

DISTRIBUTED HARDWARE TRACING

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application is related to U.S. Patent Application No.15/472,932, entitled “Synchronous Hardware Event Collection,” filed on March 29, 2017, and Attorney Docket Number 16113-8129001. The entire disclosure of U.S. Patent Application No. 15/472,932 is expressly incorporated by reference herein in its entirety.

BACKGROUND

[0002] This specification relates to analyzing execution of program code.

[0003] 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.

[0004] 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

[0004a] According to a first aspect of the present invention, there is disclosed a method performed using a tracing system, the method comprising:

providing code to a processor comprising a plurality of nodes;

executing the code using each of the plurality of nodes of the processor;

detecting, at each node of the plurality of nodes, that a trace bit is set to a first value that causes trace data to be generated at the processor;

generating, locally at each node, a respective portion of the trace data in response to detecting that the trace bit is set to the first value; and

providing each of the respective portions of trace data generated at the processor to a host for analysis at the host.

[0004b] According to a second aspect of the present invention, there is disclosed a tracing system comprising:

one or more processing devices; and

one or more non-transitory machine-readable storage devices storing instructions that are executable by the one or more processing devices to cause performance of operations comprising:

- providing code to a processor comprising a plurality of nodes;
- executing the code using each of the plurality of nodes of the processor;
- detecting, at each node of the plurality of nodes, that a trace bit is set to a first value that causes trace data to be generated at the processor;
- generating, locally at each node, a respective portion of the trace data in response to detecting that the trace bit is set to the first value; and
- providing each of the respective portions of trace data generated at the processor to a host for analysis at the host.

[0004c] According to a third aspect of the present invention, there is disclosed one or more non-transitory machine-readable storage devices storing instructions that are executable by one or more processing devices of a tracing system to cause performance of operations comprising:

- providing code to a processor comprising a plurality of nodes;
- executing the code using each of the plurality of nodes of the processor;
- detecting, at each node of the plurality of nodes, that a trace bit is set to a first value that causes trace data to be generated at the processor;
- generating, locally at each node, a respective portion of the trace data in response to detecting that the trace bit is set to the first value; and
- providing each of the respective portions of trace data generated at the processor to a host for analysis at the host.

[0005] ~~In general, one aspect of the subject matter described in this specification can be embodied in~~ There is disclosed a computer-implemented method executed by one or more processors. The method includes, 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.

[0006] The method further includes, storing, by the computing system and in at least one memory buffer, data identifying one or more hardware events occurring across processor units that include the first processor component and the second processor component. Each hardware event represents at least one of data communications associated with a memory access operation of the program code, an issued instruction of the program code, or an executed instruction of the program code. The data identifying each of the one or more hardware events includes a hardware event time stamp and metadata characterizing the

[0012] In some implementations, generating the data structure further includes: comparing, by the computing system, at least hardware event time stamps of respective events in the first subset of data identifying hardware events with at least hardware event time stamps of respective events in the second subset of data identifying hardware events; and providing, by the computing system and for presentation in the data structure, a correlated set of hardware events based, in part, on the comparison between the respective events in the first subset and the respective events in the second subset.

[0013] In some implementations, the generated data structure identifies at least one parameter that indicates a latency attribute of a particular hardware event, the latency attribute indicating at least a duration of the particular hardware event. In some implementations, at least one processor of the computing system is a multi-core multi-node processor having one or more processor components, and the one or more hardware events correspond, in part, to data transfers that occur between at least the first processor component of a first node and the second processor component of a second node.

[0014] In some implementations, the first processor component and the second processor component is one of: a processor, a processor core, a memory access engine, or a hardware feature of the computing system, and the one or more hardware events correspond, in part, to movement of data packets between a source and a destination; and metadata characterizing the hardware event corresponds to at least one of a source memory address, a destination memory address, a unique trace identification number, or a size parameter associated with a direct memory access (DMA) trace.

[0015] In some implementations, a particular trace (ID) number is associated with multiple hardware events that occur across the processor units, and wherein the multiple hardware events correspond to a particular memory access operation, and the particular trace ID number is used to correlate one or more hardware events of the multiple hardware events and is used for determining a latency attribute of the memory access operation based on the correlation.

[0016] ~~Another aspect of the subject matter described in this specification can be embodied in~~ There is disclosed a distributed hardware tracing 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 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

What is claimed is:

1. ~~A method performed using a tracing system, the method comprising:~~
 - ~~providing code to a processor comprising a plurality of nodes;~~
 - ~~executing the code using each of the plurality of nodes of the processor;~~
 - ~~for a particular node of the plurality of nodes included in the processor:~~
 - ~~detecting that a trace bit is set to a first value that causes trace data to be generated at the processor using a counter of the tracing system;~~
 - ~~generating the trace data in response to detecting that the trace bit is set to the first value, wherein the trace data is stored in the counter; and~~
 - ~~providing the trace data stored at the counter to a host for analysis at the host.~~
2. ~~The method of claim 1, wherein:~~
 - ~~at least one node of the plurality of nodes is a fixed function component node that is operable to execute a specific operation based on an instruction issued by the code executed by the processor, the specific operation being used to generate the trace data in response to detecting that the trace bit is set to the first value.~~
3. ~~The method of claim 2, wherein the fixed function component node is a memory access engine and generating the trace data comprises:~~
 - ~~generating, based on the instruction issued by the code, a descriptor for a memory access operation to be performed using the memory access engine, wherein the descriptor includes the trace bit that is set to the first value to cause the trace data to be generated at the processor using the counter.~~

4. ~~The method of claim 3, wherein:~~

~~the specific operation comprises performing the memory access operation using the memory access engine;~~

~~the memory access operation comprises multiple intermediate operations involving the particular node and at least one other node of the plurality of nodes; and~~

~~performing the memory access operation causes trace data associated with an intermediate operation to be stored in a memory buffer of the particular node.~~

5. ~~The method of claim 1, wherein generating the trace data comprises:~~

~~generating data representing a trace entry to be provided to a trace chain corresponding to a data communication bus that couples two or more nodes of the plurality of nodes.~~

6. ~~The method of claim 5, further comprising:~~

~~providing, using the trace chain, the data representing the trace entry to be stored in a chip manager of the tracing system; and~~

~~storing the data representing the trace entry in the chip manager, wherein the chip manager is operable to store multiple trace entries for a plurality of hardware events that occur at the processor when the code is being executed.~~

7. ~~The method of claim 1, further comprising:~~

~~embedding, by the host, an operand in the code provided to the processor to cause a node of the processor to store trace data that describes a hardware event, wherein the hardware event occurs during execution of the code at the processor.~~

8. ~~The method of claim 7, wherein providing the code comprises:~~

~~providing the code in response to compiling the code by the host, the host being coupled to the processor by a host interface and a node fabric of the tracing system, wherein the host uses each of the host interface and the node fabric to provide the compiled code to the processor.~~

~~9. The method of claim 8, wherein compiling the code comprises:~~

~~embedding the operand in the code provided to the processor to enable performance analysis of the compiled code using a monitoring engine of the host that communicates with the host interface and the node fabric of the tracing system.~~

~~10. A tracing system comprising:~~

~~one or more processing devices; and~~

~~one or more non-transitory machine-readable storage devices storing instructions that are executable by the one or more processing devices to cause performance of operations comprising:~~

~~providing code to a processor comprising a plurality of nodes;~~

~~executing the code using each of the plurality of nodes of the processor;~~

~~for a particular node of the plurality of nodes included in the processor:~~

~~detecting that a trace bit is set to a first value that causes trace data to be generated at the processor using a counter of the tracing system;~~

~~generating the trace data in response to detecting that the trace bit is set to the first value, wherein the trace data is stored in the counter; and~~

~~providing the trace data stored at the counter to a host for analysis at the host.~~

~~11. The tracing system of claim 10, wherein:~~

~~at least one node of the plurality of nodes is a fixed function component node that is operable to execute a specific operation based on an instruction issued by the code executed by the processor, the specific operation being used to generate the trace data in response to detecting that the trace bit is set to the first value.~~

~~12. The tracing system of claim 11, wherein the fixed function component node is a memory access engine and generating the trace data comprises:~~

~~generating, based on the instruction issued by the code, a descriptor for a memory access operation to be performed using the memory access engine, wherein the descriptor includes the trace bit that is set to the first value to cause the trace data to be generated at the processor using the counter.~~

~~13. The tracing system of claim 12, wherein:~~

~~the specific operation comprises performing the memory access operation using the memory access engine;~~

~~the memory access operation comprises multiple intermediate operations involving the particular node and at least one other node of the plurality of nodes; and~~

~~performing the memory access operation causes trace data associated with an intermediate operation to be stored in a memory buffer of the particular node.~~

~~14. The tracing system of claim 10, wherein generating the trace data comprises:~~

~~generating data representing a trace entry to be provided to a trace chain corresponding to a data communication bus that couples two or more nodes of the plurality of nodes.~~

~~15. The tracing system of claim 14, wherein the operations comprise:~~

~~providing, using the trace chain, the data representing the trace entry to be stored in a chip manager of the tracing system; and~~

~~storing the data representing the trace entry in the chip manager, wherein the chip manager is operable to store multiple trace entries for a plurality of hardware events that occur at the processor when the code is being executed.~~

~~16. The tracing system of claim 10, wherein the operations comprise:~~

~~embedding, by the host, an operand in the code provided to the processor to cause a node of the processor to store trace data that describes a hardware event, wherein the hardware event occurs during execution of the code at the processor.~~

~~17. The tracing system of claim 16, wherein providing the code comprises:~~

~~providing the code in response to compiling the code by the host, the host being coupled to the processor by a host interface and a node fabric of the tracing system, wherein the host uses each of the host interface and the node fabric to provide the compiled code to the processor.~~

~~18. The tracing system of claim 17, wherein compiling the code comprises:~~

~~embedding the operand in the code provided to the processor to enable performance analysis of the compiled code using a monitoring engine of the host that communicates with the host interface and the node fabric of the tracing system.~~

~~19. One or more non-transitory machine-readable storage devices storing instructions that are executable by one or more processing devices of a tracing system to cause performance of operations comprising:~~

~~providing code to a processor comprising a plurality of nodes;~~

~~—executing the code using each of the plurality of nodes of the processor;~~

~~—for a particular node of the plurality of nodes included in the processor:~~

~~—detecting that a trace bit is set to a first value that causes trace data to be generated at the processor using a counter of the tracing system;~~

~~generating the trace data in response to detecting that the trace bit is set to the first value, wherein the trace data is stored in the counter; and~~

~~—providing the trace data stored at the counter to a host for analysis at the host.~~

~~20. The non-transitory machine-readable storage devices of claim 19, wherein generating the trace data comprises:~~

~~generating data representing a trace entry to be provided to a trace chain corresponding to a data communication bus that couples two or more nodes of the plurality of nodes;~~

~~receiving, by a chip manager of the tracing system, the data representing the trace entry for storage in the chip manager; and~~

~~in response to receiving, storing the data representing the trace entry in the chip manager of the tracing system, wherein the chip manager is operable to store multiple trace entries for a plurality of hardware events that occur at the processor when the code is being executed.~~

1. A method performed using a tracing system, the method comprising:

providing code to a processor comprising a plurality of nodes;

executing the code using each of the plurality of nodes of the processor;

detecting, at each node of the plurality of nodes, that a trace bit is set to a first value that causes trace data to be generated at the processor;

generating, locally at each node, a respective portion of the trace data in response to detecting that the trace bit is set to the first value; and

providing each of the respective portions of trace data generated at the processor to a host for analysis at the host.

2. The method of claim 1, wherein at least one node of the plurality of nodes is a fixed-function component node that is operable to execute a specific operation and the method further comprises:

executing, using the fixed-function component node, the specific operation based on an instruction issued by the code; and

generating, based on the specific operation, a respective portion of trace data locally at the fixed-function component node in response to detecting that the trace bit is set to the first value.

3. The method of claim 2, wherein the fixed-function component node is a memory access engine and generating the respective portion of trace data comprises:

generating, based on the instruction issued by the code, a descriptor for a memory access operation to be performed using the memory access engine, wherein the descriptor includes the trace bit that is set to the first value to cause the trace data to be generated at the processor.

4. The method of claim 3, wherein:

the specific operation comprises performing the memory access operation using the memory access engine;

the memory access operation comprises multiple intermediate operations involving at least one other node of the plurality of nodes; and

performing the memory access operation causes trace data associated with an intermediate operation to be stored in a hardware trace buffer included at the other node.

5. The method of claim 1, wherein generating the trace data at the processor comprises:

generating data representing a trace entry to be provided to a trace chain corresponding to a data communication bus that couples two or more nodes of the plurality of nodes.

6. The method of claim 5, further comprising:

providing, using the trace chain, the data representing the trace entry to be stored in a chip manager of the tracing system; and

storing the data representing the trace entry in the chip manager, wherein the chip manager is operable to store multiple trace entries for a plurality of hardware events that occur at the processor during execution of the code.

7. The method of claim 1, further comprising:

embedding, by the host, an operand in the code provided to the processor to cause a node of the processor to store trace data that describes a hardware event, wherein the hardware event occurs during execution of the code at the processor.

8. The method of claim 7, wherein providing the code comprises:

providing the code in response to compiling the code by the host, the host being coupled to the processor by a host interface and a node fabric of the tracing system, wherein the host uses each of the host interface and the node fabric to provide the compiled code to the processor.

9. The method of claim 8, wherein compiling the code comprises:

embedding the operand in the code provided to the processor to enable performance analysis of the compiled code using a monitoring engine of the host that communicates with the host interface and the node fabric of the tracing system.

10. The method of claim 1, wherein the processor implements a neural network comprising a plurality of layers and the method comprises:

detecting, at each node of the plurality of nodes, that the trace bit is set to the first value that causes trace data to be generated at the processor during computations for a workload of the neural network.

11. The method of claim 10, wherein:

the processor is a multi-core processor;
at least one node of the plurality of nodes is a core of the multi-core processor; and
the method further comprises, storing, locally at the core, a portion of trace data
generated at the core during the computations for the workload of the neural network,
wherein the portion of trace data generated at the core is stored in a hardware trace
buffer included at the core.

12. A tracing system comprising:
one or more processing devices; and
one or more non-transitory machine-readable storage devices storing instructions that
are executable by the one or more processing devices to cause performance of operations
comprising:

providing code to a processor comprising a plurality of nodes;
executing the code using each of the plurality of nodes of the processor;
detecting, at each node of the plurality of nodes, that a trace bit is set to a first
value that causes trace data to be generated at the processor;
generating, locally at each node, a respective portion of the trace data in
response to detecting that the trace bit is set to the first value; and
providing each of the respective portions of trace data generated at the
processor to a host for analysis at the host.

13. The tracing system of claim 12, wherein at least one node of the plurality of
nodes is a fixed-function component node that is operable to execute a specific operation and
the operations further comprise:

executing, using the fixed-function component node, the specific operation based on
an instruction issued by the code; and
generating, based on the specific operation, a respective portion of trace data locally at
the fixed-function component node in response to detecting that the trace bit is set to the first
value.

14. The tracing system of claim 13, wherein the fixed-function component node is
a memory access engine and generating the respective portion of trace data comprises:

generating, based on the instruction issued by the code, a descriptor for a memory
access operation to be performed using the memory access engine, wherein the descriptor

includes the trace bit that is set to the first value to cause the trace data to be generated at the processor.

15. The tracing system of claim 14, wherein:
the specific operation comprises performing the memory access operation using the memory access engine;

the memory access operation comprises multiple intermediate operations involving at least one other node of the plurality of nodes; and

performing the memory access operation causes trace data associated with an intermediate operation to be stored in a hardware trace buffer included at the other node.

16. The tracing system of claim 12, wherein generating the trace data at the processor comprises:

generating data representing a trace entry to be provided to a trace chain corresponding to a data communication bus that couples two or more nodes of the plurality of nodes.

17. The tracing system of claim 16, wherein the operations comprise:
providing, using the trace chain, the data representing the trace entry to be stored in a chip manager of the tracing system; and

storing the data representing the trace entry in the chip manager, wherein the chip manager is operable to store multiple trace entries for a plurality of hardware events that occur at the processor during execution of the code.

18. The tracing system of claim 12, wherein the operations comprise:
embedding, by the host, an operand in the code provided to the processor to cause a node of the processor to store trace data that describes a hardware event, wherein the hardware event occurs during execution of the code at the processor.

19. The tracing system of claim 18, wherein providing the code comprises:
providing the code in response to compiling the code by the host, the host being coupled to the processor by a host interface and a node fabric of the tracing system, wherein the host uses each of the host interface and the node fabric to provide the compiled code to the processor.

20. The tracing system of claim 19, wherein compiling the code comprises:
embedding the operand in the code provided to the processor to enable performance
analysis of the compiled code using a monitoring engine of the host that communicates with
the host interface and the node fabric of the tracing system.

21. The tracing system of claim 12, wherein the processor implements a neural
network comprising a plurality of layers and the operations comprises:

detecting, at each node of the plurality of nodes, that the trace bit is set to the first
value that causes trace data to be generated at the processor during computations for a
workload of the neural network.

22. One or more non-transitory machine-readable storage devices storing
instructions that are executable by one or more processing devices of a tracing system to
cause performance of operations comprising:

providing code to a processor comprising a plurality of nodes;

executing the code using each of the plurality of nodes of the processor;

detecting, at each node of the plurality of nodes, that a trace bit is set to a first value
that causes trace data to be generated at the processor;

generating, locally at each node, a respective portion of the trace data in response to
detecting that the trace bit is set to the first value; and

providing each of the respective portions of trace data generated at the processor to a
host for analysis at the host.

23. The non-transitory machine-readable storage devices of claim 22, wherein
generating the trace data at the processor comprises:

generating data representing a trace entry to be provided to a trace chain
corresponding to a data communication bus that couples two or more nodes of the plurality of
nodes;

receiving, by a chip manager of the tracing system, the data representing the trace
entry for storage in the chip manager; and

in response to receiving, storing the data representing the trace entry in the chip
manager of the tracing system, wherein the chip manager is operable to store multiple trace

entries for a plurality of hardware events that occur at the processor during execution of the code.