

Алгоритм миграции ESXi-кластеров между разными vCenter серверами с возможностью возврата к исходной конфигурации

А. А. Петров^а, магистрант, orcid.org/0000-0002-7465-3317

И. В. Никифоров^а, канд. техн. наук, доцент, orcid.org/0000-0003-0198-1886, nikiforov_iv@spbstu.ru

С. М. Устинов^а, доктор техн. наук, профессор, orcid.org/0000-0003-4088-4798

^аСанкт-Петербургский политехнический университет Петра Великого, Политехническая ул., 29, Санкт-Петербург, 195251, РФ

Введение: системы управления гипервизорами могут устаревать и (или) выходить из строя, а также нуждаться в балансировке нагрузки, что влечет за собой проблемы, связанные с несовместимостью программного обеспечения с разными версиями систем управления или с уровнем интеграции приложений с ними. Эффективным решением указанных проблем является автоматизированная миграция конфигураций и ресурсов между экземплярами систем управления гипервизорами. **Цель:** разработать алгоритм автоматической миграции конфигурации и ресурсов ESXi-кластеров между разными vCenter серверами с возможностью возврата к исходному состоянию в случае возникновения ошибки. **Результаты:** предложен алгоритм автоматической миграции. На его основе разработан программный модуль для промышленной системы хранения данных, позволяющий проводить миграцию кластера ESXi-хостов, на котором развернута система хранения данных, с одного vCenter сервера на другой. Разработанный модуль обрабатывает ошибки и в случае их возникновения осуществляет действия для восстановления исходного состояния системы на первоначальном vCenter сервере. Приведены экспериментальные оценки, полученные путем сравнения ручной процедуры миграции с работой реализованного модуля. Выигрыш по времени сократился с 55 мин до приблизительно трех минут, т. е. в 15–18 раз. Также сократилось количество информации, которую пользователю необходимо знать и учитывать в процессе миграции на новый vCenter сервер. **Практическая значимость:** результаты работы использованы в промышленной системе хранения данных. Реализованный модуль позволил упростить и значительно ускорить процесс миграции системы хранения данных на новый vCenter сервер.

Ключевые слова — гипервизоры, vCenter сервер, VMware ESXi, миграция, автоматизация, распределенный виртуальный коммутатор.

Для цитирования: Петров А. А., Никифоров И. В., Устинов С. М. Алгоритм миграции ESXi-кластеров между разными vCenter серверами с возможностью возврата к исходной конфигурации. *Информационно-управляющие системы*, 2022, № 2, с. 20–31. doi:10.31799/1684-8853-2022-2-20-31

For citation: Petrov A. A., Nikiforov I. V., Ustinov S. M. Algorithm of ESXi cluster migration between different vCenter servers with the ability to rollback. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 2, pp. 20–31 (In Russian). doi:10.31799/1684-8853-2022-2-20-31

Введение

В настоящее время активно используются облачные вычисления [1, 2], в реализации которых применяются системы виртуализации. Системы виртуализации позволяют абстрагировать оборудование (аппаратное обеспечение) и предоставлять доступ пользователям и приложениям к общему набору ресурсов: памяти, вычислительным мощностям, хранилищу, — но изолированных друг от друга в виртуальных машинах (ВМ) [3–6]. Для виртуализации ресурсов требуются программные или аппаратные гипервизоры; одним из примеров широко используемых в промышленности гипервизоров является VMware ESXi [1, 4, 7]. Для управления разными ESXi-хостами и ВМ используется управляющая программа vCenter [1, 8], в которой хосты объединяются в кластер, и для них настраивается виртуальная сеть. Также vCenter позволяет настраивать пра-

вила для миграции ВМ между ESXi-хостами кластера. Примерами программных систем и организаций, использующих ESXi-кластеры под управлением vCenter, являются PowerStore X, ЭрисВМ [9], Dell Technologies, Санкт-Петербургский политехнический университет Петра Великого.

Серверы vCenter могут выходить из строя по тем или иным причинам: отсутствие интернет-соединения; отключение от электросети хоста, на котором он развернут, или отключение ВМ, где vCenter сервер установлен. В свою очередь это может вызывать ошибки у программного обеспечения, развернутого в этом vCenter сервере. Поэтому актуальной является задача миграции ресурсов и восстановления конфигурации кластера ESXi-хостов в новом vCenter сервере, что позволит:

- обеспечить отказоустойчивость;
- распределить нагрузку;
- решить проблемы при обновлениях, например, один из кластеров требует более новую вер-

сию vCenter, а другой(ие) кластер не может корректно работать в новом vCenter.

При этом процесс миграции или восстановления может быть сложным и трудоемким. В ходе миграции пользователи могут допускать ошибки, что делает актуальной задачу автоматизации миграции ESXi-кластеров между разными vCenter серверами.

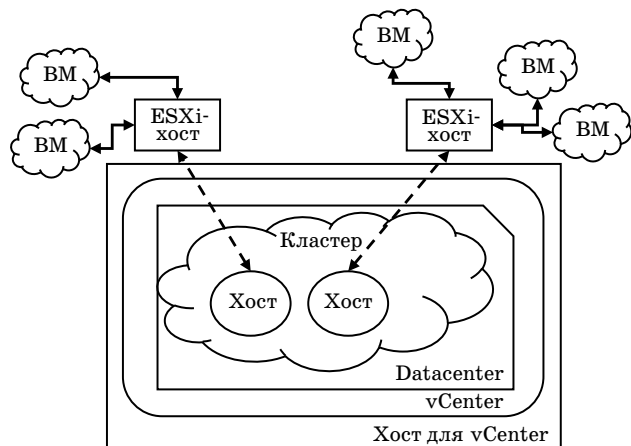
Цель и задачи работы

Рассмотрим схему компонентов кластера vCenter (рис. 1). Она состоит из хоста для vCenter, vCenter сервера, логического объекта Datacenter, ESXi-хостов и набора VM.

Сервер vCenter является программным обеспечением, установленным на выделенном vCenter-хосте под управлением операционной системы Windows или Linux. Он используется для управления VM и ESXi-хостами [1, 8]. Под ESXi-хостами в работе понимается аппаратный промышленный гипервизор компании VMware [1].

Внутри vCenter сервера создается логический объект, называемый Datacenter, объединяющий VM, ESXi-хосты, хранилища данных, кластеры ESXi-хостов и распределенный виртуальный коммутатор (Distributed Virtual Switch, DVS). При этом DVS позволяет настроить подключение для всех ESXi-хостов, к которым он относится.

В области Datacenter создан кластер, который является логическим объектом, объединяющим ESXi-хосты и VM, развернутые на них [10]. Кластер может иметь один или более ESXi-хостов, которые развернуты отдельно и независимо от самого vCenter сервера, т. е. находятся вне области Datacenter.



■ Рис. 1. Логическая схема связи ESXi-хостов, vCenter сервера и виртуальных машин

■ Fig. 1. Logical diagram of communication between ESXIs, vCenter and VMs

Именно за счет представленной логической схемы и связей vCenter сервер может осуществить миграцию VM между разными ESXi-хостами.

Целью работы является снижение трудоемкости и ускорение процесса миграции ESXi-кластеров между разными vCenter серверами. При этом на мигрируемых кластерах развернуты VM с конечными приложениями пользователя. Отличительной особенностью работы является снижение времени нахождения системы в некорректном состоянии за счет предложенного алгоритма возвращения системы к исходной конфигурации.

Снижение трудоемкости и ускорение достигаются за счет алгоритма перемещения ресурсов, реализованного в модуле промышленной системы хранения данных. К перемещаемым ресурсам можно отнести настройки DVS, ESXi-хостов, VM и кластера.

Для реализации поставленной цели необходимо выполнить следующие задачи.

1. Исследовать необходимость миграции ESXi-кластера между экземплярами vCenter серверов.
2. Провести исследование и сравнительный анализ существующих подходов к миграции ESXi-кластеров.
3. Предложить алгоритм миграции ESXi-кластеров под управлением разных vCenter серверов.
4. Реализовать предложенный алгоритм в программном средстве, которое позволяет автоматизировать миграцию. Отличительной особенностью реализации является возможность отката миграции в случае ошибки.
5. Продемонстрировать снижение трудоемкости процесса миграции.

Сравнительный анализ существующих методов миграции

Для решения поставленных задач рассмотрим необходимость восстановления конфигурации в процессе миграции, а также существующие алгоритмы миграции.

Миграция используется для переноса конфигурации и подключения существующего сервера, размещенного на хостах ESXi-кластера, к новому vCenter серверу и удаления необходимой информации со старого vCenter сервера.

Миграция также может использоваться для:

- перераспределения нагрузки [11];
- обновления версии vCenter сервера [12, 13];
- рефакторинга конфигурации IT-инфраструктуры.

Дополнительными требованиями для миграции и восстановления конфигурации являются:

- обеспечение доступности данных [14];
- невозможность отключения системных VM.

На сегодня можно выделить несколько методов миграции кластеров ВМ. Например, в статье [10] упоминаются четыре «типичных» метода: параллельной миграции с разной гранулярностью (Concurrent Migration with Various Granularity), взаимной миграции (Mutual Migration), гомогенной миграции нескольких виртуальных кластеров (Homogeneous Multi-VC Migration), гетерогенной миграции нескольких виртуальных кластеров (Heterogeneous Multi-VC Migration). Еще один метод совместной миграции виртуальных машин с поддержкой кластера для облачных медиасервисов (cluster-aware VM collaborative migration scheme for media cloud) описан в работе [15]. Также существуют методы миграции ВМ между гипервизорами и центрами обработки данных, которые предлагают оптимизацию миграций как по времени, так и по объему трафика [16]. Но если создается кластер ESXi-хостов, то за миграцию ВМ и их безотказную работу отвечает vCenter сервер [1], который в таком случае является единой точкой отказа. Поэтому рассмотрим существующие способы и подходы по миграции ESXi-кластеров между разными экземплярами vCenter серверов.

Алгоритм миграции ESXi-кластеров отражен в работе компании VMware «Перемещение vSAN-кластера с одного vCenter сервера на другой» (<https://kb.vmware.com/s/article/2151610>). Также отдельно описывается миграция ESX/ESXi-хостов, которые используют DVS, между экземплярами vCenter серверов; предлагается для упрощения использовать экспорт и импорт конфигурации DVS (<https://kb.vmware.com/s/article/1029498>), которая в свою очередь описана в материалах компании VMware (<https://kb.vmware.com/s/article/2034602>). Также существует работа компании Nutanix, которая описывает процедуру ручной миграции ESXi-кластера (База знаний о контроллере Nutanix. https://portal.nutanix.com/page/documents/details?targetId=vSphere-Admin6-AOS-v6_0:vsp-cluster-migrate-vcenter-vsphere-t.html%7D).

Все вышеперечисленные решения описывают процесс ручной миграции, который требует значительных затрат времени (суммарно около одного часа) и имеет высокий шанс ошибок. А если алгоритм имеет частичную автоматизацию, то необходима работоспособность существующего vCenter сервера, что не позволяет использовать эти решения для автоматического восстановления конфигурации.

Автоматизированный процесс миграции реализован в проекте Migrate-vCenter-with-vDS (<https://github.com/Vidanez/Migrate-vCenter-with-vDS>). В нем не содержится функциональность отката к исходной конфигурации на старом

■ **Таблица 1.** Сравнение способов миграции ESXi-кластеров между экземплярами vCenter серверов

■ **Table 1.** Comparison between different methods of migration ESXi clusters between different vCenter instances

Алгоритм	Критерий		
	Автоматизированность	Сохранение доступа по сети	Возврат к исходной конфигурации
«Перемещение vSAN-кластера с одного vCenter сервера на другой» совместно с миграцией ESXi-хостов и экспортом/импортом конфигурации DVS	Частично (за счет экспорта/импорта некоторых конфигураций)	Да	Возможен в ручном режиме
Nutanix	Нет	Да	То же
Migrate-vCenter-with-vDS	Да	Да	Нет

vCenter сервере в случае ошибки в процессе миграции [17]. Также нет возможности повторить попытку на частично восстановленной конфигурации кластера на vCenter сервере.

На основе сравнения характеристик (табл. 1) можно сделать вывод, что ни один из существующих подходов не автоматизирует миграцию с возможностью возврата к исходной конфигурации в случае ошибок во время миграции. Поэтому в настоящей работе предлагается алгоритм автоматизации, который удовлетворяет всем рассмотренным критериям и обладает следующими особенностями:

— сохраняется доступность данных во время миграции;

— имеется возможность вернуть конфигурацию на исходный vCenter в случае ошибки в процессе миграции, что позволит снизить потенциальное время нахождения системы в некорректном состоянии [17];

— есть возможность перезапустить процесс восстановления конфигурации для операции восстановления конфигурации на новом vCenter сервере.

Алгоритм миграции

Алгоритмы миграции и восстановления конфигурации имеют различия, так как первый позволяет вернуться к исходному состоянию, а второй, не имея такой возможности, обладает функцией перезапуска после исправления ошибки. Мы предлагаем алгоритм миграции, который в упрощенном виде приведен на рис. 2.

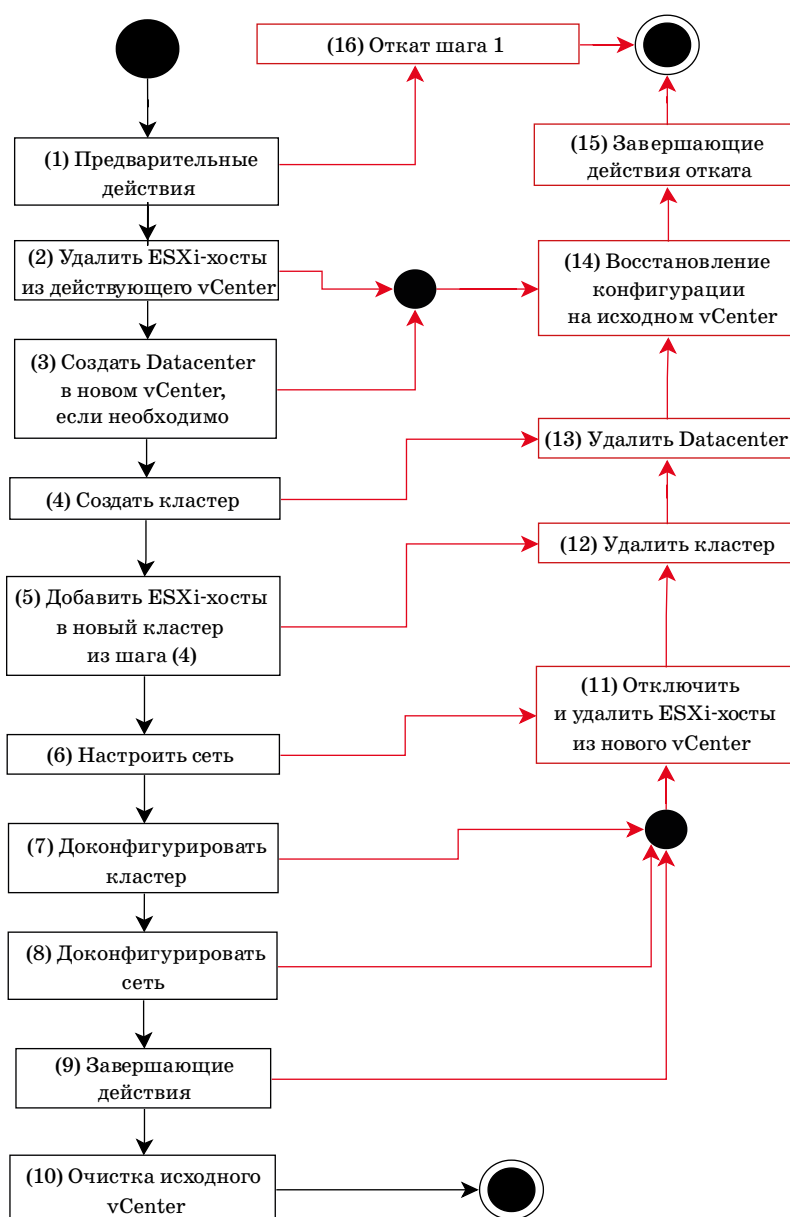
Шаг 1. Перед проведением миграции следует выполнить предварительные действия:

— получить необходимые данные и сформировать начальную конфигурацию для миграции;

— проверить состояние нового vCenter сервера и ESXi-хостов, участвующих в миграции. ESXi-хосты должны быть доступны, а имя пользователя и пароль правильными. Осуществить иные предварительные проверки, например на отсутствие в новом vCenter сервере объектов с именами, совпадающими с именами объектов, которые будут созданы в процессе миграции;

— выполнить специфичные для каждого кластера (сервера, развернутого на кластере) действия, например удалить Storage Provider и выставить параметры на сервере.

Шаг 2. Подготовить хосты к миграции, т. е. удалить их из нынешнего vCenter сервера.



■ Рис. 2. Предлагаемый алгоритм миграции
 ■ Fig. 2. Proposed migration algorithm

Шаг 3. На новом (переданном пользователем) vCenter создать объект Datacenter, если это необходимо (если клиент не передал идентификатор Datacenter, в который он хочет добавить свой кластер).

Шаг 4. После получения информации о «базовом» объекте Datacenter создать в нем кластер.

Шаг 5. Добавить хосты в созданный кластер.

Шаг 6. Сконфигурировать и создать сеть, а именно настроить стандартный виртуальный коммутатор (Standard Virtual Switch, SVS — логический коммутатор, который соединяет физические порты сетевой карты (Physical Network Adapter, PNIC) ESXi-хоста и порты в Интернет через физический коммутатор [18]) или DVS, в зависимости от используемого подключения.

Шаги 7 и 8. Дополнительная настройка параметров кластера (шаг 7) и дополнительная настройка параметров сети (шаг 8) могут проводиться в любом порядке. Примером дополнительной настройки параметров кластера может быть настройка vSphere HA.

Шаг 9. Специальные действия, связанные с завершением процесса непосредственной миграции, например: регистрация Storage Provider, сохранение необходимых данных на сервере, другие действия на vCenter сервере.

Шаг 10. Очистка ненужных ресурсов на старом vCenter сервере: удаление кластера и других объектов, например в нашем случае необходимо удалить DVS, если к нему не подключены какие-либо ВМ и (или) хосты.

Представленный алгоритм предполагает безошибочное прохождение всего процесса миграции. Но это возможно не всегда, так как может потеряться сетевое соединение между vCenter и модулем, который проводит миграцию, или vCenter сервер по каким-либо иным причинам окажется в нерабочем состоянии. Поэтому предложенный алгоритм был расширен за счет механизма возврата к изначальному состоянию. При этом механизм возврата нужен не из каждого состояния. Например, ошибка при очистке старого vCenter (шаг 10) не должна вызывать откат к исходному состоянию, а также ряд операций из завершающих действий (шаг 9) могут не вызывать откат при ошибках. Исходя из этого дополним исходный алгоритм шагами для возврата в изначальное состояние.

Часть предварительных действий (шаг 1), таких как получение конфигурации и проверки, не нуждаются в механизме возврата, но при этом часть предварительных действий, осуществленных на сервере и (или) на vCenter сервере требуют механизма возврата, который выполняется в шаге 16.

При возникновении ошибок в момент удаления ESXi-хостов из действующего кластера

следует выполнить действия по восстановлению их состояния в кластере (шаг 14), т. е. при необходимости переподключить и (или) добавить их обратно в кластер и восстановить конфигурацию сети (при необходимости). Следует осуществить дополнительные действия по настройке кластера и сервера (шаг 15). Например, в нашем случае это подключение Storage Provider, настройка vSphere HA и выставление параметров сервера в «необходимые» значения.

Если создавался Datacenter (шаг 3) и при этом создание кластера (шаг 4) было завершено с ошибкой, то удалить созданный объект Datacenter (шаг 13) и выполнить шаги 14 и 15.

В случае если не получилось успешно добавить все ESXi-хосты в кластер (шаг 5), то отключить и удалить из нового vCenter все подключенные хосты. Затем удалить кластер (шаг 12), при необходимости удалить объект Datacenter (шаг 13), а также выполнить шаги 14 и 15.

Если не удалось сконфигурировать сеть, то удалить все подключенные хосты (шаг 11), а далее выполнить шаги 12, 13 (при необходимости), 14 и 15.

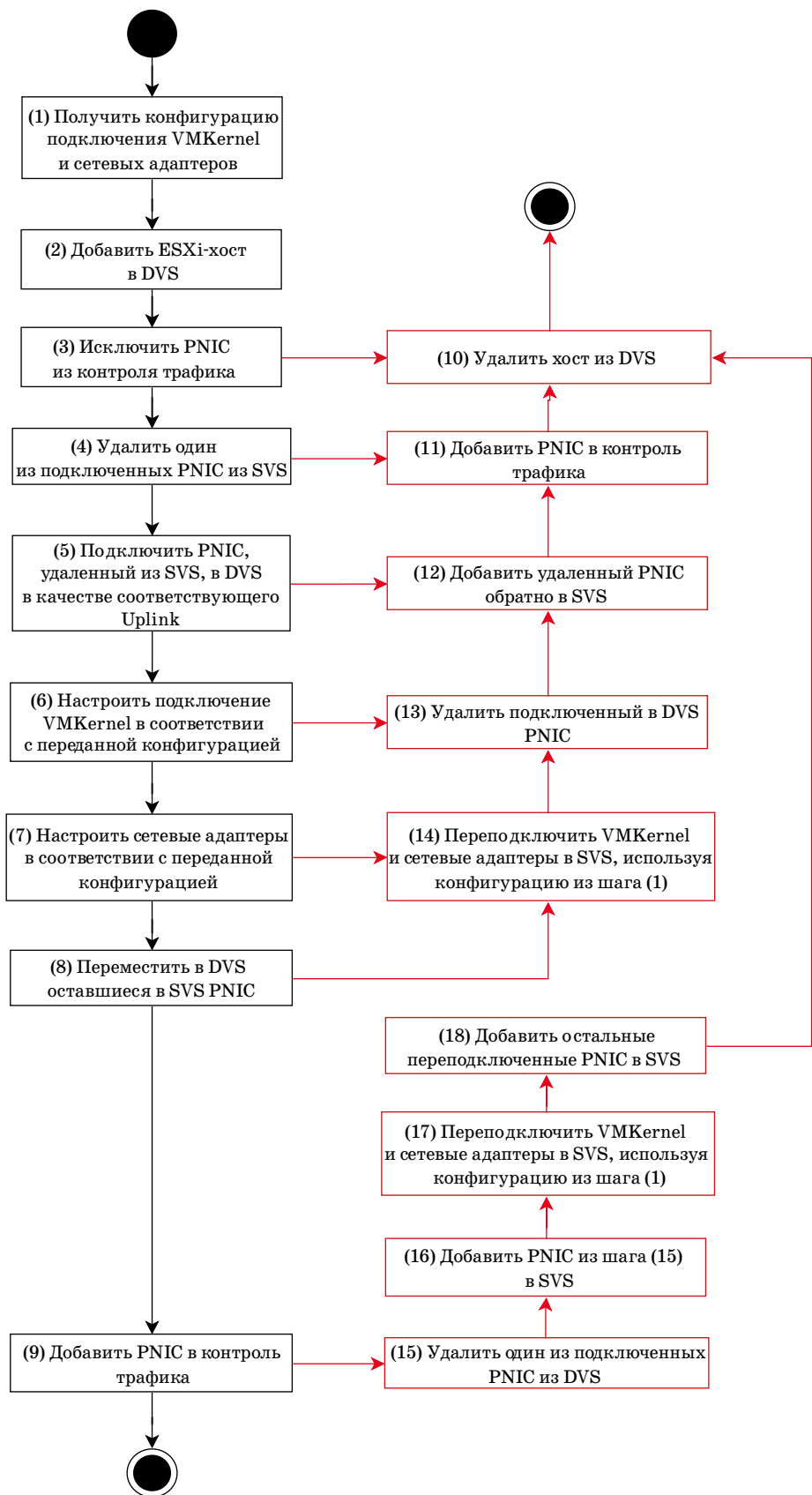
При возникновении каких-либо ошибок на шаге 7 (дополнительные настройки кластера), шаге 8 (дополнительные настройки сети) и шаге 9 (части этапов завершающих действий) выполнить все шаги по восстановлению конфигурации, а именно: шаги 11, 12, 13 (только если Datacenter создавался), 14 и 15.

Стоит, однако, заметить, что конкретные действия по восстановлению конфигурации сети на исходном vCenter могут отличаться в зависимости от того, каким образом должна быть настроена сеть и в каком состоянии была система при возникновении ошибки.

Настройка сети

Отдельно остановимся на восстановлении конфигурации сети для кластера, так как это важно при использовании DVS, который дает дополнительные возможности в конфигурации сети [18], например резервирование канала для определенного типа трафика.

До версии vCenter 7.0.0 была возможна миграция с одного DVS на другой, что упрощало как процесс миграции, так и процесс возврата к исходному состоянию. В связи с тем, что кластер подключается либо к одному DVS, либо к другому, либо, при ошибке в процессе миграции, одновременно к обоим, можно было реализовать один алгоритм миграции, который мигрировал и переподключал только те элементы и настройки, которые сейчас не подключены и не сконфигурированы для искомого DVS. Но стоит отметить,



■ **Рис. 3.** Алгоритм миграции подключения ESXi-хоста с SVS на DVS
 ■ **Fig. 3.** Algorithm of ESXi network migration from SVS to DVS

что в методике компании VMware не предусмотрено проведение миграции сети с одного DVS на другой, т. е. это являлось незадокументированной возможностью или могло даже трактоваться как ошибка.

В связи с вышесказанным и рекомендациями VMware миграция сети ESXi-хоста должна происходить в следующем порядке:

- 1) с DVS на SVS;
- 2) с SVS на DVS.

Алгоритм миграции с DVS на SVS должен допускать свой неоднократный перезапуск, что позволяет решать проблему частичного подключения через SVS. Алгоритм работы в виде псевдокода приведен в листинге.

Псевдокод миграции сети с распределенного виртуального коммутатора на стандартный виртуальный коммутатор

```
Для каждого из переданных хостов{
  Убрать pnic из IO-контроля
  active_pnic_connected = false
  Проверить, не подключен ли хост полностью или
  частично к какому-либо из SVS

  if not (передан SVS) и not (хост подключен к какому-либо
  из SVS){
    Создать SVS на хосте
    Сохранить имя SVS для хоста
  }
  Проверить, подключены ли к SVS активные pnic
  if активные pnic подключены{
    Добавить их в список мигрированных pnic
    active_pnic_connected = true
  }
  if not active_pnic_connected{
    first_pnic = Выбрать 1 из активных pnic
    Удалить first_pnic из proxySwitch хоста
    Добавить first_pnic в SVS
    Добавить first_pnic в список мигрированных pnic
  }
  Подключить в SVS необходимые VMKernel в соответ-
  ствии с конфигурацией
  Подключить в SVS необходимые сетевые адаптеры VM
  в соответствии с конфигурацией

  not_migrated_pnics = Все pnic в proxySwitch – Мигриро-
  вавшие pnic
  Удалить not_migrated_pnics из proxySwitch хоста
  Добавить not_migrated_pnics в SVS
  Добавить not_migrated_pnics в список мигрированных
  pnic
  Выполнять всегда): Добавить pnic в IO-контроль
}
```

Следует отметить, что алгоритм миграции с SVS на DVD так же, как и основной алгоритм

миграции, содержит возможность возврата к изначальному состоянию в случае ошибок.

Алгоритм, представленный на рис. 3, необходимо выполнить для каждого хоста.

А восстановление настроек сети и миграция сети на DVS в исходном vCenter сервере, с которого происходит миграция, идет по следующему алгоритму.

1. Если хосты были подключены к DVS в новом vCenter сервере, то:

1.1. Необходимо подключить их через SVS, при необходимости создать SVS.

1.2. Подключить их в необходимый DVS на исходном vCenter сервере.

1.3. Удалить созданный SVS, если создавался.

2. Если хосты не были подключены через DVS в новом vCenter сервере, то:

2.1. Если они полностью подключены через SVS, то подключить их в необходимый DVS на исходном vCenter сервере.

2.2. Если они не полностью подключены в SVS, то подключить только те pnic и сетевые адаптеры VM, которые были подключены к SVS.

2.3. Удалить SVS, если необходимо.

Предложенный подход гарантирует, что либо система мигрирует на новый vCenter сервер, либо ее состояние к концу выполнения алгоритма не изменится, т. е. система останется в полностью рабочем состоянии [17]. При этом обеспечивается бесперебойный доступ к системным VM во время миграции.

В случае если в процессе миграции потеряется связь с новым vCenter и при этом потеряется связь со старым, то система перейдет в несогласованное состояние.

Реализация алгоритма автоматической миграции

Предложенный алгоритм реализован в промышленной системе хранения данных (СХД) и может использоваться для миграции ESXi-кластера, созданного этой СХД. Для реализации использовались:

- язык программирования Java 11;
- библиотека Vert.x для реализации реактивного подхода [19];
- vim25 — SDK для работы с vSphere;
- Jira для отслеживания состояния задачи [20];
- Confluence для документирования работы;
- git/GitHub в качестве системы контроля версий;
- Jenkins для реализации Continuous Integration (CI);
- gradle в качестве сборщика проекта;
- SonarQube/SonarLint для проведения статического анализа кода.

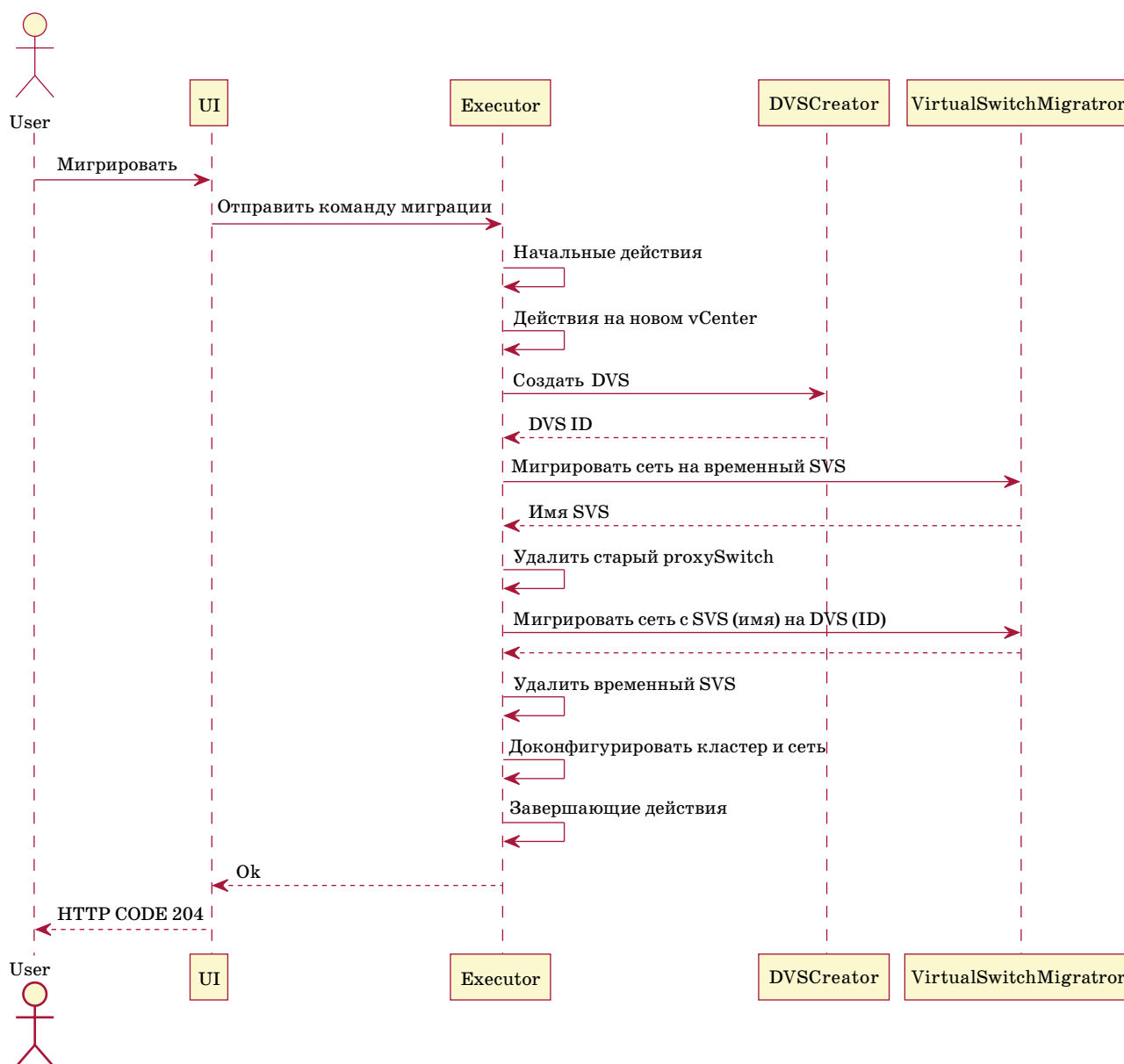
Для миграции были созданы дополнительные микросервисы: миграции с DVS на SVS и обратно и сервис по созданию и конфигурированию DVS. Эти микросервисы также использованы в других частях продукта. Сервис миграции — это тоже микросервис, диаграмма взаимодействия микросервисов представлена на рис. 4.

Для взаимодействия с ESXi-хостами и vCenter сервером используется инструментарий (SDK), представленный компанией VMware, — vim25.

Для решения проблемы отказа (отключения, перезапуска) приложения во время выполнения процесса миграции используется механизм ма-

шин состояний. Этот механизм сохраняет текущий шаг выполнения и контекст выполнения команды в базе данных, а в случае перезапуска приложения восстанавливает контекст выполнения, включая шаг, на котором выполнение прервалось, и продолжает выполнение цепочки команд с сохраненного шага.

Реализованный код проверялся модульным, интеграционным и системным тестированием. Для системного тестирования применялись следующие тесты, реализованные на языке Python 3 с использованием библиотек requests и pyVim:



■ **Рис. 4.** MSC-диаграмма взаимодействия микросервисов
 ■ **Fig. 4.** MSC-diagram of microservice communications

- 30 успешных миграций между двумя vCenter серверами;
- миграция с ошибкой во время создания DVS;
- миграция с ошибкой во время миграции сети на SVS;
- миграция с ошибкой во время миграции сети на DVS.

Результаты сравнения ручной и автоматической миграции

В результате работы был реализован и внедрен модуль миграции и внесены изменения в ряд модулей промышленной СХД. Эти изменения позволили автоматизировать процесс миграции ESXi-кластера между разными vCenter с возвратом к исходному состоянию в случае возникновения ошибки.

Для проведения ручных и автоматических экспериментов по измерению времени миграции (табл. 2) использовался ноутбук Dell latitude 5400 с 16 Гб оперативной памяти. На нем была запущена программа, реализующая алгоритм

- **Таблица 2.** Результаты сравнения ручной и автоматической миграции
- **Table 2.** Results of comparison between automated and manual migration

Критерий	Тип миграции	
	Ручная	Автоматическая
Время, кластер из двух хостов	55 мин	3 мин
Время, кластер из четырех хостов	65 мин	5 мин
Шанс ошибки	Большой (инструкция больше 20 страниц)	Крайне низкий
Возможность откатиться при ошибке	Да	Да
Возможность вернуться назад с любой точки	Да	Нет
Необходимые входные данные	Настройки кластера, ESXi-хоста, vCenter сервера, конфигурация DVS, способ регистрации Storage Provider	Настройки ESXi-хоста, vCenter сервера, логин и пароль для регистрации Storage Provider

миграции. Во время работы программы установлено защищенное соединение с vCenter сервером и промышленной СХД, которое происходило на скорости передачи данных 60 Мб/с. Во всех экспериментах на каждом из ESXi-хостов была запущена одна системная ВМ, а на кластере дополнительно развернуты четыре клиентские ВМ.

Время, необходимое на автоматическую миграцию, измерено достоверно путем получения медианного значения 30 миграций (табл. 3) одного ESXi-кластера из двух хостов (рис. 5).

Ручная миграция измерялась одним человеком, и результат, представленный в таблице, — среднее значение трех миграций, так как процесс трудоемкий. При этом следует отметить, что такие измерения могут быть недостоверными, поскольку у человека увеличивается опыт и он может выполнять операции по памяти, а не по инструкции. Но даже такие данные позволяют рассчитать примерное ожидаемое время на одну миграцию.

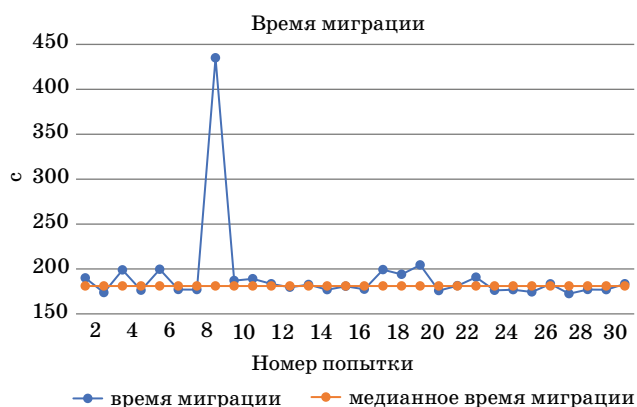
Также проведены эксперименты по тестированию работоспособности алгоритма возвращения к исходной конфигурации, для чего на каждом шаге алгоритма эмулировалась ошибка. Применялись:

- аппаратное отключение vCenter сервера;
- программное эмулирование завершения исполнимой задачи с ошибкой;
- передача некорректных данных.

- **Таблица 3.** Результаты тестирования автоматической миграции

■ **Table 3.** Results of tests for automated migration

Номер миграции	Время, затраченное на миграцию, с	Номер миграции	Время, затраченное на миграцию, с
1	189,94	16	177,41
2	173,68	17	199,19
3	198,888	18	193,95
4	176,28	19	204,40
5	199,54	20	175,87
6	177,03	21	181,27
7	176,94	22	190,71
8	435,03	23	176,14
9	187,06	24	176,87
10	189,04	25	174,35
11	183,42	26	183,35
12	179,59	27	172,46
13	182,57	28	177,11
14	176,87	29	176,88
15	180,68	30	183,37



■ **Рис. 5.** Время миграции каждой попытки и медианное время миграции для кластера из двух хостов
 ■ **Fig. 5.** Migration time for each attempt and median time

В результате во всех экспериментах на исходном vCenter сервере восстановилась корректная изначальная конфигурация системы.

Благодаря автоматизации трудоемкость процесса миграции снизилась как по времени примерно в 15–18 раз, так и в объеме необходимых входных данных. Пользователь должен предоставить только данные о паролях для ESXi-хостов, данные для доступа в vCenter сервер, логин и пароль для регистрации Storage Provider. Также пользователю необходимо узнать один идентификатор, который получается с СХД при

помощи одного REST запроса. При возникновении ошибки в процессе миграции изначальная конфигурация кластера восстанавливается на исходном vCenter сервере, что минимизирует время нахождения кластера и системы в некорректном состоянии.

Заключение

Проведены исследование и анализ существующих подходов миграции ESXi-кластеров между разными экземплярами vCenter серверов. Предложен алгоритм миграции ESXi-кластеров между разными экземплярами vCenter серверов с возможностью возврата к исходной конфигурации. Предложенный алгоритм реализован в одной из промышленных систем хранения данных.

Проведены эксперименты для оценки работоспособности алгоритма возвращения к исходной конфигурации. Продемонстрированы результаты снижения трудоемкости миграции ESXi-кластеров между разными vCenter серверами для автоматического подхода, который показал выигрыш во времени в 15–18 раз по сравнению с традиционным ручным подходом. На основе проведенных экспериментов можно также судить об уменьшении времени нахождения кластера в неконсистентном состоянии за счет предложенного механизма отката.

Литература

1. Sharma K. An alleviated model for private cloud deployment using VMware. *2017 Intern. Conf. on Information, Communication, Instrumentation and Control (ICICIC)*, 2017, pp. 1–3. doi:10.1109/ICOMICON.2017.8279164
2. Попок Е. Л., Богомолов Е. А. Методика миграции виртуальных нагрузок с платформы виртуализации VMware vsphere на платформу виртуализации Microsoft hyper-v. *Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета*, 2019, № 153, с. 66–80. doi:10.21515/1990-4665-153-007. <http://ej.kubagro.ru/2019/09/pdf/07.pdf> (дата обращения: 29.01.2022).
3. Nyrkov A. P., Katorin Y. F., Gaskarov V. D., Brazhnikova L. S., and Vikhrov N. Analysis of platform vulnerabilities for the virtualization process. *2018 IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering (EConRus)*, 2018, pp. 94–97. doi:10.1109/EConRus.2018.8317038
4. Kolahi S. S., Hora V. S., Singh A. P., Bhatti S., Yeeda S. R. Performance comparison of cloud computing/IoT virtualization software, Hyper-V vs vSphere. *2020 Advances in Science and Engineering Technology Intern. Conf. (ASET)*, 2020, pp. 1–6. doi:10.1109/ASET48392.2020.9118185
5. Anwar A. H., Atia G., Guirguis M. It's time to migrate! A game-theoretic framework for protecting a multi-tenant cloud against collocation attacks. *2018 IEEE 11th Intern. Conf. on Cloud Computing (CLOUD)*, 2018, pp. 725–731. doi:10.1109/CLOUD.2018.00099
6. Voinov N., Selin I., Drobintsev P., Kotlyarov V. An approach for managing hybrid supercomputer resources in photogrammetric tasks. *CEUR Workshop Proc.*, 2018, pp. 12–19.
7. Đorđević B., Timčenko V., Nikolić E., Davidović N. Comparing performances of native host and virtualization on ESXi hypervisor. *2021 20th Intern. Symp. INFOTEH-JAHORINA (INFOTEH)*, 2021, pp. 1–4. doi:10.1109/INFOTEH51037.2021.9400648
8. Токарева О. А. *Применение систем оптимизации информационных ресурсов*. Наука молодых — будущее России: сб. науч. ст. 2-й Междунар. науч. конф. перспективных разработок молодых ученых/под ред. А. А. Горохова, Курск, Воронежский институт высоких технологий, 13–14 декабря 2017 г., 2017, с. 387–393.

9. Свидетельство о государственной регистрации программы для ЭВМ 2016660670 РФ. *ЭrucBM*, А. А. Шарапов. № 2016617311; заявл. 07.07.2016; опублик. 20.09.2016.
10. Ye K., Jiang X., Ma R., Yan F. VC-migration: Live migration of virtual clusters in the cloud. *2012 ACM/IEEE 13th Intern. Conf. on Grid Computing*, 2012, pp. 209–218. doi:10.1109/Grid.2012.27
11. Jun D., Yuanyuan Y. A load balancing and multi-tenancy oriented data center virtualization framework. *IEEE Transactions on Parallel and Distributed Systems*, 2017, no. 8, pp. 2131–2144. doi:10.1109/TPDS.2017.2657633
12. Wahler M., Richter S., Oriol M. Non-disruptive large-scale component updates for real-time controllers. *2011 IEEE 27th Intern. Conf. on Data Engineering Workshops*, 2011, pp. 174–178. doi:10.1109/ICDEW.2011.5767631
13. Marsico A., Doriguzzi-Corin R., Gerola M., Siracusa D., Schwabe A. A non-disruptive automated approach to update SDN applications at runtime. *NOMS 2016 — 2016 IEEE/IFIP Network Operations and Management Symp.*, 2016, pp. 1007–1008. doi:10.1109/NOMS.2016.7502946
14. Bogatyrev V. A., Derkach A. N. Evaluation of a cyber-physical computing system with migration of virtual machines during continuous computing. *Computers*, 2020, vol. 9, no. 2, p. 42. doi:10.3390/computers9020042
15. Zhang W., Chen Y., Gao X., Mo Z. Cluster-aware virtual machine collaborative migration in media cloud. *IEEE Transactions on Parallel and Distributed Systems*, 2017, no. 10, pp. 2808–2822. doi:10.1109/TPDS.2017.2697381
16. Fei Z., Xiaoming F., Ramin Y. CBase: A new paradigm for fast virtual machine migration across data centers. *2017 17th IEEE/ACM Intern. Symp. on Cluster, Cloud and Grid Computing (CCGRID)*, 2017, pp. 284–293. doi:10.1109/CCGRID.2017.26
17. Cui L., Hao Z., Peng Y., Yun X. Piccolo: A fast and efficient rollback system for virtual machine clusters. *IEEE Transactions on Parallel and Distributed Systems*, 2017, no. 8, pp. 2328–2341. doi:10.1109/TPDS.2017.2668403
18. Kamla R. Z., Yahiya T., Mustafa N. B. An implementation of software routing for building a private cloud. *Intern. Journal of Computer Network and Information Security*, 2018, no. 3, pp. 1–7. doi:10.5815/ijcnis.2018.03.01
19. Rasheedh J. A., Saradha S. Reactive microservices architecture using a framework of fault tolerance mechanisms. *2021 Second Intern. Conf. on Electronics and Sustainable Communication Systems (ICESC)*, 2021, pp. 146–150. doi:10.1109/ICESC51422.2021.9532893
20. Sarkan H. M., Ahmad T. P. S., Bakar A. A. Using JIRA and Redmine in requirement development for agile methodology. *2011 Malaysian Conf. in Software Engineering*, 2011, pp. 408–413. doi:10.1109/MyS-EC.2011.6140707

UDC 004.457

doi:10.31799/1684-8853-2022-2-20-31

Algorithm of ESXi cluster migration between different vCenter servers with the ability to rollbackA. A. Petrov^a, Master Student, orcid.org/0000-0002-7465-3317I. V. Nikiforov^a, PhD, Tech., Associate Professor, orcid.org/0000-0003-0198-1886, nikiforov_iv@spbstu.ru

S. M. Ustinov, Dr. Sc., Tech., Professor, orcid.org/0000-0003-4088-4798

^aPeter the Great St. Petersburg Polytechnic University, 29, Politekhnicheskaya St., 195251, Saint-Petersburg, Russian Federation

Introduction: Hypervisor managers may become deprecated or outdated or may fail. In addition, customers may want to rebalance hypervisor load. All these can cause problems related to the incompatibility of software with different manager versions or to the level of application integration. An effective solution to such problems is to automatically migrate the configuration and resources between hypervisor manager instances. **Purpose:** To develop an algorithm of the automatic migration of configurations and resources of an ESXi cluster between different vCenter Server instances with the ability to rollback in the event of a failure. **Results:** An automatic migration algorithm has been proposed. On its basis a program module for enterprise storage has been developed, which allows migrating an ESXi cluster on which a storage system is deployed from one vCenter Server to another. The module handles errors, and in case an exception or error occurs the module initiates a rollback operation in order to revert to the original vCenter Server. The study presents an experimental evaluation based on the comparison of the manual migration and the execution of the developed module. The time needed for the migration procedure has been reduced from 55 minutes to about 3 minutes, i.e., by 15–18 times. The amount of information to be known or taken into consideration by a user has also been reduced. **Practical relevance:** The study results have been used for enterprise storage. The developed module makes migrations to a new vCenter Server easier and faster.

Keywords — hypervisors, vCenter Server, VMware ESXi, migration, automatization, Distributed Virtual Switch.

For citation: Petrov A. A., Nikiforov I. V., Ustinov S. M. Algorithm of ESXi cluster migration between different vCenter servers with the ability to rollback. *Informatsionno-upravlyaiushchie sistemy* [Information and Control Systems], 2022, no. 2, pp. 20–31 (In Russian). doi:10.31799/1684-8853-2022-2-20-31

References

- Sharma K. An alleviated model for private cloud deployment using VMware. *2017 Intern. Conf. on Information, Communication, Instrumentation and Control (ICICIC)*, 2017, pp. 1–3. doi:10.1109/ICOMICON.2017.8279164
- Popok L. E., Bogomolov E. A. Method of migration of virtual loads from VMware vSphere virtual platform to the Microsoft Hyper-V virtualization platform. *Polythematic Online Scientific Journal of Kuban State Agrarian University*, 2019, no. 153, pp. 66–80 (In Russian). doi:10.21515/1990-4665-153-007. Available at: <http://ej.kubagro.ru/2019/09/pdf/07.pdf> (accessed 29 January 2022).
- Nyrkov A. P., Katorin Y. F., Gaskarov V. D., Brazhnikova L. S., and Vikhrov N. Analysis of platform vulnerabilities for the virtualization process. *2018 IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2018, pp. 94–97. doi:10.1109/EIConRus.2018.8317038
- Kolahi S. S., Hora V. S., Singh A. P., Bhatti S., Yeeda S. R. Performance comparison of cloud computing/IoT virtualization software, Hyper-V vs vSphere. *2020 Advances in Science and Engineering Technology Intern. Conf. (ASET)*, 2020, pp. 1–6. doi:10.1109/ASET48392.2020.9118185
- Anwar A. H., Atia G., Guirguis M. It's time to migrate! A game-theoretic framework for protecting a multi-tenant cloud against collocation attacks. *2018 IEEE 11th Intern. Conf. on Cloud Computing (CLOUD)*, 2018, pp. 725–731. doi:10.1109/CLOUD.2018.00099
- Voinov N., Selin I., Drobintsev P., Kotlyarov V. An approach for managing hybrid supercomputer resources in photogrammetric tasks. *CEUR Workshop Proc.*, 2018, pp. 12–19.
- Đorđević B., Timčenko V., Nikolić E., Davidović N. Comparing performances of native host and virtualization on ESXi hypervisor. *2021 20th Intern. Symp. INFOTEH-JAHORINA (INFOTEH)*, 2021, pp. 1–4. doi:10.1109/INFOTEH51037.2021.9400648
- Tokareva O. A. Using of information resources optimization systems. *Sbornik nauchnyh statej 2-j Mezhdunarodnoj nauchnoj konferencii perspektivnyh razrabotok molodyh uchenyh "Nauka molodyh — budushchee Rossii"* [Proc. 2nd Intern. Science Conf. of Promising Developments of Young Scientists "Science of young — future of Russia"]. Eds. A. A. Gorohov. Kursk, Voronezhskij institut vysokih tekhnologij, 2017, pp. 387–393 (In Russian).
- Sharapov A. A. *ErisVM* [ErisVM]. Certificate of state registration of the computer program Russia, no. 2016660670, 2016.
- Ye K., Jiang X., Ma R., Yan F. VC-Migration: Live migration of virtual clusters in the cloud. *2012 ACM/IEEE 13th Intern. Conf. on Grid Computing*, 2012, pp. 209–218. doi:10.1109/Grid.2012.27
- Jun D., Yuanyuan Y. A load balancing and multi-tenancy oriented data center virtualization framework. *IEEE Transactions on Parallel and Distributed Systems*, 2017, no. 8, pp. 2131–2144. doi:10.1109/TPDS.2017.2657633
- Wahler M., Richter S., Oriol M. Non-disruptive large-scale component updates for real-time controllers. *2011 IEEE 27th Intern. Conf. on Data Engineering Workshops*, 2011, pp. 174–178. doi:10.1109/ICDEW.2011.5767631
- Marsico A., Doriguzzi-Corin R., Gerola M., Siracusa D., Schwabe A. A non-disruptive automated approach to update SDN applications at runtime. *NOMS 2016 — 2016 IEEE/IFIP Network Operations and Management Symp.*, 2016, pp. 1007–1008. doi:10.1109/NOMS.2016.7502946
- Bogatyrev V. A., Derkach A. N. Evaluation of a cyber-physical computing system with migration of virtual machines during continuous computing. *Computers*, 2020, vol. 9, no. 2, p. 42. doi:10.3390/computers9020042
- Zhang W., Chen Y., Gao X., Mo Z. Cluster-aware virtual machine collaborative migration in media cloud. *IEEE Transactions on Parallel and Distributed Systems*, 2017, no. 10, pp. 2808–2822. doi:10.1109/TPDS.2017.2697381
- Fei Z., Xiaoming F., Ramin Y. CBase: A new paradigm for fast virtual machine migration across data centers. *2017 17th IEEE/ACM Intern. Symp. on Cluster, Cloud and Grid Computing (CCGRID)*, 2017, pp. 284–293. doi:10.1109/CCGRID.2017.26
- Cui L., Hao Z., Peng Y., Yun X. Piccolo: A fast and efficient rollback system for virtual machine clusters. *IEEE Transactions on Parallel and Distributed Systems*, 2017, no. 8, pp. 2328–2341. doi:10.1109/TPDS.2017.2668403
- Kamla R. Z., Yahiya T., Mustafa N. B. An implementation of software routing for building a private cloud. *Intern. Journal of Computer Network and Information Security*, 2018, no. 3, pp. 1–7. doi:10.5815/ijenis.2018.03.01
- Rasheedh J. A., Saradha S. Reactive microservices architecture using a framework of fault tolerance mechanisms. *2021 Second Intern. Conf. on Electronics and Sustainable Communication Systems (ICESC)*, 2021, pp. 146–150. doi:10.1109/ICESC51422.2021.9532893
- Sarkan H. M., Ahmad T. P. S., Bakar A. A. Using JIRA and Redmine in requirement development for agile methodology. *2011 Malaysian Conf. in Software Engineering*, 2011, pp. 408–413. doi:10.1109/MySEC.2011.6140707