

## Лекция 19: Распределенные файловые системы

Алексей Линёв  
Александр Мошук  
Кирилл Погорельский

some slides are adapted from the OS course at the University of Washington



## Распределенные файловые системы

- Часто встречающиеся распределенные сервисы
  - сервер печати
  - сервер e-mail
  - файловый сервер
- Основная идея распределенных файловых систем
  - обеспечить предоставление доступа к файлам и устройствам (дискам) для всей сети
- Пользователю предоставляется "традиционная" ФС
  - централизованная, локальная, разделяемая
- Но с распределенной реализацией
  - блоки получаются с удаленного узла, а не с локального диска



## Вопросы

- Что является базовой абстракцией
  - удаленная файловая система?
    - open, close, read, write, ...
  - удаленный диск?
    - read block, write block
- Способ именования
  - как именуются файлы?
  - являются ли используемые имена прозрачными для пользователя (location transparent)?
    - знает ли пользователь, где реально располагаются файлы?
  - являются ли используемые имена независимыми от расположения (location independent)?
    - изменяются ли имена файлов при их перемещении?
    - изменяются ли имена файлов при перемещении пользователя?



## Вопросы (2)

- Кэширование
  - кэширование используется для повышения производительности
  - где кэшируются блоки файлов?
    - на файловом сервере?
    - на клиентской машине?
    - на обеих системах?
- Предоставление в общее использование и когерентность
  - какова семантика предоставления в общее использование?
  - что происходит при изменении блока/файла, находящегося в кэше?
  - как ОС на сетевом узле узнает, что содержимое блоков, находящихся в ее кэше, устарело?



## Вопросы (3)

- Репликация
  - репликация используется для увеличения производительности или работоспособности
  - может ли быть в сети несколько копий одного файла?
  - если копий несколько, как выполнять их обновления?
  - что произойдет, если нарушилась связь по сети, и пользователи работали с различными копиями?
- Производительность
  - какова стоимость удаленных операций?
  - какова стоимость предоставления файлов в совместное использование?
  - каковы возможности по масштабированию при увеличении числа клиентов?
  - что ограничивает производительность: сеть, центральный процессор, дисковая подсистема, сетевые протоколы, операции копирования данных?



## Пример: SUN Network File System (NFS)

- Sun Network File System (NFS) – общепринятый стандарт удаленного доступа к файлам в UNIX
- NFS работает в локальных сетях (и даже в глобальных, но медленно)
- Основная идея
  - позволяет выполнить монтирование удаленного каталога в качестве локального
  - это дает удобный доступ к удаленному каталогу и всем его потомкам
- Выглядит очень похоже на обычные локальные операции монтирования или создания мягкой ссылки в UNIX
  - но, конечно, отличается по реализации и производительности
  - мы, вообще-то, не говорили в наших лекциях об указанных операциях, но вы уже должны были изучить их самостоятельно



- Например:
  - я монтирую `/nethome/liniov` на узле А в `/students/tmp` на узле В
  - пользователи на узле В теперь могут работать с содержимым каталога `/students/tmp`
  - если существует файл `/nethome/liniov/p4solution.txt`, пользователям на узле В он виден как `/students/tmp/p4solution.txt`
- Точно так же на локальном узле для каталога `/home/www/education/oscourse/2006su` может быть создана локальная ссылка `/home/liniov/os2006` позволяющая удобно получать доступ к данным курса из домашнего каталога

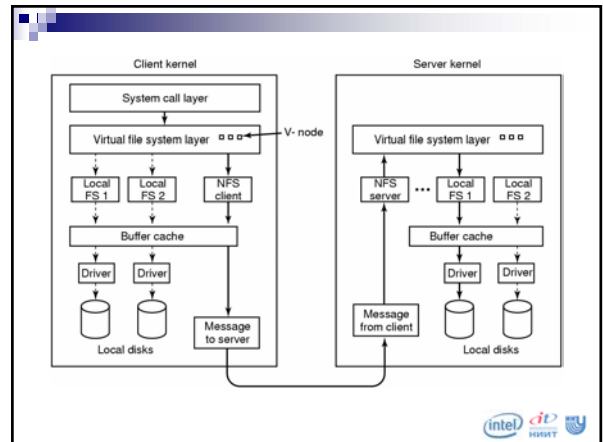


## NFS: реализация

- NFS представлена множеством операций RPC, предоставляющих удаленный доступ к файлам
  - поиск в каталоге
  - чтение записей каталога
  - управление ссылками и каталогами
  - чтение/запись файлов
- Каждый узел может быть одновременно и клиентом и сервером NFS



- NFS определяет новые уровни в файловой системе UNIX



## NFS: предоставление в совместное использование / кэширование

- При открытии файла, клиент спрашивает сервер, являются ли блоки, находящиеся в его кэше, актуальными
- После открытия файла, различные клиенты могут выполнять операции записи и получать в результате противоречивые и несовместимые данные
- Измененные данные передаются на сервер каждые 30 с



## Пример: CMU's Andrew File System (AFS)

- Разработана в Университете Карнеги-Меллона (CMU) для поддержки работы всех его студентов
- Состоит из множества рабочих станций – клиентов и выделенных файловых серверов (в отличие от NFS)
- Рабочие станции имеют локальные диски, используемые для локального кэширования файлов (изначально файлы кэшировались целиком, впоследствии – кэшировались блоки файлов размером 64 Кб) (в отличие от NFS)
- В AFS единое пространство имен – имена ваших файлов будут одними и теми же в любой точке мира (в отличие от NFS)
- AFS хорошо работает в случае продолжительных операций с удаленными файлами благодаря кэшированию на локальном диске: после медленного старта большинство операций выполняются с локальным диском



## AFS: предоставление в совместное использование / кэширование

- Для хорошего масштабирования необходимо уменьшать трафик сообщений между клиентом и сервером
- После того, как файл помещен в кэш, все операции над ним выполняются локально
- При закрытии, если файл изменялся, он сохраняется на сервере
- Клиент считает свой кэш актуальным до тех пор, пока с сервера не поступит сообщение, говорящее о его устаревании
  - если клиент получил сообщение об устаревании какого-то содержимого в кэше файла, при открытии этого файла он должен запросить новую копию с сервера; в противном случае, он использует локальную копию (в отличие от NFS)



## Пример: Berkeley Sprite File System

- Файловая система, разработанная в Калифорнийском Университете в Беркли для бездисковых рабочих станций с большим объемом памяти (отличается от NFS и AFS)
- Рассматривает память как огромный дисковый кэш
  - оперативная память распределяется между подсистемой управления файлами и подсистемой управления виртуальной памятью
- Постоянным хранилищем файлов являются серверы
  - серверы также имеют большой объем памяти, также частично используемой для целей кэширования
- Несколько рабочих станций могут кэшировать блоки файлов, открытых только для чтения
- Если файл открыт на запись более чем в одной машине, кэширование этого файла на стороне клиента запрещается, все запросы перенаправляются непосредственно на сервер (отличие от NFS и AFS)



## Выводы

- Существует множество проблем и вопросов
  - базовая абстракция
  - способ именования
  - кэширование
  - предоставление в совместное использование и когерентность
  - репликация
  - производительность
- Нет "правильного" ответа! Для различных систем требуется искать различные компромиссы!



- Производительность – всегда важный фактор
  - всегда приходится искать компромисс между производительностью и семантикой файловых операций (например, при разделении файлов)
- Кэширование содержимого файлов – ключевой момент в любой файловой системе
  - способ поддержки когерентности содержимого кэшей – важнейшее архитектурное решение
- Современные системы сталкиваются с такими проблемами, как обеспечение работы на мобильных компьютерах с непостоянным подключением к сети

