УДК 004.75

РАСПРЕДЕЛЁННЫЕ ФАЙЛОВЫЕ СИСТЕМЫ ДЛЯ ОРГАНИЗАЦИИ ХРАНИЛИЩ СТРУКТУРИРОВАННЫХ ДАННЫХ

И. Н. Цырельчук
Декан факультета непрерывного и дистанционного обучения БГУИР, кандидат технических наук, доцент

Е. Н. Шнейдеров

П. А. Берашевич

Н. А. Лось

А. С. Терешкова

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: tsyrelchuk@bsuir.by, shneiderov@bsuir.by

Аннотация. Сегодня значимую роль играют технологии, позволяющие эффективно обрабатывать и хранить данные. В статье приводится сравнение новых информационных систем, ориентированных на хранение больших мас-сивов данных в распределенных кластерных системах. Условно их можно разделить на два класса: распре-делённые файловые системы и распределённые хранилища структурированных данных.

Ключевые слова: Распределённые файловые системы, распределённые хранилища структурированных данных, Google File System, Hadoop Distributed File System, BigTable, HBase

Введение. Технологии, которые позволяют формировать параллельную обработку и распределённое хранение больших массивов данных в крупных кластерных системах, требуют значительных вычислительных ресурсов. Подобные кластерные системы состоят из сотен и тысяч серверов и требуют решения вопросов обеспечения отказоустойчивости и бесперебойного функционирования. Также важной задачей при их рассмотрении является разработка высокоуровневой модели программирования процессов обработки данных, которая могла бы скрыть от пользователя детали распределения данных.

В статье рассматриваются системы, создаваемые для хранения больших объёмов данных в распределенных системах. Обычно их делят на два класса: распределённые файловые системы (РФС) и распределённые хранилища структурированных данных (РХСД). Обозреваемые системы принципиально отличаются от традиционных файловых систем и реляционных баз данных.

Основная часть. В наши дни программные средства работают с большими объёмами данных, поэтому сейчас в РФС отдается большее предпочтение не маленькой задержке при выполнении отдельно взятой операции, а высокой пропускной способности. На данный момент существует множество РФС: Google File System (GFS), Hadoop Distributed File System (HDFS), Gluster File System, POHMELFS и др. Наиболее распространёнными и известными являются GFS и HDFS [1, 2].

GFS – закрытая разработка компании Google, которая используется для хранения больших объёмов данных. Более 200 GFS-кластеров функционирует внутри Google, в крупнейших из которых насчитывается более 5 тысяч машин. GFS, как и любая распределённая файловая система, направлена на обеспечение масштабируемости, высокой производительности, надёжности и доступности.

Специфика используемых программных средств и вычислительная инфраструктура Google обусловливают отличия архитектуры GFS от других распределённых файловых систем. Хранимые файлы, как правило, имеют большой размер (десятки гигабайт) в сравнении с простыми файловыми системами. Обычно файлы модифицируются только за счёт записи новой информации прямо в конец файла. В связи с тем, что запись может производиться сразу

несколькими клиентами одновременно, необходимы гарантии того, что атомарность отдельных операций не будет нарушена. После окончания записи, файлы в основном только считываются. Кроме того, операции чтения больших данных могут быть запущены в потоковом режиме.

Распределенная файловая система HDFS является частью общедоступной платформы распределённых вычислений Hadoop. Основные идеи при разработке HDFS были взяты из системы GFS, поэтому HDFS – открытая и свободно распространяемая альтернатива закрытой технологии GFS. HDFS разработана для запуска на кластерах из доступных и массовых комплектующих, характеризуется мощной отказоустойчивостью и осуществляет автоматическое восстановление при отказах. HDFS, как и GFS, также нацелена на гарантию высокой пропускной способности при доступе к данным в потоковом режиме и оптимизирована для хранения файлов большого размера.

Сравнение характеристик файловых систем приведено в таблице 1.

Таблица 1

Сравнительная характеристика РФС GFS и HDFS

Характеристика	**GFS**	**HDFS**
Отказоустойчивость	Автоматическое восстановление после отказа любого из серверов	Автоматическое восстановление после отказа подчинённого сервера
Запись данных	Многократная запись в файл	Однократная запись файла с последующим многократным чтением
Редактирование файла	Запись с определённым отступом (без гарантии атомарности) или присоединение (гарантия атомарности). Поддерживается многопользовательский доступ	Отсутствует поддержка записи в файл после его создания одним или несколькими клиентами одновременно
Удаление файла (отсутствует моментальное физическое удаление файлов)	Файл становится скрытым для пользователя. Физически удаляется при «сборке мусора» системой (до этого можно восстановить)	Файл перемещается в директорию /trash, а после истечения определенного времени, происходит его физическое удаление
Общая структура (схема «главный-подчинённый»)	Один главный сервер (master) и несколько chunk-серверов	Один главный сервер (namenode) и несколько datanode-серверов
Создание копий (snapshot)	Стратегия отложенного копирования	Не поддерживается
Реализация	На языке C++	На языке Java
Масштабируемость	Подключение новых chunk-серверов	Простое добавление серверов
Проверка целостности	Поддерживается ресурсами файловой системы	Перекладывает проверку целостности данных на клиентов

Одной из важнейших разновидностей информационных систем являются базы данных, в которых накапливается разнородная информация, и откуда к ней получают доступ десятки миллионов пользователей по всему миру [3].

На основе распределённых файловых систем работает множество распределённых хранилищ данных: Apache Cassandra, Рикор, BigTable, HBase и другие. Последние два работают поверх Google File System и Hadoop Distributed File System соответственно.

РХСД BigTable, как и GFS, представляет из себя закрытую разработку компании Google и используется для хранения организованных данных многочисленными проектами и сервисами этой компании [4]. Функциональность Google основана на более 500 экземплярах

BigTable, в крупнейшем из которых насчитывается около 3 тысяч машин. Самые нагруженные экземпляры BigTable круглосуточно обслуживают около полумиллиона запросов в секунду.

При разработке системы BigTable внимание акцентировалось на следующих показателях: высокая производительность, надёжность, универсальность и масштабируемость. BigTable во многом похожа на базу данных и применяет различные стратегии реализации, используемые в высокопроизводительных СУБД. Но существует ряд важнейших пунктов, по которым BigTable отличается от традиционных систем. В BigTable, в отличие от реляционной модели данных, используется более упрощённая модель многомерной, сортированной, разреженной хэш-таблицы. Любое значение в хэш-таблице индексируется при помощи ключа столбца, ключа строки и времени. Само же значение является байтовым массивом и никак не интерпретируется системой. Такую хэш-таблицу можно представить как таблицу, каждая строка и столбец в которой обладают уникальными ключами. В ячейках могут находиться значения, притом у каждого из значений может быть несколько версий, к каждому из которых привязана временная метка. Иначе говоря, таблица содержит несколько временны́х слоёв. Например, Web-страницы могут храниться в такой таблице, где ключами строк будут являться URL-адреса страниц, а ячейки будут хранить несколько версий содержимого страницы, загруженных роботом в разные моменты времени.

HBase, как и HDFS, входит в состав открытой платформы распределённых вычислений Hadoop и является общедоступным аналогом BigTable [5]. Реализация, архитектура и модель данных HBase очень близка к BigTable. HBase, в отличие от BigTable, не поддерживает определение прав доступа для семейств столбцов. HBase, как и все остальные компоненты в составе платформы Hadoop, разработана на языке Java. Система реализует ряд интерфейсов для клиентских приложений: REST-интерфейс, Java API и доступ по протоколу Thrift. Для пользовательского взаимодействия с системой предназначена командная оболочка HBase Shell, поддерживающая SQL-подобный язык HQL. Также HBase предоставляет Web-интерфейс, который позволяет пользователям просматривать информацию о хранимых таблицах и системе.

Сравнительная характеристика указанных выше хранилищ данных приведена в таблице 2.

Таблица 2

Сравнительная характеристика РХСД BigTable и HBase

Характеристика	BigTable	HBase
Общая структура (схема «главный-подчинённый»)	Кластер состоит из главного сервера (master) и множества таблет-серверов (tablet server), которые могут динамически добавляться или удаляться из кластера	Кластер состоит из главного сервера (master), множества регионов (region, аналог – tablet), и обслуживающих их регион-серверов (RegionServer)
Отказоустойчивость	Продолжение обслуживания клиентов при потере соединения таблет-сервера с главным сервером	Завершение работы и перезапуск регион-сервера при потере соединения с главным сервером
Модель хранения данных	Модель разреженной (column-oriented), многомерной, сортированной хэш-таблицы. Каждое значение, хранимое в хэш-таблице, индексируется с помощью ключа строки, ключа столбца и времени	
Реализация	На языке C++	На языке Java
Пользовательские интерфейсы	Клиентская библиотека, реализующая API	Java API, REST-интерфейс и доступ по протоколу Thrift, Web-интерфейс, для просмотра информации о системе и хранимых таблицах пользователями
SQL-диалекты	Не поддерживается	Взаимодействие с системой через командную оболочку HBase Shell, которая поддерживает SQL-подобный язык HQL
Транзакции	Не осуществляется поддержка	

Изначально описанные технологии создавались для использования в Web-сфере, которая насчитывает десятки миллиардов страниц. Роботы поисковой системы круглосуточно загружают петабайты данных с содержанием новых и измененных Web-страниц. В настоящее время появляется все больше Web-приложений, накапливающих и фильтрующих большие объемы информации, таких как социальные сети и различные сервисы агрегации. Использование реляционных баз данных в таких приложениях уже не актуально.

Кроме того, такие хранилища данных могут использоваться и в обычных программных средствах, использующих базы данных. Теперь нет монополизма реляционных баз данных, как безальтернативного источника данных. Все чаще архитекторы выбирают хранилище исходя из природы самих данных и того, как мы ими хотим манипулировать, какие объемы информации ожидаются.

Заключение. Таким образом, рассмотренные в докладе РФС имеют как схожие характеристики, так и существенные различия. HDFS обладает рядом недостатков по сравнению с GFS. Это восстановление после отказов, запись и редактирование файлов, а также отсутствие создания копий. При этом GFS не может быть использована повсеместно из-за закрытости разработки компании Google. Что касается рассмотренных распределённых хранилищ, то они не имеют существенных различий в структуре и модели хранения данных, так как HBase изначально является доступной альтернативой BigTable, реализованной на Java. При этом присутствуют расхождения в реализации механизма восстановления после отказов, который лучше реализован в BigTable. Поддержка SQL-диалектов имеет место только в HBase, что очень важно, так как информационная поддержка серверной логики систем базируется на использовании интерпретируемых языков T-SQL и PL/SQL. Общим недостатков является отсутствие поддержки транзакций.

Список литературы

[1] Распределённая файловая система GFS [Электронный ресурс]. − Режим доступа: https://habrahabr.ru/post/73673/.

[2] HDFS Architecture Guide [Электронный ресурс]. − Режим доступа: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[3] Реферат на тему «Распределенные базы и хранилища данных» [Электронный ресурс]. − Режим доступа: http://bibliofond.ru/view.aspx?id=656738#1.

[4] Распределённое хранилище Google Bigtable: не SQL единым… [Электронный ресурс]. − Режим доступа: http://www.computerra.ru/83598/bigtable/.

[5] Apache HBase™ Reference Guide [Электронный ресурс]. − Режим доступа: http://hbase.apache.org/book.html.

DISTRIBUTED FILE SYSTEMS FOR STRUCTURED DATA STORES ORGANIZATION

I.N. TSYRELCHUK, PhD
Dean of Faculty of Continuous and Distance Learning of BSUIR, Associate Professor

E. N. SHNEIDEROV

P. A. BERASHEVICH

N. A. LOS

A. S. TERESHKOVA

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: tsyrelchuk@bsuir.by, shneiderov@bsuir.by

Abstract. Technologies that allow to process and store data efficiently play a significant role nowadays. In this article, there is a comparison of new information systems oriented to store large amounts of data in distributed cluster systems. Conventionally, they can be divided into two classes: distributed file systems and distributed storages of structured data.

Key words: Distributed file systems, distributed storages of structured data, Google File System, Hadoop Distributed File Sys-tem, BigTable, HBase