Файловые системы: от дисков к облакам

Обзор тенденций развития современных файловых систем (часть 2).

Сергей Платонов – руководитель исследовательской лаборатории RAIDIX.

Введение

В первой части статьи были перечислены параллельные файловые системы, применяемые для задач BIGDATA, высокопроизводительных кластеров и облачных инфраструктур.

Во второй части приводим сравнительные характеристики (табл. 1), которые позволяют оценить их перспективность для той или иной задачи.

Архитектура и особенности облачных файловых систем

Рост популярности облачных технологий вызвал всплеск новых требований к файловым системам (ФС). Появляется новый тип файловых систем, которые используют в качестве бэкенд-интерфейса облачные системы хранения данных.

Подобные файловые системы применяются для использования одного или нескольких облачных ресурсов и позволяют решать часть ограничений, таких как обеспечение безопасности хранения данных, доступность, задержки доступа. При этом клиенты имеют доступ к данным, хранящимся в облачных ФС, как через SMB/NFS-протоколы, так и монтируя их как локальные файловые системы.

Важнейшим шагом к адаптации подобных решений стало появление единого интерфейса – CDMI (Cloud Data Management Interface), представляющего собой единый стандарт REST ful HTTP операций, по которому будут создаваться, обновляться, удаляться и запрашиваться объекты из облака. Он же будет использоваться и управляющими приложениями для редактирования метаданных контейнеров.

Файловые системы транслируют запросы к ним в запросы CDMI и обладают возможностью кэширования для обеспечения быстрого доступа к объектам.

Некоторые облачные файловые системы предоставляют также дополнительные функции, такие как сжатие и шифрование данных на стороне клиента.

Как уже упоминалось в первой части статьи, до недавнего времени существовал только один публичный стандарт, связанный с работой файловой системы.

POSIX, описывающий работу UNIX-подобных операционных систем, появился из IEEE 1003.1-1988, разработанного еще в 1988 г. Последнее изменение в POSIX filesystem IO было сделано в 1991 г.

С появлением web стала очевидным необходимость появления нового интерфейса, т.к. невозможно выполнить системный вызов через HTTP. На рубеже тысячелетий одним из основателей протокола HTTP был описан REST (Representational State Transfer, "передача представлений состояний").

В последние 3 года с переносом приложений и систем хранения данных в облако сильно возросла популярность REST-интерфейса. Существует мнение, что REST со временем заменит POSIX для всех типов приложений.

POSIX был основным стандартом в течение 35 лет – существуют миллионы приложений, поддерживающих этот стандарт. При этом он не изменялся более 20 лет. Было множество предложений по его доработке, но они требуют тщательного тестирования и обновлений стеков операционных систем, что не поддерживается многими вендорами, входящими в OpenGroup, контролирующий развитие POSIX.

Обеспечение корректности метаданных при конкурентном обращении нескольких потоков к одним и тем же данным является достаточно сложным для параллельных файловых систем, хранящих миллиарды файлов и приложений, которые могут требовать множество параллельных операций ввода-вывода.

POSIX позволяет выполнять чтение или запись части файла, чего нет в REST.

Среди основных проблем POSIX выделяют структуру файловых дискрипторов (inodes) и требования атомарности операций.

В REST используется небольшой набор предопределенных методов для доступа к объектам.

Основным преимуществом REST является его гибкость: разработчик управляющей системы в определенных границах может создавать любые новые специальные методы.

Одним из сегодняшних вызовов, как говорилось ранее, является возможность создания файловых систем, поддерживающих хранение миллиардов объектов в едином пространстве имен. Существу-

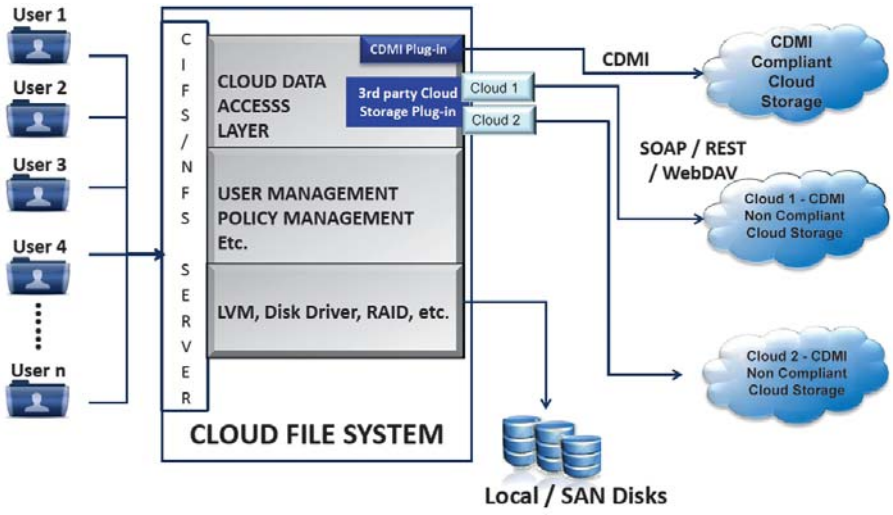

Рис. 1. Архитектура облачных файловых систем.

Табл. 1. Сравнительные характеристики параллельных файловых систем.

Критерий	GPFS	Luster FS	HDFS	Ceph	PANFS
Автор ФС	IBM	SUN Microsystems Сейчас разрабатывается консорциумом. Права на торговую марку у Xyratex	Apache Software Foundation	Inktank Storage	Panasas
Обеспечение отказоустойчивости	Нет единой точки отказа	Нет единой точки отказа	Есть единая точка отказа	Нет единой точки отказа	Нет единой точки отказа
Архитектура	Симметричная	Асимметричная	Асимметричная	Асимметричная	Асимметричная
Совместимость с POSIX	Полная	Не полная, достаточная для работы	Частичная	Полная с некоторыми изменениями	Полная
Обеспечение целостности данных	Высокий уровень обеспечения целостности данных	Планируется обеспечение высокого уровня целостности за счет интеграции с zfs	Нет	За счет файловых систем OSD	Да, за счет Panasas Tiered Parity
Поддерживаемые задачи	Файловая система общего назначения	В первую очередь файловая система разработана для задач HPC, на которых обеспечивается высочайший уровень производительности.	HDFS применима только для задач MAP-REDUCE.	Ceph создан для богатого набора задач. Поддерживается 3 вида доступа: объектный, файловый, блочный.	Файловая система общего назначения
	Высокая производительность MAP-REDUCE	Высокая производительность MAP-REDUCE.	Обеспечивает максимальную производительность на них	Ориентирован на задачи ЦОД и HPC	Высокая производительность MAP-REDUCE
	Высокая производмительность работы с потоками данных (HPC, Media and Entertaiment)				Высокая производмительность работы с потоками данных (HPC, Media and Entertaiment)
	Хорошая производительность при работе с транзакционными задачами				
Работа с небольшими файлами	Поддерживается	Поддерживается, но есть неоднозначные показатели уровня производительности	Не поддерживается	Поддерживается	Поддерживается
Дополнительные функции	Шифрование, аутентификация, репликация, функция Native RAID	Планируется расширение функциональности в будущих версиях.	Нет	Интеграция с облачными платформами, мгновенные копии, клонирование, ThinProvisioning	Мгновенные снимки, квотирование
Конкурентный доступ к одному файлу	Есть	Есть	Ограничен	Есть	Есть
Поддерживаемые типы клиентов	AIX / Linux / Windows	Linux, поддержка Windows ограничена.	Linux	Linux	Linux, остальные клиенты через CIFS и NFS
Комментарий	Файловая система общего назначения. Огромный опыт внедрения и богатый набор функций. Подойдет практически для любого типа задач.	Предназначена для задач HPC и Map-Reduce. Применение в качестве файловой системы в рамках облачных инфраструктур под сомнением.	Разработана исключительно под задачи Map-Reduce. Недостатки устраняются независимыми вендорами.	Богатый набор функций, любые типы доступа, ориентаия на облака. Но пока не очень большой опыт внедрения в производственную среду.	Применима под любой тип задач, но глубоко интегрирована с аппаратным обеспечением.

ет, однако, немного POSIX-совместимых систем, масштабирующихся до 10 петабайт данных и миллиардов объектов. При этом не все из них поддерживают конкурентные обращения к одному файлу (например, это является ограничением GlusterFS).

С другой стороны, если объекты является очень большими, мы не можем выполнять их чтение до тех пор пока операция перемещения не завершится с использованием REST-интерфейса.

REST- и SOAP-интерфейсы не смогут быть применены для приложений, требующих асинхронного доступа и случайного позиционирования внутри файла, но на инфраструктурах, требующих высокой степени масштабируемости, у POSIX появился сильный конкурент.

К файловым системам, применяемым в частных и гибридных облаках, предъявляются дополнительные требования, которым не могут удовлетворить стандартные параллельные распределенные файловые системы.

Одна из проблем распределенных файловых систем – обеспечение защиты данных клиентов. Для облачных файловых систем необходима изоляция пространств имен файловой системы для различных пользователей, шифрование на уровне файловой системы на стороне пользователей, изоляция UID/GUID для предотвращения конфликтов, обеспечение квотирования и, наконец, контроль использования для задач биллинга.

Одним из проектов, направленных на решение этих задач, является HekaFS (ранее известный как GloudFS), развиваемый в рамках Fedora Project и спонсированный RedHat.

HekaFS представляет из себя набор расширения GlusterFS, который превращает эту распределенную файловую систему в готовую к внедрению в облачную инфраструктуру.

HekaFS предоставляет следующий уровень дополнений к стандартной ПФС:

- более строгий уровень аутентификации и авторизации;
- шифрование как онлайн, так и офлайн;
- поддержка множества пользователей (изоляция пространств имен и ID);
- фреймворк (Cli и UI) для конфигурирования системы.

Также в процессе разработки функции квотирования и биллинга.

Компанией Oracle в 2011 г. была представлена Oracle Cloud File System. Основной целью разработки послужило значительное снижение стоимости и сложности управления инфраструктурой хранения в частных облаках.

Oracle Cloud File System позволяет:

– развертывать совместно используемый пул ресурсов хранения с единым пространством имен для приложений, операционных файлов и пользовательских файлов;

– предоставлять доступ к хранящимся данным по сети хранения или по обычным сетям;

– оперативно увеличивать, уменьшать и переносить ресурсы хранения без остановки работы приложений *(http://www.oracle.com/ru/corporate/press/press-release-rufeb-15-326447-ru.html)*.

В основе OSFS используется хорошо известный администраторам СУБД Oracle продукт ASM (рис. 2).

OCFS использует Automatic Storage Management Dynamic Volume Manager как службу управления томами для ASM Cluster File System и других файловых систем.

ADVM, разработанный для использования в одноузловых и кластерных средах, обладает такими функциями ASM, как динамическое выделение пространства, является кроссплатформенным, легко управляется посредством ASM CMD, EM и SQL.

Oracle Automatic Storage Management Cluster File System – это кластерная кэш-когерентная файловая система общего назначения с высоким уровнем готовности для хранения любых типов файлов.

ACFS является файловой системой, совместимой с POSIX, X/OPEN и Windows. ACFS может быть использована как в одноузловой, так и кластерной инфраструктурах и обладает следующей функциональностью:

– *мгновенные снимки*, выполняемые по технологии copy-on-write. Обеспечивается восстановление файлов по требованию;

– *тэгирование*, позволяющее пользователю объединять несколько файлов в группу и выполнять на них групповые действия;

– *репликация в режиме active-standby* для обеспечения катастрофоустойчивого решения. Репликация осуществляется за счет асинхронной передачи и проигрывания на вторичном сайте файлов журналов;

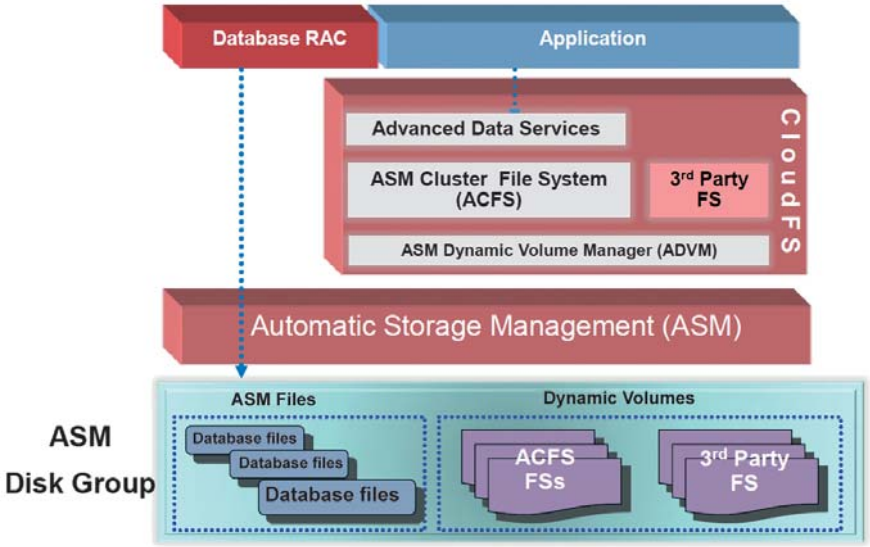

Рис. 2. Высокоуровневая архитектура Oracle Cloud File System.

– *расширенные средства обеспечения безопасности,* работающие поверх средств операционных систем. ACFS System Security позволяет пользователям создавать области безопасности (realms), определяет пользовательские и групповые политики безопасности и обеспечивает контроль доступа к объектам файловой системы.

Областью безопасности ACFS System Security является виртуальный контейнер файлов и директорий. Права доступа к контейнеру определены фильтром безопасности, состоящим из правил и наборов правил обеспечения доступа, а также прав выполнения операций.

Управление системой безопасности ACFS происходит только при применении специального пароля администратора безопасности, отличного от основного пароля администратора;

– *шифрование* – еще одним механизм защиты информации в облаке от несанкционированного доступа.

ACFS System Encryption позволяет пользователям шифровать данные, расположенные на СХД, и предоставляет ключи для дешифрования. Кодировать можно как всю файловую систему, так и отдельные папки и файлы.

Система шифрования прозрачна для авторизованных пользователей и приложений. Зашифрованные и незашифрованные файлы могут существовать одновременно в рамках одной файловой системы.

Каждый файл защищен двумя ключами – File Encryption Key и Volume Encryption Key.

Файл зашифрован FEK, который хранится на диске и зашифрован VEK. VEK хранится в OracleWallet и защищен паролем;

– *зеркалирование и страйпинг файлов*;

– *поддержка протоколов SMB (CIFS) и NFS.*

Oracle Cloud File System имеет единый интерфейс управления, единый набор инструментов установки и первоначальной конфигурации.

Заключение

Динамика роста объемов данных, новые архитектурные подходы и задачи требуют развития файловых систем. Облачные провайдеры не могут использовать стандартные распределенные файловые системы из-за проблем безопасности и конфликтов пользователей. Рынок HPC в скором времени потребует стократного роста производительности, а стандарт POSIX уже не удовлетворяет нуждам потребителей.

Скорее всего, в ближайшие пять лет мы увидим множество различных решений, кардинально отличающихся по своим функциям и предназначению, под общим флагом "файловые системы". Мы находимся на смене подходов, когда старые решения не удовлетворяют всем требованиям быстро меняющегося рынка, а новые не готовы принять на себя весь спектр задач.

Сергей Платонов,
исследовательская лаборатория RAIDIX