

Database and File-system Access Speed Comparison

Author: Jan Horak, horak.honza@gmail.com

The reason of this test was to compare access speed of the reading data from pure filesystem and from MySQL database running on the localhost. It was supposed that reading data from database will be slower than reading from filesystem. The test was run twice with different record size.

In the filesystem as well as in the MySQL database were same random data in binary or text form. The data were accessed by their 128bit-length md5 hash. There were totally 100000 records of small files with average size of 3,7kB, which means totally 366MB of data and 11MB for index in database. In the test with large files there were 780MB of data, 28kB for index in 200 records.

In the filesystem the files had to be placed in separate directories, because more than 1000 files in one directory slows down the directory searching. An example of the stored files can be as follows:

- files/df/dfdbf90903852bd6bbd66c532fceb2b5
- files/c9/c9d7b037dc8dcd52fb7a972e33a0a1cf
- files/73/73f65d20dca6fc6063c6974757cab145

In MySQL database the content of the files were stored in one InnoDB table with integer primary column ID, column HASH of the type VARCHAR(32) for MD5 hash of the content and column CONTENT of the type LONGBLOB for the content itself. Searching was performed using one UNIQUE INDEX on the HASH column.

Each test composed from 10 partial tests. 1000 random records were loaded in every partial test in the case of small files and 300 records in the case of large files. Pseudo-random generator had been initialized by the same seed, so the results from MySQL and from filesystem were the same. Each test was run twice for small files and twice for large files, because repeated reading could be influenced by using a cache.

Results of the test with small files are in the Table 1 and Figure 1, results of the test with large files are in the Table 2 and Figure 2.

	Filesystem #1	MySQL #1	Comparison #1	Filesystem #2	MySQL #2	Comparison #1
Test #1	11,29	4,81	43%	0,013	0,814	6091%
Test #2	12,84	4,74	37%	0,017	0,688	4138%
Test #3	10,69	4,56	43%	0,023	0,638	2766%
Test #4	10,67	4,32	40%	0,017	0,707	4047%
Test #5	9,92	3,98	40%	0,023	0,616	2677%
Test #6	9,86	4,07	41%	0,023	0,620	2745%
Test #7	12,37	3,78	31%	0,016	0,548	3326%
Test #8	10,19	3,25	32%	0,014	0,377	2791%
Test #9	9,43	3,50	37%	0,013	0,383	2854%
Test #10	9,34	3,00	32%	0,013	0,299	2250%

Table 1: Results of the test with small files

Small Files Data Access Speed Comparison (s)

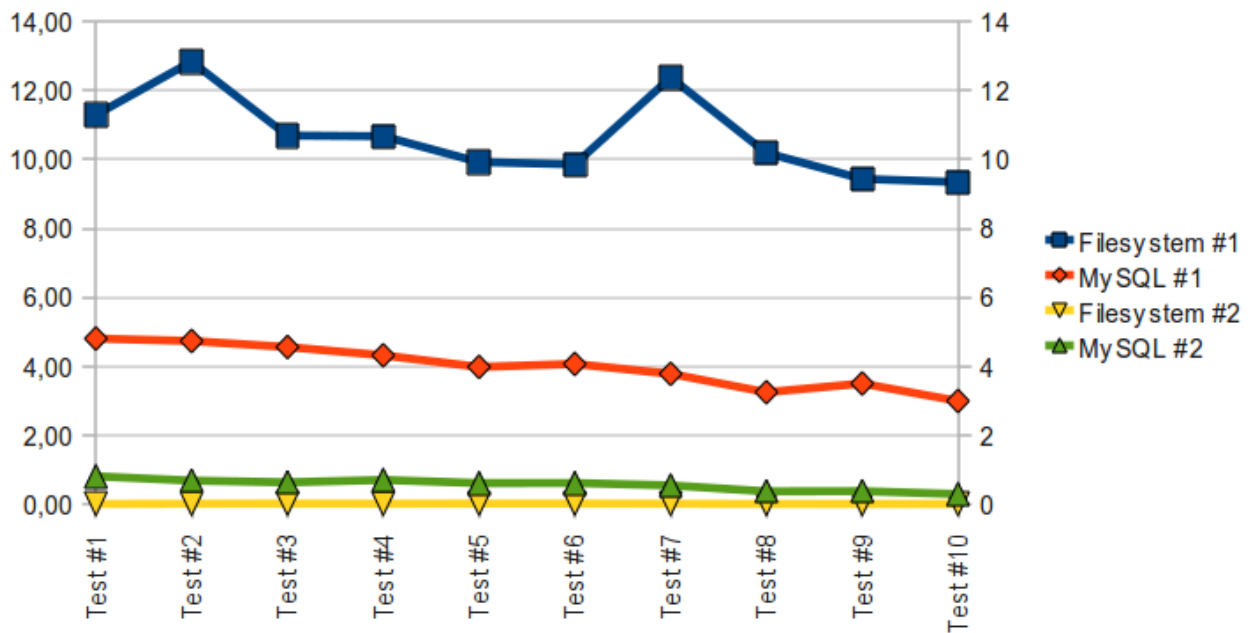


Figure 1: Results of the test with small files

	Filesystem #1	MySQL #1	Comparison #1	Filesystem #2	MySQL #2	Comparison #1
Test #1	1,81	11,52	637%	1,07	7,29	682%
Test #2	2,45	9,71	397%	0,98	6,78	689%
Test #3	2,15	10,27	478%	1,12	7,7	687%
Test #4	1,65	7,92	480%	1,04	7,17	690%
Test #5	2,19	7,63	348%	1,1	7,63	690%
Test #6	1,55	8,38	542%	1,21	8,32	689%
Test #7	1,14	7,31	642%	1,07	7,37	691%
Test #8	1,15	8,33	723%	1,23	8,42	686%
Test #9	1,08	7,52	699%	1,07	7,39	694%
Test #10	1,04	7,39	712%	1,02	7,24	707%

Table 2: Results of the test with large files

Large Files Data Access Speed Comparison (s)

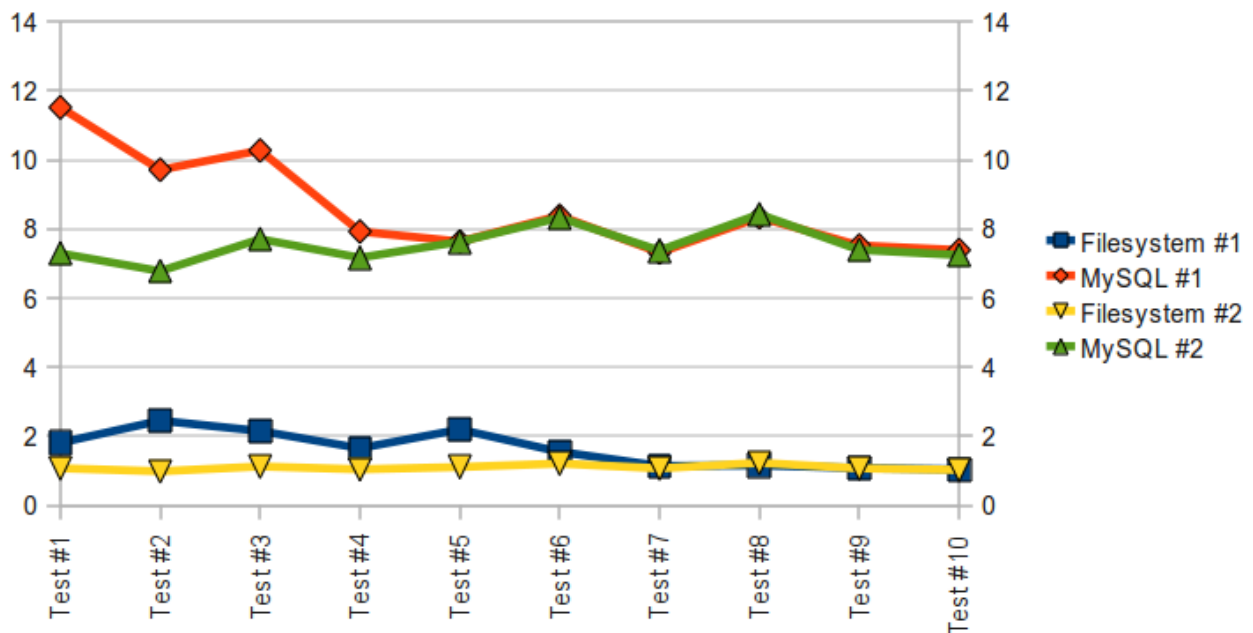


Figure 2: Results of the test with large files

Note: Tested on a computer with Intel Core2Duo T5550, 1.83GHz, 2GB RAM, HDD Hitachi HTS542525K9SA00, 5400rpm, Ubuntu 9.10, Linux Kernel 2.6.31-20.

We can see that reading data from MySQL is not always slower than reading data from pure filesystem. Especially in the case when small files are read first time, MySQL database is much faster than a pure filesystem. This is probably caused by better caching in MySQL database.

If we want to improve the reading from pure filesystem, we could store small records in larger files so the filesystem cache could be used better. The real speed also depends on the character of the work with data, it depends on repetition of reading of the same or close data and it also depends on the size of the data.