УДК 004.65: 004.9

Н. Я. Павич, О. П. Крохмальна
Національний університет “Львівська політехніка”,
кафедра програмного забезпечення

ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ОПРАЦЮВАННЯ ДАНИХ ВЕЛИКИХ ОБСЯГІВ ТЕХНОЛОГІЯМИ SPARK ТА HIVE

Проаналізовано технології опрацювання даних великих обсягів. Розроблено програмне рішення на кластері Hadoop та отримано результати порівняння ефективності опрацювання даних великих обсягів технологіями Spark та Hive за часовими характеристиками і форматами даних. Запропоновано підходи до реалізації програмних систем для опрацювання даних великих обсягів технологіями Spark та Hive.

Ключові слова: дані великих обсягів, кластер Hadoop, технологія Spark, технологія Hive.

ASSESSMENT OF BIG DATA PROCESSING EFFICIENCY WITH SPARK AND HIVE TECHNOLOGY

In this paper the contemporary technology to big data processing is analyzed. The software solution on Hadoop is developed. And the comparative results of the time efficiency in big data processing with Spark or Hive are described. The approaches to implement the software systems for big data processing with Spark or Hive are suggested.

Key words: Big Data, Hadoop, Spark, Hive.

Вступ

Сьогодні спостерігається зацікавлення технологіями класу big data, яке пов’язане з постійним зростанням обсягів даних, з якими доводиться працювати великим компаніям. Великим організаціям важливо не лише накопичувати дані, а й ефективно обробляти їх та отримувати від них користь, ухвалюючи правильні рішення.

Проблеми ефективної обробки великих обсягів інформації стосується багато наукових досліджень, на основі їх результатів побудовано чимало програмних рішень та технологій, таких як Hadoop, MapReduce, Spark, Hive, Pig та багато інших. Через різноманітність технологій для опрацювання даних великих обсягів виникає проблема ефективного вибору засобів опрацювання для розв’язання конкретної задачі. Важливими є такі критерії, як час виконання аналізу даних, ресурсозатратність, складність поставленого завдання, формати вхідних та вихідних даних. Все це потребує детального аналізу інструментарію для опрацювання даних великих обсягів та вибору оптимальної технології. Отже, дослідження ефективності опрацювання даних великих обсягів є актуальною задачею.

Стан проблеми

Із швидким накопиченням інформації з різноманітних джерел, таких як фінанси, коментарі у соціальних мережах, медіа, прогнози погоди тощо, постала проблема ефективного опрацювання даних великих обсягів.

Використання Big Data дозволяє опрацьовувати на звичайних комп'ютерах, об'єднуючи їх у кластери [1], масиви інформації великих обсягів. Також застосування технологій цього класу дає змогу працювати із неструктурованими даним, що забезпечує ефективний аналіз інформації.

Існує багато платформ та технологій для такого аналізу інформації, всіх їх об'єднує можливість розподіленого опрацювання даних великих обсягів, які є неструктурованими. Проте найпопулярнішою та найчастіше використовуваною є платформа Apache Hadoop. Проект Apache Hadoop розробляє програмне забезпечення з відкритим вихідним кодом для надійних, масштабованих, розподілених обчислень [2].

Основою програмного забезпечення Apache Hadoop є фреймфорк, що дає змогу розподіляти опрацювання великих масивів даних на кластерах комп'ютерів з використанням простих моделей програмування. Кластер забезпечує розширення обчислювальної системи від одиничних серверів до тисячі машин, кожна з яких забезпечує місцеве опрацювання та зберігання наборів даних. Дані, розміщені на кластері Hadoop, опрацьовують за допомогою серії підходів та технологій, серед яких доцільно виділити Spark та Hive [2].

Високоефективною технологією компанії Apache для опрацювання даних великих обсягів є технологія Spark [3]. Для цієї технології характерним є те, що вона може опрацьовувати як дані на кластері Hadoop, так і дані, розміщені локально на комп'ютері або ж у будь-якому розподіленому сховищі, до якого є JDBC конектор.

Інша технологія Hive є рідною для Hadoop [4], дає змогу опрацьовувати дані, розміщені на кластері, за допомогою SQL-подібних запитів. Hive працює тільки з даними на Hadoop кластері та є доволі простою для розуміння та використання.

Технології Hive та Spark набувають все більшої популярності завдяки зручності їх використання для опрацювання даних великих обсягів, адже Spark дозволяє швидко опрацьовувати дані, а Hive є зручною технологією для роботи бізнес-аналітиків, які мають досвід роботи з реляційними сховищами даних.

Результати дослідження та порівняння ефективності застосування технологій Hive та Spark будуть корисні менеджерам проектів для вибору оптимальної з технологій для задач певного класу.

Постановка задачі

Описати підходи до опрацювання даних технологіями Hive та Spark для визначення найефективнішої роботи кожної із них. Проаналізувати результати реалізації програмних систем та здійснити їх порівняння, на основі чого можна буде дати рекомендації щодо вибору оптимальної технології для вирішення конкретної проблеми.

Метою дослідження є порівняння та аналіз таких технологій опрацювання даних великих обсягів, як Hive та Spark, для отримання інформації про особливості застосування цих технологій до наборів даних різних обсягів, форматів. Результати такого аналізу повинні допомогти менеджерам програмних проектів вибирати оптимальну технологію для опрацювання Big Data з погляду швидкості опрацювання інформації, повноти такого опрацювання та ресурсозатратності.

Основні завдання дослідження такі:

– Аналіз опрацювання даних технологією Spark та Hive. Визначення часу виконання аналізу даних із Spark та Hive залежно від обсягу, над яким здійснюється дослідження. Аналіз затрат ресурсів для використання цих технологій. Виокремлення класів задач, які можна дослідити за допомогою технології Spark чи Hive.

– Аналіз застосування обчислювального ядра Spark для виконання Hive запитів.

– Узагальнення та аналіз отриманих результатів, побудова на їх основі рекомендацій щодо ефективності використання вищезгаданих технологій.

Технології Hive та Spark сьогодні активно використовуються та набувають все більшої популярності, вони мають як спільні властивості, так і принципові відмінності. Обидві технології призначені для розподіленого опрацювання даних великих обсягів, проте в їх основу покладено зовсім різні методи опрацювання цих даних. Тому здійснення порівняльного аналізу цих технологій для визначення переваг та недоліків кожної з них є доцільним.

Основні результати досліджень

Технології Spark та Hive є одними з із найперспективніших, тому доцільно проаналізувати їхні основні особливості та характеристики.

Spark – це високопродуктивний двигун для опрацювання даних у кластері Hadoop [5]. Двигун може виконуватися на вузлах кластера Hadoop як за допомогою Hadoop YARN, так і у відокремленому режимі. Підтримується опрацювання даних у сховищах HDFS, HBase, Cassandra, Hive і у будь-якому форматі введення Hadoop (InputFormat).

Spark – це щільно розподілений набір даних (Resilient Distributed Dataset – RDD) [6]. Над RDD можна виконувати операції двох типів: трансформації та дії.

Операції перетворення будують новий набір даних після кожного їх застосування над попереднім набором даних, а дії розраховують результат виконання певних операцій над набором даних, і або повертають його програмі-драйверу, або зберігають у певному зовнішньому сховищі даних.

Результатом виконання операції трансформації над набором даних буде новий набір даних. Як правило, це операції, які змінюють набір даних. Серед них виділимо такі [3]:

– .map(function) – застосовує функцію function до кожного елемента набору даних;

– .filter(function) – повертає всі елементи набору даних, на яких функція function повернула значення “істина”;

– .distinct([numTasks]) – повертає набір даних, який містить унікальні елементи вихідного набору даних.

Також варто виділити такі операції над множинами [3]:

– .union(otherDataset) – об’єднання набору даних;

– .intersection(otherDataset) – перетин набору даних;

– .cartesian(otherDataset) – новий набір даних містить в собі усі можливі пари (A, B), де перший елемент належить вихідному набору даних, а другий – набору даних-аргументу.

Дії застосовуються тоді, коли необхідно матеріалізувати результат – як правило, зберегти на диск або вивести у консоль. До найпоширеніших дій, які можна виконувати над набором даних, належать [3]:

– .saveAsTextFile(path) – зберігає дані у текстовий файл (в hdfs, на локальну машину або у будь-яку іншу файлову систему, яка підтримується);

– .collect() – повертає елементи набору даних у вигляді масиву. Як правило, функція застосовується у випадку, коли даних у наборі небагато (після застосування фільтрів і перетворень і коли необхідна візуалізація або додатковий аналіз даних);

– .take(n) – повертає у вигляді масиву перші n елементів набору даних;

– .count()– повертає кількість елементів у наборі даних;

– .reduce(function) – із механізму цієї операції випливає, що функція function (яка приймає на вхід два аргументи і повертає одне значення) обов’язково повинна бути комутативною і асоціативною.

Драйвером є процес, який запускає функцію main() у програмі. Цей процес запускає код користувача, який створює SparkContext, RDD і виконує перетворення та дії. Як тільки драйвер зупиняється, додаток завершує роботу. Коли драйвер запущений, він виконує дві функції [5]:

– Перетворює програму користувача на завдання. Spark-драйвер відповідає за перетворення програми на виконувані модулі, що називаються задачами. Загалом всі Spark програми наслідують одну структуру – створюють щільний розподілений набір даних з певного набору вхідної інформації, отримують нові набори даних за допомогою перетворень, а також виконують певні дії над цими наборами даних. Програма Spark неявно створює логічний ациклічний граф операцій DAG. Коли драйвер запущений, він перетворює цей логічний граф на фізичний план виконання. Spark виконує декілька оптимізацій, таких як «конвеєрне» об’єднання трансформацій, а також виконує перетворення графу виконання на набір етапів. Кожен етап, своєю чергою, складається з декількох завдань. Завдання готуються для відправлення на кластер. Завдання є найменшою одиницею виконання у Spark, типова програма може запустити сотні або тисячі завдань.

– Планування і розподіл завдань виконавцям (executors). Враховуючи фізичний план виконання, драйвер повинен координувати планування індивідуальних завдань виконавцям. Коли

виконавці починають працювати, вони реєструються драйверу, тому він має інформацію про дії виконавців протягом всього проміжку часу. Кожен виконавець являє собою процес, здатний виконувати завдання і зберігати дані RDD. Драйвер відстежує поточний набір виконавців і розпланову є кожне завдання у відповідному місці на основі розміщення даних. Драйвер також відстежує місце зберігання керованих даних і використовує цю інформацію для планування майбутніх завдань, які звертаються до цих даних. Драйвер надає інформацію про додаток Spark, який працює за допомогою веб-інтерфейсу.

Виконавці (Spark Executors) [5] є робочими процесами, що відповідають за функціонування окремих завдань у Spark Job. Виконавці запускаються один раз під час запуску додатка, і зазвичай працюють протягом всього життєвого циклу додатка. Виконавці виконують дві основні функції – по-перше, вони виконують завдання, які їм призначає драйвер, і повертають йому результат, а по-друге, забезпечують зберігання у пам'яті наборів даних, що кешуються користувацькими програмами, за допомогою служби Block Manager, яку містить кожен виконавець. Оскільки дані кешуються безпосередньо всередині виконавців, завдання можна виконувати на кешованих розподілених наборах даних.

До основних особливостей технології ***Hive*** можна зарахувати [4]:

– Використання індексації для прискорення опрацювання наборів даних. Використовуються такі типи індексів, як Compaction та Bitmap.

– Зберігання інформації різних типів.

– Зберігання метаданих у СУБД, що значно знижує час на виконання семантичних перевірок під час виконання запитів.

– Опрацювання архівованих даних, розміщених в екосистемі Hadoop.

– Вбудовані користувацькі функції (UDF) для опрацювання даних та інші інструменти опрацювання інформації. Hive підтримує розширення набору UDF для випадків, які не підтримуються стандартним UDF.

– SQL-подібні запити, які неявно перетворюються на серію MapReduce, Tez або Spark команд.

Сьогодні Hive є успішною технологією, яку використовують різні компанії як добре масштабовану, широкого призначення платформу.

Як правило, Hive працює на робочій станції та перетворює SQL-подібні запити на серію операцій для виконання в кластері Hadoop [7]. Hive організовує дані в таблиці, які дозволяють структурувати дані в HDFS. Метадані цих таблиць зберігаються в базі даних, яка називається Metastore. Hive використовує просту SQL-подібну мову HiveQL, що дає змогу користувачам, які працюють з SQL, швидко та ефективно опрацьовувати інформацію, яка зберігається на кластері Hadoop.

Розглянемо детальніше принципи опрацювання даних із технологією Hive. У традиційних базах даних перевірка схеми відбувається на етапі завантаження даних. У Hive дані не перевіряються на етапі завантаження, а під час виконання запитів. Це називається перевіркою схеми на читання.

Оновлення, транзакції та індекси є важливою частиною традиційних баз даних, проте у Hive вони не впроваджені й не підтримуються [4], оскільки ця технологія у своїх обчисленнях використовує парадигму MapReduce для опрацювання даних на HDFS, і сканування таблиці повністю для пошуку потрібної інформації в цьому випадку є нормою, а оновлення таблиць здійснюється за допомогою перетворень даних у нових таблицях. Для зберігання даних додатка, що опрацьовує великі масиві інформації, цей підхід є оптимальним.

Отже, з з аналізу технологій Spark та Hive можна зробити висновок, що Hive є традиційною технологією екосистеми Hadoop, і використовує як основу парадигму MapReduce, а Spark уможливлює опрацювання даних – як розміщених в екосистемі Hadoop, так і за її межами, за допомогою підключення JDBC.

Оцінювання ефективності опрацювання даних великих обсягів *виконано з розгортанням програмних рішень на кластері Hadoop. Для дослідження технологій розроблено дві програмні системи, реалізовані технологіями Hive та Spark відповідно, які використовуються для опрацювання даних із єдиного сховища, розміщеного на файловій системі HDFS.*

Розроблене програмне рішення для опрацювання даних з використанням технологій Spark та Hive (рис. 1) забезпечує такі функціональні можливості:

– завантаження нових даних;
– опрацювання даних різних форматів;
– збереження результатів опрацювання даних у розподіленій файловій системі HDFS.

Для розгортання програмного рішення вибрано дистрибутив Hortonworks [7], який є ефективним програмним інструментом організації кластера Hadoop. Hortonworks Sandbox [2] є особистим, портативним Hadoop-середовищем, яке орієнтоване на роботу з продуктами Apache, до яких можна зарахувати Spark та Hive.

Для зберігання та організації даних у кластері використано розподілену файлову систему HDFS (Hadoop Distributed File System) [2]. HDFS влаштована так, щоб забезпечити максимальну продуктивність потокового доступу до даних. До того ж структури файлової системи оптимізовані для роботи з великими файлами.

Набори даних, для який здійснюється аналіз, завантажені із сайта Американської асоціації статистики [8]. Вхідні дані розміщено на двох вузлах кластера на розподіленій файловій системі HDFS (рис. 1). Основним вузлом є вузол, що містить NameNode. Також цей вузол містить і DataNode, тобто на ньому здійснюється управління кластером і опрацьовуються дані. Також основний вузол містить клієнта Hive, тобто на ньому відбувається написання та запуск Hive-задач. На другому вузлі кластера здійснюється лише зберігання та опрацювання даних.

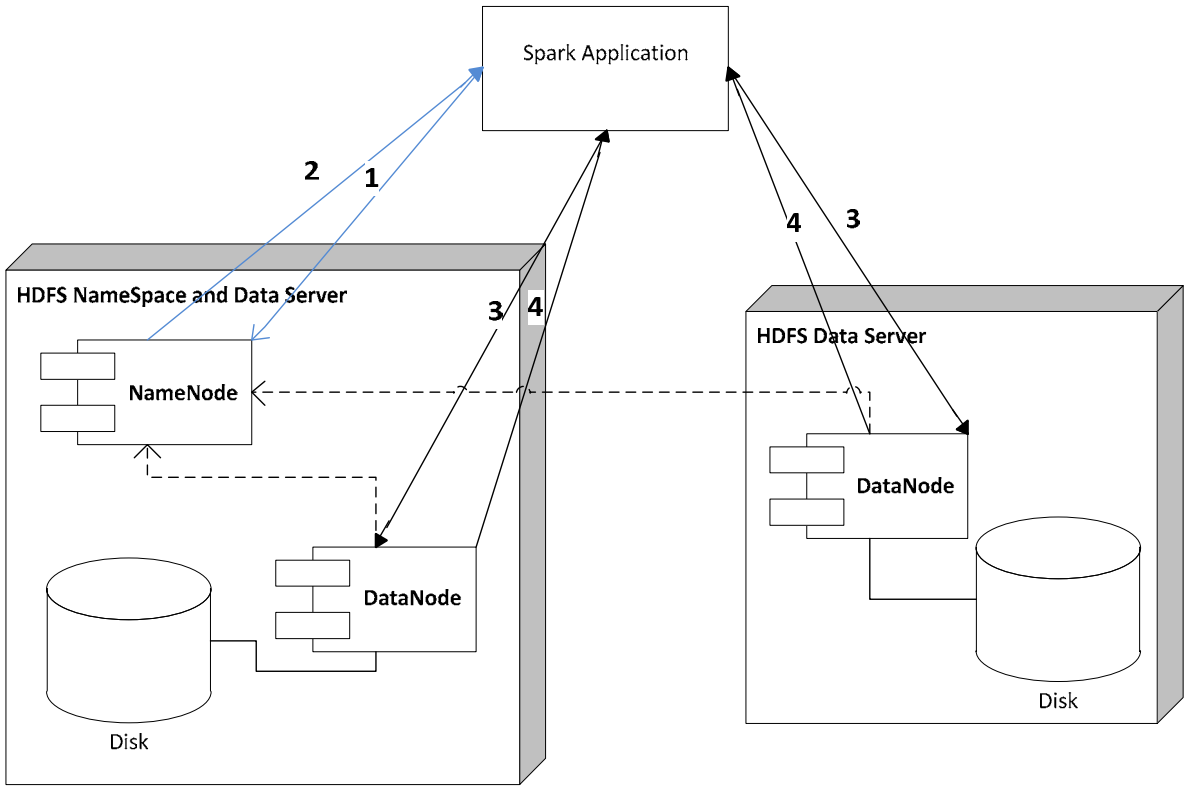

Рис. 1. Діаграма компонентів програмного рішення із використанням технологій Spark і Hive

Для кожної з технологій проаналізовано документацію для детального вивчення основних переваг та недоліків застосування такого підходу опрацювання інформації до задач певного класу.

Визначено, з якими платформами та для даних яких форматів найкраще використовувати кожну з технологій.

Платформи, на яких можуть зберігатися дані, що опрацьовуються технологіями Hive та Spark, відображено у табл. 1.

Таблиця 1

Порівняння платформ, на яких здійснюється опрацювання даних

Платформа	Технологія	
	Spark	Hive
Hadoop HDFS	+	+
Hadoop Hive Metastore	–	+
Hortonworks Sandbox	+	+
Syncfusion Big Data Platform	+	–
Cassandra	+	–
Cloudera Manager	+	+
Mesos	+	–
S3	+	–
EC2	+	–
Standalone mode	+	–

З табл. 1 можна зробити висновок, що Spark є технологією, яка може опрацьовувати дані на ширшому спектрі платформ, ніж технологія Hive. Оскільки Hive є технологією, побудованою над екосистемою Hadoop, вона може опрацьовувати дані, що зберігаються лише в Hadoop, а саме в розподіленій файловій системі HDFS або HBase – нереляційній розподіленій базі даних, реалізованій в екосистемі. Також Hive працює з такими дистрибутивами Hadoop, як Hortonworks Sandbox та Cloudera Manager. Spark, своєю чергою, є технологією, не прив'язаною до конкретної платформи опрацювання інформації, і може опрацьовувати дані, що зберігаються як на Hadoop-кластері, так і у таких розподілених системах зберігання інформації, як Mesos, S3 чи EC2. Однією із переваг Spark є можливість запускати його у локальному режимі, що дає змогу швидко перевіряти програмну систему на основі цієї технології ще на стадії розроблення.

Для порівняння швидкості та ефективності опрацювання даних, які містяться у файловій системі HDFS та у Hive Metastore, здійснено аналіз даних, що зберігаються у кожній із згаданих систем зберігання даних.

Опрацювання 657,5 MB даних, що зберігаються у сховищі Hive, тривало 2 хв і 59 с. А опрацювання запиту для опрацювання даних з таким самим обсягом (657,5 MB), що зберігаються у розподіленій файловій системі HDFS, тривало 5 хв і 3 с.

Отже, можна зробити висновок, що ***для ефективного застосування технології Hive для опрацювання даних доцільно попередньо завантажувати дані у метасховище***. Це дасть змогу пришвидшити процес аналізу інформації до двох раз. Під час аналізу даних, що зберігаються у файлі на HDFS, Hive структурує їх вже безпосередньо під час опрацювання і зберігає у тимчасовій зовнішній таблиці, що істотно сповільнює процес опрацювання інформації.

У результаті дослідження можна стверджувати, що технології Spark та Hive аналізують дані різних форматів (табл. 2).

Таблиця 2

Порівняння форматів даних, що опрацьовуються технологіями Hive та Spark

Формати	Технологія	
	Spark	Hive
sequence file	+	+
text file	+	+
RCFILE	+	+
CSV file	+	+
Avro	+	–
Parquet	+	+
JSON	+	–
Реляційні дані в сховищі Hive	+	+

З табл. 2 можна зробити висновок, що і для технології Hive, і для Spark можна опрацьовувати інформацію, яка міститься у файлах таких форматів, як sequence, text, Parquet, CSV та інші. Проте у разі використання технології Hive, навіть у випадку опрацювання зовнішнього файла, відбудеться структуризація такого файла, і дані з нього збережуться у зовнішній таблиці, до якої будуть звертатися запити, написані мовою HiveQL для аналізу інформації. Spark, своєю чергою, може опрацьовувати дані, що зберігаються у сховищі Hive, проте ця технологія працюватиме з такими даними як з простими неструктурованими файлами.

Досліджено ***час опрацювання даних різних обсягів*** із технологіями Hive та Spark. Опрацювання даних у Spark здійснювалося мовою Scala із використанням бібліотек SparkContext та RDD [5]. Час опрацювання даних різного обсягу подано у табл. 3.

Таблиця 3

Час опрацювання даних із Spark залежно від обсягу даних

Обсяг даних, МВ	Час виконання
639	33 с
1279	51 с
1949	1 хв 9 с
2606	1 хв 35 с

Графік залежності часу опрацювання даних технологією Spark залежно від обсягу інформації, аналіз якої здійснюється, зображено на рис. 2.

Рис. 2. Графік залежності часу опрацювання даних (секунди) від обсягу інформації (мегабайти) технологією Spark

Час опрацювання даних із використанням технології Hive визначили, запустивши на наборах даних різного обсягу розроблене програмне рішення для опрацювання даних за допомогою Hive (рис. 1). Мова Hive для опрацювання даних великих обсягів використовує синтаксис, подібний до Transact-SQL, що робить її простою для сприйняття та зручною у використанні, а також забезпечує можливість роботи із даними, які мають реляційну будову, хоч і орієнтована на обробку даних на кластері. Результати аналізу даних із Hive подано у табл. 4.

Таблиця 4

Час опрацювання даних із Hive залежно від обсягу даних

Обсяг даних, МВ	Час виконання
639	4 хв 43 с
1279	7 хв 59 с
1949	11 хв 33 с
2606	15 хв 21 с

Графік залежності часу опрацювання даних технологією Hive залежно від обсягу інформації зображено на рис. 3. За наведеними графіками можна визначити час опрацювання даних для різних обсягів інформації.

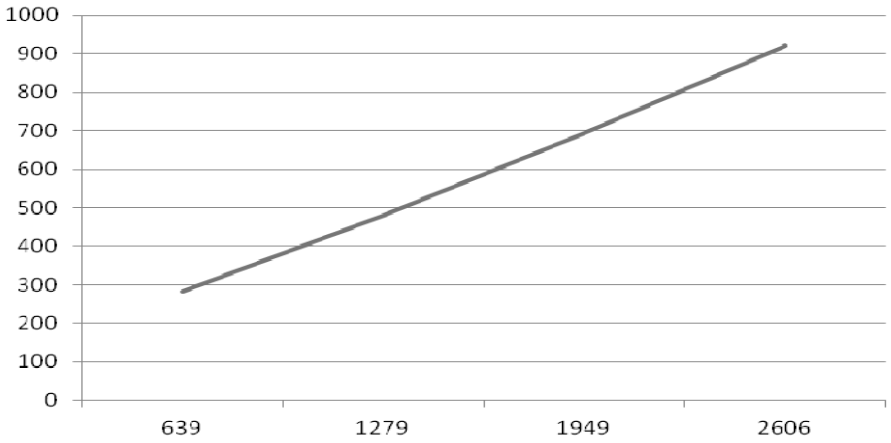

Рис. 3. Графік залежності часу опрацювання даних (секунди) від обсягу інформації (мегабайти) технологією Hive

Отже, з наведених вище таблиць і графіків можна зробити висновок, що Spark є технологією, яка у 15 разів швидше опрацьовує дані, що зберігаються на диску. Проте Hive має істотні переваги, передусім це простота та зручність використання. Обсяг коду, написаного на Hive, для однотипних задач аналізу даних у десятки разів менший, ніж обсяг аналогічного коду, реалізованого за допомогою технології Spark.

Висновки

Кожна з технологій Hive та Spark має характерні особливості, які дають змогу її оптимально використовувати для розв'язання певного класу задач.

Система аналізу даних, побудована на технології Spark, потребує великих затрат ресурсів. Для її запуску навіть на одному вузлі потрібно щонайменше 16 Гб оперативної пам'яті. Проте така система є доволі універсальним засобом аналізу інформації великих обсягів, адже дозволяє опрацьовувати дані, розміщені на різноманітних платформах зберігання даних, таких як Hadoop, Cassandra, Mesos, S3 та багато інших, тоді як система для аналізу за допомогою Hive опрацьовує лише дані на кластері Hadoop і не призначена для опрацювання даних за його межами.

Hive перед опрацюванням даних приводить їх до певного структурованого вигляду, що дає змогу користувачам, знайомим із засобами опрацювання даних у реляційних базах, легко опрацьовувати дані великих обсягів. Spark, своєю чергою, є оптимальною технологією для розробників, знайомих із такими мовами програмування, як Scala, Java та Python.

Для швидкого аналізу потокових даних у режимі реального часу доцільно використовувати Spark. Для опрацювання даних із меншими затратами обчислювальних і часових ресурсів та таких, що мають реляційну модель, ефективним є Hive.

1. Karen Montgomery. Big Data Now: 2014 Edition. O'Reilly Media. – Junuary, 2015 – 165 p. 2. White T. Hadoop: The Definitive Guide, 4th Edition. O'Reilly Media. – March, 2015 – 756 p. 3. Holden Karau, Andy Konwinski, Patrick Wendell, Matei Zaharia. Learning Spark. O'Reilly Media. – February 2015 – 276 p. 4. Edward Capriolo, Dean Wampler, Jason Rutherglen. Programming Hive. O'Reilly Media. – September, 2014. – 365 p. 5. Karau, H. Fast Data Processing With Spark. – Packt Publishing, 2013. – 120 p. 6. Gonzalez Joseph, Xin Reynold, Dave Ankur, Crankshaw Daniel, Franklin Michael, Stoica Ion (Oct 2014). GraphX: Graph Processing in a Distributed Dataflow Framework. 7. How to Process Data with Apache Hive [Електронний ресурс] // Hortonworks: сайт. – Режим доступу: http://hortonworks.com/hadoop-tutorial/how-to-process-data-with-apache-hive/ 8. Bi-Annual Data Exposition. [Електронний ресурс] // Statistical Computing: сайт. – Режим доступу: http://stat-computing.org/dataexpo/