



- (51) **International Patent Classification:**
G06F 15/16 (2006.01) *G06F 15/167* (2006.01)
- (21) **International Application Number:**
PCT/US2015/018019
- (22) **International Filing Date:**
27 February 2015 (27.02.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** NOVAKOVIC, Stanko; 1501 Page Mill Rd., Palo Alto, California 94304 (US). FARABOSCHI, Paolo; 1501 Page Mill Rd., Palo Alto, California 94304 (US). KEETON, Kimberly; 1501 Page Mill Rd., Palo Alto, California 94304 (US). ZHAO, Jishen; 1501 Page Mill Rd., Palo Alto, California 94304 (US). SCHREIBER, Robert; 1501 Page Mill Rd., Palo Alto, California 94304 (US).
- (74) **Agents:** CHEN, Lawrence M. et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— as to the identity of the inventor (Rule 4.17(i))

Published:

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

(54) **Title:** RESPONSIVE SERVER IDENTIFICATION AMONG MULTIPLE DATA SERVERS LINKED TO A SHARED MEMORY

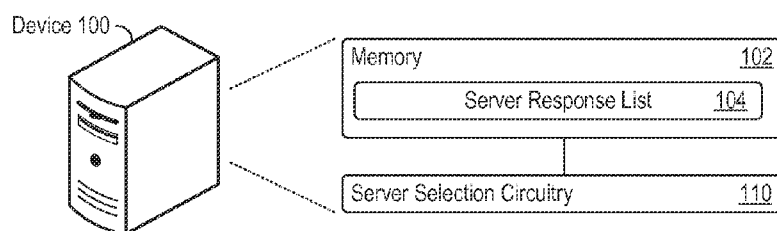


Figure 1

(57) **Abstract:** In some examples, a method may be performed by server selection circuitry in a device. The method may include sending a first data query to multiple data servers, the multiple data servers linked to a shared memory storing data requested by the first data query. The method may also include tracking response times for the multiple data servers, where the response times include a particular response time for each of the multiple data servers to service the first data query, identifying a responsive server from among the multiple data servers according to the response times, and sending a second data query to the responsive server.

RESPONSIVE SERVER IDENTIFICATION AMONG MULTIPLE DATA SERVERS LINKED TO A SHARED MEMORY

BACKGROUND

[0001] With rapid advances in technology, computing systems are increasingly prevalent in society today. Vast computing systems execute and support applications that communicate and process immense amounts of data, many times with performance constraints to meet the increasing demands of users. Increasing the efficiency, speed, and effectiveness of computing systems will further improve user experience.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Certain examples are described in the following detailed description and in reference to the drawings.

[0003] Figure 1 shows an example of a device that supports responsive server identification among multiple data servers linked to a shared memory.

[0004] Figure 2 shows an example of the server selection circuitry identifying a responsive server among multiple data servers linked to a shared memory.

[0005] Figure 3 shows an example of the server selection circuitry selecting a responsive server for servicing a data query.

[0006] Figure 4 shows an example of logic that the server selection circuitry may implement.

[0007] Figure 5 shows another example of logic that the server selection circuitry may implement.

[0008] Figure 6 shows an example of a computing device that supports responsive server identification among multiple data servers linked to a shared memory.

DETAILED DESCRIPTION

[0009] Figure 1 shows an example of a device 100 that supports server selection for data servers linked to a shared memory. The device 100 may be a client device that executes a client application. Thus, the device 100 may be, for example, a desktop or laptop computer, a web server, a data center computing device, a personal computing device, and more.

[0010] The device 100 may include a memory 102. The memory 102 may be any machine-readable storage medium such as Random Access Memory (RAM), an Electrically-Erasable Programmable Read-Only Memory (EEPROM), a storage drive, an optical disk, a processor cache, or any other volatile or non-volatile storage medium. The memory 102 may store a server response list 104, which may list multiple data servers linked to the device 100. The multiple data servers may be linked to a shared memory (not shown) storing data for querying. The device 100 (or application executing on the device 100) may request data stored in the shared memory through a data query.

[0011] The device 100 may include server selection circuitry 110 through which the device 100 may determine a particular data server to send a data query through. The server selection circuitry 110 may be implemented through any combination of hardware (e.g., a processor) or machine-readable instructions (e.g., stored on a machine-readable medium). Thus, in some examples, the server selection circuitry 110 includes a processor and a machine-readable medium storing executable instructions to perform any of the features described herein.

[0012] As described in greater detail below, the server selection circuitry 110 may identify a responsive server amongst the multiple data servers linked to the shared memory in which to send a data query to retrieve requested data. In doing so, the server selection circuitry 110 may apply any number of responsiveness criteria in determining whether a data server is responsive or

unresponsive. The server selection circuitry 110 may track responsive and unresponsive data servers through the server response list 104. In some examples, the server selection circuitry 110 maintains the server response list by 104 determining response times for the multiple data servers to service a data query, including concurrently sending the data query to the multiple data servers to track the response time for each of the multiple servers to service the data query. The server selection circuitry 110 may then identify responsive servers among the multiple data servers listed in the server response list 104 according to the response time to service the data query and select one of the responsive servers to send a subsequent data query to be serviced through the shared memory.

[0013] Figure 2 shows an example of the server selection circuitry 110 identifying a responsive server among multiple data servers linked to a shared memory. In the example shown in Figure 2, the server selection circuitry 110 is communicatively linked to the multiple data servers 202 including the servers labeled as data server A, data server B, data server C, and data server D. The multiple data servers 202 may be some or all of the data servers linked to a shared memory through which the server selection circuitry 110 may query for data.

[0014] The server selection circuitry 110 may query data through any one of the multiple data servers 202. A data server may include any combination of circuitry or logic for performing a particular task or operation. For example, the multiple data servers 202 may be part of a data serving system through which client applications (e.g., a web server) request data. An individual data server among the multiple data servers 202 may service a data query from a client application by retrieving the queried data from a shared memory 210 accessible to the multiple data servers 202. The server selection circuitry 110 may send data queries to the multiple data servers 202 through a communication network 204. The communication network 204 may include a local area network (LAN), for example as part of a data center implementation as client devices interface with backend servers.

[0015] The multiple data servers 202 may be linked to a shared memory, such as the shared memory 210 shown in Figure 2. The shared memory 210 may implement a common memory namespace accessible to client applications through the multiple data servers 202. The shared memory 210 may take the form of a random access memory (RAM), which may be a dynamic RAM or a non-volatile RAM for example. In some examples, the shared memory is byte-addressable, thus supporting access to a particular memory address or memory address range within the shared memory. In other examples, the shared memory is block addressable. A memory medium that implements the shared memory may be volatile or non-volatile. Thus, in some examples, the shared memory is a non-volatile computer storage medium, such as a non-volatile RAM, a hard drive, flash memory, optical disk, memristor array, solid state drive, and the like.

[0016] The multiple data servers 202 and shared memory 210 may be implemented, for example, as a rack-scale system. The multiple data servers 202 may access the shared memory 210 through a high-speed memory fabric or network, supporting increased access speed to retrieve data as compared to shared disk systems. A data serving system or other system implementing the multiple data servers 202 may share physical resources of the rack-scale system with other applications or services, which may cause resource contention and delay in servicing data queries. The server selection circuitry 110 may reduce such delay by identifying responsive data servers among the multiple data servers 202. In a rack-scale system, any of the multiple data servers 202 may be capable of servicing a data query, and thus the server selection circuitry 110 may leverage the common access to the shared memory 210 and flexibly select a particular data server that meets specific performance constraints or quality of service (QoS) requirements, e.g., as set forth by a client application.

[0017] Next, various features of the server selection circuitry 110 are described with respect to a probe phase and a selection phase. The server selection circuitry 110 may implement any combination of the features

described for the probe phase and selection phase, e.g., operating at separate times, sequentially, concurrently, or according to any other combination.

Probe Phase

[0018] In a probe phase, the server selection circuitry 110 may collect response data from the multiple data servers 202. Through the collected response data, the server selection circuitry 110 may categorize an individual data server as responsive or unresponsive, which may also be referred to as a responsiveness determination. In some examples, the server selection circuitry 110 collects response data as a response latency for an individual data server to service (e.g., respond to) a data query. As other examples, the server selection circuitry 110 may collect response data as bandwidth availability, resource consumption, or any other performance metric. As discussed in greater detail below, the server selection circuitry 110 may categorize responsive and unresponsive servers according to collected response data.

[0019] To collect response data from the multiple data servers 202, the server selection circuitry 110 may send a data query to some or all of the multiple data servers 202. In the example shown in Figure 2, the server selection circuitry 110 sends the data query 220 to each of the data servers A, B, C, and D. The data query 220 may be a data query or request generated by a client application, such as a web search application querying for a search term. In sending the data query, the server selection circuitry 110 may send the same data query 220 to more than one data server, even though any individual data server can service the data query 220. The server selection circuitry 110 may, for example, clone the data query 220 so as to send an instance of the data query 220 to some or all of the multiple data servers 202.

[0020] The server selection circuitry 110 may concurrently send the data query 220 to the multiple data servers 202. In that regard, the server selection circuitry 110 may duplicate and send the data query 220 to more than one data server, independent of whether any of the data servers has responded to the data query 220. Put another way, the server selection circuitry 110 may not inject a calculated or deliberate delay in sending successive clones of the data query 220 to successive data servers among the multiple data servers 202,

stopping the transmission of the data query 220 if another data server responds to earlier sent instance of the data query 220.

[0021] To illustrate in another way, the server selection circuitry 110 may unconditionally send the instances of the data queries 220 to the multiple data servers. In that regard, the server selection circuitry may inject delay (e.g., to prevent overloading the shared memory or causing resource contention), but nonetheless send each of the cloned instances of the data query 220 to the multiple data servers without dependence on whether a response has been received for a previously sent instance of the data query 220. The server selection circuitry 110 may thus send the cloned copies of the data query 220 without hedging, and may instead concurrently (or with delay) send the data query 220 to multiple data servers 202 to service the query as well as to collect response data. The server selection circuitry 110 may thus send the cloned copies of the data query 220 without hedging.

[0022] The server selection circuitry 110 may record the response time for individual data servers to respond to the data query 220. Upon receiving responses to the data query 220, the server selection circuitry 110 may populate corresponding portions of the server response list 104 with the response times. In the example shown in Figure 2, the server selection circuitry 110 records the response times of 500 microseconds (μ s), 450 μ s, 950 μ s, and 300 μ s for the data servers A, B, C, and D respectively to respond to the data query 220. As the data query 220 may originate from a client application, the server selection circuitry 110 may determine the data query 220 as serviced upon receiving the first response from the multiple data servers 202. Thus, in Figure 2, the server selection circuitry 110 may receive the requested data when the data server D responds to the data query 220 with a shortest response time of 300 μ s, providing the queried data returned by data server D to the client application.

[0023] During the probe phase, the server selection circuitry 110 may record response data for some or all of the multiple data servers 202. The server selection circuitry 110 may, for example, populate the server response list 104 through multiple, different data queries directed at different subsets of the pool

of data servers linked to the shared memory 210. The server selection circuitry 110 may probe subsets of the multiple data servers 202 through different data queries until response data has been collected for the multiple data servers 202. To illustrate, the server selection circuitry 110 may probe 'k' number of distinct data servers through a first data query, cloning and concurrently sending the first data query to the 'k' distinct data servers. The 'k' distinct data servers may be some, but not all, of the multiple data servers linked to the shared memory and capable of servicing the data query. For the example shown in Figure 2, the server selection circuitry 110 may clone the first data query for sending to the data servers A and B, for instance. The server selection circuitry 110 may record the response times for the 'k' distinct data servers to service the first data query.

[0024] Then, the server selection circuitry 110 may clone and concurrently send a second data query to another data server subset among the multiple data servers 202, such as data servers C and D in Figure 2. The server selection circuitry 110 may thus record the response times for the next subset of the multiple data servers 202. The server selection circuitry 110 may continue to clone and send data queries until, for example, response data has been collected for each of the multiple data servers 202. Thus, in a probe phase, the server selection circuitry 110 may collect response data for the multiple data servers 202, which may be tracked through the server response list 104.

Selection Phase

[0025] In a selection phase, the server selection circuitry 110 may identify responsive servers among the multiple data servers 202 and select a particular responsive server to query data through. Figure 3 shows an example of the server selection circuitry 110 selecting a responsive server for servicing a data query. In particular, the server selection circuitry 110 may perform a responsive server determination using the response data collected during the probe phase. The response data may be specified in the server response list 104, for example. In Figure 3, the server selection circuitry 110 tracks response times for the data servers A, B, C, and D through the server response list 104. In the particular example shown in Figure 3, the server response list 104 indicates

response times of 500 μ s, 450 μ s, 950 μ s, and 300 μ s for data servers A, B, C, and D respectively to service the data query 220.

[0026] The server selection circuitry 110 may identify a data server as responsive according to any number of responsiveness criteria, one example of which is through a response latency threshold. The response latency threshold may specify a time boundary for servicing a data query that a data server is to satisfy to be categorized as responsive. Thus, the server selection circuitry 110 may identify responsive servers from the server response list 104 as those with an associated response time that does not exceed a response latency threshold. In the example shown in Figure 3, the server selection circuitry 110 may define a response latency threshold as 700 μ s according to an application QoS requirement, and thus identify data servers A, B, and D as responsive and server C as unresponsive. The server response list 104 may provide an indication as to which of the multiple data servers 202 are determined as responsive according to the responsiveness criteria, and which are determined as unresponsive. For example, the server response list 104 may include a responsiveness flag or bit indicating the categorization of a particular data server.

[0027] The response latency threshold may be set in various ways. In some examples, the server selection circuitry 110 defines a response latency threshold according to a QoS requirement specified by a client application that originates the data query. The application QoS requirement may specify a memory access bandwidth requirement, throughput requirement, resource availability requirement, or any other quality or performance metric. The server selection circuitry 110 may translate the application QoS requirement into a corresponding response latency threshold that satisfies the application QoS requirement. Similarly, for other types of collected response data (e.g., bandwidth requirement), the server selection circuitry 110 may translate the application QoS requirement into a corresponding responsiveness criteria for the response data that the server selection circuitry 110 may apply for the responsive server determination. Accordingly, the server selection circuitry 110 may identify a responsive data server among the multiple data servers 202 on

an application-specific basis. As application QoS requirements may vary by application, the server selection circuitry 110 may flexibly support differentiated responsive server determination for different applications with varying quality and/or performance requirements.

[0028] The server selection circuitry 110 may select a responsive server for handling a data query. Thus, in Figure 3, the server selection circuitry 110 may select a particular responsive server among data servers A, B, and D for servicing a data query. In some examples, the server selection circuitry 110 selects among the responsive servers in a round-robin manner. In other examples, the server selection circuitry 110 randomly selects among the identified responsive servers. Selection techniques among multiple responsive servers may be feasible as any of the responsive data servers may be capable of servicing the data query, as the data servers share access to the shared memory 210. As a client application generates data queries for stored data in the shared memory 210, the server selection circuitry 110 may select responsive data servers for servicing the data queries, e.g., data servers with response times that meet the QoS requirements for the client application. Implementing the features described herein, the server selection circuitry 110 may maintain a QoS level for data service in a shared memory system, which may be provided through selection of responsive servers in the selection phase.

[0029] While the probe phase and selection phase are distinctly discussed above, the server selection circuitry 110 may implement the features described herein in combination. For example, during the selection phase, the server selection circuitry 110 may continue to monitor response times or otherwise collect response data for the multiple data servers 202, whether for responsive data servers, unresponsive data servers, or both. Even after an initial probe phase (e.g., upon startup of an application or physical device(s) implementing server selection circuitry 110), the server selection circuitry 110 may update the server response list 104 with collected response data and perform subsequent responsive and unresponsive server determinations for the multiple data servers 202.

[0030] In some examples, the server selection circuitry 110 differentiates between different types of data queries. A client application may generate multiple data query types, examples of which include search queries, scan queries, add queries, change queries, delete queries, and more. Different data query types may be characterized with different QoS requirements or response latency thresholds. The server selection circuitry 110 may separately track response times and apply separate responsiveness criteria for the multiple data query types. Thus, for a particular data query type (e.g., web search data query), the server selection circuitry 110 may identify responsive servers specifically using response data relevant to the particular data query type, such as response times for the multiple data servers 202 in servicing the particular data query type. In some examples, the server response list 104 separately tracks response times or other response data for multiple data query types. In other examples, the server selection circuitry 110 may maintain multiple server response lists 104, e.g., individually for a particular data query type. Along similar lines, the server selection circuitry 110 may apply varying response latency thresholds depending on the particular data query type being serviced, which may be preconfigured, customized, or set forth according to an application QoS requirement for the particular data query types.

[0031] Along similar lines, the server selection circuitry 110 may differentiate data queries of different sizes. The data size of returned data for the data query may impact performance, and impact meeting of application QoS requirements. As one example, data queries for selective data searches may return less result data whereas less selective or broader searches may return a greater amount of result data. The server selection circuitry 110 may differentiate between these data queries of differing size by separately collecting and tracking response data for data queries within particular data size tiers, for example. Thus, the server response list 104 may track response data for the multiple data servers with respect to the multiple data size tiers, which may support application of different or individualized response latency thresholds for different data size tiers.

[0032] Through responsive server determination and selection, the server selection circuitry 110 may reduce tail latencies. Tail latencies may refer to the latency value at the tail or upper end of a latency distribution (e.g., 95th or 99th percentile). In a data serving system, data queries or other web server operations may be subject to resource contention resulting from other applications or services concurrently utilizing the same physical resources. High server response times may result in shared resource systems, which may result in high tail latency values. The server selection circuitry 110 may reduce the tail latency values for a data serving system, through specifically querying responsive servers among the multiple data servers 202, each of which may be capable of retrieving data from the shared memory. Moreover, by concurrently sending cloned instances of a data query to multiple data servers, the server selection circuitry 110 may take advantage of the access speeds for a shared memory implemented as non-volatile memory (e.g., a memristor array). Doing so may likewise reduce tail latency values.

[0033] Figure 4 shows an example of logic 400 that the server selection circuitry 110 may implement. The server selection circuitry 110 may implement the logic 400 as hardware and/or machine-readable instructions, for example. The server selection circuitry 110 may execute the logic 400 as a process or method to identify a responsive server among multiple data servers linked to a shared memory.

[0034] The server selection circuitry 110 may send a first data query to multiple data servers, the multiple data servers linked to a shared memory storing data requested by the first data query (402). In some examples, each individual server among the multiple data servers can service the first data query by retrieving requested data from the shared memory. For example, the multiple data servers and shared memory may be implemented as part of a rack-scale system. The multiple data servers to which the server selection circuitry 110 sends the first data query may be some or all of the data servers that access the shared memory. In sending the first data query, the server selection circuitry 110 may clone the first data query to send an instance of the first data query to each of the multiple data servers.

[0035] The server selection circuitry 110 may track response times for the multiple data servers (404), and the response times may include a particular response time for each of the multiple data servers to service the first data query. In some examples, the server selection circuitry 110 tracks the response times through a server response list 104. The server selection circuitry 110 may identify a responsive server from among the multiple servers according to the response times (406), such as by applying a response latency threshold corresponding to an application QoS requirement. The server selection circuitry 110 may identify a responsive server by determining that the particular response time for the responsive server to service the first data query does not exceed the response latency threshold. Then, the server selection circuitry 110 may send a second data query to the response server (408).

[0036] Figure 5 shows an example of logic 500 that the server selection circuitry 110 may implement. The server selection circuitry 110 may implement the logic 500 as hardware and/or machine-readable instructions, for example. The server selection circuitry 110 may execute the logic 500 as a process or method to support identifying and selecting a responsive server among multiple data servers linked to a shared memory.

[0037] The server selection circuitry 110 may monitor the multiple data servers to determine a previously identified responsive server as unresponsive, for example during the selection phase and after an initial probe phase. When sending a data query to a responsive server (e.g., selected through the server response list 104), the server selection circuitry 110 may monitor and track the response time for the responsive server to service the data query. The data query may be a second data query sent to the responsive server (the first data query being sent as part of the probe phase or as the first data query described in Figure 4). When the server selection circuitry 110 determines that the response time for the responsive server to service the data query exceeds a response latency threshold (502), the server selection circuitry 110 may perform, in response, a responsiveness verification process to determine whether the responsive server has become unresponsive. The responsiveness

verification process may occur during normal operation of the server selection circuitry 110 in selecting data servers to query data to.

[0038] As part of a responsiveness verification process, the server selection circuitry 110 may send a subsequent data query for servicing by the responsive server (506). The subsequent data query may be sent as a third data query and at, for example, a subsequent time when the responsive server is selected by the server selection circuitry 110 for servicing a data query. When the response time for servicing the subsequent data query exceeds the response latency threshold, the server selection circuitry 110 may determine that the responsive server has become unresponsive (510). The server selection circuitry 110 may also identify that the responsive server as an unresponsive server instead (512), for example by updating the server response list 104 or otherwise categorizing the responsive server instead as unresponsive.

[0039] When the response time for servicing the subsequent data query does not exceed the response latency threshold, the server selection circuitry 110 may determine that the responsive server has not become responsive (514) and continue to select the responsive server for handling data queries to the shared common memory. The server selection circuitry 110 may determine not to send subsequent data queries to an unresponsive server (516), such as the previously responsive server now identified as unresponsive.

[0040] While the logic 500 shown in Figure 5 provides one example, the server selection circuitry 110 may employ any number or combination of responsiveness verification criteria to determine whether a previously identified responsive server has become unresponsive. As one illustrative example, a responsiveness verification criterion may be met when a threshold number of data queries serviced by the previously identified responsive server have a response time that exceeds the response latency threshold, e.g., when 3 or more data queries serviced by previously identified responsive server have response times that exceed the response latency threshold.

[0041] Additional examples of responsiveness verification criteria for the previously identified responsive server include: (i) when a threshold number of serviced data queries have a response time exceeding the response latency

threshold over a set or predetermined period of time, (ii) when a threshold percentage of the serviced data queries have a response time exceeding the response latency threshold (e.g., as measured over a predetermined number of previous data queries or over a predetermined period of time); and (iii) when a serviced data query exceeds the response latency threshold by more than a threshold time amount. The server selection circuitry 110 may employ any combination of responsiveness verification criteria to determine a previously-identified responsive server as unresponsive. In some examples, the server selection circuitry 110 applies the responsiveness verification criteria through normal operation of server selection and data querying, e.g., by performing responsiveness verification as a background process in the selection phase. In some examples, the server selection circuitry 110 employs a dedicated responsiveness verification, e.g., directly in response to or specifically upon detecting that a response time for the previously identified responsive server has exceeded a response latency threshold.

[0042] In some examples, the server selection circuitry 110 performs a probe phase as part of the responsiveness verification process. Thus, the server selection circuitry 110 may collect response data for some or all of the multiple data servers upon detecting that a particular data server has become unresponsive, including collecting response data for responsive data servers. In doing so, the server selection circuitry 110 may detect whether multiple responsive servers have become unresponsive, e.g., when a newly executed application, job, or task consumes computation resources or bandwidth of multiple servers, compute nodes, or other hardware.

[0043] Once categorized as unresponsive, the server selection circuitry 110 may monitor an unresponsive server to determine the unresponsive server has become responsive (518). For example, the server selection circuitry 110 may periodically probe unresponsive data servers to determine whether response data for the unresponsive data servers indicates the servers have become responsive. The server selection circuitry 110 may clone a data query to send to both a selected responsive server (e.g., for servicing the data query) and an unresponsive server (e.g., to probe the unresponsive server). The server

selection circuitry 110 track the response time for the unresponsive server to service the data query, and deem the unresponsive server as responsive instead when the response time does not exceed a response latency threshold. The server selection circuitry 110 may apply various criteria for determining whether an unresponsive server has become responsive, e.g., any of the responsiveness verification criteria discussed above. Upon determining that an unresponsive server has become responsive, the server selection circuitry 110 may update the server response list 104 and select the now responsive server for servicing subsequent data queries.

[0044] Figure 6 shows an example of a computing device 600 that supports responsive server identification among multiple data servers linked to a shared memory. In that regard, the computing device 600 may implement any of the functionality described herein, including according to any of the features described herein for the server selection circuitry 110.

[0045] The computing device 600 may include a processor 610. The processor 610 may include a central processing unit (CPU), microprocessor, and/or any hardware device suitable for executing instructions stored on a machine-readable medium. The computing device 600 may include a machine-readable medium 620. The machine-readable medium 620 may be any electronic, magnetic, optical, or other physical storage device that stores executable instructions, such as the server selection instructions 622 shown in Figure 6. Thus, the machine-readable medium 620 may be, for example, Random Access Memory (RAM), an Electrically-Erasable Programmable Read-Only Memory (EEPROM), a storage drive, an optical disk, and the like.

[0046] The computing device 600 may execute instructions stored on the machine-readable medium 620 through the processor 610. Executing the instructions may cause the computing device 600 to perform any of the features described herein. One specific example is shown in Figure 6 through the server selection instructions 622. Executing the server selection instructions 622 may cause the computing device 600 to operate according to any combination of features of the server selection circuitry 110.

[0047] As one example, executing the server selection instructions 622 may cause the computing device 600 to populate a server response list 104 in a probe phase, which may include sending a first data query to multiple data servers, the multiple data servers linked to a shared memory accessible to the multiple data servers and recording a response time for the multiple data servers to service the first data query. Executing the server selection instructions 622 may further cause the computing device 600 to identify a selected data server from the server response list 104 for servicing a second data query in a selection phase, which may include defining a response latency threshold according to a quality of service requirement specified by an application; identifying, from the server response list 104, responsive servers with a response time that does not exceed a response latency threshold; and determining the selected data server for servicing the second data query from among the responsive servers.

[0048] As additional examples, executing the server selection instructions 622 may cause the computing device 600 to monitor subsequent response times for the responsive servers in servicing data queries to determine whether a responsive server among the responsive servers has become unresponsive. Determination that a responsive server has become unresponsive may include identifying when a threshold number of the response times for the responsive server exceed the response latency threshold.

[0049] The methods, devices, systems, and logic described above, including the server selection circuitry 110, may be implemented in many different ways in many different combinations of hardware, software or both hardware and software. For example, server selection circuitry 110 may include circuitry in a controller, a microprocessor, or an application specific integrated circuit (ASIC), or may be implemented with discrete logic or components, or a combination of other types of analog or digital circuitry, combined on a single integrated circuit or distributed among multiple integrated circuits. A product, such as a computer program product, may include a storage medium and machine readable instructions stored on the medium, which when executed in an endpoint,

computer system, or other device, cause the device to perform operations according to any of the description above.

[0050] The processing capability of the systems, devices, and circuitry described herein, including the server selection circuitry 110, may be distributed among multiple system components, such as among multiple processors and memories, optionally including multiple distributed processing systems. Parameters, databases, and other data structures may be separately stored and managed, may be incorporated into a single memory or database, may be logically and physically organized in many different ways, and may implemented in many ways, including data structures such as linked lists, hash tables, or implicit storage mechanisms. Programs may be parts (e.g., subroutines) of a single program, separate programs, distributed across several memories and processors, or implemented in many different ways, such as in a library, such as a shared library (e.g., a dynamic link library (DLL)). The DLL, for example, may store code that performs any of the system processing described above.

[0051] While various examples have been described above, many more implementations are possible.

CLAIMS

1. A method comprising:
through server selection circuitry of a device:
 sending a first data query to multiple data servers, the multiple data servers linked to a shared memory storing data requested by the first data query;
 tracking response times for the multiple data servers, the response times comprising a particular response time for each of the multiple data servers to service the first data query;
 identifying a responsive server from among the multiple data servers according to the response times; and
 sending a second data query to the responsive server.
2. The method of claim 1, wherein sending the first data query to the multiple data servers comprises cloning the first data query to send an instance of the first data query to each of the multiple data servers.
3. The method of claim 1, comprising concurrently sending the first data query to the multiple data servers.
4. The method of claim 1, wherein identifying the responsive server comprises determining that the particular response time for the responsive server in servicing the first data query does not exceed a response latency threshold.
5. The method of claim 1, further comprising:
 tracking response times for the multiple data servers in servicing multiple data query types; and
 identifying a responsive server for servicing a particular data query type according to the response times for the multiple data servers in servicing the particular data query type.

6. The method of claim 1, further comprising:
 - determining that a response time for the responsive server in servicing the second data query exceeds a response latency threshold, and in response:
 - performing a responsiveness verification process to determine whether the responsive server has become unresponsive.
7. The method of claim 6, wherein performing the responsiveness verification process comprises:
 - sending a third data query to the responsive server, and when a response time for the responsive server in servicing the third data query also exceeds the response latency threshold:
 - determining that the responsive server has become unresponsive;
 - identifying the responsive server as an unresponsive server instead; and
 - further comprising:
 - determining not to send subsequent data queries to the unresponsive server.
8. The method of claim 6, wherein performing the responsiveness verification process comprises:
 - sending multiple subsequent data queries to the responsive server, and when a threshold number of the multiple subsequent data queries have a response time that exceeds the response latency threshold:
 - determining that the responsive server has become unresponsive;
 - and
 - identifying the responsive server as an unresponsive server instead.
9. A device comprising:
 - a memory to store a server response list, the server response list including a listing of multiple data servers linked to a shared memory; and
 - server selection circuitry to:

maintain the server response list by determining response times for the multiple data servers to service a data query, including concurrently sending the data query to the multiple data servers to track a response time for each of the multiple servers to service the data query;

identify responsive servers among the multiple data servers listed in the server response list according to the response times to service the data query; and

select one of the responsive servers to send a subsequent data query.

10. The device of claim 9, wherein the server selection circuitry is to maintain the server response list further to:

track response times for the multiple data servers in servicing multiple data query types; and

determine a responsive server for servicing a data query of a particular data query type according to the response times for the multiple data servers in servicing the particular data query type.

11. The device of claim 9, wherein the server selection circuitry is to identify the responsive servers by determining which of the multiple servers have a response time in servicing the data query that does not exceed a response latency threshold.

12. The device of claim 11, wherein the server selection circuitry is further to obtain the response latency threshold according to an application quality of service requirement.

13. A device comprising:

a non-transitory machine readable medium storing executable instructions to:

populate a server response list in a probe phase, including:

sending a first data query to multiple data servers, the multiple data servers linked to a shared memory accessible to the multiple data servers;

recording a response time for the multiple data servers to service the first data query; and

identify a selected data server from the server response list for servicing a second data query in a selection phase, including:

defining a response latency threshold according to a quality of service requirement specified by a client application;

identifying, from the server response list, responsive servers with a response time that does not exceed a response latency threshold; and

determining the selected data server for servicing the second data query from among the responsive servers.

14. The device of claim 13, wherein the executable instructions are further to monitor subsequent response times for the responsive servers in servicing data queries to determine whether a responsive server among the responsive servers has become unresponsive.

15. The device of claim 14, wherein the executable instructions are to determine that a responsive server has become unresponsive when a threshold number of the response times for the responsive server exceed the response latency threshold.

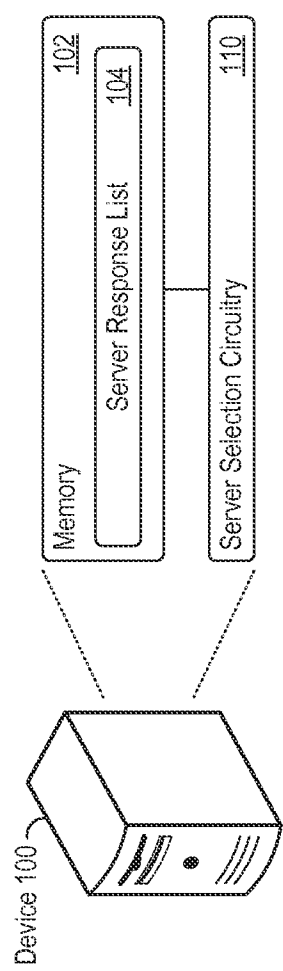


Figure 1

200 →

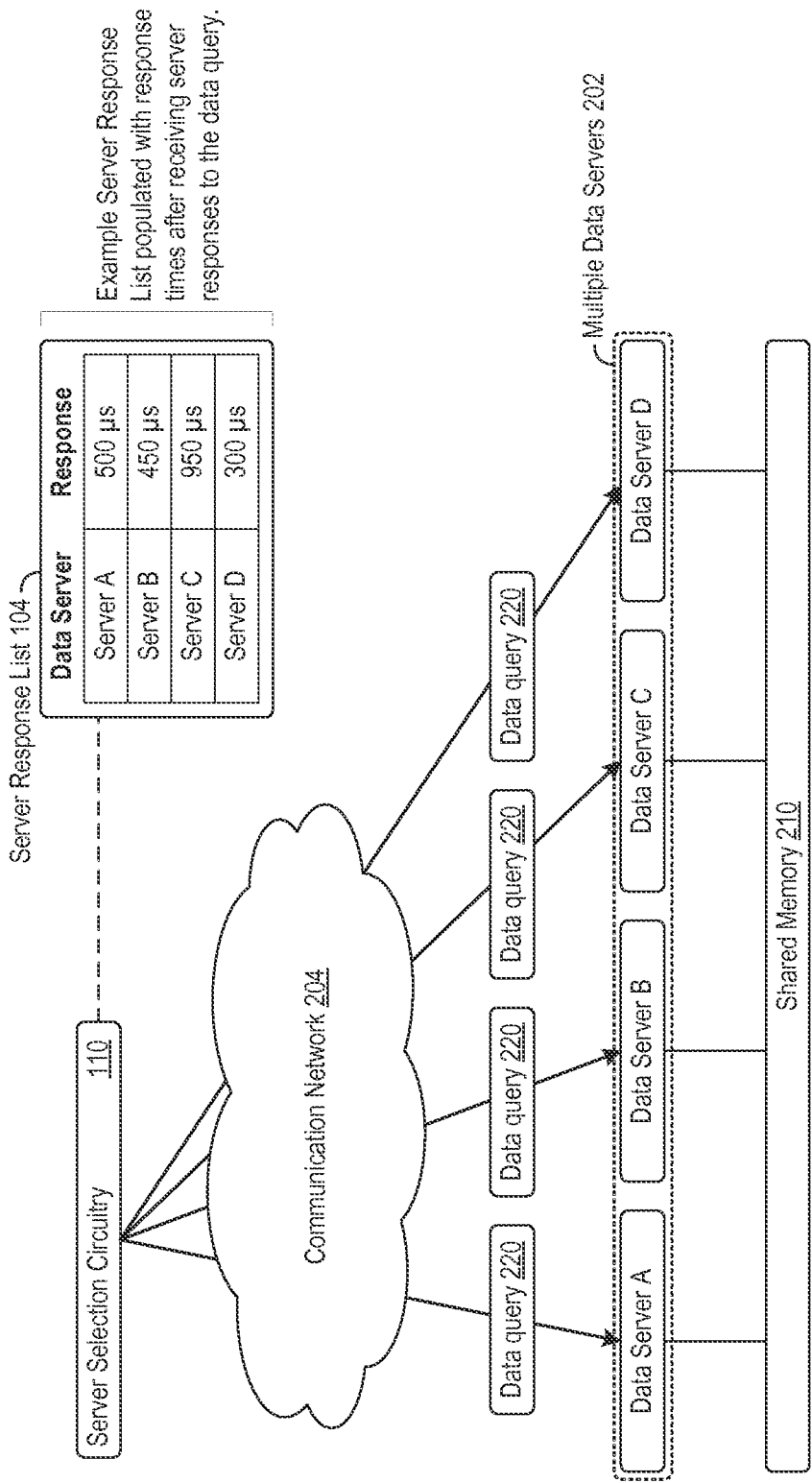


Figure 2

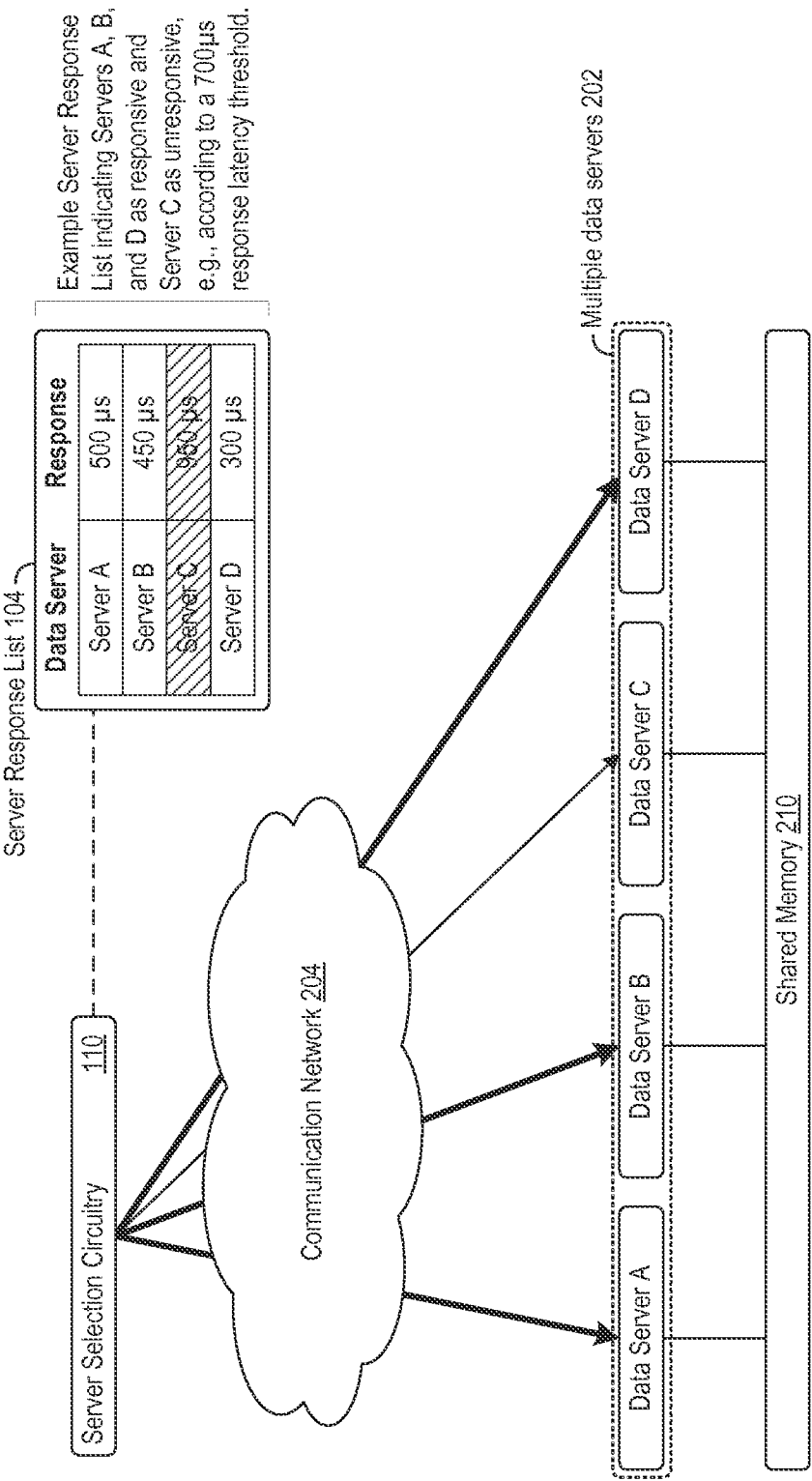


Figure 3

4 / 6

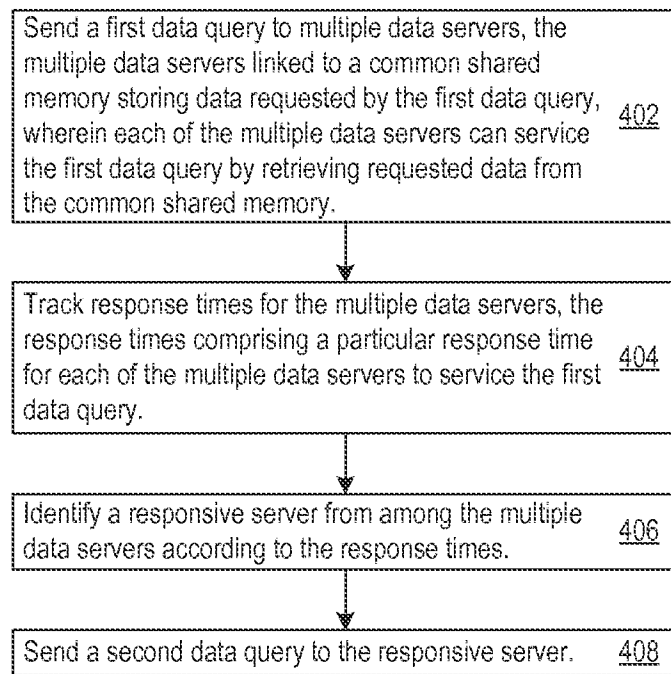
400
→

Figure 4

5 / 6

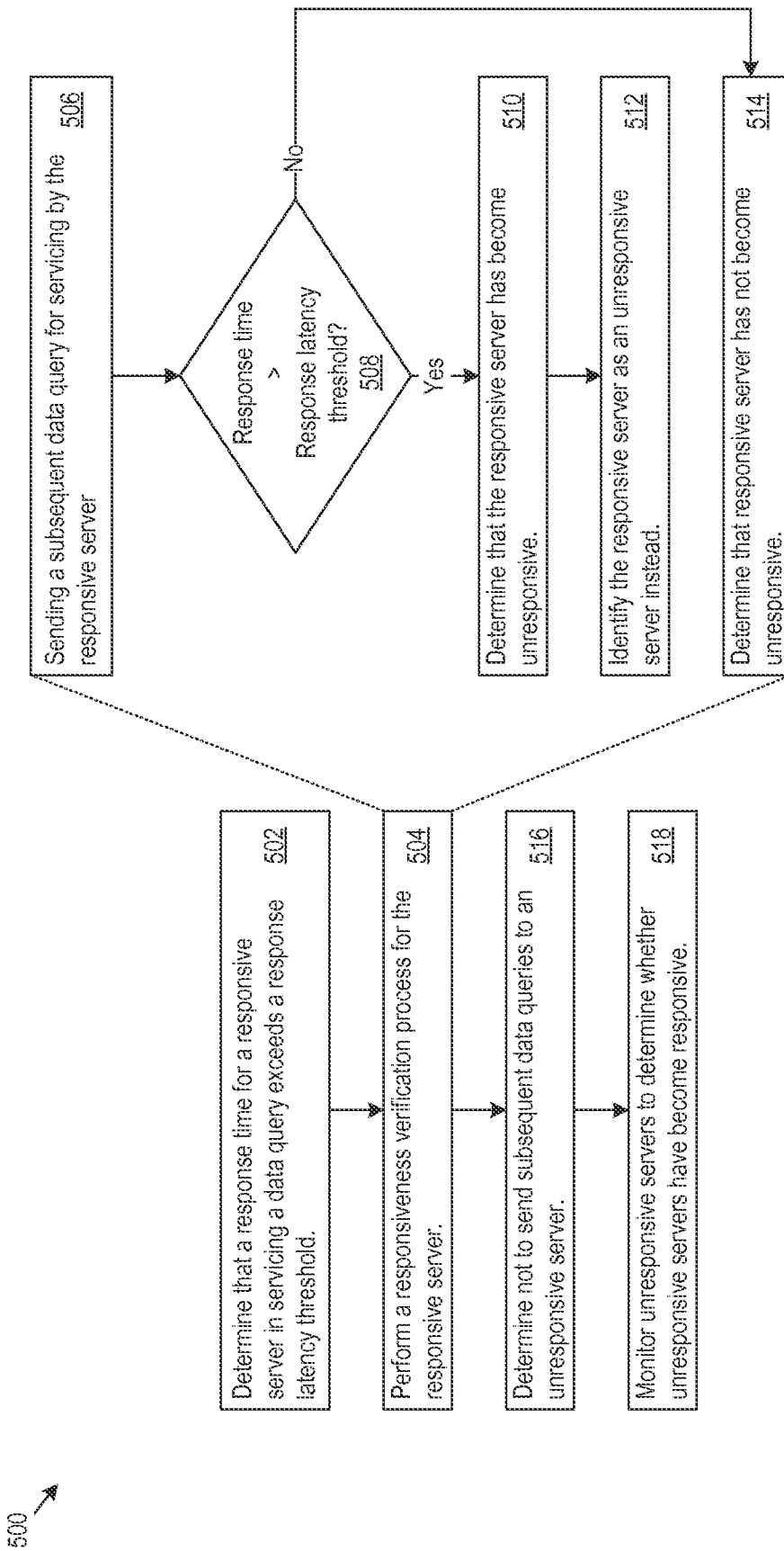


Figure 5

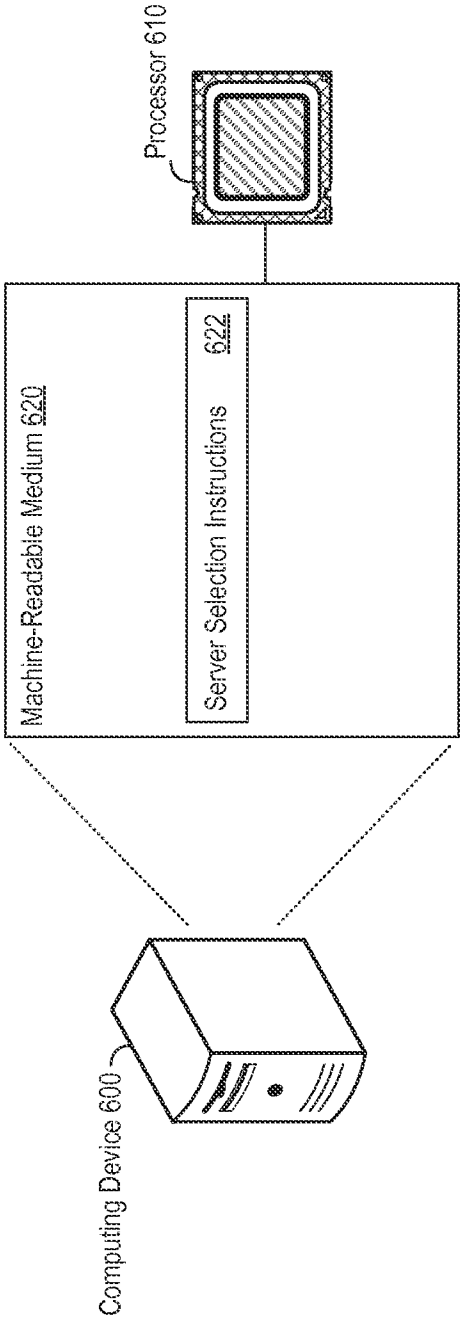


Figure 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/018019**A. CLASSIFICATION OF SUBJECT MATTER****G06F 15/16(2006.01)i, G06F 15/167(2006.01)j**

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

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06F 15/16Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords:server, selection, table, list, response time**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 8037197 B2 (KAMIYAI MASANORI et al.) 11 October 2011 See abstract; Col.5 line 10 - Col.8 line 49; figures 3-5.	1,4,5,13
A		2,3,6-12,14,15
Y	US 7937473 B2 (NAKADAI SHINJI) 03 May 2011 See abstract; Col.5 line 10 - Col.8 line 49; figures 1,7,8.	1,4,5,13
A		2,3,6-12,14,15
A	US 2009-0276520 A1 (WEERAKOON ISHAN P. et al.) 05 November 2009 See abstract; paragraphs [0059]-[0075]; figures 3-5.	1-15
A	US 8244873 B2 (BASKARAN RAMESH et al.) 14 August 2012 See abstract; Col.4 line 28 - Col.5 line 15; figure 1.	1-15

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

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
06 July 2015 (06.07.2015)Date of mailing of the international search report
07 July 2015 (07.07.2015)Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea
Facsimile No. +82-42-472-7140Authorized officer
KYUNG, Youn Jeong

Telephone No. +82-42-481-3452



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/018019

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8037197 B2	11/10/2011	US 2009-113447 A1	30/04/2009
US 7937473 B2	03/05/2011	JP 5050854 B2	17/10/2012
		US 2009-0157870 A1	18/06/2009
		WO 2007-034826 A1	29/03/2007
US 2009-0276520 A1	05/11/2009	None	
US 8244873 B2	14/08/2012	US 2010-0005173 A1	07/01/2010
		US 2011-0161470 A1	30/06/2011
		US 7970905 B2	28/06/2011