Гришутенко Павел отчет 3.3

Поисковая система Google известна практически всем, кто пользуется Интернетом.

Понимая, что в такой инфраструктуре неизбежны частые сбои, создатели Google предусмотрели в своей системе средства отказоустойчивости, разработав файловую систему GFS, позволяющую хранить по три копии каждого файла на разных компьютерах одного кластера. В системе существуют мастер-сервера и чанк-сервера, собственно, хранящие данные. Как правило, GFS-кластер состоит из одной главной машины мастера (master) и множества машин, хранящих фрагменты файлов чанк-серверы (chunkservers). Клиенты имеют доступ ко всем этим машинам. Файлы в GFS разбиваются на куски — чанки (chunk, можно сказать фрагмент). Чанк имеет фиксированный размер, который может настраиваться. Каждый такой чанк имеет уникальный и глобальный 64 — битный ключ, который выдается мастером при создании чанка. Чанк-серверы хранят чанки, как обычные Linux файлы, на локальном жестком диске. Для надежности каждый чанк может реплицироваться на другие чанк-серверы. Обычно используются три реплики.

Файлы в GFS организованы иерархически, при помощи каталогов, как и в любой другой файловой системе, и идентифицируются своим путем. С файлами в GFS можно выполнять обычные операции: создание, удаление, открытие, закрытие, чтение и запись.

Рассмотрим, как происходит чтение данных клиентом. Сначала, зная размер чанка, имя файла и смещение относительно начала файла, клиент определяет номер чанка внутри файла. Затем он шлет запрос мастеру, содержащий имя файла и номер чанка в этом файле. Мастер выдает чанк-серверы, по одному в каждой реплике, которые хранят нужный нам чанк. Также мастер выдает клиенту идентификатор чанка. Затем клиент решает, какая из реплик ему нравится больше (как правило та, которая ближе), и шлет запрос, состоящий из чанка и смещения относительно начала чанка. Дальнейшее чтения данных, не требует вмешательства мастера.

Каждое изменение чанка должно дублироваться на всех репликах и изменять метаданные. Механизм владения чанком устроен таким образом, чтобы минимизировать нагрузку на мастера. Мастер восстанавливает состояние системы, исполняя лог операций. Мастер ключевой элемент системы, через взаимодействие с ним осуществляется взаимодействие с системой.

Авторы системы считают одной из наиболее сложных проблем частые сбои работы компонентов системы. Сбой компонента может быть вызван недоступностью этого компонента или, что хуже, наличием испорченных данных. GFS поддерживает систему в рабочем виде при помощи двух простых стратегий: быстрое восстановление и репликации.

Организация работы с данными при помощи MapReduce. Пользователи могут задавать функцию, обрабатывающую пары ключ/значение для генерации промежуточных аналогичных пар, и сокращающую функцию, которая объединяет все промежуточные значения, соответствующие одному и тому же ключу. Система берет на себя детали разбиения входных данных на части, составления расписания выполнения программ на различных компьютерах, управления ошибками, и организации необходимой коммуникации между компьютерами. Это позволяет программистам, не обладающим опытом работы с параллельными и распределенными системами, легко использовать все ресурсы больших распределенных систем.

Хранение структурированных данных в BigTable

BigTable представляет собой распределенный механизм хэширования, построенный поверх GFS, а вовсе не реляционную базу данных и, как следствие, не поддерживает SQLзапросы и операции типа Join. GFS хранит данные, не поддающиеся пониманию, хотя многим приложениям необходимы структурированные данные.

Коммерческие базы данных попросту не могут масштабироваться до такого уровня и, соответственно, не могут работать с тысячами машин одновременно. BigTable является крупномасштабной, устойчивой к потенциальным ошибкам, самоуправляемой системой, которая может включать в себя терабайты памяти и петабайты данных, а также управлять миллионами операций чтения и записи в секунду.

Пример Flickr.

Flickr является мировым лидером среди сайтов размещения фотографий.

Контролировать огромное количество ежесекундно обновляющегося контента, непрерывно пополняющиеся пользователями, основная задача Flickr.

Примечательно, что на проекте Flickr используется практически только свободное программное обеспечение. Linux, MySQL, Apache и т. д.

Все, за исключением фотографий, хранится в базе данных. Для статического контента используются выделенные сервера. Фотографии хранятся в системе хранения данных.

Каждый сервер в рамках одного сегмента в обычном состоянии нагружен ровно на половину. Выключите половину серверов, в каждом сегменте и система продолжит функционировать без изменений.

Миграция пользователей проводится время от времени для того, чтобы избавиться от проблем, связанных с излишне активными пользователями. Необходима сбалансированность в этом процессе, особенно в случаях с большим количеством фотографий.

Организация резервного копирования данных реализована с помощью процесса ibbackup, который выполняется регулярно посредством cron daemon'а, причем на каждом сегменте он настроен на разное время. Каждую ночь делается снимок со всего кластера баз данных.