

What is claimed is :

CLAIMS

1. A trace replay execution control system, comprising:
 - a processor;
 - a memory in operable communication with the processor;
 - a trace recorded during an execution of a program in conjunction with calls into a runtime of the program
 - and a kernel, wherein the trace includes trace entries which include a record of
 - activities of ~~by~~ native code instructions which executed in at least one thread of
 - the program, wherein the activities of the native code instructions are traced as the program is executed, and wherein the trace does not include any executing instance of the runtime of the program ;
 - a source code of at least a portion of the program;
 - a source-intermediate mapping which automatically maps between the source code of at least the portion of the program and an
 - intermediate representation of the source code of at least the portion of the program;
 - an intermediate-native mapping which automatically maps between the intermediate
 - representation of the source code of at least the portion of the program and the trace; and
 - an execution controller in a software tool, wherein the execution controller is configured to,
 - upon execution ~~by with~~ the processor, control a replay execution of the trace in
 - response to receiving trace replay execution control requests, in which the source code is correlated with the trace
 - entries through the source-intermediate mapping and the intermediate-native mapping.
2. The trace replay execution control system of claim 1, wherein the execution controller is further configured to translate requests for at least the following operations into corresponding performance of trace native code instructions :
 - a forward execution step-in operation;
 - a forward execution step-out operation; and
 - a forward execution step-over operation.

3.2. The trace replay execution control system of claim 1, wherein the execution controller is further configured to translate requests for at least the following operations into corresponding performance of trace native code instructions:

- a backward execution step-in operation,
- a backward execution step-out operation, and
- a backward execution step-over operation.

4. The trace replay execution control system of claim 1, wherein the execution controller is further configured to translate requests for at least the following operations into corresponding performance of trace native code instructions :

continue execution forward through the trace until a location in a breakpoint range is reached; and

continue execution backward through the trace until a location in a breakpoint range is reached.

5.3. The trace replay execution control system of claim 1, wherein the trace replay execution control system is configured to step into a part of the source code of at least the portion of the program which during the replay execution of the trace has not yet been compiled into native code instructions.

6.4. The trace replay execution control system of claim 1, wherein the intermediate representation of the source code of at least the portion of the program comprises at least one of the following:

a symbol table which correlates at least two identifiers used in the source code of at least the portion of the program with data types;

an abstract —syntax tree derived from at least a portion of the source code of at least the portion of the program; or

a translation into an intermediate language of at least a portion of the source code of at least the portion of the program.

7.5. The trace replay execution control system of claim 1, wherein the trace does not contain any executing instance of the runtime of the program, in that the trace replay execution control system is not configured

to call into the trace to execute any code that is specific to the runtime of the program .

8.6. The trace replay execution control system of claim 1, further comprising a live process of the program with a callable instance of the runtime of the program, wherein the trace is recorded from the live process of the program and the trace replay execution control system is configured to replay execution of the — trace recorded from the live process of the program.

9.7. The trace replay execution control system of claim 1, wherein the trace replay execution control system is configured to skip over replay execution of a portion of the trace,

in that the execution controller moves the replay execution of the portion of the trace from a first execution time point of the trace to a second execution time point of the trace without the trace replay execution control system performing trace native code instructions that would have been performed by continued execution from the first execution time point of the trace location to the second execution time point of the trace location.

10.8. A trace replay execution control method, comprising:

identifying a traced managed program, ~~namely, a program~~ which is ~~was~~ configured to execute in conjunction with calls into a runtime of the traced managed program and a kernel and configured to be traced during the execution;

receiving a trace replay execution control request, wherein the trace replay execution control request is received in a software tool running on a computing system;

based on receiving the trace replay execution control request, automatically mapping between a source code of the traced managed program and an intermediate representation of the source code of the traced managed program, and automatically mapping between the intermediate representation of the source code of the traced managed program and a trace recorded during the execution of the traced managed program, wherein the trace includes trace entries which include a record of activities of native code instructions which executed in at least one thread of the traced managed program by at least one processor, wherein the activities of the native code instructions are traced as the traced managed program is executed, and wherein the trace does not include any executing instance of the runtime of the traced managed program; and

performing trace native code instructions which correspond to the trace replay execution control request, in which the source code is correlated with the trace entries through the source-intermediate mapping and the intermediate-native mapping, thereby replaying execution of a portion of the trace.

11.9. The trace replay execution control method of claim 108, wherein the trace replay execution control request identifies a backward execution step-out operation to step out of a routine, and the trace replay execution control method further comprises setting a breakpoint on a trace native code instruction which has a location that shortly precedes a location of a return address of a call to the routine, and wherein a location A shortly precedes a location B in a trace when A is encountered prior to B during a forward execution of the trace and there are at most five trace native code instructions between A and B.

12. 10. ~~The trace replay execution control method of claim~~ 108, wherein the source

code of the traced managed program contains statements and expressions, ~~and~~ wherein the trace replay execution control method further comprises setting a breakpoint on a trace native code instruction which is on an edge of a group of all trace native code instructions that are mapped by the trace replay execution control method to a given statement or a given expression in the source code of the traced managed program, ~~and~~ wherein an instruction X of a group of instructions G in a trace is on an edge of G when X satisfies at least one of the following conditions:

_____ X is the first instruction of G encountered during forward execution of the trace;₅

_____ X is the last instruction of G encountered during forward execution of the trace;₅

_____ X is the first — instruction of G encountered during backward execution of the trace;₅
or

_____ X is the last instruction of G encountered during backward execution of the trace.

13. The trace replay execution control method of claim 12, wherein the trace display execution control request calls for one of the following :

_____ a forward execution step-over operation to step forward over a routine;

_____ a backward execution step-over operation to step backward over a routine;

_____ a forward execution step-in operation to step forward into a routine; or

_____ a backward execution step-in operation to step backward into a routine.

~~14.11.~~ The trace replay execution control method of claim 108, wherein the trace replay execution control request identifies a forward execution step-in operation to step into a user code routine, wherein replay execution of the user code routine includes replay execution of at least one call to a runtime routine, wherein the at least one call to the runtime routine does not have a corresponding call statement or call expression in the source code of the traced managed program, and wherein the trace replay execution control method further comprises at least one of the following:

setting a breakpoint range which specifies an entire memory range that is designated to receive dynamically compiled code; or
setting a breakpoint to perform a step-over operation on the at least one -call to the runtime routine.

15.12. The trace replay execution control method of claim 108, wherein the trace replay execution control request calls for at least one of the following operations:

continue trace replay execution forward through the trace until a breakpoint is reached;

continue trace replay execution forward through the trace until a location in a breakpoint range is reached;

continue trace replay execution backward through the trace until a breakpoint is reached; ~~or~~

continue trace replay execution backward through the trace until a location in a breakpoint range is reached;

a backward execution step-in operation;

a backward execution step-out operation;

a backward execution step-over operation;

a forward execution step-in operation;

a forward execution step-out operation; or

a forward execution step-over operation.

~~16.13.~~ The trace replay execution control method of claim 108, wherein during the execution of the traced managed program, a maximum number of breakpoints was constrained by a hardware breakpoint constraint that a processor can monitor at most N addresses at once, and wherein the trace replay execution control method further comprises setting more than N breakpoints during the replay execution of the portion of the trace, thereby exceeding the hardware breakpoint constraint.

~~17.14.~~ A computer-readable storage medium stored configured with data and instructions to perform, upon execution by with a processor in operable communication with a memory, a trace replay execution control method, the trace replay execution control method comprising:

identifying a traced managed program which is configured to execute in conjunction with calls into a runtime of the traced managed program and a kernel and configured to be traced during the execution;

receiving a trace replay execution control request, wherein the trace replay execution control request is received in a software tool running on a computing system;

based on receiving the trace replay execution control request, automatically mapping between a source

code of the a-traced managed program and an intermediate representation of

the source code of the traced managed program, and automatically mapping between the intermediate representation of the source code of the traced managed program and

a trace recorded during the execution of the traced managed program, wherein the trace includes trace entries which include a record of activities of native code instructions which executed in at least one thread of the traced managed program by at least one processor, wherein the activities of the native code instructions are

~~which were~~ traced as the traced managed program executed, and wherein the trace does not include any executing instance of the runtime of the traced managed program; and

performing trace native code instructions which correspond to the trace replay execution control request or setting a breakpoint on a trace native code instruction, in which the source code is correlated with the trace entries through the source-intermediate mapping and the intermediate-native mapping, thereby controlling replay execution of the trace.

18. The computer-readable storage medium of claim 17, wherein the traced managed program, in conjunction with a debugger code, includes a live process and an executing runtime of the traced managed program, and wherein the debugger code communicates with the executing runtime of the traced managed program, and the trace replay execution control method further comprises controlling backward execution replay using the trace without terminating the traced managed program.

19. The computer-readable storage medium of claim 17, wherein the trace replay execution control method further comprises at least two of the following execution control activities:

controlling an execution of the runtime of the traced managed program at a high level using machine-level information in the trace, where controlling the execution of the runtime of the traced managed program at a high level means controlling based on a user selection that specifies an identifier or a statement or an expression in the source code of the traced managed program, and machine-level information in the trace means information which is specific to a processor architecture;

debugging both forward execution and backward execution of a process of the traced managed program;

stepping into a call shown in the source code of the traced managed program at least in part by setting a large breakpoint range, where a breakpoint range is large when it covers at least a thousand contiguous addresses or instructions in the trace;

mapping to a particular version of native code of a routine when the trace includes at least two versions of native code of the routine; or

recording trace data from a live process of the traced managed program while debugging the live process, and then performing reverse execution debugging on the live process using at least some of the recorded trace data.

20.15. The computer-readable storage medium of claim 1714, wherein the trace replay execution control method further comprises at least three of the following execution control activities:

controlling an execution of the runtime of the traced managed program at a high level using machine-level

information in the trace, where controlling the execution of the runtime of the traced managed program at a high level means controlling

based on a user selection that specifies an identifier or a statement or an expression in the source code of the traced managed program, and machine-level information in the trace means

information which is specific to a processor architecture;

debugging both forward execution and backward execution of a process of the traced managed program;

stepping into a call shown in the source code of the traced managed program at least in part by setting a large

breakpoint range, where a breakpoint range is large when it covers at least a thousand contiguous addresses or instructions in the trace;

mapping to a particular version of native code of a routine when the traced includes at least two versions of native code of the routine; or

recording trace data from a live process of the traced managed program while debugging the live process, and then performing reverse execution debugging on the live process using at least some of the recorded trace data.