

**ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КЛАСТЕРА
С КОНТЕЙНЕРНОЙ ВИРТУАЛИЗАЦИЕЙ****В. К. Фунг¹, В. А. Богатырев^{1,2*}**¹ Университет ИТМО, Санкт-Петербург, Россия² Санкт-Петербургский государственный университет аэрокосмического приборостроения,
Санкт-Петербург, Россия

* vladimir.bogatyrev@gmail.com

Аннотация. Экспериментально исследована интенсивность обслуживания запросов при автоматическом разделении ресурсов компьютерной системы между активными, т. е. используемыми в текущий момент времени, контейнерами. На первом этапе экспериментов выполняется формирование и развертывание сервиса (веб-сервера) на кластере. На втором этапе осуществляется запуск тестовой программы с сохранением результатов в лог-файле. Предложены алгоритмы проведения экспериментов на первом и втором этапе. Эксперименты проведены в лабораторной среде облачной виртуализации Proxmox и средств управления кластером Kubernetes. Для автоматизации развертывания и обновления конфигурации сервиса на кластере Kubernetes разработана программа на языке программирования Python. Программа использует библиотеки SCP и PARAMIKO для удаленного развертывания и обновления сервиса. Выявлено влияние распределения ограниченных возможностей ресурсов системы между активными и развернутыми в системе (виртуальной машине) контейнерами на интенсивность обслуживания запросов. Полученные результаты могут быть использованы при разработке аналитической модели обслуживания системы с контейнерной виртуализацией, в том числе виртуальных кластерных систем.

Ключевые слова: кластер, Kubernetes, интенсивность обслуживания, Docker, контейнер, виртуализация, виртуальная машина

Ссылка для цитирования: Фунг В. К., Богатырев В. А. Экспериментальное исследование производительности кластера с контейнерной виртуализацией // Изв. вузов. Приборостроение. 2024. Т. 67, № 8. С. 647–656. DOI: 10.17586/0021-3454-2024-67-8-647-656.

EXPERIMENTAL STUDY OF CLUSTER PERFORMANCE WITH CONTAINER VIRTUALIZATION**Phung Van Quy¹, V. A. Bogatyrev^{1,2*}**¹ ITMO University, St. Petersburg, Russia² Saint Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia

* vladimir.bogatyrev@gmail.com

Abstract. The intensity of query servicing during automatic division of computer system resources between active, i.e. currently used, containers is investigated experimentally. At the first stage of experiments, the formation and deployment of a service (web server) on a cluster is performed. At the second stage, a test program is launched with the results saved in a log file. Algorithms for conducting experiments at the first and second stages are proposed. The experiments are conducted in a laboratory environment of Proxmox cloud virtualization and Kubernetes cluster management tools. A program in the Python programming language is developed to automate the deployment and update of the service configuration on the Kubernetes cluster. The program uses the SCP and PARAMIKO libraries for remote deployment and update of the service. The influence of the distribution of limited system resource capabilities between active and deployed containers in the system (virtual machine) on the intensity of request servicing was revealed. The obtained results can be used in the development of an analytical model for servicing a system with container virtualization, including virtual cluster systems.

Keywords: cluster, Kubernetes, servicing intensity, Docker, container, virtualization, virtual machine

For citation: Quy Phung Van, Bogatyrev V. A. Experimental study of cluster performance with container virtualization. *Journal of Instrument Engineering*. 2024. Vol. 67, N 8. P. 647–656 (in Russian). DOI: 10.17586/0021-3454-2024-67-8-647-656.

Введение. Снижение рисков, связанных с цифровой трансформацией [1, 2], сопряжено, прежде всего, с возрастающими требованиями к отказоустойчивости и надежности [3, 4] современных распределенных компьютерных систем [5]. Для распределенных систем реального времени, в том числе промышленных и киберфизических [6, 7], дополнительные риски могут быть связаны с требованием выполнения запросов за ограниченное время [8, 9]. Обеспечению высокой надежности, отказоустойчивости современных инфокоммуникационных систем служит их адаптивная реконфигурация [10] с перераспределением трафика [11, 12].

Один из способов повышения отказоустойчивости, надежности и производительности компьютерных систем — объединение ресурсов в кластер [13–15] с сохранением его высокой доступности, малых задержек и высокой надежности передачи данных [16, 17].

Технология виртуальных машин, позволяющая при отказах физических серверов восстанавливать работоспособность кластера в случае миграции виртуальных машин с отказавших физических серверов на исправные, обеспечивает повышение надежности и отказоустойчивости кластерных систем. В [18–21] предложены марковские модели кластеров, обеспечивающие работоспособность кластеров с миграцией виртуальных машин. Модели оценки производительности отказоустойчивых кластеров рассмотрены в статьях [22–24]; влияние на производительность серверов миграции виртуальных машин в процессе восстановления работоспособности кластеров — в работах [25–27].

В последнее время концепция построения высоконадежных отказоустойчивых кластеров с использованием виртуальных машин получила развитие на основе технологии контейнерной виртуализации.

Виртуальная контейнеризация позволяет упаковывать приложения в контейнеры, обеспечивая наличие необходимых средств для их выполнения. В отличие от виртуальных машин, контейнеры используют общие ядра операционной системы только в рамках одного процесса и одной файловой системы. Это обеспечивает меньшие затраты ресурсов на поддержание контейнеров, делает их быстро запускаемыми и масштабируемыми, что открывает дополнительные возможности высокоскоростной реконфигурации и восстановления функционирования после обнаружения отказов оборудования и программного обеспечения.

Контейнеризация позволяет управлять ресурсами эффективно, запуская и масштабируя приложения по мере необходимости, что способствует поддержанию высокой производительности даже при растущей нагрузке.

Контейнеры создают изолированные среды выполнения, что позволяет избежать конфликтов между приложениями и обеспечивает предсказуемое поведение даже при изменениях в окружении или конфигурации. Кроме того, системы управления контейнерами, такие как Kubernetes, предоставляют средства для автоматического восстановления приложений в случае сбоев или неполадок, что повышает надежность системы в целом.

Сравнительная оценка производительности контейнеров и виртуальных машин на различных аппаратных платформах, включая различные тестовые эталонные примеры (бенчмарки) и различные типы платформ контейнеризации, проведена в статье [28]. Предложенный в работе [29] метод тестирования безопасности веб-приложений, в котором использовались контейнеры Docker при динамическом и статическом анализе безопасности, а также проверке зависимостей, продемонстрировал потенциал контейнеризации для обеспечения безопасности информационных систем. Эффективность технологий виртуальных машин и контейнерной виртуализации в вычислительных системах с анализом влияния этих технологий на производительность и масштабируемость приложений в критически важных областях рассмотрена в работе [30]. Эффективность применения виртуальных машин и контейнеров приложений для облачных платформ по таким критериям, как производительность, масштабируемость и изоляция пользователей, проанализирована в работе [31].

Особенностью контейнерной виртуализации, реализуемой в виртуальной машине, является динамическое распределение выделяемых ресурсов между активными, т. е. задействованными в процессе обслуживания запросов, контейнерами. При этом ресурсы выделяются по мере

обращения к контейнерам и закрепляются за ними на требуемое время. В настоящее время обоснование построения моделей, отражающих указанную особенность динамического разделения ресурсов системы между активными контейнерами при обслуживании запросов, требует дополнительных исследований.

Представленные в настоящей статье исследования являются развитием выполненных в [32] работ по экспериментальному определению интенсивности обслуживания запросов контейнерами. В настоящей работе предложена усовершенствованная двухэтапная методика исследования влияния совместного использования ограниченных ресурсов системы между активными и развернутыми в системе (виртуальной машине) контейнерами на интенсивность обслуживания запросов.

Средства проведения эксперимента. Целью эксперимента является анализ зависимости интенсивности обслуживания запросов каждым контейнером от общего числа контейнеров, сформированных в системе, и числа активных контейнеров, участвующих в выполнении запросов.

Эксперимент проведен с использованием стенда, содержащего компьютер с четырьмя процессорами Intel(R) Core(TM) i5-4570 @ 3,20 ГГц, работающий под управлением ядра Linux версии 6.2.16-3-pve на платформе Proxmox. Эксперимент проведен для виртуального сервера с одним виртуальным ядром, одним виртуальным сокетом и 4 ГБ оперативной памяти.

На одном узле кластера, который представляет собой изолированную виртуальную машину (рис. 1), развернут один сервис. В системе развернуто n контейнеров, функционирующих как однопоточные HTTP веб-серверы. Для исследования интенсивности обслуживания каждого контейнера в условиях наличия m активных контейнеров необходимо обеспечить одновременную работу (активность) m контейнеров. Это достигается с помощью тестовой программы, которая отправляет m запросов одновременно, а затем записывает в лог-файл информацию о времени обработки запроса каждым контейнером. Эта информация содержится в ответе на каждый запрос, отправленный тестовой программой.

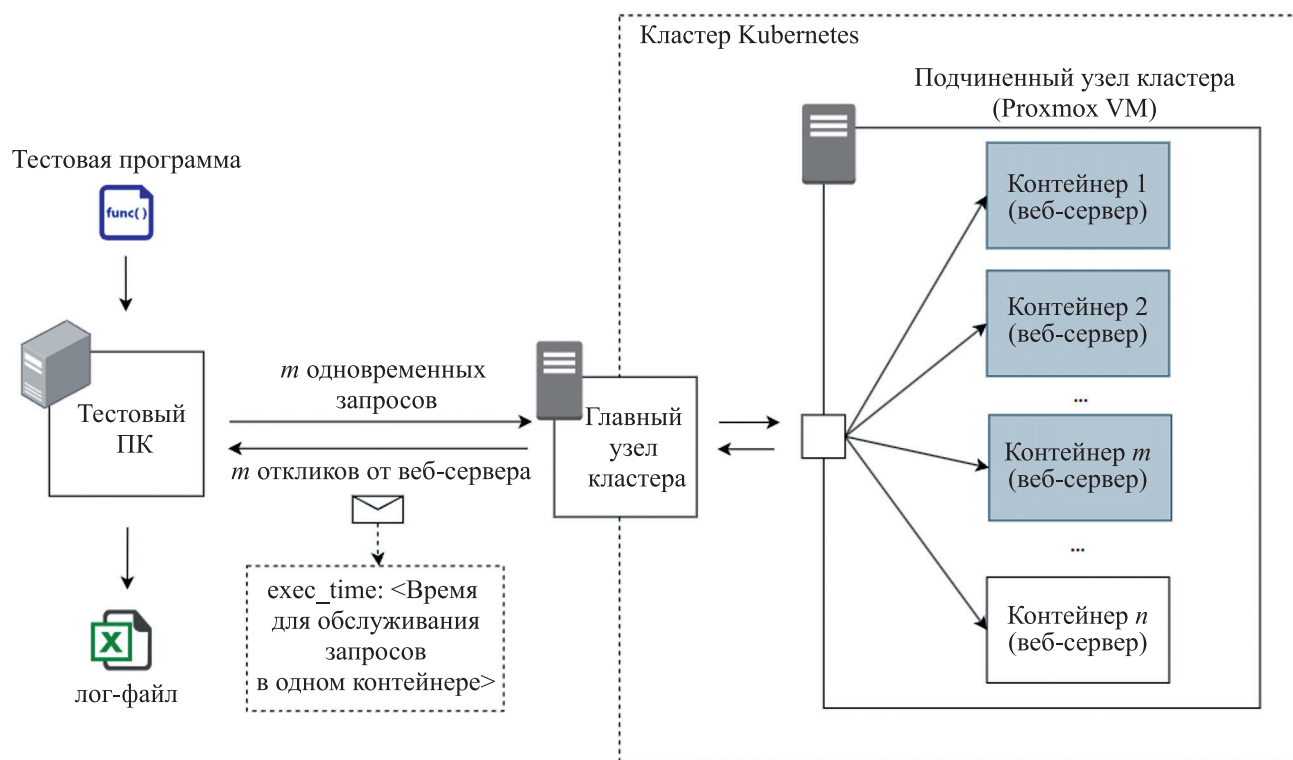


Рис. 1

Методика проведения эксперимента. Предлагаемая методика проведения эксперимента включает два этапа.

I. Создание и развертывание сервиса (веб-сервера) на кластере.

II. Создание и запуск тестовой программы с формированием результатов в лог-файле.

Алгоритм реализации эксперимента на этапе I методики (рис. 2) предусматривает выполнение следующих процедур.

1. Формируется исходный код веб-сервера образа контейнера. Для развертывания сервиса используется Dockerfile [33], при этом указываются версии применяемого программного обеспечения, производятся установка и настройка необходимых средств.
2. Полученный образ контейнера загружается в реестр контейнеров [34] (Container Registry). Реестр обеспечивает доступ к образам из любого места и облегчает автоматизацию процесса развертывания приложений в контейнерах (в качестве реестра контейнеров в настоящей работе используется DockerHub).
3. После того как образ контейнера становится доступным в реестре, создается файл конфигурации развертывания (Deployment.yaml). Этот файл определяет число реплик, стратегии обновления и образ контейнера из реестра контейнеров. Deployment.yaml упрощает процесс развертывания и управления приложением в кластере, повышая его надежность и масштабируемость.
4. После получения полного файла Deployment.yaml в кластере Kubernetes приложение развертывается при помощи API Kubernetes или широко используемого инструмента развертывания Helm. Helm представляет собой механизм управления пакетами приложений, он обеспечивает упрощенную установку, управление настройками виртуализации с возможностью отката к предыдущим значениям, параметризацию и обновление приложений, что существенно повышает эффективность и масштабируемость процесса управления приложениями в Kubernetes.



Рис. 2

Рис. 3, а иллюстрирует процесс создания образа контейнера, а рис. 3, б — его загрузку в реестр контейнеров. На рис. 4 представлен результат развертывания сервиса на узле кластера.

В ходе эксперимента после завершения этапа I методики выполняется повторный запуск команды инструмента Helm для изменения числа контейнеров или реплик приложения. Для автоматизации этого процесса во время выполнения теста потребовался инструмент, позволяющий тестовой программе подключаться к кластеру и управлять этими настройками, — в настоящей работе использованы библиотеки Python SCPClient. Этот инструмент позволяет устанавливать связь с главным узлом кластера и выполнять команды Helm для изменения конфигурации приложения, а также для повторного развертывания приложения с различными параметрами, включая число реплик (контейнеров).

a)

```
$ make build_image
docker build -t phungvanquy/loadtest .
[+] Building 3.2s (9/9) FINISHED
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 363B
=> [internal] load metadata for docker.io/library/python:3.9-alpine
=> [auth] library/python:pull token for registry-1.docker.io
=> [1/3] FROM docker.io/library/python:3.9-alpine@sha256:99161d2323b4130fed2d849dc8ba35274d1e1f35da170435627b21d305dad954
=> [internal] load build context
                                0.0s
=> => exporting layers
=> => writing image sha256:4607aabe9f6659451eb351604865edf439fa6ae78d10ba26050564e906279c8
=> => naming to docker.io/phungvanquy/loadtest

$ make publish_image
docker tag phungvanquy/loadtest:latest phungvanquy/loadtest:latest
docker push phungvanquy/loadtest:latest
The push refers to repository [docker.io/phungvanquy/loadtest]
a285dc5edc9c: Layer already exists
e4fa09a2ee3d: Layer already exists
4857056bad11: Layer already exists
69141b6c4721: Layer already exists
0bbac9765c1f: Layer already exists
4c9c2b9681ab: Layer already exists
d4fc045c9e3a: Layer already exists
latest: digest: sha256:df2eff0b271c37b9ca5fc0b30c87993a698dc407720418d012e2d5f4f9f33ebf size: 1781
```

b)

The screenshot shows the Docker Hub interface for the repository `phungvanquy/loadtest`. The page includes a search bar, navigation tabs (General, Tags, Builds, Collaborators, Webhooks, Settings), and a main content area. The main content area displays the repository name, a 'test' tag, and a 'Docker commands' section with the command `docker push phungvanquy/loadtest:tagname`. There is also an 'Automated Builds' section with a description and an 'Upgrade' button.

Рис. 3

The screenshot shows a terminal window with the following output:

```
phungvanquy@test-cluster:~/loadtest$ helm install loadtest ./loadtest --set replicas=5 --wait
NAME: loadtest
LAST DEPLOYED: Sat May 4 11:44:02 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
phungvanquy@test-cluster:~/loadtest$
```

Below the terminal output, there is a table showing the deployment status of the application:

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GA
loadtest-deployment-844c55c569-rs5zd	1/1	Running	0	3m20s	10.42.7.74	slave7	<none>	<none>
loadtest-deployment-844c55c569-wkfdm	1/1	Running	0	3m20s	10.42.7.75	slave7	<none>	<none>
loadtest-deployment-844c55c569-z7wk4	1/1	Running	0	3m20s	10.42.7.76	slave7	<none>	<none>
loadtest-deployment-844c55c569-6bjgl	1/1	Running	0	3m20s	10.42.7.77	slave7	<none>	<none>
loadtest-deployment-844c55c569-x4d2t	1/1	Running	0	3m20s	10.42.7.78	slave7	<none>	<none>

Рис. 4

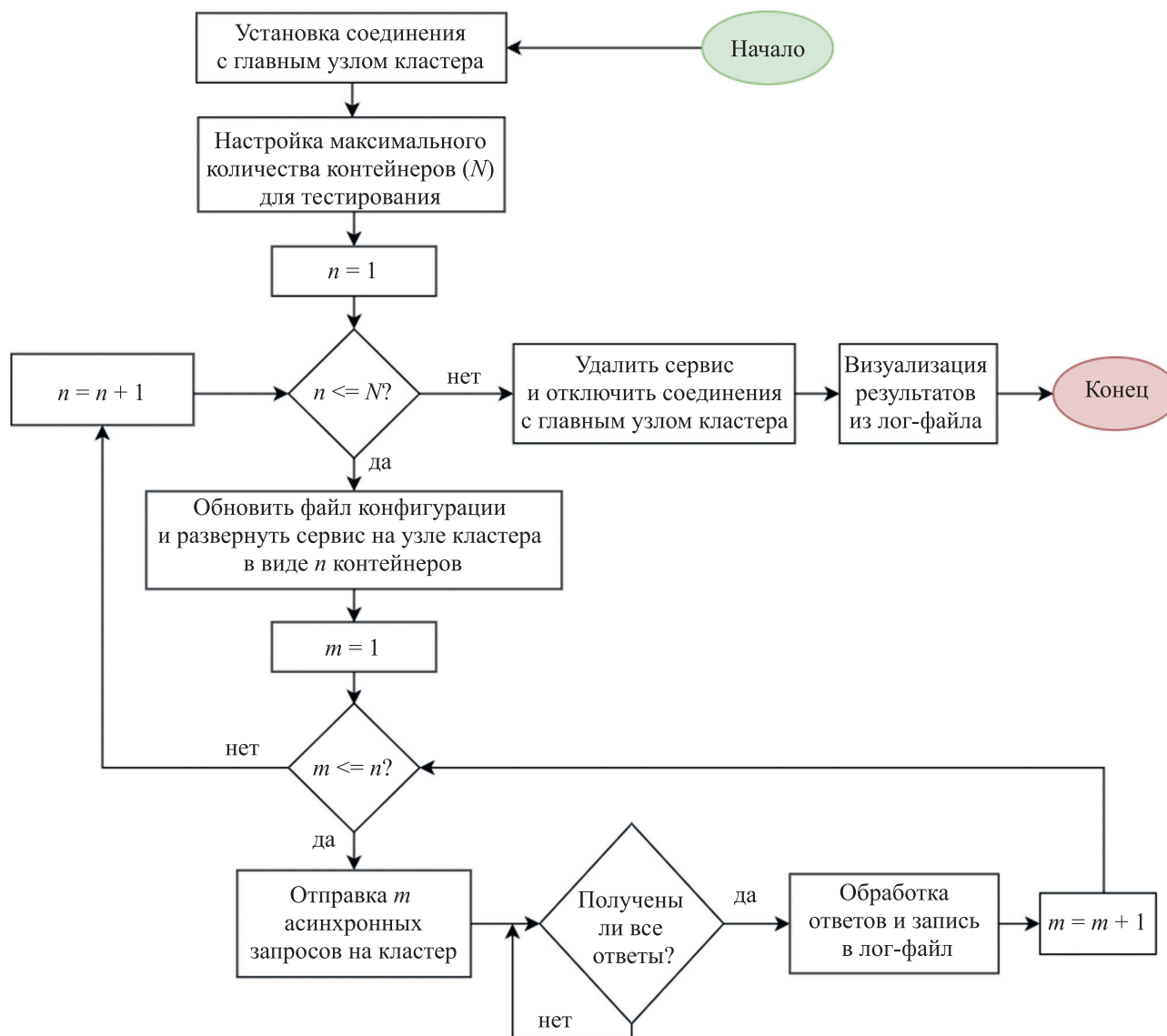


Рис. 5

Эксперимент этапа II методики выполняется в соответствии с алгоритмом, блок-схема которого приведена на рис. 5. После инициализации тестовой программы необходимо установить соединение с кластером для удаленного управления процессом развертывания сервиса. Каждый раз при изменении значения переменной n (число развернутых контейнеров), где $1 \leq n \leq N$ (N — максимальное число контейнеров для эксперимента), происходят процессы обновления файла конфигурации и развертывания сервиса. Для каждого n осуществляется отправка m ($1 \leq m \leq n$) одновременных запросов на кластер с целью обеспечения активности m контейнеров. Затем обрабатываются ответы от кластера, содержащие время обслуживания запроса в каждом контейнере. Результаты обработки усредняются, а обратные значения представляют собой среднюю интенсивность обслуживания каждого контейнера при n развернутых и m активных контейнерах.

Результаты второго этапа эксперимента представлены на рис. 6. Полученные данные описаны в файле `processing_time.csv` (рис. 6, а) в следующем формате: „<Количество развернутых контейнеров>, <Количество активных контейнеров>, <время обслуживания запросов в одном контейнере>“. На рис. 6, б проиллюстрирована зависимость интенсивности обслуживания от числа контейнеров. Полученные данные показывают, что интенсивность обслуживания запросов каждым контейнером $\mu(n, m)$ существенно снижается с увеличением числа активных контейнеров m . При одинаковом числе активных контейнеров интенсивность обработки контейнера оказывается выше, когда количество развернутых контейнеров n меньше.

a)

```

phung@DESKTOP-1TN808J MSYS /e/Main/PhD/MY_PAPER/Paper_4/source_code
$ python main.py
Command output:
NAME: loadtest
LAST DEPLOYED: Mon May 6 08:17:23 2024
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None

Testing with 1 containers...
processing time: 0.04795646667480469
processing time: 0.04504227638244629
processing time: 0.04404306411743164
processing time: 0.04422736167907715
processing time: 0.04475092887878418
processing time: 0.04456734657287598
processing time: 0.04530477523803711
processing time: 0.04499316215515137
processing time: 0.046623945236206055
processing time: 0.044248104095458984
processing time: 0.044509172439575195
processing time: 0.0447545051574707
processing time: 0.043845415115356445
processing time: 0.04502105712890625
processing time: 0.045273780822753906
processing time: 0.04488420486450195
processing time: 0.04462623596191406
processing time: 0.04764604568481445
processing time: 0.044855594635009766
processing time: 0.044187307357788086

phung@DESKTOP-1TN808J MINGW64 /e/Main/PhD/MY_PAPER/Paper_4/source_code
$ cat processing_time.csv
1,1,0.04483171075384733
2,1,0.04494697603081285
2,2,0.06680559119354695
3,1,0.04495987906134373
3,2,0.07447734390117614
3,3,0.08482721019088036
4,1,0.04518440436079817
4,2,0.0779010767916447
4,3,0.09547542732800487
4,4,0.10404659086067862
5,1,0.0451711038685311
5,2,0.08030794677837594
5,3,0.10202490792447003
5,4,0.11536908772796232
5,5,0.12777956369513513
6,1,0.045254836243063486
6,2,0.08145403479000416
6,3,0.10626920702092184
6,4,0.12435894563633909
6,5,0.1370989684290534
6,6,0.14685560187298446
7,1,0.04530305951821837
7,2,0.0830054902104169
7,3,0.1111775836274948
7,4,0.12874170543598706
7,5,0.14788187586344206
7,6,0.15788678648651286
7,7,0.16717443173502295
8,1,0.0454759087562561

```

б)

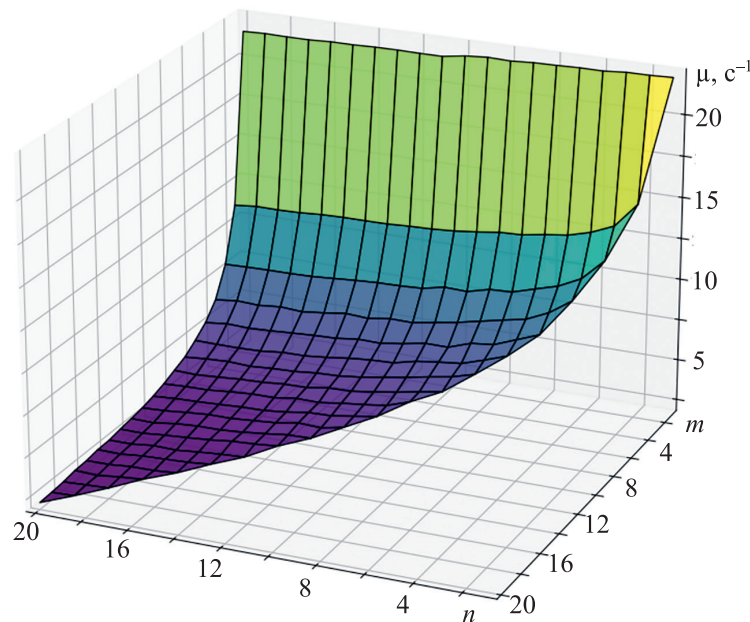


Рис. 6

Заключение. Для компьютерных систем с контейнерной виртуализацией экспериментально установлено влияние работы контейнеров на интенсивность их обслуживания при автоматическом разделении ресурсов системы между активными контейнерами. Определено влияние изменения общего числа контейнеров на интенсивность обработки запросов.

Полученные результаты могут быть использованы для разработки аналитической модели обслуживания компьютерной системы с контейнерной виртуализацией, в том числе кластерных систем виртуальных машин с развертыванием в них контейнеров.

СПИСОК ЛИТЕРАТУРЫ

1. Зинина Т. С., Рудник П. Б. Цифровая трансформация: ожидания и реальность. М.: Изд. дом Высшей школы экономики, 2022. 221 с.
2. Емельянов А. А., Кориунов И. Л. Технические риски предприятия, связанные с цифровой трансформацией // Изв. вузов. Приборостроение. 2024. Т. 67, № 2. С. 116–121.
3. Половко А. М., Гуров С. В. Основы теории надежности. СПб: БХВ-Петербург, 2006. 702 с.
4. Черкесов Г. Н. Надежность аппаратно-программных комплексов. СПб: Питер, 2005. 479 с.
5. Херсонский Н. С., Большедворская Л. Г. Анализ развития статистических методов оценки надежности как свойства качества объектов и продукции // Crede Experto: транспорт, общество, образование, язык. 2024. № 1. С. 6–23. DOI: 10.51955/2312-1327_2024_1_6.
6. Astakhova T., Shamin A., Verzun N., Kolbanev M. A. A model for estimating energy consumption seen when nodes of ubiquitous sensor networks communicate information to each other // CEUR Workshop Proceedings. 2019. Vol. 2344, N 2.
7. Gurjanov A. V., Korobeynikov A. G., Zharinov I. O., Zharinov O. O. Edge, fog and cloud computing in the cyber-physical systems networks // III International Workshop on Modeling, Information Processing and Computing. 2021. P. 103–108.
8. Богатырев В. А., Богатырев С. В., Богатырев А. В. Оценка готовности компьютерной системы к своевременному обслуживанию запросов при его совмещении с информационным восстановлением памяти после отказов // Науч.-техн. вестн. информационных технологий, механики и оптики. 2023. Т. 23, № 3. С. 608–617.
9. Feng, D., Lai, L., Luo, J. et al. Ultra-reliable and low-latency communications: applications, opportunities and challenges // Sci. China Inf. Sci. 2021. Vol. 64. P. 1–12. DOI: 10.1007/s11432-020-2852-1.
10. Богатырев В. А. Надежность и эффективность резервированных компьютерных сетей // Информационные технологии. 2006. № 9. С. 25–30.
11. Богатырев В. А., Богатырев А. В. Оптимизация резервированного распределения запросов в кластерных системах реального времени // Информационные технологии. 2015. Т. 21, № 7. С. 495–502.
12. Богатырев В. А., Богатырев А. В., Богатырев С. В. Перераспределение запросов между вычислительными кластерами при их деградации // Изв. вузов. Приборостроение. 2014. Т. 57, № 9. С. 54–58.
13. Татарникова Т. М., Архипцев Е. Д., Кармановский Н. С. Определение размера кластера и числа реплик высоконагруженных информационных систем // Изв. вузов. Приборостроение. 2023. Т. 66, № 8. С. 646–651.
14. Богатырев В. А., Богатырев А. В. Надежность функционирования кластерных систем реального времени с фрагментацией и резервированным обслуживанием запросов // Информационные технологии. 2016. Т. 22, № 6. С. 409–416.
15. Богатырев В. А., Богатырев С. В., Богатырев А. В. Оптимизация кластера с ограниченной доступностью кластерных групп // Науч.-техн. вестн. Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2011. № 1. С. 63–67.
16. Андреев С. Д., Самуйлов К. Е., Тюриков А. М. Управление потоками в гетерогенных мобильных сетях радиодоступа с соединениями устройство–устройство // Вестник Российского университета дружбы народов. Серия: Математика, информатика, физика. 2018. Т. 26, № 4. С. 357–370.
17. Bogatyrev V. A., Bogatyrev S. V., Bogatyrev A. V. Model and interaction efficiency of computer nodes based on transfer reservation at multipath routing // Wave Electronics and its Application in Information and Telecommunication Systems. 2019. P. 8840647.
18. Lambropoulos G., Douligeris C., Mitropoulos S. Improving business performance by employing virtualization technology: a case study in the financial sector // Computers. 2021. Vol. 10, N 4.
19. Bogatyrev V., Derkach A. Evaluation of a cyber-physical computing system with migration of virtual machines during continuous computing // Computers. 2020. Vol. 9, N 2. P. 42.
20. Shi F., Lin J. Virtual machine resource allocation optimization in cloud computing based on multiobjective genetic algorithm // Computational Intelligence and Neuroscience. 2022. Vol. 2022. P. 7873131.
21. Bogatyrev V. A., Aleksankov S. M., Derkach A. N. The model of reliability of dulated real-time computers for cyber-physical systems // Physical Systems: Industry 4.0 Challenges. Studies in Systems, Decision and Control. 2020. Vol. 260. P. 11–21.
22. Волков А. О. Оценка производительности кластера облачных вычислений // T-Comm: Телекоммуникации и транспорт. 2020. Т. 14, №12. С. 72–79. DOI: 10.36724/2072-8735-2020-14-12-72-79.
23. Хомоненко А. Д., Благовещенская Е. А., Проурзин О. В., Андрук А. А. Прогноз надежности кластерной вычислительной системы с помощью полумарковской модели альтернирующих процессов и мониторинга // Наукоемкие технологии в космических исследованиях Земли. 2018. Т. 10, № 4. С. 72–82. DOI: 10.24411/2409-5419-2018-10099.

24. Tourouta E., Gorodnichev M., Polyantseva K., Moseva M. Providing fault tolerance of cluster computing systems based on fault-tolerant dynamic computation planning // Digitalization of Society, Economics and Management: A Digital Strategy Based on Post-pandemic Developments. 2022. P. 143–150. DOI: 10.1007/978-3-030-94252-6_10.
25. Шибаев Р. В. Анализ методов повышения надежности сетей при виртуализации сетевых компонентов // Автоматизация. Современные технологии. 2023. Т. 77, № 2. С. 74–81.
26. Александров С. М. Модели динамической миграции с итеративным подходом и сетевой миграции виртуальных машин // Науч.-техн. вестн. информационных технологий, механики и оптики. 2015. Т. 15, № 6. С. 1098–1104.
27. Toutov A., Vorozhtsov A., Toutova N. Analytical approach to estimating total migration time of virtual machines with various applications // Intern. J. of Embedded and Real-Time Communication Systems. 2020. Vol. 11, N 2. P. 58–75.
28. Agasizade H., Ataie E., Bastam M. Experimental Assessment of Containers Running on Top of Virtual Machines // arXiv preprint. 2024. DOI: 10.13140/RG.2.2.11457.33129.
29. Pathirathna P. P. W., Ayesha V. A. I., Imihira W. A. T., Wasala W. M. J. C., Kodagoda N., Edirisinghe E. A. T. D. Security testing as a service with docker containerization // 11th Intern. Conf. on Software, Knowledge, Information Management and Applications (SKIMA). 2017. P. 1–7. DOI: 10.1109/SKIMA.2017.8294109.
30. Zhou Yuyu et al. Comparison of virtualization and containerization techniques for high performance computing // Proc. of the 2015 ACM/IEEE Conf. on Supercomputing. 2015.
31. Mavridis I., and Karatza H. Combining containers and virtual machines to enhance isolation and extend functionality on cloud computing // Future Generation Computer Systems. 2019. Vol. 94. P. 674–696. DOI: 10.1016/j.future.2018.12.035.
32. Фунг В. К., Богатырев В. А., Кармановский Н. С., Лэ В. Х. Оценка вероятностно-временных характеристик компьютерной системы с контейнерной виртуализацией // Науч.-техн. вестн. информационных технологий, механики и оптики. 2024. Т. 24, № 2. С. 249–255. DOI: 10.17586/2226-1494-2024-24-2-249-255.
33. Wu Y., Zhang Y., Wang T., Wang H. Characterizing the Occurrence of Dockerfile Smells in Open-Source Software: An Empirical Study // IEEE Access. 2020. Vol. 8. P. 34127–34139. DOI: 10.1109/ACCESS.2020.2973750.
34. Liu Guannan, Xing Gao, Haining Wang, Kun Sun. Exploring the uncharted space of container registry typosquatting // 31st USENIX Security Symposium (USENIX Security 22). 2022. P. 35–51.

СВЕДЕНИЯ ОБ АВТОРАХ

Фунг Ван Кю

— аспирант; Университет ИТМО, факультет программной инженерии и компьютерной техники; E-mail: phungvanquy97@gmail.com

Владимир Анатольевич Богатырев

— докт. техн. наук, профессор; Университет ИТМО, факультет программной инженерии и компьютерной техники; Санкт-Петербургский государственный университет аэрокосмического приборостроения, кафедра информационной безопасности; E-mail: vladimir.bogatyrev@gmail.com

Поступила в редакцию 14.03.2024; одобрена после рецензирования 20.03.2024; принята к публикации 19.06.2024.

REFERENCES

1. Zinina T.S., Rudnik P.B. *Tsifrovaya transformatsiya: ozhidaniya i real'nost'* (Digital Transformation: Expectations and Reality), Moscow, 2022, 221 p. (in Russ.)
2. Emelyanov A.A., Korshunov I.L. *Journal of Instrument Engineering*, 2024, no. 2(67), pp. 116–121. (in Russ.)
3. Polovko A.M., Gurov S.V. *Osnovy teorii nadezhnosti* (Fundamentals of Reliability Theory), St. Petersburg, 2006, 702 p. (in Russ.)
4. Cherkosov G.N. *Nadezhnost' apparatno-programmnykh kompleksov* (Reliability of Hardware and Software Systems), St. Petersburg, 2005, 479 p. (in Russ.)
5. Khersonsky N.S., Bolshedvorskaya L.G. *Crede Experto: transport, society, education, language*, 2024, no. 1, pp. 6–23, DOI: 10.51955/2312-1327_2024_1_6. (in Russ.)
6. Astakhova T., Shamin A., Verzun N., Kolbanev M.A. *CEUR Workshop Proceedings*, 2019, no. 2(2344).
7. Gurjanov A.V., Korobeynikov A.G., Zharinov I.O., Zharinov O.O. *III International Workshop on Modeling, Information Processing and Computing*, 2021, pp. 103–108.
8. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, no. 3(23), pp. 608–617. (in Russ.)
9. Feng D., Lai L., Luo J. et al. *Sci. China Inf. Sci.*, 2021, vol. 64, pp. 1–12, DOI: 10.1007/s11432-020-2852-1.
10. Bogatyrev V.A. *Information Technologies*, 2006, no. 9, pp. 25–30. (in Russ.)
11. Bogatyrev V.A., Bogatyrev A.V. *Information Technologies*, 2015, no. 7(21), pp. 495–502. (in Russ.)
12. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. *Journal of Instrument Engineering*, 2014, no. 9(57), pp. 54–58. (in Russ.)

13. Tatarnikova T.M., Arkhiptsev E.D., Karmanovskiy N.S. *Journal of Instrument Engineering*, 2023, no. 8(66), pp. 646–651. (in Russ.)
14. Bogatyrev V.A., Bogatyrev A.V. *Information Technologies*, 2016, no. 6(22), pp. 409–416. (in Russ.)
15. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2011, no. 1, pp. 63–67. (in Russ.)
16. Andreev S.D., Samouylov K.E., Tyurlikov A.M. *Discrete and Continuous Models and Applied Computational Science*, 2018, no. 4(26), pp. 357–370. (in Russ.)
17. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. *Wave Electronics and its Application in Information and Telecommunication Systems*, 2019, pp. 8840647.
18. Lambropoulos G., Douligieris C., Mitropoulos S. *Computers*, 2021, no. 4(10).
19. Bogatyrev V., Derkach A. *Computers*, 2020, no. 2(9), pp. 42.
20. Shi F., Lin J. *Computational Intelligence and Neuroscience*, 2022, vol. 2022, pp. 7873131.
21. Bogatyrev V.A., Aleksankov S.M., Derkach A.N. *Physical Systems: Industry 4.0 Challenges. Studies in Systems, Decision and Control*, 2020, vol. 260, pp. 11–21.
22. Volkov A.O. *T-Comm: Telecommunications and transport*, 2020, no. 12(14), pp. 72–79, DOI: 10.36724/2072-8735-2020-14-12-72-79. (in Russ.)
23. Khomonenko A.D., Blagoveshchenskaya E.A., Prourzin O.V., Andruk A.A. *High Technologies in Earth Space Research. H&ES Research*, 2018, no. 4(10), pp. 72–82, DOI: 10.24411/2409-5419-2018-10099. (in Russ.)
24. Tourouta E., Gorodnichev M., Polyantseva K., Moseva M. *Digitalization of Society, Economics and Management: A Digital Strategy Based on Post-pandemic Developments*, 2022, pp. 143–150, DOI: 10.1007/978-3-030-94252-6_10.
25. Shibaev R.V. *Avtomatizatsiya. Sovremennyye Tekhnologii*, 2023, no. 2(77), pp. 74–81. (in Russ.)
26. Aleksankov S.M. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2015, no. 6(15), pp. 1098–1104. (in Russ.)
27. Toutov A., Vorozhtsov A., Toutova N. *International Journal of Embedded and Real-Time Communication Systems*, 2020, no. 2(11), pp. 58–75.
28. Aqasizade H., Ataie E., Bastam M. *arXiv preprint*, 2024, DOI: 10.13140/RG.2.2.11457.33129.
29. Pathirathna P.P.W., Ayesha V.A.I., Imihira W.A.T., Wasala W.M.J.C., Kodagoda N., Edirisinghe E.A.T.D. *11th Intern. Conf. on Software, Knowledge, Information Management and Applications (SKIMA)*, 2017, pp. 1–7, DOI: 10.1109/SKIMA.2017.8294109.
30. Zhou Yuyu et al. *Proc. of the 2015 ACM/IEEE Conf. on Supercomputing*, 2015.
31. Mavridis I., and Karatza H. *Future Generation Computer Systems*, 2019, vol. 94, pp. 674–696, DOI: 10.1016/j.future.2018.12.035.
32. Phung Van Quy, Bogatyrev V.A., Karmanovskiy N.S., Le V. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2024, no. 2(24), pp. 249–255, DOI: 10.17586/2226-1494-2024-24-2-249-255. (in Russ.)
33. Wu Y., Zhang Y., Wang T., Wang H. *IEEE Access*, 2020, vol. 8, pp. 34127–34139. DOI: 10.1109/ACCESS.2020.2973750.
34. Liu Guannan, Xing Gao, Haining Wang, Kun Sun. *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 35–51.

DATA ON AUTHORS

- | | |
|------------------------------|---|
| Phung Van Quy | — Post-Graduate Student; ITMO University, Faculty of Software Engineering and Computer Science; E-mail: phungvanquy97@gmail.com |
| Vladimir A. Bogatyrev | — Dr. Sci, Professor; ITMO University, Faculty of Software Engineering and Computer Science; St. Petersburg State University of Aerospace Instrumentation, Department of Information Security; E-mail: vladimir.bogatyrev@gmail.com |

Received 14.03.2024; approved after reviewing 20.03.2024; accepted for publication 19.06.2024.