

INDEX CONFIGURATION FOR SEARCHABLE DATA IN NETWORK

BACKGROUND

[0001] Computing devices are often used to communicate over a network such as the Internet. Network based services offered by a service provider are becoming more commonplace. Computing devices are frequently used to connect to a network based service, which can provide services such as storing searchable data to be used/retrieved by the computing devices or providing additional processing power to the computing devices. With respect to the network based storage of searchable data, users of computing devices typically need to choose a configuration and/or format for their data, so that their data can be indexed and stored by the network based service. Conventional approaches typically require users to determine an appropriate configuration for their data. Conventional approaches can also demand a format to which the user's data must comply, thereby requiring the users to convert their data to the format. This can be inconvenient, cumbersome, or difficult to users who want to use the network based service for storage and search, thereby reducing the overall user experience.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Various embodiments in accordance with the present disclosure will be described with reference to the drawings, in which:

[0003] **FIG. 1** illustrates an example environment in which aspects of the various embodiments can be utilized;

[0004] **FIG. 2** illustrates an example system embodiment for index configuration for searchable data in a networked environment;

[0005] **FIG. 3** illustrates an example web browsing environment in which index configuration for searchable data in a networked environment can be utilized;

[0006] **FIG. 4** illustrates an example search index that can be generated in accordance with the various embodiments ;

[0007] **FIG. 5** illustrates an example method embodiment for index configuration for searchable data in a networked environment;

[0008] FIG. 6 illustrates an example device that can be used to implement aspects of the various embodiments;

[0009] FIG. 7 illustrates example components of a client device such as that illustrated in FIG. 6; and

[0010] FIG. 8 illustrates an environment in which various embodiments can be implemented.

DETAILED DESCRIPTION

[0011] Systems and methods to generate an index configuration that can be used to generate a search index for data received over at least one network are described. At least some embodiments enable a computing device to upload data over a network (e.g., the Internet) onto a storage allocation provided by a network service (i.e., network service provider). The network service can analyze the uploaded data to determine a type of data field (i.e., data field type) for each data field in the plurality of data fields. The network service can analyze the uploaded data to determine whether or not to enable one or more search options for each data field in the plurality of data fields included in the uploaded data.

[0012] At least some embodiments enable a computing device to upload data over a network (e.g., the Internet) onto a storage allocation provided by a network service (i.e., network service provider, network based service, etc.). One or more users/entities (e.g., using one or more computing devices) can search for the uploaded data over the network utilizing a search index, which can be provided by the network service.

[0013] In some embodiments, the uploaded data can include a plurality of data fields. The network service can analyze the uploaded data to determine a type of data field (i.e., data field type) for each data field in the plurality of data fields. For example, each data field can be of a type including an integer type, a text type, or a literal type.

[0014] Moreover, the network service can analyze the uploaded data to determine whether or not to enable one or more search options for each data field in the plurality of data fields included in the uploaded data. For example, the network service can determine, for each respective data field, whether or not to enable an option that would include the respective data field in a search index to be generated. The network service

can also determine, for each respective data field, whether to enable an option that would calculate a facet count for the respective data field. Further, the network service can determine, for each respective data field, whether to enable an option that would return/provide the value associated with the respective data field in response to a search query.

[0015] In some embodiments, the network service can generate an index configuration (i.e., search index configuration, schema, index settings, etc.) for the data based at least in part on the determined data field type(s) and the search option(s) to be enabled. The network service can generate a search index for the data based at least in part on the index configuration.

[0016] Various other functions and advantages are described and suggested below as may be provided in accordance with the various embodiments.

[0017] FIG. 1 illustrates an example environment 100 in which aspects of the various embodiments can be utilized. The example environment 100 can comprise at least one computing device 102, a network 104 (e.g., Internet, intranet, local network, local area network, etc.), and a network service 106 (i.e., network service provider, network based service, etc.). The at least one computing device 102 can be communicatively connected to the network service 106 over the network 104. In some embodiments, the computing device 102 can communicate the network service 106 without a network 104 such as the Internet. As shown in FIG. 1, there can also be a user 108 of the at least one computing device 102 or other entity (e.g., individual, company, organization, group, etc.) 108. The user or entity 108 can communicate data 110 from the at least one computing device 102 over the network 104 to the network service 106 (and vice versa).

[0018] In some embodiments, the network service 106 can comprise of and/or utilize one or more hosts or servers connected to the network 104. For example, the network service 106 can rent storage space to customers, such as the user of the device 102 or another entity(ies) (e.g., company, organization, group, individual, etc.) 108. Accordingly, the user/entity 108 of the computing device 102 can store data from the device 102 onto the network service 106 using the network 104. In other words, the user/entity 108 and/or device 102 can utilize network based computing storage via the network service 106.

[0019] In one example, the computing device 102 can transmit data 110 over the network 104 to be stored on the network service 106, as shown in FIG. 1. The data 110 can be any data utilized in network based computing, such as for search, database storage, running an application, running a virtual machine, running an operating system, etc. The computing device 102 can transmit the data 110 to be stored on a storage allocation provided by the service 106. For example, the user/entity 108 can purchase or rent storage space on the service 106 and the storage allocation can be allocated and assigned to the user/entity 108. In some embodiments, the user/entity 108 can have a particular account and/or storage allocation on the service 106; the storage space (e.g., storage allocation) allocated and assigned to the entity 108 can be associated with the account for the entity 108.

[0020] The entity 108 may also want the network service 106 to provide a search index for the data 110. Conventional approaches typically require the entity 108 to first provide a configuration (i.e., index configuration, schema, index setting, etc.) for the data 110 to be indexed, or conventional approaches can require a configuration/format (e.g., Search Data Format (SDF)) that the entity's data 110 must comply with, thus demanding the entity 110 to convert its data 110 to the required configuration. However, this can be inconvenient, cumbersome, or difficult to the entity 108.

[0021] In some embodiments, the entity 108 can transmit the data 110 to the network service 106, and the network service 106 can automatically (i.e., without an instruction or a request from the entity 108) analyze the data 110 and generate an index configuration (e.g., search index configuration, search index schema, etc.) for the data 110. For example, in some embodiments, the network service 106 can analyze the data 110 by determining a type of data field 112 for one or more data fields included in the data 110 and determining a search option 114 to be enabled for one or more data fields included in the data 110.

[0022] With regard to determining the type of data field 112, there can be a plurality of data field types that the data 110 (e.g., document, file, etc.) can be associated with, such as an integer type of data field, a literal type of data field, or a text type of data field, and so forth. In some embodiments, the data 110 can include a plurality of data fields, each data field including a value (e.g., data field "name" can have a value of "ABCD-Brand Shirt"; data field "Price" can have a value of "\$20"; etc.). The network

service 106 can analyze the plurality of data fields included in the data 110 to determine a data field type for each data field in the plurality.

[0023] For example, for each data field, the network service 106 can determine whether the value of each respective data field comprises an amount of integers above a specified integer amount threshold (e.g., value of data field “Price” is all integers); if so, then that respective data field can be determined to be of an integer data field type. The network service 106 can also determine whether a data field is of a literal data field type by, for example, determining at least one of a value associated with the data field having an amount of alphabetic characters above a specified lower literal amount threshold but below a specified upper literal amount threshold, a number of distinct values associated with the data field being below a specified literal distinct amount threshold, a percentage of distinct values being below a specified literal distinct percentage threshold, or a length of values being below a specified literal length threshold. In some embodiments, the network service 106 can, for example, consider the length of a data field value and the frequency and/or percentage of distinct values in the data field value to identify the data field as being of a text type; if there are many distinct values in a data field value and the data field value is very long (e.g., has a number of alphanumeric characters above a threshold), then the data field is likely of a text type. In some embodiments, if a data field is not of an integer type or a literal type, then the data field can be of a text type.

[0024] Regarding determining the search option 114, the network service 106 can determine one or more search options 114 to be enabled for (the data fields of) the data 110. For example, having determined a data field type for a data field included in the data 110, the network service 106 can determine whether or not to enable an option to include the data field in the search index to be generated, whether or not to enable an option to calculate a facet count for the data field, and/or whether or not to enable an option to return/provide a search value for the data type.

[0025] For example, if the data field type for a data field is determined to be a text (e.g., the data field is a “Product Description” and the value is a long paragraph), then the network service 106 can choose the option not to include the data field (and value) in the search index. In another example, for data field with an integer data field type (e.g., data field being “Production Year” and the value is a year), the network service 106 can choose to enable the option to include the data field in the search index to be

generated, and the service 106 can enable the option to calculate a facet count for the data field. A facet count can be a count of how many search results fall into a certain category for a data field. For example, if the data field is “Production Year,” the network service 106 can determine that it makes sense to provide a facet count, which indicates how many search results are associated with a certain category; e.g., “1984 (23), 2002 (12), 2010 (18)” shows an example of facet counts in which 23 search results are associated with “1984” with respect to the “Production Year” data field, 12 search results are associated with “2002,” and 18 search results are associated with “2010.”

[0026] In some embodiments, the network service 106 can also decide to enable the return of the value for a data field. For example, not all searchable data fields (and values) need to be returned (e.g., retrieved and presented) in response to a search request. The network service 106 can decide whether or not to return the value for a data field.

[0027] Turning now to the generating of a configuration for the data 110, the network service can automatically (i.e., without an instruction from the entity 108) generate a configuration (e.g., search index configuration, schema, etc.) for the data 110. In some embodiments, the configuration can, at least in part, help determine how to index the data 110; the index configuration can, at least in part, govern how the data 110 will be indexed. The configuration or schema can specify a data field type for each data field included in the data 110, indicate whether each data field is searchable, indicate whether each data field is rankable (e.g., sortable), and other similar information useful for building the index. Subsequent to generating the configuration for the data 110 to be indexed, the network service 106 can generate a search index for the data 110 based in at least part on the generated configuration.

[0028] FIG. 2 illustrates an example web browsing environment 200 in which index configuration for searchable data in a networked environment can be utilized. The example web browsing environment 200 can comprise an example web page 202 being rendered by an application, such as a web browser. In this example, the web page 202 can be provided by a network service that is associated with the domain, ABCD.com.

[0029] A user/entity (e.g., customer of the network service) can be a retailer and can upload data that is related to selling shirts, for example. The data can be indexed and stored by the network service and made searchable to others such as potential customers

of the user/entity. The network service can analyze the data to determine a type of data field (i.e., data field type) for each of the data fields included in the data. For example, the data related to the selling of shirts can include data fields such as “Color” 206, “Size” 208, “Price” 210, “Description,” and other fields. The network service can analyze the value for each data field to determine a type for each respective data field. The network service can also determine one or more options (e.g., search options) to enable for each data field. The network service can subsequently generate a configuration/schema for the data to be indexed. Then the network service can generate an index for the data based on the configuration/schema.

[0030] For example, the network service can identify the data field “Color” and determine that its value (e.g., “Red,” “Blue,” “White,” “Green,” etc.) is alphabetic/literal and may identify the type of the “Color” data field to be a literal type. (In this example, the data associated with the “Color” data field and the values (e.g., “Red,” “Blue,” “White,” “Green,” etc.) can be uploaded by the entity.) In another example, the network service can identify a “Size” data field in at least a portion of the uploaded data and determine that the values contained in the “Size” data field are numeric values. In this instance, the network service may determine that the “Size” data field is an integer type. In a further example, the network service can identify the values for the “Description” data fields in at least a portion of the uploaded data and may determine that the values include both numbers and alphabetic characters, and/or that the values are lengthy in terms of the number of characters, and/or that the values have distinct terms/phrases/symbols. In this instance, the network service may determine that the “Description” data field is a text type.

[0031] Regarding the search options, the network service can determine, for each of the data fields, whether to not to enable the option to include a respective data field in the search index to be generated. For example, in some embodiments, the “Description” data fields (and corresponding values) can be omitted from the search index. If so, then when a query is run with respect the search index, the query will not search the “Description” data field. However, some embodiments can and do include the “Description” data fields and values in the search index.

[0032] Moreover, the network service can determine whether or not to enable the option to calculate a facet count for each data field. As mentioned above, a facet count represents how many of the results matching a search query have a particular value (or

range of values) for a particular data field. For example, as shown in FIG. 2, the “Color” data field with a value of “Red” has a facet count of 23 (i.e., 23 search results for a “Red” shirt), whereas the “Blue” value of the “Color” data field has a facet count of 28 (i.e., 28 search results for a “Blue” shirt), and so forth. In some embodiments, the values can overlap (i.e., do not have to be an exact match). For example, a shirt with blue and red stripes can be associated with both the “Blue” and “Red” values, and/or with other values. In some embodiments, the network service can determine that facet counts should be calculated for some of the data fields, but not necessarily all of the data fields. For example, the network service can determine that there should be facet counts for “Color,” “Size,” and “Price,” but not for “Description.”

[0033] Furthermore, the network service can determine whether or not to enable a return of the value for a data field. For example, there can be a data field “Internal Product Identification Number” included in the data, the value of the data field being a product identification number internal to the entity and not intended to be shown to a customer of the entity; as such, the network service can determine not to enable a return of the value for such a data field.

[0034] It is contemplated that there can be additional options as well as data related to other items that a person having ordinary skill in the art would recognize. For example, the network service can determine whether or not to enable an option to make a data field rankable (e.g., sortable). With reference to FIG. 2, in some embodiments, the “Price” data field can be ranked/sorted by its values (e.g., from lowest price to highest price, from highest price to lowest price, etc.), the “Color” data field can be sorted alphabetically (not illustrated in FIG. 2), and so forth. In another example (not illustrated), there can be data related to media files, such as music, videos, books, photographs, etc. Example data fields for the media files can include, but are not limited to, “Title,” “Artist/Author,” “Year Created,” “Price,” “Rating,” etc.

[0035] Having determined the types of the data fields included in the data and the one or more search options for the data fields included in the data, the network service can generate a configuration (i.e., search index configuration, schema, etc.) for the data, the generating of the configuration being based at least in part on the determined data field types and search options.

[0036] Subsequent to generating the configuration, the network service can generate a search index for the data based at least in part on the generated configuration. Accordingly, the data provided by the entity can be stored with the network service and the search index for the data generated by the network service.

[0037] FIG. 3 illustrates an example system embodiment 300 for index configuration for searchable data in a networked environment. The example system embodiment 300 can comprise a system controller 302, at least one communication transceiver 304, a data field type analyzer 306, a search option analyzer 308, an index configuration generator 310, an index generator 312, and at least one storage allocation 314.

[0038] The system controller 302 can facilitate the system to perform various operations for index configuration for searchable data in a networked environment. The system controller 302 can communicate with the at least one communications transceiver 304 to facilitate data transmission to and/or data receipt from one or more sources external to the system 300 as well as to facilitate data communication among the system.

[0039] Data received (e.g., from an entity) by the system 300 via the communications transceiver 304 can be analyzed by the data field type analyzer 306 to determine a type associated with each of the data fields included in the data. The data can also be analyzed by the search option analyzer 308 to determine whether or not to enable one or more search options with respect to each of the data fields included in the data. Based at least in part on the determined data field types and the one or more determined search options, the index configuration generator 310 can generate a search index configuration/schema. Then based at least in part on the generated search index configuration/schema, the index generator 312 can generate a search index for the data. The data and the search index generated for the data can be stored on one or more storage allocations 314.

[0040] It is contemplated that the various components and/or portions of the example system 300 can be implemented as hardware, software, or a combination of both. For example, the various parts of the system 300 can be implemented via a circuit, a processor, an application, a portion of programming code, an algorithm, or any combination thereof, etc. It is also contemplated FIG. 3 is an example and meant to be used for illustrative purposes only. For example, the various components do not have to

be configured according to FIG. 3. In some embodiments, the various components do not have to be tightly coupled to one another and can instead be spread across a more distributed system. For example, a component such as the index generator can reside on a separate/different network and/or system, but still retain a communicative connection(s) to the other components.

[0041] FIG. 4 illustrates an example search index 400 that can be generated in accordance with the various embodiments of the present disclosure. With reference to FIG. 4, there can be a root node 402 in the search index. In the example of FIG. 4, data can be uploaded by an entity such as a T-Shirt retailer. The data can correspond to information about the T-Shirts (root node 402) that the entity has made available for sale. There can be parent nodes (e.g., 404, 406, 408) that represent data fields for the data relating to the T-Shirts. For example, the T-Shirts can have a Color data field 404, a Size data field 406, and a Price data field 408.

[0042] Continuing with the example of FIG. 4, the data fields can have child nodes (e.g., 410, 412, 414, 416, 418) that represent values within each respective data field. For example, there can be at least two Colors (Red 410 and Blue 412), one Size (Medium 414), and two Price ranges (<\$10 416 and \$10-\$20 418). There can also be a set of search results/items (e.g., T-Shirts 420, 422, 424, 426, 428, 430) that can correspond to one or more of the data fields and values.

[0043] In this example, all three data fields (Color 404, Size 406, and Price 408) are to be included in the search index, can have facet counts, and can provide/return values in response to relevant search queries. For example, as shown in FIG. 4, Color:Red 410 can have a facet count of three and Color:Blue 412 can have a facet count of two. Size:Medium 414 can have a facet count of two. Price:<\$10 416 can have a facet count of one and Price:\$10-\$20 418 can have a facet count of two. Moreover, a search query of Color:Red 410, for example, will return T-Shirts 422, 424, and 428; searching for Red 410 and Blue 412, for example, will return T-Shirt 422; and so forth. Although the example search index 400 is shown as being a tree structure, it is contemplated that the search index can be generated in many other ways and/or with other structures.

[0044] FIG. 5 illustrates an example method embodiment 500 for index configuration for searchable data in a networked environment. Again, it should be understood that there can be additional, fewer, or alternative steps performed in similar or alternative

orders, or in parallel, within the scope of the various embodiments unless otherwise stated. At step 502, the example method embodiment 500 can receive data to be indexed. For example, the method 500 can receive data, uploaded by an entity, to be indexed and the data can include a plurality of data fields (or at least one data field). In some embodiments, the example method can also determine a name for a data field associated with the data. At step 504, the example method 500 can determine a type of a data field associated with the data. For example, the method can determine a field type, of a plurality of field types, associated with each data field in the plurality of data fields. The plurality of field types can include (but is not limited to) at least one of an integer type, a literal type, or a text type. The type of the data field can be determined from a plurality of types of data fields. In some embodiments, the plurality of data fields and their types and/or names can be identified based on tags, signals, or other indications. The method 500 can determine one or more search options to be enabled with respect to the data field associated with the data, at step 506. For example, the one or more search options can include at least one of an option to include a respective data field in a search index to be generated, an option to calculate a facet count for the respective data field, or an option to provide one or more values associated with the respective data field. Step 508 can include generating an index configuration for the data based at least in part on the type of the data field and the one or more search options. Then at step 510, the method 500 can generate a search index for the data based at least in part on the index configuration for the data. In some embodiments, the search index can be generated based on whether the data is structured data, free text data, or a combination of both. In some embodiments, the example method can also provide at least one of the data, the index configuration, or the index to be searchable by one or more search queries.

[0045] There can be various other information included in index configurations. For example, a configuration can hold information regarding whether a data field is facetable or not (i.e., whether or not a facet count should be calculated for the data field), whether a data field is rankable or not (i.e., whether or not the values of search results having the data field should be sorted), etc.

[0046] In some embodiments, the network service can convert data received/uploaded in a first format to a second format, the second format being compatible with the search index and can store the data converted to the second format on one or more storage

allocations. For example, the network service can receive data from the entity, the data capable of having any one or more of several various formats, such as .PDF, .DOC, .DOCX, .CSV, .JSON, .XML, etc. The network service can automatically convert the data into a format compatible with (e.g., recognizable by, workable with, etc.) the network service, such as the Search Data Format (SDF).

[0047] In some embodiments, the network service can convert the data based on comparing the first format with the second format and modifying at least one data field associated with the first format to correspond to at least one data field associated with the second format. For example, the network service can compare a format(s) of the data received from the entity and modify/update the format such that it is compatible with the network service. This can include identifying whether one or more data fields in the format should be added, removed, or changed.

[0048] In some embodiments, the network service can determine a type of a data field to be the integer type based on determining that a value associated with the data field has an amount of integer characters above a specified integer amount threshold. Also, the network service can determine a type of a data field to be the literal type determining at least one of a value associated with the data field having an amount of alphabetic characters above a specified lower literal amount threshold but below a specified literal amount threshold, a number of distinct values associated with the data field being below a specified literal distinct amount threshold, a percentage of distinct values being below a specified literal distinct percentage threshold, or a length of values being below a specified literal length threshold. Further, the network service can determine a type of a data field to be the text type based on determining that a value associated with the data field has at least one of an amount of integer and alphabetic characters above a specified text amount threshold, a number of distinct characters above a specified text distinct amount threshold, a percentage of distinct characters above a specified text distinct percentage threshold, or a length of characters above a specified text length threshold.

[0049] In some embodiments, the network service can decide to enable the option to include a data field in a search index to be generated, the decision being based at least in part on receiving a signal included in the data field indicating that the data field is to be included in the search index. The network service can also decide to enable the option to calculate a facet count for a data field, the decision being based at least in part on

determining that a quantity for at least one value associated with the data field is above a specified facet count lower threshold and below a specified facet count upper threshold. The network service can further decide to enable the option to provide a value associated with a data field in response to a relevant search query, the decision being based at least in part on receiving a signal included in the data field indicating that the value associated with the data field is to be provided.

[0050] In some embodiments, one or more search queries (e.g., terms in the search query) can be utilized by the network service. For example, the network service can infer from the search queries that a searcher is faceting on a particular data field. As a result, the network service can determine that the data field should be of a literal type, for example.

[0051] In some embodiments, when a searcher inputs query terms and requests a search, one or more search results can be presented in a particular rank expression (e.g., order of results), such as by relevance. The present disclosure can allow for creations more complicated rank expressions that take into account other factors such as query independent factors (e.g., there can be a popularity data field included within the data, etc.). The present disclosure can also allow for analysis to propose rank expressions that can be used, by looking at the data and determining that a data field is meaningful in terms of popularity. For example, there can be a text body data field type and its length (e.g., or the inverse of its length) can be taken into account and can provide useful information for rank expressions.

[0052] In some embodiments, data field types can also include a geolocation type, a time type, a date type, or a floating point number type.

[0053] Various embodiments consistent with the present disclosure can also utilize sample data. For example, the user/entity can first provide sample data to the network service. The network service can analyze the sample data to determine types of data fields and search options. Based on the data field types and search options for the sample data, the network service can generate an index configuration, and subsequently generate a search index based on the generated index configuration.

[0054] FIG. 6 illustrates an example electronic user device 600 that can be used in accordance with various embodiments. Although a portable computing device (e.g., an electronic book reader or tablet computer) is shown, it should be understood that any

electronic device capable of receiving, determining, and/or processing input can be used in accordance with various embodiments discussed herein, where the devices can include, for example, desktop computers, notebook computers, personal data assistants, smart phones, video gaming consoles, television set top boxes, and portable media players. In some embodiments, a computing device 600 can be an analog device, such as a device that can perform signal processing using operational amplifiers. In this example, the computing device 600 has a display screen 602 on the front side, which under normal operation will display information to a user facing the display screen (e.g., on the same side of the computing device as the display screen). The computing device in this example includes at least one camera 604 or other imaging element for capturing still or video image information over at least a field of view of the at least one camera. In some embodiments, the computing device might only contain one imaging element, and in other embodiments the computing device might contain several imaging elements. Each image capture element may be, for example, a camera, a charge-coupled device (CCD), a motion detection sensor, or an infrared sensor, among many other possibilities. If there are multiple image capture elements on the computing device, the image capture elements may be of different types. In some embodiments, at least one imaging element can include at least one wide-angle optical element, such as a fish eye lens, that enables the camera to capture images over a wide range of angles, such as 180 degrees or more. Further, each image capture element can comprise a digital still camera, configured to capture subsequent frames in rapid succession, or a video camera able to capture streaming video.

[0055] The example computing device 600 also includes at least one microphone 606 or other audio capture device capable of capturing audio data, such as words or commands spoken by a user of the device. In this example, a microphone 606 is placed on the same side of the device as the display screen 602, such that the microphone will typically be better able to capture words spoken by a user of the device. In at least some embodiments, a microphone can be a directional microphone that captures sound information from substantially directly in front of the microphone, and picks up only a limited amount of sound from other directions. It should be understood that a microphone might be located on any appropriate surface of any region, face, or edge of the device in different embodiments, and that multiple microphones can be used for audio recording and filtering purposes, etc.

[0056] The example computing device 600 also includes at least one orientation sensor 608, such as a position and/or movement-determining element. Such a sensor can include, for example, an accelerometer or gyroscope operable to detect an orientation and/or change in orientation of the computing device, as well as small movements of the device. An orientation sensor also can include an electronic or digital compass, which can indicate a direction (e.g., north or south) in which the device is determined to be pointing (e.g., with respect to a primary axis or other such aspect). An orientation sensor also can include or comprise a global positioning system (GPS) or similar positioning element operable to determine relative coordinates for a position of the computing device, as well as information about relatively large movements of the device. Various embodiments can include one or more such elements in any appropriate combination. As should be understood, the algorithms or mechanisms used for determining relative position, orientation, and/or movement can depend at least in part upon the selection of elements available to the device.

[0057] FIG. 7 illustrates a logical arrangement of a set of general components of an example computing device 700 such as the device 600 described with respect to FIG. 6. In this example, the device includes a processor 702 for executing instructions that can be stored in a memory device or element 704. As would be apparent to one of ordinary skill in the art, the device can include many types of memory, data storage, or non-transitory computer-readable storage media, such as a first data storage for program instructions for execution by the processor 702, a separate storage for images or data, a removable memory for sharing information with other devices, etc. The device typically will include some type of display element 706, such as a touch screen or liquid crystal display (LCD), although devices such as portable media players might convey information via other means, such as through audio speakers. As discussed, the device in many embodiments will include at least one image capture element 708 such as a camera or infrared sensor that is able to image projected images or other objects in the vicinity of the device. Methods for capturing images or video using a camera element with a computing device are well known in the art and will not be discussed herein in detail. It should be understood that image capture can be performed using a single image, multiple images, periodic imaging, continuous image capturing, image streaming, etc. Further, a device can include the ability to start and/or stop image capture, such as when receiving a command from a user, application, or other device.

The example device similarly includes at least one audio capture component 712, such as a mono or stereo microphone or microphone array, operable to capture audio information from at least one primary direction. A microphone can be a uni-or omni-directional microphone as known for such devices.

[0058] In some embodiments, the computing device 700 of FIG. 7 can include one or more communication elements (not shown), such as a Wi-Fi, Bluetooth, RF, wired, or wireless communication system. The device in many embodiments can communicate with a network, such as the Internet, and may be able to communicate with other such devices. In some embodiments the device can include at least one additional input device able to receive conventional input from a user. This conventional input can include, for example, a push button, touch pad, touch screen, wheel, joystick, keyboard, mouse, keypad, or any other such device or element whereby a user can input a command to the device. In some embodiments, however, such a device might not include any buttons at all, and might be controlled only through a combination of visual and audio commands, such that a user can control the device without having to be in contact with the device.

[0059] The device 700 also can include at least one orientation or motion sensor 710. As discussed, such a sensor can include an accelerometer or gyroscope operable to detect an orientation and/or change in orientation, or an electronic or digital compass, which can indicate a direction in which the device is determined to be facing. The mechanism(s) also (or alternatively) can include or comprise a global positioning system (GPS) or similar positioning element operable to determine relative coordinates for a position of the computing device, as well as information about relatively large movements of the device. The device can include other elements as well, such as may enable location determinations through triangulation or another such approach. These mechanisms can communicate with the processor 702, whereby the device can perform any of a number of actions described or suggested herein.

[0060] As an example, a computing device such as that described with respect to FIG. 6 can capture and/or track various information for a user over time. This information can include any appropriate information, such as location, actions (e.g., sending a message or creating a document), user behavior (e.g., how often a user performs a task, the amount of time a user spends on a task, the ways in which a user navigates through an interface, etc.), user preferences (e.g., how a user likes to receive information), open

applications, submitted requests, received calls, and the like. As discussed above, the information can be stored in such a way that the information is linked or otherwise associated whereby a user can access the information using any appropriate dimension or group of dimensions.

[0061] As discussed, different approaches can be implemented in various environments in accordance with the described embodiments. For example, FIG. 8 illustrates an example of an environment 800 for implementing aspects in accordance with various embodiments. As will be appreciated, although a Web-based environment is used for purposes of explanation, different environments may be used, as appropriate, to implement various embodiments. The system includes an electronic client device 802, which can include any appropriate device operable to send and receive requests, messages or information over an appropriate network 804 and convey information back to a user of the device. Examples of such client devices include personal computers, cell phones, handheld messaging devices, laptop computers, set-top boxes, personal data assistants, electronic book readers and the like. The network can include any appropriate network, including an intranet, the Internet, a cellular network, a local area network or any other such network or combination thereof. Components used for such a system can depend at least in part upon the type of network and/or environment selected. Protocols and components for communicating via such a network are well known and will not be discussed herein in detail. Communication over the network can be enabled via wired or wireless connections and combinations thereof. In this example, the network includes the Internet, as the environment includes a Web server 806 for receiving requests and serving content in response thereto, although for other networks an alternative device serving a similar purpose could be used, as would be apparent to one of ordinary skill in the art.

[0062] The illustrative environment includes at least one application server 808 and a data store 810. It should be understood that there can be several application servers, layers or other elements, processes or components, which may be chained or otherwise configured, which can interact to perform tasks such as obtaining data from an appropriate data store. As used herein the term "data store" refers to any device or combination of devices capable of storing, accessing and retrieving data, which may include any combination and number of data servers, databases, data storage devices and data storage media, in any standard, distributed or clustered environment. The

application server can include any appropriate hardware and software for integrating with the data store as needed to execute aspects of one or more applications for the client device and handling a majority of the data access and business logic for an application. The application server provides access control services in cooperation with the data store and is able to generate content such as text, graphics, audio and/or video to be transferred to the user, which may be served to the user by the Web server in the form of HTML, XML or another appropriate structured language in this example. The handling of all requests and responses, as well as the delivery of content between the client device 802 and the application server 808, can be handled by the Web server 806. It should be understood that the Web and application servers are not required and are merely example components, as structured code discussed herein can be executed on any appropriate device or host machine as discussed elsewhere herein.

[0063] The data store 810 can include several separate data tables, databases or other data storage mechanisms and media for storing data relating to a particular aspect. For example, the data store illustrated includes mechanisms for storing production data 812 and user information 816, which can be used to serve content for the production side. The data store also is shown to include a mechanism for storing log or session data 814. It should be understood that there can be many other aspects that may need to be stored in the data store, such as page image information and access rights information, which can be stored in any of the above listed mechanisms as appropriate or in additional mechanisms in the data store 810. The data store 810 is operable, through logic associated therewith, to receive instructions from the application server 808 and obtain, update or otherwise process data in response thereto. In one example, a user might submit a search request for a certain type of element. In this case, the data store might access the user information to verify the identity of the user and can access the catalog detail information to obtain information about elements of that type. The information can then be returned to the user, such as in a results listing on a Web page that the user is able to view via a browser on the user device 802. Information for a particular element of interest can be viewed in a dedicated page or window of the browser.

[0064] Each server typically will include an operating system that provides executable program instructions for the general administration and operation of that server and typically will include computer-readable medium storing instructions that, when executed by a processor of the server, allow the server to perform its intended functions.

Suitable implementations for the operating system and general functionality of the servers are known or commercially available and are readily implemented by persons having ordinary skill in the art, particularly in light of the disclosure herein.

[0065] The environment in one embodiment is a distributed computing environment utilizing several computer systems and components that are interconnected via communication links, using one or more computer networks or direct connections. However, it will be appreciated by those of ordinary skill in the art that such a system could operate equally well in a system having fewer or a greater number of components than are illustrated in FIG. 8. Thus, the depiction of the system 800 in FIG. 8 should be taken as being illustrative in nature and not limiting to the scope of the disclosure.

[0066] As discussed above, the various embodiments can be implemented in a wide variety of operating environments, which in some cases can include one or more user computers, computing devices, or processing devices which can be used to operate any of a number of applications. User or client devices can include any of a number of general purpose personal computers, such as desktop or laptop computers running a standard operating system, as well as cellular, wireless, and handheld devices running mobile software and capable of supporting a number of networking and messaging protocols. Such a system also can include a number of workstations running any of a variety of commercially-available operating systems and other known applications for purposes such as development and database management. These devices also can include other electronic devices, such as dummy terminals, thin-clients, gaming systems, and other devices capable of communicating via a network.

[0067] Various aspects also can be implemented as part of at least one service or Web service, such as may be part of a service-oriented architecture. Services such as Web services can communicate using any appropriate type of messaging, such as by using messages in extensible markup language (XML) format and exchanged using an appropriate protocol such as SOAP (derived from the "Simple Object Access Protocol"). Processes provided or executed by such services can be written in any appropriate language, such as the Web Services Description Language (WSDL). Using a language such as WSDL allows for functionality such as the automated generation of client-side code in various SOAP frameworks.

[0068] Most embodiments utilize at least one network that would be familiar to those skilled in the art for supporting communications using any of a variety of commercially-available protocols, such as TCP/IP, OSI, FTP, UPnP, NFS, CIFS, and AppleTalk. The network can be, for example, a local area network, a wide-area network, a virtual private network, the Internet, an intranet, an extranet, a public switched telephone network, an infrared network, a wireless network, and any combination thereof.

[0069] In embodiments utilizing a Web server, the Web server can run any of a variety of server or mid-tier applications, including HTTP servers, FTP servers, CGI servers, data servers, Java servers, and business application servers. The server(s) also may be capable of executing programs or scripts in response requests from user devices, such as by executing one or more Web applications that may be implemented as one or more scripts or programs written in any programming language, such as Java®, C, C# or C++, or any scripting language, such as Perl, Python, or TCL, as well as combinations thereof. The server(s) may also include database servers, including without limitation those commercially available from Oracle®, Microsoft®, Sybase®, and IBM®.

[0070] The environment can include a variety of data stores and other memory and storage media as discussed above. These can reside in a variety of locations, such as on a storage medium local to (and/or resident in) one or more of the computers or remote from any or all of the computers across the network. In a particular set of embodiments, the information may reside in a storage-area network (“SAN”) familiar to those skilled in the art. Similarly, any necessary files for performing the functions attributed to the computers, servers, or other network devices may be stored locally and/or remotely, as appropriate. Where a system includes computerized devices, each such device can include hardware elements that may be electrically coupled via a bus, the elements including, for example, at least one central processing unit (CPU), at least one input device (e.g., a mouse, keyboard, controller, touch screen, or keypad), and at least one output device (e.g., a display device, printer, or speaker). Such a system may also include one or more storage devices, such as disk drives, optical storage devices, and solid-state storage devices such as random access memory (“RAM”) or read-only memory (“ROM”), as well as removable media devices, memory cards, flash cards, etc.

[0071] Such devices also can include a computer-readable storage media reader, a communications device (e.g., a modem, a network card (wireless or wired), an infrared communication device, etc.), and working memory as described above. The computer-

readable storage media reader can be connected with, or configured to receive, a computer-readable storage medium, representing remote, local, fixed, and/or removable storage devices as well as storage media for temporarily and/or more permanently containing, storing, transmitting, and retrieving computer-readable information. The system and various devices also typically will include a number of software applications, modules, services, or other elements located within at least one working memory device, including an operating system and application programs, such as a client application or Web browser. It should be appreciated that alternate embodiments may have numerous variations from that described above. For example, customized hardware might also be used and/or particular elements might be implemented in hardware, software (including portable software, such as applets), or both. Further, connection to other computing devices such as network input/output devices may be employed.

[0072] Storage media and computer readable media for containing code, or portions of code, can include any appropriate media known or used in the art, including storage media and communication media, such as but not limited to volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage and/or transmission of information such as computer readable instructions, data structures, program modules, or other data, including RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disk (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the a system device. Based on the disclosure and teachings provided herein, a person of ordinary skill in the art will appreciate other ways and/or methods to implement the various embodiments.

[0073] The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense. It will, however, be evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the claims.

[0074] Various embodiments of the disclosure can be described in view of the following clauses:

A1. A computer-implemented method for index configuration for searchable data in a networked environment, comprising:

receiving data to be indexed, the data including a plurality of data fields;

determining a name associated with each data field in the plurality of data fields;

determining a field type, of a plurality of field types, associated with each data field in the plurality of data fields, the plurality of field types including at least one of an integer type, a literal type, or a text type;

determining whether to enable one or more search options for each of the data fields, the one or more search options including at least one of an option to include a respective data field in a search index to be generated, an option to calculate a facet count for the respective data field, or an option to provide one or more values associated with the respective data field;

generating a search index configuration for the data based at least in part on the field type of each data field included the data and the determining whether to enable the one or more search options; and

generating a search index for the data based at least in part on the search index configuration for the data.

A2. The computer-implemented method of clause A1, wherein the data is of a first format, further comprising:

converting the data from the first format to a second format, the second format being compatible with the search index; and

storing the data converted to the second format on one or more storage allocations.

A3. The computer-implemented method of clause A2, wherein the converting the data from the first format to the second format comprises:

comparing the first format with the second format; and

modifying at least one data field associated with the first format to correspond to at least one data field associated with the second format.

A4. The computer-implemented method of clause A2, wherein the second format is a Search Data Format (SDF).

A5. A computer-implemented method comprising:
receiving data to be indexed;
determining a type of a data field associated with the data, the type of the data field being determined from a plurality of types of data fields;
determining one or more search options to be enabled with respect to the data field associated with the data;
generating an index configuration for the data based at least in part on the type of the data field and the one or more search options; and
generating a search index for the data based at least in part on the index configuration for the data.

A6. The computer-implemented method of clause A5, wherein the data is of a first format, further comprising:
converting the data from the first format to a second format, the second format being compatible with the search index; and
storing the data converted to the second format on one or more storage allocations.

A7. The computer-implemented method of clause A6, wherein the converting the data from the first format to the second format comprises:
comparing the first format with the second format; and
modifying at least one data tag associated with the first format to correspond to at least one data tag associated with the second format.

A8. The computer-implemented method of clause A5, wherein the plurality of types of data fields includes at least one of an integer type, a text type, a literal type, a geolocation type, a time type, a date type, or a floating point number type.

A9. The computer-implemented method of clause A8, wherein the determining the type of the data field comprises:
determining that a value associated with the data field has an amount of integer characters above a specified integer amount threshold; and
determining the type of the data field to be the integer type.

A10. The computer-implemented method of clause A8, wherein the determining the type of the data field comprises:

determining at least one of a value associated with the data field having an amount of alphanumeric characters above a specified text amount threshold, a number of distinct values associated with the data field being above a specified text distinct amount threshold, a percentage of distinct values being above a specified text distinct percentage threshold, or a length of values being above a specified text length threshold; and

determining the type of the data field to be the text type.

A11. The computer-implemented method of clause A8, wherein the determining the type of the data field comprises:

determining at least one of a value associated with the data field having an amount of alphabetic characters above a specified lower literal amount threshold but below a specified upper literal amount threshold, a number of distinct values associated with the data field being below a specified literal distinct amount threshold, a percentage of distinct values being below a specified literal distinct percentage threshold, or a length of values being below a specified literal length threshold; and

determining the type of the data field to be the literal type.

A12. The computer-implemented method of clause A5, wherein the one or more search options include at least one of an option to include the data field in the search index to be generated, an option to calculate a facet count for the data field, or an option to provide a value associated with the data field in response to a relevant search query.

A13. The computer-implemented method of clause A12, wherein the determining the one or more search options to be enabled comprises deciding to enable the option to include the data field in the search index to be generated, the decision being based at least in part on at least one of receiving a signal included in the data field indicating that the data field is to be included in the search index or determining a type of the data field to be a literal type.

A14. The computer-implemented method of clause A12, wherein the determining the one or more search options to be enabled comprises deciding to enable the option to calculate the facet count for the data field, the decision being based at least in part on determining that a quantity for a distribution of a plurality of values associated with the data field is below a specified facet count upper threshold.

A15. The computer-implemented method of clause A12, wherein the determining the one or more search options to be enabled comprises deciding to enable the option to provide the value associated with the data field in response to the relevant search query, the decision being based at least in part on at least one of receiving a signal included in the data field indicating that the value associated with the data field is to be provided or determining that a length of the value associated with the data field is below a specified return value length threshold.

A16. The computer-implemented method of clause A5, further comprising:
providing at least one of the data, the index configuration, or the index to be searchable by one or more search queries.

A17. The computer-implemented method of clause A5, further comprising:
modifying the index configuration based at least in part on one or more user-initiated inputs.

A18. A system comprising:
at least one communications transceiver;
one or more storage allocations;
at least one processor; and
a memory device including instructions that, when executed by the at least one processor, cause the system to:
receive, via the at least one communications transceiver, data to be indexed;
determine a type of a data field associated with the data, the type of the data field being determined from a plurality of types of data fields;

determine one or more search options to be enabled with respect to the data field associated with the data;

generate an index configuration for the data based at least in part on the type of the data field and the one or more search options; and

generate a search index for the data based at least in part on the index configuration for the data.

A19. The system of clause A18, wherein the data is of a first format, and wherein the instructions cause the system to further:

convert the data from the first format to a second format, the second format being compatible with the search index; and

store the data converted to the second format on the one or more storage allocations.

A20. The system of clause A19, wherein the instructions cause the system to convert the data from the first format to the second format based on comparing the first format with the second format and modifying at least one data field associated with the first format to correspond to at least one data field associated with the second format.

A21. A non-transitory computer-readable storage medium including instructions for identifying elements, the instructions when executed by a processor of a computing system causing the computing system to:

receive data to be indexed;

determine a type of a data field associated with the data, the type of the data field being determined from a plurality of types of data fields;

determine one or more search options to be enabled with respect to the data field associated with the data;

generate an index configuration for the data based at least in part on the type of the data field and the one or more search options; and

generate a search index for the data based at least in part on the index configuration for the data.

A22. The non-transitory computer-readable storage medium of clause A21, wherein the plurality of types of data fields includes at least one of an integer type, a text type, a literal type, a geolocation type, a time type, a date type, or a floating point number type.

A23. The non-transitory computer-readable storage medium of clause A22, wherein the instructions cause the computing system to determine the type of the data field to be the literal type based on determining at least one of a value associated with the data field having an amount of alphabetic characters above a specified lower literal amount threshold but below a specified upper literal amount threshold, a number of distinct values associated with the data field being below a specified literal distinct amount threshold, a percentage of distinct values being below a specified literal distinct percentage threshold, or a length of values being below a specified literal length threshold.

A24. The non-transitory computer-readable storage medium of clause A21, wherein the one or more search options include at least one of an option to include the data field in the search index to be generated, an option to calculate a facet count for the data field, or an option to provide a value associated with the data field in response to a relevant search query.

A25. The non-transitory computer-readable storage medium of clause A24, wherein the determining the one or more search options to be enabled comprises deciding to enable the option to calculate the facet count for the data field, the decision being based at least in part on determining that a quantity for at least one value associated with the data field is above a specified facet count lower threshold and below a specified facet count upper threshold.

B1. A computer-implemented method for dynamic search partitioning, comprising:

monitoring at least one of an amount of data being stored or a rate at which data is being manipulated on a first partition provided by a network service, the first partition being included in a storage allocation provided by the network service;

detecting that the at least one of the amount or the rate exceeds a specified amount threshold or a specified rate threshold, respectively;

performing, in response to the detecting, at least one of an increase to a size of the first partition or an addition of at least a second partition to the storage allocation, the at least one of the increase or the addition being based at least in part on the amount of data being stored or the rate at which data is being manipulated;

directing network traffic associated with the storage allocation to a cache provided by the network service during the at least one of the increase or the addition; and

directing the network traffic to the storage allocation when the performing the at least one of the increase or the addition is complete.

B2. The computer-implemented method of clause B1, further comprising:

monitoring a search index for the storage allocation;

detecting that a size of the search index exceeds a specified index size threshold; and

updating the search index for the storage allocation to reflect the at least one of the increase or the addition with respect to the storage allocation.

B3. The computer-implemented method of clause B1, wherein the increase to the size of the first partition is performed if the size of the first partition is below a maximum partition size threshold, and wherein the addition of at least the second partition is performed if the size of the first partition is at the maximum partition size threshold.

B4. A computer-implemented method comprising:

monitoring data usage on a storage allocation in a networked environment, the storage allocation having a number of partitions including at least one partition;

determining whether the data usage on the at least one partition included in the storage allocation exceeds a specified threshold;

modifying at least one of a size of the at least one partition or the number of partitions included in the storage allocation;

directing network traffic associated with the storage allocation away from a portion of the storage allocation associated with the modifying of the at least one of the size or the number; and

directing the network traffic to the portion of the storage allocation associated with the modifying when the modifying is complete.

B5. The computer-implemented method of clause B4, further comprising:

detecting that a size of a search index for the storage allocation exceeds a specified index size threshold; and

updating the search index for the storage allocation based on the modifying the at least one of the size of the at least one partition or the number of partitions included in the storage allocation.

B6. The computer-implemented method of clause B5, wherein the updating the search index includes rebuilding the search index for the storage allocation to reflect the modifying the at least one of the size of the at least one partition or the number of partitions included in the storage allocation.

B7. The computer-implemented method of clause B4, wherein the data usage includes at least one of an amount of data being stored on the storage allocation or a rate at which data is being manipulated on the storage allocation.

B8. The computer-implemented method of clause B7, wherein the specified threshold includes at least one of a specified amount threshold or a specified rate threshold, and wherein the data usage exceeds the specified threshold when there is an occurrence of at least one of the amount of data being stored exceeds the specified amount threshold or the rate at which data is being manipulated exceeds the specified rate threshold.

B9. The computer-implemented method of clause B8, wherein the specified threshold is calculated based at least in part on information about historical data usage.

B10. The computer-implemented method of clause B4, further comprising:

determining that an amount of network traffic directed to the storage allocation is above a specified traffic threshold; and
modifying the storage allocation based on the amount of network traffic.

B11. The computer-implemented method of clause B10, wherein the network traffic includes search query traffic for searching data stored on the storage allocation.

B12. The computer-implemented method of clause B10, wherein the modifying the storage allocation based on the amount of network traffic includes at least one of modifying the size of the at least one partition, modifying the number of partitions, or replacing at least one partition included in the number of partitions with at least one partition having different specifications.

B13. The computer-implemented method of clause B12, wherein the different specifications include at least one of a different CPU power, a different capacity of RAM, a different capacity of hard disk space, or a different capacity of bandwidth.

B14. The computer-implemented method of clause B4, wherein the modifying the at least one of the size of the at least one partition or the number of partitions includes increasing at least one of the size of the at least one partition or the number of partitions, wherein the increasing the size of the at least one partition is performed if the size of the at least one partition is below a maximum partition size threshold, and wherein the increasing the number of the partitions is performed if the size of the at least one partition is at the maximum partition size threshold

B15. The computer-implemented method of clause B4, wherein the modifying the at least one of the size of the at least one partition or the number of partitions includes decreasing at least one of the size of the at least one partition or the number of partitions, wherein the decreasing the number of the partitions is performed if the number of the partitions is greater than one partition, and wherein the decreasing the

size of the at least one partition is performed if the number of the partitions is one partition.

B16. The computer-implemented method of clause B4, further comprising:

determining a CPU usage of the storage allocation, wherein the modifying the at least one of the size or the number is based on at least one of the data usage on the storage allocation or the determined CPU usage of the storage allocation.

B17. The computer-implemented method of clause B4, further comprising:

modifying a configuration of the storage allocation based on at least one of a configuration associated with the data usage or a user-initiated input.

B18. The computer-implemented method of clause B4, further comprising:

determining when to perform the modifying of the at least one of the size or the number based on resources available to the storage allocation.

B19. A system comprising:

a storage allocation having a number of partitions including at least one partition;

at least one processor; and

a memory device including instructions that, when executed by the at least one processor, cause the system to:

monitor data usage on the storage allocation;

determine whether the data usage on the at least one partition included in the storage allocation exceeds a specified threshold;

modify at least one of a size of the at least one partition or the number of partitions included in the storage allocation;

direct network traffic associated with the storage allocation away from a portion of the storage allocation associated with the modifying of the at least one of the size or the number; and

direct the network traffic to the portion of the storage allocation associated with the modifying when the modifying is complete.

B20. The system of clause B19, further comprising:

at least one load balancer configured to facilitate in the network traffic being directed away from the portion of the storage allocation during the modifying the at least one of the size or the number and in the network traffic being directed to the portion of the storage allocation when the modifying the at least one of the size or the number is complete.

B21. The system of clause B20, wherein the at least one load balancer is configured to direct the network traffic across the number of partitions included in the storage allocation.

B22. The system of clause B19, further comprising:

at least one monitor module configured to facilitate in the monitoring the data usage on the storage allocation and in the determining whether the data usage on the at least one partition included in the storage allocation exceeds a specified threshold.

B23. A non-transitory computer-readable storage medium including instructions for identifying elements, the instructions when executed by a processor of a computing system causing the computing system to:

monitor data usage on a storage allocation in a networked environment, the storage allocation having a number of partitions including at least one partition;

determine whether the data usage on the at least one partition included in the storage allocation exceeds a specified threshold;

modify at least one of a size of the at least one partition or the number of partitions included in the storage allocation;

direct network traffic associated with the storage allocation away from a portion of the storage allocation associated with the modifying of the at least one of the size or the number; and

direct the network traffic to the portion of the storage allocation associated with the modifying when the modifying is complete.

B24. The non-transitory computer-readable storage medium of clause B23, wherein the instructions cause the computing system to further detect that a size of a search index for the storage allocation exceeds a specified index size threshold and update the search index for the storage allocation based on the modifying the at least one of the size of the at least one partition or the number of partitions included in the storage allocation.

B25. The non-transitory computer-readable storage medium of clause B24, wherein the updating the search index includes rebuilding the search index for the storage allocation to reflect the modifying the at least one of the size of the at least one partition or the number of partitions included in the storage allocation.

What is claimed is:

1. A computer-implemented method comprising:
 - monitoring data usage on a storage allocation in a networked
 - 5 environment, the storage allocation having a number of partitions including at least one partition;
 - determining whether the data usage on the at least one partition included in the storage allocation exceeds a specified threshold;
 - modifying at least one of a size of the at least one partition or the number
 - 10 of partitions included in the storage allocation;
 - directing network traffic associated with the storage allocation away from a portion of the storage allocation associated with the modifying of the at least one of the size or the number; and
 - directing the network traffic to the portion of the storage allocation
 - 15 associated with the modifying when the modifying is complete.
2. The computer-implemented method of claim 1, further comprising:
 - detecting that a size of a search index for the storage allocation exceeds a
 - specified index size threshold; and
 - 20 updating the search index for the storage allocation based on the modifying the at least one of the size of the at least one partition or the number of partitions included in the storage allocation.
3. The computer-implemented method of claim 2, wherein the updating the
- 25 search index includes rebuilding the search index for the storage allocation to reflect the modifying the at least one of the size of the at least one partition or the number of partitions included in the storage allocation.
- 30 4. The computer-implemented method of claim 1, wherein the data usage includes at least one of an amount of data being stored on the storage allocation or a rate at which data is being manipulated on the storage allocation.

5. The computer-implemented method of claim 4, wherein the specified threshold includes at least one of a specified amount threshold or a specified rate threshold, and wherein the data usage exceeds the specified threshold when there is an occurrence of at least one of the amount of data being stored exceeds the specified amount threshold or the rate at which data is being manipulated exceeds the specified rate threshold.

5
6. The computer-implemented method of claim 5, wherein the specified threshold is calculated based at least in part on information about historical data usage.

10
7. The computer-implemented method of claim 1, further comprising:
determining that an amount of network traffic directed to the storage allocation is above a specified traffic threshold; and
15 modifying the storage allocation based on the amount of network traffic.
8. The computer-implemented method of claim 7, wherein the network traffic includes search query traffic for searching data stored on the storage allocation.

20
9. The computer-implemented method of claim 7, wherein the modifying the storage allocation based on the amount of network traffic includes at least one of modifying the size of the at least one partition, modifying the number of partitions, or replacing at least one partition included in the number of partitions with at least one partition having different specifications.

25
10. The computer-implemented method of claim 9, wherein the different specifications include at least one of a different CPU power, a different capacity of RAM, a different capacity of hard disk space, or a different capacity of bandwidth.

30
11. The computer-implemented method of claim 1, wherein the modifying the at least one of the size of the at least one partition or the number of

5 partitions includes increasing at least one of the size of the at least one partition or the number of partitions, wherein the increasing the size of the at least one partition is performed if the size of the at least one partition is below a maximum partition size threshold, and wherein the increasing the number of the partitions is performed if the size of the at least one partition is at the maximum partition size threshold.

10 12. The computer-implemented method of claim 1, wherein the modifying the at least one of the size of the at least one partition or the number of partitions includes decreasing at least one of the size of the at least one partition or the number of partitions, wherein the decreasing the number of the partitions is performed if the number of the partitions is greater than one partition, and wherein the decreasing the size of the at least one partition is performed if the number of the partitions is one partition.

15 13. The computer-implemented method of claim 1, further comprising:
determining a CPU usage of the storage allocation, wherein the modifying the at least one of the size or the number is based on at least one of the data usage on the storage allocation or the determined CPU usage of the storage allocation.

20 14. The computer-implemented method of claim 1, further comprising:
modifying a configuration of the storage allocation based on at least one of a configuration associated with the data usage or a user-initiated input.

25 15. The computer-implemented method of claim 1, further comprising:
determining when to perform the modifying of the at least one of the size or the number based on resources available to the storage allocation.