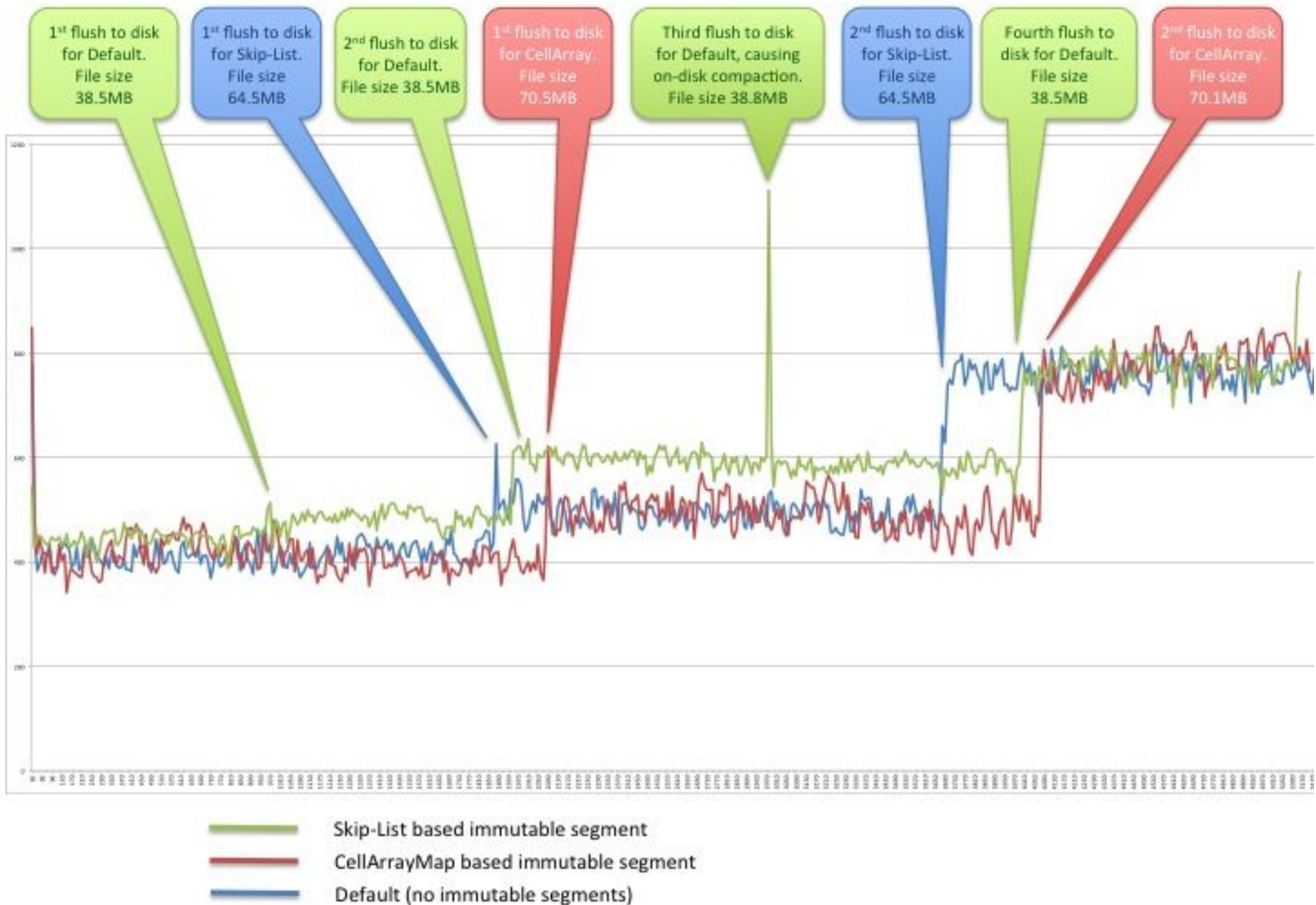


Initial CellArrayMap Evaluation

We set up a small cluster of 3 hdfs nodes, with a single region server (and a master). The heap at each node is allocated 1GB, while the global memstore size is limited to 300MB and the block cache size to 60MB (see the hbase-site.xml below). To avoid memory fragmentation we use mslab chunks of size 2MB. We use a data set of 1280KB records, record size is 100Bytes.

We do not load the table with any initial data, instead we use YCSB to run workloads of 5M operations, set target throughput to 1Kops, 50% updates, 50% reads, and to measure their performance over time. We first experimented with zipfian distribution.

The figure below depicts the average latencies of read and write operations over time. Latencies are accumulated over intervals of 10 seconds. We compare the performance of branch 2.0 in-memory column, HBASE-14920 in-memory compaction, and HBASE-14921 (with flattening disabled).



hbase-site.xml

<...cluster configuration settings...>

<property>

<name>hbase.regionserver.global.memstore.size</name>

<value>0.3f</value>

</property>

<property>

<name>hbase.regionserver.global.memstore.lowerLimit</name>

<value>1.0f</value>

</property>

<property>

<name>hfile.block.cache.size</name>

<value>0.01f</value>

</property>

<property>

<name>hbase.hregion.memstore.mslab.enabled</name>

<value>true</value>

</property>

<property>

<name>hbase.hregion.memstore.mslab.chunksize</name>

<value>2097152</value>

</property>

<property>

<name>hbase.hregion.compacting.memstore.type</name>

<value>2</value>

<description>The type of compaction, 1 - to be compacted to the skip-list, 2 - to be compacted to the CellArray</description>

</property>

<property>

<name>hbase.hstore.compaction.min</name>

<value>3</value>

<description>The minimum number of StoreFiles which must be eligible for compaction before compaction can run.</description>

</property>

</configuration>