SYNCHRONIZED PROCESSING OF DATA BY NETWORKED COMPUTING RESOURCES

Technical Field

[0001] The present disclosure relates generally to systems, methods, and machine-interpretable programming or other instruction products for the management of data processing by multiple networked computing resources. In particular, the disclosure relates to the synchronization of related requests for processing of data using distributed network resources.

[0002] Aspects of the material disclosed in this application relate to the holding, transfer, and/or administration of securities and other financial interests. Aspects of such holding, transfer, and/or administration may be subject to regulation by governmental and other agencies. The disclosure herein is made solely in terms of logical, programming, and communications possibilities, without regard to statutory, regulatory, or other legal considerations. Nothing herein is intended as a statement or representation that any system, method or process proposed or discussed herein, or the use thereof, does or does not comply with any statute, law, regulation, or other legal requirement in any jurisdiction; nor should it be taken or construed as doing so.

Background

[0003] In various forms of networked or otherwise distributed data processing systems, complex and/or multiple related processes are often routed to multiple computing resources for execution. For example, in financial and other trading systems, orders for purchases, sales, and other transactions in financial interests are often routed to multiple market or exchange servers for fulfillment. In such cases it can be advantageous for orders or other data processing requests routed to multiple servers, or other resources, to be executed simultaneously, or as close to simultaneously as possible, or to be executed in any otherwise desired-synchronized fashion, or time sequence.

[0004] For example, it has been observed that fill rates for orders related to financial interests executed in networked electronic markets decrease significantly when such orders are filled in non-synchronized fashion in multiple markets. It has further been observed that the decline in fill rate increases as such orders are routed to an increased number of electronic markets. This is at least partly due to delays in execution of subsequent portions of such orders after their first components have been filled: when an order has been executed in one market ahead of another, the intervening time period is sometimes used for price manipulation by parties trying to maximize short-term returns from offerings: when a first segment of an order has been filled, automatic changes in terms of offers or bids on parallel markets can be implemented, causing previously-publicized positions to be revoked and subsequent trade to be restrained.

[0005] For example, when a large order is routed to multiple exchanges (e.g., based on the liquidity available in each market), orders tend to arrive at the faster exchanges (i.e., those having fewer inherent latencies) before they arrive at slower exchanges (i.e., those having greater inherent latencies), and thus show in the books of different exchanges at different times. When orders begin to show on the books of the faster exchanges, other parties can detect the orders and attempt to take advantage of the latency in slower exchanges by cancelling, changing, and or otherwise manipulating quotes (e.g., bids and offers) or other market parameters on the slower exchanges, effectively increasing the implicit trading costs. As a result, orders that may have otherwise executed on any single exchange at a high fill ratio tend to exhibit a lower overall fill ratio when routed to multiple exchanges as a split trade.

[0006] Prior art documents, such as the Rony Kay article "Pragmatic Network Latency Engineering, Fundamental Facts and Analysis, have attempted to address such problems by proposing elimination of one-way communications (i.e., "packet") latencies. Such systems fail to address arbitrage opportunities and other issues caused or facilitated by variations in the time required for

multiple processors to execute individual portions of multiple-processor execution requests (i.e., execution latencies), in addition to (or as part of) communications latencies.

Summary

[0007] In various aspects the invention provides systems, methods, and computer-executable instruction mechanisms (e.g., non-transient machine-readable programming structures) such as software-coded instruction sets and data, for the management of data processing by multiple networked computing resources. In particular, for example, the invention provides systems, methods, and coded instruction sets useful in controlling the synchronization of related requests for processing of data using distributed network resources.

[0008] For example, in a first aspect the invention provides systems, methods, and programming or other machine-interpretable instructions for causing synchronized processing of data by multiple networked computing resources, such systems, for example, comprising at least one processor configured to execute machine-interpretable instructions and causing the system to:

receive from one or more data sources signals representing instructions for execution of at least one data process executable by a plurality of networked computing resources;

divide the at least one data process into a plurality of data processing segments, each data processing segment to be routed to a different one of a plurality of networked execution processors;

based at least partly on latencies in execution of prior data processing requests routed by the system to each of the plurality of networked execution processors, determine a plurality of timing parameters, each of the plurality of timing parameters to be associated with a corresponding one of the plurality of data processing segments, the

plurality of timing parameters determined to cause synchronized execution of the plurality of data processing segments by the plurality of networked execution processors; and

using the timing parameters associated with the plurality of data processing segments, routing the plurality of data processing segments to the plurality of corresponding networked execution processors.

[0009] In some embodiments, as will be explained herein, the networked execution processors can, for example, comprise exchange servers, and the data processing segments represent requests for trades in financial interests such as commodities and/or intangible interests such as stocks, bonds, and/or various forms of options.

[0010] The plurality of determined timing parameters can be used in determining and implementing timing sequences in order to implement desired sequential execution of data processing requests in accordance with the invention, and can for example represent and/or be based wholly or partially upon latencies in execution of data processing requests due many factors. For example, such parameters can be wholly or partially based on dynamically-monitored latency(ies) in execution of signal processing requests previously routed by the system to at least one of the plurality of networked execution processors. Such latencies may be caused by many factors, including, for example, various types of communication and data processing delays. Such timing parameters may further based on statistical, e.g., probability, models of observed latency data, and patterns therein.

[0011] Such systems, methods, and programming or other machine-interpretable instructions may further be configured such that they cause a system to:

associate with each of at least one of the plurality of data processing segments data representing at least one quantity term, the at least

one quantity term representing at least one quantity of a financial interest to be traded in accordance with a request represented each of the at least one data processing segments, and at least one corresponding price term associated with each such quantity term, the quantity term representing at least one proposed price at which a trade represented by the at least one data processing segment is to be executed;

the at least one quantity term larger than at least one quantity of the financial interest publicly offered at a price equivalent to the corresponding associated price term, in a market associated with the networked execution processor(s) to which the at least one data processing segment is to be routed.

[0012] Such quantity terms can, for example, be determined based at least partly on trading histories associated with the market(s) associated with the networked execution processor(s) to which the data processing segments are to be routed. They can be determined on data relating to displayed or undisplayed offerings and/or trades, including for example historical undisplayed oversize or reserve quantities.

[0013] In further aspects the invention provides systems, methods, and programming or other machine-interpretable instructions for causing synchronized processing of data by multiple networked computing resources, such systems, for example, comprising at least one processor configured to execute machine-interpretable instructions and causing the system to:

monitor execution of signal processing execution requests by each of the plurality of networked computing resources;

determine at least one timing parameter associated with a latency in execution of signal processes between the system and each of the plurality of networked computing resources; and

store the at least one timing parameter in machine-readable memory accessible by the at least one processor.

Monitoring of execution of signal processing execution requests according to such and other embodiments of the invention can be implemented on continual, periodic, and/or other suitable or desirable bases.

[0014] In various embodiments of the various aspects of the invention, the networked computing resources can include one or more exchange servers. The data sources can include one or more broker or trader systems or servers, the controlled signal processes can represent trades in financial interests, and the execution of signal processing execution requests represents the execution of transactions in financial interests, including for example stocks, bonds, options and contract interests, currencies and/or other intangible interests, and/or commodities. In such embodiments requests for execution of data processing procedures can be based wholly or partially on parameters including, for example, any one or more of current market data quotations, order routing rules, order characteristics, displayed liquidity of each networked computing resource, and a probable delay, or latency, in execution of an order quantity at each networked computing resource.

[0015] In the same and further aspects the invention provides systems for controlling or otherwise managing requests for processing of data by distributed computer resources, such systems including one or more processors configured to execute instructions for causing the system to:

monitor execution of signal processing execution requests by each of the plurality of networked computing resources;

determine at least one timing parameter associated with the latency in execution of signal processes between the system and each of the plurality of networked computing resources; and

store the at least one timing parameter for each of the plurality of networked computing resources.

[0016] Among the many advantages offered by the invention is the possibility of monitoring latencies and other factors in networked processing of multi-part or other complex data processing requests on a dynamic, or 'rolling', basis, and using such dynamically-monitored latencies and/or other factors in determining timing parameters to be used in implementing synchronized processing requests, as disclosed herein. Timing parameters used in implementing synchronized processing requests can be monitored and/or determined on continuous, continual, periodic, or other bases, depending upon the needs, objectives, and other factors of the applications in which they are to be applied.

[0017] A further advantage offered by the invention is reduction or elimination of the need for consideration of one-way communications latencies, e.g., the need to minimize latencies in communications between routing and execution processors.

[0018] As will be appreciated by those skilled in the relevant arts, once they have been made familiar with this disclosure, synchronization of execution of distributed data processing requests by, for example, synchronized transmission of requests for such processing, has a great many possible applications in a large number of data processing fields.

Brief Description of the Drawings

[0019] Reference will now be made to the drawings, which show by way of example embodiments of the present disclosure.

[0020] FIGS. 1A, 1B, and 3 show examples of systems suitable for causing processing of data by multiple networked computing resources in accordance with various aspects of the invention.

[0021] FIGS. 2 and 4 show flowcharts illustrating examples of methods for causing processing of data by multiple networked computing resources in accordance with various aspects of the invention.

[0022] FIG. 5 shows an example histogram that may be used in an example method for managing processing of data by multiple networked computing resources in accordance with various aspects of the invention.

[0023] FIGS. 6A and 6B show a comparison of fill ratios using an example method and system for processing of data by multiple networked computing resources versus using a conventional method and system.

[0024] FIG. 7 illustrates the use of an example metric for comparing an example method and system for processing of data by multiple networked computing resources versus results of using a prior art method and system.

[0025] Throughout the appended drawings, like features are identified by like reference numerals.

Description of Example Embodiments

[0026] In this disclosure, as will be understood by those skilled in the relevant arts, 'synchronized' means according to any desired timing sequence, whether regular, irregular, and/or wholly or partially simultaneous.

[0027] Figure 1 shows an example of a system 100 suitable for causing processing of data by multiple networked computing resources in accordance with the invention.

[0028] In the example shown, system 100 includes one or more signal or data sources 102 (comprising one or more each of sources102a, 102b), execution router processor(s) 104, and one or more networked computing resources, or execution processors, 106. In some embodiments, data sources 102 may include one or more internal data sources 102a, which may communicate with the router 104 directly (e.g., through private local- or wide area

network(s) or other secure wireless or wireline communication, through direct communication channel(s) or through communication(s) within a single server). In the same and/or other embodiments, data source(s) 102 may also include one or more external data sources 102b, which may for example communicate with router processor(s) 104 via one or more public networks 108 (e.g., a public or private telecommunications network such as the internet), using suitable or otherwise desired network security devices, which may for example include data encryption, etc. In the example shown, router processor(s) 104 communicate with each of the one or more networked execution, or computing, resources 106 via a network 110, which may be the same as or different than network(s) 108.

[0029] In various embodiments, data source(s) 102 may include devices that provide, on behalf of one or more entities that generate trading and/or other data processing requests, signals that communicate data and/or instructions related to execution of data processing processes to router processor(s) 104, which data and/or instructions the router processor(s) 104 may process (e.g., aggregate by summing, averaging, etc.; and/or divide into segments, etc.) and use as bases for requests for processing of data by the networked computing resources 106. Data sources 102a, 102b may include, for example, systems, servers, processors and/or any other suitable source(s) of requests for execution of data processing tasks such as offers and/or bids for purchase of commodities, intangible financial interests, etc., and/or other data processing tasks, such as word, image, and/or other communications or document processing tasks. Each or any of data source(s) 102, processor(s) 104, and resources 106 may include multiple such systems, servers or processors.

[0030] In various embodiments, some or all of data source(s) 102 and router processor(s) 104 may be combined, and/or otherwise configured to implement multiple programming or other machine instruction applications running on single machines.

[0031] Networked computing resources 106 may include any devices or other resources that communicate with router processor(s) 104 to receive and carry out any of a very wide variety of data processing requests. Such networked computing resources 106 may include systems, servers, processors or any other suitable devices adapted for execution of any processes suitable for use in implementing the invention, including, for example, processing of offers or bids for purchase of commodities, financial interests, etc., and/or other data processing tasks, such as word or document processing, image, and/or other communications or documentation tasks.

[0032] In various embodiments, the one or more data sources 102 transmit or otherwise provide to or for the router processor(s) 104 signals representing instructions, or requests, for executing data processing functions. Instructions from any given data source(s) 102 may include instructions for signal processes to be executed by any one or more networked computing resources 106. Requested signal processes may include, for example, computing operations, data manipulations, and/or communications processes or other signal exchanges, among others. In some but not necessarily all examples, such instructions may specifically identify networked computing resource(s) 106 particularly targeted for execution of such processes.

[0033] Router processor(s) 104 may parse instruction signals received from one or more source(s) 102 and use such signals to prepare instructions, or requests, to be forwarded to pluralities of execution processors 106, for execution of data processing and/or other signal processes in accordance with the received instructions. Parsing of such instructions may include, for example, identifying the type of process(es) to be requested, including for example the volume or quantity of an order or bid for a trade or an amount of document processing to be done, and the type, nature, and/or identity(ies) of networked computing resource(s) 106 to be requested to

execute, and thereby associated with, a given data processing and/or other signal processing request.

[0034] For example, in order to increase the efficiency of signal and/or other data processing functions, router processor(s) 104 may parse, sort, and aggregate instructions or requests received from multiple sources 102 for relatively smaller execution requests into one or more larger requests for processing, and further divide such aggregated request(s) into pluralities of smaller requests to be distributed to plurality(ies) of execution processors 106, depending, for example, on the current ability of the execution processors 106 to satisfy or complete such processed requests.

[0035] For example, multiple instruction signal sets received from different data sources 102a, 102b may be associated with (e.g., addressed for delivery to and execution by) individual networked computing resource(s) 106, and such instructions may be aggregated into single signal process execution requests for such networked computing resource(s) 106. In some examples, identification of the networked computing resource(s) 106 to be tasked with a given signal processing request may be performed after the aggregating. For example, multiple instructions from different data sources 102a, 102b may be sorted or otherwise associated with a single signal or data process, and such instructions may be aggregated, and the aggregated instructions may be associated with one or more identified networked computing resource(s) 106, such that one or more signal process requests may be accordingly prepared for the identified networked computing resource(s) 106. Such parsing, sorting, and/or identification may be performed according to predetermined rules or algorithms (e.g., based on continuing or current processing capabilities of one or more specific networked computing resource(s) 106), and according to requirements encoded in the instructions or otherwise provided by the originating source(s) 102, where relevant.

[0036] As a further example, single instruction sets for processing of data may be broken down by processor(s) 104 and distributed to a plurality of resources 106 for distributed execution. For example, a relatively large order for trading in one or more financial interests originating from a single source 102a, 102b, might need to be distributed to multiple exchange servers 106 in order to be completely filled; in such cases request(s) from one or more source(s) 102 may be broken down by processor(s) 104 into suitable orders for execution by a plurality of such resources 106.

[0037] Targeted, or specifically identified, networked computing resources / execution processors 106 communicate with the router processor(s) 104 to receive the segmented signal process execution requests and may thereafter execute them accordingly. Execution of such signal processes may include, for example, carrying out a text- or image-processing operation, a mathematical computation, or a communications signal exchange, among others.

[0038] As will be readily understood by those skilled in the relevant arts, various components of system 100 may combined, or may be implemented in the form of separate systems or devices. In a wide variety of configurations, such combined or separate (sub)systems may be operated by the same or distinct entities. As a particular example, one or more request source(s) 102 may be integrated with, or otherwise associated with, individual router(s) 104.

[0039] An example of an application of a system 100 for distributed execution of segmented processing requests in accordance with the invention is provided by a financial system 1000 adapted for processing of requests for processing of data representing trades and/or offers for trades, or other transactions, in tangible and/or intangible financial interests such as stocks, bonds, currencies (e.g., foreign exchange), various forms of natural resources or commodities, options, loans, etc. As shown in Figures 1A and 1B, for example, in a financial transaction data processing system 1000 in

accordance with the invention, signal or data source(s) 102 may include trader system(s) 1102, which may, for example, include trader/broker systems or servers as well as any other sources of bids, offers, or other transactions in financial interests such as currently provided by known financial trading platforms. In various embodiments, such trader systems 1102 may be referred to as order origination systems.

[0040] Order origination systems 1102, 102a may include systems operated by or on behalf of, for example, entities owned or otherwise controlled by parent or other controlling organizations such as banks or brokerage houses. Order origination systems 1102, 102b may, for example, include systems operated by or on behalf of brokers or other trading entities acting on behalf of, for example, individual investors, trading through or with the assistance of independently-controlled banks, institutional investors, and/or other brokerage houses.

[0041] Router processor(s) 104 in such embodiments may include, for example, server(s) or other system(s) 1104 that communicate with trader systems 1102, 102, for example through the receipt and transmission of encoded electronic signals representing requests for processing of data representing execution and/or acknowledgement of transactions in financial interests; and which communicate with broker, exchange, or other market systems or execution processor(s) 1106 for execution of such transactions. In such embodiments a processor 104 may be referred to as a Smart Order Router or Tactical Hybrid Order Router (in either case, "SOR") 1104, 104. An SOR 1104 may, for example, include one or more gateway(s) 1122 and/or router(s) 1124 for facilitating communications by router(s) 1104 with one or more trader systems 1102, 102 directly (e.g., through wired communication, using one or more dedicated communication channel(s), or through communication within a single server) and/or indirectly (e.g., through wireless communication, through a network 108, 1108 or through an intermediate server). Exchange or market systems 1106, or other execution processor(s) 106 may be in communication with SOR(s) 1104 through, for

example, a network 110, 1110, such as the internet or other public network, which may be the same as the network 1108.

[0042] For an embodiment of a system 100 configured as a financial trading or order execution system 1000, requested and executed signal processes provided by source(s) 102 may represent trades or other transactions in financial interests. Such transactions may include, for example, trades and/or offers for trades, or other transactions, in financial interests such as stocks, bonds, currencies (e.g., foreign exchange), various forms of natural resources or commodities, options, loans, etc.; and networked computing resources 106 may be, for example, exchange servers 1106, examples of which may include automatic or electronic market systems.

[0043] As will be well understood by those skilled in the relevant arts, an SOR (sub)system, or processor, 1104 receiving such transaction request signal sets can apply a wide variety of processes to the request(s). For example, where the signal sets represent requests for transactions in financial interests, requested transactions can be aggregated, either over time and/or across multiple transaction request sources 1102; and/or processing requests for transactions in one or more interests can be divided for routing to multiple execution handlers or processors 1106, individually or in batches.

[0044] In various embodiments, as described herein, order source(s) 102, 1102 can be implemented together with, or as part of, order router(s) 104, 1104. It will be readily understood by those skilled in the relevant arts that any or all of the various components of system(s) 100, 1000, including for example any or all of processor(s) 102, 104, 106, and methods of operating them in accordance with the disclosure herein, may be implemented using any devices, software, and/or firmware configured for the purposes disclosed herein. A wide variety of components, both hardware and software, as well as firmware, are now known that are suitable, when used singly and/or in various combinations, for implementing such systems, devices, and methods; doubtless others will hereafter be developed.

[0045] Examples of components suitable for use in implementing examples of systems 100, 1000, and the various processes disclosed herein, including for example processes 200 of Figure 2 and 300 of Figure 4, include, for example server-class systems such as the IBM x3850 M2™, the HP ProLiant DL380 G5™ HP ProLiant DL585™, and HP ProLiant DL585 G1™. A wide variety of other processors, including in some embodiments desktop, laptop, or palm model systems will serve.

[0046] An example of a method 200 for processing of a transaction request signal set generated by a transaction request signal source 102, 1102, suitable for implementation by an router processor(s) 104 such as, for example, an SOR 1104 of a system 1000, is shown in Figure 2.

[0047] Process 200 of Figure 2 can be considered to start at 202, with receipt by processor(s) 104, 1104 of signals representing a request for processing of data such as, for example, a transaction in one or more financial interests. In embodiments of systems 100, 1000 comprising SOR routing processor(s) 1104 adapted to process signals representing requests for execution of trades and/or other transactions in financial interests received from transaction signal source(s) 1102, signal sets representing requests for execution of transactions in one or more financial interests can include signals or signal sets representing, for example, one or more identifiers representing:

- the source(s) of the request, such as a URL or other network address or identifier used by or otherwise associated with a trading system 102, 1102;
- the interest(s) to be traded or otherwise transacted, such as an identifier used by one or more exchanges to identify a stock, a CUSIP number for a bond, a set of currencies to be exchanged, etc.;
- a type of transaction (e.g., buy, sell, bid, offer, etc.) to be executed or requested;

- one or more quantities (i.e., amounts or volumes) of the interest(s) to be transacted (including for example any total and/or reserve quantities); and

- corresponding price terms.

[0048] Further parameters can include, for example, current and/or historical:

- fill probability for multi-part, or segmented, transaction requests (i.e., the historical proportion of multi-part orders that result in completed transactions);

- amounts of spread between, for example, bid and offer prices, e.g., current and/or relative to historical trends in spread;

- market volatility in specific interests to be traded, or related or corresponding interest(s), or related benchmarks or indexes;

- depth of market book(s), for example current depth relative to historical trends in depth;

- reserve quantities;

- display quantities; and

- display size and backing, for example on buy and/or sell sides.

[0049] In other embodiments, such signal sets can comprise content and/or identifiers representing images, text, or other content or to be processed by one or more execution processors 104, 1104, and specific execution requests.

[0050] Among the many types of market systems 1106 suitable with various embodiments of the invention are alternative trading systems (ATSs) of the type known as 'dark' exchanges, or 'dark pools'. Typically, such exchanges do not openly display market offerings to members of the trading public. The

use of known or predicted reserve quantities can be especially useful in such embodiments.

[0051] Thus an example of a data record to be provided by a source 102, 1102 to request a transaction in a given interest, on stated terms, can include:

<source (102, 1102) of request><type of transaction><interest identifier><quantity(ies)><price term(s)>

[0052] Signal sets received by processors 104, 1104 at 202 can be stored in any volatile and/or persistent memory(ies), as appropriate, for archival and/or further processing purposes.

[0053] At 204, transaction or other data processing execution requests received at 202 can be parsed by router processor(s) 104, 1104 to place them in any suitable or desired form for use in preparing one or more instruction signal sets to be provided to execution processor(s) 106, 1106. Parsing of instruction signals may include, for example, identifying the type of transaction(s) or process(es) to be requested, including for example volumes and/or quantities of orders or bids for trades in specified interest(s), and whether such volumes are to be bought or sold, or offered for sale or purchase; amounts and/or types of document processing to be done; and the type and nature of networked computing resource(s) or execution processor(s) 106 to be requested to execute and thereby be associated with such execution or processing instructions. In various embodiments parsed instruction sets can be stored in temporary or volatile memory(ies) 118, 1018 accessible by the corresponding processor(s) 104, 1104 for aggregation with other processing requests, division for routing to multiple execution processors / resources 106, 1106, and/or preparation and forwarding of batch or other delayed-execution requests.

[0054] Instructions received at 202 may be accumulated during defined time intervals, regular or irregular, such as the duration of a business day or any

segment thereof, or any other desired time period(s), which may be preset and/or dynamically determined by processor(s) 104, 1104. Instructions may be also be processed individually, as received. If more instructions are to be received prior to processing, or may potentially be received, process 200 can return to 202.

[0055] Transaction requests / instructions may be accumulated during defined time intervals, such as the duration of a business day or any segment thereof, or a desired time period, which may be preset and/or dynamically determined by processor(s) 104, 1104. If more instructions to be received, or may potentially be received, process 200 can return to 202.

[0056] In embodiments of the invention which employ sorting / aggregation techniques in parsing or otherwise preparing order or other processing requests, at 206 processor(s) 104, 1104 can repeat process 202 – 204 until all needed or desired related or aggregatable processing request signal sets have been received from source(s) 102, 1102. For example, as described above, arbitrary numbers of data records representing orders or requests for purchase of bonds identifiable by CUSIP (Committee on Uniform Security Identification Procedures) numbers can be received from data source(s) 102, 1102, and stored in memory 118, 1018 associated with the processor(s) 104, 1104, for batch processing, thus:

<source 1><sell><CUSIP No. AA><10,000><price A><res. 9,000><price D>
<source 2><buy><CUSIP No. BB><12,000><price C><res. 1,000><price B>
<source 3><sell><CUSIP No. BB><11,000><price A><res. 8,000><price D>
<source 6><sell><CUSIP No. AA><14,000><price A><res. 2,000><price E>
<source 4><buy><CUSIP No. AA><18,000><price C><res. 7,000><price B>
<source 1><sell><CUSIP No. BB><20,000><price A><res. 3,000><price D>
<source 3><sell><CUSIP No. AA><13,000><price A><res. 6,000><price D>
<source 4><buy><CUSIP No. BB><22,000><price C><res. 4,000><price B>
<source 5><sell><CUSIP No. AA><21,000><price A><res. 5,000><price E>
<source 4><buy><CUSIP No. BB><15,000><price C><res. 7,000><price F>
<source 1><sell><CUSIP No. AA><19,000><price A><res. 3,000><price D>
<source 5><buy><CUSIP No. BB><16,000><price C><res. 8,000><price F>
<source 6><sell><CUSIP No. BB><17,000><price A><res. 6,000><price H>

[0057] Upon individual receipt, or at a given periodic rate, a given time, when a given number of orders has been received, when all desired orders have

been received, or when any other desired criteria have been satisfied, processor(s) 104, 1104 can, as a part of parsing or otherwise processing instructions at 204, sort and/or group the stored records according to any one or more desired criteria, e.g., by type of transaction request and interest identifier, thus:

<buy><CUSIP No. AA><18,000><price C><res. 7,000><price G><source 4>

<sell><CUSIP No. AA><10,000><price A><res. 9,000><price D><source 1>
<sell><CUSIP No. AA><14,000><price A><res. 2,000><price E><source 6>
<sell><CUSIP No. AA><13,000><price A><res. 6,000><price D><source 3>
<sell><CUSIP No. AA><21,000><price A><res. 5,000><price E><source 5>
<sell><CUSIP No. AA><19,000><price A><res. 3,000><price D><source 1>

<buy><CUSIP No. BB><15,000><price C><res. 7,000><price F><source 4>
<buy><CUSIP No. BB><22,000><price C><res. 4,000><price B><source 4>
<buy><CUSIP No. BB><12,000><price C><res. 1,000><price B><source 2>
<buy><CUSIP No. BB><16,000><price C><res. 8,000><price F><source 5>

<sell><CUSIP No. BB><20,000><price A><res. 3,000><price D><source 1>
<sell><CUSIP No. BB><11,000><price A><res. 8,000><price D><source 3>
<sell><CUSIP No. BB><17,000><price A><res. 6,000><price H><source 6>

As shown, various data fields in the transaction request records can be reordered or otherwise reformatted as needed or desired, to suit the processing needs of the routing processor(s) 104, 1104. For example, as shown, the association of a "source" data item associated with or otherwise accorded a different priority, to facilitate efficient ordering while permitting the processor(s) 104, 1104 to report fulfillment of transactions / requests on completion of order processing.

[0058] Process 204 can further include aggregation by processor(s) 104, 1104 of received and sorted transaction requests, into collected or consolidated order(s) for specific types of transactions in specific interest(s), e.g., by summing total or subtotal quantities associated with corresponding transaction requests, thus:

<buy><CUSIP No. AA><18,000><price C><res. 7,000><price G>

<sell><CUSIP No. AA><77,000><price A><res. 18,000><price D>
<res. 7,000><price E>

<buy><CUSIP No. BB><65,000><price C><res. 15,000><price E>
<res. 5,000><price B>

<sell><CUSIP No. BB><48,000><price A><res. 11,000><price D>
<res. 6,000><price H>

[0059] When all desired signal sets have been received at 202, and optionally sorted, accumulated, and/or otherwise processed at 204, at 208 processor(s) 104, 1104, using instruction sets processed at 204, can prepare execution-request signal sets for transmission to resources / execution processors 106, 1106. Such execution-request signal sets can comprise any necessary or desirable signals for causing requested processing, including content or data and command signals. For example, in embodiments of the invention adapted for processing of requests for transactions in financial interests, requests may be sorted and/or aggregated on the basis of interest(s) to be traded, quantities of interest(s) to be traded, price, etc., and associated with suitable execution command signals. The form of any execution command signals associated with a given request can depend, as those skilled in the relevant arts will recognize, on the nature and type of requests to be executed and the processors 106, 1106 by which they are to be executed, as well any networks 110, 1110 over which signals exchanged between processor(s) 104, 1104 and 106, 1106 are to be sent, including applicable protocols and instruction formatting requirements. Ergo, data pertaining to any or all of systems 106, 1106, 104, 1104, and 110, 1110, protocols used thereby, and/or information related to interests traded, offered, or described thereby may be accessed and used by processor(s) 104, 1104 in parsing and preparing instructions for execution of processing by any of processors or resources 106, 1106. Sources 1126 of such data may include, for example, exchange market data system 1126v (Figure 1b) which, for example, in embodiments of the invention adapted for processing of financial transactions, can include information received from various exchange systems 1106, news information sources such as Bloomberg or Reuters, and/or other sources.

[0060] It is sometimes necessary or desirable, in assembling requests for data processing using networked processing resources, including many resources configured for use in executing financial transactions, to break execution and/or other processing requests into multiple parts. Such parts, or segments, can, for example, correspond to portions of larger orders or other data processing requests, to be executed by a plurality of networked resources 106 such as exchange servers or other execution processor or handlers 1106. For example, if a plurality of exchange servers or other markets are available for execution of a transaction request representing a purchase order for a significant amount of a financial interest such as a stock or bond, it may be necessary or desirable to split the order into multiple parts, for execution in multiple markets and/or by multiple exchange servers 1106. For example, sufficient quantities of specific interests may not be available, at all or at desirable prices, on a single exchange: in order to fill an order entirely, it may be necessary or desirable to break a single order into smaller segments and route it to multiple exchanges.

[0061] Thus, for example, in various embodiments of the invention directed toward the processing of requests for transactions in financial instruments, when a router 104,1104 is requested by one or more sources 106, 1106 to complete a transaction in one or more financial interests, the router 104, 1104 can, in preparing signal set(s) representing requests for the transactions, access information available from sources such as market data source(s) 1126, as well as any one or more execution processor(s) 106, 1106, to determine the quantities of such interests available through the respective processors 106, 1106 and the terms under which such quantities are available, and can construct an execution request signal set configured for routing to each of the respective desired processors 1106, 1106, based on the number of quantities available at the most favorable terms.

[0062] For example, continuing the example above, it may be necessary or desirable to split one or more incoming processing requests into smaller parts, directed to a plurality of exchanges, in order to obtain fulfillment of the

complete order(s). This can be accomplished by, for example, accessing data representing current order books provided by one or more of exchange servers 1106 and dividing the order(s) correspondingly, using known data processing techniques. Thus, for example, the aggregated 'sell CUSIP No. AA' order above may be broken down into portions or segments and associating with data representing such segments URLs or other network resource address identifiers suitable for use in routing the various segments to a plurality of exchange servers A1 – C3, as desired, thus:

<exchange A1><sell><CUSIP No. AA><15,000><price A><res. 6,000><price D><res. 2,000><price E>

<exchange B2><sell><CUSIP No. AA><27,000><price A><res. 6,000><price D><res. 2,500><price E>

<exchange C3><sell><CUSIP No. AA><35,000><price A><res. 6,000><price D><res. 2,500><price E>

[0063] As will be appreciated by those skilled in the relevant arts, execution of individual portions of a distributed transaction or other multi-part data processing request such as a transaction in financial interests placed in multiple exchanges by a plurality of network resources, such as market or exchanger servers 1106 or other execution processors 106, typically requires different amounts of time. That is, if multiple parts of a desired transaction execution request are sent simultaneously to a plurality of exchange execution processors 106, 1106, each part or segment of the transaction request may be expected to execute at a different point in time. This is because the amount of time, or 'latency,' required for transmission of execution request signals from the order router(s) 104, 1104 to the different various resources or execution processor 106, 1106 across a network 110, 1110 or other communications path; for actual processing of corresponding portions of the execution request by the corresponding processors 106, 1106; and/or for return of confirmatory or other data to the order router(s) 104, 1104 typically varies depending upon a number of factors, including for example the network paths between the router(s) 104, 1104 and execution processors 106, 1106; the amount of network traffic being processed by the

network(s) 110, 1110; the number of requests being handled by the individual execution processors 106, 1106, etc.

[0064] For a number of reasons it can be important, in such cases, to synchronize execution of two or more portions of a multi-part execution request. As one example, when an execution request represents a request for execution of multiple parts of a financial transaction in multiple markets or on multiple exchanges, non-synchronized, staggered execution of individual portions of the transaction by multiple corresponding servers can affect both the possibility of completing later portions of the transaction and/or the terms under which such later portions may be completed.

[0065] A particular example of the desirability of synchronizing execution requests may be illustrated through reference to Figure 3. In the example shown in Figure 3, system 100, 1000 comprises order router 104, 1104 and a plurality of networked execution resources 106, exchange servers or execution processors 1106 "Exchange 1," "Exchange 2," "Exchange 3." In addition, system 100, 1000 of Figure 3 further comprises a co-located trading server 304 configured to execute trades or other transactions on execution resource 1106 "Exchange 1." As noted in the Figure, co-located trading server 304, which employs a relatively low-latency trading algorithm, is associated with Exchange 1 in such manner that it can execute transactions with Exchange 1 in a relatively short period of time compared to the amount of time required for other processors, such as router(s) 104, 1104, to complete similar transactions with Exchange 1. For example, co-located server 304 can be communicatively linked with Exchange 1 by direct wireline connection, or other rapid-processing system. Moreover, Exchange 1 is capable of completing an execution request with non co-located processor(s) 104, 1104 in a relatively shorter period of time (i.e., with a "lower latency") than is either Exchange 2 or Exchange 3. In other words, as shown in Figure 3, latency Time X < Time Y and Time X < Time Z, while an execution time for a transaction between co-located server 304 and Exchange 1 is less than any of Time X, Time Y, and Time Z.

[0066] If, for example, signals representing a request to trade in one or more financial interests is received by a router processor 104, 1104 from one or more request sources 102, 1102, and the request is of such quantity or magnitude that an order reflecting the request will be too large to be completely filled by any one of Exchanges 1, 2, or 3, the order router 104, 1104 may attempt to check availabilities on the various available processors 106, 1106 and split the order accordingly, in order to route a portion of it to each of Exchange 1, Exchange 2, and Exchange 3. If the router 104, 1104 of Figure 3 simultaneously transmits to each of execution processors 106, 1106 Exchange 1, Exchange 2, and Exchange 3 a divided portion or segment of the request for execution of the requested transaction, it is possible that trading server 304 (which might, for example, be operated by a high-frequency trading entity, or other speculative investor) will be able fill a portion of that transaction on Exchange 1 by, for example, acting as a counterparty to the proposed transaction by selling or buying all or a portion of the transaction request forwarded to that exchange by the order router 104, at terms stated in the request for the transaction, and have time in which to change or otherwise post terms for filling remaining portions of the order on Exchange 2 and/or Exchange 3, on terms more favorable to the party making the transaction(s) available (e.g., the party operating or acting through the server 304) than those offering such transactions (e.g., those behind orders provided by request processor(s) 104, 1104) might otherwise have sought. In other words, for example, the co-located trading server 304 may, due to the difference in execution latencies associated with trades with Exchange 1, Exchange 2, and Exchange 3, be able fill a portion of the requested transaction on Exchange 1 and move to improve its terms, by for example raising or lowering its bid/offered price, for filling remaining portions of the transaction on Exchange 2 or Exchange 3 before such remaining portions can execute at previously-stated prices, in order to increase its operators' or beneficiary(ies) own profits, or the profits of other traders offering similar interests on those Exchanges.

[0067] As may be seen in Figure 3, such possibilities (which can be referred to as 'latency arbitrage' opportunities) can exist when:

Time X + Time A < Time Y and/or
Time X + Time B < Time Z

[0068] It will be appreciated by those skilled in the relevant arts that, even where transaction or other processing request signals are sent simultaneously to each of Exchanges 1, 2, 3 from the router(s) 104, 1104, the time required for each divided portion of the request to be received, acknowledged, and/or processed by the respective resources 106, 1106 (e.g., Times X, Y, Z) may in general be different, for example due to differences in network communications paths and processing speeds in any or all of processor(s) 104, 1104 and/or 106, 1106. Similarly, the time required for trading server 304 to change terms of transaction offerings in each of Exchanges 2 and 3 may in general differ.

[0069] Among the disadvantages which can arise in such cases is that traders represented by request source(s) 102, 1102 may pay higher prices in executing their trades than they otherwise would have, in the absence of such arbitrage opportunities; or, if prices on subsequent exchanges are changed sufficiently to put them outside terms stated in their execution requests, they may not be able to complete transactions in desired quantities – for example, all or part of a transaction routed to an exchange processor 1106 may not trade in view of a changed price.

[0070] In such examples, in which a trade instruction may not be fully fulfilled at an exchange server 1106 due for example to price or other term manipulation by a third party taking advantage of latencies, in prosecuting data processing requests in one or more exchange servers 1106 it may be useful to time or schedule the sending of trade requests to multiple exchange servers 1106 such that the execution of such trade requests at all exchange servers 1106 happens in a synchronized manner, such as, for example, in a substantially concurrent manner. In particular, it may be useful to

synchronize the execution of signal processing execution requests, or portions or segments thereof, in multiple networked computing resources 106, 1106, for example such that the signal processes are received, acknowledged, and/or executed by the resources 106, 1106 in a substantially concurrent manner.

[0071] In some examples it may not be necessary for the signal processes to be executed in each processor 106, 1106 to be executed simultaneously, but may be sufficient that:

Time Y – Time X < Time A, and/or

Time Z – Time X < Time B,

such that execution of the request(s) or segments thereof occurs before any change in terms can be implemented by a trading server 304. The use of such synchronized timings can, for example, cause:

Time X + Time A > Time Y and/or
Time X + Time B > Time Z

and thus, for example, defeat latency arbitrage opportunities. In some embodiments, therefore, the invention provides router(s) 104, 1104 the ability to execute transactions across multiple resources 106, 1106 with minimal or no time variance, such that algorithms run by trader(s) 304 employing low-latency algorithms are given insufficient time to react to market changes.

[0072] Thus, in these and other cases where synchronization is desired, at 210 processor / router 104, 1104 can determine absolute or relative timings to be assigned to, or otherwise associated with, various portions or segments of an execution request, in order to obtain the desired sequencing. Such timings can be determined in order to cause any desired synchronization: for example, timings configured to cause simultaneous, or substantially

simultaneous, execution can be determined, or timings configured to cause any desired sequencing can be determined.

[0073] Thus at 210, a timing parameter can be determined for each signal processing execution request, or portion thereof, to be assigned to each respective networked computing resource 106, 1106. The parameters are determined in such manner as to cause synchronized execution of the signal processing execution requests at each of the respective networked computing resources 106, 1106. This determination can be based at least partly on a corresponding determined latency in the execution time of such request(s) and/or portion(s), such as for example any or all of latencies A, B, X, Y, Z of Figure 3, and/or any other relevant latencies, in the execution of signal exchanges between the router processor(s) 104, 1104 and each of the networked computing resources 106, 1106, or in the processing of other such signals by any of such devices.

[0074] Arbitrage and other problems caused by variations in execution time between servers can also be minimized or eliminated by reducing absolute latencies in transmission and execution of processing requests. Thus the determination of timing parameters as described above can be practiced in combination with procedures which also serve to minimize absolute amounts of time associated with execution and/or reporting of execution requests by resource(s) 106, 1106.

[0075] Information on determined latencies used in determining timing parameters to be associated with the various portions of a multi-part execution request provided by router(s) 104, 1104 to a plurality of execution processors 106, 1106 may include timing information (e.g., transmission delays, signal propagation delays, serialization delays, queuing delays, and/or other processing delays at the router processor(s) 104, 1104, the networked computing resource 106, 1106, and/or network(s) 110, 1110, 108, 1108). Such information may be provided by or received from any source(s), and may be stored in and retrieved from one or more data stores

214. Timing data store(s) 214, in various embodiments, may include databases or other data structures residing in memory(ies) 118, 1018 associated with or otherwise accessible by router processor(s) 104, 1104. For example, if execution of a portion of an execution request associated with a first networked computing resource 106, 1106 has a longer determined latency than that associated with a second networked computing resource 106, 1106 (as for example in the case of Exchange 1 vs. Exchanges 2 and 3 of Figure 3) timing for requests associated portions of a transaction request to be routed to these two networked computing resources 106, 1106 may be determined such that an execution request, or portion thereof, associated with the first networked computing resource 106 is timed to be sent earlier than the request associated with the second networked computing resource 106, with the aim of having the requests executed at the two networked computing resources 106 substantially concurrently, or within an effective minimum time A or B associated with possible term manipulation by a trading server 304.

[0076] In some embodiments, one or more algorithms, which may for example use a latency probability model or other predictive model, may be used in determining timing parameters to be associated with portions of execution requests to be routed to various execution processors 106, 1106, based on information associated with such communication and/or processing delays, or latencies. For example, a rolling average of historical latency data, accumulated over or relevant to any desired devices, time periods, or other timing considerations may be used to predict an expected latency for execution of a data processing request.

[0077] One example of an algorithm suitable for use in determining timing parameters to be associated by router(s) 104, 1104 with portion(s) of requests for execution provided by source(s) 102, 1102, where it is desired to cause concurrent or otherwise synchronized arrival of such portions or requests at network resources 106, 1106, is based on an average latency between transmission of request signals from router(s) 104, 1104 and an

appropriate timing reference. Such timing reference(s) can for example include start of processing by the corresponding targeted resource(s) 106, 1106, and/or receipt by routing processor(s) 104, 1104 of a confirmation signal generated by the resource(s) 106, 1106 on receipt of the request and/or completion of execution of the request. For example, in some embodiments, it can be advantageous to measure latencies between transmission to a given resource 106, 1106 and receipt by the router(s) 104, 1104 of a confirmation or acknowledgement signal, or other appropriate response signal 1260, from such resource 106, 1106, and to use such measured latency(ies) in determining timing parameter(s) at 210.

[0078] Process step 210 may for example be carried out by a application executed by, or a module of, or otherwise associated with, routing processor(s) 104, 1104 such as a capital management entity or module 1126 in the case of a financial system 1000. Determination of a timing parameter to be associated with each part or segment of a multi-part execution request may, for example, include use of an adaptive exchange round-trip latency (RTL) learning & compensation logic module 1126c, such as that shown in Figure FIG. 1B. Referring to Figure 3, such an adaptive exchange RTL learning & compensation logic module 1126c may determine the timing for each signal processing request (e.g., a trade request) as follows:

1) For each portion or segment *n* of an *m*-part multi-part processing request X, a time $T1_{x,n}$ provided by, for example, a clock associated with the processor(s) 104, 1104 is time-stamped by processor(s) 104, 1104 at a desired defined point within the process of parsing or generating the transaction order(s), or other processing request(s) X, and is associated with a processing request signal set record(s) corresponding to each part or segment *n* of the *m*-part request X.

2) $T2_{x,n}$ for each portion *n* of the multi-part request X is time-stamped by the processor(s) 104, 1104 when the corresponding n^{th} portion request signal set has been received at the targeted exchange 106, 1106, and

a corresponding exchange-generated confirmation message has been received by the requesting routing processor 104, 1104.

3) During the course of a trading day (or other data processing period), process steps 2 and 3 may be repeated, and corresponding $T1_{x,n}$ and $T2_{x,n}$ determined for each transaction segment routed to a given execution processor 106, 1106.

4) For each portion segment *n* of a subsequent pending multi-part execution request Y, the determined timing parameter $RTL_{y,n} = \Sigma(T2_{x,n} - T1_{x,n}) / Z$, where *Z* is the number of previously-executed order segments routed to a given execution processor 106, 1106 used in the calculation.

[0079] Where timing data store(s) 214 store a rolling record of past timing parameters (e.g., a plurality of determined timing parameters $RTL_{y,n}$) associated with one or more execution resources 106 / exchange server 1106, such data may be used to create a rolling histogram, which may be used to predict current or cumulative latency for each resource 106 / exchange server 1106. Because such predictions are based on a continuously-changing ("rolling") record, this process may be referred to as "online learning." There may be a component (e.g., an exchange latency histogram memory or processing component, not shown) within the adaptive exchange RTL learning & compensation logic module 1126c responsible for this.

[0080] An adaptive exchange RTL learning & compensation logic module 1126c may use predicted latencies to determine appropriate timing parameters to be used in transmitting trade (or other data processing) requests to various exchange servers 1106 in order to compensate for differences in execution latencies associated with such exchange servers 1106, in a way that reduces, controls, minimizes or eliminates differences in timing of execution of portions of divided trade requests routed to different

exchange servers 1106, and thus, for example, reduces or eliminates opportunities for latency arbitrage by opportunistic traders.

[0081] Adaptive RTL module(s) 1126c can use a variety of algorithms in determining timing parameters suitable for use in synchronizing execution of multi-part processing requests. For example, such a module may use latency values determined for the various exchanges to determine the extent to which the router(s) 104, 1104 should compensate for different exchange latencies by sending to the various processors 106, 1106 their corresponding portions of a request for processing at, for example, different times. This can minimize delay between completion of execution of each portion by, for example, minimizing the difference in time between receipt of each respective portion by its corresponding execution resource 106, 1106. (In Figure 3, for example, this would be shown as minimizing differences between times elapsed at Time X, Time Y and Time Z.) Such algorithms can also account for historical differences in the time required for execution of trade or other processing orders on the various resources 106, 1106, in addition to communication delays.

[0082] Adaptive exchange RTL learning & compensation logic module(s) 1126c may additionally collect information about market conditions prevailing in each exchange server 1106 (using, for example, sources of data such as exchange market data source(s) 1126v), wave orders/executions, actual latencies and target latencies (e.g., as predicted above) when trade requests are sent. There may be a component within the adaptive exchange RTL learning & compensation logic module 1126c responsible for this.

[0083] One or more timing parameters associated with execution requests to be routed to any one or more of execution processor(s) 106, 1106 can also be provided to the corresponding routing processor(s) 104, 1104 (e.g., to timing data store 214) by, or determined by such processor(s) 104, 1104 using related data supplied by, any one or more market data feed(s) or processor(s) 1126 (including e.g., any one or more of processors or

(sub)systems 1126a – 1126g and/or 1126v), and/or by processor(s) 106, 1106 themselves.

[0084] At 212, the various portions of the optionally aggregated and divided signal processing execution request(s) are sent to the respective networked computing resources 106 according to timing parameters or sequence(s) determined or otherwise acquired at 210. Thereafter the request(s), or the various portions thereof, may be executed by the respective execution resources 106, 1106, with subsequent signal communications and processing as needed or desired. As will be understood by those skilled in the relevant arts, once they have been made familiar with this disclosure, once the parameters of a desired execution request have been determined by router(s) 104, 1104, signals representing those parameters may be assembled, using known or specialized data processing techniques; formatted according to the Financial Information Exchange (FIX) protocol and/or any other desired protocol(s); and transmitted, written or otherwise communicated to the corresponding execution processor(s) 106, 1106 using known or specialized signal communications techniques, and executed in accordance with requested transaction or other data processes.

[0085] For example, continuing the example above, timing delays, or parameters X', Y', Z', one or all of which may be equal to zero or any other suitable time period, may be determined according the disclosure above and associated with the order segments generated by processor(s) 1104 for purchase of 77,000 bond lots of CUSIP No. AA bonds at price A, with 25,000 lots (18,000 + 7,000) in reserve at prices D and E, respectively, thus:

<delay X'><exchange A1><sell><CUSIP No. AA><15,000><price A>
<res. 6,000><price D><res. 2,000><price E>

<delay Y'><exchange B2><sell><CUSIP No. AA><27,000><price A>
<res. 6,000><price D><res. 2,500><price E>

<delay Z'><exchange C3><sell><CUSIP No. AA><35,000><price A>
<res. 6,000><price D><res. 2,500><price E>

[0086] Thereafter, routing processor(s) 104, 1104 can process the transaction segments by using timing parameters, e.g., delays X', Y', Z', to cause the corresponding transaction segments to be transmitted or otherwise provided to the exchanges 106, 1106 A1, B2, C3 for execution according to a desired timing sequence, for simultaneous or otherwise-desired sequential execution.

[0087] Following execution of all or as many portions of routed transaction or processing segments, routing processor(s) 104, 1104 can receive from corresponding execution processor(s) 106, 1106 data confirming or otherwise indicating such execution, and by accessing data records stored in associated memory(ies), can allocate execution results to the requesting source(s) 102, 1102.

[0088] Reference is now made to FIG. 4, showing an example of a method 300 of determining timing parameters to be used in managing processing of data by multiple networked computing resources 106. In the embodiment shown, method 300 is an iterative method, and each loop of the method 300 is denoted as N. Method 300 is suitable for implementation using, for example, any of various embodiments of systems 100, 1000 and components thereof, including particularly router processor(s) 104, 1104 and data source(s) 1126.

[0089] At 302, each of a plurality of networked computing resources 106, 1106 is monitored, for example by router processor(s) 104, 1104, execution processor(s) 106, 1106, external processor(s) 1126, and/or various components or modules operated by or otherwise associated therewith, for latencies associated with receipt and/or execution of signal processing execution requests. This may be carried out, for example, by a monitoring module (e.g., an exchange RTL measurement module 1126b, such as for the financial system 1000) in the router processor(s) 104, 1104. Such monitoring may comprise, for example, time stamping outgoing requests for processing of data, and comparing times of receipt of confirmation(s) or results from processing to the corresponding time-stamped outgoing request.

The difference in time between the outgoing request and the incoming receipt confirmation and/or data processing results can be defined as a data or signal processing latency, and stored in memory accessible by the router processor(s) 104, 1104. By timing differences between outgoing requests and incoming receipts, confirmations, and/or results, such latencies can be monitored on a continual, periodic, and/or other dynamic basis.

[0090] At 306, at least one timing parameter associated with latency(ies) observed in execution of signal processing requests provided to the monitored resources 106, 1106 by the routing processor(s) 104, 1104 is determined. As described herein, such timing parameter(s) may include, for example, latencies due to communication delay, such as transmission delays or other signal propagation delays, and/or processing delays, among others. Typically, corresponding timing parameter(s) are determined for each of the plurality of networked computing resources 106, 1106 to which a transaction order or other data processing request, or a portion thereof, is expected to be sent by routing processor(s) 104, 1104.

[0091] In various embodiments, such as in various forms of financial systems 1000, and depending upon the types of system(s) to be used and desired processing results, such timing parameters may be determined for one-way and/or round-trip communications between the routing processor(s) 1104 operated by or on behalf of a capital management entity and the exchange server 1106; that is, from generation of a multi-part transaction request by capital management entity's routing processor 1104 to the receipt of a response, such as confirmation of receipt of a part of a larger trading order and/or confirmation of execution of all or part of a requested trade, from the execution resource to which the processing request was directed. With reference to FIG. 1B, for example, and explained above, an RTL measurement may include latencies due any or all of transmission of signals within the capital management entity server 1104, processing of signals within the capital management entity 1104, transmission of signals between the capital management entity 1104 and a network 1110, transmission of

signals within the network 1110, transmission of signals between the network 1110 and the targeted exchange server 1106, and processing of signals within the exchange server 1106; for both communications sent from the routing processor(s) 104, 1104 and responses (e.g., acknowledgement of communication, rejection of a trade request, confirmation of a trade request, etc.) sent from the exchange server 106, 1106. In such embodiments, the timing parameter(s) may be simply the total time for the round-trip communication, or a statistical or other mathematical function thereof.

[0092] For example, an exchange RTL measurement module 1126b, such as that associated with SOR 1104 shown in FIG. 1B, may determine a timing parameter as follows:

1) A time-stamp value T1 is associated by the processor(s) 1104 with a new communication M1 (e.g., a trade request) sent to an exchange server 1106.

2) A time-stamp value T2 is associated by the processor(s) 1104 with any response to the request M1 received from the exchange processor 1106 to which the request M1 was sent. This response can be any response such as acknowledgement, rejection, whole or partial fill, etc., and may depend on the nature of the request represented by M1.

3) The RTL associated with the request M1 is calculated as the difference between T2 and T1. In some embodiments, as noted above, RTL may be calculated as an average of the time (T2 – T1) for a past number *Z* (e.g., 30) of processing requests routed to each of a plurality of targeted exchange processor(s) 1106.

[0093] At 308, timing parameter(s) associated with each networked computing resource 106 may be stored in timing data store(s) 214. As

described herein, a timing data store 214, in some examples, may be a database or other data structure residing in a memory associated with or otherwise accessible by the router processor(s) 104. Timing parameter(s) stored in timing data store(s) 214 may be employed in processes such as those described above in connection with process block 210 of Figure 2.

[0094] Timing parameter(s) determined by processor(s) 104, 1104 may for example represent rolling histogram(s) representing latencies associated with individual execution processors 106, 1106 and/or other components of system(s) 100, 1000.

[0095] FIG. 5 shows an example of a histogram illustrating stored data representing processing latency time values associated communications and/or other processing associated with an execution processor 106, 1106 in a system 100, 1000. In the example shown, round-trip latency times (in ms) are stored for the most recent 30 transaction requests or other communications with a given execution server 106. Although the example shows 30 latency times being stored, the number of stored timing parameter(s) used in determining RTLs or other timing parameters may be greater or fewer, and may vary according to conditions such as the time of day, the season, etc. The results of calculations based on the stored latencies, and other related data, may also be stored in timing data store(s) 214. For example, in the example of FIG. 5, in addition to raw latency times, a rolling average or a rolling mode of the past 30 (or other suitable number) latency times associated with communications and/or other processing with or by each execution server 106 may also be calculated and stored in timing data store(s) 214.

[0096] As will be readily understood by those skilled in the relevant arts, further factors, including for example desired fix offsets or delays, or scaling factors associated with time of day, day of week, season of year, etc., known trading or other data processing patterns, economic conditions, etc., may be used at 210 in determining timing parameters.

[0097] Timing parameters determined at 210 can be used by routing processor(s) 104, 1104 to synchronize execution of processing requests originated by source(s) 102, 1102 and directed to processor(s) 106, 1106 by, for example, associating with such requests, or portions of them to be forwarded for execution by each of multiple processor(s) 106, 1106, data items useable by the processor(s) 104, 1104 to cause communication of the requests to the corresponding processor(s) 106, 1106 at desired absolute or relative times, to achieve desired synchronization of the arrival of the requests at the corresponding execution processor(s) 106, 1106. For example, by using data items configured to cause communication of one or more portions of the requests at given time(s) according to a clock associated with the processor(s) 104, 1104, the processor(s) 104, 1104 can cause the request(s) or request portion(s) to be communicated at a desired time of day, or in any desired relative order or sequence without regard to the actual time of day, but rather with respect to each other or some third index.

[0098] At 310, N is incremented by one, or other suitable value, or control is otherwise returned to 302 so that the process 302 – 308 continues. Optionally process 302 – 310 continues until a maximum desired number of iterations has been completed, or until all requests for transactions or other processing by orders have been processed (e.g., routed to execution processors 106, 1106), or until other suitable criteria has been met.

[0099] To aid operators and users of system(s) 100, 1000, or components thereof, understand or evaluate the effect of the disclosed method and system for causing processing of data by multiple networked computing resources, in some aspects, the present disclosure also provides various metrics (e.g., trading benchmarks, in the case of a financial system 1000) which may be determined by, and through the use of data generated from, any or all of the various components of a system 100, 1000.

[00100] Reference is now made to FIG. 6, which shows comparisons of results of transmission of multi-part trade execution requests to pluralities of networked computing resources, or execution processors 106, 1106 according to an example of the disclosed method and system, to results of conventionally-transmitted multi-part trade requests.

[00101] FIG. 6a shows results of execution of a multi-part transaction request using the disclosed methods and systems to obtain synchronized (in the illustrated case, substantially simultaneous) execution of the various parts or segments 624 of the multi-part transaction request (a sell order) by a plurality of exchange servers 106, 1106. In the example shown, a fill rate of 94% of an original aggregated order was achieved at the original offer price 630 of $4.21 (shown as "Level 1"). In a second round of transactions (which was filled in a single transaction, as shown at 626) the remaining volume was sold at a less-desired but still acceptable price 632 of $4.20 (shown as "Level 2"). The cost associated with the orders filled below the requested order price (i.e., those orders in Level 2) was $53,000 for the trader systems 1102 (e.g., client systems) and $10,049 for the capital management entity 1106.

[00102] In FIG. 6b, using prior-art trading methods and systems, an unsynchronized multi-part trade request (multi-exchange sell order) consisting of multiple, unsynchronized order segments 624' for the same overall transaction request resulted in an initial fill rate of 47% at the preferred order price 630 of $4.21 (shown as "Level 1"). A further 43% of the request was subsequently filled at the less-desirable price 632 of $4.20 (shown as "Level 2"), with the remainder being filled at a further reduced price 634 of $4.19 (shown as "Level 3").

[00103] Using methods and systems in accordance with the disclosure, a volume-weighted average sale price (VWAP) 636 of $4.2094/share was realized, as shown at 628. Using prior-art methods and systems, a VWAP 638 of $4.2038/share was realized.

[00104] As will be readily understood by those skilled in the relevant arts, systems 100, 1000 can comprise devices or components suitable for providing a wide variety of further metrics and functionalities. For example, reference is now made to FIG. 7, which illustrates two examples of the provision by a routing processor 104, 1104 or other processor of a benchmark comparison relative to a market average price provided by, for example, a market news service or other market data source 1126v. At 646, performance of a system 100, 1000 in synchronized processing of a multi-part transaction request in accordance with the invention is compared to a market performance indicator "Average Price Benchmark." Such average price benchmark, or other benchmark or metric factor, can be obtained from, for example, any or all of components 1126, 1106, etc. At 644, performance of a system 100, 1000 in un-synchronized processing of a multi-part transaction request in accordance with prior art methods is compared to the same market performance indicator "Average Price Benchmark." Comparison of comparisons 646, 644 indicates that processing of transactions in accordance with the invention provides better results for a seller of financial interests. As will be understood by those skilled in the relevant arts, a wide variety of benchmarks may be used in assessing performance of systems and methods according to the invention. Such benchmarks may be determined at least partially by the nature of the system 100, 1000 used, and the types of transactions or other execution requests processed by such system.

[00105] In the embodiment shown in Figure 1B, source(s) 1126 of data useable by processor(s) 104 in preparing financial transaction or other data processing execution requests includes a plurality of modules 1126a-g useful in preparing a multi-part execution request. In the example shown, modules 1126a-g include market data processing module 1126a, exchange round-trip latency measurement module 1126b, adaptive exchange round-trip latency (RTL) learning & compensation logic module 1126c, smart sweeping share allocation logic module 1126d, smart posting logic module 1126e, regional & national exchange access logic module 1126f, and aggressiveness management module 1126g.

[00106] Market data processing module 1126a receives and processes market data, which may be the same as or different from data provided through exchange market data module 1126v of the exchange server 1106. Sources of such data may be internal to the system 1104, or external, as needed or desired, and may include any suitable private or publicly-available sources of data useful in preparing execution requests, and particularly such requests that are useful in dividing or otherwise preparing a transaction order: information provided can, for example, include the numbers or quantities and/or prices available on any particular exchanges; historical trading volumes or prices; current and historical depth of market(s) or liquidity; reserve sizes; absolute, relative, and/or average price spreads; and stock- or interest-specific heuristics; and/or trends in any or all thereof.

[00107] Exchange RTL measurement module 1126b determines timing parameters for use in synchronizing execution of multi-part trade or other data processing requests by pluralities of exchange server 1106s, as for example explained herein, using statically-defined latency data representing time(s) elapsed between sending of requests or other data to, and receipt of confirmation or execution results from, individual execution processor(s) 106, 1106.

[00108] Adaptive Exchange RTL measurement module 1126c determines timing parameters for use in synchronizing execution of multi-part trade or other data processing requests by pluralities of exchange server 1106s, as for example explained herein, using dynamically-defined ("rolling") latency data representing times elapsed between sending of multiple processing requests, or other data, to, and receipt of confirmation or execution results from, individual execution processor(s) 106, 1106. Histograms and other data models and/or structures representing such rolling data may be used by module(s) 1126c in determining timing parameters according to such processes.

[00109] Smart sweeping share allocation logic module 1126d includes a statistical model for strategically oversizing transaction requests, and/or associating reserve quantity(ies) with publicly-posted orders, based on historically observed market data. This module 1126d determines, for example, a suitable oversizing (i.e., over-ordering on a trade request) to be incorporated in an open order, taking into consideration predicted hidden reserve quantity(ies) in an exchange server 1106, based on statistical data about the hidden reserve available in that exchange server 1106 over a given period or under other specified conditions (e.g., the past 30 trade requests). Based on such predicted hidden market reserves, a suitably-sized hidden reserve can be determined, and associated with a transaction order, to result in a strategic oversizing of the publicly-viewable order and help to ensure that an actual desired trading volume is realized.

[00110] Smart posting logic module 1126e includes a statistical model for determining the probability of fills (i.e., percentage satisfaction of a trade request) expected to be realized in trade requests routed to individual exchange servers 1106. Such statistical models may for example include historical fill data realized on such individual exchanges over a given period (e.g., the past 30 trade requests, last month, previous 12 months, etc.). A smart posting logic module 1126e may take into consideration factors including, for example, the depth of the top of book at each exchange server 1106, the volatility level across exchange servers 1106 and the mean latency time to execution of a trade request, among other factors.

[00111] Regional & national exchange access logic module 1126f provides information about how a trade request should be routed to an exchange server 1106, depending on whether the exchange server 1106 is regional or national. Internally- and/or externally- stored data related to suitable protocol(s) to be employed, regulations to be observed, etc., may be employed in providing such data. Such data may be used, for example, in ensuring that trade or other processing requests forwarded to external resources 106, 1106 by routing processor(s) 104, 1104 are suitably

formatted, in view of the resource(s) 106, 1106 to which the request(s) are provided, and in ensuring that such request(s) comply with all applicable legal standards.

[00112] Aggressiveness management logic module 1126g includes a probability model for determining the probability of a fill percentage for individual exchange servers 1106, and modifying execution requests routed to such servers accordingly. Such a module 1126g may take into consideration factors such as, for example, the fill rate at each exchange server 1106, the depth of book at each exchange server 1106, and the volatility levels across exchange servers 1106, among other factors.

[00113] While the disclosure has been provided and illustrated in connection with specific, presently-preferred embodiments, many variations and modifications may be made without departing from the spirit and scope of the invention(s) disclosed herein. The disclosure and invention(s) are therefore not to be limited to the exact components or details of methodology or construction set forth above. Except to the extent necessary or inherent in the processes themselves, no particular order to steps or stages of methods or processes described in this disclosure, including the Figures, is intended or implied. In many cases the order of process steps may be varied without changing the purpose, effect, or import of the methods described. The scope of the claims is to be defined solely by the appended claims, giving due consideration to the doctrine of equivalents and related doctrines.

[0114] Also disclosed is a system for causing synchronized processing of data by multiple networked computing resources, the system comprising at least one processor configured to execute machine-interpretable instructions and causing the system to: receive from one or more data sources signals representing instructions for execution of at least one data process executable by a plurality of networked computing resources; divide the at least one data process into a plurality of data processing segments, each data processing segment to be routed to a different one of a plurality of networked execution processors; based at least partly on latencies in execution of prior data processing requests routed by the system to each of the plurality of networked execution processors, determine a plurality of timing parameters, each of the plurality of timing parameters to be associated with a corresponding one of the plurality of data processing segments, the plurality of timing parameters determined to cause synchronized execution of the plurality of data processing segments by the plurality of networked execution processors; and using the timing parameters associated with the plurality of data processing segments, route the plurality of data processing segments to the plurality of corresponding networked execution processors.

[0115] In the system, at least one of the plurality of determined timing parameters may be determined based at least partly on dynamically-monitored latency in execution of signal processing requests routed by the system to at least one of the plurality of networked execution processors.

[0116] At least one of the plurality of determined timing parameters may be determined based at least partly on at least one of: communication delay and processing delay.

[0117] At least one of the plurality of determined timing parameters may be determined based at least partly on a latency probability model.

[0118] The networked execution processors may comprise exchange servers and the data processing segments represent requests for trades in financial interests.

[0119] The financial interests may include at least one of commodities and intangible interests.

[0120] The financial interests may include at least one of stocks, bonds, and options.

[0121] The at least one processor may be further configured to execute instructions for causing the system to: associate with each of at least one of the plurality of data processing segments data representing at least one quantity term, the at least one quantity term representing at least one quantity of a financial interest to be traded in accordance with a request represented each of the at least one data processing segments, and at least one corresponding price term associated with each such quantity term, the quantity term representing at least one proposed price at which a trade represented by the at least one data processing segment is to be executed; the at least one quantity term larger than at least one quantity of the financial interest publicly offered at a price

equivalent to the corresponding associated price term, in a market associated with the networked execution processor(s) to which the at least one data processing segment is to be routed.

[0122] This at least one quantity term may be determined based at least partly on a trading history associated with the market associated with the networked execution processor to which the data processing segment is to be routed.

[0123] Also disclosed is a system for causing synchronized processing of data by multiple networked computing resources, the system comprising at least one processor configured to execute machine-interpretable instructions and causing the system to: monitor execution of signal processing execution requests by each of the plurality of networked computing resources; determine at least one timing parameter associated with a latency in execution of signal processes between the system and each of the plurality of networked computing resources; and store the at least one timing parameter in machine-readable memory accessible by the at least one processor.

[0124] This at least one latency may be associated with at least one of: communication delay and processing delay.

[0125] The execution of signal processing requests may be monitored periodically.

[0126] This execution of signal processing execution requests may be monitored continually.

[0127] The networked execution processors may comprise exchange servers and the data processing segments represent requests for trades in financial interests.

[0128] Also disclosed is a method of causing synchronized processing of data by multiple networked computing resources, the method performed by at least one processor executing machine-interpretable instructions configured to cause the at least one processor to: receive from one or more data sources signals representing instructions for execution of at least one data process executable by a plurality of networked computing resources; divide the at least one data process into a plurality of data processing segments, each data processing segment to be routed to a different one of a plurality of networked execution processors; based at least partly on latencies in execution of prior data processing requests routed by the system to each of the plurality of networked execution processors, determine a plurality of timing parameters, each of the plurality of timing parameters to be associated with a corresponding one of the plurality of data processing segments, the plurality of timing parameters determined to cause synchronized execution of the plurality of data processing segments, by the plurality of networked execution processors; and using the timing parameters associated with the plurality of data processing segments, route the plurality of data processing segments to the plurality of corresponding networked execution processors.

[0129] The machine-interpretable instructions may be further configured to cause the at least one processor to: associate with each of at least one of the plurality of data processing segments data representing at least one quantity term, the at least one

quantity term representing at least one quantity of a financial interest to be traded in accordance with a request represented each of the at least one data processing segments, and at least one corresponding price term associated with each such quantity term, the quantity term representing at least one proposed price at which a trade represented by the at least one data processing segment is to be executed; the at least one quantity term larger than at least one quantity of the financial interest publicly offered at a price equivalent to the corresponding associated price term, in a market associated with the networked execution processor(s) to which the at least one data processing segment is to be routed.

[0130] Also disclosed is a method of causing synchronized processing of data by multiple networked computing resources, the method performed by at least one processor executing machine-interpretable instructions configured to cause the at least one processor to: monitor execution of signal processing execution requests by each of the plurality of networked computing resources; determine at least one timing parameter associated with a latency in execution of signal processes between the system and each of the plurality of networked computing resources; and store the at least one timing parameter in machine-readable memory accessible by the at least one processor.

[0131] Also disclosed is a computer-readable medium or media comprising non-transient machine-readable programming structures configured to cause synchronized processing of data by multiple networked computing resources, the programming structures configured to cause at least one processor to: receive from one or more data sources signals representing instructions for execution of

at least one data process executable by a plurality of networked computing resources; divide the at least one data process into a plurality of data processing segments, each data processing segment to be routed to a different one of a plurality of networked execution processors; based at least partly on latencies in execution of prior data processing requests routed by the system to each of the plurality of networked execution processors, determine a plurality of timing parameters, each of the plurality of timing parameters to be associated with a corresponding one of the plurality of data processing segments, the plurality of timing parameters determined to cause synchronized execution of the plurality of data processing segments by the plurality of networked execution processors; and using the timing parameters associated with the plurality of data processing segments, route the plurality of data processing segments to the plurality of corresponding networked execution processors.

[0132] In this medium or media, the machine-interpretable instructions may be further configured to cause the at least one processor to: associate with each of at least one of the plurality of data processing segments data representing at least one quantity term, the at least one quantity term representing at least one quantity of a financial interest to be traded in accordance with a request representing each of the at least one data processing segments, and at least one corresponding price term associated with each such quantity term, the quantity term representing at least one proposed price at which a trade represented by the at least one data processing segment is to be executed; the at least one quantity term larger than at least one quantity of the financial interest publicly offered at a price equivalent to the corresponding

associated price term, in a market associated with the networked execution processor(s) to which the at least one data processing segment is to be routed.

[0133] Also disclosed is a computer-readable medium or media comprising non-transient machine-readable programming structures configured to cause synchronized processing of data by multiple networked computing resources, the programming structures configured to cause at least one processor to: monitor execution of signal processing execution requests by each of the plurality of networked computing resources; determine at least one timing parameter associated with a latency in execution of signal processes between the system and each of the plurality of networked computing resources; and store the at least one timing parameter in machine-readable memory accessible by the at least one processor.

CLAIMS

1. A system for coordinating processing of data by multiple networked computing resources, the system comprising at least one processor configured to:

receive or generate from one or more data sources signals representing instructions for execution of a plurality of proposed transactions in one or more financial interests by a plurality of networked computing resources;

associate each of the plurality of proposed transactions with a corresponding networked computing resource;

determine a timing sequence for routing the plurality of proposed transactions, the timing sequence defining a sequence and delays between times that the plurality of proposed transactions are to be routed, the timing sequence determined such that a difference between:

an execution or arrival time of a proposed transaction of the plurality of proposed transactions which executes or arrives the earliest at its corresponding networked computing resource, and

an execution or arrival time of a proposed transaction of the plurality of proposed transactions which executes or arrives the latest at its corresponding networked computing resource

is less than a defined time threshold; and

route, in the timing sequence, the plurality of proposed transactions to their respective corresponding networked computing resource for execution.

2. The system of claim 5, wherein the at least one processor is configured to : receive from the one or more data sources signals representing instructions for execution of at least one data process executable by the plurality of networked computing resources; and

divide the at least one data process into the plurality of proposed transactions to be routed to their respective corresponding networked computing resource.

3. A method for coordinating processing of data by multiple networked computing resources, the method performed by at least one processor configured to cause the at least one processor to:

receive or generate from one or more data sources signals representing instructions for execution of a plurality of proposed transactions in one or more financial interests by a plurality of networked computing resources;

associate each of the plurality of proposed transactions with a corresponding networked computing resource;

determine a timing sequence for routing the plurality of proposed transactions, the timing sequence defining a sequence and delays between times that the plurality of proposed transactions are to be routed, the timing sequence determined such that a difference between:

an execution or arrival time of a proposed transaction of the plurality of proposed transactions which executes or arrives the earliest at its corresponding networked computing resource, and

an execution or arrival time of a proposed transaction of the plurality of proposed transactions which executes or arrives the latest at its corresponding networked computing resource

is less than a defined time threshold; and

route, in the timing sequence, the plurality of proposed transactions to their respective corresponding networked computing resource for execution.

4. The method of claim 3, comprising:

receiving from the one or more data sources signals representing instructions for execution of at least one data process executable by the plurality of networked computing resources; and

dividing the at least one data process into the plurality of proposed transactions to be routed to their respective corresponding networked computing resource.

5. A non-transitory, computer-readable medium or media having stored thereon computer readable instructions which when executed by at least one processor cause at least one processor to:

receive or generate from one or more data sources signals representing instructions for execution of a plurality of proposed transactions in one or more financial interests by a plurality of networked computing resources;

associate each of the plurality of proposed transactions with a corresponding networked computing resource;

determine a timing sequence for routing the plurality of proposed transactions, the timing sequence defining a sequence and delays between times that the plurality of proposed transactions are to be routed, the timing sequence determined such that a difference between:

an execution or arrival time of a proposed transaction of the plurality of proposed transactions which executes or arrives the earliest at its respective networked computing resource, and

an execution or arrival time of a proposed transaction of the plurality of proposed transactions which executes or arrives the latest at its corresponding networked computing resource

is less than a defined time threshold; and

route, in the timing sequence, the plurality of proposed transactions to their respective corresponding networked computing resource for execution.

6. The medium or media of claim 5, wherein the machine-interpretable instructions are configured to cause the at least one processor to:

receive from the one or more data sources signals representing instructions for execution of at least one data process executable by the plurality of networked computing resources; and

divide the at least one data process into the plurality of proposed transactions to be routed to their respective corresponding networked computing resource.

7. The method of claim 3, wherein the defined time threshold is defined such that execution or arrival of the proposed transactions can occur before counterparty terms associated with one or more of the proposed transactions can be changed.

8. The method of claim 3, wherein the defined time threshold is defined to reduce an amount of time in which counterparty terms associated with one or more of the proposed transactions can be changed.

9. The method of claim 3, comprising determining the timing sequence based at least partly on latencies in execution or arrival of prior proposed transactions.

10. The method of claim 9, wherein the timing sequence is determined based at least partly on rolling historical latencies in execution.

11. The method of claim 9, comprising: monitoring latencies associated with routed proposed transactions and their corresponding networked computing resources.

12. The method of claim 3, wherein the associating or routing of proposed transactions is based at least in part on historical fill rates or volatility of at least one of the networked computing resources.

13. The system of claim 1, wherein the defined time threshold is defined such that execution or arrival of the proposed transactions can occur before counterparty terms associated with one or more of the proposed transactions can be changed.

14. The system of claim 1, wherein the defined time threshold is defined to reduce an amount of time in which counterparty terms associated with one or more of the proposed transactions can be changed.

15. The system of claim 1, wherein the at least one processor is configured to: determine the timing sequence based at least partly on latencies in execution or arrival of prior proposed transactions.

16. The system of claim 15, wherein the timing sequence is determined based at least partly on rolling historical latencies in execution.

17. The system of claim 15, wherein the at least one processor is configured to: monitor latencies associated with routed proposed transactions and their corresponding networked computing resources.

18. The system of claim 1, wherein the associating or routing of proposed transactions is based at least in part on historical fill rates or volatility of at least one of the networked computing resources.