

HBase-7312 Benchmark Report

This is an automated HBase scanner performance test report for **HBASE-7312** (Flatten the invoking to the compare/match methods of KeyValue.java). The workload uses **HBASE-6923** code, TestStorePerformance.java.

There are 2 parts of performance test comparison, PART 1 (test of 10,000,000 rows of data scanning) and PART 2 (long term scanning of 1,000,000 rows of data scanning). Results from both 2 PARTs of benchmark shows that there is stably about **2%** improvement brought by **HBASE-7312** code.

HBase versions compared:

- HBase-0.94.3-testscan-original (Original)
- HBase-0.94.3-testscan-7312 (New)

PART 1 -----

Workload Profile (every scan pass is under the following profile:):

- IO on local fs, not HDFS
- Read 10000000 rows
- Total bytes: 1280000000
- HFILE_BLOCK_SIZE: 64KB
- KEY_SIZE: 16 bytes
- VALUE_SIZE: 100 bytes
- Read through StoreScanner and heap
- Use raw (non-checksummed) fs when writing to local fs
- Disable use of HBase checksums

Original 0.94.3		HBASE-7312	
(ms)	Average	(ms)	Average
21224		23357	
20737		19719	
21848		20707	
27167		20211	
26607	23516.6	20143	20827.4
19948		22115	
19621		19680	
19586		20035	
19061		19558	
19331	19509.4	19235	20124.6
20474		21035	
19667		18484	

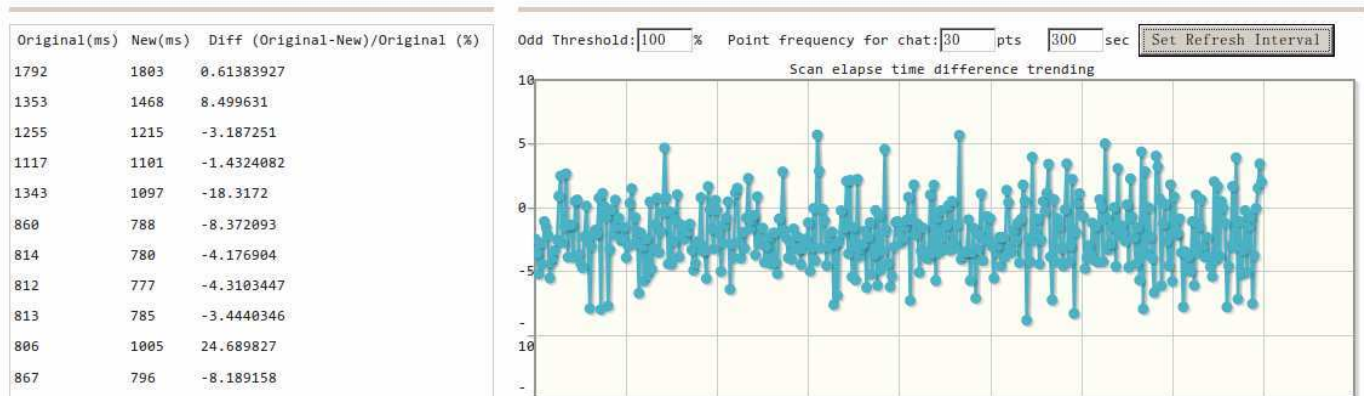
	19200		19197	
	19061		18294	
	19414	19563.2	18389	19079.8
	20080		20760	
	18890		19515	
	19852		19552	
	19102		18666	
	19036	19392	19721	19642.8
	19639		21152	
	18585		19934	
	19165		19443	
	18423		19901	
	18343	18831	19582	20002.4
	19789		19437	
	19545		19866	
	19476		19339	
	20740		18626	
	19875	19885	18722	19198
	23427		21363	
	18817		18224	
	18861		18861	
	18544		18356	
	18670	19663.8	18046	18970
	20592		19671	
	18059		18202	
	18572		18774	
	17725		18139	
	17671	18523.8	19266	18810.4
	22108		21931	
	20893		19969	
	18734		19915	
	18979		19045	
	27409	21624.6	19278	20027.6
Total/Avg	902547	20056.6	883415	19631.44444
Improvement			-2%	-2%

PART 2 -----

I wrote a benchmark scheduler on HBase server and a web console to monitoring the real-time comparison result as below.

```
HBase versions compared:
- HBase-0.94.3-testscan-original (Original)
- HBase-0.94.3-testscan-7312 (New)
-----
Workload Profile (every scan pass is under the following profile:):
- IO on local fs, not HDFS
- Read 1000000 rows
- Total bytes: 128000000
- HFILE_BLOCK_SIZE: 64KB
- KEY_SIZE: 16 bytes
- VALUE_SIZE: 100 bytes
- Read through StoreScanner and heap
- Use raw (non-checksummed) fs when writing to local fs
- Disable use of HBase checksums

=====
Total read pass for comparison: 12000
Counted read pass for comparison: 11996
Odd read pass due to noise: 4. Diff threshold as odd is +-100.0%.
Diff average: -2.2012250423431396%
```



Workload Profile (every scan pass is under the following profile:):

- IO on local fs, not HDFS
- Read 1000000 rows
- Total bytes: 128000000
- HFILE_BLOCK_SIZE: 64KB
- KEY_SIZE: 16 bytes
- VALUE_SIZE: 100 bytes
- Read through StoreScanner and heap
- Use raw (non-checksummed) fs when writing to local fs
- Disable use of HBase checksums

Test results:

Notes: for long term running, there are some passes that have huge differences, which could be the noise due to kinds of reasons like GC, etc. So I defined the “diff threshold” to exclude the odd results of those kinds of scan

passes. I got a online web console to monitoring the real-time comparison result. Odd diff threshold for counting and calculation can be set. Here just list 3 accumulated comparison results by the odd diff threshold (+-10%, +-100% and +- 1000%). Total count of scan pass is 12000 with 1000000 rows each. The HBASE-7312 improvement is stably fixed at about **2%**.

=====

Total read pass for comparison: 12000

Counted read pass for comparison: 10073

Odd read pass due to noise: 1927. Diff threshold as odd is +-10.0%.

Diff average: **-1.9083983898162842%**

=====

Total read pass for comparison: 12000

Counted read pass for comparison: 11996

Odd read pass due to noise: 4. Diff threshold as odd is +-100.0%.

Diff average: **-2.2012250423431396%**

=====

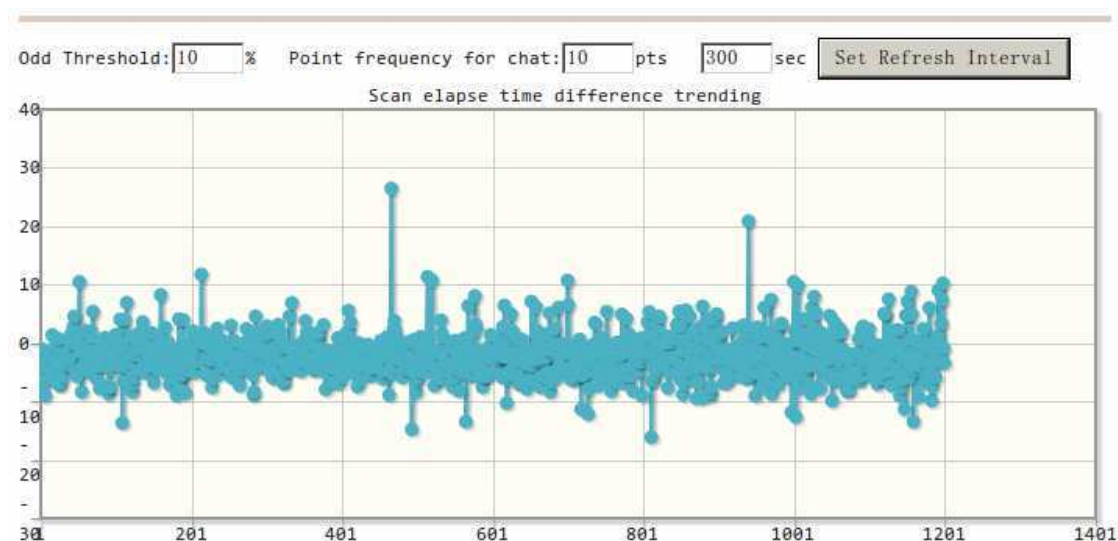
Total read pass for comparison: 12000

Counted read pass for comparison: 12000

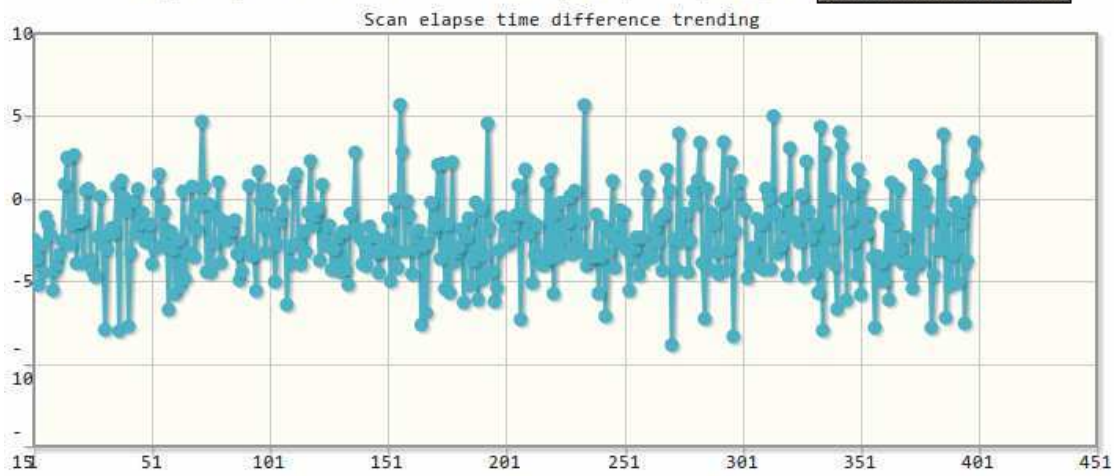
Odd read pass due to noise: 0. Diff threshold as odd is +-1000.0%.

Diff average: **-2.154961347579956%**

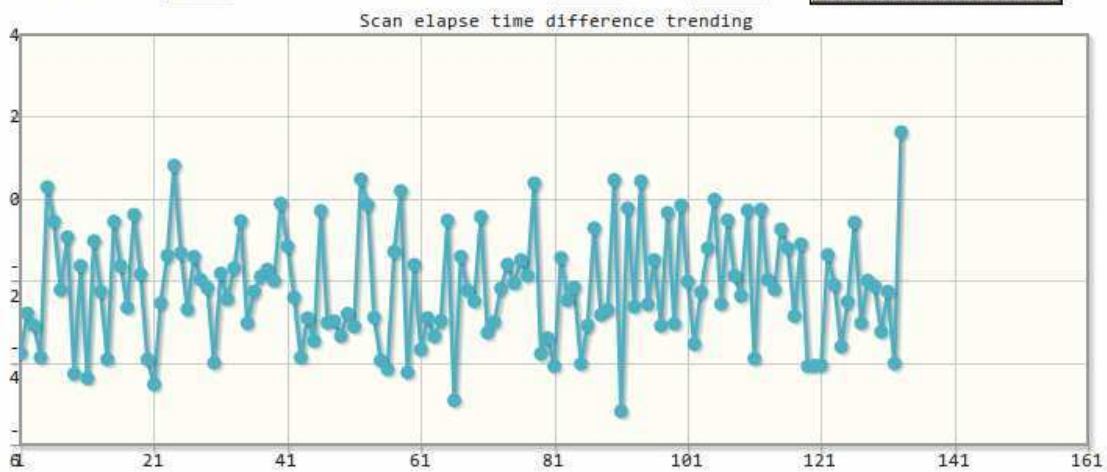
Following chats show the scan pass points difference trending. We could see that the improvements are averagely fixed at around **-2%** ((NEW-ORIGINAL)/ORIGINAL).



Odd Threshold: % Point frequency for chat: pts sec



Odd Threshold: % Point frequency for chat: pts sec



Odd Threshold: % Point frequency for chat: pts sec [Set Refresh Interval](#)

