US 20080222086A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2008/0222086 A1

(43) Pub. Date: Sep. 11, 2008

Meacham et al.

(54) **LIVE PROFILE**

(75) Inventors: **Paul Meacham**, Tinley Park, IL (US); **Jacob Doornebos**, Riverside, IL (US); **James P. Moran**, Hinsdale, IL (US)

Correspondence Address:
BANNER & WITCOFF, LTD.,
ATTORNEYS FOR CLIENT NO. 006119
10 SOUTH WACKER DRIVE, SUITE 3000
CHICAGO, IL 60606 (US)

(73) Assignee: **Chicago Mercantile Exchange, Inc.**, Chicago, IL (US)

(21) Appl. No.: **12/125,682**

(22) Filed: **May 22, 2008**

Related U.S. Application Data

(63) Continuation of application No. 11/696,555, filed on Apr. 4, 2007, which is a continuation-in-part of application No. 11/276,752, filed on Mar. 13, 2006, which is a continuation-in-part of application No. 11/234,697, filed on Sep. 23, 2005.

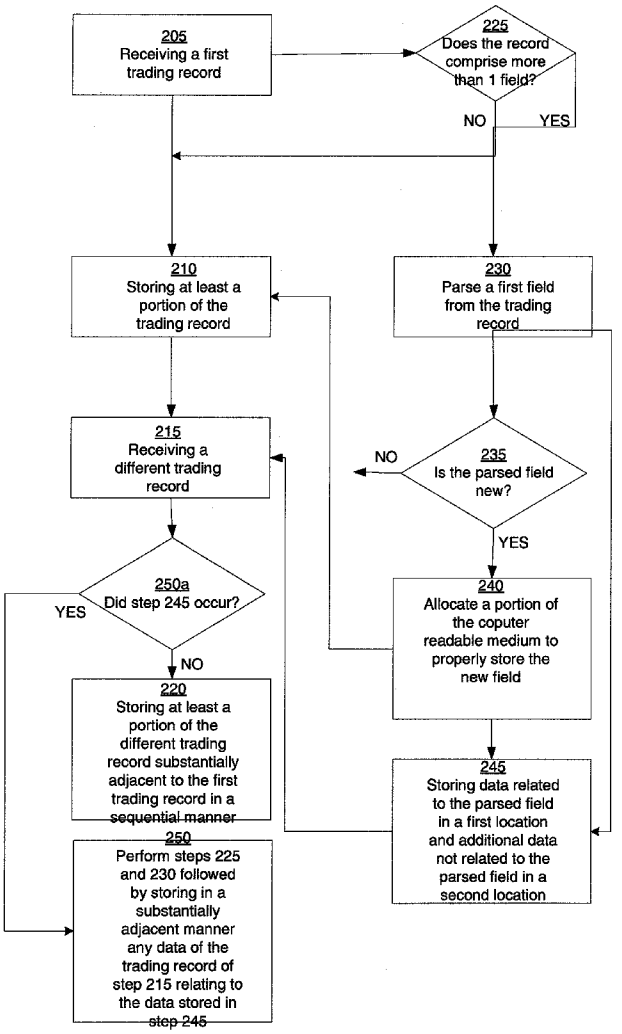
Publication Classification

(51) **Int. Cl.**
G06F 7/00 (2006.01)
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/1**

(57) **ABSTRACT**

Systems and methods for reconstructing the state of a market are provided. Orders are arranged as a non-indexed collection of orders and may be stored in the cache memory of a processor. The physical locations of orders stored in the memory may correspond to the order in which they were received at a match engine. A computer device simulates the processing of orders between any time periods to reconstruct the activity state of an entity across a trading platform and one or more order books.



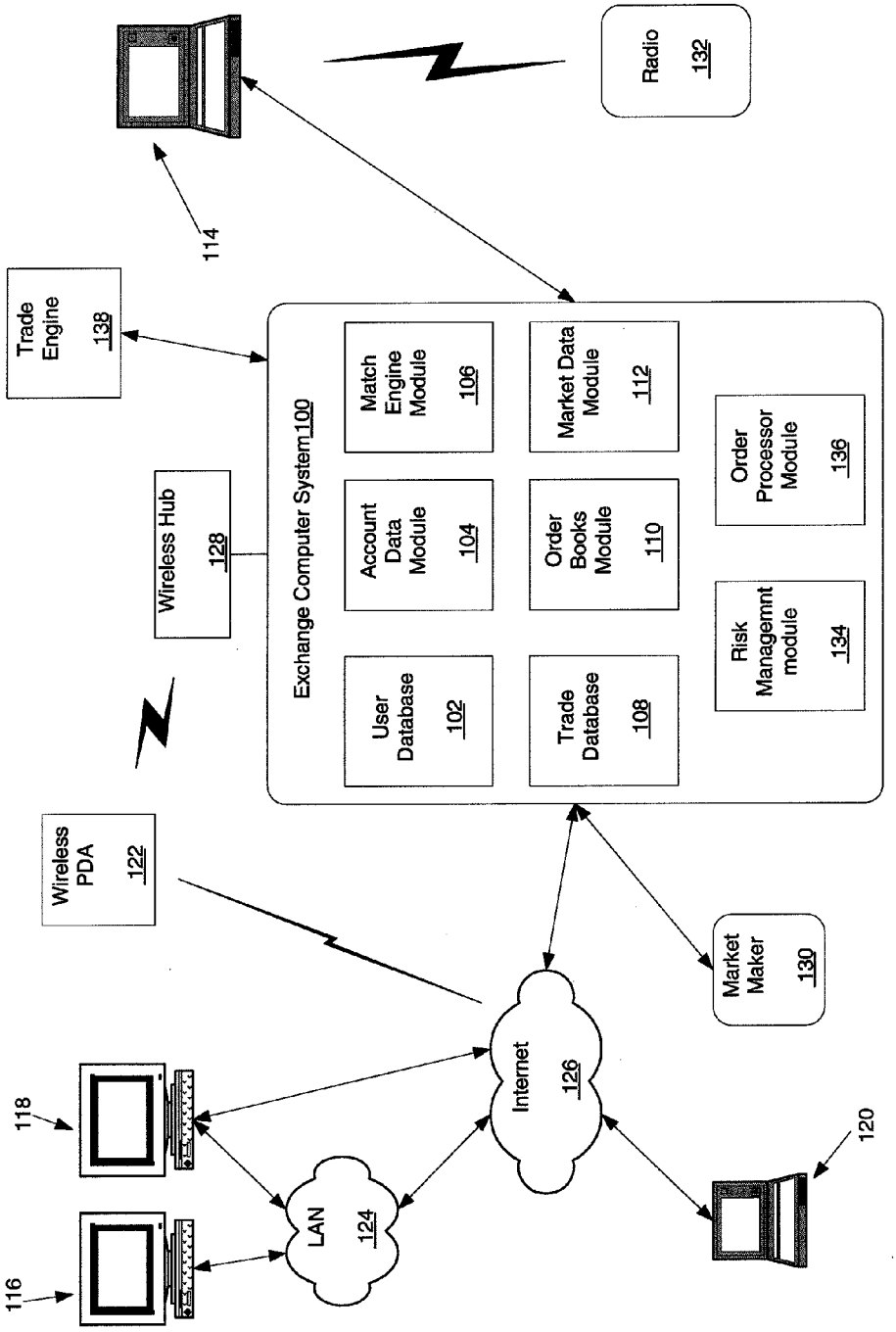

Radio
132
114
Trade Engine
138
Exchange Computer System100
Match Engine Module
106
Market Data Module
112
Account Data Module
104
Order Books Module
110
Order Processor Module
136
User Database
102
Trade Database
108
Risk Managemnt module
134
Wireless Hub
128
Wireless PDA
122
Market Maker
130
Internet
126
LAN
124
118
116
120

Figure 1

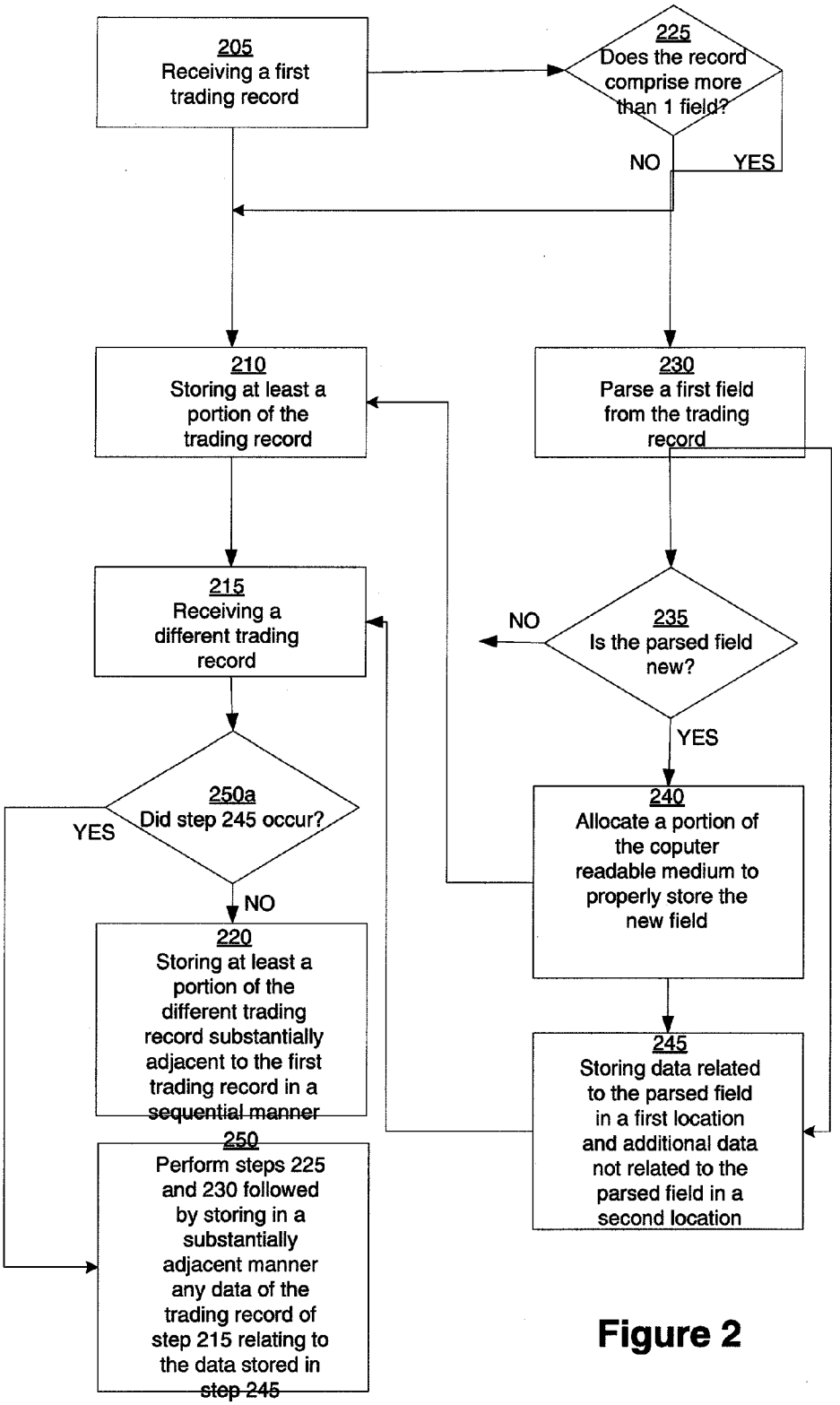

205
Receiving a first trading record
225
Does the record comprise more than 1 field?
NO
YES
210
Storing at least a portion of the trading record
230
Parse a first field from the trading record
215
Receiving a different trading record
235
Is the parsed field new?
NO
YES
250a
Did step 245 occur?
YES
NO
240
Allocate a portion of the coputer readable medium to properly store the new field
220
Storing at least a portion of the different trading record substantially adjacent to the first trading record in a sequential manner
245
Storing data related to the parsed field in a first location and additional data not related to the parsed field in a second location
250
Perform steps 225 and 230 followed by storing in a substantially adjacent manner any data of the trading record of step 215 relating to the data stored in step 245

Figure 2

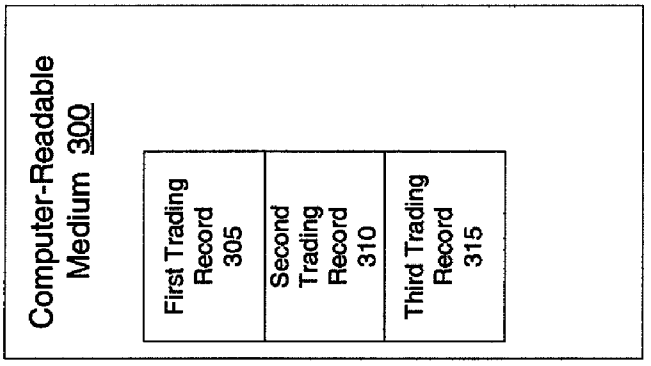


Figure 3a

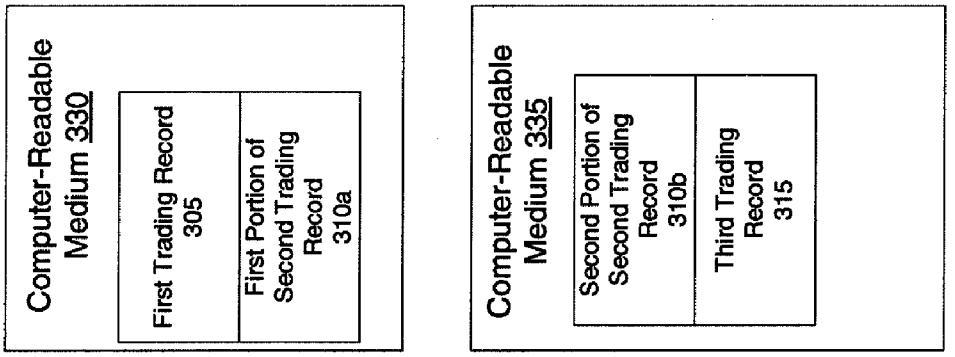


Figure 3b

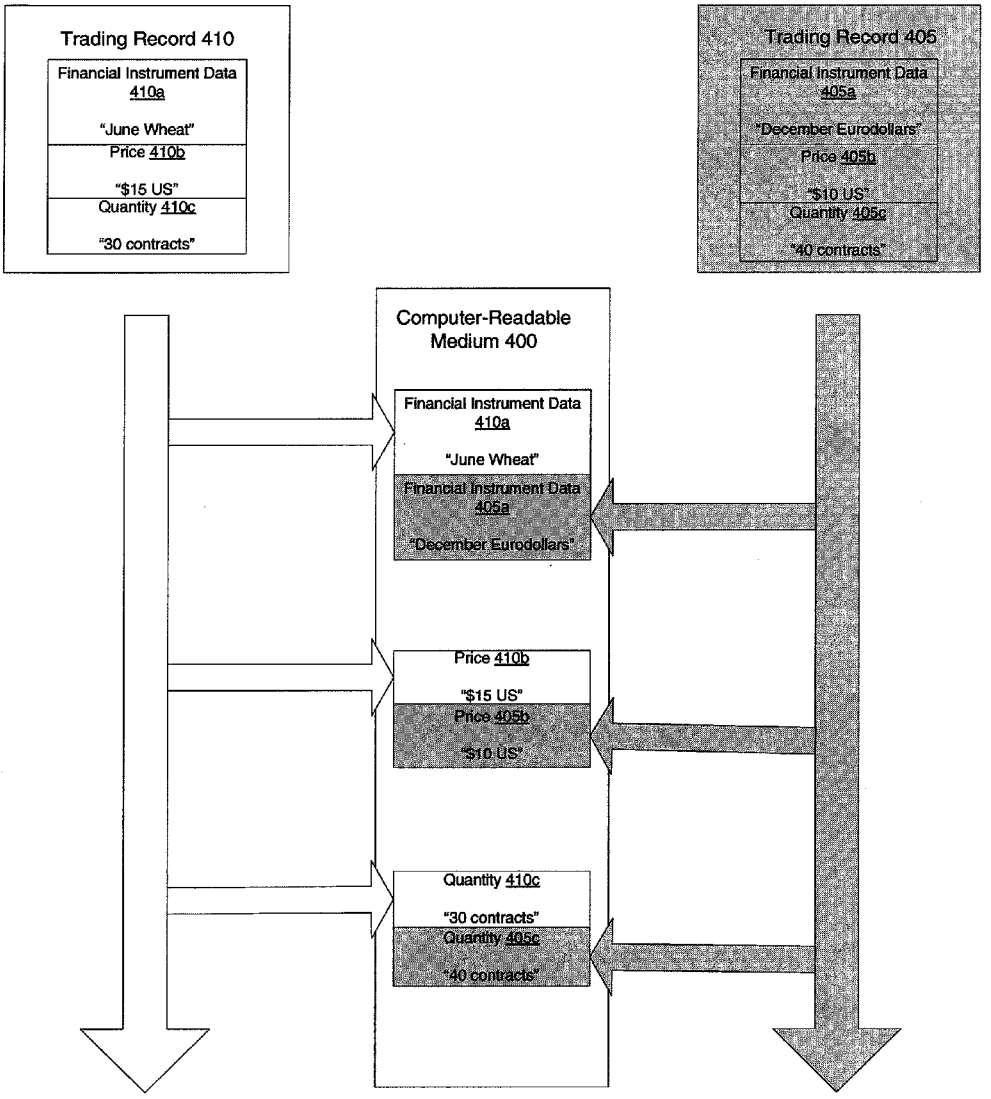

Trading Record 410
Financial Instrument Data 410a
"June Wheat"
Price 410b
"$15 US"
Quantity 410c
"30 contracts"
Trading Record 405
Financial Instrument Data 405a
"December Eurodollars"
Price 405b
"$10 US"
Quantity 405c
"40 contracts"
Computer-Readable Medium 400
Financial Instrument Data 410a
"June Wheat"
Financial Instrument Data 405a
"December Eurodollars"
Price 410b
"$15 US"
Price 405b
"$10 US"
Quantity 410c
"30 contracts"
Quantity 405c
"40 contracts"

Figure 4a

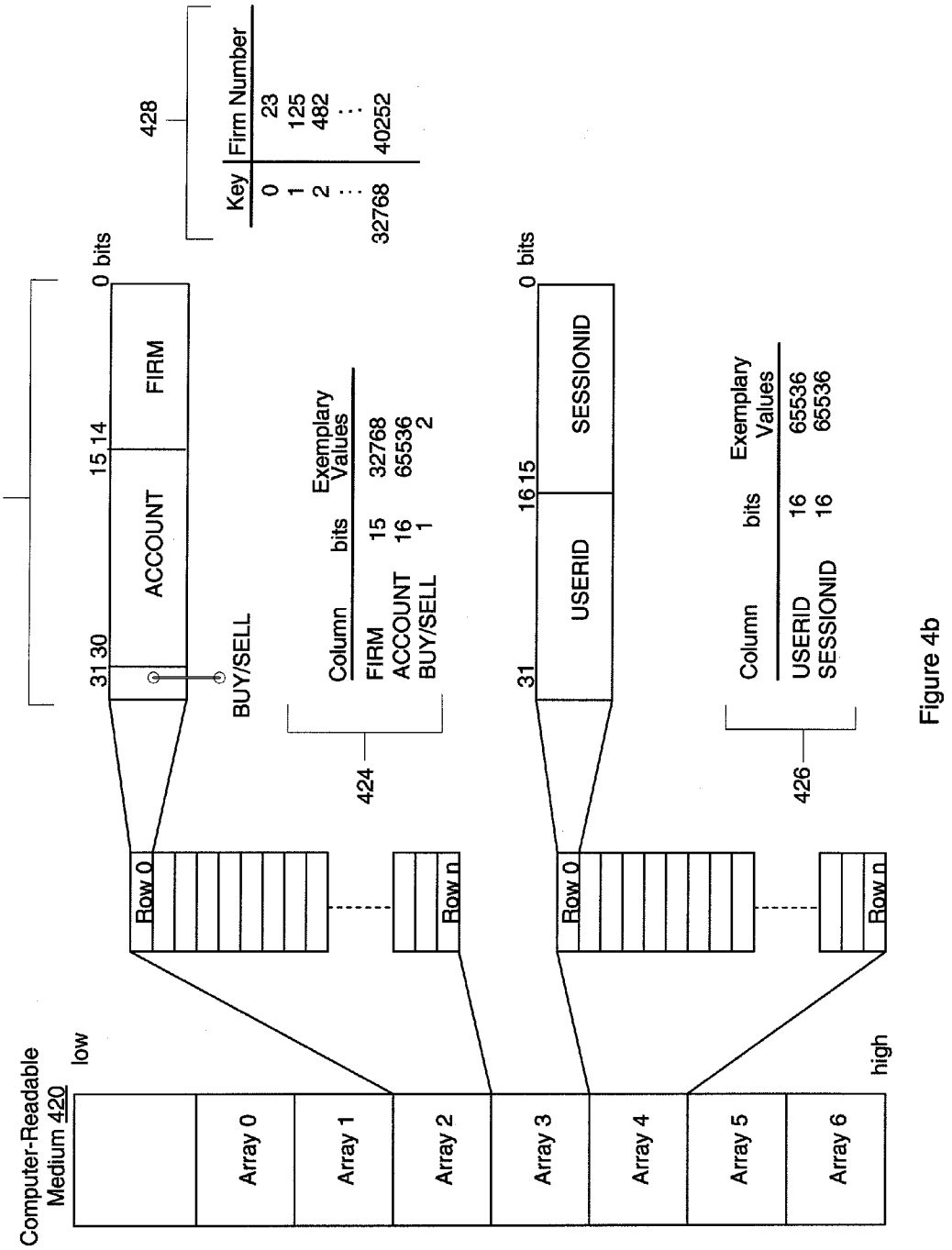
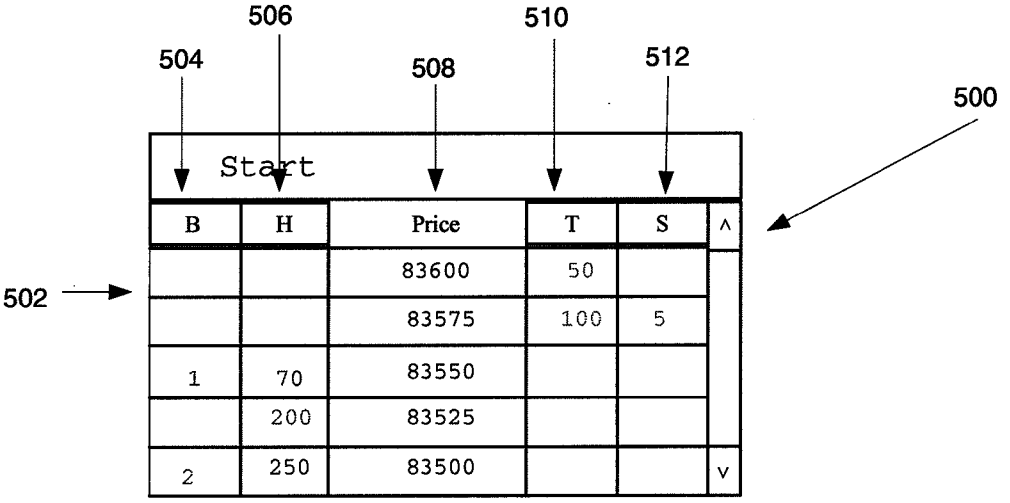

Computer-Readable Medium 420
low
high
Array 0
Array 1
Array 2
Array 3
Array 4
Array 5
Array 6
Row 0
Row n
422
0 bits
15 14
31 30
FIRM
ACCOUNT
BUY/SELL
428
Key
Firm Number
0 23
1 125
2 482
32768 40252
Column
bits
Exemplary Values
FIRM 15 32768
ACCOUNT 16 65536
BUY/SELL 1 2
424
16 15
31
SESSIONID
USERID
Column
bits
Exemplary Values
USERID 16 65536
SESSIONID 16 65536
426

Figure 4b


506
510
504
508
512
500
Start
B
H
Price
T
S
^
502
83600
50
83575
100
5
1
70
83550
200
83525
2
250
83500
v

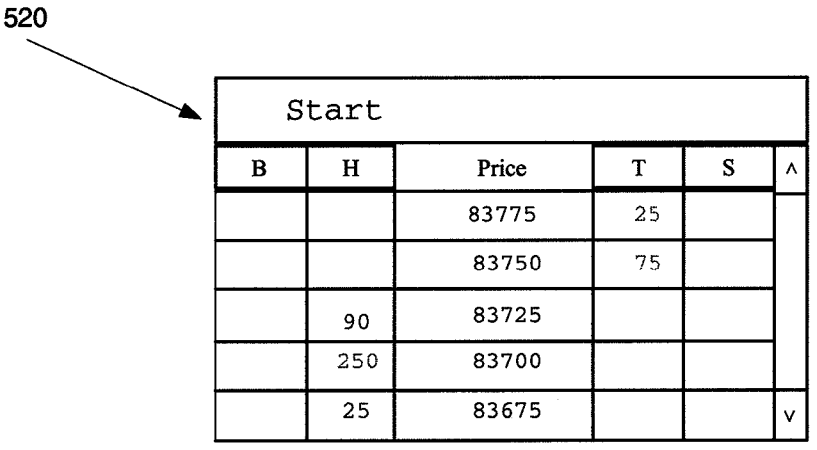

520
Start
B
H
Price
T
S
^
83775
25
83750
75
90
83725
250
83700
25
83675
v

Figure 5

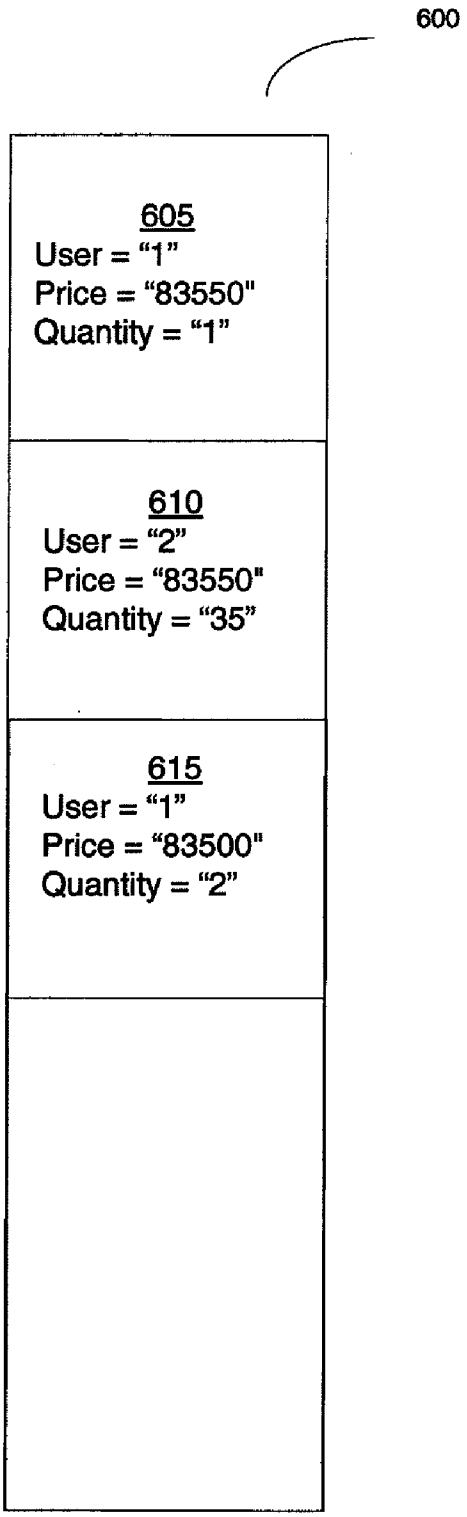

600
605
User = "1"
Price = "83550"
Quantity = "1"
610
User = "2"
Price = "83550"
Quantity = "35"
615
User = "1"
Price = "83500"
Quantity = "2"

Figure 6

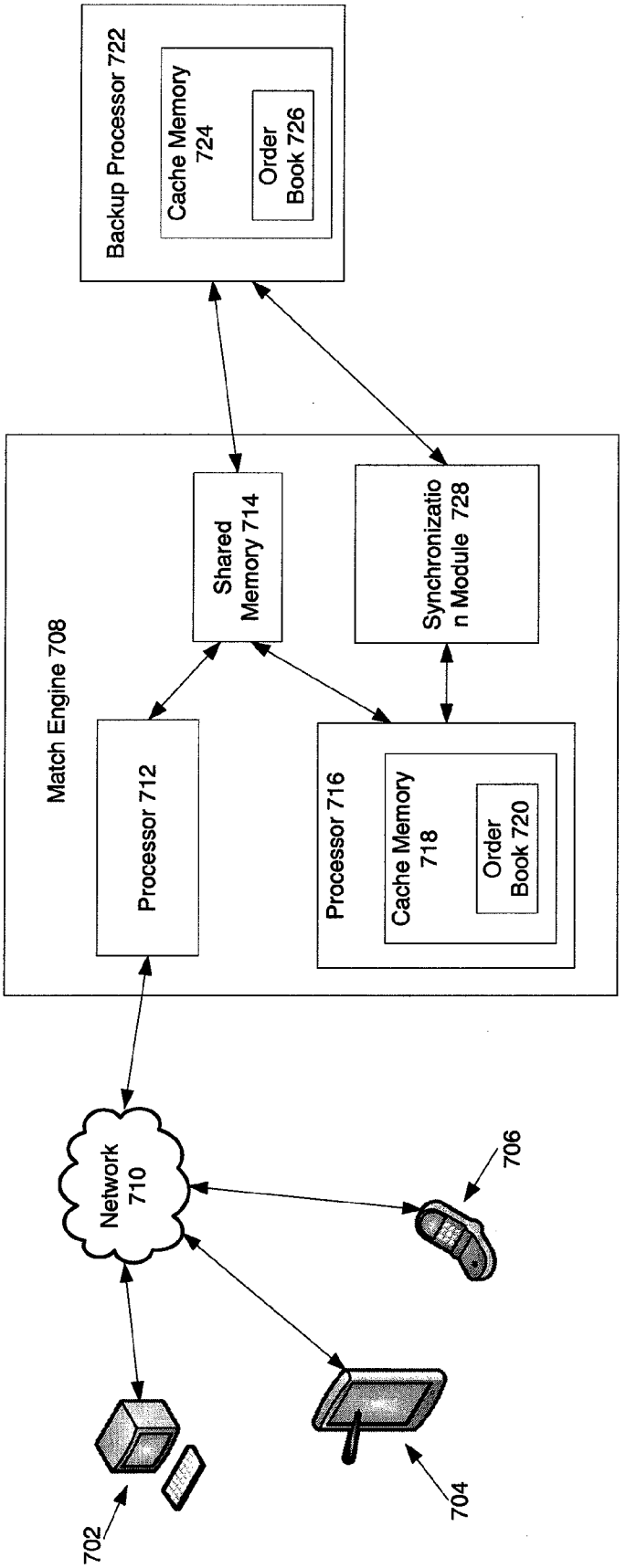

Backup Processor 722
Cache Memory 724
Order Book 726
Match Engine 708
Shared Memory 714
Synchronizatio n Module 728
Processor 712
Processor 716
Cache Memory 718
Order Book 720
Network 710
706
702
704

Figure 7

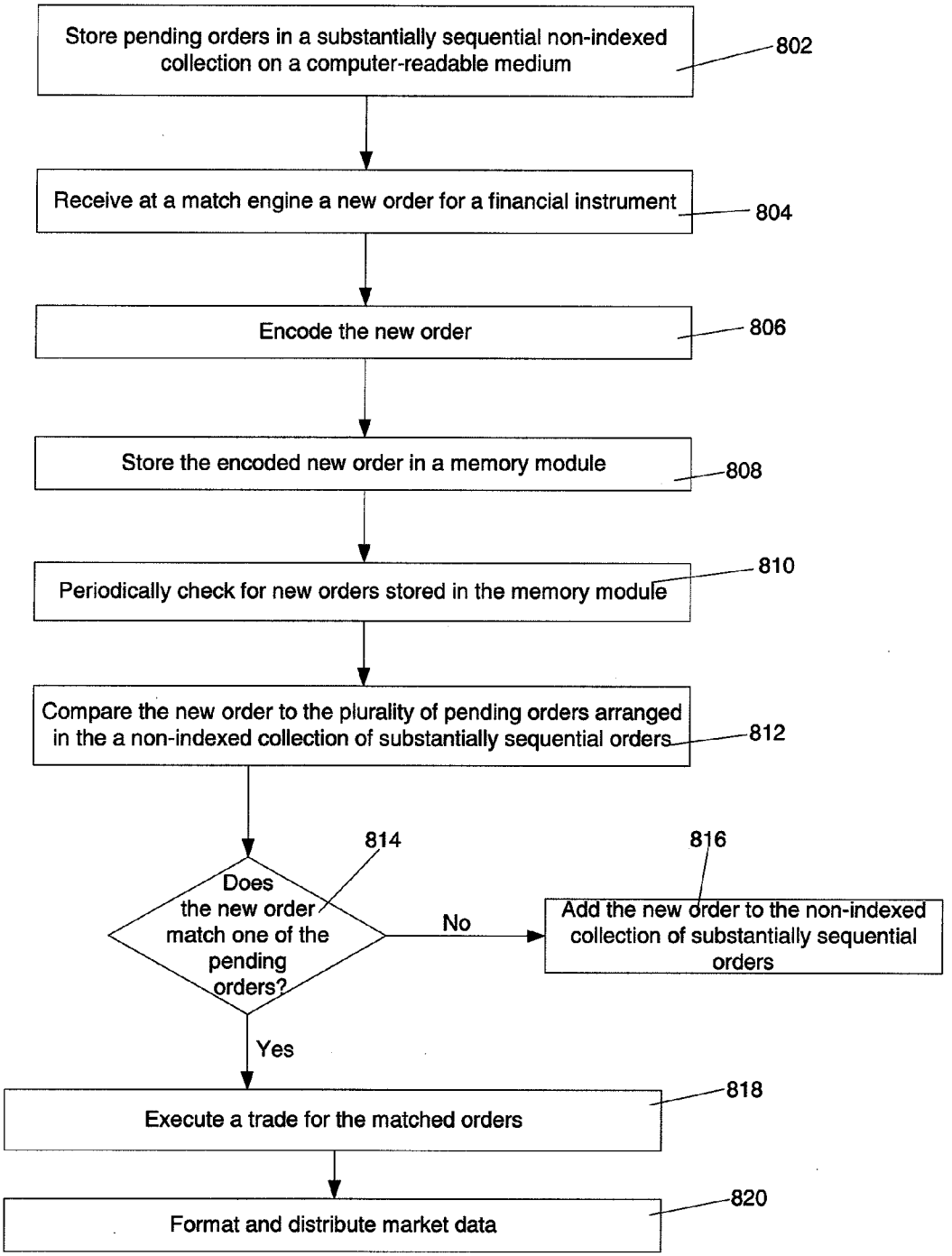

Store pending orders in a substantially sequential non-indexed collection on a computer-readable medium
802
Receive at a match engine a new order for a financial instrument
804
Encode the new order
806
Store the encoded new order in a memory module
808
Periodically check for new orders stored in the memory module
810
Compare the new order to the plurality of pending orders arranged in the a non-indexed collection of substantially sequential orders
812
814
Does the new order match one of the pending orders?
No
816
Add the new order to the non-indexed collection of substantially sequential orders
Yes
Execute a trade for the matched orders
818
Format and distribute market data
820

Figure 8

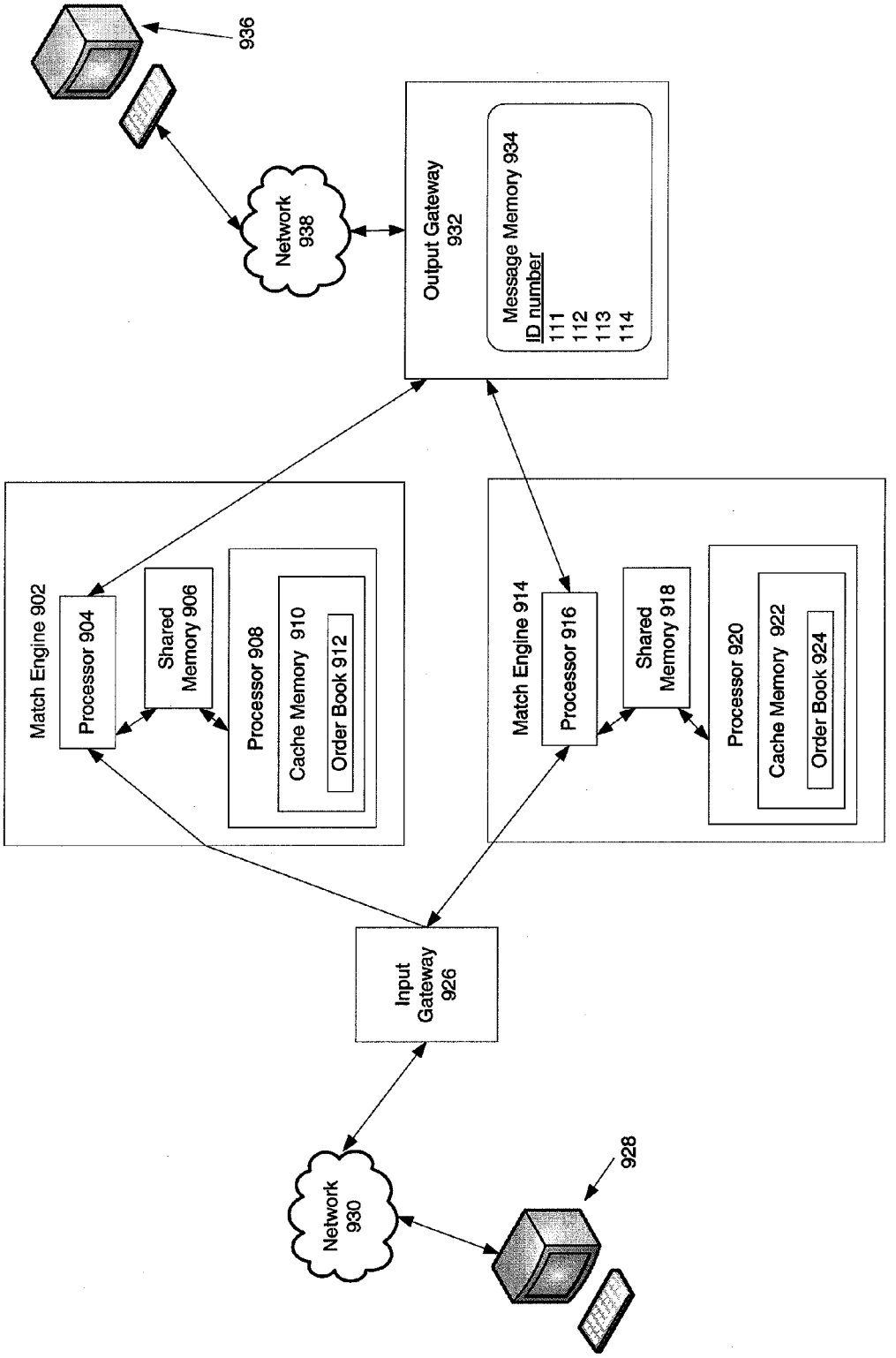

936
Network 938
Output Gateway 932
Message Memory 934
ID number
111
112
113
114
Match Engine 902
Processor 904
Shared Memory 906
Processor 908
Cache Memory 910
Order Book 912
Match Engine 914
Processor 916
Shared Memory 918
Processor 920
Cache Memory 922
Order Book 924
Input Gateway 926
Network 930
928

Figure 9

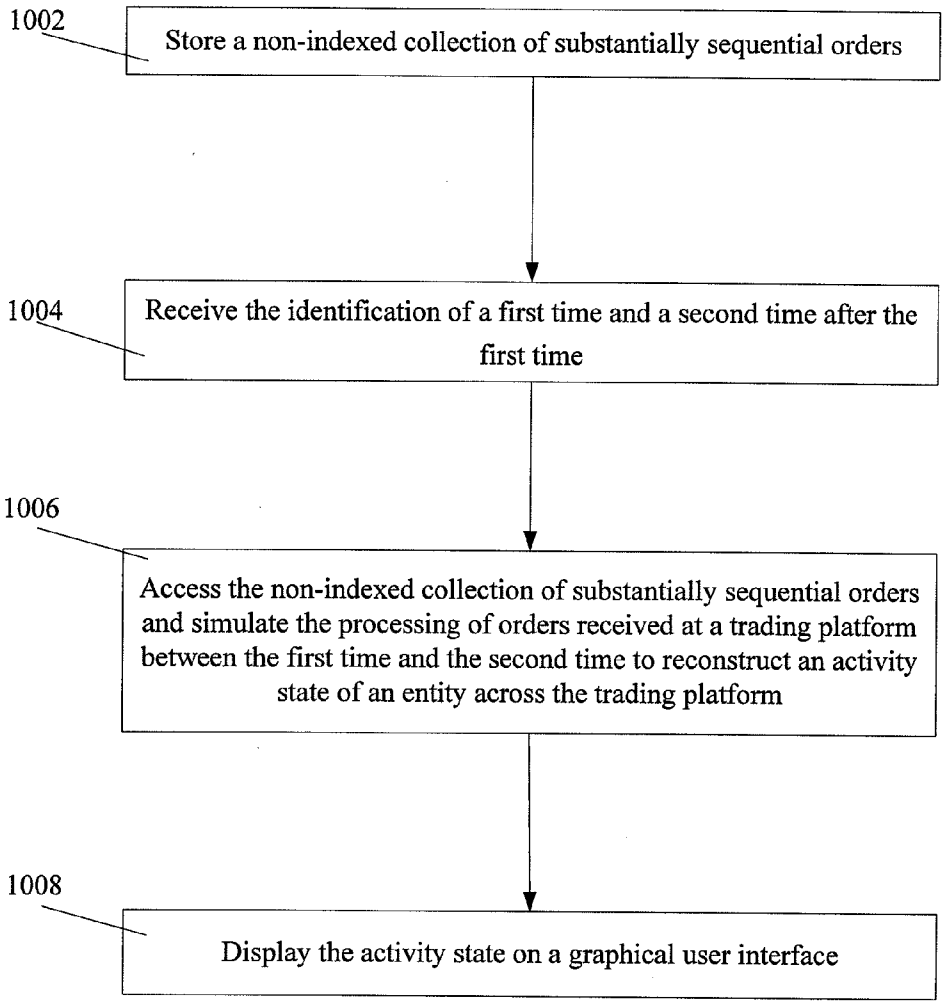

1002
Store a non-indexed collection of substantially sequential orders
1004
Receive the identification of a first time and a second time after the first time
1006
Access the non-indexed collection of substantially sequential orders and simulate the processing of orders received at a trading platform between the first time and the second time to reconstruct an activity state of an entity across the trading platform
1008
Display the activity state on a graphical user interface

Figure 10

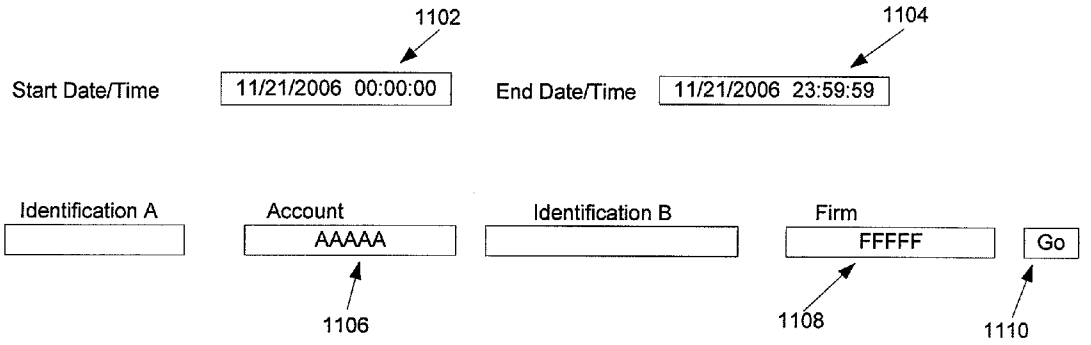


Figure 11

11/21/2006 00:00:00 - 11/21/2006 23:59:59
Account = "AAAAA" AND Firm = "FFFFF"

1202

1208

Instrument	Buy Vol	Sell Vol	Position	Last Price	P/L	Orders	Mods	Cancels	Quotes	VR
ESZ6	1730	1748	-18	140650.00	5625.00	266	4	115	0	12.88
NQZ6	8	8	0	181700.00	200.00	17	0	15	0	0.94

1206

1204

Figure 12

11/21/2006 00:00:00 - 11/21/2006 23:59:59

(Account = "AAAAA" AND Firm = "FFFFF") AND (Instrument = "ESZ6")

Executions

Buy	Sell	Net Pos	P&L	Last Price
296	298	-2	1450.00	140650.00

1302

Bid					Offer				
Price	Quant	UserID	SLE	Account	Price	Quant	UserID	SLE	Account
140600	2	UUUUU	SSSSS	AAAAA	140675	39	UUUUU	SSSSS	AAAAA
140575	7	UUUUU	SSSSS	AAAAA	140725	32	UUUUU	SSSSS	AAAAA

1304

Figure 13

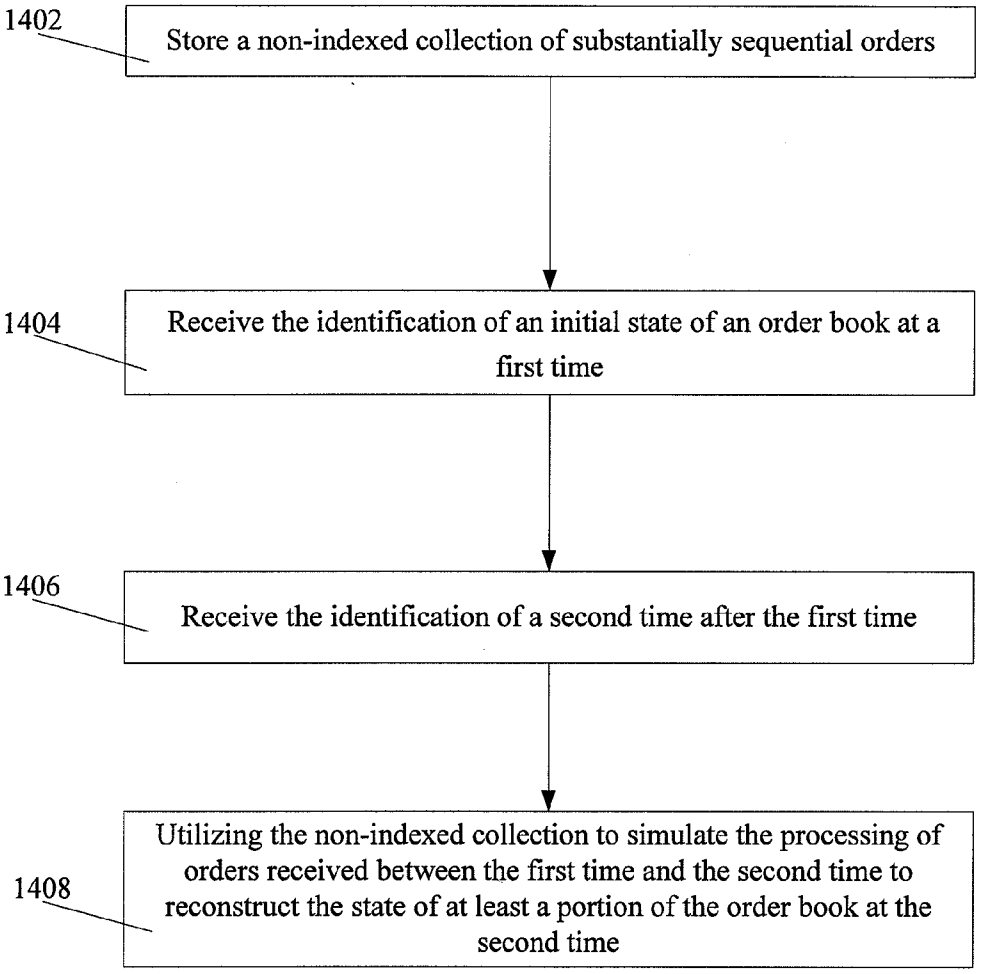

1402
Store a non-indexed collection of substantially sequential orders
1404
Receive the identification of an initial state of an order book at a first time
1406
Receive the identification of a second time after the first time
1408
Utilizing the non-indexed collection to simulate the processing of orders received between the first time and the second time to reconstruct the state of at least a portion of the order book at the second time

Figure 14

LIVE PROFILE

[0001] The present application is a continuation application of U.S. Ser. No. 11/696,555, filed Apr. 4, 2007 and entitled "Live Profile," which is a continuation-in-part application of U.S. Ser. No. 11/276,752, filed Mar. 13, 2006 and entitled "Match System That Uses A Non-Indexed Collection of Orders," which is a continuation-in-part application of U.S. Ser. No. 11/234,697, filed Sep. 23, 2005 and entitled "Non-Indexed In-Memory Data Storage and Retrieval." The entire disclosures of all of these applications are hereby incorporated in their entirety by reference herein.

FIELD OF THE INVENTION

[0002] The present invention relates to systems, methods and user interfaces that are utilized in connection with the trading of financial instruments. More particularly, a live profile provides mechanisms for recreating and displaying the bids and offers existing in a moment of time in a financial market.

DESCRIPTION OF THE RELATED ART

[0003] Modern financial institutions process and monitor a large volume of trading data, such as orders for financial instruments. Financial instruments include options contracts, futures contracts, options on futures contracts, interest rate products, securities, commodities, equity contracts, etc. Institutions that administer transactions in financial instruments process and store large amounts of trading data every second of the trading day. Moreover, upon executing a trade, processors continually access and distribute market data, which is a subset of trading data.

[0004] Large databases may be utilized to store and retrieve trading data. To select and aggregate trading data, conventional databases often use sorts, searches, indexes, and/or disc scanning techniques and lookups. These requirements result in a substantial number of chip or processing clock cycles and lead to delayed query results. Current analysis systems utilized to aggregate large quantities of trading data are often executed in batch mode overnight because of the computing resources that are consumed by these activities. Often the aggregation and retrieval of trading data is not sufficient to allow adequate information to be retrieved within the desired timeframe. Indeed, under traditional approaches, large amounts of trading data cannot be adequately analyzed in real-time, thereby preventing many uses of the data, such as for exchange risk analysis and regulation monitoring for risk and potential rule violations.

[0005] Prior art attempts to solve these problems have focused on building more intelligent indexes to speed up selection and analysis of the data stored within a database. Yet other systems have attempted to reduce response time to users through the use of precomputed summary data. These and other attempts to more efficiently store and retrieve trading data may not provide adequate solutions for many problems. For example, precomputed indexes cannot be rapidly adapted for changing user needs or changing data. Additionally, precomputed data requires the user to specify the data that needs to be precomputed. When there is a need to analyze data from different angles or perspectives, these conventional systems fail to deliver results in a rapid fashion.

[0006] Financial institutions, such as exchanges, may provide regulatory or market oversight to monitor trade activity. Regulatory or other oversight entities often analyze trading data for compliance with regulations and to ensure market and financial integrity. For example, market data representing a market order book may be analyzed when responding to complaints, reviewing trading for compliance with rules, conducting formal investigations, or checking on the financial risk of a market participant or firm.

[0007] A regulatory entity may try to determine whether a trader has a pattern of submitting and canceling large orders to create the appearance of demand in a market. Existing tools may provide periodic snapshots of the summary information for the top of the market, but since market bids and offers change rapidly and continuously, it is not practical to archive snapshots of every change for the entire book of orders showing the identity of the parties causing the changes. Systems that look at market data stored in databases may not be able to provide timely information for regulatory purposes. Such systems also have limited flexibility. As a result, the type of analysis that may be performed is limited and users cannot quickly create new queries to adapt to new situations.

[0008] Therefore, there exists a need in the art for systems and methods that allow for the rapid processing, storage searching, and sorting of large amounts of data in a time sensitive manner.

SUMMARY OF THE INVENTION

[0009] Aspects of a live profile overcome at least some of the above problems and limitations by providing one or more apparatuses, systems and methods for efficiently storing, sorting, displaying, recording, compiling and/or searching of large amounts of data.

[0010] A live profile may provide data and other information relating to a market participant's activity across a platform of a financial market, and/or provide data relating to a market book existing at some point in the past.

[0011] Information may be arranged as a non-indexed collection of data records within one or more computer-readable media. Exemplary computer-readable media include processor cache memories, magnetic memories, hard disk drives, electromagnetic memories, electronic memories, and optical disk drives. Solid-state memory modules allow for rapid queries due to the lack of moving parts, such as those associated with hard disk drives. The physical locations of information associated with pending orders stored in a computer-readable medium may correspond to the sequence in which the pending orders were received at a match engine, an exchange, or other financial institution. The state of a market may be reconstructed by simulating the processing of orders received between a first time and a second time.

[0012] Of course, the apparatuses, methods and systems disclosed herein may also include other additional elements, steps, computer-executable instructions, or computer-readable data structures. The details of these and other embodiments of the invention are set forth in the accompanying drawings and the description below. Other features and advantages will be apparent from the description and drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] A live profile may take physical form in certain parts and steps, embodiments of which will be described in detail in

the following description and illustrated in the accompanying drawings that form a part hereof, wherein:

[0014] FIG. **1** shows a computer network system that may be used to implement aspects of the invention;

[0015] FIG. **2** shows a flowchart of one illustrative method of storing trading data in a computing environment;

[0016] FIGS. **3***a* and **3***b* illustrates illustrative collections of trading data;

[0017] FIG. **4** illustrates yet another illustrative collection of trading data;

[0018] FIG. **4***b* illustrates the structure of an alternative collection of trading data;

[0019] FIG. **5** illustrates an exemplary graphical user interface that may be used to display market depth information;

[0020] FIG. **6** illustrates one exemplary collection of data arranged in a substantially sequential ordering;

[0021] FIG. **7** illustrates a system for matching trades;

[0022] FIG. **8** illustrates a method of processing orders with the system shown in FIG. **7**;

[0023] FIG. **9** illustrates a system for matching trades that uses redundant match engines;

[0024] FIG. **10** illustrates a computer implemented method of reconstructing an activity state of an entity across a trading platform;

[0025] FIG. **11** illustrates a graphical user interface that may be used to initiate a process to reconstruct an activity state of an entity across a trading platform;

[0026] FIG. **12** illustrates a display or a graphical user interface that indicates an activity state of an entity across a trading platform and that results from the input values provided in FIG. **11**;

[0027] FIG. **13** illustrates a graphical user interface that shows executed order data and pending order data; and

[0028] FIG. **14** illustrates a method for recreating a state of an order book.

DETAILED DESCRIPTION

Exemplary Operating Environment

[0029] Aspects of the invention are preferably implemented with computer devices, methods, systems and computer networks for exchange trading information. An exemplary trading network environment for implementing trading systems and methods is shown in FIG. **1**.

[0030] An exchange computer system **100** receives orders and transmits market data related to orders and trades to users. Exchange computer system **100** may be implemented with one or more mainframe, servers, gateways, desktop, handheld and/or other computers. In one embodiment, a computer device uses a 64-bit (or more) processor.

[0031] A user database **102** includes information identifying traders and other users of exchange computer system **100**. Data may include user names and passwords. An account data module **104** may process account information that may be used during trades. A match engine module **106** is included to match bid and offer prices. Match engine module **106** may be implemented with software that executes one or more algorithms for matching bids and offers. A trade database **108** may be included to store information identifying trades and descriptions of trades. In particular, a trade database may store information identifying the time that a trade took place and the contract price. An order book module **110** may be included to compute or otherwise determine current bid and offer prices. A market data module **112** may be included to collect market data and prepare the data for transmission to users. A risk management module **134** may be included to compute and determine a user's risk utilization in relation to the user's defined risk thresholds. An order processing module **136** may be included to decompose delta based and bulk order types for processing by order book module **110** and match engine module **106**.

[0032] The trading network environment shown in FIG. **1** includes computer devices **114**, **116**, **118**, **120** and **122**. Each computer device includes a central processor that controls the overall operation of the computer and a system bus that connects the central processor to one or more conventional components, such as a network card, such as an Ethernet card, or modem. Each computer device may also include a variety of interface units and drives for reading and writing data or files. Depending on the type of computer device, a user can interact with the computer with a keyboard, pointing device, microphone, pen device or other input device.

[0033] Computer device **114** is shown directly connected to exchange computer system **100**.

[0034] Exchange computer system **100** and computer device **114** may be connected via a T1 line, a common local area network (LAN) or other mechanism for connecting computer devices. Computer device **114** is shown connected to a radio **132**. The user of radio **132** may be a trader or exchange employee. The radio user may transmit orders or other information to a user of computer device **114**. The user of computer device **114** may then transmit the trade or other information to exchange computer system **100**.

[0035] Computer devices **116** and **118** are coupled to a LAN **124**. LAN **124** may have one or more of the well-known LAN topologies and may use a variety of different protocols, such as Ethernet. Computers **116** and **118** may communicate with each other and other computers and devices connected to LAN **124**. Computers and other devices may be connected to LAN **124** via twisted pair wires, coaxial cable, fiber optics or other media. Alternatively, a wireless personal digital assistant device (PDA) **122** may communicate with LAN **124** or the Internet **126** via radio waves. PDA **122** may also communicate with exchange computer system **100** via a conventional wireless hub **128**. As used herein, a PDA includes mobile telephones and other wireless devices that communicate with a network via radio waves.

[0036] FIG. **1** also shows LAN **124** connected to the Internet **126**. LAN **124** may include a router to connect LAN **124** to the Internet **126**. Computer device **120** is shown connected directly to the Internet **126**. The connection may be via a modem, DSL line, satellite dish or any other device for connecting a computer device to the Internet.

[0037] One or more market makers **130** may maintain a market by providing constant bid and offer prices for a derivative or security to exchange computer system **100**. Exchange computer system **100** may also exchange information with other trade engines, such as trade engine **138**. One skilled in the art will appreciate that numerous additional computers and systems may be coupled to exchange computer system **100**. Such computers and systems may include clearing, regulatory and fee systems.

[0038] The operations of computer devices and systems shown in FIG. **1** may be controlled by computer-executable instructions stored on computer-readable medium. For example, computer device **116** may include computer-executable instructions for receiving order information from a user and transmitting that order information to exchange

computer system **100**. In another example, computer device **118** may include computer-executable instructions for receiving market data from exchange computer system **100** and displaying that information to a user.

[0039] Of course, numerous additional servers, computers, handheld devices, personal digital assistants, telephones and other devices may also be connected to exchange computer system **100**. Moreover, one skilled in the art will appreciate that the topology shown in FIG. **1** is merely an example and that the components shown in FIG. **1** may be connected by numerous alternative topologies.

EXEMPLARY EMBODIMENTS

[0040] FIG. **2** shows a flowchart of one illustrative method of storing trading data in a computing environment. A first trading record having trading data is received in step **205**. The trading record may be an order received at an exchange, a market data distributed by an exchange or a trade. Alternatively or in addition, the trading record may be a record created by an exchange that represents an order book for one or more financial instruments, the state of an order book, orders received and processed for one or more financial instruments, or any other information that may be used, processed, delivered to, and/or received by an exchange or entity involved with trading, administering, monitoring and/or facilitating financial transactions. The trading data may include one or more specific identifiers of an order or trade, including: the time and date of the order or trade, the individual or firm that submitted an order, price information and/or the identification and quantity of financial instruments. The trading record may include an order first transmitted from computer device **116** and/or PDA **122** through LAN **124** and/or Internet **126** and may be further processed by order processing module **136**. The trading record received in step **205** additionally or alternatively may include market data distributed by one or more of the modules within Exchange Computer System **100**. Other fewer, and/or additional fields may be included within, associated with or represented by a trading record.

[0041] In step **210**, at least a portion of the data from the trading record received in step **205** is stored on a computer readable medium. As used herein, a computer-readable medium may include, for example, random access memory (RAM), dynamic random access memory (DRAM), flash memory, a hard disk drive, thumb drive, and/or an optical disk (CD ROM, DVD or other optical media). Solid-state memory modules allow for rapid queries due to the lack of moving parts, such as those associated with hard disk drives. The computer-readable medium may be integral with the Exchange Computer System **100** and/or may be associated with one or more modules, such as the match engine module **136**.

[0042] FIG. **3***a* illustrates a collection of trading data. As seen in the figure, a memory module **300**, such as a RAM or optical disk, may store a plurality of trading records. For example, the trading record stored in step **205** may be visually represented as trading record **305**. In step **215**, a second trading record may be received. The second trading record may be of a fixed length and have substantially the same information as the first trading record received in step **205**. Yet in additional or alternative embodiments, the second trading record may be of a different length than the first trading record, may have additional fields not present in the first record and/or may include additional data not present in the first record. In one embodiment, optional step **225** (explained in more detail below) may be implemented to determine if the trading record comprises more than one field.

[0043] In step **220**, the second trading record (or a portion thereof) is sequentially stored in relation to the first trading record **305**, wherein the data from the second trading record is stored substantially physically adjacent to the data received from the first trading record **305**. One of the advantages of locating records close to one another is faster reading times. Existing database systems may have data scattered throughout a memory device. Reading data arranged in this manner is time consuming because the reading process has to skip from one physical location to another physical location. For example, a hard disk drive must physically move a reading head from location to location. FIG. **3***a* shows one illustrative embodiment where data from the second trading record **310** is stored in a sequential fashion in relation to the data stored from the first trading record **305**. Unlike conventional databases, the trading data stored and as represented by **305** and **310** are not associated with an index for locating the data. As additional data and/or trading records are received, they may be stored in a substantially sequential ordering. As used herein "substantial sequential ordering" means that data is physically positioned on a computer-readable medium in a direction that a read operation will follow and does not imply that a sort operation is performed on the records or data before storing. For example, data **315** is stored substantially sequential to data **310**, which is stored substantially sequential to data **305**. When memory module **300** is implemented with a hard disk drive, for example, a reading head may follow a straight path to read records **305**, **310** and **315**.

[0044] While an exemplary embodiment may not have an index as used in conventional databases, the collection of data as represented by first trading records **305**, **310** and **315** can perform row selection at the same speed as an indexed database. Indeed, in some embodiments, eliminating conventional databases indexes results in queries that have the same speed for all columns in the table. Conventional databases typically have an index on every column. This results in deleterious effects on the insertion speed. Conversely, a collection of data organized according to the various embodiments of the present invention allows for rapid insertion speeds and is particularly useful and advantageous in real-time insertion situations, such as those routinely encountered in the trading industry. Moreover, by providing a collection of data without an associated database-type index, more space is available on the computer readable medium to store data, such as that present in the data of the first trading record and second trading record (**305**, **310**). An increase in data storage may be achieved by eliminating the use of a conventional database-type index. In at least one implementation, the elimination of an index may double the amount of data that may be stored on the computer readable memory.

[0045] As one skilled in the art will appreciate, a computer-readable medium may have multiple linked drives and/or modules, where the data is stored in a sequential fashion as described above. The drives and/or modules may be at different physical or remote locations. For example, FIG. **3***b* shows two distinct physical computer-readable media (**330**, **335**) that are configured to store data in a sequential ordering process. First trading record **305** is stored in the first portion of computer-readable medium **330**, sequentially followed by a first portion of second trading record **310***a*. Since the trading record consisting of section **310***a* and **310***b* is too large to be

entirely stored on computer-readable medium **330**, the second portion of the second trading record **310***b* is stored on the first portion of computer-readable medium **335**. Trading record **315** is also stored on computer-readable medium **335** and substantially follows the second portion of the second trading record **310***b*. In this regard, the trading records **305**, **310**, and **315** are sequentially stored on one or more computer-readable media arranged in a sequential order. The computer-readable media, such as computer-readable media **330** and **335**, may or may not be physically ordered in a sequential manner, but rather may be configured to be sequentially store data.

[0046] As briefly mentioned above, one or more trading records may include multiple fields. Optional step **225** may be implemented to determine if one or more of the trading records includes more than one field. Upon determining that more than one field exists, step **230** may optionally be initiated to parse a first field from one of the trading records. As shown in FIG. **4**, data **405**, which may be similar to trading record **305** is received and it is determined through a process, such as process **225**, that it comprises multiple fields. Optional step **230** may be initiated to parse the data into a plurality of fields. Data **405** may be parsed into at least three fields, wherein financial instrument data **405***a* includes an identification of the financial instrument, such as an option contract to purchase wheat in June, price data **405***b* comprises information on the price of the contract, and quantity data **405***c* may include information on the quantity, such as quantity of financial instruments. More or fewer fields may be utilized in various embodiments of the invention. Optional step **235** also or alternatively may be implemented to determine if a field present in received data, such as trading record **405** includes a new field, wherein upon the detection of a new field, a portion of the computer readable medium may be allocated to store data associated with the new field (step **240**). While step **225** is shown in relation to receipt of the first trading record, the step may be utilized upon receipt of any of the trading records.

[0047] In one embodiment, data associated with a first field of a trading record, such as financial instrument data **405***a* may be stored in a first location on the computer readable medium (step **245**). Data associated with other fields of the same record is not stored substantially sequential to financial instrument data **405***a*, but may be placed on the computer readable memory at a different location (see exemplary embodiment of step **250**). Upon receiving another trading record, such as trading record **410**, it too may be parsed into a plurality of data associated with different fields. For simplicity, FIG. **4** shows trading record **410** subdivided according to the same fields as record **405**. As shown in the figure, step **250** sequentially stores the data having a matching field (**410***a* comprises information relating to the field of "financial instrument") on the computer-readable memory, wherein data of the first field **405***a* that was initially stored in step **245** is substantially physically adjacent to the data of the first field **410***a* that was stored in step **245**.

[0048] Other matching fields of different trading records, such as trading records **405** and **410** may also be stored in the same manner. For example, price fields **405***b* and **410***b* are stored sequentially wherein price data **410***b* is substantially sequential to and follows price data **405***b*. (see also; quantity data **405***c* and **410***c*). The data may be stored sequentially as it is received and in at least one embodiment is stored in real time allowing for the fast storage and manipulation of the data without having to construct and update a database-type index. Moreover, while the illustrated computer-readable memory of FIG. **4** illustrates a single medium, one skilled in the art will realize a plurality of computer readable media could be utilized to achieve the same aspects of the invention.

[0049] FIG. **4***b* illustrates the structure of an alternative or additional collection of trading data. A computer-readable medium **420** includes a plurality of arrays that contain trading data. Each array includes a particular type of data for a plurality of records. For example, array **2** includes buy/sell, account and firm data for a plurality of trading records and array **4** includes userID and sessionID data. Each record is assigned to a unique row. Section **422**, which represents row **0** of array **2** includes 32 bit rows. Bits **0-14** identify the trading firm, bits **15-30** identify the account and bit **31** is used to identify data as corresponding to a buy or a sell. Section **424** illustrates exemplary values for the fields identified in array **2** and section **426** illustrates exemplary values for the fields identified in array **4**.

[0050] Keys may be used to represent values and facilitate further processing. The keys utilize table based compression to remove gaps in data. Section **428** shows exemplary firm number keys. The firm field in section **422** includes 15 bits, which allows for 32,768 unique keys. The actual firm numbers used may include gaps. For example, section **428** shows that the first firm number used is "28" and the second firm number used is "125." When assigning keys, the keys may be arranged in sequential order corresponding to the sequential order of the firm numbers to facilitate searching. Section **428** shows that the lowest firm number of "28" is assigned a key value of "0" and the next highest firm number of "125" is assigned a key value of "1."

[0051] Arranging keys in the same order as firm numbers allows for the rapid searching of firm numbers within a specified range. Of course keys for some or all of the additional fields, such as account number, sessionID, userID, price, date and time, may also be arranged in sequential order, alphabetical order or other orders that facilitate searching. When new records are added, keys may be reassigned to ensure that the order of the keys corresponds to the order of the data.

[0052] A query on a computer readable medium, such as computer readable mediums having data stored in accordance with several or all of the steps and embodiments discussed in regards to FIGS. **2-4** may be performed. A query may relate to a method of processing an order received at a match engine. An order for at least one financial instrument may be received at a match engine, which may be implemented, for example, with match engine module **106**. At least one parameter or field may be extracted from the order. Upon extracting the at least one field or parameter, the field(s) that were extracted are compared to fields and/or parameters within a non-indexed collection of data representing pending orders.

[0053] For example, looking back to FIG. **3***a*, where trading records **305**, **310** and **315** represent pending orders, the comparison may be initiated at the upper starting portion of record **305** and proceed through trading records **305**, **310**, and **315** in a sequential manner based on proximate physical location of the records. Since there is no database-type index, the data within trading records **305**, **310** and **315** may be analyzed from different angles or perspectives at a more rapid pace than utilizing conventional database structures. Indeed, in some situations certain fields of data are unlikely to have data to meet the query being searched. For example, if the query relates to the quantity of financial instrument fields, a query

against data located in currency fields is unlikely to yield useful information in many cases. Searching a collection of data arranged such that records or fields are physically located next to one another in a memory module in the direction of a read operation of the search allows for faster query execution when compared to queries performed on indexed databases having records or fields distributed throughout a memory module.

[0054] Yet in other embodiments, distinct portions of trading records may be queried and/or a query may be directed to selected portions of trading records. For example, the pending orders may be organized as set forth in FIG. **4**, wherein a plurality of trading records are stored wherein matching fields of different trading records, such as trading records **405** and **410** are stored in a substantially sequential fashion without an associated database-like index. For example, price fields **405***b* and **410***b* are stored sequentially and wherein price field **410***b* is substantially sequential to and follows price field **405***b*. According to one embodiment of the invention, only data categorized as having certain fields will be queried. This determination may be based on a user-input, an algorithm, or a combination of user preference and a function of one or software applications.

[0055] This can be more readily seen when reviewing FIG. **4**. If a query is directed towards the price of a pending order, the comparison of the query and the data stored on the computer readable medium may be initiated at beginning of price data **405***b* and proceed in a substantially sequential fashion to price data **410***b* and through any additional price data located following price data **410***b*. The searching and comparison of the price data within the compared data will be performed without the use of a database-like index and will go in the sequential manner as described above. As stated above, the individual trading records, such as **405** and **410** as well as the individual fields of data such as **405***a* and **405***b* may be of a fixed length, thereby allowing a user and/or computer device to readily and accurately estimate the time to conduct the query of the trading records and/or individual data fields.

[0056] The speed at which queries may be performed when trading records are arranged as described above may be taken advantage of for other exchange and trading related activities. For example, a match engine may match trades using aspects of the inventions.

[0057] FIG. **5** illustrates an exemplary graphical user interface **500** that may be used to display market depth information and allow traders to trade financial instruments. The figure is helpful to understand the type and amount of information that represents the state of a market. Graphical user interface **500** includes a price and quantity grid **502**. Price and quantity grid **502** may contain multiple sections, which as shown in FIG. **5** include five columns. A buy column **504** displays a user's working buy order quantities. As used herein, a user may be a trader. Each user will have different values in this column reflecting their buy order quantity. A hit column **506** displays the market bid quantities. Prices for individual rows are displayed in a price column **508**. A take column **510** displays market ask quantities. And, a sell column **512** displays a user's working sell order quantities. Individual entries may be color coded to assist users in quickly interpreting the displayed information. For example, entries in buy column **504** and hit column **506** may be in blue and entries in take column **510** and sell column **512** may be in.

[0058] A trading firm, exchange or other entity may record trading records in a non-indexed collection of data, as described above. The speed at which such a collection may be queried and processed allows such entities to quickly recreate the state of an order book, or portions thereof, for any time period. For example, an initial state of the order book may first be determined and then all of the orders placed at an exchange may be processed in the same manner that they would be processed by an exchange until the desired point in time. For example, graphical user interface **500** (shown in FIG. **5**) may represent an initial state of a market. All of the incoming orders received at an exchange may be stored sequentially in one or more memory modules as a non-indexed collection of orders such that the physical location of the orders corresponds to the order in which they were received. A computer device may then be programmed to retrieve the orders and recreate the state of the order book.

[0059] Graphical user interface **520** represents the state of a order book at some time after the state represented with graphical user interface **500**. If we assume that graphical user interface **500** represents the state that existed at 9:00 on Monday morning and graphical user interface **520** represents the state that existed at 2:00 on the following Wednesday, the state represented with graphical user interface **520** may be recreated by starting with the state representing with graphical user interface **500** and processing orders in the sequential non-indexed collection of orders received until 2:00 on Wednesday.

[0060] Unlike conventional indexed databases storing and retrieving trading data according to one or more methods of the present invention does not require large quantities of trading data to be executed in batch mode overnight. Indeed, under traditional approaches, large amounts of data could not be adequately analyzed in real-time, thereby preventing many uses of the data. Under select embodiments of the invention, the analysis of the data sequentially stored on the computer readable memory can be continually processed in real-time to monitor activity while new data is being written to the computer readable medium, all without having to create, update, and maintain a space-consuming database index and constant interruption to jump physical locations within the computer readable medium to locate a certain data piece.

[0061] FIG. **6** illustrates one exemplary sequential non-indexed collection of orders stored on a computer-readable medium **600**. As seen in the figure, computer readable medium **600** includes multiple orders. Seen at the upper end of computer readable medium **600** is order **605**. Order **605** may be, for example, any of the quantities and/or prices displayed in the price and quantity grid **502**. Order **610** may represent the next order in time received at an exchange and order **615** may represent the next order in time received at the exchange. One skilled in the art will appreciate that intervals between the receipt of orders **605**, **610** and **615** may not be uniform.

[0062] Recreating market conditions may be readily accessible by querying methods, for example, as described above. Indeed, by following one or more embodiments of the invention, the analysis of the data sequentially stored on the computer readable memory can be continually processed in real-time to monitor activity while new data is being written to the computer-readable medium, all without having to create, update, and maintain a space-consuming database index and constant interruption to jump physical locations within the computer readable medium to locate a certain data piece.

[0063] FIG. **7** illustrates a system for matching trades. Computer devices **702**, **704** and **706** may be used to transmit

orders for financial instruments to a match engine **708** via a network **710**. Network **710** may be implemented at least in part with the Internet, a WAN, a LAN, a phone network or other infrastructure used for exchanging data between computer devices. Incoming orders are received at a processor **712**. Processor **712** may encode incoming orders and transmit them to a shared memory **714**. Encoding may include reducing the size of the order. For example, an order may be received at processor **712** that includes the name of a trading firm. During the encoding process, the trading firm name may be replaced with a number that represents the trading firm.

[0064] A second processor **716** may be configured to periodically check shared memory **714** for new orders. Processor **716** may include a cache memory **718** that includes one or more order books, such as order book **720**. The physical locations of pending orders stored in cache memory **718** may correspond to the order in which they were received at the match engine. When a new order is received, the order may be compared to orders that are included in order book **720**. The use of two processors **712** and **716** allows for fast operation when matching of trades. In alternative embodiments, the functions performed by processors **712** and **716** may be performed by a single processor and/or by three or more processors. An exemplary processor that may be used with aspects of the invention is an Intel Itanium II, which contains a 9 MB byte cache memory. Another exemplary processor is the AMD Opteron processor, which utilizes Hypertransport™ technology.

[0065] After trades are matched, market data may be generated by processor **716** and then transmitted to shared memory **714**. Processor **712** may periodically check shared memory **714** for market data messages. When market data messages are received, processor **712** may format or expand the market data message and then distribute the market data message to trading entities, such as those represented by computer devices **702**, **704** and **706**. Processor **712** may also distribute other information to trading entities, such as acknowledgement messages.

[0066] The speed at which processor **716** may process orders and the low-cost of such processors facilitates the use of redundant components and backup mechanisms. For example a backup processor **722** may include a cache memory **724** that includes an order book **726**. Order book **726** may be synchronized with order book **720**, such that in the event that processor **716** fails, backup processor **722** may resume matching of trades. A synchronization module **728** may be used to synchronize order books **720** and **726**. In one embodiment, processor **716** and backup processor **722** transmit information identifying the states of order books **720** and **726** to synchronization module **728**. Synchronization module **728** may then compare the states of the order books and make any adjustments that are necessary.

[0067] Match engine **708** may include several different processors that are configured to match a variety of different trades. Shared memory **714** may group new orders such that each of the processors knows which order to process. For example, a first processor may maintain a first order book and match trades against the first order book and a second processor may maintain a second order book for a different financial instrument and match trades for those financial instruments.

[0068] Backup processor **722** may be included within match engine **708**. Alternatively or in addition, backup processor **722** may be connected to match engine **708** via a local area network or wide-area network. Backup processor **722** may be in a different geographic location than processor **716**. For example processor **716** may be located within a first premises and backup processor **722** may be located in a different premises to prevent all processors from failing because of a fire or other event at the first premises. Two or more processors may also be geographically distributed and may be configured to process orders originating from different geographic regions. For example, processor **716** may be located in Chicago and may process orders originating in the United States and a second processor may be located in London in may be configured to process orders originating in that region.

[0069] FIG. **8** illustrates a method of processing orders with the system shown in FIG. **7**, in accordance with an embodiment of the invention. First, in step **802** pending orders are stored in a substantially sequential non-indexed collection on a computer readable medium. The computer readable medium may include a cache memory of a processor to facilitate rapid processing of incoming orders. Next, in step **804** a new order for a financial instrument is received at a match engine. The new order may be encoded in step **806**. Encoding may include reducing the file size of the order and placing the order into a standard format that is recognized by components of the match engine. Next, in step **808** the encoded order may be stored in a memory module. The memory module may include a RAM memory that is accessible by more than one processor. In step **810**, a processor may periodically check for new orders stored in the memory module. Steps **806** and step **810** may be performed by the same or different processors.

[0070] In step **812** the new order is compared to the plurality of pending orders arranged in the non-indexed collection of substantially sequential orders. It is then determined whether the new order matches one of the pending orders in step **814**. When the new order does not match one of the pending orders, in step **816** the new order is added to the non-indexed collection of substantially sequential orders. Step **816** may include adding the new order to an existing order book. If the new order does match one of the pending orders, a trade for the matching orders may be executed in step **818**. Finally, in step **820** market data may be formatted and distributed.

[0071] FIG. **9** illustrates a system for matching trades that uses redundant match engines. A first match engine **902** includes a first processor **904**, a shared memory **906** and a second processor **908**. Processor **908** includes a cache memory **910** that may include one or more order books, such as order book **912**. A second match engine **914** includes a first processor **916**, a shared memory **918** and a second processor **920**. Processor **920** includes a cache memory **922** that may include one or more order books, such as order book **924**. The components within match engines **902** and **914** function similar to corresponding components located within match engine **708** (shown in FIG. **7**).

[0072] The match system shown in FIG. **9** includes an input gateway **926**. Input gateway **926** receives orders, assigns identifying information to the orders and distributes the orders to match engines **902** and **914**. Input gateway **926** may be implemented with a computer device configured to route data and assign identifying information. In operation computer device **928** may transmit an order for a financial instrument to input gateway **916** via network **930**. Network **930** may be implemented at least in part with a WAN, LAN, phone network or other infrastructure used for exchanging data

between computer devices. After receiving the order, input gateway **926** may assign an identification number to the order and transmit the order to match engines **902** and **914**. In alternative embodiments of the invention, additional match engines may be included and input gateway **926** may transmit orders to some or all of the match engines.

[0073] In the embodiment shown, match engines **902** and **914** receive orders from input gateway **926** and process those orders in parallel. Match engines **902** and **914** include the same data in order books **912** and **924** and are configured to produce the same results. Output messages from processors **904** and **916** are transmitted to an output gateway **932**. Output gateway **932** may be implemented with a computer device configured to route data. In one embodiment of the invention input gateway **926** and output gateway **932** are implemented with the same computer device. Output gateway **932** may include a message memory **934** that stores information identifying messages that have been transmitted by output gateway **932**. At least part of the identifying information may be assigned by input gateway **926**. The identifying information may be used to make sure that data for an event is only distributed once. For example, input gateway **926** may assign and identification number to a new order and then transmit the order to match engines **902** and **914**. Match engines **902** and **914** may process the order in parallel and generate market data messages. The market data messages include the identification number assigned by input gateway **926**. When market data messages are received at output gateway **932**, output gateway may be configured to search identification numbers stored in message memory **934** to determine if a corresponding market data messages has been transmitted. When a message is received that includes an identification number not stored in message memory **934**, the message is distributed to trading entities, such as to computer device **936** via network **938**. When a message is received that includes an identification number that is stored in message memory **934**, the message may be discarded.

[0074] One skilled in the art will appreciate that a variety of different protocols may be used to assign identifying information to data received at input gateway **926** and then filter data at output gateway **932**. A new order may be assigned an identification number at input gateway **926** and derivative identification numbers may be assigned to messages associated with that order. For example, a new order may be assigned a **15** digit identification number and an acknowledgment message may be assigned an identification number that consists of the 15 digit identification number followed by a character or number that identifies the type of message. This allows multiple messages associated with a single order to all be assigned unique identification numbers that are related and ensures that match engines **902** and **914** assign the same identification numbers to created messages.

[0075] In the event of a failure of one of match engines **902** or **914**, output gateway **932** would receive messages from only one match engine and continue to process messages without any failure being apparent to trading entities. In embodiments that include more than two match engines operating in parallel, the failure of one or two match engines would not be apparent to trading entities. Match engines **902** and **914**, as well as any additional match engines, may be located in the same location or may be distributed to prevent a fire, network failure or other catastrophic event from halting the operation of all match engines.

[0076] The speed at which incoming orders may be processed when pending orders are arranged in a substantially sequential non-indexed collection in the cache memory of a processor allows for the processing of many types of orders. Existing match engine systems limit the type of orders that traders may make. For example, it not practical to process an order that does not have a standard format, such as an order to buy four particular contracts and sell six other contracts, because of the time required to match a single novel order having several legs to several different orders.

[0077] With the system described above the processing of orders can be quickly performed, which allows for many types of orders. For example, a trader may enter a nonstandard order that contains multiple legs in different markets. The systems shown in FIGS. **7** and **9** may rapidly attempt to match all of the legs of the nonstandard order without incurring reductions in throughput that would be incurred with conventional systems. Match engines **708**, **902** and **914** may be programmed to perform implied pricing functions for nonstandard orders. Alternatively, a trader may provide prices for one or more legs of a nonstandard order.

[0078] The speed at which orders may be processed when they are arranged in a substantially sequential non-indexed collection allows one to quickly reconstruct an activity state of an entity across the trading platform and/or reconstruct the state of an order book at any given time. Traders, trading firms and exchange regulatory or enforcement divisions may wish to reconstruct the state of a market for a variety of purposes. FIG. **10** illustrates a computer implemented method of reconstructing an activity state of an entity across a trading platform, in accordance with an embodiment of the invention. The trading platform may include a single exchange, multiple exchanges or other entities or combinations of entities that allow for the trading of financial instruments. First, in step **1002** a non-indexed collection of substantially sequential orders are stored. The orders may be stored in a variety of different memory modules. In one embodiment, the collection of substantially sequential orders may be stored in a processor memory, a solid-state memory or a memory module having no moving parts. The orders may be orders that were received at an exchange or other entity that matches orders for financial instruments and may be orders for futures contracts, option contracts or any other type of financial instrument.

[0079] Next, in step **1004**, the identification of a first time and a second time after the first time are received. The first and second times may identify a time period over which an activity state of an entity across the trading platform will be reconstructed. For example, if a regulator wanted to determine the profit and loss of a trader between 10:00:21 and 12:31:04 on a given day, the first and second times would be selected accordingly. In step **1006**, the non-indexed collection of substantially sequential orders are accessed and a computer device simulates the processing of orders received at a trading platform between the first time and the second time to reconstruct the activity state of an entity across the trading platform. Finally, in step **1008** the activity state is displayed on a graphical user interface. An exemplary graphical user interface is described below. In alternative embodiments activity state data is printed on a report or displayed on a display device in a manner other than a graphical user interface.

[0080] FIG. **11** illustrates a graphical user interface that may be used to initiate a process to reconstruct an activity state of an entity across a trading platform, such as the process

shown in FIG. **10**. A user may enter a start date/time value **1102** and an end date/time value **1104**. The start and stop values may correspond the first and second times described above in relation to FIG. **10**. A user may also provide identifying information, such as an account value **1106** and/or a firm value **1108**. One skilled in the art will appreciate that numerous additional or alternative values may be used to identify a firm, a trader, a session, etc. When selected, a "go" icon **1110** initiates the recreation process with the values provided by the user.

[0081] FIG. **12** illustrates a display or a graphical user interface that indicates an activity state of an entity across a trading platform and results from the input values provided in FIG. **11**. The identification of the time period and trading entity are found in section **1202**. The reconstructed activity state for the entity is shown in section **1204**. The activity state may include values that identify instruments, buy and sell volume, net positions, last price, profit and loss, number of orders, number of modifications, number of cancelled orders and number of quotes.

[0082] In various embodiments of the invention, graphical user interface elements allow users to view order book data at different levels. For example, the instruments displayed in column **1206** may be implemented with hyperlinks that link to additional data. Selecting instrument element **1208** may cause computer-executable instructions to generate the graphical user interface shown in FIG. **13**. FIG. **13** shows executed order data **1302** and pending order data **1304** for the selected instrument. Of course numerous additional links may be provided to allow a user to quickly view data different levels of granularity.

[0083] FIG. **14** illustrates a method for reconstructing a state of an order book in accordance with an embodiment of the invention. First, step **1402** a non-indexed collection of substantially sequential orders are stored. The orders may be stored in a variety of different memory modules. In one embodiment, the collection of substantially sequential orders may be stored in a processor memory, a solid-state memory or a memory module having no moving parts. The orders may be orders that were received at an exchange or other entity that matches orders for financial instruments and may be orders for futures contracts, option contracts or any other type of financial instrument. Next, in step **1404**, the identification of an initial state of an order book at a first time is received. For example, a computer system may receive information regarding state of an order book that exists when opening on Monday morning. Next, the identification of a second time after the first time is received in step **1406**. Step **1406** may include receiving a time of interest to a regulator or other entity monitoring trading activities. For example, if a regulator is investigating whether or not a trader was trying to manipulate a market at a particular time, the particular time would be the second time in step **1406**.

[0084] Next, in step **1408**, the non-indexed collection of orders is utilized to simulate the processing of orders received between the first time and the second time to reconstruct the state of at least a portion of the order book at the second time. Step **1408** may include processing orders in the same manner as used by the match systems described above and may include identifying information at the second time such as the state of a trader's or firm's order book, the state of a portion of a trader's or firm's order book, the profit and loss state, a net position state of the trader and any other information that may be obtained by simulating the processing of orders that are received between a first time and a second time.

[0085] The method shown in FIG. **14** may also be modified to allow a user to replay the state of a moving market. For example, a user may create a display having windows that show different information such as a trader's profit and loss and current positions and the stepwise show changes. In one embodiment, changes may be displayed during predetermined time periods. For example, every fifteen seconds the display may be updated to reflect the state of a order book one second later. This allows a user to view 60 seconds of market activity over a fifteen minute period. In another embodiment, the display is updated only after receiving a response from the user. This embodiment allows the user to spend as much time as necessary viewing data before advancing to the next time period.

[0086] Aspects of the invention may also be used to automatically generate various alarms. For example, a predetermined profit/loss condition for a trading entity may result in the generation of an alarm. The alarm may be in the form of an email message, SMS message, computer generated alarm or other type of alarm that may alert a firm, trader, regulator, trading entity or another of a condition.

[0087] The present invention has been described herein with reference to specific exemplary embodiments thereof. It will be apparent to those skilled in the art that a person understanding this invention may conceive of changes or other embodiments or variations, which utilize the principles of this invention without departing from the broader spirit and scope of the invention as set forth in the appended claims. For example, aspects of the invention may be applied to data collections that are not related to exchanges or trading. All are considered within the sphere, spirit, and scope of the invention.

We claim:

1. A computer implemented method of reconstructing at least a portion of an order book for a financial market, the method comprising:

(a) storing a non-indexed collection of substantially sequential orders in a solid-state processor memory;

(b) receiving the identification of an initial state of the at least a portion of the order book at a first time;

(c) receiving the identification of a second time after the first time; and

(d) utilizing the non-indexed collection of substantially sequential orders at the processor to simulate the processing of orders received between the first time and the second time to reconstruct the state of the at least a portion of the order book for the financial market at the second time.

2. The method of claim **1**, wherein (b) comprises recreating the state of the at least a portion of the order book from a startup time using pending orders prior to the startup time and orders between the startup time and the first time.

3. The method of claim **1**, wherein the orders comprise an order for any of a futures contract, an options contract, a security, an equity product or combinations thereof.

4. The method of claim **1**, further comprising displaying the state of the at least a portion of the order book at the second time.

5. The method of claim **1**, wherein the non-indexed collection of substantially sequential orders contains records having a fixed length.

6. The method of claim **1**, wherein the non-indexed collection of substantially sequential orders contains records having substantially equal length.

7. The method of claim **1**, wherein (d) comprises simulating the matching of new received orders to orders resting in the non-indexed collection of substantially sequential orders.

8. The method of claim **1**, wherein (d) comprises reconstructing a state of a trader's order book.

9. The method of claim **8**, wherein (d) comprises reconstructing the profit and loss state of the trader.

10. The method of claim **8**, wherein (d) comprises reconstructing a net position state of the trader.

11. The method of claim **1**, wherein (a) comprises separating order records into fields and physically grouping like fields.

12. The method of claim **11**, wherein one field comprises an account number.

13. The method of claim **11**, wherein (a) comprises assigning an ordered list of keys to values included in at least one field.

14. A computer-readable medium containing computer-executable instructions for causing a processor to reconstruct at least a portion of an order book for a financial market by performing the steps comprising:

(a) storing a non-indexed collection of substantially sequential orders in a solid-state processor memory;

(b) receiving the identification of an initial state of the at least a portion of the order book at a first time;

(c) receiving the identification of a second time after the first time; and

(d) utilizing the non-indexed collection of substantially sequential orders to simulate the processing of orders received between the first time and the second time to reconstruct the state of the at least a portion of the order book for the financial market at the second time.

15. The computer-readable medium of claim **14**, wherein (d) comprises reconstructing a state of a trader's order book.

16. The computer-readable medium of claim **15**, wherein (d) comprises reconstructing the profit and loss state of the trader.

17. The computer-readable medium of claim **15**, wherein (d) comprises reconstructing a net position state of the trader.

18. The computer-readable medium of claim **14**, wherein (b) comprises recreating the state of the at least a portion of the order book from a startup time using pending orders prior to the startup time and orders between the startup time and the first time.

19. The computer-readable medium of claim **14**, wherein the orders comprise an order for any of a futures contract, an options contract, a security, an equity product or combinations thereof.

20. The computer-readable medium of claim **14**, wherein (a) comprises separating order records into fields and physically grouping like fields.

* * * * *