Solving Space Issues in Space

Project for the Advanced Big Data Module in CAS Machine Intelligence 2025

By Fredrik Bäckman

Code and Dataset: https://github.com/backmanai/market_data_compression

[Image Source: Gemini AI]

Data Context: Eve Online – a massive multiplayer online game

- **PLAYER-DRIVEN ECONOMY**
  - Nearly everything manufactured, mined, and traded by players
  - Complex supply chains spanning entire star systems
  - Real economists study EVE's market dynamics
  - In-Game currency ISK and Plex. 500M ISK <- 100Plex <- $5 (note: only one direction)
  - $24+ million USD in circulating currency alone (total economy value likely $300M+ including all assets)

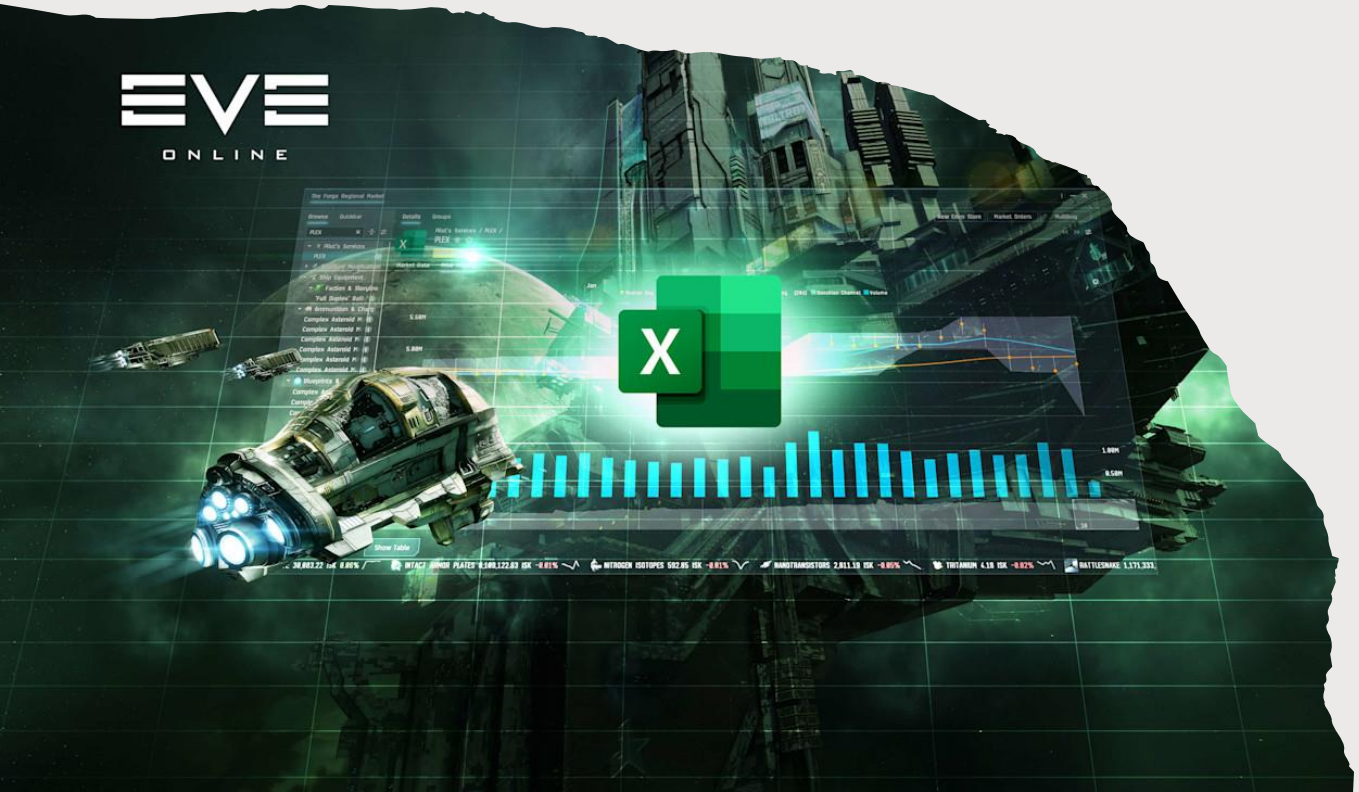
  (https://www.eveonline.com/news/view/monthly-economic-report-september-2025)

- **MASSIVE SCALE**
  - Single-shard universe with 7,000+ star systems
  - Battles involving thousands of simultaneous players
  - Largest battle (M2-XFE, 2021): 6,500+ pilots, losses valued at ~$378,000 USD

- **LONGEVITY**
  - Launched May 2003 (20+ years running)
  - Continuous updates and expansions

[Image Source: www.eveonline.com

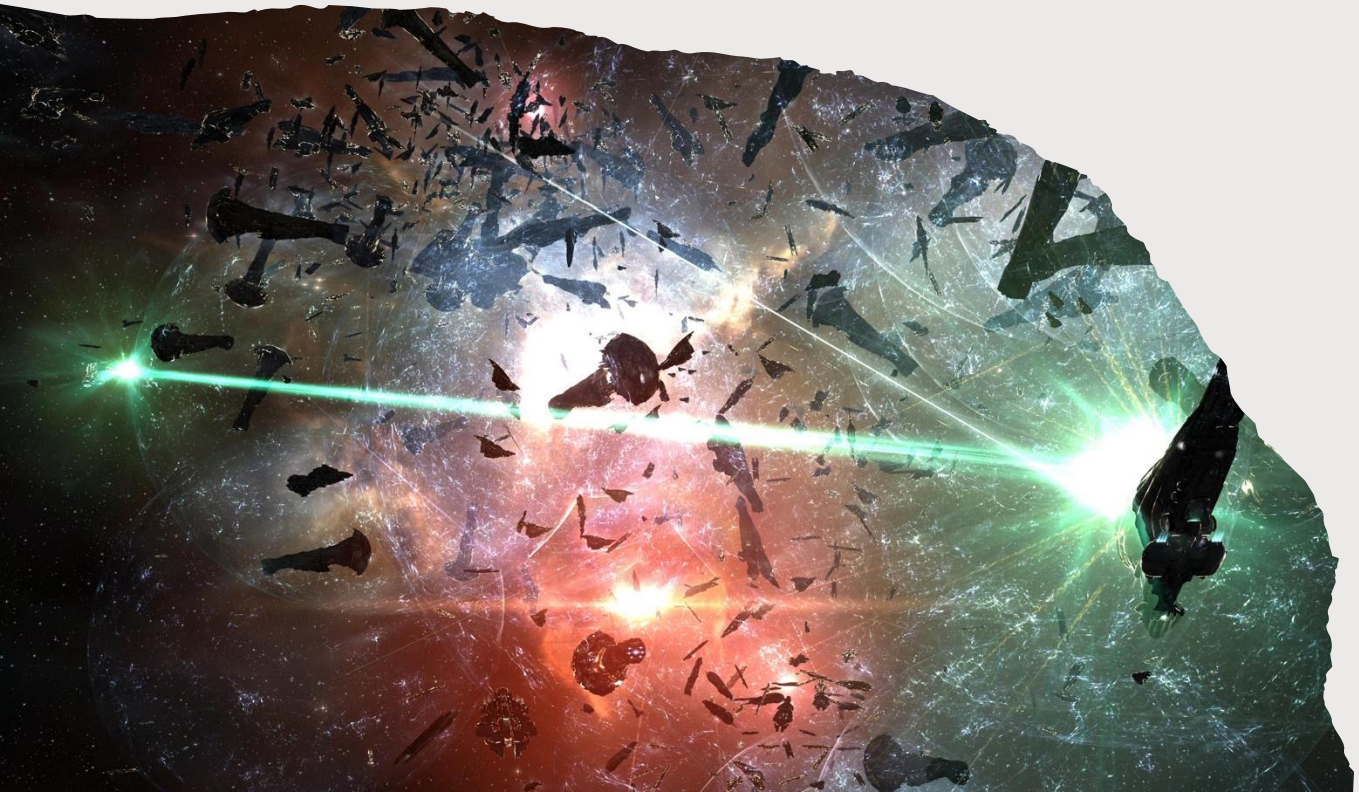
Data Context: Eve Online – spreadsheets in space



- **INDUSTRY & ECONOMY COMPLEXITY**
  - **Production Chain Tiers:**
    - **Multi-tier manufacturing**: Production chains can go 5+ levels deep
    - Raw materials (ores, gases, planetary resources)
    - Processed materials (minerals, P1 planetary commodities)
    - Intermediate components (P2, P3 planetary commodities)
    - Advanced components (P4, moon reaction products, T2 components)
    - Final products (ships, modules, ammunition)
  - **Material Sources:**
    - **Mining**: 16+ ore types producing various minerals
    - **Moon Mining**: Raw moon materials → Processed materials → Complex materials
    - **Planetary Industry**: 4-tier system (P0→P1→P2→P3→P4)
    - **Gas Harvesting**: Multiple gas types for booster and T3 production
    - **Reactions**: Combining materials to create advanced components
  - **Manufacturing Types:**
    - **Tech 1**: Simpler production using mainly minerals
    - **Tech 2**: Requires invention process + moon materials + planetary materials + T1 base items
    - **Tech 3**: Requires wormhole materials + complex reactions
    - **Capital ships**: Require components from multiple production chains
    - **Supercapitals**: Multi-month build times with dozens of component types (multi-year blueprint research!)
  - **Data Accessibility**
    - **Static Data Export (SDE)**: Complete game database dumps
    - **ESI API**: RESTful API for real-time character, market, and universe data
    - **Excel**: Official Excel plugin available

[Image Source: www.eveonline.com

Data Context: Eve Online – real world headlines



[Image Source: www.eveonline.com

- **Bloodbath of B-R5RB** (2014)
  - Biggest battle in gaming history, $300,000+ in losses
    - https://www.eveonline.com/news/view/the-bloodbath-of-b-r5rb
    - https://www.bbc.com/news/technology-25944837
    - https://en.wikipedia.org/wiki/Battle_of_B-R5RB

- **Guiding Hand Social Club Heist** (2005)
  - Player spent 10+ months social engineering and climbing ranks in a coporation.
  - $16’000 stolen
    - https://www.pcgamer.com/murder-incorporated-ten-months-of-deception-for-one-kill-in-eve-online/
    - https://www.guinnessworldrecords.com/world-records/88183-most-hostile-corporate-takeover-in-eve-online

- **Academic papers**
  - EVE Online is not for Everyone: Exceptionalism in Online Gaming Cultures
    - **https://minds.wisconsin.edu/bitstream/handle/1793/79143/11460%20OshScholar18%20Smith.pdf?sequence=1&isAllowed=y**
  - EVE: Online as a Potential Microeconomic Model
    - https://www.researchgate.net/publication/337645762_EVE_Online_is_not_for_Everyone_Exceptionalism_in_Online_Gaming_Cultures
  - Minerals, Titans, and Connections: The Political Economy of Empire in the World of EVE Online
    - **https://www.cambridge.org/core/books/abs/invisible-hand-in-virtual-worlds/minerals-titans-and-connections-the-political-economy-of-empire-in-the-world-of-eve-online/D35784F90EF7098ACD4E8582F6F77920**
  - **…**

Business Goal

- Understanding market prices and real trading volumes is key to industrial production planning and market trading.
- The game does not publish the buy and sell volume and prices directly.
- The games does publish the currently open orders through a REST API.

- Purpose:
  - Capture the order book snapshots and store the data for further analysis
  - Extract market insights

Data Source

- **API: List orders in a region**
  - https://developers.eveonline.com/api-explorer#/operations/GetMarketsRegionIdOrders
- https://esi.evetech.net**/markets/{region_id}/orders**
- region_id: 10000002 (The Forge – most active trade region)
- current snapshot is cached for 5 minutes

```
Loaded 402,300 market orders

Sample order structure:
{
  "duration": 90,
  "is_buy_order": false,
  "issued": "2025-11-15T20:13:29Z",
  "location_id": 60003760,
  "min_volume": 1,
  "order_id": 7186939904,
  "price": 232700.0,
  "range": "region",
  "system_id": 30000142,
  "type_id": 570,
  "volume_remain": 3,
  "volume_total": 5
}
```

```
JSON file size: 120.93 MB (126,801,130 bytes)
Average bytes per order: 315.2 bytes
```

Technical Challenge

- Problem:
  - The market order api returns about 400k open orders for this region.
  - The raw json output is huge (120MB)
- Solution:
  - Compression

Stage 1: JSON to Parquet

```
Compression Results (Stage 1: JSON → Parquet)
============================================================
Raw JSON:          120.93 MB
Parquet:             9.52 MB
Compression:         12.7× reduction
Avg per order:       24.8 bytes (Parquet)
                  vs 315.2 bytes (JSON)
============================================================
```

1. Dictionary Encoding:

- Replaces repeated values with small integer references
- Example: If the region_id appears 400k times, store it once + 400k tiny references
- Extremely effective for fields with limited unique values (type IDs, locations, buy/sell flags)

2. Snappy Compression:

- Fast, general-purpose compression algorithm by google, applied after encoding
- Prioritizes speed over maximum compression (still achieves 2-4× reduction)
- Unlike gzip/zstd, optimized for decompression speed in analytics workloads

3. Type Efficiency:

- Binary encoding vs. text representation
- Example: The number `*123456789*`
  - **9 bytes** in JSON (ASCII) **4 bytes** in Parquet (32-bit integer)
- Timestamps, floats, booleans all stored in compact binary form

Better but not enough…

288 snapshots per day at 9.5MB still results in 2.7GB data per day.

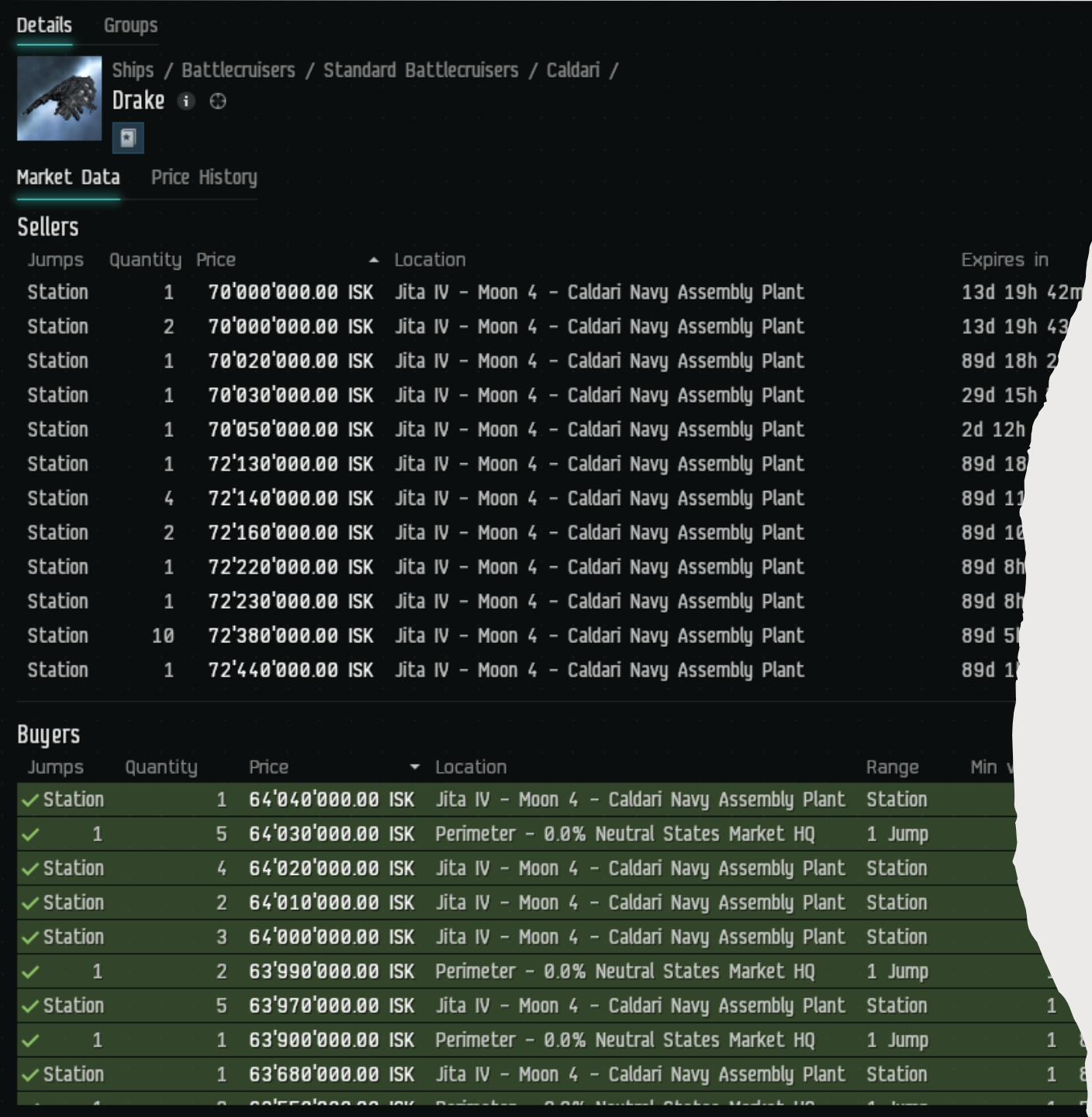
Options A) limit the order book history.

Options B) sign up for a cloud storage and help Jeff become a trillionaire.

Option C) compress further

I choose option C.

[Image Source: gemini AI



Most orders don’t change

Within a 5-minute interval, most orders remain unchanged.

Trading happens on the best priced sell and buy orders.

Occasionally a trader may update the order price stay competitive.

Occasionally orders expire or are cancelled by the trader.

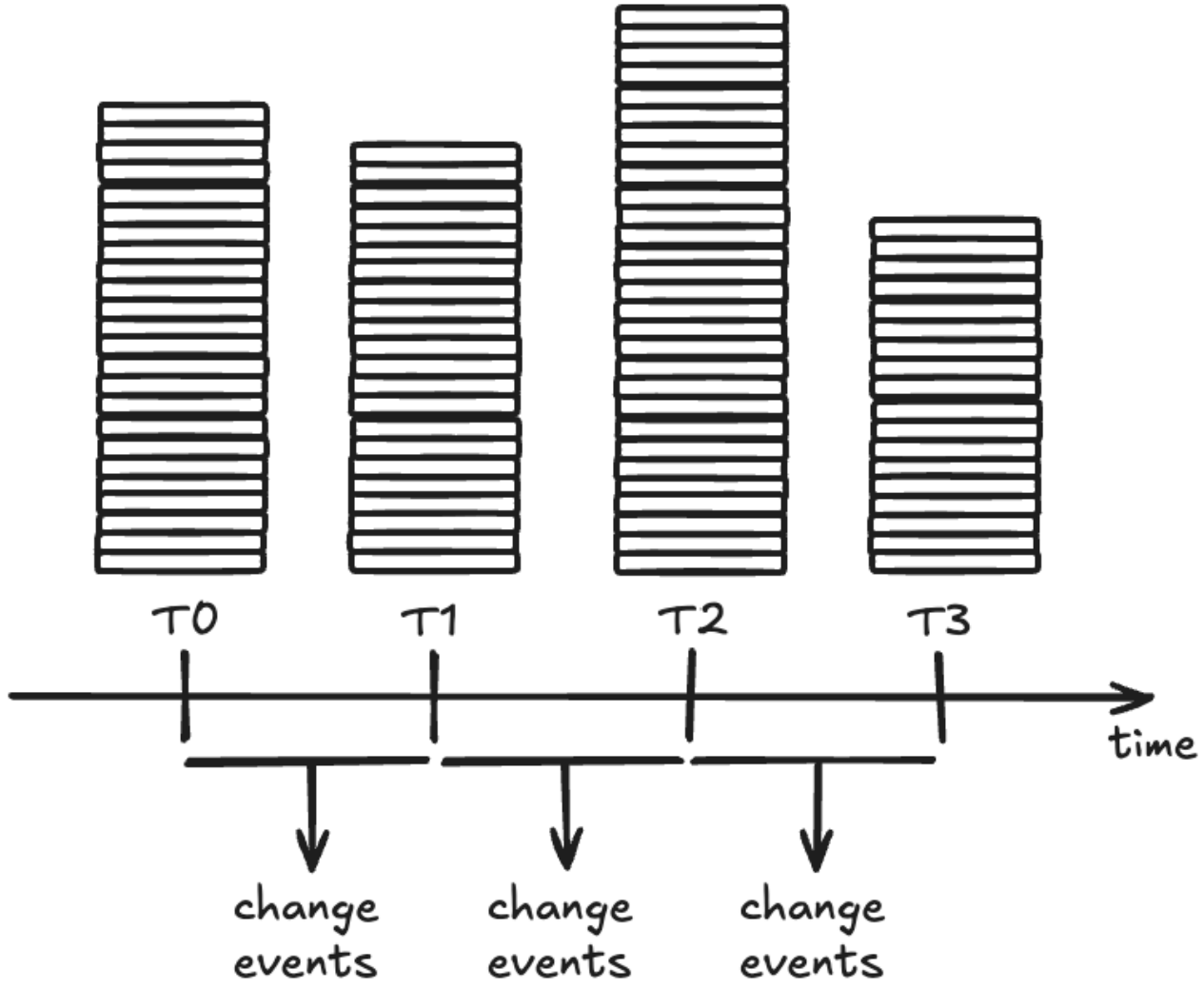
[Image Source: www.eveonline.com

Capture changes

Apply the “event sourcing” pattern.

Event sourcing persists the state of a business entity such as an order as a sequence of state-changing events.

https://martinfowler.com/eaaDev/EventSourcing.html



Change Event Types

The first snapshot which is processed will establish the baseline by initializing each order as “order_opened” event.

Event Type	Trigger
TRADE	Volume reduced
ORDER_OPENED	New order appeared
ORDER_CLOSED	Order disappeared with volume=0
ORDER_CANCELLED	Order disappeared with volume>0, not expired
PRICE_CHANGED	Price modified
ORDER_EXPIRED	Natural expiration time reached

Processing one interval

```
Loaded 2 consecutive snapshots:
   T0: region_10000002_2025-10-22T07-00-00+00-00.parquet
   T1: region_10000002_2025-10-22T07-05-00+00-00.parquet

Snapshot sizes:
   T0: 8.52 MB
   T1: 8.52 MB
   Total: 17.05 MB for 2 snapshots
```

```
Phase 1: Initialization
   Processing first snapshot at 2025-10-22 07:00:00
Initializing event log from snapshot: 375,338 orders

Initialization Results:
   Orders in T0:        375,338
   ORDER_OPENED events: 375,338
   Ratio:               1 event per order (baseline)
```

```
Phase 2: Delta Extraction
   Comparing T0 → T1 (5 minute interval)

Delta Extraction Results:
   Orders in T1:        375,320
   Delta events:        1,023
   Change rate:         0.3% of orders changed

Delta event types:
   trade                514
   price_changed        221
   order_cancelled      149
   order_opened         135
   order_closed         4
```

Extrapolate to one day data volume

```
Daily Event Volume:
======================================================================
Initialization (once/day):                                  375,338 events
Delta events (287 intervals at approx. 1023 each):         293,601 events
Total daily events:                                         668,939 events
======================================================================

Daily Storage Comparison:
======================================================================
Naive (keep all snapshots):     2454.57 MB/day
Event sourcing (Parquet):         16.26 MB/day
Compression ratio:                150.9×
======================================================================

Key Insight:
   After initialization, only 1,023 events per 5-min interval
   = 0.27% of baseline data per snapshot
   = Storing changes is 150.9× more efficient!
```

Conclusion

```
======================================================================
COMPRESSION DEMONSTRATION SUMMARY
======================================================================

Stage 1: JSON → Parquet (Columnar Storage)
   Raw JSON:                  120.93 MB
   Parquet (Snappy):          9.52 MB
   Compression:               12.7×

Stage 2: Event Sourcing (2 Snapshots)
   Two snapshots (T0 + T1):   17.05 MB
   Events (init + delta):     9.15 MB
   Compression:               1.9×

   Events breakdown:
      Initialization:         375,338 events (baseline)
      Delta (5 min):          1,023 events (0.27% changed)

Daily Projection (288 Snapshots)
   288 JSON files:            34,827 MB/day
   288 Parquet files:         2,455 MB/day (14.2× vs JSON)
   Event sourcing:            16 MB/day (150.9× vs Parquet)

   Total compression:         2141.7× (JSON 28965MB → Events 15MB)
```