

Hybrid Row-Column Partitioning in Teradata

Luca Gasparini
October 17, 2017
HADP Seminar



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Data Partitioning

- Easier management
- Assist backup/recovery
- Enhance Performance

Data Partitioning

```
SELECT avg(C)
FROM T
WHERE A between 3 and 9
```

No Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Data Partitioning

```
SELECT avg(C)  
FROM T  
WHERE A between 3 and 9
```

No Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Data Partitioning

```
SELECT avg(C)
FROM T
WHERE A between 3 and 9
```

Horizontal Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Vertical Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Data Partitioning

```
SELECT avg(C)
FROM T
WHERE A between 3 and 9
```

Horizontal Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Vertical Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Data Partitioning

```
SELECT avg(C)  
FROM T  
WHERE A between 3 and 9
```

Horizontal Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Vertical Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Data Partitioning

```
SELECT avg(C)  
FROM T  
WHERE A between 3 and 9
```

Column & Row Partitioning

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

TERADATA®

THE BEST DECISION POSSIBLE™

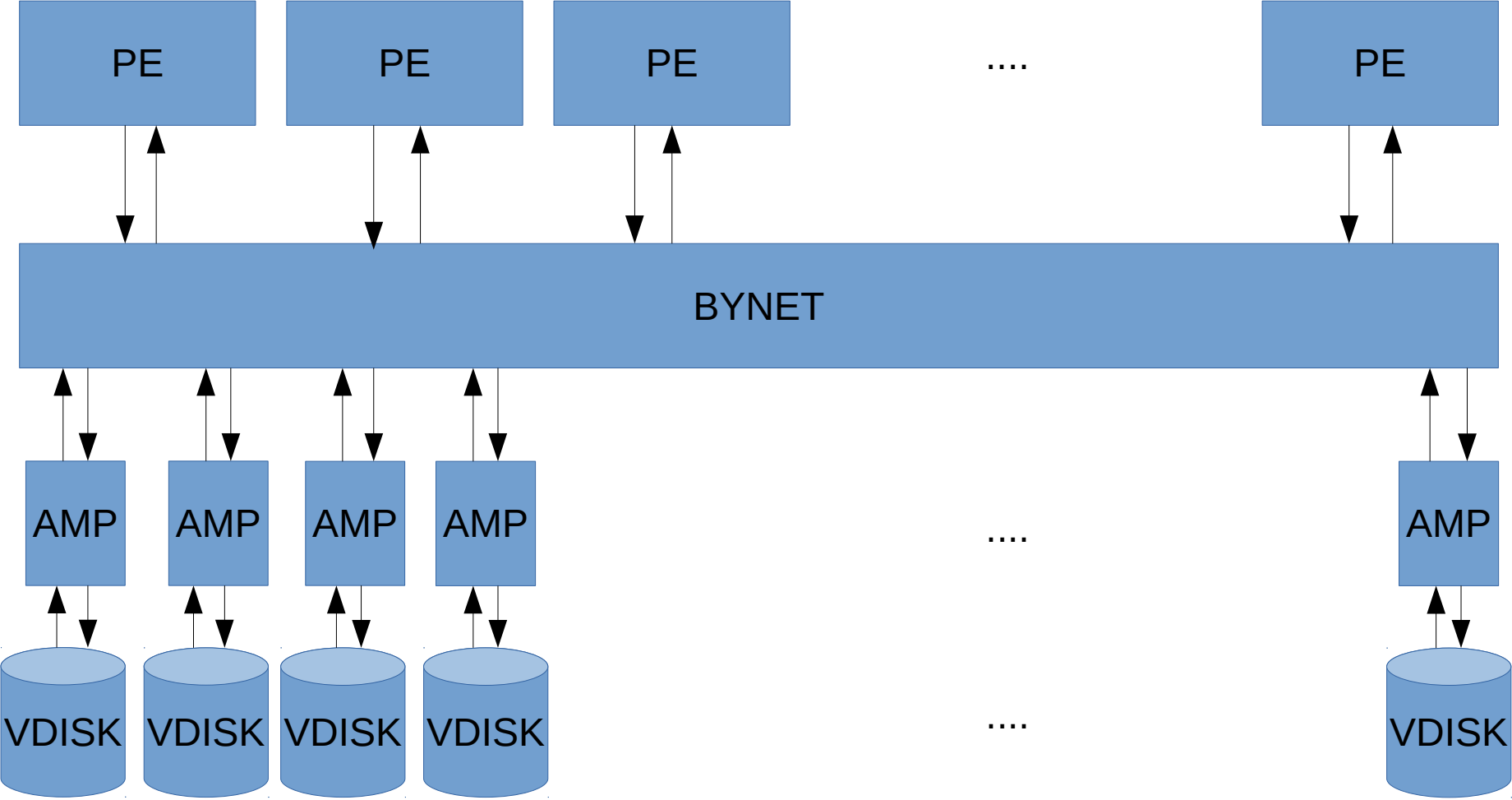
TERADATA.

THE BEST DECISION POSSIBLE™

Wikipedia:

- Integrated Data Warehouse Hardware and Software
- Unified Data Architecture
- Big Data Analytics
- Professional Services
- Customer Services

Teradata Parallel Database



Multilevel Partitioning

```
CREATE TABLE table(  
    ID INTEGER,  
    date DATE,  
    c1 TYPE,  
    ....  
    cn TYPE  
)
```

Multilevel Partitioning

```
CREATE TABLE table(  
  ID INTEGER,  
  date DATE,  
  c1 TYPE,  
  ....  
  cn TYPE  
)
```

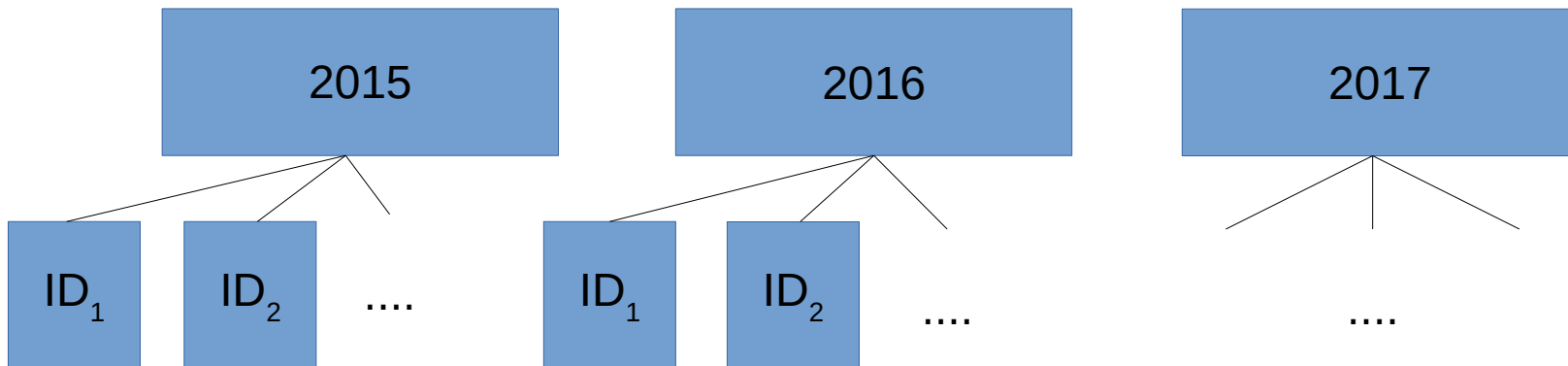
2015

2016

2017

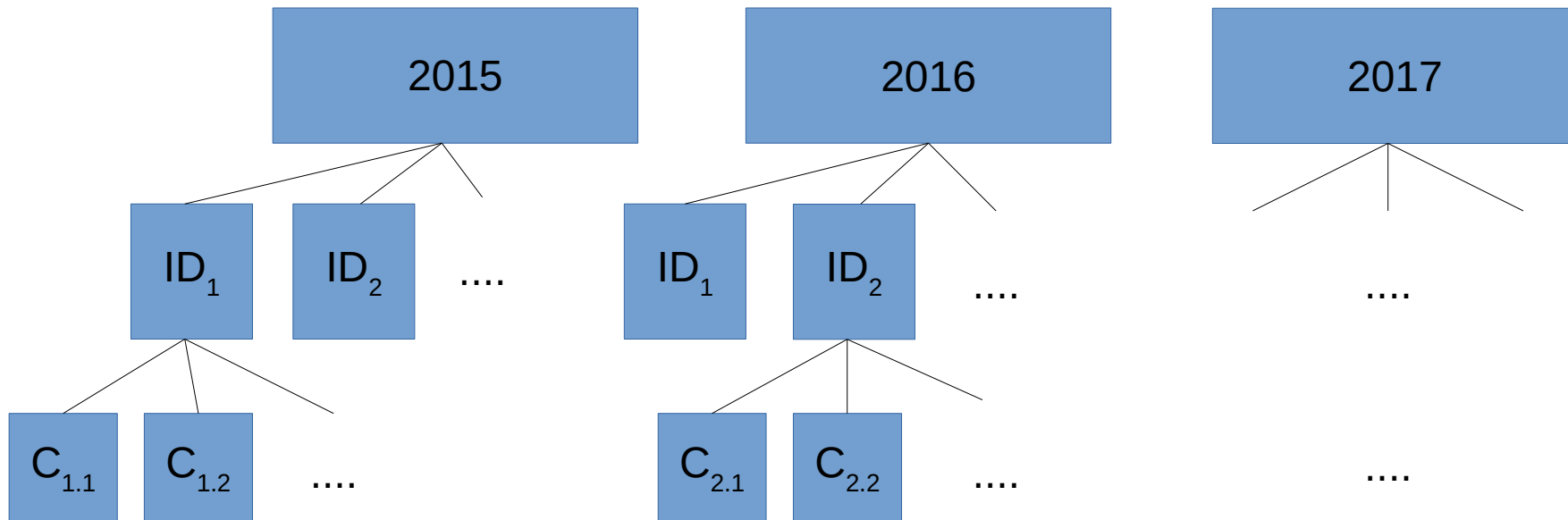
Multilevel Partitioning

```
CREATE TABLE table(  
  ID INTEGER,  
  date DATE,  
  c1 TYPE,  
  ....  
  cn TYPE  
)
```



Multilevel Partitioning

```
CREATE TABLE table(  
  ID INTEGER,  
  date DATE,  
  c1 TYPE,  
  ....  
  cn TYPE  
)
```



File System

- doesn't care about partitioning scheme
- *rowid* based (unique 16-byte key)

rowid

3 types of indices:

- PI (Primary Index)
- PA (Primary AMP Index)
- NoPI (no Primary Index)

PI & PA

Rows are hash-distributed over AMPs
on value of PI/PA column

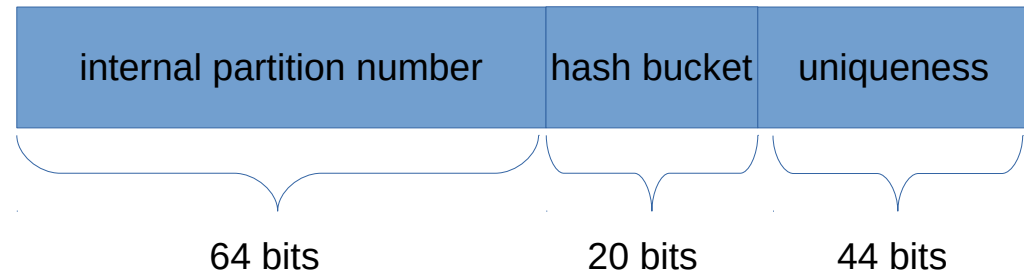
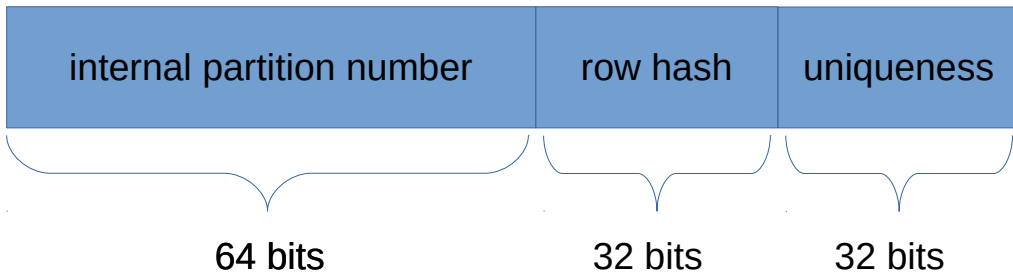
NoPI

Rows are distributed
randomly over AMPs

rowid

PI table

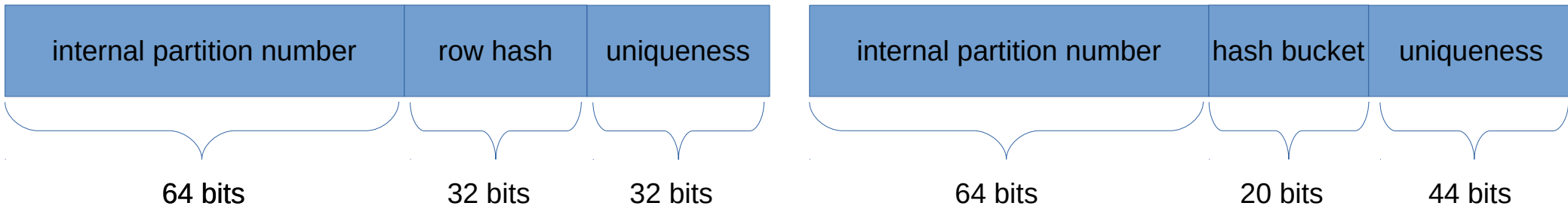
PA/NoPi table



rowid

PI table

PA/NoPi table

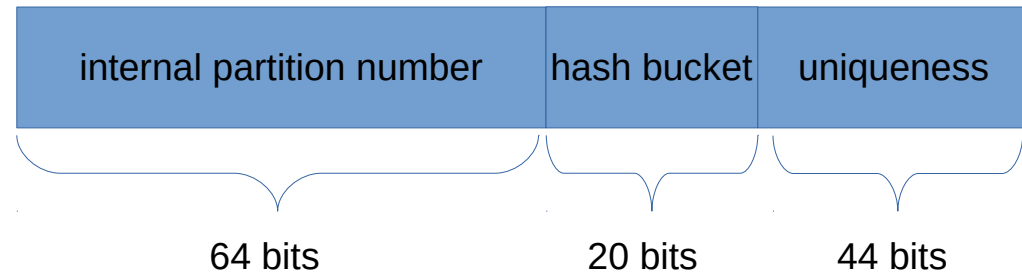
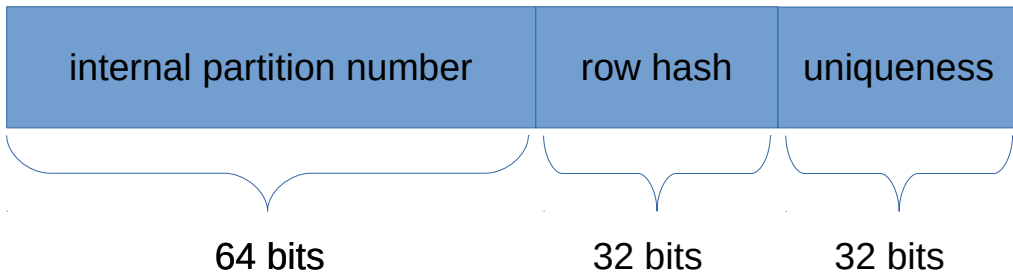


	PI tables	PA tables	NoPi tables
Hash-distribution to AMP	YES	YES	NO
Row-hash Ordering on AMP	YES	NO	NO

rowid

PI table

PA/NoPi table

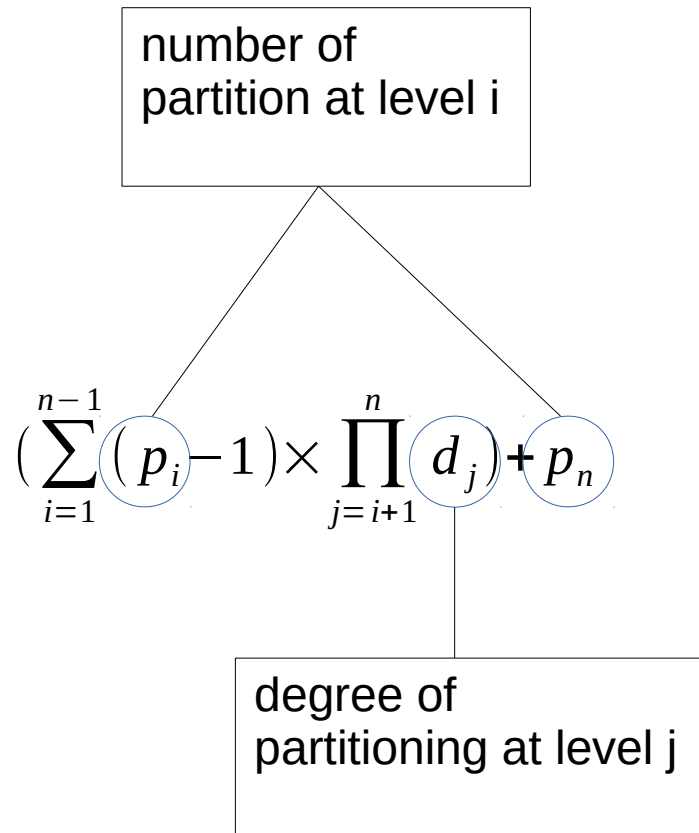


$$\prod_{i=1}^n d_i < (2^{63} - 1)$$

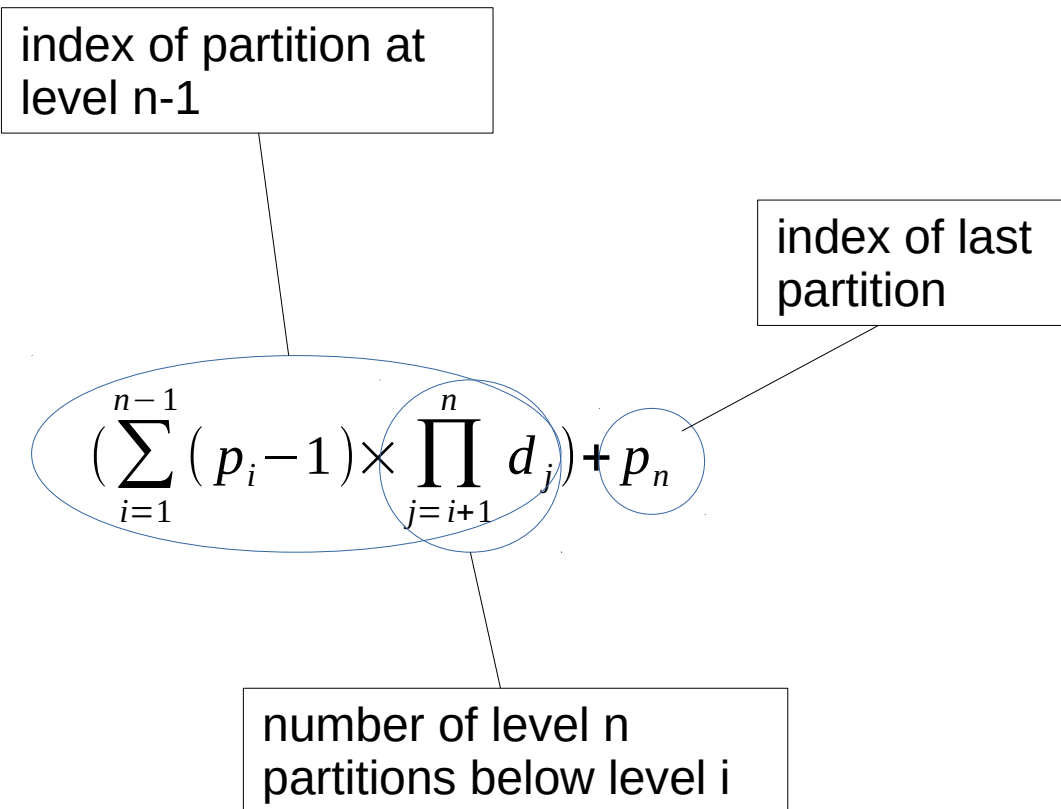
CPN (Combined Partition Number)

$$\left(\sum_{i=1}^{n-1} (p_i - 1) \times \prod_{j=i+1}^n d_j \right) + p_n$$

CPN (Combined Partition Number)

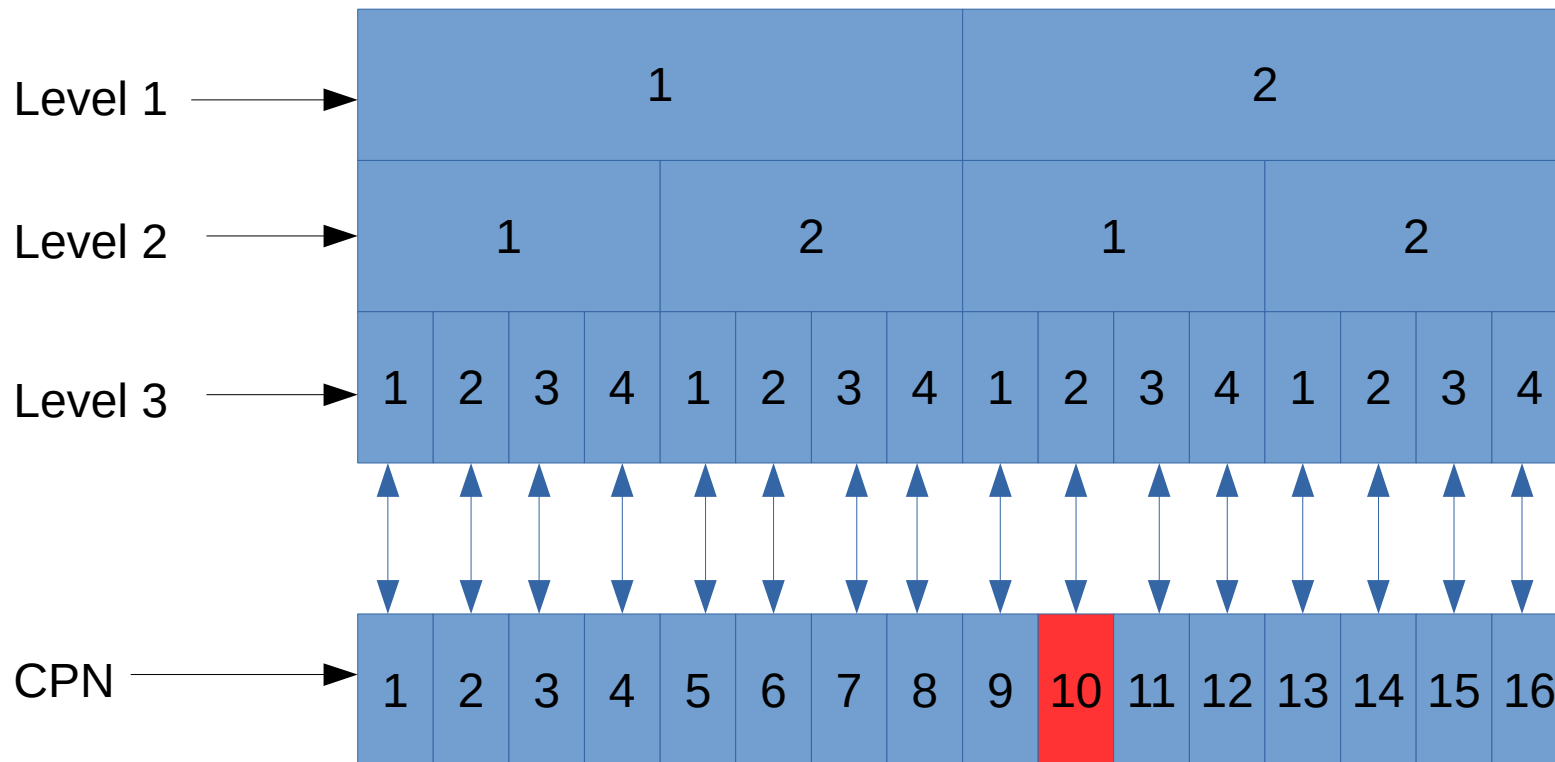


CPN (Combined Partition Number)



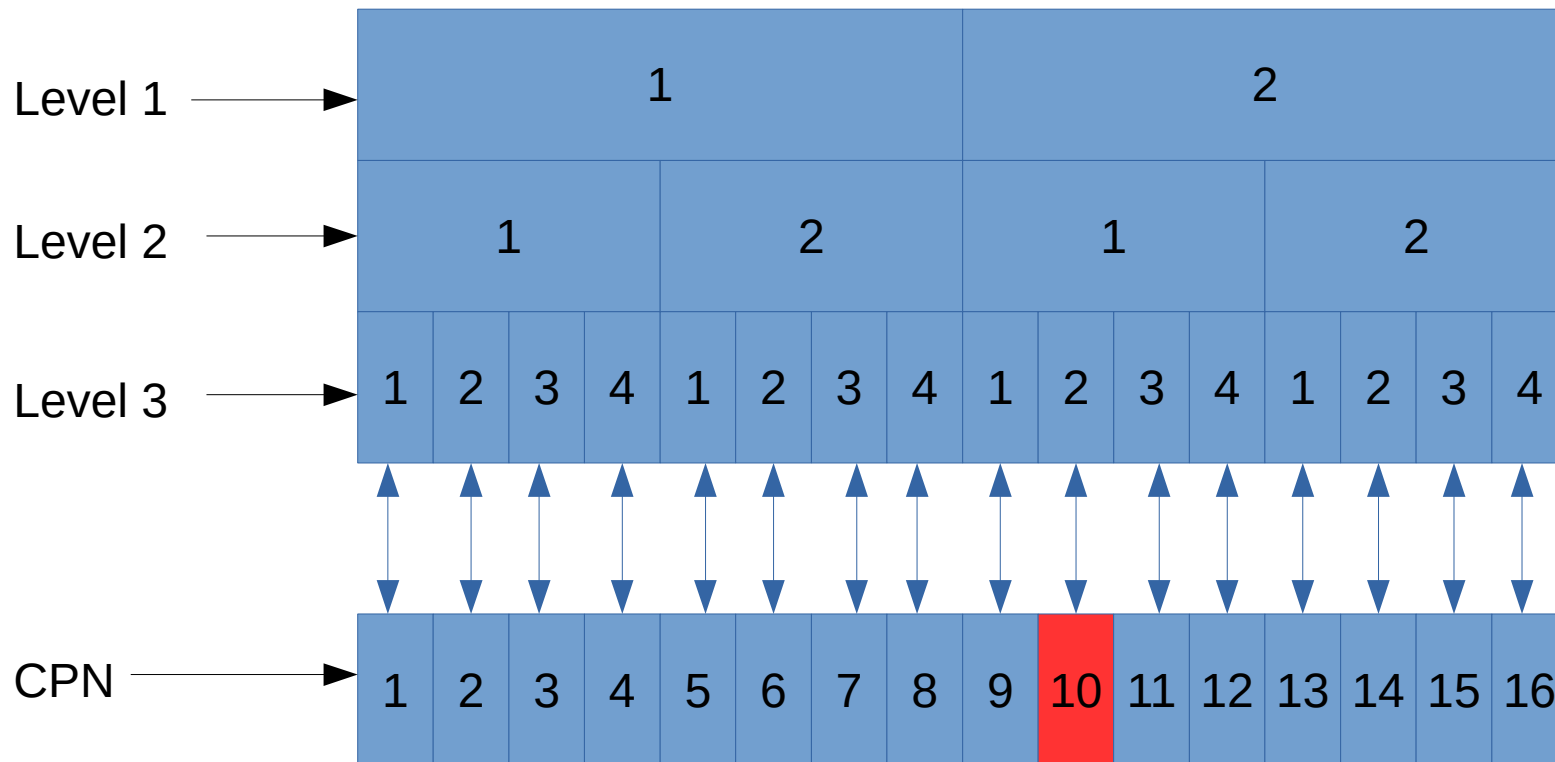
CPN (Combined Partition Number)

$$\left(\sum_{i=1}^{n-1} (p_i - 1) \times \prod_{j=i+1}^n d_j \right) + p_n$$



CPN (Combined Partition Number)

$$\left(\sum_{i=1}^{n-1} (p_i - 1) \times \prod_{j=i+1}^n d_j \right) + p_n$$



$$((2 - 1) * (2 * 4)) + ((1 - 1) * 4) + 2 = 10$$

Partitioning Expressions

PARTITION BY(

) **...**

Partitioning Expressions

COLUMN [[NO] AUTO COMPRESS]

**PARTITION BY(
 partition_expr₁,
 ...,
 partition_expr_n
)**

**RANGE_N(test-value BETWEEN
 range [, range] ...
)**

**CASE_N(condition-expr₁,
 ...,
 condition-expr_n
)**

Partitioning Expressions

```
CREATE TABLE Sales(TxnNo INTEGER,  
                    TxnDate DATE,  
                    ItemNo INTEGER,  
                    Quantity INTEGER,)  
  
NO PRIMARY INDEX,  
PARTITION BY(  
    COLUMN,  
    RANGE_N(TxnDate BETWEEN  
            Date '2011-01-01' AND DATE '2011-12-31'  
    EACH INTERVAL '1' DAY))
```

Teradata Columnar

- ROW format
- COLUMN format

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Teradata Columnar

- ROW format
- COLUMN format

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

Teradata Columnar

- ROW format

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

- COLUMN format

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

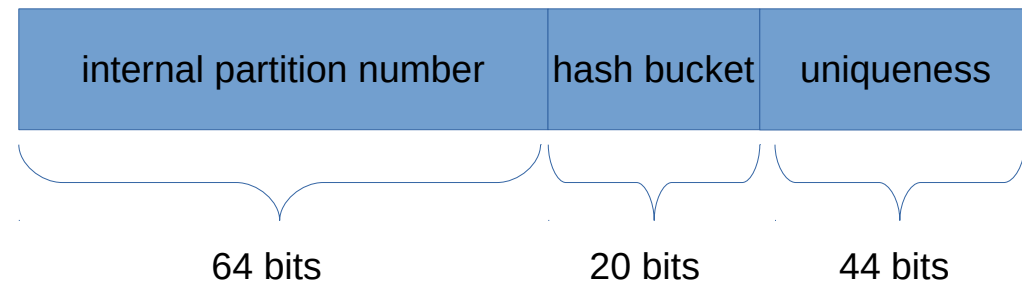
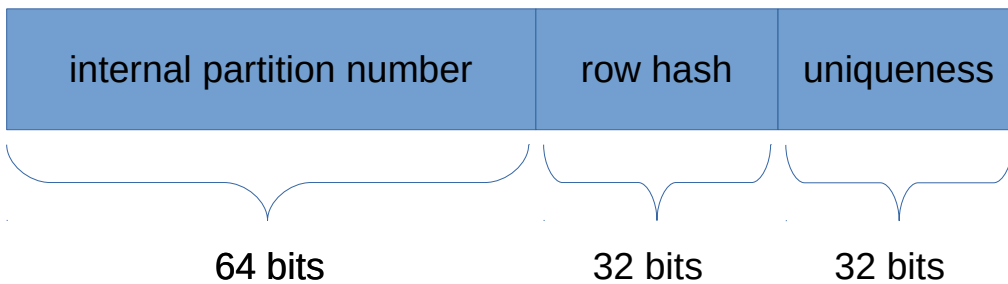
Teradata Columnar

- COLUMN format

A	B	C	D
10	11	2	8
6	3	4	7
1	12	9	10
5	0	2	8

PI table

PA/NoPi table

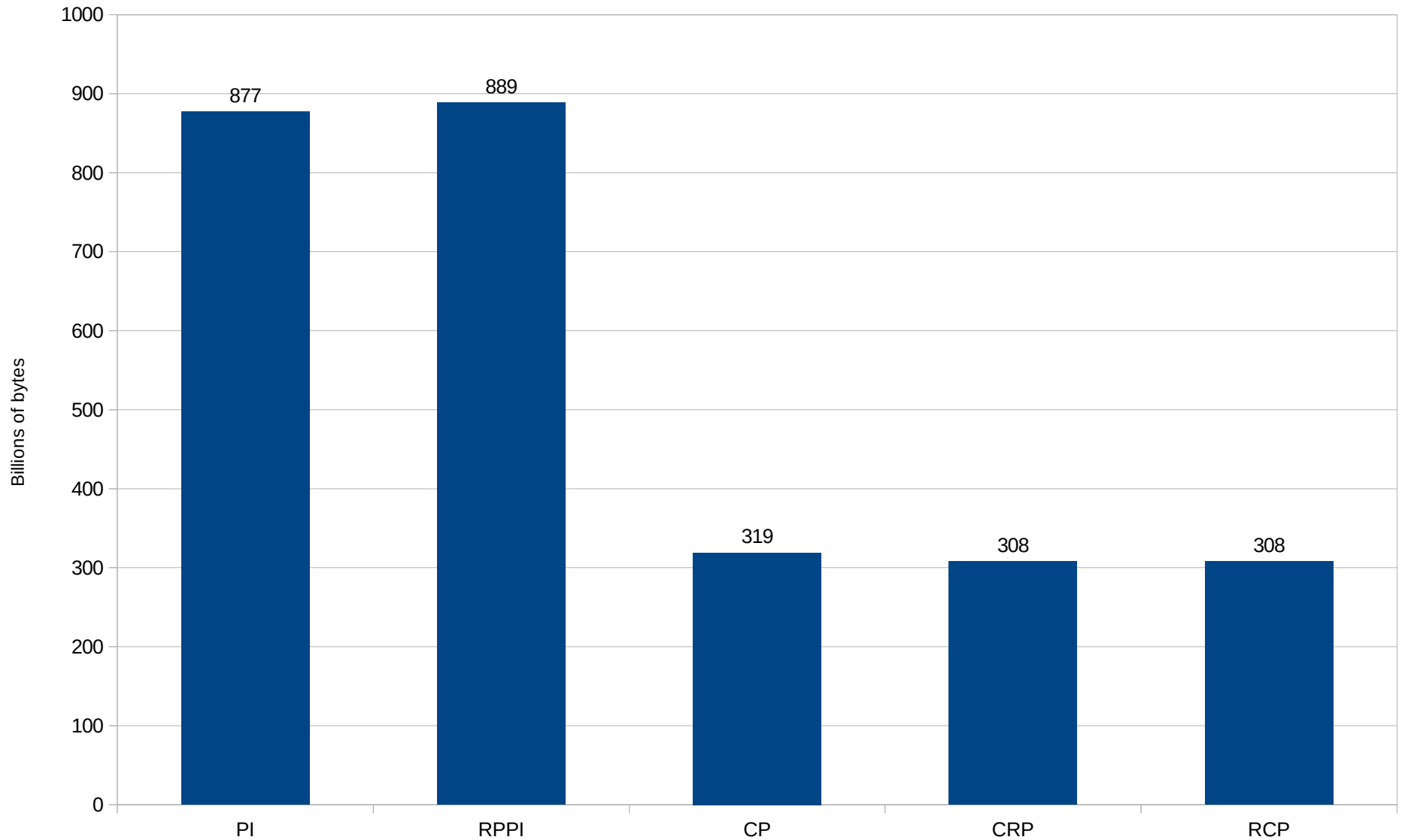


Experimental Results

PI	Regular PI table with no partitioning
RPPI	Partitioned primary index table with 84 monthly row partitions
CP	NoPI Single-column CP table with COLUMN format and autocompression
CRP	Multilevel partitioned table with column partitioning at the first level and row partitioning (by month) at the second level
RCP	Multilevel partitioned table with row partitioning (by month) at the first level and column partitioning at the second level

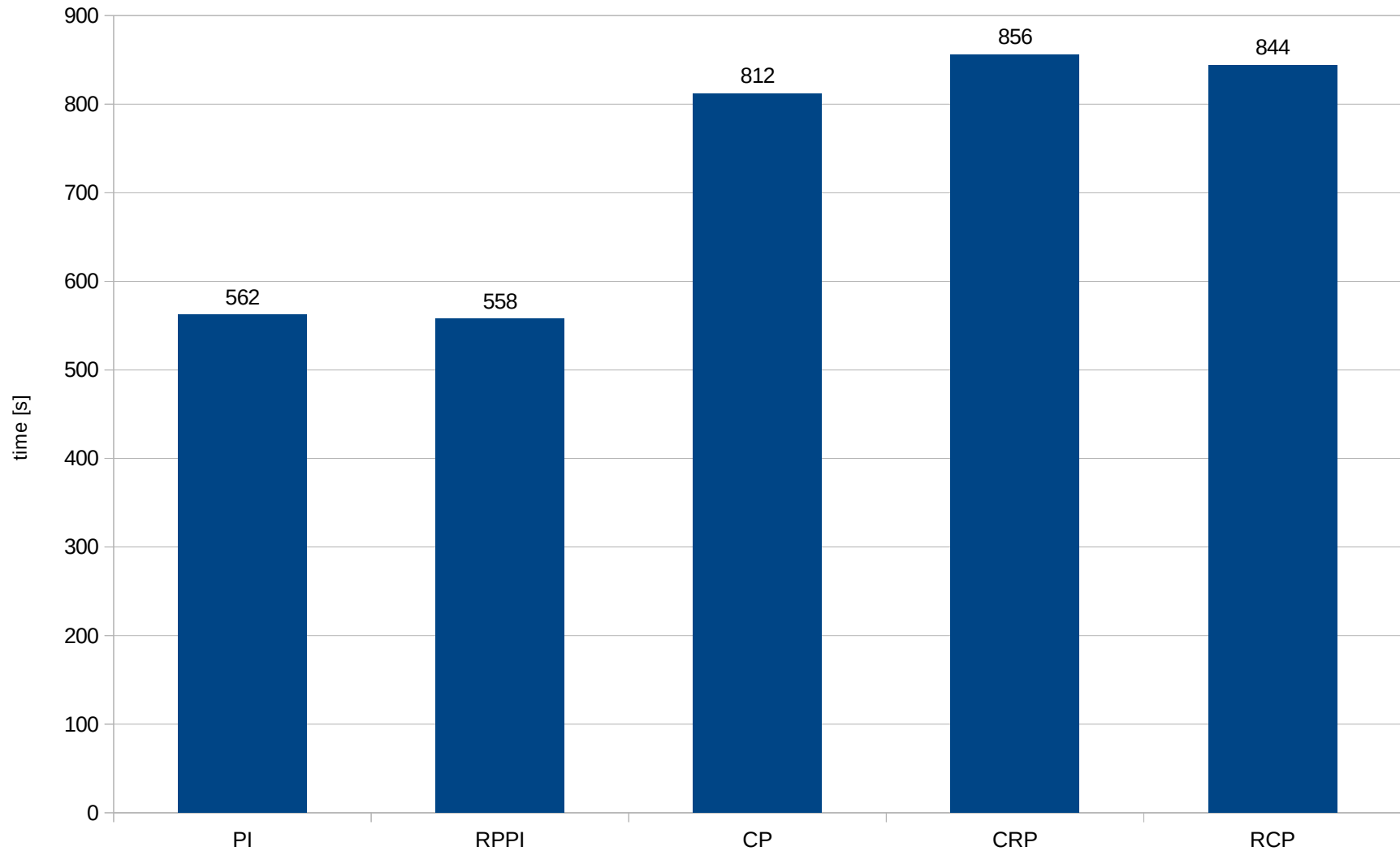
Experimental Results

Table Size



Experimental Results

Full Table Scan



Experimental Results

