

Презентация на тему: «Файловый сервер»

Выполнил студент группы
КС-306 :Чадов В

Файл-сервер — это выделенный сервер, оптимизированный для выполнения файловых операций ввода-вывода. Предназначен, для хранения файлов любого типа. Как правило, обладает большим объемом дискового пространства.

Как правило, файл-сервер оборудован RAID контроллером для обеспечения надежности сохранности данных.

Архитектура «файл-сервер»

Файл-серверные приложения — приложения, схожие по своей структуре с локальными приложениями и использующие сетевой ресурс для хранения программы и данных.



Функции сервера: хранения данных и кода программы. Функции клиента: обработка данных происходит исключительно на стороне клиента. Количество клиентов ограничено десятками. Плюсы: низкая стоимость разработки; высокая скорость разработки; невысокая стоимость обновления и изменения ПО. Минусы: низкая производительность (зависит от производительности сети, сервера, клиента); плохая возможность подключения новых клиентов; ненадежная система.



Файловый сервер

Узел вычислительной сети, реализующий начальный уровень архитектуры клиент-сервер. Обычно файловый сервер работает под управлением серверной операционной системы*. Основной задачей файл-серверов является надежное хранение и быстрый доступ к данным, находящимся на общем дисковом пространстве, при котором обеспечивается управление доступом к файлам и базам данных;



Данная серия специально спроектирована для решения задач по хранению и быстрому доступу к данным, как отдельных пользователей, так и всего предприятия в целом. Для достижения полной сохранности Ваших данных, в серверах используются RAID-контроллеры с технологиями зеркалирования данных, для коррекции ошибок, используется память FB DIMM ECC, а в старших моделях реализовано резервирование питания с технологией HOT-SWAP, что позволяет уберечь сервер от остановки даже при выходе из строя одного блока питания. Опционально данные серверы могут монтироваться в 19 дюймовую стойку.



FTP (File Transfer Protocol - Протокол передачи файлов) позволяет передавать файлы между двумя компьютерами, соединенными средствами Internet. Для доступа к FTP вам нужна программа-клиент для соединения с машиной, содержащей файлы (сервером) Если в вашей системе есть FTP-клиент и вы соединены с Internet, вы можете получить доступ к очень большому количеству файлов, доступных на FTP-серверах. Если у вас нет прямого доступа к FTP, вам следует обратить свое внимание на серверы, позволяющие получить доступ к FTP средствами E-mail.

Что значит Anonymous

Большое количество серверов в Internet предоставляют доступ к файлам средствами так называемого Anonymous FTP. Это значит, что вы можете получить доступ к машине не являясь ее официальным пользователем. Эти сервера содержат программное обеспечение, документы различного рода, картинки, тексты песен и тому подобное. Гигантский объем информации на таких серверах доступен любому.

Соединение с сервером

Просто наберите команду ftp, а затем имя системы с которой хотите соединиться, например:

```
ftp ftp.lipetsk.su
```



Через несколько секунд появится запрос login: Если вы не являетесь официальным пользователем системы, введите Anonymous. Затем появится запрос Password: Что вы введете в общем-то не имеет значения, но согласно неписанным правилам, вы должны ввести свой E-mail адрес, поскольку операторы серверов были бы не прочь узнать, кто использует их сервер. После этого вы увидите приглашение сервера (обычно ftp>) Вы в системе. Вы можете посмотреть список директорий, набрав команду dir. Если сервер использует Unix и dir не работает, попробуйте ls -l. Обычно сервер имеет файл с именем README или 00-index.txt, содержащий в себе краткое описание сервера и местонахождения файлов. Надо отметить, что FTP-клиенты, встроенные в Web-browser'ы выполняют всю процедуру соединения автоматически. Так же поступают и специализированные FTP-клиенты типа CuteFTP. Команды FTP

Все команды FTP используются для получения файлов. Некоторые команды одинаковы для всех серверов, некоторые нет. Также, некоторые серверы поддерживают свои собственные команды, например, получение целой директории одной командой, поиск по директориям, etc. Прочтите содержащийся почти на каждом сервера файл README для получения информации о таких командах. Обычно FTP-сервер выдает список команд в ответ на команду help или ? Итак, основные команды:

- `ascii` Переключение в `ascii` режим. Этот режим является
- стандартным для передачи текстов.
- `binary` Переключение в двоичный режим. Для передачи архивов, картинок, etc.)
- `cd` Смена директории на сервере.
- `dir` Список файлов в текущей директории сервера.

- `get` Копирует файл с сервера на ваш компьютер.
- `help` Список доступных команд.
- `lcd` Смена директории на вашем компьютере.
- `lpwd` Показывает текущую директорию вашего компьютера
- (Для Unix-машин используйте `!pwd`)
- `mget` Получение сразу нескольких файлов по маске (*.zip, например)
- `pwd` Показывает текущую директорию на сервере.

Основные принципы работы

Итак, вы в системе. Вы можете осмотреться, выбрать интересующие вас файлы и получить их (большинство серверов не позволяет пользователям самим посылать файлы) Обычно все самое интересное находится в директории /pub. Помните, что в Unix-системах вместо обратной дроби (\) используется прямая (/). Некоторые сервера содержат файлы типа ls-lR, которые содержат полный список имеющихся на сервере файлов. Если такового не имеется, вы можете набрать ls -lR и получить такой список. Однако помните, что это может занять много времени (списки размером 10 Mb не являются редкостью)

При приеме нетекстовых файлов вы должны использовать двоичный режим. Перед получением файла наберите команду `binary`. Обычно файлы хранятся в сжатом виде. Чаще всего в Unix-системах используется программа `compress`, результатом работы которой являются файлы с расширением `.Z`. Встречаются также `Arc`, `Zoo`, `Arj`, `Lzh`, `Gz`, `Zip`, в соответствии с используемой программой компрессии. Проблемой может стать получение файлов вида `filename.tar.gz`. MS-DOS не воспринимает подобные имена файлов, поэтому для получения такого файла используйте следующую схему:

get filename.tar.Z filename.tz

Затем используйте последовательно `compress` и `tar` для распаковки файла. Многие сервера поддерживают режим on-line распаковки файлов на лету (на тот случай, если у вас нет утилит для декомпрессии файлов) Используйте следующую схему:

get filename.z filename

Вы получите распакованный файл. Помните, что передача такого файла займет больше времени, чем передача сжатого.

Архитектура распределенных файловых систем

Распределенная система обычно имеет два существенно отличающихся компонента - непосредственно файловый сервис и сервис директорий.

Интерфейс файлового сервера

Для любой файловой системы первый фундаментальный вопрос - что такое файл. Во многих системах, таких как UNIX и MS-DOS, файл - не интерпретируемая последовательность байтов. На многих централизованных ЭВМ (IBM/370) файл представляется последовательность записей, которую можно специфицировать ее номером или содержимым некоторого поля (ключом). Так, как большинство распределенных систем базируются на использовании среды UNIX и MS-DOS, то они используют первый вариант понятия файла.

Файл может иметь атрибуты (информация о файле, не являющаяся его частью).
Типичные атрибуты - владелец, размер, дата создания и права доступа.

Важный аспект файловой модели - могут ли файлы модифицироваться после создания. Обычно могут, но есть системы с неизменяемыми файлами. Такие файлы освобождают разработчиков от многих проблем при кэшировании и размножении.

Защита обеспечивается теми же механизмами, что и в однопроцессорных ЭВМ - мандатами и списками прав доступа. Мандат - своего рода билет, выданный пользователю для каждого файла с указанием прав доступа. Список прав доступа задает для каждого файла список пользователей с их правами. Простейшая схема с правами доступа - UNIX схема, в которой различают три типа доступа (чтение, запись, выполнение), и три типа пользователей (владелец, члены его группы, и прочие).

Файловый сервис может базироваться на одной из двух моделей - модели загрузки/разгрузки и модели удаленного доступа. В первом случае файл передается между клиентом (памятью или дисками) и сервером целиком, а во втором файл сервис обеспечивает множество операций (открытие, закрытие, чтение и запись части файла, сдвиг указателя, проверку и изменение атрибутов, и т.п.). Первый подход требует большого объема памяти у клиента, затрат на перемещение ненужных частей файла. При втором подходе файловая система функционирует на сервере, клиент может не иметь дисков и большого объема памяти.

Интерфейс сервера директорий

Обеспечивает операции создания и удаления директорий, именованя и переименования файлов, перемещение файлов из одной директории в другую.

Определяет алфавит и синтаксис имен. Для спецификации типа информации в файле используется часть имени (расширение) либо явный атрибут.

Реализация распределенных файловых систем

Использование файлов

- Приступая к реализации очень важно понимать, как система будет использоваться. Приведем результаты некоторых исследований использования файлов (статических и динамических) в университетах. Очень важно оценивать представительность исследуемых данных.
- большинство файлов имеют размер менее 10К (следует перекачивать целиком).
- чтение встречается гораздо чаще записи (кэширование).
- чтение и запись последовательны, произвольный доступ редок (упреждающее кэширование, чтение с запасом, выталкивание после записи следует группировать).
- большинство файлов имеют короткое время жизни (создавать файл в клиенте и держать его там до уничтожения).
- мало файлов разделяются (кэширование в клиенте и семантика сессий).
- существуют различные классы файлов с разными свойствами (следует иметь в системе разные механизмы для разных классов).



Размножение

Система может предоставлять такой сервис, как поддержание для указанных файлов нескольких копий на различных серверах.

Главные цели:



Повысить надежность.

Повысить доступность (крах одного сервера не вызывает недоступность размноженных файлов).

Распределить нагрузку на несколько серверов.

Явное размножение (непрозрачно). В ответ на открытие файла пользователю выдаются несколько двоичных имен, которые он должен использовать для явного дублирования операций с файлами.

Ленивое размножение. Одна копия создается на одном сервере, а затем он сам автоматически создает (в свободное время) дополнительные копии и обеспечивает их поддержание.

Симметричное размножение. Все операции одновременно вызываются в нескольких серверах и одновременно выполняются.

Протоколы коррекции.

Просто посылка сообщений с операцией коррекции каждой копии является не очень хорошим решением, поскольку в случае аварий некоторые копии могут остаться не скорректированными.

Имеются два алгоритма, которые решают эту проблему.

Метод размножения главной копии. Один сервер объявляется главным, а остальные - подчиненными. Все изменения файла посылаются главному серверу. Он сначала корректирует свою локальную копию, а затем рассылает подчиненным серверам указания о коррекции. Чтение файла может выполнять любой сервер. Для защиты от краха главного сервера до завершения всех коррекций, до выполнения коррекции главной копии главный сервер запоминает в стабильной памяти задание на коррекцию. Слабость - выход из строя главного сервера не позволяет выполнять коррекции.

Метод голосования. Идея - запрашивать чтение и запись файла у многих серверов (запись - у всех!). Запрос может получить одобрение у половины серверов плюс один. При этом должно быть согласие относительно номера текущей версии файла. Этот номер увеличивается на единицу с каждой коррекцией файла. Можно использовать различные значения для кворума чтения (N_r) и кворума записи (N_w). При этом должно выполняться соотношение $N_r + N_w > N$. Поскольку чтение является более частой операцией, то естественно взять $N_r = 1$. Однако в этом случае для кворума записи потребуются все серверы.



THE END