WHAT IS CLAIMED IS:

1. An apparatus (200) for executing a software program (101), the apparatus comprising a plurality of processing units (210, 220) and at least one hardware processor (201) adapted for:

in an intermediate representation (102) of the software program, where the intermediate representation comprises a plurality of blocks, each associated with one of a plurality of execution blocks of the software program and comprising a set of intermediate instructions, identifying (301) a calling block (111) and a target block (121) , where the calling block comprises at least one control-flow intermediate instruction to execute at least one target intermediate instruction of the target block;

generating (311) a target set of executable instructions (122) using the target block;

generating (313) a calling set of executable instructions (112) using the calling block and using at least one computer control instruction for invoking the target set of executable instructions, when the calling set of executable instructions is executed by a calling processing unit (210) of the plurality of processing units and the target set of executable instructions is executed by a target processing unit (220) of the plurality of processing units;

configuring (321) the calling processing unit for executing the calling set of executable instructions; and

configuring (323) the target processing unit for executing the target set of executable instructions;

wherein the generating of the calling set of executable instructions and the target set of executable instructions is performed during runtime of the software program.

2. The apparatus of claim 1, wherein the target processing unit is further configured for providing (410) the calling processing unit with at least one value produced by the target processing unit executing (402) at least some of the target set of instructions.

3. The apparatus of claim 2, wherein the calling processing unit is further configured for receiving (413) the at least one value, produced by the target processing unit executing at least some of the target set of instructions, from the calling processing unit.

4. The apparatus of claim 1, wherein the target processing unit is further configured for instructing, after executing at least some other of the target set of instructions, the calling processing unit to execute (422) at least some of the calling set of executable instructions.

5. The apparatus of claim 4, wherein the calling processing unit is further configured for:

suspending execution of the calling set of executable instructions after executing the at least one computer control instruction for invoking the target set of executable instructions; and

executing the at least some of the calling set of executable instructions in response to receiving an instruction from the target processing unit.

6. The apparatus of claim 1, wherein the target processing unit is selected from the group of processing units consisting of: a central processing unit, a multi-core central processing unit (CPU), a data processing unit (DPU), a microcontroller unit (MCU), an accelerated processing unit (ACU), a field-programmable gate array (FPGA), a coarse-grained reconfigurable architecture (CGRA), a neural-network accelerator, an intelligence processing unit (IPU), an application-specific integrated circuit (ASIC), a quantum computer, and an interconnected computing grid, comprising a plurality of reconfigurable logical elements connected by a plurality of configurable data routing junctions.

7. The apparatus of claim 1, wherein a first computer architecture of the calling processing unit is different from a second computer architecture of the target processing unit.

8. The apparatus of claim 7, wherein at least one of the first computer architecture and the second computer architecture are selected from a group of computer architectures consisting of: a complex instruction set computer (CISC) architecture, a reduced instruction set computer (RISC) architecture, an x86-family based computer architecture, an International Business Machine (IBM) Power-family based computer architecture, and an ARM-family based computer architecture.

9. The apparatus of claim 7, wherein the calling set of executable instructions comprises a plurality of conversion instructions, implementing an application binary interface applied to the at least one control-flow intermediate instruction to convert between the first computer architecture of the calling processing unit and the second computer architecture of the target processing unit.

10. The apparatus of claim 7, wherein the target block comprises a plurality of target metadata values; and

wherein generating the target set of executable instructions comprises generating a plurality of converted target metadata values using the plurality of target metadata values according to the second computer architecture of the target processing unit.

11. The apparatus of claim 1, wherein the calling set of executable instructions comprises at least one first memory access instruction to at least one memory address in an identified range of application memory addresses; and

wherein the target set of executable instructions comprises at least one second memory access instruction to at least one other memory address in the identified range of application memory addresses.

12. The apparatus of claim 11, wherein the identified range of application memory addresses is one of: a range of physical memory addresses, a range of virtual memory addresses, a range of memory-mapped input-output addresses, and a range of bus addresses.

13. The apparatus of claim 1, wherein the calling set of executable instructions comprises accessing at least one device using at least one device handle associated with the at least one device; and

wherein the target set of executable instructions comprises accessing the at least one device using the at least one device handle.

14. The apparatus of claim 13, wherein the at least one device handle is at least one of: a network socket handle, an operating system kernel object handle, and a file descriptor.

15. The apparatus of claim 1, wherein the at least one hardware processor is further adapted for:

collecting (330) a plurality of statistical values while executing at least part of the software program; and

identifying (301) the calling block and the target block according to the plurality of statistical values.

16. The apparatus of claim 15, wherein at least one of the plurality of statistical values indicative of one of: an outcome of executing a control-flow instruction, a data value of a variable of the software program, a memory access pattern, and a memory virtual address translation.

17. The apparatus of claim 15, wherein the at least one hardware processor is further adapted for:

inserting a plurality of telemetry intermediate instructions into the intermediate representation of the software program, for the purpose of collecting the plurality of statistical values.

18. The apparatus of claim 1, wherein the calling processing unit is the at least one hardware processor.

19. A method (300) for executing a software program (101), comprising:

in an intermediate representation (102) of the software program, where the intermediate representation comprises a plurality of blocks, each associated with one of a plurality of execution blocks of the software program and comprising a set of intermediate instructions, identifying (301) a calling block (111) and a target block (121), where the calling block comprises at least one control-flow intermediate instruction to execute at least one target intermediate instruction of the target block;

generating (311) a target set of executable instructions (122) using the target block;

generating (313) a calling set of executable instructions (112) using the calling block and using at least one computer control instruction for invoking the target set of executable instructions, when the calling set of executable instructions is executed by a calling processing unit (210) of a plurality of processing units and the target set of executable instructions is executed by a target processing unit (220) of the plurality of processing units;

configuring (321) the calling processing unit for executing the calling set of executable instructions; and

configuring (323) the target processing unit for executing the target set of executable instructions;

wherein the generating of the calling set of executable instructions and the target set of executable instructions is performed during runtime of the software program.

20. A software program product for executing a software program, comprising:

a non-transitory computer readable storage medium;

first program instructions for: in an intermediate representation of the software program, where the intermediate representation comprises a plurality of blocks, each associated with one of a plurality of execution blocks of the software program and comprising a set of intermediate instructions, identifying a calling block and a target block, where the calling block comprises at

least one control-flow intermediate instruction to execute at least one target intermediate instruction of the target block;

second program instructions for generating a target set of executable instructions using the target block;

third program instructions for generating a calling set of executable instructions using the calling block and using at least one computer control instruction for invoking the target set of executable instructions, when the calling set of executable instructions is executed by a calling processing unit of a plurality of processing units and the target set of executable instructions is executed by a target processing unit of the plurality of processing units;

fourth program instructions for configuring the calling processing unit for executing the calling set of executable instructions; and

fifth program instructions for configuring the target processing unit for executing the target set of executable instructions; wherein the generating of the calling set of executable instructions and the target set of executable instructions is performed during runtime of the software program;

wherein the first, second, third, fourth and fifth program instructions are executed by at least one computerized processor from the non-transitory computer readable storage medium.