(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 April 2013 (04.04.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/048971 A1

(51) International Patent Classification:
H04L 12/28 (2006.01)

(21) International Application Number:
PCT/US2012/056950

(22) International Filing Date:
24 September 2012 (24.09.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/541,055 29 September 2011 (29.09.2011) US
13/415,700 8 March 2012 (08.03.2012) US

(71) Applicant: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, M/S 5op7, Redwood Shores, California 94065 (US).

(72) Inventors: **SHI, Peizhi**; Room 401, Building #5, Yushuguanxili Street, Xicheng District, Beijing (CN). **JIN, Yongshun**; 1-2-1004, Jiaotong University Road, Haidian District, Beijing (CN).

(74) Agents: **MEYER, Sheldon, R.** et al.; FLIESLER MEYER LLP, 650 California Street, Fourteenth Floor, San Francisco, California 94108 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

(54) Title: SYSTEM AND METHOD FOR SUPPORTING DIFFERENT MESSAGE QUEUES IN A TRANSACTIONAL MIDDLEWARE MACHINE ENVIRONMENT

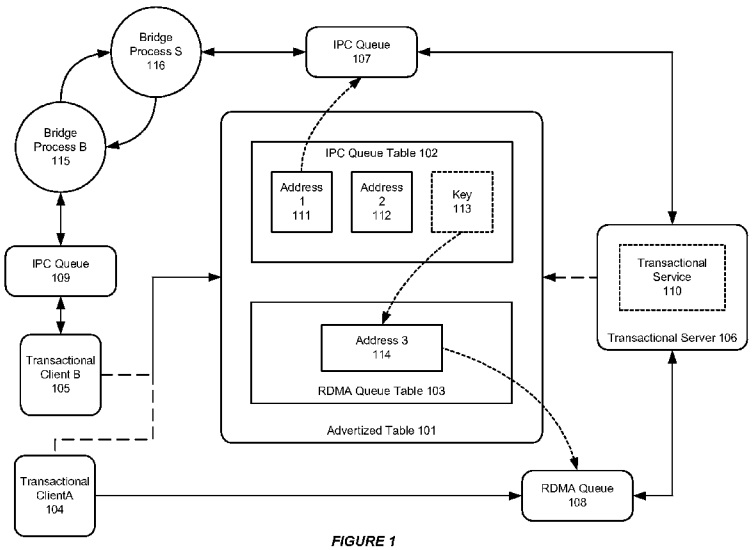

FIGURE 1

(57) Abstract: A system and method can support different message queues in a transactional middleware machine environment. The transactional middleware machine environment includes an advertized table that comprises a first queue table and a second queue table, with the first queue table storing address information for a first message queue and the second queue table storing address information for a second message queue. The advertized table is further adaptive to be used by a first transactional client to locate a transactional service provided by a transactional server. The first transactional client operates to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

WO 2013/048971 A1

SYSTEM AND METHOD FOR SUPPORTING DIFFERENT MESSAGE QUEUES IN A TRANSACTIONAL MIDDLEWARE MACHINE ENVIRONMENT

Copyright Notice:

Field of Invention:

[0002] The present invention is generally related to computer systems and software such as middleware, and is particularly related to supporting a transactional middleware machine environment.

Background:

[0003] A transactional middleware system, or a transaction oriented middleware, includes enterprise application servers that can process various transactions within an organization. With the developments in new technologies such as high performance network and multiprocessor computers, there is a need to further improve the performance of the transactional middleware. These are the generally areas that embodiments of the invention are intended to address.

Summary:

[0004] Described herein is a system and method for supporting different message queues in a transactional middleware machine environment. The transactional middleware machine environment includes an advertized table that comprises a first queue table and a second queue table, with the first queue table storing address information for a first message queue and the second queue table storing address information for a second message queue. The advertized table is further adaptive to be used by a first transactional client to locate a transactional service provided by a transactional server. The first transactional client operates to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

Brief Description of the Figures:

[0005] **Figure 1** shows an illustration of a transactional middleware machine environment that supports different message queues, in accordance with an embodiment of the invention.

[0006] **Figure 2** illustrates an exemplary flow chart for supporting different message queues in a transactional middleware machine environment, in accordance with an embodiment of the

invention.

[0007] **Figure 3** shows an illustration of a transactional middleware machine environment with bypass bridge feature activated, in accordance with an embodiment of the invention.

[0008] **Figure 4** shows an illustration of a transactional middleware machine environment with bypass bridge feature not activated, in accordance with an embodiment of the invention.

[0009] **Figure 5** shows a functional diagram of a transactional middleware machine environment in accordance with an embodiment of the invention.

[0010] **Figure 6** is a block diagram illustrating the structure of a transactional server in accordance with an embodiment of the present invention.

[0011] **Figure 7** is a block diagram illustrating the structure of a transactional client in accordance with an embodiment of the present invention.

Detailed Description:

[0012] Described herein is a system and method for supporting a transactional middleware system, such as Tuxedo, that can take advantage of fast machines with multiple processors, and a high performance network connection. An advertized service table can be used by a transactional server to locate a message queue that can send and receive messages via Remote Direct Memory Access (RDMA) protocol instead of via Inter-process Communication (IPC). The transactional middleware machine environment comprises a server that listens to a first message queue and a second message queue. The transactional middleware machine environment further comprises an advertized table that comprises a first queue table and a second queue table, wherein the first queue table stores address information for the first message queue and the second queue table stores address information for the second message queue. The advertized table is adaptive to be used by a client to locate a transactional service provided by the server. The first queue table contains a key that indicates the address information of the transactional service stored in the second queue table. The client can obtain the address information of the transactional service in the second message queue after first looking up the key in the first queue table.

[0013] In accordance with an embodiment of the invention, the system comprises a combination of high performance hardware, e.g. 64-bit processor technology, high performance large memory, and redundant InfiniBand and Ethernet networking, together with an application server or middleware environment, such as WebLogic Suite, to provide a complete Java EE application server complex which includes a massively parallel in-memory grid, that can be provisioned quickly, and can scale on demand. In accordance with an embodiment, the system can be deployed as a full, half, or quarter rack, or other configuration, that provides an application server grid, storage area network, and InfiniBand (IB) network. The middleware

machine software can provide application server, middleware and other functionality such as, for example, WebLogic Server, JRockit or Hotspot JVM, Oracle Linux or Solaris, and Oracle VM. In accordance with an embodiment, the system can include a plurality of compute nodes, IB switch gateway, and storage nodes or units, communicating with one another via an IB network. When implemented as a rack configuration, unused portions of the rack can be left empty or occupied by fillers.

[0014] In accordance with an embodiment of the invention, referred to herein as "Sun Oracle Exalogic" or "Exalogic", the system is an easy-to-deploy solution for hosting middleware or application server software, such as the Oracle Middleware SW suite, or Weblogic. As described herein, in accordance with an embodiment the system is a "grid in a box" that comprises one or more servers, storage units, an IB fabric for storage networking, and all the other components required to host a middleware application. Significant performance can be delivered for all types of middleware applications by leveraging a massively parallel grid architecture using, e.g. Real Application Clusters and Exalogic Open storage. The system delivers improved performance with linear I/O scalability, is simple to use and manage, and delivers mission-critical availability and reliability.

[0015] In accordance with an embodiment of the invention, Tuxedo is a set of software modules that enables the construction, execution, and administration of high performance, distributed business applications and has been used as transactional middleware by a number of multi-tier application development tools. Tuxedo is a middleware platform that can be used to manage distributed transaction processing in distributed computing environments. It is a proven platform for unlocking enterprise legacy applications and extending them to a services oriented architecture, while delivering unlimited scalability and standards-based interoperability.

[0016] In accordance with an embodiment of the invention, a transactional middleware system, such as a Tuxedo system, can take advantage of fast machines with multiple processors, such as an Exalogic middleware machine, and a high performance network connection, such as an Infiniband (IB) network.

[0017] In accordance with an embodiment of the invention, a transactional middleware system can exchange messages between a local machine and a remote machine using Remote Direct Memory Access (RDMA) protocol to achieve short latency in a manner like a local message transfer, e.g. bypassing the bridge process and preventing single point bottleneck. Exchanging messages between a local machine and a remote machine using RDMA protocol is disclosed in U.S. Application No. 13/415,670, filed March 8, 2012, entitled "SYSTEM AND METHOD FOR PREVENTING SINGLE-POINT BOTTLENECK IN A TRANSACTIONAL MIDDLEWARE MACHINE ENVIRONMENT," which application is incorporated herein by reference in its entirety.

Supporting different message queues

[0018] In accordance with an embodiment of the invention, different message queues, such as a RDMA message queue and a System V Inter-process Communication (IPC) message queue, can be supported in a transactional middleware machine environment, in order to prevent single-point bottleneck in transferring messages among machines.

[0019] **Figure 1** shows an illustration of a transactional middleware machine environment that supports different message queues, in accordance with an embodiment of the invention. As shown in Figure 1, a transactional server 106 can publish one or more transactional services 110 in an advertized table 101, such as a bulletin board in Tuxedo. The advertized table is adaptive to be used by one or more clients 104 and 105 to locate a transactional service provided by the server.

[0020] In accordance with an embodiment of the invention, an advertized service table can include a RDMA queue table 103, in addition to an IPC queue table 102. The IPC queue stores address information for the IPC message queue 107 and can be used by a transactional client server 105 to locate the transactional service in the IPC queue. Additionally, the RDMA queue table stores address information for the RDMA message queue 108 and can be used by a transactional client server 104 to locate the transactional service in the RDMA queue.

[0021] In accordance with an embodiment of the invention, a transactional server can listen to an IPC queue and a RDMA queue at the same time. The RDMA queue table can be implemented so that the RDMA queue can be used concurrently and consistently with the System V IPC queue.

[0022] A transactional client can look up the queue tables in the advertized table in order to obtain the location information of the service that it needs. The client can first look up the IPC queue table 102 in the advertized table 101, e.g. the Tuxedo Bulletin Board (BB). If the obtained address information is a real IPC queue address, for example address 1 with a positive value 111, the client can use this IPC queue address to access the transactional server using an IPC queue 107 for the requested transactional services.

[0023] On the other hand, the transactional client can obtain a key 113 instead of a real IPC queue address. For example, the obtained address information can be a “faked” IPC queue address with a negative value. The client can look up the RDMA queue table in turn using the “faked” IPC queue address for a queue address 114 in the RDMA queue. Then, the client can access the transactional server for the requested transactional services using the RDMA queue.

[0024] **Figure 2** illustrates an exemplary flow chart for supporting different message queues in a transactional middleware machine environment, in accordance with an embodiment of the invention. As shown in Figure 2, at step 201, an advertized table that comprises a first queue

table and a second queue table is provided, wherein the first queue table stores address information for a first message queue and the second queue table stores address information for a second message queue. Then, at step 202, the advertized table can be used by a first transactional client to locate a transactional service provided by a transactional server. Finally, at step 203, the first transactional client can look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

Referencing MSGQ queue information in Tuxedo Bulletin Board (BB)

[0025] In accordance with an embodiment of the invention, a Tuxedo application can take advantage of the RDMA protocol and use a MSGQ queue, which is a RDMA queue, in order to bypass the bridge processes and achieve a shorter latency in a manner similar to a local message transfer.

[0026] When the bypass bridge feature is activated, a Tuxedo client can use the MSGQ queue instead of System V IPC queue to send messages to a Tuxedo server. When the bypass bridge feature is not activated, a Tuxedo client can use the System V IPC queue instead of the MSGQ queue to send messages to Tuxedo server.

[0027] In accordance with an embodiment of the invention, different versions of Tuxedo can co-exist in a multi-process configuration. Processes in a machine installed with a version of Tuxedo without the bypass bridge feature can access the advertized table (or the bulletin board) in the version of Tuxedo with bypass bridge feature without problem, even though the internal structure in the Tuxedo Bulletin Board remains unchanged.

[0028] For example, in a version of Tuxedo without the bypass bridge feature, a bulletin board in a shared memory contains only System V IPC queue address, each of which is an eight-byte “long” variable. On the other hand, in a version of Tuxedo with bypass bridge feature, the RDMA MSGQ address can be an array of 128 bytes. Hence, if the Tuxedo bulletin board is changed, the offset of the bulletin board fields in different binaries can be wrong, and the binary compatibility is broken.

[0029] **Figure 3** shows an illustration of a transactional middleware machine environment with bypass bridge feature activated, in accordance with an embodiment of the invention. As shown in Figure 3, a MSGQ queue table 304 can be created in Tuxedo if the bypass bridge feature is activated. Each entry in this MSGQ queue table 306 or 308 contains information for a pair of MSGQ /System V IPC queues: {MSGQ queue name, MSGQ queue key, System V IPC queue address}.

[0030] Since the bypass bridge feature is activated, the IPC queue table 303 can contain a faked System V IPC queue address 305 or 307, which indicates that the location of the RDMA queue is stored in the RDMA queue table. For example, a faked System V IPC queue address

can be a negative value. Then, the client can look into the RDMA queue table to obtain the queue information.

[0031] As shown in **Figure 3**, the bulletin board data structure can be kept unchanged with the bypass bridge feature implemented. A separate section, containing the table of 128-bytes MSGQ address entries, can be created, appended to the end of bulletin board. The index of the entry can be stored in the System V IPC queue address field in the corresponding entry in the original bulletin board. Since there are at most 65536 RDMA MSGQ queues, the index may need 16 bits, and the other 16 bits can be used to distinguish whether this queue address is a System V IPC queue address, or an index for RDMA MSGQ queue address.

[0032] **Figure 4** shows an illustration of a transactional middleware machine environment with bypass bridge feature not activated, in accordance with an embodiment of the invention. As shown in Figure 4, when the bypass bridge feature is not activated, no MSGQ section is created, and the internal structure of the bulletin board remains unchanged. Thus, the client can only see a positive System V IPC queue address 404 and 406 in the IPC queue table 402 when the bypass BRIDGE feature is not activated. The queue addresses in the bulletin board indicate the System V IPC queue addresses, so that the binaries can have the correct offsets of any field in the bulletin board.

[0033] In accordance with an embodiment of the invention, the data structure change can be isolated, so that the code change can be limited to a smaller scope since the data structure of the advertized service table does not need to be changed. Furthermore, the data structure change introduced by the bypass bridge feature in Tuxedo, which is a C based application, maintains interoperability among different versions of Tuxedo.

[0034] With reference to **Figure 5**, a system 500 in accordance with an embodiment of the invention is described. Figure 5 is a functional diagram which indicates function realized by system 500. System 500 includes a memory 500, a locator 520 and a look-up unit 530. Memory 500 is configured to provide an advertised table that includes a first queue table and a second queue table. The first queue table stores address information for a first message queue and the second queue table stores address information for a second message queue.

[0035] Locator 520 is configured to allow the advertised table to be used by a first transactional client to locate a transactional service provided by a transactional server. Look-up unit 530 is configured to allow the first transactional client to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

[0036] **Figure 6** is a block diagram illustrating the structure of a transactional server 610 for supporting different message queues in a transactional middleware machine environment in accordance with an embodiment of the present invention. The blocks of the transactional server

610 may be implemented by hardware, software, or a combination of hardware and software to carry out the principles of the invention. It is understood by those skilled in the art that the blocks described in Figure 6 may be combined or separated into sub-blocks to implement the principles of the invention as described above. Therefore, the description herein may support any possible combination or separation or further definition of the functional blocks described herein.

[0037] As shown in **Figure 6**, the transactional server 610 may comprise a publishing unit 621 for publishing address information of one or more transactional services in an advertized table (not shown). As described above, the advertized table may comprise a first queue table and a second queue table, wherein the first queue table stores address information for a first message queue and the second queue table stores address information for a second message queue. The advertized table may be adaptive to be used by a first transactional client (not shown) to locate a transactional service of the one or more transactional services. The first transactional client may operate to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

[0038] In accordance with an embodiment of the invention, the transactional server 610 may further comprise a listening unit 612 for listening to both the first message queue and the second message queue.

[0039] In accordance with an embodiment of the invention, the first transactional client may operate to send a message to the transactional server via the second message queue, based on the address information of the transactional service stored in the second queue table.

[0040] In accordance with an embodiment of the invention, the advertized table may be adaptive to be used by a second transactional client (not shown) to locate a transactional service of the one or more transactional services. The second transactional client may operate to send a message to the transactional server via a local bridge process using the first message queue, based on the address information of the transactional service stored in the first queue table.

[0041] In accordance with an embodiment of the invention, the first message queue may be an IPC queue and the second message queue may be a RDMA queue. The key may be a faked IPC queue address.

[0042] In accordance with an embodiment of the invention, the second queue table may be appended to an end of an existing advertized table with only the first queue table.

[0043] In accordance with an embodiment of the invention, each entry in the second queue table may contain information on information for both the first message queue and the second message queue.

[0044] **Figure 7** is a block diagram illustrating the structure of a transactional client 710 for supporting different message queues in a transactional middleware machine environment in accordance with an embodiment of the present invention. The blocks of the transactional client

710 may be implemented by hardware, software, or a combination of hardware and software to carry out the principles of the invention. It is understood by those skilled in the art that the blocks described in Figure 6 may be combined or separated into sub-blocks to implement the principles of the invention as described above. Therefore, the description herein may support any possible combination or separation or further definition of the functional blocks described herein.

[0045] As shown in **Figure 7**, the transactional server 710 may comprise a using unit 711 for using an advertized table (not shown), in which address information of one or more transactional services are published by a transactional server (not shown), to locate a transactional service of the one or more transactional services. As described above, the advertized table may comprise a first queue table and a second queue table. The first queue table stores address information for a first message queue and the second queue table stores address information for a second message queue. The transactional server 710 may further comprise a looking up unit 712 for looking up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

[0046] In accordance with an embodiment of the invention, the transactional client 710 may further comprise a sending unit 713 for sending a message to the transactional server via the second message queue, based on the address information of the transactional service stored in the second queue table.

[0047] In accordance with an embodiment of the invention, the transactional client may further comprise a sending unit 713 for sending a message to the transactional server via a local bridge process using the first message queue, based on the address information of the transactional service stored in the first queue table.

[0048] In accordance with an embodiment of the invention, the transactional server may listen to both the first message queue and the second message queue.

[0049] In accordance with an embodiment of the invention, the first message queue may be an IPC queue and the second message queue may be a RDMA queue. The key may be a faked IPC queue address.

[0050] In accordance with an embodiment of the invention, the second queue table may be appended to an end of an existing advertized table with only the first queue table.

[0051] In accordance with an embodiment of the invention, each entry in the second queue table may contain information on information for both the first message queue and the second message queue.

[0052] Another embodiment comprises a system for supporting different message queues in a transactional middleware machine environment, comprising means for providing an advertized table that comprises a first queue table and a second queue table, wherein the first queue table stores address information for a first message queue and the second queue table stores address

information for a second message queue, means for allowing the advertized table to be used by a first transactional client to locate a transactional service provided by a transactional server, and means for allowing the first transactional client to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

[0053] Another embodiment comprises a system further comprising means for allowing said transactional server to listen to both the first message queue and the second message queue.

[0054] Another embodiment comprises a system further comprising means for allowing the first transactional client to send a message to the transactional server via the second message queue, based on the address information of the transactional service stored in the second queue table.

[0055] Another embodiment comprises a system further comprising means for allowing the advertized table to be used by a second transactional client to locate a transactional service provided by a transactional server.

[0056] Another embodiment comprises a system further comprising means for allowing the second transactional client to send a message to the transactional server via a local bridge process using the first message queue, based on the address information of the transactional service stored in the first queue table.

[0057] Another embodiment comprises a system wherein the first message queue is an IPC queue and the second message queue is a RDMA queue.

[0058] Another embodiment comprises a system wherein the key is a faked IPC queue address.

[0059] Another embodiment comprises a system further comprising means for allowing the second queue table to be appended to an end of an existing advertized table with only the first queue table.

[0060] Another embodiment comprises a system further comprising means for allowing the transactional server to publish address information of the transactional service in the advertized table.

[0061] Another embodiment comprises a system further comprising means for allowing each entry in the second queue table to contain information on information for both the first message queue and the second message queue.

[0062] The present invention may be conveniently implemented using one or more conventional general purpose or specialized digital computer, computing device, machine, or microprocessor, including one or more processors, memory and/or computer readable storage media programmed according to the teachings of the present disclosure. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present

disclosure, as will be apparent to those skilled in the software art.

[0063] In some embodiments, the present invention includes a computer program product which is a storage medium or computer readable medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the processes of the present invention. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

[0064] The foregoing description of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. The variations may include a combination of two or more features disclosed herein. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalence.

Claims:

What is claimed is:

1. A system for supporting different message queues in a transactional middleware machine environment, comprising:

an advertized table that comprises a first queue table and a second queue table, wherein the first queue table stores address information for a first message queue and the second queue table stores address information for a second message queue,

wherein the advertized table is adaptive to be used by a first transactional client to locate a transactional service provided by a transactional server, and

wherein the first transactional client operates to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

2. The system of claim 1, wherein said transactional server listens to both the first message queue and the second message queue.

3. The system of claim 1 or 2, wherein the first transactional client operates to send a message to the transactional server via the second message queue, based on the address information of the transactional service stored in the second queue table.

4. The system of claim 1 or 2, wherein the advertized table is adaptive to be used by a second transactional client to locate a transactional service provided by a transactional server.

5. The system of claim 4, wherein the second transactional client operates to send a message to the transactional server via a local bridge process using the first message queue, based on the address information of the transactional service stored in the first queue table.

6. The system of claim any one of claims 1 to 5, wherein the first message queue is an IPC queue and the second message queue is a RDMA queue.

7. The system of claim any one of claims 1 to 6, wherein the key is a faked IPC queue address.

8. The system of claim any one of claims 1 to 7, wherein the second queue table is

appended to an end of an existing advertized table with only the first queue table.

9. The system of any one of claims 1 to 8, wherein the transactional server publish address information of the transactional service in the advertized table.

10. The system of any one of claims 1 to 9, wherein each entry in the second queue table contains information on information for both the first message queue and the second message queue.

11. A method for supporting different message queues in a transactional middleware machine environment, comprising:

providing an advertized table that comprises a first queue table and a second queue table, wherein the first queue table stores address information for a first message queue and the second queue table stores address information for a second message queue,

allowing the advertized table to be used by a first transactional client to locate a transactional service provided by a transactional server, and

allowing the first transactional client to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table.

12. The method of claim 11, further comprising allowing said transactional server to listen to both the first message queue and the second message queue.

13. The method of claim 11 or 12, further comprising allowing the first transactional client to send a message to the transactional server via the second message queue, based on the address information of the transactional service stored in the second queue table.

14. The method of any one of claims 11 to 13, further comprising allowing the advertized table to be used by a second transactional client to locate a transactional service provided by a transactional server.

15. The method of claim 14, further comprising allowing the second transactional client to send a message to the transactional server via a local bridge process using the first message queue, based on the address information of the transactional service stored in the first queue table.

16. The method of any one of claims 11 to 15, wherein the first message queue is an IPC queue and the second message queue is a RDMA queue.

17. The method of any one of claims 11 to 16, wherein the key is a faked IPC queue address.

18. The method of any one of claims 11 to 17, further comprising allowing the second queue table to be appended to an end of an existing advertized table with only the first queue table.

19. The method of any one of claims 11 to 18, further comprising allowing the transactional server to publish address information of the transactional service in the advertized table.

20. The method of any one of claims 11 to 19, further comprising allowing each entry in the second queue table to contain information on information for both the first message queue and the second message queue.

21. A computer program for causing one or more processors to implement the method recited in any one of claims 11 to 20.

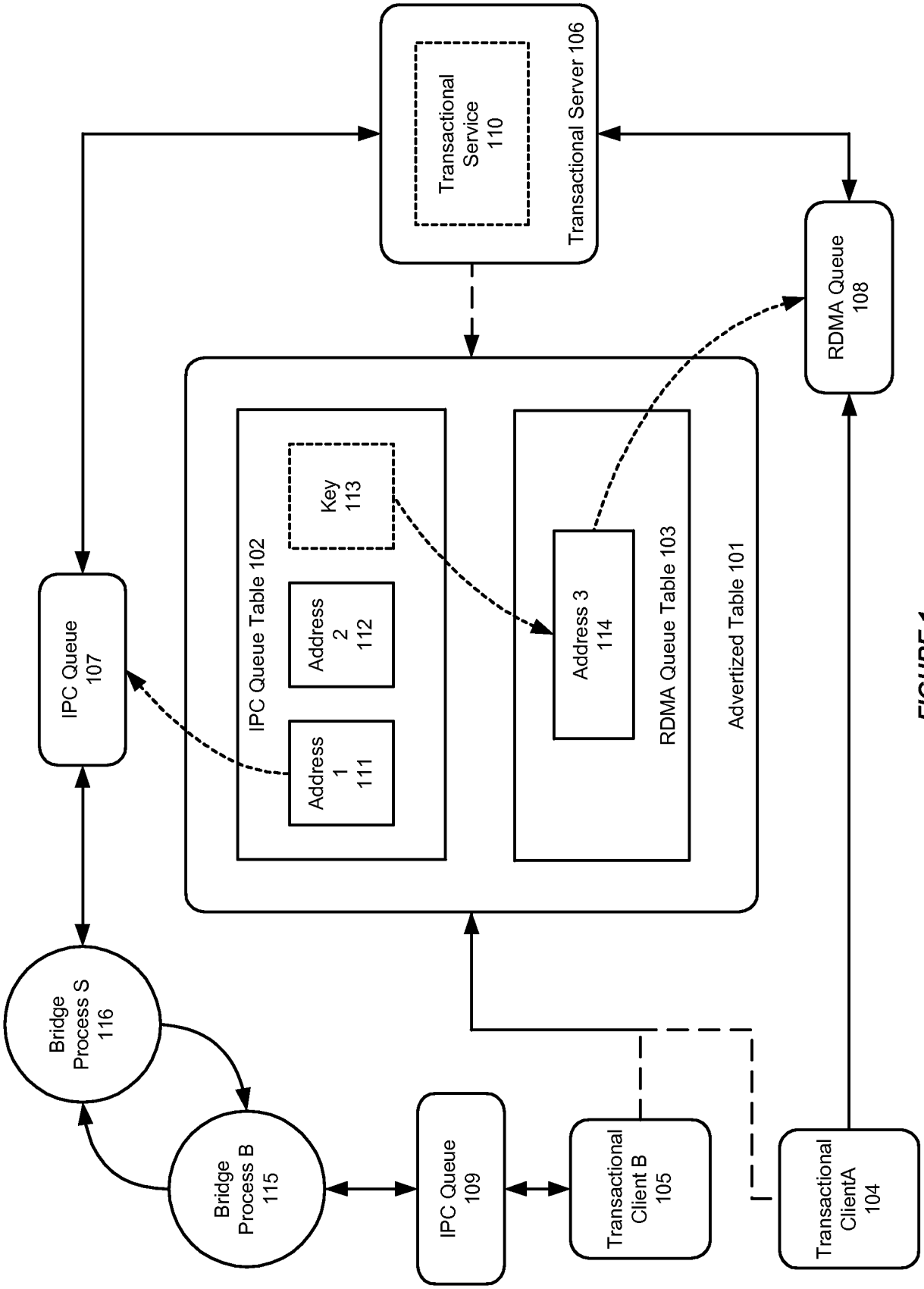
Transactional Server 106
Transactional Service 110
RDMA Queue 108
IPC Queue 107
IPC Queue Table 102
Key 113
Address 2 112
Address 1 111
RDMA Queue Table 103
Address 3 114
Advertized Table 101
Bridge Process S 116
Bridge Process B 115
IPC Queue 109
Transactional Client B 105
Transactional ClientA 104

FIGURE 1

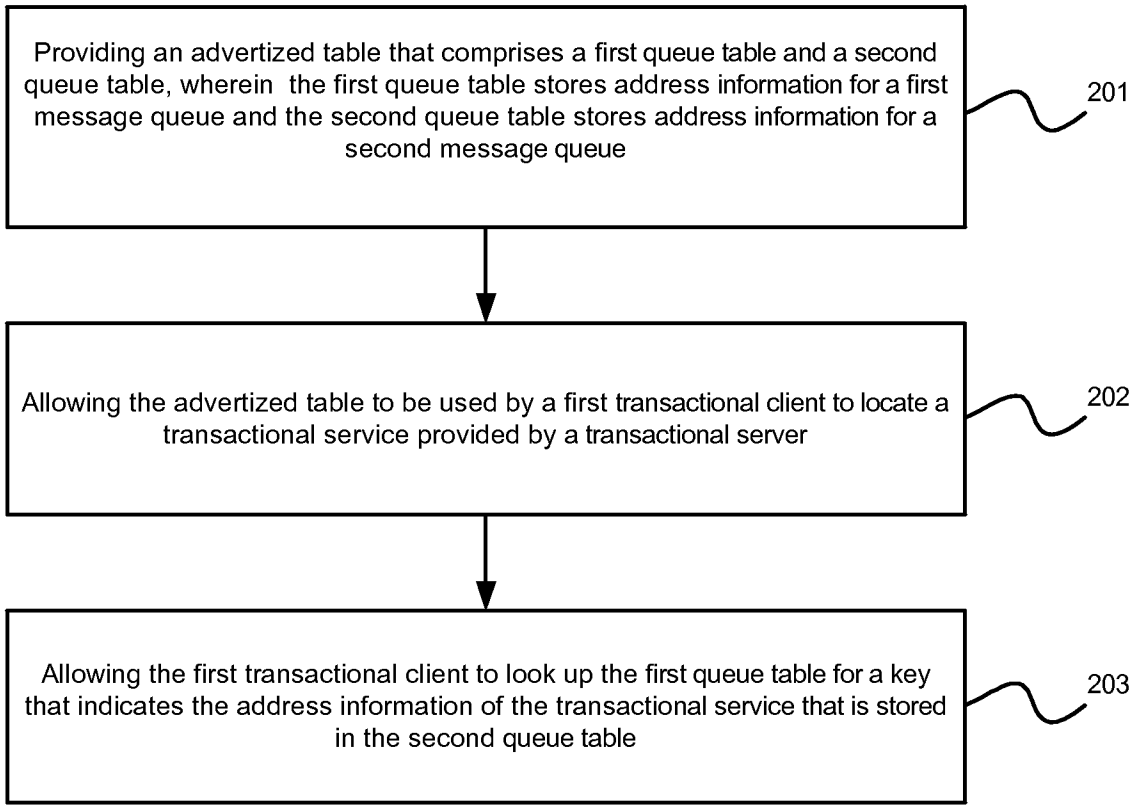
Providing an advertized table that comprises a first queue table and a second queue table, wherein the first queue table stores address information for a first message queue and the second queue table stores address information for a second message queue
201
Allowing the advertized table to be used by a first transactional client to locate a transactional service provided by a transactional server
202
Allowing the first transactional client to look up the first queue table for a key that indicates the address information of the transactional service that is stored in the second queue table
203

FIGURE 2

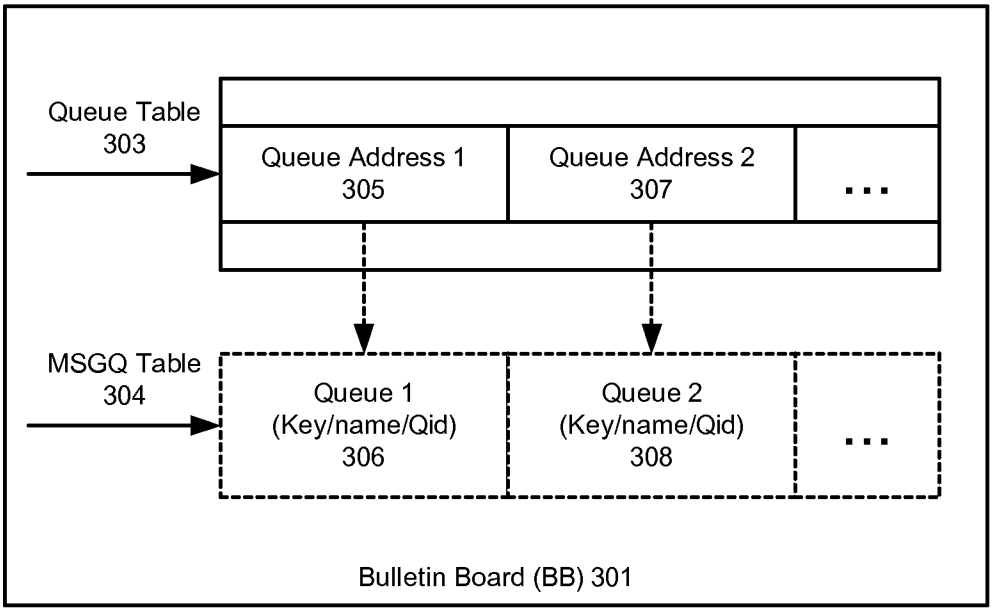

FIGURE 3

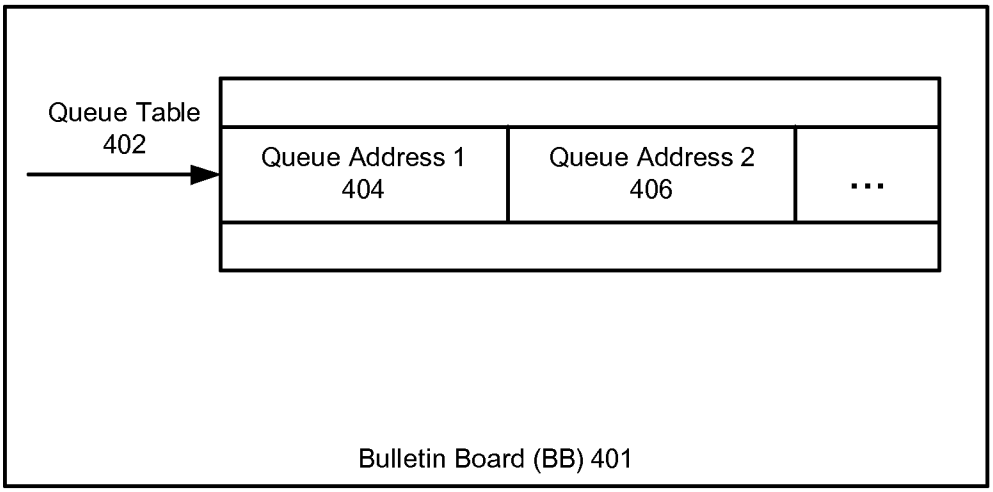

FIGURE 4

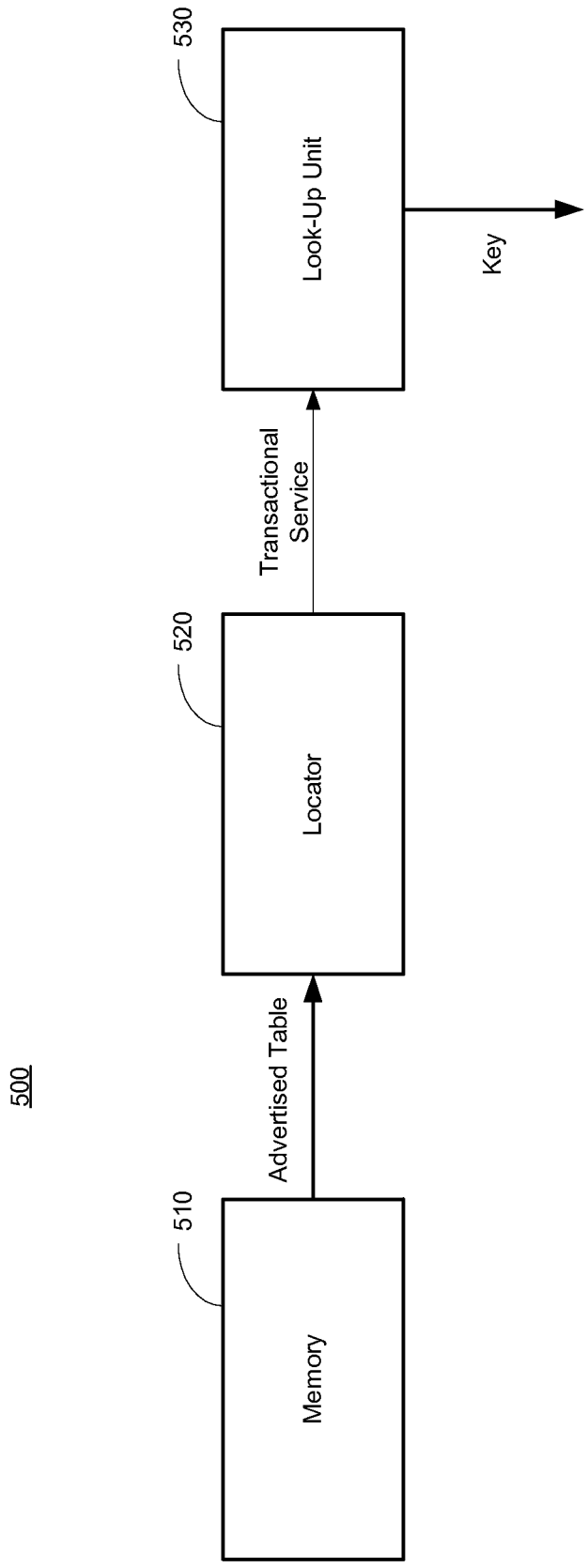
500
510
Memory
Advertised Table
520
Locator
Transactional Service
530
Look-Up Unit
Key

FIGURE 5

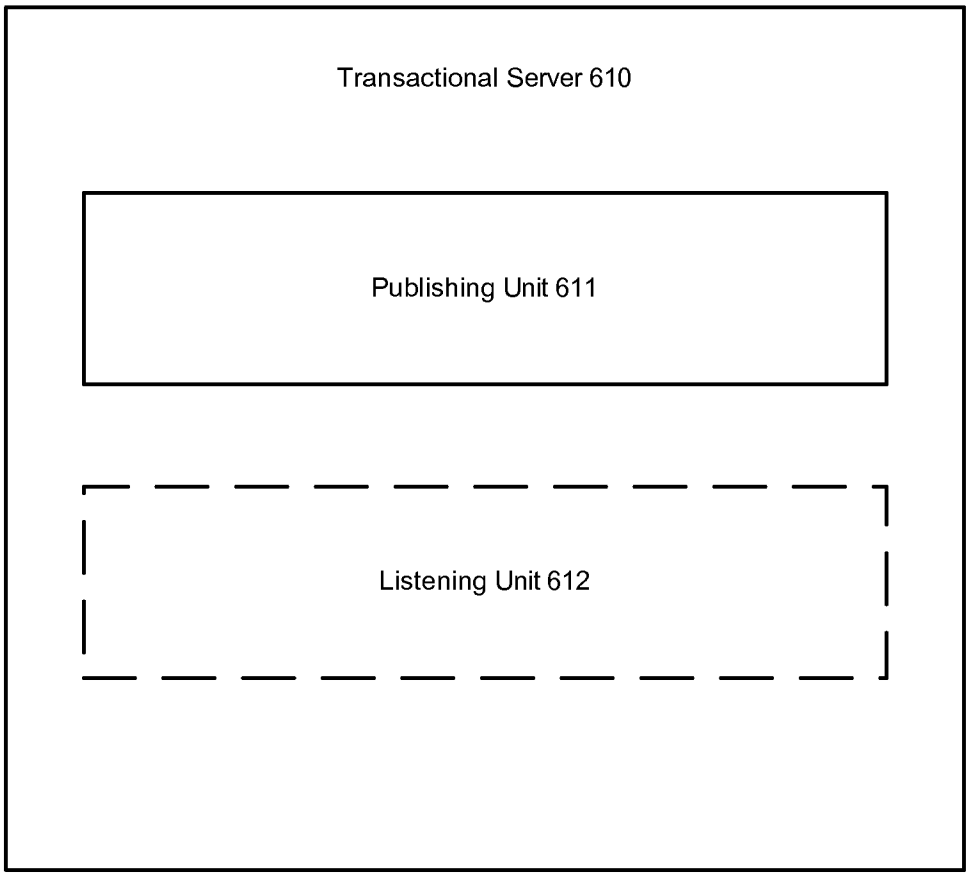
Transactional Server 610
Publishing Unit 611
Listening Unit 612

FIGURE 6

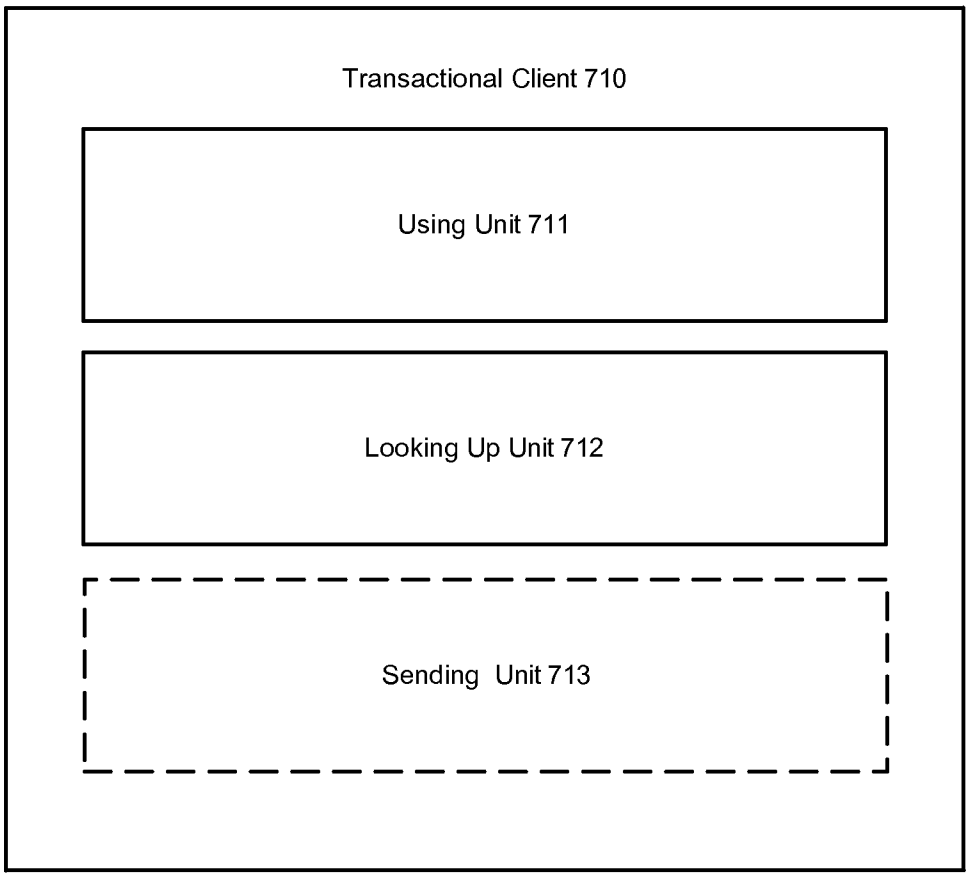
Transactional Client 710
Using Unit 711
Looking Up Unit 712
Sending Unit 713

FIGURE 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US12/56950

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 12/28 (2012.01)
USPC - 370/431, 412
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8): G06F 17/30, 15/16; H04L 12/28, 12/54 (2102.01)
USPC: 707/999.001; 370/401, 429, 412; 709/250

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
MicroPatent (US-G, US-A, EP-A, EP-B, WO, JP-bib, DE-C,B, DE-A, DE-T, DE-U, GB-A, FR-A); DialogPRO; IEEE/IEEEXplore; Google/Google Scholar; IP.com; RDMA, host adapter, InfiniBand, middleware, IPC, input, ingress, queue, output, egress, mapping, index, key, lookup, address, channel, destination, transaction, table

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7,990,994 B1 (YEH, J., et al.) August 2, 2011, figures 6B-6E, 8A, column 4, lines 31-42, column 6, lines 3-8, column 7, lines 1-4, column 21, lines 20-22, column 34, line 63, column 56, lines 9-19, column 58, lines 37-51, column 58, line 61, column 61, lines 45-47, column 62, lines 20-21, column 63, lines 36-39, column 64, lines 15-23, column 71, line 66 through column 72, line 7, column 74, lines 37-48	1, 2, 3/1, 3/2, 4/1, 4/2, 11, 12, 13/11, 13/12
A	US 2007/0183418 A1 (RIDDOCH, D., et al.) August 9, 2007, figures 15, 39-41, paragraphs [0008], [0021], [0096], [0107], [0108], [0115], [0118]-[0125], [0149], [0166]	1, 2, 11

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
7 November 2012 (07.11.2012)	20 NOV 2012
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US12/56950

Box No. II Observations where certain claims were found unsearchable (*Continuation of item 2 of first sheet*)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☒ Claims Nos.: 5-10 and 14-21
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where *unity of invention* is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.