

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ”

ОСНОВНЫЕ РЕЗУЛЬТАТЫ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

«ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА
ИНТЕГРИРОВАННОГО РЕШЕНИЯ ПО КОНСОЛИДАЦИИ
РЕСУРСОВ СХД СЕРИИ EMC VNX»

Автор Трофимов Владислав Александрович _____
(Фамилия, Имя, Отчество) (Подпись)

Направление подготовки 09.04.02 Разработка корпоративных
информационных систем

Квалификация Магистр _____
(бакалавр, инженер, магистр)

Руководитель Маятин А.В., доцент, к.п.н. _____
(Фамилия, И., О., учёное звание, степень) (Подпись)

К защите допустить

Зав. кафедрой ИС Парфёнов В.Г., проф., д.т.н. _____
(Фамилия, И., О., учёное звание, степень) (Подпись)

“ ” _____ 2018 г.

Санкт-Петербург, 2018 г.

СОДЕРЖАНИЕ

1.	Обзор предметной области и автоматизируемого процесса	3
2.	Специфика ВКР	4
3.	Результаты ВКР	5

1. Обзор предметной области и автоматизируемого процесса

В данной ВКР решается проблема интеграции 3 программных продуктов для автоматизации прикладного бизнес-процесса с учетом требования к высокой отказоустойчивости системы при наличии ненадежных каналов связи и надежных узлов обработки данных.

Предметной областью автоматизируемого процесса являются системы хранения данных (далее - *СХД*). Основным ресурсом *СХД* является дисковое пространство. Дисковое пространство объединяется в *RAID*-массивы. Однотипные *RAID* массивы объединяются в пулы, называемые Storage Pool (далее - *SP*). Примеры *SP* приведены на рисунке 1.

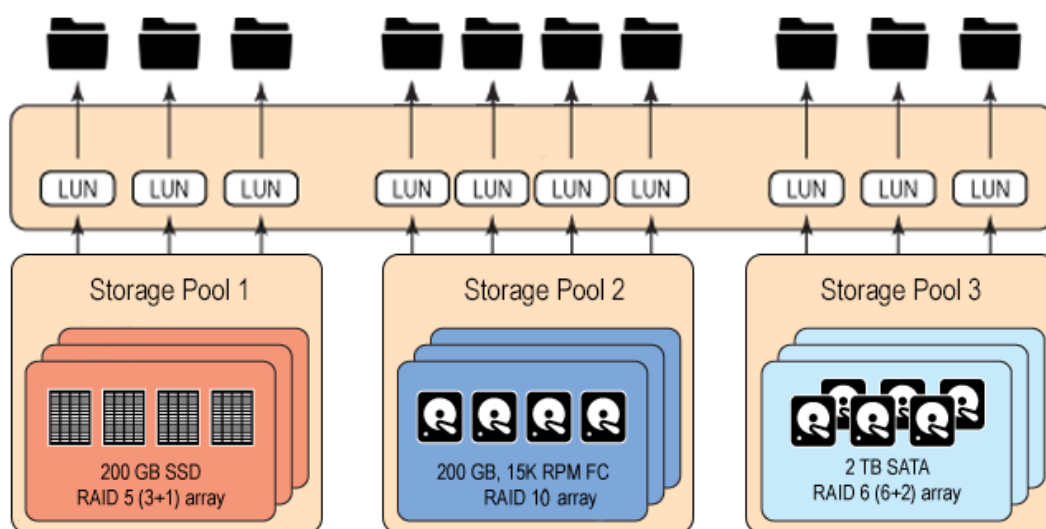


Рисунок 1 – Различные *SP*

Различные конфигурации *RAID*-массивов обладают разными показателями производительности при различных типах подаваемой на них нагрузки, а также различными показателями надежности, доступных протоколов подключения и др.

Для логической консолидации *SP* по одному или нескольким вышеописанным показателям существует концепция Virtual Pool (далее - *VP*). Примеры *VP* представлены на рисунке 2.

Продукт ViPR Controller предоставляется REST API для управления *SP* и *VP* на заранее сконфигурированных *СХД*. Конфигурирование *СХД* не является темой, рассматриваемой в данной работе.

Данный продукт не имеет возможности группировать *SP* по динамическим показателям производительности (в данной работе рассматривается время отклика). Задачей проектируемой системы является создание *VP* на основании *SP*, удовлетворяющих времени отклика под заданной пользователем нагрузкой, а также последующее выделение *LUN* на таком *VP*.

Второй интегрируемый программный продукт – ViPR SRM – предоставляет информацию о динамических показателях производительности *SP*.

Третий интегрируемый продукт – VNX Sizer – позволяет рассчитать характеристики *SP* под заданной нагрузкой используемых его приложений.

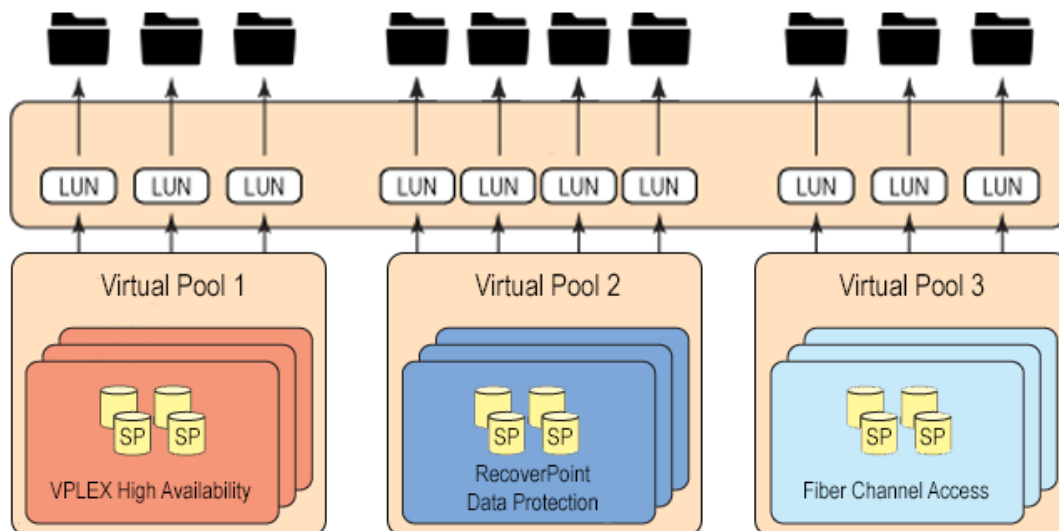


Рисунок 2 – Различные конфигурации *VP*

2. Специфика ВКР

Прототип данной системы, позволяющий только создавать требуемые *VP*, был создан автором в рамках бакалаврской ВКР. Данный прототип представляет собой однопользовательское веб-приложение с одним вычислительным узлом.

В рамках данной ВКР была спроектирована и разработана система, реализующая, помимо создания *VP*, все функциональные варианты использования, описанные в п.1. Отличительным требованием, выдвигаемым к данной системе, является возможность объединения копий системы, запущенных на вычислительных узлах в разных ЦОД. Спроектированная система является высоко надежной при наличии ненадежных каналов связи между узлами приложения и надежными вычислительными узлами. Спроектированная система относится к классу распределенных систем «узел-узел», что позволяет осуществлять управление *VP* сразу со всех доступных ЦОД. Данная система подразумевает использование в многопользовательском режиме. Требования к высокой надежности обусловлено тем, что ЦОД являются надежными, а каналы связи, соединяющие их – нет. В связи с этим администраторы должны иметь возможность конфигурировать *VP* на доступных на данный момент ЦОД посредством веб-интерфейса текущего узла приложения, а после восстановления канала связи все изменения, сделанные на изолированном узле, должны стать доступны остальным пользователям системы. Для оценки качества надежности

используются различные метрики, связанные со временем восстановления информации с изолированного узла при обрыве и восстановлении канала связи, при наличии нестабильного соединения с различными показателями задержки сигнала и потери пакетов в сети.

3. Результаты ВКР

В ходе выполнения ВКР был проведен глубокий анализ различных подходов, используемых для организации высоко надежной системы при заданных условиях, а также детальное рассмотрение возможностей, предоставляемыми уже используемыми программными продуктами, по организации высоко надежной системы. В рассмотрение также были взяты и другие современные технологии, и программные продукты, которые могут решить поставленную задачу, проведен их сравнительный анализ, обоснован выбор и определены методы оценки отказоустойчивости разработанной системы.

Для организации высокой отказоустойчивости в рамках имеющегося программного стека технологий была выбрана СУБД Hazelcast, осуществляющая хранение данных в оперативной памяти.

Имеющаяся программная архитектура приложения была подвергнута значительным изменениям, обусловленным концептуально новыми требованиями к ее использованию. При этом было осуществлено эффективное переиспользование программного кода и компонент, реализованных в рамках бакалаврской ВКР. Удачный выбор стека технологий и программной архитектуры в рамках бакалаврской ВКР позволил без значительных изменений системной архитектуры и архитектуры данных реализовать все требуемые варианты использования и минимально затрагивать имеющиеся компоненты при проектировании отказоустойчивой программной архитектуры, осуществляя в большей степени добавление новых компонент в имеющуюся программную архитектуру и интеграцию их высокоуровневых вызовов в имеющуюся.

Для реализации новых функциональных требований также потребовалось добавление новых страниц в имеющийся веб-интерфейс.

Для удобства демонстрации для каждого из компонент приложения был сконфигурирован docker-контейнер, и осуществлено конфигурирование тестового кластера из 3 узлов для демонстрации ключевых функциональных особенностей системы.

Так как интегрируемые компоненты являются зависимыми от СХД, то для удобства демонстрации программного комплекса заказчикам были также реализованы системные компоненты, эмулирующие работу интегрируемых программных продуктов, реализующих эквивалентный API.