

CLAIMS

What is claimed is:

1. A method for runtime optimization of a configurable processing architecture, comprising:
 - receiving a plurality of calls for running at least one function;
 - identifying at least one pattern among the plurality of calls; and
 - based on the at least one pattern, manipulating at least a portion of the configurable processing architecture to compute the at least one function.
2. The method of claim 1, wherein each call is directed to a computational device for computing each of the at least one function thereon.
3. The method of claim 2, wherein the computational device is any one of:
 - a multi-core central processing unit (CPU), a field-programmable gate array (FPGA), a graphics processing unit (GPU), a coarse-grained configurable architecture (CGRA), an intelligence processing unit (IPU), a neural-network accelerator, an application-specific integrated circuit (ASIC), and a quantum computer.
4. The method of claim 2, further comprising:
 - computing the at least one function by a computing grid instead of the computational device.
5. The method of claim 4, wherein the computing grid includes:
 - a data input mesh network;
 - a plurality of logical elements; and
 - a plurality of multiplexers, wherein the plurality of logical elements and the plurality of multiplexers form at least one computation group, wherein at least one

of the logical elements and at least one of the plurality of multiplexers forms each of the at least one computation group.

6. The method of claim 5, wherein each of the at least one of function is mapped for computation by at least one of the at least one computation group.

7. The method of claim 5, wherein the manipulation of the computing grid further comprises:

duplicating one of the at least one computation group mapped to a function when the pattern indicates re-occurrence of the function.

8. The method of claim 5, wherein the at least one computation group includes a first computation group and a second computation group, wherein the at least one function includes a first function and a second function, wherein the manipulation of the computing grid further comprises:

relocating the first computation group and the second computation group respectively mapped to the first function and the second function based on the pattern.

9. The method of claim 5, wherein the manipulation of the computing grid further comprises:

releasing a computation group when no calls for a function are mapped to the computation group.

10. The method of claim 5, wherein the data input mesh network includes at least one feeder.

11. The method of claim 1, wherein the manipulation is performed at runtime.

12. A configurable processing architecture, comprising:

a computing element, wherein the computing element includes at least one of: a computing grid, and a processing unit;

a memory connected to the processing unit, wherein the memory contains instructions that, when executed by the computing element, configure the configurable processing architecture to:

receive a plurality of calls for running at least one function;

identify at least one pattern among the plurality of calls; and

based on the at least one pattern, manipulate at least a portion of the configurable processing architecture, to compute the at least one function.

13. The configurable processing architecture of claim 12, wherein each call is directed to a computational device for computing each of the at least one function thereon.

14. The configurable processing architecture of claim 12, wherein the computing element is any one of: a multi-core central processing unit (CPU), a field-programmable gate array (FPGA), a graphics processing unit (GPU), a coarse-grained reconfigurable architecture (CGRA), a neural-network accelerator, an intelligence processing unit (IPU), an application-specific integrated circuit (ASIC), and a quantum computer.

15. The configurable processing architecture of claim 12, wherein the computing element further comprises:

a data input mesh network;

a plurality of logical elements; and

a plurality of multiplexers, wherein the plurality of logical elements and the plurality of multiplexers form at least one computation group, wherein at least one of the logical elements and at least one of the plurality of multiplexers forms each of the at least one computation group.

16. The configurable processing architecture of claim 15, wherein each of the at least one of function is mapped for computation by at least one of the at least one computation group.

17. The configurable processing architecture of claim 15, wherein the manipulation of the computing element further comprises:

duplicating one of the at least one computation group mapped to a function when the pattern indicates re-occurrence of the function.

18. The configurable processing architecture of claim 15, wherein the at least one computation group includes a first computation group and a second computation group, wherein the at least one function includes a first function and a second function, wherein the manipulation of the computing element further comprises:

relocating a first computation group and second computation group respectively mapped to a first function and second function, respective of the pattern.

19. The configurable processing architecture of claim 15, wherein the manipulation of the computing element further comprises:

releasing a computation group when no calls for a function are mapped to the computation group.

20. The configurable processing architecture of claim 15, wherein the data input mesh network includes at least one feeder.

21. The configurable processing of claim 12, wherein the manipulation is performed at runtime.

22. A non-transitory computer readable medium having stored thereon instructions for causing a processing unit to execute a process for runtime optimization of a configurable processing architecture, the process comprising:

receiving a plurality of calls for running at least one function;

identifying at least one pattern among the plurality of calls; and

based on the at least one pattern, manipulating at least a portion of the configurable processing architecture, to compute the least one function.