

CLAIMS

1. A distributed computer system comprising:
 - electronic memory configured to store a data structure that includes at least two different types of data transaction requests, wherein a first type of data transaction requests is contra-sided to a second type of transaction requests;
 - a transceiver configured to receive data transaction requests for processing by the distributed computer system;
 - a processing system that includes at least one hardware processor, the processing system configured to:
 - receive, via the transceiver, a first data transaction request that is of the first type of data transaction requests, the first data transaction request including a first value for a first parameter;
 - based on reception of the first data transaction request, set a status of the first data transaction request to a first status;
 - based on reception of the first data transaction request, perform a comparison that compares the first value of the first data transaction request to a second value;
 - based on the performed comparison, activate a timer that indicates how long the first data transaction request will remain in the first status;
 - during a time period in which the first data transaction request is set to the first status, perform a matching process between at least two different data transaction requests that include one of the first type and one of the second type, wherein the

performed matching process does not take into account those data transaction requests with a first status;

based on expiration of the timer, change the status of the first data transaction request to a second status; and

subsequent to changing the status of the first data transaction request to the second status, perform a second instance of the matching process, wherein the first data transaction request is taken into account for performance of the second instance of the matching process.

2. The distributed computer system of claim 1, wherein the processing system is further configured to:

obtain a reference value that indicates a current state of the data structure that includes at least two different types of data transaction requests, wherein the reference value is the second value that is used in the performed comparison; and

wherein activation of the timer is delayed until the performed comparison indicates that the first value of the first data transaction request is outside of the reference value.

3. The distributed computer system of claim 2, wherein the data structure includes a pre-pending list, a pending list, an executable list, and a passive list,

wherein the first data transaction request is placed into the pre-pending list until the first value is inside of an obtained reference value,

wherein the first data transaction request is placed into the pending list while the timer is activated,

wherein the first data transaction request is placed, after expiration of the timer, into the passive list in response to determination that the first value is outside of an obtained reference value,

wherein the matching process performs the matching process by taking in account those data transaction requests that are in the executable list.

4. The distributed computer system of claim 2, wherein the reference value is obtained from an external computer system.

5. The distributed computer system of claim 1, wherein the processing system is further configured to:

after activation of the timer and before expiration of the timer, adjust the timer for the first data transaction request based on determination that the first value is outside of a newly obtained reference value.

6. The distributed computer system of claim 5, wherein adjustment of the timer includes at least one of:

resetting the timer and only restarting the timer upon determination that the first value is inside of the reference value,

pausing the timer until determination that the first value is inside of the reference value, and

canceling the timer.

7. The distributed computer system of claim of claim 1, further comprising:
a plurality of computing nodes, which include at least one sequencer node and a plurality of matcher nodes, the sequencer node configured to sequence electronic data messages that are distributed across the distributed computer system, the matcher nodes configured to perform matching processes, wherein a first matcher node of the plurality of matcher nodes is configured to perform the second instance of the matching process, and wherein each of the plurality of computing nodes comprise at least one hardware processor.

8. The distributed computer system of claim of claim 7, wherein the first matcher node is configured to store a plurality of different data structures for each one of a plurality of different resource identifiers, wherein different matching processes are performed using each one of the plurality of different data structures for the first matcher node,

wherein electronic data transaction requests for each of the plurality of different resources identifiers that have active timers are stored in the same data structure,

wherein the matcher node is configured to, in response to expiration of a timer for an electronic data transaction request, move the corresponding electronic data transaction request to a corresponding one of the plurality of different data structures that is associated with the resource identifier of the corresponding electronic data transaction request.

9. A method of matching electronically received data transaction requests by a distributed computer system that includes electronic memory, a transceiver, and a processing system that includes at least one processor coupled to the electronic memory and the transceiver, the method comprising:

storing a data structure that includes at least two different types of data transaction requests, wherein a first type of data transaction requests is contra-sided to a second type of transaction requests;

receiving, via the transceiver, a first data transaction request that is of the first type of data transaction requests, the first data transaction request including a first value for a first parameter;

based on receiving the first data transaction request:

setting a status of the first data transaction request to a first status, and

comparing the first value of the first data transaction request to a second value that is not included with the first data transaction request;

based on how the first value compares to the second value, starting a timer that indicates how long the first data transaction request will remain in the first status;

during a time period in which the first data transaction request is set to the first status, executing a first instance of a matching process between at least two different data transaction requests that include one of the first type and one of the second type, wherein the performed matching process does not take into account those data transaction requests with a first status;

based on expiration of the timer, changing the status of the first data transaction request to a second status; and

subsequent to changing the status of the first data transaction request to the second status, executing a second instance the matching process, wherein the first data transaction request is taken into account for performance of the second instance of the matching process.

10. The method of claim 9, further comprising:

obtaining a reference value that indicates a current state of the data structure that includes at least two different types of data transaction requests, wherein the reference value is the second value that is used in the performed comparison; and

prior to activation of the timer and based on determination that the first value of the first data transaction is outside of the reference value, waiting to start the timer for the first data transaction request until the first value is inside of an obtained reference value.

11. The method of claim 10, wherein the data structure includes a pre-pending list, a pending list, an executable list, and a passive list,

wherein the first data transaction request is placed into the pre-pending list until the first value is inside of an obtained reference value,

wherein the first data transaction request is placed into the pending list while the timer is started,

wherein the first data transaction request is placed, after expiration of the timer, into the passive list in response to determination that the first value is outside of an obtained reference value,

wherein the matching process performs the matching process by taking in account those data transaction requests that are in the executable list.

12. The method of claim 10, wherein the reference value is received via an electronic data message sent from an external computer system.

13. The method of claim 9, further comprising:
after activation of the timer and before expiration of the timer, adjusting the timer for the first data transaction request based on determination that the first value is outside of second reference value.

14. The method of claim 13, wherein adjustment of the timer includes at least one of:

resetting the timer and only restarting the timer upon determination that the first value is inside of the reference value,

pausing the timer until determination that the first value is inside of the reference value, and

canceling the timer.

15. The method of claim 9, wherein the distributed computer system includes a plurality of computing nodes, which include at least one sequencer node and a plurality of matcher nodes, the sequencer node sequencing electronic data messages that are distributed across the distributed computer system, the matcher nodes executing matching processes, wherein a first matcher node of the plurality of matcher nodes is configured to perform the second instance of the matching process, and wherein each of the plurality of computing nodes comprise at least one hardware processor.

16. The method of claim 15, further comprising:

- storing, on the first matcher node, a plurality of different data structures for each one of a plurality of different resource identifiers;
- executing, on the first matcher node, different matching processes for each one of the plurality of different data structures;
- storing those electronic data transaction requests for each of the plurality of different resources identifiers that have active timers to a different data structure; and
- moving, in response to expiration of a timer for a corresponding electronic data transaction request that is stored in the different data structure, the corresponding electronic data transaction request to a corresponding one of the plurality of different data structures that is associated with the resource identifier of the corresponding electronic data transaction request.

17. A non-transitory computer readable storage medium having stored thereon computer readable instructions for use with a distributed computer system that includes at

least one processor, electronic memory, and a transceiver, the electronic memory configured to store a data structure that includes at least two different types of data transaction requests, wherein a first type of data transaction requests is contra-sided to a second type of transaction requests, the stored computer readable instructions comprising instructions that cause the computer system to:

receive, via the transceiver, a first data transaction request that is of the first type of data transaction requests, the first data transaction request including a first value for a first parameter;

in response to reception of the first data transaction request, set a status of the first data transaction request to a first status;

for the received first data transaction request, perform a comparison by using at least the first value of the first data transaction request;

based on the performed comparison, activate a timer that indicates how long the first data transaction request will remain in the first status;

during a time period in which the first data transaction request is set to the first status, perform a first instance of a matching process between at least two different data transaction requests that include one of the first type and one of the second type, wherein the performed matching process does not take into account those data transaction requests that have the first status;

based on expiration of the timer, change the status of the first data transaction request to a second status; and

subsequent to changing the status of the first data transaction request to the second status, perform a second instance of the matching process, wherein the first data

transaction request is taken into account for performance of the second instance of the matching process.

18. The non-transitory computer readable storage medium of claim 17, wherein the stored instructions comprise further instructions that cause the computer system to:

obtain a reference value that indicates a current state of the data structure that includes at least two different types of data transaction requests, wherein the comparison is performed between the first value and the reference value.

19. The non-transitory computer readable storage medium of claim 17, wherein the data structure includes a pre-pending list, a pending list, an executable list, and a passive list,

wherein the first data transaction request is placed into the pre-pending list until the first value is inside of an obtained reference value,

wherein the first data transaction request is placed into the pending list while the timer is activated,

wherein the first data transaction request is placed, after expiration of the timer, into the passive list in response to determination that the first value is outside of an obtained reference value,

wherein the matching process performs the matching process by taking in account those data transaction requests that are in the executable list.

20. The non-transitory computer readable storage medium of claim 18, wherein the performed comparison that uses the first value of the first data transaction request is performed before, during, or after the performed first instance of the matching process.