сырой материал

CloudStorageReferenceModel

При обсуждении cloud хранения и стандартов,важно различать различные ресурсы, которые предлагается в качестве услуги. Эти ресурсы предоставляются клиентам в качестве функциональных интерфейсов (Data Path) и интерфейсов управления (Control Path). Мы изучили различные типы интерфейсов, которые предоставляются на сегодняшний день и показали, как они связаны между собой. Мы предложили модель интерфейсов, которые могут быть отображены с различными приложениями, а также служить основой для разнообразных интерфейсов cloud хранения в будущем. Другая важная концепция рассматриваемая в этой статье-это метаданные. При управлении большими объемами данных с различными требованиями, метаданные-удобный механизм, который удовлетворяет требованиям, что при передаче данных может дифференцироваться обработка данных.

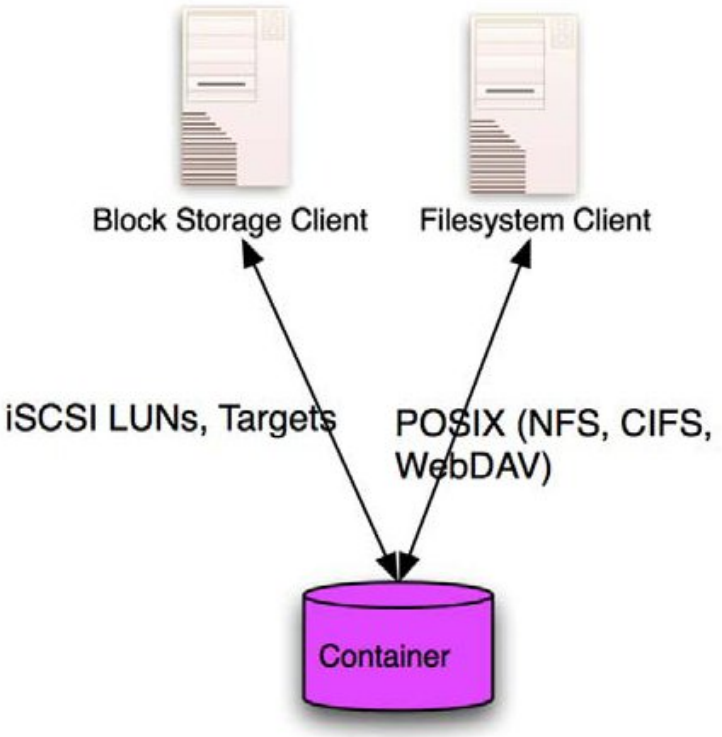
Привлекательность cloud хранения связано с некоторыми из тех атрибутов, которые определяют другие cloud услуги: pay as go,иллюзию бесконечной мощности (упругость), а также Простота использования / управления. Важно, чтобы любой интерфейс поддерживал эти атрибуты.

Что такое cloud storage?

Термин «Облако» используется как метафора, основанная на изображении интернета на диаграмме компьютерной сети. Эта абстракция имеет свои плюсы и минусы.Плюс в том, что прежде облако представляет собой множество ресурсов, независимо от типа. Важной частью облачной модели является концепция пула ресурсов, котороая расчитана на постепенное целевое приращение ресурсов (что по стоимости меньше закупки нового оборудования). Последним, что сделало это возможным является виртуализация.Таким образом cloud storage просто предоставляет виртуальные ресурсы по требованию. В терминах cloud это называется Data Storage as a Service (DaaS).

DAAS

Абстрагируется от хранения данных набором сервисных интерфейсов и предоставляют их по требованию. Единственный тип хранения, который не поддерживают эти интерфейсы-это хранение данных, не основанное на требовании, но требующее приращения мощностей. Важной частью любой слуги DaaS является поддержка уннаследованных клиентов. Это размещение с существующими стандартными протоколами, такими как ISCSI для блоков и CIFS / NFS или WebDAV для хранения сетевых файлов, как показано ниже:

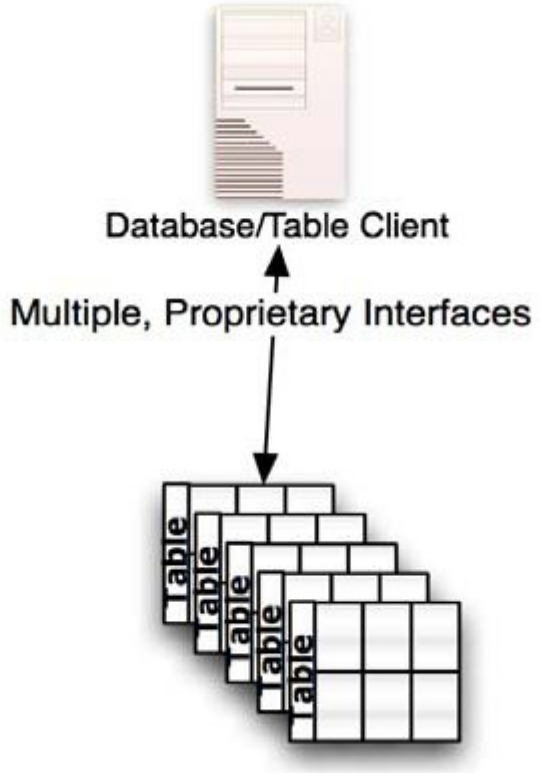


В случае с блоком хранения, LUN или виртуальный объем- это детализация распределения. Для файловых протоколов, файловая система это единица детализации. В любом случае фактическое место для хранения является тонким и его выделение предусмотрено на основе фактического использования. Услуги, такие как сжатие и дедупликация используются для последующего уменьшения фактического потребляемого пространства. Управление хранением, как правило, производится с помощью этих стандартных интерфейсов системы хранения данных, либо через API, или более общих инструментов через административный браузер с использованием пользовательского интерфейса. Этот интерфейс может быть использована для вызова других услуг передачи данных таких как snapshot и cloning.

в этой модели мы представляем основные интерфейсы использую понятие контейнер. Контейнер является не только полезной абстракцей в пространствах хранения, но так же служит средством группировки хранищихся в нем данных и является точкой контроля при применения общих сервисов к этим данным в совокупности. Другой тип предлагаемый DAAS - это одна простая таблица пространства хранения, это позволило производить горизонтальное масштабирование БД с помощью операций, необходимых определенным приложениям.Вместо того, чтобы виртуализовать случаи экземпляры реляционных БД, эти приложения предлагают новый интерфейс хранения данных с ограниченной функциональностью с акцентом на масштабируемость, а не на функциональность.

Это позволяет таблицам, раздеяемым между несколькими узлами на основе общего ключа значений, при наличии горизонтального масштабирования получить функциональность путем вертикального масштабирования реляционных БД. Существует много инноваций и модификаций этих интерфейсов, и каждое приложение имеет свой собственный уникальный

интерфейс, как показано ниже:

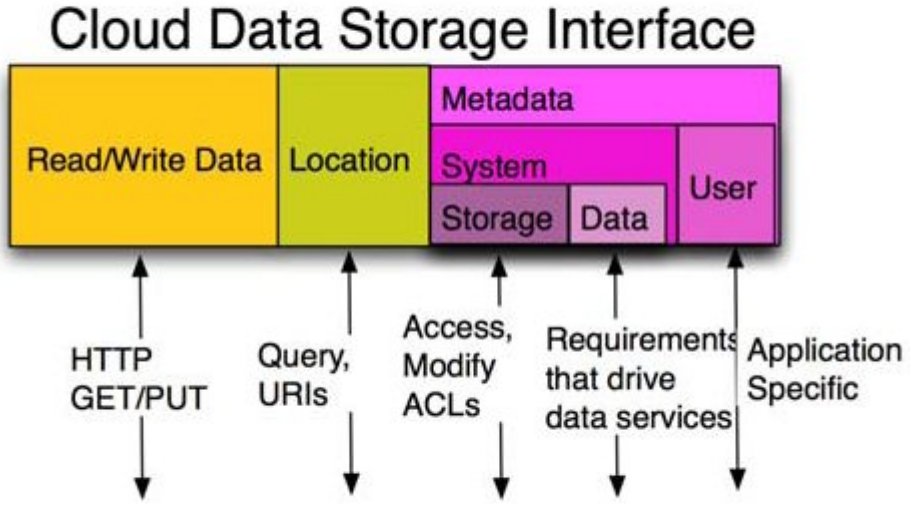


Существует и третья категория функциональных интерфейсов для хранимых данных. Этот тип интерфейсов обращается к каждому объекту данных через уникальный URI. Затем с помощью протокола HTTP и браузера можно вызвать подходящее приложение для взаиможействия с данными.

Каждый объект может быть создан, изменен, обновлен и удален как самостоятельный ресурс. В этом типе интерфейсов, контейнер, если используется, является последовательностью групп объектов данных.Этот тип контейнеров называется “мягким”.

Управление данными в облаке

Для того, чтобы удовлетворить все потребности предприятий, необходимо улучшать качество предлагаемых услуг и развертывать допонительные сервисы передачи дынных. Существует опасность, что при этом облако потеряет свою наглядность и простоту. Модель SNIA дает способ, чтобы минимизации сложности и адрес нуждающийся в cloud storage оставаться простыми. С помощью различных типов метаданных обсуждающихся в этой модели,можно создать интерфейс, который позволяют удовлетворять потребности в данных без увеличения сложности в управлении этими данными.



Управление данными и контейнерами

Любые данные помещенные в контейнер наследуют метаданные этого контейнера.Метаданные могут быть изменены в контейнере или отдельном элементе данных до желаемого уровня. Даже если функциональный интерфейс не поддерживает имеющийся тип метаданных по отдельным элементам данных, он может быть пременен по отношению к контейнеру.Для файловых интерфейсов, которые поддерживают расширеннае атрибуты(например, CIFS, NFSv4), эти расширенные атрибуты могут быть использованы для определения данных систем метаданных и последующее их и замещение на указанные в контейнере.

Эталонная модель для хранения интерфейса Cloud Storage

Собирая все вместе мы имеем следующую модель: