(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 June 2018 (28.06.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/119116 A1

(51) International Patent Classification:
G06F 9/46 (2006.01)

(21) International Application Number:
PCT/US2017/067668

(22) International Filing Date:
20 December 2017 (20.12.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/385,833 20 December 2016 (20.12.2016) US

(71) Applicant: **STRIIM, INC.** [US/US]; 575 Middlefield Rd., Palo Alto, California 94301 (US).

(72) Inventors: **PAREEK, Alok**; 1231 Southdown Rd., Hillsborough, California 94010 (US). **NADIMPALLI, Vijayakrishna**; 1988 Bellomy St., Apt. 6, Santa Clara, California 95050 (US). **KHALADKAR, Bhushan**; 3716 Redwood Circle, Palo Alto, California 94306 (US). **AGARWAL, Manish**; 17804 NE 27th St., Redmond, Washington 98052 (US). **KEENE, Nicholas**; 485 Togstad Glenn, Madison, Wisconsin 53711 (US).

(74) Agent: **GALLIANI, William S.** et al.; Cooley LLP, Attn.: IP Docketing Department, 1299 Pennsylvania Avenue NW, Suite 700, Washington, District of Columbia 20004 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: DATA STREAM PROCESSOR WITH BOTH IN MEMORY AND PERSISTED MESSAGING

(57) Abstract: A machine has a processor and a memory connected to the processor. The memory stores instructions executed by the processor to receive data from a data source that continuously generates records, selectively identify conditions within the data that satisfy specified rules, and report the conditions prior to routing the data to a message system node forming a segment of a distributed transaction log with messages persisted in non-volatile memory and replicated within a cluster of nodes.

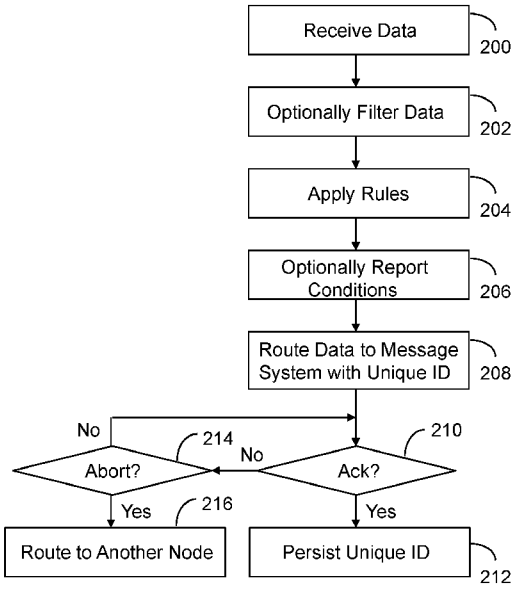


FIG. 2

WO 2018/119116 A1

[Continued on next page]

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

DATA STREAM PROCESSOR WITH BOTH IN MEMORY AND PERSISTED MESSAGING

CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority to U.S. Patent Application No. 15/385,833 filed December 20, 2016, the contents of which are incorporated herein by reference.

FIELD OF THE INVENTION

This invention relates generally to the processing and analysis of data. More particularly, this invention is directed toward a data stream processor that processes and reports events before optionally and automatically persisting data to a message system.

BACKGROUND OF THE INVENTION

As used herein, the term stream processor refers to a computing system that processes a data feed and selectively reports events that satisfy specified conditions. The stream processor is an in-memory processor, meaning it is operative in random memory on data as the data is "in flight". Examples of a data feed include output from a database log, an operating system log, an application log, a web server log, an application server log, a machine generated log, a sensor (e.g., an Internet of Things or IoT device), a social media feed, and the like. The stream processor processes the data feed before the data reaches its ultimate destination, where it may or may not be persisted from memory. The assignee of the current application discloses a stream processor in U.S. Serial Number 14/285,428, filed May 22, 2014, the contents of which are incorporated herein by reference.

It is desirable to expand the applications and uses of stream processors.

SUMMARY OF THE INVENTION

A machine has a processor and a memory connected to the processor. The memory stores instructions executed by the processor to receive data from a data source that continuously generates or publishes records, selectively identify conditions within the data that satisfy specified rules, and report the conditions prior to routing the data to a message system node forming a segment of a distributed transaction log with messages persisted in non-volatile memory and replicated within a cluster of nodes.

BRIEF DESCRIPTION OF THE FIGURES

The invention is more fully appreciated in connection with the following detailed description taken in conjunction with the accompanying drawings, in which:

FIGURE 1 illustrates a system configured in accordance with an embodiment of the invention.

FIGURE 2 illustrates stream processor operations performed in accordance with an embodiment of the invention.

FIGURE 3 illustrates message system processing performed in accordance with an embodiment of the invention.

FIGURE 4 is a user interface utilized in accordance with an embodiment of the invention.

Like reference numerals refer to corresponding parts throughout the several views of the drawings.

DETAILED DESCRIPTION OF THE INVENTION

Figure 1 illustrates a system 100 configured in accordance with an embodiment of the invention. The system 100 includes a set of data sources 102_1 through 102_N. Each data source continuously generates records (a data feed) that are routed to a server 104 or to servers 130_1 through 130_N via a network 106, which may be any combination of wired and wireless networks. Example data sources include a sensor (e.g., an Internet of Things or IoT device), database log, an operating system log, an application log, a web server log, an application server log, a machine generated log, a social media feed, and the like. Alternately, the data source may be on server 104 or on servers 130_1 through 130_N.

The data is processed at server 104. A single server is referenced for simplicity, but embodiments contemplate the use of distributed servers to implement the operations disclosed herein. The server is operative for "in flight" processing of data before its next destination at one or more servers 130_1 through 130_N.

Server 104 includes standard components, such as a central processing unit 110 connected to a set of input/output devices 112 via bus 114. The input/output devices 112 may include a keyboard, mouse, touch display and the like. A network interface circuit 116 is also connected to the bus 114 to provide connectivity to network 106. A memory 120 is also connected to the bus 114. The memory stores instructions executed by the central processing unit 110 to implement operations disclosed herein. In particular, a stream processor 122 includes instructions to receive data from a data source that is continuously

publishing or generating records. The stream processor 122 selectively identifies conditions within the data that satisfy specified rules. The stream processor then reports the conditions, for example to one or more client devices 150_1 through 150_N. The data is then sent to a messaging system implemented on a cluster of servers 130_1 through 130_N. In particular, the data is sent to a message system node (e.g., server 130_1) that forms a segment of a distributed transaction log with messages persisted in non-volatile memory and replicated within a cluster of nodes. By way of example, the message system is Apache® Kafka®. Apache® Kafka® is an open source message broker project developed by the Apache Software Foundation. The message system is a massively scalable publish/subscribe message queue architected as a distributed transaction log to process streaming data.

Server 130_1 also includes standard components, such as a central processing unit 131, input/output devices 132, a bus 134 and a network interface circuit 136 to provide connectivity to network 106. A memory 140 is connected to the bus 134. The memory stores instructions executed by the central processing unit 131 to implement operations disclosed herein. The memory 140 stores a message system module_1 142, which is a node instance of the distributed messaging system. The memory 140 also stores a restart module 144 to coordinate restart operations in the case of a node failure and reboot sequence. The memory also stores a data mining module 146. The data mining module 146 includes executable instructions to coordinate analysis of persistently stored messages within the message system.

Figure 1 also illustrates client devices 150_1 through 150_N. Each client device may be a computer, tablet, smartphone, and the like. The client devices may receive reports of the conditions identified by the stream processor 122. The client devices may also be used to access and drive the data mining module on server 130_1.

Figure 2 illustrates processing operations associated with an embodiment of the message system. Data is received 200. As previously indicated, the data is from a data source that continuously generates or publishes records.

Next, the data is optionally filtered or transformed 202. For example, the filtering may be in the form of subsampling the data or only processing data above a specified threshold. Rules are then applied to the data 204. The rules are any pre-specified set of conditions applied to the data. In general, the rules incorporate business logic and are operative to identify data of interest. When conditions of interest are identified, the conditions are reported 206. For example, the conditions are reported in the form of a message to one or more client devices 150_1 through 150_N.

The data is then routed to the message system with a unique ID 208. The routing operation is over network 106 using an application program interface (API) of the message system module 142. In the case of a Kafka implementation, the KafkaProducer API is used.

The raw transaction record, as received by server 104, is forwarded over network 106 to a node (e.g., 130_1) of the message system. The stream processor 122 assigns a unique ID to the data. For example, the unique ID may be based upon a hash of the metadata associated with the record. The unique ID facilitates the removal of duplicate records at the server 104. Existing messaging systems have a difficult time identifying duplicate or redundant data. Utilization of the disclosed stream processor advantageously provides a de-duplication facility for the message system. This guarantees exactly once processing of a message input from systems 102_1 through 102_N.

The stream processor 122 then waits for an acknowledgement or commit signal 210 from the message system. Once the acknowledgement signal is received (210 – Yes), the unique ID is persisted 212. The unique ID represents the last acknowledged transaction, which may be referenced in the event of a node failure. The unique ID may be associated with a time stamp (e.g., a universal global time). This allows a message to be identified based upon time.

While waiting for the commit signal (210 – No), an abort threshold 214 may be tested. If the abort threshold (e.g., a specified period of time to wait) is not reached (214 – No), then control returns to block 210. Otherwise, (214 – Yes) the data is routed to another node 216 of the message system. The abort test is effectively testing for a failed node, in which case another node of the distributed message system is utilized.

The previously referenced commonly owned patent application discloses the creation and processing of streams. The current invention provides a mechanism for additionally storing, reading and replaying such streams from a durable messaging system. The stream processor 122 may be configured to process a stream using the following syntax:

CREATE STREAM <name> of <type> PERSIST [USING <property set>]

The <property set> is the name of a set of server properties. In the case of a Kafka® implementation, a default Global.DefaultKafkaProperties attribute is supported. Alternately, a custom property set may be created using the following syntax:

CREATE PROPERTYSET <name> (

Zk.address: '<zookeeper IP address>:<port>',

Bootstrap.brokers:'<bootstrap IP address>:<port>',

Zookeeper is a reference to Apache® ZooKeeper®, which is also an Apache Software Foundation open source initiative. Zookeeper® provides a distributed configuration service, synchronization service and naming registry for large distributed systems. Zookeeper® supports high availability through redundant services. Kafka® works in conjunction with Zookeeper®. The bootstrap IP address is a reference to the IP address of the server running the distributed messaging system software.

Figure 4 illustrates an interface 400 to persist a stream. A persist toggle switch 402 invokes the persist operation. A property set 404 is defined as DefaultKafkaProperties or a created property set, as in the case of the command line embodiment.

Figure 3 illustrates processing operations performed by the message system. In one embodiment, the message system is Apache® Kafka® with all its attendant functionality. Initially, a queued message or batch of messages is written to persistent storage 300. The message system module 142 then waits for an acknowledgement signal 302. During the wait period an abort condition 304 is tested. If the acknowledgement signal is not obtained within a threshold period of time (304 – Yes), the stream processor is advised 306. The stream processor may then invoke another node.

If a commit signal is received (302 – Yes), the commit signal is recorded as the last transaction. The record may include the unique ID and a commit time stamp, which are used in the case of a reboot of the system. A commit signal is then sent 310 to the stream processor.

The messaging system then hosts universal replay 312. That is, the next subscriber in the stream processing pipeline can now read the message from the durable queue without risk of losing the data and without relying on the original external data source to replay the data in the case of a crash. In the event that a node fails, upon reboot, the restart module 144 is operative. The restart module identifies the partition in the cluster with the replica data and ports the replica data onto the node. Note that the messaging system facilitates exactly once processing of data from a non-replayable external data source, such as data being received from an IoT device.

The data mining module 146 includes analytical tools to evaluate messages within the message system. In a Kafka® implementation, the Kafka® Consumer API is used. In one embodiment, the data mining module 146 includes a KafkaReader with the following properties:

Property	Type	Default Value	Notes
Topic	Java.lang.String		
brokerAddress	Java.lang.String		
PartitionIDList	Java.lang.String		Partition numbers to read from, separated by semicolongs, or leave blank to read from all partitions
Blocksize	Java.lang.integer	10240	Size of the fetch buffer in KB; set this to be greater than or equal to the Kafka broker's message.max.bytes size
KafkaConfig	Java.lang.String		Optionally specify Kafka producer properties, separated by semicolons
Charset	Java.lang.String	UTF-8	
startOffset	Java.lang.Long	-1	With default value of -1, reads from the end of the partition. Change to 0 to read from the beginning of the partition

Observe that the data mining module 146 is decoupled from the stream processor 122. Thus, the data mining module 146 can operate on local data. Moreover, computational resources of the stream processor 122 need not be used for playback or analysis of data.

The data mining module 146 may identify a message based upon the unique ID or a time stamp.

An embodiment of the present invention relates to a computer storage product with a non-transitory computer readable storage medium having computer code thereon for performing various computer-implemented operations. The media and computer code may be those specially designed and constructed for the purposes of the present invention, or they may be of the kind well known and available to those having skill in the computer software arts. Examples of computer-readable media include, but are not limited to: magnetic media, optical media, magneto-optical media and hardware devices that are specially configured to store and execute program code, such as application-specific integrated circuits ("ASICs"),

programmable logic devices ("PLDs") and ROM and RAM devices. Examples of computer code include machine code, such as produced by a compiler, and files containing higher-level code that are executed by a computer using an interpreter. For example, an embodiment of the invention may be implemented using JAVA®, C++, or other object-oriented programming language and development tools. Another embodiment of the invention may be implemented in hardwired circuitry in place of, or in combination with, machine-executable software instructions.

The foregoing description, for purposes of explanation, used specific nomenclature to provide a thorough understanding of the invention. However, it will be apparent to one skilled in the art that specific details are not required in order to practice the invention. Thus, the foregoing descriptions of specific embodiments of the invention are presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed; obviously, many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, they thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the following claims and their equivalents define the scope of the invention.

In the claims:

1. A machine, comprising:

a processor; and

a memory connected to the processor, the memory storing instructions executed by the processor to:

receive data from a data source that continuously publishes records,

selectively identify conditions within the data that satisfy specified rules, and

report the conditions prior to routing the data to a message system node forming a segment of a distributed transaction log with messages persisted in non-volatile memory and replicated within a cluster of nodes.

2. The machine of claim 1 wherein the instructions to selectively identify conditions within the data include data filtering and transformation instructions.

3. The machine of claim 1 wherein the message system is Apache® Kafka®.

4. The machine of claim 1 further comprising instructions executed by the processor to route the data to the message system with a unique identification.

5. The machine of claim 4 further comprising instructions executed by the processor to route the data to the message system with the unique identification and corresponding time stamp.

6. The machine of claim 4 further comprising instructions executed by the processor to store the unique identification in persistent memory in response to an acknowledge signal from the message system.

7. The machine of claim 1 further comprising instructions executed by the processor to route the data to another node within the cluster of nodes in response to an absent acknowledgement signal from the message system.

8. The machine of claim 1 further comprising instructions executed by the processor to replay data from the message system.

9. The machine of claim 1 further comprising instructions executed by the processor to utilize a unique message identification to remove duplicate records.

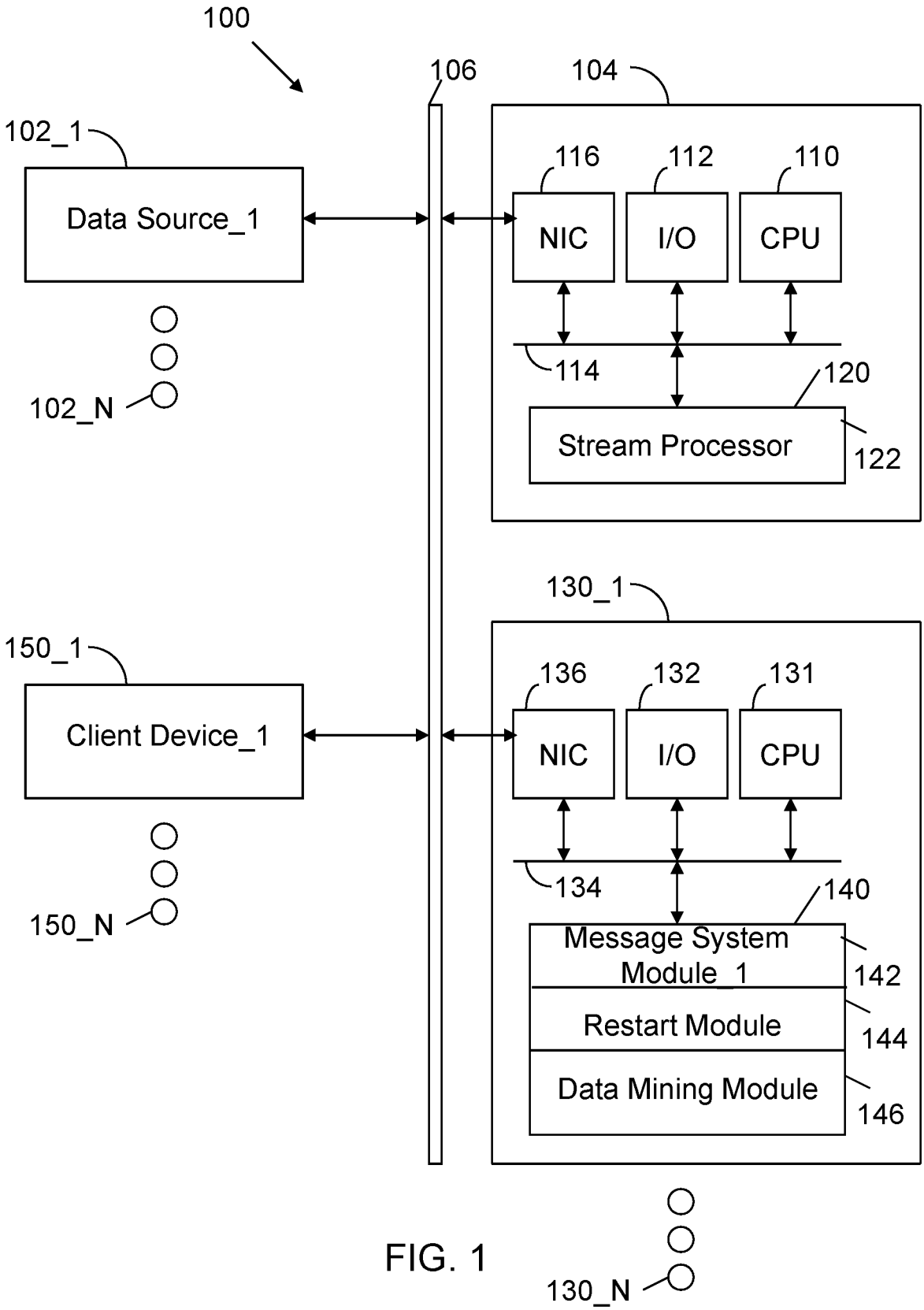

100
106
104
102_1
Data Source_1
102_N
116
112
110
NIC
I/O
CPU
114
120
Stream Processor
122
130_1
150_1
Client Device_1
150_N
136
132
131
NIC
I/O
CPU
134
140
Message System Module_1
142
Restart Module
144
Data Mining Module
146
130_N

FIG. 1

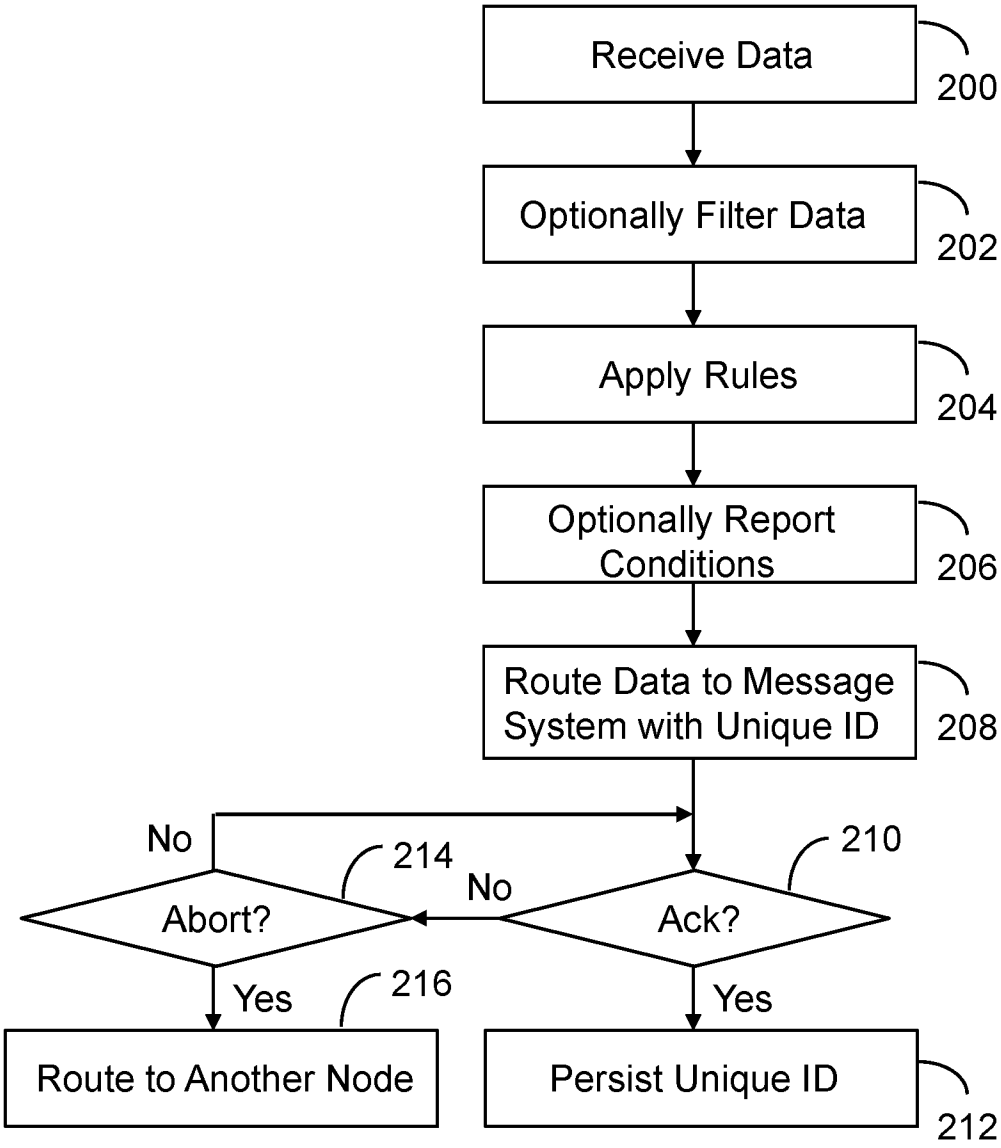

Receive Data
200
Optionally Filter Data
202
Apply Rules
204
Optionally Report Conditions
206
Route Data to Message System with Unique ID
208
No
214
Abort?
No
210
Ack?
Yes
216
Yes
Route to Another Node
Persist Unique ID
212

FIG. 2

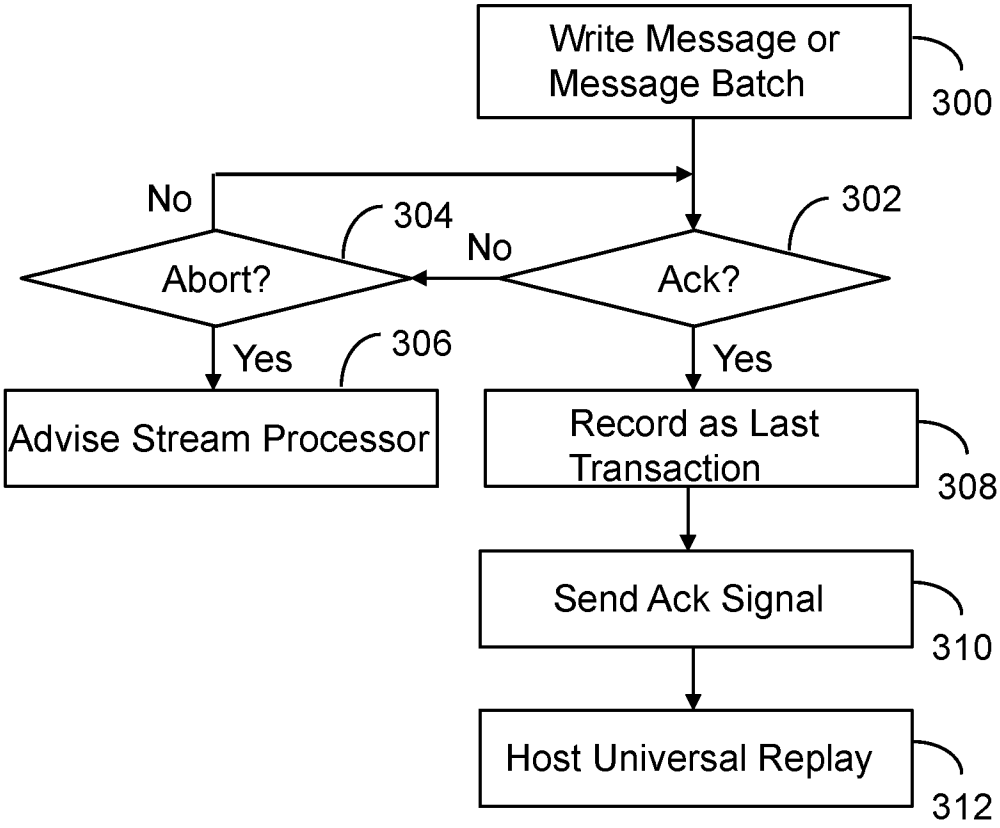

Write Message or Message Batch
300
No
304
302
No
Abort?
Ack?
Yes
306
Yes
Advise Stream Processor
Record as Last Transaction
308
Send Ack Signal
310
Host Universal Replay
312

FIG. 3

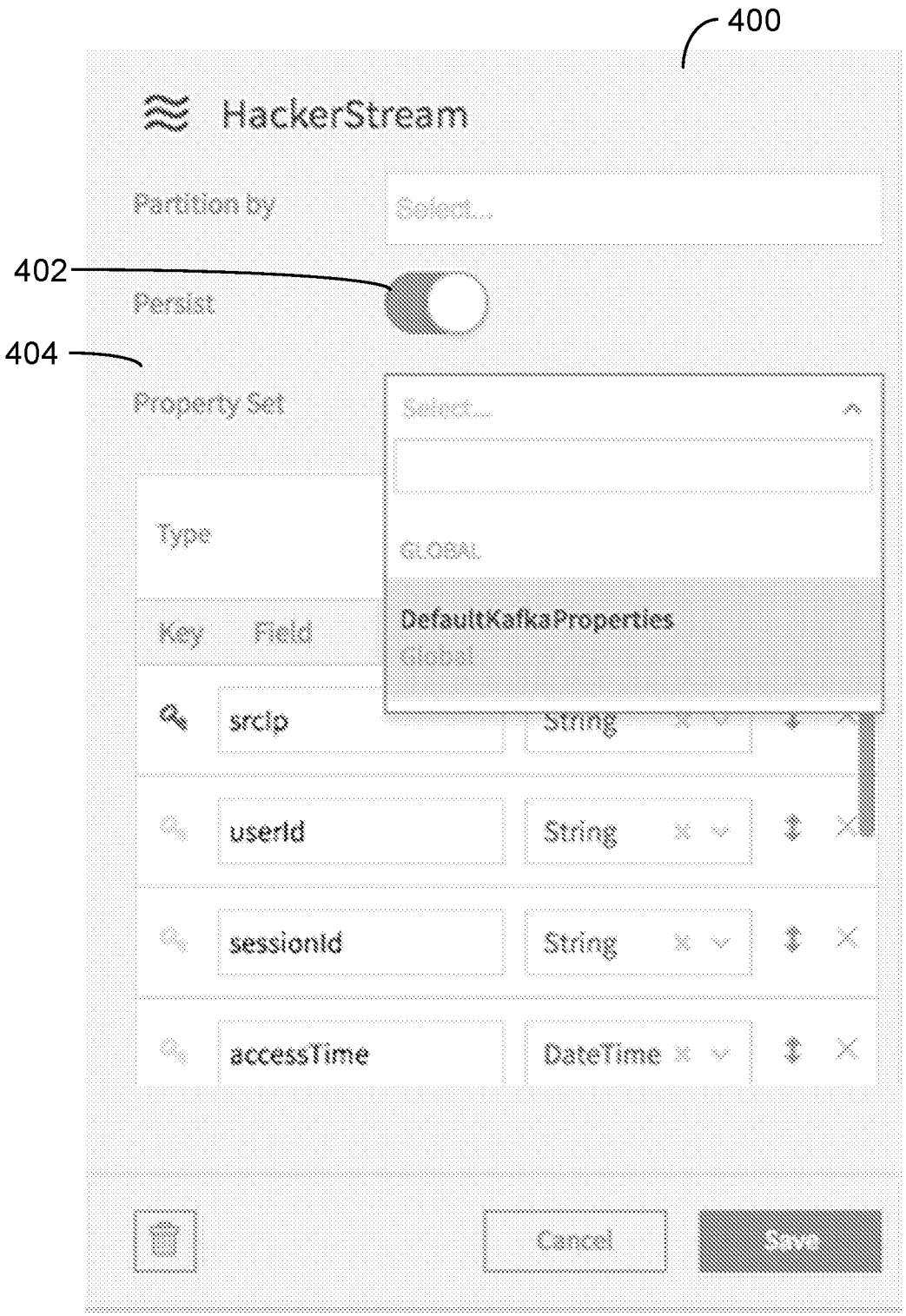

400
HackerStream
Partition by
Select...
402
Persist
404
Property Set
Select...
Type
GLOBAL
Key
Field
DefaultKafkaProperties
GLOBAL
srcIp
String
userId
String
sessionId
String
accessTime
DateTime
Cancel
Save

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 17/67668

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 9/46 (2018.01)
CPC - G06F 9/4843, G06F 9/52, G06F 9/4881, G06F 9/485, G06F 9/5027

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History Document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History Document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History Document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016/0292263 A1 (Oracle International Corporation) 06 October 2016 (06.10.2016), entire document especially paras [0042], [0061], [0076], [0078], [0101], [0226], [0263]	1-9
A	US 2005/0256825 A1 (Dettinger et al.) 17 November 2005 (17.11.2005), entire document	1-9
A	US 6,321,239 B1 (Shackelford) 20 November 2001 (20.11.2001), entire document	1-9
A	US 7,844,999 B1 (Aguilar-Macias et al.) 30 November 2010 (30.11.2010), entire document	1-9
A	US 2011/0307502 A1 (Boctor) 15 December 2011 (15.12.2011), entire document	1-9

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
19 February 2018 (19.02.2018)	09 MAR 2018
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)