WHAT IS CLAIMED IS:

1. A system for executing a plurality of software threads, comprising:

a plurality of processing circuitries (110);

a plurality of memory areas (120) connected to the plurality of processing circuitries, each memory area associated with at least one of the plurality of processing circuitries; and

at least one hardware processor (101), connected to the plurality of processing circuitries and configured for:

in each of a plurality of iterations:

while the plurality of processing circuitries executes the plurality of software threads, collecting (201), for each software thread of one or more of the plurality of software threads, a plurality of thread statistical values indicative of a plurality of memory accesses to at least some of the plurality of memory areas performed when executing the software thread;

for at least one software thread of the one or more software threads, performing an analysis (210) comprising the plurality of thread statistical values thereof and using at least one state value of the at least one software thread, where the at least one state value is at least part of the plurality of thread statistical values collected for the at least one software thread, to identify (220) a preferred memory area of the plurality of memory areas; and

configuring (230) one of the at least one processing circuitry associated with the preferred memory area to execute the at least one software thread, wherein the one of the at least one processing circuitry associated with the preferred memory area is selected according to the at least one state value of the at least one software thread.

2. The system of claim 1, wherein the plurality of processing circuitries comprises at least one of: a central processing unit (CPU), a core of a multi-core processing unit (CPU core), a data processing unit (DPU), a graphics processing unit (GPU), a microcontroller unit (MCU), an accelerated processing unit (APU), 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, a reconfigurable

interconnected computing grid, comprising a plurality of reconfigurable logical elements connected by a plurality of reconfigurable data routing junctions, and part of a reconfigurable interconnected computing grid.

3. The system of claim 1, wherein at least one of the plurality of processing circuitries is integrated into the at least one hardware processor.

4. The system of claim 1, wherein the plurality of memory areas comprises at least one of:

a memory component electrically coupled with at least one of the plurality of processing circuitries;

a cache memory of another memory component connected to at least one other of the plurality of processing circuitries, where the cache memory is electrically coupled with the at least one other processing circuitry;

yet another memory component connected to at least one other of the processing circuitries via a bus or at least one digital communication network interface; and

at least part of a reconfigurable interconnected memory grid that is part of a reconfigurable interconnected computing grid.

5. The system of claim 1, wherein each of the plurality of memory areas is associated with the at least one of the plurality of processing circuitries according to at least one memory access metric.

6. The system of claim 5, wherein the at least one memory metric comprises at least one of the group of memory metrics consisting of: access latency, throughput, a cache utilization metric, memory capacity, topological distance, memory allocation ownership, and memory coherence.

7. The system of claim 1, wherein the preferred memory area contains program data used by the at least one software thread; and

wherein performing the analysis comprising the plurality of thread statistical values of the at least one software thread further comprises identifying that the preferred memory area contains program data used by the at least one software thread.

8. The system of claim 1, wherein the at least one hardware processor is further configured for:

identifying (240) more than one software thread of the plurality of software threads accessing the preferred memory area; and

configuring (241) at least some of the at least one processing circuitry associated with the preferred memory areas to execute the more than one software thread.

9. The system of claim 1, wherein performing the analysis comprising the plurality of thread statistical values of the at least one software thread further comprises using at least one other plurality of thread statistical values of at least one other thread of the one or more software threads.

10. The system of claim 1, wherein performing the analysis comprising the plurality of thread statistical values of the at least one software thread further comprises using a plurality of control-flow statistical values collected while executing the plurality of software threads.

11. The system of claim 1, wherein the at least one hardware processor is further configured for migrating (250) at least some application data from one memory area of the plurality of memory areas to another memory area of the plurality of memory areas in response to performing the analysis;

wherein the at least some application data is accessed by at least one additional thread of the plurality of software threads;

wherein the other memory area is the preferred memory area; and

wherein the at least one hardware processor is further configured for configuring (251) the at least one processing circuitry associated with the preferred memory area to execute the at least one additional thread.

12. The system of claim 1, wherein the plurality of software threads each execute part of a compute kernel of a software program, where the compute kernel comprises a plurality of loop iterations for accessing application data of the software program such that each of the plurality of

software threads executes some of the plurality of loop iterations, accessing at least some of the application data thereby; and

wherein the at least one hardware processor is further configured for identifying the preferred memory area by additionally or alternatively performing another analysis of a code of the plurality of software threads to identify (305) the compute kernel and additionally or alternatively to identify the application data.

13. The system of claim 1, wherein the at least one software thread comprises a plurality of telemetry computer instructions for the purpose of collecting the plurality of thread statistical values.

14. The system of claim 1, wherein at least one of the plurality of processing circuitries comprises telemetry circuitry for the purpose of collecting the plurality of thread statistical values.

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

accessing (306) a plurality of memory access exception counters, each associated with at least one additional software thread of the plurality of software threads and indicative of an amount of memory exceptions when executing the additional software thread associated therewith; and

performing the analysis further comprising using the plurality of memory access exception counters;

wherein the at least one hardware processor configures the one of the at least one processing circuitry associated with the preferred memory area in an identified iteration of the plurality of iterations; and

wherein the at least one hardware processor is further configured for, in another iteration of the plurality of iterations executed after the identified iteration:

identifying (401) for the at least one software thread that the memory access exception counter associated therewith exceeds an exception threshold value;

identifying (405) a new preferred memory area of the plurality of memory areas; and

configuring (407) one of the at least one processing circuitry associated with the other preferred memory area to execute the at least one software thread.

16. The system of claim 1, wherein the plurality of thread statistical values comprises at least one of: a latency of a memory access, a throughput of a memory area, an amount of memory accesses to a memory area, an amount of memory accesses to a memory area by a thread of the plurality of software threads, an amount of memory accesses to a memory area by a subset of the plurality of software threads, an amount of cache misses, and an amount of translation lookaside buffer misses.

17. The system of claim 1,

wherein the at least one hardware processor is further configured for:

generating for a first processing circuitry of the plurality of processing circuitries a first page table for mapping at least a first subset of the plurality of memory areas according to a first set of memory attributes; and

generating for a second processing circuitry of the plurality of processing circuitries a second page table for mapping at least a second subset of the plurality of memory areas according to a second set of memory attributes;

wherein at least a first statistical value of the plurality of thread statistical values is collected according to at least one exception when accessing the first page table; and

wherein at least a second statistical value of the plurality of thread statistical values is collected according to at least one other exception when accessing the second page table.

18. The system of claim 1, wherein the at least one state value comprises at least one register value, where a register value is a value of a register of a processing circuitry of the plurality of processing circuitries when executing the at least one software thread.

19. A method for executing a plurality of software threads, comprising:

in each of a plurality of iterations:

while a plurality of processing circuitries executes the plurality of software threads, collecting (201), for each software thread of one or more of the plurality of software

threads, a plurality of thread statistical values indicative of a plurality of memory accesses to at least some of a plurality of memory areas performed when executing the software thread, where the plurality of memory areas are connected to the plurality of processing circuitries, each memory area associated with at least one of the plurality of processing circuitries;

for at least one software thread of the one or more software threads, performing an analysis (210) comprising the plurality of thread statistical values thereof and using at least one state value of the at least one software thread, where the at least one state value is at least part of the plurality of thread statistical values collected for the at least one software thread, to identify (220) a preferred memory area of the plurality of memory areas; and

configuring (230) one of the at least one processing circuitry associated with the preferred memory area to execute the at least one software thread, wherein the one of the at least one processing circuitry associated with the preferred memory area is selected according to the at least one state value of the at least one software thread.

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

a non-transitory computer readable storage medium;

program instructions for in each of a plurality of iterations:

while a plurality of processing circuitries executes the plurality of software threads, collecting (201) for each software thread of one or more of the plurality of software threads a plurality of thread statistical values indicative of a plurality of memory accesses to at least some of a plurality of memory areas performed when executing the software thread, where the plurality of memory areas are connected to the plurality of processing circuitries, each memory area associated with at least one of the plurality of processing circuitries;

for at least one software thread of the one or more software threads, performing an analysis (210) comprising the plurality of thread statistical values thereof and using at least one state value of the at least one software thread, where the at least one state value is at least part of the plurality of thread statistical values collected for the at least one software thread, to identify (220) a preferred memory area of the plurality of memory areas; and

configuring (230) one of the at least one processing circuitry associated with the preferred memory area to execute the at least one software thread, wherein the one of the at least one processing circuitry associated with the preferred memory area is selected according to the at least one state value of the at least one software thread;

wherein the program instructions are executed by at least one computerized processor from the non-transitory computer readable storage medium.