

CLAIMS

What is claimed is:

1. At least one non-transitory computer-readable storage medium storing processor-executable instructions that, when executed by at least one computer hardware processor, cause the at least one computer hardware processor to perform:

obtaining an automatically generated initial dataflow graph (1000) , the initial dataflow graph comprising a first plurality of nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a first plurality of data processing operations and a first plurality of links representing flows of data among nodes in the first plurality of nodes;

generating an updated dataflow graph by iteratively applying dataflow graph optimization rules to update the initial dataflow graph, the updated dataflow graph comprising a second plurality of nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a second plurality of data processing operations and a second plurality of links representing flows of data among nodes in the second plurality of nodes, wherein the second plurality of nodes includes a node representing a first data processing operation and another node representing the second data processing operation; wherein generating the updated dataflow graph comprises:

selecting the first optimization rule;

identifying a first portion of the initial dataflow graph (1000) to which to apply the first optimization rule, wherein identifying the first portion of the initial dataflow graph (100) is performed using a dataflow subgraph pattern matching

language, the dataflow subgraph pattern matching language including one or more expressions for identifying a second portion of the initial dataflow graph (1000) with said second portion having commuting nodes, wherein identifying the first portion of the initial data flow graph (1000) is performed using the one or more expressions; and

applying the first optimization rule to the first portion of the initial dataflow graph (100); and

executing the updated dataflow graph at least in part by executing the first data processing operation using a first computer system process and executing the second data processing operation using a second computer system process different from the first computer system process.

2. The at least one non-transitory computer-readable storage medium of claim 1, wherein the processor-executable instructions further cause the at least one computer hardware processor to perform:

assigning a processing layout to each of one or more nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) of the updated dataflow graph, preferably wherein executing the updated dataflow graph is performed in accordance with the assigned one or more processing layouts.

3. The at least one non-transitory computer-readable storage medium of claim 1, wherein the second plurality of nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) has fewer nodes than the first plurality of nodes preferably wherein the second plurality of links has fewer links than the first plurality of links.

4. The at least one non-transitory computer-readable storage medium of claim 3, wherein generating the updated dataflow graph further comprises:

- selecting a second optimization rule different from the first optimization rule;
- identifying a second portion of the initial dataflow graph (1000) to which to apply the second optimization rule; and
- applying the second optimization rule to the second portion of the initial dataflow graph.

5. The at least one non-transitory computer-readable storage medium of claim 1, wherein identifying the first portion of the initial dataflow graph comprises identifying a first node (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a first data processing operation that commutes with a second data processing operation represented by a second node connected to the first node.

6. The at least one non-transitory computer-readable storage medium of claim 1, wherein applying the first optimization rule comprises applying an optimization selected from the group consisting of: removal of a redundant data processing operation, a strength reduction optimization, a combining operations optimization, a width reduction optimization, and a deduplication optimization.

7. The at least one non-transitory computer-readable storage medium of claim 1, wherein generating an updated dataflow graph comprises identifying a first node (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a redundant operation and removing the first node from the initial dataflow graph (1000).

8. The at least one non-transitory computer-readable storage medium of claim 1, wherein generating an updated dataflow graph comprises identifying two nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) in the initial dataflow graph (1000) representing respective sort data processing operations and replacing the two nodes with a single node (836) representing a sort data processing operation, preferably wherein the two nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) are not adjacent to each other in the initial dataflow graph.

9. The at least one non-transitory computer-readable storage medium of claim 1, wherein generating an updated dataflow graph comprises replacing a first node (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a first data processing operation with a second node representing a second data processing operation of a weaker type than the first data processing operation, or wherein generating an updated dataflow graph comprises identifying two nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) in the initial dataflow graph (1000) representing respective join data processing

operations and replacing the two nodes with a single node (836) representing a join data processing operation.

10. The at least one non-transitory computer-readable storage medium of claim 1, wherein automatically generating the initial dataflow graph (1000) comprises:

- obtaining a structured query language (SQL) query;
- generating a query plan for the SQL query; and
- generating the initial dataflow graph using the query plan.

11. The at least one non-transitory computer readable storage medium of claim 1, wherein the dataflow subgraph pattern matching language includes on or more further expressions for identifying the first portion of the initial dataflow graph (1000) for the application of the first optimization rule.

12. The at least one non-transitory computer-readable storage medium of claim 11, wherein the one or more further expressions are for identifying a series of nodes of at least a threshold length representing a respective series of calculations that are suitable to be combined and represented by a single node using combining operations optimization rule.

13. The at least one non-transitory computer-readable storage medium of claim 1 or 11, wherein the processor-executable instructions further cause the at least one computer hardware processor to perform: commuting the commuting nodes for the applying of the first optimization rule to the first portion of the initial dataflow graph (1000).

14. A method, comprising:

- using at least one computer hardware processor to perform:

- obtaining an automatically generated initial dataflow graph, the initial dataflow graph comprising a first plurality of nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072,

1074, 1080, 1088) representing a first plurality of data processing operations and a first plurality of links representing flows of data among nodes in the first plurality of nodes;

generating an updated dataflow graph by iteratively applying dataflow graph optimization rules to update the initial dataflow graph, the updated dataflow graph comprising a second plurality of nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a second plurality of data processing operations and a second plurality of links representing flows of data among nodes in the second plurality of nodes, wherein the second plurality of nodes includes a node representing a first data processing operation and another node representing the second data processing operation, wherein generating the updated dataflow graph comprises:

selecting a first optimization rule;

identifying a first portion of the initial dataflow graph (1000) to which to apply the first optimization rule, wherein identifying the first portion of the initial dataflow graph (1000) is performed using a dataflow subgraph pattern matching language, the dataflow subgraph pattern matching language including one or more expressions for identifying a second portion of the initial dataflow graph (1000) with said second portion having commuting nodes, wherein identifying the first portion of the initial data flow graph (1000) is performed using the one or more expressions and applying the first optimization rule to the first portion of the initial data flow graph (1000); and

executing the updated dataflow graph at least in part by executing the first data processing operation using a first computer system process and executing the second data processing operation using a second computer system process different from the first computer system process.

15. A data processing system (100), comprising:
at least one computer hardware processor; and
at least one non-transitory computer-readable storage medium storing processor-executable instructions that, when executed by the at least one computer hardware processor, cause the at least one computer hardware processor to perform:

obtaining an automatically generated initial dataflow graph (1000), the initial dataflow graph comprising a first plurality of nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a first plurality of data processing operations and a first plurality of links representing flows of data among nodes in the first plurality of nodes;

generating an updated dataflow graph by iteratively applying dataflow graph optimization rules to update the initial dataflow graph, the updated dataflow graph comprising a second plurality of nodes (302, 304, 311, 312, 322, 401, 403, 411, 413, 502, 504, 508, 512, 514, 516, 522, 524, 526, 532, 534, 536, 602, 604, 702, 704, 706, 708, 802, 804, 806, 810, 812, 816, 826, 828, 832, 834, 902, 904, 906, 908, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1022, 1024, 1026, 1030, 1032, 1036, 1040, 1044-1046, 1048, 1054, 1056, 1058, 1064, 1070, 1072, 1074, 1080, 1088) representing a second plurality of data processing operations and a second plurality of links representing flows of data among nodes in the second plurality of nodes, wherein the second plurality of nodes includes a node representing a first data processing operation and another node representing the second data processing operation, wherein generating the updated dataflow graph comprises:

selecting a first optimization rule;

identifying a first portion of the initial dataflow graph (1000) to which to apply the first optimization rule, wherein identifying the first portion of the initial dataflow graph (1000) is performed using a dataflow subgraph pattern matching language, the dataflow subgraph pattern

matching language including one or more expressions for identifying a second portion of the initial dataflow graph (1000) with said second portion having commuting nodes, wherein identifying the first portion of the initial data flow graph (1000) is performed using the one or more expressions and applying the first optimization rule to the first portion of the initial data flow graph (1000); and

executing the updated dataflow graph at least in part by executing the first data processing operation using a first computer system process and executing the second data processing operation using a second computer system process different from the first computer system process.