

# N-diff for streaming trade logs

Reconcile trade log entries arriving in independent streams.

Copyright Marcus Schwartz <[marcus@marcus.net](mailto:marcus@marcus.net)> 28-Feb-2021

Licensed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

## Record Format

Each record **O\_R** will consist of at least the following fields:

- **TS** - Timestamp (YYYY-MM-DD HH:MM:SS.ssssss) in UTC
- **T\_ID** - Trade ID, unique to each trading transaction
- **RF\_\*** - One or more fields that will be compared for reconciliation.
- **DF\_\*** - One or more fields that will be included in output for debugging purposes.

## Input

Streams are assumed to be iterators that can be reset to a specific point in time. There may be an arbitrary number of them and they should be non-blocking.

To simplify debugging, each stream origin will be referred to with a unique identifier **O\_ID**.

## Inter-stream Jitter

There may be an inter-stream jitter of up to **J** seconds for records that are otherwise identical. Any time skew within this window should be considered a non-difference.

An outside jitter window of **E** seconds ( $E > J$ ) will be used to limit memory usage and raise alarms if an input stream is blocked.

## Output

The system will note any differences between streams as a record **D\_R** in an output stream. This output stream will include the **O\_ID** and **O\_R** for each stream. Each output record will include a list of inconsistent fields **DF**. If an input record is missing, the inconsistent field set will include "\_missing".

For cases where the inter-stream jitter is greater than **E**, the output stream will include two or more **D\_Rs** each with one or more "\_missing" DFs and null **O\_Rs**.

## Example

T\_ID, [DF, ...], [ O\_ID.0, O\_R.0 ], [ O\_ID.1, O\_R.1 ], ..., [ O\_ID.N, O\_R.N ]

# Logic Overview

## Data Structures

- **nextRecords** - A list of the next log record from each log stream, sorted by timestamp.
- **pendingTrades** - A dict of mappings from unreconciled trade IDs to a (sometimes partial) list of associated log records from each log stream.
- **pendingRecords** - A list of each unreconciled log record previously read from each log stream in the past E seconds, sorted by timestamp.

## Reconciliation

1. Compare the TS and RF\_\* fields in each of the log records for an individual trade.
  - a. If any do not match, emit D\_R records.
2. Remove the related entries from pendingTrades and pendingRecords.

## Main Loop

1. Reconcile any pendingTrades that have pendingRecords with a timestamp that is older than E
  - a. Use the next TS from nextRecords as a basis for pendingRecords ages.
2. Attempt to populate nextRecords with a new record from any log stream that isn't currently represented in it.
3. Fetch the next record from nextRecords
  - a. Attempt to backfill nextRecords from whichever log stream it originated from.
  - b. Add it to pendingRecords
  - c. If the trade is not in pendingTrades, add it and then add the record to it.
  - d. Otherwise, add the record to the existing trade in pendingTrades.
  - e. If the trade in pendingTrades has records from all log streams, reconcile it.

## Reliability Considerations

### State Persistence

State required for restarting this service should be persisted externally on a regular basis (~1 minute of data?). This state should include the most recently read timestamp from each log stream.

### Restarting

When restarting, each log stream should be rewound to the stored timestamps mentioned above, minus E seconds. This may result in some duplicate reconciliation output but should ensure that no reconciliation is missed.

## Instrumentation

Metrics should be emitted into an external store on a regular basis (~1 wallclock minute or ~1000 log records?). These metrics should include:

- Process runtime
- Resource consumption (cpu, memory, ...)
- The count of “good” and “not good” reconciled trades.
- The count of each DF value.
- A histogram of timestamp jitter ( $\text{max\_ts} - \text{min\_ts}$ ) for “good” and “not good” reconciled trades.
- The count of records read from each log stream.
- A histogram of record read latencies for each log stream.
- The latency of writes to the external state persistence store.