[HBASE-13071] Hbase Streaming Scan Feature

February 19, 2015

Eshcar Hillel (eshcar@yahoo-inc.com)

Motivation

A scan operation iterates over all rows of a table or a subrange of the table. We focus on the synchronous nature in which the data is served at the client side. This factor hinders the speed the application traverses the data: it increases the overall processing time, and may cause a great variance in the times the application waits for the next piece of data.

The scanner next() method at the client side invokes an RPC to the regionserver and then stores the results in a *cache*. The application can specify how many rows will be transmitted per RPC; by default this is set to 100 rows.
The cache can be considered as a producer-consumer queue, where the hbase client pushes the data to the queue and the application consumes it. Currently this queue is synchronous, i.e., blocking. More specifically, when the application consumed all the data from the cache---so the cache is empty---the hbase client retrieves additional data from the server and re-fills the cache with new data. During this time the application is blocked.

Under the assumption that the application processing time can be balanced by the time it takes to retrieve the data, an *asynchronous* approach can reduce the time the application is waiting for data.

Proposal

We propose to make changes to the scan implementation such that hbase can demonstrate a *streaming scan* behavior, in the sense that the time an application waits for the next piece of data (row) is equivalent to reading local data, and the differences in times waiting for next() methods to retrieve data are significantly reduced.

Streaming is enabled by adding an *asynchronous scanner* which prefetches data from the server at the background while the application is still consuming data from the client cache.

This feature requires changes only at the client side (code and configuration settings).

Cline hbase-site.xml configuration

To set async scanner at the client side set
hbase.client.scanner.async.prefetch=true
(can also be set per scan at the the application level)

Client code

Main changes are reflected in the following classes

ClientScanner

Changed to be an abstract class: most of its code remains unchanged except for next(), and new abstract initCache() method.
The parts of next() that are common to sync and async scanner are extracted to a new prefetch() method.

ClientSimpleScanner [new class]

A new class which implements the **sync scanner** behaviour (inherits from ClientScanner). Essentially, this class maintains the original behavior of ClientScanner.

ClientAsyncPrefetchScanner [new class]

A new class which implements the **async scanner** behaviour (inherits from ClientScanner). Specifically, the cache used by this scanner is a concurrent queue which allows both the producer (hbase client) and consumer (application) to access the queue in parallel. This requires some synchronization. The prefetch is invoked when the cache is half-filled, instead of waiting for it to be empty.

Scan

Defines the new configuration property (and its default value) that defines which scanner to construct simple (sync) or async.