**Description:**

1. Switch the data source from file system to DB.
2. Generate 1000 text posts.
3. Perform capacity testing.
4. Perform regular load testing with the parameters based on the results of capacity testing.
5. Perform regular load testing with the parameters based on the results from Task 8.
6. Perform stress testing.
7. Repeat volume testing with the parameters of the Task 9.
8. Prepare complex report including the comparison between the results with files system data source and DB data source if appropriate results are available.

**User role:**

                admin  
                editor  
                anonymous

**Goals:**

* Get an experience of configuration testing
* Learn complex testing during migration of test data to new platform.
* Learn complex analysis.

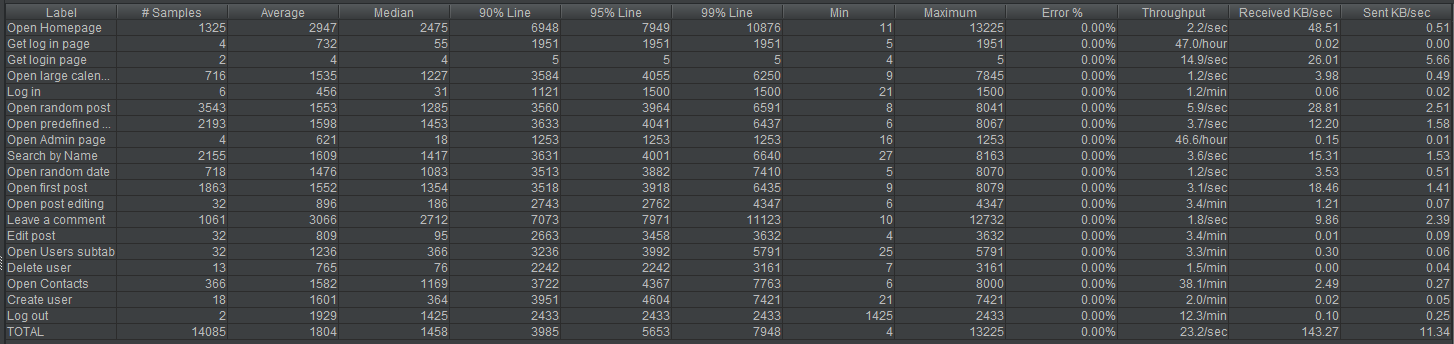
**Results:**

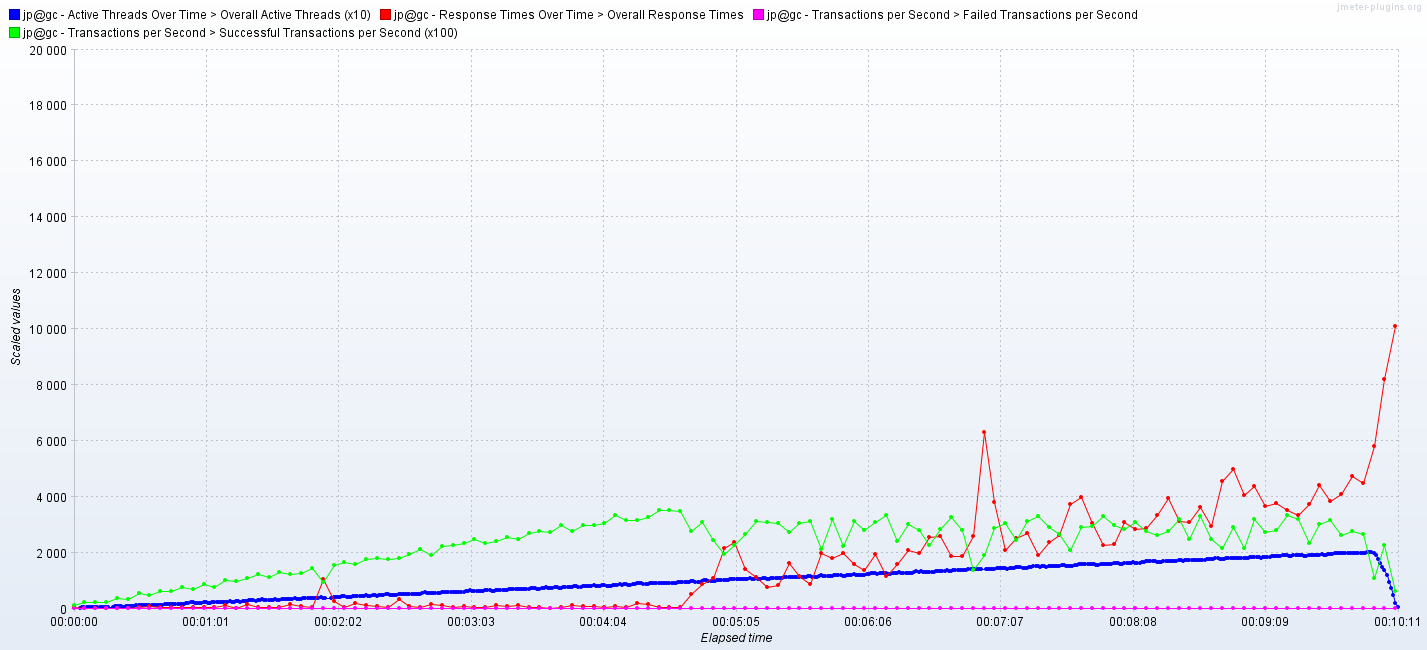
1. **Capacity testing results**

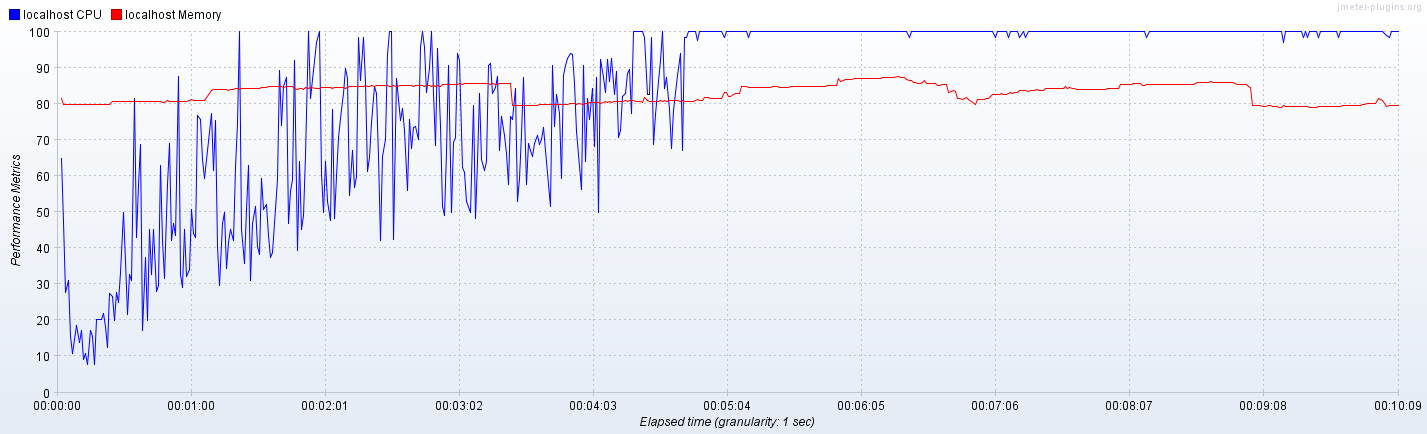
In this script 3 thread groups with different scenarios were used. Workload of this script is 200 users for anonymous scenario, and 2 users for editor and admin scenarios separately. Rampup and duration is 600 ms for all script. Workload profiles are displayed at the end of document in form of algorithms.

From graphics below we can see, that till approximately 04:40 system is in comfort zone. In this period of time system reaches its capacity point, CPU reached 100% and don`t get down till the end of test. Amount of transaction falling, and response time starts growing. At this time approximately 96 users are using the site: 2 editors, 2 admins and 92 anonymous, and approximately 35 transactions at the same time are performed.

Comparing with run of the same script using file system we can see, that in this run errors didn`t occur at all, but system reached its capacity earlier. Amount of maximum transactions is approximately the same, but we can say, that capacity of application, that using DB is lower, that capacity of the system, that is using file system as data source.







1. **Regular load testing results**

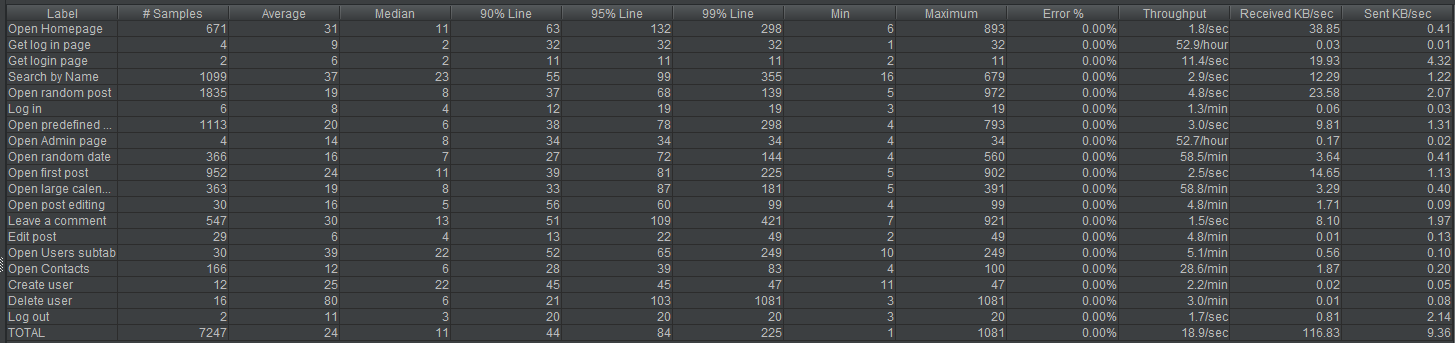
Before running regular load test such load should be defined. As already mentioned, capacity of the system, that is using database as data source, is 96 users. So, for performing load test was decided to use such workload model:

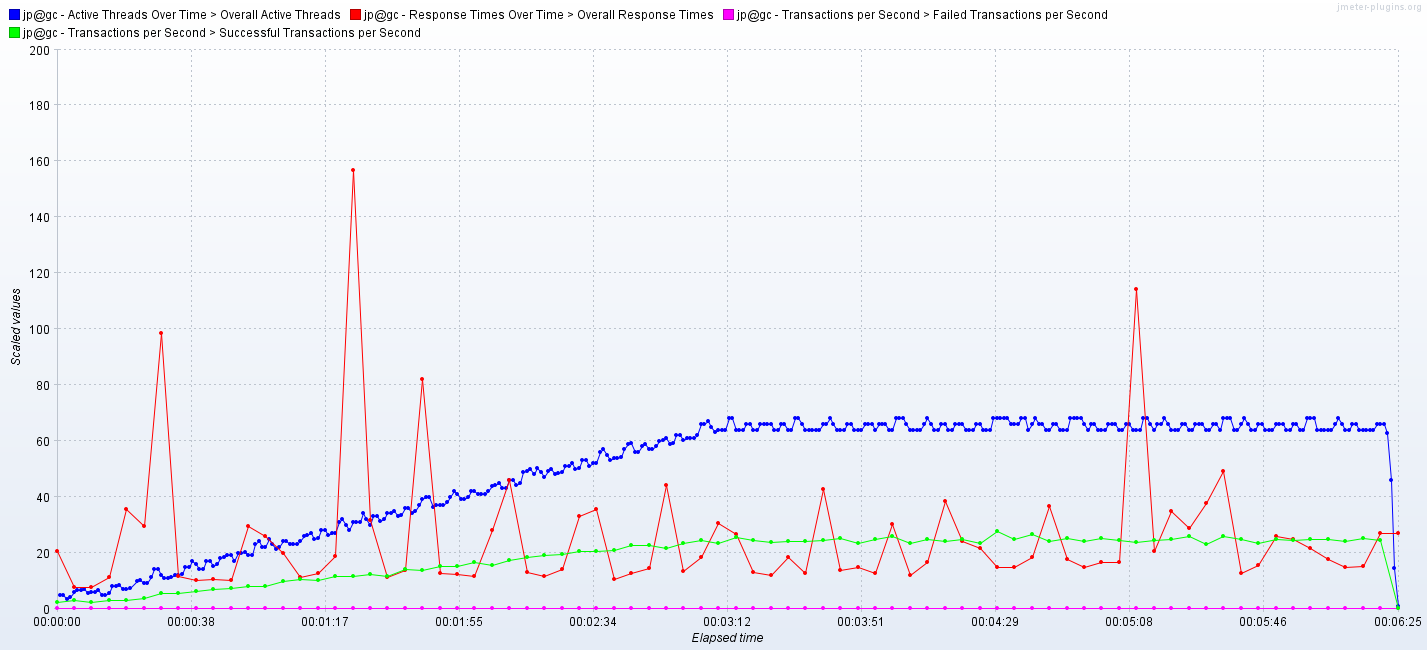
* number of threads: 64 anonymous + 2 admins + 2 editors;
* ramp up: 192 s;
* duration: 384 s.

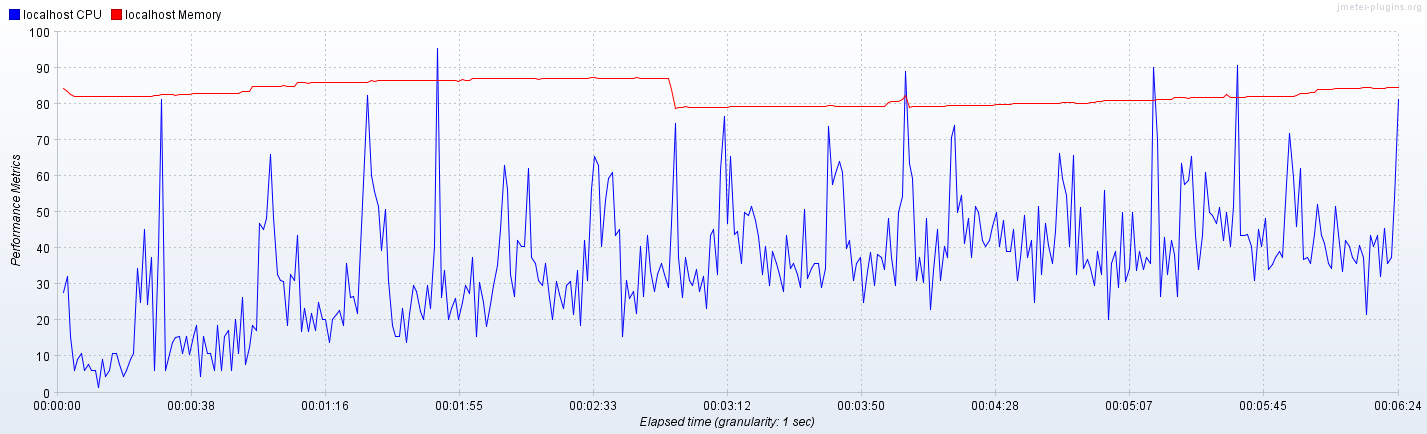
After performing regular load test and gathering KPI`s we are able to compare with results of the same test for file system data source. From the table below, we see, that response time for the system with database data source is lower, as well as CPU load and memory load. Max throughput is the same and it is good result, because load on system with DB is less, than on system, that using files as data source. Also we see, that test issues are not occurring at all, so looks like system is able to operate with defined amount of data at one time without problems.

Due to all this factors we may say, that system, that is using database as data source works more stable and quickly that system that is using file system.

|  |  |  |
| --- | --- | --- |
|  | File system | Database |
| Maxthroughput | 28 tps | 28 tps |
| Response time | 30-40 ms, up to 132 ms | 12-30 ms, up to 115 ms |
| CPU | 50-80% load, up to 100% | 28-52% load, up to 91% |
| Memory | 85-90% | 79-84% |
| Test issues | 0.02% | 0% |







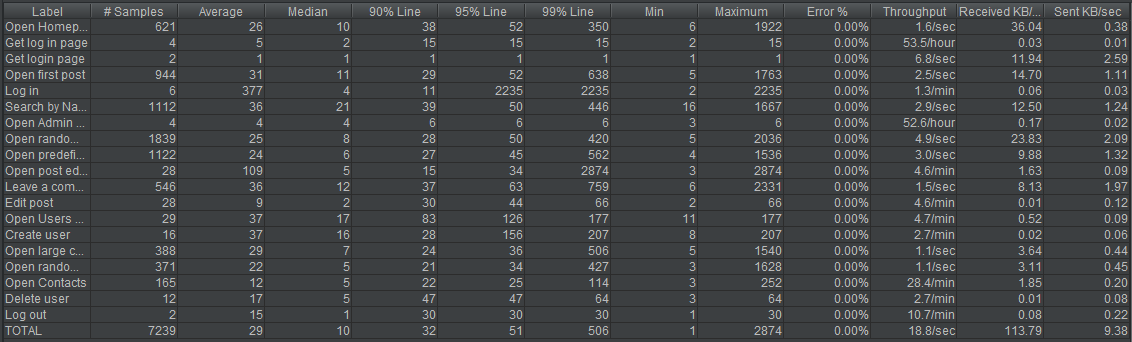
1. **Regular load testing with maximum scalable configuration results**

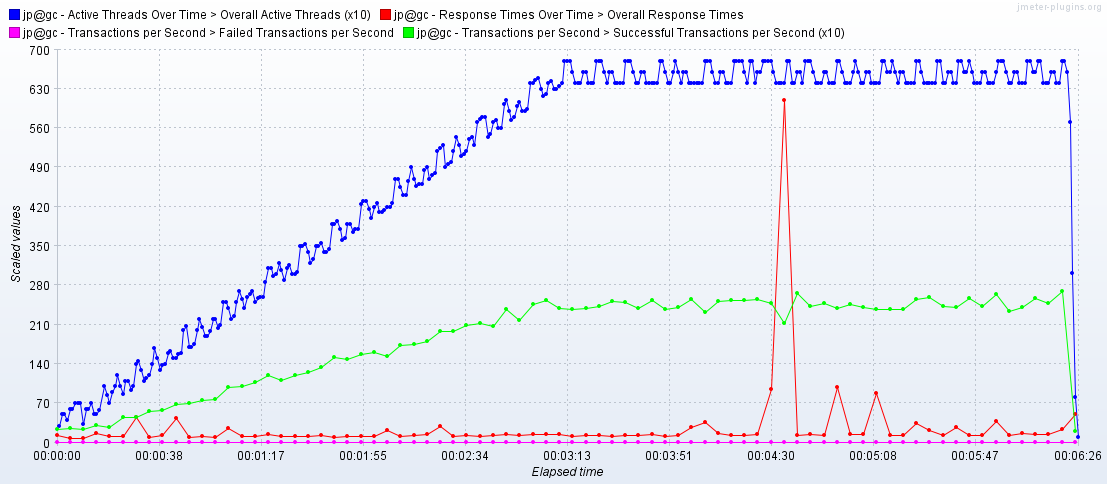
Due to results of Task 8 was decided to use such configuration: 1 CPU and 6144 MB RAM. Results of run are displayed below in table.

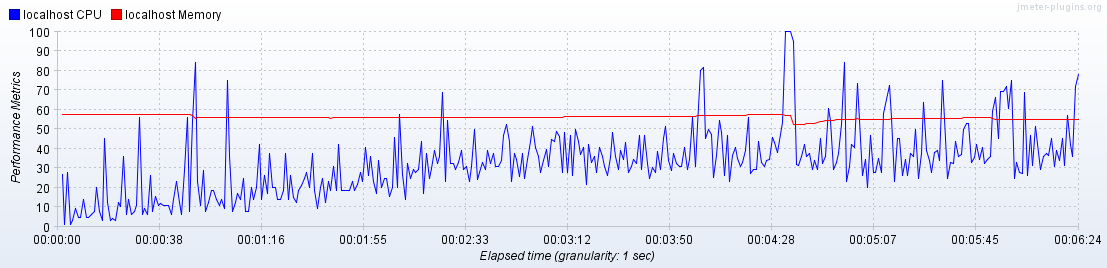
From them we see, that response time is approximately the same as well as CPU load. Maximum amount of transactions is little bit less, but as already mentioned its only due to that fact, that load on system with DB is less, than on system, that using files as data source and sometimes system performed 28-29 tps as well. Memory load is bigger, but its only due to other running processes on the system. After killing them to make state of the system similar to state, when file system was checked, it got similar percentage. So, it varies from run to run. Test issues were not occurring in both cases.

So, due to all mentioned above we may say, that system that is using DB works little bit worse but, in my opinion, it is not critical a lot. Maybe some increase in memory configuration will bring better results.

|  |  |  |
| --- | --- | --- |
|  | File system | Database |
| Maxthroughput | 29 tps | 27 tps |
| Response time | 8-20 ms, up to 270 ms | 8-24 ms, up to 607 ms |
| CPU | 30-50% load, up to 100% | 27-50% load, up to 100% |
| Memory | 42-44% | 52-56% |
| Test issues | 0% | 0% |







1. **Stress testing results**

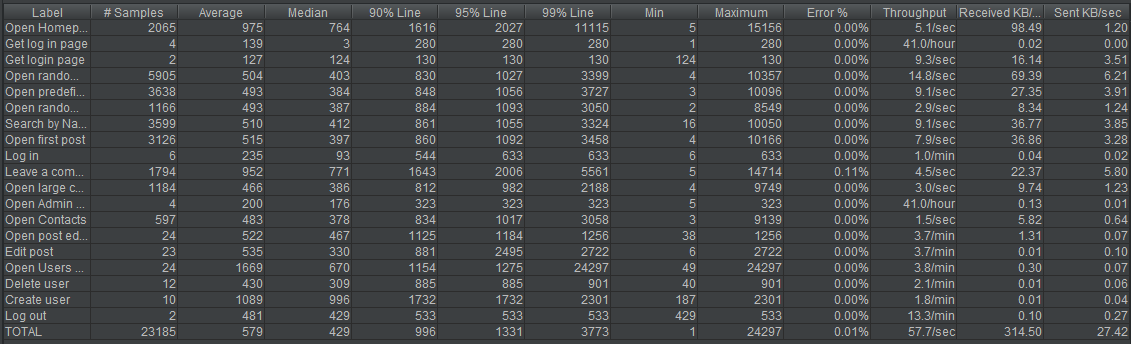
Before running stress test such load should be defined. As already mentioned, capacity of the system, that is using database as data source, is 96 users. So, for performing stress test was decided to use such workload model:

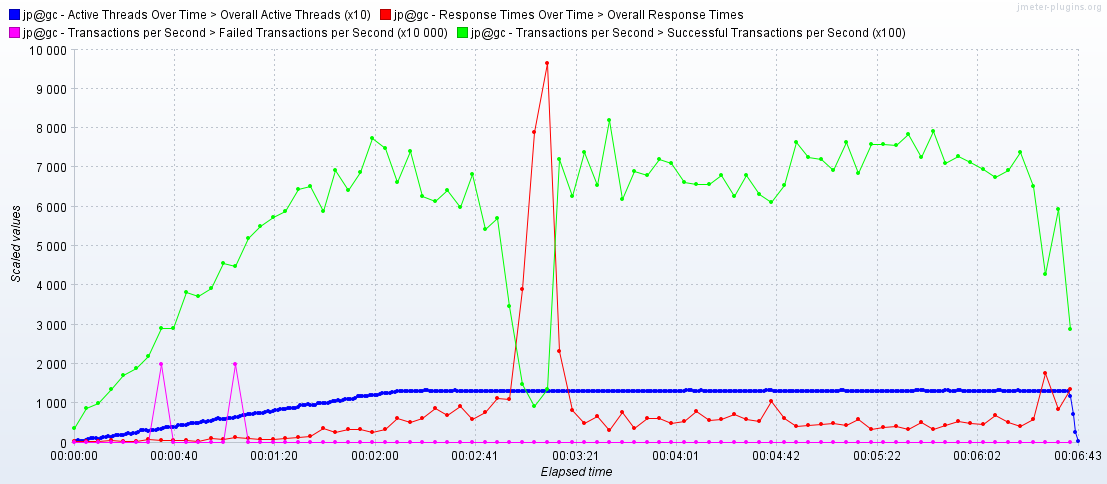
* number of threads: 130 anonymous + 2 admins + 2 editors;
* ramp up: 130 s;
* duration: 400 s.

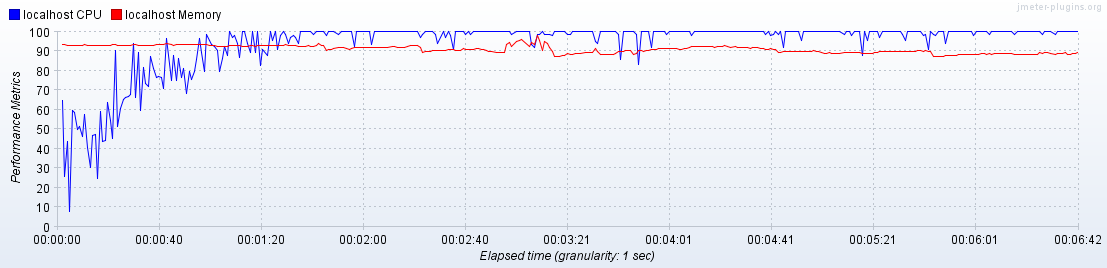
So, for performing stress load on the system was decided to increase number of users and to decrease ramp up, so to assign 1 second for connecting one user instead of 3. Duration also was decreased accordingly.

Here we have the results. Already on 01:35 system behavior of the system started degradation. CPU reached 100 % and didn`t fall a lot since this moment till the end of test. Response time started growing, amount of transactions continued growing but with jumps. On 03:04 we see the most critical moment for the system. Throughput fell in several times as well as response time increased in several times. After this this metrics came to the previous values and didn`t change till the end of test. Also we see that at the start of test some issues occurred, but according to the all test it is only 0.01% of them.

So, looks like system is able to handle with such load on some way, because system didn`t stop working and continue to perform transactions. It can normalize work after critical load moments.







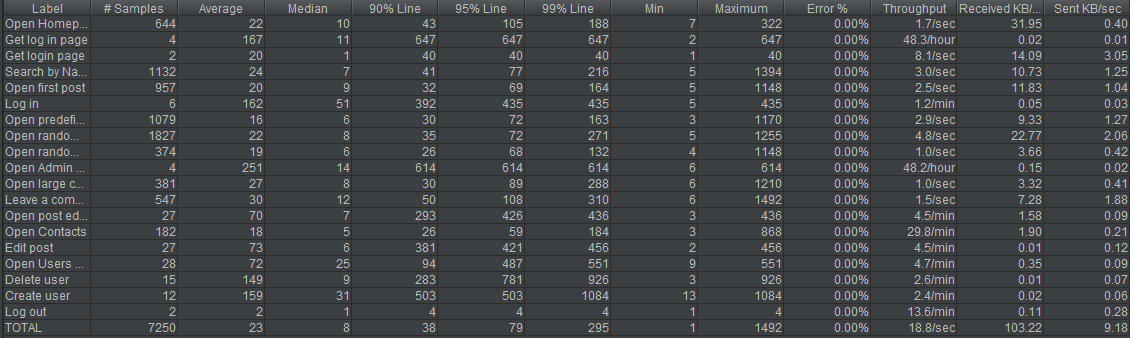
1. **Volume testing results**

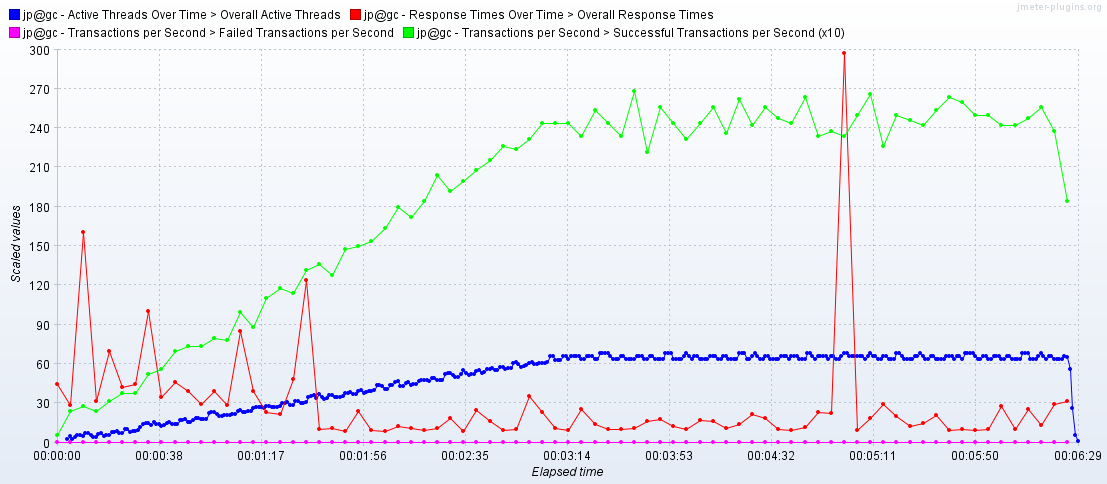
Due to run in Task 9 was found, that optimal amount of posts, that system is able to handle is 1000. As it basic value for all tests in this task and results of two data source were already compared, was decided to compare KPI`s of 2000 pure posts runs and 2000 mixed posts runs.

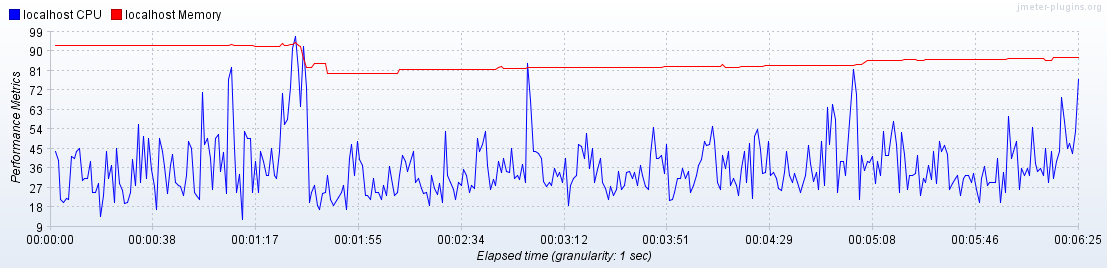
**2000 pure posts:**

From results below we see, that response time is less in two times than for system using DB, as well as CPU load and Memory. In both cases no issues occurred, but performance got higher. Looks like system using file system data source can not operate such amount of data, while system using DB is able.

|  |  |  |
| --- | --- | --- |
|  | File system | Database |
| Maxthroughput | 29 tps | 27 tps |
| Response time | 13-28 ms, up to 270 ms | 10-24 ms, up to 300 ms |
| CPU | 39-59% load, up to 100% | 22-48% load, up to 81% |
| Memory | 81-87% | 82-86% |
| Test issues | 0% | 0% |





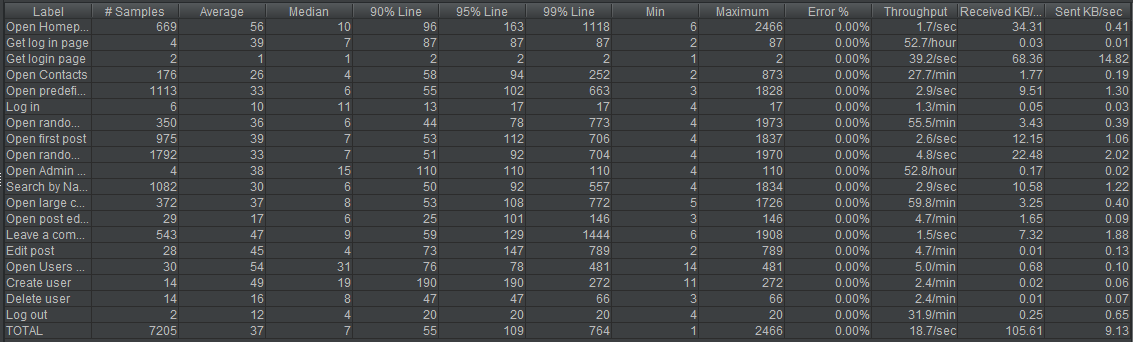


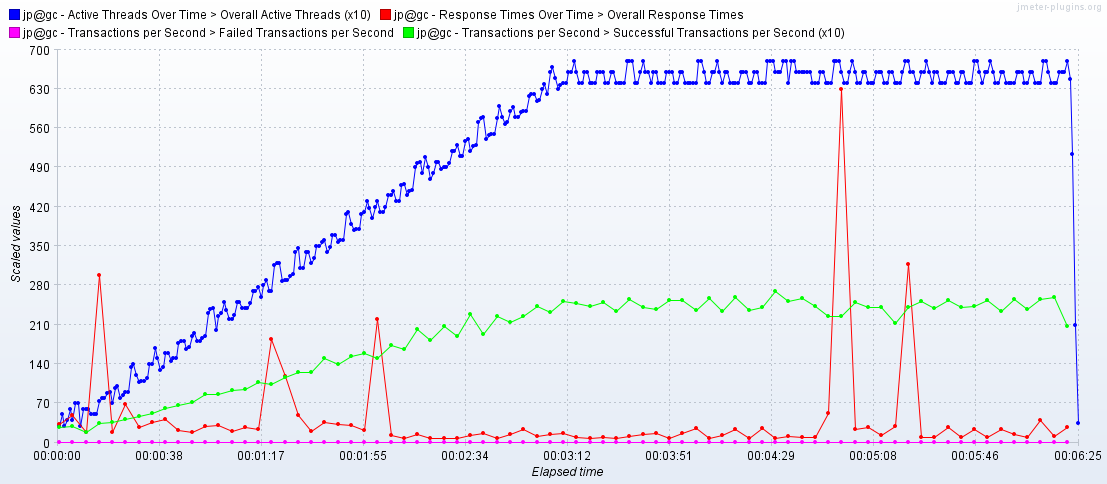
**2000 mixed posts:**

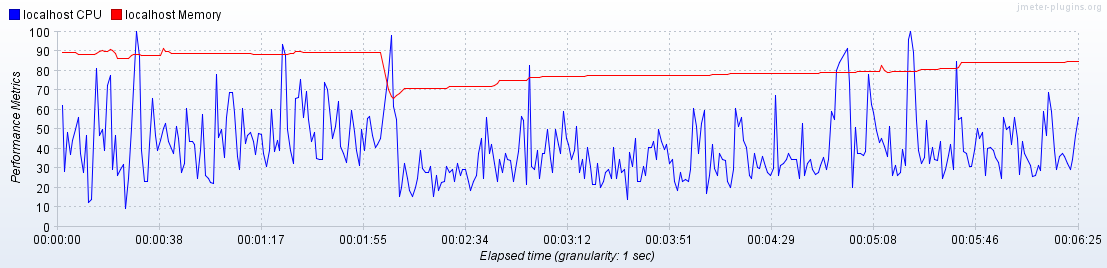
From results below we see some differences again. Response time is less in several times for file system data source, CPU load is also higher as well as memory, what was predictable. Throughput is lower for database, but it is acceptable, because as already said it vary from run to run and doesn`t relate to higher load.

So, due to all mentioned above we may say, that database can operate with heavier file more quickly and effectively.

|  |  |  |
| --- | --- | --- |
|  | File system | Database |
| Maxthroughput | 29 tps | 27 tps |
| Response time | 29-75 ms, up to 186 ms | 8-28 ms, up to 629 ms |
| CPU | 48-77% load, up to 100% | 21-50% load, up to 96% |
| Memory | 87-91% | 77-84% |
| Test issues | 0.05% | 0% |

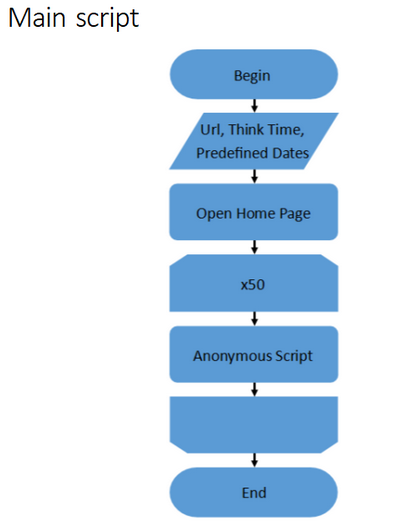


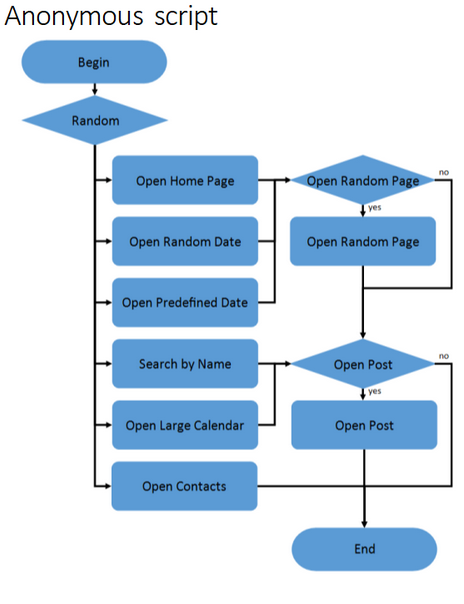


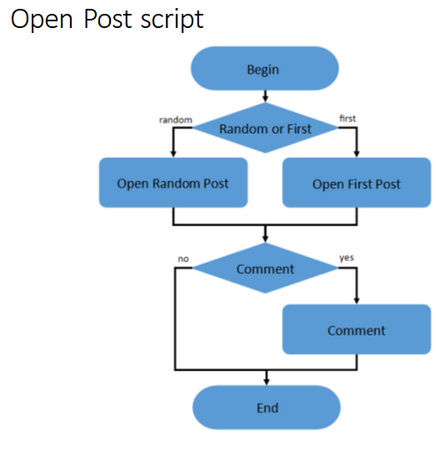


**Scenarios:**

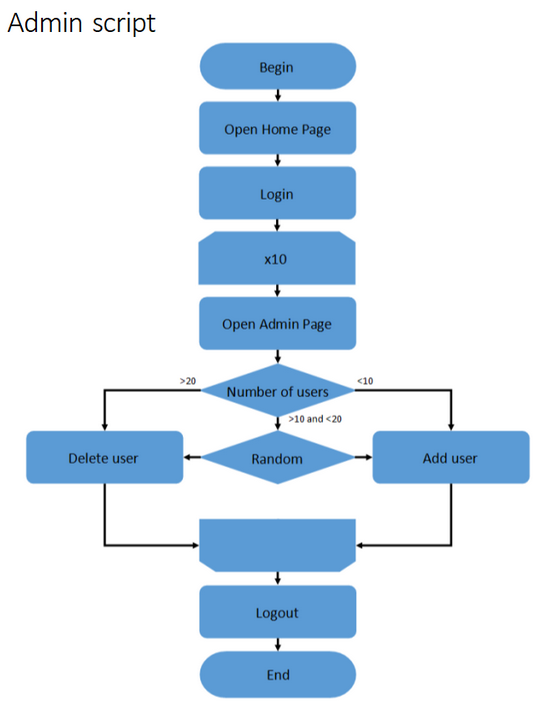
**Anonymous scenario:**







**Admin scenario:**



**Editor scenario:**

