

What is claimed is:

1. A method for distributed processing of a data collection, the method including:

receiving, over an input device or port, information for configuring a distributed processing system, the configuring including forming a first configuration of components of the distributed processing system, the first configuration including a plurality of sources of data elements of the data collection and a plurality of workers configured to process data elements of the data collection, the sources and workers being distributed among a plurality of computing resources, wherein each data element includes a data partition value that identifies a subset of the plurality of workers of the first configuration according to a data partition rule of the first configuration; and

processing data in the distributed processing system during at least two processing epochs, the processing including:

accepting data elements from the sources for a first part of the data collection in a first processing epoch and routing said data elements through the first configuration and completing processing of at least some of said data elements, wherein other of the data elements of the first part remain queued at components of the first configuration;

after accepting a first part of the data collection, initiating change of configuration of the distributed processing system from the first configuration to a second configuration;

after initiating the change of configuration, performing a succession of two or more transitions between configurations of components of the system to a succession of modified configurations of components, and after each transition causing transfer of data elements between components of the modified configuration, wherein a last of said modified configurations corresponds to the second configuration, thereby completing a transition from the first configuration to the second configuration; and

after completing a transition to the second configuration, accepting further data elements of the data collection from a plurality of sources of the second configuration in a second processing epoch.

2. The method of claim 1 wherein the plurality of computing resources includes a plurality of processors coupled via communication links.

3. The method of claim 2 wherein the plurality of computing resources includes at least one processing thread executing on each of the plurality of processors, each computing resource being associated with a distinct processing thread.

4. The method of any previous claim wherein each source of data elements is coupled to a partitioner module configured to accept data elements from the source, and wherein each partitioner is configured with the data partition rule to direct data elements to a worker identified according to the data partition rule.

5. The method of claim 4 wherein performing a first transition of the succession of two or more transitions between configurations of components of the system includes

halting operation of the partitioner modules, stopping of acceptance of data elements from the sources at the partitioner modules, reconfiguring the plurality of partition modules with a modified partition rule, and coupling at least one queue of data elements accepted from a source to provide data elements to a partition module reconfigured with the modified partition.

6. The method of claim 4 wherein each partitioner module is hosted on a same computing resource as a source coupled to said partitioner module, wherein passing data elements from said source to said partitioner is performed without requiring inter-processor communication.

7. The method of claim 4 or 6 wherein the plurality of workers includes one or more workers each worker of said one or more workers being coupled to an acceptor module configured to accept data elements from a plurality of partitioner modules.

8. The method of claim 7 wherein performing a first transition of the succession of two or more transitions between configurations of components of the system includes

halting operation of the partitioner modules, stopping of acceptance of data elements from the sources at the partitioner modules,
halting operation of the plurality of acceptor modules,
reconfiguring the plurality of partition modules with a modified partition rule, and
coupling at least one queue of an acceptor module of the plurality of acceptor modules to provide data elements to a partition module reconfigured with the modified partition.

9. The method of claim 7 or 8 wherein each acceptor module is hosted on a same computing resource as a worker coupled to said acceptor module, wherein passing data elements from said acceptor module to said worker is performed without requiring inter-processor communication.

10. The method of any of claims 7 to 9 wherein a first partitioner module is hosted on a same computing resource as a first acceptor module, and is hosted on a different computing resource than a second acceptor module, and wherein routing the data elements includes passing data elements from the first partitioner module to the first acceptor module without requiring inter-processor communication, and wherein routing the data elements includes queueing data elements at the first partitioner module prior to inter-processor communication of said data elements for passing to the second acceptor module.

11. The method of any previous claim wherein the plurality of workers includes one or more workers each worker of said one or more workers being coupled to an acceptor module configured to accept data elements from a plurality of partitioner modules.

12. The method of claim 11 wherein data elements are received from any one of the partitioner modules in a first-in-first-out order.

13. The method of any previous claim wherein each data element further includes a serialization value, and wherein during the first processing epoch, processing using the first configuration enforces a serialization policy whereby no two data elements with a same serialization value are processed by a worker concurrently with one another.

14. The method of claim 13 wherein the plurality of workers includes one or more workers each worker of said one or more workers being coupled to an acceptor module configured to accept data elements from a plurality of partitioner modules, the acceptor module being configured to enforce a serialization policy whereby no two data elements with a same serialization value are processed by a worker coupled to said acceptor module concurrently with one another.

15. The method of claim 13 or 14 wherein after the first processing epoch and prior to the second processing epoch, processing according to each modified configuration continues to enforce the serialization policy.

16. The method of any previous claim wherein during the first processing epoch, processing using the first configuration of components enforces a partition policy whereby all data elements with a same data partition value accepted from a first data source in a first order are provided to a same subset of the plurality of workers in the first order.

17. The method of claim 16 wherein after the first processing epoch and prior to the second processing epoch, data elements of the first part of the data that have not completed processing in the first processing epoch and with the same data partition value accepted from the first data source are provided to the same worker in the first order.

18. The method of claim 17 wherein at least some of said data elements are transferred between components of the modified configurations.

19. The method of any previous claim wherein the second configuration of components differs from the first configuration in at least one of: (a) a partition rule; (b) a set of sources; and (c) a set of workers.

20. Software stored in non-transitory form on a machine-readable medium, for distributed processing of a data collection, the software including instructions that, when executed by one or more processors of a data processing system, cause the data processing system to perform the method of any one of claims 1 to 19.

21. A distributed processing system, the distributed processing system including a plurality of processing engines and configured to execute configurations of components distributed among said processing engines according to a first configuration of components, and to enable a transition to a second configuration of components via a succession of transitions between configurations of components of the system to a succession of modified configurations of components, wherein the first configuration of components includes:

- a plurality of sources of data elements of the data collection and a plurality of workers configured to process data elements of the data collection, the sources and workers being distributed among a plurality of processing engines, wherein each data element includes a data partition value that identifies a subset of the plurality of workers of the first configuration according to a data partition rule of the first configuration;

- a plurality of partitioner modules, each partitioner module being configured to accept data elements from the source, and wherein each partitioner is configured with the data partition rule to direct data elements to a worker identified according to the data partition rule; and

a plurality of accepter modules, each accepter module configured to accept data elements from a plurality of partitioner modules, and to provide data elements to at least one worker;

wherein the system is configured to change from the first configuration to the second configuration by performing a succession of two or more transitions between configurations of components of the system to a succession of modified configurations of components, and after each transition causing transfer of data elements between components of the modified configuration, wherein a last of said modified configurations corresponds to the second configuration, thereby completing a transition from the first configuration to the second configuration.