Performance Report Blueriq 12.2 (Redis)

Version: 12.2
Date: 09.07.2019

DISCLAIMER

The results in the following report have been achieved on a specific hardware, software and application combination. The results in this report may differ from results on other combinations of hardware, software, custom code and application settings. No rights can be derived from this document.

It is strongly recommended to perform appropriate performance tests on the application that is modelled with Blueriq before taking it into production.

1 INTRODUCTION

Performance testing is a type of testing intended to determine the responsiveness, throughput, reliability, and/or scalability of a system under a given workload.

This report provides an overview of the performance tests for **Blueriq 12.2 when using a forms application with a Redis session store**. The purpose of this report is to give insight into the performance of such a Blueriq application on a typical hardware configuration and to validate the use of the new session store component. Based on this report, Blueriq customers should be able to review the performance characteristics of their own Blueriq application in production and to assess the infrastructure adequacy.

How to use this report
Testing and reporting on performance is complex since it depends on a large number of variables. On the [documentation page on the Blueriq Community](#)[1] the test approach and the assumptions of the tests can be found. It contains the reference application, user scenarios, the test environment, the test methodology, key performance indicators and the acceptance criteria.

Chapter 3 of this document will summarize the test results .

The results from performance testing and analysis can help Blueriq customers to estimate the hardware configuration required to support the application(s) when "going live" to production operation.

1.1 TEST ENVIRONMENT

Three machines were involved in the performance test: the Runtime, the Redis server and the JMeter application which ran the script and simulated the load. Each of those elements were deployed on their own separate machine so that the computational power could stay unaltered and unaffected for every environment.
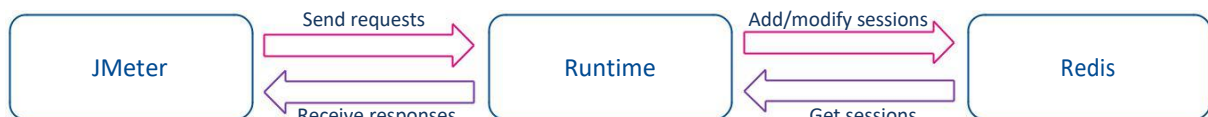
The following are the specifications for each server:

JMeter machine (bq-jmeter):

- OS Windows Server 2016 Standard
- Intel® Xeon® CPU E5-2680 v2 @ 2.80GHz 2.79GHz (2 processors)
- 2,00 GB memory (RAM)
- 39,10 GB HDD

---

1   https://my.blueriq.com/display/DOC/Performance+Reports

Runtime server (bq-perf-test):

- OS Windows Server 2016 Standard
- Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz (4 processors)
- 8,00 GB memory (RAM)
- 40 GB Hard Disk (Raid 5)
- JBoss EAP 7.2 application server
- 50 Redis connections
- 1 Runtime instance

Redis server (bq10-redis-perf):

- OS CentOS Linux 7 (Core)
- Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz
- 2,00 GB memory (RAM)
- 16,00 GB HDD

2   TEST RESULTS

The table below shows the results of all individual steps of the performance script. Please note that this test simulated a constant user load of 800 users during a one hour run. The scenario is also explained on the community page.

For each step (Key Performance Indicator) the following values are given:
- The T-value used for calculating the Apdex value
- The Apdex value with the colour indicating the rating

Step name	Actions	T-Value (ms)	Apdex value R11.6	Apdex value R11.7	Apdex value R11.8	Apdex value R11.9	Apdex value R11.10	Apdex value R11.11	Apdex value R12.2
1. Login	The user logs in with a username and password;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
2. Start Project	Start the application and display the Welcome screen;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
3. Open ATMs section	Switch to the Find Local ATMs page	500	1,00	1,00	1,00	1,00	1,00	0,99	0,99
4. Search ATMs	Creates 500 instances using the AQ_CsvConnectivityService and an external CSV file, then switches to the Confirm ATMs Found page;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00

5. Display search results	Opens the Display ATms page where the previously created instances are displayed in an AQ_InstanceList with page size 20;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
6. Go to loan offers page	Navigates to the Loan Offers page. The page contains dozens of containers and text items, each of them having presentation or content styles assigned to them;	500	1,00	1,00	1,00	0,99	0,99	0,99	0,99
7. Apply for prime offer	Updates an instance through the AQ_InstanceUpdate service and navigates to the Loan Form page;	500	1,00	1,00	1,00	1,00	1,00	1,00	0,99
8. Open calculator	Opens the Borrowing Calculator page;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
9. Fill in fields (1)	Fills in the available page fields. Other conditional fields are presented to the user because of this action;	500	1,00	1,00	1,00	0,99	1,00	1,00	1,00
10. Fill in fields (2)	Fills in the newly available page fields. New fields are displayed because of this action;	500	1,00	1,00	1,00	1,00	1,00	0,98	0,99
11. Fill in fields (3)	Fills in the newly available fields on the page;	500	1,00	1,00	1,00	1,00	1,00	1,00	0,99
12. Calculate	Updates 3 instances belonging to separate entities using the AQ_InstanceUpdate service and displays the calculation results to the user;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
13. Go to the first page	Returns to the welcome page;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
14. Open the loan offers page	Navigates to the Loan Offers page;	500	1,00	1,00	1,00	1,00	1,00	0,98	1,00

15. Apply for a fixed rate	Updates an instance through the AQ_InstanceUpdate service and navigates to the Loan Form page;	500	1,00	1,00	1,00	0,99	1,00	1,00	0,99
16. Fill in fields (4)	Fill in the available fields on the page. Some fields are refresh fields and have validations;	500	1,00	1,00	1,00	1,00	0,99	1,00	1,00
17. Fill in fields (5)	Fills in the available fields on the page. Some fields are refresh fields and have validations;	500	1,00	1,00	1,00	0,99	1,00	1,00	1,00
18. Fill in fields (6)	Fills in the available fields on the page. Some fields are refresh fields and have validations;	500	1,00	1,00	1,00	0,99	1,00	0,99	1,00
19. Continue	Navigate to the Personal Information page;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
20. Fill in personal form	Fill in all fields and navigate to the Document Upload page;	500	1,00	1,00	1,00	0,99	0,99	0,98	0,99
21. Upload file	Upload a file to certify the user's identity. As the upload is successful an AQ_FileDownload container is generated on the page;	500	1,00	1,00	1,00	0,99	1,00	1,00	1,00
22. Download file	Download the previously uploaded file;	500	1,00	1,00	1,00	1,00	1,00	0,99	0,99
23. Submit request	Navigate to the Confirmation page;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
24. Go to the loan offers page	Navigate to the Loan Offers page;	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00
25. Log out	The user logs out.	500	1,00	1,00	1,00	1,00	1,00	1,00	1,00

(*) For a description of the scenario, please refer to:
https://my.blueriq.com/display/DOC/Performance+Reports

3 CONCLUSION

All the requests have response times under 100 milliseconds and an Apdex value located at 0,98 - 1,00.

As new 12.x releases will appear this document will be updated with new performance figures and possible explanation and analyses on the performance variations.