

What is claimed is :

1. A method for managing sets of parameter values, the method including:
receiving a plurality of sets of parameter values for a generic computer program, and
processing log entries associated with prior executions of instances of the generic
computer program, an operation of each instance of the generic computer program
depending on values of one or more parameters associated with the generic
computer program, each log entry including values of the one or more parameters
for an associated execution of an instance of the generic computer program, the
processing including:
analyzing the generic computer program to classify each of the one or more
parameters associated with the generic computer program as a member of
either a first class of parameters or a second class of parameters;
processing a log entry associated with a prior execution of a first instance of the
generic computer program to form a first set of parameter values, wherein
the processing includes:
including, in the first set, any values of parameters classified as members
of the first class occurring in the log entry, and
excluding, from the first set, any values of parameters classified as
members of the second class occurring in the log entry; and
determining whether to add the first set of parameter values to the plurality of sets
of parameter values based on a comparison of a first identifier for the first
set of parameter values to identifiers for one or more of the sets of
parameter values of the plurality of sets of parameter values.
2. The method of claim 1 wherein processing the log entries includes classifying
parameters based on whether or not the parameters affect data lineage associated with the generic
computer program.

3. The method of claim 1 wherein the comparison of the first identifier for the first set of parameter values to the identifiers for one or more of the sets of parameter values of the plurality of sets of parameter values includes:

determining the first identifier based on the first set of parameter values and an identifier of the generic computer program;

determining a plurality of second identifiers, one for each set of parameter values of the one or more of the sets of parameter values; and

comparing the first identifier to each second identifier of the plurality of second identifiers to determine whether the first identifier and any of the second identifiers match.

4. The method of claim 3 wherein determining whether to add the first set of parameter values to the plurality of sets of parameter values includes determining to add the first set of parameter values to the plurality of sets of parameter values if none of the second identifiers match the first identifier.

5. The method of claim 3 wherein determining the first identifier includes computing an identification string from the contents of the first set of parameter values, and determining the second identifiers includes computing identification strings from the contents of the one or more of the sets of parameter values of the plurality of sets of parameter values.

6. The method of claim 3 wherein determining the first identifier includes forming a concatenation of one or more of the identifier of the generic computer program, name-value pairs of the first set of parameter values, a function prototype of the generic computer program, and a project scope for the first instance of the generic computer program.

7. The method of claim 3 wherein determining the first identifier includes applying a data mapping function to one or more of the identifier of the generic computer program, name-value pairs of the first set of parameter values, a function prototype of the generic computer program, and a project scope for the first instance of the generic computer program.

8. The method of claim 7 wherein the data mapping function includes a hash function.
9. The method of claim 1 wherein the first class of parameters includes parameters that affect the logical operation of the generic computer program and the second class of parameters includes parameters that do not affect the logical operation of the generic computer program.
10. The method of claim 1 wherein the generic computer program is specified as a dataflow graph that includes nodes representing data processing operations and links between the nodes representing flows of data elements between the data processing operations.
11. The method of claim 1 wherein, for each of the parameters, the analysis includes either automatically classifying the parameter or accepting a user-defined classification for the parameter.
12. The method of claim 11 wherein automatically classifying the parameter includes initially classifying the parameter as belonging to the first class of parameters, determining a number of unique values of the parameter in a plurality of executions of instances of the generic computer program, and re-classifying the parameter as belonging to the second class of parameters if the number of unique values of the parameter exceeds a predetermined threshold.
13. The method of claim 11 wherein automatically classifying the parameter includes initially classifying the parameter as belonging to the first class of parameters, determining whether or not changes in values of the parameter in a plurality of executions of instances of the generic computer program affect data lineage associated with the generic computer program, and re-classifying the parameter as belonging to the second class of parameters if changes in values of the parameter do not affect the data lineage.
14. The method of claim 1 further including forming an association between the log entry associated with the execution of the first instance of the generic computer program and the first set of parameter values.

15. The method of claim 1 wherein the log entry associated with the execution of the first instance of the generic computer program includes a log entry of an execution command used to instantiate the generic computer program including one or more parameter values supplied as arguments to the execution command.

16. The method of claim 15 wherein the log entry associated with the execution of the first instance of the generic computer program further includes one or more of an indication of the project that the first instance executed in, an indication of internal parameters for the first instance, and an indication of environmental settings, global variables, and configuration variables used by the first instance.

17. The method of claim 11 further including processing an overall plurality of sets of parameter values for a plurality of generic computer programs and an overall plurality of log entries associated with executions of instances of one or more of the plurality of generic computer programs to form a data lineage report, wherein the overall plurality of sets of parameter values includes the augmented plurality of sets of parameter values for the generic computer program, and the overall plurality of log entries associated with executions of instances of the one or more of the plurality of generic computer programs includes the log entry of the execution of the first instance of the generic computer program, including its association with the first set of parameter values.

18. The method of claim 17, wherein forming the data lineage report includes, for each set of parameter values of the overall plurality of sets of parameter values for the plurality of generic computer programs,

processing the overall plurality of log entries associated with executions of instances of the one or more of the plurality of generic computer programs to identify all log entries associated with executions of instances of the generic computer program corresponding to the set of parameter values, and identifying a most recent time of instantiation of the generic computer program from the identified log entries associated with executions of instances of the generic computer program; and

determining whether to include the set of parameters in the data lineage report based on the most recent time of instantiation of the generic computer program.

19. The method of claim 18 wherein determining whether to include the set of parameters in the data lineage report based on the most recent time of instantiation of the generic computer program includes comparing the most recent time of instantiation to a predetermined time interval and including the set of parameters in the data lineage report if the most recent time of instantiation of the generic computer program is within the predetermined time interval.

20. The method of claim 17 wherein forming the data lineage report includes, for each set of parameter values of the overall plurality of sets of parameter values for the plurality of generic computer programs,

processing the overall plurality of log entries associated with executions of instances of the one or more of the plurality of generic computer programs to determine a number of log entries associated with executions of instances of the generic computer program corresponding to the set of parameter values, and

determining whether to include the set of parameters in the data lineage report based on the number of log entries associated with executions of instances of the generic computer program.

21. Software stored in a non-transitory form on a computer-readable medium, for managing sets of parameter values, the software including instructions for causing a computing system to perform the method of any of claims 1 to 20.

22. A computing system for managing sets of parameter values, the computing system including:

an input device or port for receiving a plurality of sets of parameter values for a generic computer program, and

at least one processor configured to perform the method of any of claims 1 to 20.

23. A computing system for managing sets of parameter values, the computing system including:

means for receiving a plurality of sets of parameter values for a generic computer program, and

means for processing log entries associated with prior executions of instances of the generic computer program, an operation of each instance of the generic computer program depending on values of one or more parameters associated with the generic computer program, each log entry including values of the one or more parameters for an associated execution of an instance of the generic computer program, the processing including:

analyzing the generic computer program to classify each of the one or more parameters associated with the generic computer program as a member of either a first class of parameters or a second class of parameters;

processing a log entry associated with a prior execution of a first instance of the generic computer program to form a first set of parameter values, wherein the processing includes:

including, in the first set, any values of parameters classified as members of the first class occurring in the log entry, and

excluding, from the first set, any values of parameters classified as members of the second class occurring in the log entry; and

determining whether to add the first set of parameter values to the plurality of sets of parameter values based on a comparison of a first identifier for the first set of parameter values to identifiers for one or more of the sets of parameter values of the plurality of sets of parameter values.

24. A method for managing sets of parameter values, the method including:

receiving a generic computer program;

receiving a first set of parameter values;

generating an executable instance of the generic computer program by
instantiating the generic computer program according to the first set of
parameter values, an operation of the executable instance of the generic
computer program depending on the first set of parameter values;

receiving data from one or more data sets;

executing the executable instance of the generic computer program to process one
or more of the received data;

generating a log entry for the executable instance of the generic computer
program, the log entry including one or more of the parameter values of
the first set of parameter values;

storing the log entry;

receiving the log entry;

processing the log entry to form a first set of parameter values, wherein the
processing includes extracting the one or more of the parameter values of
the first set of parameter values from the log entry and forming the first set
of parameter values from the extracted parameter values; and

determining whether to add the first set of parameter values to a plurality of
preexisting sets of parameter values based on a comparison of a first
identifier for the first set of parameter values to identifiers for one or more
of the preexisting sets of parameter values of the plurality of preexisting
sets of parameter values.

25. The method of claim 24 wherein the comparison of the identifier for the first set of parameter values to the identifiers for one or more of the preexisting sets of parameter values of the plurality of preexisting sets of parameter values includes:

determining the first identifier based on the first set of parameter values and an identifier of the generic computer program;

determining a plurality of second identifiers, one for each preexisting set of parameter values of the one or more of the preexisting sets of parameter values; and
comparing the first identifier to each second identifier of the plurality of second identifiers to determine whether the first identifier and any of the second identifiers match.

26. The method of claim 25 wherein determining whether to add the first set of parameter values to the plurality of preexisting sets of parameter values includes determining to add the first set of parameter values to the plurality of preexisting sets of parameter values if none of the second identifiers match the first identifier.

27. The method of claim 25 wherein determining the first identifier includes computing an identification string from the contents of the first set of parameter values, and determining the second identifiers includes computing identification strings from the contents of the one or more of the preexisting sets of parameter values of the plurality of preexisting sets of parameter values.

28. The method of claim 25 wherein determining the first identifier includes forming a concatenation of one or more of the identifier of the generic computer program, name-value pairs of the first set of parameter values, a function prototype of the generic computer program, and a project scope for the executable instance of the generic computer program.

29. The method of claim 25 wherein determining the first identifier includes applying a data mapping function to one or more of the identifier of the generic computer program, name-value pairs of the first set of parameter values, a function prototype of the generic computer program, and a project scope for the executable instance of the generic computer program.

30. The method of claim 29 wherein the data mapping function includes a hash function.

31. The method of claim 24 further including analyzing the generic computer program to classify each of the one or more parameters associated with the generic computer program as a member of either a first class of parameters or a second class of parameters.

32. The method of claim 31 wherein processing the log entry to form a first set of parameter values further includes:

including, in the particular set, any extracted parameter values classified as members of the first class occurring in the log entry , and

excluding, from the particular set, any extracted parameter values classified as members of the second class occurring in the log entry.

33. The method of claim 31 wherein the first class of parameters includes parameters that affect the logical operation of the generic computer program and the second class of parameters includes parameters that do not affect the logical operation of the generic computer program.

34. Software stored in a non-transitory form on a computer-readable medium, for managing sets of parameter values, the software including instructions for causing a computing system to perform the method of any of claims 24 to 33.

35. A system for managing sets of parameter values, the system including:

a first computing system including

a first input device or port for receiving a generic computer program, a first set of parameter values, and data from one or more data sets;

a first set of one or more processors configured to:

generate an executable instance of the generic computer program
by instantiating the generic computer program according to
the first set of parameter values, an operation of the
executable instance of the generic computer program
depending on the first set of parameter values;

execute the executable instance of the generic computer program
to process one or more of the received data;

generate a log entry for the executable instance of the generic
computer program, the log entry including one or more of
the parameter values of the first set of parameter values;

a first output device or port for storing the log entry in a storage device;

a second computing system including

a second input device or port for receiving the log entry;

a second set of one or more processors configured to:

process the log entry to form a first set of parameter values,
wherein the processing includes extracting the one or more
of the parameter values of the first set of parameter values
from the log entry and forming the first set of parameter
values from the extracted parameter values; and

determine whether to add the first set of parameter values to a
plurality of preexisting sets of parameter values based on a
comparison of a first identifier for the first set of parameter
values to identifiers for one or more of the preexisting sets
of parameter values of the plurality of preexisting sets of
parameter values.