

# Распределённые файловые системы

Фишков А.А.

14.11.2011

# Понятие файлового сервиса и файлового сервера

- Файловый сервис - это то, что файловая система предоставляет клиентам, т.е. интерфейс доступа к файлам
- Файловый сервер - это процесс, который реализует файловый сервис
- Пользователь не должен знать, сколько файловых серверов имеется и где они расположены
- Файловый сервер обычно является обычным пользовательским процессом

# Интерфейс файлового сервера

- Чаще всего, файл - не интерпретируемая последовательность байтов
- Типичные атрибуты - владелец, размер, дата создания и права доступа
- Основные модели: «загрузка/разгрузка» и «удаленный доступ»
- Сервер директорий: создание/удаление директорий, именованное/переименование файлов, перемещение файлов из одной директории в другую

# Семантика разделения файлов

- **UNIX-семантика:** если за операцией записи следует чтение, то результат определяется последней из предшествующих операций записи
- **Семантика сессий:** изменения открытого файла видны только тому процессу (или машине), который производит эти изменения, а лишь после закрытия файла становятся видны другим процессам (или машинам)
- **Транзакции:** «всё или ничего», как в БД

# Исследования использования файлом

- большинство файлов имеют размер менее 10K (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).

# Структура системы (1)

- Есть ли разница между клиентами и серверами?
  - любая машина может предоставлять файловый сервис, серверы являются обычными пользовательскими процессами
  - клиенты и серверы являются фундаментально разными машинами с точки зрения аппаратуры или ПО

# Структура системы (2)

- Должны ли серверы хранить информацию о клиентах?
- Серверы с состоянием
  - короче сообщения (таблица открытых файлов)
  - выше эффективность (информация об открытых файлах может храниться в оперативной памяти)
  - блоки информации могут читаться с упреждением.
  - возможна блокировка файла
- Серверы без состояния
  - устойчивость к ошибкам
  - не требуется операций ОТКРЫТЬ/ЗАКРЫТЬ
  - не требуется память для таблиц
  - нет ограничений на число открытых файлов
  - нет проблем при крахе клиента

# Google file system

- Система специального назначения, разрабатывалась исходя из следующих основных требований:
  - В сети много дешёвых ненадёжных узлов
  - Много больших файлов
  - Обращения чтения: большие последовательные, малые произвольные
  - Обращения записи: большие последовательные
  - Много клиентов одновременно дописывают в один файл
  - Эффективное использование пропускной способности в ущерб времени отклика



# Интерфейс GFS

- Не реализует никакой стандартный API
- *create, delete, open, close, read, write*
- Иерархическая адресация по каталогам
- *snapshot* – быстрое создание копии файла или поддерева каталогов
- *record append* – одновременное дописывание в один файл

# Архитектура

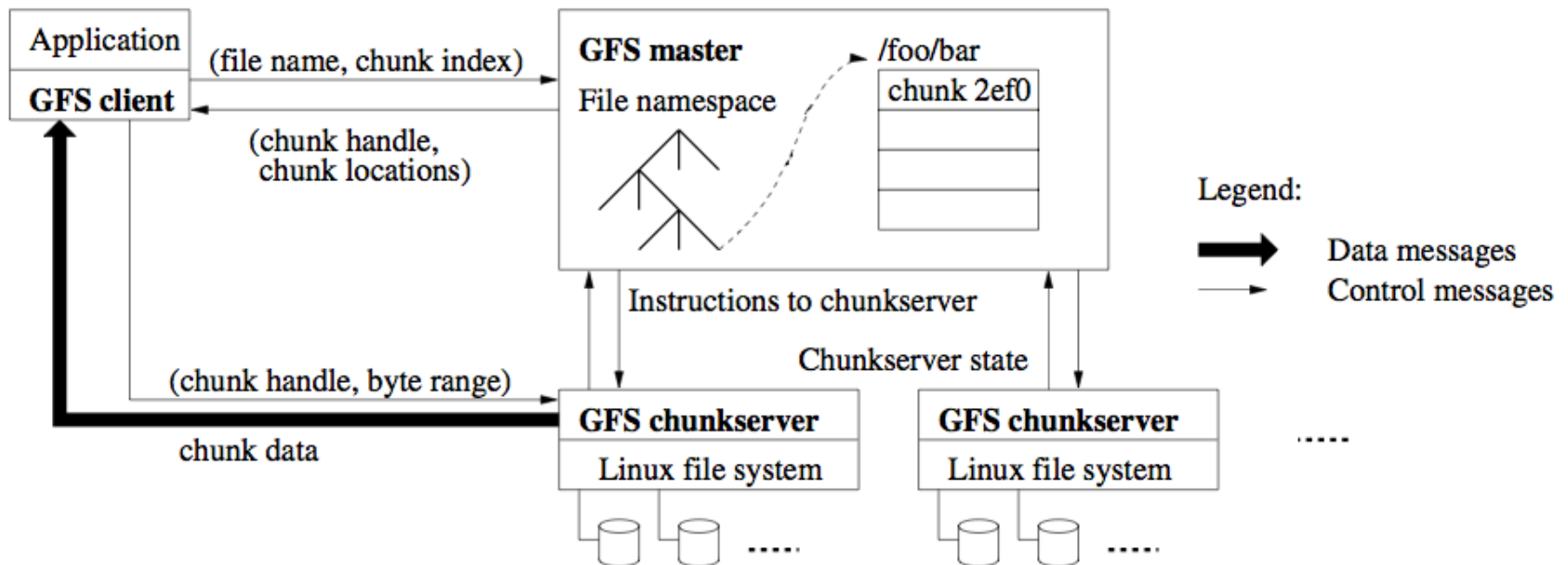


Figure 1: GFS Architecture

# Состав

- Каждый узел – машина потребительского уровня на базе Linux
- *chunkserver* – отдельный процесс, может выполняться и на клиентской машине
- *chunk* имеет фиксированный размер (64 Мб) и глобальный уникальный 64-битный *handle*
- *chunk*'и хранятся на локальных диска серверов в обычных файловых системах
- Используется репликация

# Головной сервер

- *GFS master* хранит всю информацию о структуре файловой системы, составе файлов и местоположении соответствующих *chunk*'ов
- Периодически рассылает сообщения *HeartBeat* для передачи команд и получении информации о **состоянии**
- Всё в оперативной памяти + лог

# Клиент

- Клиент обменивается метаданными с головным сервером, а запись и чтение данных – с серверов хранения
- Кэшируются только мета-данные
- Благодаря фиксированному размеру *chunk*'а, на стороне клиента определяется его положение в файле
- Сервер может дополнительно указать местоположение следующих *chunk*'ов

# Литература

1. DISTRIBUTED OPERATING SYSTEMS. Andrew S. Tanenbaum, Prentice-Hall, Inc., 1995
2. The Google File System. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung