Максименко Дмитрий Сергеевич

Реализация кластерного режима в SCSI target ядра Linux

Выпускная квалификационная работа

Научный руководитель:
д. ф.-м. н., доцент Д. В. Луцив

Рецензент:
Ведущий разработчик «Yadro»
Р. О. Большаков

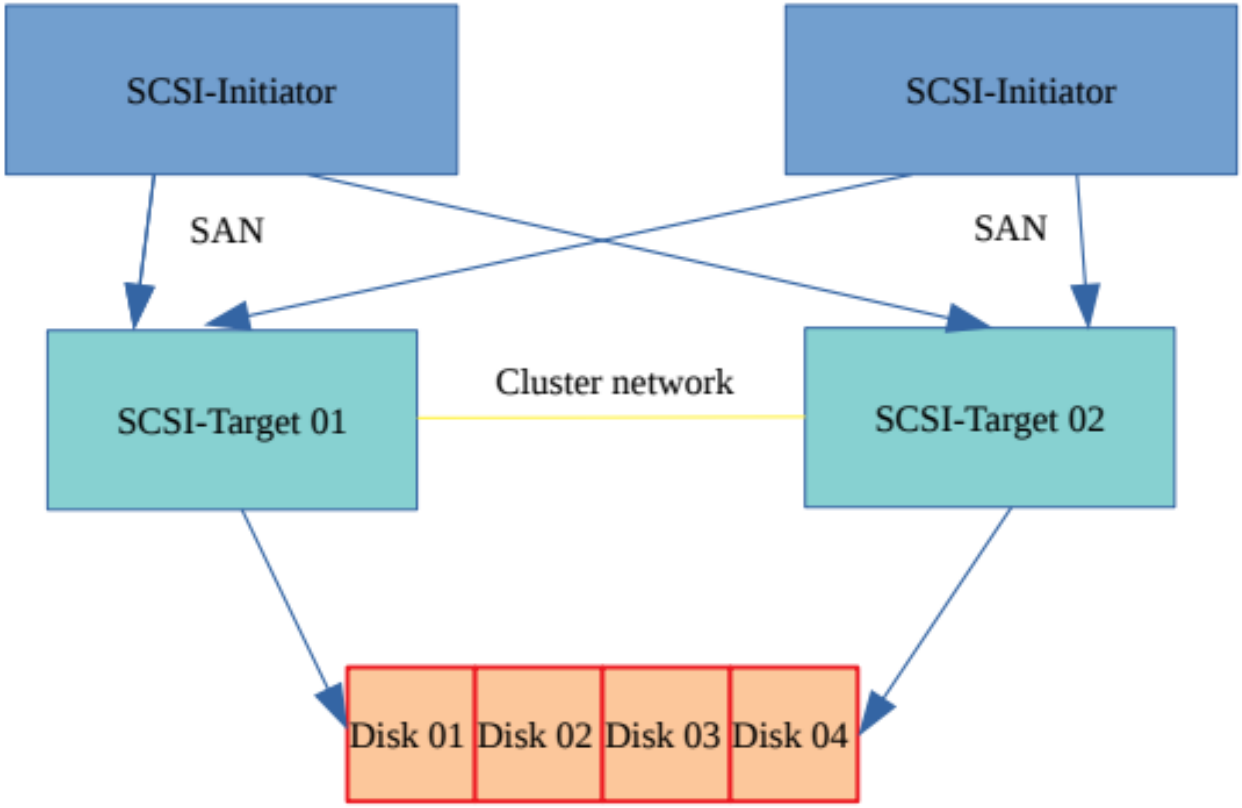
14.06.2021

Введение в предметную область

С 2011 года в ядре Linux SCSI-target подсистемой является TCM (Target Core Module)

TCM не поддерживает кластерный режим в SCSI

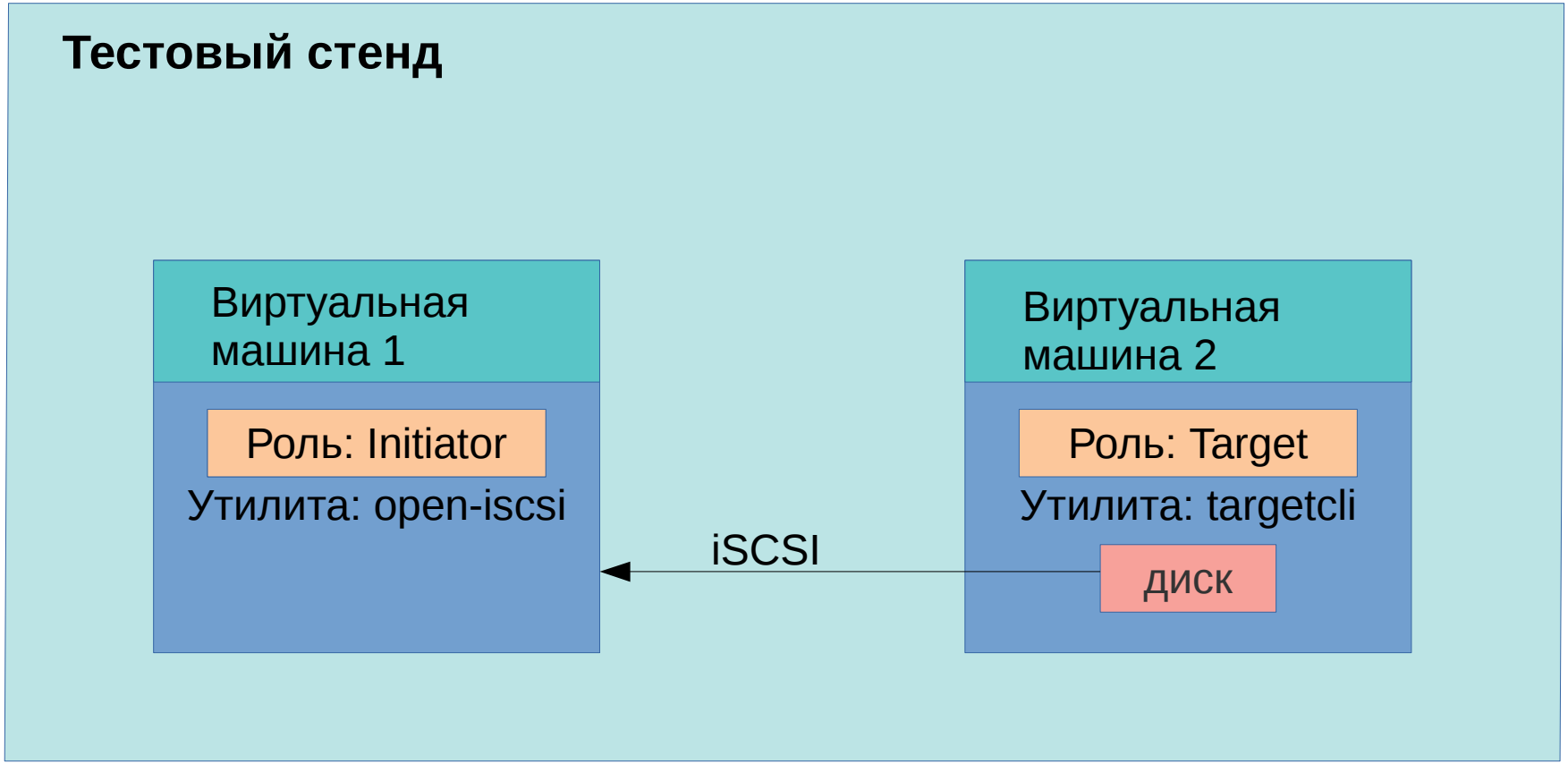
Кластерный режим нужен компаниям, разрабатывающим системы хранения данных



Постановка задачи

- Провести исследование предметной области, изучить существующие стандарты
- Произвести обзор решений, которые позволяют организовать SCSI кластер и зафиксировать лучшие аспекты этих решений
- Разработать метод реализации кластерного режима в SCSI-target ядра Linux
- Реализовать разработанный метод и добавить с помощью него кластерный режим в некоторые команды SCSI
- Провести тестирование ключевой функциональности с помощью тестовых стендов

SCSI взаимодействие



Стандарты: SCSI-3, SPC-4, SBC-3

Обзор аналогов

	SCST	Моё решение
Решает задачу кластеризации SCSI	Да	Да
Работает, как модуль ядра	Да	Да
Средство синхронизации	DLM	DLM
Является встраиваемым в ядро Linux решением	Нет	Да

Средство синхронизации

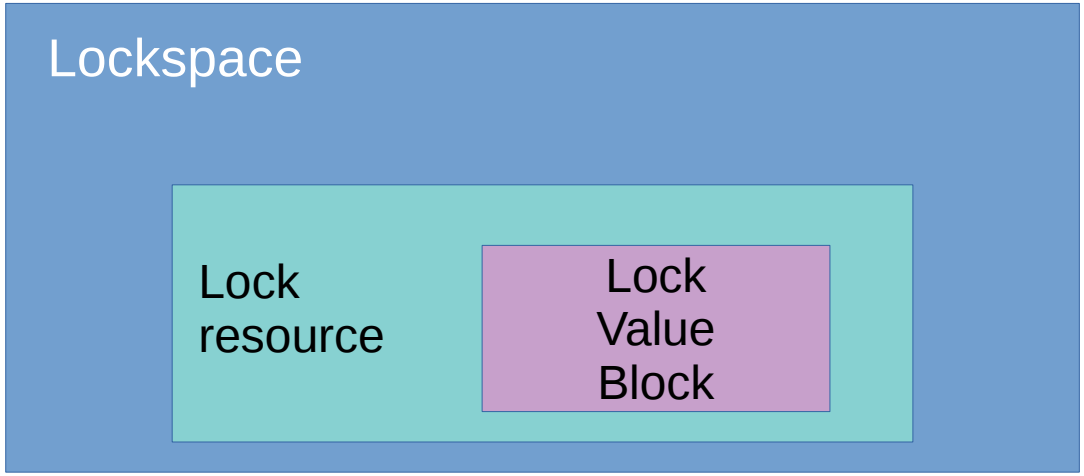
- создание и конфигурация кластера
- контроль поведения кластера

Pacemaker/Corosync

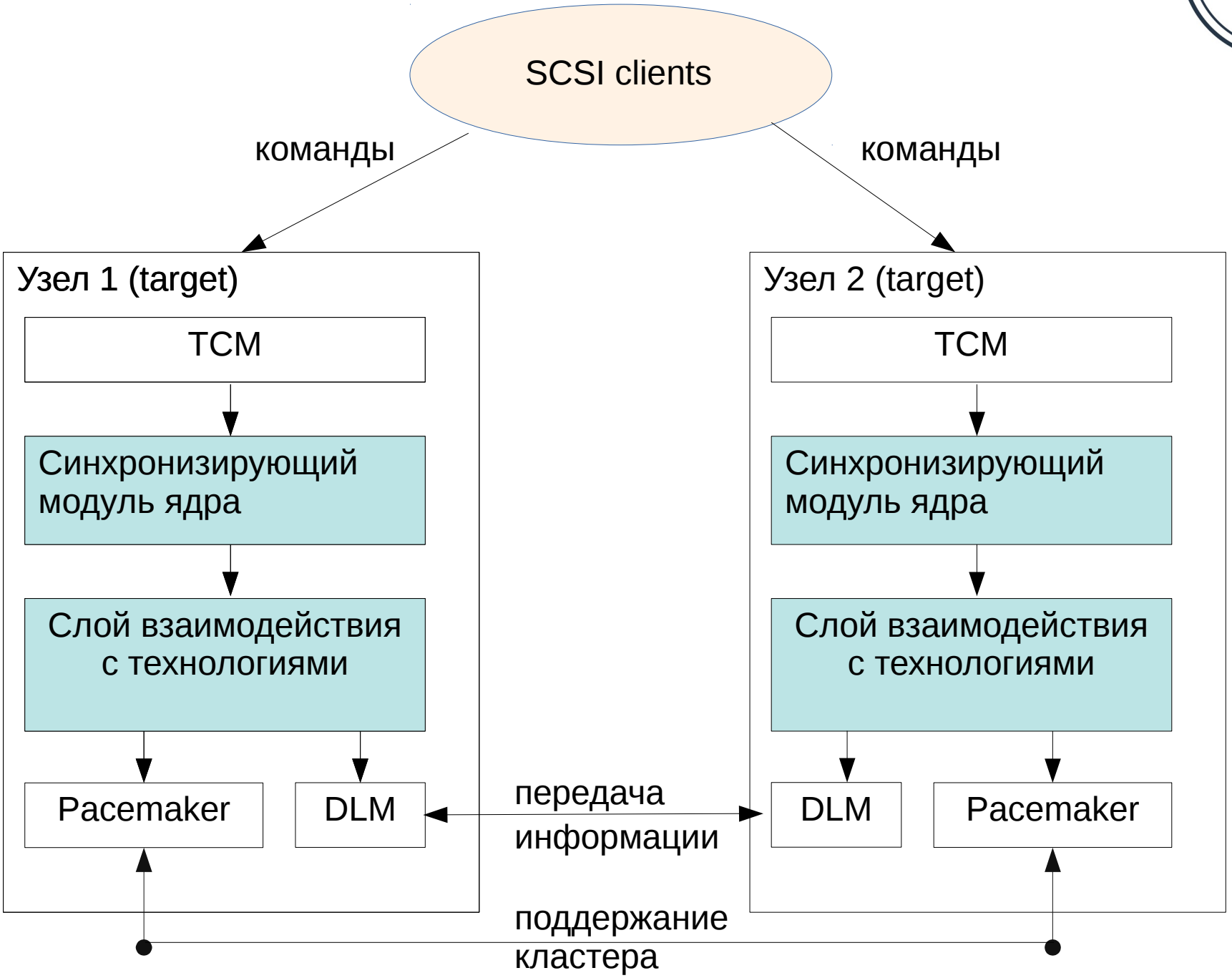
- передача информации между узлами
- корректная работа в конкурентной среде
- наличие в ядре Linux

DLM (Distributed Lock Manager)

DLM сущности:



Общая архитектура



Синхронизирующий модуль ядра

Использует DLM для построения распределенной базы данных

Два подхода:

Запрашивать актуальную информацию с других узлов при чтении состояния	Обновлять все узлы при изменении состояния
Эффективен при редких чтениях состояния	Эффективен при частых чтениях состояния

Распространение обновлений

У блокировок DLM 6 режимов: NL, CR, CW, PR, PW, EX

NL → EX	значение читается
EX → NL	значение пишется

С блокировками DLM можно связать два обратных вызова

AST (Asynchronous System Trap)
вызывается при взятии блокировки

BAST (Blocking AST)
вызывается, когда данная блокировка препятствует взятию другой блокировки

Реализация команды «CAW»

1) Создать ресурс X в синхронизирующем модуле с ключом LBA

2) Взять блокировку на ресурс X

3) Прочитать значение по адресу LOGICAL BLOCK ADDRESS (LBA)

4) Сравнить это значение со значением в буфере команды

5) В случае совпадения: происходит запись специального значения из буфера в LBA

6) В случае несовпадения: команда прекращается с соответствующим статусом

7) Снять блокировку с ресурса X

**Реализовано в ядре Linux 5.8.0-50-generic*

Слой взаимодействия

Драйвер символьного устройства

1) запись (echo >) : исполнение команд
2) чтение (cat) : возвращение состояния

Команды

1) add-key <key> <value> : создание или модернизация ключа
2) remove-key <key> : удаление ключа
3) lock-key <key> : блокировка ключа
4) unlock-key <key> : снятие блокировки

Тестирование

1) Виртуальные машины создавались с использованием VirtualBox и QEMU

2) Кластеры создавались с использованием Pacemaker

3) Тестирование синхронизирующего модуля и драйвера символьного устройства выполнять с использованием фреймворка PyTest

4) Для тестирования команды «Compare And Write» использовался набор утилит sg3_utils

Итог

- Проведено исследование предметной области, изучены существующие стандарты
- Произведен обзор решений, которые позволяют организовать SCSI кластер и зафиксированы лучшие аспекты этих решений
- Разработан метод реализации кластерного механизма в SCSI-target ядра Linux
- Реализован разработанный метод и с помощью него добавлен кластерный режим в команду SCSI «Compare And Write»
- Проведено тестирование ключевой функциональности с помощью тестовых стендов