

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
12 January 2012 (12.01.2012)

PCT

(10) International Publication Number
WO 2012/004707 - A 4(51) International Patent Classification:
G06F 9/44 (2006.01)(21) International Application Number:
PCT/IB20 11/052878(22) International Filing Date:
30 June 2011 (30.06.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/830,451 6 July 2010 (06.07.2010) US

(72) Inventor; and

(71) Applicant : ELDAR, Adi [IL/IL]; 68b Shtern St., 55602
Kiryat Ono (IL).(74) Agent: DR. MARK FRIEDMAN LTD.; Moshe Aviv
Tower, 54th Floor 7, Jabotinsky St., 52520 Ramat Gan
(IL).(81) Designated States (unless otherwise indicated, for every
kind of national protection available): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO,DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD,
SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.(84) Designated States (unless otherwise indicated, for every
kind of regional protection available): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG,
ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ,
TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

— with amended claims (Art. 19(1))

Date of publication of the amended claims: 22 March 2012

(54) Title: EFFICIENT RECORDING AND REPLAYING OF THE EXECUTION PATH OF A COMPUTER PROGRAM

Code at address 30145B before
instrumentation:

7D 0C 8B 45 EC

JGE... MOV...

310

and after instrumentation:

CC 0C 8B 45 EC

INT 3 0C MOV...

320

Figure 3

(57) Abstract: To monitor the execution path of executable code, only non-deterministic jump instructions of the executable code are instrumented by replacing them with respective recording instructions that record the results of executions of the non-deterministic jump instructions and then emulate those executions, thereby providing instrumented code, and the instrumented code is executed. Preferably, the recording instructions are one byte long and invoke an interrupt service routine that does the recording and the emulating. Optionally, selected instructions of the executable code are replaced with trigger instructions for turning the recording on and off. Preferably, after the instrumented code is executed, the addresses of the instrumented instructions and the results of their executions are played back either forward or backward. Optionally, the instrumented code is executed a second time and the results of the executions of the instrumented instructions in the two executions of the instrumented code are compared.



Wo 2°12/©°47°7 A

1. A method of monitoring an execution path of executable code that includes at least one non-deterministic jump instruction, comprising the steps of:
 - (a) storing the executable code in a first machine-readable medium;
 - (b) only for each non-deterministic jump instruction of at least a portion of the at least one non-deterministic jump instruction:
 - (i) identifying an address of said each non-deterministic jump instruction, and
 - (ii) replacing only said each non-deterministic jump instruction with a respective recording instruction for
 - (A) recording a result of each execution of said each non-deterministic jump instruction in a second machine-readable medium, and
 - (B) emulating said each execution of said each non-deterministic jump instruction,thereby providing instrumented executable code; and
 - (c) effecting a first execution of the instrumented executable code.
2. The method of claim 1, wherein said identifying and replacing is effected off-line.
3. The method of claim 1, wherein said identifying and replacing is effected on-line.
4. The method of claim 1, wherein said address of every said non-deterministic jump instruction is identified and wherein every said non-deterministic jump instruction is replaced by said respective recording instruction thereof.
5. The method of claim 1, wherein each said recording instruction is one byte long.
6. The method of claim 1, wherein said identifying includes reading an assembly file of the executable code.

7. The method of claim 1, wherein said identifying includes disassembling the executable code.

8. The method of claim 7, wherein said disassembling is effected on-line.

9. The method of claim 1, wherein said first and second machine-readable media are identical.

10. The method of claim 1, wherein said first and second machine-readable media are different.

11. The method of claim 1, wherein said recording of said result and said emulating of said each execution of said respective non-deterministic jump instruction is effected by an interrupt service routine that is invoked by said recording instructions.

12. The method of claim 11, further comprising the step of:

(d) installing said interrupt service routine.

13. The method of claim 12, wherein said installing is effected using a device driver.

14. The method of claim 1, further comprising the step of:

(d) prior to said effecting of said first execution, setting up, in said second machine-readable medium, a recording structure that includes, for each said non-deterministic jump instruction of said at least portion of the at least one non-deterministic jump instruction:

(i) an address of said each non-deterministic jump instruction,

(ii) a pointer to a buffer in said second machine-readable medium for recording said result of said each execution of said each non-deterministic jump instruction, and

(iii) a pointer to a code stub for emulating said each execution of said each non-deterministic jump instruction;

AMENDED SHEET (ARTICLE 19)

said recording and said emulating then being effected with reference to said recording structure.

15. The method of claim 1, further comprising the step of:
- (d) replacing at least one other selected instruction of the executable code with a trigger instruction.

16. The method of claim 15, wherein said selected instruction is selected from the group consisting of an entry point of a function of the executable code and an exit point of a function of the executable code.

17. The method of claim 1, further comprising the step of:
- (d) recording, in a third machine-readable medium, said address of each said non-deterministic jump instruction of said at least portion of the at least one non-deterministic jump instruction, along with all results of any said executions of said each non-deterministic jump instruction, thereby providing a first record of the execution path.

18. The method of claim 17, wherein said second and third machine-readable media are identical.

19. The method of claim 17, wherein said second and third machine-readable media are different.

20. The method of claim 17, further comprising the step of:
- (e) playing said record of the execution path forward.

21. The method of claim 17, further comprising the step of:
- (f) playing said record of the execution path backward.

22. The method of claim 21, further comprising the step of:
- (g) during said forward playing of the execution path, constructing a backward recording array and a backward replay state vector, said backward playing of said record of the execution path being in

accordance with said backward recording array and said backward replay state vector.

23. The method of claim 17, further comprising the step of:
 - (e) subsequent to said providing of said first record of the execution path, effecting a second execution of the instrumented executable code.
24. The method of claim 23, further comprising the steps of:
 - (f) recording, in a fourth machine-readable medium, said address of each said non-deterministic jump instruction of said at least portion of the at least one non-deterministic jump instruction, along with all results of any said executions, during said second execution, of said each non-deterministic jump instruction, thereby providing a second record of the execution path; and
 - (g) comparing said first and second records of the execution path.
25. The method of claim 24, wherein said third and fourth machine-readable media are identical.
26. The method of claim 24, wherein said third and fourth machine-readable media are different.
27. The method of claim 24, further comprising the step of:
 - (h) inserting, in the executable code, a synchronization instruction for recording, in said first and second records of the execution path, every arrival of said first and second executions at a location in the executable code where said synchronization instruction has been inserted.
28. The method of claim 23, further comprising the step of:
 - (f) during said second execution, comparing said result of each said execution of each said non-deterministic jump instruction with said first record of the execution path.

AMENDED SHEET (ARTICLE 19)

29. The method of claim 28, wherein said second execution is stopped when said comparing shows that said result of one said execution of one said non-deterministic instruction during said second execution differs from said result of said one execution of said one non-deterministic instruction during said first execution.

30. The method of claim 28, further comprising the steps of:

- (h) inserting, in the executable code, a synchronization instruction for:
 - (i) recording, in said first record of the execution path, every arrival of said first execution at a location in the executable code where said synchronization instruction has been inserted; and
 - (ii) counting every arrival of said second at said location in the executable code where said synchronization instruction has been inserted;

and wherein said second execution is stopped when said comparing shows that, starting from an execution of said synchronization instruction a pre-determined number of times by both said first execution and said second execution, said result of one said execution of one said non-deterministic instruction during said second execution differs from said result of said one execution of said one non-deterministic instruction during said first execution.

31. A computer-readable storage medium having computer-readable code embodied on said computer-readable storage medium, the computer-readable code for monitoring an execution path of executable code that includes at least one non-deterministic jump instruction, the computer-readable code comprising:

- (a) program code for: only for each non-deterministic jump instruction of at least a portion of the at least one non-deterministic jump instruction:
 - (i) identifying an address of said each non-deterministic jump instruction; and
 - (ii) replacing only said each non-deterministic jump instruction with a respective recording instruction for:
 - (A) recording a result of each execution of said each non-deterministic jump instruction, and

AMENDED SHEET (ARTICLE 19)

- (B)) emulating said each execution of said each non-deterministic jump instruction.

32. The computer-readable storage medium of claim 31, wherein said recording instruction is one byte long.

33. The computer-readable storage medium of claim 31, wherein the computer-readable code further comprises

- (b) program code for an interrupt service routine to which said recording instructions jump for said recording and said emulating.

34. The computer-readable storage medium of claim 33, wherein the computer-readable code further comprises:

- (c) program code for a device driver for installing said interrupt service routine.

35. The computer-readable storage medium of claim 31, wherein the computer-readable code further comprises:

- (b) program code for setting up a recording structure that includes, for each said non-deterministic jump instruction of said at least portion of the at least one non-deterministic jump instruction:
- (i) an address of said each non-deterministic jump instruction,
 - (ii) a pointer to a buffer for recording said result of said each execution of said each non-deterministic jump instruction, and
 - (iii) a pointer to a code stub for emulating said each execution of said each non-deterministic jump instruction; and
- (c) program code for said code stub.

36. The computer-readable storage medium of claim 31, wherein the computer-readable code further comprises:

- (b) program code for replacing at least one other selected instruction of the executable code with a trigger instruction.

AMENDED SHEET (ARTICLE 19)

37. The computer-readable storage medium of claim 31, wherein said replacing provides instrumented executable code, and wherein the computer-readable code further comprises:

- (b) program code for, subsequent to execution of said instrumented executable code, recording said address of each said non-deterministic jump instruction of said at least portion of the at least one non-deterministic jump instruction, along with all results of any said executions of said each non-deterministic jump instruction, thereby providing a record of the execution path.

38. The computer-readable storage medium of claim 37, wherein the computer-readable code further comprises:

- (c) program code for playing said record of the execution path forward.

39. The computer-readable storage medium of claim 38, wherein the computer-readable code further comprises:

- (d) program code for playing said record of the execution path backward.

40. The computer-readable storage medium of claim 38, wherein the computer-readable code further comprises:

- (e) program code for constructing a backward recording array and a backward replay state vector during said forward playing of the execution path.

41. The computer-readable storage medium of claim 37, wherein the computer-readable code further comprises:

- (c) program code for comparing two said records of respective execution paths of two executions of said instrumented code.

42. The computer-readable storage medium of claim 31, wherein the computer-readable code further comprises:

- (b) program code for replacing at least one other selected instruction of the executable code with a synchronization instruction.

AMENDED SHEET (ARTICLE 19)