

Theme: DEVELOPMENT OF A METHODOLOGY FOR CLOUD SOFTWARE TESTING IN ORDER TO REDUCE THE COST OF ITS DESIGN

Annotation: The application of the cloud testing methodology developed in the framework of the study will ensure a reduction in software design costs and increase the efficiency of software vendors. To develop high-quality cloud testing techniques, it is necessary to take into account the features of the cloud platform that will be used as a cloud testing environment.

Keywords: Testing Methodology, 1C:Bitrix, IaaS and SaaS services, cost reduction

Let's consider the process of developing a methodology using the example of cloud testing of the application 1C:Bitrix.

1C 8 Cloud Application Testing Methodology

Based on the analysis of known approaches to cloud software testing, taking into account their functional and architectural features, a methodology for cloud testing of applications implemented on the basis of a technology platform has been developed «1C: Bitrix » [1].

The methodology of cloud testing of applications «1C: Bitrix » "Site Management"

"Site Management" is a software product designed for the creation and management of online stores, business card sites, etc. There are a huge number of editions of this software product, but they are all designed for one purpose – to create a website and further maintain it (filling, editing, etc.) [1].

Activity diagram of the 1C application cloud testing methodology:Bitrix is shown in Figure 1.

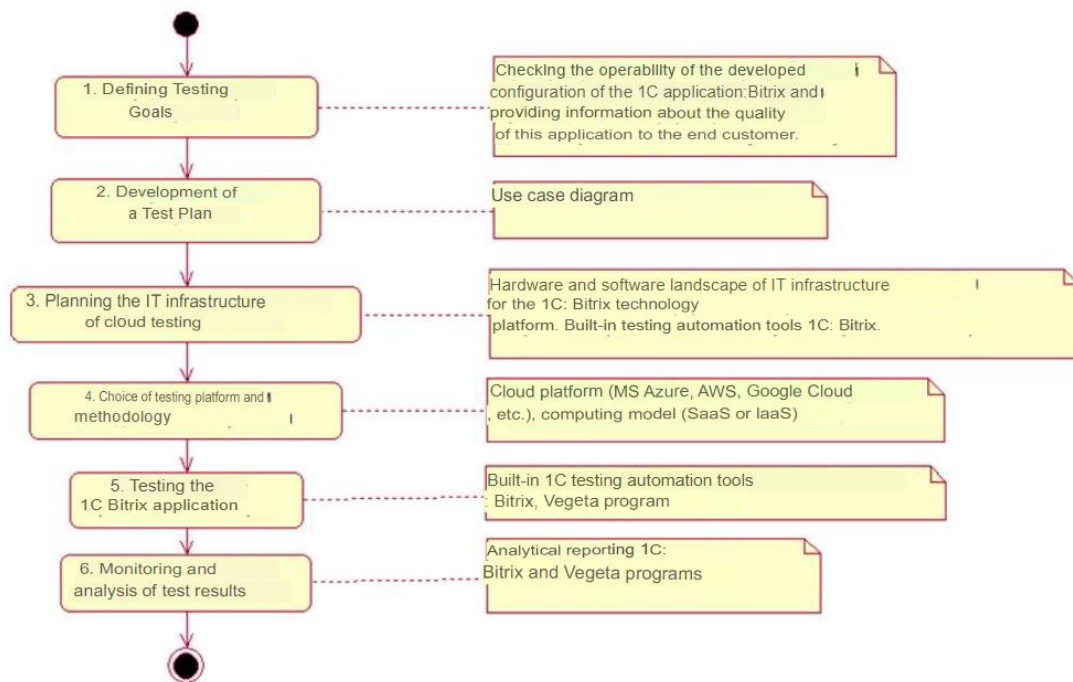


Figure 1 - Activity diagram of the application cloud testing methodology 1C:
Bitrix

Let's consider the stages of this technique[2]

1. Testing objectives: checking the operability of the developed configuration of the 1C application:Bitrix and providing information about the quality of this application to the end customer.

2. Development of a test plan.

Testing the 1C application:Bitrix includes the following types of testing:

- Web application testing;
- testing of the server part.

The actors in this case are: Tester, 1C testing automation tools:Bitrix, Vegeta program[4]

Use cases (use cases) are presented in Tables 1, 2.

Table 1 - Description of the use case: Testing a web application 1C: Bitrix

| |
|--|
| Use case: Testing a 1C:Bitrix web application |
| ID: 1 |
| Brief description: performing site speed testing, web application configuration testing, site page performance testing and testing on scaling. |
| Main actor: Tester |
| Secondary actors: Test automation tools 1C: Bitrix |
| Precondition: the use case starts at the initiative of the Tester |
| Basic flow: Tester Launches process testing web applications 1C: Bitrix |
| Postcondition: means automation testing 1C: Bitrix formed test results. |
| Alternate streams: no |

Table 2 - Description of the precedent: Testing the server part of the 1C: Bitrix application

| |
|---|
| Use case: Testing the server part of the 1C: Bitrix application |
| ID: 2 |
| Short description: performing load testing of the server part applications 1C8. |
| Main actor: Tester |
| Minor actor: Vegeta program |
| Precondition: the use case starts at the initiative of the Tester |
| Main flow: The tester starts the load testing process of the backend parts of the application 1C: Bitrix. |
| Postcondition: using the Vegeta program, the test results are generated. |
| Alternate streams: no |

A diagram of options for using the cloud testing plan for the 1C: Bitrix application, developed on the basis of the above recommendations, is shown in Figure 2[3]

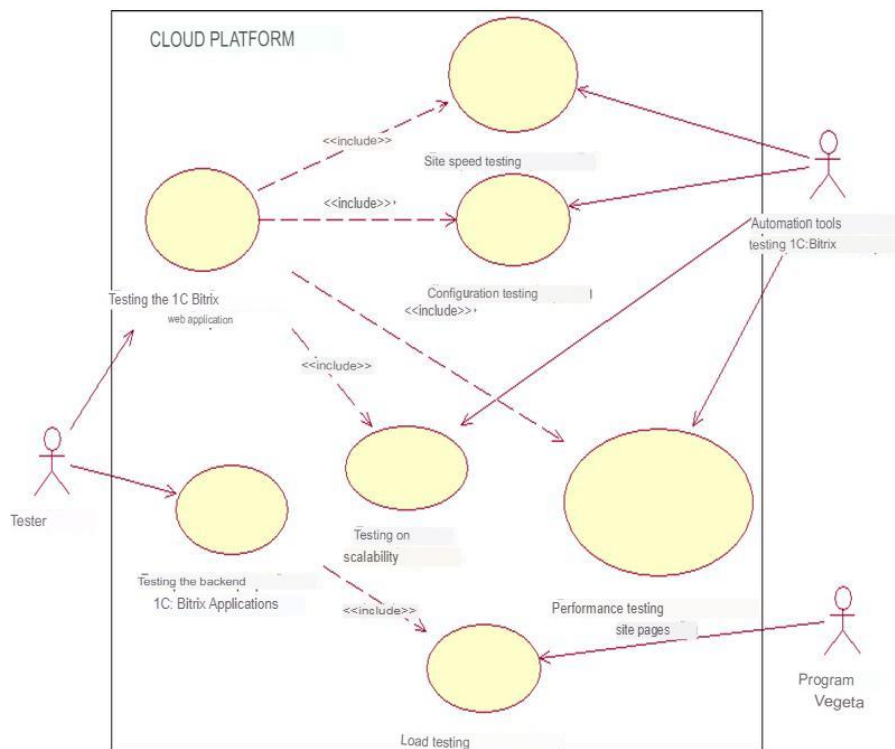


Diagram 2– Diagram of options for using the cloud testing plan for the 1C: Bitrix application

As follows from the diagram, in addition to the built-in testing tools of 1C: Bitrix, the Vegeta benchmark program is used to automate testing of the server part [2].

Vegeta is a free command-line tool for testing HTTP services written in the Go language. It can be connected as a library to create your own load testing tools.

1. Infrastructure planning.

The hardware and software landscape of the IT infrastructure of the testing environment is determined by the relevant requirements of the 1C platform: Bitrix, Vegeta program.

2. The choice of platform and cloud service provider is carried out in accordance with the recommendations given above. We choose SaaS or IaaS as methodologies.

3. Testing the application 1C: Bitrix. For automation, built-in 1C test automation tools are used: Bitrix and the Vegeta program.

4. Monitoring and analysis of test results.

To collect, process and visualize the results of cloud testing, the built-in 1C tools: Bitrix and the Vegeta program are used, which allows you to create reports in text and graphic format.

Conclusions to the third chapter. To automate cloud testing of the 1C8 application, the built-in test automation tools of the 1C: Enterprise 8 technology platform are used. To collect, process and visualize the results of cloud testing, it is recommended to use an Excel spreadsheet[6]

In the process of cloud testing of the 1C: Bitrix application, in addition to the built-in testing tools of this platform, the Vegeta benchmark program is used to automate testing of the server side. To collect, process and visualize the results of cloud testing, the built-in tools of the 1C: Bitrix content management system and the Vegeta program are used, which allows you to create reports in text and graphic format.

Evaluation of the effectiveness of the cloud testing methodology for 1C:Bitrix applications. Consider the results of cloud testing of an IT solution based on the 1C: Bitrix application.

AT quality cloudy platforms chosen platform MS
azure,providing IaaS and SaaS services.

1) Website speed testing (figure 3).

This page displays site load statistics collected while users are browsing the site.

Functions:

- displays the download speed for users from different regions;
- allows you to find out the minimum, average and maximum page loading speed;
- allows you to see which subsystems affect the page rendering speed: server upload speed, dns server, html processing, other processing.

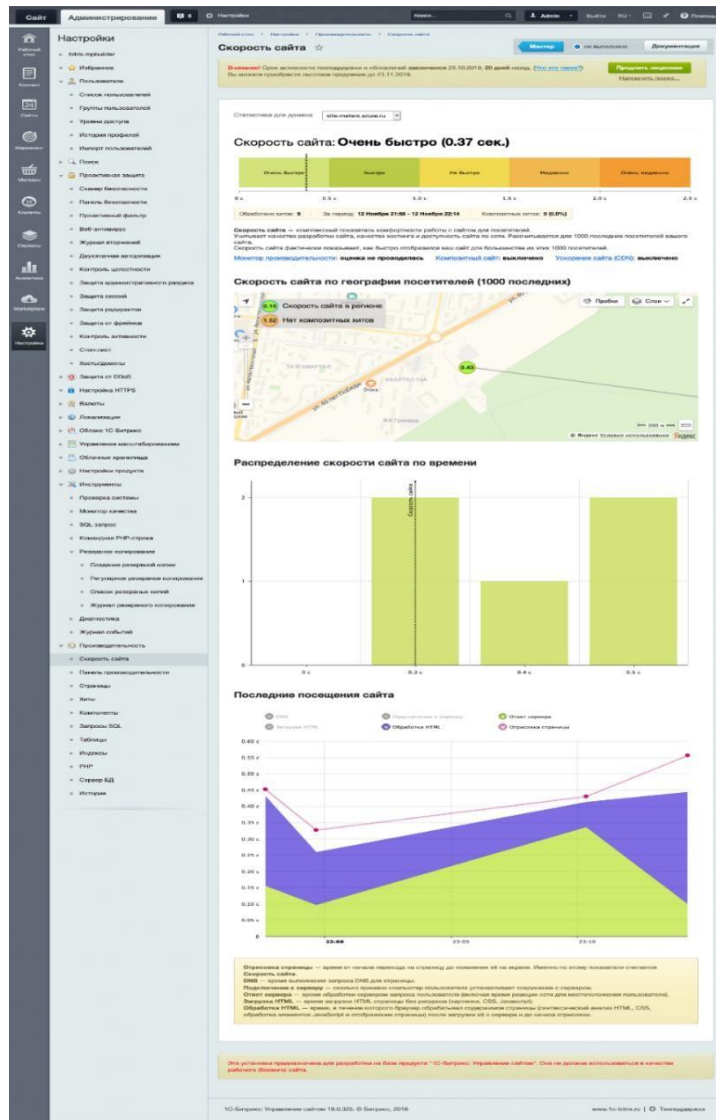


Figure 3 - Page for testing the speed of the site 1C: Bitrix

2) Configuration testing.

Allows you to determine how much hosting is enough for optimal system operation.

The processor, memory, file subsystem are checked. The work of php, mail, MySQL is also checked (Figure 4)

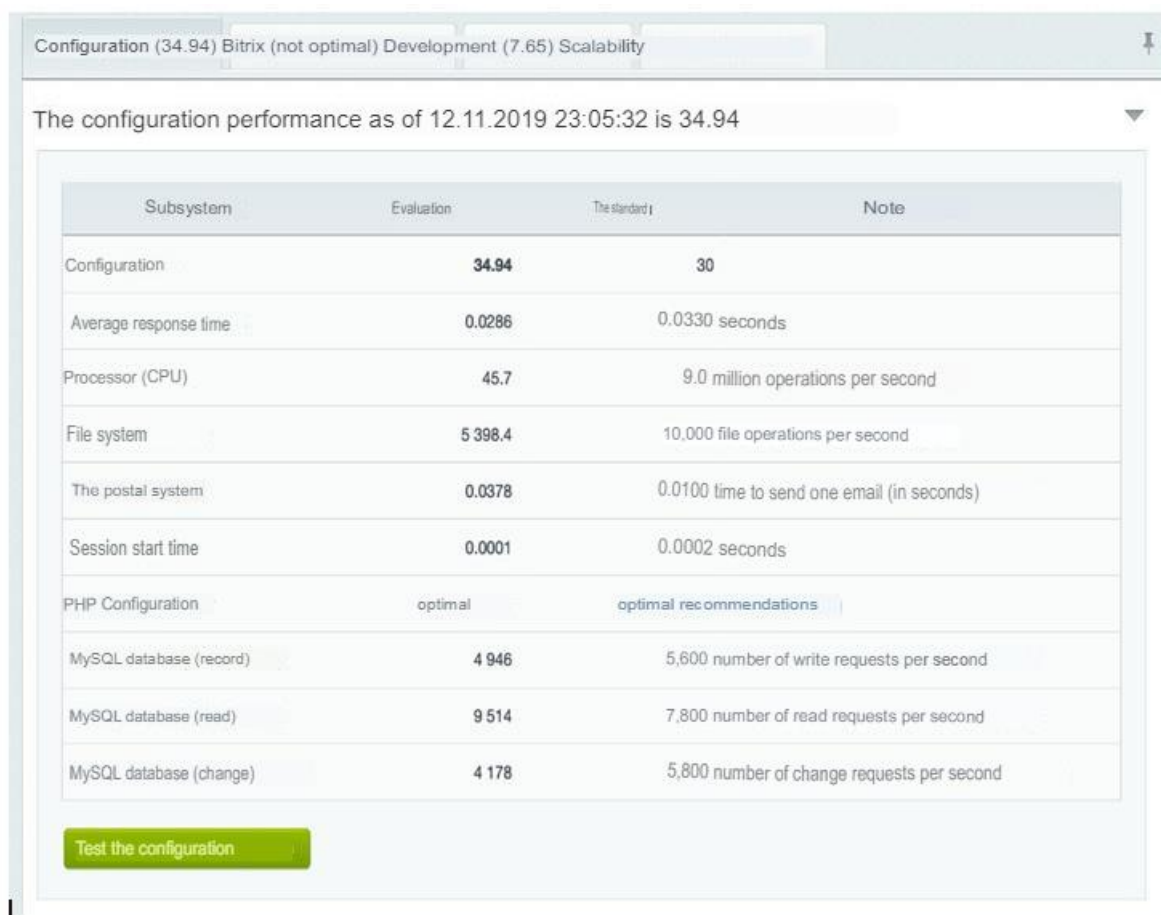


Figure 4 - Fragment of the web configuration testing page applications

1) Testing the performance of site pages (Figure 5). Runs on the development panel.

Allows you to find slow site pages, identify how elements on the site slow down its work, find a disabled cache for "heavy" pages, etc.

2) Scalability test.

Allows you to get for one or more server solutions basic metrics for returned pages per second and page generation time (Figure 6).

3) Load testing of servers.

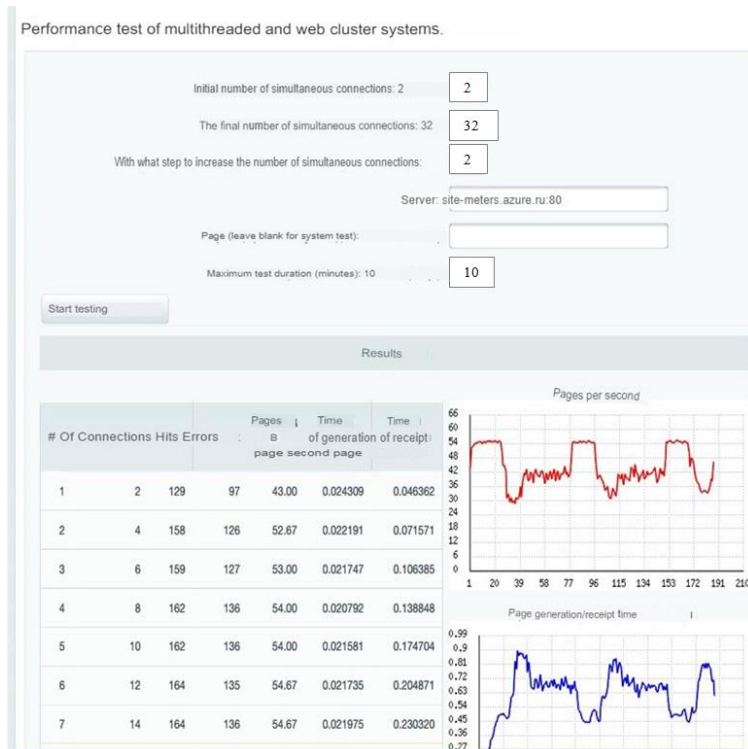


Figure 5 - Fragment of the performance testing page
site pages

For server load testing, it is better to use external testing applications.

Since they better allow us to calculate the maximum capabilities of the service, unnecessary server resources are not wasted on the monitoring system inside the service.

Average performance 7.65 (measured 300 seconds, 70 hits) 12.11.2019 23:07:42

| 20 most loading pages | Mistakes development2 | Load | Number of hits | Average time (sec.) |
|--|-----------------------|--------|----------------|---------------------|
| /api/index.php | | 58.28% | 25 | 0.2133 |
| /api/settings/index.php | 1 | 11.90% | 15 | 0.0726 |
| /personal/index.php | | 7.00% | 2 | 0.3202 |
| /home/bitrix/www/bitrix/modules/main/tools/eron_events.php | | 5.18% | 2 | 0.2370 |
| /index.php | 1 | 4.55% | 5 | 0.0832 |
| /buy/index.php | 1 | 4.47% | 8 | 0.0512 |
| /catalog/detail/index.php | | 3.66% | 3 | 0.1116 |
| /catalog/color/index.php | | 2.40% | 6 | 0.0366 |
| /profile/index.php | | 1.39% | 3 | 0.0424 |
| /history/index.php | | 1.16% | 1 | 0.1065 |

All pages

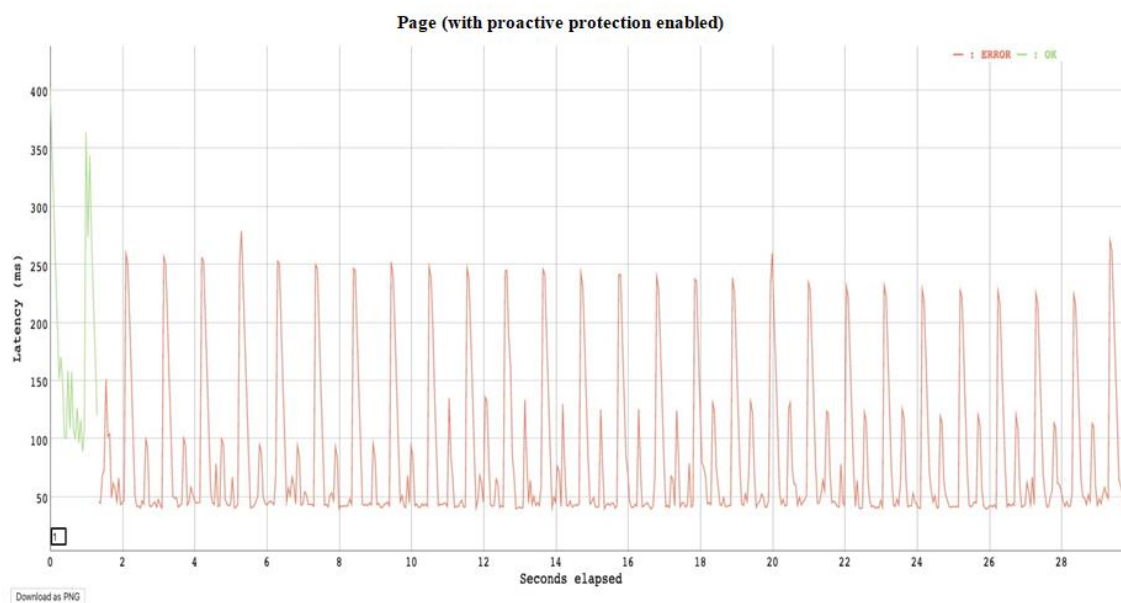
Figure 6 - Fragment of the scalability testing page

For this purpose, we use the Vegeta application.

It should be noted that the product "1C-Bitrix: Site Management" implements a mechanism that allows you to connect any "clouds" to the site and easily manage them - up to the exchange of data between storages[7]

Consider reports of test results using the Vegeta program.

Graphs of load testing results for pages with enabled proactive protection (allows you to cut off too frequent access to the server from the user) are shown in Figure 7.



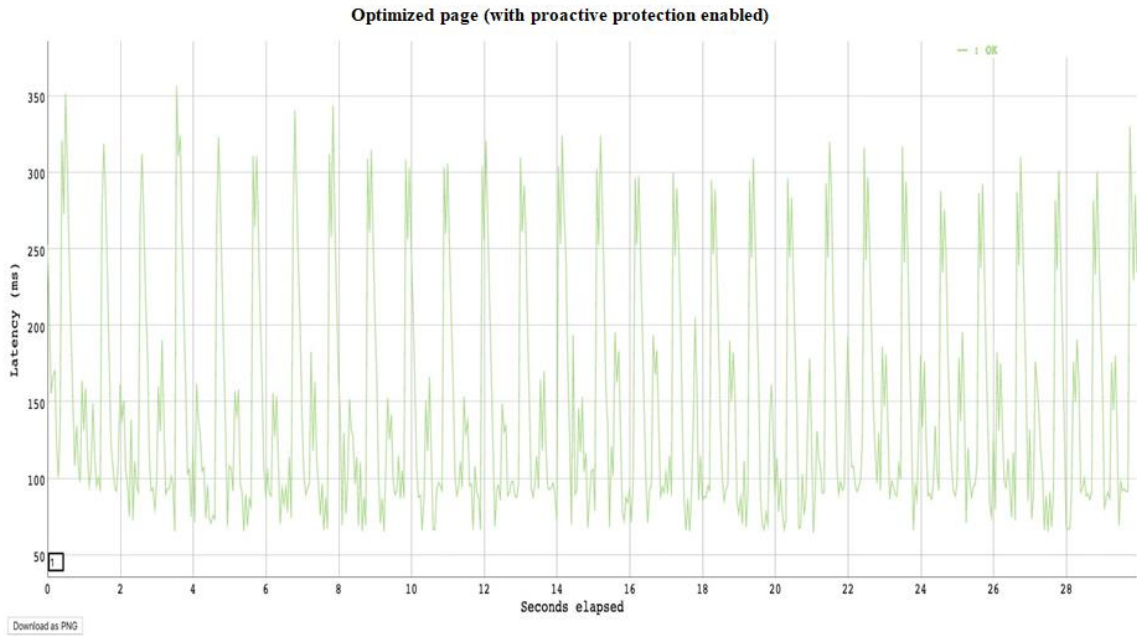
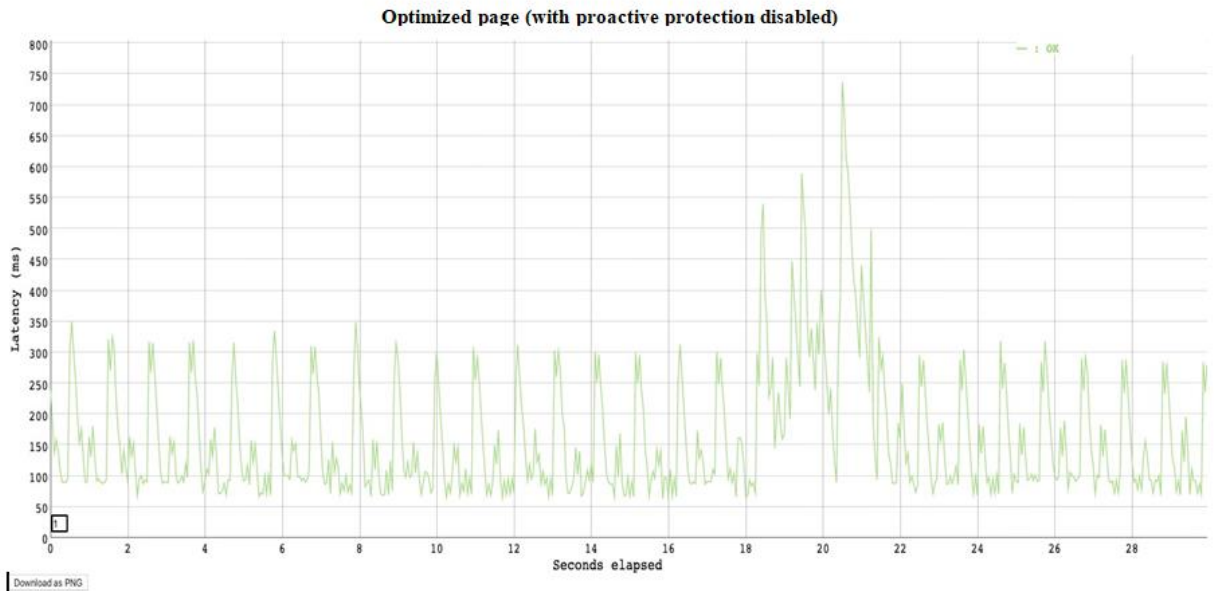


Figure 7 Graphs of load testing results for pages with enabled proactive protection

Graphs results load testing for pages with disabled proactive protection are shown in Figure 8.



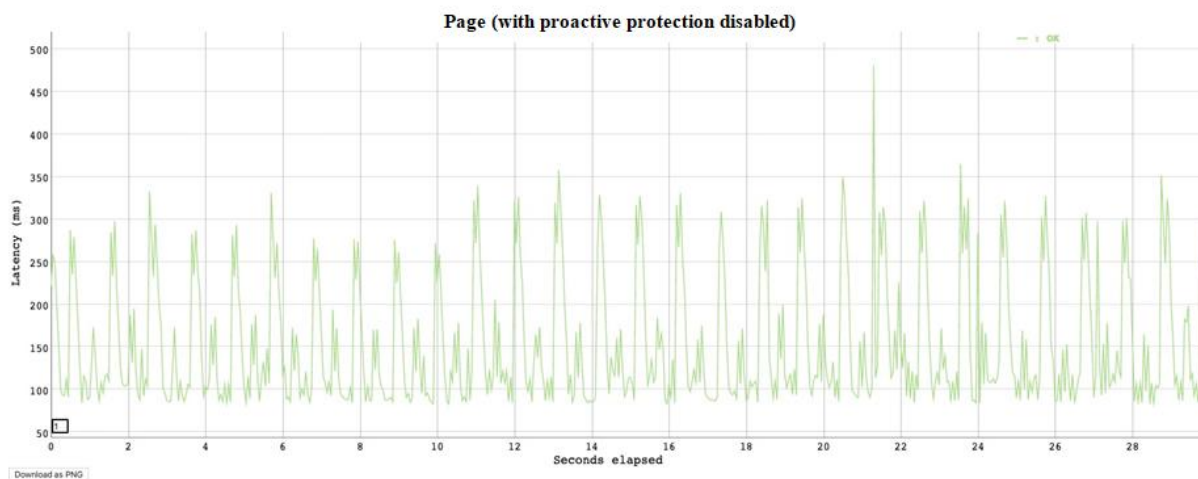


Figure 8 – Graphs of load testing results for pages with enabled proactive protection

Figure 9 shows the Vegeta program commands for displaying test results.

```
echo "GET http://site-meters.azure.ru/" | vegeta attack -duration=30s -
rate=20 | tee results.bin vegeta report
vegeta plot -title=Results results.bin > results-plot1.html
echo "GET http://site-meters.azure.ru/optimaze\_page/" | vegeta attack
duration=30s -rate=20 | tee results.bin vegeta report
vegeta plot -title=Results results.bin > results-plot2.html
echo "GET http://site-meters.azure.ru/" | vegeta attack -duration=30s -
rate=20 | tee results.bin vegeta report
vegeta plot -title=Results results.bin > results-plot3.html
echo "GET http://site-meters.azure.ru/optimaze\_page/" | vegeta attack
duration=30s -rate=20 | tee results.bin vegeta report
vegeta plot -title=Results results.bin > rcsults-plot4.html
```

Figure 9 - Commands of the Vegeta program for displaying test results

Thus, the proposed method provides testing of various types of software, which confirms its versatility and effectiveness.

Conclusions: When choosing a cloud for 1C, it is recommended to choose a specialized hosting provider that can provide the most productive platform for the user information system.

The proposed method provides testing of various types of software, which confirms its versatility and effectiveness.

List:

1. 1C:Bitrix [Electronic resource]. <https://www.1c-bitrix.ru/>

2. Open benchmarks for load testing of servers and web applications [Electronic resource]. <https://habr.com/ru/company/1cloud/blog/474474/>
3. Cloud Computing and Testing Cloud based Applications [Electronic resource]. <https://www.360logica.com/blog/cloud-computing-and-testing-cloud-based-applications/>
4. SaaS vs PaaS vs IaaS: What's The Difference and How To Choose [Electronic resource]. —<https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>
5. Cloud Testing Tutorial with SaaS and PaaS [Electronic resource]. <https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>
6. Kotlyarov V. P. Fundamentals of software testing [Electronic resource] / V. P. Kotlyarov. — M.: Internet University of Information Technologies (INTUIT), 2016. — 334 p.
7. Burnyashov B. Ah. Information technologies in management. Cloudburneы [элект resource]: textbook posobie / B. Ah. Burnя this. - 2nd ed. - Saratov: Wuzovskoe education, 2019. — 87 с.