December 2022

METHOD AND SYSTEM FOR PROCESSING QUERIES WITH QUERY FILTERS ON A SERVER

MOHANKUMAR RAMACHANDRAN
VISA

TITLE: “METHOD AND SYSTEM FOR PROCESSING QUERIES WITH QUERY FILTERS ON A SERVER”

VISA

MOHANKUMAR RAMACHANDRAN

TECHNICAL FIELD

The present subject matter relates to a field of data analytics and data storage on servers, more particularly, but not exclusively to a system and method for processing queries with query filters.

BACKGROUND

Every day, several quintillion bytes of data may be created around the world. These data come from everywhere: posts to social media sites, digital pictures and videos, purchase transaction records, bank transactions, sensors used to gather data and intelligence such as climate information, cell phone GPS signals, and many others. This type of data and its vast accumulation is often referred to as “big data.” Big data is managed and analysed using an Apache Spark® or Hadoop software® framework. Apache Spark® is an open-source cluster computing framework that provides distributed task dispatching, scheduling, and basic functionality. Spark® revolves around a concept of a Resilient Distributed Dataset (RDD), which is a fault-tolerant collection of elements that can be operated in parallel. RDDs are fundamental data structures of Spark® and are immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster.

One of the major use cases of spark® is to work as big data processing engine for Business Intelligence (BI) tools. When Spark® is used with BI, the queries processed by spark® is only in terms of “read” with or without filters and not “write.” When Spark® processes a read query, it reads every single source file and then processes it every single time. This is in efficient in cases where the filter condition is filtering out significant number of records, but still spark® must process every single file for every query irrespective of the filter. Therefore, in the existing techniques, Spark® reads every single file for a given query irrespective of whether a particular location data is present in a file. For example, one million files create one million RDDs (assuming 128 mb files and fits max partition bytes), then one million CPU cores would be required to process the query reading stage.

Further, the existing systems allow to read all the files in case of flat files, and in case of other formats such as Optimized Row Columnar (ORC) format or Parquet format, a task is generated and then the task checks the file's inbuilt metadata to filter out. There is no optimization in case

of flat files, whereas in terms of compressed formats like ORC and Parquet, the task which does not satisfy a given filter gets completed faster, but the major drawback is that a task must be generated first and then the metadata of the files are filtered out. Therefore, the procedure followed for compressed formats lacks the ability to process the data faster as it relies on the generation of task which is time consuming.

The information disclosed in this background of the disclosure section is only for enhancement of understanding of the general background of the disclosure and should not be taken as an acknowledgement or any form of suggestion that this information forms prior art already known to a person skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, serve to explain the disclosed principles. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Some embodiments of device or system and/or methods in accordance with embodiments of the present subject matter are now described, by way of example only, and with reference to the accompanying figures, in which:

Figure 1a illustrates an exemplary environment for processing queries with query filters on a server;

Figure 1b illustrates a detailed block diagram of a server including a system for processing queries with query filters; and

Figure 2 shows a flow diagram that illustrates a method of processing queries with query filters on a server.

The figures depict embodiments of the disclosure for purposes of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the disclosure described herein.

DESCRIPTION OF THE DISCLOSURE

In the present document, the word "exemplary" is used herein to mean "serving as an example, instance, or illustration." Any embodiment or implementation of the present subject matter described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

While the disclosure is susceptible to various modifications and alternative forms, specific embodiment thereof has been shown by way of example in the drawings and will be described in detail below. It should be understood, however that it is not intended to limit the disclosure to the particular forms disclosed, but on the contrary, the disclosure is to cover all modifications, equivalents, and alternative falling within the spirit and the scope of the disclosure.

The terms “comprises,” “comprising,” or any other variations thereof, are intended to cover a non-exclusive inclusion, such that a setup, device or method that comprises a list of components or steps does not include only those components or steps but may include other components or steps not expressly listed or inherent to such setup or device or method. In other words, one or more elements in a device or system or apparatus proceeded by “comprises… a” does not, without more constraints, preclude the existence of other elements or additional elements in the device or system or apparatus.

The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the invention(s)" unless expressly specified otherwise.

The terms "including", "comprising", “having” and variations thereof mean "including but not limited to", unless expressly specified otherwise.

The present disclosure discloses a method and a system for processing queries with query filters on a server. The method includes processing queries on large datasets by using resilient distributed datasets (RDDs) under Apache Spark® framework. The query filters are provided directly to the server for processing the query such that the step of generating a task is eliminated, thus reducing the cost of processing significantly.

Figure 1a illustrates an exemplary environment for processing queries with query filters on a server 101 in accordance with the present invention. The environment 100 comprises a user device 102 and a system 104.The system 104 is connected to the user device 102 for processing queries sent by the user device 102, the query may be a read query with a query filter. A person skilled in the art may understand that the system 104 may be connected to a plurality of user devices 104 for processing the queries with query filters. The system 104 is implemented on a server 101, the server may be an Apache Spark® server. The system 104 may include one or more processors 105, I/O interface 106, and a memory 107. In some embodiments, the memory 107 may be communicatively coupled to the one or more processor 105. The memory 107 stores instructions, executable by the one or more processor 105, which, on execution, may cause the system 104 to process the query with a query filter, as disclosed in the present disclosure. In an embodiment, the memory 107 may include one or more modules 108 and data 109. The one or more modules 108 may be configured to perform the steps of the present disclosure using the data 109, to process the query with a query filter. In an embodiment, each of the one or more modules 108 may be a hardware unit which may be present outside the memory 107 and coupled with the system 104. Further, the system 104 may comprise a plurality of files associated with a particular data. Fig.1b shows a detailed block diagram of the server along with the system 104. As shown in **Figure 1b,** the system 104 may comprise dataset107 stored in the data 109. The dataset 107a may include the plurality of files corresponding to a plurality of immutable resilient distributed datasets (RDDs). The system 101 may generate metadata for each of the plurality of RDDs when the corresponding file are read for the first time.

In an embodiment, each of the plurality of files may be read and accordingly the RDDs corresponding to each of the plurality of files may have the corresponding metadata stored against each RDD.

In another embodiment, one or more files of the plurality of files may be read and hence only the RDDs corresponding to the one or more files may have the corresponding metadata before processing the query sent by the user device 102.

In yet another embodiment, the plurality of files may be newly stored in the system 104 and hence may not have been read even once. Accordingly, the RDDs corresponding to the plurality of files may not include any metadata before processing the query sent by the user device 102.

Returning to Fig.1a, in an embodiment, the system 104 implemented on the server 101, and the user device 102 may communicate via the communication network 103, for processing the queries along with respective query filters. The communication network 103 may include, without limitation, a direct interconnection, Local Area Network (LAN), Wide Area Network (WAN), wireless network (e.g., using Wireless Application Protocol), the Internet, and the like. In an embodiment, the system 104 may be implemented in a server configured to process a read query with filter. In an embodiment, such server may be a dedicated server or a cloud-based server.

The system 104 may receive a query along with a query filter from the user device 102 for performing a read from one or more files of the plurality of files stored in the system 104. On receiving the query, the system 104 may analyse the query along with the query filter. As shown in **Fig. 1b**, the system 104 may perform a search across the available metadata of RDDs.

In an embodiment, the metadata may be available for each of the plurality of RDDs. Hence, the search is performed across metadata of each of the plurality of RDDs that are associated with respective files from the plurality of files to identify whether the query filter is within a threshold of each of the plurality of RDD's. The threshold of a given RDD corresponds to a range comprising minimum and maximum values of the given RDD. In an embodiment, the metadata comprises minimum and maximum values for each of the one or more RDDs.

In another embodiment, the metadata may be available only for one or more RDDs. Hence, the search is performed across the metadata associated with only the one or more RDDs of one or more files from the plurality of files to identify whether the query filter is within the threshold of the one or more RDDs. In case the available metadata of the one or more RDDs may not satisfy the query filter, the system 104 may perform a read of files other than the one or more files which may not have been read previously and create metadata for the same. Thus, in such cases, the metadata associated with the other files is considered for identifying whether the query filter is within the threshold of the one or more RDDs.

In yet another embodiment, metadata may not be available before processing the query sent by the user device 102. Hence, the metadata maybe be created on reading each of the plurality of files. Subsequent to which the metadata is considered for the search for identifying whether the query filter is within the threshold of the RDDs corresponding to the plurality of files.

Based on the search, the system 104 may identify at least one RDD that satisfies the query filter. In an embodiment, the number of RDDs satisfying the query filter may be less than or equal to the total number of RDDs. Finally, the query sent by the user device 102 may get executed to read the content from the one or more files which are associated with the at least one identified RDDs that satisfy the query filter. Therefore, the query sent by the user device 102 is executed based on the query filter that avoids reading the RDDs of files that do not satisfy the query filter. Thus, the cost and time of processing a query reduces as the number of cores required are equal to the number of RDDs.

Figure 2 illustrates flow diagram showing methods for processing queries with query filters on a server, in accordance with some embodiments of the present disclosure.

As illustrated in Fig. 2, the method 200 comprises one or more blocks illustrating a method for processing queries with query filters on a server in accordance with some embodiments of the present disclosure. The method 200 may be described in the general context of computer-executable instructions. Generally, computer-executable instructions can include routines, programs, objects, components, data structures, procedures, modules, and functions, which perform functions or implement abstract data types.

The order in which the method 200 is described is not intended to be construed as a limitation, and any number of the described method blocks can be combined in any order to implement the method 200. Additionally, individual blocks may be deleted from the methods without departing from scope of the subject matter described herein. Furthermore, the method 200 can be implemented in any suitable hardware, software, firmware, or combination thereof.

At block 201 of **Fig. 2,** the method may include receiving, by a system 104, a query with a query filter to be executed on the dataset 107a, wherein the datasets 107a comprise a plurality of files that correspond to a plurality of immutable resilient distributed datasets (RDDs). The query may correspond to accessing the plurality of files present in the dataset 107a and the

filter may correspond to a condition based on which the user device 102 accesses the plurality of files.

For example, the user initiates the following query:
SELECT dt, sum(sales_amount) FROM "<sale_files_folder>" WHERE loc = "xyz" GROUP BY dt
The above query reads files inside sale_files_folder and gives sum of sales amount by date filtered for some specific location.

At block 203 of **Fig. 2**, the method may include analysing, by the system 104, the received query and the query filter.

At block 205 of **Fig. 2**, the method may include performing, by the system 104, a search across available metadata of RDDs associated with files from a plurality of files to identify whether the query filter is within a threshold of the RDDs associated with the respective files. The metadata for the RDDs is created when the system 104 identifies a file associated with a given RDD being read for the first time, and wherein the threshold of a given RDD corresponds to a range comprising minimum and maximum values of the given RDD.

At block 207 of **Fig. 2**, the method may include identifying, by the system 104, at least one RDD satisfying the query filter based on the search. The number of identified RDDs may be less than or equal to the total number of RDDs.

At block 209 of **Fig. 2**, the method may include executing, by the system 104, the query to read the content from the one or more files associated with the at least one identified RDD.

Thus, by processing the query only on the identified RDDs that satisfy the query filter, instead of the all the RDDs corresponding to the files in the dataset 107a, the cost and time of processing a query reduces as the number of RDDs equal to number of cores required.

With respect to the use of substantially any plural and/or singular terms herein, those having skill in the art can translate from the plural to the singular and/or from the singular to the plural as is appropriate to the context and/or application. The various singular/plural permutations may be expressly set forth herein for sake of clarity.

Any of the software components or functions described in this application, may be implemented as software code to be executed by a processor using any suitable computer language such as, for example, Java, C++, or Perl using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions, or commands on a computer readable medium, such as a random-access memory (RAM), a read only memory (ROM), a magnetic medium such as a hard-drive or a floppy disk, or an optical medium such as a CD-ROM. Any such computer readable medium may reside on or within a single computational apparatus and may be present on or within different computational apparatuses within a system or network.

The above description is illustrative and is not restrictive. Many variations of the invention may become apparent to those skilled in the art upon review of the disclosure. The scope of the invention can, therefore, be determined not with reference to the above description, but instead can be determined with reference to the pending claims along with their full scope or equivalents.

One or more features from any embodiment may be combined with one or more features of any other embodiment without departing from the scope of the invention.

A recitation of "a", "an" or "the" is intended to mean "one or more" unless specifically indicated to the contrary.

All patents, patent applications, publications, and descriptions mentioned above are herein incorporated by reference in their entirety for all purposes. None is admitted being prior art.

Although the invention has been described in detail for the purpose of illustration based on what is currently considered to be the most practical and preferred embodiments, it is to be understood that such detail is solely for that purpose and that the invention is not limited to the disclosed embodiments, but, on the contrary, is intended to cover modifications and equivalent arrangements that are within the spirit and scope of the appended claims. For example, it is to be understood that the present invention contemplates that, to the extent possible, one or more features of any embodiment can be combined with one or more features of any other embodiment.

METHOD AND SYSTEM FOR PROCESSING QUERIERS WITH QUERY FILTERS ON A SERVER

ABSTRACT

The present disclosure provides a method and a system for processing queries with query filter on a server. The method includes receiving a query to be executed. The method analyses the received query and query filter. The method includes performing a search across metadata of Resilient Distributed Datasets (RDDs) associated with one or more files to identify whether the query filter is within a threshold of a given RDD. The method further identifies at least one RDD satisfying the query filter based on the search. Finally, the method includes executing the query to read the content from one or more files associated with the identified at least one RDD.

Figure 1a & 1b

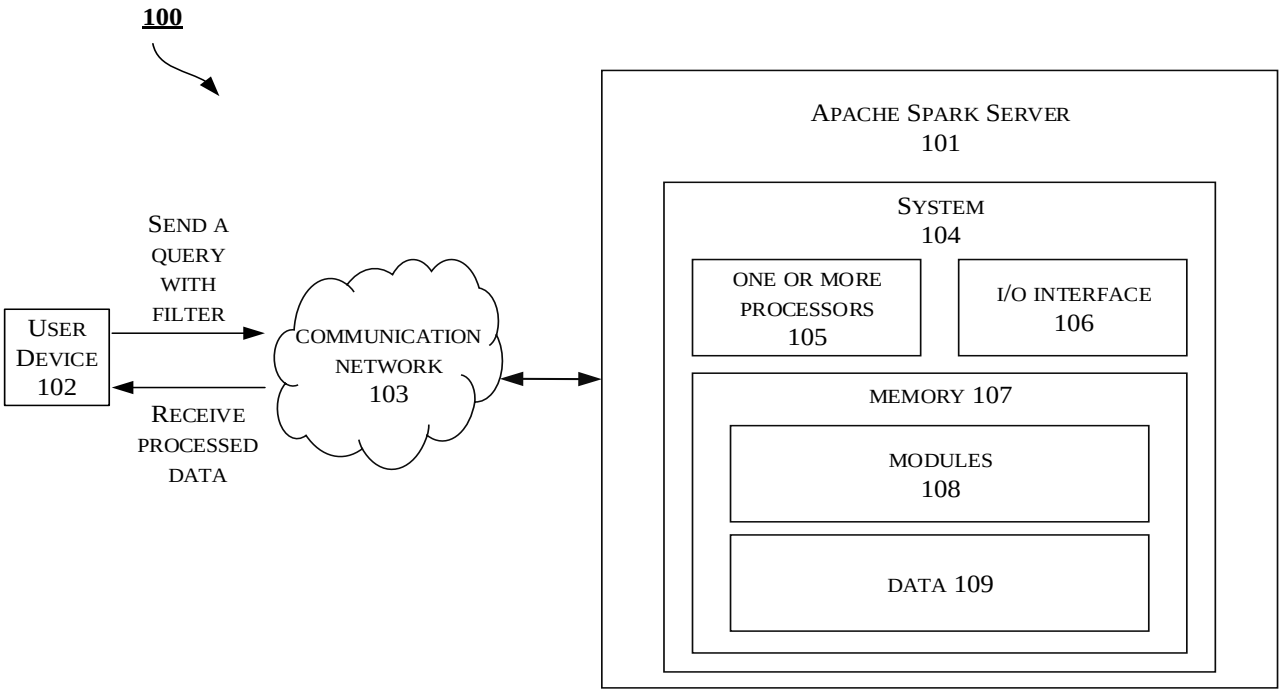
100
APACHE SPARK SERVER
101
SYSTEM
104
ONE OR MORE PROCESSORS
105
I/O INTERFACE
106
MEMORY 107
MODULES
108
DATA 109
SEND A QUERY WITH FILTER
USER DEVICE
102
COMMUNICATION NETWORK
103
RECEIVE PROCESSED DATA

Fig. 1a

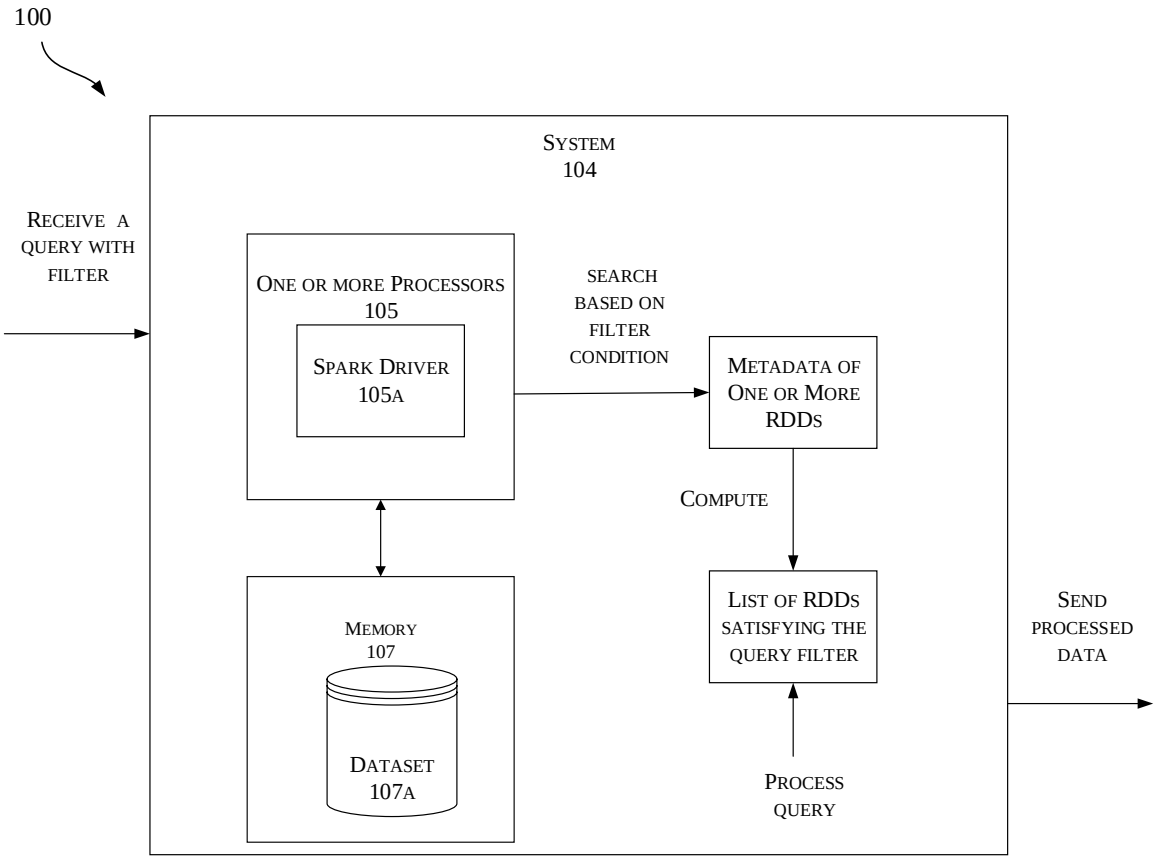
100
SYSTEM
104
RECEIVE A QUERY WITH FILTER
ONE OR MORE PROCESSORS
105
SPARK DRIVER
105A
SEARCH BASED ON FILTER CONDITION
METADATA OF ONE OR MORE RDDS
COMPUTE
MEMORY
107
DATASET
107A
LIST OF RDDS SATISFYING THE QUERY FILTER
SEND PROCESSED DATA
PROCESS QUERY

Fig. 1b

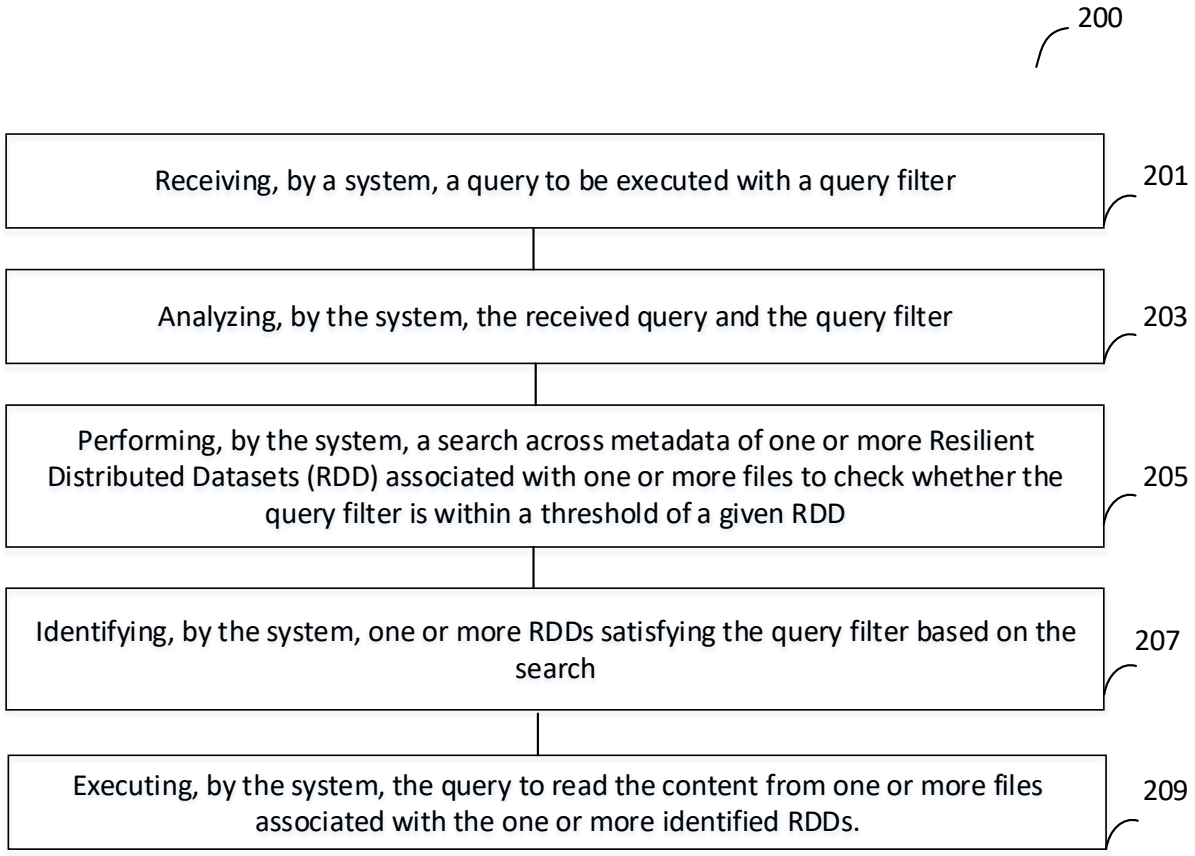
200
Receiving, by a system, a query to be executed with a query filter
201
Analyzing, by the system, the received query and the query filter
203
Performing, by the system, a search across metadata of one or more Resilient Distributed Datasets (RDD) associated with one or more files to check whether the query filter is within a threshold of a given RDD
205
Identifying, by the system, one or more RDDs satisfying the query filter based on the search
207
Executing, by the system, the query to read the content from one or more files associated with the one or more identified RDDs.
209

Fig. 2