codeplay

Enabling AI & HPC To Be Open Safe & Accessible To All

Executing Graphs with OpenCL

Erik Tomusk

IWOCL ‘21

26-29 April 2021

Products

Acoran

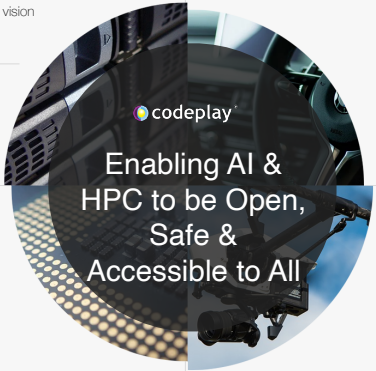
Integrates all the industry standard technologies needed to support a very wide range of AI and HPC

ComputeCpp

C++ platform via the SYCL™ open standard, enabling vision & machine learning e.g. TensorFlow™

ComputeAorta

The heart of Codeplay's compute technology enabling OpenCL™, SPIR-V™, HSA™ and Vulkan™



Markets

High Performance Compute (HPC)
Automotive ADAS, IoT, Cloud Compute
Smartphones & Tablets
Medical & Industrial

Technologies: Artificial Intelligence
Vision Processing
Machine Learning
Big Data Compute

Company

Leaders in enabling high-performance software solutions for new AI processing systems

Enabling the toughest processors with tools and middleware based on open standards

Established 2002 in Scotland with ~80 employees

Partners

BROADCOM
CEVA
Imagination
intel
KNC
NSI-TEXE
BERKELEY LAB
RENESAS
SYNOPSYS
Argonne NATIONAL LABORATORY

And many more!

© 2021 Codeplay Software Ltd.

About me

© 2021 Codeplay Software Ltd.

About me

- Software engineer

© 2021 Codeplay Software Ltd.

About me

- Software engineer
- ComputeAorta team — OpenCL runtime

© 2021 Codeplay Software Ltd.

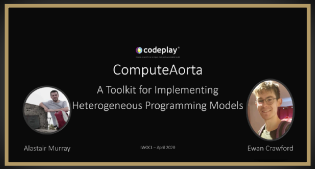
About me

- Software engineer
- ComputeAorta team — OpenCL runtime
  - www.youtube.com/watch?v=enyvwRWJ7PA

© 2021 Codeplay Software Ltd.

About me

- Software engineer
- ComputeAorta team — OpenCL runtime
  - www.youtube.com/watch?v=enyvwRWJ7PA



Thanks:
Romain Biessy
Mehdi Goli
Victor Lomüller
Andrew Richards
ComputeAorta team
… and many others at Codeplay

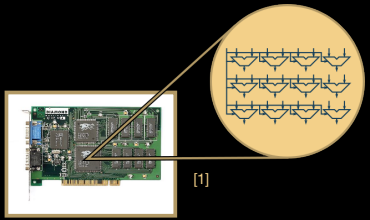
© 2021 Codeplay Software Ltd.

Motivation

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos
[1] https://en.wikipedia.org/wiki/File:3Dfx_Diamond_Monster3D_Voodoo_1.jpg
[2] https://www.youtube.com/watch?v=31C4HmTXBHg
[3] https://www.youtube.com/watch?v=w34dwqDEyv8
[4] https://www.youtube.com/watch?v=ME-BuxSoVm0

© 2021 Codeplay Software Ltd.

Motivation



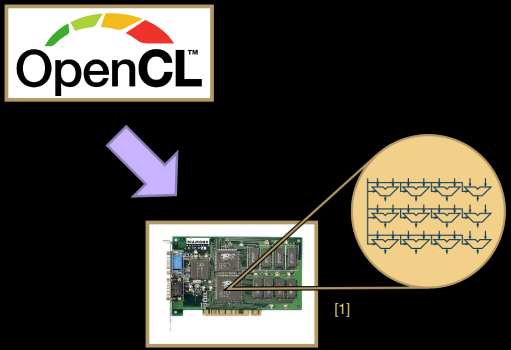
[1]

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

[1] https://en.wikipedia.org/wiki/File:3Dfx_Diamond_MonsterII_Voodoo_1.jpg
[2] https://www.youtube.com/watch?v=31C4HmTXBIg
[3] https://www.youtube.com/watch?v=w34dwqDEyI8
[4] https://www.youtube.com/watch?v=ME-8uxSqVmQ

© 2021 Codeplay Software Ltd.

Motivation

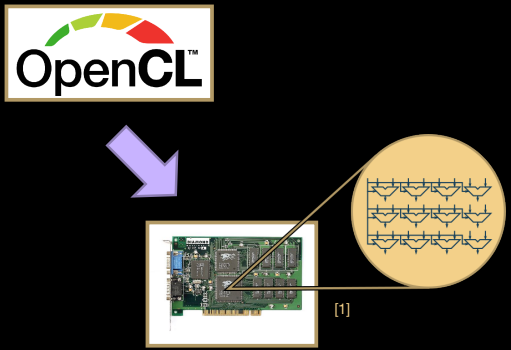


OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

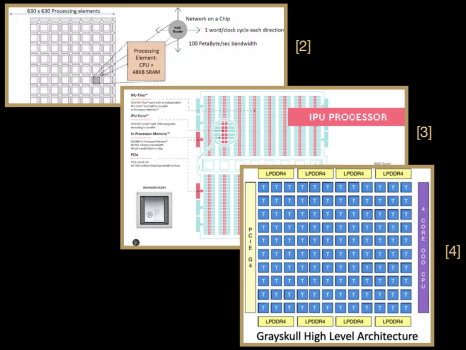
[1] https://en.wikipedia.org/wiki/File:3dfx_Diamond_MonsterSD_Voodoo_1.jpg
[2] https://www.youtube.com/watch?v=31C4HmTXBIg
[3] https://www.youtube.com/watch?v=w34dwqDEyv8
[4] https://www.youtube.com/watch?v=ME-8ux5pVmQ

© 2021 Codeplay Software Ltd.

Motivation



[1]



[2]

[3]

[4]

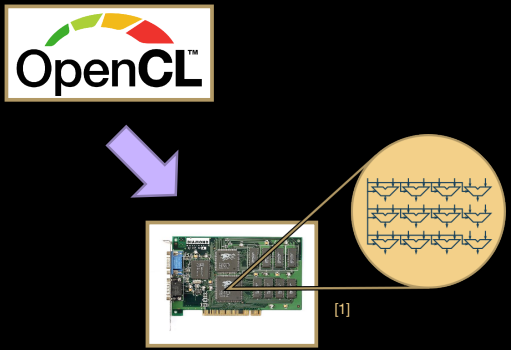
OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

[1] https://en.wikipedia.org/wiki/File:3dl_Diamond_MonsterSD_Voodoo_1.jpg
[2] https://www.youtube.com/watch?v=31C4HmTXBIg
[3] https://www.youtube.com/watch?v=w34dwqDEy8
[4] https://www.youtube.com/watch?v=ME-8wxSqVmg

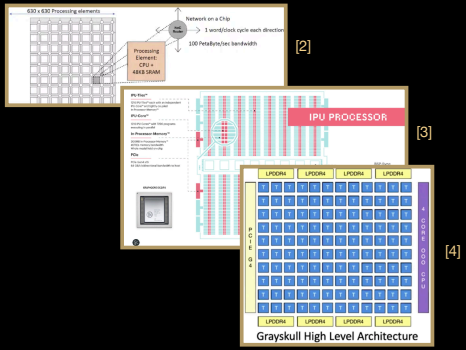
© 2021 Codeplay Software Ltd.

Motivation

?



[1]



[2] [3] [4]

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos

[1] https://en.wikipedia.org/wiki/File:3d_Diamond_MonsterSD_Voodoo_1.jpg
[2] https://www.youtube.com/watch?v=31C4HmTXBIg
[3] https://www.youtube.com/watch?v=w34dwqDEy8
[4] https://www.youtube.com/watch?v=ME-8wxSqVmQ

© 2021 Codeplay Software Ltd.

Motivation

© 2021 Codeplay Software Ltd.

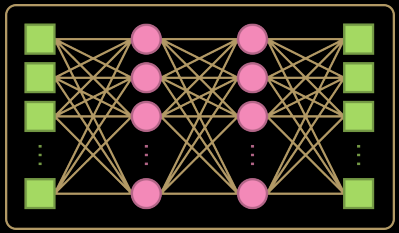
Motivation

- Machine learning requires huge amounts of compute resources

© 2021 Codeplay Software Ltd.

Motivation

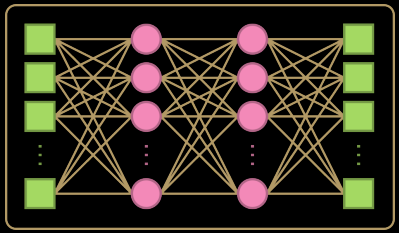
- Machine learning requires huge amounts of compute resources
- Machine learning problems are shaped like graphs



© 2021 Codeplay Software Ltd.

Motivation

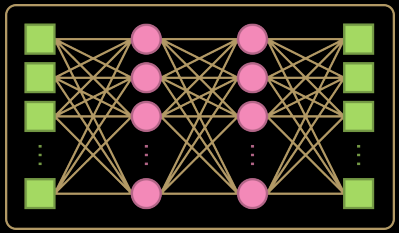
- Machine learning requires huge amounts of compute resources
- Machine learning problems are shaped like graphs
- Large accelerators are being designed to execute graphs



© 2021 Codeplay Software Ltd.

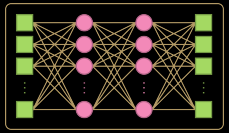
Motivation

- Machine learning requires huge amounts of compute resources
- Machine learning problems are shaped like graphs
- Large accelerators are being designed to execute graphs
- Where does OpenCL fit in?



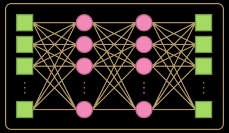
© 2021 Codeplay Software Ltd.

Motivation



© 2021 Codeplay Software Ltd.

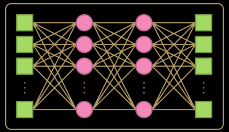
Motivation

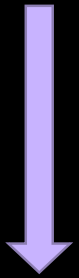


Accelerator
Device

© 2021 Codeplay Software Ltd.

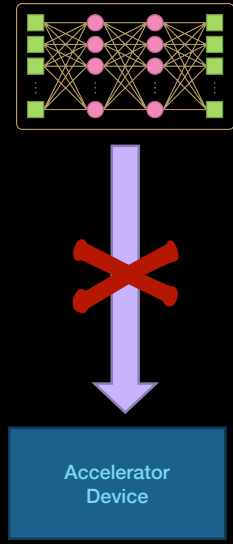
Motivation





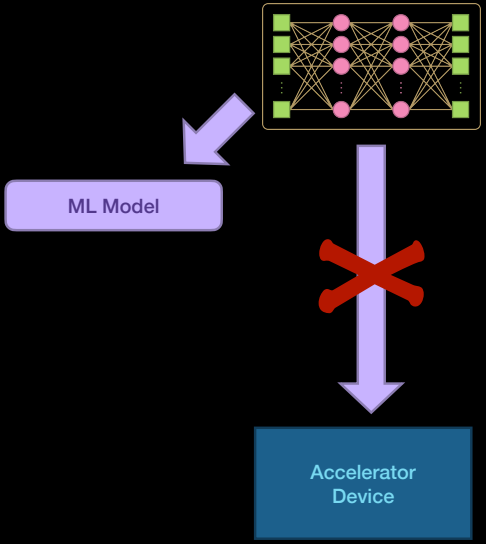
© 2021 Codeplay Software Ltd.

Motivation



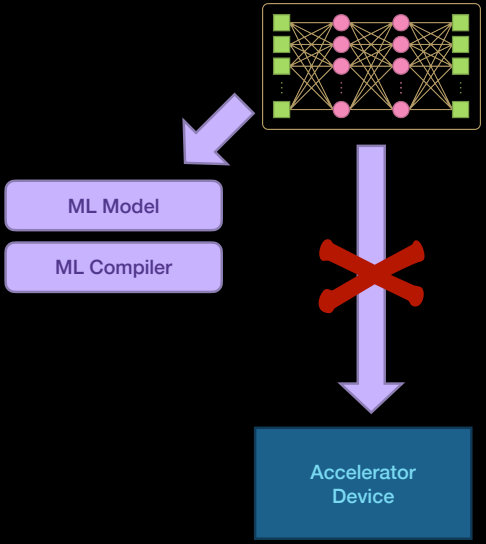
© 2021 Codeplay Software Ltd.

Motivation



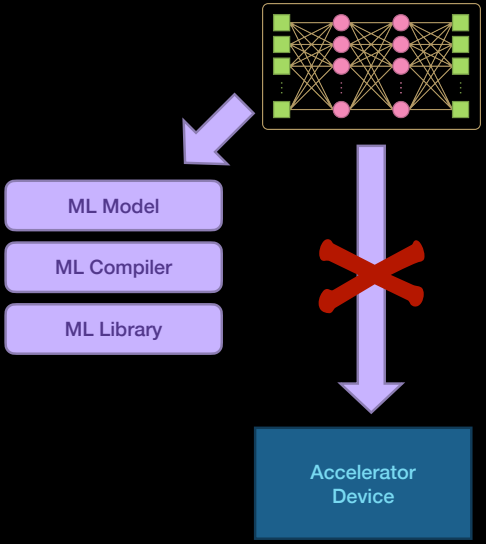
© 2021 Codeplay Software Ltd.

Motivation



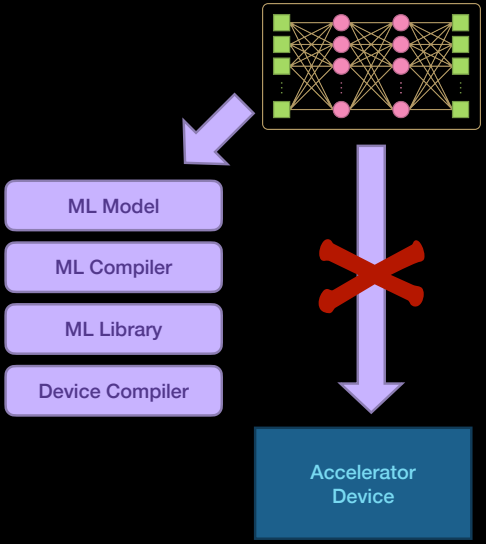
© 2021 Codeplay Software Ltd.

Motivation



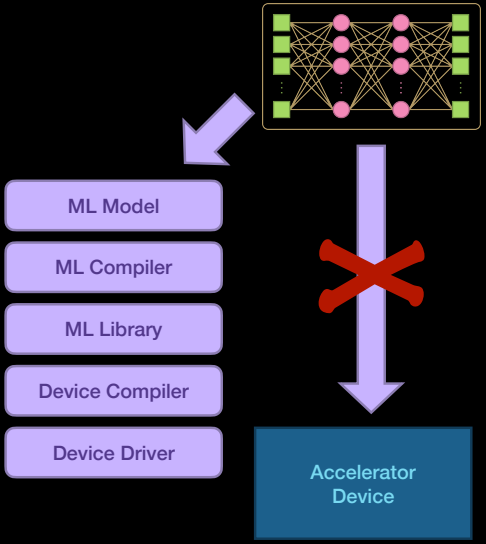
© 2021 Codeplay Software Ltd.

Motivation



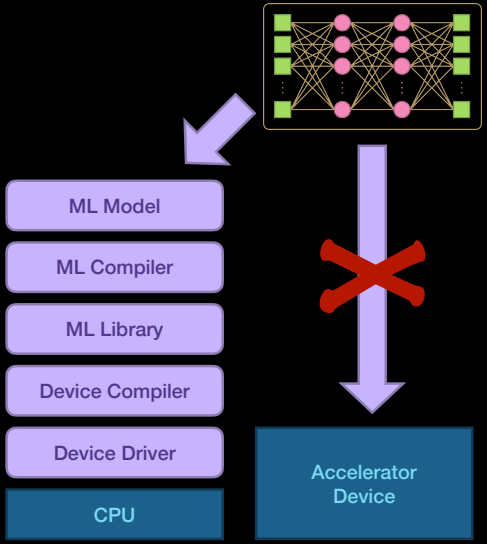
© 2021 Codeplay Software Ltd.

Motivation



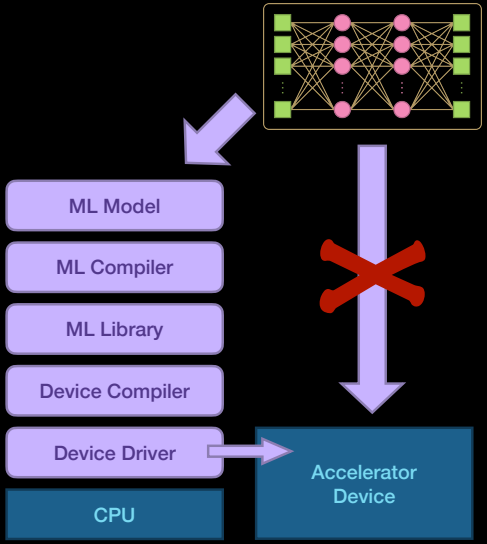
© 2021 Codeplay Software Ltd.

Motivation



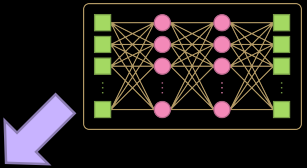
© 2021 Codeplay Software Ltd.

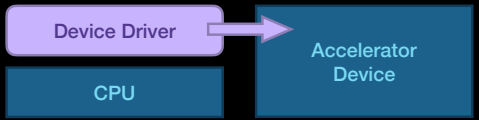
Motivation



© 2021 Codeplay Software Ltd.

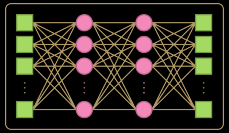
Motivation



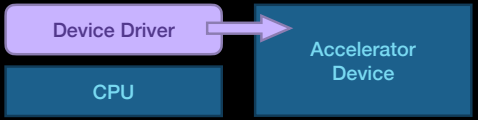


© 2021 Codeplay Software Ltd.

Motivation

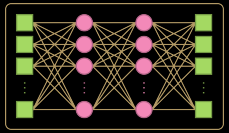


Python Interface



© 2021 Codeplay Software Ltd.

Motivation



Python Interface

TensorFlow

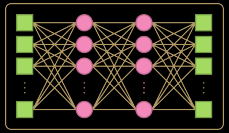
Device Driver

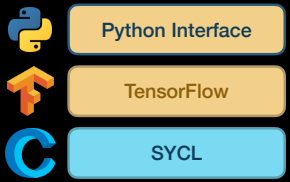
CPU

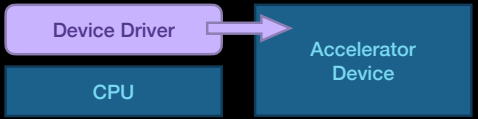
Accelerator Device

© 2021 Codeplay Software Ltd.

Motivation

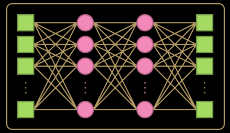


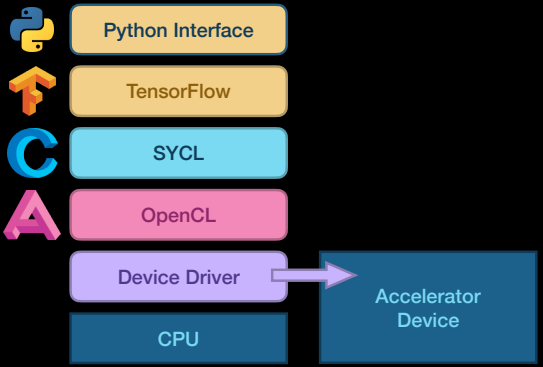




© 2021 Codeplay Software Ltd.

Motivation

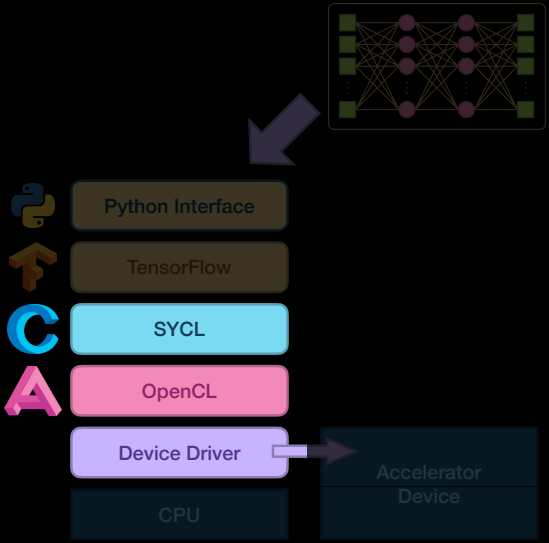




© 2021 Codeplay Software Ltd.

Motivation

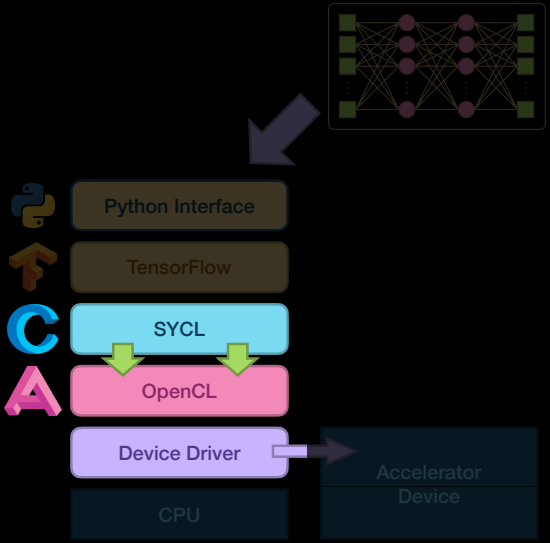
- Can the OpenCL layer “see” the execution graph
  - … and explain it to the device driver?



© 2021 Codeplay Software Ltd.

Motivation

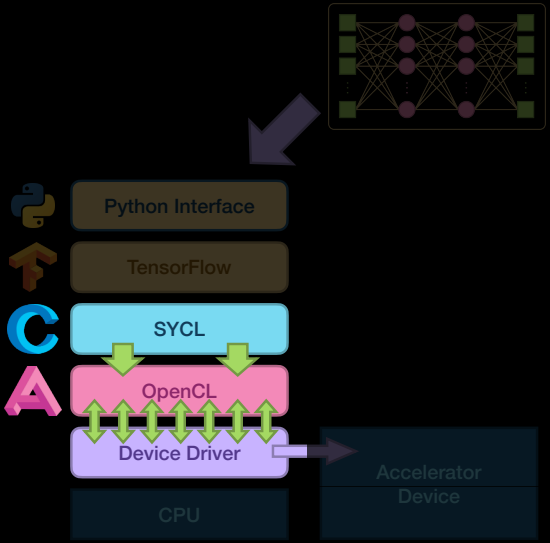
- Can the OpenCL layer “see” the execution graph
  - … and explain it to the device driver?
- Execution graph comes down from SYCL



© 2021 Codeplay Software Ltd.

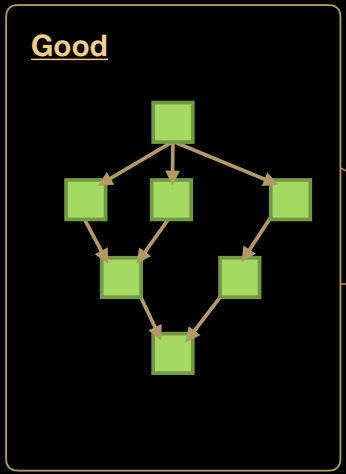
Motivation

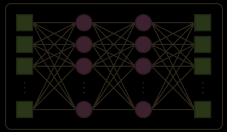
- Can the OpenCL layer “see” the execution graph
  - … and explain it to the device driver?
- Execution graph comes down from SYCL
- The OpenCL implementation and the device driver are tightly integrated

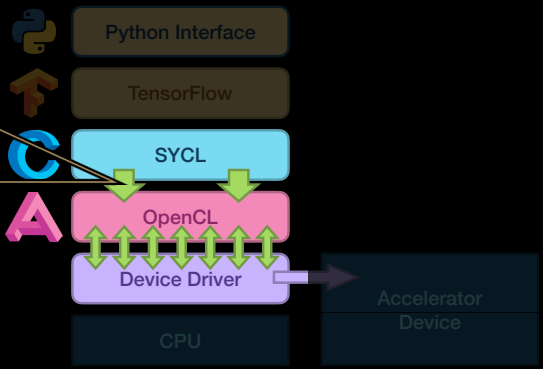


© 2021 Codeplay Software Ltd.

Motivation

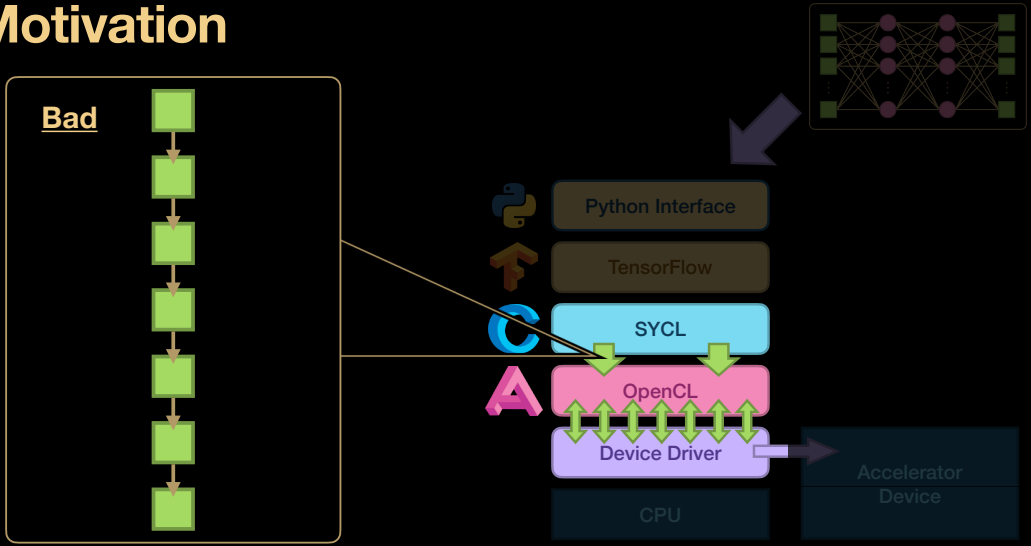






© 2021 Codeplay Software Ltd.

Motivation



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- What does the OpenCL API “see”?

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- What does the OpenCL API “see”?
- Kernels (functions) execute code

© 2021 Codeplay Software Ltd.

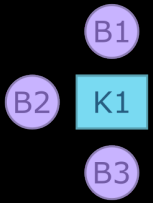
Graphs in OpenCL

- What does the OpenCL API “see”?
- Kernels (functions) execute code

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

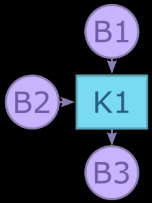
- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

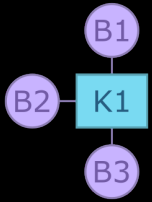
- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data
  - Input, output, or inout



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

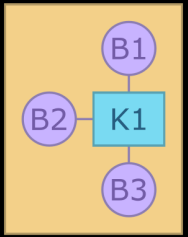
- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data
  - Input, output, or inout



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

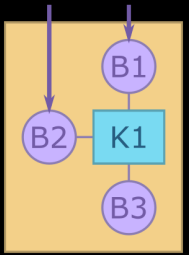
- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data
  - Input, output, or inout
- Kernels are enqueued for execution



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

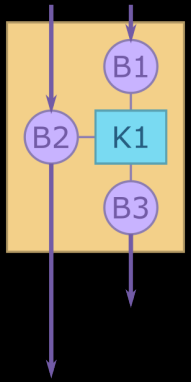
- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data
  - Input, output, or inout
- Kernels are enqueued for execution
- Some buffers written previously
- Some buffers are new



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

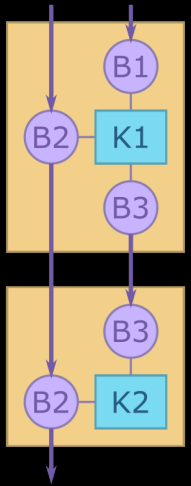
- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data
  - Input, output, or inout
- Kernels are enqueued for execution
- Some buffers written previously
- Some buffers are new
- Some buffers are reused, others aren’t



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

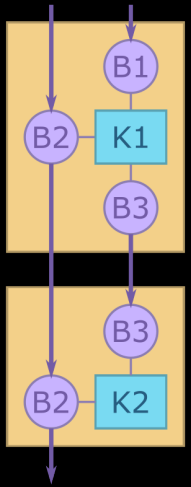
- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data
  - Input, output, or inout
- Kernels are enqueued for execution
- Some buffers written previously
- Some buffers are new
- Some buffers are reused, others aren’t



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- What does the OpenCL API “see”?
- Kernels (functions) execute code
- Buffers store data
  - Input, output, or inout
- Kernels are enqueued for execution
- Some buffers written previously
- Some buffers are new
- Some buffers are reused, others aren’t
- *Buffer usage implies dependencies between kernels*



© 2021 Codeplay Software Ltd.

Graphs in OpenCL

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Commands are enqueued onto command-queues

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Commands are enqueued onto command-queues
- Command-queues enforce happens-before dependencies

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Commands are enqueued onto command-queues
- Command-queues enforce happens-before dependencies
- ***In-order*** queues infer dependencies from enqueue order

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Commands are enqueued onto command-queues
- Command-queues enforce happens-before dependencies
- ***In-order*** queues infer dependencies from enqueue order
- ***Out-of-order*** queues require explicit event wait lists from OpenCL user

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Commands are enqueued onto command-queues
- Command-queues enforce happens-before dependencies

> *For an in-order command queue, the kernel instances appear to launch and then execute in the same order; where we use the term appear to emphasize that when there are no dependencies between commands and hence differences in the order that commands execute cannot be observed in a program,*
> *an implementation can reorder commands even in an in-order command queue.*
>
> — The OpenCL™ Specification v3.0.6 §3.2.2 (emphasis added)

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Commands are enqueued onto command-queues
- Command-queues enforce happens-before dependencies
- ***In-order*** queues infer dependencies from enqueue order
- ***Out-of-order*** queues require explicit event wait lists from OpenCL user

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Commands are enqueued onto command-queues
- Command-queues enforce happens-before dependencies
- ***In-order*** queues infer dependencies from enqueue order
- ***Out-of-order*** queues require explicit event wait lists from OpenCL user
- Focus on in-order queues for the rest of the talk

© 2021 Codeplay Software Ltd.

Empirical Study

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library
- TensorFlow using SYCL backend [1]

[1] https://github.com/codeplaysoftware/tensorflow

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library
- TensorFlow using SYCL backend [1]
  - Codeplay's ComputeCpp SYCL implementation

[1] https://github.com/codeplaysoftware/tensorflow

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library
- TensorFlow using SYCL backend [1]
  - Codeplay's ComputeCpp SYCL implementation
- SYCL using OpenCL backend

[1] https://github.com/codeplaysoftware/tensorflow

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library
- TensorFlow using SYCL backend [1]
  - Codeplay's ComputeCpp SYCL implementation
- SYCL using OpenCL backend
  - Codeplay's ComputeAorta OpenCL

[1] https://github.com/codeplaysoftware/tensorflow

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library
- TensorFlow using SYCL backend [1]
  - Codeplay's ComputeCpp SYCL implementation
- SYCL using OpenCL backend
  - Codeplay's ComputeAorta OpenCL
  - Instrumented to record OpenCL API calls

[1] https://github.com/codeplaysoftware/tensorflow

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library
- TensorFlow using SYCL backend [1]
  - Codeplay's ComputeCpp SYCL implementation
- SYCL using OpenCL backend
  - Codeplay's ComputeAorta OpenCL
  - Instrumented to record OpenCL API calls
- Trace training and inference stages separately

[1] https://github.com/codeplaysoftware/tensorflow

© 2021 Codeplay Software Ltd.

Empirical Study

- Train and use a simplified handwriting detection neural network
  - TensorFlow 1.9.0
  - Standard MNIST example
  - Simplified neural network so graphs fit on slides
  - Pre-Keras Python library
- TensorFlow using SYCL backend [1]
  - Codeplay's ComputeCpp SYCL implementation
- SYCL using OpenCL backend
  - Codeplay's ComputeAorta OpenCL
  - Instrumented to record OpenCL API calls
- Trace training and inference stages separately
- Use hacky scripts to convert traces into graphs

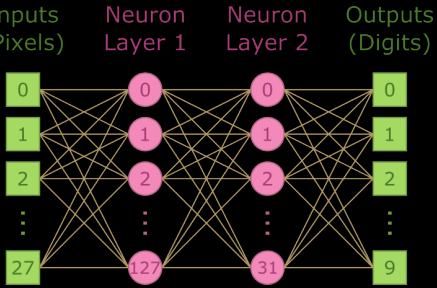
[1] https://github.com/codeplaysoftware/tensorflow

© 2021 Codeplay Software Ltd.

Empirical Study

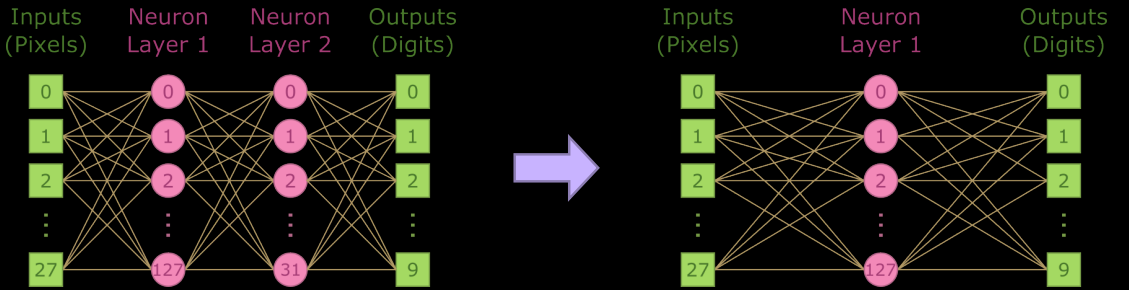
© 2021 Codeplay Software Ltd.

Empirical Study



© 2021 Codeplay Software Ltd.

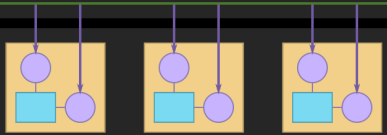
Empirical Study



© 2021 Codeplay Software Ltd.

MNIST Inference Execution Graph

Data In

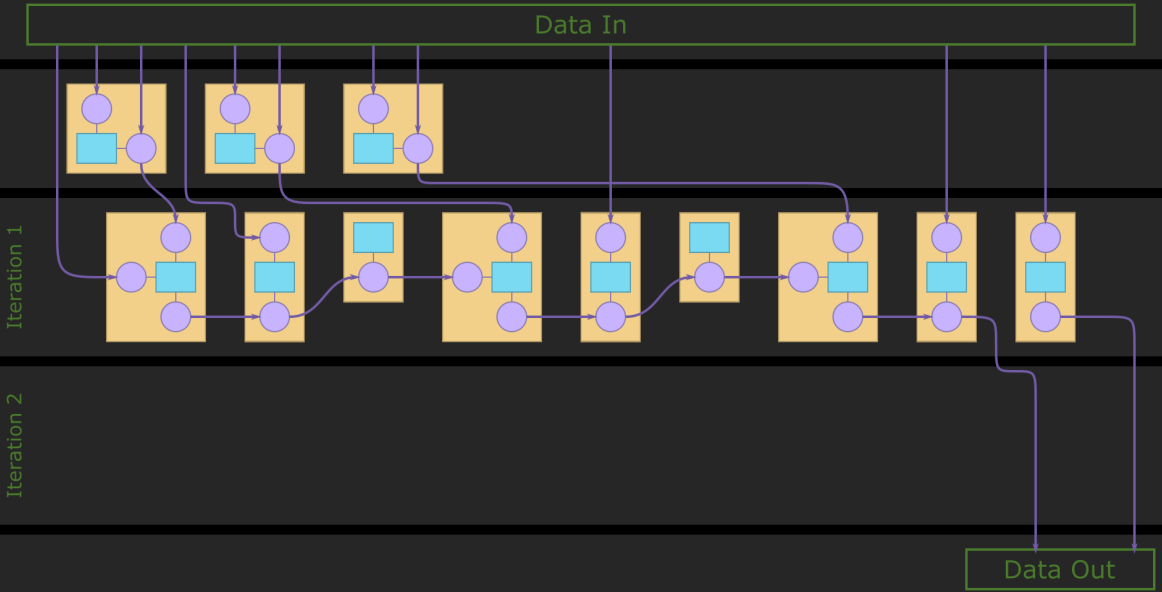


Iteration 1

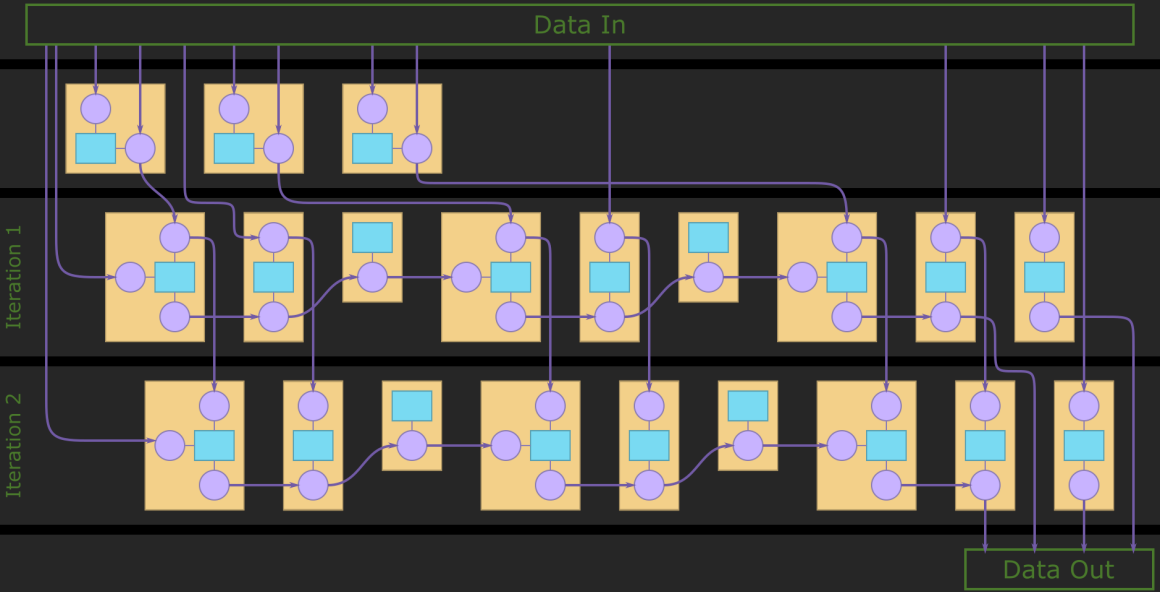
Iteration 2

Data Out

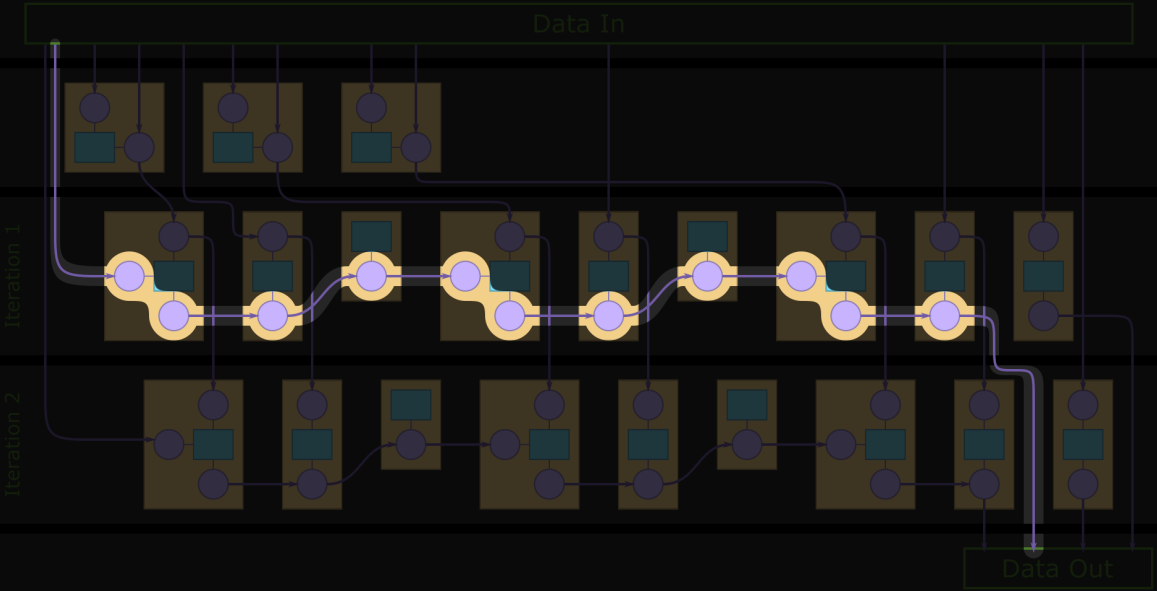
© 2021 Codeplay Software Ltd.



© 2021 Codeplay Software Ltd.

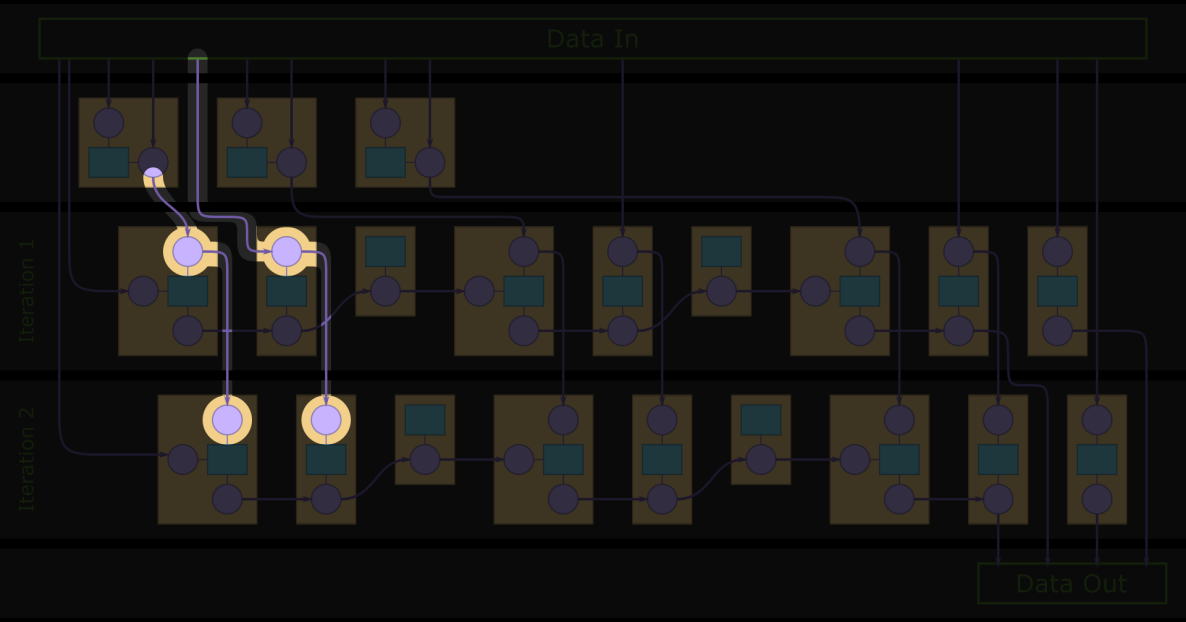


© 2021 Codeplay Software Ltd.

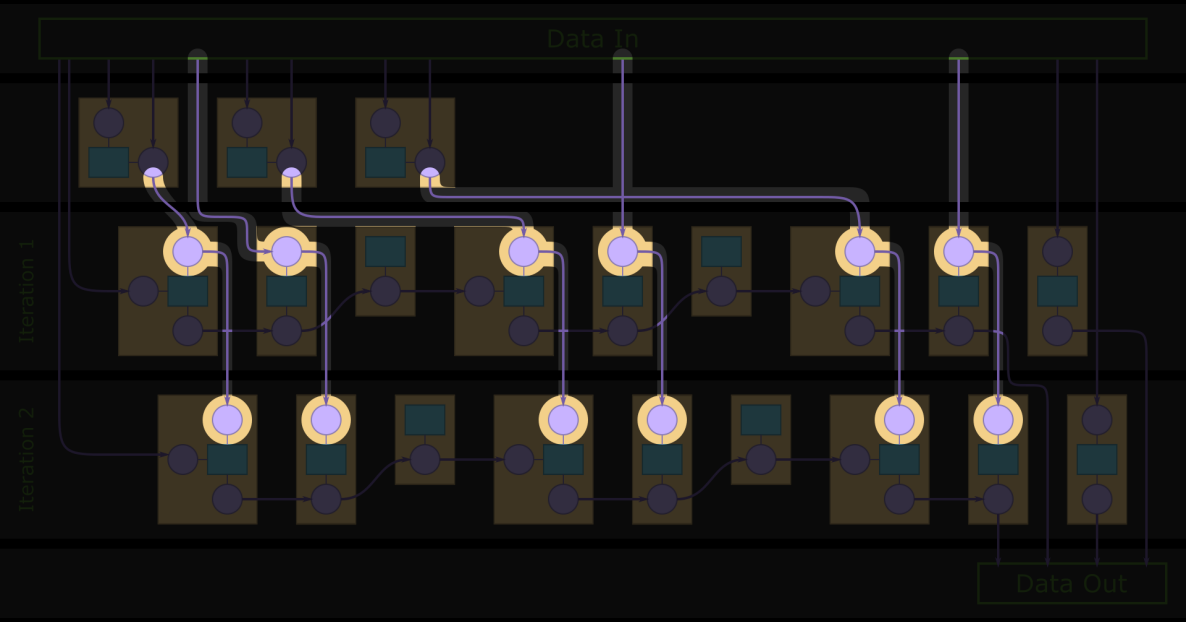


© 2021 Codeplay Software Ltd.

© 2021 Codeplay Software Ltd.



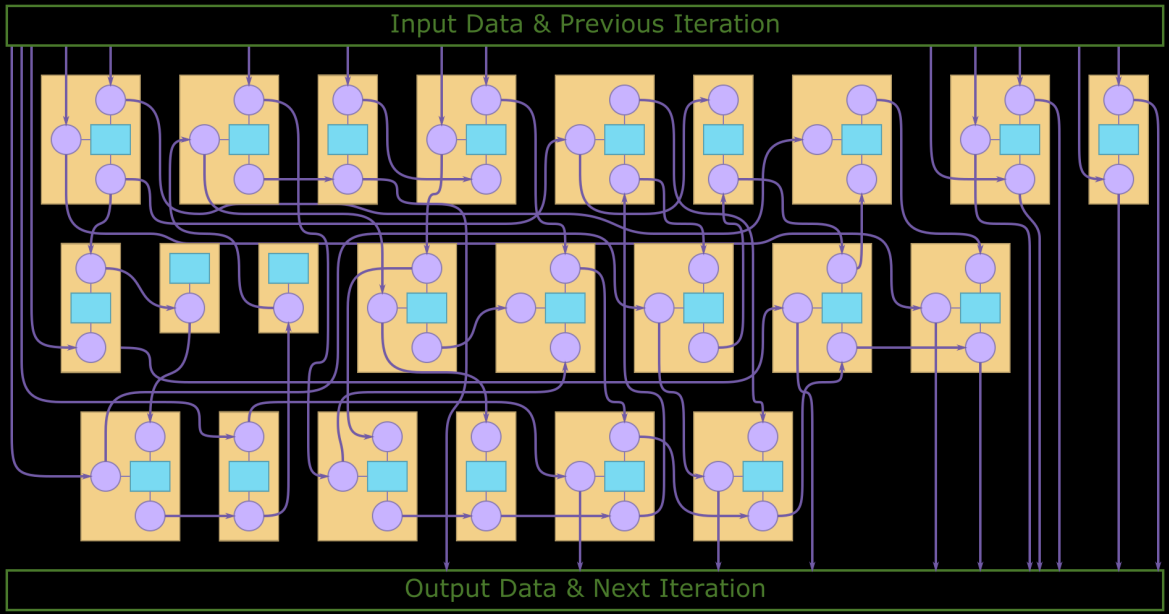
© 2021 Codeplay Software Ltd.



© 2021 Codeplay Software Ltd.

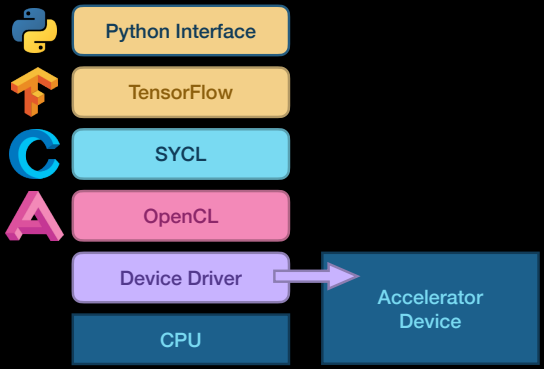
© 2021 Codeplay Software Ltd.

MNIST Training Execution Graph



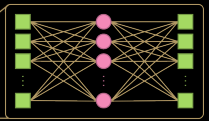
© 2021 Codeplay Software Ltd.

Graphs in OpenCL



© 2021 Codeplay Software Ltd.

Graphs in OpenCL



Python Interface

TensorFlow

SYCL

OpenCL

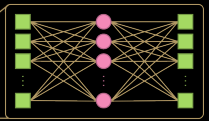
Device Driver

CPU

Accelerator Device

© 2021 Codeplay Software Ltd.

Graphs in OpenCL



Python Interface

TensorFlow

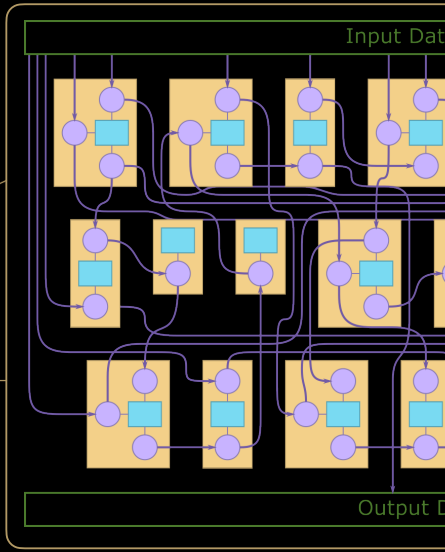
SYCL

OpenCL

Device Driver

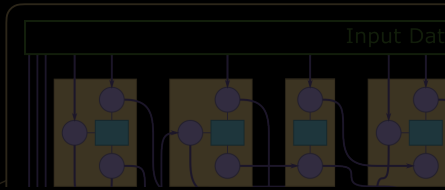
CPU

Accelerator Device

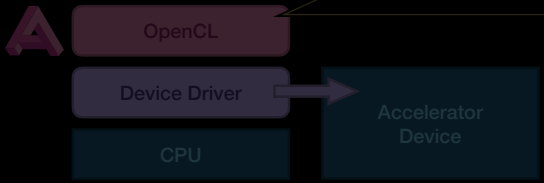


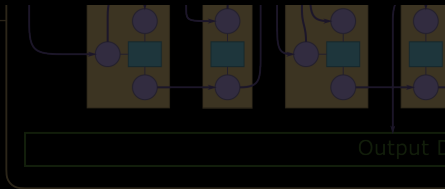
© 2021 Codeplay Software Ltd.

Graphs in OpenCL



Execution graph is visible to OpenCL





© 2021 Codeplay Software Ltd.

Graphs in OpenCL

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer
  - Deep neural networks can have hundreds of layers

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer
  - Deep neural networks can have hundreds of layers
- This particular API stack hides some producer/consumer information from OpenCL

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer
  - Deep neural networks can have hundreds of layers
- This particular API stack hides some producer/consumer information from OpenCL
  - Could be improved with:

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer
  - Deep neural networks can have hundreds of layers
- This particular API stack hides some producer/consumer information from OpenCL
  - Could be improved with:
    - Explicitly declaring read-only and write-only buffers

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer
  - Deep neural networks can have hundreds of layers
- This particular API stack hides some producer/consumer information from OpenCL
  - Could be improved with:
    - Explicitly declaring read-only and write-only buffers
    - Using the `__constant` address space

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer
  - Deep neural networks can have hundreds of layers
- This particular API stack hides some producer/consumer information from OpenCL
  - Could be improved with:
    - Explicitly declaring read-only and write-only buffers
    - Using the `__constant` address space
    - Using out-of-order command-queues

© 2021 Codeplay Software Ltd.

Graphs in OpenCL

- Execution graph is visible to OpenCL
  - OpenCL implementation can work with the device driver to execute the graph
- Diagram showed one iteration of training a simplified MNIST example
  - This model takes a few thousand iterations to train
  - Vulkan-style command buffers could reduce the number of times the graph must be constructed
- Simplified MNIST model has one layer
  - Deep neural networks can have hundreds of layers
- This particular API stack hides some producer/consumer information from OpenCL
  - Could be improved with:
    - Explicitly declaring read-only and write-only buffers
    - Using the `__constant` address space
    - Using out-of-order command-queues
    - Using device-specific OpenCL extensions

© 2021 Codeplay Software Ltd.

Conclusion

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators
- OpenCL by itself does not guarantee that an accelerator's device driver will see a graph

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators
- OpenCL by itself does not guarantee that an accelerator's device driver will see a graph
  - Many ways to get it wrong

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators
- OpenCL by itself does not guarantee that an accelerator's device driver will see a graph
  - Many ways to get it wrong
  - In the API stack

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators
- OpenCL by itself does not guarantee that an accelerator's device driver will see a graph
  - Many ways to get it wrong
  - In the API stack
  - In the OpenCL implementation

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators
- OpenCL by itself does not guarantee that an accelerator's device driver will see a graph
  - Many ways to get it wrong
  - In the API stack
  - In the OpenCL implementation
- Most users don't write OpenCL code, but use an API with an OpenCL backend

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators
- OpenCL by itself does not guarantee that an accelerator's device driver will see a graph
  - Many ways to get it wrong
  - In the API stack
  - In the OpenCL implementation
- Most users don't write OpenCL code, but use an API with an OpenCL backend
  - API developer's responsibility to ensure API communicates user's graph to OpenCL

© 2021 Codeplay Software Ltd.

Conclusion

- OpenCL can be used to represent graphs to graph accelerators
- OpenCL by itself does not guarantee that an accelerator's device driver will see a graph
  - Many ways to get it wrong
  - In the API stack
  - In the OpenCL implementation
- Most users don't write OpenCL code, but use an API with an OpenCL backend
  - API developer's responsibility to ensure API communicates user's graph to OpenCL
  - OpenCL implementor's responsibility to communicate graph to device driver

© 2021 Codeplay Software Ltd.

We're Hiring!

codeplay.com/careers/

codeplay

Enabling AI & HPC To Be Open Safe & Accessible To All

erik@codeplay.com

@codeplaysoft

info@codeplay.com

codeplay.com