

SYNCHRONOUS HARDWARE EVENT COLLECTION

BACKGROUND

[0001] This specification relates to analyzing execution of program code.

[0002] Effective performance analysis of distributed software executing within distributed hardware components can be a complex task. Distributed hardware components can be respective processor cores of two or more Central Processing Units (CPUs) (or Graphics Processing Units (GPUs)) that cooperate and interact to execute portions of a larger software program or program code.

[0003] From the hardware perspective (e.g., within the CPUs or GPUs), there are generally two types of information or features available for performance analysis: 1) hardware performance counters and 2) hardware event traces.

SUMMARY

[0003a] According to a first aspect of the present invention, there is provided a computer-implemented method performed by an event collection system having one or more hardware circuit components, the method comprising:

monitoring execution of program code by a first multi-core neural network processor in the event collection system, the first multi-core neural network processor being configured to execute a first portion of the program code to perform computations for inference workloads of a multi-layer neural network, wherein the multi-layer neural network is implemented on a hardware circuit;

monitoring execution of the program code by a second multi-core neural network processor in the event collection system, the second multi-core neural network processor being configured to execute a second portion of the program code to perform the computations for the inference workloads of the multi-layer neural network;

detecting, by the event collection system, that a trigger condition is satisfied by identifying an occurrence of an operand in the first portion of the program code or the second portion of the program code, wherein each portion of the program code includes multiple code sequences that represent computational steps of the inference workload and the operand includes a first parameter value that indicates a particular code sequence of the multiple code sequences for initiating synchronized trace events;

responsive to detecting that the trigger condition is satisfied, initiating, by the event collection system, a first synchronized trace event that occurs during a particular computational step of the inference workload to generate trace data identifying respective hardware events occurring across distributed hardware circuit components of the event collection system, and for each of the respective hardware events, the trace data comprises a hardware event time stamp;

generating, by the event collection system and using the trace data, a data structure that correlates the trace data based on the particular computational step of the inference workload;

transmitting, to a host controller and using a host interface block of the event collection system, the data structure that includes the trace data correlated based on the particular computational step of the inference workload;

determining, by the host controller, one or more performance attributes based on the respective hardware events that were identified by the trace data in the data structure received using the host interface block; and

using, by the host controller, the one or more performance attributes to analyze execution of the program code by the first and second multi-core neural network processors when the particular computational step is executed at the first and second multi-core neural network processors at different respective times.

[0003b] According to a second aspect of the present invention, there is provided a hardware event collection system, comprising:

one or more processor components including one or more processor cores;

one or more machine-readable storage units storing instructions that are executable by the one or more processor components to cause performance of operations comprising:

monitoring execution of program code by a first multi-core neural network processor in the event collection system, the first multi-core neural network processor being configured to execute a first portion of the program code to perform computations for inference workloads of a multi-layer neural network, wherein the multi-layer neural network is implemented on a hardware circuit;

monitoring execution of the program code by a second multi-core neural network processor in the event collection system, the second multi-core neural network processor being configured to execute a second portion of the program code to perform the computations for the inference workloads of the multi-layer neural network;

detecting, by the event collection system, that a trigger condition is satisfied by identifying an occurrence of an operand in the first portion of the program code or the second portion of the program code, wherein each portion of the program code includes multiple code sequences that represent computational steps of the inference workload and the operand includes a first parameter value that indicates a particular code sequence of the multiple code sequences for initiating synchronized trace events;

responsive to detecting that the trigger condition is satisfied, initiating, by the event collection system, a first synchronized trace event that occurs during a particular computational step of the inference workload to generate trace data identifying respective hardware events occurring across distributed hardware circuit components of the event collection system, and for each of the respective hardware events, the trace data comprises a hardware event time stamp;

generating, by the event collection system and using the trace data, a data structure that correlates the trace data based on the particular computational step of the inference workload;

transmitting, to a host controller and using a host interface block of the event collection system, the data structure that includes the trace data correlated based on the particular computational step of the inference workload;

determining, by the host controller, one or more performance attributes based on the respective hardware events that were identified by the trace data in the data structure received using the host interface block; and

using, by the host controller, the one or more performance attributes to analyze execution of the program code by the first and second multi-core neural network processors when the particular computational step is executed at the first and second multi-core neural network processors at different respective times.

[0003c] According to a third aspect of the present invention, there is provided a computer-implemented method performed by an event collection system having one or more hardware circuit components, the method comprising:

monitoring execution of program code by a multi-core neural network processor in the event collection system, the program code being executed to perform computations for inference workloads of a multi-layer neural network that is implemented on a hardware circuit, and the multi-core neural network processor being configured to:

execute a first portion of the program code using a first core of the multi-core neural network processor; and

execute a second portion of the program code using a second core of the multi-core neural network processor;

detecting, by the event collection system, that a trigger condition is satisfied by identifying an occurrence of an operand in the first or second portion of the program code, wherein the program code includes multiple code sequences that represent computational steps of the inference workload and the operand includes a first parameter value that indicates a particular code sequence of the multiple code sequences for initiating synchronized trace events;

responsive to detecting that the trigger condition is satisfied, generating, by the event collection system and during a particular computational step of the inference workload, a control signal received by a count register of the multi-core neural network processor, the control signal causing count data for a hardware event to be stored in the count register when a trace event synchronized between the first and second processor cores is initiated by the event collection system during the particular computational step of the inference workload;

generating, by the event collection system and using the count data, a data structure that correlates the count data based on the particular computational step of the inference workload;

transmitting, to a host controller and using a host interface block of the event collection system, the data structure that includes the count data correlated based on the particular computational step of the inference workload;

determining, by the host controller, one or more performance attributes based on the count data for the hardware event that was stored in the count register and included in the data structure received using the host interface block; and

using, by the host controller, the one or more performance attributes to analyze the executing program code when the particular computational step is executed at the first and second cores of the multi-core neural network processor at different respective times.

[0004] There is disclosed a computer-implemented method executed by one or more processors, the method including, monitoring execution of program code by a first processor component, the first processor component being configured to execute at least a first portion of the program code; and monitoring execution of the program code by a second processor component, the second processor component being configured to execute at least a second portion of the program code.

[0005] The method further includes, detecting, by a computing system, that a trigger condition is satisfied based on at least one of: i) identifying an occurrence of an operand in at least the first portion of the program code or the second portion of the program code, the operand including a first parameter value used to initiate one or more trace events, or ii) determining that a current time of at least one clock of the computing system indicates a predefined time value used to initiate one or more trace events.

[0006] Responsive to detecting that the trigger condition is satisfied, the method includes, initiating, by the computing system, at least a first trace event that generates trace data, the trace data identifying respective hardware events occurring across distributed processor units that include at least the first processor component and the second processor component. For each of the respective hardware events, the trace data comprises at least a hardware event time stamp. The method further includes, using, by the computing system, the trace data to generate

[0012] In some implementations, the method further includes: inserting, by a compiler of the computing system, the operand of the trigger condition into at least the first portion of program code executed by the first processor component; and inserting, by the compiler of the computing system, at least one predefined time value of the trigger condition into at least the second portion of program code executed by the second processor component.

[0013] In some implementations, initiating at least one of the first trace event or the second trace event, includes: generating, by the computing system, a first control signal received by a first count register of the first processor core, the first control signal causing data associated with a first hardware event to be stored in the first count register; and generating, by the computing system, a second control signal received by a second count register of the second processor core, the second control signal causing data associated with a second hardware event to be stored in the second count register.

[0014] In some implementations, the data associated with one of the first hardware event or the second hardware event, comprises at least one of: a number of bytes written to a particular memory buffer of a particular processor core of the distributed processor units; or a number of instructions executed by a particular processor core of the distributed processor units.

[0015] In some implementations, the method further includes: identifying, by the computing system, an occurrence of a second operand in at least a portion of the program code executed by the first or second processor components, the second operand including a second parameter value; determining, by the computing system, that a filter condition is satisfied based on the second parameter value of the second operand one of: exceeding a particular bound value of a register or being below a particular bound value of the register; and responsive to determining that the filter condition is satisfied, filtering, by the computing system, one or more trace events, wherein filtering the one or more trace events comprises precluding storage of trace data associated with one or more hardware events.

[0016] There is disclosed a hardware event collection system, including: one or more processors including one or more processor cores; one or more machine-readable storage units for storing instructions that are executable by the one or more processors to perform operations comprising: monitoring execution of program code by a first processor component, the first processor component being configured to execute at least a first portion of the program code; and monitoring execution of the program code by a second processor component, the second processor component being configured to execute at least a second portion of the program code.

[0017] The method further includes, detecting, by a computing system, that a trigger condition is satisfied based on at least one of: i) identifying an occurrence of an operand in at least the first portion of the program code or the second portion of the program code, the operand including a first parameter value used to initiate one or more trace events, or ii) determining that a current time of at least one clock of the computing system indicates a predefined time value used to initiate one or more trace events.

[0018] Responsive to detecting that the trigger condition is satisfied, the method includes, initiating, by the computing system, at least a first trace event that generates trace data, the trace data identifying respective hardware events occurring across distributed processor units that include at least the first processor component and the second processor component. For each of the respective hardware events, the trace data comprises at least a hardware event time stamp. The method further includes, using, by the computing system, the trace data to generate a correlated set of trace data that indicates at least a time ordered sequence of the respective hardware events that are generated when the trigger condition is satisfied.

[0019] There is disclosed a computer-implemented method executed by one or more processors, the method including, monitoring execution of program code by a processor component, the processor component being configured to execute at least a first portion of the program code.

[0020] The method further includes, detecting, by a computing system, that a trigger condition is satisfied based on at least one of: i) identifying an occurrence of an operand in at least the first portion of the program code, the operand including a first parameter value used to initiate one or more trace events, or ii) determining that a current time of at least one clock of the computing system indicates a predefined time value used to initiate the one or more trace events.

[0021] Responsive to detecting that the trigger condition is satisfied, the method further includes, generating, by the computing system, a control signal received by a count register of the processor component, the control signal causing count data associated with a hardware event to be stored in the count register; and generating, by the computing system, a data structure that indicates one or more performance attributes associated with the executing program code, the data structure being generated based on one or more count parameters of the stored count data.

[0022] These and other implementations can each optionally include one or more of the following features. For example, in some implementations, the count register is one of multiple

performance counters configured to store count data about the performance of one or more processor components of the computing system, and wherein at least one performance counter includes one of: an activity counter, a stall counter, statistical counter, or a sampling counter.

[0023] In some implementations, the one or more count parameters indicates one of: i) a number of instructions received by a particular processor component; ii) a number of instructions processed by the particular processor component; iii) a number of instructions executed by the particular processor component; or iv) a number of memory reads, or a number of memory writes, performed by the particular processor component.

[0024] In some implementations, a performance attribute associated with the executing program code comprises one of: i) a stall frequency of a particular processor component executing the program code; ii) an indication that utilization of the particular processor component is below a threshold utilization; or iii) an indication that a data storage queue used by the particular processor component is at, or below, a queue occupancy threshold.

[0025] There are disclosed corresponding systems, apparatus, and computer programs, configured to perform the actions of the methods, encoded on computer storage devices. A system of one or more computers can be so configured by virtue of software, firmware, hardware, or a combination of them installed on the system that in operation cause the system to perform the actions. One or more computer programs can be so configured by virtue of having instructions that, when executed by data processing apparatus, cause the apparatus to perform the actions.

[0026] The subject matter described in this specification can be implemented in particular embodiments so as to realize one or more of the following advantages. The described hardware tracing system enhances computational efficiency by using dynamic triggers that execute through hardware knobs/features. The triggers enable synchronized capture of events based, in part, on a global time counter, embedded logical operands, and hardware registers, rather than purely through coded sequences which typically require system resources to monitor and execute event capture.

[0027] Likewise, system memory use is optimized when hardware triggers are tuned to capture short duration events and synchronized global captures that occur simultaneously instead of as uncorrelated inefficient event captures. Controls for shorter duration synchronized event capture mitigates information overload thereby enabling some memory resources to remain unused and available for other system processes.

CLAIMS

1.A computer-implemented method performed by an event collection system having one or more hardware circuit components, the method comprising:

monitoring execution of program code by a first multi-core neural network processor in the event collection system, the first multi-core neural network processor being configured to execute a first portion of the program code to perform computations for inference workloads of a multi-layer neural network, wherein the multi-layer neural network is implemented on a hardware circuit;

monitoring execution of the program code by a second multi-core neural network processor in the event collection system, the second multi-core neural network processor being configured to execute a second portion of the program code to perform the computations for the inference workloads of the multi-layer neural network;

detecting, by the event collection system, that a trigger condition is satisfied by identifying an occurrence of an operand in the first portion of the program code or the second portion of the program code, wherein each portion of the program code includes multiple code sequences that represent computational steps of the inference workload and the operand includes a first parameter value that indicates a particular code sequence of the multiple code sequences for initiating synchronized trace events;

responsive to detecting that the trigger condition is satisfied, initiating, by the event collection system, a first synchronized trace event that occurs during a particular computational step of the inference workload to generate trace data identifying respective hardware events occurring across distributed hardware circuit components of the event collection system, and for each of the respective hardware events, the trace data comprises a hardware event time stamp;

generating, by the event collection system and using the trace data, a data structure that correlates the trace data based on the particular computational step of the inference workload;

transmitting, to a host controller and using a host interface block of the event collection system, the data structure that includes the trace data correlated based on the particular computational step of the inference workload;

determining, by the host controller, one or more performance attributes based on the respective hardware events that were identified by the trace data in the data structure received using the host interface block; and

using, by the host controller, the one or more performance attributes to analyze execution of the program code by the first and second multi-core neural network processors when the particular computational step is executed at the first and second multi-core neural network processors at different respective times.

2. The method of claim 1, wherein the first synchronized trace event generates trace data indicating a unique trace identifier for respective hardware events, and wherein the hardware events include a plurality of synchronized hardware events, and two hardware events are synchronized when the events are associated with the same computational step of the inference workload.

3. The method of claim 1, wherein detecting that the trigger condition is satisfied comprises:

detecting, by the event collection system, that one of:

i) the first parameter value of the operand exceeds a first bound value of a register, or

ii) a current time indicated by a clock of the event collection system exceeds a first bound value of a register; and

responsive to detecting that the trigger condition is satisfied, initiating, by the event collection system, a second synchronized trace event that generates trace data, wherein the trace data identifies at least one attribute that is shared among the respective hardware events occurring across the event collection system.

4. The method of claim 3, further comprising:

detecting, by the event collection system, that one of:

a second parameter value of the operand exceeds a second bound value of the register, or

the current time indicated by the clock of the event collection system exceeds a second bound value of the register; and

responsive to detecting, stopping, by the event collection system, the second trace event when the second parameter value of the operand exceeds the second bound value, or when the second predefined time value exceeds the second bound value.

5. The method of claim 3, wherein:

the operand further includes:

a global control parameter that indicates a particular performance state of the event collection system; and the first bound value of the register includes at least one of:

a predefined time value indicated by a global time clock of the event collection system;

or

a particular time value of a predefined time window associated with the global time clock.

6. The method of claim 3, wherein:

the operand has a first binary data structure and the first parameter value of the first operand corresponds to a trace-mark,

the first bound value of the register has a second binary data structure and the current time is indicated by a global time clock,

and wherein the global time clock is used by the event collection system to generate one or more hardware event time stamps.

7. The method of claim 3, further comprising:

inserting, by a compiler of the event collection system, the operand of the trigger condition into the first portion of program code executed by the first multi-core neural network processor; and

inserting, by the compiler of the event collection system, a predefined time value of the trigger condition into the second portion of program code executed by the second multi-core neural network processor.

8. The method of claim 3, wherein initiating at least one of the first synchronized trace event or the second synchronized trace event, comprises:

generating, by the event collection system, a first control signal received by a first count register of the first multi-core neural network processor, the first control signal causing data for a first hardware event to be stored in the first count register; and

generating, by the event collection system, a second control signal received by a second count register of the second multi-core neural network processor, the second control signal causing data for a second hardware event to be stored in the second count register.

9. The method of claim 8, wherein the data for one of the first hardware event or the second hardware event, comprises at least one of:

a number of bytes written to a particular memory buffer of a particular multi-core neural network processor of the event collection system; or

a number of instructions executed by a particular multi-core neural network processor of the event collection system.

10. The method of claim 1, further comprising:

identifying, by the event collection system, an occurrence of a second operand in the portion of the program code executed by the first or second multi-core neural network processor, the second operand including a second parameter value;

determining, by the event collection system, that a filter condition is satisfied based on the second parameter value of the second operand exceeding a particular bound value of a register or being below a particular bound value of the register; and

responsive to determining that the filter condition is satisfied, filtering, by the event collection system, one or more trace events, wherein filtering the one or more trace events comprises precluding storage of trace data for one or more hardware events.

11. A hardware event collection system, comprising:

one or more processor components including one or more processor cores;

one or more machine-readable storage units storing instructions that are executable by the one or more processor components to cause performance of operations comprising:

monitoring execution of program code by a first multi-core neural network processor in the event collection system, the first multi-core neural network processor being configured to execute a first portion of the program code to perform computations for inference workloads of a multi-layer neural network, wherein the multi-layer neural network is implemented on a hardware circuit;

monitoring execution of the program code by a second multi-core neural network processor in the event collection system, the second multi-core neural network processor being configured to execute a second portion of the program code to perform the computations for the inference workloads of the multi-layer neural network;

detecting, by the event collection system, that a trigger condition is satisfied by identifying an occurrence of an operand in the first portion of the program code or the second portion of the program code, wherein each portion of the program code includes multiple code sequences that represent computational steps of the inference workload and the operand

includes a first parameter value that indicates a particular code sequence of the multiple code sequences for initiating synchronized trace events;

responsive to detecting that the trigger condition is satisfied, initiating, by the event collection system, a first synchronized trace event that occurs during a particular computational step of the inference workload to generate trace data identifying respective hardware events occurring across distributed hardware circuit components of the event collection system, and for each of the respective hardware events, the trace data comprises a hardware event time stamp;

generating, by the event collection system and using the trace data, a data structure that correlates the trace data based on the particular computational step of the inference workload;

transmitting, to a host controller and using a host interface block of the event collection system, the data structure that includes the trace data correlated based on the particular computational step of the inference workload;

determining, by the host controller, one or more performance attributes based on the respective hardware events that were identified by the trace data in the data structure received using the host interface block; and

using, by the host controller, the one or more performance attributes to analyze execution of the program code by the first and second multi-core neural network processors when the particular computational step is executed at the first and second multi-core neural network processors at different respective times.

12. The hardware event collection system of claim 11, wherein the first synchronized trace event generates trace data indicating a unique trace identifier for respective hardware events, and wherein the hardware events include a plurality of synchronized hardware events, and two hardware events are synchronized when the events are associated with the same computational step of the inference workload.

13. The hardware event collection system of claim 11, wherein detecting that the trigger condition is satisfied comprises:

detecting, by the event collection system, that one of:

i) the first parameter value of the operand exceeds a first bound value of a register, or

ii) a current time indicated by a clock of the event collection system exceeds a first bound value of a register; and

responsive to detecting that the trigger condition is satisfied, initiating, by the event collection system, a second synchronized trace event that generates trace data, wherein the trace data identifies at least one attribute that is shared among the respective hardware events occurring across the event collection system.

14. The hardware event collection system of claim 13, wherein the operations further comprise:
detecting, by the event collection system, that one of:

a second parameter value of the operand exceeds a second bound value of the register,

or

the current time indicated by the clock of the event collection system exceeds a second bound value of the register; and

responsive to detecting, stopping, by the event collection system, the second trace event when the second parameter value of the operand exceeds the second bound value, or when the second predefined time value exceeds the second bound value.

15. The hardware event collection system of claim 13, wherein:

the operand further includes:

a global control parameter that indicates a particular performance state of the event collection system; and

the first bound value of the register includes at least one of:

a predefined time value indicated by a global time clock of the event collection system;

or

a particular time value of a predefined time window associated with the global time clock.

16. The hardware event collection system of claim 13, wherein:

the operand has a first binary data structure and the first parameter value of the first operand corresponds to a trace-mark,

the first bound value of the register has a second binary data structure and the current time is indicated by a global time clock,

and wherein the global time clock is used by the event collection system to generate one or more hardware event time stamps.

17. A computer-implemented method performed by an event collection system having one or more hardware circuit components, the method comprising:

monitoring execution of program code by a multi-core neural network processor in the event collection system, the program code being executed to perform computations for inference workloads of a multi-layer neural network that is implemented on a hardware circuit, and the multi-core neural network processor being configured to:

execute a first portion of the program code using a first core of the multi-core neural network processor; and

execute a second portion of the program code using a second core of the multi-core neural network processor;

detecting, by the event collection system, that a trigger condition is satisfied by identifying an occurrence of an operand in the first or second portion of the program code, wherein the program code includes multiple code sequences that represent computational steps of the inference workload and the operand includes a first parameter value that indicates a particular code sequence of the multiple code sequences for initiating synchronized trace events;

responsive to detecting that the trigger condition is satisfied, generating, by the event collection system and during a particular computational step of the inference workload, a control signal received by a count register of the multi-core neural network processor, the control signal causing count data for a hardware event to be stored in the count register when a trace event synchronized between the first and second processor cores is initiated by the event collection system during the particular computational step of the inference workload;

generating, by the event collection system and using the count data, a data structure that correlates the count data based on the particular computational step of the inference workload;

transmitting, to a host controller and using a host interface block of the event collection system, the data structure that includes the count data correlated based on the particular computational step of the inference workload;

determining, by the host controller, one or more performance attributes based on the count data for the hardware event that was stored in the count register and included in the data structure received using the host interface block; and

using, by the host controller, the one or more performance attributes to analyze the executing program code when the particular computational step is executed at the first and second cores of the multi-core neural network processor at different respective times.

18. The method of claim 17, wherein the count register is one of multiple performance counters configured to store count data about the performance of the inference workloads by multi-core

neural network processors of the event collection system, and wherein at least one performance counter includes one of: an activity counter, a stall counter, statistical counter, or a sampling counter.

19. The method of claim 18, wherein one or more count parameters of the count data indicates at least one of:

- i) a number of instructions received by a particular multi-core neural network processor;
- ii) a number of instructions processed by the particular multi-core neural network processor;
- iii) a number of instructions executed by the particular multi-core neural network processor; or
- iv) a number of memory reads and a number of memory writes performed by the particular multi-core neural network processor.

20. The method of claim 17, wherein a performance attribute that is used to analyze the executing program code comprises at least one of:

- i) a stall frequency of a particular multi-core neural network processor executing the program code;
- ii) an indication that utilization of the particular multi-core neural network processor is below a threshold utilization; or
- iii) an indication that a data storage queue used by the particular multi-core neural network processor is at, or below, a queue occupancy threshold.