WHAT IS CLAIMED IS:

1. A computer-implemented method for transforming a first dataflow graph into a second dataflow graph, wherein the first dataflow graph includes a plurality of first nodes representing a plurality of first computer operations, wherein the second dataflow graph includes a plurality of second nodes representing a plurality of second computer operations, including:

generating the first dataflow graph with the plurality of first nodes representing first computer operations in processing data, with at least one of the first computer operations being a declarative operation that specifies one or more characteristics of one or more results of processing of data;

identifying a pattern among at least some declarative operations represented in the first dataflow graph;

based on the pattern identified among at least some declarative operations, transforming the first dataflow graph into the second dataflow graph for processing data in accordance with the first computer operations, the second dataflow graph including the plurality of second nodes representing second computer operations, with at least one of the second nodes representing one or more imperative operations that implement the logic specified by the declarative operation; and

storing, in a data store, the second dataflow graph.

2. The computer-implemented method of claim 1, wherein transforming the first dataflow graph into the second dataflow graph includes:

creating an imperative operation; and

creating a given second node that represents the imperative operation, with the given second node being unrepresented in the first dataflow graph.

3. The computer-implemented method of claim 1, wherein one or more of the second computer operations at least:

(i) are required for processing data in accordance with one or more of the first computer operations specified in the first dataflow graph, or

(ii) improve processing data in accordance with one or more of the first computer operations specified in the first dataflow graph, relative to processing data without the one or more second computer operations.

4. The computer-implemented method of claim 1, further including:

transforming the second dataflow graph into an optimized dataflow graph by applying one or more dataflow graph optimization rules to the second dataflow graph to improve a computational efficiency of the second dataflow graph, relative to a computational efficiency of the second dataflow graph prior to the applying.

5. The computer implemented method of claim 4, wherein the one or more dataflow graph optimization rules include at least one of removing a redundant node from the second dataflow graph, removing a dead node from the second dataflow graph, changing an order of nodes in the second dataflow graph, reducing a strength of a node in the second dataflow graph, combining two or more nodes in the second dataflow graph, transforming a node in the second dataflow graph from serial operation to parallel operation, or inserting a partition operation in the second dataflow graph.

6. The computer-implemented method of claim 1, wherein at least one of the second computer operations includes an automatic parallelism operation, a sort operation, an operation to specify metadata among one or more of the second nodes or an automatic partitioning operation.

7. The computer-implemented method of claim 1, further including:

providing data to generate a graphical editor interface that includes a canvas portion and a catalog portion, with the catalog portion including one or more selectable icons for visually depicting, in the canvas portion, logic of a computation;

receiving icon selection data representing logic of a computation depicted in the canvas portion, with the icon selection data specifying at least one of the one or more selectable icons selected from the catalog portion and included in the canvas portion; and

generating, from the icon selection data received, the first dataflow graph including the plurality of first nodes that represent the logic specified in the canvas portion, with at least one of the first nodes representing the least one of the one or more selectable icons selected from the catalog portion.

8. The computer-implemented method of claim 1, further including:

providing data to generate a graphical editor interface that includes a canvas portion and a catalog portion, with the catalog portion including plural dataset selection icons and plural transformation selection icons;

generating an initial node in the first dataflow graph in accordance with elements stored in a storage unit represented by a selected dataset selection icon and a selected transformation selection icon;

labeling the initial node to provide a labeled node; and

rendering, in the canvas portion, a visual representation of the labeled node.

9. The computer-implemented method of claim 8, further including:

providing data to render the first dataflow graph in the canvas portion of the graphical editor interface.

10. The computer-implemented method of claim 8, wherein upon labeling of all of the initial nodes that were generated, the method further includes:

compiling all labeled nodes of the first dataflow graph into the second dataflow graph that is a computational dataflow graph.

11. The computer-implemented method of claim 8, wherein upon labeling of all of the initial nodes that were modified, the method further includes:

optimizing all labeled nodes of the first dataflow graph, wherein optimizing the labeled nodes of the first dataflow graph further includes optimizing the elements stored in at least one of the labeled nodes.

12. The computer-implemented method of claim 8, further including:

accessing a prototype node; and

applying an algorithm that copies parameters from the accessed prototype node to modify at least one of the initial nodes.

13. The computer-implemented method of claim 8, wherein the labels refer to one or more of keys, values, names, and sources.

14. The computer-implemented method of claim 1, including:

rendering a visual representation of the first dataflow graph, with the visual representation including component icons representing one or more of the first computer operations;

receiving input data,

processing the input data with the second dataflow graph, including:

for a given component icon,

updating the visual representation with a visualization of a result of processing an item of input data with one or more of the second computer operations included in the second dataflow graph based on inclusion of a first computer operation represented by the given component icon.

15. A system for transforming a first dataflow graph into a second dataflow graph, wherein the first dataflow graph includes a plurality of first nodes representing a plurality of first computer operations, wherein the second dataflow graph includes a plurality of second nodes representing a plurality of second computer operations, including:

one or more processors and one or more storage devices storing instructions that are operable, when executed by the one or more processors, to cause the one or more processors to perform the steps of the method of any one of claims 1 to 14.

16. A non-transitory computer readable medium for transforming a first dataflow graph into a second dataflow graph, wherein the first dataflow graph includes a plurality of first nodes representing a plurality of first computer operations, wherein the second dataflow graph includes a plurality of second nodes representing a plurality of second computer operations, the non-transitory computer readable medium storing instructions for causing a computing system to perform the steps of the method of any one of claims 1 to 14.