

(19) **United States**
(12) **Patent Application Publication** (10) **Pub. No.: US 2025/0258592 A1**
Burke et al. (43) **Pub. Date: Aug. 14, 2025**

(54) **PRESENTING EVENTS BASED ON USER-SELECTED FIELDS**

- (71) Applicant: **Splunk Inc.**, San Francisco, CA (US)
- (72) Inventors: **Cory Eugene Burke**, San Bruno, CA (US); **Katherine Kyle Feeney**, Oakland, CA (US); **Divanny I. Lamas**, San Francisco, CA (US); **Marc Vincent Robichaud**, San Francisco, CA (US); **Matthew G. Ness**, Oakland, CA (US); **Clara E. Lee**, Pacifica, CA (US)
- (21) Appl. No.: **19/191,695**
- (22) Filed: **Apr. 28, 2025**

Related U.S. Application Data

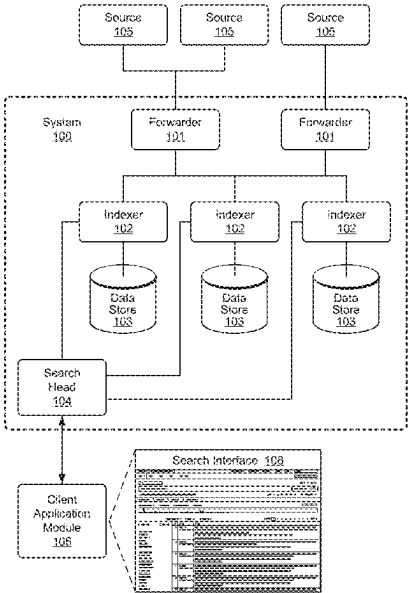
- (63) Continuation of application No. 18/403,594, filed on Jan. 3, 2024, which is a continuation of application No. 17/952,950, filed on Sep. 26, 2022, now Pat. No. 11,868,158, which is a continuation of application No. 16/397,393, filed on Apr. 29, 2019, now Pat. No. 11,455,087, which is a continuation of application No. 14/526,406, filed on Oct. 28, 2014, now Pat. No. 10,303,344.
- (60) Provisional application No. 62/060,545, filed on Oct. 6, 2014, provisional application No. 62/060,551, filed on Oct. 6, 2014, provisional application No. 62/060,560, filed on Oct. 6, 2014, provisional application No. 62/059,988, filed on Oct. 5, 2014, provisional application No. 62/059,989, filed on Oct. 5, 2014, provisional application No. 62/059,993, filed on Oct. 5, 2014, provisional application No. 62/059,944, filed on Oct. 5, 2014, provisional application No. 62/059,998, filed on Oct. 5, 2014, provisional application No. 62/060,001, filed on Oct. 5, 2014.

Publication Classification

- (51) **Int. Cl.**
G06F 3/04842 (2022.01)
G06F 3/0482 (2013.01)
G06F 3/04847 (2022.01)
G06F 9/451 (2018.01)
G06F 16/22 (2019.01)
G06F 16/242 (2019.01)
G06F 16/2455 (2019.01)
G06F 16/248 (2019.01)
G06F 16/25 (2019.01)
G06F 16/951 (2019.01)
G06F 40/18 (2020.01)
G06V 10/22 (2022.01)
- (52) **U.S. Cl.**
CPC **G06F 3/04842** (2013.01); **G06F 3/0482** (2013.01); **G06F 3/04847** (2013.01); **G06F 16/221** (2019.01); **G06F 16/242** (2019.01); **G06F 16/2455** (2019.01); **G06F 16/248** (2019.01); **G06F 16/252** (2019.01); **G06F 16/951** (2019.01); **G06F 40/18** (2020.01); **G06V 10/22** (2022.01); **G06F 9/451** (2018.02); **G06F 16/2425** (2019.01)

(57) **ABSTRACT**

In embodiments of field value search drill down, a search system exposes a search interface that displays one or more events returned as a search result set. A field-value pair can be emphasized in the field-value pairs of an event displayed in the search interface, and a menu is displayed with search options that are selectable to operate on the emphasized field-value pair of the event. The menu includes the search options to add search criteria of the emphasized field-value pair to a search command in a search bar of the search interface, exclude the search criteria of the emphasized field-value pair from a search, or create a new data search based on the emphasized field-value pair. A selection of one of the search options in the menu can be received, and the search command in the search bar is updated based on the search option that is selected.



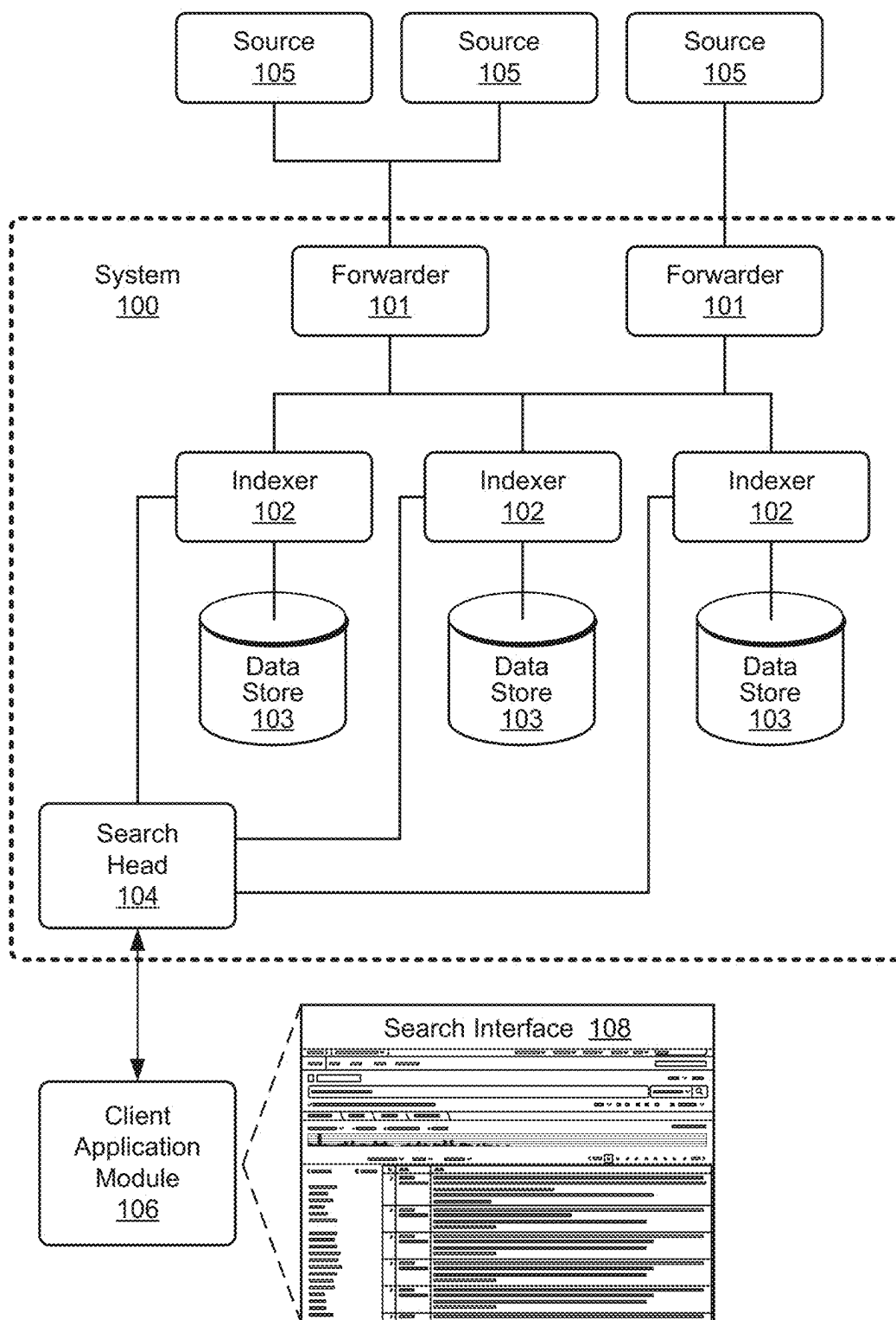


FIG. 1

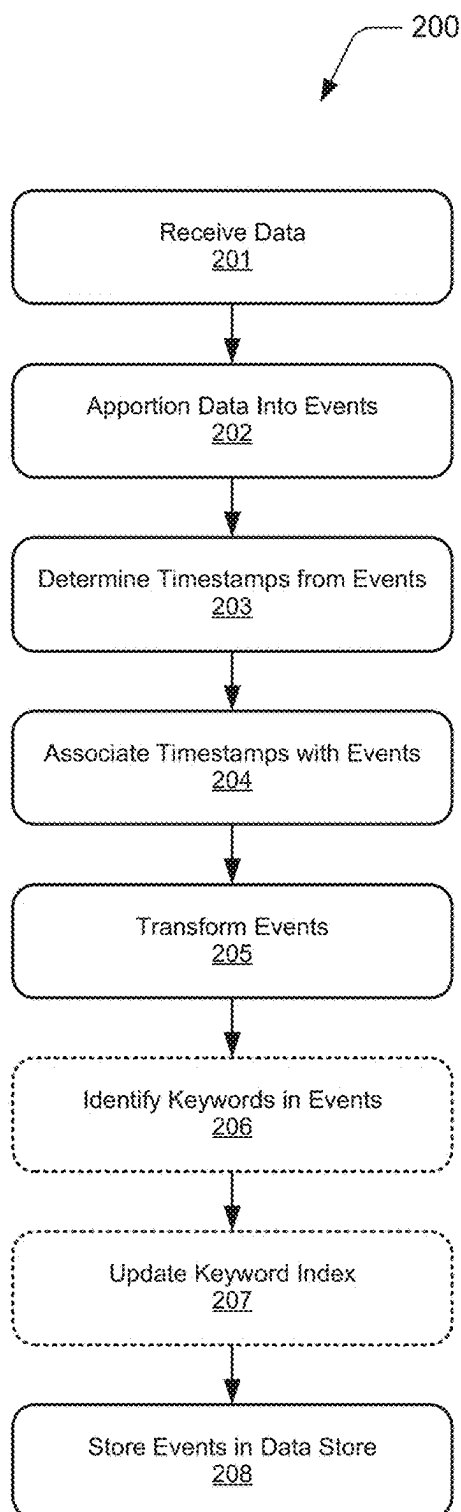


FIG. 2

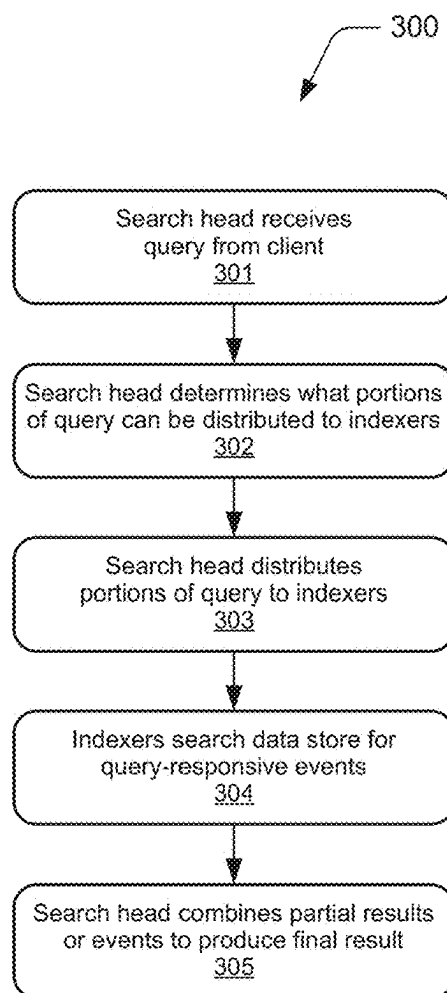


FIG. 3

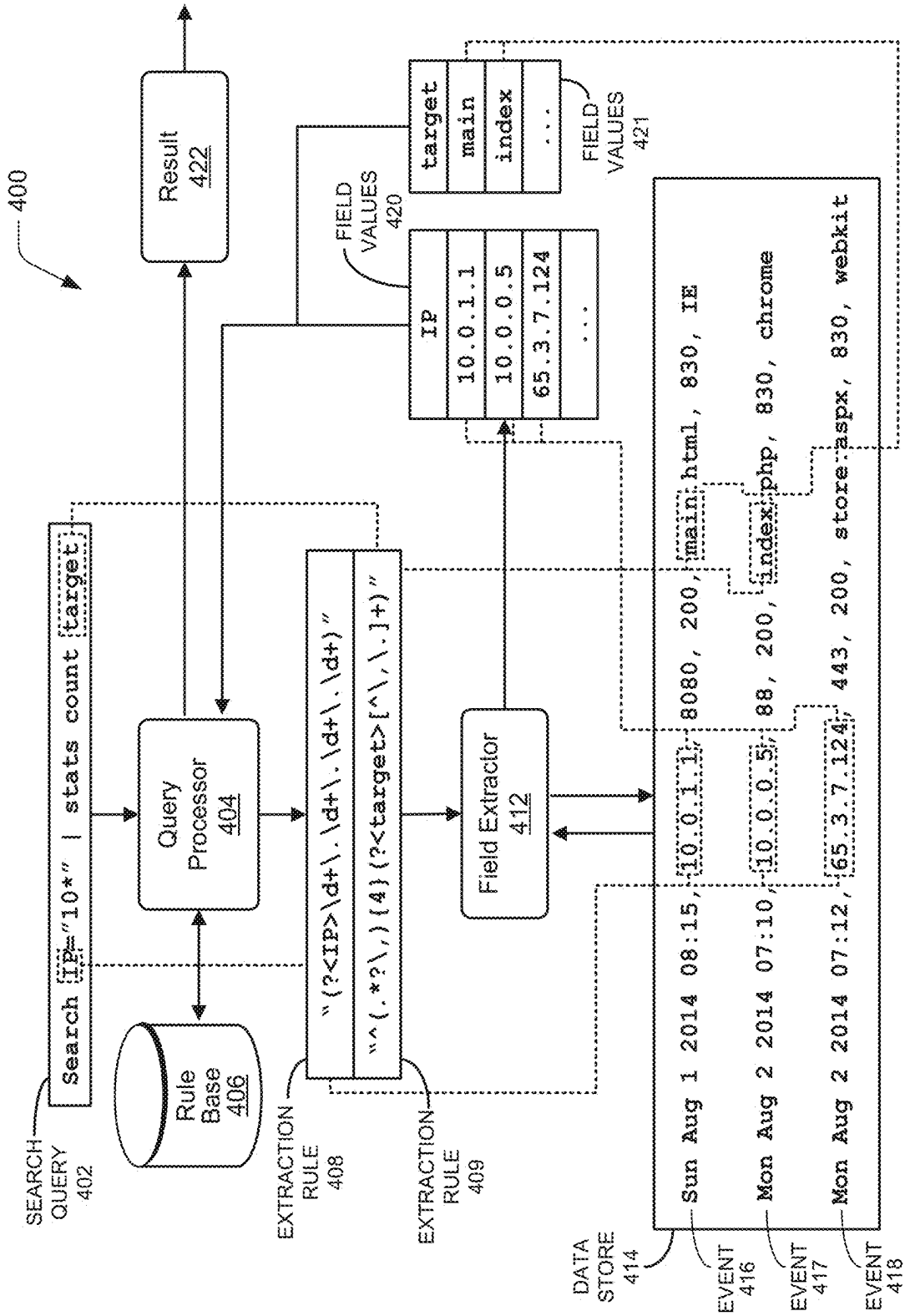


FIG. 4

500

Original Search: 501
search "error | stats count BY host

Sent to peers: 502
search "error | prestats count BY host (map)

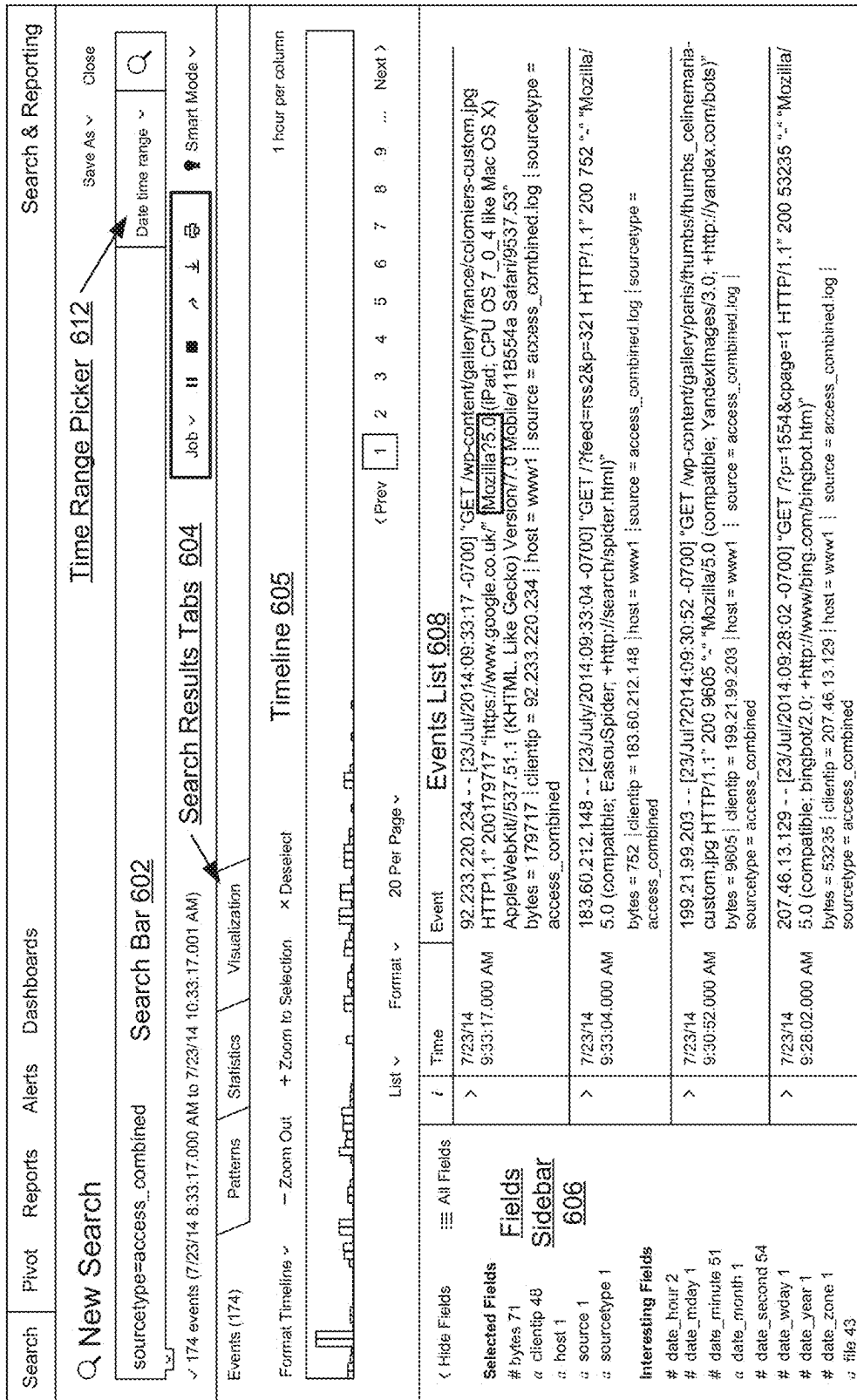
Executed by search head: 503
Merge prestats results received from peers (reduce)

FIG. 5

610

Data Summary x			
<div> <div>Hosts (5)</div> <div>Sources (8)</div> <div>Sourcetypes (3)</div> </div>			
<div>filter</div>			
Host	all	Count	Last Update
mailsv	all	9,829	4/29/14 1:32:47.000 PM
vendor_sales	all	30,244	4/29/14 1:32:46.000 PM
www1	all	24,221	4/29/14 1:32:44.000 PM
www2	all	22,595	4/29/14 1:32:47.000 PM
www3	all	22,975	4/29/14 1:32:45.000 PM

FIG. 6B



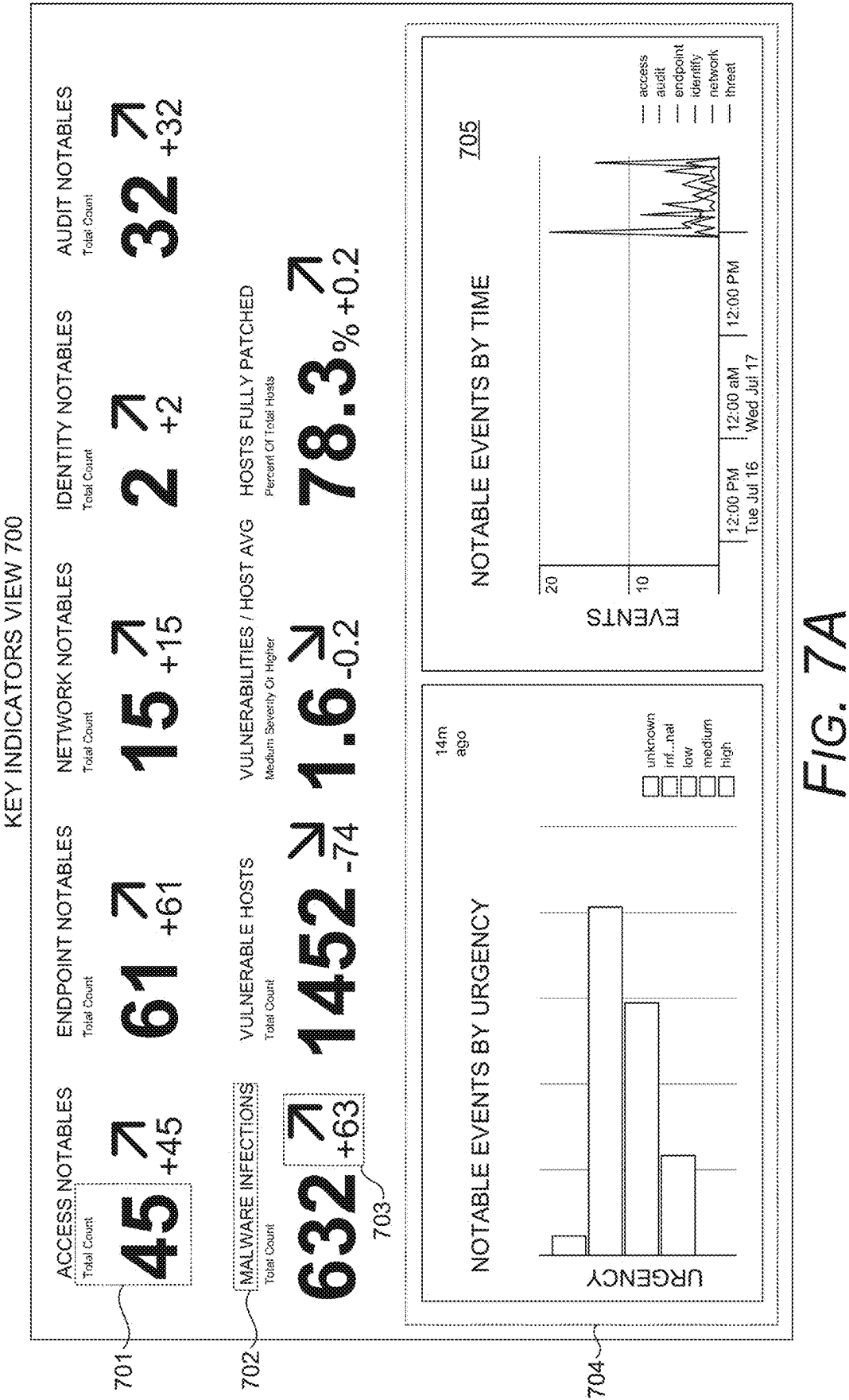


FIG. 7A

INCIDENT REVIEW DASHBOARD 710

Incident Review | Actions

Status:

Urgency:

Owner:

Title:

Security domain:

Governance:

Search:

TIME RANGE FIELD 712

24 hour window

Last 15 minutes

Last 60 minutes

Last 4 hours

Last 24 hours

Last 7 days

Last 30 days

Last year

Real-time

Other

All time

Custom time...

INCIDENT ATTRIBUTE FIELDS 711

225 matching events

Hide

Zoom out

Zoom to selection

Reset

120

60

4:00 AM

Sun Aug 26 2012

6:00 AM

8:00 AM

TIMELINE 713

Linear scale • 1 bar = 1 hour

120

60

EVENTS LIST 714

225 events in a 24 hour window (real-time) (from 11:29:20 AM August 25 to 11:29:20 AM August 26, 2012)

Select all | Unselect all | 1 2 3 4 5 6 7 8 9 10 next » | Edit selected events | Edit all 225 matching

Select	Options	Time	Security Domain	Title	Urgency	Status	Owner
<input type="checkbox"/>	<div>▼</div>	8/26/12 11:11:03:000 AM	Access	Insecure Or Cleartext Authentication Detected	① High	New	unassigned
<input type="checkbox"/>	<div>▼</div>	8/26/12 11:10:37:000 AM	Access	Insecure Or Cleartext Authentication Detected	① High	New	unassigned
<input type="checkbox"/>	<div>▼</div>	8/26/12 11:00:39:000 AM	Access	Account (blinbry) Deleted On (PROD-PDS-001)	① High	New	unassigned
<input type="checkbox"/>	<div>▼</div>	8/26/12 11:00:39:000 AM	Access	Account (beu) Deleted On (COREDEV-006)	① High	New	unassigned
<input type="checkbox"/>	<div>▼</div>	8/26/12 11:00:39:000 AM	Access	Account (combs) Deleted On (HOST-005)	① High	New	unassigned
<input type="checkbox"/>	<div>▼</div>	8/28/12	Access	Account (wisner) Deleted On (BUSDEV-005)	① High	New	unassigned

FIG. 7B

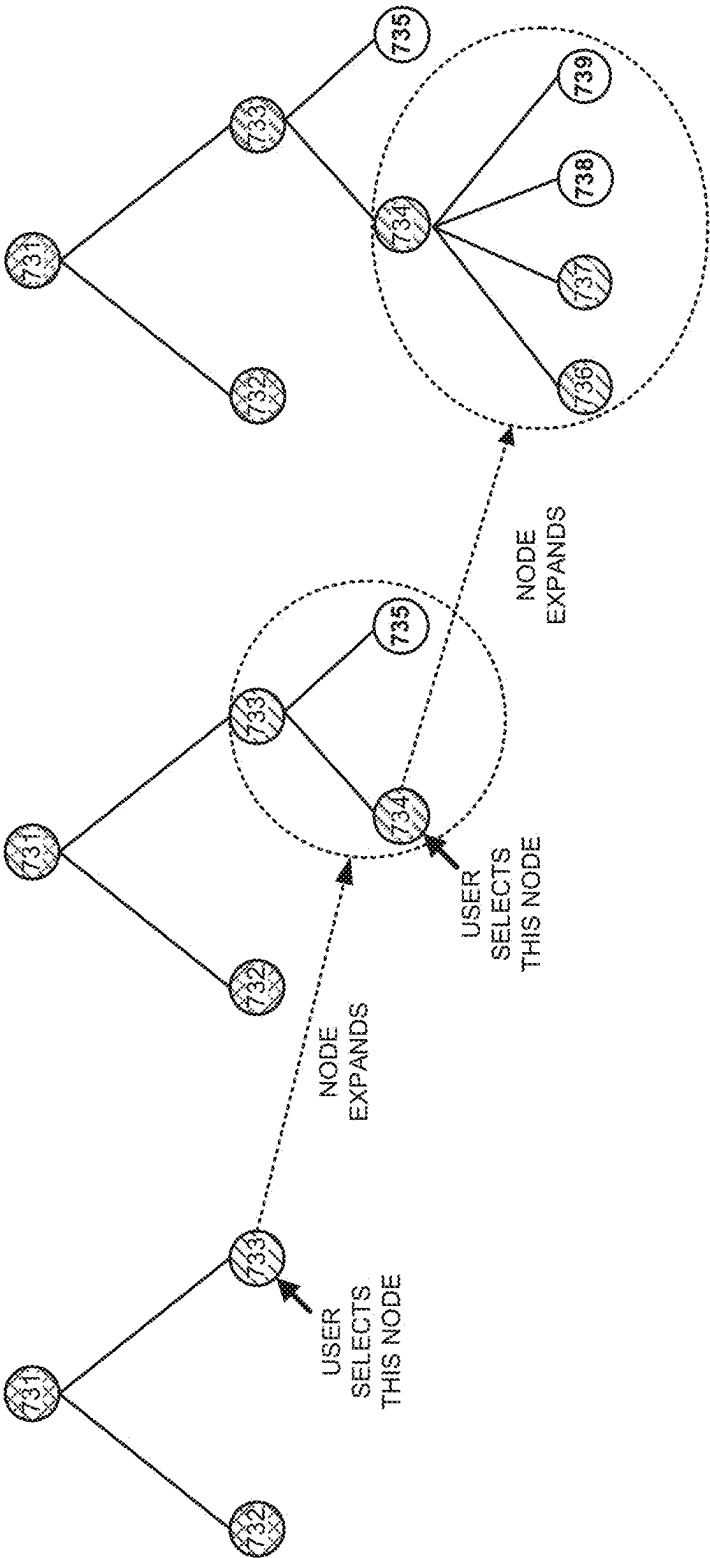
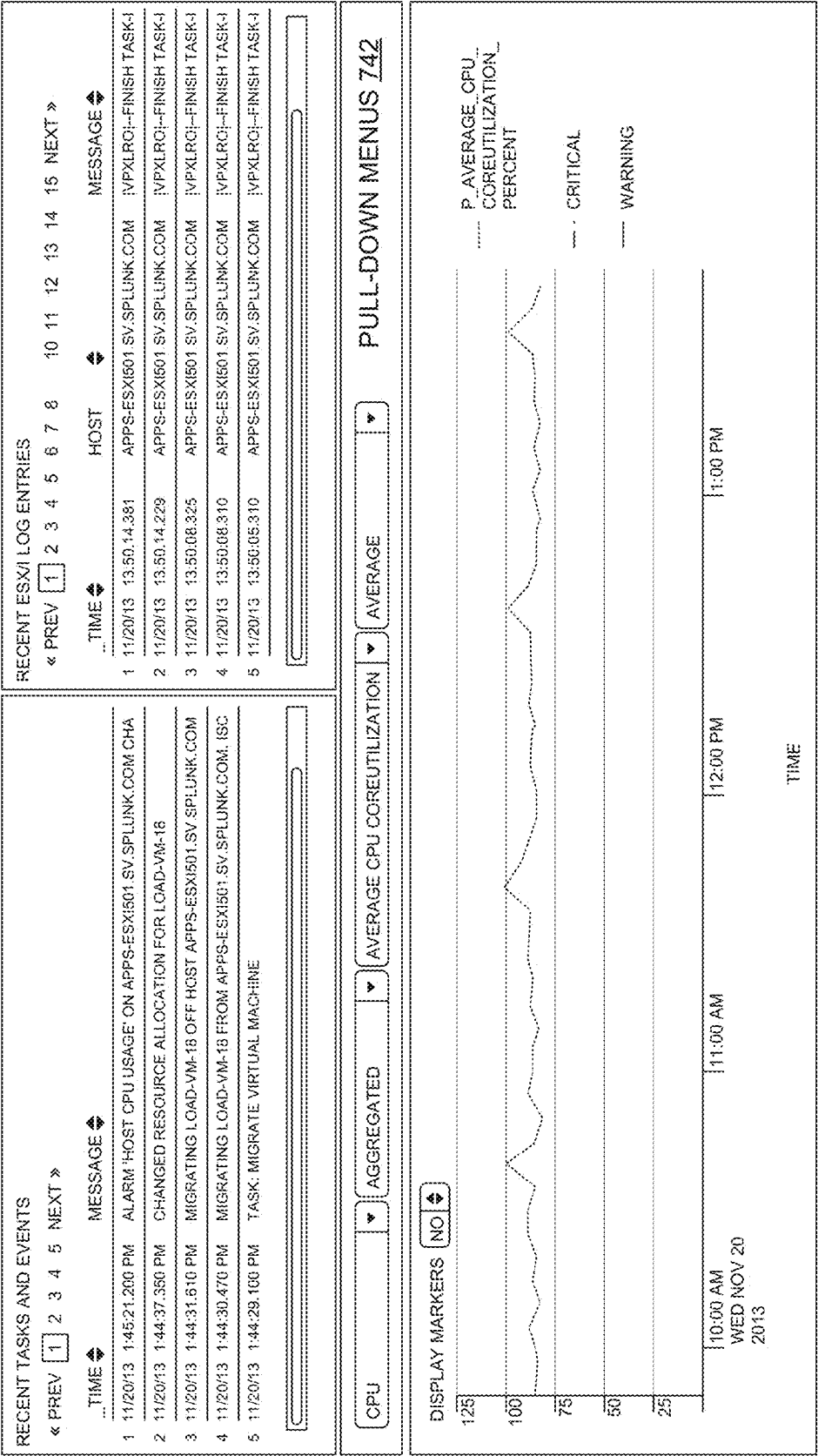
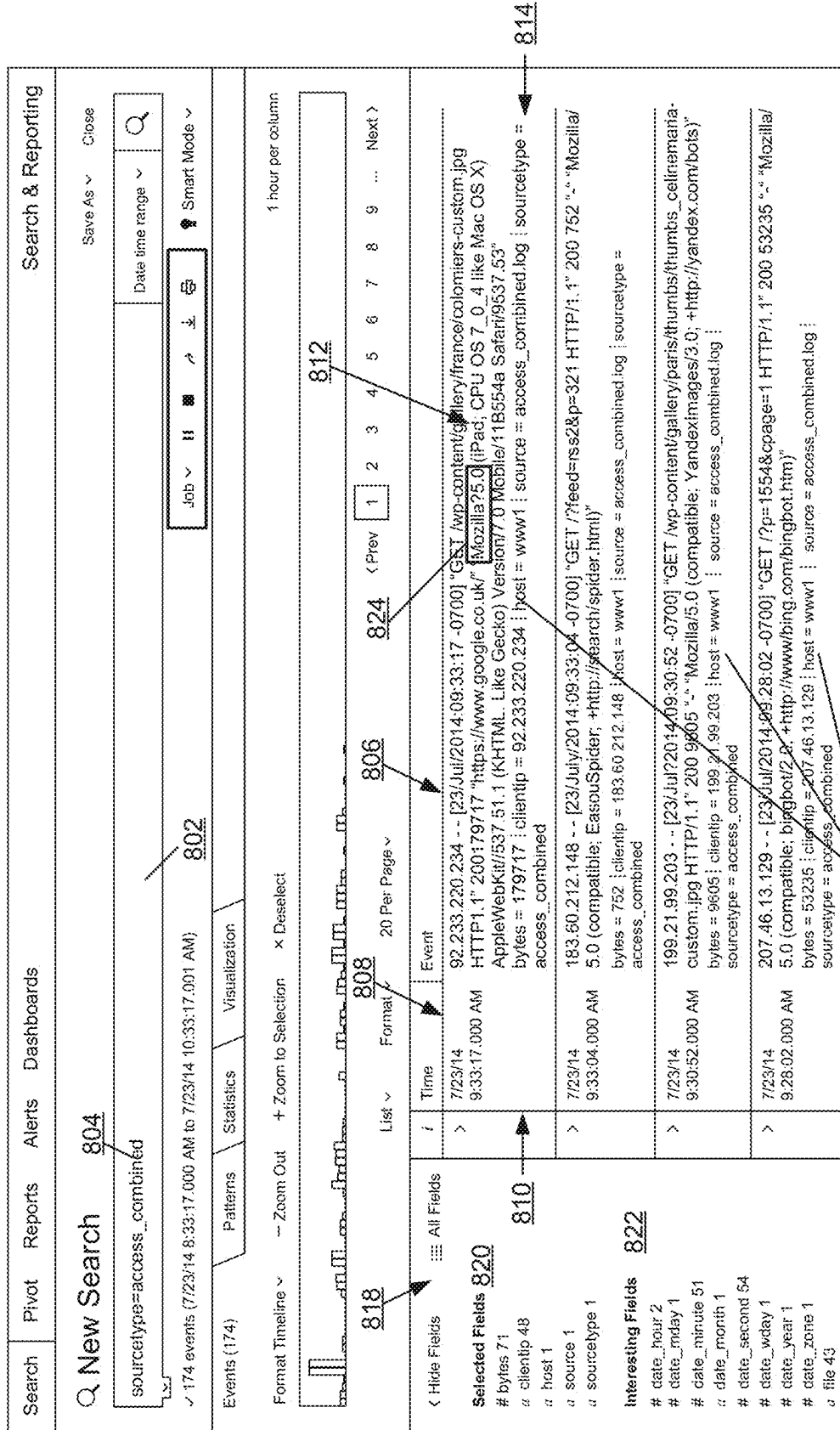
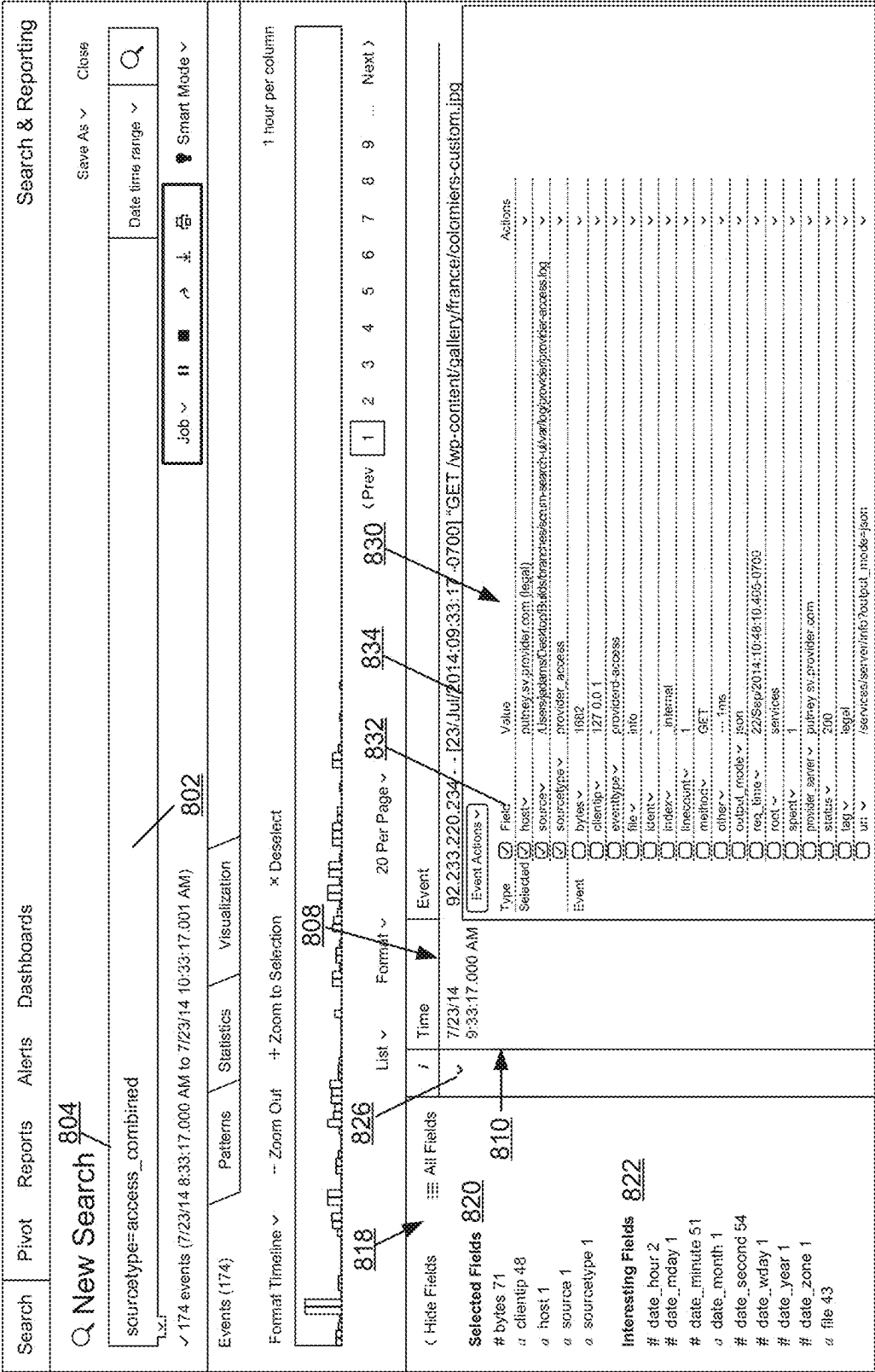


FIG. 7C

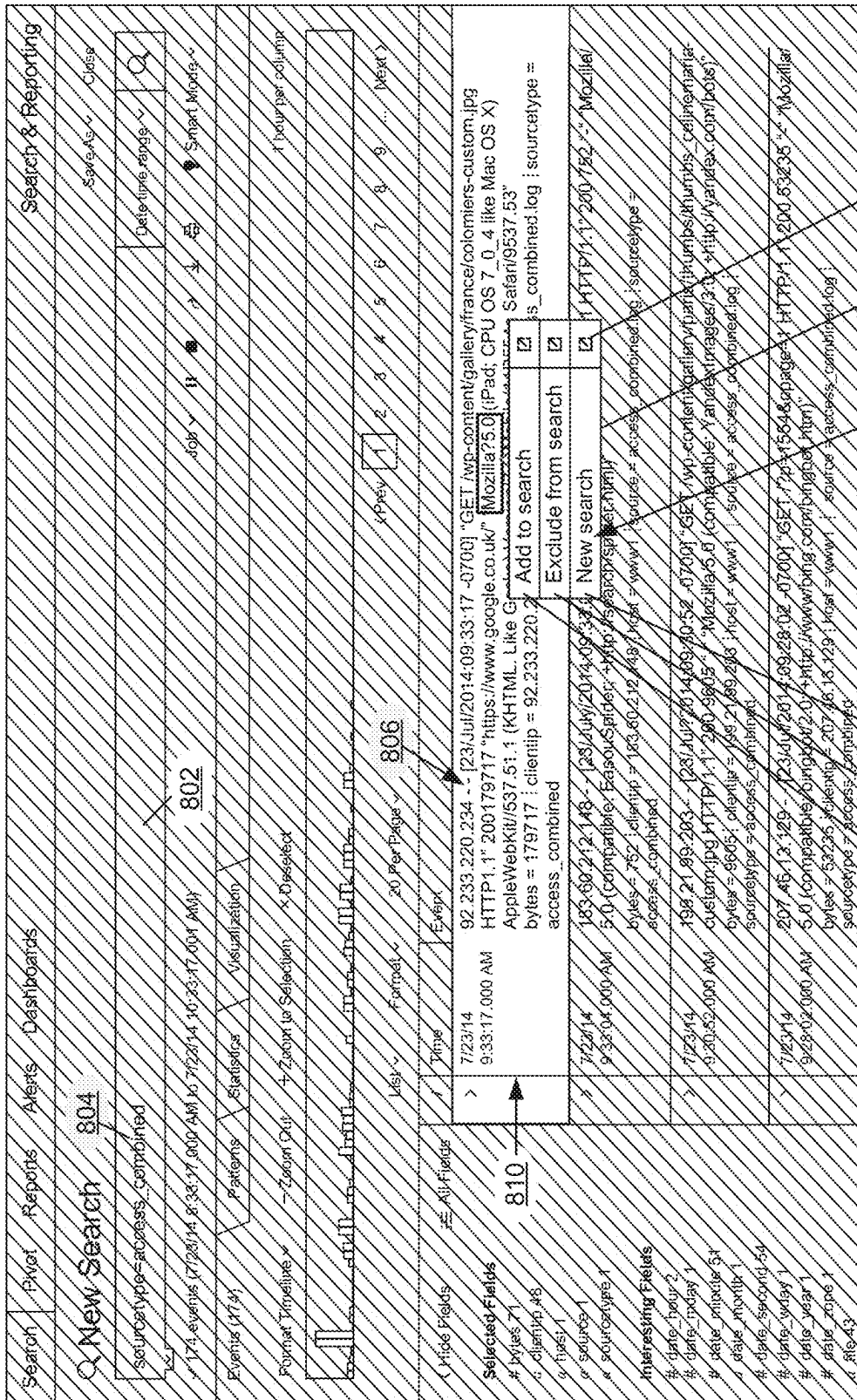






Event Field-Picker
Interface 828

FIG. 8B



Search Interface 800

FIG. 8C

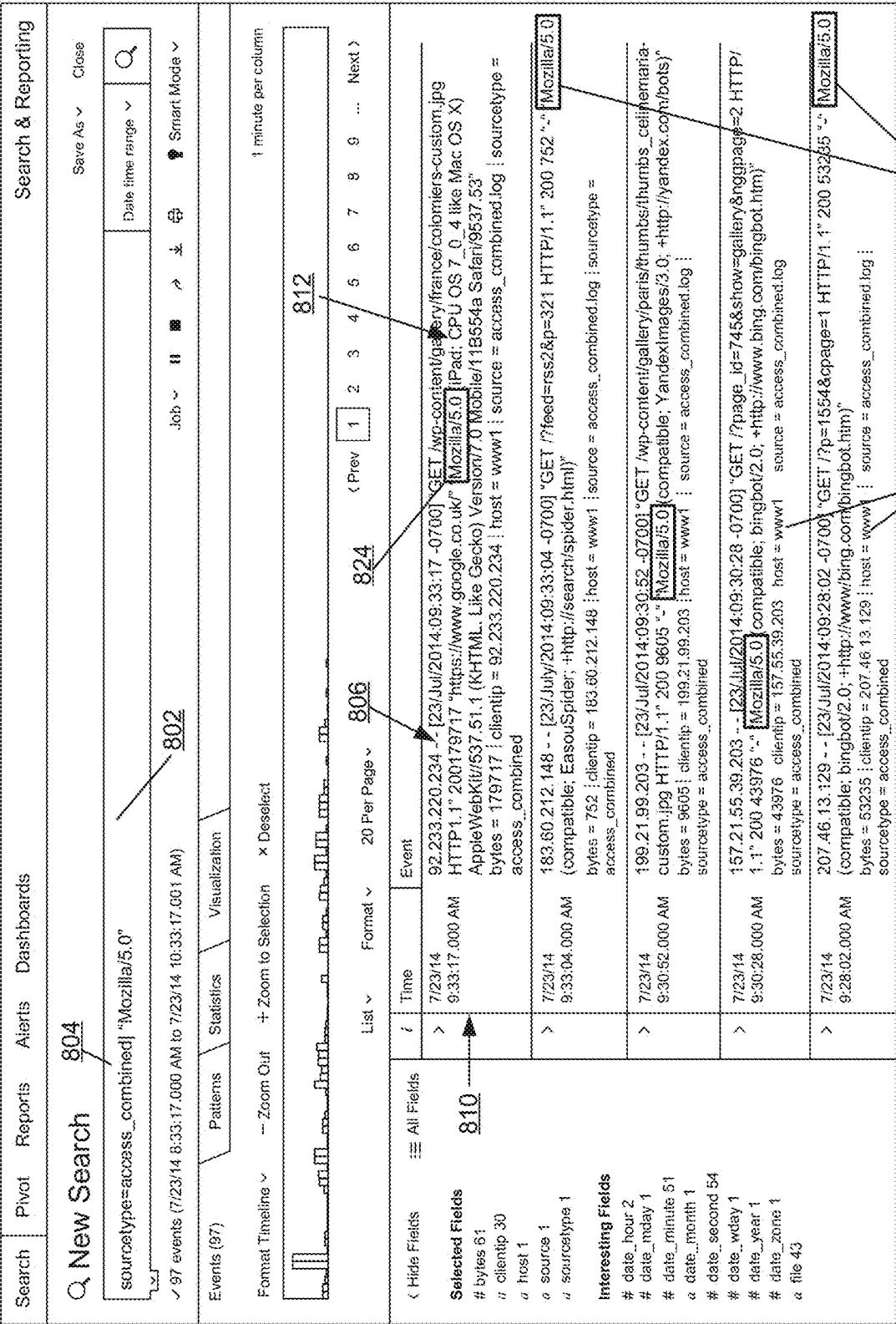
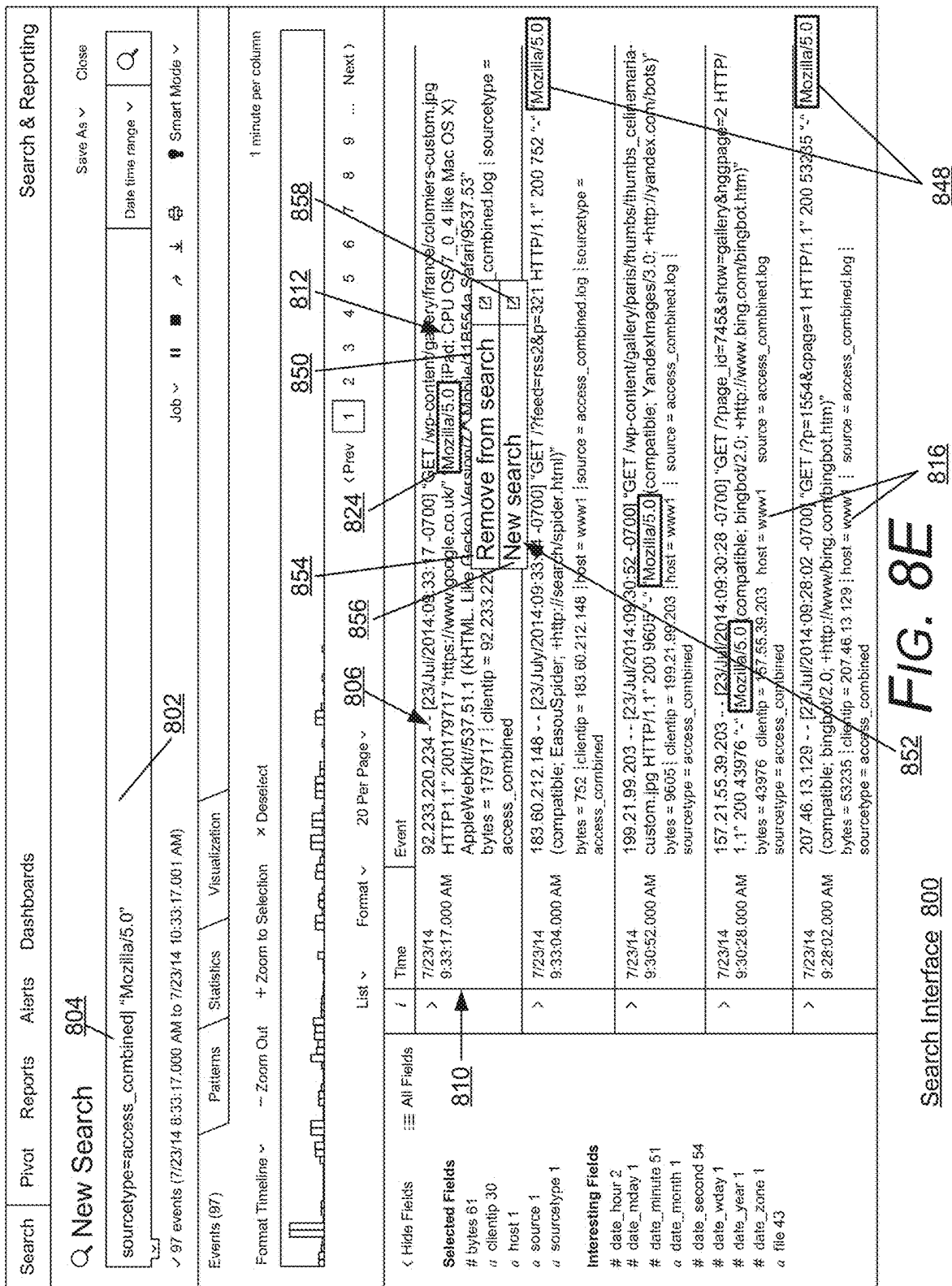


FIG. 8D

Search Interface 800



Search & Reporting

New Search

Save As ▾ Close

sourcetype=access_combined NOT "Mozilla/5.0"

Date time range ▾ 🔍

✓ 77 events (7/23/14 8:33:17.000 AM to 7/23/14 10:33:17.001 AM) 802

Events (77)

1 minute per column

Format Timeline ▾ -- Zoom Out + Zoom to Selection × Deselect

Visualization

i	Time	Event
>	7/23/14 9:27:41.000 AM	89.248.168.164 - [23/Jul/2014:09:27:41 -0700] "POST /xmlrpc.php HTTP/1.0" 200 370 "-" Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0) bytes = 370 clientip = 89.248.168.164 host = www1 source = access_combined sourcetype = access_combined
>	7/23/14 9:27:384.000 AM	93.174.93.204 - [23/Jul/2014:09:27:38 -0700] "POST /xmlrpc.php HTTP/1.0" 200 370 "-" Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0) bytes = 370 clientip = 93.174.93.204 host = www1 source = access_combined sourcetype = access_combined
>	7/23/14 9:24:56.000 AM	94.242.206.244 - [23/Jul/2014:09:24:56 -0700] "POST /xmlrpc.php HTTP/1.0" 200 370 "-" Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0) bytes = 370 clientip = 94.242.206.244 host = www1 source = access_combined sourcetype = access_combined
>	7/23/14 9:24:41.000 AM	10.1.10.14 - [23/Jul/2014:09:24:41 -0700] "GET / HTTP/1.0" 200 21453 "-" bytes = 370 clientip = 10.1.10.14 host = www1 source = access_combined sourcetype = access_combined
>	7/23/14 9:22:51.000 AM	94.242.206.244 - [23/Jul/2014:22:51 -0700] "POST /xmlrpc.php HTTP/1.0" 200 370 "-" Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0) bytes = 370 clientip = 94.242.206.244 host = www1 source = access_combined sourcetype = access_combined

< Hide Fields ≡ All Fields

810

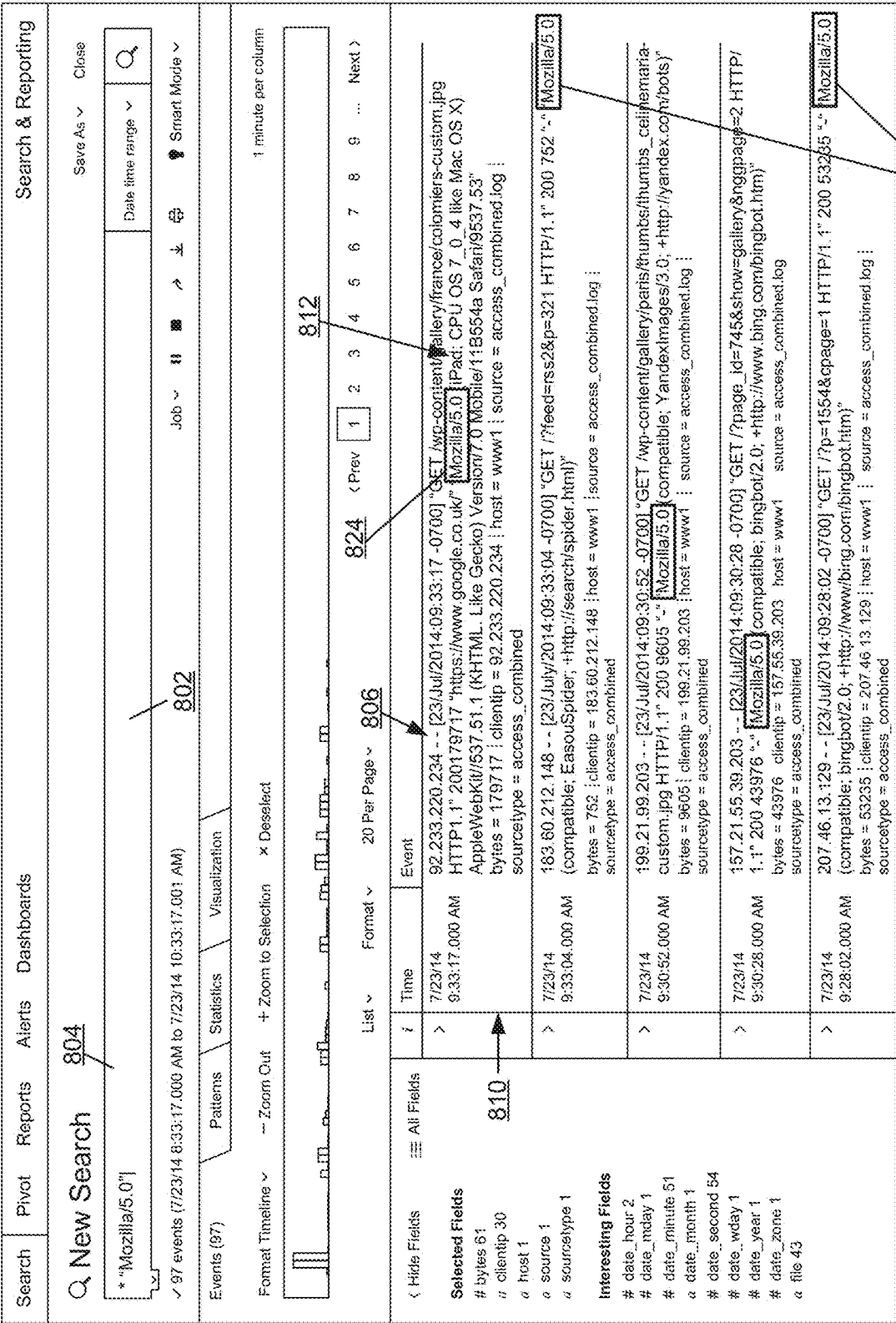
Selected Fields

- # bytes_61
- a clientip_30
- a host_1
- a source_1
- a sourcetype_1

Interesting Fields

- # date_hour_2
- # date_mday_1
- # date_minuta_51
- a date_month_1
- # date_second_54
- # date_wday_1
- # date_year_1
- # date_zone_1
- a file_43

FIG. 8F



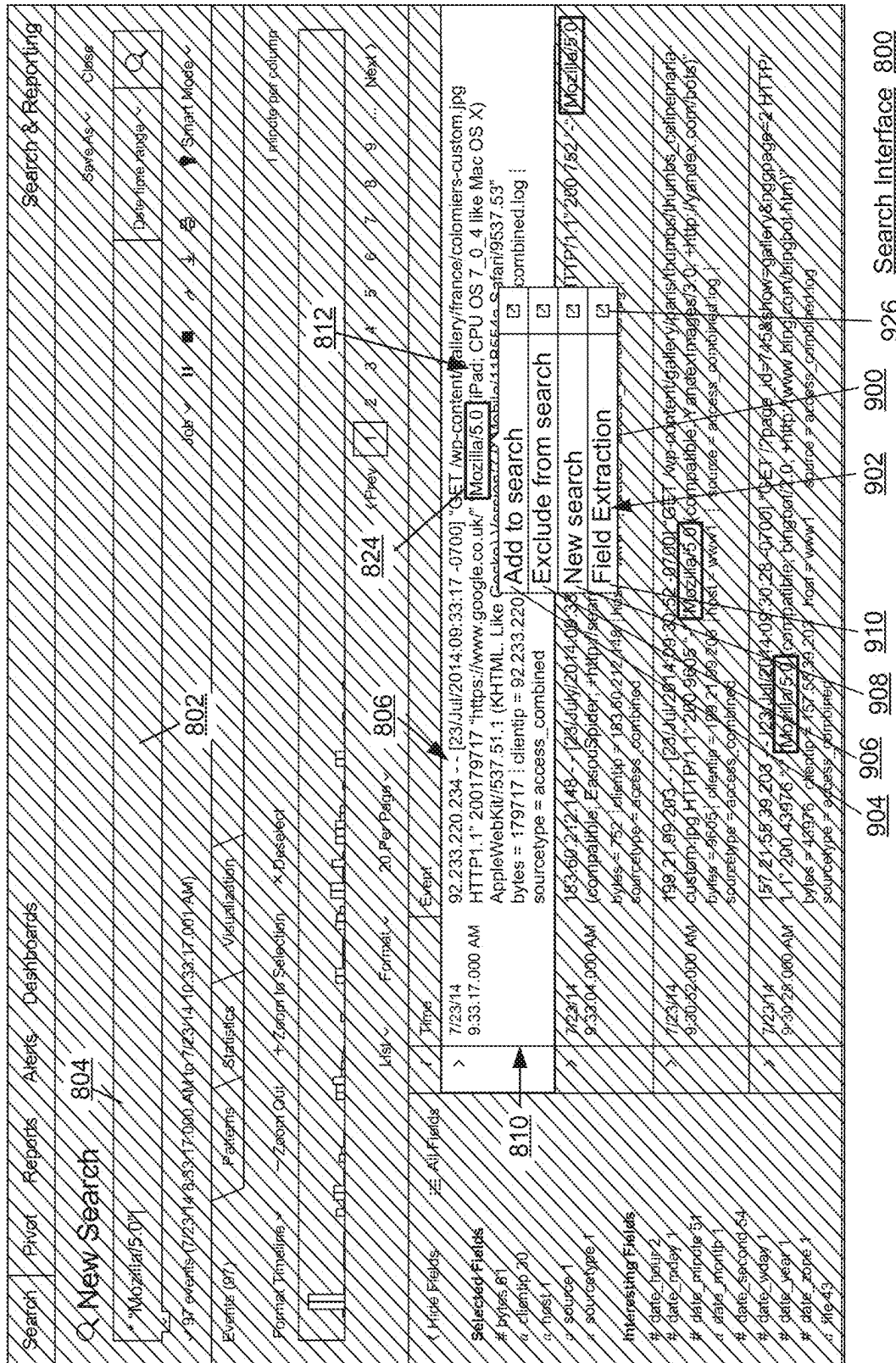


FIG. 9A

Extract Fields

Select source type Select sample Select fields Validate fields Save

Select Fields

Highlight one or more values in the sample event to create fields. You can indicate one value is required, meaning it must exist in an event for the regular expression to match. Click on highlighted values in the sample event to modify them. [Learn more](#)

127.0.0.1 -- admin [22/Sep/2014:10:16:18 -0700] "GET /en-US/api/shelher?snippet=true&shippeEmbedUS=false&namesplace=search&search=search-index%3D_internet+useAssistant&useTypeahead=true&showCommand+help=true&showCommandHistory=true&showFieldInfo=false&-141147search79=search%20index%3D_internet%20useAssrv-32.0) Gecko/20100101 Firefox/32.0" - 542069

Field Name

Sample Value owFieldInfo

920 922

916 924 918

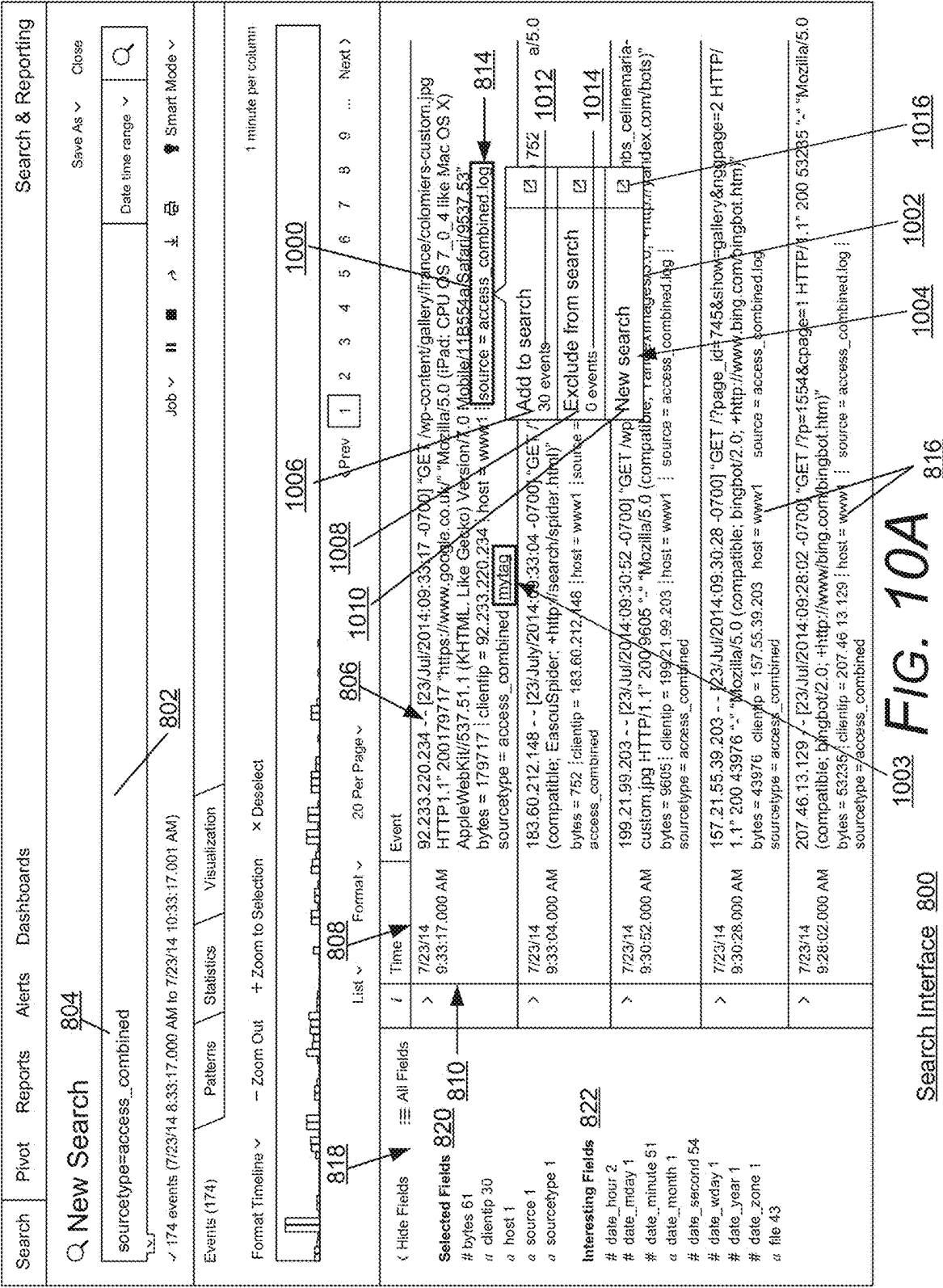
Extract Require Add Extraction

Next >

About Support File a Bug Documentation Privacy Policy

Extract Fields Interface 912

FIG. 9B



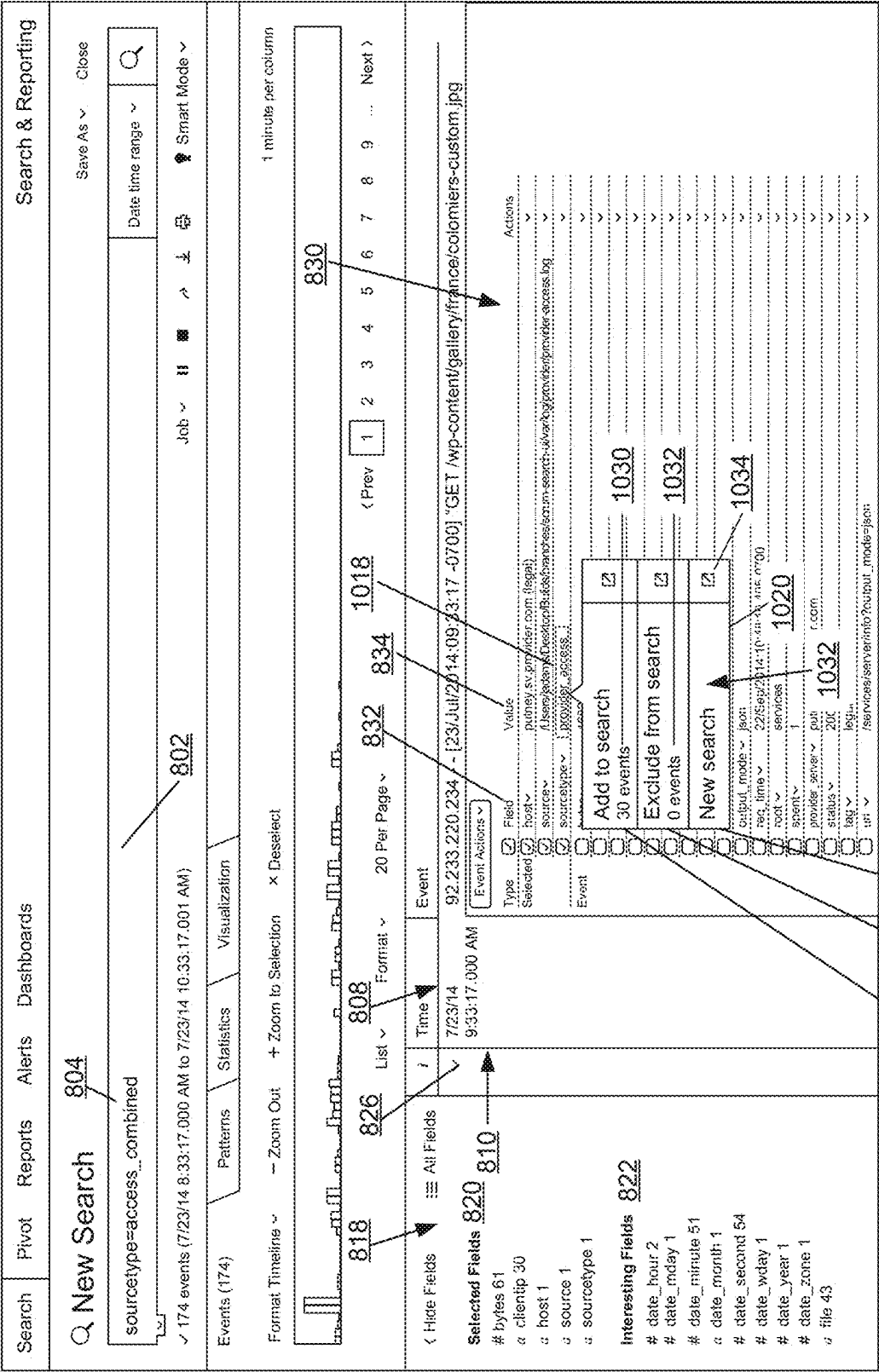


FIG. 10B

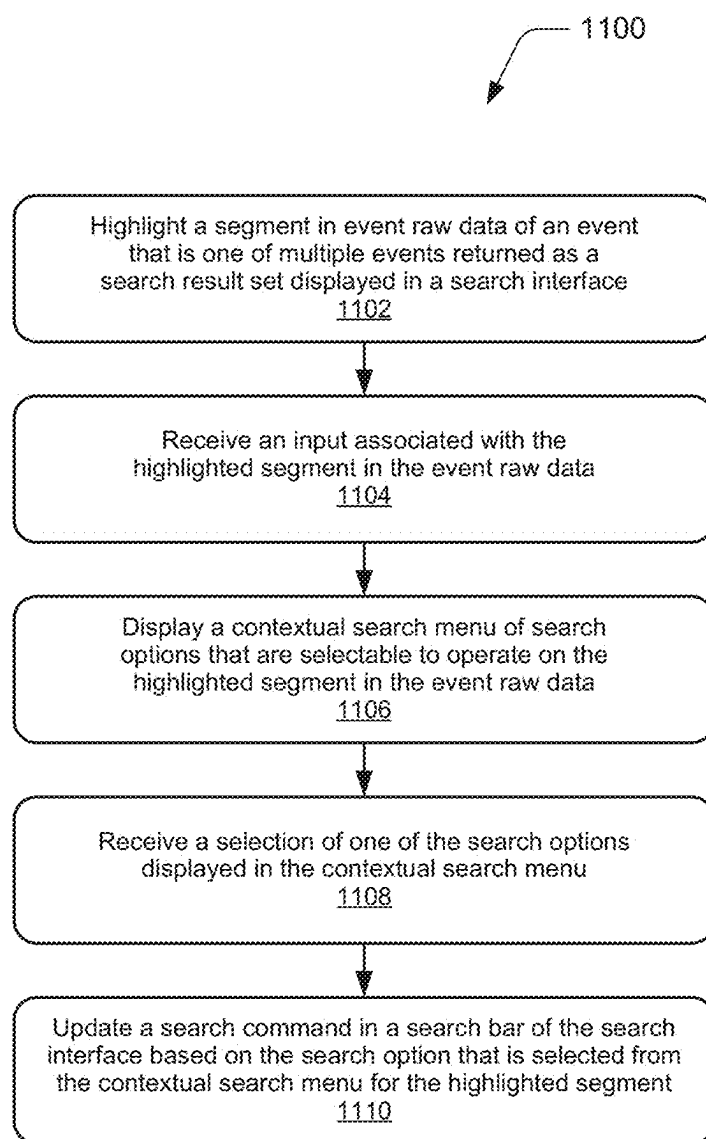


FIG. 11A

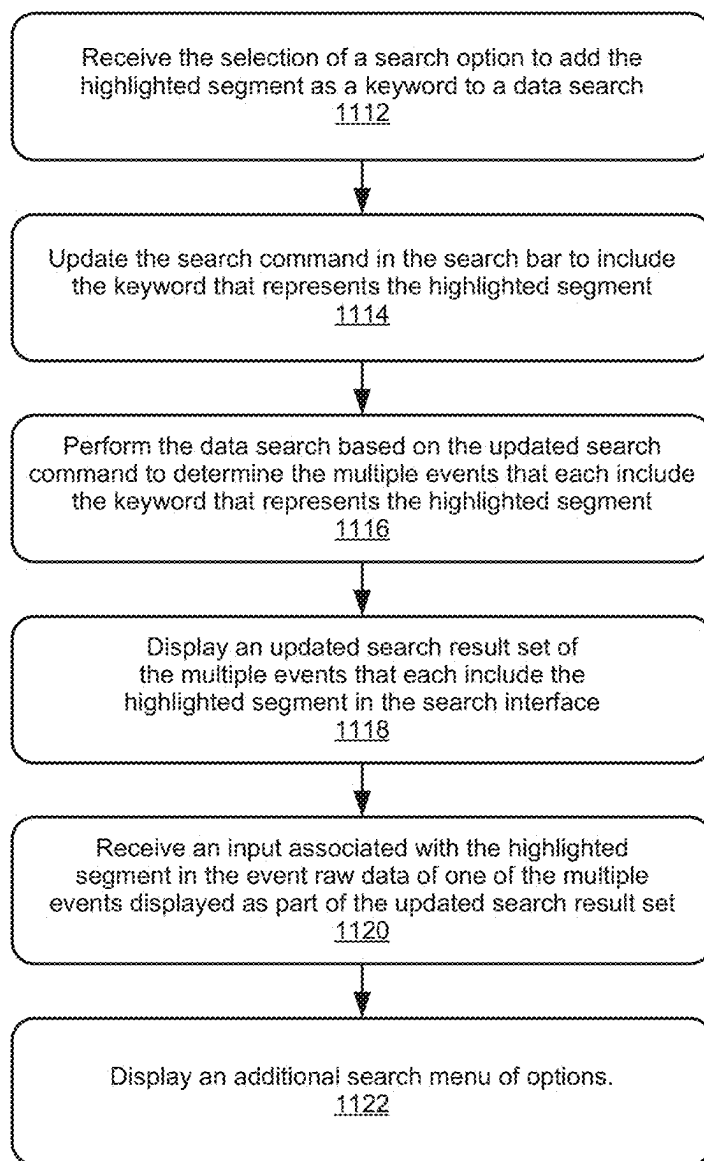


FIG. 11B

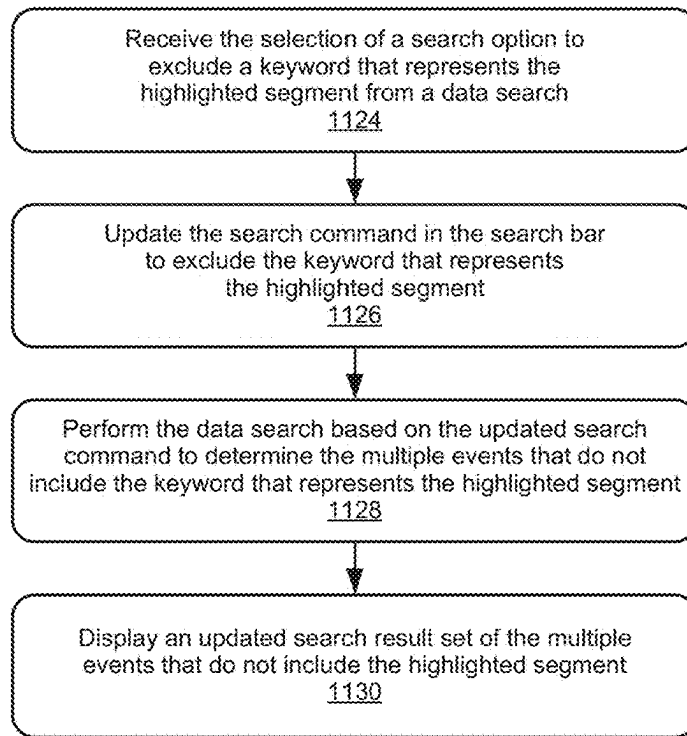


FIG. 11C

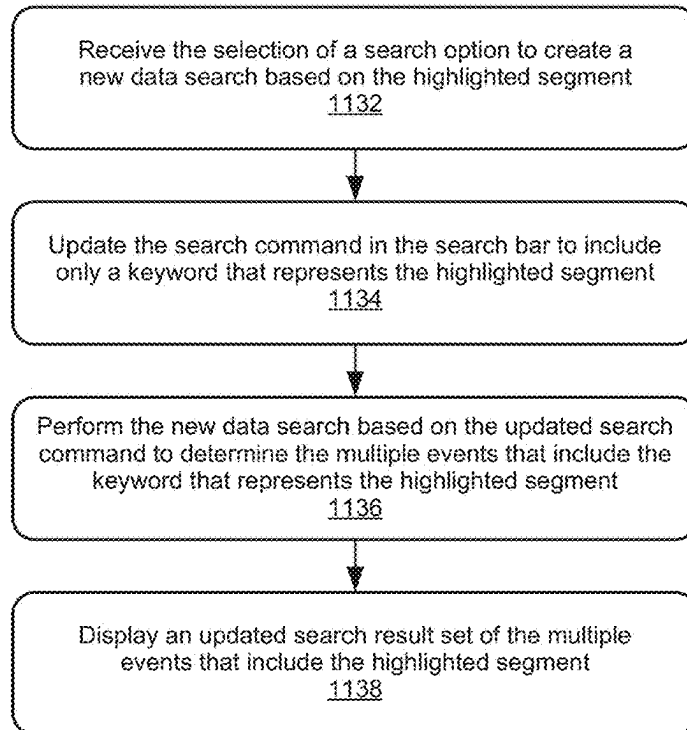


FIG. 11D

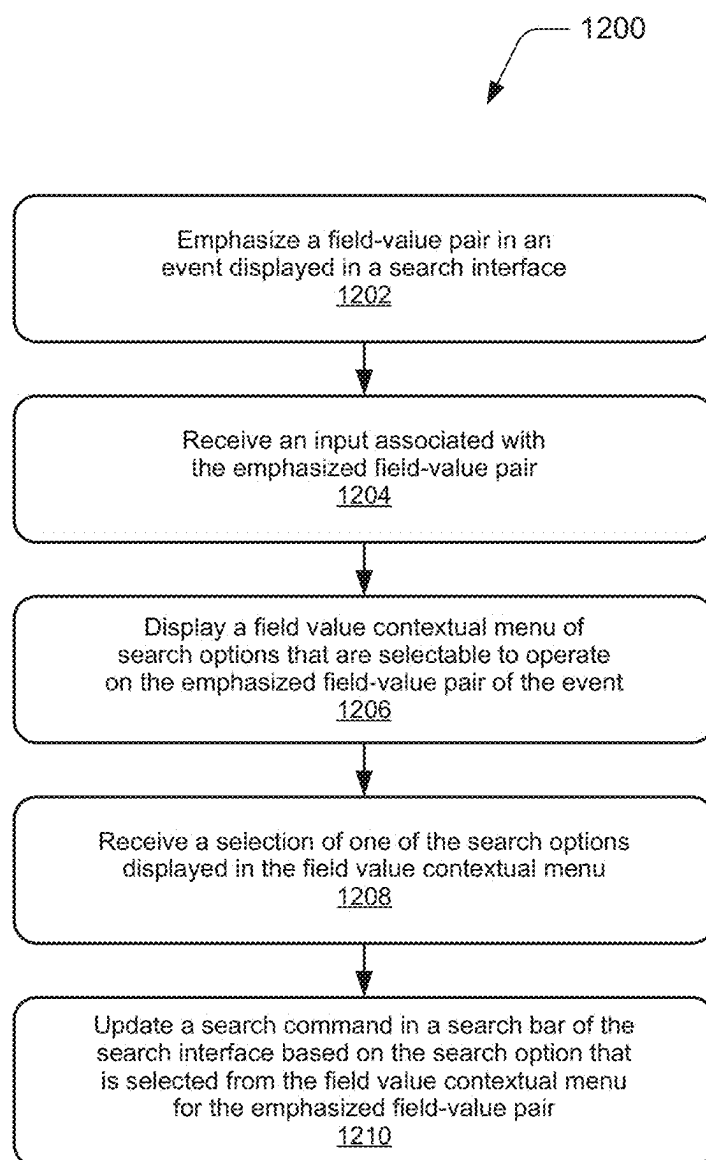


FIG. 12A

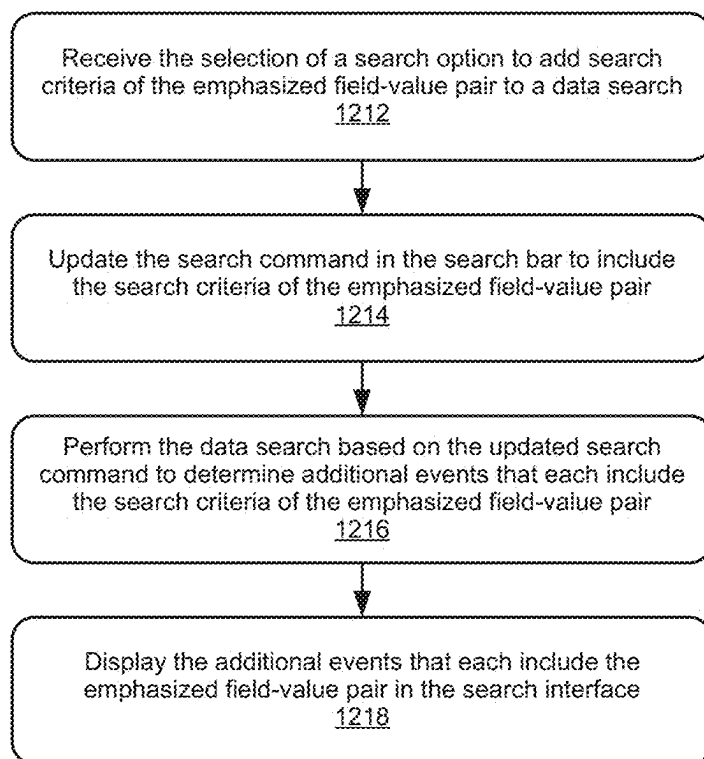
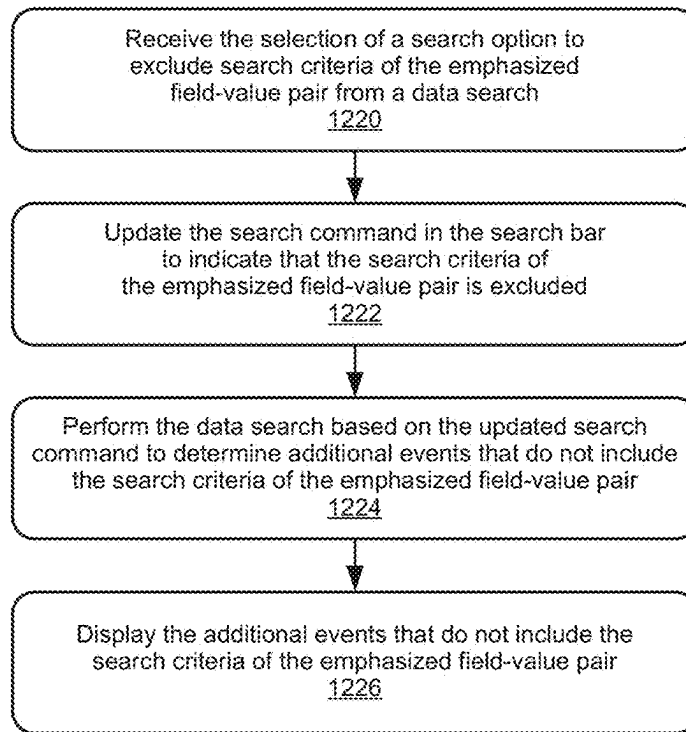
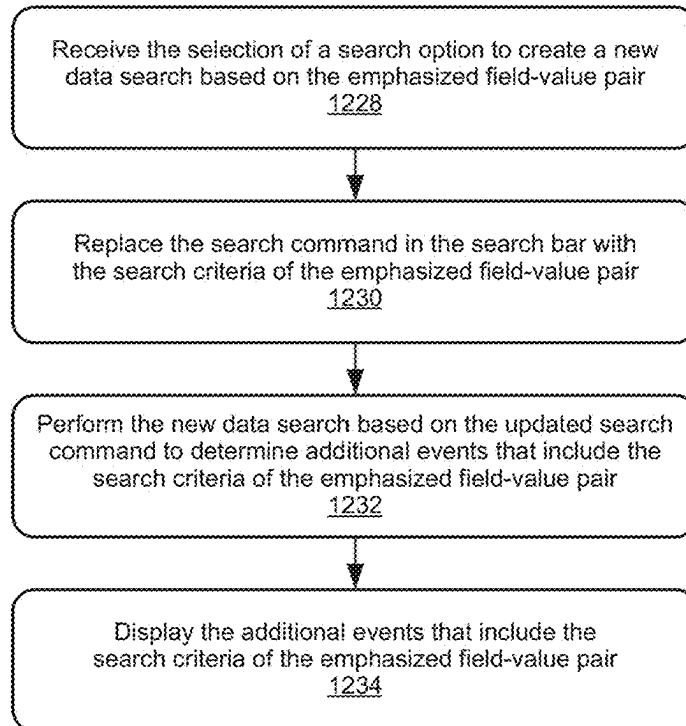


FIG. 12B

*FIG. 12C**FIG. 12D*

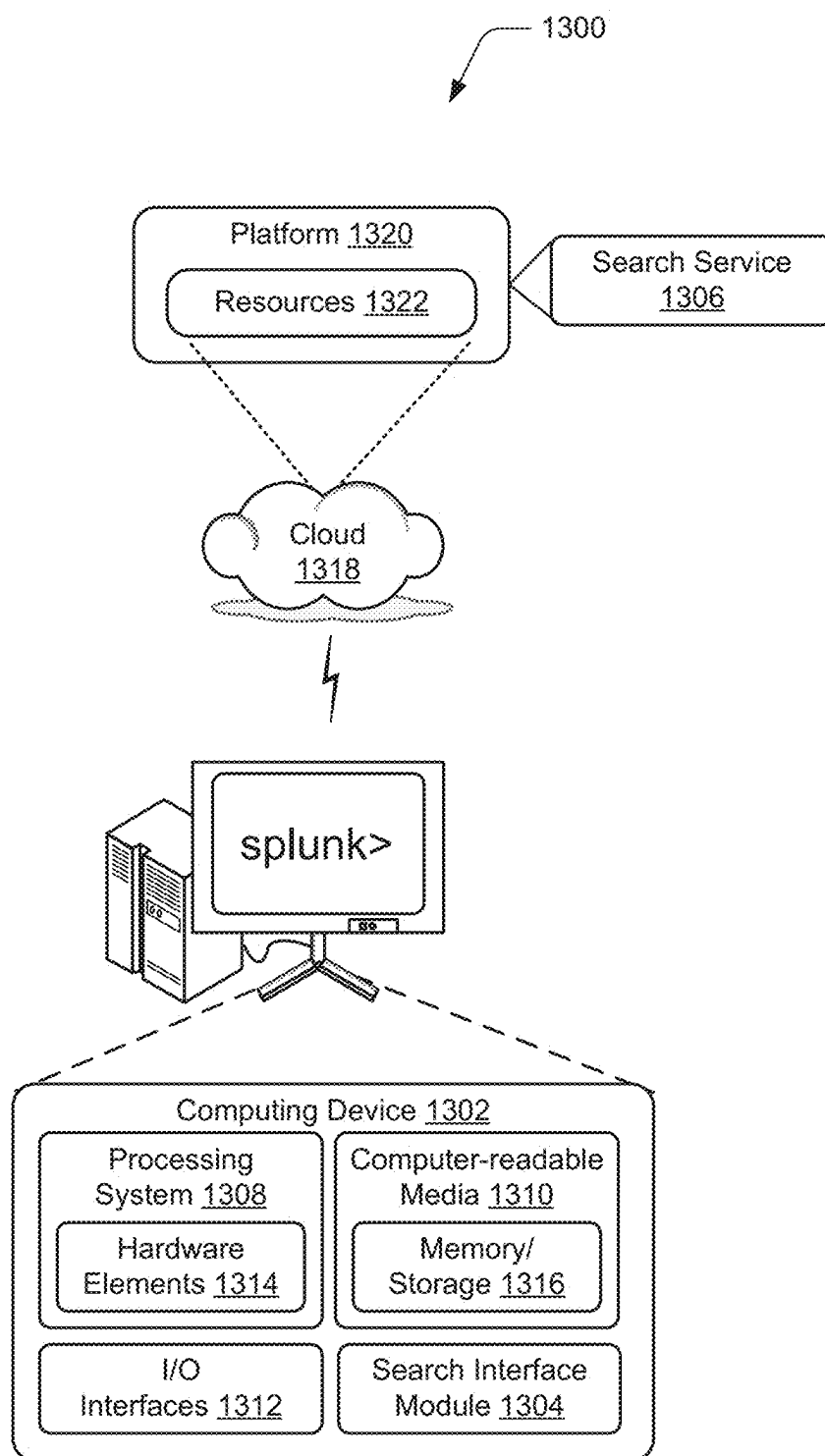


FIG. 13

PRESENTING EVENTS BASED ON USER-SELECTED FIELDS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a Continuation of U.S. patent application Ser. No. 18/403,594, filed on Jan. 3, 2024, which itself is a Continuation of U.S. patent application Ser. No. 17/952,950, filed on Sep. 26, 2022, which itself is a continuation of U.S. Pat. No. 11,455,087, filed Apr. 29, 2019, which is itself a Continuation of U.S. Pat. No. 10,303,344, filed Oct. 28, 2014. U.S. Pat. No. 10,303,344 claims priority to U.S. Provisional Patent Application Nos. 62/059,988, 62/059,989, 62/059,993, 62/059,944, 62/059,998, and 62/060,001 all filed Oct. 5, 2014. U.S. patent application Ser. No. 14/526,406 also claims priority to U.S. Provisional Patent Application Nos. 62/060,545, 62/060,551, 62/060,560, and 62/060,567 all filed Oct. 6, 2014. The contents of each of the foregoing applications are incorporated by reference herein in their entirety.

BACKGROUND

[0002] Data analysts for many businesses face the challenge of making sense of and finding patterns in the increasingly large amounts of data in the many types and formats that such businesses generate and collect. For example, accessing computer networks and transmitting electronic communications across the networks generates massive amounts of data, including such types of data as machine data and Web logs. Identifying patterns in this data, once thought relatively useless, has proven to be of great value to the businesses. In some instances, pattern analysis can indicate which patterns are normal and which ones are unusual. For example, detecting unusual patterns can allow a computer system manager to investigate the circumstances and determine whether a computer system security threat exists.

[0003] Additionally, analysis of the data allows businesses to understand how their employees, potential consumers, and/or Web visitors use the company's online resources. Such analysis can provide businesses with operational intelligence, business intelligence, and an ability to better manage their IT resources. For instance, such analysis may enable a business to better retain customers, meet customer needs, or improve the efficiency of the company's IT resources. Despite the value that one can derive from the underlying data described, making sense of this data to realize that value takes effort. In particular, patterns in underlying data may be difficult to identify or understand when analyzing specific behaviors in isolation, often resulting in the failure of a data analyst to notice valuable correlations in the data from which a business can draw strategic insight.

SUMMARY

[0004] This Summary introduces features and concepts of field value search drill down, which is further described below in the Detailed Description and/or shown in the Figures. This Summary should not be considered to describe essential features of the claimed subject matter, nor used to determine or limit the scope of the claimed subject matter.

[0005] Field value search drill down is described. In embodiments, a search system exposes a search interface

that displays one or more events returned as a search result set. A field-value pair can be emphasized in the field-value pairs of an event displayed in the search interface, and a menu is displayed with search options that are selectable to operate on the emphasized field-value pair of the event. The menu includes the search options to add search criteria of the emphasized field-value pair to a search command in a search bar of the search interface, exclude the search criteria of the emphasized field-value pair from a search, or create a new data search based on the emphasized field-value pair. A selection of one of the search options in the menu can be received, and the search command in the search bar is updated based on the search option that is selected for the emphasized field-value pair.

[0006] In implementations, the search interface is an event field-picker interface that displays a listing of multiple field-value pairs of the event. The field-value pair in the event can be highlighted responsive to detection of an input pointer over the field-value pair. An input associated with the emphasized field-value pair can be received, such as when initiated by a user in the search interface, and the menu of the search options is displayed proximate the emphasized field-value pair in the search interface. For example, the menu may pop-up or drop-down just below the emphasized field-value pair. A received input that is associated with the emphasized field-value pair initiates a display of the menu with the search options that are selectable to operate on the emphasized field-value pair. The search options that are displayed in the menu include an add to search option, an exclude from search option, and a new search option. The menu also includes a first statistical event count that indicates a number of multiple events that include the emphasized field-value pair, and the menu includes a second statistical event count that indicates a number of multiple events that exclude the emphasized field-value pair.

[0007] In embodiments, the search system can receive a selection of the search option to add search criteria of the emphasized field-value pair to a data search, and update the search command in the search bar to include the search criteria of the emphasized field-value pair. The search system can then perform the data search based on the updated search command to determine additional events that each include the search criteria of the emphasized field-value pair, and display the additional events that each include the emphasized field-value pair in the search interface. Alternatively, the search system can receive a selection of the search option to exclude search criteria of the emphasized field-value pair from a data search, and update the search command in the search bar to exclude the search criteria of the emphasized field-value pair. The search system can then perform the data search based on the updated search command to determine additional events that do not include the search criteria of the emphasized field-value pair, and display the additional events that do not include the emphasized field-value pair. Alternatively, the search system can receive a selection of the search option to create a new data search based on the emphasized field-value pair, and update the search command in the search bar to replace the search command with search criteria of the emphasized field-value pair. The search system can then perform the new data search based on the updated search command to determine additional events that include the search criteria of the emphasized field-value pair, and display the additional events that include the emphasized field-value pair.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments of field value search drill down are described with reference to the following Figures. The same numbers may be used throughout to reference like features and components that are shown in the Figures:

[0009] FIG. 1 illustrates a block diagram of an event-processing system in accordance with the disclosed implementations of field value search drill down.

[0010] FIG. 2 illustrates a flowchart of how indexers process, index, and store data received from forwarders in accordance with the disclosed implementations.

[0011] FIG. 3 illustrates a flowchart of how a search head and indexers perform a search query in accordance with the disclosed implementations.

[0012] FIG. 4 illustrates a block diagram of a system for processing search requests that uses extraction rules for field values in accordance with the disclosed implementations.

[0013] FIG. 5 illustrates an exemplary search query received from a client and executed by search peers in accordance with the disclosed implementations.

[0014] FIG. 6A illustrates a search screen in accordance with the disclosed implementations.

[0015] FIG. 6B illustrates a data summary dialog that enables a user to select various data sources in accordance with the disclosed implementations.

[0016] FIG. 7A illustrates a key indicators view in accordance with the disclosed implementations.

[0017] FIG. 7B illustrates an incident review dashboard in accordance with the disclosed implementations.

[0018] FIG. 7C illustrates a proactive monitoring tree in accordance with the disclosed implementations.

[0019] FIG. 7D illustrates a screen displaying both log data and performance data in accordance with the disclosed implementations.

[0020] FIGS. 8A-8G illustrate examples of a search interface in accordance with the disclosed implementations.

[0021] FIG. 9A illustrates an example of a search interface in accordance with the disclosed implementations.

[0022] FIG. 9B illustrates an example of an extract fields interface in accordance with the disclosed implementations.

[0023] FIGS. 10A and 10B illustrate examples of a search interface in accordance with the disclosed implementations.

[0024] FIGS. 11A-11D illustrate example method(s) of event segment search drill down in accordance with one or more embodiments.

[0025] FIGS. 12A-12D illustrate example method(s) of field value search drill down in accordance with one or more embodiments.

[0026] FIG. 13 illustrates an example system with an example device that can implement embodiments of field value search drill down.

DETAILED DESCRIPTION

[0027] Embodiments of event segment search drill down and field value search drill down are described and can be implemented to facilitate user-initiated search options when performing data searches in search interfaces that include events, highlighted segments in event raw data of the events, values of field-value pairs in the events, and tagged field-value pairs in the events. Although described in the context of an event segment that is highlighted or otherwise visually emphasized in event raw data of a displayed event, the techniques described herein can be implemented and applied

to any text selection, alphanumeric selection, or searched text and/or alphanumeric string.

[0028] In embodiments, a segment in the event raw data of an event can be highlighted (or otherwise emphasized) and a contextual search menu is displayed with search options that are selectable to operate on the highlighted segment. Similarly, a field-value pair in an event can be emphasized (e.g., highlighted or any other type of visual emphasis) and a field value contextual menu is displayed with search options that are selectable to operate on the emphasized field-value pair.

[0029] The contextual search menu and the field value contextual menu includes search options, such as to add the highlighted segment as a new keyword to a search command in a search bar of the search interface, add the keyword that represents the highlighted segment to the search command to exclude the highlighted segment from a search, or create a new data search based on the keyword that represents the highlighted segment. Similarly, the search options include an option to add search criteria of the emphasized field-value pair to the search command in the search bar of the search interface, add the search criteria of the emphasized field-value pair to the search command as the search criteria excluded from events that do not include the emphasized field-value pair, or create a new data search based on the emphasized field-value pair, where the search criteria of the emphasized field-value pair replaces the search command in the search bar. The user can select one of the search options in the contextual search menu or field value contextual menu, and the search command in the search bar of the search interface is updated based on the search option that is selected for the highlighted segment or emphasized field-value pair.

Example Environment

[0030] Modern data centers often comprise thousands of host computer systems that operate collectively to service requests from even larger numbers of remote clients. During operation, these data centers generate significant volumes of performance data and diagnostic information that can be analyzed to quickly diagnose performance problems. In order to reduce the size of this performance data, the data is typically pre-processed prior to being stored based on anticipated data-analysis needs. For example, pre-specified data items can be extracted from the performance data and stored in a database to facilitate efficient retrieval and analysis at search time. However, the rest of the performance data is not saved and is essentially discarded during pre-processing. As storage capacity becomes progressively cheaper and more plentiful, there are fewer incentives to discard this performance data and many reasons to keep it.

[0031] This plentiful storage capacity is presently making it feasible to store massive quantities of minimally processed performance data at “ingestion time” for later retrieval and analysis at “search time.” Note that performing the analysis operations at search time provides greater flexibility because it enables an analyst to search all of the performance data, instead of searching pre-specified data items that were stored at ingestion time. This enables the analyst to investigate different aspects of the performance data instead of being confined to the pre-specified set of data items that were selected at ingestion time.

[0032] However, analyzing massive quantities of heterogeneous performance data at search time can be a challeng-

ing task. A data center may generate heterogeneous performance data from thousands of different components, which can collectively generate tremendous volumes of performance data that can be time-consuming to analyze. For example, this performance data can include data from system logs, network packet data, sensor data, and data generated by various applications. Also, the unstructured nature of much of this performance data can pose additional challenges because of the difficulty of applying semantic meaning to unstructured data, and the difficulty of indexing and querying unstructured data using traditional database systems.

[0033] These challenges can be addressed by using an event-based system, such as the SPLUNK® ENTERPRISE system produced by Splunk Inc. of San Francisco, California, to store and process performance data. The SPLUNK® ENTERPRISE system is the leading platform for providing real-time operational intelligence that enables organizations to collect, index, and harness machine-generated data from various websites, applications, servers, networks, and mobile devices that power their businesses. The SPLUNK® ENTERPRISE system is particularly useful for analyzing unstructured performance data, which is commonly found in system log files. Although many of the techniques described herein are explained with reference to the SPLUNK® ENTERPRISE system, the techniques are also applicable to other types of data server systems.

[0034] In the SPLUNK® ENTERPRISE system, performance data is stored as “events,” in which each event comprises a collection of performance data and/or diagnostic information that is generated by a computer system and is correlated with a specific point in time. Events can be derived from “time series data,” in which time series data includes a sequence of data points (e.g., performance measurements from a computer system) that are associated with successive points in time and are typically spaced at uniform time intervals. Events can also be derived from “structured” or “unstructured” data. Structured data has a predefined format, in which specific data items with specific data formats reside at predefined locations in the data. For example, structured data can include data items stored in fields in a database table. In contrast, unstructured data does not have a predefined format. This means that unstructured data can include various data items having different data types that can reside at different locations. For example, when the data source is an operating system log, an event can include one or more lines from the operating system log containing raw data that includes different types of performance and diagnostic information associated with a specific point in time.

[0035] Examples of data sources from which an event may be derived include, but are not limited to web servers, application servers, databases, firewalls, routers, operating systems, and software applications that execute on computer systems, mobile devices, and sensors. The data generated by such data sources can be produced in various forms including, for example and without limitation, server log files, activity log files, configuration files, messages, network packet data, performance measurements and sensor measurements. An event typically includes a timestamp that may be derived from the raw data in the event, or may be determined through interpolation between temporally proximate events having known timestamps.

[0036] The SPLUNK® ENTERPRISE system also facilitates using a flexible schema to specify how to extract information from the event data, in which the flexible schema may be developed and redefined as needed. Note that a flexible schema may be applied to event data “on the fly” as desired (e.g., at search time), rather than at ingestion time of the data as in traditional database systems. Because the schema is not applied to event data until it is desired (e.g., at search time), it is referred to as a “late-binding schema.”

[0037] During operation, the SPLUNK® ENTERPRISE system starts with raw data, which can include unstructured data, machine data, performance measurements or other time-series data, such as data obtained from weblogs, syslogs, or sensor readings. It divides this raw data into “portions,” and optionally transforms the data to produce time-stamped events. The system stores the timestamped events in a data store, and enables a user to run queries against the data store to retrieve events that meet specified criteria, such as containing certain keywords or having specific values in defined fields. Note that the term “field” refers to a location in the event data containing a value for a specific data item.

[0038] As noted above, the SPLUNK® ENTERPRISE system facilitates using a late-binding schema while performing queries on events. A late-binding schema specifies “extraction rules” that are applied to data in the events to extract values for specific fields. More specifically, the extraction rules for a field can include one or more instructions that specify how to extract a value for the field from the event data. An extraction rule can generally include any type of instruction for extracting values from data in events. In some cases, an extraction rule includes a regular expression, in which case the rule is referred to as a “regex rule.”

[0039] In contrast to a conventional schema for a database system, a late-binding schema is not defined at data ingestion time. Instead, the late-binding schema can be developed on an ongoing basis until the time a query is actually executed. This means that extraction rules for the fields in a query may be provided in the query itself, or may be located during execution of the query. Hence, as an analyst learns more about the data in the events, the analyst can continue to refine the late-binding schema by adding new fields, deleting fields, or changing the field extraction rules until the next time the schema is used by a query. Because the SPLUNK® ENTERPRISE system maintains the underlying raw data and provides a late-binding schema for searching the raw data, it enables an analyst to investigate questions that arise as the analyst learns more about the events.

[0040] In the SPLUNK® ENTERPRISE system, a field extractor may be configured to automatically generate extraction rules for certain fields in the events when the events are being created, indexed, or stored, or possibly at a later time. Alternatively, a user may manually define extraction rules for fields using a variety of techniques. Also, a number of “default fields” that specify metadata about the events, rather than data in the events themselves, can be created automatically. For example, such default fields can specify: a timestamp for the event data; a host from which the event data originated; a source of the event data; and a source type for the event data. These default fields may be determined automatically when the events are created, indexed, or stored.

[0041] In some embodiments, a common field name may be used to reference two or more fields containing equivalent

lent data items, even though the fields may be associated with different types of events that possibly have different data formats and different extraction rules. By enabling a common field name to be used to identify equivalent fields from different types of events generated by different data sources, the system facilitates use of a “common information model” (CIM) across the different data sources.

Data Server System

[0042] FIG. 1 illustrates a block diagram of an example event-processing system 100, similar to the SPLUNK® ENTERPRISE system, and in which embodiments of event segment search drill down can be implemented. The example event-processing system 100 includes one or more forwarders 101 that collect data obtained from a variety of different data sources 105, and one or more indexers 102 that store, process, and/or perform operations on this data, in which each indexer operates on data contained in a specific data store 103. A search head 104 may also be provided that represents functionality to obtain and process search requests from clients and provide results of the search back to the clients, additional details of which are discussed in relation to FIGS. 3 and 4. The forwarders 101, indexers 102, and/or search head 104 may be configured as separate computer systems in a data center, or alternatively may be configured as separate processes implemented via one or more individual computer systems. Data that is collected via the forwarders 101 may be obtained from a variety of different data sources 105.

[0043] As further illustrated, the search head 104 may interact with a client application module 106 associated with a client device, such as to obtain search queries and supply search results or other suitable data back to the client application module 106 that is effective to enable the client application module 106 to form search user interfaces 108 through which different views of the data may be exposed. Various examples and details regarding search interfaces 108, client application modules 106, search queries, and operation of the various components illustrated in FIG. 1 are discussed throughout this document.

[0044] During operation, the forwarders 101 identify which indexers 102 will receive the collected data and then forward the data to the identified indexers. The forwarders 101 can also perform operations to strip out extraneous data and detect timestamps in the data. The forwarders next determine which of the indexers 102 will receive each data item and then forward the data items to the determined indexers 102. Note that distributing data across the different indexers 102 facilitates parallel processing. This parallel processing can take place at data ingestion time, because multiple indexers can process the incoming data in parallel. The parallel processing can also take place at search time, because multiple indexers can search through the data in parallel.

[0045] The example event-processing system 100 and the processes described below with respect to FIGS. 1-5 are further described in “Exploring Splunk Search Processing Language (SPL) Primer and Cookbook” by David Carasso, CITO Research, 2012, and in “Optimizing Data Analysis With a Semi-Structured Time Series Database” by Ledion Bitincka, Archana Ganapathi, Stephen Sorkin, and Steve Zhang, SLAML, 2010, each of which is hereby incorporated herein by reference in its entirety for all purposes.

Data Ingestion

[0046] FIG. 2 illustrates a flowchart 200 of how an indexer processes, indexes, and stores data received from forwarders in accordance with the disclosed embodiments. At block 201, the indexer receives the data from the forwarder. Next, at block 202, the indexer apportions the data into events. Note that the data can include lines of text that are separated by carriage returns or line breaks and an event may include one or more of these lines. During the apportioning process, the indexer can use heuristic rules to automatically determine the boundaries of the events, which for example coincide with line boundaries. These heuristic rules may be determined based on the source of the data, in which the indexer can be explicitly informed about the source of the data or can infer the source of the data by examining the data. These heuristic rules can include regular expression-based rules or delimiter-based rules for determining event boundaries, in which the event boundaries may be indicated by predefined characters or character strings. These predefined characters may include punctuation marks or other special characters including, for example, carriage returns, tabs, spaces or line breaks. In some cases, a user can fine-tune or configure the rules that the indexers use to determine event boundaries in order to adapt the rules to the user’s specific requirements.

[0047] Next, the indexer determines a timestamp for each event at block 203. As mentioned above, these timestamps can be determined by extracting the time directly from data in the event, or by interpolating the time based on timestamps from temporally proximate events. In some cases, a timestamp can be determined based on the time the data was received or generated. The indexer subsequently associates the determined timestamp with each event at block 204, for example by storing the timestamp as metadata for each event.

[0048] Then, the system can apply transformations to data to be included in events at block 205. For log data, such transformations can include removing a portion of an event (e.g., a portion used to define event boundaries, extraneous text, characters, etc.) or removing redundant portions of an event. Note that a user can specify portions to be removed using a regular expression or any other possible technique.

[0049] Next, a keyword index can optionally be generated to facilitate fast keyword searching for events. To build a keyword index, the indexer first identifies a set of keywords in block 206. Then, at block 207 the indexer includes the identified keywords in an index, which associates each stored keyword with references to events containing that keyword (or to locations within events where that keyword is located). When an indexer subsequently receives a keyword-based query, the indexer can access the keyword index to quickly identify events containing the keyword.

[0050] In some embodiments, the keyword index may include entries for name-value pairs found in events, wherein a name-value pair can include a pair of keywords connected by a symbol, such as an equals sign or colon. In this way, events containing these name-value pairs can be quickly located. In some embodiments, fields can automatically be generated for some or all of the name-value pairs at the time of indexing. For example, if the string “dest=10.0.1.2” is found in an event, a field named “dest” may be created for the event, and assigned a value of “10.0.1.2” as a field-value pair.

[0051] Finally, the indexer stores the events in a data store at block 208, where a timestamp can be stored with each event to facilitate searching for events based on a time range. In some cases, the stored events are organized into a plurality of buckets, where each bucket stores events associated with a specific time range. This not only improves time-based searches, but it also allows events with recent timestamps that may have a higher likelihood of being accessed to be stored in faster memory to facilitate faster retrieval. For example, a bucket containing the most recent events can be stored as flash memory instead of on a hard disk.

[0052] Each indexer 102 is responsible for storing and searching a subset of the events contained in a corresponding data store 103. By distributing events among the indexers and data stores, the indexers can analyze events for a query in parallel, for example using map-reduce techniques, in which each indexer returns partial responses for a subset of events to a search head that combines the results to produce an answer for the query. By storing events in buckets for specific time ranges, an indexer may further optimize searching by looking only in buckets for time ranges that are relevant to a query. Moreover, events and buckets can also be replicated across different indexers and data stores to facilitate high availability and disaster recovery as is described in U.S. patent application Ser. No. 14/266,812 filed on 30 Apr. 2014, and in U.S. patent application Ser. No. 14/266,817 also filed on 30 Apr. 2014.

Query Processing

[0053] FIG. 3 illustrates a flowchart 300 of how a search head and indexers perform a search query in accordance with the disclosed embodiments. At the start of this process, a search head receives a search query from a client (e.g., a client computing device) at block 301. Next, at block 302, the search head analyzes the search query to determine what portions can be delegated to indexers and what portions need to be executed locally by the search head. At block 303, the search head distributes the determined portions of the query to the indexers. Note that commands that operate on single events can be trivially delegated to the indexers, while commands that involve events from multiple indexers are harder to delegate.

[0054] Then, at block 304, the indexers to which the query was distributed search their data stores for events that are responsive to the query. To determine which events are responsive to the query, the indexer searches for events that match the criteria specified in the query. This criteria can include matching keywords or specific values for certain fields. In a query that uses a late-binding schema, the searching operations in block 304 may involve using the late-binding scheme to extract values for specified fields from events at the time the query is processed. Next, the indexers can either send the relevant events back to the search head, or use the events to calculate a partial result, and send the partial result back to the search head.

[0055] Finally, at block 305, the search head combines the partial results and/or events received from the indexers to produce a final result for the query. This final result can comprise different types of data depending on what the query is asking for. For example, the final results can include a listing of matching events returned by the query, or some type of visualization of data from the returned events. In

another example, the final result can include one or more calculated values derived from the matching events.

[0056] Moreover, the results generated by system 100 can be returned to a client using different techniques. For example, one technique streams results back to a client in real-time as they are identified. Another technique waits to report results to the client until a complete set of results is ready to return to the client. Yet another technique streams interim results back to the client in real-time until a complete set of results is ready, and then returns the complete set of results to the client. In another technique, certain results are stored as “search jobs,” and the client may subsequently retrieve the results by referencing the search jobs.

[0057] The search head can also perform various operations to make the search more efficient. For example, before the search head starts executing a query, the search head can determine a time range for the query and a set of common keywords that all matching events must include. Next, the search head can use these parameters to query the indexers to obtain a superset of the eventual results. Then, during a filtering stage, the search head can perform field-extraction operations on the superset to produce a reduced set of search results.

Field Extraction

[0058] FIG. 4 illustrates a block diagram 400 of how fields can be extracted during query processing in accordance with the disclosed embodiments. At the start of this process, a search query 402 is received at a query processor 404. The query processor 404 includes various mechanisms for processing a query, where these mechanisms can reside in a search head 104 and/or an indexer 102. Note that the exemplary search query 402 illustrated in FIG. 4 is expressed in Search Processing Language (SPL), which is used in conjunction with the SPLUNK® ENTERPRISE system. The SPL is a pipelined search language in which a set of inputs is operated on by a first command in a command line, and then a subsequent command following the pipe symbol “|” operates on the results produced by the first command, and so on for additional commands. Search query 402 can also be expressed in other query languages, such as the Structured Query Language (“SQL”) or any suitable query language.

[0059] Upon receiving the search query 402, the query processor 404 identifies that the search query 402 includes two fields, “IP” and “target.” The query processor 404 also determines that the values for the “IP” and “target” fields have not already been extracted from events in a data store 414, and consequently determines that the query processor 404 needs to use extraction rules to extract values for the fields. Hence, the query processor 404 performs a lookup for the extraction rules in a rule base 406, in which rule base 406 maps field names to corresponding extraction rules and obtains extraction rules 408 and 409, where extraction rule 408 specifies how to extract a value for the “IP” field from an event, and extraction rule 409 specifies how to extract a value for the “target” field from an event.

[0060] As is illustrated in FIG. 4, the extraction rules 408 and 409 can include regular expressions that specify how to extract values for the relevant fields. Such regular-expression-based extraction rules are also referred to as “regex rules.” In addition to specifying how to extract field values, the extraction rules may also include instructions for deriving a field value by performing a function on a character

string or value retrieved by the extraction rule. For example, a transformation rule may truncate a character string, or convert the character string into a different data format. In some cases, the query itself can specify one or more extraction rules.

[0061] Next, the query processor 404 sends the extraction rules 408 and 409 to a field extractor 412, which applies the extraction rules 408 and 409 to events 416-418 in the data store 414. Note that the data store 414 can include one or more data stores, and the extraction rules 408 and 409 can be applied to large numbers of events in the data store 414, and are not meant to be limited to the three events 416-418 illustrated in FIG. 4. Moreover, the query processor 404 can instruct the field extractor 412 to apply the extraction rules to all of the events in the data store 414, or to a subset of the events that have been filtered based on some criteria.

[0062] Next, the field extractor 412 applies the extraction rule 408 for the first command “Search IP=“10*”” to events in the data store 414, including the events 416-418. The extraction rule 408 is used to extract values for the IP address field from events in the data store 414 by looking for a pattern of one or more digits, followed by a period, followed again by one or more digits, followed by another period, followed again by one or more digits, followed by another period, and followed again by one or more digits. Next, the field extractor 412 returns field values 420 to the query processor 404, which uses the criterion IP=“10*”” to look for IP addresses that start with “10”. Note that events 416 and 417 match this criterion, but event 418 does not, so the result set for the first command is events 416 and 417.

[0063] The query processor 404 then sends the events 416 and 417 to the next command “stats count target.” To process this command, the query processor 404 causes the field extractor 412 to apply the extraction rule 409 to the events 416 and 417. The extraction rule 409 is used to extract values for the target field for the events 416 and 417 by skipping the first four commas in the events, and then extracting all of the following characters until a comma or period is reached. Next, the field extractor 412 returns field values 421 to the query processor 404, which executes the command “stats count target” to count the number of unique values contained in the target fields, which in this example produces the value “2” that is returned as a final result 422 for the query.

[0064] Note that query results can be returned to a client, a search head, or any other system component for further processing. In general, the query results may include: a set of one or more events; a set of one or more values obtained from the events; a subset of the values; statistics calculated based on the values; a report containing the values; or a visualization, such as a graph or chart, generated from the values.

Example Search Screen

[0065] FIG. 6A illustrates an example of a search screen 600 in accordance with the disclosed embodiments. The search screen 600 includes a search bar 602 that accepts user input in the form of a search string. It also includes a date time range picker 612 that enables the user to specify a date and/or time range for the search. For “historical searches” the user can select a specific time range, or alternatively a relative time range, such as “today,” “yesterday,” or “last week.” For “real-time searches,” the user can select the size of a preceding time window to search for real-time events.

The search screen 600 also initially displays a “data summary” dialog 610 as is illustrated in FIG. 6B that enables the user to select different sources for the event data, such as by selecting specific hosts and log files.

[0066] After the search is executed, the search screen 600 can display the results through search results tabs 604, where the search results tabs 604 include: an “Events” tab that displays various information about events returned by the search; a “Patterns” tab that can be selected to display various patterns about the events returned by the search; a “Statistics” tab that displays statistics about the search results and events; and a “Visualization” tab that displays various visualizations of the search results. The “Events” tab illustrated in FIG. 6A displays a timeline graph 605 that graphically illustrates the number of events that occurred in one-hour intervals over the selected time range. It also displays an events list 608 that enables a user to view the raw data in each of the returned events. It additionally displays a fields sidebar 606 that includes statistics about occurrences of specific fields in the returned events, including “selected fields” that are pre-selected by the user, and “interesting fields” that are automatically selected by the system based on pre-specified criteria.

Acceleration Techniques

[0067] The above-described system provides significant flexibility by enabling a user to analyze massive quantities of minimally processed performance data “on the fly” at search time instead of storing pre-specified portions of the performance data in a database at ingestion time. This flexibility enables a user to see correlations in the performance data and perform subsequent queries to examine interesting aspects of the performance data that may not have been apparent at ingestion time.

[0068] However, performing extraction and analysis operations at search time can involve a large amount of data and require a large number of computational operations, which can cause considerable delays while processing the queries. Fortunately, a number of acceleration techniques have been developed to speed up analysis operations performed at search time. These techniques include: (1) performing search operations in parallel by formulating a search as a map-reduce computation; (2) using a keyword index; (3) using a high performance analytics store; and (4) accelerating the process of generating reports. These techniques are described in more detail below.

Map-Reduce Technique

[0069] To facilitate faster query processing, a query can be structured as a map-reduce computation, wherein the “map” operations are delegated to the indexers, while the corresponding “reduce” operations are performed locally at the search head. For example, FIG. 5 illustrates an example 500 of how a search query 501 received from a client at search head 104 can split into two phases, including: (1) a “map phase” comprising subtasks 502 (e.g., data retrieval or simple filtering) that may be performed in parallel and are “mapped” to indexers 102 for execution, and (2) a “reduce phase” comprising a merging operation 503 to be executed by the search head 104 when the results are ultimately collected from the indexers.

[0070] During operation, upon receiving search query 501, search head 104 modifies search query 501 by substi-

tuting “stats” with “prestats” to produce search query **502**, and then distributes search query **502** to one or more distributed indexers, which are also referred to as “search peers.” Note that search queries may generally specify search criteria or operations to be performed on events that meet the search criteria. Search queries may also specify field names, as well as search criteria for the values in the fields or operations to be performed on the values in the fields. Moreover, the search head may distribute the full search query to the search peers as is illustrated in FIG. 3, or may alternatively distribute a modified version (e.g., a more restricted version) of the search query to the search peers. In this example, the indexers are responsible for producing the results and sending them to the search head. After the indexers return the results to the search head, the search head performs the merging operations **503** on the results. Note that by executing the computation in this way, the system effectively distributes the computational operations while minimizing data transfers.

Keyword Index

[0071] As described above with reference to the flow charts **200** and **300** shown in respective FIGS. 2 and 3, the event-processing system **100** can construct and maintain one or more keyword indices to facilitate rapidly identifying events containing specific keywords. This can greatly speed up the processing of queries involving specific keywords. As mentioned above, to build a keyword index, an indexer first identifies a set of keywords. Then, the indexer includes the identified keywords in an index, which associates each stored keyword with references to events containing that keyword, or to locations within events where that keyword is located. When an indexer subsequently receives a keyword-based query, the indexer can access the keyword index to quickly identify events containing the keyword.

High Performance Analytics Store

[0072] To speed up certain types of queries, some embodiments of system **100** make use of a high-performance analytics store, which is referred to as a “summarization table,” that contains entries for specific field-value pairs. Each of these entries keeps track of instances of a specific value in a specific field in the event data and includes references to events containing the specific value in the specific field. For example, an entry in a summarization table can keep track of occurrences of the value “94107” in a “ZIP code” field of a set of events, where the entry includes references to all of the events that contain the value “94107” in the ZIP code field. This enables the system to quickly process queries that seek to determine how many events have a particular value for a particular field, because the system can examine the entry in the summarization table to count instances of the specific value in the field without having to go through the individual events or do extractions at search time. Also, if the system needs to process each of the events that have a specific field-value combination, the system can use the references in the summarization table entry to directly access the events to extract further information without having to search each of the events to find the specific field-value combination at search time.

[0073] In some embodiments, the system maintains a separate summarization table for each of the above-described time-specific buckets that stores events for a specific

time range, where a bucket-specific summarization table includes entries for specific field-value combinations that occur in events in the specific bucket. Alternatively, the system can maintain a separate summarization table for each indexer, in which the indexer-specific summarization table only includes entries for the events in a data store that is managed by the specific indexer.

[0074] The summarization table can be populated by running a “collection query” that scans a set of events to find instances of a specific field-value combination, or alternatively instances of all field-value combinations for a specific field. A collection query can be initiated by a user, or can be scheduled to occur automatically at specific time intervals. A collection query can also be automatically launched in response to a query that asks for a specific field-value combination.

[0075] In some cases, the summarization tables may not cover each of the events that are relevant to a query. In this case, the system can use the summarization tables to obtain partial results for the events that are covered by summarization tables, but may also have to search through other events that are not covered by the summarization tables to produce additional results. These additional results can then be combined with the partial results to produce a final set of results for the query. This summarization table and associated techniques are described in more detail in U.S. Pat. No. 8,682,925, issued on Mar. 25, 2014.

Accelerating Report Generation

[0076] In some embodiments, a data server system such as the SPLUNK® ENTERPRISE system can accelerate the process of periodically generating updated reports based on query results. To accelerate this process, a summarization engine automatically examines the query to determine whether generation of updated reports can be accelerated by creating intermediate summaries. This is possible if results from preceding time periods can be computed separately and combined to generate an updated report. In some cases, it is not possible to combine such incremental results, for example where a value in the report depends on relationships between events from different time periods. If reports can be accelerated, the summarization engine periodically generates a summary covering data obtained during a latest non-overlapping time period. For example, where the query seeks events meeting a specified criteria, a summary for the time period includes only the events within the time period that meet the specified criteria. Similarly, if the query seeks statistics calculated from the events, such as the number of events that match the specified criteria, then the summary for the time period includes the number of events in the period that match the specified criteria.

[0077] In parallel with the creation of the summaries, the summarization engine schedules the periodic updating of the report associated with the query. During each scheduled report update, the query engine determines whether intermediate summaries have been generated covering portions of the time period covered by the report update. If so, then the report is generated based on the information contained in the summaries. Also, if additional event data has been received and has not yet been summarized, and is required to generate the complete report, the query can be run on this additional event data. Then, the results returned by this query on the additional event data, along with the partial results obtained from the intermediate summaries, can be

combined to generate the updated report. This process is repeated each time the report is updated.

[0078] Alternatively, if the system stores events in buckets covering specific time ranges, then the summaries can be generated on a bucket-by-bucket basis. Note that producing intermediate summaries can save the work involved in re-running the query for previous time periods, so only the newer event data needs to be processed while generating an updated report. These report acceleration techniques are described in more detail in U.S. Pat. No. 8,589,403, issued on Nov. 19, 2013, and in U.S. Pat. No. 8,412,696, issued on Apr. 2, 2011.

Security Features

[0079] The SPLUNK® ENTERPRISE platform provides various schemas, dashboards, and visualizations that make it easy for developers to create applications to provide additional capabilities. One such application is the SPLUNK® APP FOR ENTERPRISE SECURITY, which performs monitoring and alerting operations, and includes analytics to facilitate identifying both known and unknown security threats based on large volumes of data stored by the SPLUNK® ENTERPRISE system. This differs significantly from conventional Security Information and Event Management (SIEM) systems that lack the infrastructure to effectively store and analyze large volumes of security-related event data. Traditional SIEM systems typically use fixed schemas to extract data from pre-defined security-related fields at data ingestion time, where the extracted data is typically stored in a relational database. This data extraction process (and associated reduction in data size) that occurs at data ingestion time inevitably hampers future incident investigations, when all of the original data may be needed to determine the root cause of a security issue, or to detect the tiny fingerprints of an impending security threat.

[0080] In contrast, the SPLUNK® APP FOR ENTERPRISE SECURITY system stores large volumes of minimally processed security-related data at ingestion time for later retrieval and analysis at search time when a live security threat is being investigated. To facilitate this data retrieval process, the SPLUNK® APP FOR ENTERPRISE SECURITY provides pre-specified schemas for extracting relevant values from the different types of security-related event data, and also enables a user to define such schemas.

[0081] The SPLUNK® APP FOR ENTERPRISE SECURITY can process many types of security-related information. In general, this security-related information can include any information that can be used to identify security threats. For example, the security-related information can include network-related information, such as IP addresses, domain names, asset identifiers, network traffic volume, uniform resource locator strings, and source addresses. The process of detecting security threats for network-related information is further described in U.S. patent application Ser. Nos. 13/956,252, and 13/956,262. Security-related information can also include endpoint information, such as malware infection data and system configuration information, as well as access control information, such as login/logout information and access failure notifications. The security-related information can originate from various sources within a data center, such as hosts, virtual machines, storage devices, and sensors. The security-related information can also originate

from various sources in a network, such as routers, switches, email servers, proxy servers, gateways, firewalls and intrusion-detection systems.

[0082] During operation, the SPLUNK® APP FOR ENTERPRISE SECURITY facilitates detecting so-called “notable events” that are likely to indicate a security threat. These notable events can be detected in a number of ways: (1) an analyst can notice a correlation in the data and can manually identify a corresponding group of one or more events as “notable;” or (2) an analyst can define a “correlation search” specifying criteria for a notable event, and every time one or more events satisfy the criteria, the application can indicate that the one or more events are notable. An analyst can alternatively select a pre-defined correlation search provided by the application. Note that correlation searches can be run continuously or at regular intervals (e.g., every hour) to search for notable events. Upon detection, notable events can be stored in a dedicated “notable events index,” which can be subsequently accessed to generate various visualizations containing security-related information. Also, alerts can be generated to notify system operators when important notable events are discovered.

[0083] The SPLUNK® APP FOR ENTERPRISE SECURITY provides various visualizations to aid in discovering security threats, such as a “key indicators view” that enables a user to view security metrics of interest, such as counts of different types of notable events. For example, FIG. 7A illustrates an exemplary key indicators view **700** that comprises a dashboard, which can display a value **701**, for various security-related metrics, such as malware infections **702**. It can also display a change in a metric value **703**, which indicates that the number of malware infections increased by sixty-three (63) during the preceding interval. The key indicators view **700** additionally displays a histogram panel **704** that displays a histogram of notable events organized by urgency values, and a histogram panel **705** of notable events organized by time intervals. This key indicators view is described in further detail in pending U.S. patent application Ser. No. 13/956,338 filed Jul. 31, 2013.

[0084] These visualizations can also include an “incident review dashboard” that enables a user to view and act on “notable events.” These notable events can include: (1) a single event of high importance, such as any activity from a known web attacker; or (2) multiple events that collectively warrant review, such as a large number of authentication failures on a host followed by a successful authentication. For example, FIG. 7B illustrates an example of an incident review dashboard **710** that includes a set of incident attribute fields **711** that, for example, enables a user to specify a time range field **712** for the displayed events. It also includes a timeline **713** that graphically illustrates the number of incidents that occurred in one-hour time intervals over the selected time range. It additionally displays an events list **714** that enables a user to view a list of each of the notable events that match the criteria in the incident attributes fields **711**. To facilitate identifying patterns among the notable events, each notable event can be associated with an urgency value (e.g., low, medium, high, or critical), which is indicated in the incident review dashboard. The urgency value for a detected event can be determined based on the severity of the event and the priority of the system component associated with the event. The incident review dashboard is

described further on-line (e.g., at an HTTP://site), “docs.splunk.com/Documentation/PCI/2.1.1/User/Incident-Reviewdashboard.”

Data Center Monitoring

[0085] As mentioned above, the SPLUNK® ENTERPRISE platform provides various features that make it easy for developers to create various applications. One such application is the SPLUNK® APP FOR VMWARE®, which performs monitoring operations and includes analytics to facilitate diagnosing the root cause of performance problems in a data center based on large volumes of data stored by the SPLUNK® ENTERPRISE system.

[0086] This differs from conventional data-center-monitoring systems that lack the infrastructure to effectively store and analyze large volumes of performance information and log data obtained from the data center. In conventional data-center-monitoring systems, this performance data is typically pre-processed prior to being stored, for example by extracting pre-specified data items from the performance data and storing them in a database to facilitate subsequent retrieval and analysis at search time. However, the rest of the performance data is not saved and is essentially discarded during pre-processing. In contrast, the SPLUNK® APP FOR VMWARE® stores large volumes of minimally processed performance information and log data at ingestion time for later retrieval and analysis at search time when a live performance issue is being investigated.

[0087] The SPLUNK® APP FOR VMWARE® can process many types of performance-related information. In general, this performance-related information can include any type of performance-related data and log data produced by virtual machines and host computer systems in a data center. In addition to data obtained from various log files, this performance-related information can include values for performance metrics obtained through an application programming interface (API) provided as part of the vSphere Hypervisor™ system distributed by VMware, Inc. of Palo Alto, California. For example, these performance metrics can include: (1) CPU-related performance metrics; (2) disk-related performance metrics; (3) memory-related performance metrics; (4) network-related performance metrics; (5) energy-usage statistics; (6) data-traffic-related performance metrics; (7) overall system availability performance metrics; (8) cluster-related performance metrics; and (9) virtual machine performance statistics. For more details about such performance metrics, please see U.S. Pat. No. 14/167,316 filed 29 Jan. 2014, which is hereby incorporated herein by reference. Also, see “vSphere Monitoring and Performance,” Update 1, vSphere 5.5, EN-001357-00 on-line (e.g., at an HTTP://site), “pubs.vmware.com/vsphere-55/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-551-monitoring-performance-guide.pdf.”

[0088] To facilitate retrieving information of interest from performance data and log files, the SPLUNK® APP FOR VMWARE® provides pre-specified schemas for extracting relevant values from different types of performance-related event data, and also enables a user to define such schemas. The SPLUNK® APP FOR VMWARE® additionally provides various visualizations to facilitate detecting and diagnosing the root cause of performance problems. For example, one such visualization is a “proactive monitoring tree” that enables a user to easily view and understand relationships among various factors that affect the perfor-

mance of a hierarchically structured computing system. This proactive monitoring tree enables a user to easily navigate the hierarchy by selectively expanding nodes representing various entities (e.g., virtual centers or computing clusters) to view performance information for lower-level nodes associated with lower-level entities (e.g., virtual machines or host systems). Exemplary node-expansion operations are illustrated in FIG. 7C, where nodes 733 and 734 are selectively expanded. Note that the nodes 731-739 can be displayed using different patterns or colors to represent different performance states, such as a critical state, a warning state, a normal state, or an unknown/off-line state. The ease of navigation provided by selective expansion in combination with the associated performance-state information enables a user to quickly diagnose the root cause of a performance problem. The proactive monitoring tree is described in further detail in U.S. patent application Ser. No. 14/235,490 filed on 15 Apr. 2014, which is hereby incorporated herein by reference for all possible purposes.

[0089] The SPLUNK® APP FOR VMWARE® also provides a user interface that enables a user to select a specific time range and then view heterogeneous data, comprising events, log data, and associated performance metrics, for the selected time range. For example, the interface screen illustrated in FIG. 7D displays a listing of recent “tasks and events” and a listing of recent “log entries” for a selected time range above a performance-metric graph for “average CPU core utilization” for the selected time range. Note that a user is able to operate pull-down menus 742 to selectively display different performance metric graphs for the selected time range. This enables the user to correlate trends in the performance-metric graph with corresponding event and log data to quickly determine the root cause of a performance problem. This user interface is described in more detail in U.S. patent application Ser. No. 14/167,316 filed on 29 Jan. 2014, which is hereby incorporated herein by reference for all possible purposes.

Event Segment Search Drill Down

[0090] FIG. 8A illustrates an example of a search interface 800 displayed as a graphical user interface in accordance with the disclosed embodiments for event segment search drill down. The search interface 800 includes a search bar 802 that displays a search command 804, which is “sourcetype=access_combined” in this example. The search interface 800 also displays events 806 that are each correlated by a date and time 808. As described previously, the events 806 are a result set of performing the search command 804 that is currently displayed in the search bar 802, and only a subset of the events are shown in the search interface. A user can scroll the list of events 806 in the search interface 800 to view additional events of the search result set that are not displayed.

[0091] An event 810 (e.g., the first displayed event in the list of events 806) generally includes displayed event information, depending on a selected event view from which a user can select a format to display some or all of the event information for each of the events 806 in the search interface. In the example search interface 800, the events 806 are displayed in a list view, in which case the displayed event information for event 810 includes event raw data 812 displayed in an upper portion of the event display area, and includes field-value pairs 814 displayed in a lower portion of the event display area. The field-value pairs 814 correlate to

selected fields **820** that are also displayed in a fields sidebar **818**. In this example, each of the events **806** include “host=jmiller-mbpr15.sv.splunk.com” as a field-value pair **816**. The search interface **800** includes the fields sidebar **818**, which displays the selected fields **820** that are also displayed as the fields **816** for each of the events **806**, and the fields sidebar **818** also includes other interesting fields **822**.

[0092] In this example search interface **800**, a user may highlight any of the segments (e.g., terms or a combination of terms) in the event raw data **812**, such as “Mozilla/5.0” shown as the highlighted segment **824** in the event raw data **812** of the event **810**. In implementations, the SPLUNK® ENTERPRISE system includes a segmenter that is implemented to analyze the event raw data as a data string and determine which of the terms or combinations of terms are the contextually interesting segments that users (e.g., the data analysts) would most likely be interested in searching on or otherwise looking into. The segmenter identifies the segments (also referred to as terms or keywords) in the event raw data, and when a user moves a mouse pointer or other input device over a segment that the segmenter has identified, then the segment is highlighted in the display of the search interface.

[0093] In implementations, a segment may be highlighted or otherwise emphasized when a pointer that is displayed in the search interface **800** moves over a particular segment. This feature is also referred to as highlight with rollover (e.g., detected when a pointer moves over a segment). For example, a user may move a mouse pointer over the “Mozilla/5.0” segment, which is then displayed as the highlighted segment **824**. Alternatively, a user can highlight a segment in the event raw data **812** by initiating a selection of the segment, such as with a computer mouse, stylus, or other input device. A highlighted segment can then be selected in response to a user input, such as with a mouse click or touch input to select a particular segment.

[0094] The search interface **800** also includes an event field-picker toggle **826** that a user can select and initiate a transition to an alternate view of the search interface for the displayed event **810**, which is shown as an event field-picker interface **828** and further described with reference to FIG. **8B**. The event-limited field picker interface enables user selection of fields associated with individual events to display in the view of the events in the search interface.

[0095] FIG. **8B** illustrates an example of the event field-picker interface **828**, as an alternate view of the search interface **800** described with reference to FIG. **8A**. A user can transition from the displayed view of the search interface **800** (shown in FIG. **8A**) to the displayed view of the event field-picker interface **828** (shown in FIG. **8B**) by selecting the event field-picker toggle **826** that corresponds to the displayed event **810**. In this example, the event field-picker interface **828** includes a listing **830** of the field **832** and value **834** pairs in the event **810**. The event field-picker interface **828** is further described with reference to FIG. **10B** in accordance with the disclosed embodiments for field value search drill down.

[0096] FIG. **8C** further illustrates the example of the search interface **800** described with reference to FIG. **8A** in accordance with the disclosed embodiments for event segment search drill down. In this example display of the search interface **800**, a user has initiated the segment **824** being highlighted, such as with a mouse pointer moved over the “Mozilla/5.0” segment, which is displayed as the high-

lighted segment **824** in the event raw data **812**. Additionally, the user has selected the highlighted segment **824**, such as with a mouse click or touch input, and a contextual search menu **836** is displayed responsive to the user input. In implementations, the contextual search menu **836** is displayed proximate the highlighted segment **824** in the search interface **800**, such as a pop-up or drop-down menu just below the highlighted segment. Although described in the context of an event segment that is highlighted in event raw data of a displayed event, the techniques described herein can be implemented and applied to any text selection, alphanumeric selection, or searched text and/or alphanumeric string.

[0097] The contextual search menu **836** includes search options **838** that are selectable to operate on the highlighted segment **824** in the event raw data **812** of the displayed event **810**. For example, the search options **838** displayed in the contextual search menu **836** include: an option “Add to search” **840** that a user can select to add the highlighted segment **824** as a new keyword to the search command **804** in the search bar **802**; an option “Exclude from search” **842** that the user can select to exclude the keyword that represents the highlighted segment **824** from searches; and an option “New search” **844** that the user can select to create a new data search based on the highlighted segment **824** (e.g., replacing the search command **804** in the search bar **802** with the keyword that represents the highlighted segment **824**). A user selection of one of the search options **838** in the contextual search menu **836** can be received, and the search command **804** in the search bar **802** is updated based on the search option that is selected for the highlighted segment. In this example, the contextual search menu **836** also includes selectable interface links **846** that are each associated with a corresponding search option **838** in the search menu. A selectable interface link **846** for an associated search option can be selected by a user to initiate a new search interface.

[0098] FIG. **8D** further illustrates the example of the search interface **800** described with reference to FIGS. **8A** and **8C** in accordance with the disclosed embodiments for event segment search drill down. In this example display of the search interface **800**, a user has selected the option “Add to search” **840** from the contextual search menu **836** (shown in FIG. **8C**) to add the highlighted segment **824** as a new keyword to a data search, and update the search command **804** in the search bar **802** to include the keyword that represents the highlighted segment, which is shown as “Mozilla/5.0” added to the search command **804**. The search system can then perform the data search based on the updated search command **804** to determine the multiple events **806** that each include the keyword that represents the highlighted segment, and display an updated search result set of the events **806** that each include the highlighted segment in the search interface **800**. For example, each of the displayed events **806** in the search interface **800** include the highlighted “Mozilla/5.0” segment, as shown generally at **848**. Note that each of the displayed events **806** in the search interface **800** also include the rest of the search command **804** (e.g., “sourcetype=access_combined”) as a field-value pair **816**.

[0099] FIG. **8E** further illustrates the example of the search interface **800** described with reference to FIGS. **8A**, **8C**, and **8D**, in which the multiple events **806** that each include the highlighted “Mozilla/5.0” segment **824** are displayed, as shown generally at **848**. In this example display

of the search interface **800**, a user has selected the highlighted segment **824** (or any of the similar highlighted segments **848**), such as with a mouse click or touch input, and an additional search menu **850** is displayed responsive to the user input. In implementations, the additional search menu **850** is displayed proximate the highlighted segment **824** in the search interface **800**, such as a pop-up or drop-down menu just below the highlighted segment.

[0100] The additional search menu **850** includes search options **852** that are selectable to operate on the highlighted segment **824** in the event raw data **812** of the displayed event **810**. For example, the search options **852** displayed in the additional search menu **850** include: an option “Remove from search” **854** that a user can select to remove the keyword that represents the highlighted segment **824** from the search command **804** in the search bar **802**; and includes an option “New search” **856** that the user can select to create a new data search based on the highlighted segment **824** (e.g., replacing the search command **804** with the keyword that represents the highlighted segment **824**). A user selection of one of the search options **852** in the additional search menu **850** can be received, and the search command **804** in the search bar **802** is updated based on the search option that is selected for the highlighted segment. In this example, the additional search menu **850** also includes selectable interface links **858** that are each associated with a corresponding search option **852** in the additional search menu. A selectable interface link **858** for an associated search option can be selected by a user to initiate a new search interface.

[0101] FIG. 8F further illustrates the example of the search interface **800** described with reference to FIGS. 8A and 8C in accordance with the disclosed embodiments for event segment search drill down. In this example display of the search interface **800**, a user has selected the option “Exclude from search” **842** from the contextual search menu **836** (shown in FIG. 8C) to exclude the keyword that represents the highlighted segment **824** from a data search, and update the search command **804** in the search bar **802** to indicate that the keyword that represents the highlighted segment is excluded, which is shown as the keywords “NOT “Mozilla/5.0”” added to the search command **804**. The search system can then perform the data search based on the updated search command **804** to determine the multiple events **806** that do not include the highlighted segment, and display an updated search result set of the events **806** that do not include the highlighted segment in the search interface **800**. For example, each of the displayed events **806** in the search interface **800** do not include the highlighted “Mozilla/5.0” segment, but still do include the rest of the search command **804** (e.g., “sourcetype=access_combined”) as a field-value pair **816**.

[0102] FIG. 8G further illustrates the example of the search interface **800** described with reference to FIGS. 8A and 8C in accordance with the disclosed embodiments for event segment search drill down. In this example display of the search interface **800**, a user has selected the option “New search” **844** from the search contextual menu **836** (shown in FIG. 8C) to create a new data search based on the highlighted segment **824**, and update the search command **804** in the search bar **802** to include only the keyword that represents the highlighted segment, which is shown as “Mozilla/5.0” as the keyword in the search command **804**. The search system can then perform the new data search based on the updated search command **804** to determine the multiple

events **806** that each include the keyword that represents the highlighted segment, and display an updated search result set of the events **806** that each include the highlighted segment in the search interface **800**. For example, each of the displayed events **806** in the search interface **800** include the highlighted “Mozilla/5.0” segment, as shown generally at **848**.

[0103] FIG. 9A further illustrates the example of the search interface **800** described with reference to FIG. 8A in accordance with the disclosed embodiments for event segment search drill down. In this example display of the search interface **800**, a user has initiated the segment **824** being highlighted, such as with a mouse pointer moved over the “Mozilla/5.0” segment, which is displayed as the highlighted segment **824** in the event raw data **812**. Additionally, the user has selected the highlighted segment **824**, such as with a mouse click or touch input, and a contextual search menu **900** is displayed responsive to the user input. In implementations, the contextual search menu **900** is displayed proximate the highlighted segment **824** in the search interface **800**, such as a pop-up or drop-down menu just below the highlighted segment.

[0104] The contextual search menu **900** includes search options **902** that are selectable to operate on the highlighted segment **824** in the event raw data **812** of the displayed event **810**. For example, the search options **902** displayed in the contextual search menu **900** include: an option “Add to search” **904** that a user can select to add the highlighted segment **824** as a new keyword to the search command **804** in the search bar **802**; an option “Exclude from search” **906** that a user can select to exclude the keyword that represents the highlighted segment **824** from searches; an option “New search” **908** that a user can select to create a new data search based on the highlighted segment **824** (e.g., replacing the search command **804** with the keyword that represents the highlighted segment **824**); and an option “Field Extraction” **910** that a user can select to initiate an extract fields interface **912** shown in FIG. 9B that is usable to define a custom event field for an event **806**.

[0105] A custom event field is a field that has been extracted from an event, such as by using a regex rule or other techniques. The field extraction process creates a regex that is used to extract fields from an event and the fields that are extracted are custom event fields. An extract fields menu **916** includes an entry “Field Name” **918** that a user can enter to name a field that is to be extracted. The extract fields menu **916** also includes selectable options to “Extract” **920**, “Require” **922**, and “Add Extraction” **924**. The user can select the option “Extract” **920** to create a field extraction (or regex) for the text selected. For example, if an event includes the text “foobar-baz” and the user clicks to select the text “baz”, then selects the option “Extract”, enters the “Field Name” as “foobarField”, and clicks on the option “Add Extraction”, then the user has created a field extraction that has a regex that determines whether an event has the text “foobar=[any value]”. If an event includes this text, then a field will be created for this event with the field name “foobarField” and the value [any value]. If the user selects the option “Require” **922**, then the selected text is not extracted, but rather, the events that include the selected text are identified for the user while setting up the field extraction. The option to “Add Extraction” **924** then saves the regex rule for the field extraction.

[0106] A user selection of one of the search options **902** in the contextual search menu **900** can be received, and the search command **804** in the search bar **802** is updated based on the search option that is selected for the highlighted segment. In this example, the contextual search menu **900** also includes selectable interface links **926** that are each associated with a corresponding search option **902** in the contextual search menu. A selectable interface link **926** for an associated search option can be selected by a user to initiate a new search interface.

Field Value Search Drill Down

[0107] FIG. 10A further illustrates an example of the search interface **800** described with reference to FIG. 8A in accordance with the disclosed embodiments for field value search drill down. In this example display of the search interface **800**, a user has initiated a field-value pair **1000** of a field-value pair **814** being emphasized (e.g., highlighted), such as with a mouse pointer moved over the “access_combined.log” value, which is displayed as the emphasized field-value pair **1000** in the field-value pairs of the event **810**. Additionally, the user has selected the emphasized field-value pair **1000**, such as with a mouse click or touch input, and a field value contextual menu **1002** is displayed responsive to the user input. In implementations, the field value contextual menu **1002** is displayed proximate the emphasized field-value pair **1000** in the search interface **800**, such as a pop-up or drop-down menu just below the emphasized field-value pair. The described implementations of field value search drill down can also be applied to tagged field-value pairs, where a tag **1003** identifies a specific field-value pair **814** and any of the events **806** that have the tagged field-value pair can be displayed with the associated tag. A tag **1003** is an alias that designates a field-value pair, and can be selected to add search criteria that represents the tag to the search command **804** to search for events that have the field-value pair designated by the tag.

[0108] The field value contextual menu **1002** includes search options **1004** that are selectable to operate on the emphasized field-value pair **1000** in the displayed event **810**. For example, the search options **1004** displayed in the field value contextual menu **1002** include: an option “Add to search” **1006** that a user can select to add search criteria of the emphasized field-value pair **1000** to the search command **804** in the search bar **802**; an option “Exclude from search” **1008** that the user can select to add search criteria of the emphasized field-value pair **1000** to the search command in the search bar as the search criteria excluded from events that do not include the emphasized field-value pair; and an option “New search” **1010** that the user can select to create a new data search based on the emphasized field-value pair **1000** (e.g., replacing the search command **804** with the search criteria of the emphasized field-value pair **1000**). A user selection of one of the search options **1004** in the field value contextual menu **1002** can be received, and the search command **804** in the search bar **802** is updated based on the search option that is selected for the emphasized field-value pair.

[0109] In this example, the field value contextual menu **1002** includes a statistical event count **1012** that is associated with the option “Add to search” **1006**, and that indicates a number of multiple events that include the search criteria of the emphasized field-value pair **1000** of the field-value pairs **814**. The field value contextual menu **1002** also

includes a statistical event count **1014** that is associated with the option “Exclude from search” **1008**, and that indicates a number of multiple events that exclude the search criteria of the emphasized field-value pair **1000**. The field value contextual menu **1002** also includes selectable interface links **1016** that are each associated with a corresponding search option **1004** in the field value contextual menu. A selectable interface link **1016** for an associated search option can be selected by a user to initiate a new search interface.

[0110] A user may select the option “Add to search” **1006** from the field value contextual menu **1002** to add search criteria of the emphasized field-value pair **1000** to a data search, and the search command **804** in the search bar **802** is updated to include the search criteria of the emphasized field-value pair (similar to the example of the highlighted segment being added as a keyword to the search command as shown in FIG. 8D). The search system can then perform the data search based on the updated search command **804** to determine the multiple events **806** that each include the search criteria of the emphasized field-value pair, and display an updated search result set of the events **806** that each include the emphasized field-value pair in the search interface **800**.

[0111] Similar to the example of the additional search interface **850** shown and described with reference to FIG. 8E, multiple events may include the search criteria of the emphasized field-value pair that has been added to the search. A selection of the emphasized field-value pair in a displayed event initiates the additional search menu **850**, which includes the option “Remove from search” **854** that a user can select to remove the search criteria of the emphasized field-value pair from the search command **804** in the search bar **802**; and includes the option “New search” **856** that the user can select to create a new data search based on the search criteria of the emphasized field-value pair (e.g., replacing the search command **804** with the search criteria of the emphasized field-value pair).

[0112] Alternatively, a user may select the option “Exclude from search” **1008** from the field value contextual menu **1002** to exclude the search criteria of the emphasized field-value pair **1000** from a data search, and update the search command **804** in the search bar **802** to indicate that the emphasized field-value pair is excluded (similar to the example of the highlighted segment shown following the “NOT” operator in the search command as shown in FIG. 8F). The search system can then perform the data search based on the updated search command **804** to determine the multiple events **806** that do not include the search criteria of the emphasized field-value pair, and display an updated search result set of the events **806** that do not include the emphasized field-value pair in the search interface **800**.

[0113] Alternatively, a user may select the option “New search” **1010** from the field value contextual menu **1002** to create a new data search based on the emphasized field-value pair **1002**, and update the search command **804** in the search bar **802** to replace the search command in the search bar with search criteria of the emphasized field-value pair (similar to the example of the keyword that represents the highlighted segment added as the only search command **804** in the search bar **802** as shown in FIG. 8G). The search system can then perform the new data search based on the updated search command **804** to determine the multiple events **806** that each include the search criteria of the emphasized field-value pair, and display an updated search

result set of the events **806** that each include the emphasized field-value pair in the search interface **800**.

[0114] FIG. 10B further illustrates an example of the event field-picker interface **828** described with reference to FIG. 8B in accordance with the disclosed embodiments for field value search drill down. In this example display of the field-picker interface **828**, a user has initiated a field-value pair **1018** of one of the field-value pairs (**832**, **834**) being emphasized (e.g., highlighted), such as with a mouse pointer moved over the “splunkid_access” value, which is displayed as the emphasized field-value pair **1018** in the listing **830** of the field **832** and value **834** pairs in the event **810**. Additionally, the user has selected the emphasized field-value pair **1018**, such as with a mouse click or touch input, and a field value contextual menu **1020** is displayed responsive to the user input. In implementations, the field value contextual menu **1020** is displayed proximate the emphasized field-value pair **1018** in the event field-picker interface **828**, such as a pop-up or drop-down menu just below the emphasized field-value pair. The described implementations of field value search drill down can also be applied to tagged field-value pairs, where a tag identifies a specific field-value pair (**832**, **834**) and any of the events **806** that have the tagged field-value pair can be displayed with the associated tag.

[0115] The field value contextual menu **1020** includes search options **1022** that are selectable to operate on the emphasized field-value pair **1018** in the displayed event **810**. For example, the search options **1022** displayed in the field value contextual menu **1020** include: an option “Add to search” **1024** that a user can select to add search criteria of the emphasized field-value pair **1018** to the search command **804** in the search bar **802**; an option “Exclude from search” **1026** that a user can select to add search criteria of the emphasized field-value pair **1018** to the search command in the search bar as the search criteria excluded from events that do not include the emphasized field-value pair; and an option “New search” **1028** that a user can select to create a new data search based on the emphasized field-value pair **1018** (e.g., replacing the search command **804** with the search criteria of the emphasized field-value pair **1018**). A user selection of one of the search options **1022** in the field value contextual menu **1020** can be received, and the search command **804** in the search bar **802** is updated based on the search option that is selected for the emphasized field-value pair.

[0116] In this example, the field value contextual menu **1020** includes a statistical event count **1030** that is associated with the option “Add to search” **1024**, and that indicates a number of multiple events that include the search criteria of the emphasized field-value pair **1018**. The field value contextual menu **1020** also includes a statistical event count **1032** that is associated with the option “Exclude from search” **1026**, and that indicates a number of multiple events that exclude the search criteria of the emphasized field-value pair **1018**. The field value contextual menu **1020** also includes selectable interface links **1034** that are each associated with a corresponding search option **1022** in the field value contextual menu. A selectable interface link **1034** for an associated search option can be selected by a user to initiate a new search interface.

[0117] A user may select the option “Add to search” **1024** from the field value contextual menu **1020** to add search criteria of the emphasized field-value pair **1018** to a data

search, and the search command **804** in the search bar **802** is updated to include the search criteria of the emphasized field-value pair (similar to the example of the highlighted segment being added as a keyword to the search command as shown in FIG. 8D). The search system can then perform the data search based on the updated search command **804** to determine additional events that each include the search criteria of the emphasized field-value pair, and display an updated search result set of the events that each include the emphasized field-value pair, such as in the search interface **800** that lists the events **806**.

[0118] Alternatively, a user may select the option “Exclude from search” **1026** from the field value contextual menu **1020** to exclude the search criteria of the emphasized field-value pair **1018** from a data search, and update the search command **804** in the search bar **802** to indicate that the emphasized field-value pair is excluded (similar to the example of the highlighted segment shown following the “NOT” operator in the search command as shown in FIG. 8F). The search system can then perform the data search based on the updated search command **804** to determine additional events that do not include the search criteria of the emphasized field-value pair, and display an updated search result set of the events that do not include the emphasized field-value pair, such as in the search interface **800** that lists the events **806**.

[0119] Alternatively, a user may select the option “New search” **1028** from the field value contextual menu **1020** to create a new data search based on the emphasized field-value pair **1018**, and update the search command **804** in the search bar **802** to replace the search command in the search bar with search criteria of the emphasized field-value pair (similar to the example of the keyword that represents the highlighted segment added as the only search command **804** in the search bar **802** as shown in FIG. 8G). The search system can then perform the new data search based on the updated search command **804** to determine additional events that each include the search criteria of the emphasized field-value pair, and display an updated search result set of the events that each include the emphasized field-value pair, such as in the search interface **800** that lists the events **806**.

Example Methods

[0120] Example methods **1100** are described with reference to FIGS. 11A-11D in accordance with one or more embodiments of event segment search drill down, and example methods **1200** are described with reference to FIGS. 12A-12D in accordance with one or more embodiments of field value search drill down. Generally, any of the components, modules, methods, and operations described herein can be implemented using software, firmware, hardware (e.g., fixed logic circuitry), manual processing, or any combination thereof. Some operations of the example methods may be described in the general context of executable instructions stored on computer-readable storage memory that is local and/or remote to a computer processing system, and implementations can include software applications, programs, functions, and the like. Alternatively or in addition, any of the functionality described herein can be performed, at least in part, by one or more hardware logic components, such as, and without limitation, Field-programmable Gate Arrays (FPGAs), Application-specific Integrated Circuits (ASICs), Application-specific Standard Products (ASSPs),

System-on-a-chip systems (SoCs), Complex Programmable Logic Devices (CPLDs), and the like.

[0121] Computing devices (to include server devices) can be implemented with various components, such as a processing system and memory, and with any number and combination of different components as further described with reference to the example device shown in FIG. 13. One or more computing devices can implement the search system, in hardware and at least partially in software, such as executable software instructions (e.g., computer-executable instructions) that are executable with a processing system (e.g., one or more computer processors) implemented by the one or more computing devices. The search system can be stored on computer-readable, non-volatile storage memory, such as any suitable memory device or electronic data storage implemented by the computing devices.

[0122] FIGS. 11A-11D illustrate example method(s) 1100 of event segment search drill down, which may be implemented by a computing device, a distributed system of computing devices, and/or by one or more user client devices. The order in which a method is described is not intended to be construed as a limitation, and any number or combination of the method operations and/or methods can be performed in any order to implement a method, or an alternate method.

[0123] At 1102, a segment is emphasized in event raw data of an event that is one of multiple events returned as a search result set displayed in a search interface. For example, the segment 824 in the event raw data 812 of the event 810 is emphasized (e.g., highlighted) in the search interface 800 (FIG. 8A). The segment is emphasized when a pointer that is displayed in the search interface 800 moves over the segment, such as to highlight the segment with rollover (e.g., detected when a pointer moves over a segment). For example, a user may move a mouse pointer over the segment, which is then displayed as the highlighted segment 824. Alternatively, a user can emphasize a segment in the event raw data 812 by initiating a selection of the segment, such as with a computer mouse, stylus, or other input device.

[0124] At 1104, an input associated with the emphasized segment in the event raw data is received and, at 1106, a contextual search menu is displayed with search options that are selectable to operate on the emphasized segment in the event raw data. For example, the highlighted segment is selected in response to a user input, such as with a mouse click or touch input to select the highlighted segment. The contextual search menu 836 (FIG. 8C) is then displayed with the search options 838 responsive to the received user input, and the contextual search menu 836 is displayed proximate the highlighted segment 824 in the search interface 800. The search options 838 displayed in the contextual search menu 836 include the option “Add to search” 840 that a user can select to add the highlighted segment 824 as a new keyword to the search command 804 in the search bar 802; the option “Exclude from search” 842 that the user can select to exclude the keyword that represents the highlighted segment 824 from searches; and an option “New search” 844 that the user can select to create a new data search based on the highlighted segment 824 (e.g., replacing the search command 804 in the search bar 802 with the keyword that represents the highlighted segment 824). Additionally, the contextual search menu 836 of the search options 838 includes selectable interface links 846, each associated with

a corresponding search option 838, and a selectable interface link 846 is selectable to initiate a new search interface.

[0125] At 1108, a selection of one of the search options displayed in the contextual search menu is received and, at 1110, a search command is updated in a search bar of the search interface based on the search option that is selected from the contextual search menu for the highlighted segment. For example, a user selection of one of the search options 838 in the contextual search menu 836 is received, and the search command 804 in the search bar 802 is updated based on the search option that is selected for the highlighted segment 824. These features are further described with reference to FIGS. 11B-11D.

[0126] FIG. 11B illustrates an example method of event segment search drill down, and is generally described with reference to adding a highlighted segment as a new keyword to a data search.

[0127] At 1112, the selection of a search option is received to add the emphasized segment as a keyword to a data search and, at 1114, the search command in the search bar is updated to include the keyword that represents the emphasized segment. For example, a user selects the option “Add to search” 840 from the contextual search menu 836 (FIG. 8C) to add the highlighted segment 824 as a keyword to a data search, and update the search command 804 in the search bar 802 to include the keyword that represents the highlighted segment, which is shown added to the search command 804 (FIG. 8D).

[0128] At 1116, the data search is performed based on the updated search command to determine the multiple events that each include the keyword that represents the emphasized segment. Additionally, at 1118, an updated search result set of the multiple events that each include the emphasized segment is displayed in the search interface. For example, the search system performs the data search based on the updated search command 804 to determine the multiple events 806 that each include the keyword that represents the highlighted segment, and an updated search result set of the events 806 that each include the highlighted segment is displayed in the search interface 800 (FIG. 8D).

[0129] At 1120, an input is received that is associated with the emphasized segment in the event raw data of one of the multiple events displayed as part of the updated search result set. Additionally, at 1122, an additional search menu of options is displayed. For example, a user selects the highlighted segment 824 (or any of the similar highlighted segments 848), such as with a mouse click or touch input, and an additional search menu 850 is displayed proximate the highlighted segment 824 in the search interface 800 responsive to the user input (FIG. 8E). The additional search menu 850 includes search options 852 that are selectable to operate on the highlighted segment 824 in the event raw data 812 of the displayed event 810. For example, the search options 852 include: an option “Remove from search” 854 that a user can select to remove the highlighted segment 824 from a search; and an option “New search” 856 that the user can select to create a new data search based on the highlighted segment 824 (e.g., replacing the search command 804 in the search bar with the keyword that represents the highlighted segment 824).

[0130] FIG. 11C illustrates an example method of event segment search drill down, and is generally described with reference to excluding a highlighted segment from a data search.

[0131] At 1124, the selection of a search option is received to exclude the keyword that represents the emphasized segment from a data search and, at 1126, the search command in the search bar is updated to exclude the keyword that represents the emphasized segment. For example, a user selects the option “Exclude from search” 842 from the contextual search menu 836 (FIG. 8C) to exclude a keyword that represents the highlighted segment 824 from a data search, and the search command 804 in the search bar 802 is updated to indicate a keyword that represents the highlighted segment is excluded, such as with a “NOT” operator (FIG. 8F).

[0132] At 1128, the data search is performed based on the updated search command to determine the multiple events that do not include the keyword that represents the emphasized segment. Additionally, at 1130, an updated search result set of the multiple events that do not include the emphasized segment is displayed. For example, the search system performs the data search based on the updated search command 804 to determine the multiple events 806 that do not include the keyword that represents the highlighted segment, and an updated search result set of the events 806 that do not include the highlighted segment is displayed in the search interface 800 (FIG. 8F).

[0133] FIG. 11D illustrates an example method of event segment search drill down, and is generally described with reference to creating a new data search based on a highlighted segment.

[0134] At 1132, the selection of a search option is received to create a new data search based on the emphasized segment and, at 1134, the search command in the search bar is updated to include only the keyword that represents the emphasized segment. For example, a user selects the option “New search” 844 from the contextual search menu 836 (FIG. 8C) to create a new data search based on the highlighted segment 824, and the search command 804 in the search bar 802 is updated to include only the keyword that represents the highlighted segment (FIG. 8G).

[0135] At 1136, the new data search is performed based on the updated search command to determine the multiple events that include the keyword that represents the emphasized segment. Additionally, at 1138, an updated search result set of the multiple events that include the emphasized segment is displayed. For example, the search system performs the new data search based on the updated search command 804 to determine the multiple events 806 that each include the keyword that represents the highlighted segment, and an updated search result set of the events 806 that each include the highlighted segment is displayed in the search interface 800 (FIG. 8G).

[0136] FIGS. 12A-12D illustrate example method(s) 1200 of field value search drill down, which may be implemented by a computing device, a distributed system of computing devices, and/or by one or more user client devices. The order in which a method is described is not intended to be construed as a limitation, and any number or combination of the method operations and/or methods can be performed in any order to implement a method, or an alternate method.

[0137] At 1202, a field-value pair is emphasized in an event displayed in a search interface. For example, the field-value pair 1000 of the field-value pairs 814 in the event 810 is emphasized (e.g., highlighted) in the search interface 800 (FIG. 10A), such as with a mouse pointer moved over the emphasized field-value pair 1000. For example, a user

may move a mouse pointer over the field-value pair, which is then displayed as the emphasized field-value pair 1000. Alternatively, a user can emphasize a field-value pair in the search interface by initiating a selection of the field-value pair, such as with a computer mouse, stylus, or other input device. In implementations, the search interface is the event field-picker interface 828 (FIG. 10B) that displays a listing 830 of multiple field-value pairs of the event. The field-value pair 1018 is emphasized responsive to detection of an input pointer over the field-value pair.

[0138] At 1204, an input is received that is associated with the emphasized field-value pair and, at 1206, a field value contextual menu is displayed with search options that are selectable to operate on the emphasized field-value pair of the event. For example, the emphasized field-value pair is selected in response to a user input, such as with a mouse click or touch input to select the emphasized field-value pair. The field value contextual menu 1002 (FIG. 10A) is displayed with the search options 1004 proximate the emphasized field-value pair 1000 in the search interface 800 responsive to the received input. Similarly, the field value contextual menu 1020 (FIG. 10B) can be displayed with the search options 1022 proximate the emphasized field-value pair 1018 in the event field-picker interface 828. The field value contextual menu 1020 includes a first statistical event count 1012 that indicates a number of multiple events 806 that include the search criteria of the emphasized field-value pair 1000, and includes a second statistical event count 1014 that indicates a number of multiple events that exclude the search criteria of the emphasized field-value pair.

[0139] The search options displayed in the field value contextual menu include an option “Add to search” 1006 that a user can select to add search criteria of the emphasized field-value pair 1000 to the search command 804 in the search bar 802; an option “Exclude from search” 1008 that the user can select to add search criteria of the emphasized field-value pair 1000 to the search command in the search bar as the search criteria excluded from events that do not include the emphasized field-value pair; and an option “New search” 1010 that the user can select to create a new data search based on the emphasized field-value pair 1000 (e.g., replacing the search command 804 with the search criteria of the emphasized field-value pair 1000). The field value contextual menu 1002 of the search options 1004 also includes selectable interface links 1016, each associated with a corresponding search option 1004, and a selectable interface link 1016 is selectable to initiate a new search interface.

[0140] At 1208, a selection of one of the search options displayed in the field value contextual menu is received and, at 1210, a search command in a search bar of the search interface is updated based on the search option that is selected from the field value contextual menu for the emphasized field-value pair. For example, a user selection of one of the search options 1004 in the field value contextual menu 1002 can be received, and the search command 804 in the search bar 802 is updated based on the search option that is selected for the emphasized field-value pair. These features are further described with reference to FIGS. 12B-12D.

[0141] FIG. 12B illustrates an example method of field value search drill down, and is generally described with reference to adding search criteria of an emphasized field-value pair to a data search. A field-value pair search returns events that have the emphasized field-value pair, and the

value of the field for an event matches the selected value. A value for a field is an extraction from a specific location in an event (e.g., the location defined by an extraction rule). If a value of an emphasized field-value pair appears in an event, but is not the value of the field for that event because it is in a location not extracted by the extraction rule defining the field, that event does not meet the search criteria of the emphasized field-value pair.

[0142] At 1212, the selection of a search option is received to add search criteria of the emphasized field-value pair to a data search, and at 1214, the search command in the search bar is updated to include the search criteria of the emphasized field-value pair. For example, a user selects the option “Add to search” 1006 from the field value contextual menu 1002 to add the search criteria of the emphasized field-value pair 1000 to a data search, and the search command 804 in the search bar 802 is updated to include the search criteria of the emphasized field-value pair (similar to the example of the highlighted segment being added as a keyword to the search command as shown in FIG. 8D).

[0143] At 1216, the data search is performed based on the updated search command to determine additional events that each include the search criteria of the emphasized field-value pair. Additionally, at 1218, the additional events that each include the search criteria of the emphasized field-value pair are displayed in the search interface. For example, the search system performs the data search based on the updated search command 804 to determine the multiple events 806 that each include the search criteria of the emphasized field-value pair 1000, and an updated search result set of the events 806 that each include the search criteria of the emphasized field-value pair is displayed in the search interface 800.

[0144] FIG. 12C illustrates an example method of field value search drill down, and is generally described with reference to adding search criteria of an emphasized field value pair that matches events excluding the emphasized field-value pair.

[0145] At 1220, the selection of a search option is received to exclude the search criteria of the emphasized field-value pair from a data search and, at 1222, the search command in the search bar is updated to indicate that the search criteria of the emphasized field-value pair is excluded. For example, a user selects the option “Exclude from search” 1008 from the field value contextual menu 1002 to exclude the search criteria of the emphasized field-value pair 1000 from a data search, and update the search command 804 in the search bar 802 to indicate that the search criteria of the emphasized field-value pair is excluded (similar to the example of the keyword that represents the highlighted segment shown following the “NOT” operator in the search command as shown in FIG. 8F).

[0146] At 1224, the data search is performed based on the updated search command to determine additional events that do not include the search criteria of the emphasized field-value pair. Additionally, at 1226, the additional events that do not include the search criteria of the emphasized field-value pair are displayed. For example, the search system performs the data search based on the updated search command 804 to determine the multiple events 806 that do not include the search criteria of the emphasized field-value pair 1000, and an updated search result set of the events 806 that do not include the search criteria of the emphasized field-value pair is displayed in the search interface 800.

[0147] FIG. 12D illustrates an example method of field value search drill down, and is generally described with reference to creating a new data search based on an emphasized field-value pair.

[0148] At 1228, the selection of a search option is received to create a new data search based on the emphasized field-value pair and, at 1230, the search command in the search bar is replaced with the search criteria of the emphasized field-value pair. For example, a user selects the option “New search” 1010 from the field value contextual menu 1002 to create a new data search based on the emphasized field-value pair 1002, and the search command 804 in the search bar 802 is replaced with the search criteria of the emphasized field-value pair (similar to the example of the keyword that represents the highlighted segment replacing the search command 804 in the search bar 802 as shown in FIG. 8G).

[0149] At 1232, the new data search is performed based on the updated search command to determine additional events that include the search criteria of the emphasized field-value pair. Additionally, at 1234, the additional events that include the search criteria of the emphasized field-value pair are displayed. For example, the search system performs the new data search based on the updated search command 804 to determine the multiple events 806 that each include the search criteria of the emphasized field-value pair 1000, and an updated search result set of the events 806 that each include the search criteria of the emphasized field-value pair is displayed in the search interface 800.

Example System and Device

[0150] FIG. 13 illustrates an example system generally at 1300 that includes an example computing device 1302 that is representative of one or more computing systems and/or devices that may implement the various techniques described herein. This is illustrated through inclusion of the search interface module 1304 that is representative of functionality to interact with a search service 1306, e.g., to specify and manage searches using a late-binding schema and events as described above and thus may correspond to the client application module 106 and system 100 of FIG. 1. The computing device 1302 may be, for example, a server of a service provider, a device associated with a client (e.g., a client device), an on-chip system, and/or any other suitable computing device or computing system.

[0151] The example computing device 1302 as illustrated includes a processing system 1308, one or more computer-readable media 1310, and one or more I/O interface 1312 that are communicatively coupled, one to another. Although not shown, the computing device 1302 may further include a system bus or other data and command transfer system that couples the various components, one to another. A system bus can include any one or combination of different bus structures, such as a memory bus or memory controller, a peripheral bus, a universal serial bus, and/or a processor or local bus that utilizes any of a variety of bus architectures. A variety of other examples are also contemplated, such as control and data lines.

[0152] The processing system 1308 is representative of functionality to perform one or more operations using hardware. Accordingly, the processing system 1308 is illustrated as including hardware element 1314 that may be configured as processors, functional blocks, and so forth. This may include implementation in hardware as an application spe-

cific integrated circuit or other logic device formed using one or more semiconductors. The hardware elements **1314** are not limited by the materials from which they are formed or the processing mechanisms employed therein. For example, processors may be comprised of semiconductor(s) and/or transistors (e.g., electronic integrated circuits (ICs)). In such a context, processor-executable instructions may be electronically-executable instructions.

[0153] The computer-readable storage media **1310** is illustrated as including memory/storage **1316**. The memory/storage **1316** represents memory/storage capacity associated with one or more computer-readable media. The memory/storage component **1316** may include volatile media (such as random access memory (RAM)) and/or nonvolatile media (such as read only memory (ROM), Flash memory, optical disks, magnetic disks, and so forth). The memory/storage component **1316** may include fixed media (e.g., RAM, ROM, a fixed hard drive, and so on) as well as removable media (e.g., Flash memory, a removable hard drive, an optical disc, and so forth). The computer-readable media **1310** may be configured in a variety of other ways as further described below.

[0154] Input/output interface(s) **1312** are representative of functionality to allow a user to enter commands and information to computing device **1302**, and also allow information to be presented to the user and/or other components or devices using various input/output devices. Examples of input devices include a keyboard, a cursor control device (e.g., a mouse), a microphone, a scanner, touch functionality (e.g., capacitive or other sensors that are configured to detect physical touch), a camera (e.g., which may employ visible or non-visible wavelengths such as infrared frequencies to recognize movement as gestures that do not involve touch), and so forth. Examples of output devices include a display device (e.g., a monitor or projector), speakers, a printer, a network card, tactile-response device, and so forth. Thus, the computing device **1302** may be configured in a variety of ways as further described below to support user interaction.

[0155] Various techniques may be described herein in the general context of software, hardware elements, or program modules. Generally, such modules include routines, programs, objects, elements, components, data structures, and so forth that perform particular tasks or implement particular abstract data types. The terms “module,” “functionality,” and “component” as used herein generally represent software, firmware, hardware, or a combination thereof. The features of the techniques described herein are platform-independent, meaning that the techniques may be implemented on a variety of commercial computing platforms having a variety of processors.

[0156] An implementation of the described modules and techniques may be stored on or transmitted across some form of computer-readable media. The computer-readable media may include a variety of media that may be accessed by the computing device **1302**. By way of example, and not limitation, computer-readable media may include “computer-readable storage media” and “computer-readable signal media.”

[0157] “Computer-readable storage media” may refer to media and/or devices that enable persistent and/or non-transitory storage of information in contrast to mere signal transmission, carrier waves, or signals per se. Thus, computer-readable storage media refers to non-signal bearing media. The computer-readable storage media includes hard-

ware such as volatile and non-volatile, removable and non-removable media and/or storage devices implemented in a method or technology suitable for storage of information such as computer readable instructions, data structures, program modules, logic elements/circuits, or other data. Examples of computer-readable storage media may include, but are not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, hard disks, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other storage device, tangible media, or article of manufacture suitable to store the desired information and which may be accessed by a computer.

[0158] “Computer-readable signal media” may refer to a signal-bearing medium that is configured to transmit instructions to the hardware of the computing device **1302**, such as via a network. Signal media typically may embody computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier waves, data signals, or other transport mechanism. Signal media also include any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media.

[0159] As previously described, hardware elements **1314** and computer-readable media **1310** are representative of modules, programmable device logic and/or fixed device logic implemented in a hardware form that may be employed in some embodiments to implement at least some aspects of the techniques described herein, such as to perform one or more instructions. Hardware may include components of an integrated circuit or on-chip system, an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a complex programmable logic device (CPLD), and other implementations in silicon or other hardware. In this context, hardware may operate as a processing device that performs program tasks defined by instructions and/or logic embodied by the hardware as well as a hardware utilized to store instructions for execution, e.g., the computer-readable storage media described previously.

[0160] Combinations of the foregoing may also be employed to implement various techniques described herein. Accordingly, software, hardware, or executable modules may be implemented as one or more instructions and/or logic embodied on some form of computer-readable storage media and/or by one or more hardware elements **1314**. The computing device **1302** may be configured to implement particular instructions and/or functions corresponding to the software and/or hardware modules. Accordingly, implementation of a module that is executable by the computing device **1302** as software may be achieved at least partially in hardware, e.g., through use of computer-readable storage media and/or hardware elements **1314** of the processing system **1308**. The instructions and/or functions may be executable/operable by one or more articles of manufacture (for example, one or more computing devices **1302** and/or processing systems **1308**) to implement techniques, modules, and examples described herein.

[0161] The techniques described herein may be supported by various configurations of the computing device 1302 and are not limited to the specific examples of the techniques described herein. This functionality may also be implemented all or in part through use of a distributed system, such as over a “cloud” 1318 via a platform 1320 as described below.

[0162] The cloud 1318 includes and/or is representative of a platform 1320 for resources 1322. The platform 1320 abstracts underlying functionality of hardware (e.g., servers) and software resources of the cloud 1318. The resources 1322 may include applications and/or data that can be utilized while computer processing is executed on servers that are remote from the computing device 1302. Resources 1322 can also include services provided over the Internet and/or through a subscriber network, such as a cellular or Wi-Fi network.

[0163] The platform 1320 may abstract resources and functions to connect the computing device 1302 with other computing devices. The platform 1320 may also serve to abstract scaling of resources to provide a corresponding level of scale to encountered demand for the resources 1322 that are implemented via the platform 1320. Accordingly, in an interconnected device embodiment, implementation of functionality described herein may be distributed throughout the system 1300. For example, the functionality may be implemented in part on the computing device 1302 as well as via the platform 1320 that abstracts the functionality of the cloud 1318.

[0164] Although embodiments of field value search drill down have been described in language specific to features and/or methods, the appended claims are not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed as example implementations of field value search drill down, and other equivalent features and methods are intended to be within the scope of the appended claims. Further, various different embodiments are described and it is to be appreciated that each described embodiment can be implemented independently or in connection with one or more other described embodiments.

1. A computer-implemented method, comprising:
 - receiving, at a computing device, a first input indicating a selection of an event field-picker toggle associated with an event in a display of a set of events in a search interface;
 - causing, in response to the first input indicating the selection of the event field-picker toggle associated with the event, display of a listing of selectable field-value pairs corresponding to the event;
 - receiving a second input corresponding to a selection associated with a field-value pair of the listing of selectable field-value pairs; and
 - causing display of a second set of events corresponding to the selection associated with the field-value pair.
2. The computer-implemented method of claim 1, wherein the second set of events is retrieved by executing a search command that is generated based on the selection associated with the field-value pair.
3. The computer-implemented method of claim 1, wherein the set of events are a results set of performing an initial search command displayed in the search interface.
4. The computer-implemented method of claim 1, wherein the selection associated with the field-value pair

comprises selecting a selectable search option of a set of selectable search options displayed within a menu that is proximate to the field-value pair within the search interface.

5. The computer-implemented method of claim 1, wherein the selection associated with the field-value pair comprises selecting a selectable search option of a set of selectable search options, wherein the set of selectable search options include an add to search option, an exclude from search option, or a new search option.

6. The computer-implemented method of claim 1, wherein the selection associated with the field-value pair comprises selecting a selectable search option comprising an add to search option to automatically add text defining search criteria to a search command.

7. The computer-implemented method of claim 1, wherein the selection associated with the field-value pair comprises selecting a selectable search option comprising an exclude from search option to automatically add text defining search criteria to a search command.

8. The computer-implemented method of claim 1, wherein the selection associated with the field-value pair comprises selecting a selectable search option comprising a new search option to automatically create a new data search based on the field-value pair.

9. The computer-implemented method of claim 1, further comprising causing display of a set of selectable search options, wherein at least one selectable search option of the set of selectable search options corresponds with an indication of a number of events of the set of events associated with the field-value pair.

10. The computer-implemented method of claim 1, further comprising causing display of a set of selectable search options, wherein at least one selectable search option of the set of selectable search options corresponds with an indication of a number of events of the set of events that exclude the field-value pair.

11. The computer-implemented method of claim 1, further comprising causing display of a set of selectable search options, wherein the selectable search options are further displayed with selectable interface links each associated with a particular search option of the selectable search options, and wherein a selectable interface link, when activated, initiates a new search interface.

12. The computer-implemented method of claim 1, further comprising causing visual emphasis of the field-value pair.

13. The computer-implemented method of claim 1, wherein the set of events are returned as a search result, the set of events being identified from collected data that comprises at least one of raw data, machine data, performance data, log data, diagnostic information, transformed data, or mashup data combined from multiple sources.

14. The computer-implemented method of claim 1, wherein the set of events are returned as a search result performed using a late-binding schema on data collected from one or more sources, and wherein the event comprises a portion of raw data that is associated with a timestamp indicating a respective point in time associated with the event.

15. A computer-implemented system, comprising:
one or more processors; and
one or more computer-readable media storing instructions thereon that, when executed by the one or more processors, cause the one or more processors to perform operations comprising:
receiving, at a computing device, a first input indicating a selection of an event field-picker toggle associated with an event in a display of a set of events in a search interface;
causing, in response to the first input indicating the selection of the event field-picker toggle associated with the event, display of a listing of selectable field-value pairs corresponding to the event;
receiving a second input corresponding to a selection associated with a field-value pair of the listing of selectable field-value pairs; and
causing display of a second set of events corresponding to the selection associated with the field-value pair.

16. The computer-implemented system of claim **15**, wherein the second set of events is retrieved by executing a search command that is generated based on the selection associated with the field-value pair.

17. One or more non-transitory computer-storage media storing computer-useable instructions that, when executed by at least one computing device, cause the at least one computing device to perform a method, the method comprising:

receiving, at a computing device, a first input indicating a selection of an event field-picker toggle associated with an event in a display of a set of events in a search interface;
causing, in response to the first input indicating the selection of the event field-picker toggle associated with the event, display of a listing of selectable field-value pairs corresponding to the event;
receiving a second input corresponding to a selection associated with a field-value pair of the listing of selectable field-value pairs; and
causing display of a second set of events corresponding to the selection associated with the field-value pair.

18. The one or more non-transitory computer-storage media of claim **17**, wherein the selection associated with the field-value pair comprises selecting a selectable search option comprising an add to search option to automatically add text defining search criteria to a search command.

19. The one or more non-transitory computer-storage media of claim **17**, wherein the selection associated with the field-value pair comprises selecting a selectable search option comprising an exclude from search option to automatically add text defining search criteria to a search command.

20. The one or more non-transitory computer-storage media of claim **17**, wherein the selection associated with the field-value pair comprises selecting a selectable search option comprising a new search option to automatically create a new data search based on the field-value pair.

* * * * *