



HBase Read Path

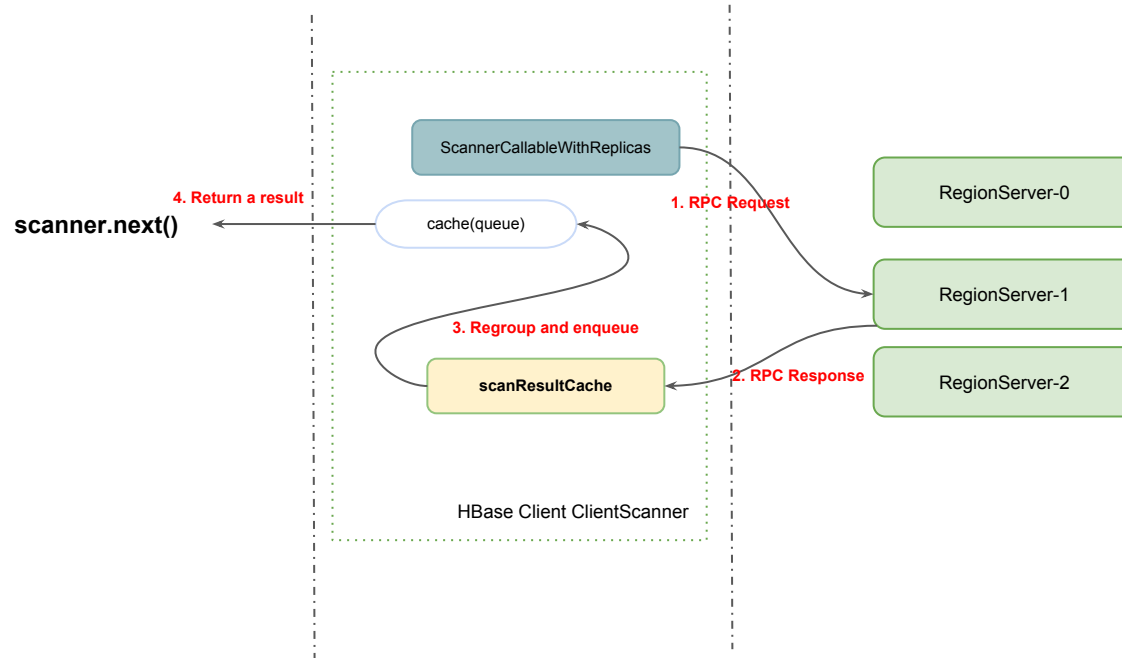
openinx@apache.org

Abstract

- ❑ Client Side
- ❑ Server Side
- ❑ Tuning

Part-1 Client Side

ClientScanner



Step.1 + Step.2 + Step.3 = loadCache

Concepts in Scan

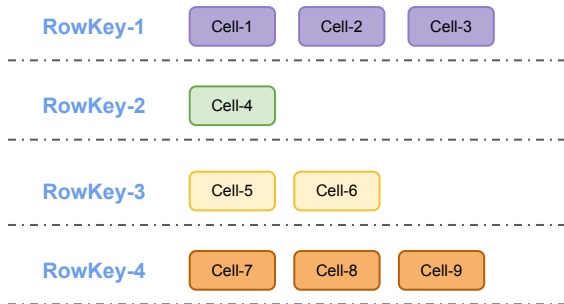
- caching
- batch
- `maxResultSize`
- `allowPartialResults`
- `limit`
- `maxVersion`
- `needCursorResult`
- `filter`
- `isolationLevel`
- `asyncPrefetch`



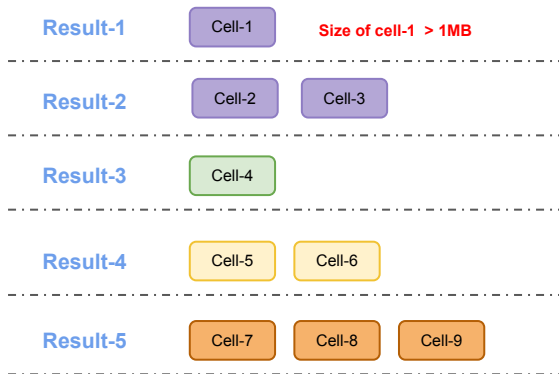
AllowPartialScanResultCache

`scan.setCaching(2).setAllowPartialResults(true).setMaxResultSize(1MB)`

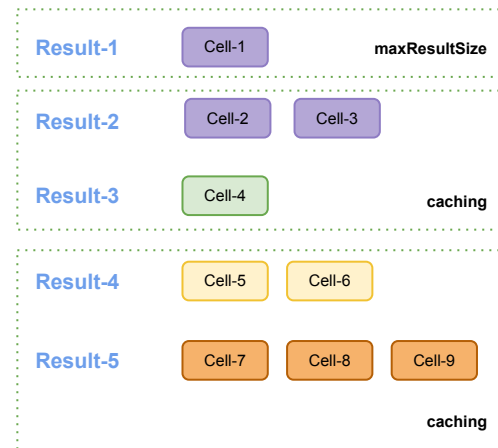
RegionServer Row Data



RPC Response Received from RS



Results get from scanner.next()



One load cache loop

maxResultSize

One load cache loop break because of maxResultSize limit

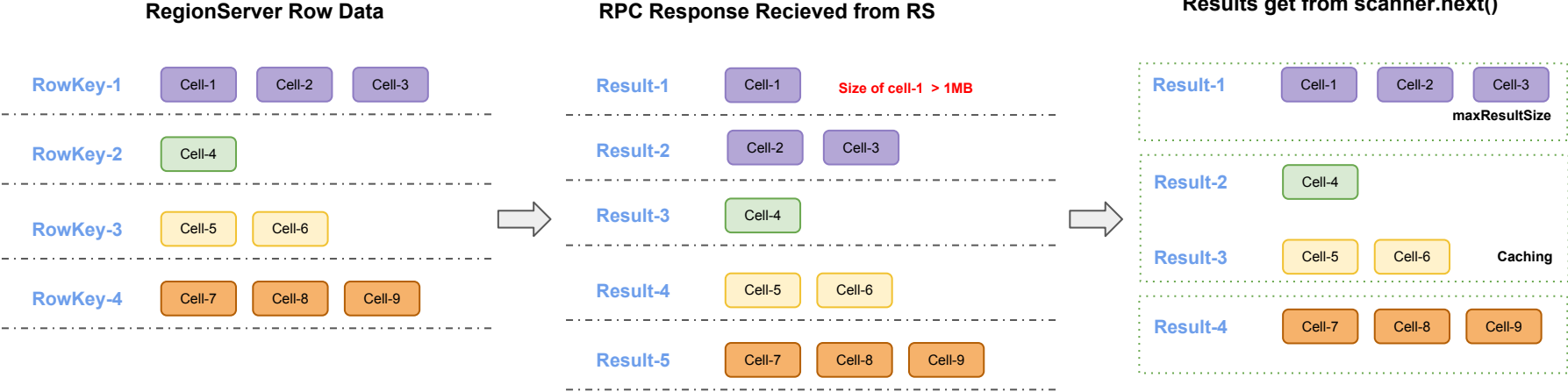
Caching

One load cache loop break because of caching limit



CompleteScanResultCache

```
scan.setCaching(2).setMaxResultSize(1MB)
```



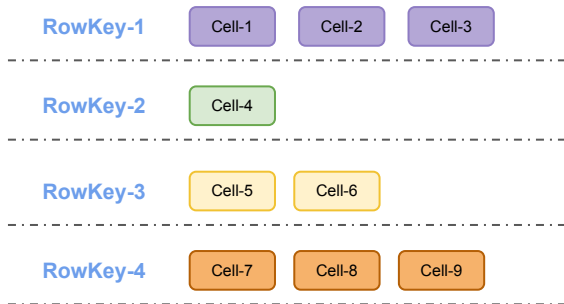
- One load cache loop**
- maxResultSize** One load cache loop break because of maxResultSize limit
- Caching** One load cache loop break because of caching limit



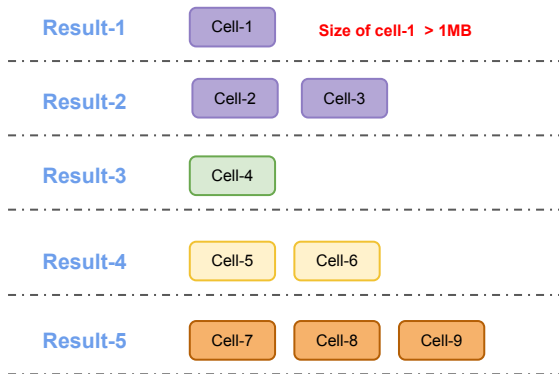
BatchScanResultCache

`scan.setCaching(2).setBatch(2).setMaxResultSize(1MB)`

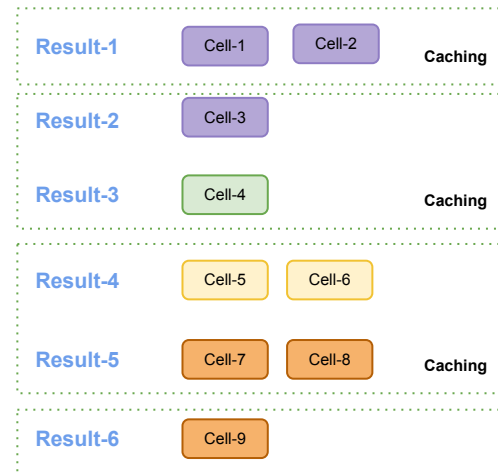
RegionServer Row Data



RPC Response Received from RS



Results get from scanner.next()



One load cache loop

maxResultSize

One load cache loop break because of maxResultSize limit

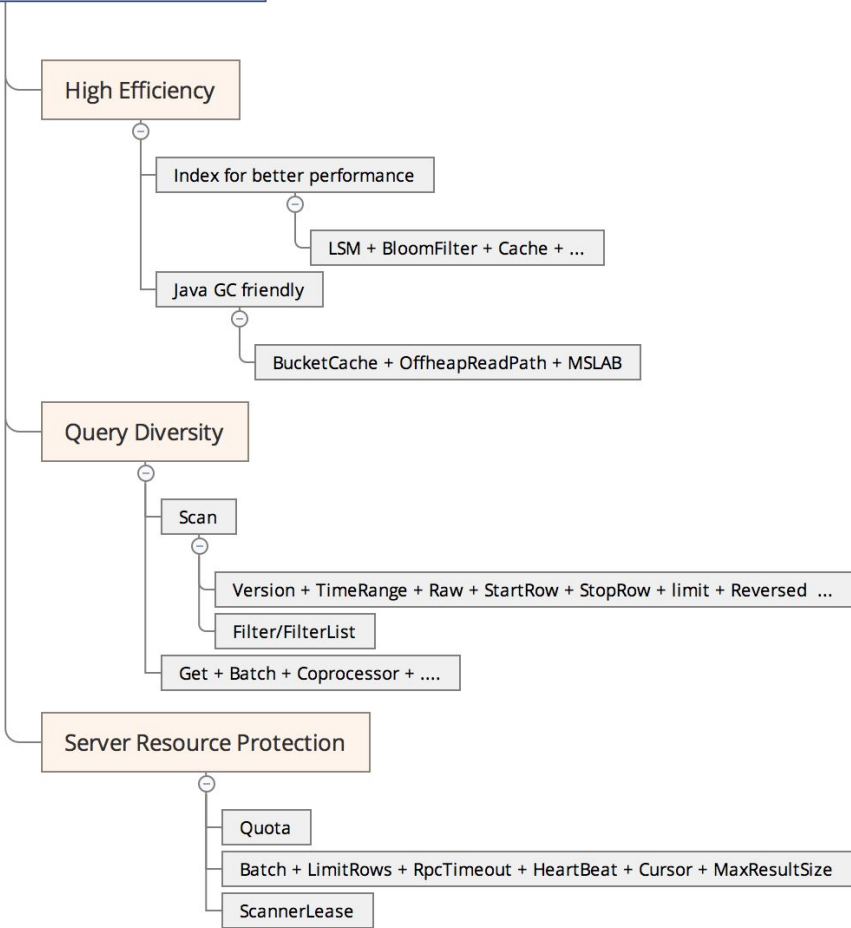
Caching

One load cache loop break because of caching limit

Part-2 Server Side



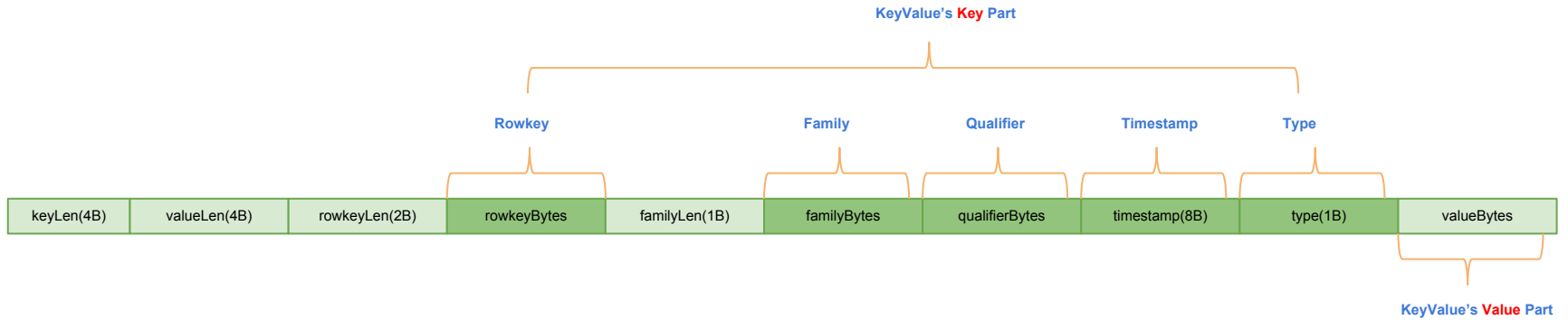
Read Path Design Purpose



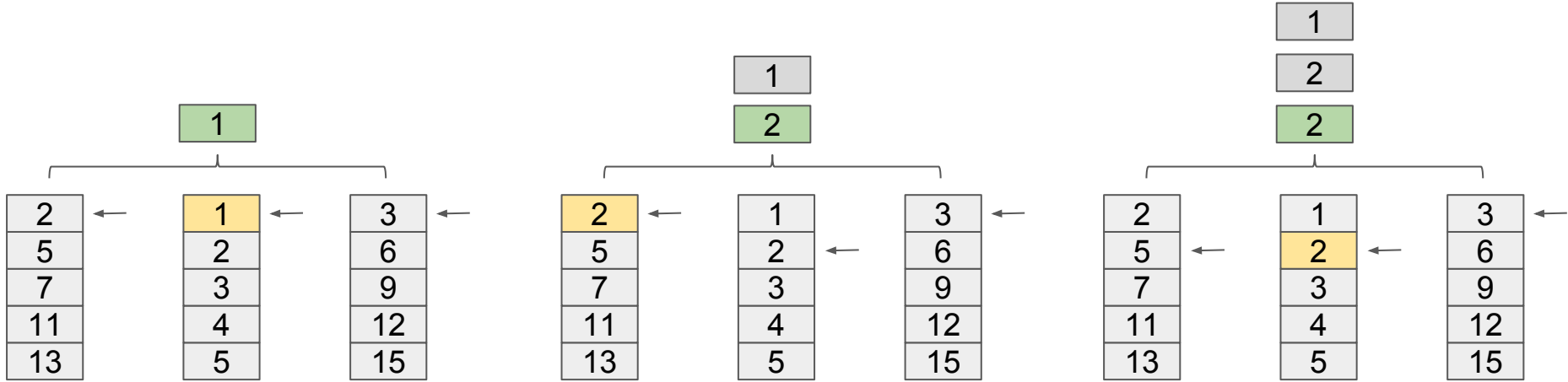
Abstract In Server Side

- ❑ **High Efficiency**
- ❑ Query Diversity
- ❑ Server Resource Protection

KeyValue / Cell

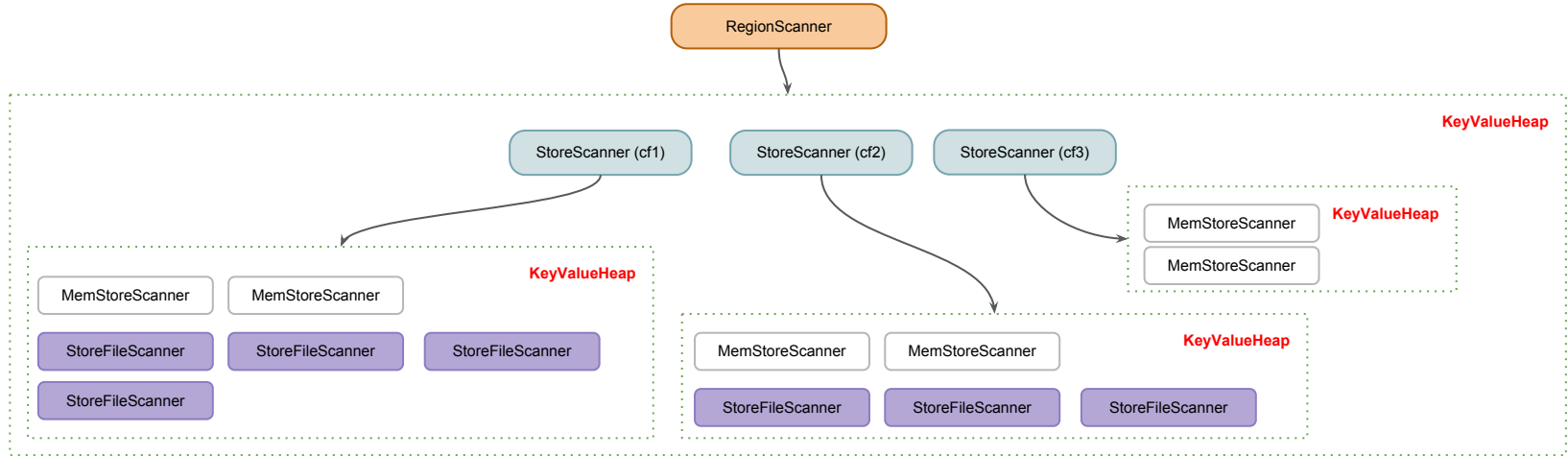


K-MergeSort

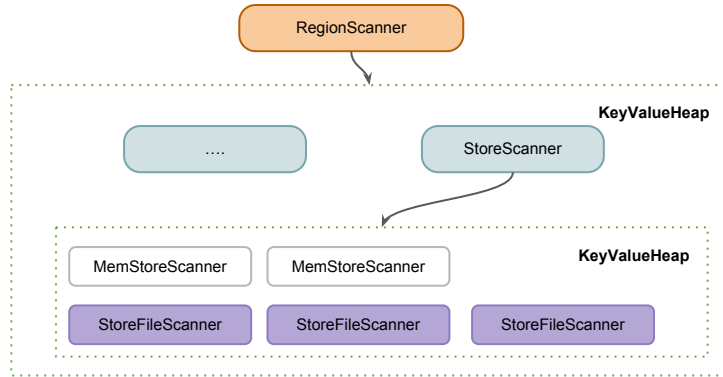


$A = \{N_1, N_2, \dots, N_k\}$, So the complexity is ?

RegionScanner



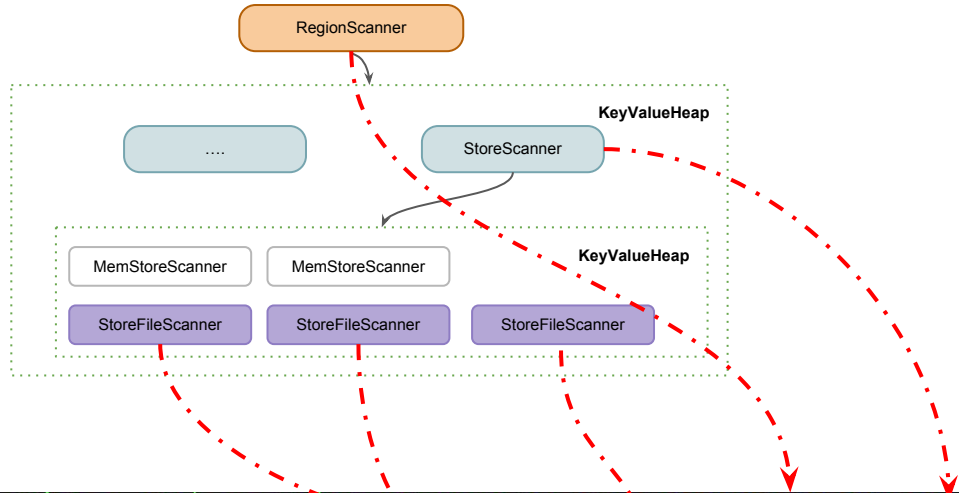
RegionScanner



```

$ ./bin/hdfs dfs -ls -R /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B
-rw-r-x----+ 3 hbase_tst supergroup 6248 2018-07-20 09:35 /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B/09a4a9e68c4845766dd8c146ef6dc4ef
-rw-r-x----+ 3 hbase_tst supergroup 728 2018-07-20 10:32 /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B/f938fbd8cbc745d58757ae79cf239644
-rw-r-x----+ 3 hbase_tst supergroup 1728 2018-07-20 11:20 /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B/ee8fbd8tbc745d58757ae79cf239644a
  
```

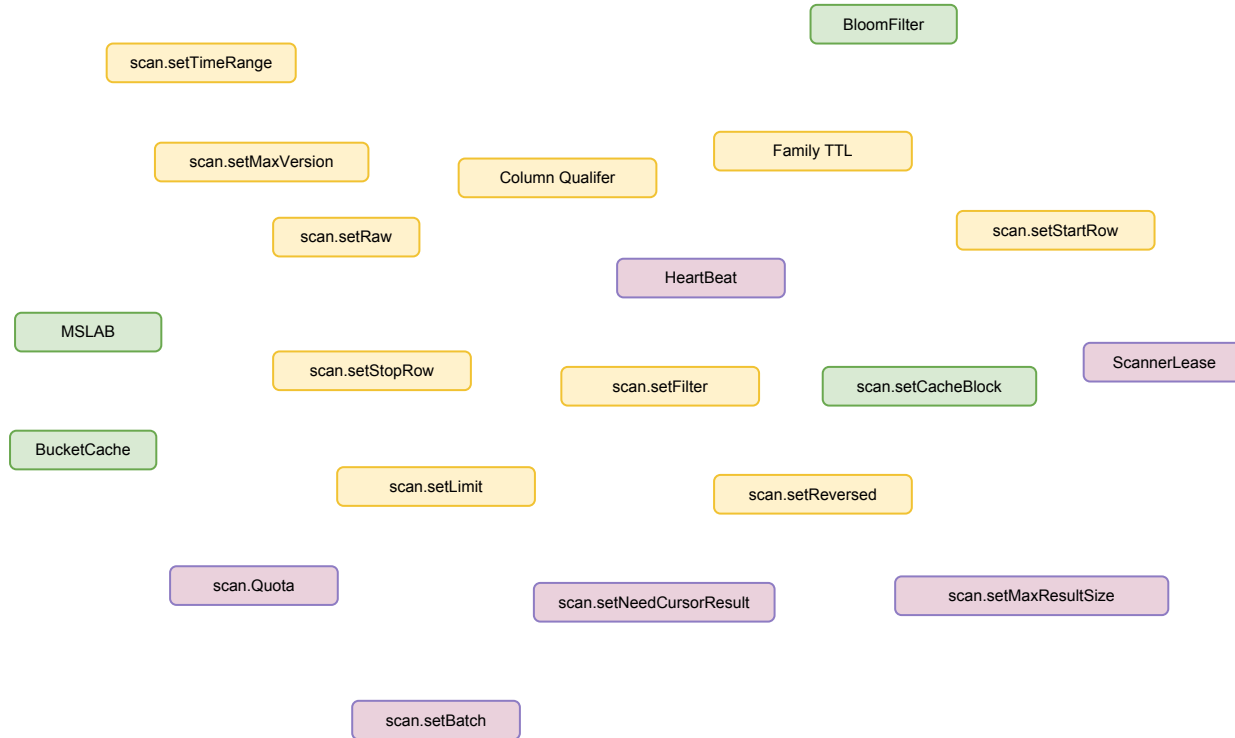
RegionScanner



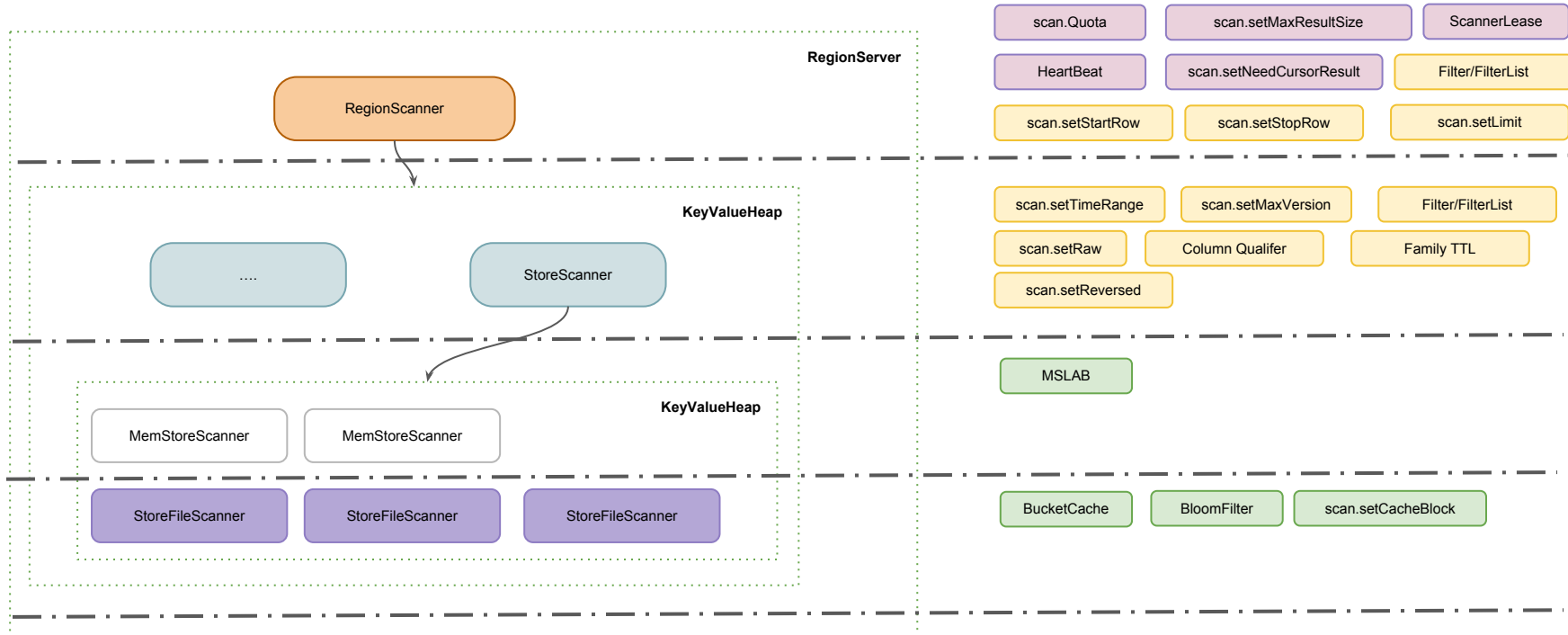
```

$ ./bin/hdfs dfs -ls -R /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B
-rw-r-x----+ 3 hbase_tst supergroup 6248 2018-07-20 09:33 /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B/09a4a9e68c4845766dd8c146ef6dc4ef
-rw-r-x----+ 3 hbase_tst supergroup 728 2018-07-20 10:32 /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B/f938fbd8cbc745d58757ae79cf239644
-rw-r-x----+ 3 hbase_tst supergroup 1728 2018-07-20 11:20 /hbase/test-hbase/data/mina/crawl_base/a09a4a9e68c4845766dd8c146ef6dc4e/B/ee8fbd8tbc745d58757ae79cf239644a
  
```

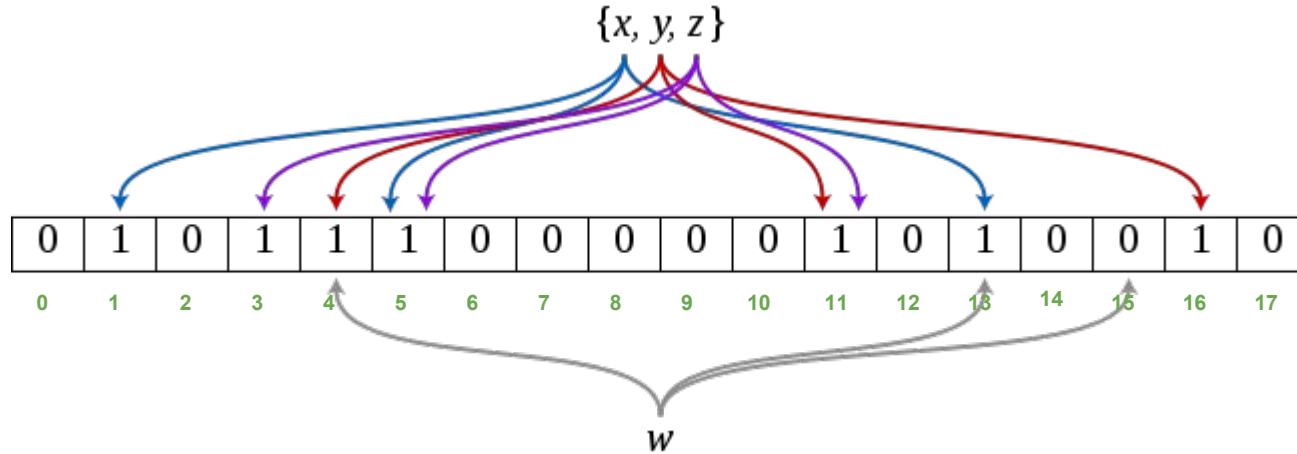

Revisit Concepts



Revisit Concepts



Bloom Filter

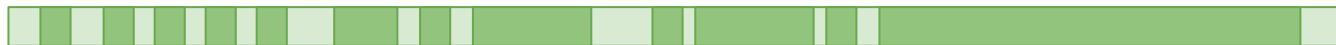


$f(w) =$ $\begin{cases} 1 : w \text{ may exist in the set.} \\ 0 : w \text{ does definitely not exist in the set.} \end{cases}$





LruCache + onheap

Before GC (G1)

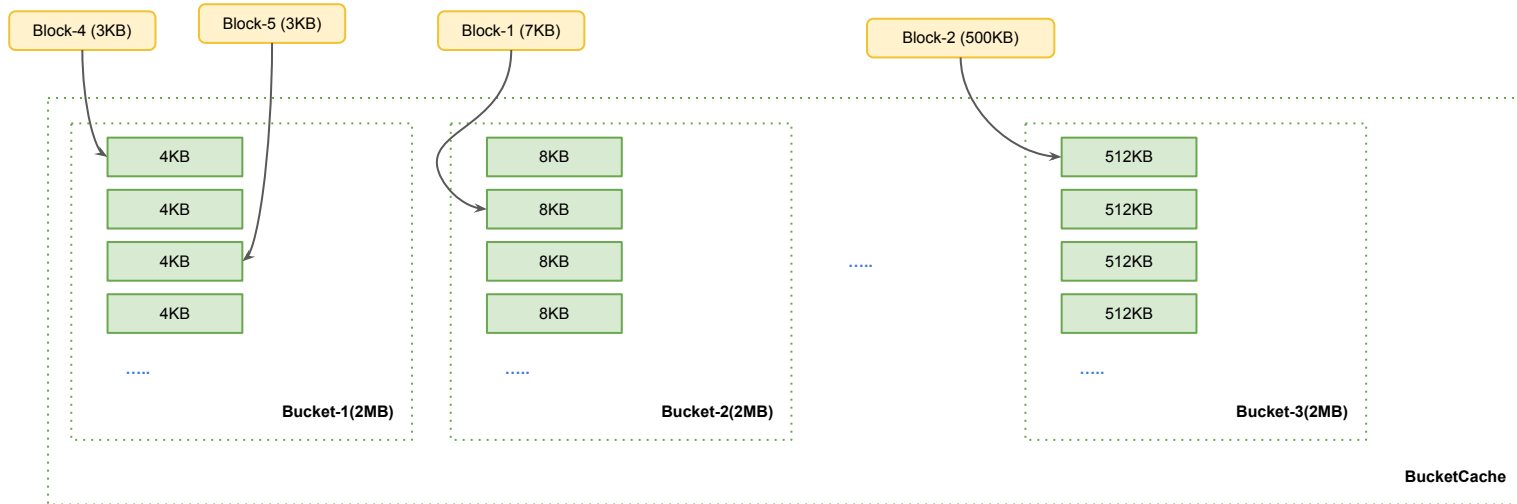


After GC (G1)

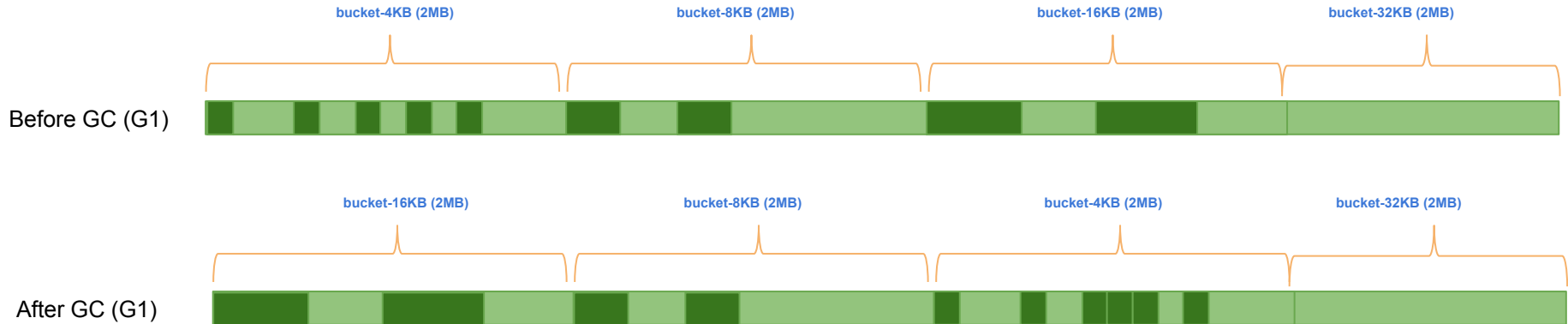


-  Memory allocated on Java heap
-  Free memory on Java heap

BucketCache



BucketCache + OnHeap



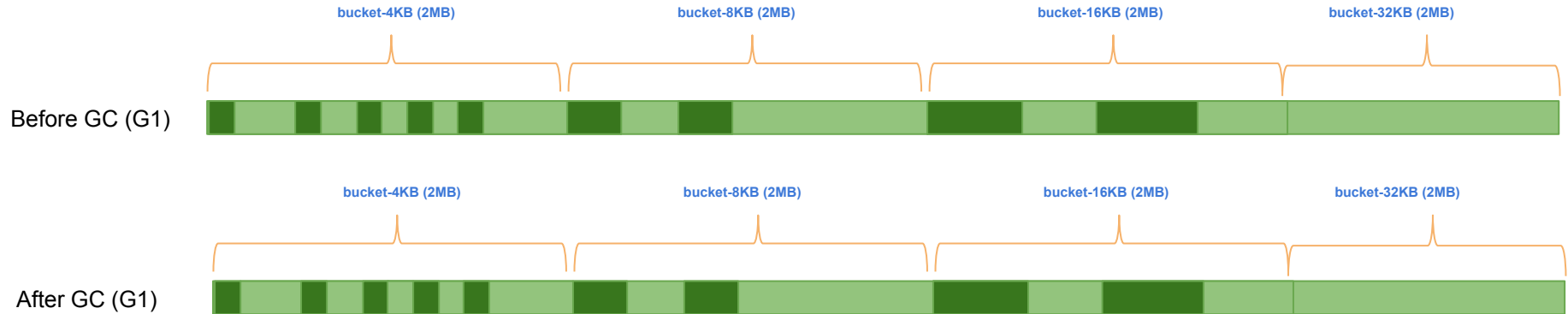
Memory allocated on Java heap and occupied by cache block



Memory allocated on Java heap and with no data.

- **Less fragment(Allocate 2MB one time) , Less Mixed GC.**
- JVM will never free BucketCache's byte buffer.
- But JVM will still sweep the buffer and **compact** them. (old generation)

BucketCache + OffHeap



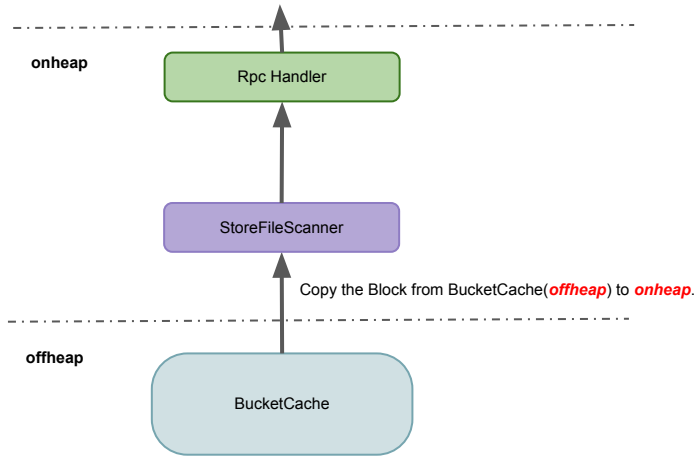
Memory allocated **on Java heap** and occupied by cache block



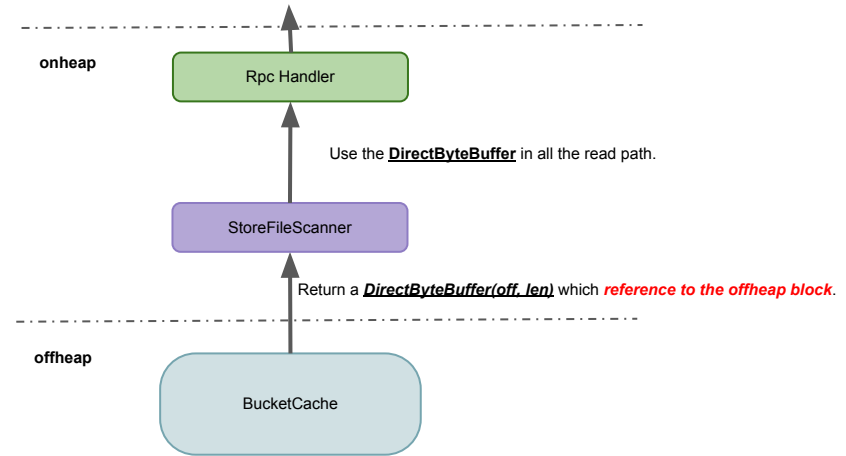
Memory allocated **on Java heap** and with no data.

- JVM will **NOT** sweep the cache and compacting them (old generation)
- **Less mixed GC(s) and shorter STW time.**

End-to-end offheap on the read-path (HBASE-11425)



branch-1.5



branch-2

- No need to copy block from offheap to onheap.
- Less onheap occupied.
- **Less mixed gc and shorter stw time.**

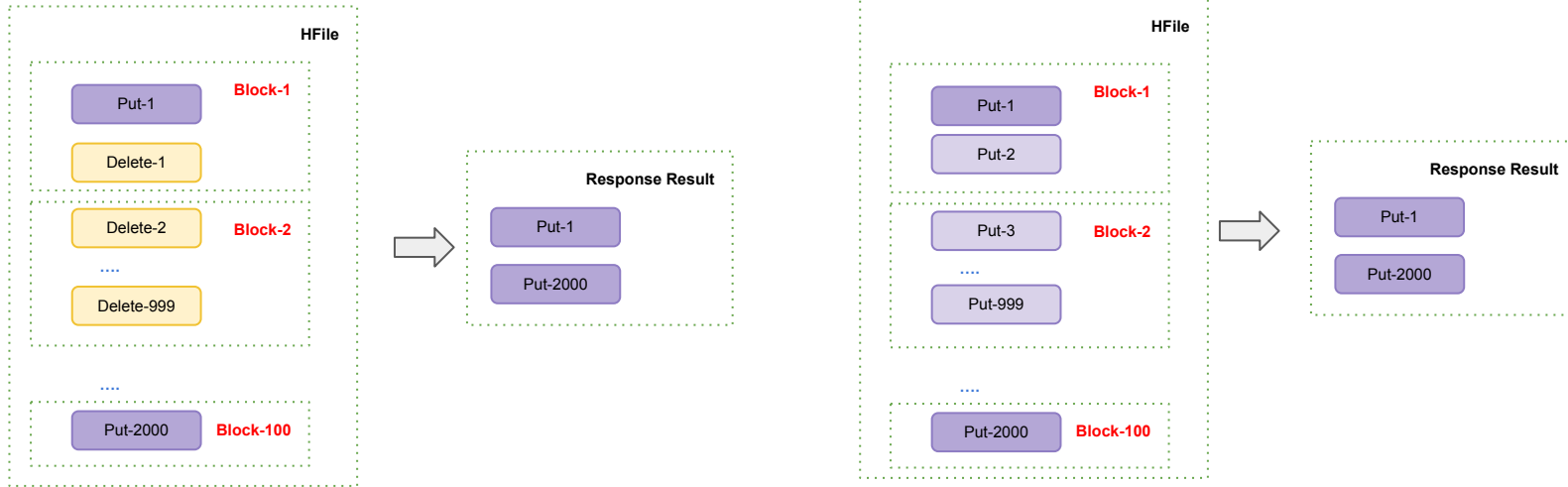
Abstract In Server Side

- ❑ High Efficiency
- ❑ **Query Diversity - TODO**
- ❑ Server Resource Protection

Abstract In Server Side

- ❑ High Efficiency
- ❑ Query Diversity
- ❑ **Server Resource Protection**

Two Cases



Scan the HFile with so many deletes

`scan.setFilter(new SingleColumnValueFilter(...))`

Read too many block data into memory, which may cause GC or OOM

Server Side Limit

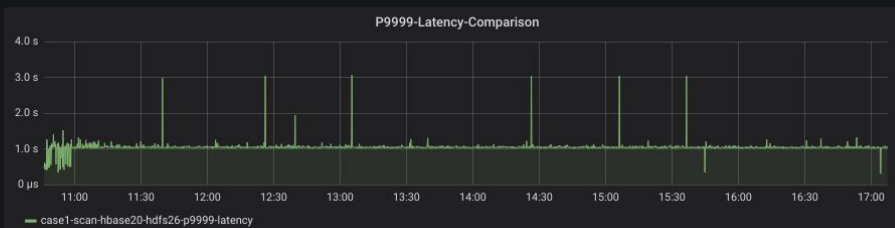
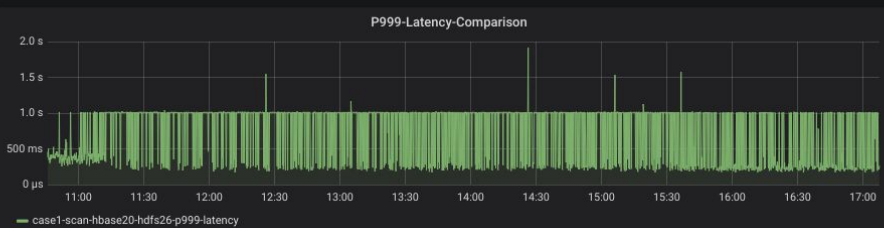
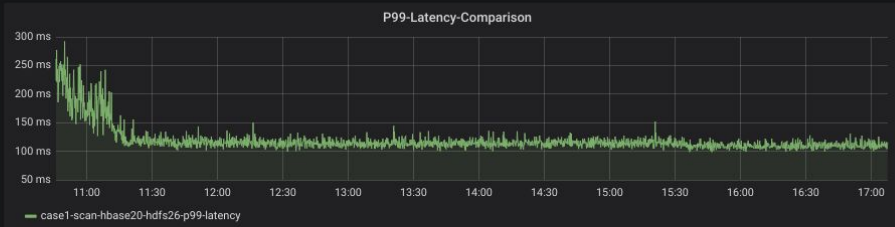
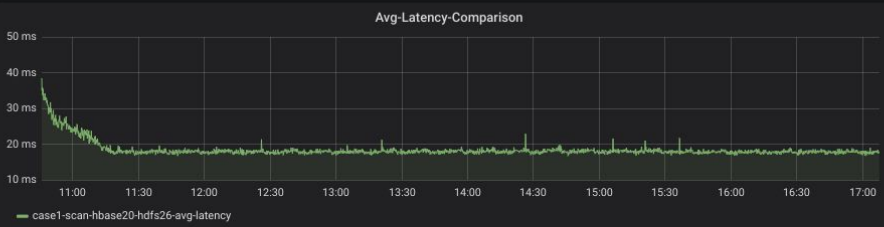
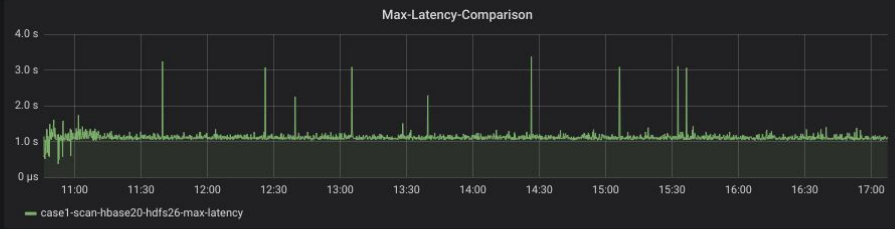
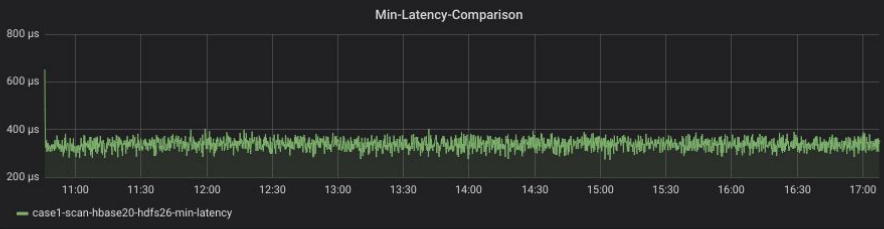
- Data Size / Heap Size
 - MaxResultSize
- Timeout
 - HeartBeat: abort this rpc once timeout and just return the current results to client.
 - Cursor: return a fake result with the current rowkey for next rpc once timeout.
- Batch
 - RS will still accumulate multiple results until reach max result size even if reach batch limit
 - Related issue: [HBASE-21206](#)
- *BlockSize ?*

Part-3 Tuning

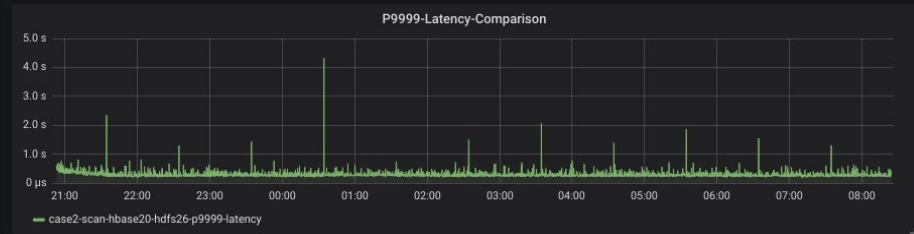
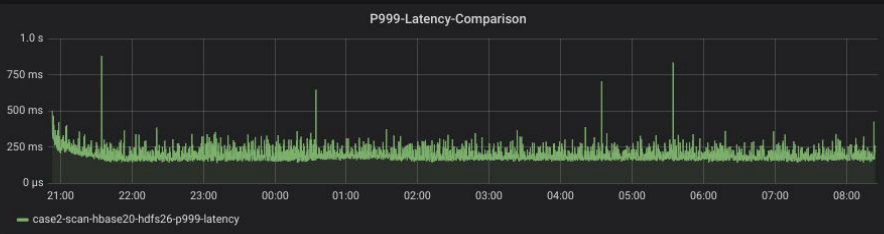
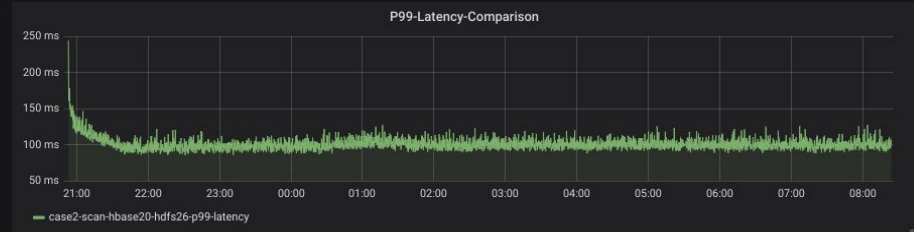
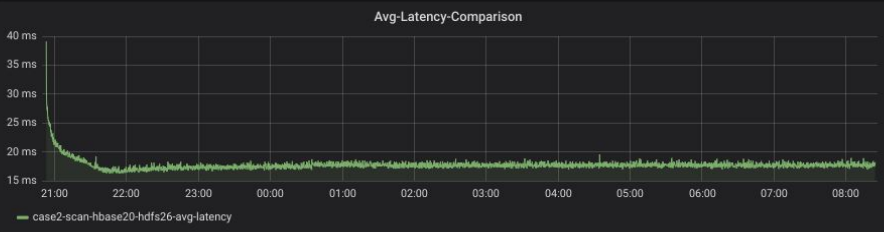
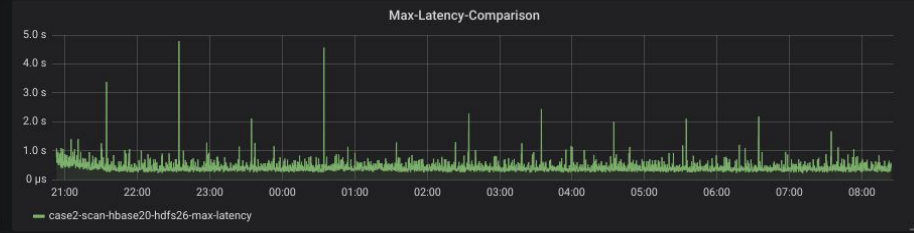
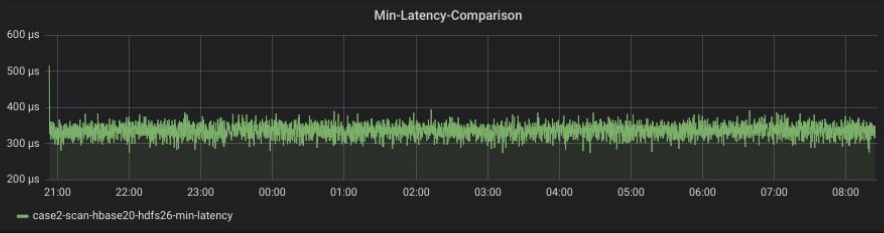
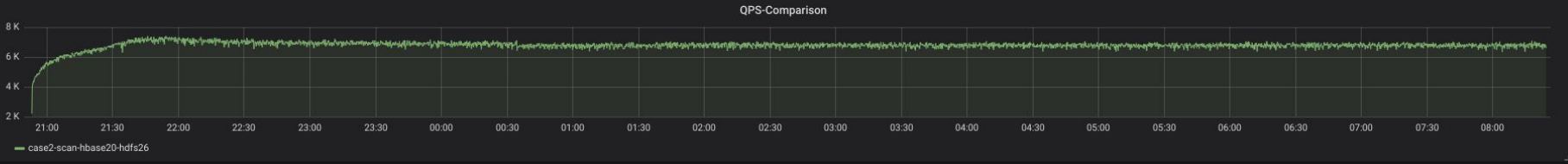
Tuning

- Read Distribution
- Locality
- *Short Circuit Read*
- CacheHitRatio
- StoreFileCount
- *ReadRawCells / ResponseCells*
- Java GC
- Scan Table OR Snapshot

case1-scan-hbase20-hdfs26 -



Rows(10^11) + onheap(12g)/offHeap(12g) + Balanced + Locality(1.0) + MajorCompaction + **enabledShortCircuitRead**





Short Circuit Read

- Disable Short Circuit Read
 - P99~120ms, but P999~1s.
 - Serious impact on P999.
 - One DN has 128 socket backlog, easy to happen “slow tcp connection”
- Enable Short Circuit Read
 - P99~100ms, P999~250ms.
 - Both QPS and latency are more stable.

ReadRawCells / ResponseCells

<input type="checkbox"/>	region server	memstore size (MB)	storefile size (MB)	read qps	get qps	write qps	read capacity (units/sec)	write capacity (units/sec)	response cells/sec	read raw cells/sec
<input checked="" type="checkbox"/>				3102	0	895	4651	1870	30077	398039155
<input checked="" type="checkbox"/>				2140	0	1083	4331	952	23563	294790225
<input checked="" type="checkbox"/>				4534	0	1769	8905	1935	42322	273242359
<input checked="" type="checkbox"/>				2335	0	571	4184	673	20559	257900269

398039155 / 30077 = 13234

ReadRawCells / ResponseCells

```
{
  "Scan": {
    "batch": -1,
    "cacheBlocks": true,
    "caching": 500,
    "families": {
      "C": [
        "ALL"
      ]
    },
    "filter": "FilterList AND (1/1): [SingleColumnValueFilter (C, s, EQUAL, \\x00\\x00\\x00\\x02)]",
    "loadColumnFamiliesOnDemand": null,
    "maxResultSize": -1,
    "maxVersions": 1,
    "startRow": "...",
    "stopRow": "37880",
    "timeRange": [
      0,
      9223372036854775807
    ],
    "totalColumns": 1
  },
  "class": "HRegionServer",
  "method": "Scan",
  "processingtimems": 1281,
  "queuetimems": 1444,
  "region": "...c3c62ba257913dc81d41745b5dd09f70.",
  "responsesize": 493,
  "starttimems": 1538399143265
}
```

ReadRawCells / ResponseCells

<input type="checkbox"/>	region server	memstore size (MB)	storefile size (MB)	read qps	get qps	write qps	read capacity (units/sec)	write capacity (units/sec)	response cells/sec	read raw cells/sec
<input checked="" type="checkbox"/>				3102	0	895	4651	1870	30077	398039155
<input checked="" type="checkbox"/>				2140	0	1083	4331	952	23563	294790225
<input checked="" type="checkbox"/>				4534	0	1769	8905	1935	42322	273242359
<input checked="" type="checkbox"/>				2335	0	571	4184	673	20559	257900269

$398039155 / 30077 = 13234$

<input type="checkbox"/>	region server	memstore size (MB)	storefile size (MB)	read qps	get qps	write qps	read capacity (units/sec)	write capacity (units/sec)	response cells/sec	read raw cells/sec
<input type="checkbox"/>				4210	0	1535	5036	1327	27692	4905492
<input type="checkbox"/>				4366	0	602	6134	1304	21177	3097303
<input type="checkbox"/>				3148	0	755	5912	767	16473	1911371
<input type="checkbox"/>				2765	0	509	7334	636	15366	1570779

$4905492 / 27692 = 177$



ReadRawCells / ResponseCells

- Solution
 - Scan the offline cluster's snapshot directly.
 - After the scan, we can get a rowkey subset.
 - For each rowkey in subset, checkAndPut the online cluster.

Thank You