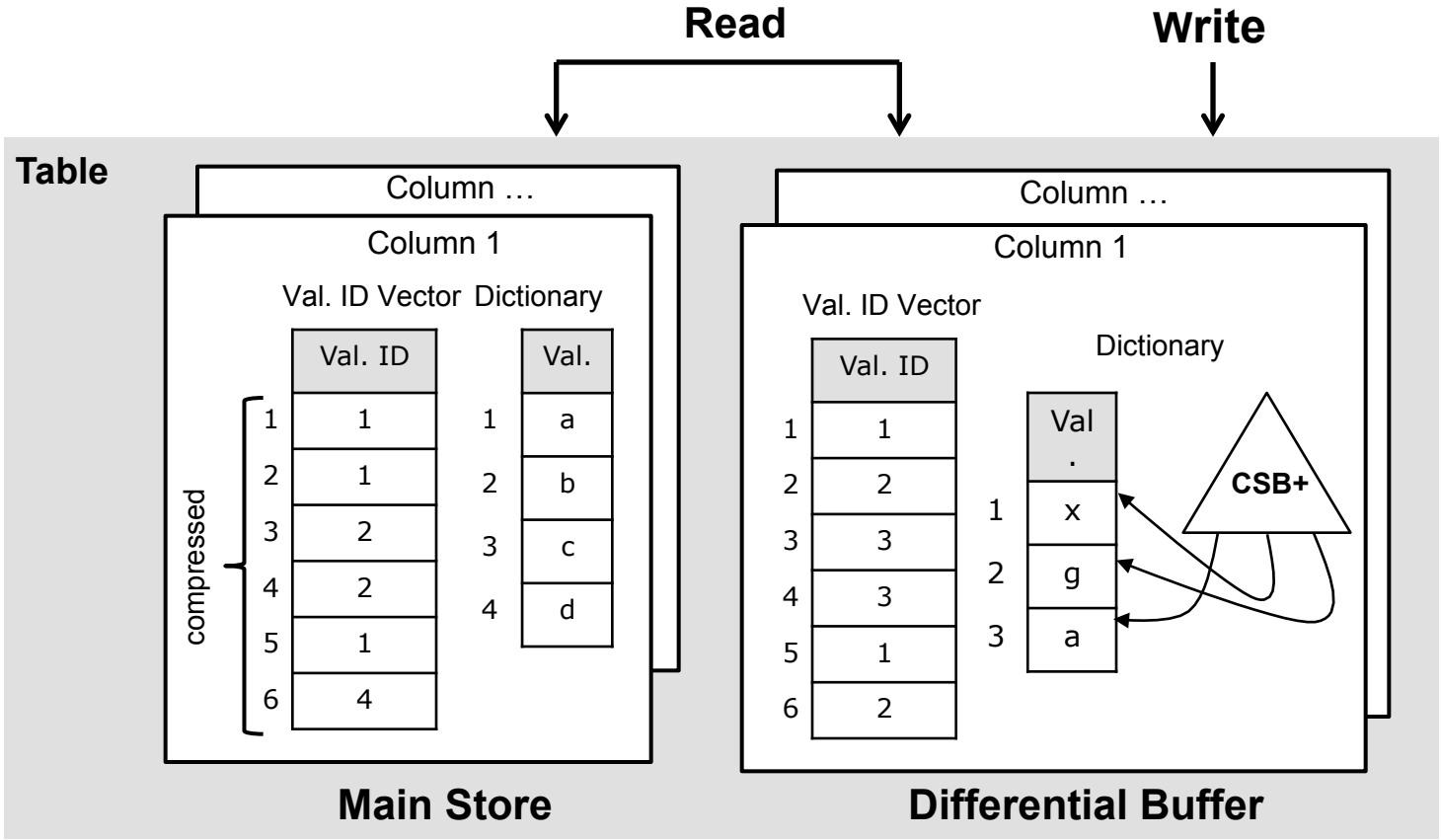
Handling Data Modifications

Jens Krueger

Enterprise Platform and Integration Concepts
Hasso Plattner Intitute

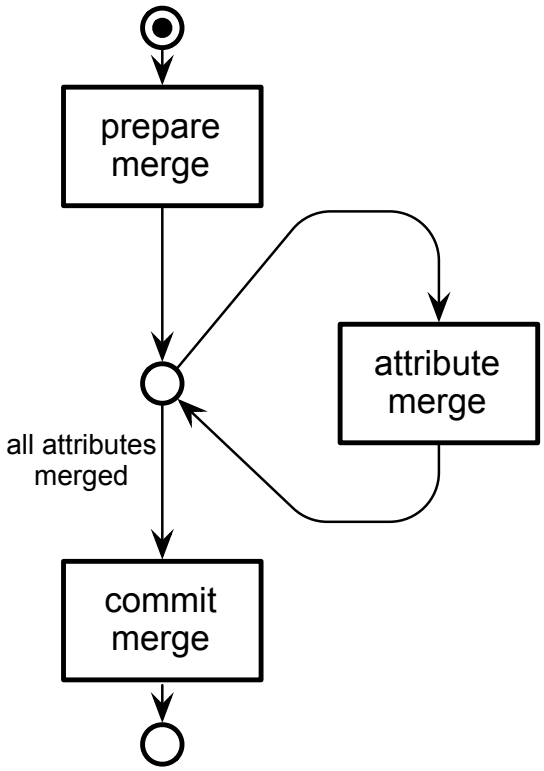
Handling Write Operations

- Inserting new tuples directly into a compressed structure is prohibitively expensive
- New values are written to a dedicated write-optimized differential buffer (delta)



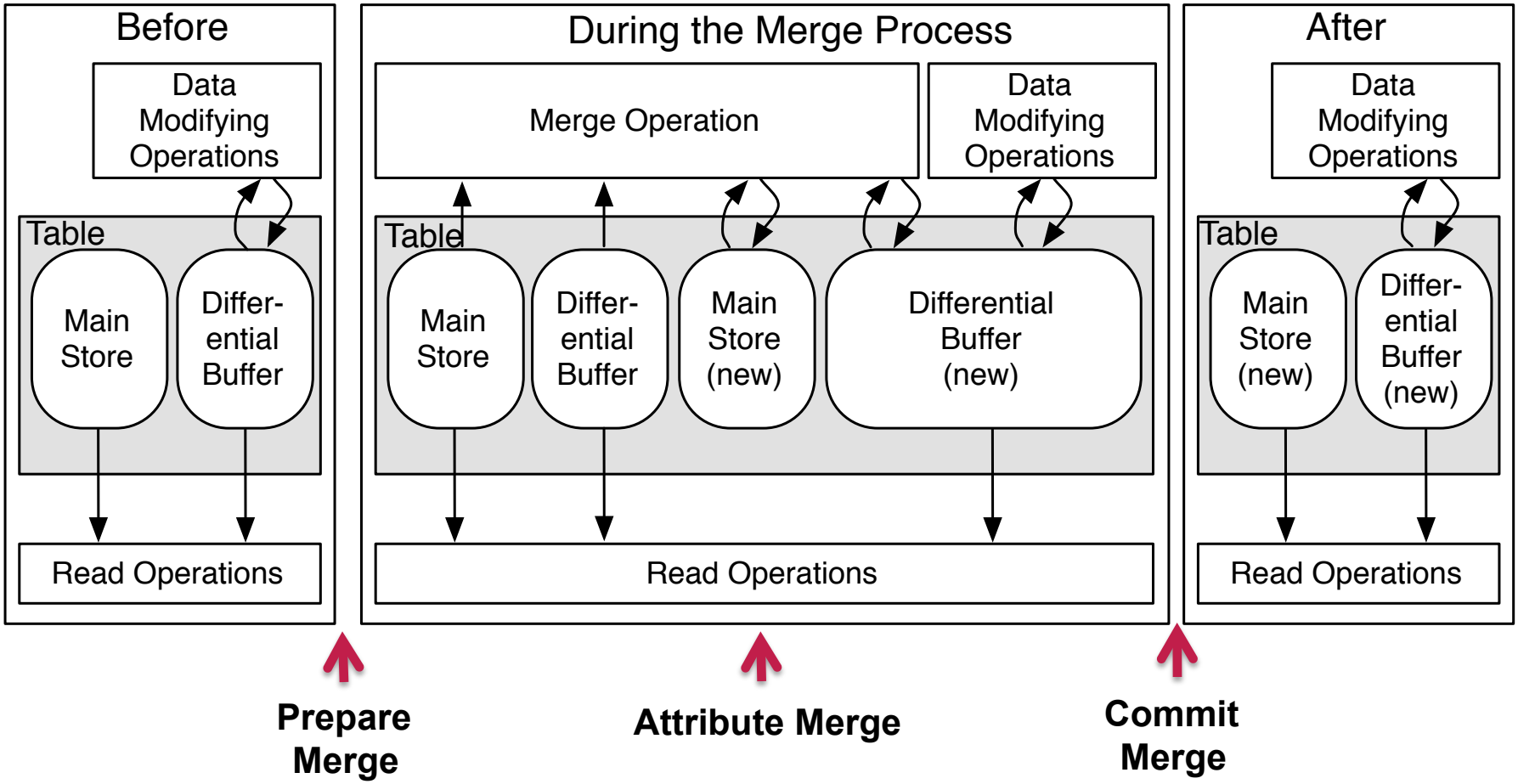
Merge Process Overview

- The merge process is working table wise
- Consists of:
  - Prepare merge
  - Attribute Merge
  - Commit Merge
- Is triggered by:
  - Amount of tuples in buffer
  - Cost model to
    - Schedule
    - Take query cost into account



Online Merge

- Working on data copies allows asynchronous merge
- Very limited interruption due to short lock
- But: at least twice the memory needed

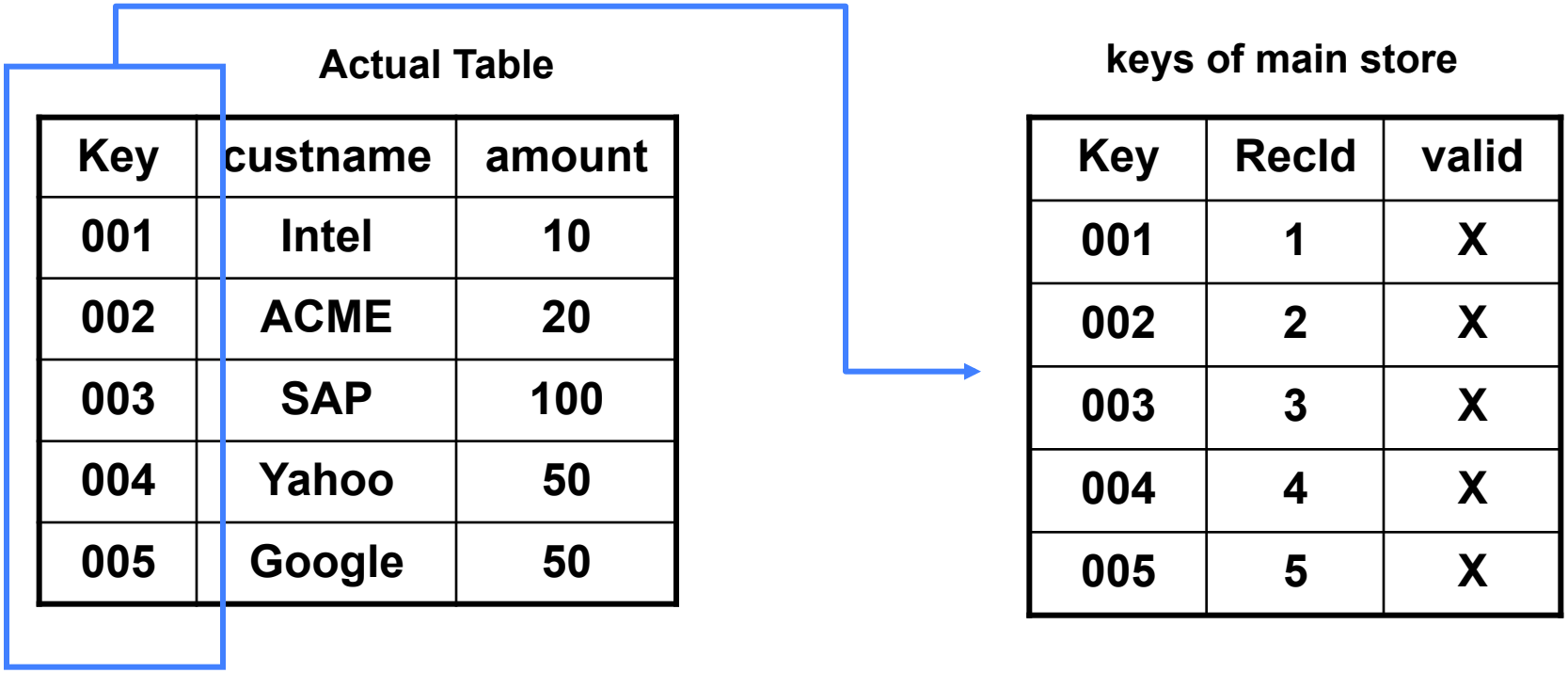


Attribute Merge

- Step 1
  - Extract delta dictionary from delta column
  - Merge main and delta dictionary
  - Create Mapping if valueID changed
- Step 2
  - Create new main column (if needed)

- Main store column needs to be rewritten if
  - Dictionary increases over #bit value range
  - Sort order changes

Main Store – Valid Tuples



- **RecId** is a synonym for row identifier
- Auxiliary structure to maintain valid tuples
- A bit vector represents valid tuples
  (for fast invalidation, e.g. updates)

Column *NAME* in Main Store

NAME
Intel
ACME
SAP
Yahoo
Google

=

Dictionary vector

Value	ValueID
ACME	1
Google	2
Intel	3
SAP	4
Yahoo	5

+

Attribute vector

RecId	ValueID
1	3
2	1
3	4
4	5
5	2

- Implicit identifiers
- Attribute Vector
  - RecID implicit
  - Insert order
- Dictionary Vector
  - ValueID is implicit
  - Sorted by the value (**fast find**)

Column *AMOUNT* in Main Store

AMOUNT
10
20
100
50
50

=

Dictionary vector

Value	Value Id
10	1
100	2
20	3
50	4

+

Attribute vector

RecId	Value Id
1	1
2	3
3	2
4	4
5	4

- Implicit identifiers
- Attribute Vector
  - RecID implicit
  - Insert order
- Dictionary Vector
  - ValueID is implicit
  - Sorted by the value
    (**fast find and late materializing range query**)

Delta Store: *Key* handling

Key	*NAME*	*AMOUNT*
007	Microsoft	40
003	SAP	500
007	Microsoft	100

Insert key in delta store

insert key in delta store

keys of main store

Key	RecId	valid
001	1	X
002	2	X
003	3	X
004	4	X
005	5	X

keys of delta store

Key	RecId	valid
007	1	X

check, if key in main store has to be invalidated

Delta Store: Non-Key Attribute *NAME*

Key	*NAME*	*AMOUNT*
007	Microsoft	40
003	SAP	500
007	Microsoft	100

Insert non-key attribute (*NAME*)

inserts new value into Btree attribute

BTree Attribute

Value	Microsoft
Value Id	1

Attribute vector

RecId	1
Value Id	1

inserts new entry in attribute vector

Delta Store: Non-Key Attribute *AMOUNT*

Key	*NAME*	AMOUNT
007	Microsoft	40
003	SAP	500
007	Microsoft	100

Insert non-key attribute (amount)

inserts new value into Btree attribute

BTree Attribute

Value	40
Value Id	1

Attribute vector

RecId	1
Value Id	1

inserts new entry in attribute vector

Again: 2nd tuple

Key	*NAME*	AMOUNT
007	Microsoft	40
003	SAP	500
007	Microsoft	100

insert second tuple (key)

insert key in delta store

keys of main store

Key	RecID	valid
001	1	X
002	2	X
003	3	-
004	4	X
005	5	X

keys of delta store

Key	RecID	valid
007	1	X
003	2	X

checks and invalidates key in main store, checks if key also exists in delta store

Again: 2nd tuple

Key	*NAME*	AMOUNT
007	Microsoft	40
003	SAP	500
007	Microsoft	100

Insert non-key attribute (*NAME*)

BTree Attribute

Value	Microsoft	SAP
Value Id	1	2

inserts new value into Btree attribute

Attribute vector

RecID	1	2
Value Id	1	2

inserts new entry in Attribute vector

Again: 2nd tuple

Key	*NAME*	AMOUNT
007	Microsoft	40
003	SAP	500
007	Microsoft	100

Insert non-key attribute (*AMOUNT*)

inserts new value into Btree attribute

BTree Attribute

Value	40	500
Value Id	1	2

Attribute vector

RecID	1	2
Value Id	1	2

inserts new entry in Attribute vector

Again: 3rd tuple - Update

Key	*NAME*	AMOUNT
007	Microsoft	40
003	SAP	500
007	Microsoft	100

insert third tuple (key)

insert key in delta store

keys of delta store

Key	RecID	valid
007	1	-
003	2	X
007	1	X

keys of main store

Key	RecID	valid
001	1	X
002	2	X
003	3	-
004	4	X
005	5	X

checks key in main store,
invalidates old key in delta store

Again: 3rd tuple - Update

Key	*NAME*	AMOUNT
007	Microsoft	40
003	SAP	500
007	Microsoft	100

insert third tuple (name)

BTree Attribute

Value	Microsoft	SAP
Value Id	1	2

value allready exists in BTree

Attribute vector

RecID	1	2	3
Value Id	1	2	1

inserts new entry in attribute vector

Again: 3rd tuple - Update

Key	*NAME*	AMOUNT
007	Microsoft	40
003	SAP	500
007	Microsoft	100

insert third tuple (amount)

inserts new value into Btree attribute

BTree Attribute

Value	40	500	100
Value Id	1	2	3

Attribute Vector

RecID	1	2	3
Value Id	1	2	3

inserts new entry in attribute vector

Attribute Merge – **First**: Merge of Keys

keys of main store

Key	RecId	valid
001	1	X
002	2	X
003	3	X
004	4	X
005	5	X
007	6	X

keys of delta delta store

Key	RecID	valid
007	1	-
003	2	X
007	3	X

will be moved to the history partition

RecId Mappings (Delta Store → Main Store)

RecID (Delta)	2	3
RecID (Main)	3	6

RecId for history

RecID (Delta)	1
RecID (Main)	3

Merge Dictionary: Column *NAME*

Hasso Plattner Institut

new dictionary (Main Store)

Value	Value Id
ACME	1
Google	2
Intel	3
Microsoft	4
SAP	5
Yahoo	6

BTree Attribute (Delta Store)

Value	Microsoft	SAP
Value Id	1	2

ValueId Mapping (Delta Store)

Value ID (old)	1	2
Value ID (new)	4	5

ValueID Mapping (Main Store)

Value ID (old)	1	2	3	4	5
Value ID (new)	1	2	3	5	6

Adapt Attribute Vector (*NAME*)

Hasso Plattner Institut

ValueId Mapping (Main Store)

Value Id (old)	1	2	3	4	5
Value Id (new)	1	2	3	5	6

RecID Mappings (Delta Store → Main Store)

RecID (Delta)	2	3
RecID (Main)	3	6

ValueID Mapping (Delta Store)

ValueID (old)	1	2
ValueID (new)	4	5

Attribute Vector (Main Store)

RecID	Value Id
1	3
2	1
3	5
4	6
5	2
6	4

Attribute Vector (Delta Store)

RecID	1	3	6
ValueID	1	5	4

The New Main Store

NAME
Intel
ACME
SAP
Yahoo
Google
Microsoft

=

Man Store Dictionary

Value	ValueID
ACME	1
Google	2
Intel	3
Microsoft	4
SAP	5
Yahoo	6

+

Main Store Attribute Vector

RecID	ValueID
1	3
2	1
3	5
4	6
5	2
6	4

same will be done for attribute *AMOUNT*