

Е.Қ. Майлыбаев¹, Д.С.Жамангарин², Н.П. Сапарходжаев³

¹Международный транспортно-гуманитарный университет, Алматы, Казахстан

²Казахский университет технологии и бизнеса, Астана, Казахстан

³East Kazakhstan Technical University, Усть-Каменогорск, Казахстан

E-mail: ersind@mail.ru

РЕАЛИЗАЦИЯ ОТКАЗОУСТОЙЧИВОЙ СЕРВЕРНОЙ ИНФРАСТРУКТУРЫ

Аннотация. В данной статье были рассмотрены методы обеспечения отказоустойчивости серверов и серверных систем. Главной целью являлось уменьшение времени простоя и последующих от них убытков, а также обеспечение постоянной работы и доступности задействованных сервисов. Были исследованы физические и программные методы отказоустойчивости, и в результате исследования были определены наиболее эффективные методы для обеспечения отказоустойчивости. Исследованы основные виды кластеров. Рассмотрены преимущества кластеризации, перед традиционными методами резервного копирования. Проведен анализ актуальности обеспечения отказоустойчивости серверных систем. Проанализированы разновидности кластеров, таких как, High Availability (HA) - кластеры, NLB - кластеры, High Performance Computing (HPC) - кластеры и системы распределенных вычислений. Разработана UML - диаграмма, описывающая работу созданного кластера, обеспечивающую отказоустойчивость серверных систем. В виртуальной среде, с помощью инструмента VMware ESXi создана отказоустойчивая система, состоящая из трех серверов на основе Network Load Balancing (NLB) кластера средствами операционной системы семейства Windows Server. Процесс настройки серверов выполнялся с помощью программной оболочки Power Shell.

Результаты исследования могут быть использованы для создания отказоустойчивых серверных систем и улучшения их производительности и доступности.

Ключевые слова. Отказоустойчивость, проектирование, кластеризация, баланс, сервер.

Введение.

Развитие интернет-сетей приводит к большому количеству циркулирующей информации. В наши дни актуально проектирование и построения надежной, защищенной и отказоустойчивой сети предприятий. Стабильная и непрерывная работа серверов, которые поддерживают функционирование сетей, достигается при помощи методов обеспечения отказоустойчивости. Отказоустойчивость работы серверной инфраструктуры возможно достичь как программными методами, так и аппаратными [1-2].

Материалы и методы.

С развитием интернета и серверов появлялась необходимость постоянного хранения больших данных и поддержания круглосуточной работы сервисов. Главной проблемой в обеспечении постоянной работы сервисов являлось возникновение различных сбоев. Если обратить внимание на рисунок 1, в котором показано количество отчетов о публично сообщаемых сбоях, за 2023 год, отслеживаемых Uptime, то можно утверждать, что на данный момент все также имеется необходимость в обеспечении отказоустойчивости.

2016	2017	2018	2019	2020	2021	2022	Total
27	57	71	165	118	109	111	656

UPTIME INSTITUTE DATA CENTER RESILIENCY SURVEY 2023




Рисунок 1 - Количество сбоев с 2016 по 2022 год

В обеспечении физической устойчивости серверов используют метод дублирования критически важных компонентов сервера. Этот метод в своем роде уже стал обязательным и традиционным. Большинство компаний могут позволить себе использовать резервированные сервера. Это полные копии основных серверов, с теми же конфигурациями и исполняемыми функциями, работающие в автономном режиме. При возникновении проблем в работе или полная поломка основного сервера, резервный сервер, выходя из автономного режима, незамедлительно заменяет вышедший из строя сервер. Такой метод дублирования является дорогостоящим. Если обратиться к статистике гарантийных случаев, то ниже мы можем составить диаграмму, которая показано на рисунке 2, по которой можно понять, что компонентами, которые наиболее часто подвергаются поломке, являются жесткие диски и блоки питания. Исходя из этого, при выборе того или иного оборудования, необходимо учитывать возможность горячей замены, вышедшего из строя элемента. Горячая замена компонента подразумевает быструю смену без остановки работы системы. Также при построении физической отказоустойчивости можно применить концепцию фактора N+1, где N — это количество активных компонентов, которые необходимы для обеспечения нормальной работы системы, а +1 — это резервный компонент, который используется в случае отказа активного компонента. Суть концепции заключается в том, что для обеспечения отказоустойчивости системы необходимо иметь достаточное количество резервных компонентов, чтобы можно было продолжить работу системы в случае отказа одного или нескольких активных компонентов [3-4].

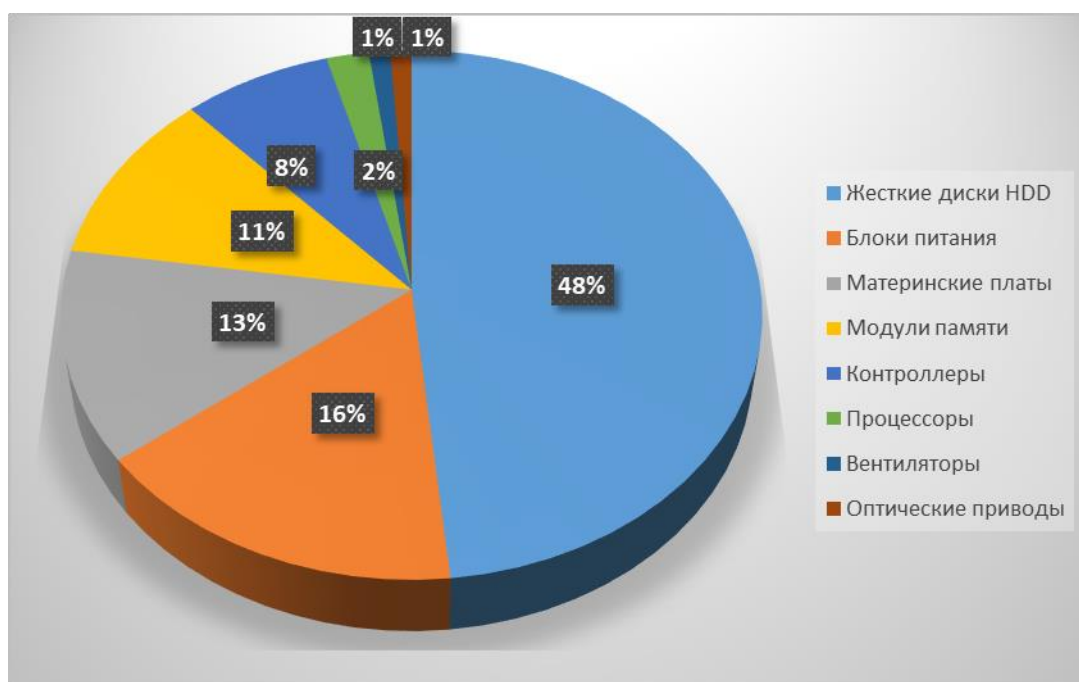


Рисунок 2 - Диаграмма частоты поломки серверных компонентов

В то время, когда для отказоустойчивости использовались резервные серверы, которые простаивали в ожидании поломки основного, целесообразно оптимизировать простой серверов путем их группировки в кластерные системы. Эти системы можно считать первым случаем реализации кластеров. Таким образом возникло понятие кластерные системы. Кластеризация серверов помогает достичь большей надежности и масштабируемости системы. Основные преимущества кластерных систем: высокая доступность (достигаемая путем взаимозаменяемости серверов), увеличенная производительность (получаемая за счет распределения нагрузки), упрощенное администрирование (возможность управления с одного сервера), повышенная надежность данных. В настоящее время имеется классификация кластерных систем. Основные виды кластеров:

- отказоустойчивые кластеры;
- кластеры непрерывной доступности;
- вычислительные кластеры.

Исходя из принципа работы кластерных систем, в зависимости от их построения можно выделить несколько основных разновидностей кластеров: High Availability (HA)-кластеры, NLB-кластеры, High Performance Computing (HPC)-кластеры и системы распределенных вычислений.

HA кластеры – это кластеры, которые обеспечивают высокую доступность. За счет объединения нескольких серверов в одну общую сеть обеспечивается отказоустойчивость и непрерывная работа системы. Принцип работы HA кластеров основан на репликации данных и функций между серверами.

NLB кластеры – это кластеры, которые используют балансировщики нагрузки для распределения трафика между узлами кластера. Эти кластеры используются для распределения трафика с высокой пропускной способностью, что делает их особенно подходящими для обработки больших объемов запросов и данных.

HPC кластеры — это кластеры, специализированные для высокопроизводительных вычислений. Это мощные компьютерные системы, которые состоят из большого количества вычислительных узлов, объединенных с помощью сети высокой скорости. HPC-кластеры используются для решения сложных задач, требующих больших вычислительных ресурсов, таких как научные расчеты, моделирование, анализ данных и другие.

В целом, выбор типа кластера зависит от конкретных потребностей организации или проекта. HA-кластеры подходят для обеспечения высокой доступности сервисов и приложений, NLB-кластеры - для балансировки нагрузки веб-серверов и других приложений, а HPC-кластеры - для обработки научных и технических вычислений.

В статье будет рассмотрена реализация отказоустойчивого кластера NLB с балансировкой нагрузки средствами Windows Server. На рисунке 3 показана схема реализуемого кластера.

Главной задачей является создание отказоустойчивой системы, которая будет продолжать бесперебойно свою работу при выходе из строя узлов кластера. Наравне с отказоустойчивостью также стоит задача достижения хорошей производительности за счет использования балансировки нагрузки.

Выбор именно NBL кластера можно обосновать несколькими возможностями, которые предоставляет данный вид кластера. Во-первых, благодаря созданию кластера достигается высокая отказоустойчивость. В случае выхода из строя одного из узлов кластера клиент, работающий в данный момент на сервере, будет моментально переключен на другой сервер, где он сможет продолжить свою работу. Во-вторых, данный кластер будет производить распределение нагрузки, что непосредственно скажется на производительности всего сервиса [5-6].

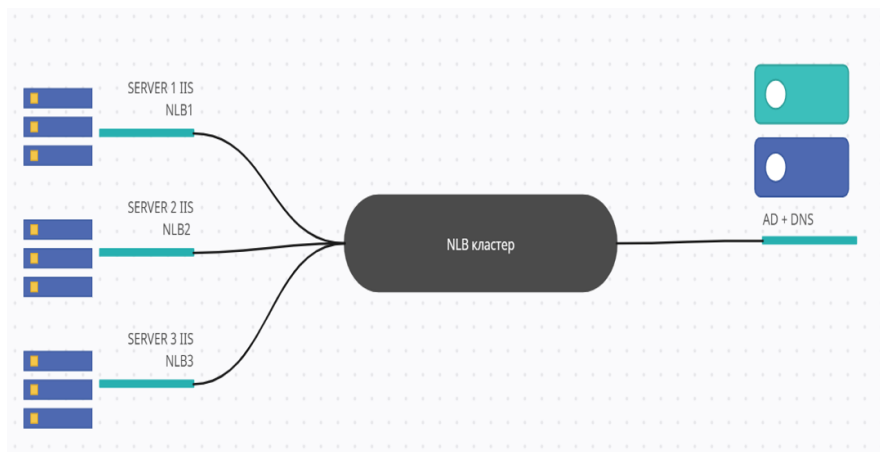


Рисунок 3 - Схема проектируемого кластера

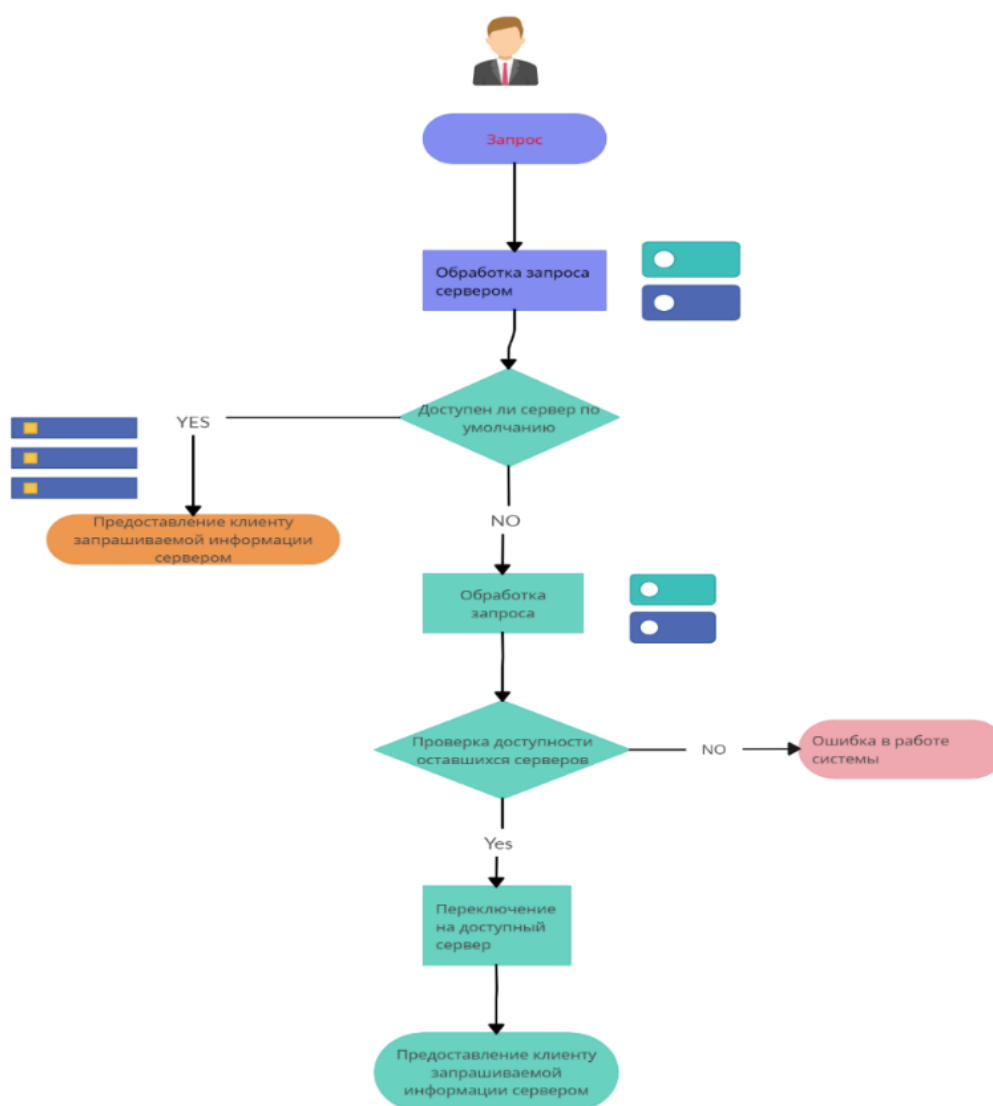


Рисунок 4 - UML-диаграмма алгоритма работы кластера

Данный кластер будет включать в себя 3 узла на каждом из которых будут установлены веб сервера и роли NLB. В целом средства Windows Server дают возможность создания кластеров, включающих объединение до 32 узлов. Помимо этого,

для реализации кластера будет необходим еще один сервер контролер домена с Active Directory и службой DNS. Кластер будет работать в режиме Multicast. Алгоритм работы кластера в виде UML диаграммы показана на рисунке 4. Взаимодействовать между собой сервера будут при помощи сообщения пульс, который будет производиться всеми серверами каждые пять секунд. При помощи данного сообщения будет производиться проверка работоспособности каждого из узлов кластера. В случае поломки или выхода из строя одного из узлов кластера переключение и перераспределение нагрузки будет производиться в течении десяти секунд.

Результаты и обсуждение.

Проект кластера была реализована в виртуальной среде VMware ESXi. Для этого необходимо установить данное программное обеспечение на одну из доступных машин. После установки и корректной настройки можно будет перейти в веб-приложение откуда можно будет производить управление данным сервером. После всего этого в данной программе необходимо будет создать и сконфигурировать четыре виртуальных сервера с операционной системой семейства Windows Server. Успешно созданные виртуальные сервера можно увидеть на рисунке 5.

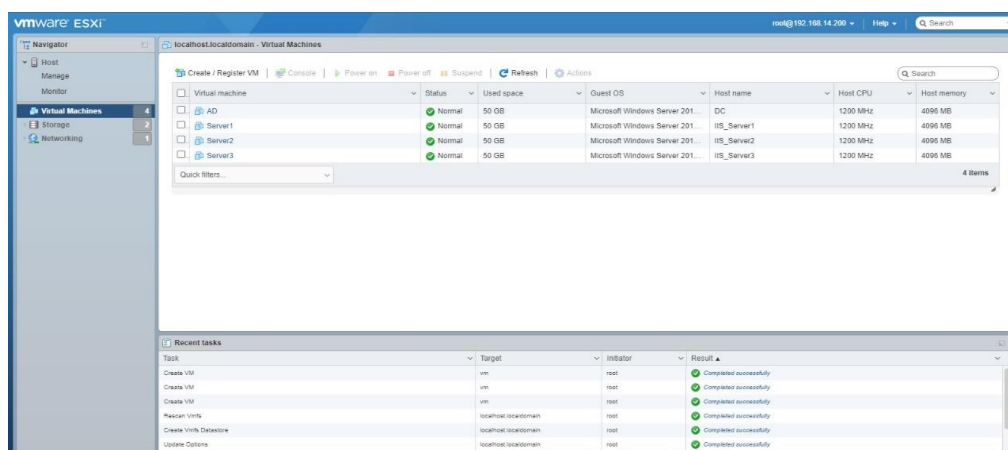


Рисунок 5 - Виртуальные серверы в VMware ESXi

В начале создания кластера необходим будет сервер с установленными на нем службами Active Directory и DNS. Все имеющиеся сервера должны быть в одном доменном пространстве и иметь свои статические IP-адреса. Реализация NLB кластера будет на server1. Данная машина будет сервером по умолчанию. Все подключения сначала будут поступать на этот сервер после чего при необходимости будет происходить переадресация на другие узлы кластера. Для построения кластера будут использованы команды в Windows PowerShell, как показано на рисунке 6. Для начала необходимо будет проверить можно ли установить на сервер NLB и Web-сервер, используя команду PowerShell.

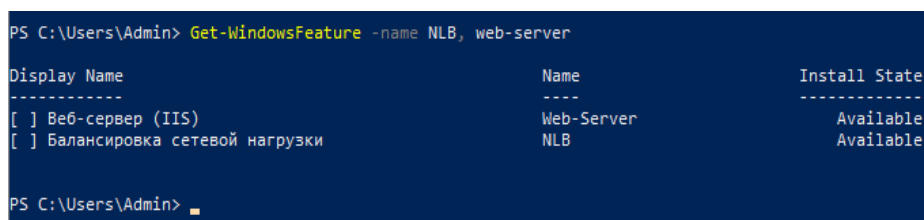


Рисунок 6 - Команда проверки наличия модуля

На рисунке 7 можно увидеть, что службы можно установить на сервер и в данный момент они не установлены. Установка данных компонентов будет производиться с машины server2 сразу на все три узла кластера при помощи Invoke-Command. Это позволяет сэкономить время, не прописывая все команды отдельно на каждом сервере, а при помощи одной команды сделать все удаленно на одном из узлов кластера.

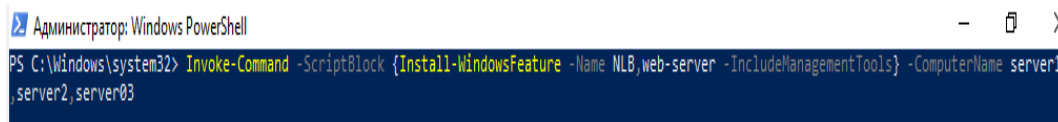


Рисунок 7 - Команда установки модулей

В данной команде:

- 1) Invoke-Command - команда PowerShell для выполнения команды на удаленном компьютере.
- 2) -ComputerName - параметр, указывающий список имен компьютеров, на которых необходимо выполнить команду.
- 3) -ScriptBlock - параметр, который содержит команду или набор команд, которые нужно выполнить на удаленном компьютере.
- 4) Install-WindowsFeature - команда PowerShell для установки роли Windows.
- 5) -Name - параметр, указывающий имя роли или ролей, которые нужно установить.
- 6) NLB, Web-Server, Web-ASP, Web-Asp-Net, Web-Asp-Net45 - имена ролей, которые нужно установить.
- 7) -IncludeManagementTools - параметр, указывающий, что нужно установить дополнительные инструменты управления для этих ролей.

После успешной установки всех необходимых компонентов можно будет приступить непосредственно к созданию NLB кластера, как показано на рисунке 8. Для этого существует специальная команда New-NlbCluster, в которой необходимо будет указать следующие параметры: название узла, на котором будет подниматься кластер, режим работы кластера, название кластера, интерфейс и адрес кластера.

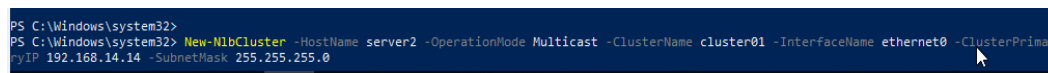


Рисунок 8 - Команда создания кластера

В данной команде:

- HostName – это имя хоста, который будет использоваться для создания кластера.
- OperationMode – это режим работы NLB. В данном случае, значение параметра равно "Multicast", что означает, что сетевой трафик будет отправлен на множество узлов в кластере, используя один и тот же IP-адрес.
- ClusterName – это имя кластера NLB.
- InterfaceName – это имя сетевого интерфейса, который будет использоваться для обработки трафика в кластере NLB.
- ClusterPrimaryIP – это IP-адрес, который будет использоваться в качестве первичного IP-адреса для кластера NLB.
- SubnetMask – это маска подсети, которая будет использоваться в кластере NLB для разрешения трафика.

После того как будет создан кластер в его составе будет находится лишь один сервер, на котором его создали. Поэтому необходимо вручную добавлять новые узлы в кластер при помощи команды `Add-NlbClusterNode`, как показано на рисунке 9.

```
PS C:\Windows\system32>  
PS C:\Windows\system32>  
PS C:\Windows\system32>  
PS C:\Windows\system32> Add-NlbClusterNode -InputObject $NLB -NewNodeName server1 -NewNodeInterface Ethernet
```

Рисунок 9 - Команда добавления узла

В данной команде:

-`InputObject` – указывает объект кластера NLB, к которому нужно добавить новый узел. В данном случае используется переменная `$NLB`, которая должна содержать объект кластера NLB.

-`NewNodeName` – указывает имя нового узла, который будет добавлен в кластер NLB. В данном случае имя узла указано как `server1`.

-`NewNodeInterface` – указывает имя сетевого интерфейса нового узла. В данном случае используется имя `Ethernet`.

Команда `Add-NlbClusterNode` позволяет автоматически настроить сетевой интерфейс нового узла и добавить его в кластер NLB. В результате новый узел будет готов к обработке сетевого трафика вместе с другими узлами кластера.

Для того чтобы удостовериться в том, что узлы были добавлены в кластер, можно будет воспользоваться командой `Get-NlbClusterNode`. Данная команда используется для получения данных о состоянии узлов кластера, их интерфейсах и ID-номерах. Из рисунка 10 можно будет увидеть, что сервером по умолчанию является узел `server2`. Это означает, что все соединения первоначально будут приниматься этим сервером после чего при необходимости они будут переправляться на другие узлы кластера.

```
PS C:\Windows\system32> Get-NlbClusterNode -HostName server2  
  
Name      State      Interface HostID  
----      -  
Server1   Converged  Ethernet  2  
Server03  Converged  Ethernet  3  
Server2   Converged(default) Ethernet  1
```

Рисунок 10 - Информация о кластере

Для удобства пользования веб-сервером следует создать DNS-запись, как показано на рисунке 11. Это необходимо для того, чтобы к нашему кластеру была возможность обращаться по заданному DNS-имени, а не по IP-адресу. С этим нам поможет команда `Add-DnsServerResourceRecordA`.

```
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.  
PS C:\Users\Администратор> Add-DnsServerResourceRecordA -ZoneName Malikov.kz -Name cluster01 -IPv4Address 192.168.14.14  
PS C:\Users\Администратор>
```

Рисунок 11 - Создание DNS-записи

После того как успешно будут пройдены все шаги установки и настройки NLB кластера, можно будет проверить отказоустойчивость системы. Для этого перейдем по созданному выше адресу cluster01 к веб-серверу. На данной странице будет указано, что в данный момент подключение происходит на узле server2. Выбор подключения пал именно на этот сервер потому, что он является сервером по умолчанию. Для проверки отказоустойчивой работы кластера необходимо будет отключить интернет-соединения узла server2. Таким образом будет проведена имитация выхода из строя одного из узлов кластера.

После того как будет произведено отключение узла server2, необходимо будет обновить страницу или же снова зайти на веб-сервер. После можно увидеть, как клиент беспрепятственно продолжил работу на сервере, но в этом случае подключение происходит уже через другой узел кластера server1. Если же данный алгоритм действий повернуть и с узлом server1, то работа сервиса также продолжится через последний узел кластера server3.

Таким образом в результате проделанной работы была достигнута стабильная работа веб-сервера на отказоустойчивом NLB кластере, что повышает удобство использования сервиса для пользователей и обеспечивает высокую отказоустойчивость и доступность.

Выводы.

В статье была рассмотрена формирования отказоустойчивой серверной инфраструктуры. Основываясь на проделанную работу, можно сформулировать следующие результаты:

- 1) Была изучена проблематика и необходимость формирования отказоустойчивой инфраструктуры.
- 2) Были рассмотрены основные методы позволяющие достичь высокой отказоустойчивости.
- 3) Выявлены основные достоинства отдельных методов.
- 4) Сформирована отказоустойчивая система на основе NLB кластера.

ЛИТЕРАТУРА

- [1] Morokina G.S., Umbetov U. U., Mailybayev Y.K. Automation design systems for mechanical engineering and device node design, *Journal of Physics: Conference Series*, 2020, 1515(3), 032061
- [2] Morokina G.S., Umbetov U. U., Mailybayev Y.K. Computer-Aided Design Systems of Decentralization on Basis of Trace Mode in Industry, *Proceedings - 2019 International Russian Automation Conference, RusAutoCon 2019*, 2019, 8867817
- [3] Mailybayev, Y., Muratbekova, G., Altayeva, Z., Zhatkanbayev, O. Development of models and improvement of methods for formalization of design problems and automating technical and operational works of railway stations, *Eastern-European Journal of Enterprise Technologies*, 2022, 4(3-118), 8–9p.
- [4] Mailybayev, Y., Umbetov, U., Lakhno, V., Ammanova, M., Sauanova K. Development of mathematical and information support for solving prediction tasks of a railway station development, *Journal of Theoretical and Applied Information Technology*, 2021, 99(3), 583–593p.
- [5] Shayakhmetova, A., Litvinenko, N., Mamyrbayev, O., Wójcik, W., Zhamangarin, D. Cycles in bayesian networks, *International Journal of Electronics and Telecommunications*, 2021, 67(2), 181–186p

[6] Saparkhojayev, N., Shakhov, E., Mailybayev, Y. Mobile Attendance Checking System on Android Platform for Kazakhstani University, Journal of Physics: Conference Series, 2016, 710(1), 12013

Ерсайын Майлыбаев, PhD, Ғылым және цифровизация проректоры, Халықаралық көліктік-гуманитарлық университет, Алматы, Қазақстан, ersind@mail.ru

Дусмат Жамангарин, PhD, Оқу-әдістемелік жөніндегі проректор, Қазақ технология және бизнес университеті, Астана, Қазақстан, dus_man89@mail.ru

Нурбек Сапарходжаев, PhD, Басқарма мүшесі - бірінші проректор, East Kazakhstan Technical University, Өскемен, Қазақстан, nursp81@gmail.com

АҚАУЛАРҒА ТӨЗІМДІ СЕРВЕРЛІК ИНФРАҚҰРЫЛЫМДЫ ІСКЕ АСЫРУ

Аңдатпа. Бұл мақалада серверлік жүйелердің ақауларға төзімділігін қамтамасыз ету әдістері қарастырылды. Негізгі мақсат тоқтап қалуды және олардан кейінгі мүліктік зиянды қысқарту, сондай-ақ қызметтердің үздіксіз жұмысы мен қолжетімділігін қамтамасыз ету болды. Физикалық және бағдарламалық әдістер зерттелді және зерттеу нәтижесінде ақауларға төзімділікті қамтамасыз етудің ең тиімді әдістері анықталды. Мақалада Windows Server көмегімен NLB кластеріне негізделген үш серверден тұратын ақауларға төзімді жүйе құрылды. Зерттеу нәтижелерін ақауларға төзімді серверлік жүйелерді құру және олардың өнімділігі мен қолжетімділігін жақсарту үшін пайдалануға болады.

Түйінді сөздер. Ақауларға төзімділік, жобалау, кластерлеу, баланс, сервер.

Yersayin Mailybayev, PhD, Vice-Rector for Science and Digitalization, International University of Transport and Humanities, Almaty, Kazakhstan, ersind@mail.ru

Dusmat Dzhamangarin, PhD, Vice-Rector for Academic and Methodological Work, Kazakh University of Technology and Business, Astana, Kazakhstan, dus_man89@mail.ru

Nurbek Saparkhojayev, PhD, Board Member - First Vice-Rector, East Kazakhstan Technical University, Ust-Kamenogorsk, Kazakhstan, nursp81@gmail.com

IMPLEMENTATION OF FAULT-TOLISTANT SERVER INFRASTRUCTURE

Abstract. In this article, methods for ensuring the fault tolerance of server systems were considered. The main goal was to reduce downtime and subsequent losses from them, as well as ensure the continued operation and availability of services. Physical and software methods were investigated, and as a result of the study, the most effective methods for ensuring fault tolerance were identified. In the article, a fault-tolerant system was created, consisting of three servers based on an NLB cluster using Windows Server. The results of the study can be used to create fault-tolerant server systems and improve their performance and availability.

Keywords. Fault tolerance, design, clustering, balance, server.
