Министерство образования и науки Российской Федерации

САНКТ–ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,   
МЕХАНИКИ И ОПТИКИ

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРОГРАММИРОВАНИЯ

КАФЕДРА ИНФОРМАЦИОННЫХ СИСТЕМ

ПРЕДВАРИТЕЛЬНАЯ АННОТАЦИЯ К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

на тему:

«ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ПРОТОТИПА ИНТЕГРИРОВАННОГО РЕШЕНИЯ ПО КОНСОЛИДАЦИИ РЕСУРСОВ СХД СЕРИИ EMC2 VNX НА ОСНОВЕ СУЩЕСТВУЮЩИХ КОМПОНЕНТОВ»

Исполнитель:

Студент 4 курса направления

Информационные системы и технологии

очной формы обучения

Трофимов Владислав Александрович

Консультант:

Инженер-программист компании EMC2

специалист, аспирант

Одеров Роман Сергеевич

Санкт–Петербург

2015

## Цель ВКР

Целью данной ВКР является проектирование и разработка прототипа интегрированного решения в виде Веб-приложения с пользовательским интерфейсом, которое автоматизирует процесс консолидации ресурсов блокового доступа СХД серии EMC2 VNX на основе требований приложений к параметрам производительности используемых ими ресурсов блокового доступа СХД.

## Описание процесса автоматизации

Задачей автоматизации является интеграция 3 имеющихся программных продуктов для осуществления автоматизация подбора наиболее подходящих *Storage Pool* под требования приложения, для которого необходимо выделить *LUN*. Подобранные пулы необходимо агрегировать в *Virtual Pool*.

Компания EMC2 предоставляет 3 программных продукта:

* EMC2 ViPR – WEB-приложение для централизованного управления большим количеством систем хранения данных. Привносит уровень абстракции над ресурсами СХД: *Virtual Pool.* Имеет REST API для взаимодействия.
* EMC2 SRM – WEB-приложение для мониторинга и анализа использования ресурсов СХД в реальном времени. Имеет SOAP API для взаимодействия.
* EMC2 Sizer – консольная утилита для расчета характеристик СХД под определенной нагрузкой. Также имеется версия с графической оболочкой для интерактивных расчетов.

На данный момент данные программные продукты не интегрированы посредством какого-либо другого программного продукта EMC2.

На данный момент концепция *Virtual Pool* в ViPR не поддерживает консолидацию по параметрам производительности. Также ViPR не отслеживает нагрузку, оказываемую на *Storage Pools*, входящие в состав *Virtual Pool*. Тем самым, при выделении *LUN*, он может оказаться расположенным на сильно загруженном *Storage Pool*.

На данный момент задача группировки подходящих по показателям производительности *Storage Pool* решается вручную. Для этого администратор с помощью веб-интерфейса продукта SRM просматривает информацию о том, как сильно нагружены *Storage Pools*. Данная информация представлена в интерфейсе в виде графиков и таблиц.

Подобрав «на глаз» менее загруженные пулы с типом дисков и уровнем RAID, которые теоретически подойдут для конечного приложения, администратор вводит информацию о приложении и о характеристиках выбранного пула вручную в интерфейс утилиты VNX Sizer, после чего вручную проверяет, какие из отобранных пулов после применения на них дополнительной нагрузки будут удовлетворять требуемым параметрам производительности.

После окончательного отбора *Storage Pools*, администратор вручную добавляет каждый из них в новый *Virtual Pool* через веб-интерфейс продукта ViPR.

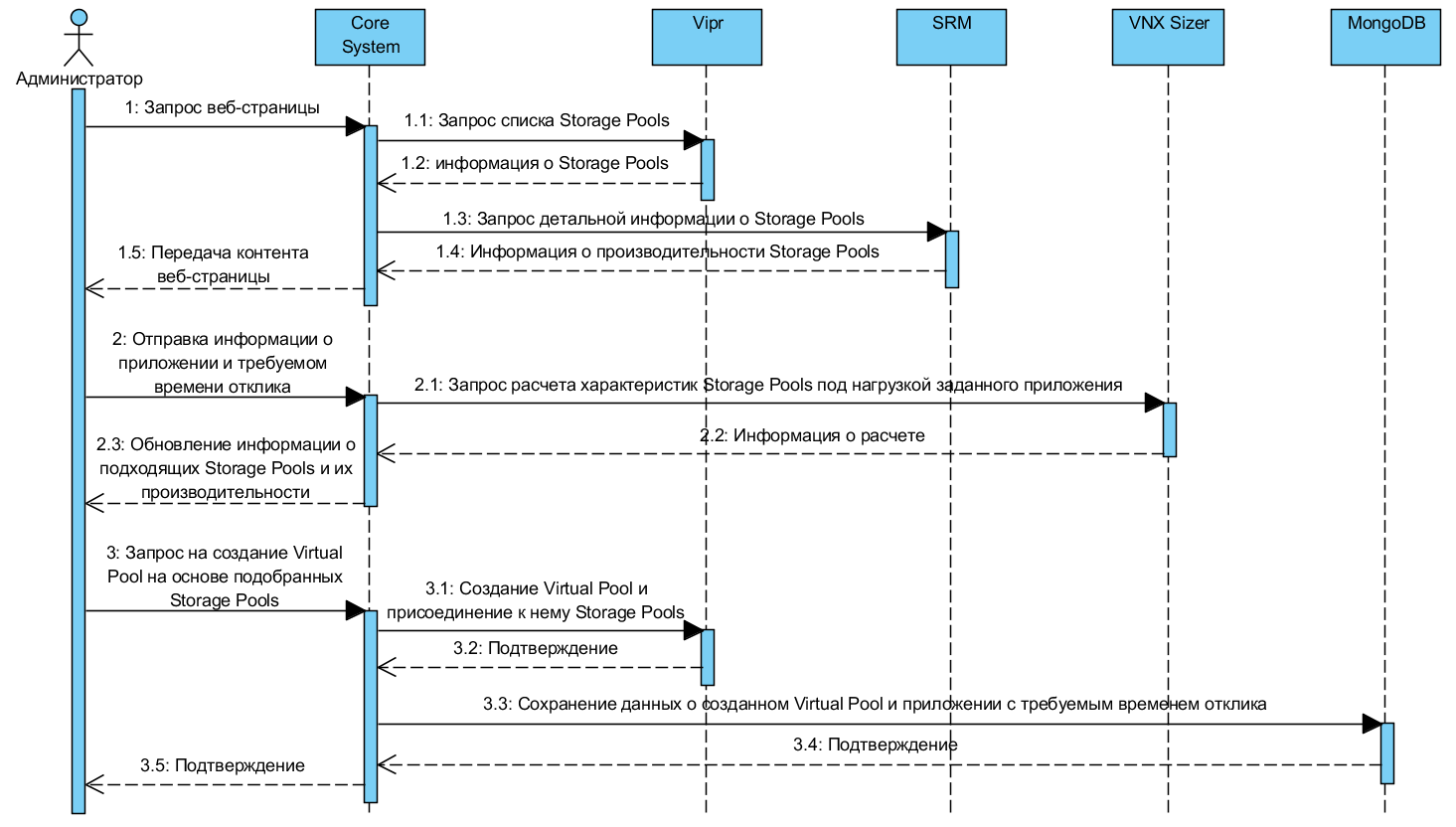
Для данного процесса предлагается следующая автоматизация:

Создание стороннего по отношению к используемым продуктам веб-приложения, которое позволит администратору не производить какие-либо оценки и перенос данных о *Storage Pools* вручную, а централизованно предоставить информацию о том, какие *Storage Pools* будут удовлетворять требованиям приложения, которое администратор хочет разместить на СХД, а также автоматически создавать *Virtual Pool* под приложения с заданными требованиями к производительности.

Пример описания требований приложения: необходим *Virtual Pool* под OLTP базу данных, в которой объем таблиц будет порядка 2ТБ, с заданной верхней границей времени выполнения транзакции 50мс, количеством пользователей 200 и временем отклика диска 10мс.

## 

1. Диаграмма последовательности для ручного процесса



2. Диаграмма последовательности для автоматизированного процесса

## Объем информации для выполнения ВКР

Данный проект в компании является прототипом. Четко сформулированное ТЗ по данному процессу отсутствует. Система должна быть итеративно доведена до состояния, при котором она успешно работает на основном сценарии использования.

## Функциональные требования к разрабатываемой системе

При открытии главной страницы приложения пользователю должна быть предоставлена информация о существующих *Storage Pools*, о показателях их загруженности и характеристиках. На этой же странице у пользователя должна быть возможность ввести информацию о требованиях приложения в заранее заданных форматах (для прототипа необходимо поддержать форматы Oracle OLTP и Free Form).

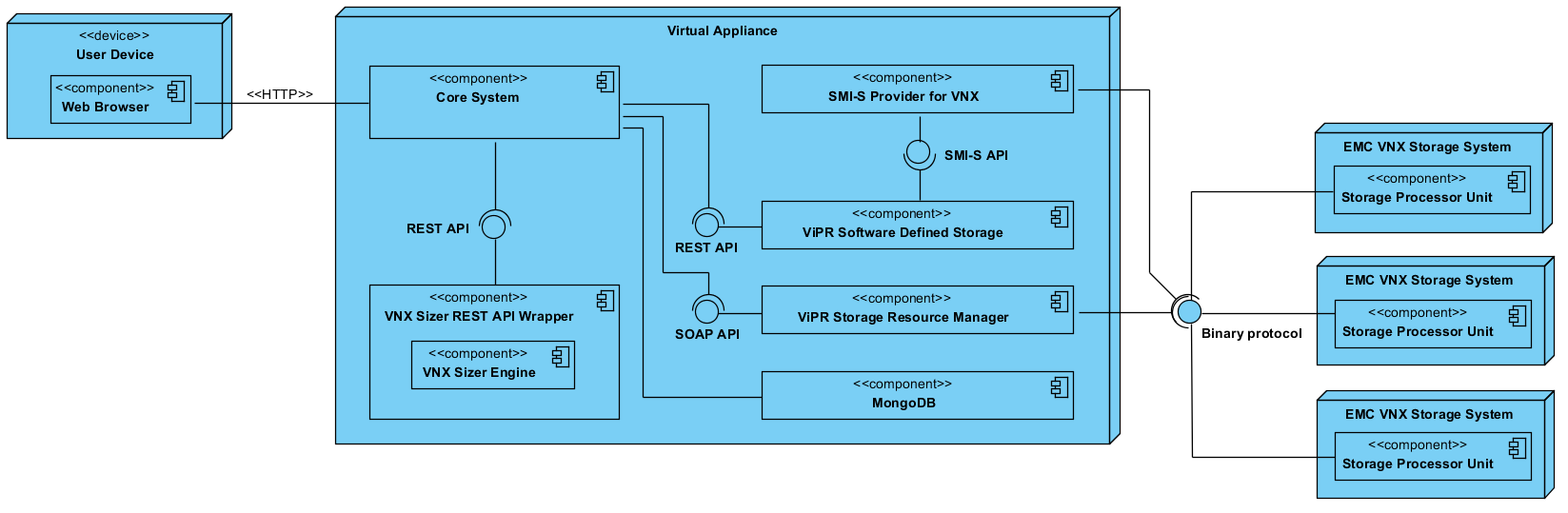
Информация о СХД, список *Storage Pools* и их статические характеристики должны загружаться из ViPR через REST API. Информация об их загруженности должна поступать из SRM по SOAP API. На данном шаге необходимо преобразование полученных из SRM данных, а также валидация того, что и ViPR, и SRM обслуживают одно и то же множество *Storage Pools*. Если валидация неудачная, или в процессе получения информация возникает ошибка, информацию о ней необходимо отобразить пользователю на странице.

После ввода пользователем информации о требованиях приложения, необходимо как на стороне клиента, так и на стороне сервера произвести проверку корректности введенных данных. После этого необходимо сгруппировать эти данные с информацией о СХД, информацией о списке *Storage Pools* и преобразовать в формат, который будет совместим с разработанным REST API утилиты Sizer.

Для расчета характеристик *Storage Pool* с конкретной конфигурацией под конкретной нагрузкой информация должна быть направлена в утилиту Sizer. Для нее необходимо разработать REST API обертку, которая позволит в будущем использовать ее как сервис на облачных платформах (н-р, Pivotal CloudFoundry). Обертка должна обеспечить прием и передачу параметров к утилите и от нее обратно на сервер. После осуществления расчетов утилитой полученные данные должны быть направлены приложению. Приложение должно проверить, что полученные показатели удовлетворяют требованиям приложения.

В конечном счете, для каждого *Storage Pool* из отображенных пользователю должна быть предоставлена информация о том, возможно ли размещение нагрузки на данный пул, и если да, то какие при этом будут параметры производительности у данного пула.

Далее, пользователь отмечает несколько (или все) из предложенных *Storage Pool* и через тот же интерфейс инициирует процесс создания *Virtual Pool* из выбранных *Storage Pool*. Для этого приложение должно делегировать создание пула продукту VIRP через REST API.



2. Диаграмма развертывания

## Используемые технологии

* Главный компонент и обертка для утилиты EMC Sizer должны быть спроектированы с учетом необходимости последующего развертывания разработанного прототипа в облачный контейнер Pivotal CloudFoundry
* Внешние компоненты: EMC ViPR, EMC ViPR SRM, EMC Sizer

## Тестирование и качество

Так как проектируемое решение не является решением продуктового уровня, наличие тестов не является обязательным. Однако, архитектура основного компонента и оболочки для утилиты Sizer должны гибко поддерживать последующее их покрытие тестами.

## Результаты

В результате выполнения ВКР должна быть получена информационная система, работоспособность которой сможет быть продемонстрирована на сценарии при использовании реальных данных на реальной СХД. Ввиду небольшого количества ресурсов СХД, должен быть предоставлен базовый сценарий, в котором будет осуществляться подбор *Storage Pools* по параметру «время отклика». Также в результате должны быть представлены UML диаграммы вариантов использования, действия, классов, последовательности и развертывания, которые должны описывать основной сценарий работы системы, описанный функциональных требованиях.

## Приложение

## Терминологическое описание

Современные СХД используются для хранения данных разнообразных Enterprise приложений – OLTP и OLAP баз данных (Oracle, MS-SQL), почтовых серверов (MS Exchange), файловых хранилищ (Samba), виртуальных рабочих станций (VMware, Citrix Xen). Каждому типу приложений требуются ресурсы СХД с разным уровнем производительности (время отклика, пропускная способность) и надежности хранения (поддержка репликации и/или бэкапов).

Система хранения данных для администратора представляет собой большой набор разнообразных дисков (SATA, NL-SAS, SAS, SSD). В рамках данной ВКР диски уже сгруппированы в RAID-массивы различных уровней, а однотипные RAID-массивы сгруппированы в *Storage Pools*. Также считается, что репликация и/или резервное копирование уже настроены (при их наличии).

*Storage Pool* представляет собой уровень абстракции над RAID-массивами. Он позволяет выделять блоковое пространство необходимого размера с заданным типом RAID прозрачно для пользователя, автоматически распределяя выделенное пространство по RAID-массивам данного типа. Также это позволяет выделять блоковое пространство с размером, превосходящим размер одного RAID-массива.

*LUN (Logical Unit Number)* – термин, используемый далее для обозначения выделенного блокового пространства фиксированного размера на *Storage Pool*. Это пространство в дальнейшем может быть сопоставлено с некоторым сервером или рабочей станцией (как физической, так и виртуальной), используя один из поддерживаемых сетевых протоколов (например, iSCSI или Fiber Channel). *LUN*, в зависимости от того, на каком *Storage Pool* он выделен, будет обладать теми или иными показателями производительности (по времени отклика, пропускной способности, надежности хранения). Эти показатели зависят не только от типа RAID и количества дисков, входящих в *Storage Pool*, но также от того, какую нагрузку создают приложения, использующие *LUNs*, выделенные на том же *Storage Pool*. Именно *LUN* можно назвать основным ресурсом СХД, которым пользуются приложения.

До появления СХД, способных автоматически определять, на каких *Storage Pools* необходимо выделять *LUN* для обеспечения требуемого уровня производительности, администраторам приходилось вручную (или с помощью вспомогательных утилит) рассчитывать, какую нагрузку будет генерировать то или иное приложение (например, какую нагрузку будет генерировать OLTP база данных, в которой объем таблиц составляет порядка 2ТБ, с заданной верхней границей времени выполнения транзакции с учетом дополнительного пространства под логи, бэкапы и кэш базы), и подбирать удовлетворяющие этим требованиям *Storage Pool*, и выделять на нем *LUN* требуемого размера.

Задача также усложняется тем, что приложения, для которых ранее были выделены *LUN*, далеко не всегда ведут себя именно так, как прогнозировалось – приложения, которые должны генерировать большое количество нагрузки, могут простаивать, и наоборот – легковесные приложения могут начать внезапно создавать большую нагрузку на используемый *Storage Pool*, отчего может страдать производительность приложений, для которых выделены *LUN* на том же *Storage Pool*.

Современные СХД имеют механизм, который позволяет избавиться от этих проблем путем анализа поведения приложений и автоматического переноса наиболее сильно нагруженных приложений в кэш, который состоит из высокопроизводительных дисков. Имея этот механизм, при выделении *LUN* достаточно указать требуемое время отклика, и в дальнейшем СХД будет пытаться автоматически удовлетворить требуемые характеристики для каждого приложения с учетом их поведения с помощью кэша.

Для более удобного и автоматизированного выделения ресурсов СХД компанией EMC2 был разработан продукт ViPR, который позволяет осуществлять централизованное и унифицированное управление ресурсами большого количества СХД одновременно. СХД подключаются к ViPR, который автоматически собирает с них всю информацию о существующих *Storage Pool* и *LUN*. Также ViPR привносит новый уровень абстракции над *Storage Pool – Virtual Pool.*

*Virtual Pool* –уровень абстракции, позволяющий консолидировать несколько *Storage Pool* в единое пространство для выделения *LUN*. При этом консолидирование может происходить по любым характеристикам (например, по уровню RAID, типу дисков, типу репликации, типу резервного копирования). Консолидирование может осуществляться сразу по нескольким характеристикам – например, «Виртуальный пул с SSD дисками на основе RAID 10 и синхронной репликацией».

Концепция *Virtual Pool* очень удобна в случае, когда у приложения есть определенное требование к какому-либо параметру (например, репликации), и нет требований к другим характеристикам. В таком случае администратору не придется вручную искать *Storage Pool*, который поддерживает данную репликацию – он просто выделит *LUN,* используя ранее созданный *Virtual Pool*, при этом *Storage Pool* будет выбран автоматически.