

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TESSELL, INC.,
Petitioner,

v.

NUTANIX, INC.,
Patent Owner.

Case No. IPR2025-00733
Patent No. 11,640,340

DECLARATION OF SAMRAT BHATTACHARJEE, PH.D.

Tessell Exhibit 1003 Tessell v. Nutanix
--

TABLE OF CONTENTS

I.	PERSONAL AND PROFESSIONAL BACKGROUND	4
II.	MATERIALS REVIEWED AND CONSIDERED	8
III.	MY UNDERSTANDING OF PATENT LAW	11
	A. Anticipation	13
	B. Obviousness.....	14
IV.	THE '340 PATENT	16
	A. Technical Background and Specification.....	16
	B. Challenged Claims.....	35
	C. Prosecution History	35
	D. Person of Ordinary Skill in the Art (“POSA”).....	36
V.	<u>GROUND 1: CLAIMS 1-30 ARE UNPATENTABLE OVER</u> <u>CHIKKANAYAKANAHALLY, WITH OR WTIHOUT MERRIMAN</u>	38
	A. Chikkanayakanahally (EX1005)	39
	B. Mapping to Challenged Claims	46
	1. Claim 1	46
	a. [1PRE]: “A system comprising:”	46
	b. [1A]: “a processor executing computer-readable instructions stored on a memory to”	47
	c. [1B1] “select one of a plurality of nodes as a primary node capable of storing a source database, the remaining ones of the plurality of nodes being a secondary node, each of the secondary node capable of storing a copy of the source database”	59
	d. [1B2] “wherein updates made to the source database on the primary node are able to be copied to the copy of the source database on the secondary node”.....	67
	e. [1B3] “and wherein in response to the primary node becoming unavailable, one of the secondary node is designated to become the primary node based on a first policy”	69

f.	[1C] “designate a first node from the primary node or the secondary node as an initial active node”	78
g.	[1D] “capture one or more snapshots of the source database or the copy of the source database from the first node designated as the initial active node”	84
h.	[1E] “determine that the first node has become unavailable”	88
i.	[1F] “select, in response to the first node becoming unavailable and based on a user defined switch over policy, a second node from the primary node or the secondary node as a next active node, wherein the switch over policy defines a criteria for selecting the next active node from the plurality of nodes to capture the one or more snapshots from; and”	92
j.	[1G] “capture the one or more snapshots of the source database or the copy of the source database from the second node in response to selecting the second node as the next active node”	97
2.	Claim 2: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node.”	102
3.	Claim 3: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node”	107
4.	Claim 4: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node, or upon determining that none of the plurality of nodes are the secondary node, select the next active node from the plurality of nodes that is the primary node”	108
5.	Claim 5: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node, or upon determining that none of the plurality of nodes is the primary node, select the next active node from the plurality of nodes that is the secondary node”	111

6. Claim 6: “The system of claim 1, wherein the switch over policy defines a priority assigned to at least a subset of the plurality of nodes and wherein the processor further executes the computer-readable instructions to select the next active node based upon the priority”	114
7. Claim 19: “The system of claim 1, wherein the plurality of nodes are distributed between a first cluster and a second cluster”	117
8. Claim 20: “The system of claim 1 wherein the switch over policy is defined during provisioning of the source database”	125
9. Claim 21: “The system of claim 1, wherein the source database on the primary node and the copy of the source database on the secondary node comprise a database cluster the source database”	128
10. Claim 22: “The system of claim 21, wherein the switch over policy is part of a user defined active policy defining...”	129
a. “a number of nodes in the plurality of nodes based on a number of copies of the source database that are desired, ...”	130
b. “a location of the nodes, ...”	132
c. “and one or more conditions for selecting a new primary node when a current primary node becomes unavailable”	134
11. Claim 7	135
a. [7PRE]: “A non-transitory computer-readable media comprising computer readable instructions stored thereon that when executed by a processor causes the processor to”	135
b. [7A1]-[7F]	135
12. Claims 8-12, 23-26	136
13. Claim 13	137
a. [13PRE]: “A method comprising”	137
b. [13A1]-[13F]	137
14. Claims 14-18, 27-30	138
VI. CLAIM LISTING	139

I, Samrat Bhattacharjee, Ph.D., declare:

1. I have been retained by Wolf, Greenfield & Sacks, P.C., counsel for Petitioner Tessell, Inc., to assess claims 1-30 (the “challenged claims”) of U.S. Patent No. 11,640,340 (“the ’340 patent,” EX1001). I am being compensated for my time at my standard rate of \$700.00 per hour, plus actual expenses. My compensation is not dependent in any way upon the outcome of the *inter partes* review of the ’340 patent.

I. PERSONAL AND PROFESSIONAL BACKGROUND

2. My qualifications for forming the opinions set forth in this declaration are summarized below and explained in more detail in my current curriculum vitae, provided as EX1004. EX1004 also includes a list of my publications and patents.

3. I am a tenured professor in the Computer Science Department at the University of Maryland. I am also an Affiliate Professor with the Department of Electrical and Computer Engineering at the University of Maryland. I have been an appointed faculty member at the University of Maryland since 1999.

4. I received a Ph.D. in computer science from the Georgia Institute of Technology in 1999. Before that I received a B.S. degree in Mathematics and Computer Science with Highest Distinction (Summa Cum Laude) and was recognized as the Outstanding Department Major at Georgia College and State University, Milledgeville, Georgia in 1994.

5. I have been involved in the research and development of computing systems for over two decades. I perform research and I teach in the areas of computer networks, distributed systems, data storage and backup, and networking, among others. A major theme of my research has been the development of technology to improve the performance and security of data transfers and data storage on networks. My research has examined problems ranging from precursors to software-defined networking, operating systems, secure data transfer and storage, large scale Internet video distribution, the design of novel network architectures for better and secure content distribution and storage, efficiency of content distribution over wireless networks, and so on. In the course of my research, I have incorporated various types of databases and other mass storage systems into systems that I have built. I have also written numerous software programs, including software programs with user interfaces.

6. For example, starting in the late 1990s, the focus of my research was the development of network and operating system technology to design a new network architecture that allowed for more efficient data transfer. Part of my Ph.D. research was to develop new and more efficient architectures for video delivery on the Internet, and I have published papers on this architecture during my graduate studies. I continued to work on video delivery and security and storage of digital content as a faculty member, and have published various papers in these

areas. During 2007, I was a visiting researcher at AT&T Labs, and one of the projects I focused on was a video content delivery platform, which was implemented using networked massed storage. This work resulted in both publications and a granted U.S. patent (U.S. 8,752,100 B2).

7. Over the course of my career, I have conducted a number of research projects that have related to the technologies underlying databases or that have used databases as components of larger systems. For example, I have conducted research on various techniques used to improve reliability and data recovery, including replication and snapshots, for distributed datasets, including storage infrastructure for databases. In 2006, I presented a paper on possible techniques for implementing a distributed version of Google's search capability at the 2nd International IEEE Workshop on Networking Meets Databases. In 2007, I was granted a patent related to secure data transfer that used a database to store cryptographic and other information. In addition, as part of my research, I have used cloud-based databases, including those hosted on Amazon AWS. For example, in 2018, I published a paper related to a project that began in 2016 for allowing users to anonymously upload sensitive information. This system used MySQL databases implemented on the Amazon AWS service. Furthermore, I have worked on the intersection of operating systems, security, and highly performant large-scale data access systems for over 20 years, all of which is related

to and relevant to database research, and I have reviewed database literature in the course of my academic research.

8. Along with working on research problems related to the topics mentioned above, I routinely taught this material in both graduate and undergraduate courses. A number of my papers in these domains are with students at the University of Maryland, many of whom started in these areas after taking my classes.

9. For over 25 years, I have developed professional and academic experience in the field of computer systems. For example, I have authored or co-authored over 100 articles in peer-reviewed journals, conference proceedings, and workshops relating to computer networking, data storage, data security. Over the years I have received several fellowships, prizes, and awards for teaching excellence and technical papers, including an Alfred P. Sloan Jr. Fellowship in 2004 and the National Science Foundation CAREER award in 2001. I have also received an award from the SIGCOMM foundation for a “Test of Time” paper (that is given to an influential paper published ten years ago). My research work has been supported by multiple grants from the US National Science Foundation, and the Department of Defense. I have also started a Joint Ph.D. program with the University of Maryland and the Max Planck Society in Germany, and co-founded

the annual Cornell, Maryland, Max Planck Research School that provides research exposure to about 80 students from across the world during a week-long school.

10. I have served on numerous paper and proposal review panels and participated on program committees for ACM/IEEE groups, technical journals, and conferences in Networking, Security and Systems in the aforementioned areas of computer science. I have created research software for all of my projects, much of which is available online for others to build on.

11. I am a named inventor on four U.S. Patents. These patents are generally related to computer networking and the delivery of services over networks and video delivery systems, and some of the systems described in these patents used databases.

12. I have served as an expert witness and technical consultant in litigation and Inter Partes Review matters concerning videoconferencing, data conferencing and screen sharing, voice over IP (VoIP) telephony, multimedia networking, distributed systems, operating systems, computer networks, and datacenter networking, among others. I have testified in several trials and depositions as an expert witness. My curriculum vitae is provided as Exhibit 1004.

II. MATERIALS REVIEWED AND CONSIDERED

13. My findings, as explained below, are based on my years of education, research, experience, and background in the field of computer science, as well as

my investigation and study of relevant materials for this declaration. When developing the opinions set forth in this declaration, I assumed the perspective of a person having ordinary skill in the art, as set forth in Section IV below. In forming my opinions, I have studied and considered the materials identified in the list below, as well as any additional materials cited herein.

Exhibit	Description
1001	U.S. Patent No. 11,640,340
1002	Prosecution History of U.S. Patent No. 11,640,340
1005	U.S. Patent No. 9,934,107 (“Chikkanayakanahally”)
1006	U.S. Patent No. 8,150,808 (“Zha”)
1007	U.S. Patent No. 10,235,250 (“Ambastha”)
1008	U.S. Patent No. 6,542,893 (“Quernemoen”)
1009	The American Heritage Dictionary of the English Language (5th ed. 2011) (“Heritage”)
1010	U.S. Patent No. 11,609,828 (“Khandkar”)
1011	U.S. Patent No. 11,169,887 (“Bajaj”)
1012	Using Oracle Database Cloud Service – Creating a Database Deployment (2017) (“Oracle”) (Wayback Machine Capture, January 18, 2017), available at https://web.archive.org/web/20170118071307/https://docs.oracle.com/en/cloud/paas/database-dbaascloud/csdbi/create-db-deployment.html
1013	Amazon Relational Database Service User Guide (2016) (“RDS”) (Wayback Machine Capture, April 12, 2016), available at https://web.archive.org/web/20160412105355/http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-ug.pdf
1014	U.S. Patent App. Pub. No. 2009/0319582 (“Simek”)
1015	U.S. Patent No. 7,331,000 (“Ohno”)
1016	U.S. Patent No. 8,918,673 (“Rangaiah”)
1017	U.S. Patent App. Pub. No. 2016/0077925 (“Tekade”)
1018	U.S. Patent No. 12,093,715 (“Zhao”)
1019	U.S. Provisional Patent App. No. 63/073,513

Exhibit	Description
1020	U.S. Patent App. Pub. No. 2011/0099420 (“McAliser”)
1021	U.S. Patent No. 9,405,634 (“Ambastha-634”)
1022	“Taking a Backup Using Snapshots” (“LVM-Howto”) (Wayback Machine Capture, January 20, 2007), available at https://web.archive.org/web/20070120200004/https://tldp.org/HOWTO/LVM-HOWTO/snapshots_backup.html
1023	U.S. Patent App. Pub. No. 2020/0314173 (“Pahwa”)
1028	U.S. Patent App. Pub. No. 2016/0057077 (“Gomatam”)
1029	U.S. Patent No. 11,036,696 (“Higginson”)
1032	“Creating Instances,” Google Cloud (“Google”) (Wayback Machine Capture, September 18, 2018), available at https://www.archive.org/web/20180918103721/https://cloud.google.com/sql/docs/mysql/create-instance
1033	U.S. Patent App. Pub. No. 2003/0046353 (“Chung”)
1058	Affidavit of Nathaniel E. Frank-White
1060	U.S. Patent No. 11,120,011 (“Priebe”)
1106	U.S. Patent No. 8,572,031 (“Merriman”)
1107	<i>7 more ways to Query Always on Availability Groups</i> , as archived by the Internet Archive on March 30, 2019, available at https://web.archive.org/web/20190330182441/https://sqlundercover.com/2019/02/19/7-more-ways-to-query-always-on-availability-groups/
1108	<i>SQL Server AlwaysOn Availability Group Backup Preference Setting</i> , as archived by the Internet Archive on September 25, 2020, available at https://www.mssqltips.com/sqlservertip/4976/sql-server-alwayson-availability-group-backup-preference-setting/
1109	<i>Use the Availability Group Wizard (SQL Server Management Studio)</i> , as archived by the Internet Archive on January 20, 2020, available at https://web.archive.org/web/20200120100053/https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/use-the-availability-group-wizard-sql-server-management-studio?view=sql-server-ver15
1110	U.S. Patent No. 8,713,060 (“Sivasubramanian”)
1112	Excerpts of <i>The Merriam-Webster Dictionary</i> (2019)
1113	U.S. Patent No. 9,311,330 (“Chockalingam”)
1114	“Nutanix Era: Databases Made Simple,” as archived by the Internet Archive on September 26, 2020, available at http://web.archive.org/web/20200926222827/https://www.nutanix.com/blog/nutanix-era-databases-made-simple

Exhibit	Description
1116	U.S. Patent No. 7,046,134 (“Hansen”)
1117	U.S. Patent No. 8,656,303 (“Hughes”)
1118	U.S. Patent No. 6,766,477 (“Grucci”)
1119	U.S. Patent No. 9,116,862 (“Rath”)
1120	U.S. Patent No. 8,843,441 (“Rath-441”)
1121	“ <i>Windows Geo-Clustering: SQL Server</i> ,” published October 12, 2014

III. MY UNDERSTANDING OF PATENT LAW

14. In developing my opinions, I discussed various relevant legal principles with Petitioner’s attorneys. Though I do not purport to have prior knowledge of such principles, I understood them when they were explained to me and have relied upon such legal principles, as explained to me, in the course of forming the opinions set forth in this declaration. My understanding in this respect is as follows:

15. I understand that “*inter partes* review” (IPR) is a proceeding before the United States Patent & Trademark Office for evaluating the patentability of an issued patent’s claims based on prior-art patents and printed publications.

16. I understand that, in this proceeding, Petitioner has the burden of proving that the challenged claims of the ’340 patent are unpatentable by a preponderance of the evidence. I understand that “preponderance of the evidence” means that a fact or conclusion is more likely true than not true.

17. I understand that, in IPR proceedings, claim terms in a patent are given their ordinary and customary meaning as understood by a person of ordinary skill in the art (“POSA”) in the context of the entire patent and the prosecution history pertaining to the patent. If the specification provides a special definition for a claim term that differs from the meaning the term would otherwise possess, the specification’s special definition controls. If a claim element is expressed as a “means” for performing a specified function, I understand that it covers the corresponding structure described in the specification and equivalents of the described structure. I have applied these standards in preparing the opinions in this declaration.

18. I understand that determining whether a particular patent or printed publication constitutes prior art to a challenged patent claim can require determining the effective filing date (also known as the priority date) to which the challenged claim is entitled. I understand that for a patent claim to be entitled to the benefit of the filing date of an earlier application to which the patent claims priority, the earlier application must have described the claimed invention in sufficient detail to convey with reasonable clarity to the POSA that the inventor had possession of the claimed invention as of the earlier application’s filing date. I understand that a disclosure that merely renders the claimed invention obvious is

not sufficient written description for the claim to be entitled to the benefit of the filing date of the application containing that disclosure.

19. I understand that for an invention claimed in a patent to be patentable, it must be, among other things, new (novel—*i.e.*, not anticipated) and not obvious from the prior art. My understanding of these two legal standards is set forth below.

A. Anticipation

20. I understand that, for a patent claim to be “anticipated” by the prior art (and therefore not novel), each and every limitation of the claim must be found, expressly or inherently, in a single prior-art reference. I understand that a claim limitation is disclosed for the purpose of anticipation if a POSA would have understood the reference to disclose the limitation based on inferences that a POSA would reasonably be expected to draw from the explicit teachings in the reference when read in light of the POSA’s knowledge and experience.

21. I understand that a claim limitation is inherent in a prior art reference if that limitation is necessarily present when practicing the teachings of the reference, regardless of whether a person of ordinary skill recognized the presence of that limitation in the prior art.

B. Obviousness

22. I understand that a patent claim may be unpatentable if it would have been obvious in view of a single prior-art reference or a combination of prior-art references.

23. I understand that a patent claim is obvious if the differences between the subject matter of the claim and the prior art are such that the subject matter as a whole would have been obvious to a person of ordinary skill in the relevant field at the time the invention was made. Specifically, I understand that the obviousness question involves a consideration of:

- the scope and content of the prior art;
- the differences between the prior art and the claims at issue;
- the knowledge of a person of ordinary skill in the pertinent art; and
- if present, objective factors indicative of non-obviousness, sometimes referred to as “secondary considerations.” To my knowledge, the Patent Owner has not asserted any such secondary considerations with respect to the ’340 patent.

24. I understand that in order for a claimed invention to be considered obvious, a POSA must have had a reason for combining teachings from multiple prior-art references (or for altering a single prior-art reference, in the case of obviousness in view of a single reference) in the fashion proposed.

25. I further understand that in determining whether a prior-art reference would have been combined with other prior art or with other information within the knowledge of a POSA, the following are examples of approaches and rationales that may be considered:

- combining prior-art elements according to known methods to yield predictable results;
- simple substitution of one known element for another to obtain predictable results;
- use of a known technique to improve similar devices in the same way;
- applying a known technique to a known device ready for improvement to yield predictable results;
- applying a technique or approach that would have been “obvious to try,” *i.e.*, choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success.
- known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to one of ordinary skill in the art;
- some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior-art reference or

to combine prior-art reference teachings to arrive at the claimed invention. I understand that this teaching, suggestion or motivation may come from a prior-art reference or from the knowledge or common sense of one of ordinary skill in the art.

26. I understand that for a single reference or a combination of references to render the claimed invention obvious, a POSA must have had a reasonable expectation of success in arriving at the claimed invention by altering or combining the teachings of the applied references.

IV. THE '340 PATENT

A. Technical Background and Specification

27. The '340 patent relates to a “database management system” providing “database management services.” EX1001, 3:10-23 (“The present disclosure is generally directed to a hyperconverged virtual computing system.... The virtual computing system may include a **database management system configured for providing database management services.**”).¹

28. Databases predated the '340 patent's priority date. Quernemoen (EX1008), which was filed in 2000 and issued in 2003, mentions that “[r]elational databases came into common use in computers over twenty years ago.” Quernemoen, 1:36-37. This is consistent with my personal understanding and

¹ All emphasis added unless otherwise indicated.

experience; I took a course on relational databases as an undergraduate in the early 1990s, and my understanding is that relational databases have been around since the 1970s.

29. Databases are collections of organized data. Heritage (EX1009), 462 (“database” is “[a] collection of data arranged for ease and speed of search and retrieval.”). A database, and the data within it, is managed by database software. Various types of database software, which the ’340 patent refers to as “database engine” types, were known in the art (*e.g.*, Oracle, PostgreSQL, etc.). EX1001, 15:13-17.

30. Data in a database can be modified (*e.g.*, new data added or existing data changed) by “writing” to the database, and data can be retrieved by “reading” from the database. EX1001, 3:40-43 (“...a user may perform operations (*e.g.*, read, write, etc.) on the database...”); Merriman (EX1106), 1:37-49 (“Some conventional approaches resolve some of these reliability issues by separating responsibility for **read and write** operations among elements of a distributed database.... In the MySQL database, processing of **write** requests can be limited to the master system which can then propagate **changes** to its slave systems. The slave systems are then used to respond to **read** requests....”).

31. Databases run on computer devices, *e.g.*, servers; the ’340 patent—consistent with the prior art—refers to such devices as “nodes.” EX1001, Abstract

(“A system and method includes a plurality of nodes distributed between a first cluster and a second cluster, each of the plurality of nodes storing a copy of a source database...”), 6:67-7:6 (“The first node 105 includes database virtual machines (‘database VMs’) 120A and 120B (collectively referred to herein as ‘database VMs 120’), a hypervisor 125 configured to create and run the database VMs...”), 7:58-8:5 (“In some embodiments, each of the first node 105, the second node 110, and the third node 115 may be a hardware device, such as a server.... In other embodiments, one or more of the first node 105, the second node 110, or the third node 115 may be another type of hardware device, such as a personal computer, an input/output or peripheral unit such as a printer, or any type of device that is suitable for use as a node within the cluster 100.”); Chikkanayakanahally (EX1005), 9:5-10; Khandkar (EX1010), 3:63-4:17 (“The source host 106 includes a source node 114 (e.g., server machine) (e.g., PRIMARY) that is communicatively coupled to the production system 116 and a storage device 117 that stores the database 118.”).

32. The ’340 patent’s “database management services” include “copy data management services,” which “involve protecting a database” by creating “cop[ies]” of the database and “restor[ing]” a database from a “copy.” EX1001, 3:29-60 (“One example database management service provided by the database management system may include copy data management services. Copy data

management services involve protecting a database. Protecting a database means replicating a state of the database to enable creating a fully functional copy of the database, as well as restoring a database from the replicated state of the database.... Protecting a database may also include restoring a source or original or production database. In some embodiments, the production database may become corrupted (e.g., bad data). Restoring the production database may involve fixing the corruption by using the replicated state of the production database.”).

33. The patent refers to a database copy as a “clone,” and refers to a database from which a clone/copy is created as a “source database.” EX1001, 3:38-43.

34. One known technique for creating copies of a database was to create “snapshots” of the database. Consistent with the prior art, the ’340 patent uses the term “snapshot” to refer to a record of the database’s state at a point in time. EX1001, 3:65-4:1 (“A snapshot stores the state of the source database at the point in time at which the snapshot is captured. The snapshot is thus a point in time image of [a] source database.”), 5:2-6 (“cloned databases of the source database may be created to a point in time...based upon available snapshots (e.g., at the point of snapshot capture”); Simek (EX1014), [0002] (“Database Snapshots are a feature of some database management systems (‘DBMS’) and allow a user to specify a point in time at which to preserve the contents of the database.”); Ohno

(EX1015), 1:54-61 (“In order to restore the contents of the database up to a point in time when the data has consistency, a snapshot must be acquired in advance in a state in which data has consistency.... Snapshot is to maintain data to be snapshot at a certain point in time in an external storage, and it is a function provided by most storage systems.”).

35. It was known to use snapshots for database backup. A snapshot could be stored separately from the database, and in the event of the database’s failure, the snapshot could be used to restore the database to the database’s state at the time the snapshot was taken. EX1001, 3:55-60 (“Protecting a database may also include restoring a source or original or production database. In some embodiments, the production database may become corrupted (e.g., bad data). Restoring the production database may involve fixing the corruption by using the replicated state of the production database.”); Simek (EX1014), [0002] (“Database Snapshots are a feature of some database management systems (‘DBMS’) and allow a user to specify a point in time at which to preserve the contents of the database.”), [0003] (“...it is currently necessary to restore an entire database snapshot even when it is only desirable to restore a small portion of the data contained within a database snapshot”). *See also* my discussion in, *e.g.*, paragraphs 45-47 below.

36. It was also known in the art to concurrently maintain multiple up-to-date copies (which the '340 patent also calls "instances," see EX1001, 5:45-61) of a single database concurrently for redundancy, a technique sometimes called "replication." EX1001, 39:28-55 (discussing different types of "replication" to keep nodes "up to date" with each other); Merriman (EX1106), 2:3-10. Consistent with the prior art, the '340 patent refers to such a database with a set of replicated copies as a "highly available" database, because if one copy of the database becomes unavailable, another copy can quickly be made available in its place. EX1001, 5:43-49; Chikkanayakanahally, 5:41-44 (referring to "high-availability database system").

37. As was known in the art, each copy of a "highly available" database may run on a separate "node." EX1001, Abstract, 6:67-7:6, 7:58-8:5; Chikkanayakanahally, 9:5-10; Khandkar (EX1010), 3:63-4:17.

38. In a "highly available"/replicated database, one copy/instance of the database may be designated as a "primary" copy. EX1001, 24:9-16; Chikkanayakanahally, Abstract. When data is written to the database, the data may be written to a primary copy initially, and then may be replicated on the other copies (either substantially synchronously, or in batches), to keep all the replicated copies consistent. Merriman (EX1106), 1:40-52, 2:56-67; EX1001, 25:6-21; Chikkanayakanahally, 8:57-67, 9:57-67. When data is to be read, it is available

from multiple copies. Merriman (EX1106), 1:42-55; EX1001, 3:40-43; Chikkanayakanahally, 5:60-6:2 (disclosing that “in a high-availability environment a first node in the cluster” may host a “primary database[],” and a “second node in the cluster may host... corresponding secondary databases [that] are synchronized with the primary database and may... be made available for read-only access”), 6:9-13 (similar), 9:57-67 (disclosing details of how the data copies on secondary databases are synchronized with the data on the primary database). Consistent with the prior art, the ’340 patent calls the non-primary copies of the database “secondary” copies, and refers to the nodes on which the secondary copies ran as “secondary” nodes. EX1001, Abstract, 39:59-63; Merriman (EX1106), 2:11-23; Chikkanayakanahally, 5:65-6:19.

39. In a “highly available”/replicated database, the node hosting the primary database could become unavailable (*e.g.*, due to hardware failure), causing the primary database to become unavailable. EX1001, 24:49-51; Chikkanayakanahally, 6:11-19, 13:66-14:3; Merriman, 2:32-35. In this case, one of the secondary nodes is designated as the new primary node, and the secondary copy of the database running on that node becomes the new primary copy (thus taking on primary responsibility for real-time writes to the database). EX1001, 24:49-51; Chikkanayakanahally, 5:51-59, 6:11-19, 13:66-14:3; Merriman, 2:11-26.

40. As I discuss further in Section V.B.1.e below (for [1B3]), the '340 patent's claims refer to a "policy" for designating a secondary node as the new primary node in such a scenario. I have reviewed the '340 patent's specification, and the specification does not use the term "policy" when describing how a new primary node is chosen. However, the specification describes various "criteria" that a "database controller" can use to "designate a new primary node," including selecting a secondary node that has been "synchronous[ly]" updated with replicated data that was written to the primary node (as opposed to selecting a batch-updated secondary node that might not be completely up-to-date at a given moment) or selecting which secondary will become the new primary based on an "order" that "the user may define." EX1001, 25:22-36.

41. I have reviewed the '340 patent, and the patent nowhere purports to have invented snapshots or replication (and I note that both snapshots and replication were known in the art, as I discussed above). Rather, as I discuss below, the '340 patent relates to using replication techniques and snapshot-based backup together. Using snapshots and replication together was well-known in the art. My testimony is corroborated by, *e.g.*: Chikkanayakanahally, 1:14-32; Merriman, 14:12-22; Bajaj (EX1011), 2:36-3:7 ("A **database** and its associated data **may be backed up to a secondary storage system**.... A stand-alone database that has a **standby role** may be **used to perform a backup snapshot** instead of a

stand-alone database that has a **primary** role to free up resources of the database node hosting the stand-alone database that has the primary role and preserve the resources available to the stand-alone database that has the primary role for one or more other purposes, such as to fulfill one or more database requests.... In some embodiments, a database cluster is comprised of a primary database instance and one or more standby database instances. **A database node hosting a standby database instance may be used to perform a backup snapshot.”).**

42. The '340 patent discloses functionality that it calls a “highly available time machine” for “automatically capturing backups of [a] source database.” EX1001, 2:35-39 (“FIGS. 4A and 4B are example block diagrams showing a highly available time machine of the database management system of FIG. 2 for automatically capturing backups of the source database...”), Fig. 4A-4B (shown below). POSAs would have understood that the patent describes its system as a “time machine” because, as I will discuss below, and as was known in the art, the database backups that the system captures allow the database to be restored to the state it was in at a previous point in time. EX1001, 3:55-60 (“Protecting a database may also include restoring a source or original or production database.... Restoring the production database may involve fixing the corruption by using the replicated state of the production database.”); Simek (EX1014), [0002] (“Database Snapshots are a feature of some database management systems (‘DBMS’) and allow a user to

specify a point in time at which to preserve the contents of the database.”), [0003]
(“...it is currently necessary to restore an entire database snapshot even when it is
only desirable to restore a small portion of the data contained within a database
snapshot”).

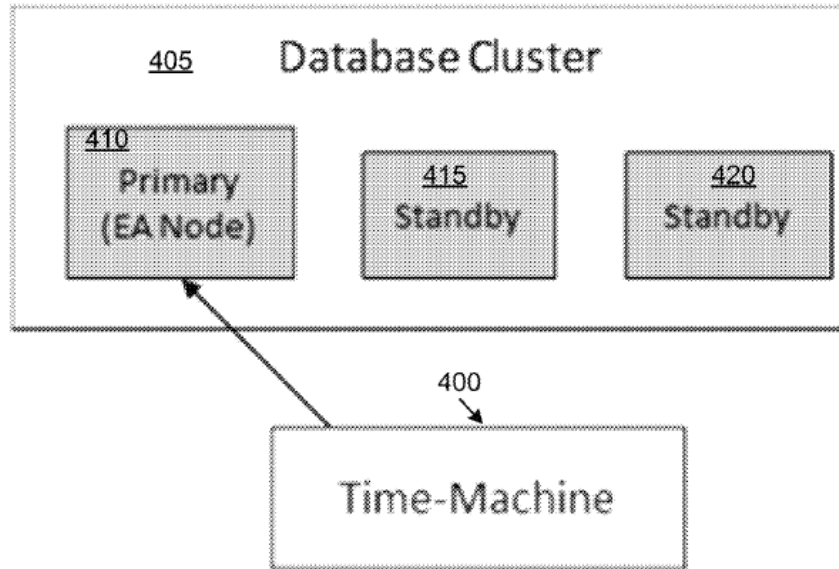


FIG. 4A

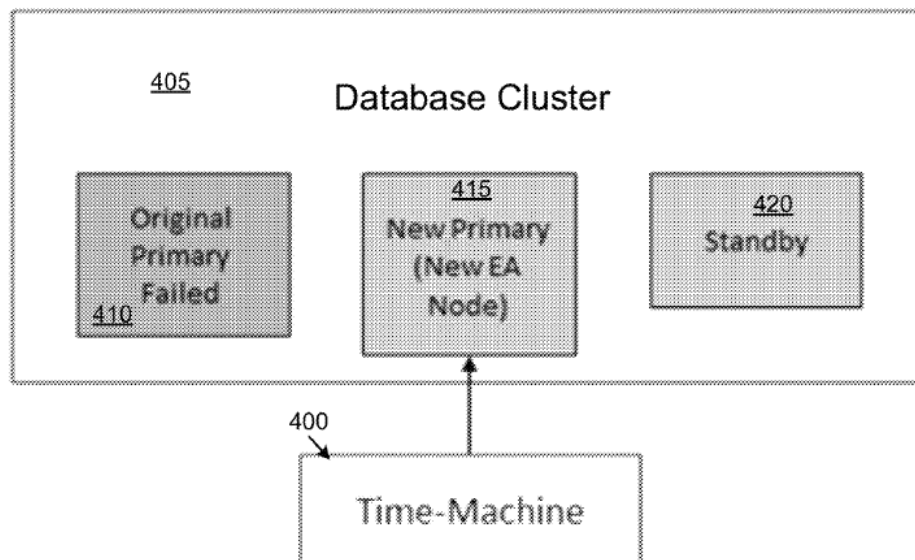


FIG. 4B

43. The “highly available time machine” may provide backup services for a “database cluster,” which consists of multiple copies of a single database where each database copy runs on a separate “node.” EX1001, 23:61-24:8. As shown in Figure 4A (below), one of the nodes in the database cluster is initially designated as the “primary” node; the remaining nodes may be designated as “secondary” (also called “standby”) nodes. EX1001, Abstract, 24:9-24.

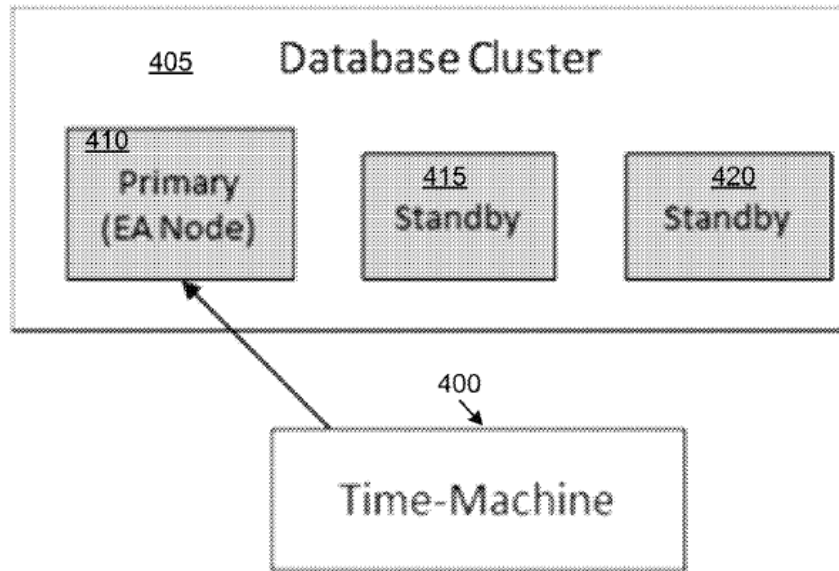


FIG. 4A

44. Consistent with the prior art I discussed above, if/when the primary node fails, one of the standby/secondary nodes may assume the role of the primary node, as shown in Figure 4B (below). EX1001, 24:49-51.

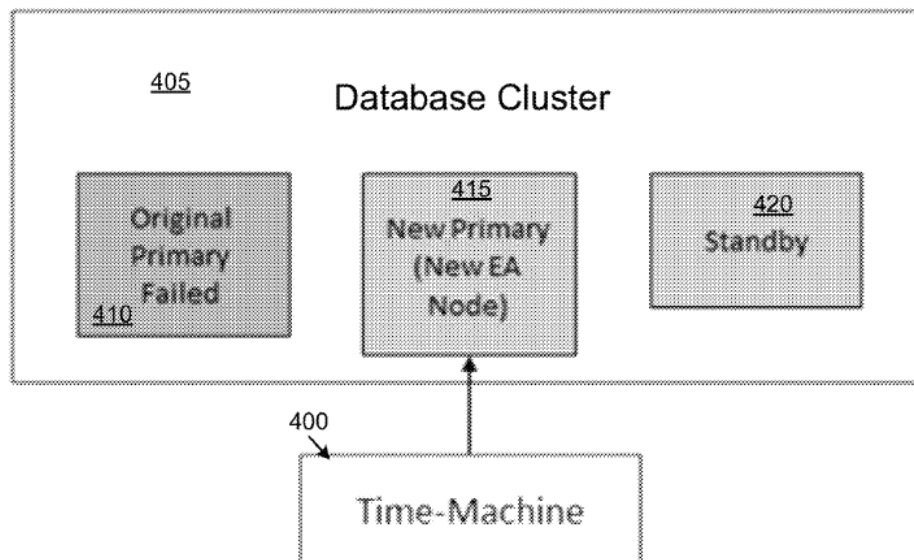


FIG. 4B

45. In addition, as was known in the art, backups of the database can be created by taking snapshots from either a primary node or a standby/secondary node, as I discuss in paragraphs 46-47 below.

46. POSAs understood that taking a backup from a database typically required pausing that database while the snapshot was taken. My testimony regarding a POSA's background knowledge is corroborated by, *e.g.*, Ohno (EX1015), 1:54-2:21. Ohno explains that "to restore the contents of the database up to a point in time when the data has consistency, a snapshot must be acquired in advance in a state in which data has consistency. For this purpose, it is necessary to staticize the database and acquire a snapshot thereof in a state in which data has consistency." Ohno, 1:54-59. Ohno explains that some database management

systems (“DBMS,” see Ohno, 1:28-29) “provide a function to staticize a database virtually.” Ohno, 2:3-6. In such systems, “[a] typical approach is to staticize a virtual database using a virtual database staticizing function of a DBMS and to create a snapshot of a file or logical volume which is to be snapshot at the point in time when the database is staticized.” Ohno, 2:18-21. Additionally, Chikkanayakanahally describes a “shadow copy service” that “quiesc[es] the database” to take a snapshot. Chikkanayakanahally, 8:18-32 (describing using “snapshot” created by “shadow copy service” for “backup”); 12:41-51 (“shadow copy service” “quiesce[s] the database” to “gather information or metadata for the backup operation”).

47. Because of the typical need to pause the database to take snapshots, as well as other factors like the time needed to take a snapshot, POSAs understood that taking backups from the secondary node rather than the primary node beneficially allowed writes to continue to the primary database while the snapshot was being taken from the secondary. My testimony regarding a POSA’s background knowledge is corroborated by, *e.g.*, Chikkanayakanahally, 1:33-41 (describing problems that motivate ability to take “backups” from “redundant,” *i.e.*, secondary, databases), 8:18-32 (describing using “snapshot” created by “shadow copy service” for “backup”); 12:41-51 (“shadow copy service”

“quiesce[s] the database” to “gather information or metadata for the backup operation”).

48. The '340 patent's specification refers to the node from which snapshots are taken as the “Era Active node” or “EA node,” while the claims use the term “active node.” EX1001, 25:56-60, 43:1-2 (claim 1). “Era” refers to Nutanix Era software suite. See EX1114 (blog post regarding Nutanix Era software). I understand that EX1114 was downloaded from the URL <http://web.archive.org/web/20200926222827/https://www.nutanix.com/blog/nutanix-era-databases-made-simple>. I understand that the DNS name web.archive.org represents the website of something known as the “Wayback machine” that is maintained by an organization known as the Internet Archive. I am familiar with the Wayback machine from having used it prior to my work in this proceeding, and I understand that the Wayback Machine allows users to search archived websites maintained by the Internet Archive. Furthermore, I have reviewed EX1058, which I understand is an Affidavit of Mr. Nathaniel E. Frank-White, an employee of the Internet Archive. Based on Mr. Frank-White's explanation of Internet Archive URLs in paragraph 5 of his declaration, and based on my own familiarity with the Wayback Machine, I understand that in an Internet Archive URL, the numerical field that appears after “web/” indicates the date and time at which the page following the number was archived. Thus, I understand the URL from which

EX1114 was downloaded is the Internet Archive's record of the website at the URL <https://www.nutanix.com/blog/nutanix-era-databases-made-simple> archived on September 26, 2020.

49. The patent says “[a] user defined policy may determine which node... is designated as the EA node.... When the EA node fails,” the new EA node to “capture backups from” is “automatically” selected. EX1001, 25:67-26:7.

50. The patent uses the term “switch over policy” to describe a user-defined “policy” that may select both an “initial” EA node from which to take snapshots and a new EA node from which to take snapshots if the initial EA node fails. EX1001, 33:6-19 (“if the **switch over policy** is the primary-only EA policy... the database controller designates the primary node... as the **initial EA node**. If the **switch over policy** is the secondary-only EA policy... the database controller... selects one of the standby nodes as the **initial EA node**...”), 34:11-16 (“if... a new EA node is needed,” then the “**next EA node**” is “select[ed]” “based upon **the switch over policy**”).

51. Examples of “switch over policies” include a “primary-only EA policy” and a “secondary-only EA policy;” the former only allows the primary node to be selected as the new EA node, while the latter only allows a secondary node to be selected as the new EA node. EX1001, 27:7-49.

52. Once the new EA node is selected, collection of snapshots resumes, with snapshots now being collected from the new EA node. EX1001, Abstract, 28:65-29:7.

53. In summary, as discussed above, the '340 patent discusses a system that uses two well-known techniques for protecting databases from failure, namely replication and backup, with separate policies for each. It was well-known in the art to use replication and backup together, as the '340 patent describes, because POSAs understood that the two techniques provided different lines of defense against failure, as I discuss in paragraphs 54-55 below.

54. Replication ensured that if the primary node failed, there was an up-to-date copy of the database available on a different node (a “secondary” node) that could quickly and seamlessly take over from the primary node, thus minimizing downtime of the database. EX1001, 5:40-53; Chikkanayakanahally, 1:14-27; Chockalingam (EX1113), 1:12-27.

55. At the same time, snapshot-based backup allowed the database’s state to be rolled back to some previous point in time in the event of, *e.g.*, some corruption in the data or some catastrophic failure of all replicated copies of the database. EX1001, 3:55-60; Chikkanayakanahally, 1:27-32; Merriman, 14:12-22; Bajaj (EX1011), 2:36-3:7 (discussing backup in systems where there are both a “primary database” and one or more “standby databases”).

56. Furthermore, in systems where replication and snapshots were used together, it was known to POSAs that taking snapshots from **all** of the replicas would be inefficient, because this would result in unnecessary duplication of snapshots of the same data. The '340 patent states:

However, since each redundant instance or copy of the source database is same or substantially similar, creating such multiple backups of the same source databases may be redundant, consume a lot of system resources, require a lot of space to store, may be costly to implement, and therefore, generally undesirable.

EX1001, 6:4-9. In my opinion, this passage in the '340 patent states something that any POSA would have recognized, because POSAs would have understood that it is not necessary to take snapshots from all copies of a database because the copies of the database are all intended to have the same data.

57. Because taking snapshots from all the replicas was inefficient, the '340 patent allows a system administrator to specify one node from which to take snapshots, as was known in the art and discussed above. EX1001, 25:56-60 (referring to the “time machine” being “configured to capture backups,” including snapshots, from “the node that is designated an Era Active node”); Chikkanayakanahally, Abstract (discussing a “backup preference... to identify a first preferred node for a first preferred backup”), 6:31-34 (“...a feature of the

system provides for the ability to designate and trigger a backup on a preferred node...”).

58. The ’340 patent, like the prior art, allows the choice of taking backups from either the primary or a secondary node. *See* EX1001, 26:64-27:6; Chikkanayakanahally, 12:41-51. POSAs understood that the reason for allowing a choice between using the primary node or secondary nodes for backup was because the choice of which database instance to take snapshots from could impact important considerations such as performance. For instance, POSAs understood that because taking snapshots often required pausing the database as I noted above (*see* paragraphs 46-47, where I discuss Chikkanayakanahally, 1:33-41, 8:18-32, 12:41-51; Ohno (EX1015), 1:54-2:21), a requirement or preference for taking snapshots from secondary copies would avoid interrupting write operations to the primary copy of the database.

59. Because the node from which snapshots are taken may itself fail, the ’340 patent, like the prior art, also implements a policy for choosing a new backup node (i.e., a new node from which to take snapshots); this policy, which the ’340 patent calls the “switch over policy” as noted above, is separate from the policy for choosing a new primary node for replication, because the node for backup may not be the primary node. EX1001, 33:6-19; Chikkanayakanahally, 1:33-41, 12:41-51.

B. Challenged Claims

60. The '340 patent's independent claims (1, 7, 13) recite a plurality of nodes including a primary node and one or more secondary nodes. In response to the primary node becoming unavailable, one of the secondary nodes is designated to become the primary node based on a "first policy." Additionally, one of the nodes (which could be the current primary or a secondary node) is designated as an "active" node from which snapshots are collected. In a scenario where the initially-designated active node becomes unavailable, a different node is selected as the "next active node" to capture snapshots from; this selection is "based on a user defined switch over policy." *See* the claim listing in Section VI below.

C. Prosecution History

61. I have reviewed the prosecution history of the '340 patent. During prosecution, the claims were rejected over prior art, and amended in response, multiple times. EX1002, 238-262 (office action rejecting claims), 289 (record of interview discussing rejection, at which no agreement was reached), 305-314 (amendment to claims and accompanying remarks), 338-362 (rejection of claims), 392-393 (record of interview discussing rejection, at which no agreement was reached), 400-428 (amendment to claims and accompanying remarks), 431-434 (advisory action and interview summary). In the last amendment before

allowance, the applicants extensively edited the claims into their final form.

EX1002, 420-428.

62. In allowing these claims, the examiner stated that “[a] reasonable combination of systems that designate a **secondary** node as an initial active node while the primary node is still functional, in combination with a user defined switch over policy, and **capturing snapshots** from the source database or copy of the source database from the **second** node was not found in the prior art.”

EX1002, 451-452.² However, the examiner did not consider references such as Chikkanayakanahally, which, as I discuss in Section V.A below, discloses capturing snapshots from either a primary or secondary node based on user preferences.

D. Person of Ordinary Skill in the Art (“POSA”)

63. I have been informed and understand that for purposes of assessing whether prior-art references disclose every element of a patent claim (thus “anticipating” the claim) and/or would have rendered the claim obvious, the patent and the prior-art references must be assessed from the perspective of a person having ordinary skill in the art (“POSA”) to which the patent is related, based on the understanding of that person at the time of the patent claim’s priority date. I have been informed and understand that a POSA is presumed to be aware of all

² All emphases added unless otherwise indicated.

pertinent prior art and the conventional wisdom in the art, and is a person of ordinary creativity. I have applied this standard throughout my declaration.

64. The '340 patent involves technology in the field of “database management services,” including “protect[ion]” of databases. EX1001, Abstract, 3:10-23, 3:32-60; *see also* my discussion of the '340 patent in Section IV.A above.

65. I have been asked to provide my opinions as to the state of the art in this field by October of 2020. I use this timeframe because the face of the '340 patent indicates an earliest claimed priority date of October 20, 2020. Whenever I offer an opinion in this declaration about the knowledge of a POSA, the manner in which a POSA would have understood the claims of the '340 patent or its description, the manner in which a POSA would have understood the prior art, or what a POSA would have been led to do based on the prior art, I am referencing the October 2020 timeframe, even if I do not say so specifically in each case.

66. In my opinion, a person of ordinary skill in the art (“POSA”) at the time of the '340 patent's claimed October 20, 2020 priority date would have had a bachelor's degree in computer science, computer engineering, or a related field, with two years of academic and/or industry experience with database technologies. A higher level of education may substitute for less experience. This person would have been capable of understanding and applying the teachings of the '340 patent and the prior-art references discussed in this declaration.

67. By 2020, I held a Ph.D. in computer science from the Georgia Institute of Technology, and I had more than 20 years of experience using, researching, and/or teaching about networked storage, storage security, storage virtualization, and databases. Therefore, I was a person of more than ordinary skill in the art during the relevant timeframe. However, I worked with many people who fit the characteristics of the POSA, and I am familiar with their level of skill. When developing the opinions set forth in this declaration, I assumed the perspective of a person having ordinary skill in the art, as set forth above.

V. GROUND 1: CLAIMS 1-30 ARE UNPATENTABLE OVER CHIKKANAYAKANAHALLY, WITH OR WITHOUT MERRIMAN

68. I have been asked to provide my opinion concerning whether claims 1-30 of the '340 patent are anticipated or would have been obvious to a POSA in light of the prior-art references identified in the Petition.

69. For the reasons explained below, it is my opinion that each of claims 1-30 would have been obvious over Chikkanayakanahally (with or without Merriman, as I discuss in Section V.B.1.e below for [1B3]). Furthermore, as I discuss below, POSAs would have found it obvious to apply well-known aspects of POSAs' background knowledge when implementing Chikkanayakanahally's teachings (alone or combined with Merriman's teachings).

A. Chikkanayakanahally (EX1005)

70. As I explain in this section, although Chikkanayakanahally uses slightly different terminology than the '340 patent, POSAs understood that Chikkanayakanahally teaches the same techniques that the '340 patent describes and claims for handling failover in a replicated environment that uses snapshot-based backup.

71. For example, as I discuss further below, Chikkanayakanahally uses the term “preferred node” to refer to what the '340 patent’s claims call the “active node,” but uses the terms “primary node” and “secondary node” to refer to the same nodes that the '340 patent’s claims call “primary node” and “secondary node.” *E.g.*, Chikkanayakanahally, 5:41-6:19 (describing a “computing environment” where there is “an **active or primary** node and an inactive, non-active, passive or **secondary** node,” where “[i]f the active node fails, the other node can assume the responsibilities of the failed node.... The node having the primary databases may be referred to as an active node. A second node in the cluster may host another database instance which in turn hosts a set of corresponding secondary databases.”), 12:18-29 (referring to “primary/secondary node”), 11:9-14 (“...[b]ased on the backup preference, the system identifies a node in the cluster to perform the backup. The identified node for the backup may be referred to as the preferred node.”). Chikkanayakanahally also uses the term

“active node,” but not to refer to what the ’340 patent’s claims call the “active node.” Chikkanayakanahally sometimes uses the term “active node” to refer to the “primary node.” Chikkanayakanahally, 5:41-6:2.

72. As I discuss below, Chikkanayakanahally allows a database to be replicated across a set of nodes such that there is a primary copy and secondary copies distributed across the nodes. Chikkanayakanahally’s policies for choosing the node from which to take snapshots of the database match exemplary policies described in the ’340 patent, as I discuss below. Chikkanayakanahally additionally goes further and describes how its techniques for replication and backup can be scaled to a system that replicates multiple databases across nodes, such that a node can host primary or secondary copies of multiple databases. Chikkanayakanahally, Abstract, 5:60-67. This aspect of Chikkanayakanahally is not relevant to my analysis, however, as more than one replicated database is neither required nor excluded by the ’340 patent’s claims.

73. Chikkanayakanahally “relates generally to... backing up computer data... in a high-availability environment.” Chikkanayakanahally, 1:7-10. The “high-availability environment” may be a “[h]igh-availability cluster[],” which is a “group[] of computers that support server applications that can be reliably utilized with a minimum of down-time.” Chikkanayakanahally, 1:14-17.

74. Like in the '340 patent, the “computers” in Chikkanayakanahally’s “cluster” are called “nodes.” Chikkanayakanahally, 5:17-26. Each node can host one or more databases (or database copies). Chikkanayakanahally, Abstract, 5:60-67.

75. A database on one node can be a “primary database” that is “available for read-write connections from clients.” Chikkanayakanahally, 5:60-64. That primary database can be replicated to one or more corresponding “secondary databases” on other nodes, which “are synchronized with the primary database[]” and “can assume the responsibilities of” the primary database if it fails.

Chikkanayakanahally, Abstract, 5:51-54, 5:65-6:6, 9:10-52, Fig. 2 (describing Figure 2 and noting that “nodes” host “database instances” and that “each database instance hosts one of a primary replica having one or more primary databases or a secondary replica having one or more secondary databases”).

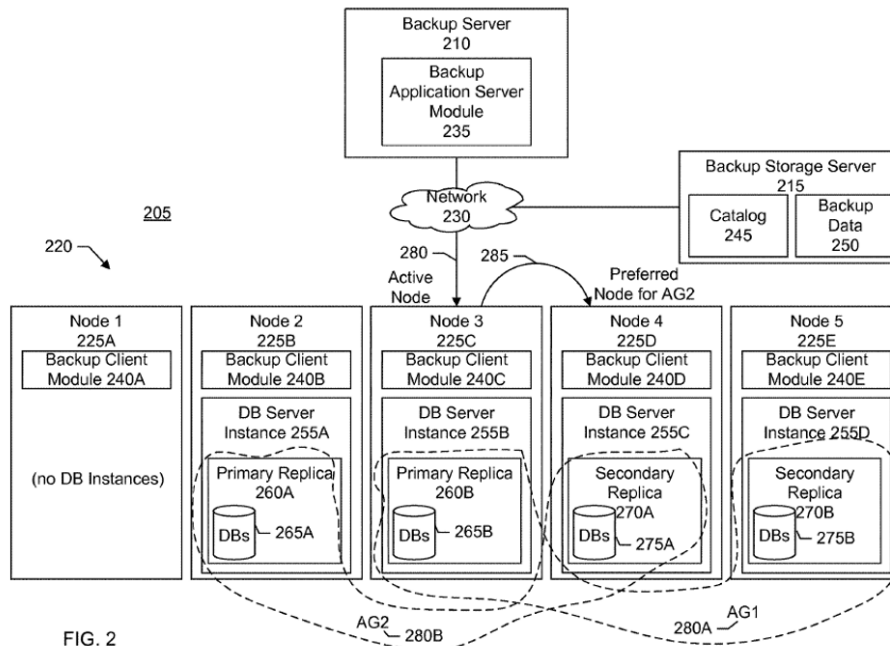


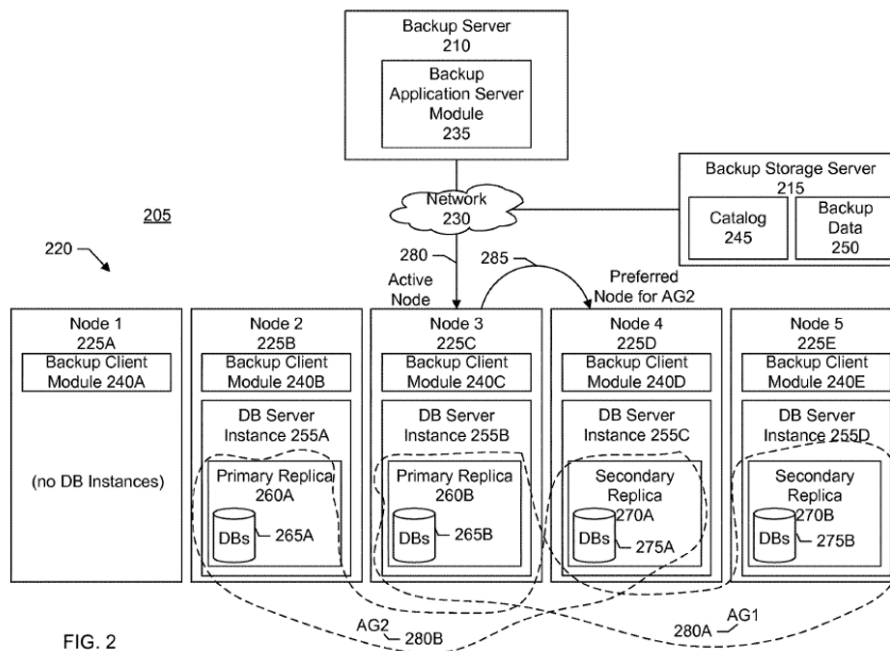
FIG. 2

Chikkanayakanahally, Fig. 2

76. As I discussed above (Paragraph 72), Chikkanayakanahally additionally goes further and describes how its techniques can be scaled to a system that replicates multiple databases. In this scaled system, Chikkanayakanahally teaches that a primary node can host a set of primary databases (collectively called a “primary replica”), each of which has a corresponding secondary database, where those secondary databases form a set (called a “secondary replica”) hosted on a secondary node. Chikkanayakanahally, Abstract, 9:14-52, Fig. 2. The primary replica and secondary replica (potentially including more than one database having replicated copies) are together called an “availability group,” and failover policies can be set to apply to the whole group. Chikkanayakanahally, 6:3-19. This aspect of Chikkanayakanahally is not relevant

to my analysis, however, as more than one replicated database is neither required nor excluded by the '340 patent's claims.

77. Chikkanayakanahally calls the nodes hosting the “primary” and “secondary” databases “primary node[s]” and “secondary nodes,” respectively, and such nodes can be grouped into an “availability group (AG)” as shown in, *e.g.*, Fig. 2. Chikkanayakanahally, 5:41-50, 9:5-37, 12:18-29.



Chikkanayakanahally, Fig. 2

78. Chikkanayakanahally’s secondary databases are replicated copies of the corresponding primary database, just like in the '340 patent. When a write operation is performed on a primary database, that operation is logged in a “transaction log”; “[t]o synchronize the primary and secondary databases, the primary replica sends transaction log records of each primary database to every

secondary database. Every secondary replica... then applies [the logged transactions] to its corresponding secondary database.” Chikkanayakanahally, 9:59-64. In other words, POSAs understood that write operations performed on the primary database are logged, and those same operations can then be performed on any secondary databases that correspond to the primary database, thus making the state of the secondaries consistent with the state of the primary.

79. In the event of a failure of the primary database, *e.g.*, a failure of the node hosting the primary database, “the secondary database[] on a particular node can transition to become the primary database[]. Once the former secondary database[] is brought online or made available as [the] primary database[], client applications can then connect to [it] for read-write access.” Chikkanayakanahally, 6:11-19.

80. Chikkanayakanahally, like the '340 patent, also teaches “a system... for backing up and recovering” these “high-availability databases.” Chikkanayakanahally, 5:17-28, Fig. 2. The system includes a “backup server,” which issues commands (which I discuss in Section V.B below) to perform the backup operation, and a “backup storage server” that stores backups taken from databases. Chikkanayakanahally, 5:17-19, 7:13-17, 10:67-11:2. The “backup server” runs a “backup application server module” that communicates with “backup client modules” (also called “backup application client modules” at, *e.g.*,

Chikkanayakanahally, 5:29-39) that are installed on the nodes.

Chikkanayakanahally, 5:29-39, Fig. 2. The nodes are the hardware devices on which databases run. Chikkanayakanahally, 5:23-28, 5:60-67, Fig. 2.

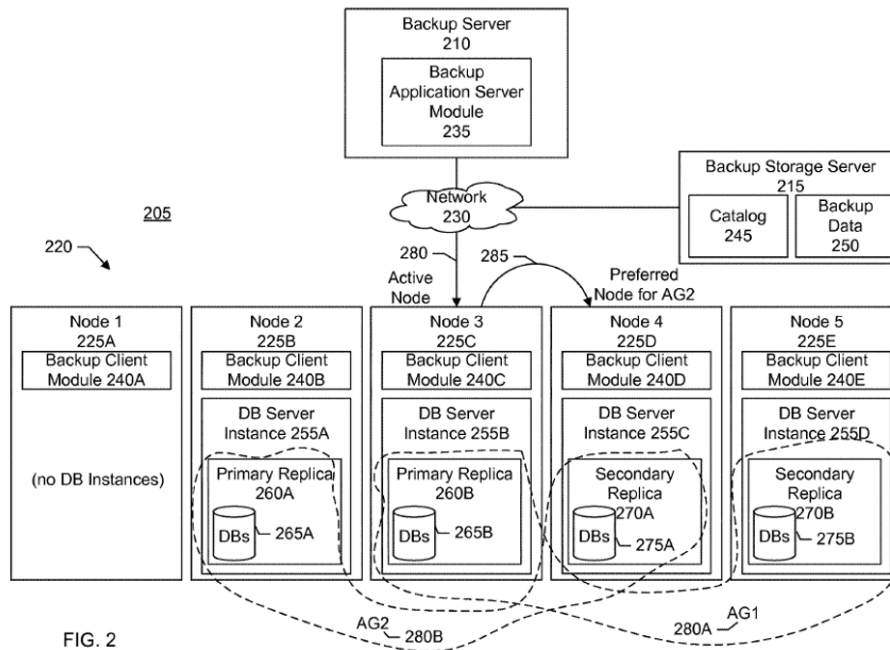


FIG. 2

Chikkanayakanahally, Fig. 2

81. The backup for a database is taken from a “preferred node.”

Chikkanayakanahally, 6:56-61. The backup can be taken by using a “shadow copy service” running on the preferred node to create a “snapshot” of the data stored in the database on that node. Chikkanayakanahally, 8:18-32, 12:30-51.

82. The “preferred node” for creating the backup is selected based on a “backup preference”; the “backup preference” is defined by “[a] user (*e.g.*, administrator)” via a user interface as part of a “configuration” that the “user (*e.g.*, administrator)” defines. Chikkanayakanahally, 10:24-55.

83. Chikkanayakanahally discloses multiple options for how the “backup preference” may indicate the preferred node. For example, the “backup preference” may “specify[] a preferred node on which backups should be performed.” Chikkanayakanahally, 10:41-45. Alternatively, the backup preference may specify a list of nodes in priority order, such that if one node in the list is unavailable for backup, the next node is selected and the backup is attempted again. Chikkanayakanahally, 13:9-19, 14:4-12. The backup preference may also “specify a preference for backups on the primary replica..., a preference for backups on the secondary replica..., a preference (or requirement) for backups on the secondary replica only..., or a preference for backups on any type of replica..., *e.g.*, no preference.” Chikkanayakanahally, 12:10-17. Thus, as I discuss further in my analysis of the claims below, Chikkanayakanahally teaches the same types of node designation and user defined switch over policies for capturing snapshots of a replicated database as the ’340 patent.

B. Mapping to Challenged Claims

1. Claim 1

a. [1PRE]: “A system comprising:”

84. Chikkanayakanahally discloses a “*system[]*” for “backing up in a high-availability environment.” Chikkanayakanahally, 1:7-10; *see also* Chikkanayakanahally, 5:41-42 (“high-availability database *system*”), 19:56-20:49 (claim 8 reciting a “*system* for backing up in a high-availability environment”). As

I discuss in Sections V.B.1.b-V.B.1.j below, Chikkanayakanahally's "system" *comprises* the remaining elements of claim 1.

b. [1A]: "a processor executing computer-readable instructions stored on a memory to"

85. Chikkanayakanahally discloses that "[a] computer-implemented or computer-*executable* version of" its "system may be embodied using, *stored* on, or associated with *computer-readable medium*.... A computer-readable medium may include any medium that participates in providing *instructions* to one or more *processors* for *execution*." Chikkanayakanahally, 4:14-19. The "computer-readable... medium" may be various types of "*memory*." Chikkanayakanahally, 2:33-40. Thus, Chikkanayakanahally's *system comprises a processor executing computer-readable instructions stored on a memory*. As I discuss in Sections V.B.1.c-V.B.1.j below, the tasks performed in Chikkanayakanahally's system include the tasks recited in the remainder of claim 1; thus, Chikkanayakanahally discloses a processor *executing* those *instructions to* perform the remaining tasks recited in the claim.

86. In describing an exemplary "database controller," the '340 patent's specification states that "[t]he processor 230 may include a single stand-alone processing unit, or a plurality of processing units that use the same or different processing technology." EX1001, 14:62-65.

87. As I discuss in Sections V.B.1.c-V.B.1.j below, the tasks recited in claim 1 relate to replication and backup. As I noted in Section V.A above, Chikkanayakanahally's "system" for performing backups (Chikkanayakanahally, 1:7-10) includes a "backup server" and a cluster of nodes. Chikkanayakanahally, 5:18-19, Fig. 2.

88. Chikkanayakanahally discloses an exemplary implementation in which responsibility for choosing which node from which to take backups—i.e., selecting the "preferred node"—is shared between the backup server and one of the nodes. Specifically, Chikkanayakanahally discloses an exemplary algorithm where (1) the "backup server" chooses the "active node" as the "coordinating node" for a backup, and (2) the "coordinating node" uses the backup preference to choose a "preferred node." Chikkanayakanahally, 10:62-11:17 (describing algorithm with reference to Figure 4), Fig. 4. Chikkanayakanahally also calls the "active node" the "primary node." Chikkanayakanahally, 5:43-49.

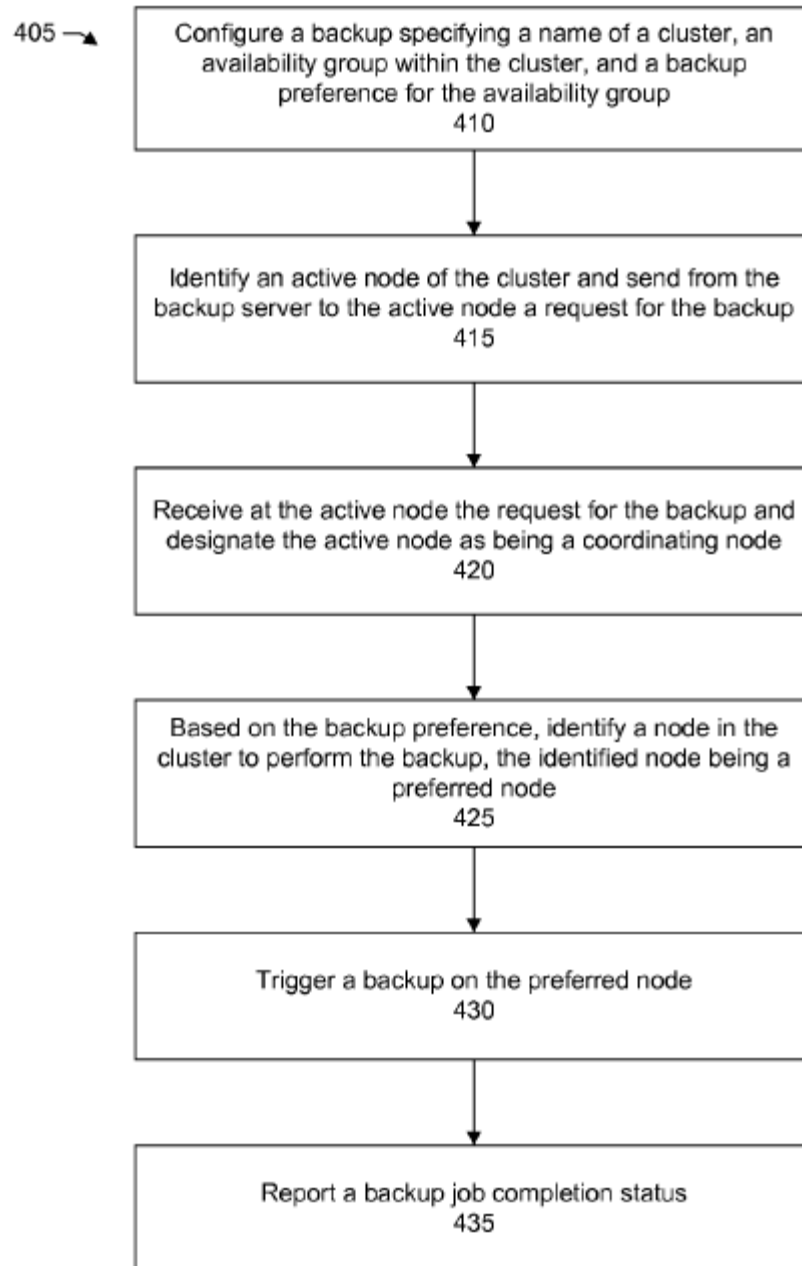


Figure 4

Chikkanayakanahally, Fig. 4

89. In addition, as I discuss in Sections V.B.1.g and V.B.1.j below for [1D] and [1G], the backup server captures snapshots by instructing the nodes to create snapshots and then retrieving the created snapshots. *See also*

Chikkanayakanahally, 11:20-23 (“The backup server, in turn, may generate a backup job request that the backup server sends to the preferred node. In another specific embodiment, the active (or coordinating) node may send the backup job request to the preferred node.”)

90. The nodes and the backup server perform their tasks using “*processors*.” Chikkanayakanahally, 4:14-19 (“A computer-implemented or computer-executable version of the system may be embodied using, stored on, or associated with computer-readable medium or non-transitory computer-readable medium. A computer-readable medium may include any medium that participates in providing instructions to one or more *processors* for execution.”).

91. Additionally, in the exemplary implementation I discussed above, a processor configures the availability group, including designating the primary node. Chikkanayakanahally, 7:42-46 (describing software, which a POSA would have understood is run using processors, used to configure availability groups), 8:57-60 (referring to “databases which are configured with SQL AlwaysOn Groups,” which POSAs understood to be a reference to availability groups configured using software run by a processor); *see also* my discussion of [1B1] in Section V.B.1.c below.

92. Element [1C] recites “a processor.” I have been informed and understand that under the law of claim construction, the indefinite article “a” in a

claim typically means “one or more.” In addition, as I noted in paragraph 86 above, in describing an exemplary “database controller,” the ’340 patent’s specification states that “[t]he processor 230 may include a single stand-alone processing unit, or a plurality of processing units that use the same or different processing technology.” EX1001, 14:62-65. Thus, the exemplary implementation of Chikkanayakanahally I described in Paragraphs 85-91 above meets claim 1 because each claimed step is performed by some *processor* in the system; thus the set of all claimed steps is performed by a set of *processors*. However, to the extent claim 1 is interpreted to require that a single *processor* execute the instructions for all the steps recited in the claim, this implementation would have been obvious over Chikkanayakanahally, for the reasons I explain below.

93. The implementation I discussed above that uses a “coordinating node” is just one exemplary implementation described in Chikkanayakanahally. Chikkanayakanahally, 11:15-17 (“**In a specific embodiment**, the active (or coordinating node) coordinates with the backup server application module to trigger the backup on the preferred node.”), 10:9-13 (saying “FIG. 4 shows an overall flow[] for backing up” but noting that “the process” of Chikkanayakanahally “is not limited to the specific flows and steps presented”). Furthermore, after describing that implementation, Chikkanayakanahally later states that “[t]he backup application module (*e.g.*, **server module**, client module,

or both) may include one or more processes for **determining the preferred node.**” Chikkanayakanahally, 11:24-26. The “server module” of the “backup application” is on the “backup server.” Chikkanayakanahally, 5:29-31, Fig. 2. Additionally, when discussing “backup scenarios” shown in Figs. 2 and 6-7 (Chikkanayakanahally, 14:23), Chikkanayakanahally says its “system executes **one or more** steps of the process shown in FIGS. 4 and 5 and identifies the preferred node.” Chikkanayakanahally, 14:34-36. Thus, in my opinion, POSAs understood Chikkanayakanahally to also disclose an implementation where the backup server alone is responsible for choosing the preferred node.

94. At a minimum, as I explain in paragraphs 95-100 below, POSAs would have found it obvious to implement Chikkanayakanahally’s backup techniques such that the backup server alone performs all steps of the algorithm for choosing the preferred node.

95. Chikkanayakanahally does not say anywhere that sharing responsibility for choosing the preferred node between the backup server and the nodes is critical to Chikkanayakanahally’s teachings. In addition, in my opinion, nothing in Chikkanayakanahally teaches away from an implementation where the backup server alone determines the preferred node.

96. Chikkanayakanahally is directed to solving the “need for improved systems and techniques that allow for parallel backups and the offloading of

backups to redundant computers in a high-availability cluster.”

Chikkanayakanahally, 1:42-45. POSAs understood that these problems can be addressed in an implementation where the backup server chooses the preferred node, because regardless of where the choice is made, parallel backups and offloading of backups to redundant computers would still be achieved.

97. POSAs would have been motivated to implement selection of the preferred node on Chikkanayakanahally’s “backup server” for multiple reasons, as I explain below.

98. First, it was well-known in the art to implement selection of a node from which to take backups on another computer, such as Chikkanayakanahally’s “backup server,” separate from the nodes themselves. My testimony regarding a POSA’s background knowledge is corroborated by various references. For example, McAlister (EX1020), published in 2011, describes a “control plane” that is separate from a “data plane” containing the hardware hosting databases, where the “control plane” includes a “sweeper component” that is responsible for managing “workflows” for tasks including “recovery and backup.” McAlister, [0032], Fig. 2 (below). As another example, Ambastha (EX1007), which shares inventors with Chikkanayakanahally (*see* Ambastha, p. 1, item 72; Chikkanayakanahally, p. 1, item 72) describes a “backup application” that selects a node for backup and that runs on a “server” separate from the servers that host

“availability groups.” Ambastha, 5:60-6:4 (describing system of Figure 2 including “first” through “fifth” “server[s]” hosting “availability group[s]” and a “sixth server” including a “backup application”), 6:22-29 (“FIG. 2 depicts the backup application 228 residing completely on the sixth server...”), 8:4-25 (“backup application” on backup server sets “preferred node”), Fig. 2 (below). Another example is Ambastha-634 (EX1021), which is cited on the face of Chikkanayakanahally (*see* Chikkanayakanahally, p. 1) and has similar disclosures as I discussed above to Ambastha of a “backup application” that selects a node for backup and that runs on a “server” separate from the servers that host “availability groups.” Ambastha-634, 6:17-28, 6:46-48, 7:34-49, Fig. 2.

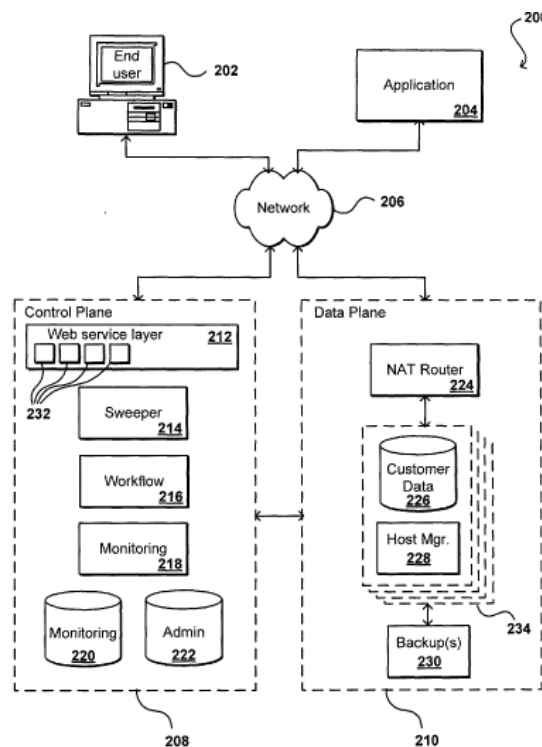


FIG. 2

McAlister, Fig. 2

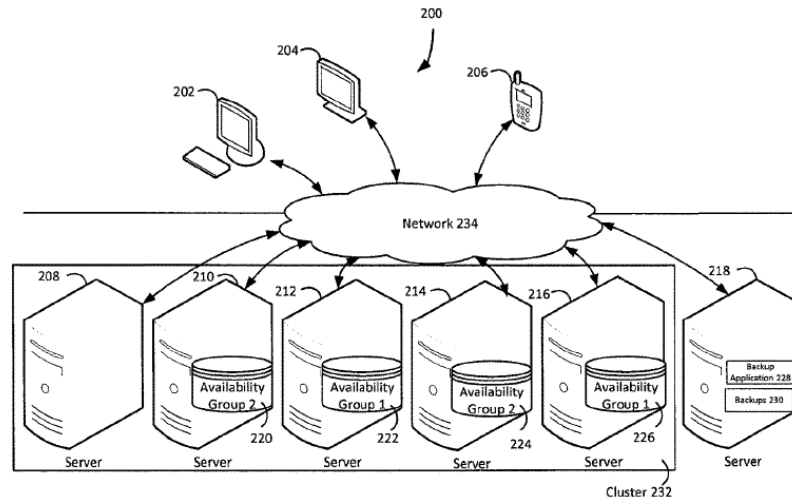


FIG. 2

Ambastha, Fig. 2

99. Second, POSAs understood that having the selection process take place in one location is beneficially simpler than having it split between multiple locations. POSAs also understood that having the backup server choose the preferred node, rather than having one of the nodes in the cluster choose the preferred node, would reduce the number of tasks the primary node (also called “active node” in Chikkanayakanahally) had to perform, which would beneficially contribute to Chikkanayakanahally’s goal of reducing the use of “computing resources” of the “active node” that “could otherwise service client requests.” Chikkanayakanahally, 6:23-31.

100. POSAs would reasonably have expected success in such an implementation. Chikkanayakanahally discloses that the “backup server” communicates with the nodes on which the primary and secondary databases run. Chikkanayakanahally, 5:29-39, 5:60-67, Fig. 2. For example, the backup server can send requests to nodes. Chikkanayakanahally, 10:67-11:2. The “backup server” can also receive information, including the backup data itself, from the nodes. Chikkanayakanahally, 5:31-38, 11:47-51, Fig. 2. Thus, POSAs understood that the backup server had the necessary capabilities to communicate with the nodes to coordinate the backup. Additionally, in my opinion, based on my knowledge of a POSA’s skill as I discussed in Section IV.D above, implementing the selection process in one computer instead of two separate computers was a simple programming task that was within a POSA’s skill.

101. Thus, for the reasons I discussed in paragraphs 95-100 above, it would have been obvious to implement a *processor* in Chikkanayakanahally’s “backup server” to perform limitations [1C], [1E], and [1F], which are the limitations related to selecting a node for backup, as I discuss below in Sections V.B.1.f, V.B.1.h, and V.B.1.i, respectively.

102. Further, as I discussed in Section V.A, Chikkanayakanahally discloses “availability groups” containing primary and secondary databases, where a secondary database can become a primary in the event of a failure of the primary.

Chikkanayakanahally does not expressly specify which computer is responsible for initially selecting the primary database or selecting a new primary database if the current primary fails. However, POSAs understood that, in a replicated database environment where replicas of a database ran on a collection of nodes, it was well-known for there to be an additional computer, **distinct** from the nodes on which the replicas run, that selects the primary replica and then designates a new primary in case the initial primary failed. My testimony regarding a POSA's background knowledge is corroborated by, for example, Zhao (EX1018), 10:42-46 (noting it was "conventional" that "control plane processors" that are separate from nodes "control[] designation and de-designation of primary and secondary nodes"), Fig. 2. Zhao was filed in 2021, but Zhao claims priority to a provisional application filed on September 2, 2020, which has the same relevant disclosure. *See* Zhao, p. 1, item (60); U.S. Provisional Patent Application No. 63/073,513 (EX1019), [0050], Fig. 2. *See also* McAlister (EX1020), [0073] ("promoting a secondary replica to be the new primary replica.... may be performed by the control plane"); Rangaiah (EX1016), 4:20-34 (referring to "an identification module 104 programmed to identify both (1) a primary node that services an application and (2) at least one failover node designated to service the application if the primary node were to fail"), 4:45-57 (saying the various "modules" in figure 1, including

the “identification module,” can be on the nodes **or** on a separate “management server” as shown in figs. 2 and 3), Figs. 1-3.

103. POSAs would at minimum have found it obvious to implement Chikkanayakanahally in this well-known manner such that the Chikkanayakanahally’s “backup server,” which communicates with the nodes for the purpose of configuring backups, also identifies the initial and replacement primary nodes. POSAs would have been motivated to implement the backup server to manage both replication and backups for the convenience of having a single entity handle the related tasks of managing backups and replication. POSAs would have reasonably expected success with such an implementation, for multiple reasons. As I noted in paragraph 100 above, POSAs understood that Chikkanayakanahally’s backup server has the capability of communicating with the nodes to configure backups; thus, implementing the backup server to also communicate with the nodes for purposes of designating nodes for replication would simply require the use of communication capabilities that the backup server already has. Furthermore, implementing the backup server to perform an additional task was a software engineering task that was within a POSA’s skill.

104. Thus, it would have been obvious to implement the *processor* in Chikkanayakanahally’s backup server to perform limitations [1B1]-[1B3] in claim 1, which relate to designating nodes for replication (as I discuss in Sections

V.B.1.c-V.B.1.e below), in addition to limitations [1C], [1E], and [1F], which are the limitations related to selecting a node for backup.

105. Finally, as I detail in Sections V.B.1.g and V.B.1.j below for [1D] and [1G], the *processor* in the backup server captures snapshots by instructing nodes to take snapshots, in the same manner described in the '340 patent.

- c. **[1B1] “select one of a plurality of nodes as a primary node capable of storing a source database, the remaining ones of the plurality of nodes being a secondary node, each of the secondary node capable of storing a copy of the source database”**

106. In the '340 patent's exemplary system, the “primary” node is the node hosting the copy of the database to which updates are made first, while the “secondary” or “standby” nodes are the nodes to which the updates to the “primary” copy are replicated. EX1001, 25:3-21, *see also* my discussion in Section IV.A.

107. Similarly, in an “availability group” of Chikkanayakanahally, a “*primary* node” hosts the “primary database” to which updates are made, and those updates are copied to the “secondary databases” on “secondary nodes,” where the updates are copied using the “transaction log” as, I discussed in Section V.A above. Chikkanayakanahally, 5:41-6:2, 9:5-37, 9:59-64, 12:18-29.

108. As I discussed in paragraph 71 above, Chikkanayakanahally also uses “active node” as another term for the “primary node,” and uses “inactive, non-

active, [or] passive” as other terms for the “secondary node.”

Chikkanayakanahally , 5:43-49 (referring to “active or primary node” where “[a]n active node can be a node... actively servicing database requests”), 14:62-15:11 & Fig. 7 (showing example where “active node 2” hosts “primary replica 260A” for availability group AG2).

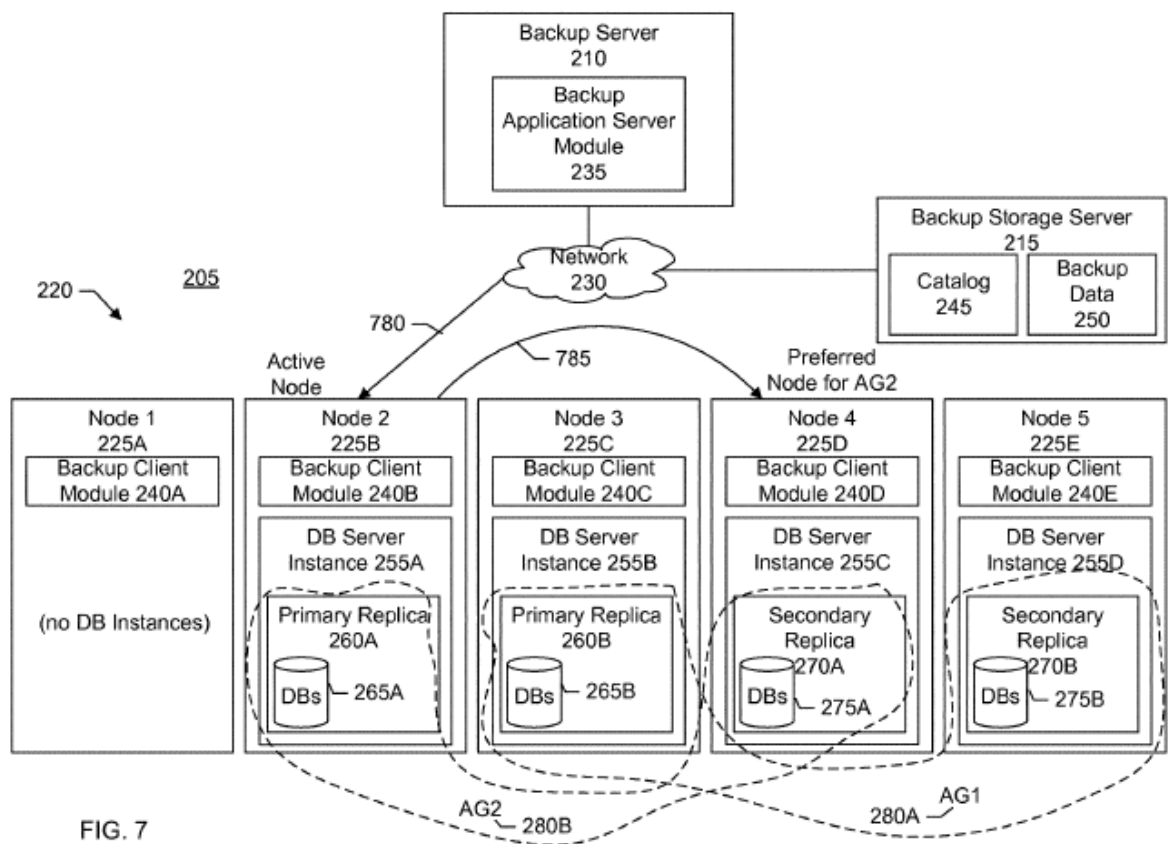


FIG. 7

Chikkanayakanahally, Fig. 7

109. Thus, in Chikkanayakanahally’s exemplary “availability group” in Figure 7, Node 2, which hosts the “Primary Replica” of AG2, is a *primary node*,

and Node 4, which is the *remaining one of the* availability group's *plurality of nodes*, is a *secondary node*.

110. The '340 patent describes a “*source database*” as a database from which snapshots are captured. EX1001, 3:63-65 (“a cloned database may be created from a snapshot captured from the source database”), 3:38-40 (“[t]he database from which the cloned database is created may be considered an original or source... database”). A “fully functional copy of [a] database may be considered a clone of [the] database.” EX1001, 3:36-38.

111. In Chikkanayakanahally, “backups” of a database in an availability group are created by taking “snapshots” of the database, from either the primary or secondary node, as I discussed in Section V.A above. Chikkanayakanahally, 8:18-32, 12:10-17, 12:30-51. POSAs understood that these backups may be used to create fully functional copies of the database, at least because Chikkanayakanahally discloses using the snapshot-based backups to “rollback” the database to a certain point in time. Chikkanayakanahally, 1:28-32.

112. Thus, Node 2 in Figure 7, which hosts the primary replica containing the primary database, is *capable of storing a source database*, where the *source database* is the primary database.

113. Furthermore, Node 4 stores a “Secondary Replica” with a “secondary database” that is a copy of the primary database. Chikkanayakanahally, 6:6-16,

9:14-52, 9:59-64; *see also* my discussion of Chikkanayakanahally in Section V.A above. Thus, Node 4, which is a *secondary node*, is *capable of storing a copy of the source database*.

114. As I explain in paragraphs 115-120 below, POSAs understood that a *processor* in Chikkanayakanahally's system *executes computer-readable instructions stored on a memory to select the primary node* in an availability group, which is sometimes called the "active node" as discussed above.

115. The '340 says that in some embodiments, a "database controller" including a processor may "select... a primary node... based upon selection by a user or other criteria." EX1001, 14:40-43, 24:51-56, Fig. 2.

116. Chikkanayakanahally discloses that availability groups may be configured using existing prior art software products such as Microsoft's "AlwaysOn" product. Chikkanayakanahally, 7:42-62, 8:57-60, 10:21-23. POSAs understood that configuring an availability group using "AlwaysOn" involved using a *processor* to execute software to configure the availability group, including setting the primary node (either automatically or as directed by a user). My testimony regarding a POSA's background knowledge is corroborated by, *e.g.*, *Use the Availability Group Wizard (SQL Server Management Studio)* (EX1109), which discusses how to use the SQL Server Management Studio software to configure availability groups. *See* EX1109, 3-4 (discussing "select databases" interface to let

user “*select* one or more of the listed databases” as “*initial primary* databases,” where the databases run on “servers” in a “cluster”). Another example is “SQL Server AlwaysOn Availability Group Backup Preference Setting” (EX1108), which similarly discusses how to configure availability groups. *See* EX1108, p. 1. *See also* “7 more ways to Query Always on Availability groups,” (EX1107), p. 1.

I understand that EX1107 was downloaded from the URL

<https://web.archive.org/web/20190330182441/https://sqlundercover.com/2019/02/19/7-more-ways-to-query-always-on-availability-groups/>. Based on my familiarity

with the Wayback Machine as detailed above in paragraph 48, I understand the URL from which EX1107 was downloaded is the Internet Archive’s record of the website at the URL <https://sqlundercover.com/2019/02/19/7-more-ways-to-query-always-on-availability-groups/> archived on February 19, 2019. I also understand that EX1108 was downloaded from the URL

<https://web.archive.org/web/20190330182441/https://www.mssqltips.com/sqlservertip/4976/sql-server-alwayson-availability-group-backup-preference-setting/>.

Based on my familiarity with the Wayback Machine as detailed above in paragraph 48, I understand the URL from which EX1108 was downloaded is the Internet Archive’s record of the website at the URL

<https://www.mssqltips.com/sqlservertip/4976/sql-server-alwayson-availability-group-backup-preference-setting/> archived on March 30, 2019. I understand that

EX1109 was downloaded from the URL

<https://web.archive.org/web/20200120100053/https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/use-the-availability-group-wizard-sql-server-management-studio?view=sql-server-ver15>. Based on my

familiarity with the Wayback Machine as detailed above in paragraph 48, I

understand the URL from which EX1109 was downloaded is the Internet

Archive's record of the website at the URL [https://docs.microsoft.com/en-](https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/use-the-availability-group-wizard-sql-server-management-studio?view=sql-server-ver15)

[us/sql/database-engine/availability-groups/windows/use-the-availability-group-wizard-sql-server-management-studio?view=sql-server-ver15](https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/use-the-availability-group-wizard-sql-server-management-studio?view=sql-server-ver15) archived on January

20, 2020.

117. POSAs understood that software comprises *executable instructions stored on a memory*. My testimony regarding a POSA's background knowledge is corroborated by, e.g., Hansen (EX1116), 4:39-42 ("Host device 37 includes a processor 39 and a memory 40, which stores software 41 (*i.e.*, machine-executable instructions) and database 42."), 5:46-48 (describing the processor "executes software (machine-executable instructions)[] stored in memory[] to perform its portion of process"); Hughes (EX1117), 11:63-66 ("As will be appreciated by those skilled in the art,...software [is] executable instructions that cause a...processor to implement the functionality described above."); Grucci (EX1118),

4:43-46 (describing that a computer “includes a processor[] used to perform executable instructions... stored in a memory”).

118. At minimum, a *processor* executing the recited *instructions* to perform the *selection* would have been obvious, as it was customary for processors to perform such selection and thus would have been within a POSA’s skill to implement. See Zhao (EX1018), 10:42-46 (noting it was “conventional” for “control plane processors” to “control[] designation and de-designation of primary and secondary nodes”); EX1019 (Zhao provisional application, filed September 2020), [0050].

119. Moreover, as I discussed in Section V.B.1.b for [1A], in an obvious implementation of Chikkanayakanahally, the *processor* in the “backup server” executes instructions for performing tasks related to replication. Because selection of a primary node is a task related to replication, the backup server’s *processor* performs this task in such an implementation.

120. Furthermore, Chikkanayakanahally discloses that as part of one exemplary technique for performing backups, “[a] request for the backup is sent from the backup server to the active node, e.g., node 2” in the Figure 7 example. Chikkanayakanahally, 10:67-11:2. As I explained in Paragraph 108 above and in Section V.A, Chikkanayakanahally uses the term “active node” to refer to what the ’340 patent refers to as a “*primary node*,” since the “active node” hosts the

primary database. Chikkanayakanahally, 5:43-49, 14:62-15:11, Fig. 7. POSAs understood, or at a minimum would have found obvious, that since the backup server knows which node is the active node (because the backup server sends commands to the active node), the backup server is the one that initially *selects* the active, *i.e. primary*, node.

121. Chikkanayakanahally also discloses a system where there is more than one *secondary node* that are each *capable of storing a copy of the source database*. Chikkanayakanahally's Figure 7 illustrates exemplary availability groups with one secondary node each, but Chikkanayakanahally also explicitly discloses availability groups with multiple secondary nodes. *See, e.g.*, Chikkanayakanahally, 9:38-67 (disclosing three-node availability group with a "primary replica" on the "first node" and a "first secondary replica" and "second secondary replica" on the "second node" and "third node" respectively), 6:6-15.

122. An exemplary illustration of such a group is shown in the modified version of Figure 7 below, which illustrates the second secondary replica on a "Node 6" within AG2. In this modified Figure 7 example, Nodes 2, 4, and 6 are a *plurality of nodes*, with Node 2 being the recited *primary node* and Nodes 4 and 6 each *being a secondary node*, each of which is *capable of storing a copy of the source database, wherein updates made to the source database on the primary*

node are able to be copied to the copy of the source database on the secondary node.

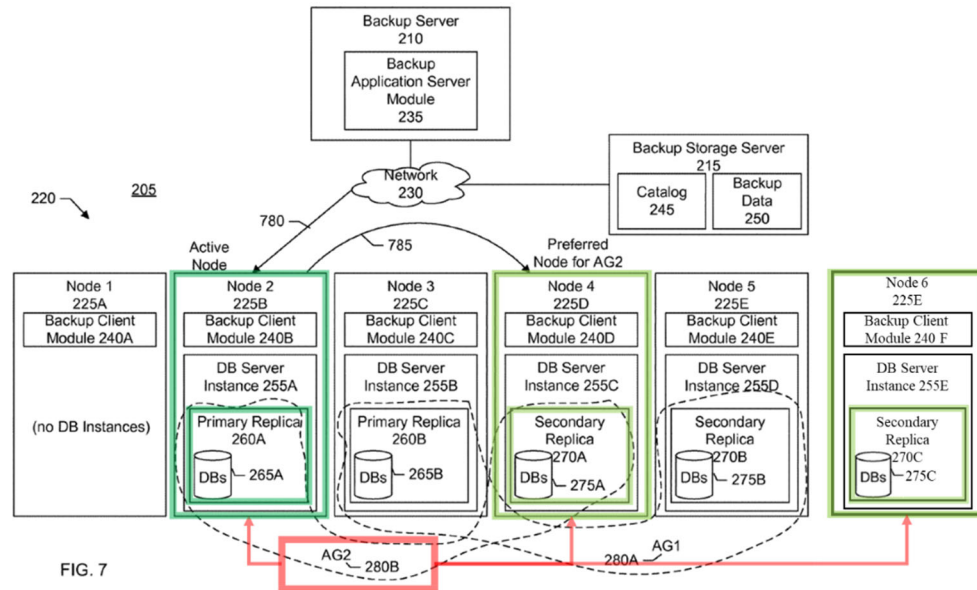


Illustration of Chikkanayakanahally embodiment (adapted from Fig. 7)

- d. [1B2] “wherein updates made to the source database on the primary node are able to be copied to the copy of the source database on the secondary node”

123. As I discussed in Section V.B.1.c for [1B1], the “primary database” on the “Primary Replica” on Node 2 in Chikkanayakanahally’s Figure 7 is a *source database on the primary node*, while the “secondary database” on the “Secondary Replica” on Nodes 4 is a *copy of the source database on a secondary node*.

124. Chikkanayakanahally also explains that “secondary databases are synchronized with the primary databases,” and describes an exemplary technique “to synchronize the primary and secondary databases” in which “the primary replica sends transaction log records of each primary database to every secondary

database. Every secondary replica caches the transaction log records (hardens the log) and then applies them to its corresponding secondary database.”

Chikkanayakanahally, 5:67-6:1, 9:53-67.

125. POSAs understood that the “transactions” logged in the “transaction logs” are *updates* to the database, and that “appl[ying]” those transactions to the “secondary database” results in those updates being *copied* to the secondary database, as [1B2] recites. *See, e.g.*, EX1001, 4:52-56 (“transaction logs may be captured to record all transactions or changes,” *i.e.*, *updates*, “in...the source database”), 18:3-6 (“cloned databases... may be refreshed **updated** to incorporate any **changes** that may have occurred in the source databases”), 25:13-21 (“To **update** the standby nodes..., synchronous or asynchronous policies may be used.... With a synchronous policy, the standby nodes 415 and 420 may be **updated** simultaneously or substantially simultaneously..... With an asynchronous policy, **changes** to the source database on the primary node... may be carried over to the standby nodes... in batches.”). My testimony regarding a POSA’s background knowledge is also corroborated by EX1006, 20:59-64 (disclosing that “transaction logs represent[] updates to the database since a previous point in time.”); EX1060, 6:66-7:31 (“Once... the operations in a transaction have been applied to the database, ... confirmation [is provided] that the transaction has been successfully completed (or committed)....[T]he **transaction log** 260 may contain

log entries 265 representing transactions [*i.e.*, **updates**] which have been committed to the database....”).

126. Therefore, *updates made to the source database on the primary database are able to be copied to the copy of the source database on the secondary node* as required.

- e. **[1B3] “and wherein in response to the primary node becoming unavailable, one of the secondary node is designated to become the primary node based on a first policy”**

127. Chikkanayakanahally discloses that in an availability group (AG), “the secondary databases serve as potential failover targets” for the primary database, such that “[d]uring a failover” of the primary database, “the secondary databases on a particular node... can transition to become the primary databases.” Chikkanayakanahally, 6:3-19. POSAs understood that one cause of a “failover” of a “primary database” is that the *node* on which the primary database runs becomes *unavailable* due to, *e.g.*, a hardware failure. Chikkanayakanahally, 1:20-22 (“if a server running a particular application crashes, the application will be **unavailable** until the crashed server is fixed”), 13:66-14:3 (“[a] replica may be unavailable or offline for backups if... the node hosting the replica may have suffered a failure or the administrator may have flagged or configured the node to be **unavailable** for backup”).

128. Thus, POSAs understood that in the Figure 7 example (below), *in response to the primary node, i.e. Node 2, becoming unavailable, a secondary database on a secondary node, e.g., Node 4, becomes the primary database. As I discussed in Section V.B.1.c for [1B1], the node hosting the primary database is the primary node; thus, POSAs understood that in the event of a failover, the node hosting the secondary database that becomes the primary database is designated to become the primary node.*

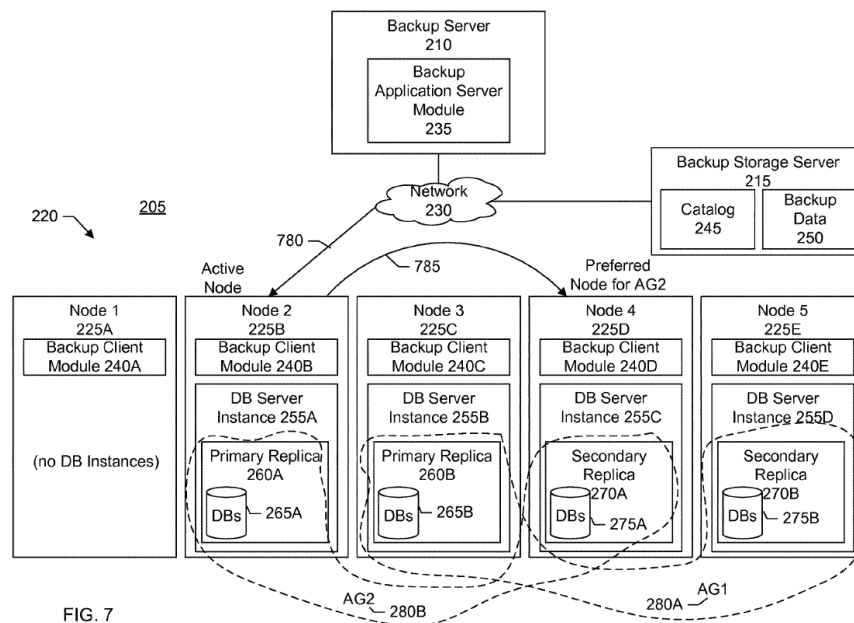


FIG. 7

Chikkanayakanahally, Figure 7

129. Furthermore, this designation happens *based on a first policy* as [1B3] recites. The '340 patent's specification does not define the word "policy" or use that word in the context of describing how to choose a new primary node (on the

other hand, the patent does use the word “policy” in the context of describing how to choose a new **active** node, as I will discuss in Sections V.B.1.f and V.B.1.i below for [1C] and [1F]).

130. As I noted in Section IV.A above, the ’340 patent mentions various “criteria” that a “database controller” may use to “select a new primary node,” but these are merely examples that “may” be used in “embodiments.” EX1001, 25:22-36.

131. Thus, POSAs understood that Chikkanayakanahally’s above-discussed teaching of choosing a new primary node from among the secondary nodes (Chikkanayakanahally, 6:3-19) satisfies the claimed *first policy*. See Webster’s (EX1112), 386 (dictionary definition of “policy” is “a definite course or method of action selected to guide and determine present or future decisions”).

132. Alternatively, choosing the new primary node based on a *policy* including a set of criteria would have been an obvious implementation of Chikkanayakanahally’s system to, *e.g.*, beneficially implement Chikkanayakanahally’s teachings and select the optimal new primary node, in view of POSAs’ background knowledge of well-known such policies and reasonable expectation of success to do so, as I discuss in paragraphs 133-136 below.

133. POSAs understood that various policies with various sets of criteria for choosing the new primary node were well-known in the art. My testimony

regarding a POSA's background knowledge is corroborated by, *e.g.*: Merriman (EX1106), 2:52-55 (corroborating POSAs' background knowledge to apply "consensus protocols" for "elect[ing]" new primary node including "freshest data, best location, most robust hardware, among other options"); Rath (EX1119), Abstract (corroborating background knowledge to implement "master failover protocol" for choosing new "master replica for a replica group" based on factors such as "whether another replica supports" a given replica to become the master or "stores more recent data or metadata"); Rath-441 (EX1120), Abstract, 63:41-50, Fig. 28 (corroborating background knowledge to elect replica as new "master replica" if it "succeeds in building a quorum" to become the master).

134. For example, Merriman (EX1106), like Chikkanayakanahally, relates to "distributed database systems and methods for implementing asynchronous replication with automatic failover." Merriman, 1:6-9. Like Chikkanayakanahally's "availability group," Merriman's "replica set" includes "a cluster of nodes that host a database." Merriman, 2:15-20. And like Chikkanayakanahally, Merriman discloses "establish[ing] a primary node in a plurality of nodes hosting database instances" where "[t]he primary node is responsible for accepting write operations and propagating them to the secondary nodes in its replica set." Merriman, 2:11-15.

135. Merriman teaches that in such a system, the selection of the primary node “can take place as part of an initial setup operation and/or can occur as part of an election protocol.” Merriman, 8:63-67, Fig. 6.

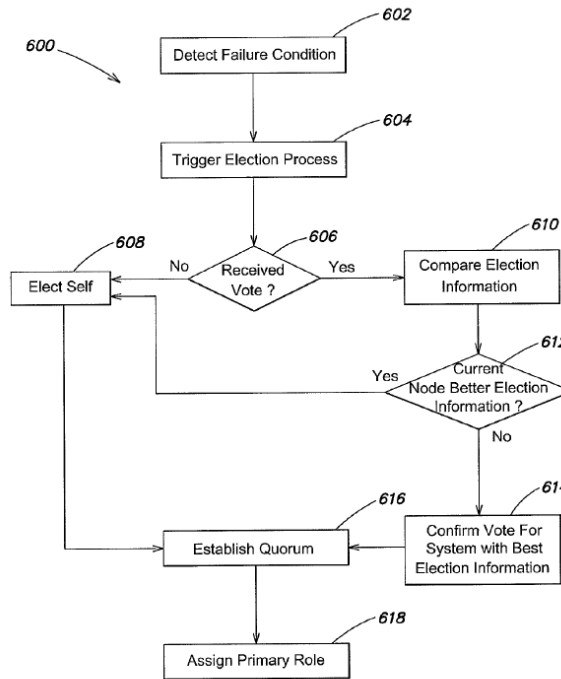


FIG. 6

Merriman, Fig. 6

136. Elections can take place “at initialization to elect a primary node” and also “in response to failure.” Merriman, 8:67-9:3. The criteria that may be considered in electing a new primary node include selecting the node having the “freshest data, best location, [or] most robust hardware, among other options,” including random selection. Merriman, 2:52-55, 6:1-7, 10:3-27, 22:46-53; 30:50-59. When an election takes place in response to a failure of the original primary

node, the candidates to be elected as the new “primary node” are the “secondary nodes” (which Merriman also refers to as “slave systems”). Merriman, 22:11-12 (“election of a new primary identifies secondary node with the freshest data” as the primary), 9:4-5, 10:34-49, 16:10-14, 21:11-19.

137. As I discuss below, the election can be conducted by the secondary nodes communicating with each other, or by another computer that is distinct from the nodes.

138. POSAs looking to implement Chikkanayakanahally’s teaching to *designate a secondary node to become the new primary node in response to the primary node becoming unavailable* (as I discussed in paragraph 127 above, citing Chikkanayakanahally, 1:20-22, 6:3-19, 13:66-14:3) would have been motivated to implement Merriman’s above-discussed “election” scheme to achieve the benefits Merriman describes, including maintaining consistency across nodes, reducing complexity, and choosing the best candidate as the replacement primary node. Merriman, 2:32-43, 2:49-55.

139. Furthermore, as I explain below, POSAs would have been motivated to implement the election either by having Chikkanayakanahally’s secondary nodes conduct the election among themselves, or by having Chikkanayakanahally’s “backup server” choose the new primary node.

140. Merriman discloses embodiments where the “secondary nodes” “agree” among themselves on a new “master” (*i.e.*, primary) node. Merriman, 9:4-7, 10:3-14. For example, in one “example process” for the election shown in Figure 6, each node participating in the election compares its “election values, for example, a higher priority, fresher data, size of hardware, etc.” against the information from other nodes (*see* step 610 in figure 6); based on this, the node determines to either vote for itself as the new primary node, or back another node. Merriman, 22:30-67, Fig. 6 (annotated below).

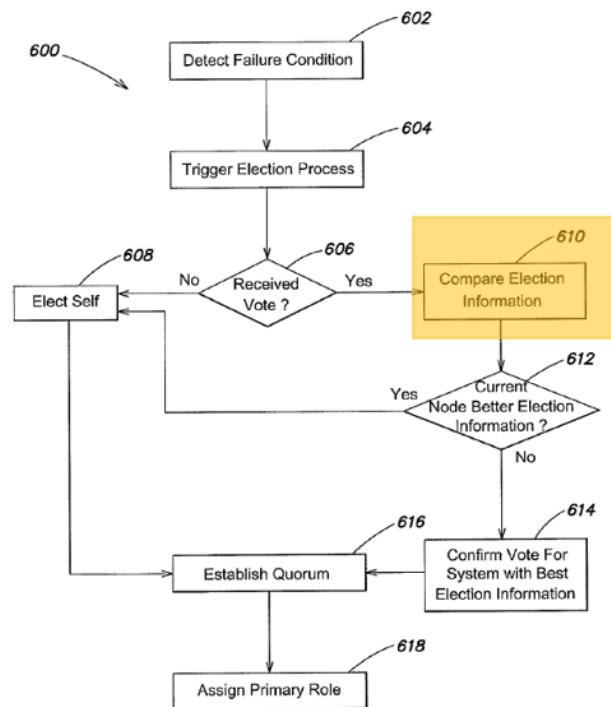


FIG. 6

Merriman, Fig. 6 (annotated)

141. Furthermore, to the extent this claim element requires a single processor chooses the new primary node and also perform the other recited steps, this would have been an obvious implementation of Chikkanayakanahally in view of Merriman, as I discuss in the next paragraph.

142. Merriman does not require all nodes to participate in the election, and instead discloses that voting by “at least one node” is sufficient. *E.g.*, Merriman, 31:35-62 (claim 1 reciting “establishing one of the plurality of secondary nodes as the new primary node” without requiring specific process), 36:37-39 (claim 11 reciting “voting by at least one node... for the new primary node”). Additionally, Merriman discloses that a computer that is distinct from the nodes can serve as an “arbiter” that “facilitates the election protocol” and that is “outside a particular data center” hosting the nodes in the cluster. Merriman, 10:57-11:10. POSAs would have been motivated to have Chikkanayakanahally’s backup server choose the new primary, as an alternative to having the nodes vote among themselves, to achieve the stated goal of Merriman’s “arbiter” to “improve efficiency during election of a new master.” Merriman, 10:57-11:10. POSAs would also have been motivated to implement Merriman this way because having a distinct computer from the nodes choose the primary node was well known. I discuss corroborating evidence in Section V.B.1.b above for [1A]. *See, e.g.*, paragraph 102, discussing Zhao

(EX1018), 10:42-46; Zhao provisional (EX1019), [0050]; McAlister (EX1020), [0073]; Rangaiah (EX1016), 4:20-34 , 4:45-58, Figs. 1-3.

143. POSAs would reasonably have expected success in either above-discussed Chikkanayakanahally+Merriman implementation (election conducted by nodes or backup server) for the following reasons.

144. As I discussed above, Merriman, like Chikkanayakanahally (*see* Section V.A), teaches that its techniques work in a system in which a database is replicated across primary and secondary nodes. Moreover, Merriman discloses that nodes can communicate with each other and with other entities (such as a backup server) using a network such as the Internet, which Chikkanayakanahally also discloses. Merriman, 9:16-29, 29:4-21; Chikkanayakanahally, 3:8-24, 5:17-23, Figs. 1-2. Thus, Chikkanayakanahally discloses a system with sufficient components to implement Merriman's teachings.

145. Based on my knowledge of a POSA's skill as I discussed in Section IV.D above, it would have been within a POSA's ordinary software engineering skill to implement Merriman's primary-node selection teachings in Chikkanayakanahally's system, either on the secondary nodes or on the backup server.

f. [1C] “designate a first node from the primary node or the secondary node as an initial active node”

146. In discussing Figures 4A-4B, the '340 patent refers to designating an “initial EA node (*e.g.*, ***initial active node***)” as “the node in the database cluster... from which the highly available time machine... captures snapshots.” EX1001, 32:55-33:5. The '340 patent's specification also describes “designat[ing]” either a “***primary node***” or one of the “standby nodes” (a.k.a. “***secondary nodes***”) as the “initial EA node” based whether the “switch over policy” is a “primary-only EA policy,” a “prefer-primary EA policy,” a “secondary-only EA policy,” a “prefer-secondary EA policy,” or an “Any’ EA policy.” EX1001, 33:6-24, Abstract, 24:9-24; *see also* my discussion in Section IV.A above.

147. Chikkanayakanahally discloses that “[b]ased on the backup preference, the system identifies a node in the cluster to perform the backup. The identified node for the backup may be referred to as a preferred node.” Chikkanayakanahally, 11:9-13. Backups may be performed by capturing “snapshots” from the database. Chikkanayakanahally, 8:18-32, 12:30-51; *see also* my discussion of Chikkanayakanahally in Section V.A above, and my discussion of [1D] in Section V.B.1.g below.

148. Thus, POSAs understood that a “preferred node” in Chikkanayakanahally (*i.e.*, the node used to perform backup by capturing snapshots from its database) is what the '340 patent calls an “*active node*.”

149. In Chikkanayakanahally’s Figure 7 example (annotated below), Node 4, which is a *secondary node* (see my discussion of [1B1] in Section V.B.1.c), is the “preferred node,” *i.e.*, the *active node* as claimed, for AG2.

Chikkanayakanahally, 15:3-5.

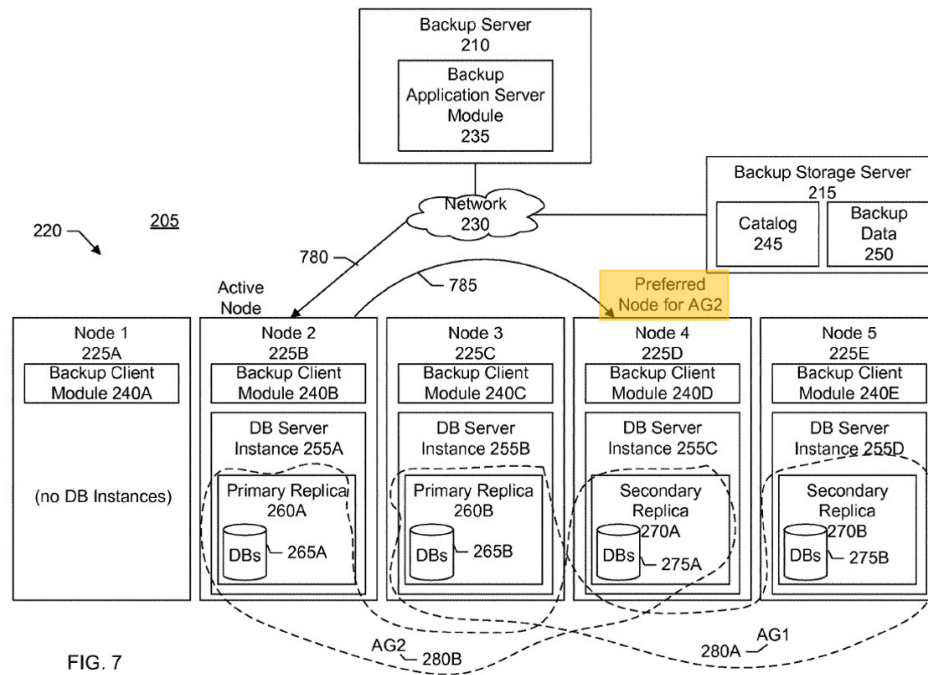


FIG. 7

Chikkanayakanahally, Fig. 7 (annotated)

150. Node 4 is an *initial* active node because, as I explain in Section V.B.1.i below for [1F], a different active node may be subsequently selected. EX1001, 34:33-44 (distinguishing between the “initial EA node” from which snapshots are taken and the “next EA node” selected when the initial becomes unavailable). This node is also a *first node* that is distinct from a *second node* that

is selected when the *first node* becomes unavailable, as I also discuss in Section V.B.1.i below for [1F].

151. POSAs understood Chikkanayakanahally to disclose that a *processor* in Chikkanayakanahally's system *designates* a node as the preferred node for the reasons I discussed in Section V.B.1.b above for [1A]. If claim 1 requires that a single *processor* performs both the designation in [1C] and the selection in [1B1], this is met by the obvious implementation of Chikkanayakanahally in which the *processor* in the "backup server" performs all steps of the claim, as I also discussed in Section V.B.1.b above for [1A].

152. Furthermore, Chikkanayakanahally discloses multiple different ways in which Node 2 (or a different node) is designated as the preferred node, such that the node is designated *from the primary node or the secondary node*, as I explain below.

153. Chikkanayakanahally discloses that the "backup preference" may "specify[] a preferred node on which backups should be performed." Chikkanayakanahally, 10:41-45. Chikkanayakanahally also discloses that the "backup preference" specifies preferences for backups from the "primary replica" or "secondary replica" of a database. Chikkanayakanahally, 12:10-17. Additionally, Chikkanayakanahally's claim 1 recites a "backup preference for the first availability group" that "specif[ies]" "backing up on a node having a primary

replica type” or “backing up on a node having a secondary replica type.”

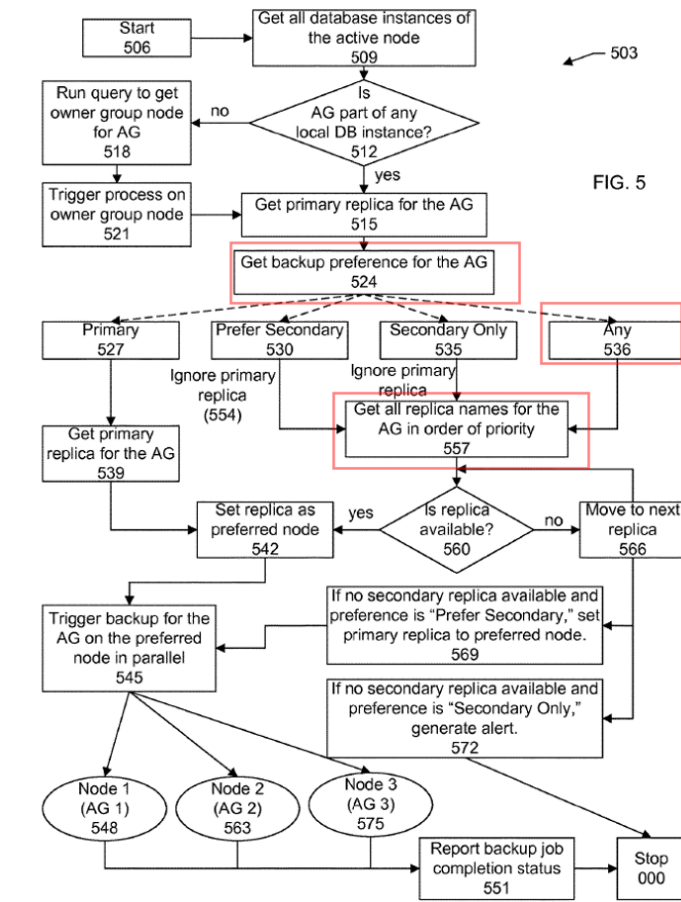
Chikkanayakanahally, 18:53-60; *see also* Chikkanayakanahally, 12:24-29

(explaining that in the example of a “primary replica” preference, “a node of the cluster hosting the primary replica... is designated or identified as being the preferred node for the backup”). Thus, POSAs understood that the “backup preference” may designate a particular node from which to take the backup that can be either a node having a primary replica or a node having a secondary replica; in other words, *from the primary node or the secondary node*.

154. The “backup preference” may also specify a preference or requirement for types of nodes without specifying a specific node. For example, the backup preference may be for “any type of replica.” Chikkanayakanahally, 12:16-17, Fig. 5 (annotated below). In this case, all of the replicas are placed in a “priority order”; the list is then traversed to find the first replica that is available, and the backup is taken from that replica’s node, *e.g.*, “[i]f secondary replica B is available the replica is set as the preferred node for backup.”

Chikkanayakanahally, 12:52-13:24, Fig. 5. In this case, the node is designated *from the primary node or the secondary node* because the list of possible nodes for designation includes all nodes hosting either primary or secondary replicas. The ’340 patent similarly describes choosing the initial node using an “Any’ EA policy” that selects from a list of all nodes. EX1001, 33:16-21 (“If the switch

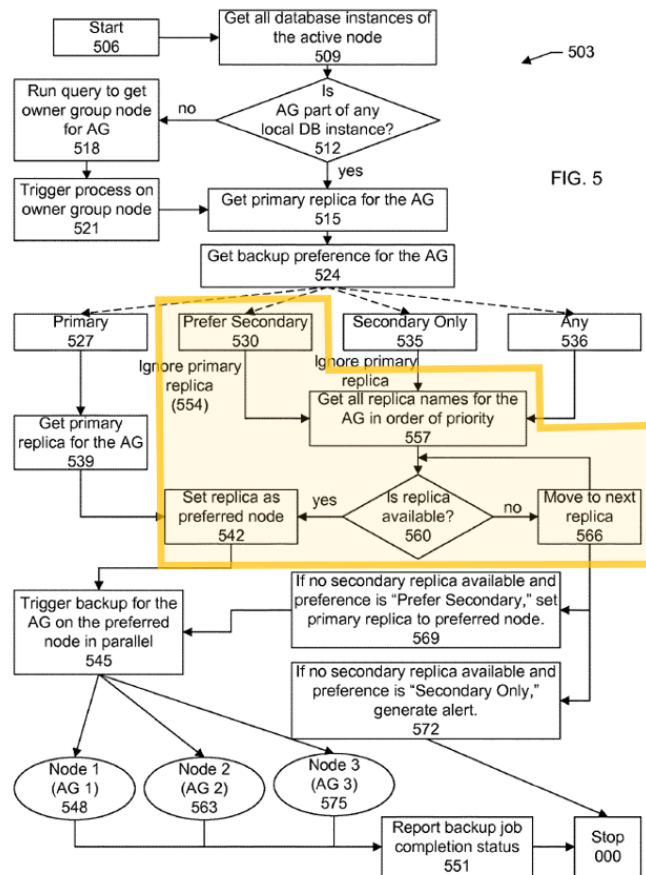
policy is the ‘Any’ EA policy, the database controller 205 may select any one node from the nodes of the database cluster 405 as the initial EA node. The database controller 205 may consider any priorities assigned to the nodes in designating the EA node such that the highest priority node is the initial EA node.”).



Chikkanayakanahally, Fig. 5 (annotated)

155. Chikkanayakanahally also specifies that the “backup preference” may be a “Prefer Secondary” policy; in this case, the node for one of the secondary replicas will be set as the preferred node for backup unless none of the secondary

replicas are available, in which case the node for the primary replica will be set as the preferred node. Chikkanayakanahally, 12:52-62, 13:9-24; 12:10-17, Fig. 5 (annotated below). In this case, as in the “any” case discussed above, the node is designated *from the primary node or the secondary node* because the list of possible nodes for designation includes all nodes hosting either secondary or primary replicas (with the secondary replicas being preferred). The ’340 patent likewise describes a “prefer-secondary EA policy” that chooses the highest-priority standby/secondary node. EX1001, 33:9-21.



Chikkanayakanahally, Fig. 5 (annotated)

156. It would also have been obvious to implement Chikkanayakanahally such that a record was created for the future of which node was originally chosen. As I noted above, Chikkanayakanahally discloses placing the potential replicas, and by extension the nodes on which the replicas reside, in “priority order” and then taking backups from the first node in the list that is available. POSAs would have found it obvious to simply place the previously-selected node at the top of the priority order and keep it there until such time as it becomes unavailable; this beneficially reduces the need to re-traverse the list the next time a backup is to be taken, and required simple programming techniques that were within a POSA’s skill.

g. [1D] “capture one or more snapshots of the source database or the copy of the source database from the first node designated as the initial active node”

157. As I discussed in Section V.B.1.f above for [1C], the node designated as the “preferred node” in Chikkanayakanahally is a claimed first node designated as the initial active node. The “preferred node” is a node hosting either a primary replica containing a primary database (source database) or a secondary replica containing a secondary database (a copy of the source database), as I explained in Section V.B.1.c for [1B1].

158. Chikkanayakanahally discloses that the “preferred node” is the node from which backups of the database are taken. Chikkanayakanahally, 6:56-61; *see also* Section V.A above. As I explain below, POSAs understood Chikkanayakanahally to disclose that the “backups” are performed by *capturing snapshots* of the database *from* the designated node.

159. Chikkanayakanahally discloses that the “client node[s]... of a high-availability cluster of nodes,” *i.e.*, the nodes on which replicas are hosted, “hav[e] a shadow copy service” that “facilitates the creation of a *snapshot* of data... upon a request... from a requestor such as a backup application client module.” Chikkanayakanahally, 8:18-32.

160. Chikkanayakanahally also describes the “shadow copy service” as the service used to perform backups on the “preferred node,” and expressly notes that as part of this process, “[t]he shadow copy service... prepare[s] the databases residing on the preferred node for the **backup** operation of that database instance hosted by the preferred node.” Chikkanayakanahally, 12:30-40, 11:52-63.

161. POSAs thus understood that backups of the database are created by using the shadow copy service on the preferred node to create snapshots of the database replica on the preferred node. This is consistent with a POSA’s background understanding that taking *snapshots* was a customary way to perform database backup. My testimony regarding a POSA’s background knowledge is

corroborated by, *e.g.*: Tekade (EX1017) [0046] (mentioning use of “snapshots” for “backup”); EX1014, [0002] (explaining that “Database snapshots are a feature of some database management systems... and allow a user to specify a point in time at which to preserve,” *i.e.*, back up, “the contents of the database”); Ohno (EX1015), 1:54-61 (“In order to restore the contents of the database up to a point in time when the data has consistency, a snapshot must be acquired in advance in a state in which data has consistency.... Snapshot is to maintain data to be snapshot at a certain point in time in an external storage, and it is a function provided by most storage systems.”); LVM-Howto (EX1022), 1 (discussing how to back up a database using snapshots). I understand that EX1022 was downloaded from the URL

https://web.archive.org/web/20070120200004/https://tldp.org/HOWTO/LVM-HOWTO/snapshots_backup.html. Based on my familiarity with the Wayback Machine as detailed above in paragraph 48, I understand the URL from which EX1022 was downloaded is the Internet Archive’s record of the website at the URL https://tldp.org/HOWTO/LVM-HOWTO/snapshots_backup.html archived on January 20, 2007.

162. POSAs understood that the above-discussed “shadow copy service” is software, *i.e.*, *computer-readable instructions stored on a memory*, that is *executed* by a *processor*, such as the processor on the node. *See* my discussion of software

in paragraph 117 above (citing Hansen (EX1116), 4:39-42, 5:46-58; Hughes (EX1117), 11:63-66; Grucci (EX1118), 4:43-46).

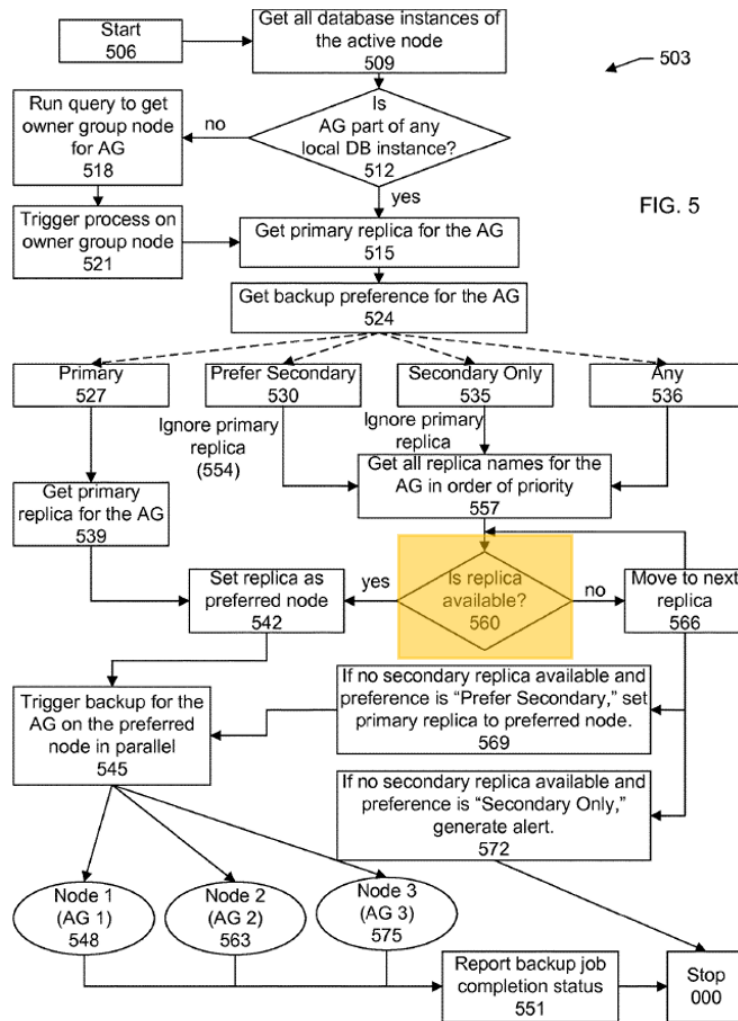
163. Furthermore, Chikkanayakanahally meets [1D] even if the claim requires that the same *processor* that performs the other steps of claim 1 *capture* the snapshots. Chikkanayakanahally discloses that the “shadow copy service” running on each node (*see* Chikkanayakanahally, 8:18-21, Fig. 3) runs in response to being “notif[ied]” to create the snapshot. Chikkanayakanahally, 12:30-40. In the obvious implementation of Chikkanayakanahally I discussed in Section V.B.1.b above for [1A], where the “backup server” performs the steps of the backup process, this notification originates from the backup server. *See also* Chikkanayakanahally, 11:20-23 (“The backup server, in turn, may generate a backup job request that the backup server sends to the preferred node. In another specific embodiment, the active (or coordinating) node may send the backup job request to the preferred node.”).

164. The ’340 patent similarly describes an embodiment where snapshots are captured as a result of a “database controller” sending “work requests... to capture snapshots” (also referred to as “snapshot requests”) to the designated nodes from which snapshot are to be taken. EX1001, 36:42-55, 38:66-39:5.

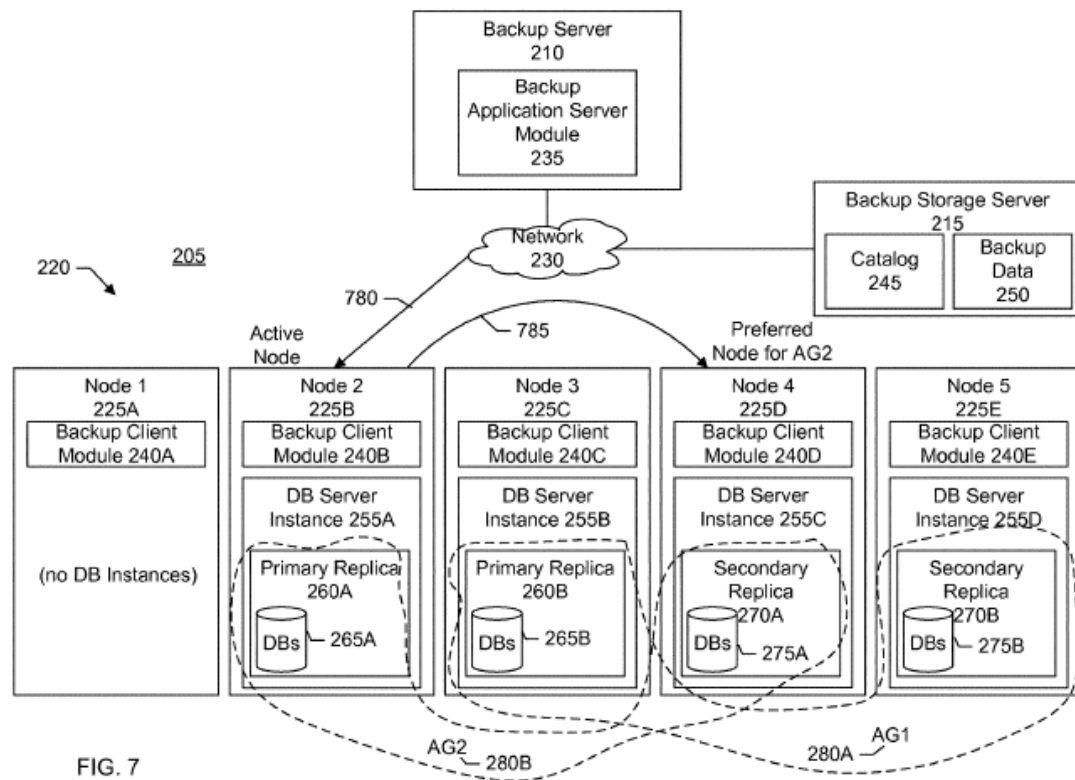
h. [1E] “determine that the first node has become unavailable”

165. As I explain below, Chikkanayakanahally discloses that when performing a backup, the backup algorithm may *determine* that the *first node, i.e.,* the node that was initially chosen for backups, *has become unavailable*.

166. Chikkanayakanahally’s Figure 5 (annotated below) shows a “detailed flow” of an algorithm for “identifying and backing up on a preferred *node*.” Chikkanayakanahally, 11:52-54. Chikkanayakanahally uses the terms “node” and “replica” interchangeably in Figure 5 and its corresponding description; POSAs understood that this is because there is a one-to-one correspondence between Chikkanayakanahally’s replicas and nodes. *See* Chikkanayakanahally, Fig. 5 (elements 542 and 569 instructing to “set” a “replica” as the “preferred node”), 13:13-19 (referring to setting “replica[s]” as “preferred node[s]”), Fig. 7 (showing one replica per node below).



Chikkanayakanahally, Fig. 5 (annotated)



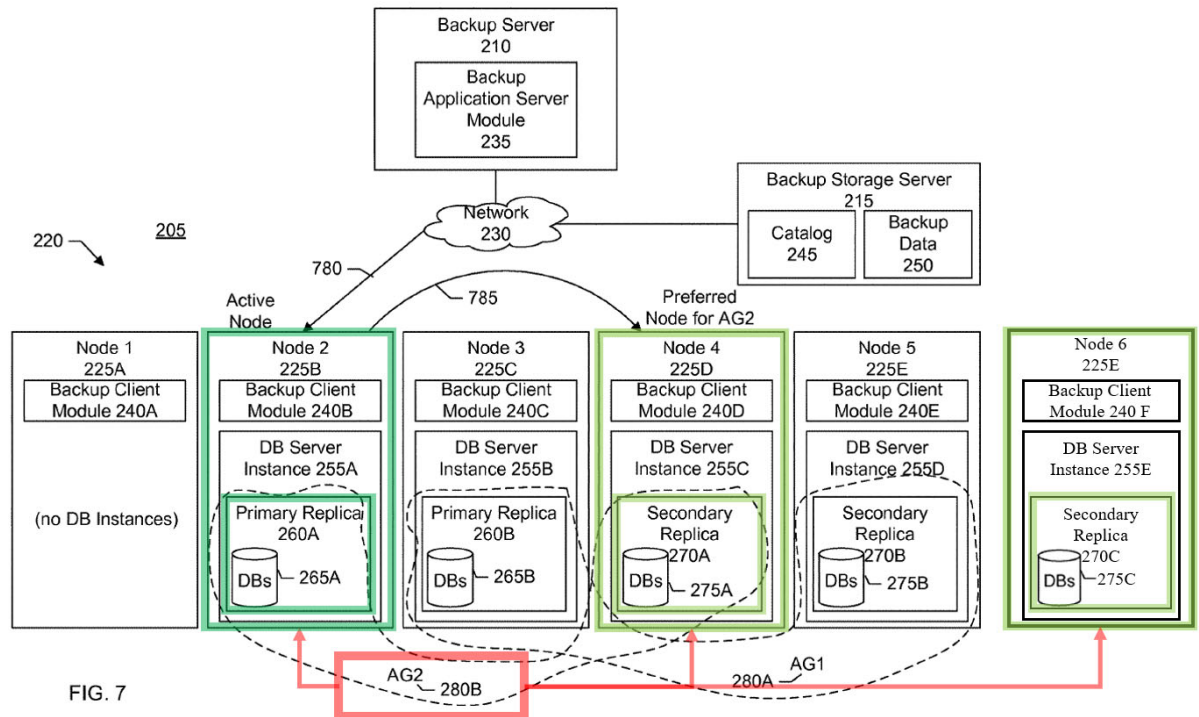
Chikkanayakanahally, Fig. 7

167. As shown in Figure 5, at step 560, the algorithm *determines* if the replica being considered for backup is available (“yes” branch) or *unavailable* (step 560, “no” branch), and thus also determines that the corresponding *node* is *unavailable*. Chikkanayakanahally, 13:66-14:3 (explaining that replica may be “unavailable” because the “node hosting the replica” has “suffered a failure” or has been “configured” to be “unavailable”).

168. As I discussed in Section V.B.1.f for [1C], in the example shown in Figure 7 (annotated below), Chikkanayakanahally’s “backup server” designates

Node 4, which hosts a “secondary replica” for AG2, as the first node.

Chikkanayakanahally discloses that the backup preference may be a “preference for the secondary replica (530)” or a requirement for “secondary replicas only (535).” Chikkanayakanahally, 12:5-53, 13:47-48, Fig. 5.



Chikkanayakanahally, Fig. 7 (modified)

169. In either case, Chikkanayakanahally’s algorithm puts the nodes/replicas in an “order of priority,” traverses the list to find the highest-priority one that is available, and then sets the available node as the “preferred node” for performing backups. Chikkanayakanahally, 12:55-57, 13:9-19, Fig. 5. Thus, POSAs understood that, if Node 4 has become unavailable, then at step 560, Chikkanayakanahally’s Figure 5 algorithm will (i) *determine that the first node*

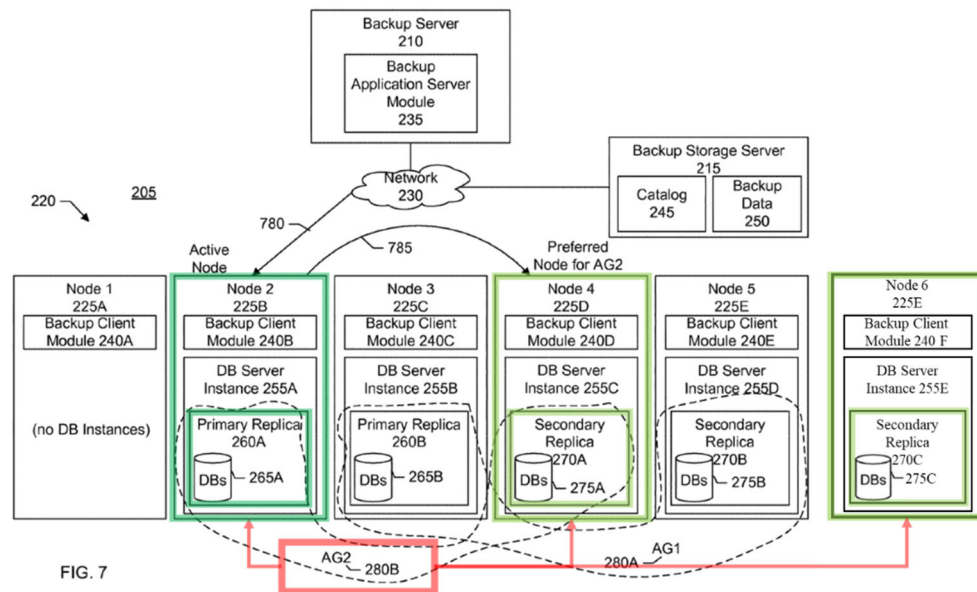
(Node 4) *has become unavailable* and, in response, (ii) move on to the next node in the list (e.g., Node 6 in the modified version of Figure 7 shown above) and set it as the “preferred node” if it is available.

170. POSAs understood Chikkanayakanahally to disclose that a *processor executing computer-readable instructions stored on a memory* in Chikkanayakanahally’s system makes this determination because a processor performs the task of selecting a preferred node, as I discussed in Section V.B.1.b above for [1A]. If claim 1 requires that a single *processor* performs all steps, this is met by the obvious implementation of Chikkanayakanahally in which the *processor* in the “backup server” performs all steps of the claim, as also I also discussed in Section V.B.1.b for [1A].

- i. **[1F] “select, in response to the first node becoming unavailable and based on a user defined switch over policy, a second node from the primary node or the secondary node as a next active node, wherein the switch over policy defines a criteria for selecting the next active node from the plurality of nodes to capture the one or more snapshots from; and”**

171. As I discussed in Section V.B.1.h for [1E], Chikkanayakanahally (1) determines that the *first node is unavailable*, and then (2) *in response, selects a different node as the “preferred node” from which to take backups*. See, e.g., Chikkanayakanahally, 12:55-57, 13:9-19, Fig. 5, which I discussed in paragraph 169 above. POSAs understood that selecting a different node results in *selecting a*

second node that is different from the first node. For example, in the 3-node availability group AG2 illustrated in the extended version of Figure 7 shown below, if Node 4 that was selected as the *first node* (see my discussion of [1B1] in Section V.B.1.c) becomes unavailable, then Node 6 becomes the new “preferred node.” Chikkanayakanahally, 11:9-13, Fig. 5.



Chikkanayakanahally, Figure 7 (extended and annotated)

172. The *second node* is selected from the *primary node* or the *secondary node*, because the second node is selected based on the “backup preference,” which permits the *second node* to be either a *primary node* or a *secondary node*.

Chikkanayakanahally, 10:41-45 (backup preference “specif[ies] a preferred node on which backups should be performed”), 12:10-17 (possible preferences include “preference for backups on the primary replica [*i.e.*, node]..., a preference for

backups on the secondary replica..., or a preference for backups on any type of replica”)

173. As I discussed in Section V.B.1.f for [1C], the node from which backups are taken is the claimed *active node*. And as I discussed in Section V.B.1.g for [1D], in Chikkanayakanahally, backups are taken by *capturing snapshots*. Thus, Chikkanayakanahally discloses instructions to *select a second node from the primary node or the secondary node as a next active node, in response to the first node becoming unavailable*, where the active node is the node *to capture the one or more snapshots from*, as claimed.

174. As I explain in Paragraphs 175-180 below, this selection is also based on a user defined switch over policy as claimed.

175. The '340 patent's specification refers to an “Era Active node (also referred to... as an EA node or *active node*)” as the “node” from which to “capture backups.” EX1001, 25:56-60.

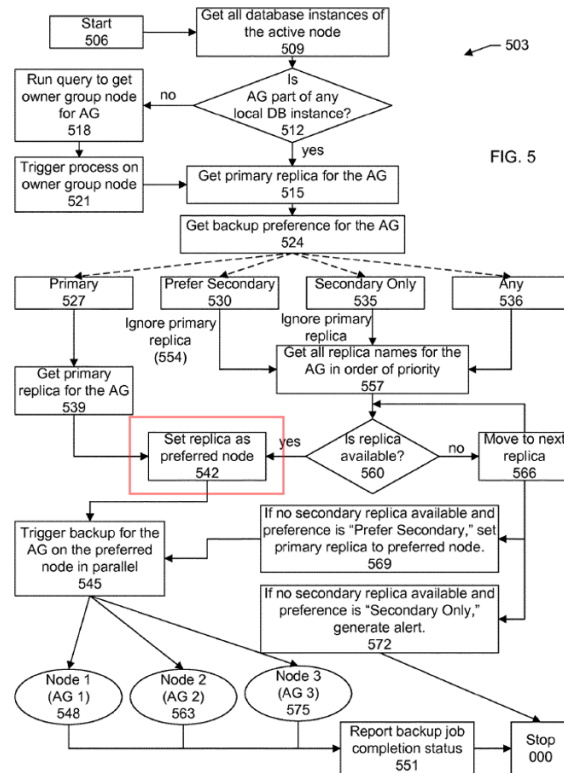
176. The specification provides examples of “switch over polic[ies]” that can be used to select a new EA node in the event of failure of one EA node, including “primary-only,” “secondary-only,” “prefer-primary,” “prefer-secondary,” and “[a]ny.” EX1001, 26:64-27:6. POSAs understood that these policies each defined “*criteria*” for selecting the next EA node, *i.e.*, the next *active node*, such as requiring that the next node be a primary node, a secondary node, or any type of

node. EX1001, 35:14-15 (referring to “criteria in the switch over policy portion of EA policy for designating EA nodes”).

177. Chikkanayakanahally discloses that the various options for the “backup preference” that selects the “preferred *node*” for the backup (Chikkanayakanahally, 10:24-40) *define* the same *criteria* as the exemplary policies in the '340 patent

...the system determines the backup preference for the availability group to be backed up. In a specific embodiment, the backup preference may specify a preference for backups on the primary replica [*i.e.*, *node*]... a preference for backups on the secondary replica... a preference (or requirement) for backups on the secondary replica only... or a preference for backups on any type of replica... e.g., no preference.

Chikkanayakanahally, 12:10-17, Fig. 5 (element 542: “Set replica as preferred node”) (annotated below).



Chikkanayakanahally, Fig. 5 (annotated)

178. Thus, Chikkanayakanahally’s “backup preference” is a *switch over policy defines a criteria* for selecting a *node* as recited.

179. Furthermore, the *node* that is selected by the backup preference is the *next active node* because, as discussed above, the selected node is the next node from which backups are taken by capturing snapshots. And the *node* is from the *plurality of nodes* because the universe of possible nodes is the *plurality of nodes* containing the primary and secondary nodes, as also discussed above.

180. Additionally, Chikkanayakanahally teaches that the “backup preference,” which is the *switch over policy* as discussed above, is *user defined*.

Chikkanayakanahally, 10:24-40 (explaining a “configuration” for a “backup”

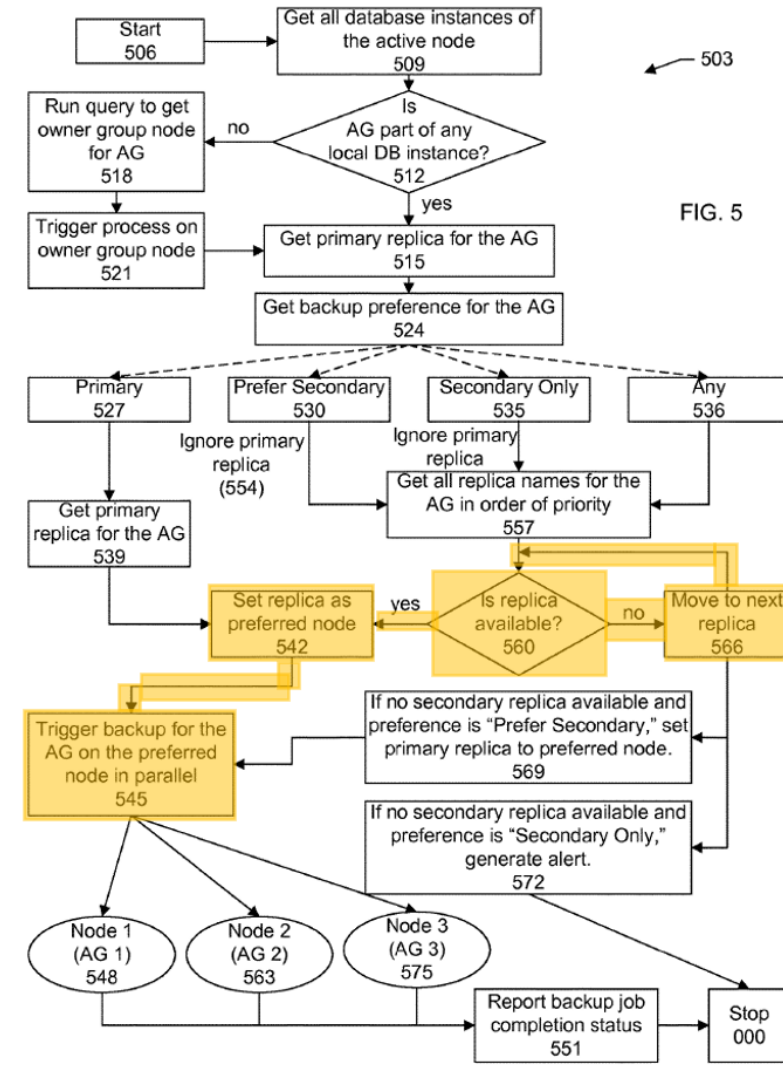
includes a “backup preference for [an] availability group” and “may be specified through a *user* interface”).

181. Finally, POSAs understood Chikkanayakanahally to disclose that a *processor* in Chikkanayakanahally’s system *selects* a new preferred node, for the reasons I discussed in Section V.B.1.b for [1A]. If the claim requires that a single *processor* performs all steps recited in the claim, this is met by the obvious implementation of Chikkanayakanahally in which the *processor* in the “backup server” performs all steps of the claim, as I also discussed in Section V.B.1.b for [1A].

- j. **[1G] “capture the one or more snapshots of the source database or the copy of the source database from the second node in response to selecting the second node as the next active node”**

182. As I discussed in Section V.B.1.i for [1F], the recited *selecting* of the *second node* occurs when the new “preferred node” is chosen for taking backups. Chikkanayakanahally, 12:55-57, 13:9-19, Fig. 5.

183. Chikkanayakanahally discloses that “backups” are “trigger[ed]” on the node once it is selected. Chikkanayakanahally, Abstract, 6:32-35, 6:61-63, 9:1-2, 11:14, 12:30-31, Figs. 4, 5 (annotated below).



Chikkanayakanahally, Fig. 5 (annotated)

184. Furthermore, the backups are *of the source database or the copy of the source database*, because the *second node* is a node that contains a replica of the source database. See Section V.B.1.c above, discussing [1B1]. Performing the backup involves *capturing the one or more snapshots* as I explained in Section V.B.1.g for [1D]. POSAs understood that a *processor executing computer-*

readable instructions stored on a memory captures the snapshots (whether or not the claim requires a single processor) for the reasons I explained in Section V.B.1.g for [1D].

185. Thus, for the reasons I discussed in paragraphs 182-184 above, Chikkanayakanahally meets [1G]. Chikkanayakanahally also meets [1G], and the rest of the claim, by capturing snapshots from **both** the *initial active node* (see [1D]) and the *next active node*, as I explain below.

186. POSAs understood that in Chikkanayakanahally's "system... for backing up and recovering high-availability databases" (Chikkanayakanahally, 5:17-28), backups are taken from databases multiple times, rather than just once. Chikkanayakanahally notes that "in a highly available environment," "backups" provide "benefits" such as "rollbacks to older **points** in time" and "archiving." Chikkanayakanahally, 1:28-32. POSAs understood that providing rollback to "points" (plural) in time involves taking backups at plural points in time.

187. Additionally, Chikkanayakanahally says the "backup preference" may include a "backup window." Chikkanayakanahally, 10:43-55. POSAs understood that the term "backup window" was customarily used in the art to denote a time for periodic backups. My testimony regarding a POSA's background knowledge is corroborated by, e.g., RDS (EX1013), 187 (discussing a "backup window" for taking "snapshot[s]"). I understand that EX1013 was downloaded from the URL

<https://web.archive.org/web/20160412105355/http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-ug.pdf>. Based on my familiarity with the Wayback Machine as detailed above in paragraph 48, I understand the URL from which EX1013 was downloaded is the Internet Archive’s record of the website at the URL <http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-ug.pdf> archived on April 12, 2016.

188. POSAs further understood that taking multiple backups, *e.g.*, on user-settable schedule, was customary and obvious. My testimony regarding a POSA’s background knowledge is corroborated, *e.g.*, RDS (EX1013), 16-17, 187 (daily “backup window”); Tekade (EX1017), [0046] (“The backup will run every 24 hours...”); Google (EX1032), 2-3; Zha (EX1006), 11:33-43 (discussing “schedule” for “retriev[ing] PIT [point-in-time] copies of databases”); Abastha-634 (EX1021), 1:9-14. I understand that EX1032 was downloaded from the URL <https://www.archive.org/web/20180918103721/https://cloud.google.com/sql/docs/mysql/create-instance>. Based on my familiarity with the Wayback Machine as detailed above in paragraph 48, I understand the URL from which EX1032 was downloaded is the Internet Archive’s record of the website at the URL <https://www.archive.org/web/20180918103721/https://cloud.google.com/sql/docs/mysql/create-instance> archived on September 18, 2018.

189. Chikkanayakanahally teaches “a system... for backing up and recovering high-availability databases” (Chikkanayakanahally, 5:17-28), and implementing this system to repeat the backup operation, *e.g.* at a user-defined frequency, would have required no more than simple programming techniques within a POSA’s skill.

190. POSAs also understood that periodic backups would have enabled the above-discussed “benefits” Chikkanayakanahally discloses, such as the ability to roll back to multiple points in time. Chikkanayakanahally, 1:28-32.

191. Because Chikkanayakanahally’s system takes periodic backups, POSAs understood that, for example, in the in the 3-node availability group AG2 illustrated in the extended version of Figure 7 shown below, Chikkanayakanahally’s system includes at least one *processor executing instructions* as recited in [1A] (*see* Section V.B.1.b above) to perform the following steps: (1) at the time a first backup is to be taken, designate Node 4 as the *initial active node*; (2) take the first backup by *capturing a snapshot* from node 4; (3) at the time a later backup is to be taken, *determine* that Node 4 has become *unavailable* and *select* Node 6 as the *next active node*; and (4) take the later backup by *capturing a snapshot* from Node 6.

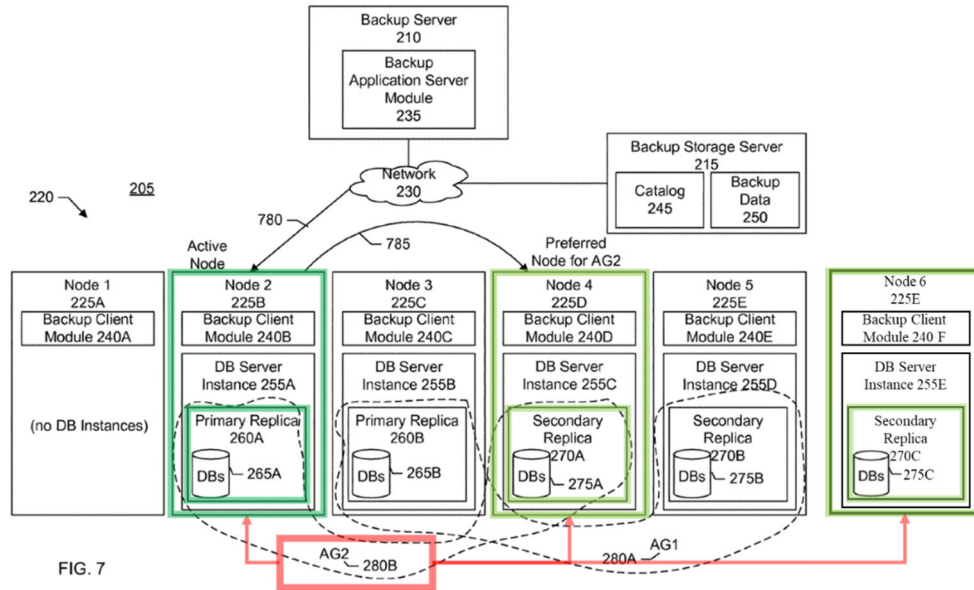


FIG. 7

Chikkanayakanahally, Figure 7 (extended and annotated)

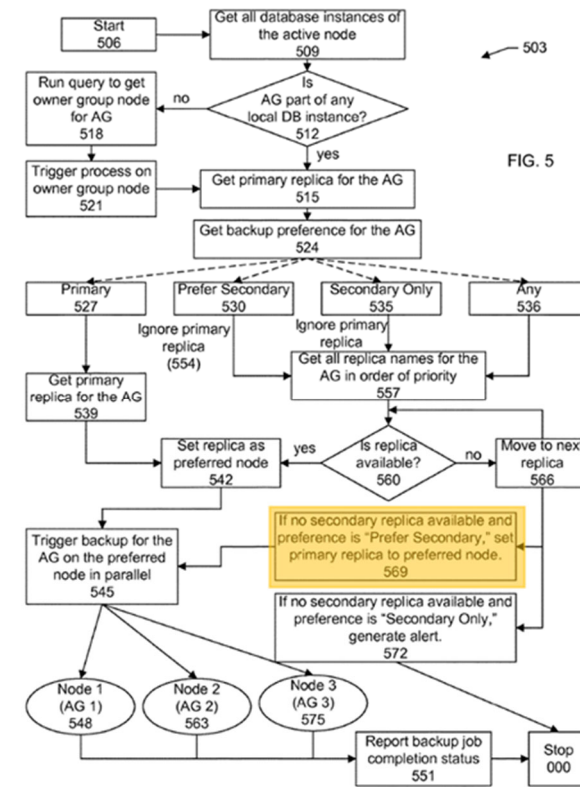
2. **Claim 2: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node.”**

192. Chikkanayakanahally meets claim 2 in alternative ways, as I show below.

193. **First**, as I further explain below, Chikkanayakanahally discloses that, where the backup preference is a “preference for the secondary replica (530)” or “any type of replica (536)”, the primary replica may be chosen if no secondaries are available. Chikkanayakanahally, 12:52-62, 13:24-46, 12:10-17, 14:4-12, Fig. 5.

194. Chikkanayakanahally discloses that, where the backup preference is a “**preference** for the secondary replica” rather than a **requirement** for backups only on the secondary replica, the primary replica may be chosen if no secondaries are

available. Chikkanayakanahally, 12:52-62 (disclosing that “if the backup preference specifies a preference for the secondary replica (530), the primary replica is... temporarily ignored as a candidate for backup [until being considered at step 569]” and then, initially at step 557, “the identified replicas include a list of one or more **secondary** replicas. The list may order or rank the secondary replicas in an order that indicates the backup priority.”), 13:24-34 (“In step 560, if secondary replica B is unavailable, the system loops back to examine the next replica in the backup priority list (step 566)... If the replica is unavailable, the system continues to loop through and examine the next replica in the priority list.”), 13:34-46 (disclosing that, at “step 569,” “if there are no secondary replicas available or online for backup and the backup preference specifies a ‘preference’ for backups on secondary replicas, the system **sets the primary replica to the preferred node,**” and that “**a node of the cluster hosting the primary replica of the availability group to be backed up is designated the preferred node** for the backup operation because all secondary replicas were unavailable and the backup preference specified a ‘preference’ (**rather than a requirement**) for backups on secondary replicas.”), Fig. 5 (annotated below, showing that where the backup preference being a “preference for the secondary replica (530)”, the system may go through the steps 557, 560, and 566, and then choose a primary node as the preferred node at step 569, as highlighted in Fig. 5).



Chikkanayakanahally, Fig. 5 (annotated)

195. Additionally, Chikkanayakanahally discloses an “any” replica preference, which may also choose the primary replica. Chikkanayakanahally, 12:10-17 (“In a step 524, the system determines the backup preference for the availability group to be backed up. In a specific embodiment, the backup preference may specify... a preference for backups on any type of replica (536)....”), 14:4-12 (“Alternatively, the backup preference may specify backups on **any** replica (536). The flow proceeds to step 557 and the steps following as shown in FIG. 5 and discussed above. For example, the backup priority list may specify a backup priority order as secondary replica B, secondary replica A, secondary

replica C, secondary replica D, and primary replica. The system traverses the list to examine in priority order which replica is available or online for backup and the flow proceeds as shown in FIG. 5 and discussed above.”).

196. In either case I explained above, the “preference” is a *switch over policy* that *defines selection of the next active node from the plurality of nodes*, where the next active node is *the primary node*.

197. **Second**, the ’340 patent discloses that if the primary node fails, a “database controller” may first designate a secondary node to become the new primary, and then “designate that new primary node as the next EA node.”

EX1001, 27:15-29 (“**when that primary node fails, the database controller designates another node (e.g., one of the standby [a.k.a. ‘secondary’] nodes 415 or 420) as the new primary node. The database controller 205 may designate that new primary node as the next EA node** and the highly available time machine 400 may automatically switch over to capturing backups from the newly designated primary node. Thus, the highly available time machine 400 may wait for a new node to be designated as the primary node for capturing snapshots and transactional logs....”).

198. Chikkanayakanahally likewise discloses that a secondary node can become a primary node following the primary node’s unavailability, as I discussed in Section V.B.1.e for [1B3]. Chikkanayakanahally also discloses that the “backup

preference,” *i.e.* the switch over policy as I discussed above for [1F] (Section V.B.1.i), can be for taking “backups on the primary replica.”

Chikkanayakanahally, 12:10-17 (“In a step 524, the system determines the backup preference for the availability group to be backed up....”).

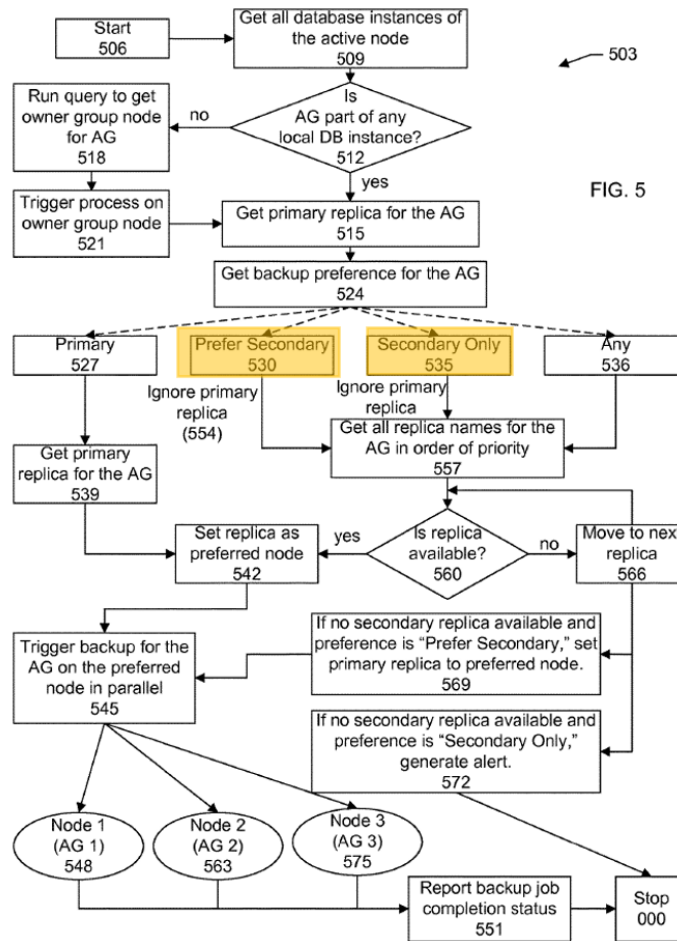
199. POSAs would have found it obvious to implement Chikkanayakanahally’s primary-replica “backup preference” such that, if the primary node hosting the primary replica is unavailable, a secondary replica is converted to a primary replica, making its hosting node the primary node, and then backups are taken from that node, just as the ’340 patent describes. This would beneficially enable honoring the preference for primaries, and it would have been within a POSA’s skill to implement because Chikkanayakanahally discloses turning a secondary into a primary (*see* Section V.A above).

200. In this circumstance, the new primary node *is the next active node* from which backups are taken. *See* my discussion of [1C] in Section V.B.1.f above. The new node is *from the plurality of nodes* because it is one of the nodes in the availability group, as I discussed above for [1B1] (Section V.B.1.c).

3. Claim 3: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node”

201. As I discussed in Section V.B.2 for claim 2, Chikkanayakanahally’s “backup preference” is a *switch over policy* and *defines selection of the next active node*.

202. Chikkanayakanahally discloses that the “backup preference” may either be a preference or requirement for secondary replicas. Chikkanayakanahally, 12:10-17 (“In a step 524, the system determines the backup preference for the availability group to be backed up. In a specific embodiment, the backup preference may specify... a preference for backups on the secondary replica (530), a preference (or requirement) for backups on the secondary replica only (533)...”), Fig. 5 (annotated below).



Chikkanayakanahally, Fig. 5 (annotated)

203. This preference *defines selection of the next active node from the plurality of nodes that is the secondary node.*

4. **Claim 4: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node, or upon determining that none of the plurality of nodes are the secondary node, select the next active node from the plurality of nodes that is the primary node”**

204. The first clause of claim 4 before “or”, i.e., “*wherein the switch over policy defines selection of the next active node from the plurality of nodes that is*

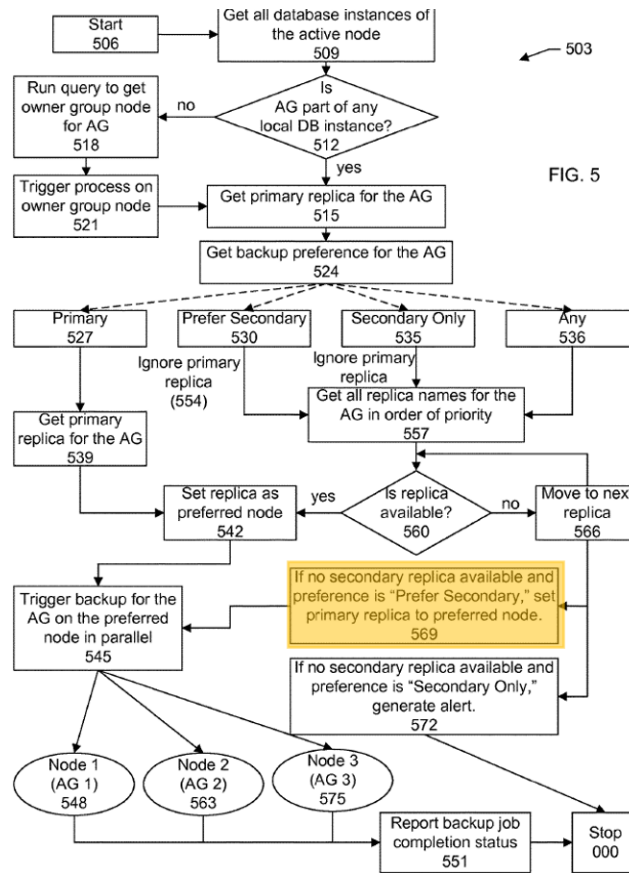
the secondary node,” is identical to claim 3. Chikkanayakanahally’s secondary-replica “backup preference” meets claim 3’s switch over policy for the reasons I discussed in Section V.B.3 for claim 3. If satisfying the first clause alone is sufficient to meet claim 4, then Chikkanayakanahally meets claim 4.

205. Chikkanayakanahally also meets claim 4 if the second clause after “*or,*” reciting that “*none of the plurality of nodes are the secondary node*” (contradicting [1B1’s] recitation of at least one secondary node), is required, as I explain below.

206. The ’340 patent’s specification describes a “prefer-secondary EA policy” that selects a primary node only if all secondaries are unavailable. EX1001, 28:1-10 (“With a prefer-secondary EA policy, the user may define that the database controller 205 may pick any of the available nodes as the next EA node. However, if a standby [a.k.a. ‘secondary’] node (*e.g.*, the standby nodes 415, 420) is available, the database controller 205 is to pick the standby node as the next EA node. Thus, as long as either a primary node or a secondary node is available in the database cluster 405, the highly available time machine 400 may continue to capture snapshots and transactional logs similar to the prefer-primary EA policy.”).

207. Chikkanayakanahally likewise discloses that its “backup preference” that prefers (as opposed to requires) secondary replicas selects a primary if it *determines* no secondaries are available. Chikkanayakanahally, 12:52-62 (“in a

step 554, if the backup preference specifies a preference for the secondary replica (530), the primary replica is... temporarily ignored as a candidate for backup [until being considered at step 569]....”), 13:34-46 (“In a step 569, **if there are no secondary replicas available or online for backup and the backup preference specifies a ‘preference’ for backups on secondary replicas**, the system **sets the primary replica to the preferred node**. In other words, **a node** of the cluster **hosting the primary replica** of the availability group to be backed up **is designated the preferred node** for the backup operation because all secondary replicas were unavailable and the backup preference specified a **‘preference’ (rather than a requirement)** for backups on secondary replicas.”), Fig. 5 (annotated below). By selecting a primary replica/node (Chikkanayakanahally, 13:34-46), this *switch over policy selects the next active node from the plurality of nodes that is the primary node*, at least because the universe of possible nodes is *the plurality of nodes* containing the primary and secondary nodes (name, the nodes in the availability group), as I discussed above for [1B1] (Section V.B.1.c) and [1F] (Section V.B.1.i).



Chikkanayakanahally, Fig. 5 (annotated)

208. Thus, Chikkanayakanahally also meets the second clause of claim 4 after “or,” reciting that “none of the plurality of nodes are the secondary node” and “select[ing] the next active node from the plurality of nodes that is the primary node.”

5. **Claim 5: “The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node, or upon determining that none of the plurality of nodes is the**

primary node, select the next active node from the plurality of nodes that is the secondary node”

209. The first clause of claim 5 before “or,” i.e., “*wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node,*” is identical to claim 2. Chikkanayakanahally’s primary replica “backup preference” meets claim 2’s switch over policy for the reasons I discussed in Section V.B.2 for claim 2. If satisfying the first clause alone is sufficient to meet claim 4, then Chikkanayakanahally meets claim 5.

210. Chikkanayakanahally also meets claim 5 if the second clause after “or,” reciting “none of the plurality of nodes is the primary node” (contradicting [1B1’s] recitation of a primary node), is required, as I explain below.

211. The ’340 patent’s specification describes a “prefer-primary EA policy” that picks a secondary node only if the primary is unavailable. EX1001, 27:50-67 (“with the prefer-primary EA policy, the user may define that the database controller 205 may pick any of the available nodes as the next EA node. However, **if a primary node** (e.g., the primary node 410) **is available, the database controller 205 is to pick the primary node** as the next EA node. Thus, in such cases, **if no primary node is available** in the database cluster 405 to capture backups from, **the database controller 205 may designate a standby [a.k.a. ‘secondary’] node** (e.g., the standby nodes 415 or 420) as the next EA node.”).

212. Chikkanayakanahally likewise discloses a “backup preference” for “backups on the primary replica.” Chikkanayakanahally, 12:10-17 (“In a step 524.... the backup preference may specify a preference for backups on the primary replica (527)....”). POSAs understood that this preference selects a secondary node *upon determining* the primary node is unavailable, because Chikkanayakanahally analogously discloses that the “preference” for secondary replicas selects a primary if no secondaries are available, as discussed I discussed in Section V.B.2 for claim 2. At minimum, POSAs would have found this implementation obvious; such an implementation of the “primary” preference beneficially provided a fallback option for database backup if the primary is unavailable, and was within a POSA’s skill to implement because Chikkanayakanahally already teaches taking backups from different types of nodes.

213. In such an implementation, if a secondary replica is selected, resulting in its hosting node becoming the preferred node (as I discussed in Section V.B.1.f for [1C], discussing, *e.g.*, Chikkanayakanahally, 18:53-60 (claim 1), 12:24-29), then this the next active node that is the secondary node is selected from the plurality of nodes, at least because the universe of possible nodes is *the plurality of nodes* containing the primary and secondary nodes (namely, the nodes of the

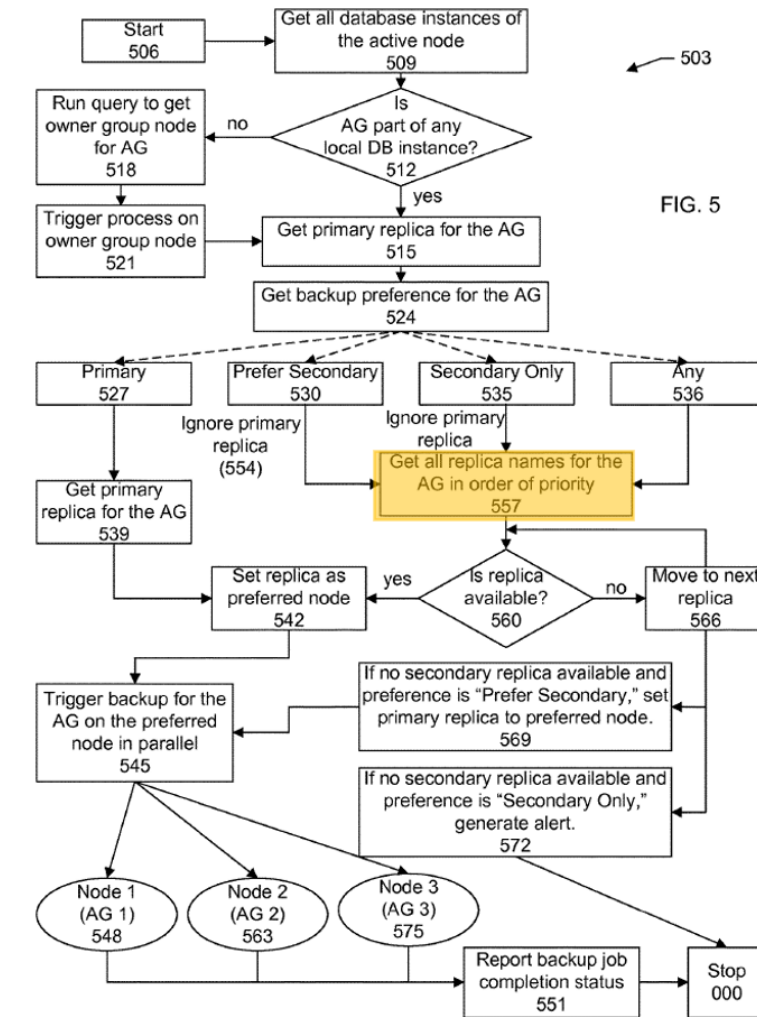
availability group), as I discussed above for [1B1] (Section V.B.1.c) and [1F] (Section V.B.1.i).

214. Thus, Chikkanayakanahally also meets the second clause of claim 5 after “or,” reciting that “none of the plurality of nodes is the primary node” and “select[ing] the next active node from the plurality of nodes that is the secondary node.”

6. Claim 6: “The system of claim 1, wherein the switch over policy defines a priority assigned to at least a subset of the plurality of nodes and wherein the processor further executes the computer-readable instructions to select the next active node based upon the priority”

215. As I discussed in Section V.B.1.i for [1F], the “backup preference” that meets claim 1’s *switch over policy* places *at least a subset of the plurality of nodes* (e.g., the secondary nodes, if the preference is “prefer secondary”) in “**priority** order”; in other words, it *defines a priority assigned to at least a subset of the plurality of nodes*. Chikkanayakanahally, 12:52-13:34 (describing the example of backup preference being a preference for the secondary replica, at step 557, “[t]he list [of identified secondary replicas] may order or rank the secondary replicas in an order that indicates the **backup priority**.... The secondary replicas may be sorted in backup **priority order**.”), 14:4-12 (describing the example where the backup preference is “any” replica, at step 557, “the backup priority list may specify a backup priority order as secondary replica B, secondary replica A,

secondary replica C, secondary replica D, and primary replica. The system traverses the list to examine in priority order which replica is available or online for backup”), Fig. 5 (annotated below).



Chikkanayakanahally, Fig. 5 (annotated)

216. For example, “if the backup preference specifies a preference for the secondary replica (530),” at step 557, “the system obtains or identifies all replica names for the availability group **in order of priority**.... In this specific embodiment, **the identified replicas include a list of one or more secondary**

replicas. The list may order or rank **the secondary replicas in an order** that indicates the **backup priority**.... The secondary replicas may be sorted in backup ***priority order.***” Chikkanayakanahally, 12:52-63. POSAs understood that, in this example of backup preference being “prefer secondary,” at step 557, the backup preference defines a “backup priority” assigned to the secondary replicas, and the secondary replicas “may be sorted in backup priority order.” Here, POSAs understood that the secondary replicas are a *subset* of the availability group’s plurality of replicas containing the primary and secondary replicas. POSAs also understood that, because choosing a particular replica for backup results in the node hosting that replica being chosen for backup (Chikkanayakanahally, 13:34-46, as I explained above for [1[C] (Section V.B.1.f)), placing the *at least a subset of the replicas* in “priority order” effectively places *at least a subset of the plurality of nodes* in “priority order,” since the recited plurality of nodes are the nodes hosting the replicas (as I explained above for [1B1] (Section V.B.1.c)).

217. As I discussed *supra* [1A] (Section V.B.1.b), the *processor* executes the *computer-readable instructions* for the tasks recited in [1B1]-[1G], including *selecting the next active node* based on the *switch over policy* as [1F] recites and thus *based upon the priority* defined by the policy. Thus, Chikkanayakanahally meets claim 6.

7. **Claim 19: “The system of claim 1, wherein the plurality of nodes are distributed between a first cluster and a second cluster”**

218. The '340 patent describes an exemplary “cluster” as a collection of nodes. EX1001, 6:63-64 (“The **cluster 100 includes a plurality of nodes**, such as a **first node 105**, a **second node 110**, and a **third node 115**”), Fig. 1 (below). Although the '340 patent does not expressly specify how to define a cluster’s boundaries, the patent says Figure 1’s exemplary “cluster” may be in a “data center,” with the cluster’s nodes connected to a common “storage pool.” EX1001, 7:23-37 (“The cluster 100[] includes and/or is associated with a **storage pool 170**.... includ[ing]... direct-attached storage 180A, 180B, and 180C. ... [T]he direct-attached storage 180A, 180B, and 180C includes storage components that are provided internally within each of the first node 105, the second node 110, and the third node 115, respectively....”), 8:4-7 (“In some embodiments, the cluster 100 may be part of a **data center**. Further, one or more of the first node 105, the second node 110, and the third node 115 may be organized in a variety of network topologies”).

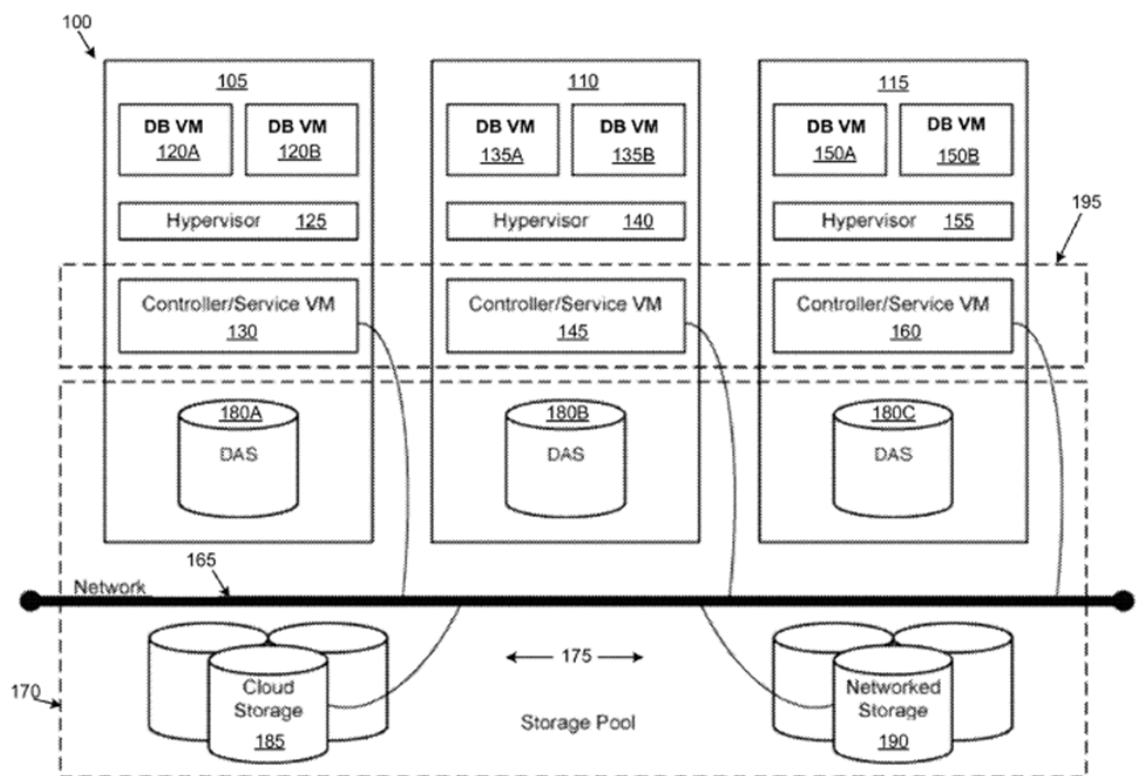


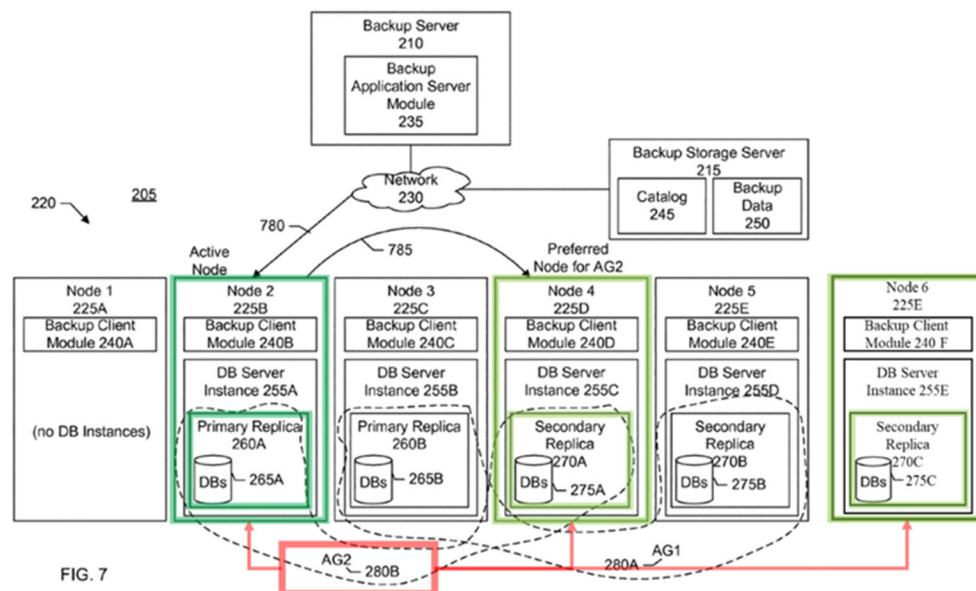
FIG. 1

219. Chikkanayakanahally describes the collection of nodes for an availability group as a “high-availability cluster.” *E.g.*, Chikkanayakanahally, Abstract (“A high-availability cluster includes a subset of nodes...”), 9:5-6 (“...cluster 220 includes a set of databases instances 255A-D”), 12:24-29 (“In a step 542, the replica (*e.g.*, primary replica) is set as the preferred node. In other words, a node of the cluster hosting the primary replica of the availability group to be backed up is designated or identified as being the preferred node for the backup operation.”).

220. However, Chikkanayakanahally does not define a “high-availability cluster” as nodes in one data center or connected to one storage pool, as in the ’340 patent’s example. Rather, Chikkanayakanahally defines a “cluster” functionally, as a “group[] of computers that support[s] server applications that can be reliably utilized with a minimum of down-time.” Chikkanayakanahally, 1:14-18 (“High-availability clusters (also known as HA clusters or fail over clusters) are groups of computers that support server applications that can be reliably utilized with a minimum of down-time.”).

221. Additionally, Chikkanayakanahally says a “Windows cluster” is an example of what Chikkanayakanahally calls a “cluster” (Chikkanayakanahally, 6:39-41), and POSAs understood that a Windows Cluster can comprise nodes geographically distributed at different data centers (a feature called “Geo-Clustering”). My testimony regarding a POSA’s background knowledge is corroborated by, *e.g.*, *Windows Geo-Clustering: SQL Server* (EX1121), 1 (describing that Geo-clustering addresses concerns such as risks of “severe catastrophic failures like the loss of an entire data center” by “having two or more cluster nodes... located in different geographical locations forming a single highly available system”, *e.g.*, with one node at the production data center and another node at a remote disaster recovery (DR) data center.).

222. Thus, POSAs understood that in the extended Chikkanayakanahally Figure 7 example (below), AG2's nodes can be distributed among two geographically dispersed data centers (e.g., Node 2 in one data center and Nodes 4 and 6 in another). In such an implementation, *the plurality of nodes are distributed between a first cluster and a second cluster*, because each data center is an example of what the '340 patent calls a "cluster." EX1001, 7:23-37 ("The cluster 100[] includes and/or is associated with a **storage pool** 170.... includ[ing] storage components that are provided internally within each of the first node 105, the second node 110, and the third node 115, respectively...."), 8:4-7 ("In some embodiments, the cluster 100 may be part of a **data center**. Further, one or more of the first node 105, the second node 110, and the third node 115 may be organized in a variety of network topologies").



Chikkanayakanahally, Figure 7 (extended and annotated)

223. Alternatively, distributing the nodes among what the '340 patent describes as *clusters* would have been obvious. It was well-known to POSAs to have a single application, *e.g.*, a replicated database, distributed among multiple '340 patent *clusters*. My testimony regarding a POSA's background knowledge is corroborated by, *e.g.*: Pahwa (EX1023), [0003] (disclosing "a multi-cluster service managing access to a set of destination clusters hosting a software application"), Fig. 1 (showing that EX1023 refers to different "clusters" as different geographic groupings of nodes); Bajaj (EX1011), 4:3-16 (describing that "a database node may be running one or more other databases", where the "one or more databases may include... one or more database nodes that are part of one or more database clusters"), 6:7-15 ("Database content associated with a database may be stored in one or more storage volumes 104", which may be "shared by a plurality of database nodes."), Fig. 1 (teaching database nodes distributed among "clusters" where nodes in a "cluster" are associated with common storage); *Windows Geo-Clustering* (EX1121), 1 (describing that Geo-clustering addresses concerns related to recovery time for "a business process or an application system" by "having two or more cluster nodes... located in different geographical locations forming a single highly available system", *e.g.*, with one node at the production data center and another node at a remote disaster recovery (DR) data center), 2 ("in geo-

clustering, the nodes can be in different geographical locations and are connected to different [data center's] storage subsystem. However, the storage subsystem used by one node is an exact replica of the storage subsystem being used by the other nodes.”).

224. POSAs would have been motivated to use a multi-*cluster* (of the type in the '340 patent's example) implementation because POSAs understood that distributing a replicated database across multiple such '340 patent *clusters* provided benefits such as increased failure protection; for instance, if the database were distributed between two clusters in two different physical locations, and one cluster failed due to a location-specific problem (*e.g.*, power failure), the other cluster would still be available. While such distribution in different physical locations could introduce some reasonable communication overhead, POSAs understood that there was a straightforward tradeoff between enhanced failure protection and the associated communication overhead, and understood that the benefits of enhanced failure protection often justified the overhead. My testimony regarding a POSA's background knowledge is corroborated by, *e.g.*, *Windows Geo-Clustering* (EX1121), 1 (discussing that “[w]hile local high availability would be sufficient for most mission-critical business applications, it does not address the risk of severe catastrophic failures like the loss of an entire data center or natural calamities like earthquakes and hurricanes”, and explaining that Geo-clustering

addresses such concerns, among other concerns such as recovery time, by “having two or more cluster nodes... located in different geographical locations forming a single highly available system”, *e.g.*, with one node at the production data center and another node at a remote disaster recovery (DR) data center.), 3 (recognizing communication overhead “when the cluster nodes are separated by geographical locations” as a known “consideration when working with geo-cluster”, *i.e.*, recognizing communication overhead as a known factor in the tradeoff to achieve enhanced failure protection using geo-cluster), 3-4 (disclosing known solutions implemented in geo-cluster environment to mitigate the communication overhead by “[a]djust[ing] cluster [heartbeat] parameters accordingly and monitor[ing] appropriately” where “heartbeat” is “commonly known”).

225. Additionally, POSAs understood a multi-*cluster* implementation would retain the benefits of Chikkanayakanahally’s teachings, including (1) allowing backups to proceed at the level of what Chikkanayakanahally calls a “high availability cluster” and (2) allowing backups on secondary nodes to avoid performance impacts on the primary. Chikkanayakanahally, 1:33-45 (“In some cases a user may desire to have backups proceed on one or more redundant computers in the high-availability cluster rather than on the server actively responding to client requests.”), 6:20-31 (explaining that a “problem with traditional backup applications [in] high-availability environment includes the

inability to have backup preferences” and for example, “a backup[] performed on the active node may.... lead to a severe degradation in the performance of the active node” because “computing resources that could otherwise be servicing client requests will be occupied by the backup operation”), 6:38-42 (“There is no need to configure backups against each node [of a cluster]”).

226. POSAs would reasonably have expected success with the above-described multi-cluster implementation for multiple reasons. First, such implementations were well-known as I noted above. Second, POSAs understood that Chikkanayakanahally’s teachings regarding replication and backup would work in a multi-cluster implementation. Chikkanayakanahally discloses that the various components of its exemplary system, such as the “nodes” and “backup server,” are connected by a “network,” *e.g.*, the “Internet.” Chikkanayakanahally, 3:8-24 (disclosing that the communication network shown in Fig. 1 may be, “in one embodiment, ... the Internet, in other embodiments, ... may be any suitable communication network....”), 5:17-23 (describing that the system “includes a backup server 210, a backup storage server 215, and a cluster 220 of backup client nodes such as first, second, third, fourth, and fifth nodes 225A-E, each of which are connected through a communication network [such as the] communication network shown in FIG. 1 and described above”), Figs. 1-2. POSAs understood that

a network like the Internet could be used for the necessary communications even in a multi-cluster implementation.

8. Claim 20: “The system of claim 1 wherein the switch over policy is defined during provisioning of the source database”

227. “Provision[ing]” in the ’340 patent refers to either “creat[ing]” a database or “register[ing]” a database with a “database management system.” EX1001, 11:58-62 (“For example, the database controller 205 is configured to provide copy data management services, or in other words, services to protect databases **provisioned** (e.g., **created or registered**) by the **database management system 200**.”).

228. The “backup preference,” which is claim 1’s *switch over policy*, is user-defined as part of a “configuration” of a backup. *See* Section V.B.1.i above for [1F]; Chikkanayakanahally, 10:24-35 (“The **configuration may specify** a name of a cluster, an availability group within the cluster to backup, a **backup preference** for the availability group, or combinations of these.... [T]he configuration may be specified through a **user interface or graphical user interface (GUI)**.... **A user** (e.g., administrator) may access the interface to configure the backup....”).

229. POSAs would have found it obvious to implement Chikkanayakanahally so that this preference is specified *during provisioning* of the source database, *e.g.*, during creation, for the reasons I explain below.

230. POSAs understood that including options related to database protection in database-provisioning user interfaces (UIs), including database-creation UIs, was well-known in the art. My testimony regarding a POSA's background knowledge is corroborated by, *e.g.*: Gomatam (EX1028), [0046] (disclosing "cloud infrastructure system" providing "database services"), [0056]-[0058] (disclosing allowing customer to order "services" including "database services", where customers may specify a service level as part of the "order" to determine "resources to be allocated for providing the requested [database] service"); Higginson (EX1029), 9:49-10:12, Fig. 4 (showing various "levels" in "database provisioning wizard"); RDS (EX1013), 16-17, 187 (disclosing that the database creation options include a daily "backup window" for taking "snapshots"); Oracle (EX1012), 7 (disclosing that options for provisioning a new database in Oracle Database Cloud include a "Backup Destination"); Google (EX1032), 2-3 (disclosing that the options for creating a MySQL instance in Google Cloud SQL included "Automatic Backups" and "Binary Logging"). I understand that EX1012 was downloaded from the URL <https://web.archive.org/web/20170118071307/https://docs.oracle.com/en/cloud/paa>

s/database-dbaascloud/csdbi/create-db-deployment.html. Based on my familiarity with the Wayback Machine as detailed above in paragraph 48, I understand the URL from which EX1012 was downloaded is the Internet Archive’s record of the website at the URL <https://docs.oracle.com/en/cloud/paas/database-dbaascloud/csdbi/create-db-deployment.html> archived on January 18, 2017.

231. Indeed, specifying a “backup preference” was a known creation option for Microsoft’s “AlwaysOn group database” product discussed in Chikkanayakanahally. Chikkanayakanahally, 6:53-55 (disclosing that the “preferred node of the AlwaysOn group”, selected based on the backup preference, “can change dynamically”), 7:42-62 (disclosing that “the system provides for **backing up** Microsoft SharePoint® data that is stored in Microsoft SQL Server® databases, ... where the SQL Server database has been **configured** with Microsoft **AlwaysOn** Availability Groups”, and that “SharePoint can be **configured** to use Microsoft SQL **AlwaysOn** Groups”). My testimony regarding a POSA’s knowledge of the AlwaysOn product is corroborated by, *e.g.*: SQL Server AlwaysOn Availability Group Backup Preference Setting (EX1108), 1 (disclosing that the backup preference may be set by users via a user interface (e.g., Backup Preferences setting is accessible) when configuring the “SQL Server AlwaysOn Availability Group (AG)”, for example, users may define backup preference via “Backup Preferences setting form” when using the user interface “New

Availability Group Wizard” for configuration, where “Prefer Secondary” is a “default configuration”; also disclosing that “SQL Server AlwaysOn Availability Group (AG) allows backups to be taken from the primary replica or any of the secondary replicas in the AG”); “7 more ways to Query Always on Availability groups” (EX1107), 4-5 (disclosing that setting “Availability group backup preference” for AlwaysOn includes at least four options: “Primary replica only”, “Secondary only,” “Prefer secondary,” and “Any replica”).

232. Moreover, POSAs understood that accepting the backup configuration during provisioning allowed a database to be protected once it is provisioned, since parameters for the backup would already be specified. POSAs would have reasonably expected success including backup configuration in a provisioning UI because it was well-known, as I noted above in this Section, and because such a UI implementation was within a POSA’s software skill.

9. Claim 21: “The system of claim 1, wherein the source database on the primary node and the copy of the source database on the secondary node comprise a database cluster the source database”

233. The ’340 patent says “[a] source database that resides on multiple nodes... forms a ‘*database cluster*.’” EX1001, 23:39-41.

234. The *source database on the primary node and the copy of the source database on the secondary node* are both replicas of the same *source database* (see

[1B1], which I discuss in Section V.B.1.c); thus, they collectively *comprise a database cluster*.

235. Claim 21 ends with “*the source database*,” which may have unintentionally resulted from a mistake during the final claim amendment. Compare EX1002, 426 (amendment to then-numbered claim 33) with EX1002, 427 (amendments to then-numbered claims 37 and 41). Nevertheless, Chikkanayakanahally discloses a *source database*. See [1B1] (Section V.B.1.c).

10. Claim 22: “The system of claim 21, wherein the switch over policy is part of a user defined active policy defining...”

236. The ’340 patent specification’s only reference to “*active policy*” is to an “Era Active Policy,” which is another term for a “user defined policy” that may specify the “switch over policy” as well as other parameters such as “the number and location of nodes in the database cluster.” EX1001, 25:65-26:45 (“The user defined policy may define **the number and location of nodes in the database cluster**[], the states of the nodes that designate a node as a primary node or a standby node, and a **switch over policy** to designate EA nodes. The **user defined policy is also referred to herein as ‘Era Active Policy’ or ‘EA policy’**”). Thus, the “*switch over policy*” (for selecting a *next active node*) can be part of the “*active policy*.”

237. Chikkanayakanahally teaches that a backup “configuration” may be “specified through a **user interface** or graphical **user interface (GUI)**,” *i.e.*, *user*

defined. Chikkanayakanahally, 10:24-35 (“The **configuration** may specify a name of a cluster, an availability group within the cluster to backup, a backup preference for the availability group, or combinations of these. [T]he configuration may be specified through a **user interface** or graphical **user interface** (GUI).... A **user** (e.g., administrator) may access the interface to configure the backup....”). See also Section V.B.1.i for [1F], where I discuss the “configuration” being user defined; Section V.B.8 for claim 20, where I discuss that the switch over policy may be user defined during provisioning. The “backup preference” that meets the recited *switch over policy* is *part of* the “configuration,” as I discussed in Section V.B.1.i for [1F].

238. As I explain below, POSAs would have found it obvious to implement Chikkanayakanahally in a manner in which the “configuration” also includes the other recited components of claim 22’s *active policy*.

- a. **“a number of nodes in the plurality of nodes based on a number of copies of the source database that are desired, ...”**

239. Each of the *secondary nodes* in the *plurality of nodes* contains a *copy of the source database*. See [1B1] (Section V.B.1.c).

240. POSAs would have been motivated to allow users to set the *number of secondary nodes based on the number of copies of the source database that are desired* to beneficially enable users to configure the number of replicas according

to their needs. UIs allowing such configuration were well-known, including for the “Microsoft SQL Server” product Chikkanayakanahally mentions, as I further explain in paragraph 241 below. Chikkanayakanahally, 10:4-8 (“In a specific embodiment, the database instances include **Microsoft SQL Server** instances, the cluster includes a Windows Server Failover Clustering (WSFC) cluster, and **the availability groups are Microsoft AlwaysOn Availability Groups** deployed in a WSFC cluster.”); EX1109, 1 (“This topic describes how to use the New Availability Group Wizard [*i.e.*, a UI,] in SQL Server Management Studio to create and **configure** an Always On availability group in SQL Server....”).

241. The mentioned Microsoft SQL Server, a well-known database management system, has its UI named “New Availability Group Wizard” allowing users to “specify and configure one or more replicas for the new availability group” on the “Specify Replicas page” of the UI when “configuring an Always On availability group in SQL Server.” EX1109, 1 (“This topic describes how to use the New Availability Group Wizard... to create and **configure** an Always On availability group in SQL Server....”), 4 (“On the **Specify Replicas** page, specify and configure one or more replicas for the new availability group.”). On the UI’s “Specify Replicas page,” the “Replicas” tab allows users “to specify *each instance of SQL Server* that will host *a* secondary replica” while “the server instance to which you are currently connected must host the primary replica”. EX1109, 4.

POSAs understood that each “instance of SQL Server” here corresponds to a node in the cluster. *See, e.g.,* Chikkanayakanahally, Fig. 7 (each DB Server Instance corresponds to a Node in the cluster, *e.g.,* DB Server Instance corresponds to Node 2). POSAs thus understood that, when users configure *a number of secondary replicas* (“copies of the source database”) *that are desired* using the UI, users are effectively defining a number of nodes based on a number of secondary replicas, because “*each instance of SQL Server... will host a secondary replica*”. EX1109, 4.

242. Further, as I discussed above for [1B1] (Section V.B.1.c) and [1F] (Section V.B.1.i), because the universe of possible nodes is the availability group’s *plurality of nodes* containing the primary and secondary nodes, when users are defining a number of nodes “that will [each] host a secondary replica” (*i.e.,* secondary nodes of the plurality of nodes), the users are defining a number of secondary nodes *in the plurality of nodes*, as recited in claim 22.

243. POSAs would reasonably have expected success with such an implementation because adding options for configuring such policies to “user interface[s]” (Chikkanayakanahally, 10:24-31) was within a POSA’s skill.

b. “a location of the nodes, ...”

244. The ’340 patent does not specify how a node’s “location” is identified. However, the ’340 patent notes—consistent with the prior art—that a “node” may

be a computer such as server. EX1001, 6:67-7:6 (describing that a “node[] includes database virtual machines..., a hypervisor..., and a controller/service VM....”), 7:58-8:5 (describing that “each [] node... may be a hardware device, such as a server”, or, for example, “a NX-1000 server, ..., or server computers,” or “other type of hardware device, such as a personal computer, an input/output or peripheral unit such as a printer, or any type of device that is suitable for use as a node within the cluster”); Chikkanayakanahally, 9:5-10 (“A node of a cluster may... host a database instance.”).

245. POSAs understood that a computer’s **network** location was commonly treated as the computer’s “location.” EX1110, 2:40-44 (“Once provisioned, a user can native access to the data instance(s) in the data plane, and can simply point existing applications (such as MySQL applications) to the **DNS (domain name system) address** or other **location information** for the particular instance”); EX1033, [0003] (disclosing that “DNS facilitates Internet communication” and that DNS has “resource records includ[ing] information concerning each network node, including the IP address of the network node, and the domain names associated with the IP address”, where the IP address is the network location of “each network node”).

246. Chikkanayakanahally’s nodes are networked with each other. Chikkanayakanahally, 2:65-3:24 (describing “a distributed computer network 100”

with respect to FIG. 1), 5:17-22 (describing “a cluster 220 of backup client nodes such as first, second, third, fourth, and fifth nodes 225A-E, each of which are connected through a communication network [such as the] communication network shown in FIG. 1 and described above), Figs. 1-2.

247. POSAs would have been motivated to allow the user to specify the *location of the nodes* to beneficially ensure that the backup server has the information needed to communicate with the nodes. POSAs would reasonably have expected success because adding network-location specification options to “user interface[s]” (Chikkanayakanahally, 10:24-31) was within a POSA’s skill.

c. “and one or more conditions for selecting a new primary node when a current primary node becomes unavailable”

248. In obvious implementations of Chikkanayakanahally, *one of the secondary node is designated to become the primary node based on the first policy, which specifies one or more conditions for selecting a new primary node when a current primary node becomes unavailable. See Section V.B.1.e above for [1B3].*

249. POSAs would have been motivated to let users specify this policy in the “configuration” (i.e., part of the *active policy*) to beneficially increase user configurability, and POSAs would reasonably have expected success because adding such policy-configuration options to “user interface[s]” (Chikkanayakanahally, 10:24-31) was within a POSA’s skill.

11. Claim 7

- a. [7PRE]: “A non-transitory computer-readable media comprising computer readable instructions stored thereon that when executed by a processor causes the processor to”**

250. As I discussed above for [1A] (Section V.B.1.b), Chikkanayakanahally’s system includes a *processor that executes computer-readable instructions*. These “instructions” may be stored in various “*computer-readable storage medi[a]*” such as “a persistent store,” examples of which include “hard drives, CDROM, [and] DVDROM,” which POSAs understood are *non-transitory*. Chikkanayakanahally, 2:21-40. As I explain below for [7A1]-[7F] (Section V.B.11.b), these instructions *when executed by the processor cause the processor to* perform claim’s recited steps.

- b. [7A1]-[7F]**

251. [7A1]-[7F] recite identical language to the corresponding Claim 1 elements indicated below; thus, Chikkanayakanahally meets [7A1]-[7F] for the same reasons I gave above at Sections V.B.1.c-V.B.1.j for the corresponding claim 1 elements (whether or not claim 7 requires that a single processor perform all steps).

Claim 7 Element	Claim 1 Element
-----------------	-----------------

[7A1]-[7A3]	[1B1]-[1B3]
[7B]	[1C]
[7C]	[1D]
[7D]	[1E]
[7E]	[1F]
[7F]	[1G]

12. Claims 8-12, 23-26

252. The claims depending from claim 7 recite identical or nearly-identical language to the claims depending from claim 1 indicated below, and Chikkanayakanahally thus meets them for the same reasons I gave above at Sections V.B.2-10.

Claim 7 Dependent Claim	Claim 1 Dependent Claim
8	2
9	3
10	4
11	5
12	6
23	19
24	20

25	21 (claim 21 ends with “ <i>the source database</i> ,” unlike claim 25)
26	22

13. Claim 13

a. [13PRE]: “A method comprising”

253. As I discussed above for [1A] (Section V.B.1.b), the processor in Chikkanayakanahally’s system executes instructions. POSAs understood that when the processor executes those instructions, it performs a method comprising the instructions’ corresponding steps. *E.g.*, Chikkanayakanahally, 2:21-28 (“described embodiments can be implemented in numerous ways, including... a method”), claim 1 (18:30) (“A method...”). This method comprises the claim’s remaining steps. *See* below Section V.B.13.b.

b. [13A1]-[13F]

254. [13A1]-[13A3] recite steps performed “by a processor executing computer-readable instructions stored on a memory,” and [13B]-[13F] recite steps performed by “the processor.” As I explained above for [1A] (Section V.B.1.b), the processor in Chikkanayakanahally’s system executes computer-readable instructions stored on a memory. The steps in [13A1]-[13F] recite identical language to the corresponding elements of claim 1 indicated below; thus,

Chikkanayakanahally meets [13A1]-[13F] for the same reasons as I gave above at Sections V.B.1.c-V.B.1.j for the corresponding claim 1 elements (whether or not claim 13 requires that a single processor perform all the steps).

Claim 13 Element	Claim 1 Element
[13A1]-[13A3]	[1B1]-[1B3]
[13B]	[1C]
[13C]	[1D]
[13D]	[1E]
[13E]	[1F]
[13F]	[1G]

14. Claims 14-18, 27-30

255. The claims depending from claim 13 recite identical or nearly-identical language to the claims depending from claim 1 indicated below, and Chikkanayakanahally thus meets them for the same reasons I gave above at Sections V.B.2-10.

Claim 13 Dependent Claim	Claim 1 Dependent Claim
14	2
15	3
16	4

17	5
18	6
27	19
28	20
29	21 (the end of claim 21 recites “ <i>the source database</i> ”; the end of claim 29 does not)
30	22

VI. CLAIM LISTING

The following claim listing assigns element labels (*e.g.*, [1B1], [1B2], etc.) to certain claims for clarity.

Claim 1
[1PRE] A system comprising:
[1A] a processor executing computer-readable instructions stored on a memory to:
[1B1] select one of a plurality of nodes as a primary node capable of storing a source database, the remaining ones of the plurality of nodes being a secondary node, each of the secondary node capable of storing a copy of the source database,
[1B2] wherein updates made to the source database on the primary node are able to be copied to the copy of the source database on the secondary node,
[1B3] and wherein in response to the primary node becoming unavailable, one of the secondary node is designated to become the primary node based on a first policy;
[1C] designate a first node from the primary node or the secondary node as an initial active node;

[1D] capture one or more snapshots of the source database or the copy of the source database from the first node designated as the initial active node;
[1E] determine that the first node has become unavailable;
[1F] select, in response to the first node becoming unavailable and based on a user defined switch over policy, a second node from the primary node or the secondary node as a next active node, wherein the switch over policy defines a criteria for selecting the next active node from the plurality of nodes to capture the one or more snapshots from; and
[1G] capture the one or more snapshots of the source database or the copy of the source database from the second node in response to selecting the second node as the next active node.
Claim 2
The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node.
Claim 3
The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node.
Claim 4
The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node, or upon determining that none of the plurality of nodes are the secondary node, select the next active node from the plurality of nodes that is the primary node.
Claim 5
The system of claim 1, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node, or upon determining that none of the plurality of nodes is the primary node, select the next active node from the plurality of nodes that is the secondary node.
Claim 6
The system of claim 1, wherein the switch over policy defines a priority assigned to at least a subset of the plurality of nodes and wherein the processor further executes the computer-readable instructions to select the next active node based upon the priority.
Claim 7
[7PRE] A non-transitory computer-readable media comprising computer readable instructions stored thereon that when executed by a processor causes the processor to:
[7A1] select one of a plurality of nodes as a primary node capable of storing a source database and the remaining ones of the plurality of nodes being a

secondary node, each of the secondary node capable of storing a copy of the source database,
[7A2] wherein updates made to the source database on the primary node are able to be copied to the copy of the source database on the secondary node,
[7A3] and wherein in response to the primary node becoming unavailable, one of the secondary node is designated to become the primary node based on a first policy;
[7B] designate a first node from the primary node or the secondary node as an initial active node;
[7C] capture one or more snapshots of the source database or the copy of the source database from the first node designated as the initial active node;
[7D] determine that the first node has become unavailable;
[7E] select, in response to the first node becoming unavailable and based on a user defined switch over policy, a second node from the primary node or the secondary node as a next active node, wherein the switch over policy defines a criteria for selecting the next active node from the plurality of nodes to capture the one or more snapshots from; and
[7F] capture the one or more snapshots of the source database or the copy of the source database from the second node in response to selecting the second node as the next active node.
Claim 8
The non-transitory computer-readable media of claim 7, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node.
Claim 9
The non-transitory computer-readable media of claim 7, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node.
Claim 10
The non-transitory computer-readable media of claim 7, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node, or upon determining that none of the plurality of nodes are the secondary node, select the next active node from the plurality of nodes that is the primary node.
Claim 11
The non-transitory computer-readable media of claim 7, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node, or upon determining that none of the plurality of nodes is the

primary node, select the next active node from the plurality of nodes that is the secondary node.

Claim 12

The non-transitory computer-readable media of claim 7, wherein the switch over policy defines a priority assigned to at least a subset of the plurality of nodes and wherein the processor further executes the computer-readable instructions to select the next active node based upon the priority.

Claim 13

[13PRE] A method comprising:

[13A1] selecting, by a processor executing computer-readable instructions stored on a memory, one of a plurality of nodes as a primary node capable of storing a source database and the remaining ones of the plurality of nodes being a secondary node, each of the secondary node capable of storing a copy of the source database,

[13A2] wherein updates made to the source database on the primary node are able to be copied to the copy of the source database on the secondary node,

[13A3] and wherein in response to the primary node becoming unavailable, one of the secondary node is designated to become the primary node based on a first policy;

[13B] designating, by the processor, a first node from the primary node or the secondary node as an initial active node;

[13C] capturing, by the processor, one or more snapshots of the source database or the copy of the source database from the first node designated as the initial active node;

[13D] determining, by the processor, that the first node has become unavailable;

[13E] selecting, by the processor, in response to the first node becoming unavailable and based on a user defined switch over policy, a second node from the primary node or the secondary node as a next active node, wherein the switch over policy defines a criteria for selecting the next active node from the plurality of nodes to capture the one or more snapshots from; and

[13F] capturing, by the processor, one or more snapshots of the source database or the copy of the source database from the second node in response to selecting the second node as the next active node.

Claim 14

The method of claim 13, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node.

Claim 15

The method of claim 13, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node.

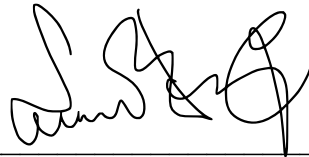
Claim 16
The method of claim 13, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the secondary node, or upon determining that none of the plurality of nodes are the secondary node, select the next active node from the plurality of nodes that is the primary node.
Claim 17
The method of claim 13, wherein the switch over policy defines selection of the next active node from the plurality of nodes that is the primary node, or upon determining that none of the plurality of nodes is the primary node, select the next active node from the plurality of nodes that is the secondary node.
Claim 18
The method of claim 13, wherein the switch over policy defines a priority assigned to at least a subset of the plurality of nodes and wherein the method further comprises selecting, by the processor, the next active node based upon the priority.
Claim 19
The system of claim 1, wherein the plurality of nodes are distributed between a first cluster and a second cluster.
Claim 20
The system of claim 1, wherein the switch over policy is defined during provisioning of the source database.
Claim 21
The system of claim 1, wherein the source database on the primary node and the copy of the source database on the secondary node comprise a database cluster the source database.
Claim 22
The system of claim 21, wherein the switch over policy is part of a user defined active policy defining a number of nodes in the plurality of nodes based on a number of copies of the source database that are desired, a location of the nodes, and one or more conditions for selecting a new primary node when a current primary node becomes unavailable.
Claim 23
The non-transitory computer-readable media of claim 7, wherein the plurality of nodes are distributed between a first cluster and a second cluster.
Claim 24
The non-transitory computer-readable media of claim 7, wherein the switch over policy is defined during provisioning of the source database.
Claim 25

The non-transitory computer-readable media of claim 7, wherein the source database on the primary node and the copy of the source database on the secondary node comprise a database cluster.
Claim 26
The non-transitory computer-readable media of claim 25, wherein the switch over policy is part of a user defined active policy defining a number of nodes in the plurality of nodes based on a number of copies of the source database that are desired, a location of the nodes, and one or more conditions for selecting a new primary node when a current primary node becomes unavailable.
Claim 27
The method of claim 13, wherein the plurality of nodes are distributed between a first cluster and a second cluster.
Claim 28
The method of claim 13, wherein the switch over policy is defined during provisioning of the source database.
Claim 29
The method of claim 13, wherein the source database on the primary node and the copy of the source database on the secondary node comprise a database cluster.
Claim 30
The method of claim 29, wherein the switch over policy is part of a user defined active policy defining a number of nodes in the plurality of nodes based on a number of copies of the source database that are desired, a location of the nodes, and one or more conditions for selecting a new primary node when a current primary node becomes unavailable.

I declare that all statements made herein of my own 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 are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

I declare under penalty of perjury that the foregoing is true and correct.

Dated: March 20, 2025

A handwritten signature in black ink, appearing to read 'Samrat Bhattacharjee', written over a horizontal line.

Samrat Bhattacharjee, Ph.D.