

CloudTurbine File Indexing

Matt Miller, Cycronix, 02/23/2018

Synopsis: An in-memory index of file-pointers enhances CloudTurbine read-performance.

A fundamental tenet of CloudTurbine (CT) is that all streaming data is stored in a sequence of regular files. This is convenient and efficient for a data source, as it enables writing new data at intervals, as it arrives, without need to keep track of prior data nor apply on-the-fly conversions or formatting. Multiple data channels (e.g. sensors A,B,C) from a common source can be multiplexed per file, as shown in Figure 1.

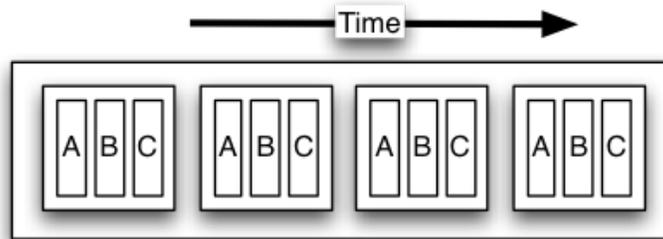


Figure 1: Time-Ordered Real-Time Streaming Data

On the other hand, a data sink (consumer, viewer) typically wants to access selected channels of data over time intervals. For example, to display a strip-chart of a channel (e.g. 'A'), a viewer requests a specified time-interval of data for a demultiplexed channel, such as shown in Figure 2.

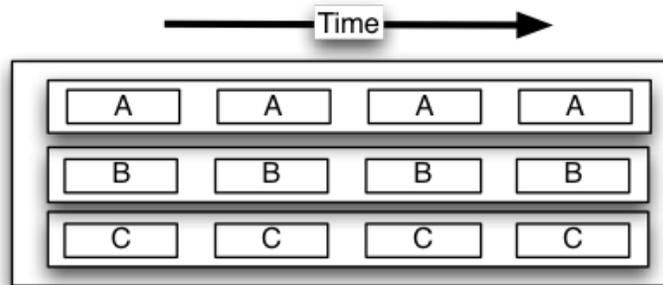


Figure 2: Channel-Ordered Recorded Playback Data

To get from time-ordered data (Fig 1) to channel-ordered data (Fig 2), a common approach is to convert from the recorded data structure to new files as a post-processing operation. This lack of a common format can make real-time and playback/analysis software incompatible. CloudTurbine, however, is designed to have a single common streamable data format, compatible for both live and playback viewers.

To efficiently provide channel-ordered data from time-ordered storage, the CloudTurbine “CTreader” API builds an in-memory time/channel index to automatically convert from the source-preferred (time-ordered) format to the sink-preferred (channel-ordered) format. This index is built on-demand as new streaming data is added.

The bottom diagram of Figure 3 shows the resulting channel ordered “view” into the time-ordered data, without having to physically demultiplex the data, leaving the source-preferred, streamable (CT) data

structure intact on-disk. The index takes a small amount of memory, consisting of light-weight pointers to the data. The result is a faster data-gathering operation, with simpler and more robust code, eliminating the need to search disk files for each data channel request.

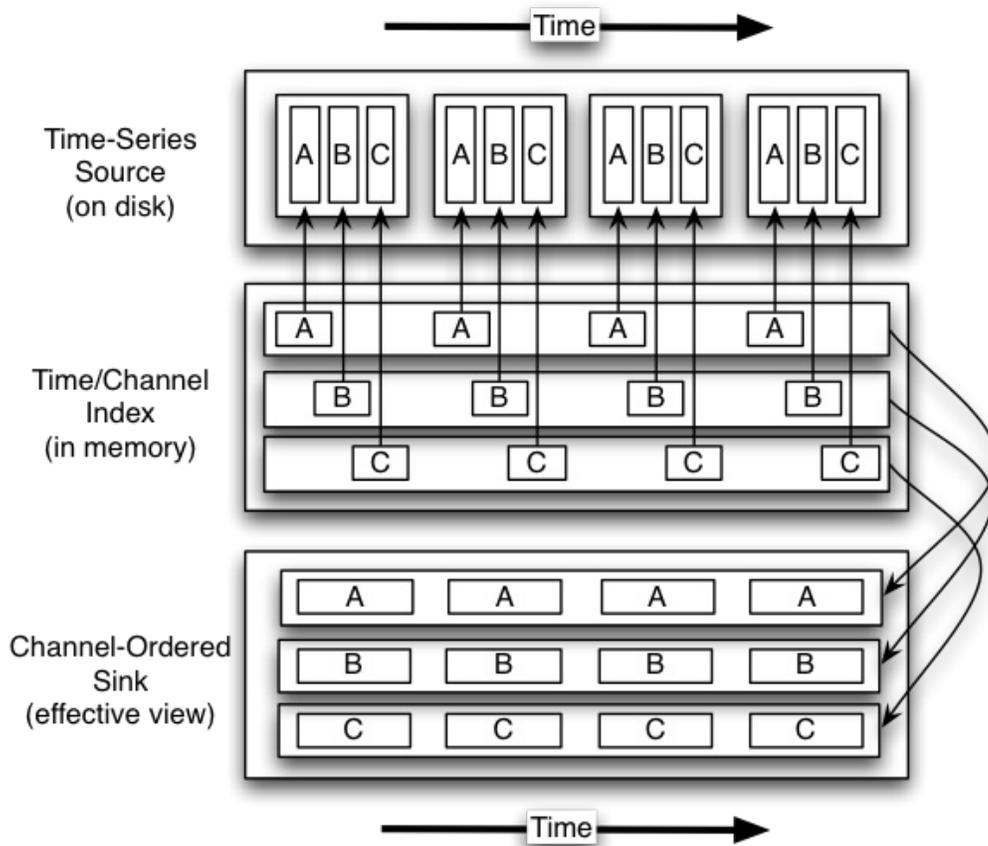


Figure 3: CloudTurbine API Time/Channel Index Efficiently Bridges Time-Ordered Source Data to Channel-Ordered Sink Requests

The performance benefit (faster data access) of this index-cache depends on the relative size and structure of the associated CT file archives. The benefit is minimal for a single channel with a large data-payload per CT file “block”, e.g. a sequence of JPEG images. The benefit is more significant (>10x) for multi-channel data in small (low-latency) CT blocks, e.g. unpacked numerical data. Thus, the in-memory CT data index significantly enhances low-latency data access for the most resource-intensive cases.