

СОЗДАНИЕ СЕТЕВОГО ФАЙЛОВОГО ХРАНИЛИЩА СРЕДСТВАМИ СПЕЦИАЛИЗИРОВАННОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ

Чусов П.А., студент УрГЭУ, *chusic@uspu.me*
Стариченко Е.Б., к.п.н., профессор кафедры ИКТО

Аннотация

В статье излагается результат исследования, проведённого в рамках выпускной квалификационной работы. В её ходе нами был проведён анализ современных средств создания разделяемых файловых ресурсов, а также определены современные требования к подобного рода системам. Одно из них — использование сетевой файловой системы ZFS, поддерживающей большие объёмы данных, объединяющей концепции файловой системы и менеджера логических дисков и физических носителей, использующую новаторскую структуру данных на дисках, а также простое управление томами хранения данных. ZFS способна обеспечить высокую скорость доступа к ним, контроль их целостности, а также минимизацию фрагментации данных.

В работе предлагается и описывается вариант создания файлового хранилища с помощью специализированной операционной системы Nas4Free, как наиболее полно отвечающей требованиям, предъявляемым к системам подобного рода.

Ключевые слова: системы хранения данных, файловый сервер, разделение ресурсов

Современную жизнь уже достаточно сложно представить без постоянного обмена информацией. В глобальной сети представлено множество сервисов, предлагающих свои ресурсы для хранения ваших данных. Безусловно это удобно, однако имеет и негативные стороны. Так, чаще всего, бесплатно предоставляются только ограниченные возможности сервиса. Например, может быть установлено ограничение скорости, размера пересылаемого файла или выделено хранилище небольшого размера, может быть предложен низкий уровень безопасности. Кроме этого, мы оказываемся в зависимости от доступа к интернет, работоспособности и доброй воли организации, предоставляющей сервис. Это не всегда допустимо, особенно, если мы говорим о данных, используемых в рабочем процессе.

Сегодня практически любой организации приходится хранить довольно большие объёмы информации, используемой, как для внутренних нужд, так и для взаимодействия с внешним миром. И если крупные структуры могут позволить себе создавать собственные центры обработки данных, то небольшие предприятия вынуждены решать эти проблемы более бюджетными средствами. Одно из них – создание собственной системы хранения данных, как части IT-инфраструктуры.

Человечество давно разрабатывает различные способы хранения информации, начиная от наскальной живописи и заканчивая облачными хранилищами. Для хранения цифрового контента наиболее популярными являются жёсткие диски (hard disk drive – HDD), так как обладают наименьшей ценой за 1 Гб ёмкости и являются достаточно надёжным хранилищем. Помимо HDD, существуют так же SSD накопители (solid-state drive), обладающие

большей скоростью обмена данными, за счёт отсутствия механических деталей, но пока ещё менее надёжные и значительно более дорогие. Мы рассмотрим создание сетевого хранилища данных на основе жёстких дисков.

Стоимость гигабайта дискового пространства постоянно снижается, однако пока скорость роста объёма информации превышает скорость разработки более ёмких носителей. К тому же у жёстких дисков существует предел объёма, связанный с форм-фактором и физическими ограничениями. В ходе исследований напылений кобальта на поверхности золота, проведённых в EPFL (Ecole Polytechnique Federale de Lausanne) Харальдом Бруне, была зарегистрирована максимально-возможная плотность магнитной записи, в 200 раз превышающая плотность записи современных жестких дисков. Однако результат не может быть использован для традиционных компьютерных систем, т.к. исследования проводились при температуре -223 градуса по Цельсию. При более высоких температурах тепловое движение не позволит сохранить информацию с такой плотностью в течении длительного времени [4].

К началу 2000-х жесткие диски почти достигли технологического предела по миниатюрности, энергопотреблению, быстродействию, отказоустойчивости и себестоимости. С тех пор удалось существенно повысить только их объем, а остальные характеристики изменились мало. За последние 15 лет быстродействие жестких дисков росло примерно в 100 раз медленнее, чем процессоров [0]. Для решения этой проблемы были созданы системы (сети) хранения данных (СХД), основанные на серверах и разграничении доступа к ним. СХД – это комплексное программно-аппаратное решение по организации надёжного хранения информационных ресурсов и предоставления гарантированного доступа к ним. Она представляет собой архитектурное решение для подключения внешних устройств хранения данных, таких как дисковые массивы, ленточные накопители, оптические приводы, таким образом, чтобы операционная система распознала эти ресурсы как локальные.

Существует несколько подходов к построению систем хранения данных. Один из них – NAS (Network Attached Storage). NAS – это устройство для файлового обмена по IP сети. Он поддерживает работу с протоколами CIFS и NFS, FTP, SSH, SMB, DFS и т.д. NAS может существовать, как элемент СХД или как самостоятельный сервер на базе собственной операционной системы.

Любой NAS, как файловый сервер, прежде всего должен обеспечивать надежность хранения данных. Для решения этой проблемы в 1987 году учеными из Калифорнийского университета был представлен дисковый массив с избыточностью данных, которого принято называть RAID (redundant array of independent disks — избыточный массив независимых дисков). Параметры этой системы отвечают требованиям быстродействия и надежности, поэтому она получила огромное распространение по всему миру.

Большое значение для СХД имеет файловая система. Она должна обеспечивать высокую скорость доступа к данным, контроль их целостности, а также минимизацию фрагментации данных. Не все существующие файло-

вые системы отвечают этим требованиям. В результате сравнения нескольких вариантов мы остановились на ZFS (Zettabyte File System).

ZFC изначально создана в Sun Microsystems для операционной системы Solaris. Эта файловая система поддерживает большие объёмы данных, объединяет концепции файловой системы и менеджера логических дисков (томов) и физических носителей, новаторскую структуру данных на дисках, легковесные файловые системы (lightweight filesystems), а также простое управление томами хранения данных. ZFS является проектом с открытым исходным кодом и лицензируется под CDDL (Common Development and Distribution License). Основное преимущество ZFS — это её полный контроль над физическими и логическими носителями. Кроме того это 128-битная файловая система, что позволяет ей хранить в $18,4 \times 10^{18}$ раз больше данных, чем все известные 64-битные системы [3].

В отличие от традиционных файловых систем, которые располагаются на одном устройстве и, следовательно, при использовании более чем на одном устройстве для них требуется менеджер томов, ZFS строится поверх виртуальных пулов хранения данных, называемых zpool. Пул построен из виртуальных устройств (vdevs), каждое из которых является либо физическим устройством, либо зеркалом (RAID 1) одного или нескольких устройств, либо (RAID Z) — группой из двух или более устройств. Емкость всех vdevs затем доступна для всех файловых систем в zpool. Для ограничения пространства, доступного конкретной файловой системе, может быть установлена квота. Кроме того, возможно использование дискового резервирования — это гарантирует, что всегда будет оставаться некоторый доступный объём для конкретной файловой системы [6].

Основные возможности ZFC, как файловой системы:

- хранит контрольные суммы и не позволяет считывать мусор вместо данных;
- сохраняет целостность до такой степени, что утилиты вроде `chkdsk` или `fsck` для нее просто нет;
- снабжена инструментом проверки целостности холодных данных и их автоматической коррекции, если данные сохранены с избыточностью;
- способна мгновенно создавать снимки своего состояния и хранить их хоть за каждую минуту месяца, монтировать любой набор снимков, откатываться к снимку.

Основные возможности ZFS, как RAID-массива:

- программный, т.е. не требующий аппаратного контроллера;
- аппаратно-независимый, достаточно наличия SATA-портов;
- способный к реконструкции деградированного массива с частично нечитаемыми дисками с потерей только тех данных, для которых нет ни одной копии;
- работающий при проверке целостности и реконструкции только с полезными данными, а не со всем массивом;

- с поддержкой аналогов RAID1 (зеркало), RAID5 (избыточность в размере одного диска), RAID6 (двух) и даже RAID7 (сохраняющий данные при выходе из строя любых трех дисков массива), а также более сложных вариантов, подобных RAID50 или RAID60.

Существует множество вариантов создания сетевого хранилища, от готовых устройств до самостоятельной установки необходимых компонентов операционной системы. Мы выбрали вариант на основе специализированной *nix системы. Наиболее популярны три сборки — NAS4Free и FreeNAS, основанные на FreeBSD и OMV (openmediavault), основанная на Debian Linux. Приведем табличное сравнение аппаратных требований:

Продукт	Процессор	Оперативная память	Системный накопитель
Nas4Free 9.x, с ZFS	x86, x86-64	Минимум 512 Мб, рекомендуется 4+ Гб	400 Мб (остальное доступно для данных)
Nas4Free 9.x, без ZFS	x86, x86-64	256 Мб для full, 384 Мб для embedded, 512 Мб рекомендуется	400 Мб (остальное доступно для данных)
FreeNas 8.x, с ZFS	x86-64	6+ Гб	2 Гб и более (будет занят целиком)
FreeNas 8.x, без ZFS	x86, x86-64	4+ Гб	2 Гб и более (будет занят целиком)
OMV	x86, x86-64	1 Гб	2 Гб и более

Для реализации проекта, был выбрана Nas4Free, так как она является абсолютно бесплатной по сравнению с FreeNAS и распространяется по свободной лицензии BSD. Так же в Nas4Free присутствует ZFS, чего нет в OMV. Так же Nas4Free менее требовательна к аппаратной части. Проект имеет многолетнюю историю (происходит от FreeNAS 0.7, 0.6 и ранее), старшая цифра текущей версии — 9. То есть сборка хорошо организована и продолжает развиваться. Основанная на последнем релизе FreeBSD 9.1, Nas4Free скомпилирована в двух версиях, для 32- и 64-разрядных Intel x86-совместимых процессоров. Возможности выбранной системы следующие:

- WebGUI (управление и настройка через web-интерфейс);
- диски и массивы: ZFS v28 (зеркало, RAID-Z 1, 2, 3...), софт-RAID 0, 1, 5 и смесь (1+0, 1+1 и т.п.), шифрование дисков (с использованием аппаратного ускорения, если присутствует), файловые системы ZFS, UFS, ext2/3, FAT, NTFS, iSCSI-инициатор;
- сетевые протоколы и службы: SMB/CIFS (Samba) — для Windows-машин, AFP (Netatalk) — для Mac OS, NFS — для *nix, FTP (ProFTPD), TFTP (tftp-hpa), Rsync (client/server) — мощный инструмент, в том числе для бэкапа, SCP (SSH), iSCSI-таргет;
- сервисы: UPnP server, Webserver (lighttpd), Network bandwidth measure (Iperf), Bittorrent client (Transmission);

- мониторинг: S.M.A.R.T (self-monitoring, analysis and reporting technology — технология самоконтроля, анализа и отчётности) — технология оценки состояния жёсткого диска встроенной аппаратурой самодиагностики, а также механизм предсказания времени выхода его из строя, e-mail, SNMP, Syslog, UPS (NUT) [5].

Здесь можно увидеть абсолютно все что пригодиться для создания NAS сервера. Но наиболее важным достоинством является использование файловой системы ZFS.

Для создания файлового хранилища был выбран путь через самостоятельную подготовку аппаратной части и установки на нее ОС Nas4Free, с дальнейшей конфигурацией этой системы. Местом проведения работ стал педагогический университет, где файловое хранилище было необходимо для обмена данными и повышения эффективности работы сотрудников Управления информатизации.

Для аппаратной части был собран системный блок. В него встроена материнская плата ECS 915GV-M10 Rev. 1.2 с 32-битным процессором Intel Pentium 4 и 1 Гб оперативной памяти. Специализация NAS на выполнение только файловых операций позволяет использовать в качестве вычислительного ядра относительно маломощные процессоры, поэтому для небольшого сервера можно использовать не самые технически новые компоненты. Объём памяти также достаточен для работы операционной системы и кэширования дисковых операций. Добавлен CD-ROM и четыре жёстких диска по 200 Гб под файловое хранилище и один на 80 Гб под операционную систему. Диски хранилища были объединены в RAID-массив нулевого уровня для повышения производительности. В ходе эксперимента выяснилось, что жёсткие диски, как наиболее нагруженный элемент, сильно нагреваются. Для предотвращения их повреждения системный блок был оснащен тремя дополнительными вентиляторами, что позволило выровнять температурный режим и привести его к приемлемым значениям.

Установка 32-битной версии Nas4Free была поведена с помощью виарда и показала отсутствие конфликтов с оборудованием. Последующая работа с системой осуществляется через Web-интерфейс, доступ к которому возможен через сконфигурированный при инсталляции IP-адрес. Интерфейс имеет русскую локализацию, есть встроенная справка и активное сообщество, поэтому сложности с настройкой оперативно решаются.

Конфигурируя сервер, прежде всего мы смонтировали диски, организовав RAID-массив и установили файловую систему ZFS. Подключили поддержку S.M.A.R.T. и отправку отчетов о состоянии на e-mail. Тем самым мы обеспечили нашей системе надежность и быстродействие. Поскольку сервер предполагается использовать в сети Windows-машин, мы установили службу CIFS/SMB (NetBIOS), чтобы открыть к нему удалённый доступ. При этом создали сетевой ресурс, который будет виден в Windows с правами на запись для всех пользователей.

SMB-ресурсы используются в локальной сети, а для доступа извне необходимо включить FTP-сервер. Мы установили по-умолчанию порт доступа 21 и включили DDNS-службу.

Nas4Free предоставляет гибкие возможности настройки прав доступа и политики безопасности. Мы ограничили доступ к FTP, разрешив его только авторизованным пользователям. Каждому пользователю системы было выделено персональное дисковое пространство, к которому имеет доступ только он, а также создан общий ресурс для обмена файлами, права на запись в который были предоставлены всем зарегистрированным участникам.

Поскольку графический интерфейс является надстройкой над операционной системой и возможны ситуации, когда он оказывается недоступен или недостаточно функционален, мы установили службу SSH (Secure Shell — «безопасная оболочка»). Это сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений (например, для передачи файлов). По функциональности он схож с протоколами telnet и rlogin, но, в отличие от них, шифрует весь трафик, включая и передаваемые пароли. SSH допускает выбор различных алгоритмов шифрования. Кроме того, SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол. Таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео [2].

Нам удалось настроить созданный NAS на полноценную и разнообразную работу в сети Управления информатизации университета. Далее планируется портировать полученное решение и создать подобные сервера для учебных подразделений и служб университета, чтобы обеспечить оперативность их работы. Однако развитием данного проекта является переход на облачные технологии и создание частного облака. Связано это исключительно с сервисными возможностями — облако предоставляет современный Web-интерфейс не только администратору, но и пользователям, есть возможность on-line просмотра документов (txt, pdf, odf) и редактирования файлов различного формата (txt, odf), календарь, задачи, адресная книга, синхронизация по протоколу WebDAV, поиск по содержимому, а что самое главное — возможность монтирования локальных папок и внешних хранилищ по протоколам FTP, Samba, и др. Необходимость этого продиктована тем, что пользователи, не являющиеся специалистами в информационных технологиях, с большой неохотой принимают необходимость «ручного» управления своими данными и предпочитают избегать применения «лишних» с их точки зрения инструментов.

ЛИТЕРАТУРА:

1. Конец эры жестких дисков: скрытая угроза и новая надежда. // iBusiness. - Р/д: http://ibusiness.ru/blog/upravlyeniye_proizvodstvom/35110 (ссылка действительна на 15.04.2015)

2. Робачевский А., Немнюгин С., Стесик О. Операционная система UNIX. — Спб: БХВ-Петербург, 2010. — 656 с.
3. Руководство по администрированию файловых систем ZFS Solaris. — Р/д: <http://docs.oracle.com/cd/E19253-01/820-0836/820-0836.pdf> (ссылка действительна на 15.04.2015)
4. Установлен предел плотности записи информации на жёсткий диск. // Радиодлоцман. — Р/д: <http://www.rlocman.ru/news/new.html?di=3899> (ссылка действительна на 15.04.2015)
5. NAS4Free — The Free Network Attached Storage Project. — Р/д: <http://wiki.nas4free.org/doku.php> (ссылка действительна на 15.04.2015)
6. ZFS Best Practices Guide. — Р/д: http://www.solarisinternals.com/wiki/index.php/ZFS_Best_Practices_Guide (ссылка действительна на 15.04.2015)