High-Performance Deep-Learning Operators on NVIDIA GPUs via Multi-Dimensional Homomorphisms

a.rasch@wwu.de

Ari Rasch, Richard Schulze, Sergei Gorlatch

WWU MÜNSTER

Generation

Let T and T' be two arbitrary types. A function $h : T[N_1]\ldots[N_d] \to T'$ on d-dimensional arrays is called a *Multi-Dimensional Homomorphism (MDH)* iff there exist *combine operators* $\circledast_1, \ldots, \circledast_d : T' \times T' \to T'$, such that for each $k \in [1,d]$ and arbitrary, concatenated input array $a \mathbin{+\!\!+}_k b$ in dimension k:

$$h(\, a \mathbin{+\!\!+}_k b \,) \;=\; h(a) \circledast_k h(b)$$

[IJPP'18]

MDHs can be uniformly represented via our `md_hom` parallel pattern:

$$\texttt{md\_hom}(\, f,\ (\circledast_1, \ldots, \circledast_d)\,)(\,a\,) \;=\; \underset{i_1 \in [1,N_1]}{\circledast_1} \;\ldots\; \underset{i_d \in [1,N_d]}{\circledast_d} \; f(\,a[\,i_1\,]\ldots[\,i_d\,]\,)$$

<u>Important Deep-Learning Operators can be expressed as MDHs:</u>

Linear Algebra (BLAS)

```
GEMM = md_hom( *, (++, ++, +) ) o view(A,B)(i,j,k)(A[i,k],B[k,j])
GEMV = md_hom( *, (++,     +) ) o view(A,B)(i,  k)(A[i,k],B[k]  )
DOT  = md_hom( *, (        +) ) o view(A,B)(    k)(A[k]  ,B[k]  )
```

Convolution

```
CONV = md_hom( * , (++,++,++,++,+,+,+) ) o
  view(images,filter)(n,k,p,q,c,r,s)(images[n,c,p+r,q+s],filter[k,c,r,s])
```

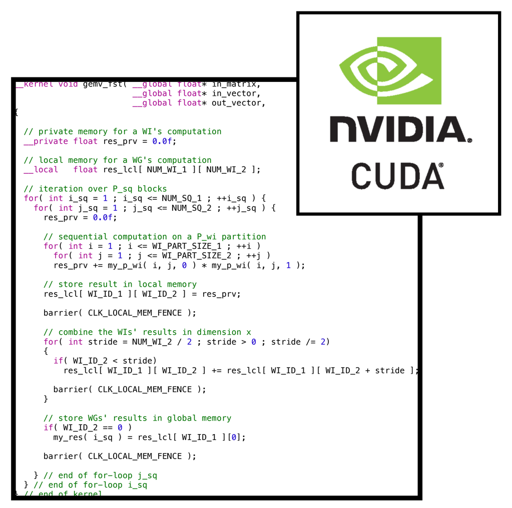
Tensor Contractions

```
TC = md_hom( *, (++,…,++,++,+,…,+) ) o view(…)
```

$$\texttt{md\_hom}(\, f,\ (\circledast_1, \ldots, \circledast_k)\,)$$

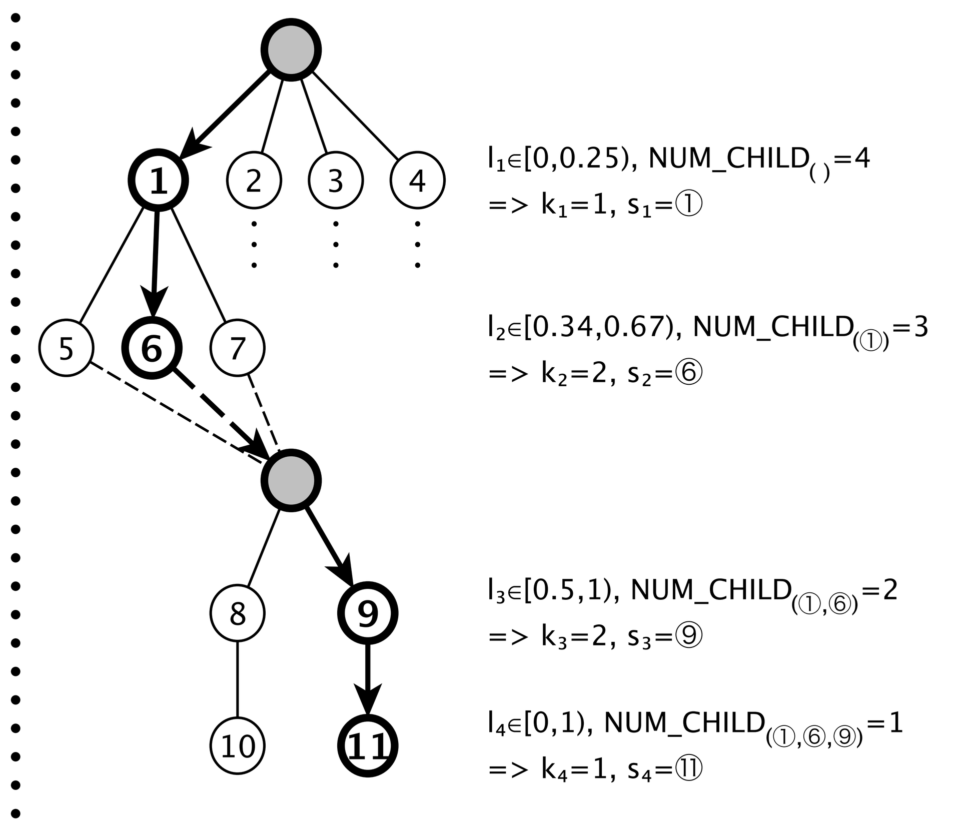
Generating Auto-Tunable CUDA Code

[PACT'19]



Optimization

Our **Auto-Tuning Framework (ATF)** is a **general-purpose approach** that supports auto-tuning of programs with **interdependent tuning parameters.** ***[CCPE'18]***



We provide a novel ***<u>chain-of-trees</u>*** *search space structure* for interdependent tuning parameters.

```
#atf::tp name       /* name       */
         range      /* range      */
         constraint /* constraint */
```

We extend the traditional definition of *tuning parameters by a* ***<u>parameter constraint</u>***.

ATF efficiently ***generates / stores / explores*** the search spaces of **interdependent tuning parameters**

2.91x faster than *NVIDIA cuBLAS*

1.18x faster than *NVIDIA cuBLASLt*

Our MDH approach achieves on **NVIDIA V100 GPU** often **better performance** than **hand-optimized approaches** on **real-world, deep-learning input sizes**.

3.31x faster than NVIDIA cuDNN

2.75x faster than *TVM*