Universal Tensor Network Library, Uni10

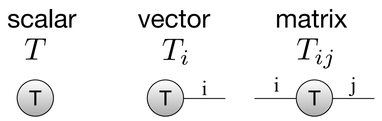
Yun-Da Hsieh, Ying-Jer Kao

Department of Physics, National Taiwan University

Background

Uni10 is a general purpose tensor network library. It has comprehensive tensor operations and network construction methods. With Nvidia GPU, applications using Uni10 may have more than 50x speedup by simply linking GPU version without modifying the programs.

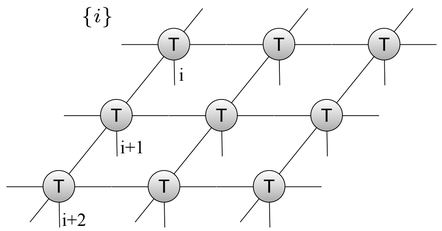
Tensor network is a promising tool in numerical methods for many-body systems in condensed-matter physics and quantum chemistry and recently in big data analysis. The followings are three key features:



Tensors are generalizations of scalars, vectors(one index), and matrices to any number of indices.

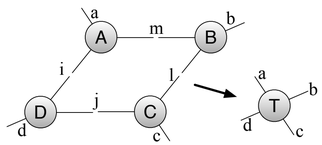
- Natural representation in lattice systems.

$$|\Psi\rangle = \sum_{\{i\}} \Psi_{i_1 i_2 \ldots i_N} |i_1\rangle \otimes |i_2\rangle \otimes \ldots \otimes |i_N\rangle$$



- Concise way to express tensor contractions.

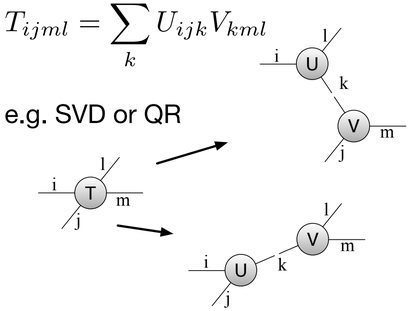
$$T_{abcd} = \sum_{ijml} A_{ami} B_{mlb} C_{ljc} D_{dij}$$



- Intuitive representation for numerical decompositions.

$$T_{ijml} = \sum_{k} U_{ijk} V_{kml}$$

e.g. SVD or QR



Inside Uni10

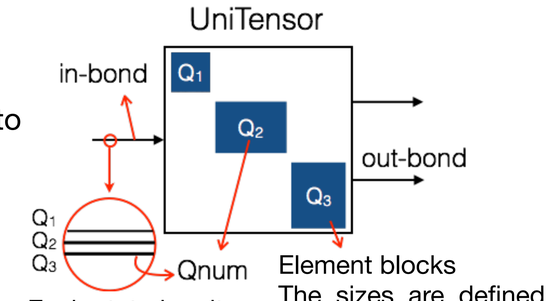
In general, tensor-network algorithms are complicated and hard to maintain and so are CUDA programs.
Uni10 makes these easier and more efficient:

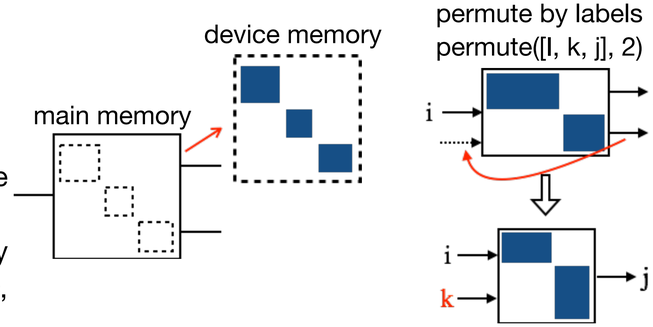
- Tensor structure, with U1 and Z2 symmetry group

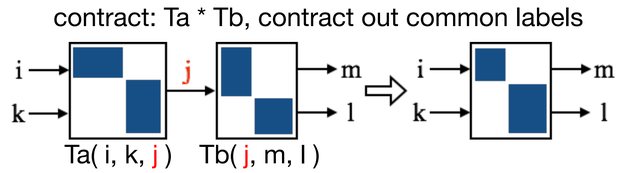
- Bond: an index of a tensor, possessing dimension and symmetry quantum numbers (Qnum: U1 and Z2).
- Element: elements of tensor. With symmetry, elements are in block-diagonal form with symmetry numbers.
- Label: labeling the bonds, used when operating tensor.

- GPU version

- Memory managements:
  - Elements are allocated in GPU device memory if possible to avoid data transfer between device and host.
  - For huge tensor, elements are allocated in main memory and transferred piece-wise to GPU for linear operations, e.g. huge tensor contraction.
  - Automatic transfer when needed, e.g. print tensor.
- Operations on GPU:
  - Operations with CUDA: allocation, memory copy, resize(), permute(), and etc..
  - Linear algebra with cuBLAS and CULA: contraction, svd(), eigh(), and etc..

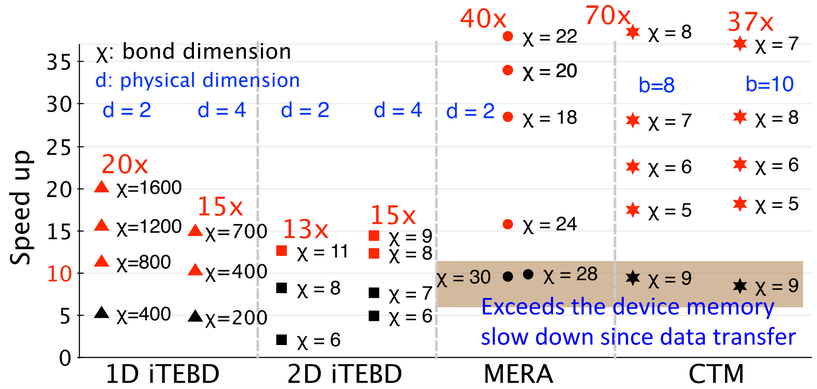






Features & Vision

- Label-based operations API for tensor-network algorithm, concise and intuitive.
- Optimize and parallelize with Nvidia CUDA. Speedup for now as show on the left.
- Painless deployment to GPU version, without any modification in user programs using Uni10.
- Python wrapper. Utilize these features in python.



Accelerate by Nvidia Tesla M2090 with 6GB device memory. Results are compared to single CPU.
In general, large tensor size can have more significant speedup.