

### StoreFile

SortedList<KeyValue>

Sort(row,family,column,ts,type)

RegionUniqueID = Creation Stamp

Represents Puts and Deletes that occurred over a Range of Time [ID, 0)

### Generic Outline of Client-Side Processing a Get Request

Instantiate client-side GetX object

GetX(byte[] row, List<Family>, TimeRange, Filter)

Family(byte[] name, List<byte[] column>)

TimeRange(long tsMax, long tsMin)

### Generic Outline of Server-Side Processing a Get Request

Instantiate server-side GetXServer object

GetXServer(GetX)

Seek to start of GetXServer.getRow()

Iterate down KeyValues

For each KeyValue "cur",

GetXServer.compareTo(cur)

### Get Request

All GetX implement Get and  
Get.compareTo(KeyValue)

### GetServer.compareTo(cur) returns

- 0 if cur should not be included in result
- 1 if cur should be included in result
- 2 if cur is past this StoreFiles boundary for this Get
- 3 if cur is past this Stores boundary for this Get

### GetServer.compareTo(cur) implementation

if nothing left to match on this StoreFile, return 2

if nothing left to match on this Store, return 3

compare row

if next row, return 2

compare family

if next family (in the future), return 2

compare timestamp against

TTL

if expired, return 0

TimeRange

if not included, return 0

compare if cur is a Delete

if yes, add to NewDeletes and return 0

compare column

if past end of column match list, return 2

if does not match, return 0

if matches, check if deleted

if != 0 (deleted), return 0

if 0, Filter.match(cur)

if -1, return 2

if 0, return 0

if 1, return 1

### Server-side Handling of GetServer.compareTo() result

- 0 look to the next KeyValue
- 1 add cur to ResultSet<KeyValue>, move to next KV
- 2 go to next StoreFile
- 3 stop, return current ResultSet (early-out)

### DeleteSet

SortedList<KeyValue>

DeleteSet.compareTo(KeyValue)

Represents the Deletes that have been read from previous StoreFiles and the Memcache

### DeleteSet.compareTo(cur)

- !0 if cur has been deleted
- 0 if cur has not been deleted

### DeleteSet.addDeletes(List<KeyValue>)

Merges existing DeleteSet with deletes in specified list  
Removes any overlapping Deletes

### Filter

Black box that contains a match function that returns done/false/true (-1/0/1)

Filters are to be used in cases where there is little or no opportunity for early-out optimizations.

As the last check, they can be stateful to implement limits.

### Filter.match(cur)

- 1 if nothing else will match (done)
- 0 if cur should not be included in result
- 1 if cur should be included in result