PlaidML

Portable Deep Learning Compiler

Seo Sanghyeon

sanxiyn@gmail.com

PlaidML

PlaidML is an advanced and **portable** tensor **compiler for** enabling **deep learning** on laptops, embedded devices, or other devices where the available computing hardware is not well supported or the available software stack contains unpalatable license restrictions.

-- https://plaidml.github.io/plaidml/

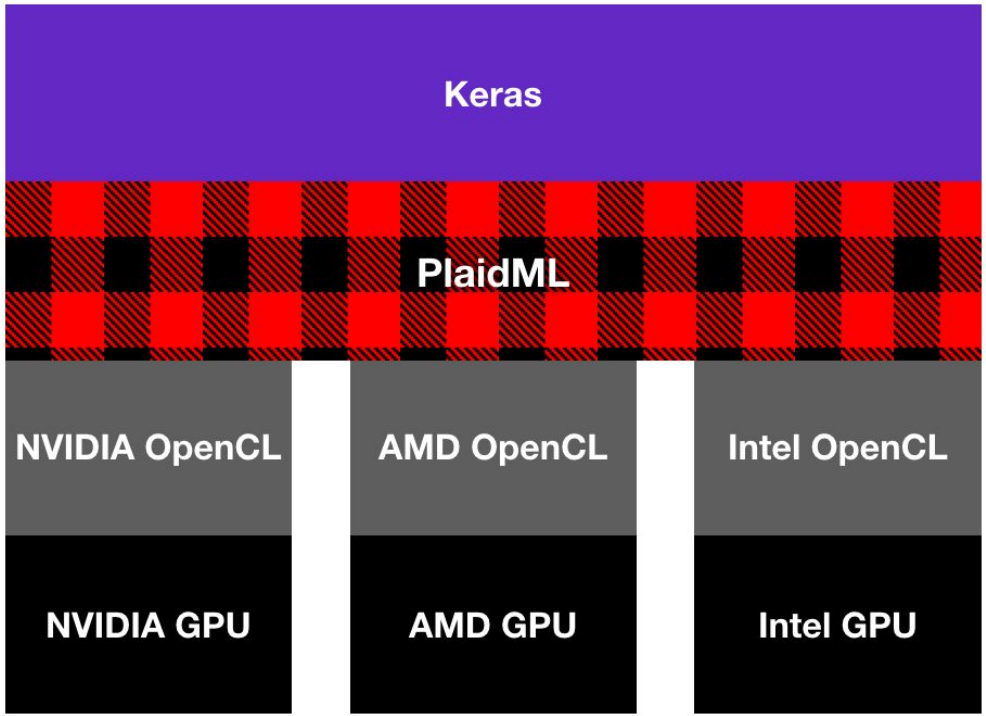
History

2017-10-20: Announcing PlaidML: Open Source Deep Learning for Every Platform
http://web.archive.org/web/20171021200126/vertex.ai/blog/announcing-plaidml

2018-08-16: Intel buys deep-learning startup Vertex.AI to join its Movidius unit
https://techcrunch.com/2018/08/16/intel-buys-deep-learning-startup-vertex-ai-to-join-its-movidius-unit/

2019-03-14: Stripe: Tensor Compilation via the Nested Polyhedral Model
https://arxiv.org/abs/1903.06498

PlaidML (2017)



Problems (Keras)

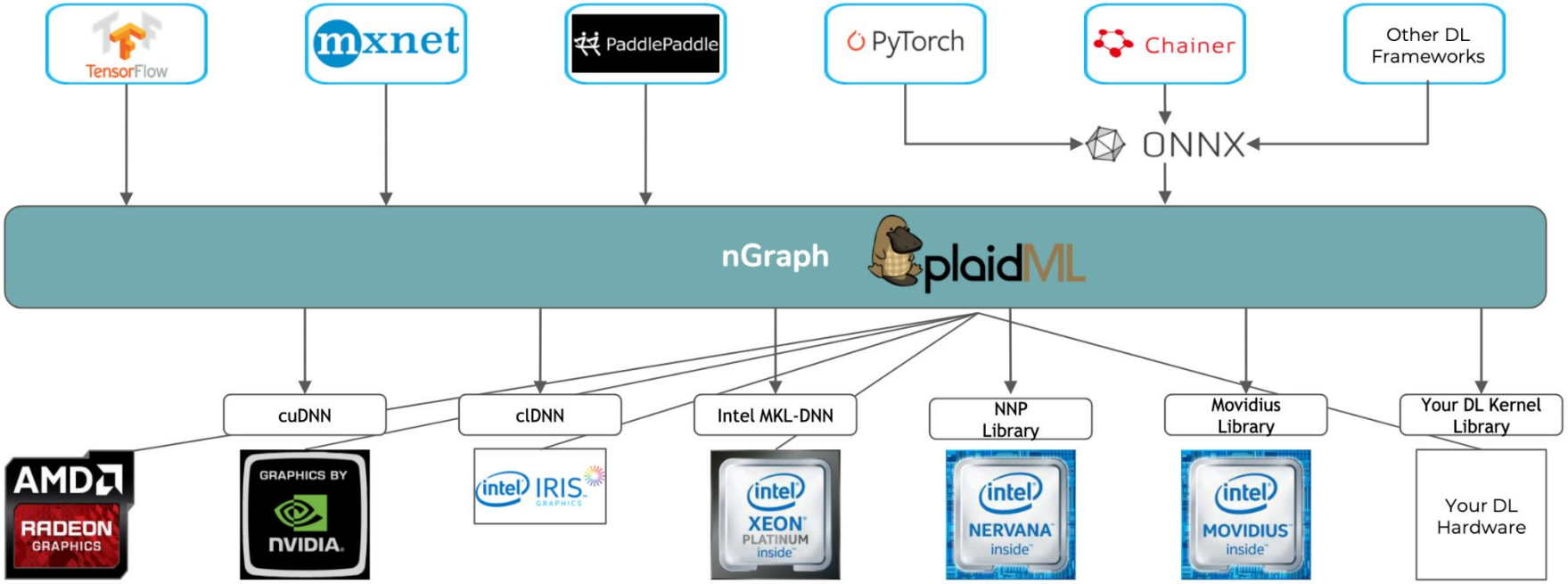
- Keras was the wrong choice
- It was an obvious choice, because it supported multiple backends: TensorFlow, Theano, and CNTK
- Keras stopped supporting multiple backends in 2.3.0 (2019)
- TensorFlow and PyTorch won

Problems (OpenCL)

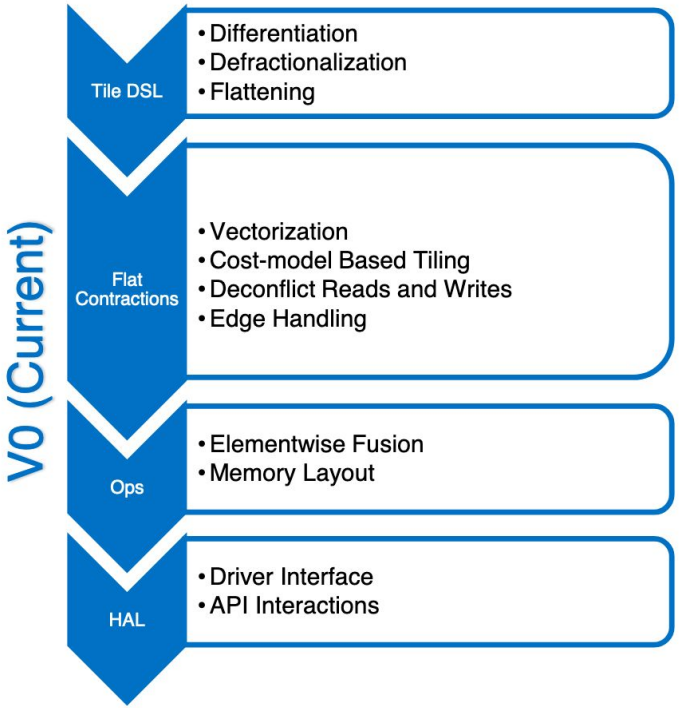
- OpenCL was the wrong choice
- It was an obvious choice, because it supported multiple hardwares
- OpenCL was developed by Apple, but Apple switched to Metal
- OpenCL compilers were of low quality
  See: Many-Core Compiler Fuzzing (PLDI 2015)
- OpenCL suffers from poor performance portability
  See: Machine Learning Based Auto-tuning for Enhanced OpenCL Performance Portability
- It probably makes more sense to target SPIR-V now

PlaidML (2018)

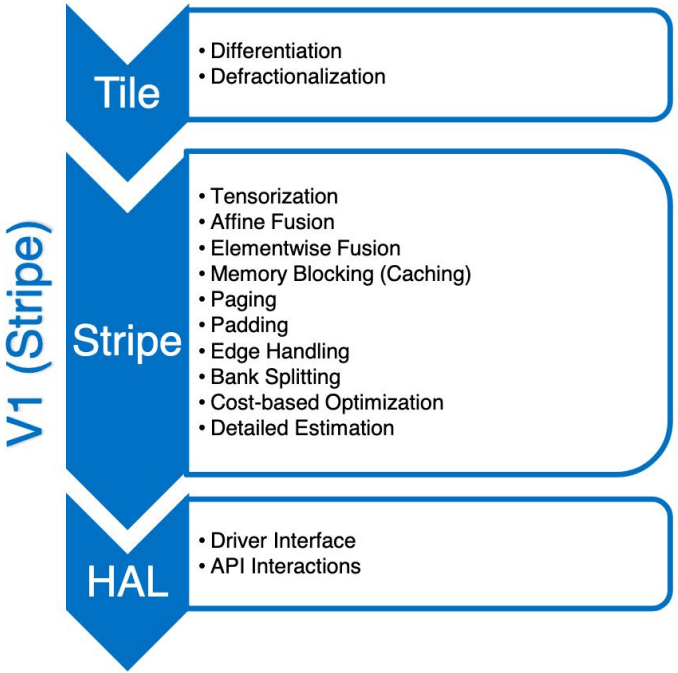
IEEE Silicon Valley Machine Learning Compiler workshop (Fall 2018)
https://r6.ieee.org/scv-cis/event/fall2018_workshop/



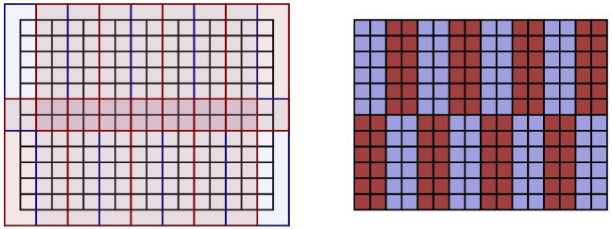
PlaidML V0


V0 (Current)
Tile DSL
• Differentiation
• Defractionalization
• Flattening
Flat Contractions
• Vectorization
• Cost-model Based Tiling
• Deconflict Reads and Writes
• Edge Handling
Ops
• Elementwise Fusion
• Memory Layout
HAL
• Driver Interface
• API Interactions

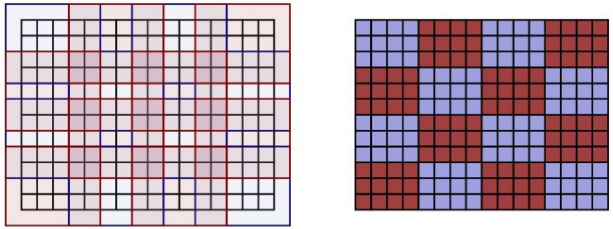
PlaidML V1


V1 (Stripe)
Tile
• Differentiation
• Defractionalization
Stripe
• Tensorization
• Affine Fusion
• Elementwise Fusion
• Memory Blocking (Caching)
• Paging
• Padding
• Edge Handling
• Bank Splitting
• Cost-based Optimization
• Detailed Estimation
HAL
• Driver Interface
• API Interactions

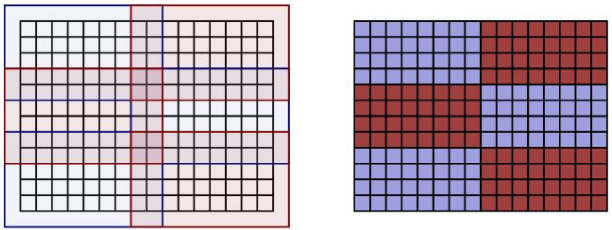
PlaidML (2019)



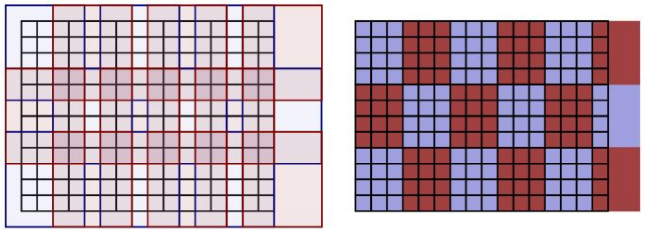
(a) Cost: 4.666



(b) Cost: 4.5



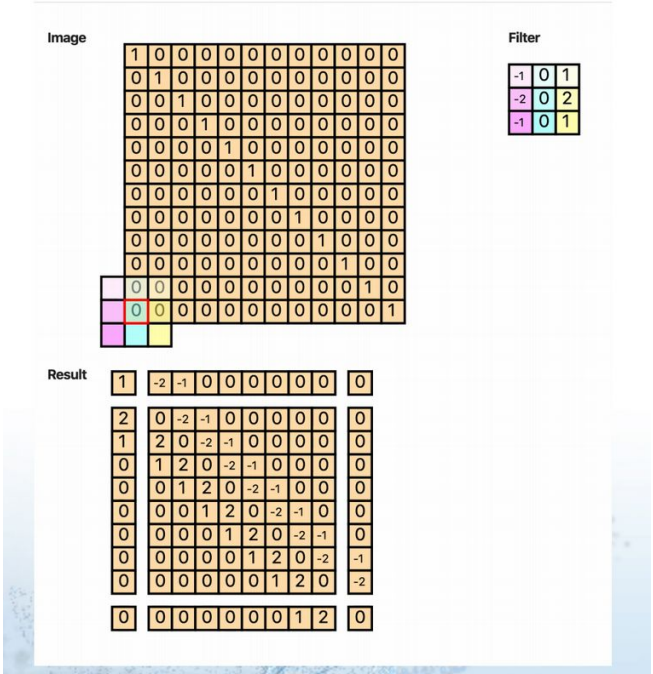
(c) Excluded from search space for requiring too many elements in memory for a single tile.



(d) Cost: 5.625.

PlaidML (2020)

Compilers for Machine Learning (C4ML) workshop at CGO 2020
https://www.c4ml.org/c4ml2020



Jigsaw Pass

https://github.com/plaidml/plaidml/blob/master/pmlc/dialect/stripe/jigsaw_pass.cc

References

nGraph + PlaidML
Using PlaidML for Affine Parallel Optimizations in MLIR