

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google LLC,

Petitioner

v.

Sandpiper CDN, LLC,

Patent Owner

Case: IPR2025-00826

U.S. Patent No. 9,021,112

Issue Date: April 28, 2015

Title: Content Request Routing and Load Balancing for
Content Distribution Networks

**DECLARATION OF TODD C. MOWRY, PH.D.
IN SUPPORT OF PETITION FOR INTER PARTES REVIEW OF
U.S. PATENT NO. 9,021,112**

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Qualifications and Background	2
III.	Materials Considered	6
IV.	Legal Standards	7
	A. Claim Construction	8
	B. Obviousness Under 35 U.S.C. § 103.....	8
V.	Technical Background.....	12
VI.	The '112 patent	15
	A. Overview of the '112 Patent.....	15
	B. Summary of the Prosecution History of the '112 patent.....	17
	C. Claim Construction of Terms in the '112 patent.....	18
VII.	Person of Ordinary Skill in the Art.....	18
VIII.	The Prior Art.....	19
	A. Overview of Glines (EX-1002).....	19
	B. Overview of Gupta (EX-1003).....	21
	C. Overview of Chung (EX-1004).....	22
	D. Overview of Chiou (EX-1005).....	23
	E. Overview of Colby (EX-1006).....	24
IX.	Opinions on Unpatentability.....	25
	A. Grounds 1A and 1B Render Obvious Claims 1-8 and 10-23.....	26
	1. Motivation to Combine Glines and Gupta.....	26
	2. Claim 1	28

3. Claim 2: “The method as recited in claim 1 wherein at least some of said plurality of CDN DNS servers are located at network Points of Presence (POPs).” 61
4. Claim 3: “The method as recited in claim 2 wherein each of said plurality of CDN DNS servers is located at a network POP.” 65
5. Claim 4: “The method as recited in claim 2 wherein at least one of said plurality of CDN DNS servers and respective plurality of content servers is grouped as a cluster.” 66
6. Claim 5: “The method as recited in claim 1 wherein each of said plurality of content servers is associated with a network Point of Presence (POP).” 70
7. Claim 6: “The method as recited in claim 1 further comprising: (D) by the content server, attempting to serve the content.” 71
8. Claim 7: “The method as recited in claim 6 wherein said step (D) of attempting comprises: (d1) if a valid version of the content is available on the content server, serving the content; otherwise (d2) obtaining the content from a content source and then serving the content.” 72
9. Claim 8: “The method as recited in claim 7 wherein, said content source is an origin server associated with said content provider.” 78
10. Claim 10: “The method as recited in claim 1, wherein selection of the content server is based on a load balancing policy.” 79
11. Claim 11: “The method of claim 1 further comprising: (c3) by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.” 81

12.	Claim 12: “The method of claim 11, wherein the step of (c3) is performed in accordance with a predetermined selection policy.”	82
13.	Claim 13	83
14.	Claim 14: “The method as recited in claim 13, wherein selection of at least one of the first content server and the second content server is based on a load balancing policy.”	98
15.	Claim 15: “The method as recited in claim 13, wherein the resolving DNS server of step (c11) is the same as the resolving DNS server performing of step (c21).”	99
16.	Claim 16: “The method as recited in claim 15, wherein the switch of step (c12) is the same as the switch of step (c22).”	100
17.	Claim 17: “The method as recited in claim 15, wherein the first content server is the same as the second content server.”	101
18.	Claim 18	102
19.	Claim 19	108
20.	Claim 20: “The method as recited in claim 19, wherein selection of the content server is based on a load balancing policy.”	113
21.	Claim 21	113
22.	Claim 22	117
23.	Claim 23:	122
B.	Grounds 2A and 2B Render Obvious Claim 4	132
1.	Motivation to Combine Glines with Chung (EX-1004), and Glines-Gupta with Chung	132

2.	Claim 4: “The method as recited in claim 2 wherein at least one of said plurality of CDN DNS servers and respective plurality of content servers is grouped as a cluster.”	137
C.	Grounds 3A and 3B Render Obvious Claim 7-9, 19, and 20	139
1.	Motivation to Combine Glines with Chiou (EX-1005), and Glines-Gupta with Chiou	139
2.	Claim 7: “The method as recited in claim 6 wherein said step (D) of attempting comprises: (d1) if a valid version of the content is available on the content server, serving the content; otherwise (d2) obtaining the content from a content source and then serving the content.”	145
3.	Claim 8: “The method as recited in claim 7 wherein, said content source is an origin server associated with said content provider.”	147
4.	Claim 9: “The method as recited in claim 7 wherein, if a valid version of the content is not available on the content server, the content server uses a table to ascertain where to obtain the content.”	148
5.	Claim 19	150
6.	Claim 20: “The method as recited in claim 19, wherein selection of the content server is based on a load balancing policy.”	155
D.	Grounds 4A/4B and 5A/5B Render Obvious Claim 10-22	156
1.	Motivation to Combine Glines, Glines-Gupta, Glines-Chiou, or Glines-Gupta-Chiou with Colby (EX-1006)	156
2.	Claim 10: “The method as recited in claim 1, wherein selection of the content server is based on a load balancing policy.”	161
3.	Claim 11: “The method of claim 1 further comprising: (c3) by at least one switch, selecting the content server	

	from said respective plurality of content servers associated with said resolving CDN DNS.”	161
4.	Claim 12: “The method of claim 11, wherein the step of (c3) is performed in accordance with a predetermined selection policy.”	162
5.	Claim 13	163
6.	Claim 14: “The method as recited in claim 13, wherein selection of at least one of the first content server and the second content server is based on a load balancing policy.”	166
7.	Claim 15: “The method as recited in claim 13, wherein the resolving DNS server of step (c11) is the same as the resolving DNS server performing of step (c21).”	167
8.	Claim 16: “The method as recited in claim 15, wherein the switch of step (c12) is the same as the switch of step (c22).”	168
9.	Claim 17: “The method as recited in claim 15, wherein the first content server is the same as the second content server.”	169
10.	Claim 18	170
11.	Claim 19	171
12.	Claim 20: “The method as recited in claim 19, wherein selection of the content server is based on a load balancing policy.”	172
13.	Claim 21	173
14.	Claim 22	174
X.	Secondary Considerations	176
XI.	Conclusion	176

I. INTRODUCTION

1. I, Todd Mowry, submit this declaration to state my opinions on the matters described below.

2. I have been retained by Google LLC (“Petitioner”) as an independent expert in this proceeding before the United States Patent and Trademark Office. I am being compensated at the rate of \$800 per hour for time that I spend on this case. No part of my compensation depends on the outcome and I have no other interest in this proceeding.

3. I understand that this proceeding involves U.S. Patent No. 9,021,112 (“the ’112 patent,” EX-1001), and I have been asked to provide my opinions as to the patentability of the claims of the ’112 patent.

4. I have been asked to consider the validity of certain claims of the ’112 patent based on certain prior art references. I have also been asked to consider the state of the art and prior art available as of October 18, 2001, which is the earliest claimed priority date on the face of the ’112 patent. Based on the combinations of prior art references discussed in this declaration, it is my opinion that claims 1-23 of the ’112 patent are unpatentable.

II. QUALIFICATIONS AND BACKGROUND

5. Below is a summary of my education, work experience, and other qualifications. My qualifications are stated more fully in my curriculum vitae, which I understand is provided in EX-1009.

6. I am a Professor in the Department of Computer Science at Carnegie Mellon University. I also have a courtesy appointment in the Department of Electrical and Computer Engineering. I have served on the faculty of Carnegie Mellon University for 27 years, starting in 1997 through the present (2025).

7. I served on the faculty of the University of Toronto for four years between 1993 and 1997, in the Department of Electrical and Computer Engineering and by courtesy appointment in the Department of Computer Science. Prior to that appointment, I served as a Graduate Research Assistant in the Department of Electrical Engineering at Stanford University for four years between 1989 and 1993.

8. As a faculty member, I have taught and continue to teach courses and direct research in computer systems and software, operating systems, distributed and network systems, parallel processing, and compiler optimization.

9. I received a B.S. degree in Electrical Engineering with Highest Distinction from the University of Virginia in May 1988. I received an M.S. in Electrical Engineering from Stanford University in June 1989, and a Ph.D. in Electrical Engineering from Stanford University in March 1994.

10. I have worked in the computer industry in various capacities. I was a part-time Computer Architect and then Computer Architecture Consultant at Silicon Graphics, Inc., in Mountain View, California (formerly MIPS Computer Systems in Sunnyvale, California), from 1989 to 1993 and 1993 to 1996, respectively. I was a Visiting Scientist at IBM in Toronto from 1996 to 2004. During that same time period (1996 to 2004), I was also a Member of the Technical Advisory Board of SandCraft, Inc. in Santa Clara, California. I was the Director of the Intel Research Pittsburgh Lab at Intel Corporation in Pittsburgh, Pennsylvania from 2004 to 2007.

11. I have authored 21 journal articles and 71 conference papers. I am also an inventor on five patents.

12. I have published a number of papers in the top research conferences in the fields of data storage, caching, operating systems and distributed systems (including a paper that won the Best Paper Award at the USENIX 2nd Symposium on Operating Systems Design and Implementation (OSDI '96)). *See, e.g.*, Charles Garrod, Amit Manjhi, Anastasia Ailamaki, Bruce Maggs, Todd Mowry, Christopher Olston, and Anthony Tomasic, *Scalable Query Result Caching for Web Applications*, in Proceedings of the 34th International Conference on Very Large Databases (VLDB) (2008); Andrew Pavlo, Gustavo Angulo, Joy Arulraj, Haibin Lin, Jiexi Lin, Lin Ma, Prashanth Menon, Todd Mowry, Matthew Perron, Ian Quah, Siddharth Santurkar, Anthony Tomasic, Skye Toor, Dana Van Aken, Ziqi Wang,

Yingjun Wu, Ran Xian, and Tieying Zhang, *Self-Driving Database Management Systems*, in Proceedings of the 2017 Conference on Innovative Data Systems Research (CIDR) (2017); Amit Manjhi, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Anthony Tomasic, *Holistic Query Transformations for Dynamic Web Applications*, in 2009 IEEE 25th International Conference on Data Engineering (ICDE) (2009); and Amit Manjhi, Phillip B. Gibbons, Anastassia Ailamaki, Charles Garrod, Bruce M. Maggs, Todd C. Mowry, Christopher Olston, Anthony Tomasic, and Haifeng Yu, *Invalidation Clues for Database Scalability Services*, in Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering (ICDE) (2007).

13. I have also been an active member of the Parallel Data Lab at Carnegie Mellon University since 1997, which is often described as “academia’s premiere data systems research center.”

14. I am the recipient of several honors and awards:

- Arthur Samuel Thesis Award (awarded by the Stanford Computer Science department to the top two Ph.D. theses in a given year);
- IBM Faculty Development Awards (1996, 1997, 1998, 2000, 2001, 2002, and 2003);
- Best Paper Awards (Second Symposium on Operating Systems Design and Implementation in 1996; 20th International Conference on Data Engineering (ICDE) in 2004);

- Alfred P. Sloan Research Fellowship (awarded to researchers in recognition of distinguished performance and a unique potential to make substantial contributions to their field);
- Most Thought-Provoking Idea Award in 2004 (awarded by the Architectural Support for Programming Languages and Operating Systems (ASPLOS));
- TR100 Award in 1999 (awarded by MIT's Technology Review magazine to the top 100 most promising young innovators in science and technology); and
- I became an ACM Fellow in 2016.

15. I am a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association of Computing Machinery (ACM). I was the Editor-in-Chief of ACM Transactions on Computer Systems (the premier journal for computer systems research) from 2013 to 2018. I was an Associate Editor for the journal prior to that, since 2001. I was the Program Chair of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS) in 2010. I was the Co-Program Chair of the International Conference on Parallel Architectures and Compilation Techniques (PACT) in 2001. I have been on the programming committee in various years for ASPLOS, the International Symposium on Computer Architecture (ISCA), the International Symposium on Microarchitectures, and the Workshop on Architectural and System Support for Improving Format.

16. Overall, I have over 30 years of experience in the field of computer science and, specifically, computer architecture, operating systems, and document storage and retrieval.

17. In light of the foregoing, I consider myself to be an expert in the field of computer science, and I believe that I am qualified to provide an opinion as to what a person of ordinary skill in the art would have understood, known, or concluded regarding the subject matter of the '112 patent at the time of its alleged invention.

III. MATERIALS CONSIDERED

18. In forming my opinions, I have reviewed the following documents:

Exhibit	Description
EX-1001	U.S. Patent No. 9,021,112 (“the ’112 patent”)
EX-1002	WO2001039470 A1 to Glines (“Glines”)
EX-1003	U.S. Patent No. 6,405,252 to Gupta
EX-1004	U.S. Patent No. 6,470,389 to Chung
EX-1005	U.S. Patent No. 6,792,507 to Chiou
EX-1006	U.S. Patent No. 6,006,264 to Colby
EX-1007	Prosecution History of the ’112 patent
EX-1010	U.S. Patent No. 8,041,814 to Chase
EX-1011	James Kurose & Keith Ross, Computer Networking: A Top-Down Approach Featuring the Internet (1st ed. 2000)
EX-1012	Andrew Tanenbaum, Computer Networks (3rd ed. 1996)
EX-1013	U.S. Patent No. 6,829,654 to Jungck
EX-1014	James Kurose & Keith Ross, Computer Networking: A Top-Down Approach Featuring the Internet (3rd ed. 2005)
EX-1015	Mark Baker & Rajkumar Buyya, Cluster Computing at a Glance <i>in</i> High Performance Cluster Computing – Architectures and Systems, 3-47 (1999).

EX-1016	U.S. Patent No. 6,578,066 to Logan
EX-1017	U.S. Patent No. 8,041,814 to Chase
EX-1018	U.S. Patent No. 7,523,181 to Swildens
EX-1019	U.S. Patent No. 7,594,030 to Teodosiu
EX-1023	Schroeder, et al., <i>Scalable Web Server Clustering Technologies</i> , IEEE Network, Vol. 14, Issue 3 (June 2000).
EX-1024	The Wayback Machine – Yahoo! News, http://dailynews.yahoo.com/ , dated February 2, 2001 (<i>available at</i> https://web.archive.org/web/20010202075000/http://dailynews.yahoo.com/)
EX-1025	The Wayback Machine, “U.S. Senate Approves Ashcroft, 58-42,” http://dailynews.yahoo.com/h/nm/20010201/ts/congress_ashcroft_dc_71.html , dated February 2, 2001 (<i>available at</i> https://web.archive.org/web/20010202073200/http://dailynews.yahoo.com/h/nm/20010201/ts/congress_ashcroft_dc_71.html)
EX-1026	A. Bestavros et al., <i>Distributed Packet Rewriting and its Application to Scalable Server Architectures</i> , Proceedings Sixth International Conference on Network Protocols, IEEE (Oct. 1998).

IV. LEGAL STANDARDS

19. In considering the subject matter of the '112 patent and its claims in light of the prior art and forming my opinions, I am relying on certain legal principles that counsel in this case explained to me. My understanding of these concepts is summarized below.

20. I understand that the claims define the invention. I also understand that an unpatentability analysis is a two-step process. First, the claims of the patent are construed to determine their meaning and scope. Second, after the claims are construed, the content of the prior art is compared to the construed claims.

21. I understand that an invention, as defined by the claims of the patent, must not be anticipated or rendered obvious by the prior art.

A. Claim Construction

22. I understand that the United States Patent and Trademark Office interprets claim terms in an *inter partes* review proceeding under the same claim construction standard that is used in a United States federal court. I understand that, under this standard, the meaning of claim terms is considered from the viewpoint of a person of ordinary skill in the art (“POSITA”) at the time of the alleged invention.

23. I have been informed that claim terms are generally given their ordinary and customary meaning as understood by a POSITA in light of the specification and the prosecution history pertaining to the patent. I understand, however, that claim terms are generally not limited by the embodiments described in the specification.

24. I understand that, in addition to the claims, specification, and prosecution history, other evidence may be considered to ascertain the meaning of claim terms, including textbooks, encyclopedias, articles, and dictionaries. I have been informed that this other evidence is often less significant and less reliable than the claims, specification, and prosecution history.

B. Obviousness Under 35 U.S.C. § 103

25. I understand that a patent claim is invalid as obvious if the claimed invention would have been obvious to a POSITA at the time the claimed invention

was made. I understand that in an obviousness determination, the POSITA is presumed to have knowledge of all material prior art. I understand that whether a claim is obvious is based upon the determination of several factual issues.

26. In considering obviousness, I understand that one must determine the scope and content of the prior art. I understand that, in order to be considered prior art to a patent, a reference must be reasonably related to the claimed invention of that patent. A reference is reasonably related if it is in the same field as the claimed invention or is from another field to which a person of ordinary skill in the art would look to solve a known problem.

27. I understand that one must determine what differences, if any, existed between the claimed invention and the prior art. In evaluating whether such a claim would have been obvious, one may consider whether a reason has been identified that would have prompted a POSITA to combine the elements or concepts from the prior art in the same way as recited in the claimed invention.

28. I understand that one may consider whether (1) the change is merely the predictable result of using prior art elements according to their known functions or the result of true inventiveness; (2) there is some teaching or suggestion in the prior art to make the modification or combination of limitations claimed in the patent; (3) the innovation applies a known technique that had been used to improve a similar device or method in a similar way; (4) the claimed invention would have

been obvious to try, meaning that the claimed innovation is one of a relatively small number of possible approaches to the problem that could have been used by those of ordinary skill in the art with a reasonable expectation of success; (5) the invention merely substitutes one known element for another known element in order to obtain predictable results; (6) the invention merely applies a known technique to a known device, method, or product to yield predictable results; or (7) known work in the field may have prompted variations of use of the same inventions in the same or different fields due to market forces or design incentives that would have been predictable to a person of ordinary skill in the art.

29. I understand that one must be careful not to determine obviousness using the benefit of hindsight. I understand that the determination should be based on the position of a POSITA at the time the claimed invention was made and should not reflect what is known today or what is learned from the teaching of the patent.

30. I understand that certain secondary considerations may be considered in evaluating obviousness in order to prevent hindsight bias. These secondary considerations include the commercial success of products that practice the patent, long-felt need for the patented technology, failure by others to solve the problem addressed by the patent, initial skepticism by others in the industry, industry recognition and praise of the patented products, and efforts by others to copy the patented technology.

31. I have been informed that these secondary considerations of non-obviousness must be balanced against the strength of the *prima facie* case of obviousness. Where the invention represents no more than the predictable use of prior art elements according to their established functions, secondary considerations are inadequate to establish non-obviousness.

32. I have also been informed that the patentee must establish a nexus between any secondary considerations and the claimed invention. Where a secondary consideration results from something other than what is both claimed and novel in the patent, there is no nexus to the merits of the claimed invention.

33. I have been informed that, for example, the nexus requirement as it applies to evidence of commercial success requires that the patentee must show that the sales of a product that allegedly practices the claimed invention must be a direct result of the unique characteristics of the claimed invention, as opposed to other economic and commercial factors that are unrelated to the patented invention. I understand this to mean that if the product's commercial success is due to an element that existed in the prior art or that is not claimed by the patent, then no nexus exists.

34. I have been further informed that, for example, the nexus requirement as it applies to the long-felt need element must likewise be supported by evidence. Where the differences between the prior art and the claimed invention are minimal,

it cannot be said that any long-felt need was met by the patented invention or that the patent solved any technological problems that were unaddressed by others.

35. I have been informed that evidence of industry praise must also show a nexus with the claimed invention, and that self-serving statements by the patentee regarding the merits of the claimed invention are generally not given weight.

V. TECHNICAL BACKGROUND

36. DNS (Domain Name System) resolution is the process of translating human-readable domain names, such as “example.com,” into IP addresses (strings of numbers) that computers use to communicate. When a user seeks content from a particular web site, the user may type the website domain name into a web browser. Typically, the browser checks its local cache, followed by the operating system’s cache, to find the corresponding IP address. If the domain name is not cached, the query is sent to a DNS resolver, typically provided by the user’s Internet Service Provider (ISP) or a third-party service. The resolver acts as an intermediary, performing recursive queries by contacting DNS servers in a hierarchical order: root servers, top-level domain (TLD) servers (e.g., .com), and finally the domain’s authoritative DNS server (e.g., the “example” company server, often referred to as the origin server), which maintains the actual IP address of the requested domain name. Once the authoritative DNS server provides an IP address of the requested domain name, the resolver returns it to the user’s browser, allowing the browser to

connect to the web server hosting the domain. This conventional method of domain name resolution was well known in the art before October 2001 and is described by the '112 patent. *See* EX-1001, 3:13-54, Fig. 1; *see also* EX-1002, 1:1-20; EX-1011, 124-132; EX-1012, 622-630.

37. To improve efficiency and offload demand on the origin server, it was known to cache popular content at various locations on the network. *See* EX-1012 at 628-630 (“In theory at least, a single server could contain the entire DNS database and respond to all queries about it. In practice, this server would be so overloaded as to be useless.”); *see also id.* at 629 (“Once these records get back to the [] name server, they will be entered into cache there, in case they are needed later.”); EX-1011, 132 (“DNS extensively exploits caching in order to improve the delay performance and to reduce the number of DNS messages in the network.”); *id.*, 97-103 (“Multiple Web caches, located at different places in the Internet, can cooperate and improve overall performance.”).

38. In the DNS resolution process, caching can occur at each step (e.g., at the browser, OS level, and at the resolver) for a specific duration called Time-To-Live (TTL). EX-1012, 624-625, 629-630 (“[C]ache entries should not live too long. This is the reason that the *Time_to_live* field is included in each resource record.”); EX-1011, 132-134. As the '112 patent acknowledges, it was known to employ caching servers across a network to replicate popular content from an origin

server and for DNS queries to direct users to those content servers instead of an origin server. *See* EX-1001, 2:32-33 (“Fig. 2 is a depiction of a prior Web transaction using reverse-proxy caching”), 3:55-4:17 (“[T]he cache server 42 assumes the identity of the web server 12 so that a web request directed to the Web server 12 (in the example, ‘www.abcd.com’) is instead directed to the cache server 42.”), Fig. 2; EX-1002, 1:21-3:13 (“The introduction of caching can therefore markedly increase the apparent bandwidth of the original server, and provide redundancy for the original server.”), 3:19-26; *see also* EX-1011, 132 (“Given a cached hostname/IP address translation pair, if another query arrives to the name server for the same hostname, the name server can provide the desired IP address, even if it is not authoritative for the hostname.”); EX-1012, 622-630.

39. The purported improvement of the ’112 patent includes locating a cache server using anycast addressing. EX-1001, 5:33-6:6. Anycast addressing, however, was known in the art as of October 18, 2001, which I understand to be the claimed date of invention of the ’112 patent. Anycast addressing was a known network addressing and routing method in which multiple devices or servers share the same IP address. *See* EX-1002, 3:14-18, 4:10-5:7; EX-1011, 342 (“[A] new type of address, called an anycast address, has also been introduced [in IPv6], which allows a packet addressed to an anycast address to be delivered to any one of the a group of hosts.”); EX-1012, 442 (same). Traffic is routed to the nearest or best-performing

node based on routing protocols. EX-1002, 4:19-22; EX-1012, 442 (“For example, when contacting a group of cooperating file servers, a client can use anycast to reach the nearest one, without having to know which one that is.”).

40. The prior art, including Glines, demonstrates that it was known to use anycast addressing in DNS and content delivery networks (CDNs) to improve performance, reduce latency, and enhance redundancy. *See* EX-1002, 3:14-18, 4:10-5:7.

VI. THE '112 PATENT

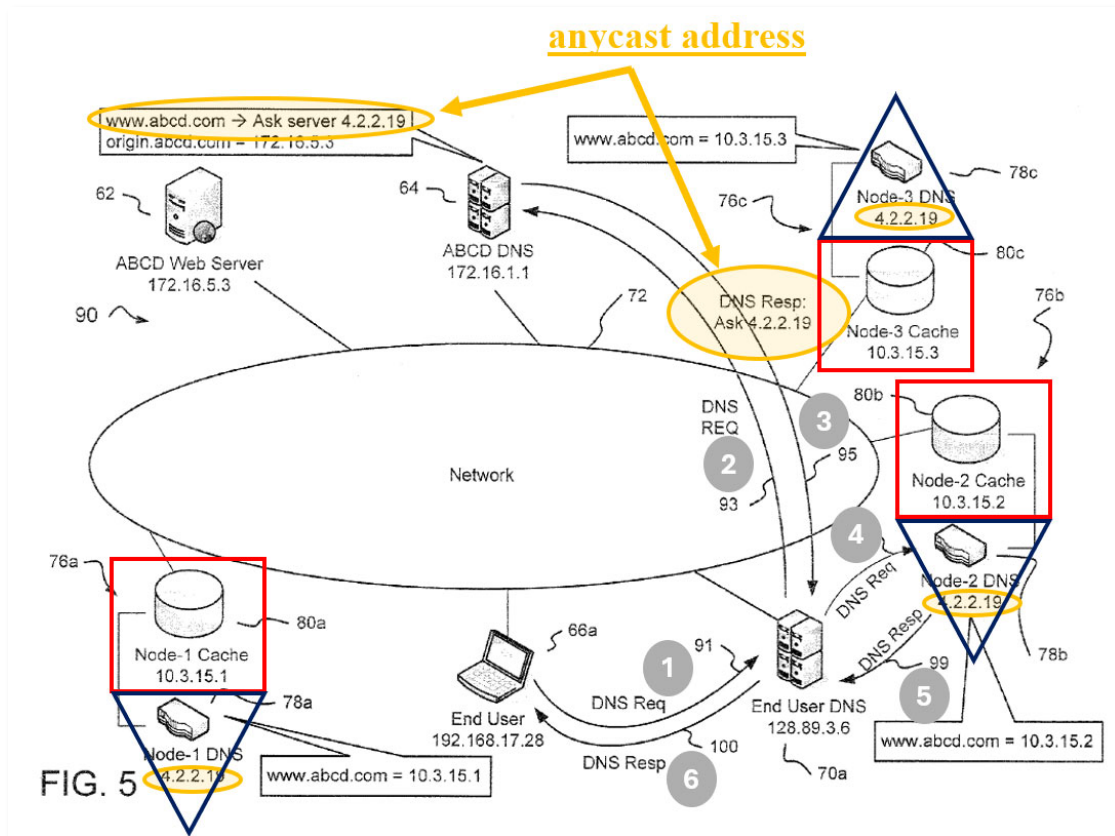
A. Overview of the '112 Patent

41. The '112 patent was filed on March 18, 2008 and claims priority to an earlier-filed application dated on October 18, 2001. For purposes of this declaration, I have assumed the '112 patent is entitled to claim October 18, 2001, as the date of invention.

42. The '112 patent discloses methods for “hosting and distributing content on a content delivery network such as the Internet.” EX-1001, 1:16-19. The patent discloses a content distribution network (CDN) that provides a content distribution service for a hostname (e.g., www.abcd.com), where content is delivered from **cache servers 80** (squares) instead of an original web server 62. *Id.*, 6:32-7:32, Fig. 5.

43. A CDN DNS system 78 made up of several CDN DNS servers 78a-78c, is made the authoritative server for a hostname. *Id.* To resolve the hostname, system

78 returns an anycast address associated with one or more CDN DNS servers 78a-c (triangles). *Id.* An anycast address is “a unicast address that is used in multiple places” and may “specify more than one interface” on a network. *Id.*, 5:32-58. For example, an anycast addresses “may be used to specify a group of one or more servers on a computer network.” *Id.*



Id., Fig. 5 (annotated).

44. The user may then request and receive the requested content from a cache server 80 (squares) whose CDN DNS servers 78 (triangles) responds to the anycast address (ovals). *Id.*, 6:32-7:32. This “enables distribution of the Web site among the cache servers 80 in their respective nodes and locations.” *Id.* In Figure 5

(above), I annotate these resolution steps by numerals 1-6 (in gray circles). *See also id.*, 6:32-7:32.

B. Summary of the Prosecution History of the '112 patent

45. I have reviewed the prosecution history of the application that led to the '112 patent. *See* EX-1007. During prosecution, the claims were rejected several times based on U.S. Patent Publication No. 2005/0010653 to McCanne, alone or in combination with other references. *See id.*, 338-349, 286-306, 244-258, 191-202, 156-170, 104-114, 64-76.

46. The applicant amended the claims in response to these office actions, including requiring the hostname to be resolved to the common anycast address “by an Internet Service Provider (ISP) DNS server.” *Id.*, 130-143. The applicant argued during prosecution that the prior art failed to teach “two DNS servers performing two resolves,” including a first resolution to the common anycast address by an ISP DNS server, and then a second resolution by a CDN DNS server identifying an IP address of a content server from which the user could request the content. *Id.*, 86-90. The applicant also filed an appeal brief to overcome the Examiner’s rejections, making these same arguments. *Id.*, 25-49; *see also id.*, 36-37 (“Neither McCanne nor Bahl teaches two DNS servers performing two resolves.”). In response to the appeal brief, the Examiner withdrew the rejections and allowed the claims without providing reasons for the allowance. *Id.*, 6-11.

47. The prior art presented below, including Glines (EX-1002), discloses the two DNS resolutions: a first resolving a request for content to an anycast address by an ISP DNS server, and a second resolving the user request to a particular content server peered to the name server responding to the anycast address. *See, e.g.*, Sections VIII.A., IX.A.2.

C. Claim Construction of Terms in the '112 patent

48. As discussed above, I have been informed that, for purposes of *inter partes* review, the standard for construction of terms in a patent's claims is the same as that applied in federal district court litigation.

49. I understand that a district court has yet to construe the claim terms of this patent. At this time, I see no need to construe any terms of the challenged claims for the purposes of this proceeding. Accordingly, in preparing my analysis I have applied the ordinary and customary meaning that each claim term would have had to POSITA in light of the '112 patent's specification and prosecution history.

50. If any claim terms are differently construed, I reserve the right to reassess my opinions under those constructions.

VII. PERSON OF ORDINARY SKILL IN THE ART

51. As discussed above, I have been informed that patentability must be analyzed from the perspective of a "person of ordinary skill in the art" in the same field as the patent-in-suit at the time of the invention. I also understand that several

factors are considered in assessing the level of ordinary skill in the art, including (1) the types of problems encountered in the art; (2) the prior art solutions to those problems; (3) the rapidity with which innovations are made; (4) the sophistication of the technology; and (5) the education level of active workers in the field.

52. I refer to a person of ordinary skill in the art here as a “POSITA.” A POSITA at the time of the invention would have had at least a bachelor’s degree in computer science, electrical engineering, or a related field, and at least two years of work or research experience in the field of content delivery management or networks. Work experience can substitute for formal education and additional formal education can substitute for work experience. I at least meet this definition of a POSITA, now and as of the priority date of the ’112 patent.

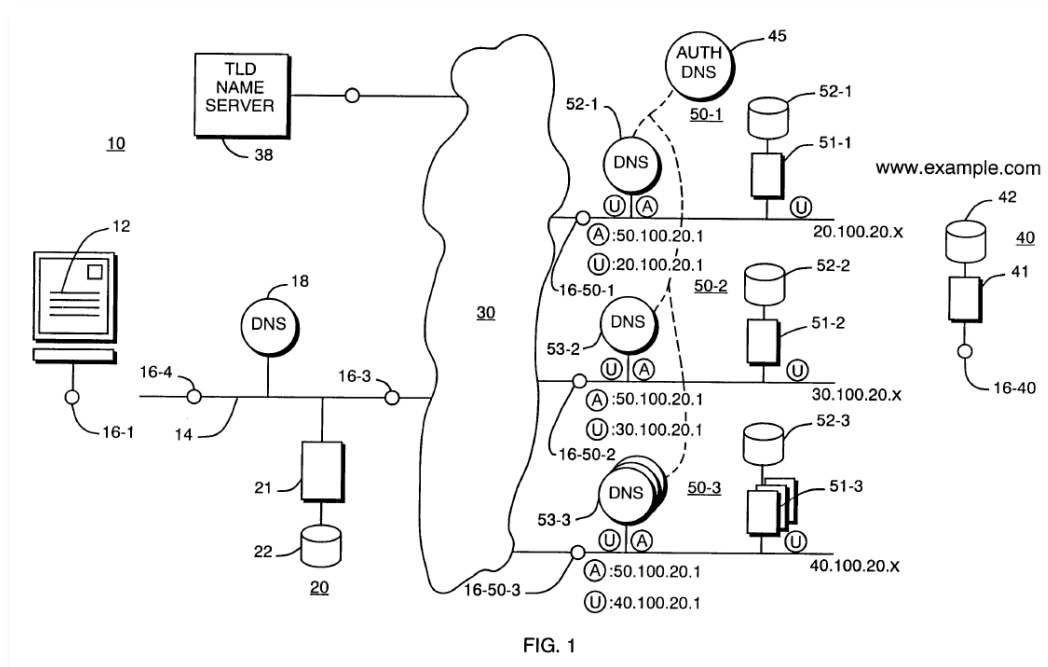
53. I am also familiar with the knowledge of a POSITA as of the priority date of the ’112 patent. I am able to opine on how a POSITA would have understood the disclosure and claims of the ’112 patent, the disclosures of the prior art, the various motivations to combine the prior art, and what combinations would have been obvious to a POSITA.

VIII. THE PRIOR ART

A. Overview of Glines (EX-1002)

54. Glines published on May 31, 2001, and claims priority to November 23, 1999. Glines, Cover.

55. Like the '112 patent, Glines discloses techniques for “redirecting client computer requests for content files to the closest replica of the requested content, by using anycast messaging.” Glines, Abstract, 11:1-13:14. Glines’s network includes replica content servers 50 that include DNS servers (or name servers) 53 and content servers 51. *Id.*, 11:1-13:14, Fig. 1. A user request to resolve a domain name is “forwarded as an anycast message” over the internetwork (e.g., Internet) 30 to the DNS servers 53, and the closest server 53 responds with a network address of an associated content server 51. *Id.*, Abstract, 11:1-13:14, 13:15-16:8, Figs. 1, 2. The user then requests the content from that content server 51. *See id.*



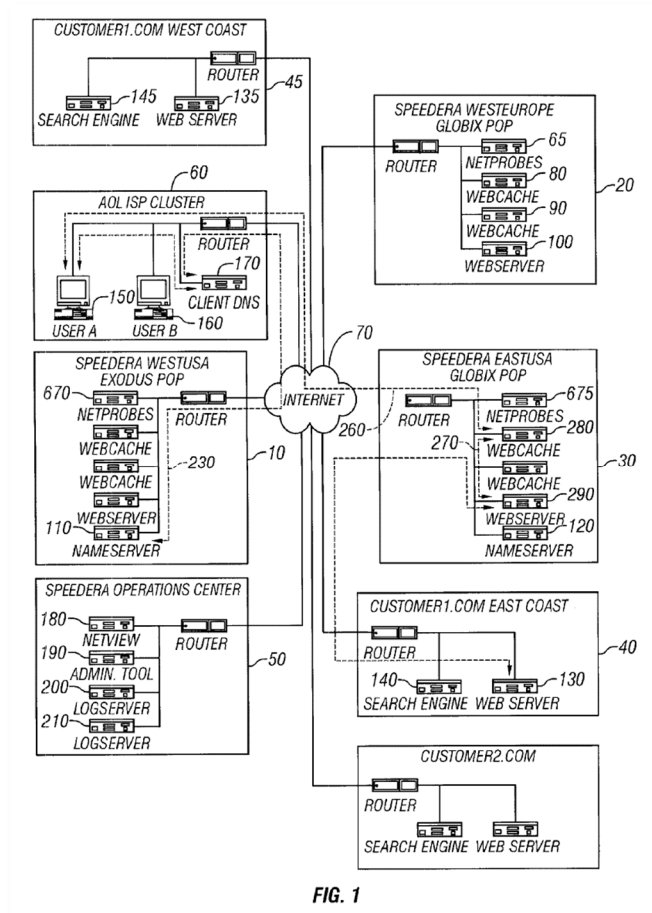
B. Overview of Gupta (EX-1003)

56. Gupta was filed on August 23, 2000, issued on June 11, 2002, and claims priority to November 22, 1999. Gupta, Cover.

57. Gupta discloses a network of point of presence (“POP”) servers. Gupta, Abstract, 2:9-3:8, Fig. 1. The servers share a hostname and determine an appropriate web (content) server to provide content in response to a request. *Id.* A “POP” is a point or physical location where two networks connect using, for example, a router. *See, e.g.*, EX-1013 at 4:65-5:54 (“The POP 114, 116 is the connecting point which separates the client 102 104, 106 from the network 100. In a public network 100, such as the Internet, the POP 114, 116 is the logical (and possibly physical) point where the public network 100 ends, after which comes the private hardware or private network of the client 102, 104, 106.”).

58. Gupta’s POP servers 10-30 are connected via computer network 70, such as the Internet. *Id.*, 4:45-67. The POP servers may include a name server (e.g., name server 110, 120) and multiple associated WebCache servers (e.g., WebCache servers in POP 10 and 30). *Id.*, 5:16-22, 10:16-22, 10:40-50, 11:1-7, Fig. 1. Using the POP network, content providers can provide “content such as HTML, images, video, sound, software, or the like on the network for fast and highly available access by clients (end users).” *Id.*, 2:9-17.

59. Gupta discloses that a user reaches the Internet via an Internet Service Provider (ISP) cluster 60 that includes a client DNS server 170. *Id.*, 5:65-6:9, 4:45-53, Fig. 1.



Id., Fig. 1.

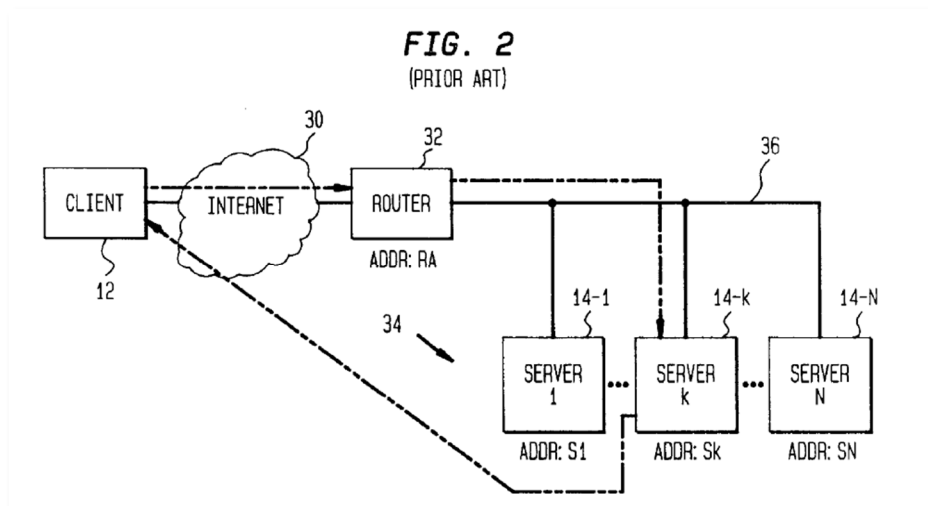
C. Overview of Chung (EX-1004)

60. Chung was filed on March 14, 1997 and issued on October 22, 2002. Chung, Cover.

61. Chung discloses the “explosive growth of the World Wide Web” and managing this growth by “deploying a set of machines, also known as a cluster, and

configuring the machines to work together to host a single service.” Chung, Abstract, 1:15-40, 2:33-50. “Such a server cluster should preferably publicize only one server name for the entire cluster so that any configuration change inside the cluster does not affect client applications.” *Id.*

62. Chung responds to user requests for content via router 32, which “dispatches the request to a selected server 14-k of server cluster 34 based on load characteristics.” *Id.*, 2:41-3:10, Fig. 2. The “selected server 14-k responds to the client request by sending reply packets over the established TCP/IP connection, as indicated by the dashed line connecting server 14-k to client 12.” *Id.*



Id., Fig. 2.

D. Overview of Chiou (EX-1005)

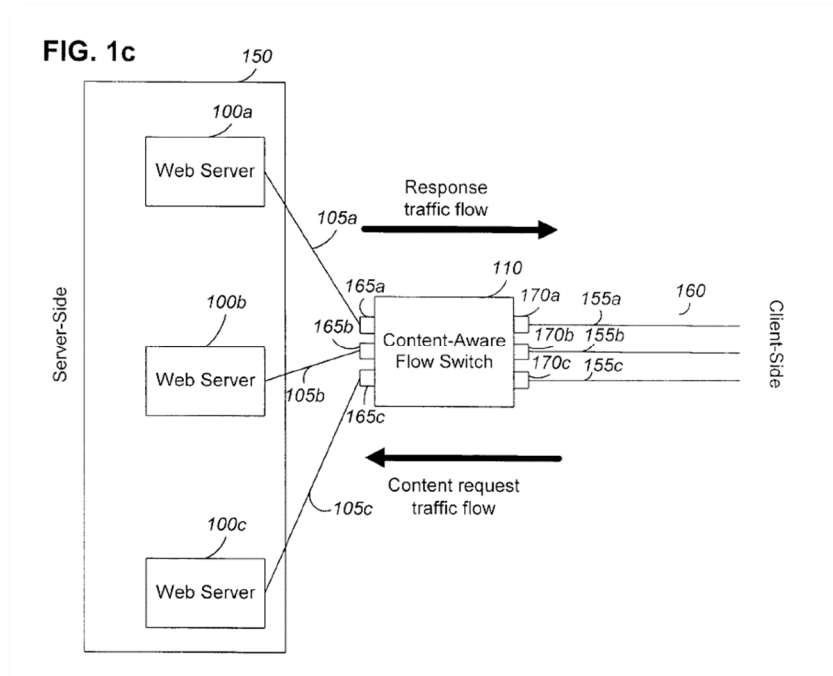
63. Chiou was filed on December 14, 2000 and issued on September 14, 2004. Chiou, Cover.

64. Chiou discloses a system for updating a content cache to prevent stored content from becoming out of date or “stale.” Chiou, Abstract, 1:28-40, 2:56-58. Chiou’s cache management method includes receiving a request for content, determining if there is a cache hit for the content in memory, and then determining if the stored data is “valid data” before returning the requested content. *Id.*, 12:52-13:40, Fig. 4. If a cache hit does not occur, the requested content is read from a “target” disk and invalid data is purged from the cache. *Id.*, 9:29-35, 13:18-40.

E. Overview of Colby (EX-1006)

65. Colby was filed on March 30, 1998, issued on December 21, 1999, and claims priority to August 1, 1997. Colby, Cover.

66. Colby discloses a “content-aware flow switch.” Colby, Abstract, 2:48-58, 5:43-6:28, Fig. 1c. The switch directs user content requests to a server associated with the switch “that is well-suited to serve the content request.” *Id.*, 2:48-58. The server is selected, for example, “based on the type of content requested, ... , the degree of load on available servers, network congestion information, and the proximity of the client to available servers.” *Id.*, Abstract, 2:48-58; *see also id.*, 5:43-6:28, Fig. 1c.



Id., Fig. 1c.

IX. OPINIONS ON UNPATENTABILITY

67. The following combinations of prior art, in my opinion, render the claims of the '112 patent obvious under 35 U.S.C. § 103:

Ground	Prior Art	Basis	Claims
1A	Glines (EX-1002)	§ 103	1-8 and 10-23
1B	Glines (EX-1002) and Gupta (EX-1003)	§ 103	1-8 and 10-23
2A	Glines (EX-1002) and Chung (EX-1004)	§ 103	4
2B	Glines (EX-1002), Gupta (EX-1003), and Chung (EX-1004)	§ 103	4
3A	Glines (EX-1002) and Chiou (EX-1005)	§ 103	7-9, 19, and 20

Ground	Prior Art	Basis	Claims
3B	Glines (EX-1002), Gupta (EX-1003), and Chiou (EX-1005)	§ 103	7-9, 19, and 20
4A	Glines (EX-1002) and Colby (EX-1006)	§ 103	10-22
4B	Glines (EX-1002), Gupta (EX-1003), and Colby (EX-1006)	§ 103	10-22
5A	Glines (EX-1002), Chiou (EX-1005), and Colby (EX-1006)	§ 103	19 and 20
5B	Glines (EX-1002), Gupta (EX-1003), Chiou (EX-1005), and Colby (EX-1006)	§ 103	19 and 20

A. Grounds 1A and 1B Render Obvious Claims 1-8 and 10-23

1. Motivation to Combine Glines and Gupta

68. As further discussed below, it would have been obvious to combine Glines and Gupta.

69. For example, both Glines and Gupta disclose methods of delivering content over the Internet. *See* Glines, 1:4-21, 5:24-6:7 (“[C]ontent distribution is provided as a service where client requests are automatically routed to the closest available content server. The performance of the system as a whole may be increased by simply adding more server resources to the parts of the network that need it, with minimal change or no change to the network existing infrastructure.”); Gupta, 1:41-2:17, 2:23-64 (“[T]he POP network selects the optimal server/servers for the user based on rules defined in the POP network configuration. [] These rules define

how criteria such as POP/customer server health, latency from POP/customer servers to the user making the request, packet loss experienced at each POP, cost associated with each POP/customer server, and the like.”).

70. Both Glines and Gupta disclose, for example, resolving DNS requests and identifying the most efficient server from which to serve content. Glines, 9:9-11:12; Gupta, 9:1-13, 12:13-37.

71. Glines and Gupta also share a similar architecture to one another—both disclose DNS servers and associated content servers that are connected by a network and function together to resolve hostnames and deliver content. *See* Glines, 11:1-13:14, Fig. 1; Gupta, 4:45-5:19, 5:54-6:21, Fig. 1.

72. Given their similarities and shared desire to increase content delivery performance, a POSITA would have understood that Glines and Gupta to be analogous. A POSITA would have accordingly had a reasonable expectation of success in their combination given both teach well-known methods of delivering content over the Internet.

73. The result of the combination would have been applying Gupta’s use of multiple WebCache servers with each name server to Glines’s system to yield multiple content servers associated with *each* name server in Glines. As discussed below, doing so would have allowed more content to be stored in association with each name server, improving Glines’s ability to deliver content to a user via the

closest name server. This also would have improved Glines's ability to manage high request volumes without degrading performance, for example by load balancing across multiple servers during periods of high traffic. This also would have improved Glines's ability to manage server faults at each name server location—other associated content servers would handle content requests and ensure uninterrupted service. This also would have ensured a name server in Glines remains operational during scheduled maintenance or unexpected outages, and avoided single points of failure.

74. The combination further would have applied Gupta's use of an ISP DNS server to Glines to resolve hostnames as part of the content delivery process for residential Internet users, to the extent not taught or rendered obvious by Glines alone. As discussed below, a POSITA would have been motivated to implement this modification to ensure compatibility with "typical" Internet service architecture, and a POSITA would have had an expectation of success in the combination.

2. Claim 1

a. [1p] "A method of content delivery in a network, the method comprising:"

75. Glines, alone or in combination with Gupta, satisfies this limitation by disclosing methods of delivering content, including content files, over a network that includes the Internet.

76. For example, Glines “redirect[s] client computer requests for *content files* to the closest replica of the requested content, by using anycast messaging.” Glines, Abstract, 1:4-2:3.¹ Glines “replicat[es] copies of popular *content* and locat[es] the copies throughout various locations in the network” to reduce original-server demand. *Id.*, 2:4-3:3; *also id.*, 5:24-8:23 (disclosing “increas[ing] performance in the delivery of requested content files in networked computer environments.”), 9:9-21 (disclosing “client computer 12 run[ning] a web browser program that enables a user to submit requests for content files”), 10:8-11:5 (overcoming “difficulties . . . with th[e] standard content delivery scheme”), 12:1-3 (“content files may include web pages, multimedia files, graphics, pictures, other computer files that are suitable for network transmission using well know protocols such as the HyperText Transfer Protocol (HTTP)”). Glines’s system improves existing methods where “the number of requests for *content* files (i.e. ‘hits’) exceeds the ability of a specific server to handle the demand.” *Id.*, 1:4-2:3.

77. Glines delivering “content” and “content files” from a content server using network protocols such as HTTP discloses “a method of content delivery in a network,” as claimed.

78. Glines’s content is delivered “in a network.”

¹ All emphases added unless otherwise noted.

79. For example, Glines generally discloses delivering content over the Internet. *See id.*, 1:4-3:3 (discussing known methods of delivering content via the Internet), 3:19-5:7 (same), 6:24-6:2 (describing a need “to provide for increased performance in the delivery of requested content files in networked computer environments”), 6:3-7:13 (disclosing that a “name server responding to the anycast returns an **Internet Protocol (IP) address** which is the unique public unicast address for its associated content server that contains and/or is able to honor the request for the specified content file.”), 8:14-23 (using Internet routing).

80. With respect to Figure 1, Glines’s content is delivered over, for example, network 10, [internetwork 30](#) (e.g., the Internet), local area network (LAN) 14, and the “common local area network segment[s]” of replica content servers 50-1, 50-2, 50-3, each of which is a network as claimed. Glines, 1:4-3:3, 3:19-5:7, 6:24-6:2, 6:3-7:13, 8:14-23, 9:9-15 (“The network 10 may be a typical client-server distributed system, such as is now popularly implemented using personal computer platforms and web based file 15 servers connected via an internetwork 30 such as the Internet.”), 11:14-28 (“[T]he client computer 12 is connected to a **local 25 area network (LAN) 14** that consists of **internetwork devices 16-3 and 16-4**, which in this instance are routers.”), 13:3-8, 15:15-25

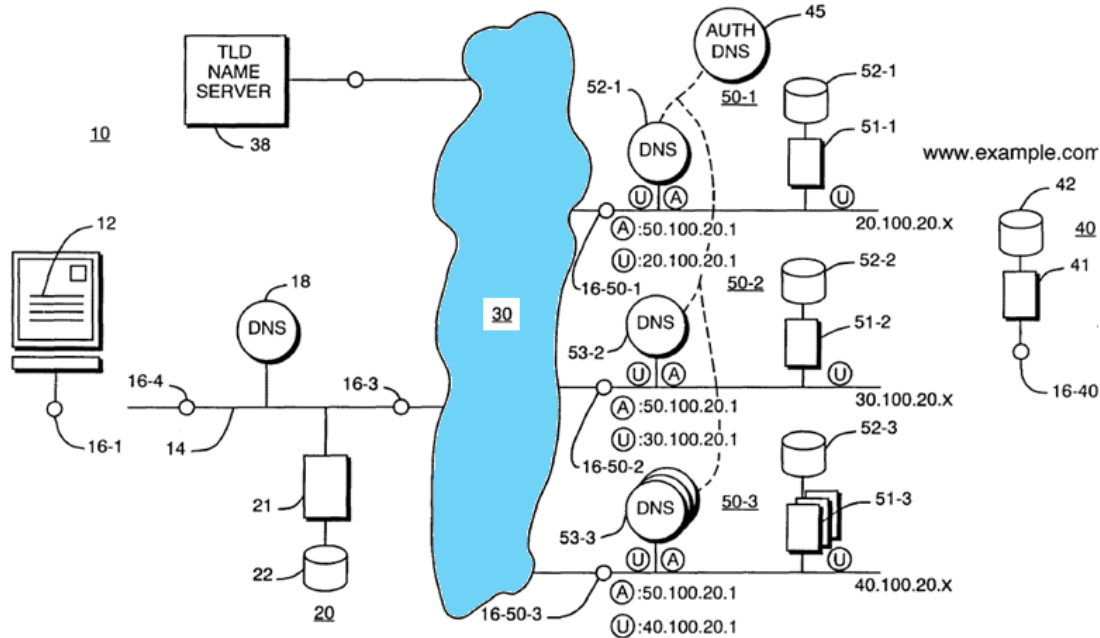


FIG. 1

Id., Fig. 1 (annotating internetwork 30).

81. Glines therefore satisfies a method of content delivery in a network, as claimed. Combining Glines with Gupta would not alter this analysis.

b. [1a] “(A) providing a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), the plurality of CDN DNS servers sharing a common anycast address, and where each CDN DNS server is associated with a respective plurality of content servers;

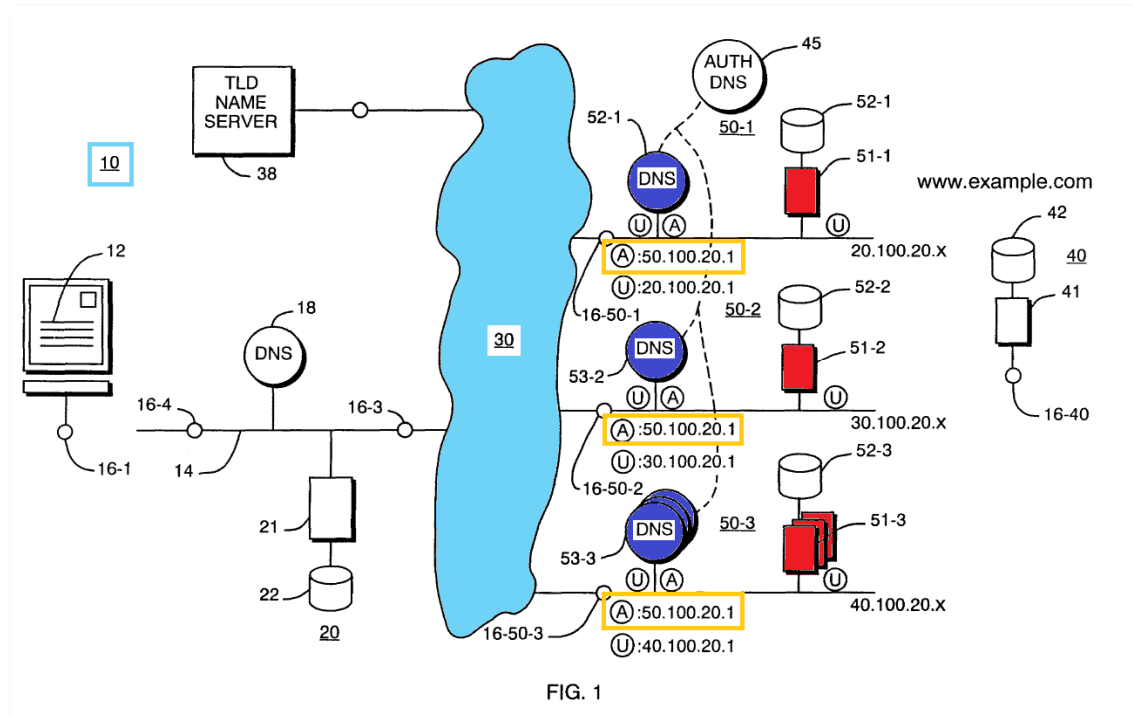
82. Glines, alone or in combination with Gupta, would have rendered limitation [1a] obvious.

83. For example, Glines provides “a group of name servers ... located throughout [a] network.” Glines, 6:9-22, 9:9-12. The name servers are DNS servers.

84. Glines's computer network 10, including internetwork 30, local area network 14, and name server local area networks, is a Content Delivery Network (CDN). Computer network 10 (square, below) "redirect[s]s client computer requests for content files to the closest replica ... to a name service provided by a group of name servers distributed in the network." Glines, 9:9-15; *also id.*, 11:14-22 (personal computer 12 connecting to network 10 and internetwork 30).

85. Each of Glines's name servers is "addressable via a common anycast address" as claimed. *Id.*, 6:9-22, 3:4-18, 4:10-5:8, 6:8-22, 7:9-8:23. Glines further discloses that its name servers 53 "advertise themselves as reachable at an anycast address (rectangles, below)," such as 50.100.20.1. *Id.*, 11:1-12, 12:24-13:2, 13:9-14, 14:17-15:14; *also id.*, 9:9-15, 16:15-17:2

86. Each of Glines's name server (filled circles, below) "is peered with a nearby content server [filled rectangles, below] ... that contains (or can serve) desired content file replicas via unicast addressing." *Id.*, 6:9-22; *also id.*, 7:1-18.



Id., Fig. 1 (annotated).

87. Therefore, Glines provides a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), the plurality of CDN DNS servers sharing a common anycast address, where each CDN DNS server is associated with one or more content servers.

88. Glines also discloses providing CDN DNS servers that are associated with a respective plurality of content servers, as claimed.

89. Glines “associat[es] multiple replica content servers 50-1, 50-2, ... 50-3, with the origin server 40” to serve content associated with the origin server. Glines, 10:13-18.

90. Each replica content server includes a name server, including “name servers 53-1, 53-2, 53-3.” *Id.*, 10:13-25. The name servers 53-1, 53-2, 53-3 are “peered with content file servers 51-1, 51-2, 51-3 that contain replica copies of files stored at the origin server 40.” *Id.*; *also id.*, 12:11-23, 12:24-13:8 (“Each name server 53 and file server 51 associated with particular content server 50 are considered to be connected in a peering arrangement.”), 6:9-22 (“Each name server is peered with a nearby content server, cache server, or other file server that contains (or can serve) desired content file replicas via unicast addressing.”), 7:1-18 (“[T]he group of name server/content server peers [] share a common anycast address” and “every router that is connected to one of the name server/content server pairs is advertising a route to the group address.”).

91. Glines depicts content servers 51-1, 51-2, and 51-3 as being addressable by unicast addresses “40.100.20.x,” “30.100.20.x,” and “20.100.20.x,” respectively. *Id.*, Fig. 1; *also id.*, 6:8-22 (“The name server responding to the anycast returns an Internet Protocol (IP) address which is the unique public unicast address for its associated content server that contains and/or is able to honor the request for the specified content file.”), 7:18-8:2 (providing “the unique (unicast) address of the associated content replica server”), 13:9-13 (same), 15:26-16:2 (same).

92. The variable “x” in this unicast address is the host part of these IP addresses, which is meant to identify particular machines on the network associated

with the content servers (e.g., “40.100” “30.100,” and “20.100.”). *See* EX-1012, 416-418. This signifies more than one machine associated with the content servers (e.g., “40.100” “30.100,” and “20.100.”), and at least suggests each name server 53-3 has a plurality of associated content servers. *Id.*; Glines, 8:19-23 (disclosing “[e]ach name server return[ing] the IP address of **one or many** (such as through round robin or other mechanisms) closely bound and topologically nearby **content servers**”)

93. Further, Glines refers to each name server-content server group as a “service group” and discloses functionality that would have been relevant only if more than one content server was in each group. Glines, 16:15-17:2, 17:20-19:16. For example, Glines discloses fallover or fallback policies to avoid issues with content server outages or overloads in each group. *Id.*, 17:20-19:16 (describing solutions “[i]f a content server in the service group fails”). This includes “provid[ing] redundancy in the service group” by “return[ing] **two or more IP** address[es] in the service group” in response to the anycast message. *Id.*, 18:20-19:3. Load-balancing techniques “within the service group” are also described using “layer 4 switches.” *Id.*, 19:10-16 (“[W]ithin the service group, standard load balancing and replication techniques can also apply, such as multiple content servers (returning multiple local IP addresses to a name lookup), layer 4 switches, etc.”). Glines also discloses “map[ping] the lookup of [a] name into a local IP address of a **member** of the service group.” *Id.*, 16:28-17:2.

94. A “member” of a service group teaches more than one content server for each name server.

95. Glines also discloses that “[e]ach name server returns the IP address of *one or many* (such as through round robin or other mechanisms) closely bound and topologically nearby *content servers*” when resolving a request for content via an anycast message. Glines, 8:19-23. By each name server returning one or many IP addresses of closely bound content servers, or selecting a particular content server using “round robin” selection (e.g., selecting the next server in a sequence if a first server is not available), Glines teaches or at least suggests that each name server is associated with more than one content server 51.

96. Glines therefore teaches *each* name server in its network being paired with *multiple* content servers.

97. **To the extent Glines does not expressly disclose each name server associated with a plurality of content servers 51 (it does), it would have been obvious to implement and/or modify Glines in this manner, just as Glines suggests.**

98. First, a POSITA would have known, for example, that having multiple content servers associated with each name server in Glines would have improved the performance, reliability, and scalability of Glines’s network.

99. For example, associating multiple content servers with each name server would have increased storage capacity at each name server location and would have allowed more content to be stored and managed by each name server. This would have increased the ability of Glines's system to retrieve content from a name server that is closest to the user, as more content would be stored at each name server, thereby improving the ability to replicate content across more locations and increase system performance. This would have reduced system latency in Glines.

100. Indeed, this concept was well known and understood in the prior art and would have been obvious to apply to Glines:

[w]ith the explosive growth of the World Wide Web, many popular Internet web sites are heavily loaded with client requests As a result, clients may experience slow response times and may be unable to access certain web sites. Upgrading the servers to more powerful machines may not always be cost-effective. Another common approach involves *deploying a set of machines*, also known as a cluster, and configuring the machines to *work together to host a single service*. Such a server cluster should preferably publicize only one server name for the entire cluster so that any configuration change inside the cluster does not affect client applications.

Chung (EX-1004) at 1:16-40.

101. Second, employing multiple content servers would have improved each of Glines's name servers' ability to manage higher request volumes without degrading performance.

102. For example, with multiple content servers, each name server-content server group would have been able to load balance requests for content across multiple content servers, particularly during periods of high traffic. Glines discloses this technique expressly: “within [a] service group, standard load balancing and replication techniques can also apply, such as multiple content servers (returning multiple local IP addresses to a name lookup), layer 4 switches, etc.” Glines, 19:13-15.

103. Third, a POSITA would have recognized that, by deploying a plurality of content servers with each name server, each name server service group would have been able to manage individual content server faults.

104. For example, if one content server fails, others would continue to handle content requests and ensure uninterrupted service. This would have ensured a name server remains operational during scheduled maintenance or unexpected outages, and avoided single points of failure. *See, e.g.*, Chung (EX-1004), 4:3-28 (“[E]venly distribut[ing] the client request load among the various servers of the cluster, mask[s] the failure of any server or servers of the cluster by distributing client requests to the remaining servers without bringing down the service, and permit[ting] additional servers to be added to the cluster without bringing down the service.”).

105. Thus, a POSITA would have been motivated to modify Glines in this manner at least to improve performance, improve load balancing, and better manage sever faults.

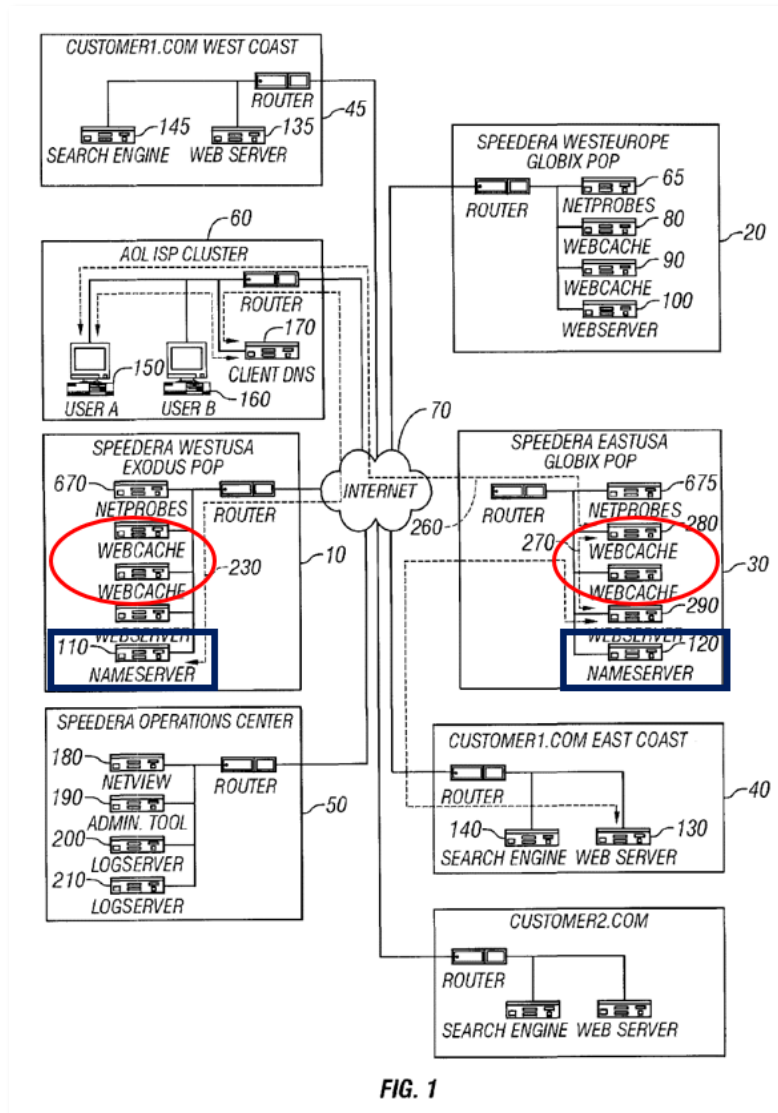
106. Modifying Glines such that each name server 53 would be associated with multiple content servers 51 would have been well within the capabilities of a POSITA. Indeed, this would have been nothing more than combining known features in the prior art according to known methods (e.g., applying Glines's teaching of multiple content servers 51-3 for name server 53-3 and its suggestion to load balance within service groups to the rest of its name server service groups) to yield predictable results. Doing so would have been nothing more than combining known replication and failover techniques in the prior art to yield predictable results.

107. The result would have been more than one content server for each name server, which is a common and understood practice in the art to increase storage, performance, and reliability. As discussed above, Glines teaches or at least suggests this modification. *See* Glines, 6:8-22, 7:18-8:2, 13:9-13, 15:26-16:2, Fig. 1 (depicting more than on content server 51-3 for name server 53-3, and referencing unicast addresses "40.100.20.x," "30.100.20.x," and "20.100.20.x" for content servers 51-3, 51-2, and 51-1), 19:13-16 ("standard load balancing and replication techniques" for each service group).

108. This would have been well within the capabilities of a POSITA and the result would have been more than one content server for each name server, which is a common practice to increase storage, performance, and reliability. A POSITA therefore would have had a reasonable expectation of success in the combination.

109. **It also would have been obvious to combine Glines and Gupta to deploy each name server with an associated plurality of content servers 51.**

110. Gupta discloses multiple “point of presence” (POP) servers that include a name server and WebCache servers. Gupta, 4:45-67, 5:16-53, 6:10-21, 10:4-22, Fig. 1. Gupta’s POP servers 10 and 30 each include multiple **WebCache servers** (ovals) with an associated **name server** (rectangles). *Id.*



Id., Fig. 1 (annotated).

111. It would have been obvious to combine and apply Gupta's teaching of multiple WebCache servers with each name server to Glines's system, to the extent Glines alone does not teach or render this feature obvious. Indeed, like Gupta, Glines discloses that its content servers 51 may be or include "*cache* server[s]." Glines, 6:8-15 ("Each name server is peered with a nearby content server, cache server, or

other file server that contains (or can serve) desired content file replicas via unicast addressing.”), 8:4-6 (“In order for this process to operate properly, each physical system which shares the anycast address must perform certain functions, such as a standard but customized name server function, as well as a content server, *such as a cache server.*”). Gupta’s WebCache servers and Glines’s content servers 51 were therefore compatible.

112. As discussed with respect to Glines, employing multiple WebCache servers, as taught by Gupta and suggested by Glines, would have improved the ability of each name server to handle manage high request volumes without degrading performance. For example, each name server would have been able to load balance requests across those servers during periods of high traffic. This also would have improved managing server faults without significant performance degradation. For example, in the event of a content server failure, other associated content servers would handle content requests and ensure uninterrupted service. This also would have ensured a name server remains operational during scheduled maintenance or unexpected outages, and avoided single points of failure.

113. Associating multiple content servers with each name server in Glines’s system also would have increased Glines’s storage capacity and allowed more content to be stored at each name server. This would have increased the ability for

users to retrieve content from the nearest name server in Glines's system, thereby increasing system performance by reducing latency.

114. Therefore, a POSITA would have been motivated to combine Glines with Gupta in this manner to improve performance, reliability, and scalability of Glines's network.

115. Combining Gupta's use of multiple WebCache servers with Glines's name servers would have been nothing more than combining known elements in the prior art according to understood principles to yield predictable results.

116. Glines's system would have benefited from multiple content servers at each name server location, to the extent not already disclosed or rendered obvious, by enabling load balancing, redundancy, and failure tolerance at these locations, which both Glines and Gupta contemplate.

117. Indeed, Gupta discloses load balancing across its network and accounting for latency of its web cache servers. Gupta, 3:26-51, 7:2-20, 8:12-48, 10:4-22. As discussed, Glines also discloses load balancing across content servers: "within [a] service group, standard load balancing and replication techniques can also apply, such as multiple content servers (returning multiple local IP addresses to a name lookup), layer 4 switches, etc." Glines, 19:13-15; *see also id.*, 18:20-19:16 (discussing replication techniques and failure tolerance). Given the similarities and

overlap of Gupta and Glines, a POSITA would have had a high expectation of success in the combination.

118. Combining Glines with Gupta in this way would not have altered the features of Glines discussed elsewhere in my declaration.

- c. **[1b] “(B) causing said plurality of CDN DNS servers to be authoritative for a hostname associated with a content provider by causing said common anycast address to be associated with said hostname;”**

119. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

120. Glines’s client enters a URL into a browser, which “will typically contain a domain name” (e.g., a hostname). Glines, 9:16-10:4. Glines discloses “www.example.com,” as an exemplary hostname associated with a content provider. *Id.*, 6:23-7:13, 13:15-15:10, Figs. 1, 2.

121. This corresponds with the ’112 patent. *See* EX-1001, 3:24-54 (listing hostname as www.abcd.com).

122. In conventional DNS systems, the root DNS name server 38 would return the IP address of the authoritative DNS server for www.example.com that would include, for example, “the four-digit address 62.104.11.12 associated with a particular origin server 40” associated with that domain. Glines, 14:8-16.

123. Using Glines’s system, however, “replica content servers 50 may [instead] contain replicas of one or more content files that originate at the origin

server 40.” *Id.*, 9:16-10:25. The browser is thus “redirected to the associated nearest cache server 50-1, 50-2, 50-3 that can honor the remaining expected HTTP request message sequence, without adding any unnecessary traffic to the associated domain name server 53 and/or paths in the internetwork 30 to the origin server 40.” *Id.*, 11:1-12; *see also id.*, 11:29-12:13:2, 14:13-15:4, Fig. 1.

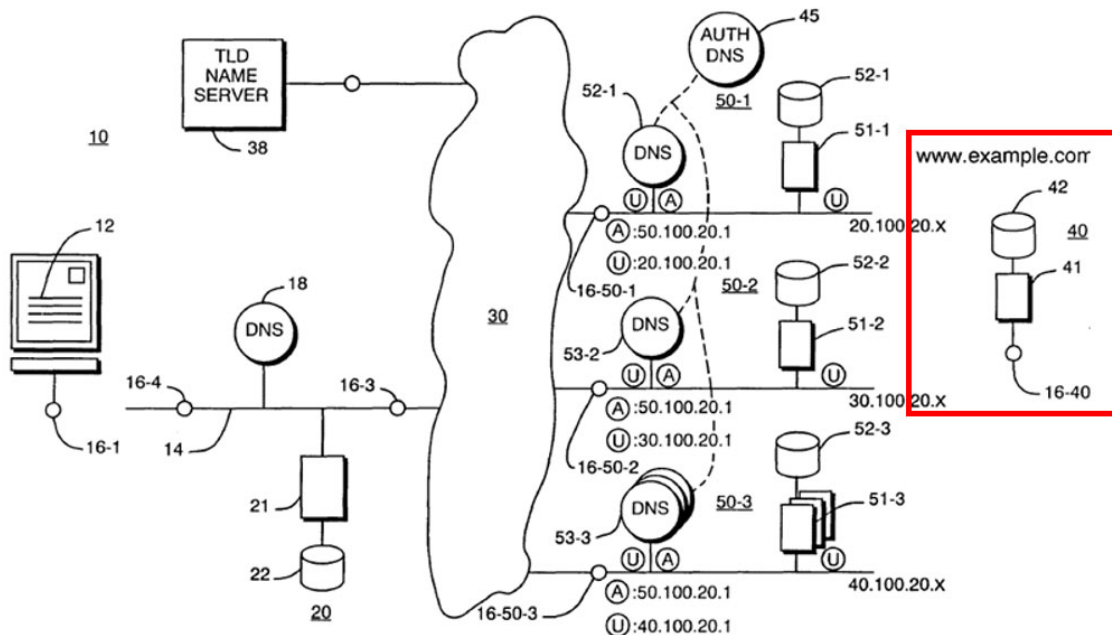


FIG. 1

Id., Fig. 1 (annotated).

124. Glines’s resolving a URL in “www.example.com” discloses a hostname associated with a content provider, as claimed.

125. Glines also teaches that each of its name servers 53 (e.g., the claimed CDN DNS servers) is “authoritative” for any domain name (hostname) for which the name server’s associated content servers have replica content. For example,

Glines discloses that “[e]ach name server 53 is considered to be an authoritative DNS resolver for domain names [(e.g., hostnames)] associated with the replica content files stored in its associated replica content server 51.”). Glines, 13:9-14; *see also id.*, 6:8-15 (“Each name server . . . advertises itself as being authoritative for the domains associated with the content files stored (or to be served from) the associated content server.”), 12:24-13:2, 14:17-25.

126. To be authoritative for `www.example.com`, the replica name servers 53 “have received appropriate information from an authoritative DNS 45 for the domains in server 40.” *Id.*, 12:24-13:2. “Specifically, the name servers 53-1, 53-2, 53-3 have been designated as being the authoritative name servers for ‘example.com’ through previous network management level configuration information.” *Id.*, 14:17-25.

127. Glines therefore discloses CDN DNS servers that are authoritative for a hostname associated with a content provider.

128. Glines also discloses causing the common anycast address to be associated with said hostname.

129. When a user enters “`www.example.com`” into a browser in Glines, the local resolver 18 “consult[s] public name servers that have been defined as being authoritative for the ‘example.com’ domain.” Glines, 7:1-8. Rather than returning the address of the origin server associated with the “example.com” domain, the

resolver “instead return[s] the anycast address 50.100.20.1,” which is associated with the name servers. Glines, 14:13-25; *also id.*, 15:1-4 (“[T]he primary name service listed by organizations responsible for maintaining the state of internetwork 30, such as the Internic, will point to this common address 50.100.20.1 of the caching servers 50, instead of the origin server 40.”); *also id.*, 7:1-8 (programming root domain name servers to return the common anycast address), 12:24-13:2, 13:9-11.

130. Glines’s Figure 2, annotated below, illustrates this process

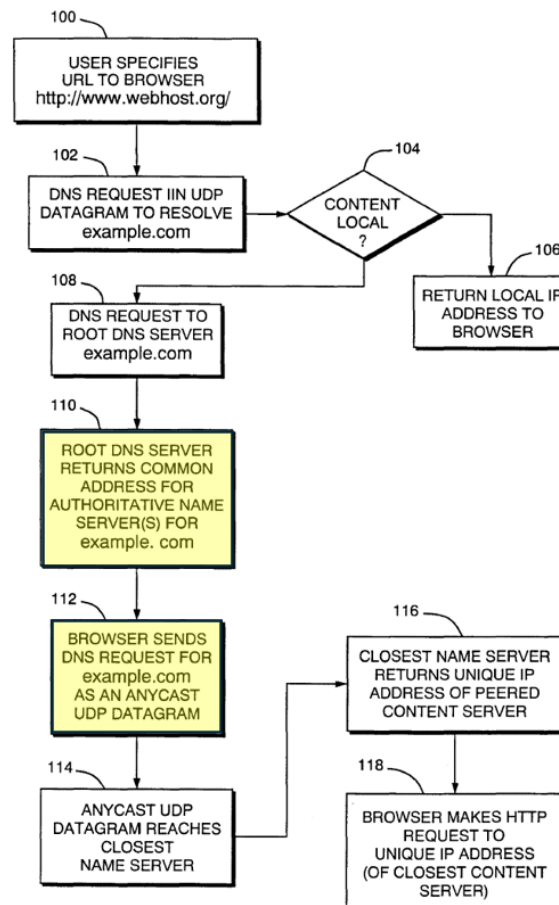
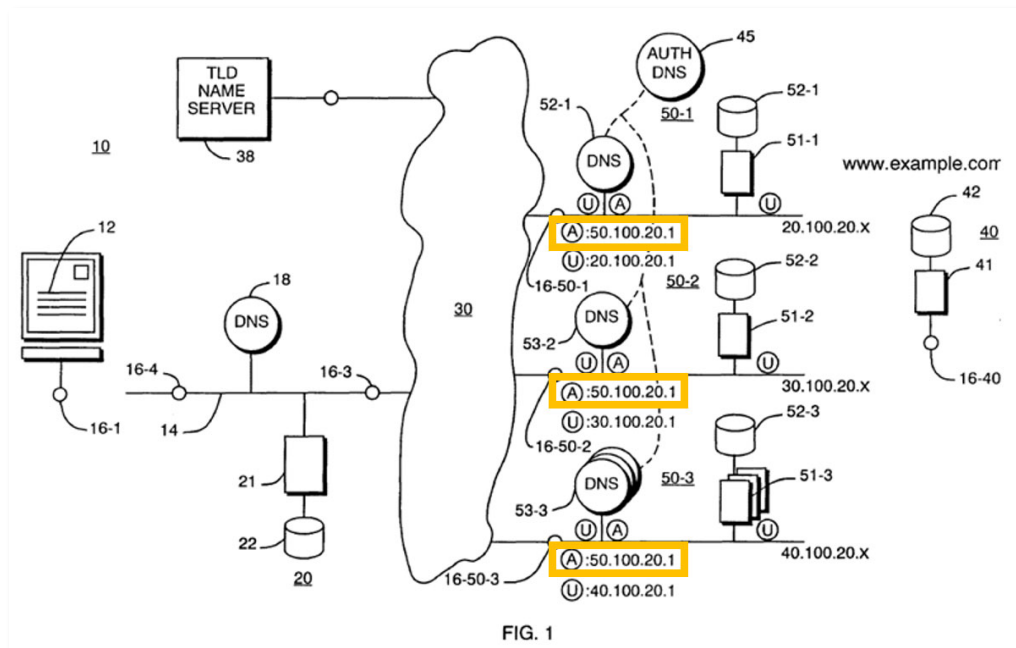


FIG. 2

Id. at Fig. 2 (annotated).

131. Indeed, Glines discloses that “[t]he name servers 53-1, 53-2, 53-3 are addressable via both a common anycast address as well as a unique or unicast address,” and “a DNS request may be sent to the name servers 53 as an anycast datagram.” *Id.*, 12:24-13:2; *see also id.*, 13:9-11 (same).

132. As shown in Figure 1, each of the authoritative name servers 53-1, 53-2, 53-3 is addressable by the common anycast address “50.100.20.1” to resolve www.example.com. *See id.*, Fig. 1.



Id., Fig. 1 (annotated).

133. Combining Glines with Gupta would not alter this analysis.

d. [1c] “(C) responsive to a request for content associated with the content provider and issued by a client, said request including at least said hostname,”

134. Glines, alone or in combination with Gupta, would have rendered this limitation obvious. For example, Glines discloses a user requesting a website (e.g., a request for content associated with the content provider issued by a client) via a URL (e.g., a request that includes the hostname).

135. Glines discloses that a user operating client computer 12 “runs a web browser program that enables a user to submit requests for content files that are nominally located at an origin file server 40.” Glines, 9:9-22; *also id.*, 9:9-15, 11:14-12:10 (“[T]he environment 10 consists of a client computing device, such as a computer 12, that is running a data file retrieval program such as a web browser”). The user’s request for a content file nominally located at an origin server is a request for content associated with the content provider issued by a client as claimed.

136. Glines’s client issues the request in the form of a Uniform Resource Locator (URL) that “will typically contain a domain name, a file name for the content file, and an access method.” Glines, 9:9-10:4, 13:15-14:25; *see also* 6:23-7:13, Fig. 2.

137. A POSITA would have understood Glines’s “domain name” to be a hostname. EX-1011, 124-125 (describing, for example, “cnn.com” and resolving this hostname via a domain name system); EX-1012, 623-624 (stating that domains

may represent “a single host, or it may represent a company and contains thousands of hosts”). “The client browser 12 then makes a request to discover the correct Internet Protocol (IP) address of the origin server 40 that contains the requested content file.” Glines, 9:9-10:4. For example, Glines discloses requesting content from `www.example.com` (a hostname) and specifying the URL “`http://www.example/homepage.html`.” *Id.*, 13:15-14:25. The request accordingly includes a hostname as claimed.

138. Glines also discloses that its system is “responsive” to the user’s request. For example, in response to the user’s request, the root DNS name server 38 is “programmed to . . . return the anycast address 50.100.20.1” of name servers 53-1, 53-2, 53-3 that “have been designated as being the authoritative name servers for ‘example.com.’” *Id.*, 14:4-25; *also infra* Limitation [1d].

139. The name server then provides the browser with a content server IP address so that the user may retrieve the requested content. Glines, 7:18-8:2, *also infra* Limitation [1e].

140. Combining Glines with Gupta would not alter this analysis.

- e. **[1d] “(c1) causing said hostname to be resolved to said common anycast address by an Internet Service Provider (ISP) DNS server; and then”**

141. Glines, alone or in combination with Gupta, discloses this limitation.

142. As discussed in Limitation [1c], Glines discloses “causing said hostname to be resolved to said common anycast address.”

143. Glines teaches that, in response to a user entering a URL in a web browser for content, and “[i]n the case where the user is associated with an Internet Service Provider (ISP) operating the local area network 20, this first name request is made to a local DNS resolver 18, to determine the location of the domain “example.com.”” Glines at 13:21-14:3.

144. In response, the root DNS name server 38 is “programmed to . . . return the anycast address 50.100.20.1” of name servers 53-1, 53-2, 53-3 that “have been designated as being the authoritative name servers for ‘example.com.’” *Id.* at 14:4-15:4; *see also id.*, 7:1-7 (“If the ‘example.com’ domain is not located in a local domain, the local resolver will then proceed to consult public name servers that have been defined as being authoritative for the ‘example.com’ domain. The root domain name servers defined, for example, by the Internic, may be programmed to return the common address of the group of name server/content server peers that share a common anycast address.”), 9:9-15 (same).

145. The browser or local resolver then “sends out a DNS request as a UDP datagram to [the] anycast message” to resolve the content. *Id.*, 7:9-13; *see also id.*, 6:16-17 (“A request to resolve a domain name originates as a datagram addressed to the common anycast address.”).

146. By returning an anycast address (e.g., 50.100.20.1) associated with name servers 53-1, 53-2, and 53-3 in response to a user entering a request into a browser (e.g., via URL or hostname) for a website (e.g., content), Glines discloses causing a hostname to be resolved to a common anycast address, as claimed.

147. **Glines alone discloses resolving “by an Internet Service Provider (ISP) DNS server.”**

148. Glines discloses local DNS resolver 18, which is an ISP DNS server.

149. A POSITA would have understood that residential users’ Internet access is usually provided by an internet service provider (ISP). *See* EX-1011, 3-4 (“Roughly speaking, from bottom-to-top, the hierarchy consists of end systems connected to local Internet service providers (ISPs) through access networks.”), 128 (“Each ISP—such as a university, an academic department, an employee’s company, or a residential ISP—has a local DNS resolver (also called a default name *server*). When a host issues a DNS query message, the message is first sent to the host’s local name *server*.”); EX-1012, 229-230 (describing Internet providers); EX-1013, 4:65-5:53 (describing an “Internet Service Provider (‘ISP’) 118, 120, which provides connectivity to the network 100 on a fee for service basis.”).

150. Typically, a user would have accessed the Internet using their ISP’s DNS server utilizing a router or device that automatically obtains DNS settings from the ISP (e.g., through the Dynamic Host Configuration Protocol (DHCP)). *See*

EX-1014, 128-129, 338-339 (“DHCP allows a host to obtain . . . the address of its local DNS *server*”); *id.*, 338-339 (“Because of DHCP’s ability to automate the network-related aspects of connecting a host into a network, it is often referred to as a plug-and-play protocol DHCP is also enjoying widespread use in residential Internet access networks and in wireless LANs, where hosts join and leave the network frequently.”).

151. In many cases, the ISP DNS server would have been the default service to access the Internet for users. *See id.*; *see also* EX-1011, 307 (“DHCP is used extensively in LANs and in residential Internet access.”). Alternatively, these settings may be manually entered by the user. EX-1011, 128 (“The IP address of the local name *server* is typically configured by hand by the host [using Windows 95/98].”)

152. A POSITA would have accordingly understood that when Glines describes the browser of a “user associated with an Internet Service Provider” interacting with local DNS resolver 18, local DNS resolver 18 is an ISP DNS server, as claimed.

153. This understanding is consistent with the ’112 patent, which describes a “local DNS server 70a” or an “end user DNS 70a” but does not discuss a separate “ISP DNS server.” EX-1001, 4:56-5:31, 6:65-7:32, Fig. 5.

154. Glines further teaches that, when a user “associated with an Internet Service Provider” enters a URL (which includes a hostname, *supra* Limitation [1c]) in a web browser, local DNS resolver 18, shown below, sends a request to a root DNS server 38. Glines, 13:24-14:12; *also id.*, 11:24-28.

155. In turn, root DNS name server 38 is “programmed to ... return the anycast address 50.100.20.1” of name servers 53-1, 53-2, 53-3 that “have been designated as being the authoritative name servers for ‘example.com.’ *Id.*, 14:4-15:4; *also id.*, 7:1-7, 9:9-15. The browser or local resolver then “sends out a DNS request as a UDP datagram to [the] anycast message” to resolve the content. *Id.*, 7:9-13; *also id.*, 6:16-17

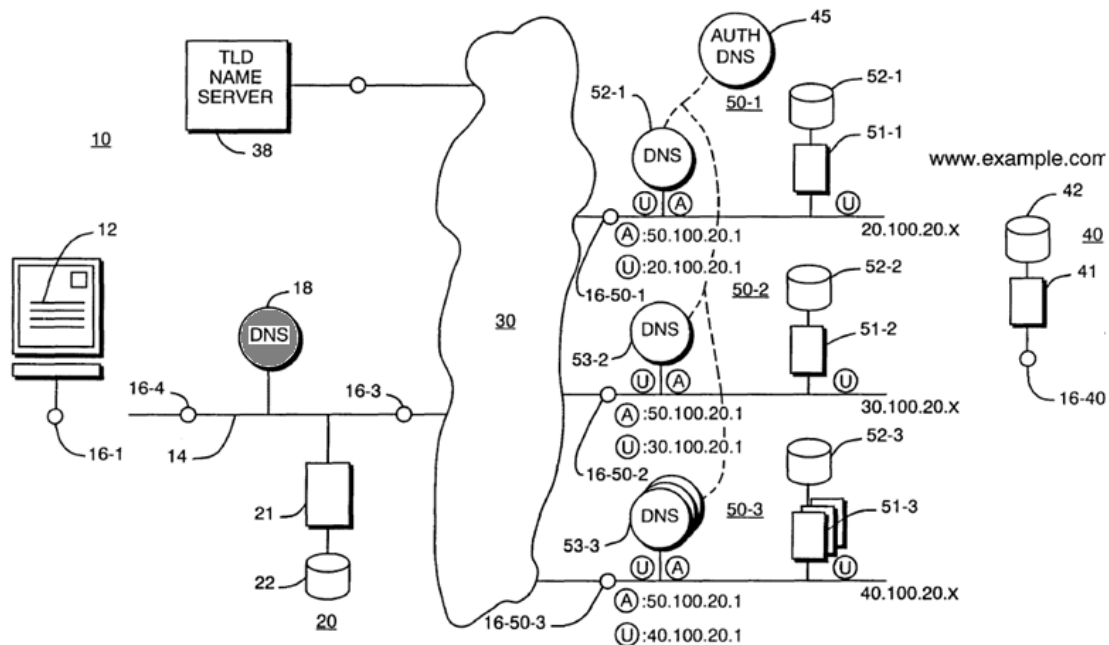


FIG. 1

Id., Fig. 1 (annotated).

156. Glines's local DNS resolver discloses an ISP DNS server that causes the hostname entered as part of the URL to be resolved to the name servers' common anycast address, as claimed.

157. To the extent Glines does not expressly disclose this feature (it does), it would have been obvious to implement and/or modify Glines's local resolver 18 as an ISP DNS server.

158. As noted above, a POSITA would have understood that Glines's local DNS resolver 18/local resolver is an ISP DNS server. But even if Glines did not disclose this feature, it would have been obvious to substitute an ISP DNS server for Glines's local resolver 18. This would have been nothing more than the simple substitution of one known element for another to obtain predictable results.

159. It was known to locate ISP-provided DNS servers in close proximity to a user's physical location, just as Glines' discloses via its *local* resolver. This makes them faster and more efficient than other, more distance DNS servers. *See* EX-1011, 128 ("The local name server is typically 'close to' the client" and may either be on "the same LAN" or "separated from the client by no more than a few routers."); EX-1013, 5:35-54 ("A[n] [Internet] service provider 118, 120 will generally provide POPs 114, 116 which are geographically proximate to the clients 102, 104, 106 being serviced.").

160. It was also known to configure residential Internet access to utilize ISP DNS servers. *See, e.g.*, EX-1014, 128-129 (“Each ISP . . . has a local DNS *server* (also called a default name server)”), 338-339 (“DHCP allows a host to obtain . . . the address of its local DNS *server*”); EX-1011, 128 (“The IP address of the local name *server* is typically configured by hand by the host [using Windows 95/98].”)

161. A POSITA therefore would have found it obvious to substitute or implement Glines’s local resolver 18 as an ISP DNS server.

162. A POSITA would have had an expectation of success in the substitution because using a local ISP DNS server was known and understood in the art for accessing the Internet and retrieving content by resolving domain names in systems such as Glines.

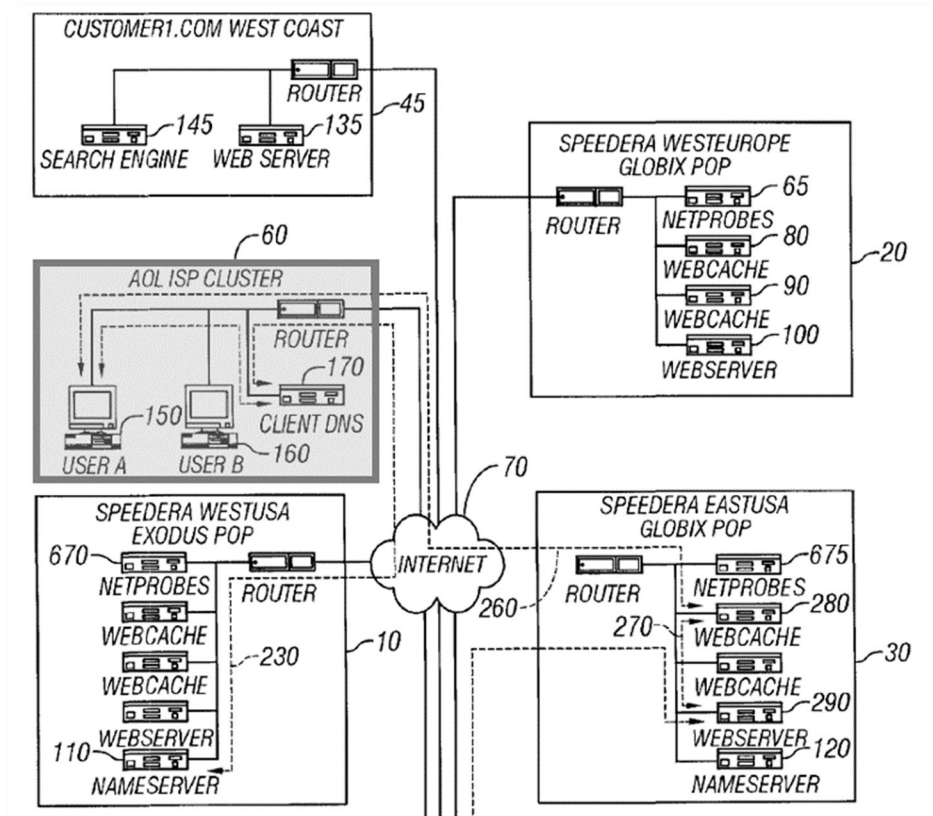
163. The modified Glines’s ISP DNS server would perform the same functions as Glines ascribes to local DNS resolver 18, and would thus cause the hostname to be resolved to the common anycast address by an ISP DNS server, as claimed

164. It also would have been obvious to combine Glines and Gupta to arrive at this feature.

165. For example, a POSITA also would have found it obvious to combine Glines and Gupta to cause a hostname to be resolved to a common anycast address by an Internet Service Provider (ISP) DNS server, as taught by Gupta.

166. Gupta discloses that “users 150 and 160 may be home users using dial-up lines, cable modems, DSL, or the like, to connect with **ISP 60**.” Gupta, 6:1-9, Fig. 1; *also id.*, 4:45-53. Gupta further discloses that “client DNS server 170 is typically provided within **ISP 60**.” *Id.*, 6:1-9. And Gupta further discloses that “client DNS server 170 is typically used to resolve a host name into an IP address.”

Id.; *also id.*, Fig. 1



Id., Fig 1. (cropped and annotating ISP 60).

167. Gupta’s ISP 60 thus performs the same functions as Glines’s local DNS resolver 18 (e.g., resolving a host name into an IP address). In view of Glines’s teaching that local DNS resolver 18 is used “where the user is associated with an

Internet Service Provider (ISP),” Glines, 13:24-27, a POSITA would have understood Gupta’s ISP 60 to be an obvious and known substitute for Glines’s local resolver 18.

168. A POSITA would have been motivated to make the modification to ensure compatibility with “typical” internet service architecture, *e.g.*, Gupta, 4:45-51, 5:65-6:9, and to achieve the speed and efficiency associated with ISP-provided DNS servers, *see, e.g.*, EX-1011, 128; EX-1013, 5:35-54.

169. Given the ubiquity of ISP-provided DNS servers and the disclosures of Gupta and Glines, a POSITA would have viewed the combination as nothing more than the simple substitution of known elements in the prior art according to understood principles to yield predictable results. A POSITA also would have had an expectation of success in the substitution. Combining Glines with Gupta in this way would not alter the features of Glines discussed elsewhere in this declaration.

- f. **[1e] “(c2) by one of said plurality of CDN DNS servers, resolving said hostname to identify an IP address for use by the client to retrieve the content from a content server.”**

170. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

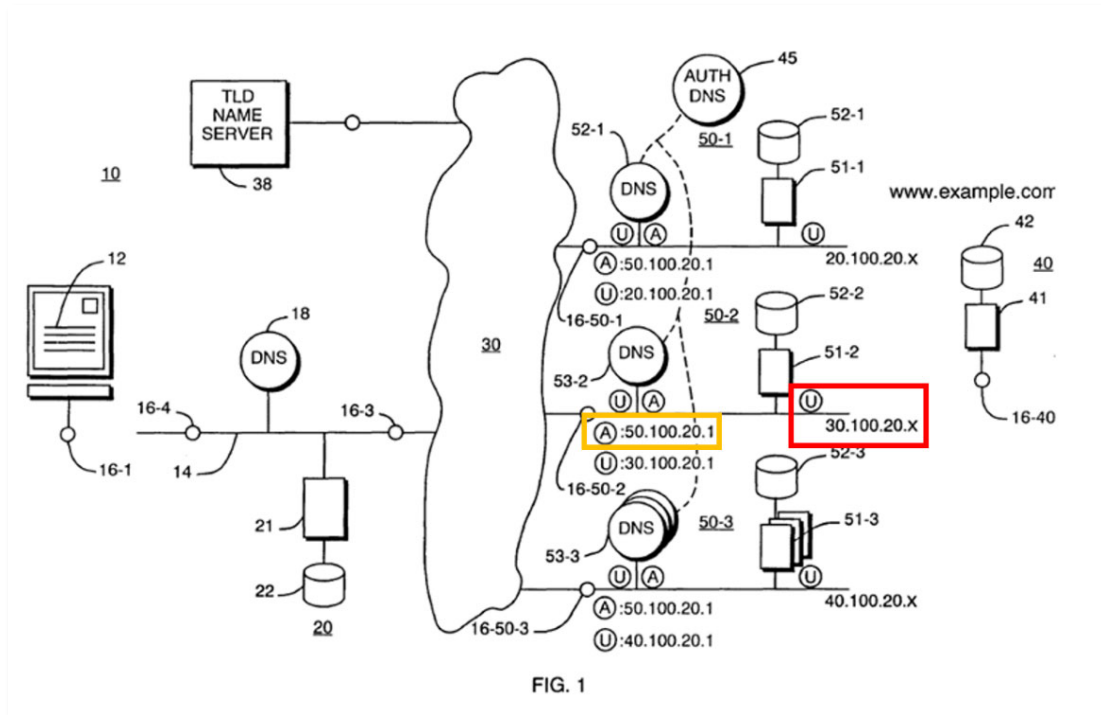
171. Glines discloses that, after receiving the common anycast address, the user sends a request to the anycast address associated with the name servers. Glines, 15:6-16:2 (“[T]he anycast datagram will reach one of the name servers 53-1, 53-2,

and 53-3 in the group associated with IP address 50.100.20.1, the one reached first being the one closest to the requesting client 12 or DNS resolver 18.”).

172. The closest name server 53, one of the plurality of CDN DNS servers, “will then respond by reporting the unique IP address of its associated replica content server 51,” thus resolving the hostname to identify an IP address for use by the client to retrieve the content from a content server as claimed. *Id.*; *also id.*, 11:1-12 (“The closest name server 53 then responds to the anycast message by returning a unique IP 5 address for its associated content server 51.”).

173. For example, the user may send a “DNS request for the IP address of ‘www.example.com’” to the anycast address of the name servers. Glines, 15:15-25. Where “the name server 53-2 may appear to be only one hop away,” it will respond and return the IP address of its associated replica content server 51-2, e.g., unicast address 30.100.20.x. *Id.*, 15:15-25, Fig. 1.² The “x” in this IP address identifies the host address of a machine within content server 51-2, allowing the client to use that address to receive the requested content. *Id.*, 15:26-16:8.

² I note that Glines discloses name server 53-2 being the closest in terms of network hops and responding with its unique IP address. *See* Glines 15:15-16:8. Glines incorrectly identifies this server as 53-3 in one instance. *See id.*



Glines at Fig. 1 (annotating the anycast address 50.100.20.1 and unicast address 40.100.20.x of content server 51-2).

174. This discloses, by one of the plurality of CDN DNS servers, resolving the hostname to identify an IP address for use by the client to retrieve the content from a content server, as claimed.

175. The user's browser program can then "make an HTTP level request for the file 'homepage.html' [of www.example.com] using the standard TCP/IP and HTTP network protocols" using "the unicast address for the content server 51." *Id.*, 16:3-8; *see also id.* at 11:5-7.

176. Combining Glines with Gupta would not alter this analysis.

3. Claim 2: “The method as recited in claim 1 wherein at least some of said plurality of CDN DNS servers are located at network Points of Presence (POPs).”

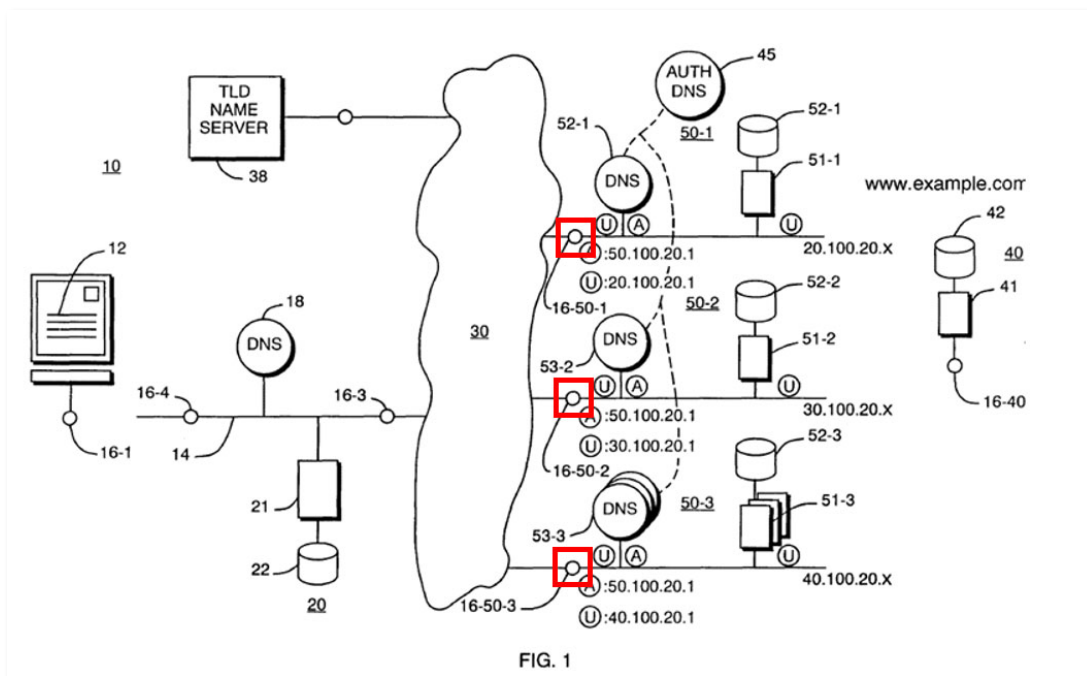
177. Glines, alone or in combination with Gupta, would have rendered this limitation obvious

178. Glines alone teaches this limitation.

179. Glines’s name servers 53 and associated content servers 51 are physically co-located and share access points to the internetwork 30 via internetwork devices 16-50: “[e]ach name server 53 and file server 51 operate quite closely together and, in fact, are preferably located physically near one another, such as on a common local area network segment sharing the same internetwork device 16-50-1.” Glines, 13:3-8, 8:19-23 (“Each name server returns the IP address of one or many (such as through round robin or other mechanisms) *closely bound and topologically nearby* content servers.”), 6:8-15 (“Each name server is *peered with a nearby* content server”), 7:6-7 (same), 10:23-25 (same), 13:3-9 (same), 16:6-8 (same), Figs. 1, 2 (Step 116). Glines discloses that the associated name server and content servers may be on “the same physical machine, if desired.” *Id.*, 8:11-13.

180. Glines also discloses that each of the peered name and content servers may access Internetwork 30 (e.g., Internet) via the same “internetwork devices 16.” *Id.*, 11:14-22, Fig. 1 (internetwork devices 16-50-1, 16-50-2, 16-50-3); *also id.*, Abstract, 6:3-7, 8:21-23, 9:9-12, 12:24-13:2, 15:11-16:8, Fig. 2 (Step 116).

181. The internetwork devices 16, including internetwork devices 16-50-1, 16-50-2, and 16-50-3, as depicted in Figure 1, *id.*, 11:14-22, Fig. 1, may include a “modem, network interface card, router, switch, bridge, gateway, or the like,” and “provide the ability for connections to be made between various computing system elements using network infrastructure such as the internetwork structure 30, which may be a corporate intranet or the Internet.” *Id.*



Id. at Fig. 1 (annotating internetwork devices 16-50-1, 16-50-2, 16-50-3).

182. As discussed with respect to Gupta, a “POP” is a point or physical location where two networks connect using, for example, a router. *E.g.*, EX-1013 at 4:65-5:54. Glines collocating its name server 53 and content server 51 service groups and causing them to share the same entry into the greater network (e.g., Internetwork

30), discloses at least some of the CDN DNS servers (and in fact, all of the CDN DNS servers) being located at network Points of Presence (POPs), as claimed.

183. Glines's disclosure corresponds with the '112 patent, which discloses nodes 76 comprised of DNS and content servers that are "located so as to be as close as possible to various network entry points." EX-1001, 5:11-18, Fig. 5.

184. **Alternatively, it also would have been obvious to combine Glines and Gupta to arrive at this feature.**

185. Gupta expressly discloses "a series of 'point of presence' (POP) servers," including POP servers 10, 20, and 30. Gupta, 4:45-67. These "are typically coupled via a computer network 70, such as the Internet." *Id.* Each POP server "include[s] a plurality of dedicated servers," including for example both WebCache servers (content servers) and "server[s] ... that include[] Domain Name Server (DNS) software," (CDN DNS servers). *Id.*

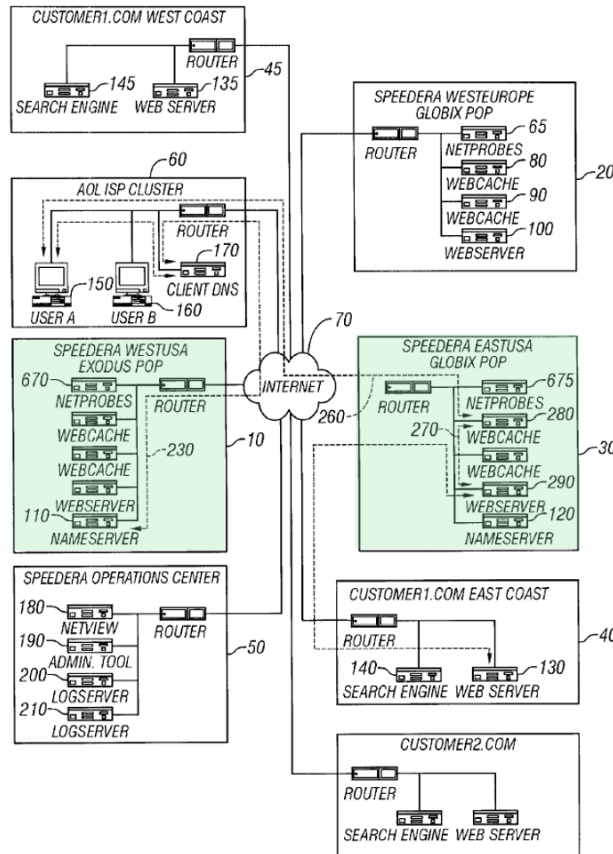


FIG. 1

Id. at Fig. 1 (annotating POP servers 10 and 20).

186. A POSITA would have combined Gupta and Glines to implement and/or modify Glines's name server and content server groups to be located at network POPs.

187. A POSITA would have been motivated to do so in order to distribute Glines's service groups to different access points on the network, including different geographic and topological locations. By distributing content servers at different POPs, content and services would be delivered from locations closer to users, thereby reducing latency. *See* Gupta 2:46-3:25 (selecting the optimal server to

“improve[] the user experience by reducing connection time, download time, errors, and the like.”). By localizing content delivery through POPs, for example regional POPs, Glines’s system would be improved by reducing the distance to users and reducing network congestion across the larger network. *See, e.g.*, EX-1011, 126-127 (noting benefits of decentralized DNS by multiple DNS servers, including “Load Distribution” and placing servers “close to” clients to reduce latency). This would have reduced bandwidth costs associated with sending data over longer distances from more centralized content servers. *See id.*

188. Combining Glines with Gupta in this way would not alter the features of Glines discussed elsewhere in this declaration.

4. Claim 3: “The method as recited in claim 2 wherein each of said plurality of CDN DNS servers is located at a network POP.”

189. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

190. Glines, alone or in combination with Gupta, renders obvious placing “at least some of said plurality of CDN DNS servers” at POPs. *Supra* claim 2.

191. For the same reasons, Glines, alone or in combination with Gupta, renders obvious placing *each* of the CDN DNS servers at a network POP. Glines’s disclosures expressly apply to “each” name server and file server 51. *See, e.g.*, Glines, 13:3-8, Fig. 1; *also id.*, 8:19-23 (“Each name server . . . [is] closely bound

and topologically nearby content servers.”), 6:8-15 (“Each name server is peered with a nearby content server”), 7:6-7 (same), 10:23-25 (same), 13:3-9 (same), 16:6-8 (same), Fig. 2.

192. To the extent Glines alone does not disclose this limitation, the benefits described in claim 2 of modifying Glines to position the servers at POPs, as expressly disclosed by Gupta, *e.g.*, Gupta, 4:45-67, Fig. 1, would be heightened by placing each name server at a POP. *Supra* Claim 2.

5. Claim 4: “The method as recited in claim 2 wherein at least one of said plurality of CDN DNS servers and respective plurality of content servers is grouped as a cluster.”

193. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

194. And as discussed in claim 1, Glines, alone or in combination with Gupta, discloses that each name server 53 (CDN DNS server) is associated with a plurality of content servers. *Supra* Limitation [1a].

195. Glines further teaches that its name servers and associated content servers may be arranged as a “server pair,” and that the name servers may report the address of an associated content server by sharing a “cluster address.” *See* Glines, 7:14-24 (explaining name server “responds by reporting the unique (unicast) address of the associated content replica server (*or cluster address* if the systems are arranged in a round robin fashion)”); *see also id.*, 8:14-23 (“Each name server returns

the IP address of one or many (such as through round robin or other mechanisms) closely bound and topologically nearby content servers.”).

196. Glines also discloses that the name server 53 and file server 51 pairs are “peers” and part of “service groups.” *Id.*, 13:3-8 (“peering arrangement”), 6:8-15 (same), 7:6-7 (same), 10:23-25 (same), 13:3-9 (same), 16:6-8 (same), Figs. 1, 2 (Step 116), 17:15-19:16 (describing “defined groups of machines” and “service group[s]”).

197. By providing the name server 53 associated file servers 51 that are pairs/peers and part of “service group[s]” for responding to user requests for content, and by disclosing that user requests may be addressed to a “cluster address,” Glines teaches or at least renders obvious “at least one of the CDN DNS servers and respective plurality of content servers being grouped as a cluster” as claimed.

198. Cluster computing was well known in the art. As was known, a cluster may be a group of interconnected computers that work together as a unified system to perform tasks, provide services, and/or share resources. *See* EX-1015 at 9-11 (“An interconnected (LAN-based) cluster of computers can appear as a single system to users and applications. Such a system can provide a cost-effective way to gain features and benefits (fast and reliable services) that have historically been found only on more expensive proprietary shared memory systems.”). A POSITA would have grouped Glines’s name server/content server pairs as a cluster based on

Glines's teachings and the benefits of cluster computing, including high performance, scalability, high throughput, and high availability over existing systems. *See id.*, 11.

199. It was also known at the time to utilize cluster computing for serving Internet content. *See* EX-1023, 38 (“The exponential growth of the Internet . . . has created the need for more and faster Web servers capable of serving the over 100 million Internet users.”).

200. In doing so, the systems serving content can be more easily scaled using commodity machines, and do so in a transparent manner to the user. *See id.* (“A pool of servers tied together to act as a single unit, or server clustering, provides [] incremental scalability” to grow capacity with demand), 39 (“[A] dispatcher receives service requests from clients and selects a server from the server pool to process the request. Depending on the clustering technology, the dispatcher appears as either a switch (processing incoming data only) or a network gateway (processing incoming and outgoing data) to the servers in the pool.”); EX-1011, 102-103 (“An alternative form of cooperative caching involves **clusters** of caches, often collocated on the same LAN. A single cache is often replaced with a **cluster of caches** when the single cache is not sufficient to handle the traffic or provide sufficient storage capacity.”).

201. Such clusters may utilize switches, such as Layer 4 switches, and implement load-balancing policies to prevent overloading of the cluster. *See*

EX-1023, 39 (discussing Layer 4 switches and load balancing); *see also id.*, 40 (disclosing that “[s]erver selection is based on some load sharing algorithm, which may be as simple as round-robin.”). This corresponds with Glines’s disclosure, which also discloses utilizing Layer 4 switches, load balancing, and selecting content servers using “round robin” techniques. Glines, 19:13-16.

202. Implementing and/or modifying Glines’s system such that at least one (or more) of its name server/content server pairs are grouped as a cluster would have been nothing more than combining prior art elements to yield predictable results (e.g., combining Glines’s system with known cluster computing techniques to serve Internet content).

203. The benefits would have included improved scalability, fault tolerance, and load-balancing in Glines’s system. For example, machines added to Glines’s system could be done transparently via each cluster. *See* EX-1015, 9-10; EX-1023, 38. This would improve the ability to add machines to and thus improve scalability, increase load-balancing to match high demand, and improve availability in the event of individual machine failure. Glines teaches these benefits and responding to content requests with a “cluster address,” thereby suggestion the modification. *See* Glines, 7:14-24, 19:13-16.

204. Combining Glines with Gupta would not alter this analysis.

6. Claim 5: “The method as recited in claim 1 wherein each of said plurality of content servers is associated with a network Point of Presence (POP).”

205. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

206. Claim 5 requires “*each* of the plurality of content servers” to be associated with a network Point of Presence (POP).

207. Glines, with or without Gupta, teaches each CDN DNS server being located at network Points of Presence (POP). *Supra* Claim 3

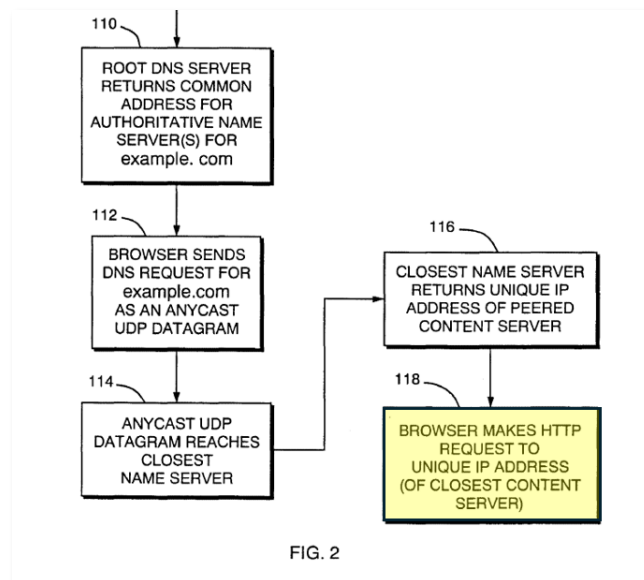
208. Further, Glines, with or without Gupta, teaches that each name server (CDN DNS server) is clustered with its plurality of content servers, and a POSITA would have been motivated to group Glines’s name and associated content servers as a cluster. *Supra* Claim 4

209. For the same reasons, Glines, alone or in combination with Gupta, discloses that each content server is associated with a network POP. Each of the content servers would be associated with and clustered with a name server, *supra* claim 4, and each of those name servers would be associated with a POP, *supra* claim 3, such that each of the content servers would also be associated with a POP as claimed.

7. Claim 6: “The method as recited in claim 1 further comprising: (D) by the content server, attempting to serve the content.”

210. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

211. Glines explains that after receiving the IP address of the content server, *see* Limitation [1e], the user’s browser makes an HTTP level request for the content (e.g., “homepage.html”) using “standard TCP/IP and HTTP network protocols” via the returned “unicast address for the content server 51.” Glines, 15:21-16:8; *also id.*, 11:1-12 (“This scheme permits the browser to subsequently establish the higher level protocol access method, such as a Hyper Text Transfer Protocol (HTTP) request, to open a connection and deliver the content file.”). In response, “[t]he requested file is then returned from the content server 51-2.” *Id.* at 16:3-8



Id. at Fig. 2 (cropped and annotating step 118 of requesting content and content being returned from closest content server).

212. Because the content server returns the requested file, Glines discloses the content server *attempting* to serve the content, as claimed. In order to successfully serve content, the content server must attempt to serve the content.

213. This is consistent with the '112 patent, which “serves the object to the end-user system that requested it.” EX-1001, 7:24-32 (disclosing that “the cache server 80*b* ... serves the object to the end-user system that requested it.”). The specification does not otherwise use the term “attempt.” EX-1001. This is also consistent with Patent Owner’s allegations. EX-1022, 18-20 (“Google, by the content server, attempts to serve the content. Google CDN delivers content from content server(s) through the method of a content delivery network.”)

214. Combining Glines with Gupta would not alter this analysis.

8. **Claim 7: “The method as recited in claim 6 wherein said step (D) of attempting comprises: (d1) if a valid version of the content is available on the content server, serving the content; otherwise (d2) obtaining the content from a content source and then serving the content.”**

215. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

216. Glines, alone or in combination with Gupta, discloses claim 6. *Supra* Claim 6.

This at least renders obvious “if a valid version of the content is available on the content server, serving the content,” as claimed.

219. A POSITA also would have understood from Glines that, if the content was not available on a particular content server or is otherwise out-of-date or stale, it would have been obvious in Glines for the content server to request the content from the origin server 40 before providing it to the user.

220. This includes, for example, when Glines’s name server provides the IP address of a content server to the user or after the user requests the content from the content server. This is taught or at least rendered obvious by Glines’s disclosure that content files are propagated to the content servers 50-1, 50-2, 50-3 “upon demand.” Glines, 12:11-23.

221. Both options would have ensured content is widely distributed and provided via the closest available content server, now and for future requests. This would have been preferable to a name server not responding to the anycast address if content is unavailable at a peered content server, for example, because responding and requesting the content from the content source would add otherwise missing content to that content server, thereby distributing the content across Glines’s system to efficiently fulfill future requests. Not doing so would also force a user to request content from potentially distant content servers that have the content stored.

222. Accordingly, a POSITA would have found it obvious to implement Glines as claimed.

223. **Gupta discloses these features.**

224. For example, Gupta's WebCache servers, like Glines's content servers, "respond[] to requests for static web content from clients." Gupta, 5:16-40, 13:23-30, Fig. 4. Gupta teaches, however, first determining "[i]f the requested content does not exist in server memory," thus determining if a valid version of the content is available on the content server or not. *Id.*, 5:16-40. Where content is available, the server responds to the request "from memory," serving the content as claimed. *Id.*

225. However, if the requested content is not cached, Gupta teaches that its WebCache server software "will generate a request to a WebServer, that is known to include the content, for example, web server 130." *Id.* The WebServer then sends the requested content to the WebCache server and "the WebCache software caches the content in memory." *Id.*; *see also id.*, 13:23-30 ("If the content exists on the local disk, the WebServer will pass the data to the WebCache."). The WebCache in turn "returns the static content to the user." *Id.*, 10:40-11:7; *also id.*, 5:16-40, 13:23-30, Fig. 4.

226. A POSITA would have understood that Gupta's WebCache checking whether the requested content is cached and either signaling a cache miss or

returning the requested content, discloses determining if a valid version of the content is available on the content server before serving the content, as claimed. *See id.*, 5:16-40, 13:23-30, Fig. 4.

227. Gupta's WebCache requesting the missing content from a WebServer to fulfill the request, receiving the content from the WebServer, and providing it to the user, discloses "obtaining the content from a content source and then serving the content," as claimed.

228. This is consistent with Patent Owner's allegations, which include "serv[ing] content from the content server (such as cached sources) *if available* on the content server." EX-1022, 20-21.

229. A POSITA would have implemented Gupta's teachings in Glines and would have implemented this check before a name server responds to the anycast address or after Glines's name server provides the IP address of the particular content server (e.g. WebCache). As discussed, this would have improved content distribution across Glines's system and improve future service of that content at the closest available content server to the user. For example, in addition to fulfilling a user's request, this would have added otherwise missing content to that content server for future requests.

230. The combination also would have improved Glines's efficiency. If requested content is not available on a responding content server, a user would still

require the content. Requesting the content from the content source and then delivering it via the content server would have avoided the user having to reinitiate another request for content or request to resolve a hostname.

231. A POSITA would have understood the combination would have been nothing more than combining known elements in Gupta with Glines's system to yield predictable results. A POSITA would have had an expectation of success because Gupta and Glines disclose similar content caching techniques and it is foundational to first check that content is available in memory before retrieving and/or providing it.

232. A POSITA would have implemented Gupta's content-checking and cache miss signal in Glines—either before a name server responds to the anycast address or after Glines's name server provides the IP address of a content server. This includes requesting the content from origin server 40 or from another content server. In addition to fulfilling a user's request, this would have added otherwise missing content to that content server, thereby distributing the content across Glines's system to efficiently fulfill future requests. It also would have reduced latency and improved the usefulness of Glines. Requesting the content from the content source and delivering it via the content server would have avoided the user having to reinitiate another request for content or request to resolve a hostname.

9. Claim 8: “The method as recited in claim 7 wherein, said content source is an origin server associated with said content provider.”

233. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

234. Glines, alone or in combination with Gupta, discloses claim 7. *Supra* claim 7.

235. As discussed in claim 7, Glines discloses a content source that includes origin file server 40. Glines, 2:9-13, 9:16-10:25, 12:11-23 (“[R]eplica content servers 50-1, 50-2, 50-3 store replicas of one or more of the content files that ***originate at the origin server 40.***”), 14:13-25. The origin server is associated with the content provider; for example, Glines’s content provider “example” provides content from “www.example.com.” *Id.*, 13:15-14:25.

236. A POSITA would have understood that content files that “originate” at an origin server in Glines discloses a content source that is an origin server associated with a content provider. The content provider in Glines may include the creator of the content and the party associated with the origin server (e.g., an organization associated with a particular domain name).

237. Gupta similarly discloses customer web server 130, which is the content source from where content that is not already cached is obtained. *Supra* claim 7.

238. Gupta discloses a “customer” that is a content provider. Gupta, 5:23-33, 2:10-17, 8:53-59 (a user entering “the name of a customer’s web site”), Fig. 1. A POSITA would have understood Gupta’s customer web server 130 to be an origin server, as claimed.

239. For the reasons discussed in claim 7, a POSITA would have combined Glines and Gupta, and the combination would have rendered obvious an origin server in Glines’s system that is “associated with said content provider,” as claimed. *Supra* claim 7.

10. Claim 10: “The method as recited in claim 1, wherein selection of the content server is based on a load balancing policy.”

240. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

241. Glines discloses each name server and content server “service group” may include fallover/fallback policies to avoid issues with content server outages or overloads. Glines, 18:20-19:16. These policies include, “within the service group, standard load balancing” using “layer 4 switches.” *Id.*, 19:10-16.

242. Layer 4 switches are commonly used in the art in advanced networking scenarios for load balancing, where the switch can distribute HTTP/HTTPS requests to multiple web servers based on Layer 4 attributes (e.g., TCP ports) or sessions, ensuring optimal server utilization. *See, e.g.*, EX-1006, 1:49-2:7 (“Applications that

communicate over the Internet typically communicate with each other over a transport layer (layer 4) Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) connection.”); EX-1016, 1:28-53 (“Server load balancers use information in the Layer 3 and Layer 4 packet headers to identify and manage application layer sessions.”).

243. Layer 4 switches are known to be useful for web servers to improve performance and reliability by distributing load across many servers. *See* EX-1017, 1:19-48 (“Many of today’s web sites are hosted on server farms, where a number of servers are grouped together to serve web requests as they arrive. To avoid overloading individual servers within the farm, load balancing techniques balance the load across servers so that the best total throughput of the farm and smallest response delay for the user is achieved.”); EX-1026 at Abstract (describing “[p]revious methods for connection routing (Layer 4 Switching) have employed a centralized node to handle all incoming requests.”), 2-3 (disclosing these methods as implementing a policy in which “load is balanced across server hosts by providing a mapping from a single host name to multiple IP addresses.”), Figs. 1(a) (depicting switch/router forwarding requests to servers in server farm), 1(b) (same).

244. Glines’s “standard load balancing” using “layer 4 switches” therefore discloses selection of a content server based on a load balancing policy, as claimed.

245. Glines further teaches that when the “local load reache[s] some watermark, it can load shed to content servers in other service groups by returning their IP address to name lookups coming at the anycast address,” which is also selection of a content server based on a load balancing policy. Glines, 19:17-22.

246. Combining Glines with Gupta would not alter this analysis.

11. Claim 11: “The method of claim 1 further comprising: (c3) by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.”

247. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

248. As explained in claim 10, Glines discloses using a Layer 4 switch for “standard load balancing” among multiple content servers in order to apply a load balancing policy and select a content server. *Supra* Claim 10.

249. For the same reasons, Glines discloses selecting the content server by at least one switch: Glines’s switch is used for the load balancing and determines which content server should be selected.

250. For example, Glines discloses that a name server 53 responds with the “unique IP address of its associate replica content server 51” from which the user can retrieve the requested content. Glines, 15:26-16:2. Glines discloses that, “within [a] service group, standard load balancing and replication techniques can also apply,

such as multiple content servers (returning multiple local IP addresses to a name lookup), layer 4 switches, etc.” *Id.*, 19:13-16.

251. As discussed, the “service group” in Glines is a peered name server and its content servers. *See id.*, 19:17-22 (discussing load balancing “across service groups”), 18:17-19 (discussing content servers “in the service group”); *see also id.*, 6:11-13 (“Each name server is peered with a nearby content server, cache server, or other file server that contains (or can serve) desired content file replicas via unicast addressing.”).

252. Glines’s applying “standard load balancing” techniques within a service group, including use of “layer 4 switches,” discloses or at least renders obvious using a Layer 4 switch to load balance across the content servers in the service group. A POSITA therefore would have understood that Glines teaches or renders obvious, by at least one switch (e.g., a Layer 4 switch), selecting the content server from the respective plurality of content servers associated with the resolving CDN DNS, as claimed.

253. Combining Glines with Gupta would not alter this analysis.

12. Claim 12: “The method of claim 11, wherein the step of (c3) is performed in accordance with a predetermined selection policy.”

254. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

255. As explained in claims 10 and 11, Glines discloses that its Layer 4 switch is used for “standard load balancing” across content servers. This discloses selection of a content server by a switch in accordance with a “predetermined selection policy” (here, a load balancing policy). *Supra* Claims 10, 11. A POSITA would have understood that a system that load-balances requests across multiple servers is a “policy” that is “predetermined” by the administrators of the system.

256. This corresponds with the '112 patent, which discloses selecting a cache server based on “a predetermined selection policy, for example, . . . , load balancing, and so forth, using known techniques.” EX-1001, 8:52-58. Because the specification contemplates load balancing as a predetermined selection policy and nothing otherwise requires the “predetermined” selection policy to be, for example, a predetermined selection of a specific server, Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

257. Combining Glines with Gupta would not alter this analysis.

13. Claim 13

- a. [13p] “A method of content delivery for delivering content on behalf of a plurality of content providers, the method comprising:”**

258. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

259. As described above, Glines discloses a method of content delivery for delivering content. Limitation [1p].

260. Glines further discloses a method of content delivery for delivering content on behalf of a plurality of content providers because Glines expressly discloses that its system may be used with either a “single provider” or “multiple providers”:

In the case of deployment within a single provider, where the anycast routing is internal to that network (and transparent to the outside - the Internet), and an internal routing protocol like iBGP or OSPF is used, pro[pa]gation of changes should be fast. In the case of deployment across **multiple providers**, full fledged external BGP (‘eBGP’) preferably would be used.

Glines, 17:3-9.

261. Glines thus discloses content delivery on behalf of a **plurality** of content providers, as claimed.

262. A POSITA would have understood Glines’s disclosure of providing content from an origin source would have included supporting more than one origin source (e.g., more than one website or content provider). *See id.*, 12:11-22 (storing replica files that originate at an origin server 40), 13:21-14:3 (requesting “example.com”).

263. Adding additional origin servers and content providers to Glines’s system, as expressly contemplated in Glines, would have been nothing more than

replicating Glines's teaching of storing and providing content associated with one origin server, with expected results. Doing so would have enabled Glines's system to support more (and different) content to fulfill user requests, including content from different content providers.

264. It also would have been obvious to combine Glines and Gupta to arrive at this feature.

265. Gupta discloses content delivery on behalf of a plurality of content providers, and for this additional reason, it would have been obvious to provide content delivery on behalf of a plurality of content providers in a combined Glines-Gupta system.

266. For example, Gupta discloses that “[c]ustomers”—plural—can store content “such as HTML, images, video, sound, software, or the like on the network for fast and highly available access by clients (end users).” Gupta, 2:9-17. Gupta also discloses that its POP server network may provide traffic management and content hosting “to *more than one specific customer*,” including “customer1 and customer2” that share the server network.” *Id.*, 7:8-20, Fig. 1 (depicting customer1 and customer2 web servers); *see also id.*, 8:53-59 (user typing in request for a *customer's* web site “customer1.com.”).

267. It would have been obvious to apply these teachings to Glines's system. The combined system would have stored and provided content from its content

servers from multiple content providers (e.g., multiple customers and their associated websites, etc.). The combination would have been nothing more than combining known elements in the prior art according to understood principles, with an expectation of success. Doing so would have been expected and well within the capabilities of a POSITA to, for example, maximize the usefulness and value of Glines's system by providing Glines's service to more than one customer/content provider.

- b. [13a] “(A) providing a plurality of Domain Name System (DNS) servers, said plurality of DNS servers sharing a common anycast address, wherein each DNS server is associated with a respective plurality of content servers;”**

268. This limitation is met for the same reasons described in connection with Limitation [1a], which recites “providing a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), said plurality of CDN DNS servers sharing a common anycast address, wherein each CDN DNS server is associated with a respective plurality of content servers.”

269. The limitations differ only in that Limitation [13a] does not require the DNS servers to be associated with a CDN.

270. Accordingly, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1a]. *See* Section IX.A.2.b.

- c. **[13b] “(B) configuring each DNS server and respective plurality of content servers to support content delivery for a plurality of content providers, wherein each of the plurality of content providers has a respective hostname associated therewith;”**

271. This limitation is met for the same reasons described in connection with Limitation [1b], which recites “causing said plurality of CDN DNS servers to be authoritative for a hostname associated with a content provider by causing said common anycast address to be associated with said hostname.”

272. The limitations differ only in that Limitation [13b] recites “configuring each DNS server and respective plurality of content servers to support content delivery for a plurality of content providers,” where each content provider “has a respective hostname associated therewith.”

273. Limitation [13a]’s “configurating” is indistinguishable from Limitation [1b]’s “causing” the CDN DNS servers to be authoritative for a hostname associated with a content provider.

274. Accordingly, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1b]. *See* Section IX.A.2.c.

275. It also would have been obvious to support more than one content provider from each name server and content server peer group in Glines, where each

content provider has an associated hostname (e.g., “customer1.com” and “customer2.com” in Gupta). *See* Limitation [13p] (Section IX.A.13.a).

276. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13p]’s content delivery for a plurality of content providers with respective hostnames.

- d. **[13c] “(C1) responsive to a first request for first content associated with a first content provider of said plurality of content providers, said first request including at least a first hostname associated with the first content provider,”**

277. This limitation is met for the same reasons described in connection with Limitation [1c], which recites “responsive to a request for content associated with the content provider and issued by a client, said request including at least said hostname.”

278. The limitations differ only in that Limitation [13c] recites a “first” request, “first” content, “first” hostname, and a “first” content provider.

279. Responding to a request for content associated with a content provider including a hostname is indistinguishable from responding to a first request for first content associated with a first content provider including a first hostname.

280. Accordingly, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1c]. *See* Sections IX.A.2.d.

- e. **[13d] “(c11) by a first DNS server of said plurality of DNS servers, resolving said first hostname to identify a first IP address to be used for retrieval of the first content from a first content server of said content servers associated with said first DNS server;”**

281. This limitation is met for the same reasons described in connection with Limitation [1e], which recites “by one of said plurality of CDN DNS servers, resolving said hostname to identify an IP address for use by the client to retrieve the content from a content server.”

282. Doing so by a “first” DNS server of a plurality to resolve a “first” hostname via a “first” IP address to retrieve the “first” content from a “first” content server is indistinguishable from Limitation [1e].

283. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1e]. *See* Sections IX.A.2.f.

- f. **[13e] “(c12) by a switch associated with said resolving DNS server of step (c11), selecting the first content server from said respective plurality of content servers associated with the DNS server of step (c11); and”**

284. This limitation is met for the same reasons described in connection with claim 11.

285. This limitation differs from claim 11 only that the switch is “associated with said resolving DNS server of step (c11),” and the switch selects the “first” content.

286. Glines, alone or in combination with Gupta, would have rendered this limitation obvious by Glines's Layer 4 switch as explained in claim 11. *See* Section IX.A.11.

287. Glines teaches or at least suggests that its Layer 4 switch is associated with its name server/content server peer groups, and therefore satisfies a switch "associated with said resolving DNS server," as claimed. *See* Glines, 18:20-19:16 (disclosing that each name server and content server "service group" may include fallover/fallback policies to avoid issues with content server outages or overloads, including, "within the service group, standard load balancing" using "layer 4 switches").

g. [13f] "(c13) attempting to serve said first content from the first content server; and"

288. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 6. *See* Sections IX.A.7.

289. Limitation [13f] only differs from claim 6 in that the content being served is a "first" content from a "first" content server, which is subsumed into claim 6.

- h. [13g] “(C2) responsive to a second request for said first content associated with said first content provider of said plurality of content providers, said second request including at least the first hostname associated with the first content provider,”**

290. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13c]. *See* Section IX.A.13.d.

291. Limitation [13g] differs from Limitation [13c] in that it recites a “second” request for the first content.

292. Glines, with or without Gupta, would have rendered this limitation obvious.

293. Glines discloses providing content distribution “as a service where client requests are automatically routed to the closest available content server.” Glines 6:3-15; *id.*, Abstract, 9:9-17 (“requests” for content), 10:5-12 (handling “the number of requests”). Similar or repeated processing would occur to fulfill those “requests.”

294. As discussed, repeating a process is an obvious and expected implementation of Glines’s system. *See* Limitation [13p]. This includes Glines providing a user the same or different content at different times, or the same or different content to different users.

295. This corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those “requests.” Repeating the same process twice is an obvious and expected implementation of Glines’s system.

- i. **[13h] “(c21) by a second DNS server of said plurality of DNS servers, resolving said first hostname to identify a second IP address for retrieval of the first content from a content server associated with the second DNS server;”**

296. Limitation [13h] differs from Limitation [13d] in that a “second” DNS server resolves the first hostname to identify a “second” IP address for retrieval of the first content.

297. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

298. Dependent claim 15 recites that the first DNS server and second DNS server are the same server. EX-1001, claim 15. Dependent claim 17 further recites that the second content server (and thus the “second IP address”) and the first content server are the same server. EX-1001, claim 17.

299. Element [13h] therefore encompasses repeating a DNS resolution for the first content based on a second user request.

300. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitations [1e] and [13d]. *See* Section IX.A.2.f, IX.A.13.e.

301. As explained, it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. *See* Limitations [13p], [13g]. Glines's resolution process would have been handled in the same way for each of the requests, and Glines, with or without Gupta, would have rendered this limitation obvious as explained in Limitations [1e] and [13d].

302. This corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those requests. Repeating the same process twice is an obvious and expected implementation of Glines's system. This would have included Glines providing the same user the same or different content at different times, or the same or different content to different users.

303. Even if the first and “second” DNS servers are different servers, it would also have been obvious to resolve a second request for particular content via a different name server in Glines (e.g., via a different DNS server 52 and different associated content server 51. Glines, 14:17-16:8, Fig. 1.

304. For example, if a first name server is overloaded or offline, or if a second name server is available that is “closer” to the user sending the second request, it would have been obvious to use the second name server in Glines’s network 10 to resolve the content. *Id.*, 14:17-16:8, 18:13-19:20 (disclosing that, “if the host or gated dies unexpectedly, then failover needs to happen,” including “[l]oad balancing across service groups” to “load shed to content servers in other service groups by returning their IP address to name lookups coming at the anycast address.”), 15:11-14 (“the anycast datagram will reach one of the name servers 53-1, 53-2, and 53-3 in the group associated with IP address 50.100.20.1, the one reached first being the one closest to the requesting client 12 or DNS resolver 18.”).

305. The second name server in Glines would respond to the anycast address by identifying an associated content server 51, and thus satisfy identifying a “second IP address” that is different than the “first IP address” provided in response to the first request. Indeed, Glines contemplates this with respect to its system in Figure 1. *See* Glines, 14:17-16:8, Fig. 1.

- j. **[13i] “(c22) by a switch associated with said resolving DNS server of step (c21), selecting the second content server from said respective plurality of content servers associated with the DNS server of step (c21); and”**

306. Limitation [13i] differs from [13e] and in that a switch associated with the “second” DNS server selects the “second” content server.

307. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

308. Dependent claim 15 recites that the first DNS server and second DNS server are the same server. EX-1001, claim 15. Dependent claim 17 further recites that the second content server (and thus the “second IP address”) and the first content server are the same server. EX-1001, claim 17. Dependent claim 16 additionally recites the switch of (c22), limitation [13e], and the switch of (c12), limitation [13i], are the same switch.

309. Element [13i] therefore encompasses a DNS resolution of the second request using the same switch.

310. Glines, with or without Gupta, would have rendered this limitation obvious for the reasons explained in Claim 11 and Limitation [1e].

311. As discussed, it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. This corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest

available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”).

312. Similar or repeated processing would occur to fulfill those requests, including using the same switch (or a second switch) to select a particular content server to provide the content to the user. As discussed, Glines teaches or at least suggests that its Layer 4 switch is associated with its name server/content server peer groups. *See* Glines, 18:20-19:16 (disclosing that each name server and content server “service group” may include fallover/fallback policies to avoid issues with content server outages or overloads, including, “within the service group, standard load balancing” using “layer 4 switches”). Repeating the same process twice is an obvious and expected implementation of Glines’s system. This would have included Glines providing the same user the same or different content at different times, or the same or different content to different users.

313. A POSITA alternatively would have found it obvious to associate a switch, such as Glines’s Layer 4 switch, at each name server location in order implement load-balancing for associated content servers. This would have been nothing more than combining known elements in the prior art, including Glines’s Layer 4 switch with Glines’s name server locations, with predictable results.

314. Combining Glines with Gupta would not alter this analysis.

315. Even if the first and second DNS server are understood to be different servers, this feature would have been obvious over Glines and/or Gupta. Glines and Gupta both disclose DNS server-content server pairs with similar configurations. *See* Glines, 12:11-13:14, 14:17-16:8, Fig. 1; Gupta, 4:45-67, 5:41-6:21, Fig. 1. Limitation [13i] would have obvious in either Glines or Glines-Gupta because it would have involved mere replication of steps (e.g., replication of limitation [13e]) in a different CDN node (e.g., different DNS server-content server pair).

316. Indeed, Glines contemplates load balancing using Layer 4 switches “within the service group” and discloses multiple service groups in 50-1, 50-2, and 50-3. *See* Glines, 19:17-22. It therefore would have been obvious in Glines to select a content server via a switch associated with different name server in response to the second request.

k. [13j] “(c23) attempting to serve said second content from the second content server.”

317. Whether [13j] is understood to be attempting to serve the same or different content in response to the second request, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 6. *See* Section IX.A.7.

318. As explained, it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. This corresponds with the objectives of Glines, including providing content

distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those requests, including attempting to serve content to users based on different requests. Repeating the same process twice is an obvious and expected implementation of Glines’s system. This would have included Glines providing the same user the same or different content at different times, or the same or different content to different users.

319. A POSITA thus would have understood that each request would be resolved in the same way, such that in response to a second request, the system would attempt to serve the requested content from the identified content server in the same manner described in claim 6. *Supra* Claim 6.

14. Claim 14: “The method as recited in claim 13, wherein selection of at least one of the first content server and the second content server is based on a load balancing policy.”

320. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

321. As explained, in connection with claims 10 and 11, Glines, alone or in combination with Gupta, would have rendered obvious selection of a content server based on a load balancing policy. *Supra* Claims 10, 11.

322. The only difference between claim 14 and claim 10 is that the “first” and “second” content servers are selected by a load balancing policy. As explained, it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. *See* Limitations [13g], [13h]. Similar or repeated processing would occur to fulfill those requests. Repeating the same process twice, for example selecting a content server by a load balancing policy across the same or different service groups would have been an obvious and expected implementation of Glines’s system, and this Limitation would have been obvious for the reasons explained in claims 10 and 11.

15. Claim 15: “The method as recited in claim 13, wherein the resolving DNS server of step (c11) is the same as the resolving DNS server performing of step (c21).”

323. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13h]. *See* Section IX.A.13.i.

324. As discussed, it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. *See* Limitations [13p], [13g]. Glines’s resolution process would have been handled in the same way for each of the requests, and Glines, with or without Gupta, would have rendered this Limitation obvious as explained in Limitation [13h].

325. Indeed, this corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those requests. Repeating the same process twice is an obvious and expected implementation of Glines’s system. This would have included Glines providing the same user the same or different content at different times, or the same or different content to different users.

16. Claim 16: “The method as recited in claim 15, wherein the switch of step (c12) is the same as the switch of step (c22).”

326. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13i]. *See* Section IX.A.13.j.

327. As explained, it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. This corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to

fulfill those requests, including attempting to serve content to users based on different requests. Repeating the same process twice is an obvious and expected implementation of Glines's system. This would have included Glines providing the same user the same or different content at different times, or the same or different content to different users.

328. A POSITA thus would have understood that each request would be resolved in the same way, such that in response to a second request, the system would attempt to serve the requested content from the identified content server in the same manner described in Limitation [13i].

17. Claim 17: "The method as recited in claim 15, wherein the first content server is the same as the second content server."

329. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13h]. *See* Section IX.A.13.i.

330. As discussed, it would have been obvious for Glines to manage more than one request for content, including multiple requests for content. *See* Limitations [13p], [13g]. Glines's resolution process would have been handled in the same way for each of the requests, and Glines, with or without Gupta, would have rendered this limitation obvious as explained in Limitations [13h]-[13j].

331. This corresponds with the objectives of Glines, including providing content distribution "as a service where client *requests* are automatically routed to

the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those requests. Repeating the same process twice is an obvious and expected implementation of Glines’s system. This would have included Glines providing the same user content at different times, or content to different users.

18. Claim 18

- a. **[18p] “A content delivery system for delivering content on behalf of a plurality of content providers, each of said plurality of content providers having a hostname associated therewith, the system comprising:”**

332. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13p]. *See* Section IX.A.13.a.

333. Limitation [18p] and [13p] differ only in that Limitation [18p] is a “content delivery system” as opposed to a “method.” Glines discloses a content delivery system for performing the method as recited in claim 1. This includes, for example, the system features of Figure 1. *See* Glines, 9:9-21, 10:13-18, 11:14-13:14 (describing the system of Figure 1), Fig. 1.

- b. [18a] “(A) a plurality of Domain Name System (DNS) servers, said plurality of DNS servers sharing a common anycast address,”**

334. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13a], which recite “providing a plurality of Domain Name System (DNS) servers, said plurality of DNS servers sharing a common anycast address,” and “ configuring each DNS server . . . to support content delivery for a plurality of content providers, wherein each of the plurality of content providers has a respective hostname associated therewith.”

335. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13a]. *See* Section IX.A.13.b.

- c. [18b] “said plurality of DNS servers being authoritative for said hostnames associated with said plurality of content providers;”**

336. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13b], which recites “configuring each DNS server and respective plurality of content servers to support content delivery for a plurality of content providers, wherein each of the plurality of content providers has a respective hostname associated therewith.”

337. As discussed, it would have been obvious for Glines to manage more content for more than one content provider. *See* Limitation [13p]. Doing so would

have been expected and well within the capabilities of a POSITA to, for example, maximize the usefulness and value of Glines's system by providing Glines's service to more than one customer/content provider.

338. Glines's resolution process would have been handled in the same way for requests for different content from different content providers, and Glines, with or without Gupta, would have rendered this limitation obvious as explained in Limitations [13b].

339. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitations [13b]. *See* Section, IX.A.13.c.

d. [18c] “(B) a set of content servers associated with each DNS server, each set comprising at least a plurality of content servers; and”

340. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13a], which recites “wherein each DNS server is associated with a respective plurality of content servers.”

341. A “set” of content servers associated with each DNS server, where each set includes a plurality of content servers, is indistinguishable from each DNS server is associated with a respective plurality of content servers, as recited in limitation [13a].

342. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitations [13a]. *See* Section, IX.A.13.b.

e. [18d] “(C) at least one switch associated with each set of content servers, wherein”

343. For the reasons explained in claim 11 and limitations [13e] and [13i], Glines, alone or in combination with Gupta, would have rendered this limitation obvious. *See* Sections IX.A.13.f, IX.A.13.j.

344. Each of claim 11 and limitations [13e] and [13i] require a switch used to select among a set of content servers.

345. Glines, alone or in combination with Gupta, therefore would have rendered obvious at least one switch associated with each set of content servers as claimed. *Supra* Claim 11, Limitations [13e], [13i].

346. For example, Glines teaches or at least suggests that its Layer 4 switch is associated with its name server/content server peer groups. *See* Glines, 18:20-19:16 (disclosing that each name server and content server “service group” may include fallover/fallback policies to avoid issues with content server outages or overloads, including, “within the service group, standard load balancing” using “layer 4 switches”). Glines therefore discloses a switch associated with a set of content servers.

347. Combining Glines with Gupta would not alter this analysis

- f. **[18e] “(D) responsive to a request to resolve a hostname associated with a particular content provider of said plurality of content providers, said request being associated with an end-user request for content of said particular content provider, one of said plurality of DNS servers is operable to resolve said hostname to identify an IP address for use by the end-user to retrieve the content from a content server,”**

348. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitations [13c] and [13d]. *See* Section IX.A.13.d, IX.A.13.e.

349. Limitation [13c] recites “responsive to a first request for first content associated with a first content provider of said plurality of content providers, said first request including at least a first hostname associated with the first content provider.”

350. Limitation [13d] recites “by a first DNS server of said plurality of DNS servers, resolving said first hostname to identify a first IP address to be used for retrieval of the first content from a first content server of said content servers associated with said first DNS server.”

351. Each of Limitations [13c] and [13d] require one of the DNS servers to resolve the hostname by identifying an IP address in response to a user’s request.

352. Glines, alone or in combination with Gupta, accordingly would have rendered obvious Limitation [18e]. *Supra* Limitations [13c], [13d].

353. Combining Glines with Gupta would not alter this analysis.

- g. **[18f] “(E) the at least one switch associated with the set of content servers of the resolving DNS server is operable to perform load balancing to select the content server from the set of content servers associated with the resolving DNS server, and”**

354. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 10, claim 11, and Limitation [13e]. *See* Sections IX.A.10, IX.A.11, IX.A.13.f;

355. Claim 10 recites “wherein selection of the content server is based on a load balancing policy.”

356. Claim 11 recites “by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.”

357. Limitation [13e] recites “by a switch associated with said resolving DNS server . . . , selecting the first content server from said respective plurality of content servers associated with the DNS server”

358. Each of claims 10, 11 and Limitation [13e] require selection of a content server based on load balancing, and doing so by a switch.

359. Therefore, Glines, alone or in combination with Gupta, satisfies this limitation as explained in claim 10, claim 11, and Limitation [13e].

360. Combining Glines with Gupta would not alter this analysis.

- h. [18g] “(F) the content server is operable to serve the content to the end-user.”**

361. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 6. *See* Section IX.A.7. This limitation is substantively identical to claim 6, addressed above, and would have been obvious for the same reasons

19. Claim 19

- a. [19p] “A method of content delivery, the method comprising:”**

362. This limitation is verbatim and therefore subsumed into limitation [1p]. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1p]. *See* Section IX.A.2.a.

- b. [19a] “(A) providing a plurality of CDN Domain Name System (DNS) servers, said plurality of DNS servers sharing a common anycast address, each DNS server being associated with a respective plurality of content servers,”**

363. This limitation is substantively identical to Limitation [1a], which recites “providing a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), said plurality of CDN DNS servers sharing a common anycast address, wherein each CDN DNS server is associated with a respective plurality of content servers.”

364. The only difference is that this limitation recites a plurality of “CDN Domain Name System (DNS) servers” instead of a plurality of “Domain Name System (DNS) servers associated with a Content Delivery Network (CDN).”

365. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1a]. *See* Section IX.A.2.b.

366. Combining Glines with Gupta would not alter this analysis.

c. [19b] “wherein at least some of said plurality of content servers are located at network Points of Presence (POPs);”

367. This limitation is substantively identical to claim 2, which recites “wherein at least some of said plurality of CDN DNS servers are located at network Points of Presence (POPs).”

368. The only difference is that claim 2 recites some of the “CDN DNS servers” being located at POPs, while limitation [19b] recites some of the “content servers” being located at POPs.

369. As described with respect to claim 2, Glines and Gupta both a name server and paired content servers being located at POPs. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 2. *See* Section IX.A.3.

- d. **[19c] “(B) causing said plurality of DNS servers to be authoritative for a hostname associated with a content provider by causing said common anycast address to be associated with said hostname;”**

370. This limitation is substantively identical to Limitation [1b], which recites “causing said plurality of CDN DNS servers to be authoritative for a hostname associated with a content provider by causing said common anycast address to be associated with said hostname.”

371. Limitation [1b]’s “plurality of CDN DNS servers” is indistinguishable from limitation [19b]’s “plurality of DNS servers.”

372. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1b]. *See* Section IX.A.2.c.

- e. **[19d] “(C) responsive to a request for content associated with the content provider by a client, said request including at least said,”**

373. This limitation is substantively identical to Limitation [1c], which recites “responsive to a request for content associated with the content provider and issued by a client, said request including at least said hostname.”

374. Limitation [1c]’s “request for content . . . issued by a client” is indistinguishable from limitation [19d]’s “request for content . . . by a client.”

375. For the purposes of this Declaration, I also assume “said request including at least said” of limitation [19d] intends to recite “at least said hostname”

and is therefore taught by the same prior art as limitation [1c]’s “said request including at least said hostname.”

376. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1c]. *See* Section IX.A.2.d.

- f. [19e] “(c1) by one of said plurality of DNS servers, resolving said hostname to identify an IP address for use by the client to retrieve the content from a content server; and”**

377. This limitation is substantively identical to Limitation [1e], which recites “by one of said plurality of CDN DNS servers, resolving said hostname to identify an IP address for use by the client to retrieve the content from a content server.”

378. Limitation [1e]’s “plurality of CDN DNS servers” is indistinguishable from limitation [19e]’s “plurality of DNS servers.”

379. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1e]. *See* Section IX.A.2.f.

- g. [19f] “(c2) by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving DNS server,”**

380. This limitation is substantively identical to claim 11, which recites “by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.”

381. Claim 11’s “CDN DNS” is indistinguishable from Limitation [19f]’s “DNS server.”

382. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 11. *See* Section IX.A.11.

- h. [19g] “(D) by the content server, (d1) attempting to serve the content by:”**

383. This limitation is substantively identical to claim 6, which recites “by the content server, attempting to serve the content,”

384. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 6. *See* Section IX.A.7.

- i. [19h] “(d11) if a valid version of the content is available on the content server, serving the content; otherwise (d12) obtaining the content from a content source and then serving the content.”**

385. This limitation is substantively identical to claim 7, which recites “(d1) if a valid version of the content is available on the content server, serving the content;

otherwise (d2) obtaining the content from a content source and then serving the content,”

386. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 7. *See* Section IX.A.8.

20. Claim 20: “The method as recited in claim 19, wherein selection of the content server is based on a load balancing policy.”

387. This limitation is substantively identical to claim 10, which recites “wherein selection of the content server is based on a load balancing policy.”

388. Each of claims 10, 12, and 14 also require selection of a content server based on a load balancing policy. Glines, alone or in combination with Gupta, accordingly would have rendered obvious selection of the content server based on a load balancing policy as claimed. *See* Section IX.A.10 (claim 10), IX.A.12 (claim 12), IX.A.14 (claim 14).

21. Claim 21

a. [21p] “A method of content delivery in a network, the method comprising:”

389. This limitation is a verbatim copy of, and therefore is subsumed into, Limitation [1p]. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1p]. *See* Section IX.A.2.a.

- b. **[21a] “(A) providing a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), the plurality of DNS servers sharing a common anycast address, wherein each DNS server is associated with a set of content servers, each set comprising at least a plurality of content servers;”**

390. This limitation is substantively identical to Limitation [1a], which recites “providing a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), said plurality of CDN DNS servers sharing a common anycast address, wherein each CDN DNS server is associated with a respective plurality of content servers.”

391. Limitation [21a]’s “DNS servers” and “set of” content servers is indistinguishable from limitation [1a]’s “CDN DNS servers” and “respective plurality” of content servers.

392. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1a]. *See* Section IX.A.2.b; *see also* Section IX.A.18.d (Limitation [18c], reciting a “set of” content servers).

- c. **[21b] “(B) causing the plurality of DNS servers to be authoritative for a hostname associated with a content provider by causing the common anycast address to be associated with the hostname;”**

393. This limitation is substantively identical to Limitation [1b], which recites “causing said plurality of CDN DNS servers to be authoritative for a

hostname associated with a content provider by causing said common anycast address to be associated with said hostname.”

394. As explained above, Limitation [21b]’s “DNS servers” is indistinguishable from Limitation [1b]’s “CDN DNS servers.”

395. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1b]. *See* Section IX.A.2.c.

- d. **[21c] “(C) responsive to a request for content associated with the content provider and issued by a client, the request including at least the hostname,”**

396. This limitation is substantively identical to Limitation [1c], which recites “said” request instead of Limitations [21c]’s “the” request.

397. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1c]. *See* Section IX.A.2.d.

- e. **[21d] “(c1) by one of the plurality of DNS servers, resolving the hostname to identify an IP address for use by the client to retrieve the content from a content server; and”**

398. This limitation is substantively identical to Limitation [1e], which recites “by one of said plurality of CDN DNS servers, resolving said hostname to identify an IP address for use by the client to retrieve the content from a content server.”

399. As explained above, Limitation [21d]’s “DNS servers” is indistinguishable from Limitation [1e]’s “CDN DNS servers.”

400. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1e]. *See* Section IX.A.2.f.

- f. **[21e] “(c2) by at least one switch associated with the set of content servers of the resolving DNS server, selecting the content server from the set of content servers associated with the resolving DNS server.”**

401. This limitation is substantively identical to claim 11, which recites “by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.”

402. As discussed with respect to limitation [18d], a POSITA would have understood that Glines’s disclosure of a Layer 4 switch applies to each name server and peered set of content servers to load balance each set of content servers. *See* Glines, 19:13-16. Glines therefore discloses a switch associated with a set of content servers.

403. Limitation [21e]’s “DNS servers” is also indistinguishable from claim 11’s “CDN DNS.”

404. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 11. *See* Section IX.A.11.

22. Claim 22

a. [22p] “A content delivery system comprising a plurality of nodes, wherein each node comprises:”

405. Glines, alone or in combination with Gupta, would have rendered this limitation obvious as explained in Limitations [1p] and [1a]. *See* Sections IX.A.2.a, IX.A.2.b.

406. The '112 patent discloses that a node (e.g., “node 76”) includes a DNS server 78 and cache server 80. EX-1001, 6:65-32, Fig. 5. Each of Glines’s server groups likewise include name and content servers and therefore discloses a “node.” Glines, 10:13-18, 12:11-23, 13:3-7 13:3-7 (“Each name server 53 and file server 51 associated with particular content server 50 are considered to be connected in a peering arrangement” and are “on a common local area network segment sharing the same internetwork device.”).

407. Glines also discloses a content delivery system for performing the method as recited in claim 1. This includes, for example, the system features of Figure 1. *See* Glines, 9:9-21, 10:13-18, 11:14-13:14 (describing the system of Figure 1), Fig. 1.

408. Limitations [1p] and [1b] also require a plurality of DNS servers and associated content servers. Accordingly, Glines, alone or in combination with Gupta, would have rendered obvious selection a plurality of nodes, as claimed. *Supra* Sections IX.A.2.a, IX.A.2.b.

409. Combining Glines with Gupta would not alter this analysis.

b. [22a] “a Domain Name System (DNS) server sharing an anycast address with DNS servers in other nodes of the content delivery system,”

410. This limitation is substantively identical to Limitation [1a], which recites “providing a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), the plurality of CDN DNS servers sharing a common anycast address, and where each CDN DNS server is associated with a respective plurality of content servers.”

411. The same disclosures of Glines, alone or in combination with Gupta, render obvious limitation [22a]’s “sharing an anycast address with DNS servers in other nodes of the content delivery system” and limitation [1a]’s “a plurality of Domain Name System (DNS) servers . . . sharing a common anycast address.”

412. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1a]. *See* Section IX.A.2.b.

c. [22b] “wherein the DNS server is authoritative for hostnames associated with a plurality of content providers;”

413. As discussed in Limitation [13p], Glines expressly discloses that its system may be used with “multiple providers,” and adding additional content providers would have been obvious. *See* Section IX.A.13.a (Limitation [13p]).

414. Gupta likewise discloses multiple content providers and it would have been obvious to apply these teachings to Glines's system. *See* Section IX.A.13.a (Limitation [13p]).

415. Limitation [13p] requires delivering content on behalf of a plurality of content providers. Accordingly, Glines, alone or in combination with Gupta, would have rendered obvious Limitation [22b]. *See* Section IX.A.13.a (Limitation [13p]).

416. Combining Glines with Gupta would not alter this analysis

d. [22c] “a plurality of content servers; and”

417. This limitation is substantively identical to Limitation [1a], which recites “each CDN DNS server is associated with a respective plurality of content servers.”

418. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1a]. *See* Section IX.A.2.b.

419. Combining Glines with Gupta would not alter this analysis.

e. [22d] “at least one switch, wherein”

420. This limitation is substantively identical to claim 11, which recites “at least one switch.”

421. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 11. *See* Section IX.A.11.

422. Combining Glines with Gupta would not alter this analysis.

- f. **[22e] “responsive to a request by a client for content associated with a particular content provider of the plurality of content providers, the request including at least a hostname associated with the particular content provider,”**

423. This limitation is substantively identical to Limitation [13c], which recites “responsive to a first request for first content associated with a first content provider of said plurality of content providers, said first request including at least a first hostname associated with the first content provider.”

424. A POSITA would have understood that Limitation [13c]’s “first request,” “first content,” “first content provider of the plurality of content providers,” and “first hostname” are indistinguishable from limitation [22e]’s “request,” “content associated with a particular content provider of the plurality of content providers,” and “hostname.”

425. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13c]. *See* Section IX.A.13.d.

426. Combining Glines with Gupta would not alter this analysis.

- g. [22f] “the DNS server is operable to resolve the hostname to identify an IP address for use by the client to retrieve the content from a content server,”**

427. This limitation is substantively identical to Limitation [13d], which recites “by a first DNS server ... , resolving said first hostname to identify a first IP address to be used for retrieval of the first content from a first content server of said content servers associated with said first DNS server.”

428. A POSITA would have understood that Limitation [13d]’s “first DNS server,” “first hostname,” “first IP address,” “first content,” and “first content server” are indistinguishable from limitation [22f]’s “DNS server,” “hostname,” “IP address,” “content,” and “content server.”

429. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13d]. *See* Section IX.A.13.e.

430. Combining Glines with Gupta would not alter this analysis.

- h. [22g] “the at least one switch is operable to select the content server from the plurality of content servers, and”**

431. This limitation is substantively identical to claim 11, which recites “by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.”

432. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 11. *See* Section IX.A.11.

433. Combining Glines with Gupta would not alter this analysis.

i. [22h] “the content server is operable to serve the content to the client.”

434. This limitation is substantively identical to Limitation [13f], which recites “attempting to serve said first content from the first content server.” As discussed in claim 6, because Glines discloses its content server returning the requested content to the client, Glines discloses the content server *attempting* to serve the content. This is indistinguishable from the content server is operable to serve the content to the client, as claimed.

435. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [13f]. *See* Section IX.A.13.g.

436. Combining Glines with Gupta would not alter this analysis.

23. Claim 23:

a. [23p] “A method comprising:”

437. This limitation is substantively identical to Limitation [13p], which recites “[a] method of content delivery for delivering content on behalf of a plurality of content providers, ... comprising.”

438. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious as explained in Limitation [13p]. *See* Sections IX.A.13.a.

439. Combining Glines with Gupta would not alter this analysis.

- b. **[23a] “(A) providing a plurality of Domain Name System (DNS) servers, said plurality of DNS servers sharing a common anycast address, wherein each DNS server is associated with a respective plurality of content servers,”**

440. This limitation is verbatim of Limitation [13a], which recites “providing a plurality of Domain Name System (DNS) servers, said plurality of DNS servers sharing a common anycast address, wherein each DNS server is associated with a respective plurality of content servers.”

441. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious as explained in Limitation [13a]. *See* Sections IX.A.13.b.

442. Combining Glines with Gupta would not alter this analysis.

- c. **[23b] “wherein each DNS server and respective plurality of content servers is configured to support content delivery for a plurality of content providers;”**

443. This limitation is substantively identical of Limitation [13b], which recites “configuring each DNS server and respective plurality of content servers to support content delivery for a plurality of content providers.”

444. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious as explained in Limitation [13b]. *See* Sections IX.A.13.c.

445. Combining Glines with Gupta would not alter this analysis.

- d. **[23c] “(B) responsive to a first request for first content associated with a first content provider of said plurality of content providers, said first request including at least a first hostname associated with the first content provider,”**

446. This limitation is verbatim Limitation [13c], which recites “responsive to a first request for first content associated with a first content provider of said plurality of content providers, said first request including at least a first hostname associated with the first content provider.”

447. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious as explained in Limitation [13c]. *See* Sections IX.A.13.d.

448. Glines, alone or in combination with Gupta, additionally renders Limitation [23c] obvious based on Internet content as of the priority date of the ’112 patent (October 18, 2001). For example, it would have been obvious for Glines’s name servers to respond to an anycast message based on a first request for first content associated with a first content provider that includes a request for a news

aggregator website, such as Yahoo! News (e.g., <http://dailynews.yahoo.com>), as of this date. *See* EX-1024.

449. News aggregator websites, including Yahoo! News, were known in the art before October 18, 2001. *See id.* These websites, such as Yahoo! News, were known to provide links to articles from different news organizations via one webpage for users to browse. *See id.* The links typically would have included a hostname and a path to a particular news source. In EX-1024, for example, the link to “U.S. Senate Approves Ashcroft, 58-42,” is associated with the URL http://dailynews.yahoo.com/h/nm/20010201/ts/congress_ashcroft_dc_71.html, where the hostname is “<http://dailynews.yahoo.com>,” and the path is “[h/nm/20010201/ts/congress_ashcroft_dc_71.html](http://dailynews.yahoo.com/h/nm/20010201/ts/congress_ashcroft_dc_71.html).” *See* EX-1024; EX-1025. The hostname is associated with Yahoo! News, while the path is associated with news content from Reuters. *See id.*

450. Applying Glines, alone or in combination with Gupta, to respond to a first request for the URL <http://dailynews.yahoo.com> would have been an obvious application of Glines’s system. The first content would have been the Yahoo! News webpage (e.g., a first content provider of said plurality of content providers), and the first request would have included a first hostname associated with the first content provider (e.g. <http://dailynews.yahoo.com>).

451. Submitting a first request to a news aggregator website such as this would have been nothing more than applying the art in understood ways to resolve particular content. It would have involved nothing more than combining known prior art techniques with predictable results, for example combining known content available via the Internet with Glines's content delivery techniques (with or without Gupta), which resolve and deliver Internet content. *See* Glines, 1:4-20, 9:9-21. This would have required no modifications of Glines or Gupta to implement. Instead, this would have involved one exemplary and known content request for Glines's system to respond to, with or without Gupta. Resolving this particular content request is indistinguishable from Glines's exemplary request for content via the URL "http://www.example.com/homepage.html." *See id.*, 6:23-7:24, 13:15-15:10.

452. Glines, with or without Gupta, accordingly would have rendered this limitation obvious, in addition to the reasons explained in Limitation [13c].

453. Combining Glines with Gupta would not alter this analysis.

- e. **[23d] "by a first DNS server of said plurality of DNS servers, resolving said first hostname to identify a first IP address of a content server of the plurality of content servers associated with said first DNS server; and"**

454. This limitation is substantively identical to Limitation [13d], which recites "by a first DNS server of said plurality of DNS servers, resolving said first hostname to identify a first IP address to be used for retrieval of the first content

from a first content server of said content servers associated with said first DNS server.”

455. Identifying “a first IP address of a content server of the plurality of content servers” is indistinguishable from identifying “a first IP address to be used for retrieval of the first content from a first content server of said content servers.”

456. It further would have been obvious to resolve the first request discussed in Limitation [23c] using Glines’s content delivery network. This would have been nothing more than applying the art in understood ways for resolve particular content, and combining known prior art resolve techniques with known content available via the Internet, with predictable results. This would have required no modifications of Glines or Gupta to implement. Instead, this would have been one exemplary content request in Glines’s system to respond to, with or without Gupta.

457. Glines, alone or in combination with Gupta, also would have rendered this limitation obvious as explained in Limitation [13d]. *See* Sections IX.A.13.e.

458. Combining Glines with Gupta would not alter this analysis.

f. [23e] “(C) responsive to a second request for second content associated with a second content provider of said plurality of content providers, said second request including at least said first hostname associated with the first content provider,”

459. Glines, alone or in combination with Gupta, would have rendered this limitation obvious.

460. This limitation recites a second request for “second” content from a “second” content provider, where the second request includes the “first” hostname associated with the “first” content provider.

461. Glines, alone or in combination with Gupta, renders this feature obvious. For example, it would have been obvious for Glines’s name servers to respond to an anycast message based on a second request for second content associated with a second content provider that includes a request for a particular news article via a news aggregator website, such as Yahoo! News. *See* EX-1024; *see also* Limitation [23c]. As shown in EX-1025, it would have been obvious to respond to a request for a news article via Yahoo! News, such as the article “U.S. Senate Approves Ashcroft, 58-42” at http://dailynews.yahoo.com/h/nm/20010201/ts/congress_ashcroft_dc_71.html, where the hostname is “http://dailynews.yahoo.com,” and the second content is a news article. *See* EX-1025. The hostname is associated with the first content provider (Yahoo! News), and the second content is associated with a second content provider (e.g., Reuters). *See id.*

462. It would have been obvious for Glines, with or without Gupta, to respond to two requests for content, including a first request via the URL <http://dailynews.yahoo.com>, and a second request via the URL http://dailynews.yahoo.com/h/nm/20010201/ts/congress_ashcroft_dc_71.html. *See*

EX-1024; EX-1025. Both requests would include the first hostname associated with the first content provider (e.g., <http://dailynews.yahoo.com> associated with Yahoo! News).

463. As discussed, repeating a process for the same or different content is an obvious and expected implementation of Glines's system. *See* Limitation [13g]. This includes Glines providing a user content at different times, or content to different users. Indeed, Glines discloses providing content distribution "as a service where client requests are automatically routed to the closest available content server." Glines 6:3-15; *id.*, Abstract, 9:9-17 ("requests" for content), 10:5-12 (handling "the number of requests"). Similar or repeated processing would occur to fulfill those "requests."

464. Resolving a second request such as this would have been nothing more than applying the art in understood ways to resolve particular content. It would have involved nothing more than combining known prior art techniques with predictable results, for example combining known content available via the Internet with Glines's content delivery techniques (with or without Gupta), which resolves and delivers Internet content. *See* Glines, 1:4-20, 9:9-21. This would have required no modifications of Glines or Gupta to implement. Instead, this would have involved one exemplary and known content request for Glines's system to respond to, with or without Gupta. Resolving this particular content request is indistinguishable from

Glines's exemplary request for content via the URL

“http://www.example.com/homepage.html.” *See id.*, 6:23-7:24, 13:15-15:10.

465. Combining Glines with Gupta would not alter this analysis.

- g. [23f] “by a second DNS server of said plurality of DNS servers, resolving said first hostname to identify a second IP address associated with a content server of the plurality of content servers associated with the second DNS server.”**

466. This limitation is substantively identical of Limitation [13h], which recites “by a second DNS server of said plurality of DNS servers, resolving said first hostname to identify a second IP address for retrieval of the first content from a content server associated with the second DNS server.”

467. Identifying “a second IP address associated with a content server of the plurality of content servers” is indistinguishable from identifying “a second IP address for retrieval of the first content from a content server.”

468. It further would have been obvious to resolve the second request discussed in Limitation [23e] in this way using Glines’s content delivery network.

469. As discussed in Limitation [13h], it would have been obvious to resolve a second request for particular content via the same or different name server in Glines (e.g., via the same or different DNS server 52 and a different associated content server 51). Glines, 14:17-16:8, Fig. 1.

470. For example, if a first name server is overloaded or offline, or if a second name server is available that is “closer” to the user sending the second request, it would have been obvious to use the second name server in Glines’s network 10 to resolve the content. *Id.*, 14:17-16:8, 18:13-19:20 (disclosing that, “if the host or gated dies unexpectedly, then failover needs to happen,” including “[l]oad balancing across service groups” to “load shed to content servers in other service groups by returning their IP address to name lookups coming at the anycast address.”), 15:11-14 (“the anycast datagram will reach one of the name servers 53-1, 53-2, and 53-3 in the group associated with IP address 50.100.20.1, the one reached first being the one closest to the requesting client 12 or DNS resolver 18.”).

471. The second name server in Glines would respond to the anycast address by identifying an associated content server 51, and thus satisfy identifying a “second IP address” that is different than the “first IP address” provided in response to the second request. Indeed, Glines contemplates this with respect to its system in Figure 1. *See* Glines, 14:17-16:8, Fig. 1

472. This would have been nothing more than applying the art in understood ways to resolve particular content, and combining known prior art resolution techniques with known content available via the Internet, with predictable results. This would have required no modifications of Glines or Gupta to implement. Instead,

this would have been one exemplary content request in Glines's system to respond to, with or without Gupta.

473. Glines, alone or in combination with Gupta, also would have rendered this limitation obvious.

474. Combining Glines with Gupta would not alter this analysis.

B. Grounds 2A and 2B Render Obvious Claim 4

1. Motivation to Combine Glines with Chung (EX-1004), and Glines-Gupta with Chung

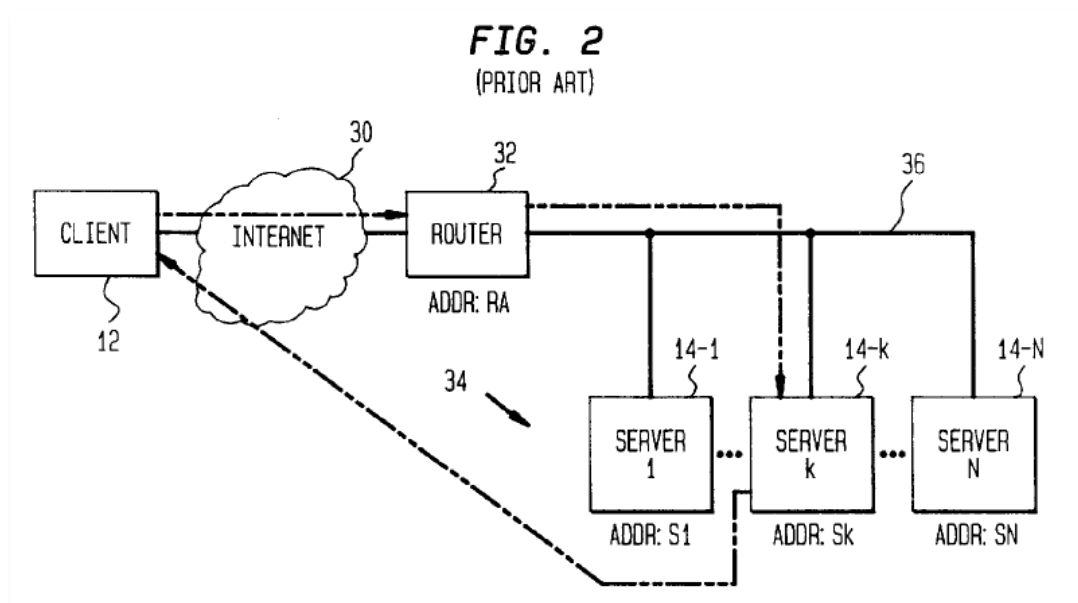
475. With the expansion of the Internet, hosting content on multiple servers, including server clusters, was known and understood in the prior art. Chung, 1:15-40, 2:33-3:10; *see also id.*, Abstract, 3:65-4:17, 5:39-6:15.

476. To meet the demand, Chung discloses a server cluster, which is “a group or set of servers interconnected or otherwise configured to host a network service.” *Id.*, 5:47-52. The server cluster advertises a single cluster address for the servers in the cluster to support the service, for example for serving content in response to client requests. *Id.*, 1:29-33 (“Such a server cluster should preferably publicize only one server name for the entire cluster so that any configuration change inside the cluster does not affect client applications.”), 2:33-3:10, Fig. 2.

477. Chung discloses “[t]here are many known techniques for distributing HTTP client requests to a cluster of servers,” including “publicizing the address . . .

of the server-side router 32 [associated with a cluster of servers] to the clients via the DNS.” *Id.*, 2:33-3:10, Fig. 2.

478. As shown below in Figure 2, a client request is directed to the router 32, “which then dispatches the request to a selected server 14-k of server cluster 34 based on load characteristics, as indicated by the dashed line connecting client 12 to server 14-k via router 32.” *Id.* The selected server 14-k then responds to the client “as indicated by the dashed line connecting server 14-k to client 12,” where the selected server changes the source IP address in its reply packets to the router address RA. *Id.* This makes the transaction “totally transparent to the clients.” *Id.*



Id., Fig. 2.

479. A POSITA would have found it obvious to combine Glines, with or without Gupta, with Chung.

480. For example, a POSITA would have combined Chung's use of a server cluster with Glines's system such that each of Glines's name servers includes a plurality of content servers grouped as a cluster.

481. The content servers and name server in Glines would be connected, as both Glines and Chung teach, to a common LAN 36 (*see* Glines, 13:3-8; Chung, 2:43-46, Fig. 2), and the particular content server in the cluster would be selected by a switch or router, as taught by Chung, (*see* Chung, 2:33-3:10).

482. Chung explains the benefits of doing so, including increasing performance of the system to provide content to users, allowing individual server failures or maintenance to not affect the operation of the system, and to increase capacity and scale. *See* Chung, 1:16-40 ("A single server hosting a service is usually not sufficient to handle this . . . aggressive growth [in popular Internet web sites that are heavily loaded with client requests]."); *see also id.* ("Such a server cluster should preferably publicize only one server name for the entire cluster so that any configuration change inside the cluster does not affect client applications.").

483. Cluster computing was well known in the art and a POSITA would have looked to Chung to improve Glines and Glines-Gupta. *See* Section IX.A.4 (claim 4); EX-1015 at 9-11 ("Such a system can provide a cost-effective way to gain features and benefits (fast and reliable services) that have historically been found only on more expensive proprietary shared memory systems.").

484. In doing so, the systems serving content can be more easily scaled using commodity machines, and do so in a transparent manner to the user. *See id* (“A pool of servers tied together to act as a single unit, or server clustering, provides [] incremental scalability” to grow capacity with demand), 39 (“[A] dispatcher receives service requests from clients and selects a server from the server pool to process the request. Depending on the clustering technology, the dispatcher appears as either a switch (processing incoming data only) or a network gateway (processing incoming and outgoing data) to the servers in the pool.”); EX-1011, 102-103 (“An alternative form of cooperative caching involves clusters of caches, often collocated on the same LAN. A single cache is often replaced with a cluster of caches when the single cache is not sufficient to handle the traffic or provide sufficient storage capacity.”).

485. A POSITA would have understood that Glines and Glines-Gupta were compatible with the teachings of Chung.

486. For example, Chung teaches resolving a DNS request to a common cluster address and then responding for the request for content from a particular server in the cluster server (e.g., one of server 14-1 through 14-N). *See* Chung, 2:33-3:10, Fig. 2. Glines and Glines-Gupta similarly include name servers that are addressable by an anycast address and provides the IP address of a particular content server from which the user can directly retrieve the content. *See* Sections IX.A.2 (claim 1).

487. A POSITA also would have understood that Chung contemplates “dispatching techniques” to distribute requests among its cluster of servers when delivering contents, and that Chung’s techniques are just like those contemplated in Glines. *See, e.g.*, Chung, 2:51-61 (“The [client] request is directed to the router 32, which then dispatches the request to a selected server 14-k of server cluster 34 based on load characteristics . . . ”); Glines, 19:13-16 (“[W]ithin the service group, standard load balancing and replication techniques can also apply”). A POSITA therefore would have understood that Chung’s cluster is compatible with the content distribution methods and architecture of Glines, and in view of this compatibility, would have been motivated to implement Chung’s cluster with Glines (or Gline’s-Gupta) at least in view of the benefits discussed above.

488. Combining Chung’s use of a cluster with Glines’s/Glines’s-Gupta’s replica content servers 50-1, 50-2, and 50-3, which include a name server 53 and content servers 51, would have been nothing more than combining known features in the art according to understood principles to yield predictable results.

489. The result would have been a cluster of servers responding to the request for content after one of the anycast DNS servers resolves the hostname to the cluster address, and then providing content in Glines using the principles of Chung’s server cluster 34. A POSITA would have had an expectation of success in

the combination because deploying multiple servers as a server cluster was known and understood in the art. *See* Chung, 1:16-40.

490. Indeed, Glines suggests deploying its name server and associated content servers as a server cluster by disclosing these are arranged in a peering arrangement and “operate quite closely together and, in fact, are preferably located physically near one another, such as on a common local area network segment sharing the same internetwork device 16-50-1.” Glines, 13:3-8, Fig. 1.

491. Glines also discloses that the internetwork device may be a “router,” similar to Chung. *Id.*, 15:23-25, 20:1-2, 7:14-18, 11:18-20. A POSITA thus would have understood Glines suggests deploying its system as Chung expressly discloses: using a router (e.g., internetwork device 16-50-1) to connect users with a server cluster (e.g., Glines’s peered name server and content servers “operat[ing] quite closely together.”). This would have improved the capacity, scalability, and performance of Glines’s system. *See* Section IX.A.5 (claim 4).

2. Claim 4: “The method as recited in claim 2 wherein at least one of said plurality of CDN DNS servers and respective plurality of content servers is grouped as a cluster.”

492. Glines, alone or in combination with Gupta, discloses claim 2, from which claim 4 depends.

493. Glines, with or without Gupta, in further combination with Chung, would have rendered this limitation obvious.

494. Hosting content on multiple servers, including server clusters, was known and understood in the prior art. Chung, 1:15-40, 2:33-3:10; *also id.*, Abstract, 3:65-4:17, 5:39-6:15. For example, Chung discloses a server cluster that is “a group or set of servers interconnected or otherwise configured to host a network service.” Chung, 5:47-52. Chung’s server cluster may advertise a single cluster address for its servers in the cluster to support the service, including serving content. *Id.*, 1:29-33, 2:33-3:10, Fig. 2.

495. Glines itself suggests deploying its servers as a cluster by disclosing they are arranged in a peering arrangement and “operate quite closely together and, in fact, are preferably located physically near one another, such as on a common local area network segment sharing the same internetwork device 16-50-1.” Glines, 13:3-8, Fig. 1. Glines also discloses reporting a “cluster address” in response to the anycast message, and that the internetwork device may be a “router,” similar to Chung. *Id.*, 7:18-20, 15:23-25, 20:1-2, 7:14-18, 11:18-20.

496. This at least suggests deploying Glines (and Glines-Gupta) as Chung expressly discloses, including using a router (e.g., internetwork device 16-50-1) to connect users with a server cluster. This would have improved the capacity, scalability, and performance of Glines’s system, as discussed above. *See* Sections IX.A.4 (claim 4 (Grounds 1A/1B)), X.B.1 (Motivation to Combine Chung).

497. A POSITA would accordingly have combined Chung's use of a server cluster with Glines (or Glines and Gupta) so that each name server includes a plurality of content servers grouped as a cluster as claimed.

498. Combining Glines or Glines-Gupta with Chung in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

C. Grounds 3A and 3B Render Obvious Claim 7-9, 19, and 20

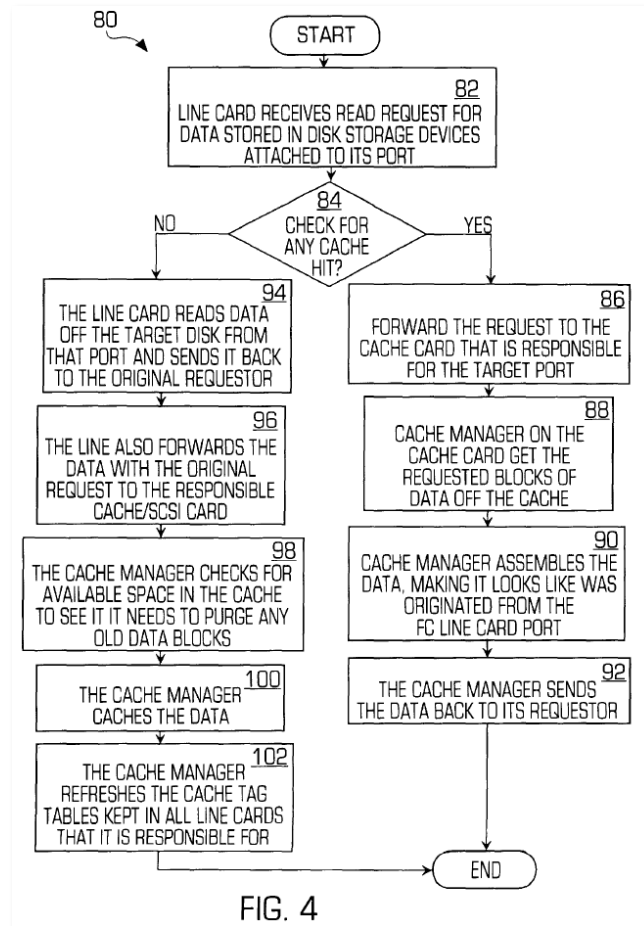
1. Motivation to Combine Glines with Chiou (EX-1005), and Glines-Gupta with Chiou

499. Chiou discloses methods of improving cache coherency in Internet caching systems. Chiou, 1:4-41. To avoid network congestion across the Internet, Chiou discloses that it was common to store frequently used data in a local cache that is "near the requesting host device in order to decrease/eliminate the delay due to network traffic congestion." *Id.*; *see also id.*, 2:15-46, 2:56-67. When data is replicated on cache servers, "[t]he greatest challenge associated with using caches is ensuring data coherency, i.e. to make sure that the data in the cache are always the same as the data stored in the target storage device." *Id.*, 1:4-41. For data that is updated frequently, it is necessary "for the caches to either refresh themselves frequently to obtain the current data or to invalidate the data stored in the cache so that applications reading from the cache will not get 'stale' (e.g., invalid) data from the cache." *Id.*

500. Chiou discloses receiving requests for content, determining if the content is cached, and either serving the content from the cache or requesting the content from an origin or “target” server. *See id.*, 12:52-13:40, Fig. 4. A user requests data and a cache manager checks a “cache tag table” to determine whether the data is available (e.g., determines a cache hit or catch miss in response to the request). *Id.*

501. The cache tag table “contains pointers to the data stored in the cache,” and that “[f]or file-based cache systems, each cache entry is identified by its target IP address, file identifier (or name), file offset, and data length.” *Id.*, 11:33-51, 12:61-67. Chiou discloses that, “[i]f a cache hit occurs (e.g., the requested data is located in the SSC cache ***and is valid data***), then the read request is forward to the cache card that is responsible for the target port.” *Id.*, 13:5-17, Fig. 4. The cache manager then retrieves the requested data from the cache location, assembles it “to make it look more like it was originated from the original target storage device,” and sends the data to the requester. *Id.*

502. If a cache hit does not occur, either because no data is cached or because the cached data is stale, Chiou discloses that the data is read from the “target disk,” e.g., the target location of the requested data. *Id.*, 13:17-40, Fig. 4. The data is then sent back to the requestor. *Id.* The cache manager purges “old cache data” based on, for example, “pre-set time limits,” and stores the received data in cache and updates the cache tag table. *Id.*



Id., Fig. 4.

503. Chiou thus discloses verifying that requested data is cached and “valid” before serving it, or otherwise requesting the data from a “target” source. *Id.*, 12:52-13:40, Fig. 4. This may include determining whether or not the requested content in cache is “stale” data. *Id.* Chiou explains that stale data may be determined based on Time-to-Live (TTL) expiration. *See id.* (using “pre-set time limits” to determine whether to purge “old cache data”), 1:29-40 (stating that it is “necessary for the caches to either refresh themselves frequently to obtain the current data or to invalidate [old] data . . . so that applications reading from the cache will not get

‘stale’ (e.g., invalid) data from the cache.”). Using time duration or TTL techniques, a cache stores content for a predefined duration, after which it is considered stale.

504. It would have been obvious to combine Glines, with or without Gupta, with Chiou to ensure the system serves only valid data stored in content servers.

505. Determining whether cached data is present and valid was common in cache systems, including in response to requested content. This includes using identifiers to verify validity, using time limits on cached data, and other methods. *See, e.g.*, EX-1019, 13:44-15:28 (“For resources that can be cached in multiple locations and that can change over time, cache coherency can be important.”); *id.* (“If a peer 140 or the gate server 120 fetches a resource from a caching publisher, they can check to ensure that the version indication in the resource identifier matches the expected version that was returned as a result of a resource locator request.”); *id.*, (“In one embodiment, caching information about resources published by master publishers . . . can be implemented on a leasing basis, i.e. the cached information can be assumed to be valid for a specified time after it was obtained”); EX-1013, 24:7-51 (“[T]he server 108 can update cached content when it determines that such content has changed or can provide time durations or other form of expiration notification after which the edge cache 508 purges the content.”).

506. In the combination, the name server in Glines that first responds to a user request for content via the anycast address that is the “closest” to the user.

Glines, 15:6-16:8. The response would include an IP address of a content server that will serve the content. *Id.*; *see also* Section IX.A.2.f (Limitation [1e]). The user would then request the content from the content server and the content server in the combined system would determine a cache hit or miss as taught by Chiou. *See* Chiou, 12:52-13:40, Fig. 4. This would have included, for example, determining if “the requested data is located in the [] cache and is valid data.” *Id.*, 13:5-17, Fig. 4.

507. A POSITA would have understood that determining whether data is valid may include verifying the data is stale or not using, for example, TTL “pre-set time limits.” *See id.*, 12:52-13:40, 1:29-40.

508. If the data is either not available or is stale, the content server in the combined system would request the content from Glines’s origin server according to Chiou’s disclosure of reading requested data from a “target disk” and Glines’s disclosure of receiving content from an origin server. *See* Glines, 12:11-23, 13:1-2, 14:17-25, Fig. 1; Chiou, 13:17-40.

509. Combining Chiou with Glines and Glines-Gupta would have been nothing more than combining known features in the prior art according to understood principles (e.g., combining Chiou’s cache hit/miss techniques with Glines’s and Glines-Gupta’s anycast messaging the content distribution system) to yield predictable results.

510. Indeed, Chiou contemplates that its methods may be employed on distributed cache systems, including those serving Internet content from distributed servers. *See* Chiou, 1:5-2:53. Chiou's techniques were compatible with Glines and Glines-Gupta, therefore a POSITA would have had a reasonable expectation of success in the combination.

511. A POSITA also would have combined Chiou with Glines and Glines-Gupta to maintain Glines's use of the "closest" content server and to provide the most up-to-date data to users.

512. The request from a content server for an updated version from Glines's origin server would have been implemented in the combined system either when the name server in Glines identifies the IP address of the content server (e.g., before the user requests the content from the particular content server), or after the user requests the content from the content server. By identifying the closest content server regardless of whether the particular content server has the content in storage, the system would have supported greater data dissemination without the need to push data across the network or to actively manage its validity.

513. For example, each content server would request the most updated information without the origin server having to push the data to each content server, which would otherwise increase bandwidth across the system. While identifying only those content servers that have valid versions of the content would have been

one option, this would have potentially excluded a content server that is “closest” to a user, increasing latency. It would have been equally obvious and understood to maintain Glines’s techniques of identifying the closest content server and thereafter requesting the content from an origin server if the requested content is not available or is stale. Particularly for static content, such as images and webpages that are not frequently updated, using timing techniques such as Chiou’s (e.g., TTL) and determining whether content is valid based on a time limit would have been a simple implementation that reduces the need for content management across Glines’s system, and a POSITA would have been motivated to make this combination.

2. Claim 7: “The method as recited in claim 6 wherein said step (D) of attempting comprises: (d1) if a valid version of the content is available on the content server, serving the content; otherwise (d2) obtaining the content from a content source and then serving the content.”

514. Glines, alone or in combination with Gupta, discloses claim 6, from which claim 7 depends. *See* Section IX.A.7.

515. Glines, with or without Gupta, in further combination with Chiou, would have rendered claim 7 obvious.

516. Chiou discloses confirming the validity of requested content. *E.g.*, Chiou, 12:52-13:40, Fig. 4. Chiou discloses that, “[i]f a cache hit occurs (e.g., the requested data is located in the SSC cache ***and is valid data***), then the read request is forward to the cache card that is responsible for the target port.” *Id.*, 13:5-17,

Fig. 4. If valid, the cache manager retrieves the content from the cache and sends it the requester. *Id.* If the requested content is not valid, Chiou discloses obtaining valid content from a “target” server and serving that content. *Id.*, 12:52-13:40, Fig. 4. This discloses attempting to server content by serving it “if a valid version ... is available on the content server, otherwise “obtaining [it] from a content source and then serving the content,” as claimed.

517. A POSITA would have been motivated to implement these teachings in Glines or Glines-Gupta to ensure only valid content is served and improve user experience. *See* Section IX.C.1 (Motivation to Combine Chiou). As disclosed in Chiou, if a cache hit does not occur (e.g., the requested data is not located in cache or is not “valid data”), the data is requested “from the target disk and sen[t] . . . back to the requestor in step 94.” Chiou, 12:52-13:40.

518. As discussed, in Glines-Chiou and Glines-Gupta-Chiou, the user would request content from a particular content server, which would determine a cache hit or miss as taught by Chiou. *See id.*, 12:52-13:40, 13:5-17 (determining if “the requested data is located in the [] cache and is valid data”), Fig. 4. If the data is either not available or is stale, the content server would request the content from the origin server according to Chiou’s teaching of reading requested data from a “target disk” and Glines’s and Gupta’s teachings regarding receiving content from an origin

server. Chiou, 13:17-40; *supra* Section IX.A.8 (claim 7, regarding teachings of Glines and Gupta).

519. Combining Glines or Glines-Gupta with Chiou in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

3. Claim 8: “The method as recited in claim 7 wherein, said content source is an origin server associated with said content provider.”

520. Glines, alone or in combination with Gupta and/or Chiou, discloses claim 7, from which claim 8 depends. *See* Sections IX.A.8, IX.C.2.

521. Glines, with or without Gupta, in further combination with Chiou, would have rendered this limitation obvious.

522. As explained in claim 7, in the combined Glines-Chiou and Glines-Gupta-Chiou systems, the content server would request the content from the origin server. *Supra* Sections IX.A.8, IX.C.2.

523. For example, Glines discloses a content source that includes origin file server 40. Glines, 2:9-13, 9:16-10:25, 12:11-23, 14:13-25. The origin server is associated with the content provider; for example, Glines’s content provider “example” provides content from “www.example.com.” *Id.*, 13:15-14:25. A POSITA would have understood this corresponds with Chiou’s “target disk” from which content is retrieved if unavailable or invalid in cache. *See* Chiou, 12:52-13:40, Fig. 4. In the combined Glines-Chiou or Glines-Gupta-Chiou system, Glines’s name

server or content server would determine if a valid version of content is available, and if not obtain the content from Glines's origin server. This discloses obtaining the content from a content source where the content source "is an origin server associated with said content provider."

524. For those reasons, Glines-Chiou or Glines-Gupta-Chiou would have rendered this limitation obvious.

525. Combining Glines or Glines-Gupta with Chung in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

4. Claim 9: "The method as recited in claim 7 wherein, if a valid version of the content is not available on the content server, the content server uses a table to ascertain where to obtain the content."

526. Glines, alone or in combination with Gupta and/or Chiou, discloses claim 7, from which claim 9 depends. *See* Sections IX.A.8, IX.C.2.

527. Glines, with or without Gupta, and Chiou, would have rendered this limitation obvious.

528. Chiou discloses a "cache tag table" that includes pointers to cached data. Chiou, 12:52-13:40. For each piece of data stored in cache, the cache tag table includes an entry that provides a "target device address" or "target IP address" associated with the data. *Id.* The "target" in Chiou is the server or location where the data is requested if the data is not located (or is stale) in cache, e.g., the origin server. ("If a cache hit does not occur (e.g., the requested data is not located in the SSC

cache), then the line card reads the requested data *from the target disk* and sends the data back to the requestor in step 94.”); *see also id.*, 2:4-45 (discussing “target web server[s]” and “target devices [that] are geographically separated as in the Internet.”). Chiou discloses that the cache tag table is checked and if there is a cache miss (e.g., the content is stale), the system uses the table to identify the target device from which to obtain the content, as claimed. *Id.*, 12:52-13:40.

529. By disclosing that the cache tag table includes the target device address or IP address, Chiou discloses or at least renders obvious using a table to ascertain where to obtain content if a valid version of the content is not available on a content server, as claimed.

530. For example, a POSITA would have understood that Chiou first consults the cache tag table to determine if content is in cache, and with that determines the target device associated with that content. *See id.*, 12:52-13:40. If the content is in cache is stale (e.g., “not valid”), Chiou would know to read the data from the target associated with the data from the table. *See id.* This teaches using the table to ascertain where to obtain a valid version of the data, or at a minimum renders this feature obvious.

531. A POSITA would have found it obvious to combine Chiou’s cache tag table with Glines, with or without Gupta, for example, because using the target device from the table would have reduced lookup time by determining the source of

data directly and without additional DNS resolution, which can add latency. This also would have made the system more robust against DNS outages, misconfigurations, or delays. The combination would have been nothing more than combining known elements to yield predictable results, and there would have been a reasonable expectation of success in the combination.

532. Combining Glines or Glines-Gupta with Chung in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

5. Claim 19

a. [19p] “A method of content delivery, the method comprising:”

533. This limitation is verbatim and therefore subsumed into limitation [1p]. Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1p]. *See* Section IX.A.2.a.

b. [19a] “(A) providing a plurality of CDN Domain Name System (DNS) servers, said plurality of DNS servers sharing a common anycast address, each DNS server being associated with a respective plurality of content servers,”

534. This limitation is substantively identical to Limitation [1a], which recites “providing a plurality of Domain Name System (DNS) servers associated with a Content Delivery Network (CDN), said plurality of CDN DNS servers sharing a common anycast address, wherein each CDN DNS server is associated with a respective plurality of content servers.”

535. The only difference is that this limitation recites a plurality of “CDN Domain Name System (DNS) servers” instead of a plurality of “Domain Name System (DNS) servers associated with a Content Delivery Network (CDN).”

536. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1a]. *See* Section IX.A.2.b.

537. Combining Glines with Gupta would not alter this analysis.

- a. **[19b] “wherein at least some of said plurality of content servers are located at network Points of Presence (POPs);”**

538. This limitation is substantively identical to claim 2, which recites “wherein at least some of said plurality of CDN DNS servers are located at network Points of Presence (POPs).”

539. The only difference is that claim 2 recites some of the “CDN DNS servers” being located at POPs, while limitation [19b] recites some of the “content servers” being located at POPs.

540. As described with respect to claim 2, Glines and Gupta both a name server and paired content servers being located at POPs. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 2. *See* Section IX.A.3.

- b. [19c] “(B) causing said plurality of DNS servers to be authoritative for a hostname associated with a content provider by causing said common anycast address to be associated with said hostname”**

541. This limitation is substantively identical to Limitation [1b], which recites “causing said plurality of CDN DNS servers to be authoritative for a hostname associated with a content provider by causing said common anycast address to be associated with said hostname.”

542. Limitation [1b]’s “plurality of CDN DNS servers” is indistinguishable from limitation [19b]’s “plurality of DNS servers.”

543. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1b]. *See* Section IX.A.2.c.

- c. [19d] “(C) responsive to a request for content associated with the content provider by a client, said request including at least said,”**

544. This limitation is substantively identical to Limitation [1c], which recites “responsive to a request for content associated with the content provider and issued by a client, said request including at least said hostname.”

545. Limitation [1c]’s “request for content . . . issued by a client” is indistinguishable from limitation [19d]’s “request for content . . . by a client.”

546. For the purposes of this Declaration, I also assume “said request including at least said” of limitation [19d] intends to recite “at least said hostname”

and is therefore taught by the same prior art as limitation [1c]’s “said request including at least said hostname.”

547. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1c]. *See* Section IX.A.2.d.

- d. **[19e] “(c1) by one of said plurality of DNS servers, resolving said hostname to identify an IP address for use by the client to retrieve the content from a content server; and”**

548. This limitation is substantively identical to Limitation [1e], which recites “by one of said plurality of CDN DNS servers, resolving said hostname to identify an IP address for use by the client to retrieve the content from a content server.”

549. Limitation [1e]’s “plurality of CDN DNS servers” is indistinguishable from limitation [19e]’s “plurality of DNS servers.”

550. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in Limitation [1e]. *See* Section IX.A.2.f.

- e. **[19f] “(c2) by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving DNS server,”**

551. This limitation is substantively identical to claim 11, which recites “by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.”

552. Claim 11’s “CDN DNS” is indistinguishable from Limitation [19f]’s “DNS server.”

553. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 11. *See* Section IX.A.11.

- f. **[19g] “(D) by the content server, (d1) attempting to serve the content by:”**

554. This limitation is substantively identical to claim 6, which recites “by the content server, attempting to serve the content,”

555. Therefore, Glines, alone or in combination with Gupta, would have rendered this limitation obvious for the reasons explained in claim 6. *See* Section IX.A.7.

- g. **[19h] “(d11) if a valid version of the content is available on the content server, serving the content; otherwise (d12) obtaining the content from a content source and then serving the content.”**

556. This limitation is substantively identical to claim 7, which recites “(d1) if a valid version of the content is available on the content server, serving the content;

otherwise (d2) obtaining the content from a content source and then serving the content,”

557. Therefore, Glines, alone or in combination with Gupta and/or Chiou, would have rendered this limitation obvious for the reasons explained in claim 7. *See* Section IX.C.2 (Grounds 3A, 3B).

6. Claim 20: “The method as recited in claim 19, wherein selection of the content server is based on a load balancing policy.”

558. Glines, alone or in combination with Gupta and/or Chiou, discloses claim 19, from which claim 20 depends. *See* Sections IX.A.19, IX.C.5.

559. Glines, with or without Gupta, and Chiou, would have rendered this limitation obvious.

560. As explained above, *supra* Section IX.A.20 (claim 20, Grounds 1A, 1B), Glines alone or in combination with Gupta, would have rendered claim 20 obvious. The addition of Chiou to the combination would not have disturbed Glines’s (or Glines-Gupta’s) teaching that content server selection may be based on a load balancing policy. *Id.* Accordingly, Glines, with or without Gupta, and Chiou, would have rendered this limitation obvious.

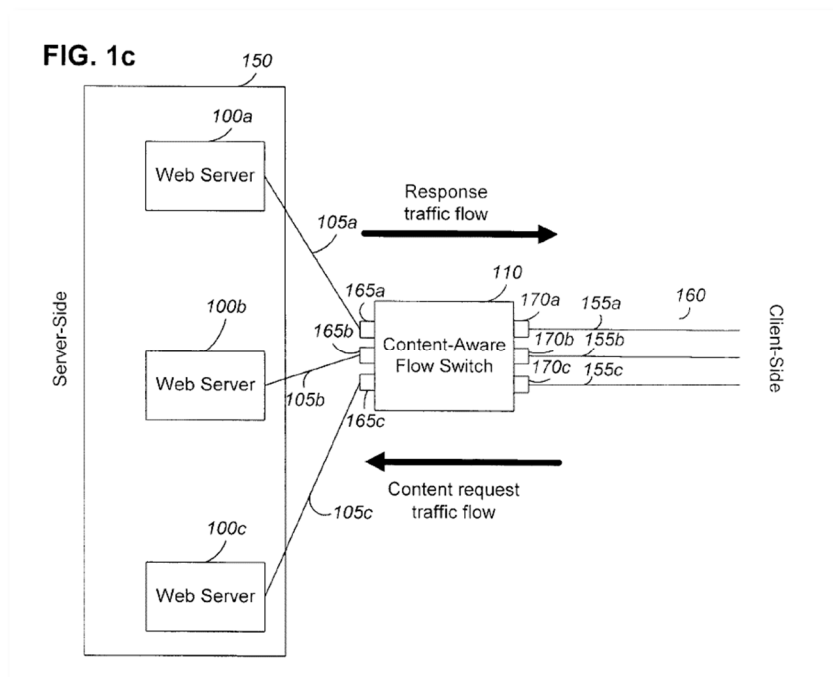
561. Combining Glines or Glines-Gupta with Chung in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

D. Grounds 4A/4B and 5A/5B Render Obvious Claim 10-22

1. Motivation to Combine Glines, Glines-Gupta, Glines-Chiou, or Glines-Gupta-Chiou with Colby (EX-1006)

562. Colby discloses a content-aware flow switch that can ““front-end[]’ (i.e., intercept[] all packets received from and transmitted by) a set of local web servers 100a-c, constituting a web server farm 150.” Colby, 5:43-45.

563. The web servers may be connected to the switch 110 via LAN links 105a-c, and the flow switch 100 “may connect to the network 100 (FIG. 1a) through a router 130” and the Internet through uplinks 155a-c. *Id.*, 5:23-41, 5:62-6:6, Figs. 1b, 1c.



Id., Fig. 1c.

564. Switch 110 may switch packets “based on a combination of source and destination IP addresses, transport layer protocol, and transport layer source and destination port numbers.” *Id.*, 6:7-28. Server loading and network path optimization are also considered “when selecting the most optimal path for a packet.” *Id.*; *see also id.*, Abstract (“The best-fit server is chosen based on the type of content requested, . . . the degree of load on available servers, network congestion information, and the proximity of the client to available servers.”), 20:1-33 (selecting the “best-fit server” based on “server load metrics descriptive of the current load and recent activity on the candidate servers”).

565. To facilitate these policies, Colby discloses a Content Server Database (CSD) that contains information about the load on servers 100a-c. *Id.*, 6:35-67, Fig. 2.

566. One database maintained by the CSD “contains content rules, which are defined by the system administrator and which indicate how the flow switch 110 should handle requests for content.” *Id.*

567. Another database maintained by the CSD contains “content records which are derived from the content rules.” *Id.* These records include “information related to particular content, such as its associated IP address, URL, protocol, layer 4 port number, QoS indicators, and the load balance algorithm to use when accessing the content.” *Id.*

568. Another database maintained by the CSD includes information about particular servers, including “load metrics indicating the load on the server.” *Id.* Using this information, the CSD is at least in part responsible for “selecting the server to service a content request based on a variety of criteria.” *Id.*

569. A POSITA would have found it obvious to combine Glines, Glines-Gupta, Glines-Chiou, or Glines-Gupta-Chiou with Colby to utilize a switch such as Colby’s flow switch 100 at each name server group to select a content server based on a policy. Colby, Abstract, 5:23-6:28, 6:35-67. In the combined system, a switch like Colby’s flow switch 100 would consider load metrics on each of the content servers and select the best-fit content server of the plurality of content servers associated with a particular grouping to serve content based on a load-balancing policy, as disclosed by Colby. *See id.*, Abstract, 5:23-6:6, 6:35-67, Figs. 1c, 2.

570. A POSITA would have understood that load-balancing requests for content across multiple servers was known and understood in the art, including load-balancing techniques applied when resolving requests for Internet content using DNS resolution. *See, e.g.*, Glines, 19:13-22 (“[W]ithin the service group, standard load balancing and replication techniques can also apply, such as multiple content servers (returning multiple local IP addresses to a name lookup), layer 4 switches, etc.”); EX-1010, 1:19-36 (“Many of today’s web sites are hosted on server farms, where a number of servers are grouped together to serve web requests as they arrive.

To avoid overloading individual servers within the farm, load balancing techniques balance the load across servers so that the best total throughput of the farm and smallest response delay for the user is achieved.”), 2:11-58 (same), Fig. 1; EX-1011, 126 (“DNS is also being used to perform load distribution among replicated servers, such as replicated Web servers.”).

571. The combination of Glines, Glines-Gupta, Glines-Chiou, or Glines-Gupta-Chiou with Colby would have been nothing more than combining known elements in the prior art (e.g., a load-balancing switch with Glines’s system to select a “best-fit” content server) according to understood principles with predictable results.

572. A POSITA would have had a reasonable expectation of success in the combination because it was known to apply such load-balancing switches to server farms that respond to user requests for content, including based on DNS requests. *See* Glines, 19:13-22; Chase, 1:19-36; Colby, 5:23-41 (describing flow switch 110 connected to “web servers 100a-c.”); EX-1011, 126.

573. Glines also suggests the combination by disclosing that “standard load balancing” techniques may apply to its name server-content server peer groups, and Glines contemplates utilizing a Layer 4 switch similar to that disclosed in Colby. *See* Glines, 19:13-22; Colby, 6:13-16 (“The content-aware flow switch 110 switches

packets based on a combination of source and destination IP addresses, *transport layer protocol*, and transport layer source and destination port numbers.”).

574. A POSITA would have been motivated to apply a switch such as Colby’s switch 110 to Glines (whether alone or in combination with Gupta) to distribute load across the multiple content servers at each name server location.

575. Based on server load, connection count, and/or capacity, individual servers would not be overloaded with requests for content and instead the “best-fit” server would be selected based on these load metrics, as contemplated by Colby. Colby, Abstract, 6:7-28, 20:1-33.

576. This would have maintained server stability across Glines’s content servers by preventing overloading, ensuring high availability of the content servers, and optimizing resource utilization. The content server would be selected, for example, using a round-robin or least load algorithm using a switch. *See, e.g.*, Glines, 8:19-23 (“Each name server returns the IP address of one or many (such as through round robin or other mechanisms) closely bound and topologically nearby content servers.”); Colby, 6:50-55 (disclosing use of a “load balancing algorithm to use when accessing the content”), 16:48-53 (disclosing an embodiment using a weighted round-robin scheduling algorithm). The name server in Glines would utilize the switch to select the particular content server by identifying the IP address of the content server for a user to retrieve the content.

2. Claim 10: “The method as recited in claim 1, wherein selection of the content server is based on a load balancing policy.”

577. Glines, alone or in combination with Gupta, discloses claim 1, from which claim 10 depends. *See* Section IX.A.2.

578. Glines, with or without Gupta, in further combination with Colby, would have rendered this limitation obvious.

579. Colby discloses a switch 100 for selecting a server from a plurality of servers to provide content to a user using, e.g., a load-balancing algorithm. Colby, Abstract, 6:7-28, 20:1-33, Fig. 1c. Colby also discloses rules that identify a load balance algorithm to use when accessing particular content. *Id.*, 6:35-67, Fig. 2.

580. A POSITA would have been motivated to combine Colby’s switch and load-balancing algorithm (a policy) to select a content server as claimed for the reasons explained in Section IX.D.1 (Motivation to Combine).

581. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

3. Claim 11: “The method of claim 1 further comprising: (c3) by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving CDN DNS.”

582. Glines, alone or in combination with Gupta, discloses claim 1, from which claim 11 depends. *See* Section IX.A.2.

583. Glines, with or without Gupta, in further combination with Colby, would have rendered this limitation obvious.

584. As discussed in claim 10, Colby's switch 110 applies an algorithm, such as a load-balancing algorithm, to select a server from multiple servers. *See* Section IX.D.2; Colby, Abstract, 6:7-28, 20:1-33, Fig. 1c. Colby also discloses content rules that provide a load-balancing algorithm to use when accessing particular content. *Id.*, 6:35-67, Fig. 2.

585. A POSITA would have implemented these teachings in Glines-Colby or Glines-Gupta-Colby to apply a load-balancing algorithm (a policy) to select a content server using a switch as claimed for the reasons explained in Section IX.D.1 (Motivation to Combine).

586. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

4. Claim 12: "The method of claim 11, wherein the step of (c3) is performed in accordance with a predetermined selection policy."

587. Glines, alone or in combination with Gupta and/or Colby, discloses claim 11, from which claim 12 depends. *See* Sections IX.A.11, IX.D.3.

588. Glines, with or without Gupta, in further combination with Colby, would have rendered this limitation obvious.

589. As discussed in claims 10 and 11, Colby discloses a switch selecting a server based on, for example, a load-balancing algorithm. *See* Sections IX.D.2, IX.D.3. Colby's load-balancing algorithm is a "predetermined selection policy," as claimed. Colby further discloses that its switch may consider factors such as source and destination IP addresses, transport layer protocol, and transport layer source and destination port numbers when selecting a content server. Colby, 6:7 28. Consideration of these factors would likewise be a "predetermined selection policy," as claimed.

590. A POSITA would have implemented these teachings in Glines-Colby or Glines-Gupta-Colby to select a content server using a switch based on a predetermined selection policy as claimed for the reasons explained in Section IX.D.1 (Motivation to Combine).

591. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

5. Claim 13

592. Glines, alone or in combination with Gupta, and Colby discloses claim 13. *See* Section IX.A.13.

593. Limitations [13a]-[13d], [13f]-[13h], and [13j] are disclosed by Glines, alone or in combination with Gupta for the reasons discussed above. *See* Section IX.A.13.

594. Colby discloses limitations [13e], and [13i] as discussed below, and a POSITA would have been motivated to combine Glines, with or without Gupta, and Colby for the reasons discussed. Section IX.D.1 (Motivation to Combine).

595. A POSITA would have understood that this combination would not have altered the features of Glines and Gupta discussed in connection with Limitations [13a]-[13d], [13f]-[13h], and [13j].

- a. **[13e] “(c12) by a switch associated with said resolving DNS server of step (c11), selecting the first content server from said respective plurality of content servers associated with the DNS server of step (c11); and”**

596. Glines, with or without Gupta, in further combination with Colby, would have rendered the “switch” of Limitation [13e] obvious.

597. As discussed in claims 10 and 11, Colby discloses a switch for selecting a server. *See* Sections IX.D.2, IX.D.3. Colby applies a load-balancing algorithm, and further discloses that its switch may consider factors such as source and destination IP addresses, transport layer protocol, and transport layer source and destination port numbers when selecting a content server. Colby, 6:7 28.

598. A POSITA would have implemented these teachings in Glines-Colby or Glines-Gupta-Colby to select a content server using a switch for the reasons explained in Section IX.D.1 (Motivation to Combine).

599. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines or Gupta discussed elsewhere in this declaration.

- b. **[13i] “(c22) by a switch associated with said resolving DNS server of step (c21), selecting the second content server from said respective plurality of content servers associated with the DNS server of step (c21); and”**

600. Glines, with or without Gupta, in further combination with Colby, would have rendered this limitation obvious.

601. Limitation [13i] differs from Limitation [13e] in that a switch associated with the “second” DNS server selects the “second” content server.

602. Dependent claim 16 recites the switch of (c22), limitation [13e], and the switch of (c12), limitation [13i], are the same switch. EX-1001, claim 16.

603. Element [13i] therefore encompasses a DNS resolution of the second request using the same switch.

604. Glines and Colby, with or without Gupta, would have rendered this limitation obvious for the same reasons provided for [13e]. Sections IX.D.5.a (Grounds 4A/4B, 5A/5B), IX.A.13.f (Grounds 1A, 1B).

605. It would have been obvious to manage more than one request for content, including multiple requests for content. *See* Section IX.A.13i (Limitation [13h]). Similar or repeated processing would occur to fulfill those requests, including using the same switch (or a second switch) to select a particular content server to provide the content to the user. *See* Section IX.A.13i.

6. Claim 14: “The method as recited in claim 13, wherein selection of at least one of the first content server and the second content server is based on a load balancing policy.”

606. Glines, alone or in combination with Gupta, and/or Colby, discloses claim 13, from which claim 14 depends. *See* Sections IX.A.13, IX.D.5.

607. Glines, with or without Gupta, in further combination with Colby, would have rendered this limitation obvious.

608. As explained, in connection with claims 10 and 11, Glines, alone or in combination with Gupta, in further combination with Colby, would have rendered obvious selection of a content server based on a load balancing policy. *See* Sections IX.D.2, IX.D.3.

609. Further, it would have been obvious to manage more than one request for content, including multiple requests for the same or different content, and content server selection would proceed in the same way for each request. *See* Section IX.A.13.h (Limitation [13g]). Repeating the same process twice, for example selecting a content server by a load balancing policy across the same or different service groups would have been an obvious and expected implementation of the system as explained in claims 10 and 11.

610. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines, Gupta, or Colby discussed elsewhere in this declaration.

7. Claim 15: “The method as recited in claim 13, wherein the resolving DNS server of step (c11) is the same as the resolving DNS server performing of step (c21).”

611. Glines, alone or in combination with Gupta, and/or Colby, renders obvious this limitation as explained in Limitation [13h], which recites “by a second DNS server of said plurality of DNS servers, resolving said first hostname to identify a second IP address for retrieval of the first content from a content server associated with the second DNS server.”

612. As discussed in Limitation [13h], it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. *See* Section IX.A.13.i (Limitation [13h]). Glines’s resolution process would have been handled in the same way for each of the requests.

613. This corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those requests, including by the same DNS server. Repeating the same process twice is an obvious and expected implementation of Glines’s system.

614. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines, Gupta, or Colby discussed elsewhere in this declaration.

8. Claim 16: “The method as recited in claim 15, wherein the switch of step (c12) is the same as the switch of step (c22).”

615. Glines, alone or in combination with Gupta, and/or Colby, renders obvious this limitation as explained in Limitation [13i], which recites “by a switch associated with said resolving DNS server of step (c21), selecting the second content server from said respective plurality of content servers associated with the DNS server of step (c21).”

616. As discussed in Limitation [13i], it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. *See* Section IX.A.13.j (Limitation [13i]). Glines’s resolution process would have been handled in the same way for each of the requests.

617. This corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those requests, including by the same switch. Repeating the same process twice is an obvious and expected implementation of Glines’s system.

618. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines, Gupta, or Colby discussed elsewhere in this declaration.

9. Claim 17: “The method as recited in claim 15, wherein the first content server is the same as the second content server.”

619. Glines, alone or in combination with Gupta, and/or Colby, renders obvious this limitation as explained in Limitation [13h], which recites “by a second DNS server of said plurality of DNS servers, resolving said first hostname to identify a second IP address for retrieval of the first content from a content server associated with the second DNS server.”

620. As discussed in Limitation [13h], it would have been obvious for Glines to manage more than one request for content, including multiple requests for the same or different content. *See* Section IX.A.13.i (Limitation [13h]). Glines’s resolution process would have been handled in the same way for each of the requests.

621. This corresponds with the objectives of Glines, including providing content distribution “as a service where client *requests* are automatically routed to the closest available content server” to improve system performance. Glines 6:3-15; *see also id.*, Abstract, 9:9-17 (plural “requests” for content), 10:5-12 (solving problems handling “the number of requests”). Similar or repeated processing would occur to fulfill those requests, including by the same content server. Repeating the same process twice is an obvious and expected implementation of Glines’s system.

622. Combining Glines or Glines-Gupta with Colby in this way would not alter the features of Glines, Gupta, or Colby discussed elsewhere in this declaration.

10. Claim 18

623. Glines, alone or in combination with Gupta, and Colby, discloses claim 18. *See* Section IX.A.18.

624. Limitations [18p]-[18c], [18e], and [18g] are disclosed by Glines, alone or in combination with Gupta for the reasons discussed above. *See* Section IX.A.18.

625. Colby discloses limitations [18d] and [18f] as discussed below, and a POSITA would have been motivated to combine Glines, with or without Gupta, and Colby for the reasons discussed. *See* Section IX.D.1 (Motivation to Combine).

626. A POSITA would have understood that this combination would not have altered the features of Glines and Gupta discussed in connection with Limitations [18p]-[18c], [18e], and [18g].

- a. **[18d] and [18f] “(C) at least one switch associated with each set of content servers, wherein,” and “(E) the at least one switch associated with the set of content servers of the resolving DNS server is operable to perform load balancing to select the content server from the set of content servers associated with the resolving DNS server”**

627. Glines, with or without Gupta, in further combination with Colby, would have rendered the “switch” of Limitation [18d] and the “switch ... operable to perform load balancing” of Limitation [18f] obvious.

628. As discussed in claims 10 and 11, Colby discloses a switch associated with and for selecting a server. *See* Sections IX.D.2, IX.D.3. Colby applies a

load-balancing algorithm, and further discloses that its switch may consider factors such as source and destination IP addresses, transport layer protocol, and transport layer source and destination port numbers when selecting a content server. Colby, 6:7-28. A POSITA would have implemented these teachings in Glines-Colby or Glines-Gupta-Colby to select a content server using a switch for the reasons explained in Section IX.D.1. (Motivation to Combine).

11. Claim 19

629. Glines, alone or in combination with Gupta and/or Chiou, and Colby, discloses claim 19. *See* Section IX.A.19.

630. Limitations [19p]-[19e], [19g], and [19h] are disclosed by Glines, alone or in combination with Gupta for the reasons discussed above. *See* Section IX.A.19.

631. Chiou alternatively discloses Limitation [19h] as discussed above, and a POSITA would have been motivated to combine Glines, alone or in combination with Gupta, with Chiou for the reasons discussed. *See* Section IX.D.1 (Motivation to Combine)

632. Colby discloses Limitation [19f] as discussed below, and a POSITA would have been motivated to combine Glines, with or without Gupta, and Colby for the reasons discussed. *See* Section IX.D.1 (Motivation to Combine).

633. A POSITA would have understood that the combination of Colby would not have altered the features of Glines, Gupta, and Chiou discussed in connection with Limitations [19p]-[19e], [19g], and [19h].

- a. **[19f] “(c2) by at least one switch, selecting the content server from said respective plurality of content servers associated with said resolving DNS server,”**

634. Glines, with or without Gupta and/or Chiou, in further combination with Colby, would have rendered the “switch” of Limitation [19f] obvious..

635. As discussed in claims 10 and 11, Colby discloses a switch associated with and for selecting a server. *See* Sections IX.D.2, IX.D.3. Colby applies a load-balancing algorithm, and further discloses that its switch may consider factors such as source and destination IP addresses, transport layer protocol, and transport layer source and destination port numbers when selecting a content server. Colby, 6:7 28.

636. A POSITA would have implemented these teachings with Glines, with or without Gupta and/or Chiou, to select a content server using a switch for the reasons explained in Section IX.D.1. (Motivation to Combine).

- 12. Claim 20: “The method as recited in claim 19, wherein selection of the content server is based on a load balancing policy.”**

637. Glines, alone or in combination with Gupta, Chiou, and/or Colby, discloses claim 19, from which claim 20. *See* Sections IX.A.19, IX.C.5, IX.D.11.

638. Glines, with or without Gupta and/or Chiou, in further combination with Colby, would have rendered this limitation obvious.

639. As explained above, Section IX.A.20, Glines alone or in combination with Gupta, would have rendered claim 20 obvious. The addition of Colby to the combination would not have disturbed Glines's (or Glines-Gupta's, Glines-Colby's, or Glines-Gupta-Colby's) teaching that content server selection may be based on a load balancing policy. *Id.* Accordingly, Glines, with or without Gupta, Chiou, and Colby would have rendered this limitation obvious.

640. Combining Glines, Glines-Gupta, Glines-Chiou, or Glines-Gupta-Chiou with Colby in this way would not alter the features of Glines, Gupta, Colby, or Chiou discussed elsewhere in this declaration.

13. Claim 21

641. Glines, alone or in combination with Gupta, and Colby discloses claim 21. *See* Section IX.A.21.

642. Limitations [21p]-[21d] are disclosed by Glines, alone or in combination with Gupta for the reasons discussed above. *See* Section IX.A.21.

643. Colby discloses Limitation [21e] as discussed below, and a POSTA would have been motivated to combine Glines, with or without Gupta, and Colby for the reasons discussed. *See* Section IX.D.1 (Motivation to Combine).

644. A POSITA would have understood that this combination would not have altered the features of Glines and Gupta discussed in connection with Limitations [21p]-[21d].

- a. **[21e] “(c2) by at least one switch associated with the set of content servers of the resolving DNS server, selecting the content server from the set of content servers associated with the resolving DNS server”**

645. Glines, with or without Gupta, in further combination with Colby, would have rendered the “switch” of Limitation [21e] obvious.

646. As discussed in claims 10 and 11, Colby discloses a switch for selecting a server. *See* Sections IX.D.2, IX.D.3. Colby applies a load-balancing algorithm, and further discloses that its switch may consider factors such as source and destination IP addresses, transport layer protocol, and transport layer source and destination port numbers when selecting a content server. Colby, 6:7-28. A POSITA would have implemented these teachings in Glines-Colby or Glines-Gupta-Colby to select a content server using a switch for the reasons explained in Section IX.D.1. (Motivation to Combine).

14. Claim 22

647. Glines, alone or in combination with Gupta, and Colby, discloses claim 22. *See* Section IX.A.22.

648. Limitations [22p]-[22c], [22e], and [22f] are disclosed by Glines, alone or in combination with Gupta, for the reasons discussed above. *See* Section IX.A.22.

649. Colby discloses Limitations [22d] and [22g] as discussed below, and a POSTA would have been motivated to combine Glines, with or without Gupta, and Colby for the reasons discussed. *See* Section IX.D.1 (Motivation to Combine).

650. A POSITA would have understood that this combination would not have altered the features of Glines and Gupta discussed in connection with Limitations [22p]-[22c], [22e], and [22f].

- a. **[22d] and [22g]: “at least one switch, wherein” and “the at least one switch is operable to select the content server from the plurality of content servers, and”**

651. Glines, with or without Gupta, in further combination with Colby, would have rendered the “switch” of Limitation [22d] and the switch “operable to select the content server” of Limitation [22g] obvious.

652. As discussed in claims 10 and 11, Colby discloses a switch associated with and for selecting a server. *See* Sections IX.D.2, IX.D.3. Colby applies a load-balancing algorithm, and further discloses that its switch may consider factors such as source and destination IP addresses, transport layer protocol, and transport layer source and destination port numbers when selecting a content server. Colby, 6:7-28. A POSITA would have implemented these teachings in Glines-Colby or Glines-Gupta-Colby to select a content server using a switch for the reasons explained in Section IX.D.1. (Motivation to Combine).

X. SECONDARY CONSIDERATIONS

653. I have considered the issue of secondary considerations and am currently unaware of any evidence of secondary considerations that would support a finding of non-obviousness or otherwise impact my opinions herein.

XI. CONCLUSION

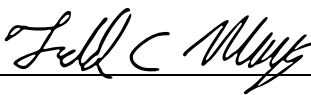
654. For the reasons set forth above, it is my opinion that claims 1-23 of the '112 patent are unpatentable in view of the prior art.

655. In signing this declaration, I understand that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I acknowledge that I may be subject to cross-examination in this case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

656. I declare that all statements made herein of my knowledge are true, that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Declaration of Todd C. Mowry, Ph.D.
U.S. Patent No. 9,021,112

Dated: April 14, 2025

By: 

Todd C. Mowry, Ph.D.