Double Glazing: Two Years of Windowing Improvements

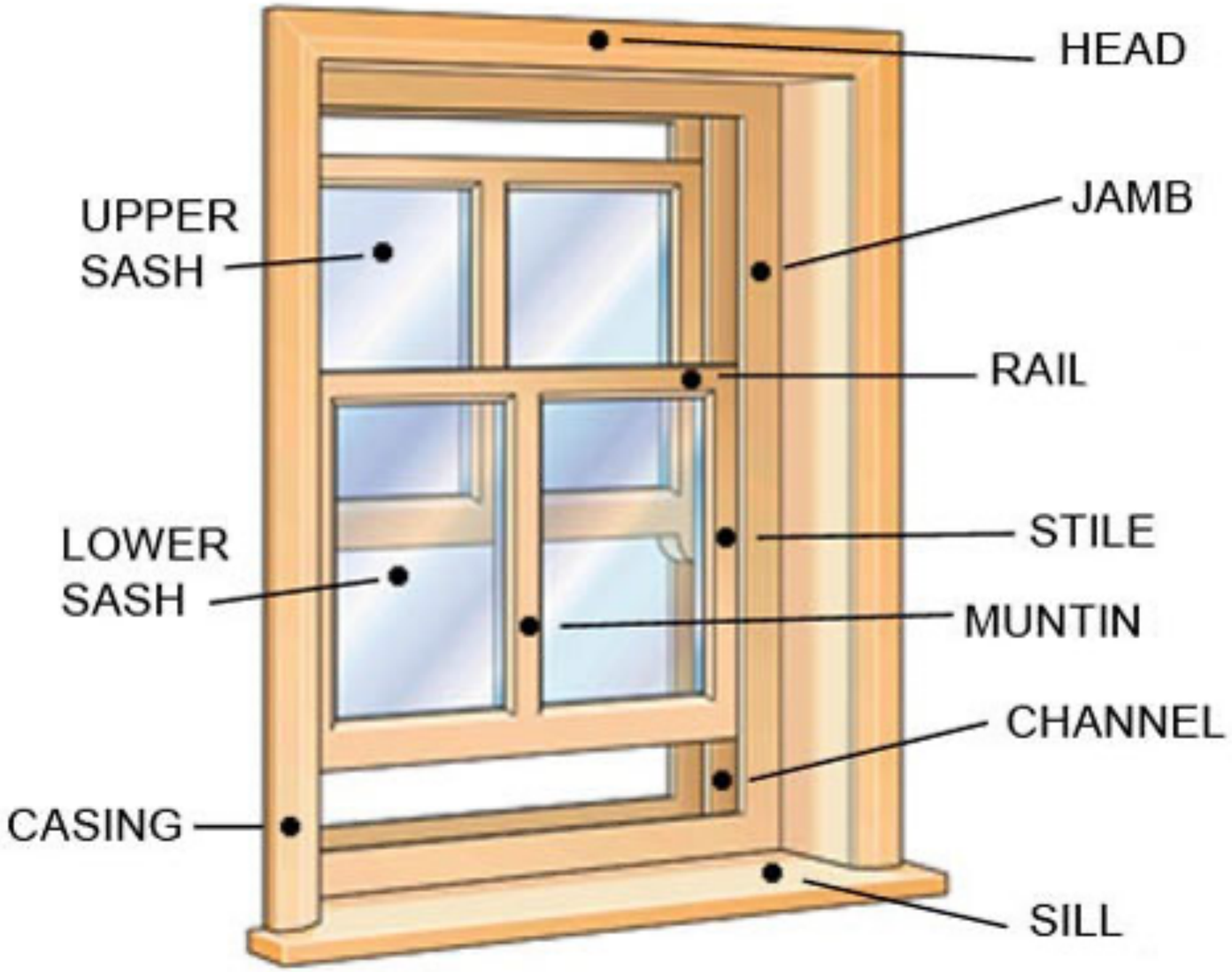
Richard Wesley

TL;DL

- Functionality
  - **`DISTINCT`**, `EXCLUDE`, `QUALIFY`
- Performance
  - **Partition Fusion**
  - Vectorisation
  - Streaming
- **Memory**
  - One partition at a time

Lightning on the Space Needle

The Windowing Model



Single Row Calculations

- Ordinary calculations
  - Produce a **new attribute**...
  - ...from a **single row**
- Only need the row
  - Trivial to **stream**
  - Trivial to parallelise

```
SELECT a + b AS c
```

a	b	c
1	1	2
5	-2	3
2	NULL	NULL

Multi-Row Calculations

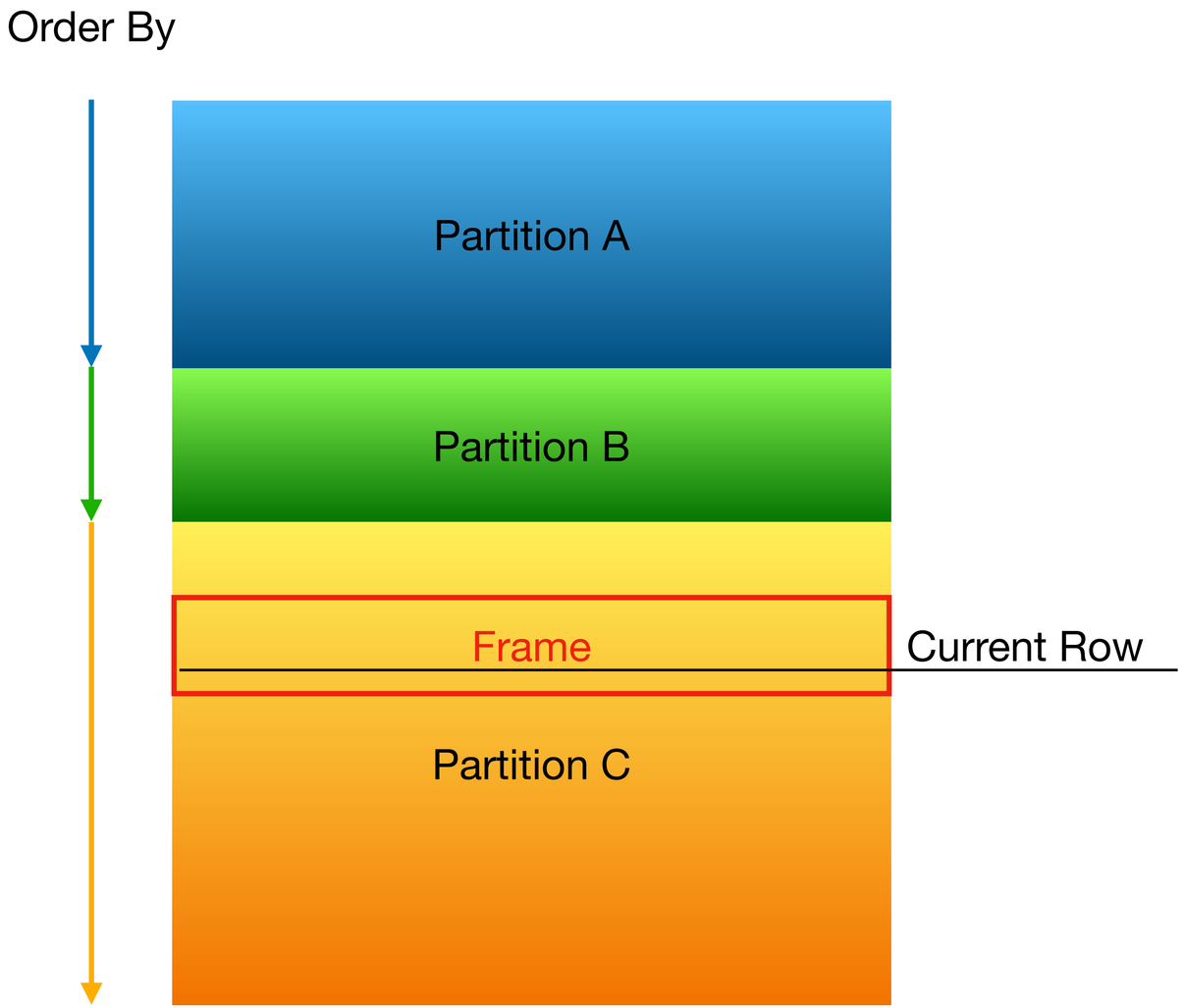
```
SELECT running_total(b) AS c
```

a	b	c
1	1	1
5	-2	-1
2	NULL	-1

- Windowing calculations
  - Produce a **new attribute**
  - From **adjacent rows**

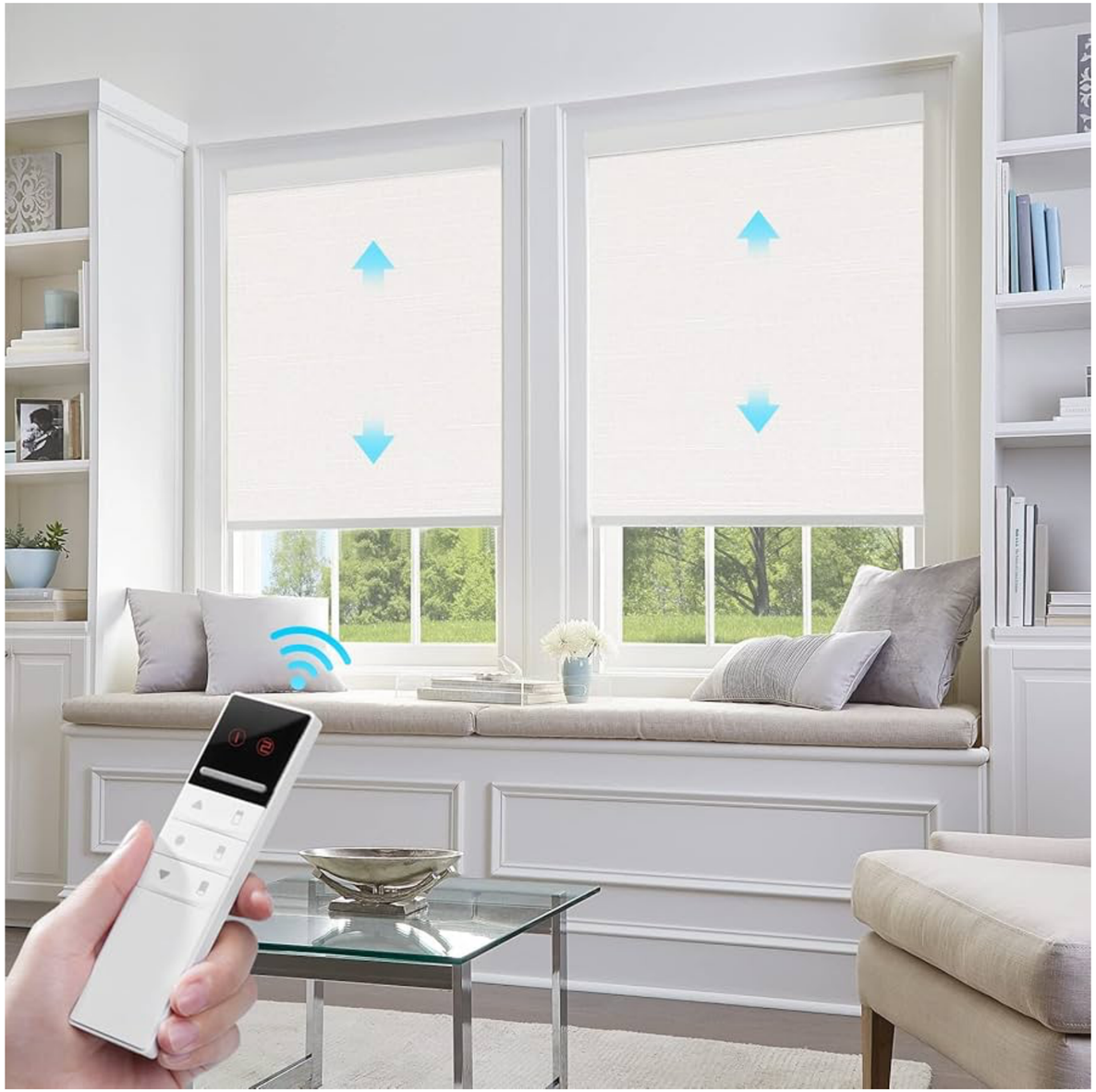
- What does **adjacent mean?**
  - Sets are unordered!
  - Need boundaries
  - Need ordering
  - Need distance

Visualising Windowing

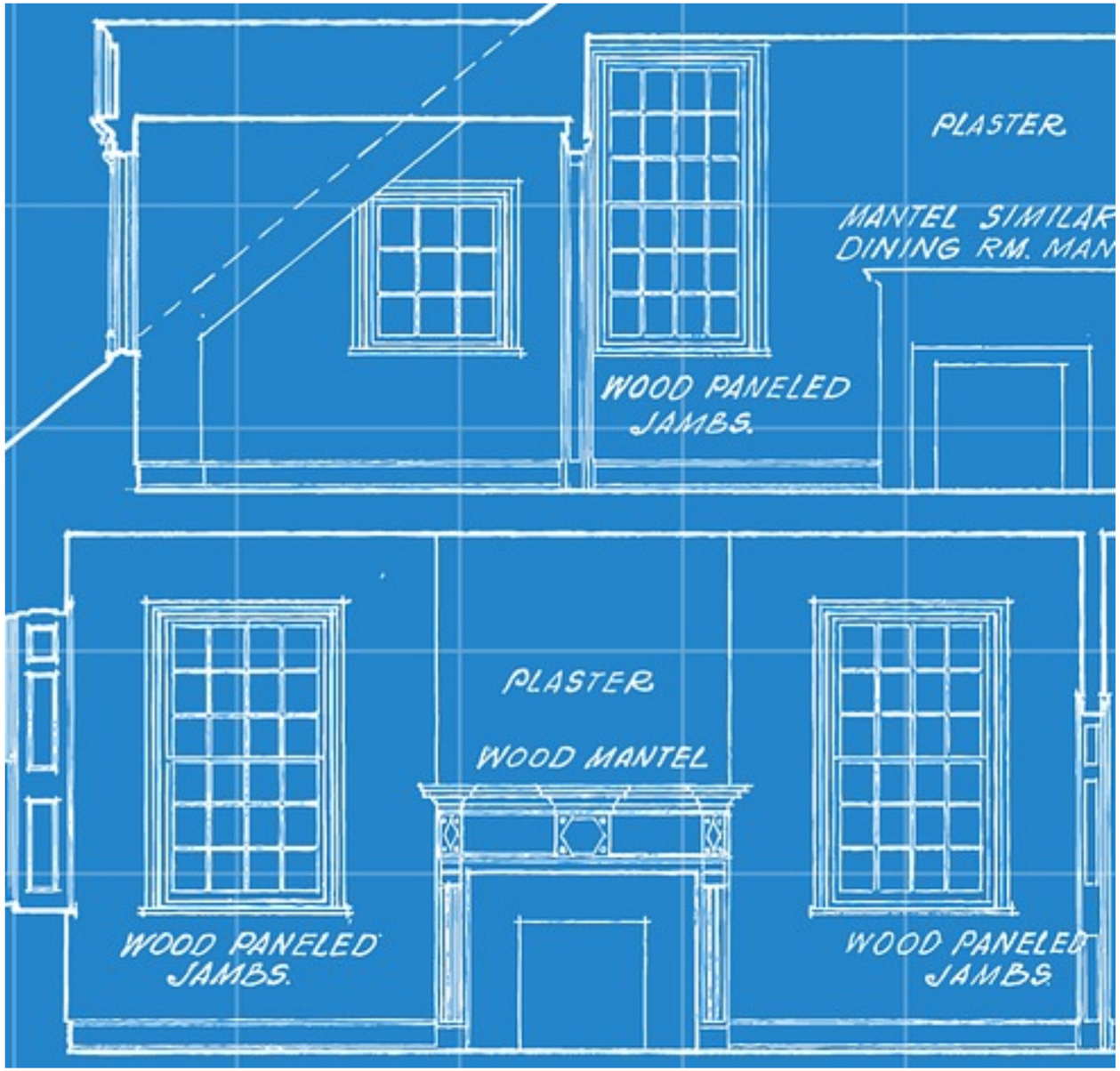


- **`PARTITION BY`**
  - Independent blocks of rows
- **`ORDER BY`**
  - Sort the partitions
- **`ROWS/RANGE BETWEEN`**
  - Distance from the current row

The Window Operators



Window Planning



Windows on a Blueprint

- **Streaming** or **materialised**?
  - Depends on the functions
  - Split out streamable ones
- Can we **combine partitions**?
  - Same partitioning keys
  - Ordering key prefixes
  - Cao et al. for general solution

Cao et al., Optimization of Analytic Window Functions, VLDB 2012

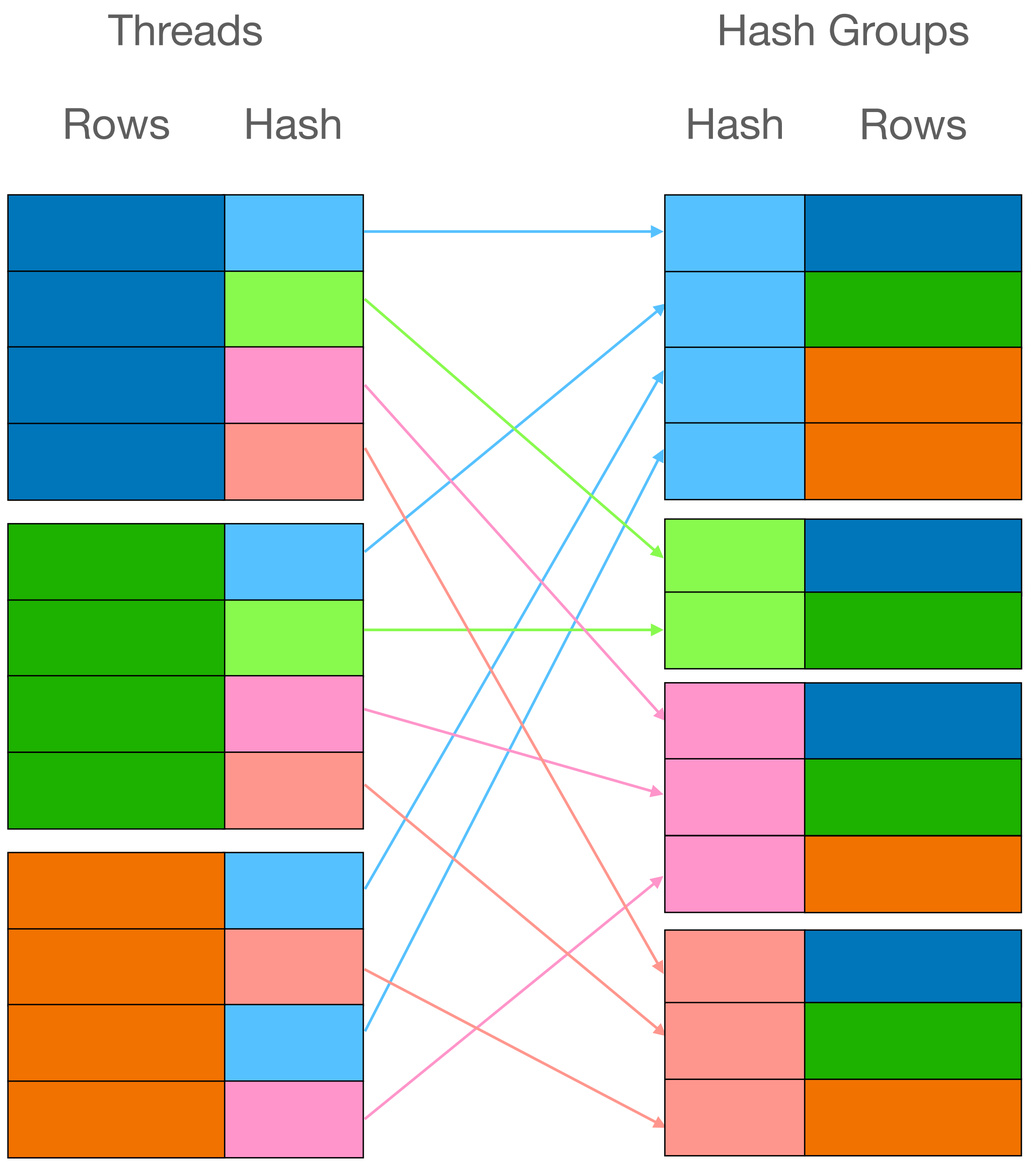
Streamed Windowing

- Can we **stream evaluation**?
  - No `PARTITION` or `ORDER`
  - No `IGNORE NULLS`
  - “Simple” non-aggregates
  - “Running total” aggregates

- Recent functionality
  - **`FILTER`** and **`DISTINCT`**
  - **`LEAD`** and **`LAG`** (close by)

Water streaming down a window

Hash Grouping



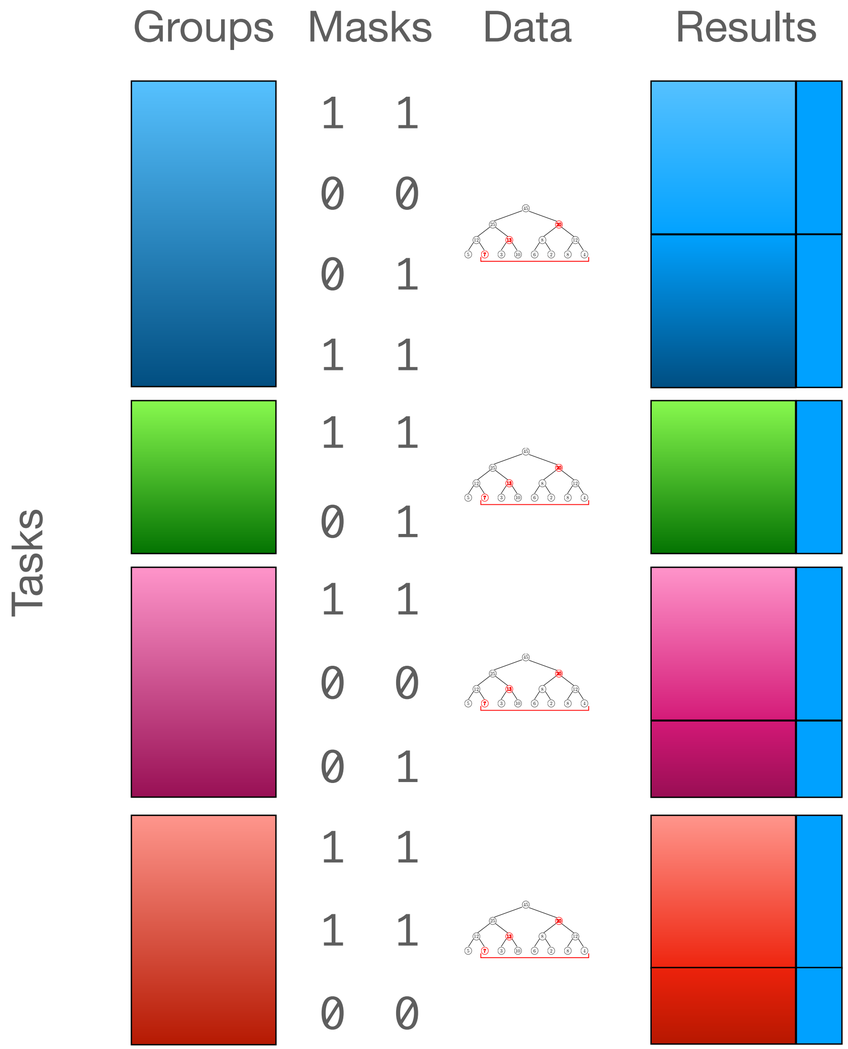
Hash Partitioning by Thread

- Hash chunks in **`Sink`**
  - Lies et al., VLDB 2015
  - Max of 1024 hash groups
- Only **hash the partition keys**
  - **Reduces sorting size**
  - `N * log (N/p)`
- Hash collisions?
  - Multiple partitions per group

Source Implementation

- Minimise memory footprint
  - **One partition at a time**
  - Use all threads
  - Size scheduling to largest
  - Smaller partitions can share

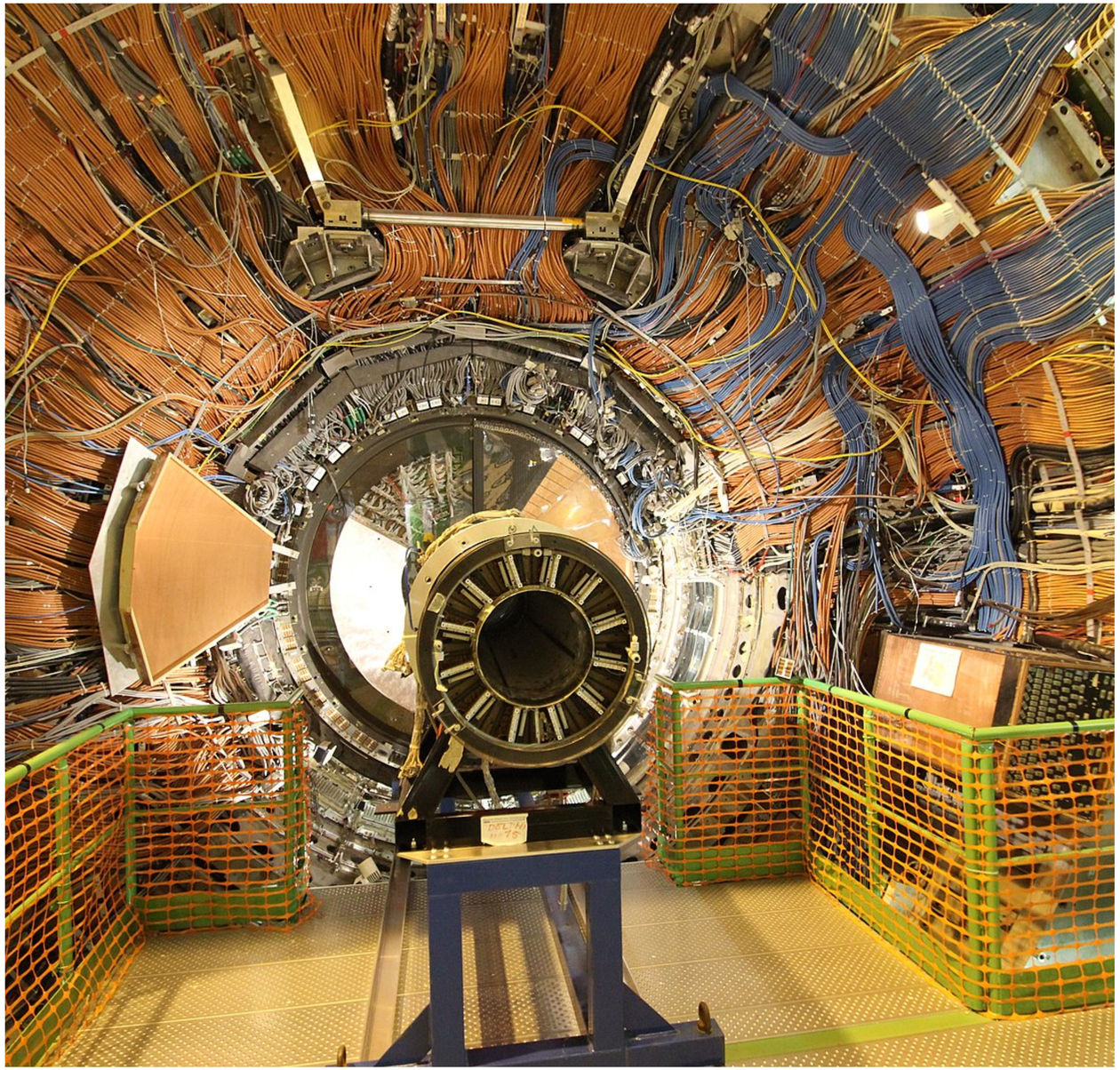
- Do **everything in parallel**
  - Build **acceleration** structures
  - Compute result chunks



Source Task Diagram

Function Evaluation

Aggregation Accelerators



CERN Particle Accelerator, Wikimedia Commons

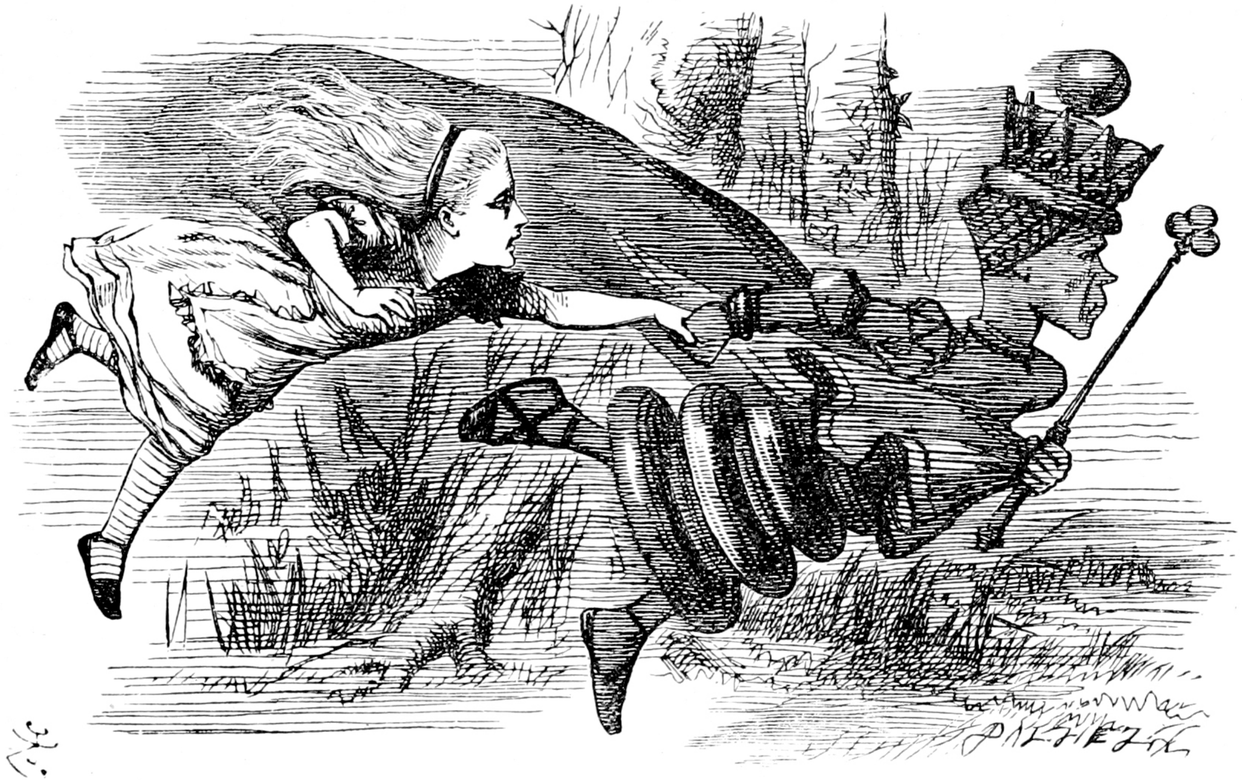
- Na ve evaluation is s l o w!
  - Independent row evaluation
  - No history reuse

- Accelerators
  - Segment Trees
  - Custom Window APIs
  - **Single Value Aggregation**
  - **Merge Sort Trees**
  - **Na ve** (for testing)

Single Value Aggregation

- Unsorted frames
  - No `ORDER BY`
  - Frame is **entire partition**
  - Only **one value** (“constant”)

- We detect this
  - Only **compute it once**
  - Copy to all rows
  - Often constant vector



Alice and the Red Queen, John Tenniel, Public Domain

Merge Sort Trees

- Window has frame order

- Aggregate needs another order
  - **`DISTINCT`** arguments
  - **`quantile`**/**`mad`**
  - Order-sensitive (`first`, …)

- **Doubly ordered tree**!
  - Built and queried in parallel

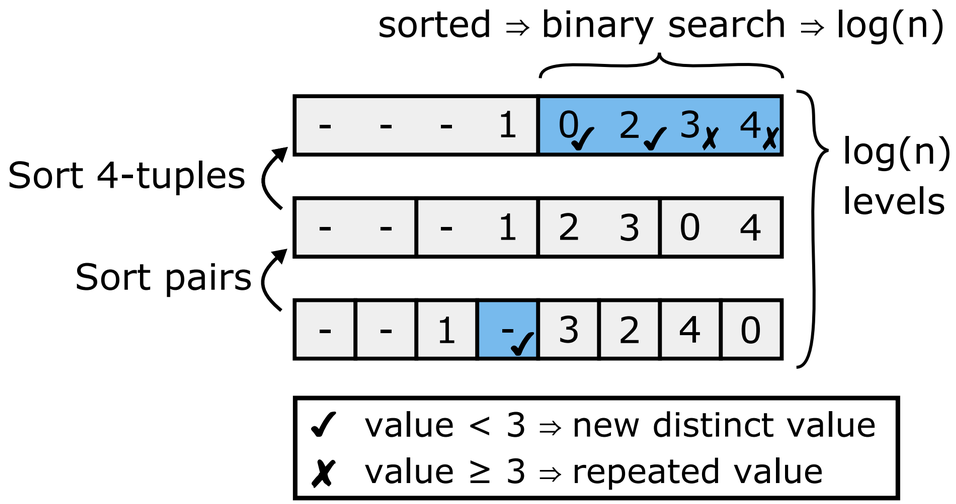


Figure 2: A merge sort tree improves query time to $O(n(\log n)^2)$ by utilizing a tree of sorted lists.

Basic Merge Sort Tree, from Vogelsgesang et al.

Questions?

- Why is a **duck** staring at me through a plane **window**?