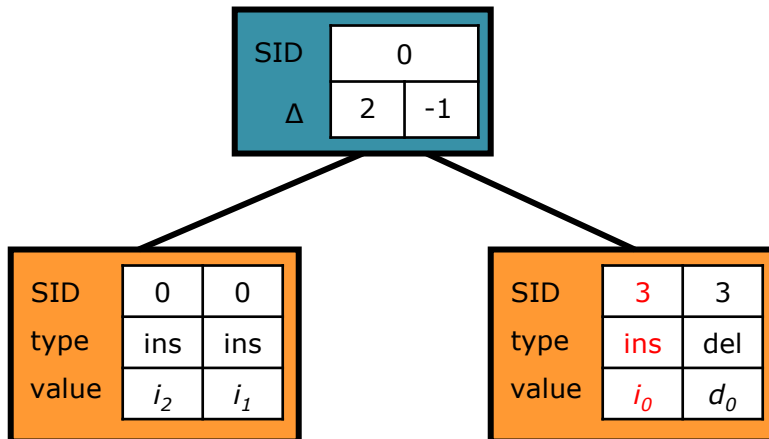
RID

0
1
2
3
4
5

INSERT INTO *inventory*
VALUES('Paris', 'rack', Y, 4)

Insert before RID = 5,
which is at SID = 3

Positional Delta Tree



Insert Value Table

	store	prod	new	qty
i_0	Paris	rack	Y	4
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

Delete Value Table

	store	prod
d_0	Paris	rug

An Example of PDT

<i>SID</i>	store	prod	new	qty
0	Berlin	chair	Y	20
0	Berlin	cloth	Y	5
0	London	chair	N	30
1	London	stool	N	10
2	London	table	N	20
4	Paris	stool	N	5

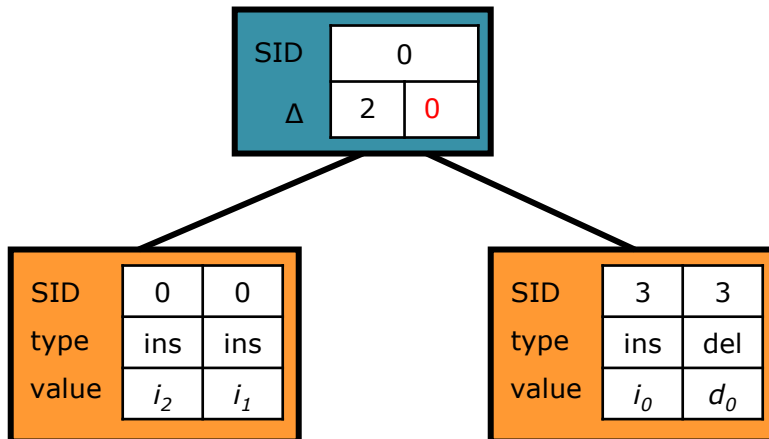
RID

0
1
2
3
4
5

INSERT INTO *inventory*
VALUES('Paris', 'rack', Y, 4)

Insert before RID = 5,
which is at SID = 3

Positional Delta Tree



Insert Value Table

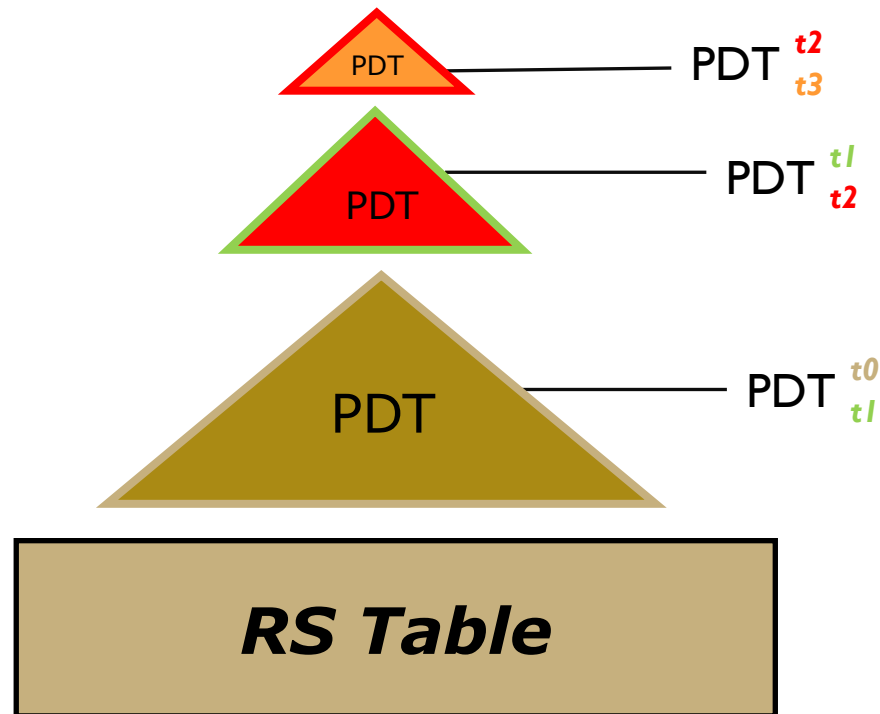
	store	prod	new	qty
i_0	Paris	rack	Y	4
i_1	Berlin	chair	Y	5
i_2	Berlin	cloth	Y	20

Delete Value Table

	store	prod
d_0	Paris	rug

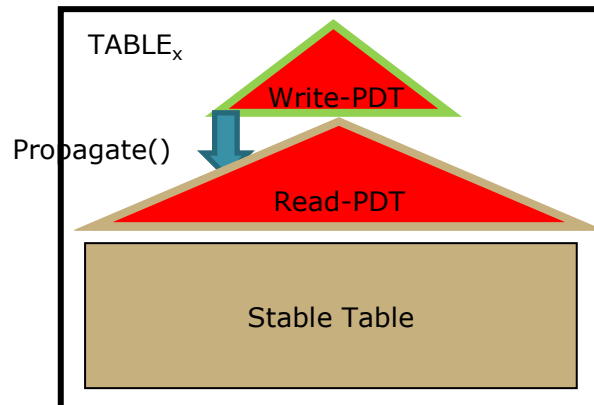
Stacking PDTs

- Layers of PDTs: “deltas on deltas on ..”
 - PDT_{hi}^{lo} contains all differences in time $[lo, hi]$

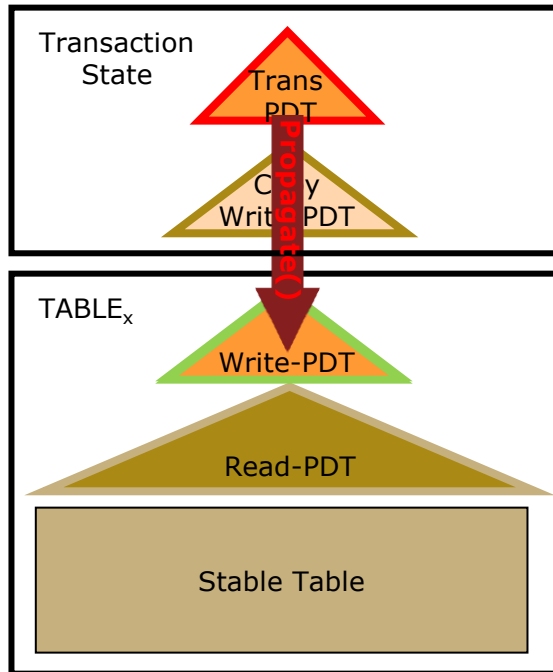


Stacking for Isolation

- Immutable **read-PDT** → *BIG: main memory*
- Updateable **write-PDT** → *SMALL: L2 cache*
 - Periodically propagate changes in write-PDT to read-PDT

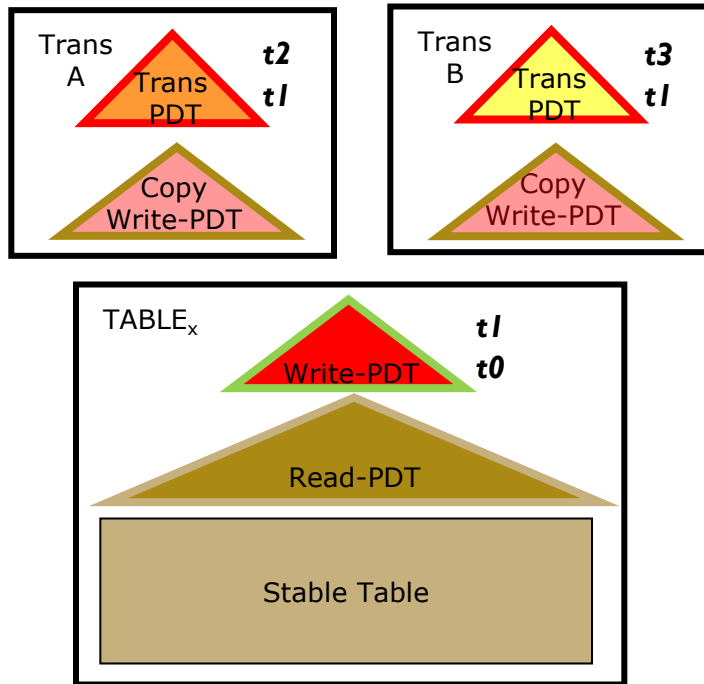


Stacking for Isolation



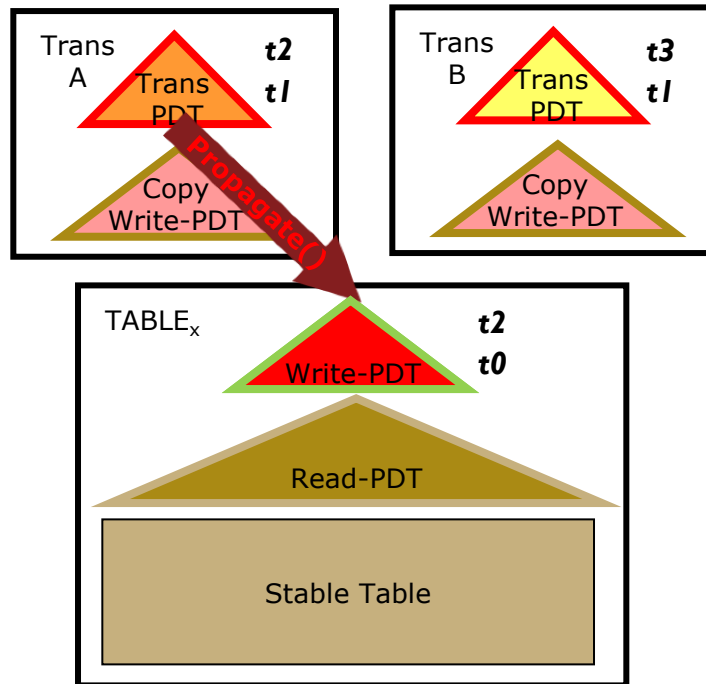
- Transaction creates a snapshot copy of the write-PDT at startup
- Updates go into *trans-PDT*
- On commit, propagate trans-PDT into write-PDT
 - May not succeed if write-PDT has already been updated

Optimistic Concurrency Control



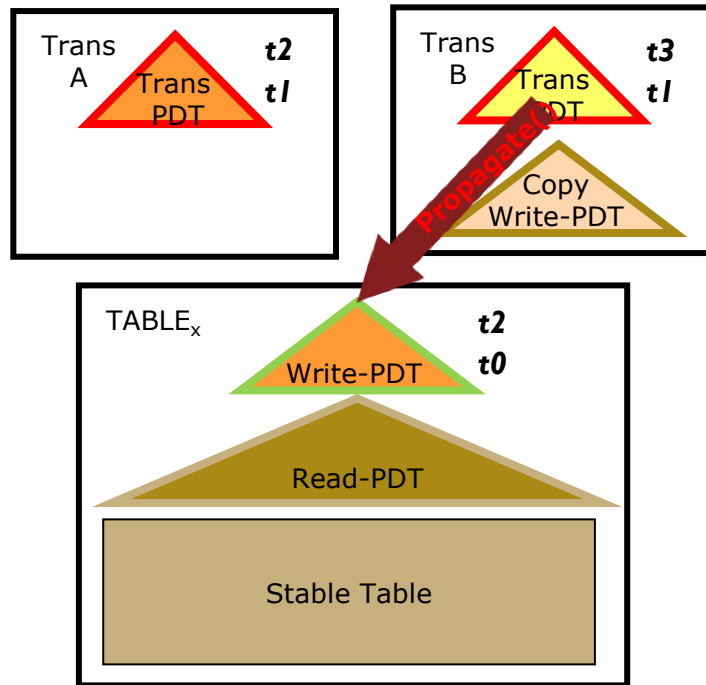
- Two concurrent transactions

Optimistic Concurrency Control



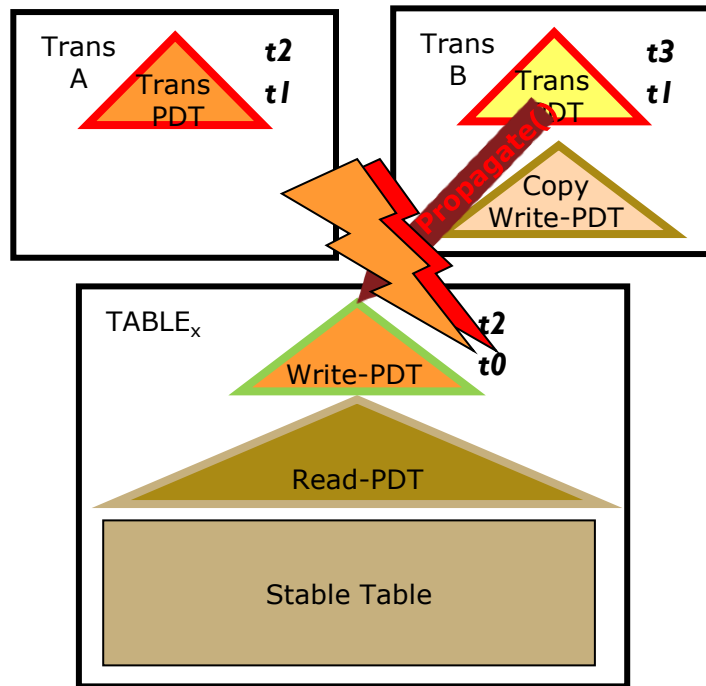
- Two concurrent transactions
- A commits before B

Optimistic Concurrency Control



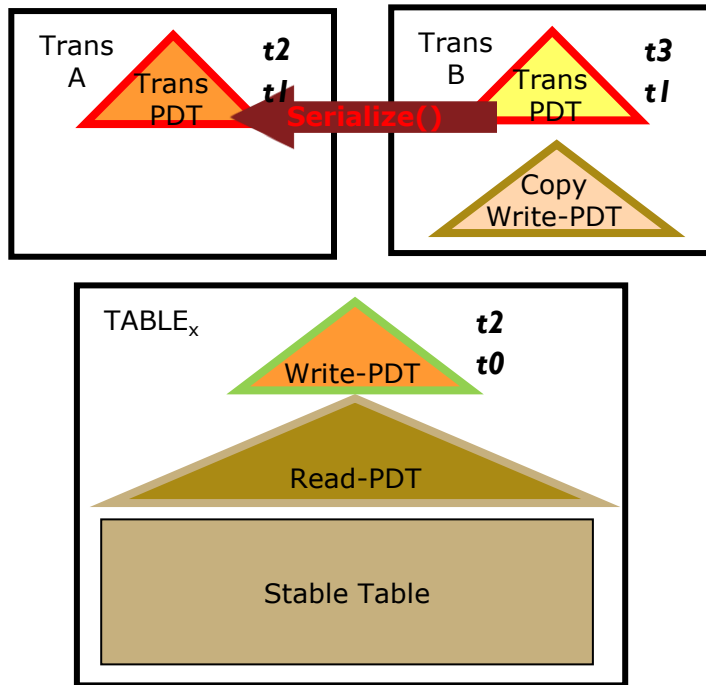
- Two concurrent transactions
- A commits before B
- B wants to commit, tries to propagate its trans-PDT

Optimistic Concurrency Control



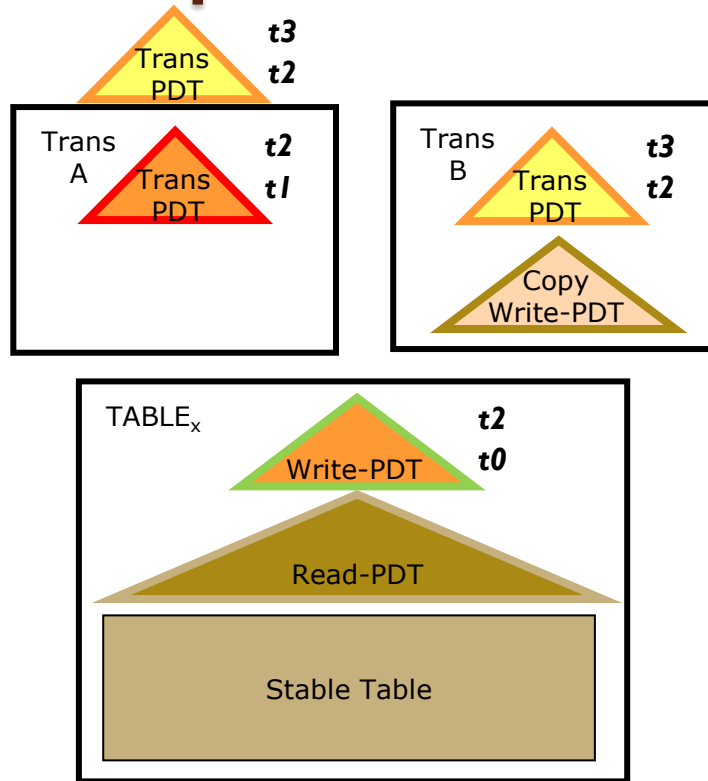
- If conflict detected when enumerating RIDs
 - Cannot commit B since write-set overlaps, abort

Optimistic Concurrency Control



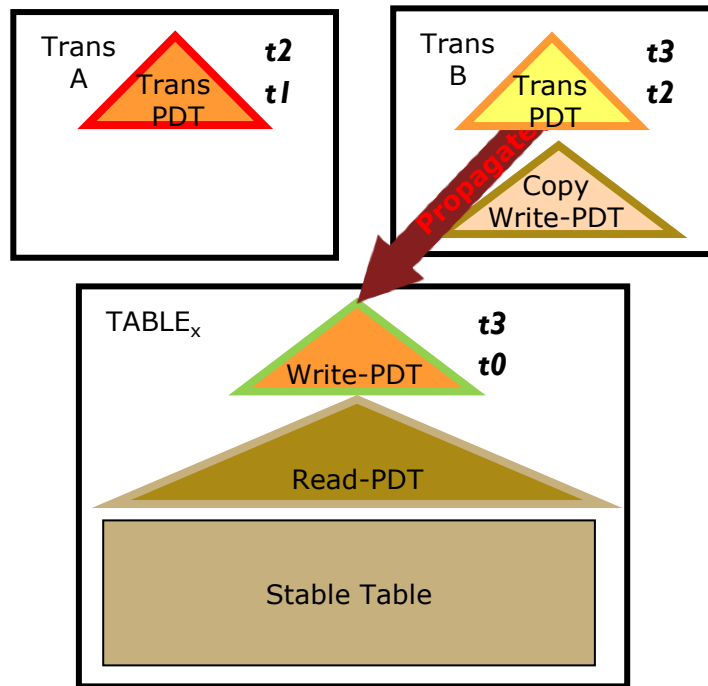
- If conflict detected when enumerating RIDs
 - Cannot commit B since write-set overlaps, abort
- Otherwise B can commit
 - Generate a new PDT aligned with A's PDT

Optimistic Concurrency Control



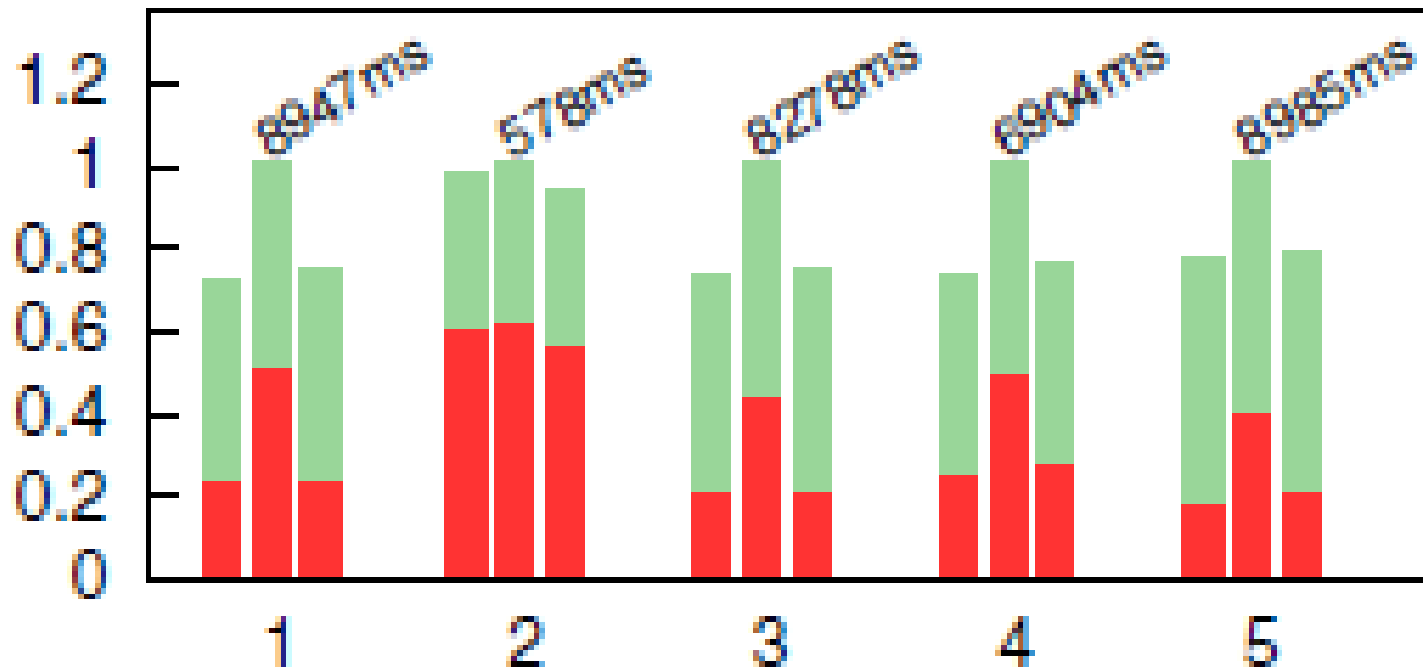
- If conflict detected when enumerating RIDs
 - Cannot commit B since write-set overlaps, abort
- Otherwise B can commit
 - Generate a new PDT aligned with A's PDT

Optimistic Concurrency Control



- If conflict detected when enumerating RIDs
 - Cannot commit B since write-set overlaps, abort
- Otherwise B can commit
 - Generate a new PDT aligned with A's PDT
 - Propagate the new PDT

Evaluation



- 48GB RAM, 3GBs I/O performance
- PTC-H Benchmark

Conclusion

Thank you

- PDTs speed-up differential update merging
 - Reduced I/O volume
 - Reduced CPU merge overhead
- PDTs are stackable
 - Suitable for Snapshot Isolation