

Санкт-Петербургский государственный университет

Программная инженерия

Сергеев Евгений Дмитриевич

# Интеграция системы хранения данных Dell EMC Unity с гипервизором Microsoft Hyper-V

Бакалаврская работа

Научный руководитель:  
ст. преп. Сартасов С.Ю.

Рецензент:  
ведущий инженер-программист  
ООО «Санкт-Петербургский Центр Разработок EMC»  
Беликов Я.С.

Санкт-Петербург  
2018

SAINT PETERSBURG STATE UNIVERSITY

Software engineering

Sergeev Evgeny

Integration of data storage system Dell EMC Unity with  
hypervisor Microsoft Hyper-V

Graduation Thesis

Scientific supervisor:

senior lecturer, Stanislav Sartasov

Reviewer:

software principal engineer, Dell EMC, Yakov Belikov

Saint Petersburg

2018

# Оглавление

<b>Введение</b>	<b>5</b>
<b>1. Постановка задачи</b>	<b>6</b>
<b>2. Обзор</b>	<b>7</b>
2.1 СХД Dell EMC Unity	7
2.2 Гипервизор Hyper-V	8
2.3 Windows Management Instrumentation	10
2.4 System Center Virtual Machine Manager	11
<b>3. Требования к системе</b>	<b>12</b>
3.1 Функциональные требования	12
3.2 Нефункциональные требования	14
<b>4. Архитектура и детали реализации</b>	<b>17</b>
4.1 Сетевое взаимодействие	17
4.2 Архитектура системы	19
4.3 Реализованные операции	21
4.4 Тестирование	22
<b>Заключение</b>	<b>23</b>
<b>Список литературы</b>	<b>24</b>

# Введение

На сегодняшний день вычислительные ресурсы большинства серверов при выполнении ими повседневных задач используются не более чем на 15% [1], что приводит к увеличению их количества. Виртуализация позволяет сократить количество серверов путем запуска на одном физическом устройстве множества операционных систем в виде виртуальных машин, каждая из которых использует вычислительные мощности одного и того же физического сервера. Это позволяет экономить на стоимости оборудования, снизить электропотребление и тепловыделение системы. Одним из инструментов, позволяющих использовать виртуальные машины, является технология Hyper-V [2]. Подобные технологии называются гипервизорами.

Надежное хранение данных – это задача, которую приходится решать каждой организации. Система хранения данных (СХД) является важной частью серверной инфраструктуры компании. Современные системы хранения данных – это сложные программно-аппаратные комплексы. Примером СХД является система Dell EMC Unity [3].

Dell EMC Unity включает программную часть, которая позволяет управлять интеграцией СХД с гипервизорами VMware ESXi Server. Интеграция включает предоставление доступа к памяти дискового массива в виде сетевого хранилища или блочного устройства, выполнение резервных копий, управление запущенными на серверах виртуальными машинами. Интеграция с гипервизором автоматизирует однообразные действия и снимает нагрузку с системного администратора.

# 1. Постановка задачи

Целью данной работы является разработка системы, предоставляющей интерфейс для управления взаимодействием СХД Dell EMC Unity с серверами под управлением ОС Microsoft Windows и интеграцией с гипервизором Microsoft Hyper-V. Для достижения этой цели были сформулированы следующие задачи.

- Провести анализ требований к системе.
- Выполнить проектирование системы.
- Реализовать требуемую систему.

## 2. Обзор

### 2.1 СХД Dell EMC Unity

Система хранения данных (СХД) — это комплекс программного и аппаратного обеспечения, предназначенного для надежного хранения информации и передачи большого объема данных. Надежность хранения достигается использованием технологии RAID [4].

Выделяют следующие виды систем хранения данных: сетевые системы хранения данных (Network Attached Storage, NAS) [5] и сети хранения данных (Storage Area Network, SAN) [6].

SAN — это выделенная сеть, которая объединяет серверы с внешними устройствами хранения данных. SAN предоставляет для СХД блочный интерфейс доступа, обычно реализованный посредством протоколов Fibre Channel [7] и iSCSI [8]. Дисковый массив СХД делится на логические единицы, к которым получают доступ клиентские устройства. При этом серверная ОС распознает подключенные ресурсы как локальные. СХД оперирует только логическими единицами, а создание на них файловых систем и размещение файлов выполняет клиентское устройство, которому СХД предоставила доступ.

NAS представляет собой файловый сервер, который предоставляет другим устройствам в сети возможность хранить файлы и получать к ним доступ.

Система Dell EMC Unity представляет собой СХД на базе жестких дисков и твердотельных накопителей. Dell EMC Unity поддерживает как

технологии SAN, так и NAS и поддерживает как блочные протоколы Fibre Channel и iSCSI, так и файловые – SMB/CIFS и NFS.

СХД находится под управлением операционной системы SUSE Linux Enterprise Server, на которой выполняется программное обеспечение, обеспечивающее интеграцию с гипервизором VMware ESXi Server [9]. Программное обеспечение имеет три вида интерфейсов: консольный, REST API и графический на основе REST API. Интеграция включает следующее:

- предоставление доступа к памяти дискового массива в виде сетевых хранилищ или блочных устройств;
- выполнение снимков состояния и резервных копий блочных устройств, файловых устройств и виртуальных машин;
- миграцию виртуальных машин;
- предоставление информации о виртуальных машинах;
- журнализацию выполняемых операций.

## **2.2 Гипервизор Hyper-V**

Гипервизор — это программное обеспечение, которое позволяет параллельно выполняться на одном физическом компьютере нескольким операционным системам. Гипервизор предоставляет операционным системам изолированное окружение, доступ к аппаратным ресурсам и средства взаимодействия друг с другом так, как если бы эти ОС были запущены на различных физических устройствах.

Гипервизоры разделяют на два типа по способу запуска: первого типа и второго типа. Гипервизоры первого типа ("bare-metal") запускаются

непосредственно на физическом устройстве и работают с оборудованием напрямую. Гипервизоры второго типа ("hosted") запускаются поверх некоторой операционной системы и абстрагируют виртуальные машины, запущенные на ней, от ОС, запущенной на физическом устройстве, и этого устройства.

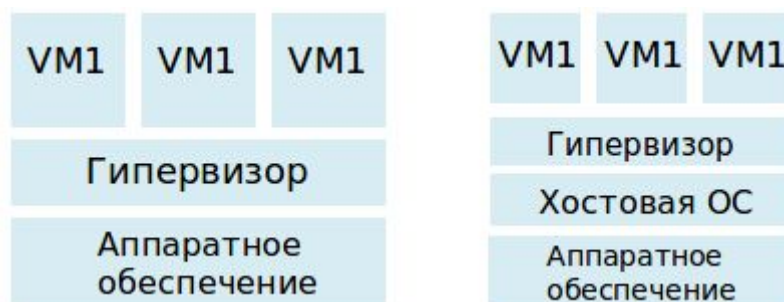


Рис. 1. Гипервизоры первого и второго типа.

По архитектуре гипервизоры первого типа делят на монолитные и микроядерные. Гипервизоры с монолитной архитектурой включают драйверы аппаратных устройств в свой код. Драйверы устройств в гипервизорах с микроядерной архитектурой работают внутри хостовой ОС, которая, в свою очередь, работает в виртуальном окружении, как и остальные виртуальные машины. Непосредственный доступ к физическому оборудованию при такой архитектуре имеет только хостовая ОС, а гипервизор выполняет функции выделения памяти и распределением процессорного времени.



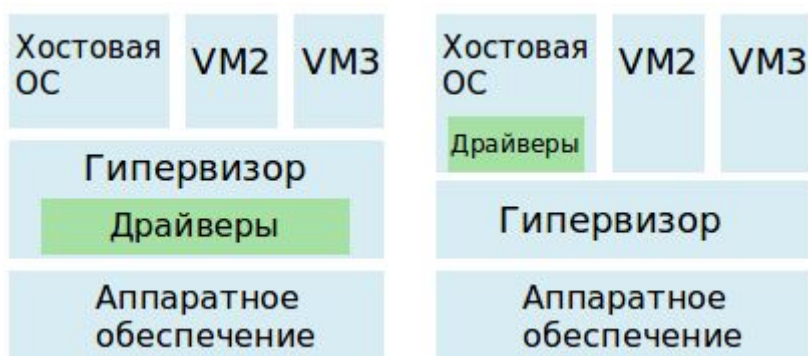


Рис. 2. Монолитные и микроядерные гипервизоры.

Microsoft Hyper-V — это гипервизор первого типа с микроядерной архитектурой. Hyper-V существует в двух вариантах: как отдельная операционная система Microsoft Hyper-V Server и как роль серверной операционной серии Windows Server.

## 2.3 Windows Management Instrumentation

Windows Management Instrumentation [10] (WMI) — стандартный инструмент, предназначенный для удаленного администрирования серверов под управлением ОС Windows. Он предоставляет открытый интерфейс для управления различными компонентами ОС и поддерживает протоколы DCOM [11] и WS-Management [12]. В частности, гипервизор Hyper-V предоставляет интерфейс, позволяющий пользователям управлять всеми его компонентами посредством WMI. WMI является реализацией стандарта CIM [13] в ОС Windows

Common Information Model (CIM) — открытый стандарт, определяющий объектно-ориентированную архитектуру, которая представляет управляемые элементы в виде совокупности объектов и их отношений и предоставляет способ управления этими элементами. Архитектура построена по объектно-ориентированному принципу:

элементы представлены классами CIM, между которыми определены отношения ассоциации и наследования, и у которых можно вызывать определенные методы с заданными параметрами. Классами CIM обычно описываются такие физические и логические компоненты, как компьютеры, сети, логические диски, процессы и т.д.

Со стандартом CIM связан протокол WS-Management, который описывает удаленное управление CIM объектами. Он основан на протоколе SOAP [14] и работает поверх протокола HTTP/HTTPS. WS-Management предоставляет следующие возможности:

- запрос, создание, изменения и удаление CIM объекта;
- получение множества объектов одного типа запросом на SQL-подобном языке;
- подписка на события (создание, изменение, удаление, вызов метода CIM объекта);
- вызов метода со строго типизированными входными и выходными параметрами у указанного CIM объекта.

Для запроса множества объектов используется язык запросов WQL [15] (Windows Management Instrumentation Query Language). Он представляет собой подмножество стандарта ANSI SQL, в котором возможен только запрос данных (доступен только оператор SELECT) с заданными параметрами.

## **2.4 System Center Virtual Machine Manager**

SCVMM (System Center Virtual Machine Manager) — продукт серии Microsoft System Center. Он предоставляет центр управления кластером серверов и запущенными на различных гипервизорах виртуальными машинами. При этом физический или виртуальный сервер с ролью SCVMM представляет собой узел, через который происходит взаимодействие со всеми серверами кластера.

## 3. Требования к системе

В этом разделе описываются требования к системе с обсуждением причин, по которым они были выдвинуты. Требования разделены на две категории — функциональные и нефункциональные.

### 3.1 Функциональные требования

#### 3.1.1 Интерфейс командной строки

Система должна предоставлять интерфейс для выполнения реализуемых команд на скриптовом языке `bash` [16].

Данное требование поставлено в связи с тем, что основная задача разрабатываемой системы — предоставить системному администратору возможность управлять взаимодействием СХД с кластером серверов из единого места и автоматизировать однотипные операции. Использование скриптового языка `bash` позволяет писать переиспользуемые сценарии.

#### 3.1.2 Получение информации о конечных серверах и виртуальных машинах

Система должна предоставлять возможность получения следующей информации о конечных серверах:

- IP адреса;
- IQN (iSCSI Qualified Name);
- WWN (World Wide Name);
- логические диски с указанием свободного и занятого места;
- сетевые диски с указанием свободного и занятого места.

Кроме того, система должна предоставлять возможность получения следующей информации о виртуальных машинах:

- имя;
- операционная система;
- IP адреса;
- логические диски с указанием свободного и занятого места;
- сетевые диски с указанием свободного и занятого места;
- состояние (“power on”, “power off”, “suspended”).

### **3.1.3 Добавление блочного устройства на сервер**

Система должна реализовывать возможность предоставления блочного устройства СХД на конечный сервер в виде логического диска, используя протоколы iSCSI и Fibre Channel.

### **3.1.4 Добавление сетевого диска на сервер**

Система должна реализовывать возможность предоставления сетевого хранилища СХД на конечный сервер в виде сетевого диска, используя протоколы CIFS [17] и NFS [18].

### **3.1.5 Выполнение снимка состояния виртуальной машины**

Система должна предоставлять возможность выполнения снимка состояния виртуальной машины, который должен быть сохранен на указанном логическом или сетевом диске.

### **3.1.6 Миграция виртуальной машины**

Система должна предоставлять возможность миграции виртуальной машины с одного гипервизора на другой. При этом виртуальные диски должны быть сохранены на указанном логическом или сетевом диске.

### **3.1.7 Журналирование операций**

Система должна выполнять журналирование всех совершаемых операций с указанием:

- названия операции;
- системного времени начала операции;
- системного времени окончания операции;
- состояния операции (RUNNING, SUCCESS, FAILED, REVERTED, HALT);
- IP адрес сервера, над которым операция выполняется.

### **3.1.8 Отмена операций**

Система должна предоставлять возможность отмены любой завершенной операции (находящейся в состоянии “SUCCESS”).

## **3.2 Нефункциональные требования**

### **3.2.1 Язык программирования**

Разрабатываемая система также должна быть реализована на языке программирования C++.

Данное требование поставлено в связи с тем, что разрабатываемая система будет являться частью существующей программной платформы, которая выполняет управление СХД Dell EMC Unity, и реализуемые в рамках разрабатываемой системы компоненты будут линковаться к

существующей программной платформе как разделяемые библиотеки (“shared object”). Существующая программная платформа реализована с использованием языка программирования C++, поэтому разрабатываемая система также должна быть реализована на языке программирования C++.

### **3.2.2 Транзакционность операций**

Выполняемые разрабатываемой системой команды должны выполняться транзакционно. При отмене выполняемой команды система должна отменить все выполненные на данный момент в рамках данной команды действия.

### **3.2.3 Ограничение используемой оперативной памяти**

Система должна использовать не более 1 Гб оперативной памяти.

Данное требование поставлено в связи с тем, что при выполнении определенных операций, таких как запрос информации о всех виртуальных машинах на всех доступных серверах, объем полученной информации может превышать объем доступной оперативной памяти. Таким образом, система должна быть способна фрагментировать большие запросы и обрабатывать их отдельными частями.

### **3.2.4 Поддерживаемые операционные системы и их конфигурация**

Разрабатываемая система должна поддерживать следующие операционные системы: Windows Server 2012, Windows Server 2012 R2, Windows Server 2016.

Система должна поддерживать интеграцию с сервером в стандартной конфигурации с использованием только стандартных компонент, без каких-либо дополнительных надстроек и расширений. Гарантируется, что

у сервера будет активна роль Hyper-V ([1]), а также будет запущен сервис WMI ([9]).

### **3.2.5 Поддерживаемые типы пользователей Windows**

Для того, чтобы СХД могла производить операции на удаленном сервере под управлением операционной системы Windows, СХД должны быть предоставлены имя и пароль пользователя, обладающего полномочиями администраторы на целевом сервере. СХД должна осуществлять поддержку как локальных пользователей, так и пользователей активного каталога (“Active Directory”).

### **3.2.6 Поддержка SCVMM**

Должна быть возможность выполнить все реализуемые операции на сервере как напрямую, так и посредством интерфейса SCVMM.



## 4. Архитектура и детали реализации

### 4.1 Сетевое взаимодействие

На рисунке 3 представлена диаграмма компонентов программной системы, включающей СХД Dell EMC Unity и ряд серверов под управлением ОС Windows. Реализованная в рамках работы компонента обозначена “Hyper-V integration component”.

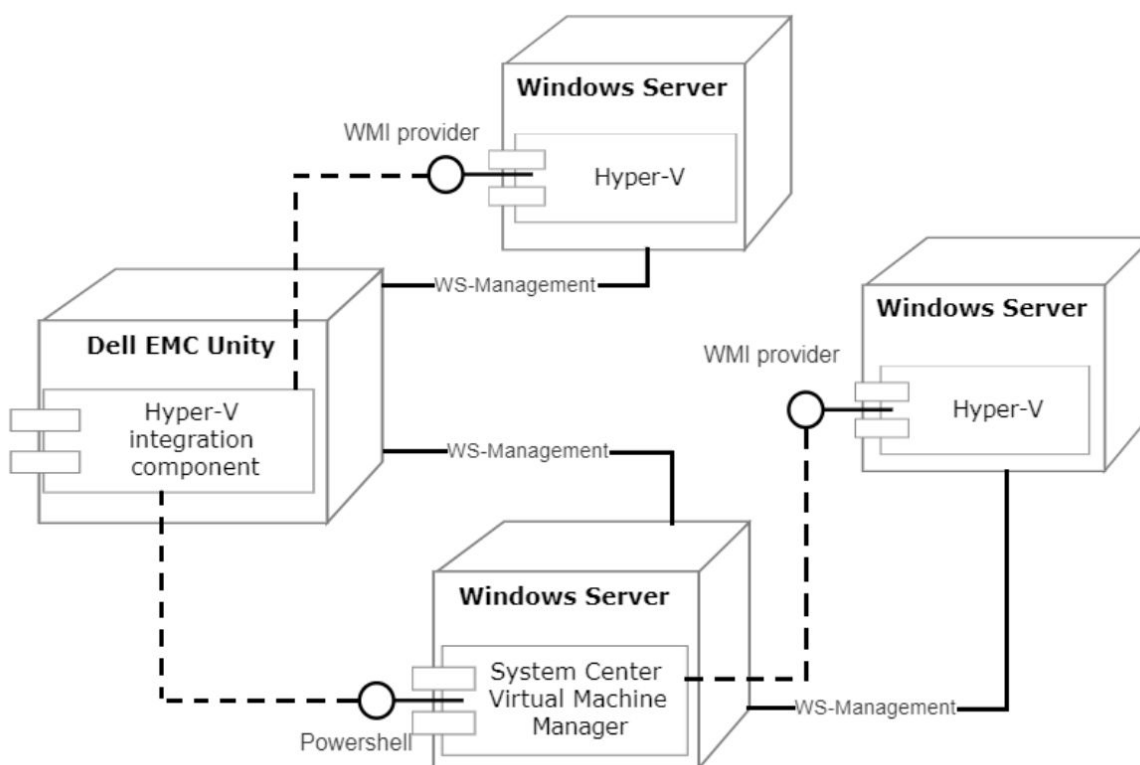


Рис. 3. Диаграмма компонентов системы виртуализации.

Для сетевого взаимодействия СХД с серверами под управлением ОС Windows и гипервизорами используется библиотека OpenWSMAN с открытым исходным кодом. Она посредством

протокола WS-Management взаимодействует с WMI провайдерами систем Windows, такими как система виртуализации Hyper-V и системы управления блочными и сетевыми дисками. Была выбрана библиотека OpenWSMAN, так как она поддерживает протокол сетевой аутентификации Керберос, который используется для аутентификации пользователей Windows Active Directory.

SCVMM, в отличие от большинства подсистем ОС семейства Windows Server, не имеет WMI интерфейса. Вместо этого, помимо графического интерфейса, он предоставляет команды языка сценариев Windows Powershell. Взаимодействие с SCVMM происходит путем вызова команд Powershell с СХД, что осуществляется посредством протокола WS-Management, используя библиотеку OpenWSMAN.

Как в случае взаимодействия с Hyper-V напрямую, как и с SCVMM, можно было бы использовать удаленный вызов Powershell команд. Это не было сделано, так как Powershell отправляет сообщения, закодированные в base64. Base64 — это метод кодирования, при котором каждые 3 исходных байта кодируются 4 символами (происходит увеличение длины сообщения на треть), что значимо при больших объемах данных.

При выполнении определенных операций, таких как запрос информации о всех виртуальных машинах на всех доступных серверах, объем полученной информации может превышать объем доступной оперативной памяти. Чтобы не превысить объем доступной оперативной памяти, система фрагментирует большие запросы и обрабатывает их отдельными частями.

## 4.2 Архитектура системы

На рисунке 4 представлена упрощенная диаграмма классов реализованной системы.

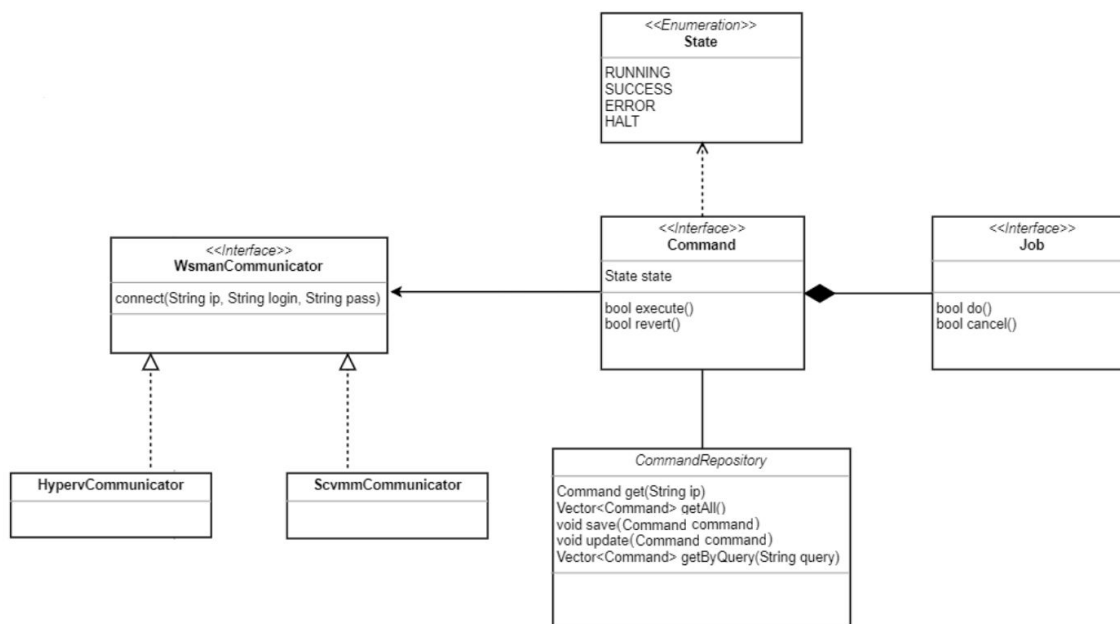


Рис. 4. Диаграмма классов реализованной системы.

Класс `Command` представляет отдельную операцию, выполняемую на сервере, такую как предоставление серверу блочного или файлового устройства, сохранение состояния виртуальной машины. Каждая команда может быть выполнена на сервере как напрямую, так и через предоставляемый интерфейс `SCVMM`. За взаимодействие СХД с сервером отвечает класс `WsmannCommunicator`. Объект класса команды при создании параметризуется объектом класса `HyperVCommunicator` или `ScvmmCommunicator`, которые наследуют от абстрактного класса

WsmnCommunicator и реализуют сетевое взаимодействие с сервером путем вызова WMI или Powershell команд соответственно.

Для исполнения одной команды может потребоваться выполнить несколько различных задач последовательно. Класс конкретной команды наследует от абстрактного класса Command и включает несколько задач, представленных классами, наследуемыми от абстрактного класса Job. Job реализует выполнение задачи и ее отмену. Класс команды исполняет задачи по очереди, с первой до последней. Если при выполнении какой-либо задачи возникает ошибка и система не может продолжить выполнение команды, то все предыдущие задачи отменяются в порядке, обратном их выполнению. Это представляет собой простейший конечный автомат. Такой подход обеспечивает транзакционность выполнения операций, так как в случае ошибки во время выполнения операции, система вернется в изначальное корректное состояние. Во время исполнения команды, она автоматически сохраняет свое состояние, используя класс репозитория. Также реализована функция отмены успешно завершенной команды. Кроме того, осуществляется журналирование всех выполняемых операций.

При успешном выполнении команда переходит в состояние SUCCESS. При возникновении ошибки команда отменяет выполненные задачи, после чего переходит в состояние ERROR. Если невозможно ни продолжение операции, ни ее отмена, то команда переходит в состояние HALT. Это позволяет корректно обрабатывать некоторые ошибки, такие как выключение системы, прерывание интернет соединения. После восстановления системы можно корректно завершить задачу, или отменить ее.

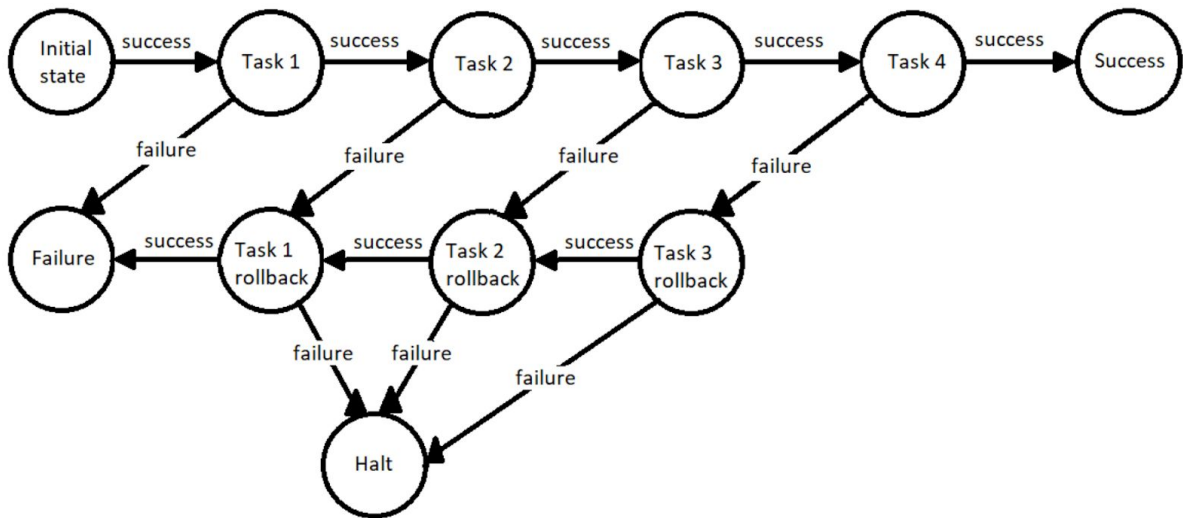


Рис. 5. Диаграмма состояний класса Command.

### 4.3 Реализованные операции

Реализованная система позволяет пользователям вызывать реализованные команды по взаимодействию СХД с сервером, используя интерфейс командной строки.

В рамках работы реализованы следующие команды.

Реализовано выполнение снимка состояния виртуальной машины. Снимок состояния виртуальной машины — это файловая копия состояния виртуальной машины, данных на диске и ее конфигурации. При необходимости, можно вернуться к предыдущему состоянию, сохраненному в снимке.

Реализована миграция виртуальных машин. Миграция виртуальных машин – это перенос процесса выполнения виртуальных машин между гипервизорами, запущенными на различных серверах. Миграция может выполняться для распределения нагрузки между гипервизорами.

Реализовано получение информации о серверах и виртуальных машинах, такой как состояние, информация о логических и сетевых дисках, включая свободный объем; IP адреса, WWN (World Wide Name), IQN (iSCSI Qualified Name).

Реализована функция добавления блочного устройства на сервер с использованием протоколов iSCSI и Fibre Channel.

Реализована функция добавления сетевого диска на сервер с использованием протоколов CIFS и NFS.

## **4.4 Тестирование**

В рамках работы были написаны автоматические модульные и интеграционные тесты.

Интеграционные тесты представляют собой сценарии на языке Python, которые вызывают заданные команды реализованной системы по взаимодействию с гипервизором. Затем, используя удаленные вызовы к гипервизору, в сценарии осуществляется проверка того, что тестируемые команды действительно выполнили предполагаемые действия.

# Заключение

В ходе выполнения данной работы были достигнуты следующие результаты.

- Проведен анализ требований к системе.
- Создана архитектура системы для управления взаимодействием СХД и гипервизора.
- Разработана система, реализующая взаимодействие СХД с сервером и гипервизором, основанная на протоколе WS-Management. Реализованы следующие операции:
  - предоставление блочного устройства и файлового хранилища;
  - миграция виртуальных машин;
  - получение информации о виртуальных машинах;
  - выполнение снимка состояния виртуальной машины.

# Список литературы

- [1] A Measurement Study of Server Utilization in Public Clouds. — URL: <https://ieeexplore.ieee.org/document/6118751/> (online; accessed: 29.05.2018).
- [2] Introduction to Hyper-V. — URL: <https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/about/> (online; accessed: 29.05.2018).
- [3] Dell EMC Unity, технические характеристики. — URL: <https://russia.emc.com/collateral/data-sheet/h14958-unity-hybrid-family-ss.pdf?isKoreaPage=false&domainUrlForCanonical=https%3A%2F%2Frussia.emc.com> (online; accessed: 29.05.2018).
- [4] Redundant Arrays of Inexpensive Disks (RAID) technology overview. — URL: <https://www2.eecs.berkeley.edu/Pubs/TechRpts/1987/CSD-87-391.pdf> (online; accessed: 29.05.2018).
- [5] Introduction to Storage Area Networks. — URL: <http://www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf> (online; accessed: 29.05.2018).
- [6] Y. Deng. Deconstructing Network Attached Storage Systems. Journal of Network and Computer Applications. Vol.32, No.5, 2009, pp.1064-1072.
- [7] RFC 3720: Internet Small Computer Systems Interface (iSCSI). — URL: <https://tools.ietf.org/html/rfc3720> (online; accessed: 29.05.2018).
- [8] RFC 2625: IP and ARP over Fibre Channel. — URL: <https://tools.ietf.org/html/rfc2625> (online; accessed: 29.05.2018).
- [9] ESXi overview. — URL: <https://www.vmware.com/products/esxi-and-esx.html> (online; accessed: 29.05.2018).



- [10] Windows Management Instrumentation overview — URL:  
[https://msdn.microsoft.com/en-us/library/windows/desktop/aa394582\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/aa394582(v=vs.85).aspx)  
(online; accessed: 29.05.2018).
- [11] Distributed Component Object Model overview. — URL:  
<https://technet.microsoft.com/en-us/library/cc958799.aspx> (online; accessed:  
29.05.2018).
- [12] Web Services Management, DMTF Specifications. — URL:  
<https://www.dmtf.org/standards/ws-man> (online; accessed: 29.05.2018).
- [13] Common Information Model, DMTF Specifications. — URL:  
<https://www.dmtf.org/standards/cim> (online; accessed: 29.05.2018).
- [14] Simple Object Access Protocol, specification. — URL:  
<https://www.w3.org/TR/2000/NOTE-SOAP-20000508/> (online; accessed:  
29.05.2018).
- [15] WQL Operators. — URL:  
[https://msdn.microsoft.com/ru-ru/library/aa394605\(v=vs.85\).aspx](https://msdn.microsoft.com/ru-ru/library/aa394605(v=vs.85).aspx) (online; accessed:  
29.05.2018).
- [16] Bash Reference Manual. — URL:  
<https://www.gnu.org/software/bash/manual/bashref.html> (online; accessed:  
29.05.2018).
- [17] Server Message Block (SMB) Protocol Specification. — URL:  
<https://msdn.microsoft.com/en-us/library/cc212363.aspx> (online; accessed:  
29.05.2018).
- [18] RFC 1094: Network File System Protocol Specification. — URL:  
<https://tools.ietf.org/html/rfc1094> (online; accessed: 29.05.2018).