

APPLICATION UNDER UNITED STATES PATENT LAWS

Atty. Dkt. No. PW 0283007
(M#)

Invention: CONFIGURABLE ADAPTIVE GLOBAL TRAFFIC CONTROL AND MANAGEMENT

Inventor (s): P.E. Stolorz, J.K. Salmon, M.S. Warren, J. Koller, A. Hagberg, M. Yevmenkin, M. Brady, and D. Pfitzner.

Pillsbury Winthrop LLP
Intellectual Property Group
1600 Tysons BoulevardMcLean, VA
22102
Tel: (703) 905-2000
Attorneys
Telephone:

This is a:

- ☒ Provisional Application
- ☐ Regular Utility Application
- ☐ Continuing Application
 - ☒ The contents of the parent are incorporated by reference
- ☐ PCT National Phase Application
- ☐ Design Application
- ☐ Reissue Application
- ☐ Plant Application
- ☐ Substitute Specification
 - Sub. Spec Filed _____
 - in App. No. _____ / _____
- ☐ Marked up Specification re
 - Sub. Spec. filed _____
 - In App. No _____ / _____

SPECIFICATION

CONFIGURABLE ADAPTIVE GLOBAL TRAFFIC CONTROL AND MANAGEMENT

1. Reservation of Copyright

[0001] This patent document contains information subject to copyright protection.

The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent, as it appears in the U.S. Patent and Trademark Office files or records but otherwise reserves all copyright rights whatsoever.

2. Field of the Invention

[0002] Aspects of the present invention relate to network management. Other aspects of the present invention relate to configurable, adaptive, global traffic control and management in networks such as the Internet.

BACKGROUND & SUMMARY

[0003] As the volume of Internet traffic grows, providers of web content and applications increasingly need to deliver content from multiple servers at widely-separated locations in order to sustain a good end-user experience under high traffic loads. This need generates several difficult challenges, including:

- how to guarantee the fault-tolerance of such a multiple-server system in the face of failure of one or more individual servers;
- how to control the way in which requests from end-users are distributed to each server according to important content provider policy constraints imposed for economic, contractual or other reasons; and
- how to guarantee high performance as experienced by end-users as network conditions change.

[0004] This invention solves these and other problems by providing a means to distribute network (e.g., Internet) traffic according to a configurable set of rules. The rules can be configured to take into account key factors such as:

- server availability.
- state of the network (Internet) at any given moment, including measures of network latency.
- specific requirements of content providers who deploy the invention, e.g. distribution based upon geography, position in IP address space, load share, etc.

[0005] These rules together provide an extremely fine-grained level of Internet traffic control to providers of internet content and applications, enabling them to dramatically improve the end-user experience (measured by speed of request resolution and associated download time) over that provided by conventional web servers and mirrored server farms.

[0006] There are many potential uses for the invention. One use is to provide a stand-alone service directing traffic exclusively to a set of designated servers managed by a single organization. The invention may also be used in more general ways – for example, one or more of the designated destinations can refer to servers (or server collections) outside the organization's control. The latter case includes, for example, Content Delivery Networks (CDN's), as well as local load-balancing servers, as potential destinations. The invention can also be used to provide the DNS component of a Content Delivery Network itself. It can be deployed as a service on behalf of subscribers, or it can be deployed as software to be used directly by subscribers themselves.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The present invention is further described in terms of exemplary embodiments, which will be described in detail with reference to the drawings. These embodiments are non-limiting exemplary embodiments, in which like reference numerals represent similar parts throughout the several views of the drawings, and wherein:

[0008] FIG. 1 depicts a framework according to embodiments of the present invention;

[0009] FIG. 2 describes exemplary types of subscriber policies, according to embodiments of the present invention;

[0010] FIG. 3 shows an exemplary subscriber policy hierarchy or a decision tree built based on various subscriber policies, according to embodiments of the present invention;

[0011] FIG. 4 shows an exemplary relationship between a subscriber's network hierarchy and subscriber policies, according to embodiments of the present invention;

[0012] FIG. 5 depicts a high level architecture of an adaptive traffic control framework according to embodiments of the present invention;

[0013] FIG. 6 depicts a high level functional block diagram of an administrative master agent (AMA) according to embodiments of the present invention;

[0014] FIG. 7 depicts the internal functional block diagram of a monitoring mechanism, according to embodiments of the present invention;

[0015] FIG. 8 is an exemplary flowchart of a process, in which a name server agent serves a DNS request based on subscriber policies, according to embodiments of the present invention;

[0016] FIGS. 9(a) – 9(d), 10(a) - 10(k) illustrate various exemplary GUI interfaces according to embodiments of the present invention;

60325477-092801

[0017] FIG. 11(a) shows an exemplary secure web based graphical interface, through which a subscriber may define load share and shed fraction policies among static resources, according to embodiments of the present invention;

[0018] FIG. 11(b) shows an exemplary interface for defining policies for dynamic servers, according to embodiments of the present invention;

[0019] FIG. 11(c) shows an exemplary graphical user interface through which an overflow server may be defined using a canonical name, according to embodiments of the present invention;

[0020] FIGS. 12(a) – 12(b) show example subscriber policies, defined based on a set of resource servers, that govern the selection according to geographical location of an incoming request and the time zone of each of the locations, according to embodiments of the present invention;

[0021] FIGS. 12(c) – 12(d) show example subscriber policies, defined based on geographical locations of an incoming request with overflow policies that allow the ATC mechanism 150 to direct traffic to pre-defined services when primary servers are not available, according to embodiments of the present invention;

[0022] FIGS. 13(a) – 13(c) show exemplary archived log information that can be displayed and viewed through a graphical user interface, according to embodiments of the present invention;

[0023] FIG. 14 is an exemplary flowchart of a process, in which an administrative master agent manages subscriber policies and monitors associated name server agents, according to embodiments of the present invention; and

[0024] FIG. 15 is an exemplary flowchart of a process, in which a monitoring mechanism in an administrative master agent monitors the performance of name server agents, according to embodiments of the present invention.

DETAILED DESCRIPTION

[0025] The invention is described below, with reference to detailed illustrative embodiments. It will be apparent that the invention can be embodied in a wide variety of forms, some of which may be quite different from those of the disclosed embodiments. Consequently, the specific structural and functional details disclosed herein are merely representative and do not limit the scope of the invention.

[0026] Although there are a number of different scenarios in which the invention might be deployed, this description will focus, for clarity, on a scenario in which DNS service is provided by a third party on behalf of a content or applications provider.

[0027] FIG. 1 depicts a high-level of a framework or system **100** according to embodiments of the present invention. A subscriber **102** provides content via a subscriber server network **104** which is made up of one or more server groups **106-1, . . . , 106-k** (collectively **106**).

[0028] As used in this description, the framework according to embodiments of this invention contemplates three groups of users:

- An **end-user** who wishes to access content and/or applications over the Internet
- An **operator** who offers and supports the service on a managed network of computers.
- A **subscriber** (typically a content or applications provider) who subscribes to the operator's service to help support the needs of the end user.

[0029] The terminology end-user, operator and subscriber is used throughout this description to distinguish these three roles, although there are many scenarios in which more than one role can be taken by a single entity. Such scenarios are contemplated by this invention.

[0030] Each server group 106 includes one or more primary servers and possibly one or more overflow servers. In general, for the *j*-th server group, the primary servers of server group 106-*j* are denoted 108-*j*, and the overflow servers of server group 106-*j* are denoted 110-*j*.

[0031] Each server in a server group is any process or collection of processes that provide resources in response to requests, e.g., from a client. A server can be any off-the-shelf Web server. In some embodiments, servers are typically a Web server such as the Apache server or Netscape Communications Corporation's Enterprise™ server.

[0032] Client 112 accesses the subscriber server network 104 in order to obtain content from the subscriber. Content includes any kind of data, including, without limitation, video and audio data and the like. To achieve this access, a user at client 112 enters a resource locator, e.g., a Universal Resource Locator ("URL"), into a browser 114 on client 112. URLs specify the location of resources (information, data files, etc.) on the network. URLs are defined in detail in T. Berners-Lee et al, Uniform Resource Locators (URL), Network Working Group, Request for Comments: 1738, Category: Standards Track, December 1994, located at "<http://ds.internic.net/rfc/rfc1738.txt>", which is hereby incorporated herein by reference. URLs generally have the following form:

scheme://*host*[:*port*]/*url-path*

[0033] , where "*scheme*" can be a symbol such as "*file*" (for a file on the local system), "*ftp*" (for a file on an anonymous FTP file server), "*http*" (for a file on a file on a Web server), and "*telnet*" (for a connection to a Telnet-based service). Other schemes, can also be used and new schemes are added every now and then. The port number is optional, the system substituting a default port number (depending on the scheme) if none is provided. The "*host*" field (the hostname) maps to one or more particular network addresses for particular computer(s). The "*url-path*" is relative to the computer specified

in the “host” field. A *url-path* is typically, but not necessarily, the pathname of a file in a web server directory.

[0034] The framework or system **100** includes at least one Domain Name Service (DNS) name server **118-1**. In preferred embodiments, the system **100** includes DNS name servers **118-1**, **118-2**, . . . , **118-n**, (collectively referred to as name servers **118**) all operated by a single particular entity. In the embodiment shown in **FIGURE 1**, one of the name servers **118-1** serves the U.S.A., another name server **118-2** serves the U.K., another name server **118-3** serves Japan, and so on. The various name servers **118-1**, **118-2**, . . . , **118-n** can be configured to serve geographical regions, geopolitical regions, corporate structures or any other logical organizational structure. The structure and operation of each name server **118** is described below.

[0035] When the client’s browser **114** obtains a request (in the form of a URL), the browser queries its resolver **116** for the hostname specified in the requested URL. The resolver **116** eventually queries a particular name server **118-1**. The name server **118-1** returns the IP (Internet Protocol) address (or addresses) of a server (or servers) in the subscriber server network. The particular IP address returned to the resolver **116** is based on a number of factors, including the resolver’s location (e.g., as determined from the resolver’s IP address) and various policies (e.g., subscriber policies **120**, other policies **122**) in a policy database **124**. The client’s browser **114** is then able to communicate with the selected server in the subscriber server network in order to obtain the desired resource.

[0036] The policies **120**, **122** may be defined by subscribers of the service or by other entities, or both, according to embodiments of the present invention. The policies in the policy database **124**, described in greater detail below, may include the following policy categories:

- **Geographic Policy:** Decisions are based on the IP address of an end-user's resolver. For example, if the IP address is within the U.S., the request may be directed to a "domestic" group of servers in the subscriber server network 104, otherwise it may be directed to "international" servers in the network. The request may be directed to a proprietary Content Delivery Network (CDN) or to another service specified by the subscriber. Any combination of IP address location and request direction is possible.
- **Load Share Policy:** The subscriber can explicitly specify the amount of traffic (load share) to be directed to each of their servers within a defined server set in the subscriber server network 104. Typically the percentages will be chosen according to the capacity at each server.
- **Overflow (Load Shed) Policy:** In some circumstances, the "load" on one or more subscriber servers may exceed a prescribed level. In these cases, a subscriber-specified fraction of traffic (shed fraction) that would otherwise be directed to a given server may be shed to an overflow service outside the scope of the standard capacity policy.
- **CIDR Policy:** Policy decisions are supported based upon CIDR blocks of IP address space. (CIDR denotes Classless Inter-Domain Routing, an IP addressing scheme that replaces the system based on classes A, B, and C. With CIDR, a single IP address can be used to designate many unique IP addresses.)
- **Static versus Dynamic Policy:** A policy may be static or dynamic.
- **Network Aware Policy:** Decisions may be based upon an adaptive, regularly updated map of the state of the Internet. The map can cluster IP addresses together according to their network latency to a selected set of

network agents. This enables subscriber servers to be selected according to their "network proximity" to an end-user's browser, optimizing resulting download time.

[0037] Name servers **118-1, 118-2, . . . , 118-n**, according to embodiments of the present invention, include an adaptive traffic control (ATC) mechanism **126** which provides domain name service to a client based on policies in the policy database **124**. Each name server **118** includes or has access to a location determination mechanism **128** for associating the client's request with the client's location. Such a mechanism **128** may be, e.g., the TraceWare™ product of Digital Island, Inc. TraceWare™ is suite of services which provide an Internet Atlas which gives real-time, geographic intelligence through its ability to recognize the originating country, region and metropolitan area of Internet consumers.

[0038] The subscriber server network **104** represents a network of servers that provides, on behalf of an underlying subscriber, Internet content or services. For example, a subscriber may be the Cable News Network (CNN), which has its own network of servers that deliver news content to end users' browsers via the Internet. The subscriber server network **104** may be configured in such a way that the processing of the service requests may be reasonably distributed among all the servers in the server network **104**, according to some criteria. To achieve that distribution, the subscriber server network **104** may be organized, for example, to have one or more server groups (e.g., server group **106-1, . . . , server group k 106-k**), each of which may be responsible for processing a portion of the service requests. For example, service requests coming from Finland may be routed to a server group that is physically located in Europe.

[0039] Each server group **106** in the subscriber server network **104** may comprise a plurality of servers to further share the load. For example, a service request from

Finland may be subsequently routed to a server located in Scandinavia. Routing service requests to different server groups and subsequently to different servers may be based on various criteria. For example, such routing may be based on the distance between the origin of the request and the location of the server. For example, for a service request originated from Finland, it may be more effective and efficient to direct the request to a server located in Norway instead of directing it to a server located in the U.S. Server load may also be used to determine where to route a service request. For example, a service request originated from the U.S. may be routed to a server group in Mexico if the server group in the U.S. is overloaded while the server group in Mexico is relatively idle.

[0040] Servers in a server group **106** may be classified into different categories. Such classification may be based on the specific role of the underlying server. For example, servers in a server group **106** may be classified into a primary server group and an overflow server group, where a overflow server may be called upon to provide services when either a primary server is overloaded or not functioning.

[0041] A primary server group may further include one or more servers. When multiple servers are present in a primary server group, servers in a same primary server group may be configured in such a way that they share the load. For example, if there are five servers in a primary server group, they may be configured so that each takes 20% of the total service requests routed to the primary server group. For each of such servers, it may be further configured so that when a particular server is overloaded, it may shed a portion of its load to a particular overflow server group.

[0042] The distribution of service requests (from a client **112**) within the subscriber server network **104** may be controlled through a set of ATC policies (**120, 122**) stored in the policy database **124**. Various involved authorities may influence the ATC policies and may create policies in the policy database **124**. For example, the subscriber

may set up policies to direct traffic with respect to considerations such as the geographical locations and the capacities of the underlying servers in the subscriber's network 104.

Other policy entities 130, including, for example, the operator of the name servers 118 or various geo-political entities may also have policies regarding how the network traffic should be managed and directed. For example, the government or some organizations may regulate some aspects of network traffic policies. Such regulation policies may be required to be incorporated so that a service request from the client 112 can be routed in a manner that satisfies both regulatory and subscriber policies.

[0043] Thus, ATC policies in the policy database 124 may be a combination of subscriber policies 120 and other policies 122 from different sources such as subscriber 102 and policy entity 130. Policies from different sources may be defined and accessed by the ATC mechanism 126 in a similar fashion. From the point of view of the ATC mechanism 126, the source of a policy may not be relevant or even determinable. For example, policies may be defined in a textual file which may be downloaded to the ATC mechanism 126 via either a graphical user interface (GUI), a file transfer protocol (FTP), or some other mechanism. Policy makers may also enter policies directly via a web-based GUI interface 132 provided by the ATC mechanism 126. Preferably the interface 132 is secure (e.g., implemented using the "https" protocol).

[0044] The policies from the policy database 124 used by the ATC mechanism are collectively referred to as policies 120 or ATC policies 120.

[0045] The ATC policies may be organized in a way that is appropriate to govern and / or control the traffic at different levels of the subscriber server network 104. FIG. 2 shows exemplary types of subscriber policies 120 according to embodiments of the present invention. There may be different types of ATC policies which may include geo-political policies, load share policies, and overflow policies. Geo-political policies may

govern the selection of a server according to where the client is located. Load share policies may govern the selection according to the capacities of the servers. Overflow policies may govern the selection process when the primary servers in a server group are overloaded. As shown in FIG. 2, the geo-political policies may further classified into continental based distribution policies, country based distribution policies, region based distribution policies and time zone based distribution policies. These policies may be set up to govern the process of selecting a particular server group according to different geographical or political criteria. On the other hand, the load share and overflow policies may be designed to govern the process of selecting one or more servers within a particular server group according to the dynamic server load and performance information.

[0046] The ATC policies may be hierarchically constructed to form a decision tree, e.g., as shown in FIG. 3 which depicts an exemplary subscriber policy hierarchy or a decision tree built based on various subscriber policies, according to an embodiment of the present invention. In the example of FIG. 3, the continental based distribution policies are used to first direct a selection process limited to a particular continent. For example, if the requesting client 112 is located in Asia, the continental based distribution policies may direct the further selection within servers that are located in Asia. Similarly, if the client 112 is located in either North America or Europe, the further selection process may be directed to the appropriate servers located in the corresponding continent.

[0047] Once a DNS request is directed to an appropriate continent, the country based distribution policies may further constrain the selection to particular servers that are in the same or close-by countries where the client 112 is located. For example, if the client 112 is located in the U.S., the country based policies may direct the selection from the servers located in the U.S. Similarly, region based distribution policies may further

constrain the selection to, for example, the west coast or east coast depending on where the client 112 is located.

[0048] When a particular server group is selected (e.g., after a hierarchical decisions based on continent-based policies to region-based policies), the load share policies govern the process whereby servers in a given server group should be selected. Such policies may be determined based on the servers' capacities or may be adaptively revised based on the dynamic performance or load of the underlying servers. Load share policies may specify how much percentage (share) of the total requests that each primary server should handle. For example, if a server group comprises a total of three primary servers (server 1, server 2, server 3), a load share policy for this server group may specify the load share as (0.3, 0.5, 0.2), indicating that server 1 should take 30% of the total load, server 2 should take 50% of the load, and server 3 should take 20% of the total load.

[0049] The overflow policies govern the selection when a primary server in a server group is either overloaded or not available. Such situations may be detected according to the response time of the server. For example, if the response time becomes long, it may indicate that the server is overloaded. If the server is simply not responding, the server may be not available. In this case, a dynamic policy can be used to shed traffic away from a given server. The selection process may examine the overflow policies to see whether an overflow (alternative) server or servers are designated and then the selection will be directed to the overflow servers.

[0050] FIGURE 4 depicts an exemplary relationship between the subscriber server network 104 and subscriber policies 120, according to embodiments of the present invention. The subscriber server network 104 is depicted in FIG. 4 as a network hierarchy or a tree in which the subscriber server network 104 includes server group 1 106-1 through server group k 106- k , and each server group may include its own primary server group 108

and overflow server group **110** (e.g., server group **1** **106-1** has primary server group **108-1** and overflow server group **110-1** associated therewith, while server group k **106- k** has primary server group **108- k** and overflow server group **110- k** associated therewith).

[0051] The ATC policies **120** govern selection of one or more servers within the subscriber server network **104** (or in some other network of servers) so that their IP addresses may be returned in response to DNS requests from the client **112** (resolver **116**). To determine or to select appropriate servers in the subscriber server hierarchy, the decisions may also be hierarchical. For example, as shown in **FIGURE 4**, at the level of the subscriber server network **104**, the ATC mechanism **126** may make geo-political distribution decisions to determine which server group should be selected. Once a particular server group is selected, the ATC mechanism **126** may further determine which particular servers in the group should be selected according to, for example, how the traffic load should be shared among the servers within a particular server group. In this case, the decisions at server group level may be based on load share policies. That is, at each branch node in the server hierarchy, appropriate ATC policies may be applied to govern the selection of an appropriate server or servers.

[0052] For example, as illustrated in **FIG. 4**, the subscriber server network **104** is related to the ATC policies **120**. For example, geo-political policies may be applied to control the selection of a particular server group, load share policies may be applied to select the servers within a particular server group, and overflow policies may be applied to determine when overflow servers, instead of primary servers, within a particular server group should be selected. That is, the subscriber server network **104** forms a tree and the ATC policies govern the flow from the top of the tree to one or more leaves of the tree. The ATC mechanism **126** controls the flow by applying the ATC policies **120**.

[0053] FIG. 5 depicts a high level architecture of an adaptive traffic control (ATC) framework according to embodiments of the present invention. The ATC framework comprises an ATC administrative framework 132 and an ATC name server network 134. The ATC administrative framework 132 is responsible for various administrative tasks associated with subscribers, include manipulating and storing ATC policies, propagating ATC policies to name servers in the ATC name server network 134, monitoring name server behavior, generating status reports, and displaying monitoring results on appropriate media.

[0054] The ATC name server network 134 is responsible for responding to DNS requests, including processing DNS requests, applying ATC policies to select one or more servers from the subscriber server network (or another designated network), and replying with the IP address(es) of the selected server(s). The ATC name server network 134 dynamically receives updated policy information from the ATC administrative framework 132, providing domain name-to-IP address resolutions based on such updated ATC policies, and supplying logging and monitoring data to the ATC administrative framework 132.

[0055] The ATC administrative framework 132 may be designed to have fault-tolerance. For example, as depicted in FIGS. 1 and 5, the ATC administrative framework 132 may include an administrative master agent (AMA) 136, and one or more AMA backups 138-1, . . . , 138-*m* (collectively 138). All AMA agents may be capable of performing the exact functions except that one of them (e.g., master AMA 136) may be a central or primary administrative master agent and others (e.g., AMA backups 138) may be backup or secondary or redundant master agents. The central or primary AMA 136 may be responsible for regularly backing up the AMA backup agents 138. When, for

whatever reason, the primary AMA **138** is no longer functioning properly, one of the AMA backups **138** may take on the role of the central or master AMA.

[0056] The ATC name server network **134** may comprise a plurality of name server agents **118-1, 118-2, . . . , 118-k**, each of which may be designed to be responsible for the DNS requests of a particular geographical (or any other administrative or functional) region. For example, the name server agent **118-1** may be responsible for processing all the DNS requests from North America, the name server **118-2** may be responsible for Europe's DNS requests, and the name server **118-3** may be responsible for DNS requests from Japan. In addition, a name server agent may also serve as a back up name server agent for other name server agents in the network. For example, if the North America name server agent is not functioning properly, the Europe name server agent may be temporarily assigned to handle the DNS requests from North America. For that purpose, all ATC policies may be propagated to all of the name server agents in the ATC name server network **134**.

[0057] FIG. 6 depicts a high level functional block diagram of an administrative master agent (AMA) **136** according to embodiments of the present invention. The AMA **136** comprises a secure web-based graphical user interface **138**, a policy editing mechanism **140**, an ATC policy database **142**, an administrative policy update mechanism **144**, an ATC administrative browser **146**, an ATC policy management mechanism **148**, a report generation mechanism **150**, a monitoring mechanism **152**, and an administrative master backup mechanism **154**.

[0058] The ATC policy database **142** stores the ATC policies. As discussed above, the ATC policies may include policies from different sources (e.g., from subscribers and from other entities that control network traffic). Both subscriber policies and other policies may be defined and provided to the AMA **136**. In the exemplary

configuration shown in **FIG. 6**, the policies are received at the **AMA 136** through the secure web based **GUI 138**. There are other means through which the policies may be delivered to the **AMA 136** and will be discussed later. The policy editing mechanism **140** organizes the received policies to form the ATC policies and then stores them in the ATC policy database **142**.

[0059] The ATC policy management mechanism **148** may broadcast or propagate the ATC policies to all the name server agents in the ATC name server network **134** so that the ATC policies may be used to control the domain name translation service. The stored ATC policies may be dynamically updated via different means. For example, the policies may be updated through the ATC administrative browser **146**, and the administrative policy update mechanism **144** may revise existing ATC policies. As another example, the ATC policy management mechanism **148** may regularly poll dynamically updated policies from locations that are specified as part of some of the ATC policies.

[0060] Broadcasting ATC policies may take place periodically according to some pre-defined interval or may be triggered whenever the stored ATC policies are updated. The ATC policy management mechanism **148** may monitor changes made to the existing ATC policies. The ATC policy management mechanism **148** may poll the ATC policies stored in the ATC policy database and see whether there are changes. On the other hand, whenever the ATC policy management mechanism **148** polls dynamic policies from specified locations (specified in existing ATC policies), it may determine whether the dynamically polled policies differ from existing ATC policies. In the event that updated policies are different from the existing ATC policies, the ATC policy management mechanism **148** may re-broadcast the updated ATC policies to the ATC name server network **134**.

[0061] The monitoring mechanism **150** may collect DNS log summaries from different name servers in the ATC name server network **134**. Such summary log data may provide information such as, for example, the number of requests directed to particular servers in a given time period. Each administrative Report Generation agent receives and collectively processes DNS log summaries from the entire ATC system. The report generation mechanism **150** generates status reports from these summaries and makes such reports available to subscribers via the secure web-based GUI **138**.

[0062] The administrative master backup mechanism **154** may periodically update the AMA backups **138-1, . . . , 138-m** to ensure that all the backup agents are current. This may include replicating the ATC policies, the operational status of various control mechanisms (including the ATC policy management mechanism **148**), the policy editing mechanism **140**, and the administrative policy update mechanism **144**, and providing the up-to-date monitoring information to the AMA backups.

[0063] A major function of an administrative master agent is to manage the ATC policies, to make sure that updated ATC policies are supplied to the name server agents in the ATC name server network **134**, to monitor the various name servers' performance, to generate dynamic status report of system performance, and to maintain a connection through which policies may be updated dynamically and status report can be examined.

[0064] Information flagging system errors and other anomalous conditions is collected by the monitoring mechanism or agent **152**. FIG. 7 depicts an internal functional block diagram of a monitoring mechanism **152**, according to embodiments of the present invention. The monitoring mechanism **152** includes a trap handler **158**, a processing mechanism **160**, and an alert generation mechanism **162**. The trap handler **158** traps events from the name servers in the name server network **134**. The processing mechanism **160** analyzes both the collected trapped events, and, based on analyzed information, the

alert generation mechanism **162** generates alerts when necessary, and reports such alerts to, for example, the ATC's network operation center (NOC) **164** (**FIG. 1**).

[0065] The trap handler **158** further comprises an event receiver **166** that intercepts trap events from the name servers and a trap consolidation mechanism **168** which may classify the trapped events and organize them in a reasonable and appropriate fashion. The processing mechanism **160** may process the consolidated events to identify useful or informative patterns which may be further used, by the alert generation mechanism **162** to identify problematic patterns which may significantly affect the system performance.

[0066] The Adaptive Traffic Control (ATC) framework according to the present invention may be deployed as a stand-alone service directing traffic solely to the subscriber's servers, in conjunction with another CDN provider, or in conjunction with any other service.

[0067] A subscriber may activate (turn up) the ATC system (DNS servers) in one of two ways: using a DNS CNAME or using NS delegation. Instead of using ATC to direct traffic for a single DNS hostname, the subscriber may have many different subdomains that it would like to direct to ATC. For example, the subscriber might want all downloads from *dl.customer.com* handled by ATC, together with all of its subdomains, but have all other domains that it controls, such as *www.customer.com*, resolved by its own name server:

```
dl.customer.com. => ATC
any.thing.dl.customer.com. => ATC
www.customer.com. => not ATC
customer.com. => not ATC
```

[0068] In this case, instead of adding the CNAME record in the subscriber's DNS zone file, it simply *delegates* the *dl.customer.com.* name to ATC via NS records in its zone file.

[0069] The delegation (NS-records) method is preferred as it is more flexible and offers all of the reliability, scalability and flexibility of ATC. After the initial contact for delegation, a properly operating resolver making occasional queries will not contact the subscriber's own name server or name servers again. The result is much better name resolution performance since the unnecessary overhead of routing each fresh DNS request through the subscriber's name server is eliminated. The CNAME method keeps the subscriber's own name server in the loop. That is, whenever the CNAME TTL expires, client resolvers will return to the subscriber's name servers to refresh the record.

[0070] For both CNAME and NS delegation methods, the procedure to turn off the ATC switch is the same – the subscriber edits its DNS zone files to remove the lines that delegate authority to ATC. DNS requests will continue to be served in accordance with the subscriber's defined ATC policies until the TTL's have expired on the appropriate delegations.

The System in Operation

FIG. 8 is an exemplary flowchart of a process, in which a name server agent serves a DNS request based on subscriber policies, according to an embodiment of the present invention. First, a user enters a URL into the user's browser **114** (or into any application that accepts URLs as input and obtains the corresponding resource for the client) (at **170**). The client's resolver **116** attempts to resolve the hostname of the URL in order to obtain an IP address of a server from which the resource identified by the URL can be obtained (at **172**). The resolver **116** will be directed by the client's DNS (not shown) to a DNS Name Server **118** in the ATC name server network **134** (at **174**). The resolver **116** provides the name server **118** with the hostname it is attempting to resolve.

The name server **118** first determines the location of the requesting resolver **116** using the location determinator **128** (at **176**). Using this determined location and policies

from the policy database 124, the ATC mechanism 126 selects one or more servers in the subscriber server network 104 that can process the client's request (at 178). The IP address(es) of the selected server(s) are returned to the requesting resolver 116 (at 180). The browser 114 then connects to one of the servers (at 182) in order to obtain the requested resource.

Policy Administration

[0071] The ATC policies may be initially set up and later adaptively updated according to servers' dynamic operational status. The ATC policies may be formed via various means. The described approaches may also be applied to form other policies. A GUI method or a file-based method may be employed to set up subscriber policies. Through the GUI method, different means to form subscriber policies may be adopted. For example, a browser may be used so that a subscriber can enter policies directly. An XML file containing descriptions of the subscriber policies may also be loaded using GUI method so that the policies may be parsed and store. As another alternative, a file containing descriptions of the subscriber policies may also be loaded in a similar fashion and parsed accordingly. When a file based method is used, a file containing descriptions of the subscriber policies, either constructed based on XML or some other structure, may be accessed via other means such as FTP.

[0072] In some preferred embodiments, a subscriber accesses a secure web-based GUI 138 (FIG. 6) using subscriber 112 secure browser interface 132.

[0073] FIGS. 9(a) – 10(k) illustrate various exemplary GUI interfaces through which network resources (servers in a subscriber server network) may be defined and the ATC policies that determine how network resources should be utilized may be specified. FIGS. 9(a) – 9(e) show an exemplary process, in which a subscriber specifies resources (servers) in a subscriber server network 104 through secure web based graphical user

interfaces 138, according to embodiments of the present invention. FIG. 9(a) shows a web-based interface for creating a new policy, which allows a subscriber to perform a variety of tasks listed in the Policy Editor section on the left side of the window. These listed tasks are related to subscriber policies. For example, the first on that list is “*Decision Tree Window*” which allows a subscriber to construct policies. Next is the task “*Resource window*” that allows a subscriber to define network resources. Furthermore, once a subscriber server network and its governing policies are defined, a subscriber can save, discard, or activate a policy.

[0074] FIG. 9(b) shows a “*Resource Window*” (e.g., invoked through selecting the “*Resource window*” option in FIG. 9(a)) through which a subscriber can specify relevant network resources. Different kinds of network resources may be defined. For example, a server may be defined by its IP address by selecting the “*IP address(es)*” option which invokes the interface shown in FIG. 9(c), in which a subscriber may specify a server by a so-called Nickname (or alias), its IP address, and its time to live (TTL) parameter. For example, two servers are defined via the interface in FIG. 9(c). One defined server is a server named “New York” (which may be located in New York) with IP address 10.0.0.1 with TTL 1d. Another server named “London” (which may be located in London) is defined as having IP address 10.2.2.2 with TTL 1d. Once these two servers are defined, they are appropriately listed in the Resource window (as shown in FIG. 9(d)).

[0075] With defined network resources (server “New York” and server “London”), a subscriber may further define the policies that govern the process of selecting a server. This can be accomplished by selection of the “*Decision Tree Window*” on the left frame of the window shown in FIG. 9(a). A decision tree window is then invoked through which a subscriber can define policies.

60325177-093901
109260-715300

[0076] FIGS. 10(a) – 10(k) show an exemplary process, in which a subscriber specifies geo-political policies through secure web based graphical user interfaces, according to an embodiment of the present invention. FIG. 10(a) illustrates a window interface for construct a decision tree. A decision tree may comprise a plurality of nodes, each corresponding to a branch point in the tree and the branch decision may be governed by a rule (policy) defined based on some criteria. To add a node in a decision tree, a subscriber may select the option “*Add New Node*” in the interface shown in FIG. 10(a). This selection invokes an *Add New Node* window interface illustrated in Fig 10(b). In the *Add New Node* window, a subscriber may specify at which node the policy is to be defined (see example described in reference to FIGS. 10(d) – 10(h)) and the type of subscriber policy to be defined for that node. For example, the policy to be defined may be a geo-political policy that split based on continent, on country, or on time zone. In the example shown in FIG. 10(b), the type of the policy to be defined is specified as a policy that splits based on country.

[0077] The GUI interface shown in FIG. 10(c) is a window that allows a subscriber to define a geo-political policy based on country. To proceed further, a subscriber selects the option “*Add New Node*,” which further invokes the interface shown in FIG. 10(d). Through this interface, a subscriber can specify the server nodes involved in a split based on country. For example, in FIG. 10(d), a subscriber selects United States as a country from a list of countries displayed in the upper left window frame and then further specifies the corresponding resource node type as IP address(es). That is, according to this example, the requests from the United States will be handled by a server or servers that will be specified based on its IP address(es) (see FIG. 10(e)).

[0078] FIGS. 10(f) and 10(g) describe a similar selection scenario, in which the country selected is “*Others*” (i.e., all other countries) and the resource node type is IP

address(es). So, at this point, two “countries” (U.S.A. and all others) are specified and requests from these two parts of the world will be directed to servers that are defined according to their IP addresses. This policy is summarized in **FIG. 10(g)**.

[0079] **FIG. 10(h)** describes the interface through which a subscriber can further specify the IP addresses of the servers that are assigned to handle the requests for these two parts of the world (U.S.A. and all other countries). To specify the IP address of the server that is assigned to handle requests that are not initiated from the United States, a subscriber can simply select the “*IP Address(es)*” option following the “*Other*” option. This selection will invoke the interface shown **FIG. 10(h)**, which lists all the network resources that are defined according to their IP addresses. A subscriber can simply select the server(s) by selecting the appropriate boxes. For example, in the interface shown in **FIG. 10(h)**, server *London* with IP address 10.2.2.2 is selected to handle requests from all other countries. This policy is summarized in the interface shown in **FIG. 10(i)**. Similarly, to specify the server to handle the requests initiated from the United States, a subscriber may simply select the option “*IP address(es)*” following the option “*United States*” (in **FIG. 10(i)**) and then select one of the servers in their IP addresses in the interface shown in **FIG. 10(j)** that is subsequently invoked. For instance, as shown in **FIG. 10(k)**, server New York with IP address 10.0.0.1 is selected to handle requests initiated from the United States. **FIG. 10(k)** summarizes the subscriber’s policy based on split on country, which specifies that all requests initiated from the United States will be handled by server New York and all requests initiated from other parts of the world will be handled by server London.

[0080] **FIGS. 11(a)-11(c)** show exemplary secure web based graphical interfaces, through which a subscriber may define load share policies and overflow policies with respect to specified network resources, according to an embodiment of the present

invention. There may be different types of load sharing servers: (1) static servers, (2) dynamic servers, and (3) overflow servers. As can be seen in **FIG. 9(b)**, a resource server may be defined as static, dynamic, or an overflow server.

[0081] Correspondingly, the policies governing routing requests to different types of servers may also be defined accordingly. First, the policies that govern static servers are applied when servers are relatively static and do not change often over time. This may mean that the availability of the servers is fairly stable and load sharing among different servers is also relatively stable. The policies that control dynamic servers are applied when servers are expected to change frequently. The overflow policies control the flow when, for some reason, primary servers, either static or dynamic, become unavailable.

[0082] The load share with respect to a given server represents the amount of traffic that will be sent to that server within a static server setting. An exemplary method to compute the fraction of the traffic sent to the server is to sum the load share numbers for all servers in a current static server setting and divide the load share for that server by the summed total. The shed fraction for a given server represents the percentage of traffic that should be redirected away from the server. The redirection may be carried out after a decision based upon load share policies. **FIG. 11(a)** gives an example interface, in which each of the servers listed is assigned various selection related parameters. For example, for each server, a TTL is specified. In addition, a load share and a shed fraction may also be specified. For example, both server New York and Seattle have load share of 0.1. The shed fraction assigned to server New York is 0.2, meaning that 20% of its load is re-directed to some other server, and the shed fraction assigned to server Seattle is 0.3, meaning that 30% of its load is re-directed to some other server.

[0083] For each resource server, an on-line flag may also be set to indicate whether the server is currently available. This flag is shown in **FIG. 11(a)** in the rightmost

column. With this flag, a server may be temporarily removed from the service, for instance, a server needs to be taken down for maintenance.

[0084] Due to the fact that dynamic servers are expected to change relatively frequently, policies that govern their selection may be defined in two stages. The policies for dynamic servers may be initially specified in a similar fashion as for static servers. For instance, load share and shed fraction can be defined for dynamic servers New York and Seattle, as shown in **FIG. 11(a)**. During operation, however, the policies that govern the selection of dynamic servers may be established dynamically. For example, initially defined policies (e.g., load share, shed fraction, and on-line flag) may be adaptively revised based on, for instance, an on-line status report retrieved from a specified location on the network. Dynamic policies provide considerable flexibility in how the selection may be conducted in a manner that is adaptive to the network health or any other network performance related factors (e.g., maintenance).

[0085] The on-line status report may be provided at a network location specified by the subscriber. The location may be specified as a Uniform Resource Locator (URL) and it may be accessed through a HTTP. The location of the status report may be specified in the initial policy, providing a constant link to the location. The access may be authenticated and such authentication requirements may be also specified in the initial policies. The on-line accessed status report may be constructed based on information gathered during monitoring the servers' performance. The frequency at which the status report is accessed may also be specified explicitly. For example, it may require the ATC mechanism (specifically, the ATC policy management mechanism) to poll the status report every thirty seconds.

[0086] The status report may be constructed using some standard language such as extendible Markup Language (XML). The status report may contain the revised policies,

which may be determined by the subscriber manually based on network performance or may be devised by an automated process based on network performance. In this scenario, the ATC mechanism updates the dynamic policies based on the information contained in the status report. For example, a status report may indicate that the current load share for server New York should be 0.4 (up from 0.2 previously) with the same shed fraction and the current load share for server London should be 0.0 (down from previous 0.3) with 1.0 shed fraction (100% shed fraction). An alternative scenario may be that the status report provides network performance data about each dynamic server and the ATC mechanism, after accessing the status report, determines how the current policies may be revised accordingly.

[0087] When the status report provides dynamic policies, it may, in general include, for each dynamic server, the following information: `<ServerStatus loadShare="LS" shedFraction="SF" online="Boolean"/>`, where LS and SF represent numerical numbers and "Boolean" represents a logical value of either "true" or "false".

[0088] With the mechanism of dynamic policies described above, a subscriber can easily implement changes to existing policies (e.g., to load share, shed fraction policies and the On-line flag) without having to update an entire ATC policy hierarchy. In addition, a parameter called "Down on line" can be set in an ATC policy to control the behavior of the ATC mechanism under the circumstance that the ATC policy management mechanism is unable to properly access the status report due to reasons such as a failed polling or retrieval or failing to parse the status report's content. According to some embodiments of the present invention, three different methods may be applied to handle the situation.

[0089] With the first method, a subscriber instructs the ATC mechanism not to select a particular server if the status report for that server can not be properly obtained. The subscriber specifies this policy by setting parameter “Down on line” to false. In this case, the ATC mechanism will not consider the underlying server as a candidate for selection process until the polling mechanism retrieves a valid status report.

[0090] FIG. 11(b) shows an exemplary GUI interface for specifying a dynamic policy. For each server (row), next to the server name and IP address, various parameters concerning policies may be set, including TTL, URL for the location of the status report, *Down on line*, default load share, default shed fraction, and basic authentication requirement (e.g., a password). These parameters constitute the original policies or a given dynamic server.

[0091] The dynamic aspects of a dynamic server can be further seen through parameters such as download load share and download shed fraction. These are the parameters that are updated through the dynamic process described above.

[0092] The second method to deal with a polling failure is to allow the ATC mechanism to select the underlying dynamic server according to its default policies (or original policies). To specify this solution, the subscriber sets the “Down on line” parameter true. The third method to handle the situation where the status report can not be properly retrieved is to allow the ATC mechanism to select the underlying server if its “on line” flag is on (it is available). To achieve this, a subscriber specifies only the “on line” ServerStatus in the status report, with parameter “Down on line” set to either true or false. In this case, the ATC mechanism will use the Default load share and Default shed fraction specified in the policy when the “On line” flag is set to true.

[0093] Both static and dynamic servers are primary servers, although their selection may be controlled by operationally different policies. The third type of servers

are called overflow servers. They provide alternatives when primary servers, for some reason, can not be selected. Typically, an overflow server corresponds to a Canonical name or *CName*, pointing to a service providing a CDN (such as, e.g., Digital Island's Footprint service). The choice of overflow servers may be determined based on the belief that they are in general always available. An overflow server is defined through a Overflow Server Window illustrated in FIG. 11(c), which can be invoked by selecting the option "*Overflow Server(s)*" in the Resource window shown in FIG. 9(b). In the Overflow Server window in FIG 11(c), the overflow server is defined by a CName, which points to the CDN server address "*customer.footprint.net*" (e.g., CDN 105, shown in FIG. 1).

[0094] FIGS. 12(a) – 12(b) show example subscriber policies, defined based on a set of resource servers, that govern the selection according to geographical location of an incoming request and the time zone of each of the locations. In FIG. 12(a), three primary resource servers are defined, including "eastserver" with IP address 10.0.0.2, "westserver" with IP address 10.0.0.1, and a service network with CName "*customer.footprint.net*".

FIG. 12(b) shows an exemplary decision tree that are built based on the three primary servers as well as a set of geographical policies and time zone based policies.

[0095] In the decision tree depicted in FIG. 12(b), the selection is first directed to different resource servers at the top level according to a split between the United States and rest of the world. A request initiated from non U.S. geographical regions is directed to the CDN denoted "*customer.footprint.net*". Within the United States, time zone based policies are further applied. A request initiated from a region within the central time zone is directed to the server denoted "eastserver," as are requests initiated from a region within the eastern time zone. A request initiated from a region within either the mountain time zone or the pacific time zone is directed to the server designated "west server." Finally, a

request initiated from a region within the United States that does not fall in any of these time zones (e.g., Alaska or Hawaii) is also directed to “west server”. Note that any or all of the servers may in fact be groups of servers or CDNs.

[0096] FIGS. 12(c) – 12(d) show another example decision tree constructed based on a set of resource servers and subscriber policies that are defined based on geographical locations of an incoming request with overflow policies that allow the ATC mechanism to direct traffic to pre-defined services when primary servers are not available. In the Resource window shown in FIG. 12(c), four resources are defined: static servers New York and Seattle, each defined based on its IP address, a service network named “International” with CName “*customer.fp.sandpiper.net*”, and another service network with CName “*www.cdn.download.customer.com*” as an overflow server network. These resource servers are the end node in a decision tree. Their configuration is determined by the governing policies. FIG. 12(d) illustrates an example decision tree built based on these resources.

[0097] The decision tree in FIG. 12(d) first splits three ways at the top level based on country. It is classified as from Canada, the United States, or others. A request initiated from neither United States nor Canada is directed to server “International”. A request initiated from Canada is directed first to static server Seattle (a primary server). If primary server Seattle is not available, the request is directed to overflow server service (CDN) “*www.cdn.download.customer.com*”. For a request from the United States, the second level of split is carried out based on time zones. If the request is the central or the eastern time zones, the request is directed to primary server “New York.” If the request is from the western, mountain, or other time zones, it is directed to primary server “Seattle”. If any of the two primary servers is not available, the request is to be directed to the corresponding overflow server (CDN) service “*www.cdn.download.customer.com*.”

[0098] In addition to the various web-based GUI interfaces described herein, there are also other means through which subscriber policies may be defined. As discussed earlier, for example, subscriber policies may also be constructed or specified in XML.

[0099] Policies defined via different means (e.g., through web based GUI or XML file) may be converted into some pre-defined format within the ATC mechanism. Such pre-defined format may be designed for efficiency in manage and handling the ATC policies. For example, the internal format for ATC policies may be designed so that, internally, the AMAs can conveniently store, access, and broadcast the ATC policies to the name server agents and the name server agents can efficiently apply the policies.

[00100] As mentioned earlier, in additional to manage the ATC policies, the AMA may also monitor the performance of name servers and generates viewable DNS log reports. The monitoring mechanism may gather performance information from either the DNS logs of the name servers or the events trapped from the name servers. Such gathered information may be used by the report generation mechanism to construct informative reports. The report generation mechanism may also make such reports available to the subscribers via the secure web-based GUI. **FIG. 13 (a) – 13(c)** show exemplary types of report information that is accessible via the secure web-based GUI. **FIG. 13(a)** depicts a web interface that allows a subscriber to view the DNS log data of a specified resource server. A subscriber can specify the name of the server under review (e.g., “ncust.atc.foorprint.net”), the form in which the log data is to be organized (e.g., table or graph), the time period under review, and the time resolution used in displaying the log data.

[00101] **FIG. 13 (b)** shows a plot of the log data for a specified server that is displayed according to the viewing parameters specified in the view archived logs window shown in **FIG. 13 (a)**. For example, the plot is a graph form and it is plotted against a

specified period of time and with certain resolution. It can be seen in this example plot that there is a surge in a measure called “answers per minute” (or APM) after 18:00 on the 31st of August 2001. FIG. 13(c) shows a graphical display for the same log data in FIG. 13(b) in a table form, in which the log data is listed in chronicle order.

[00102] FIG. 14 is an exemplary flowchart of a process, in which a central or master administrative master agent 136 (FIGS. 1, 5, 6) manages subscriber policies and monitors associated name server agents 118, according to embodiments of the present invention. Policies are received (at 182). This may include both subscriber policies and other policies from other entities. The received policies are used to construct ATC policies that are stored (at 184) in the ATC policy database 142. The AMA then broadcasts (at 186), the ATC policies to the name server agents in the ATC name server network 134. The broadcast may take place at some specified times according to, for example, certain pre-defined periodicity. (Each name server 118 stores the policies in its policy database 124.)

[00103] While maintaining the ATC policies, the central AMA monitors (at 188), the status of the ATC name servers 118. As discussed earlier, this may include gathering DNS log summary data by polling the individual name servers. Based on collected log data, the AMA generates report (at 190) and makes such reports viewable via the secure web-based GUI interface 138. The central AMA regularly determines (at 192) whether or not update copies (at 194) other distributed AMAs. The AMA may also regularly accesses status reports (which contain adaptive policies) from subscriber specified locations as part of dynamically maintaining adaptive ATC policies. The AMA determines (at 196) whether on-line ATC policy update is required. If update is required, the processing returns back to obtain on-line policies (at 182).

60325477 "052804
100250 4432200

[00104] FIG. 15 is an exemplary flowchart of a process, in which the monitoring mechanism 152 in the administrative master agent (136) monitors the performance of name server agents 118, according to embodiments of the present invention. Traps are collected from different name servers 118 (at 198). If an event is trapped (determined at 200), the trapped event is consolidated (at 202). The consolidated events (if any) are processed (at 204). Based on the processing results, alerts are generated (at 206) and are sent to the ATC browser (or some other device) for display (at 208).

[00105] The DNS service log data is stored in the DNS log storage from which the DNS log reporting mechanism reports the DNS log data of the underlying name server agent to the AMA. On the name server agent side, the trap monitor also sends events to the AMA if irregular behavior has been observed.

[00106] The various mechanisms described herein, including, without limitation, the adaptive traffic control (ATC) mechanism, the location determination mechanism, policy editing mechanism, administrative policy update mechanism, ATC policy management mechanism, report generation mechanism, a monitoring mechanism, and an administrative master backup mechanism may be implemented in hardware, software or a combination thereof. When implemented in software, they may be implemented in any type of appropriate interpreted or compiled programming language. When implemented fully or partially in software, aspects of the invention can reside on any memory or storage medium, including but not limited to a ROM, a disk, an ASIC, a PROM and the like. While the invention has been described with reference to particular mechanisms (algorithms, processes and functions) and architectures, one skilled in the art would realize that other mechanisms and/or architectures could be used while still achieving the invention.

100200 09301

[00107] When the various mechanisms of the present invention are running on a particular machine (e.g., the at a client or on a server), they may reside in the memory of the machine or on a storage device or in a combination. Further, while many of the operations have been shown as being performed in a particular order, one skilled in the art would realize that other orders, including some parallelization of operations, are possible and are considered to be within the scope of the invention.

[00108] The present invention has been described above in connection with a preferred embodiment thereof; however, this has been done for purposes of illustration only, and the invention is not so limited. Indeed, variations of the invention will be readily apparent to those skilled in the art. Such variations also fall within the scope of the invention. Thus, while the invention has been described with reference to the certain illustrated embodiments, the words that have been used herein are words of description, rather than words of limitation. Changes may be made, within the purview of the appended claims, without departing from the scope and spirit of the invention in its aspects. Although the invention has been described herein with reference to particular structures, acts, and materials, the invention is not to be limited to the particulars disclosed, but rather extends to all equivalent structures, acts, and, materials, such as are within the scope of the appended claims.

[00109] The processing described may be performed by a properly programmed general-purpose computer alone or in connection with a special purpose computer. Such processing may be performed by a single platform or by a distributed processing platform. In addition, such processing and functionality can be implemented in the form of special purpose hardware or in the form of software being run by a general-purpose computer. Any data handled in such processing or created as a result of such processing can be stored in any memory as is conventional in the art. By way of example,

such data may be stored in a temporary memory, such as in the RAM of a given computer system or subsystem. In addition, or in the alternative, such data may be stored in longer-term storage devices, for example, magnetic disks, rewritable optical disks, and so on. For purposes of the disclosure herein, a computer-readable media may comprise any form of data storage mechanism, including such existing memory technologies as well as hardware or circuit representations of such structures and of such data.