

Claims

1. A computer-implemented method of generating a computer-executable graph for debugging operations based on a specification that is separate from the computer-executable graph, the method including: receiving a first computer-executable graph that includes components and flows, the components representing operations performed on data records, the flows representing flows of data records between components; receiving a specification that is separate from the first computer-executable graph, the specification defining one or more insertions, each of the insertions being indicative of at least one graph component to be inserted into the first computer-executable graph and identifying a flow of the first computer-executable graph into which the at least one graph component is to be inserted; generating one or more graph components that correspond to the one or more insertions defined by the specification; and generating a second computer-executable graph that includes components and flows that correspond to at least some of the components and flows of the first computer-executable graph and the one or more generated graph components that correspond to the one or more insertions defined by the specification.
2. The method of claim 1, wherein at least one of the generated graph components is a probe that is configured to monitor data related to one or more data records as the one or more data records pass between components of the second computer-executable graph.
3. The method of claim 2, wherein the probe is configured to store the data related to the one or more data records.
4. The method of claim 1, wherein at least one of the generated graph components is a test source that is configured to provide data to one of the components of the second computer-executable graph.
5. The method of claim 1, further including executing the second computer-executable graph.
6. The method of claim 5, wherein fewer than all of the components of the second computer-executable graph are executed.

7. The method of claim 6, wherein the components of the second computer-executable graph that are executed include components whose operations impact one or more of the generated graph components.

8. The method of claim 7, wherein one of the generated graph components is a probe, and the components of the second computer-executable graph that are executed include components that provide data to the probe.

9. The method of claim 8, wherein one or more components of the second computer-executable graph that reside downstream from the probe are not executed.

10. The method of claim 7, wherein one of the generated graph components is a test source, and one or more components of the second computer-executable graph that reside upstream of the test source are not executed.

11. The method of claim 5, further including: modifying a parameter associated with one of the components of the second computer-executable graph; determining whether the modified component, when executed, impacts any of the generated components; and re-executing the second computer-executable graph.

12. The method of claim 11, wherein, if the modified component, when executed, does not impact any of the generated components, the modified component is not re-executed.

13. The method of claim 12, wherein one of the generated components is a probe, and the modified component resides upstream of a flow associated with the probe.

14. The method of claim 1, wherein the first computer-executable graph and the specification are stored in separate files.

15. The method of claim 1, wherein the first computer-executable graph and the second computer-executable graph are stored in separate files.

16. The method of claim 1, wherein the second computer-executable graph is a shell script that is not stored in a file.

17. The method of claim 1, wherein, in the second computer-executable graph, at least one of the generated graph components provides data to a component that corresponds to a component in the first computer-executable graph that receives data from the flow that is associated with the at least one generated graph component.

18. The method of claim 1, wherein, in the second computer-executable graph, at least one of the generated components receives data from a component that corresponds to a component in the first computer-executable graph that provides data to the flow that is associated with the at least one generated graph component.

19. The method of claim 1, wherein the second computer-executable graph is generated without the first computer-executable graph being modified.

20. The method of claim 1, wherein generating the second computer-executable graph includes: generating a copy of at least a portion of the first computer-executable graph; and modifying the copy of the first computer-executable graph to include the one or more generated graph components.

21. A system including: means for receiving a first computer-executable graph that includes components and flows, the components representing operations performed on data records, the flows representing flows of data records between components; means for receiving a specification that is separate from the first computer-executable graph, the specification defining one or more insertions, each of the insertions being indicative of at least one graph component to be inserted into the first computer-executable graph and identifying a flow of the first computer-executable graph into which the at least one graph component is to be inserted; means for generating one or more graph components that correspond to the one or more insertions defined by the specification; and means for generating a second computer-executable graph that includes

components and flows that correspond to at least some of the components and flows of the first computer-executable graph and the one or more generated graph components that correspond to the one or more insertions defined by the specification.

22. A computer system including:

a processor configured to: receive, from a code repository, a first computer-executable graph that includes components and flows, the components representing operations performed on data records, the flows representing flows of data records between components; and receive, through a user interface, a specification that is separate from the first computer-executable graph, the specification defining one or more insertions, each of the insertions being indicative of at least one graph component to be inserted into the first computer-executable graph and identifying a flow of the first computer-executable graph into which the at least one graph component is to be inserted; and a compiler configured to: generate one or more graph components that correspond to the one or more insertions defined by the specification; and generate a second computer-executable graph that includes components and flows that correspond to at least some of the components and flows of the first computer-executable graph and the one or more generated graph components that correspond to the one or more insertions defined by the specification.

23. A non-transitory computer-readable device storing a computer program, the computer program including executable instructions for causing a computer to: receive a first computer-executable graph that includes components and flows, the components representing operations performed on data records, the flows representing flows of data records between components; receive a specification that is separate from the first computer-executable graph, the specification defining one or more insertions, each of the insertions being indicative of at least one graph component to be inserted into the first computer-executable graph and identifying a flow of the first computer-executable graph into which the at least one graph component is to be inserted; generate one or more graph components that correspond to the one or more insertions defined by the specification; and generate a second computer-executable graph that includes components and flows that correspond to at least some of the components and flows of the first computer-executable graph and the one or more generated graph components that correspond to the one or more insertions defined by the specification.

24. A computer-implemented method of generating a computer program for debugging operations based on a specification, the method including: identifying a data processing element of a first version of a computer program, the data processing element configured to i) receive data from or ii) provide data to a first data element; generating a component that represents a

second data element according to a specification defining one or more insertions, at least a first insertion of the one or more insertions identifying the component that represents the second data element and a location in the computer program into which the component that represents the second data element is to be inserted; and generating a second version of at least a portion of the computer program, the second version of the computer program including the identified data processing element and including the generated component, the generated component being generated and arranged within the second version of the computer program according to the defined first insertion such that the identified data processing element in the second version of the computer program i) receives data from or ii) provides data to the generated component.

25. The method of claim 24, including receiving the first version of the computer program.

26. The method of claim 24, wherein generating the second version of at least a portion of the computer program includes generating a copy of the portion of the computer program.

27. The method of claim 26, including modifying the copy of the portion of the computer program to include the generated component.