Database Cracking

September 7, 2016
Row Stores

Traditional DB: Lay out data on disk in rows
Column Stores

Columnar DB: Lay out data on disk in columns
Column Stores

Store with Row ID to recover original table
Why use a Column Store?
Immediate Data Access

**Problem:** Data is initially unsorted

**Query:** Find all rows where $100 < A \leq 200$

What is the fastest way to answer this query?
Immediate Data Access

**Problem:** Data is initially unsorted

What if you get 2 queries?

... 3 queries?

... 100 queries?
Immediate Data Access

**Problem:** Data is initially unsorted

**Strategy 1:** Index the data then run queries
First few queries are much slower (upfront indexing cost)

**Strategy 2:** Linear scans over the data
Last few queries are much slower (no indexing!)
Immediate Data Access

**Problem:** Data is initially unsorted

**Strategy 3:** Index *while* you run queries!

Re-use compute effort of scans.
Cracking

Query 1: Find $4 < X \leq 7$
Cracking

**Step 1:** Split into 2 bins: $> 4$ and $\leq 4$

Query 1: Find $4 < X \leq 7$
**Step 1:** Split into 2 bins: > 4 and ≤ 4

**Query 1:** Find 4 < X ≤ 7
Cracking

**Step 1:** Split into 2 bins: $> 4$ and $\leq 4$

Low

\[
\begin{array}{cccccccc}
0 & 9 & 3 & 5 & 4 & 8 & 7 & 1 & 2 & 6 \\
\end{array}
\]

Current

**Query 1:** Find $4 < X \leq 7$
Cracking

**Step 1:** Split into 2 bins: $> 4$ and $\leq 4$

**Query 1:** Find $4 < X \leq 7$
Cracking

**Step 1**: Split into 2 bins: \( > 4 \) and \( \leq 4 \)

Query 1: Find \( 4 < X \leq 7 \)
Cracking

Step 1: Split into 2 bins: > 4 and ≤ 4

Query 1: Find 4 < X ≤ 7
Cracking

**Step 1:** Split into 2 bins: > 4 and ≤ 4

![Number sequence diagram]

Query 1: Find 4 < X ≤ 7
Cracking

**Step 1:** Split into 2 bins: $> 4$ and $\leq 4$

Low

```
0 3 4 5 9 8 7 1 2 6
```

Current

**Query 1:** Find $4 < X \leq 7$
Cracking

**Step 1:** Split into 2 bins: $> 4$ and $\leq 4$

- Low
- Current

**Query 1:** Find $4 < X \leq 7$
Cracking

**Step 1:** Split into 2 bins: $> 4$ and $\leq 4$

- Low

```
0 3 4 1 9 8 7 5 2 6
```

- Current

**Query 1:** Find $4 < X \leq 7$
Cracking

**Step 1:** Split into 2 bins: > 4 and ≤ 4

![Diagram](chart)

Query 1: Find 4 < X ≤ 7
Cracking

**Step 2:** Split into 2 bins: $> 7$ and $\leq 7$

```
0 3 4 1 2 8 7 5 9 6
```

**Query 1:** Find $4 < X \leq 7$
Cracking

**Step 2:** Split into 2 bins: > 7 and ≤ 7

![Array Diagram](image)

**Query 1:** Find 4 < X ≤ 7
Cracking

**Step 2:** Split into 2 bins: $> 7$ and $\leq 7$

Current

**Query 1:** Find $4 < X \leq 7$
Cracking

**Step 2:** Split into 2 bins: $> 7$ and $\leq 7$

```
Low
0 3 4 1 2 7 5 6 9 8
High
```

**Query 1:** Find $4 < X \leq 7$
Cracking

Step 2: Split into 2 bins: \( > 7 \) and \( \leq 7 \)

Low \( \downarrow \) High

Result

Query 1: Find \( 4 < X \leq 7 \)
Query 1: Find 4 < X ≤ 7
Query 2: Find $2 < X \leq 5$
Query 2: Find $2 < X \leq 5$
3-Way Cracking

Query 1: Find $4 < X \leq 7$
3-Way Cracking

Query 1: Find $4 < X \leq 7$
3-Way Cracking

Query 1: Find $4 < X \leq 7$
3-Way Cracking

Query 1: Find $4 < X \leq 7$
3-Way Cracking

Query 1: Find $4 < X \leq 7$
3-Way Cracking

Query 1: Find $4 < X \leq 7$
Discussion Questions…

Does cracking work with a row-oriented database?
Discussion Questions…

How would one crack a multi-attribute index?

(e.g., a spatial index?)
Discussion Questions…

Can updates be performed efficiently on a cracker index?
Discussion Questions…

Can updates be performed efficiently on a cracker index?

What constraints are required?
Discussion Questions…

What applications would cracking work well on?
What applications would cracking work poorly on?
Discussion Questions…

Upfront Indexing vs Sequential Scan vs Cracking…

Where is the cutoff?