Accelerating the calculation of correlation functions in redstar

Eloy Romero [1], Jie Chen [1], Robert Edwards [1], Jefferson Lab [1]

Correlation functions evaluation

Context:

- Evaluation of Lattice Quantum Chromodynamics (LQCD) correlation functions coming from the integration over the fermionic quark fields
- Large number of graphs to evaluate especially for multi-hadron systems
- Each graph involves the contraction of many 2D (for mesons) and 3D (for baryons) tensors when using *distillation*

Impact:

- First three-body scattering amplitude computation directly from LQCD
- First calculations of an exotic meson decay estimating mass and branching fraction
- Relevant for on-going Jefferson Lab GlueX experiment

Examples

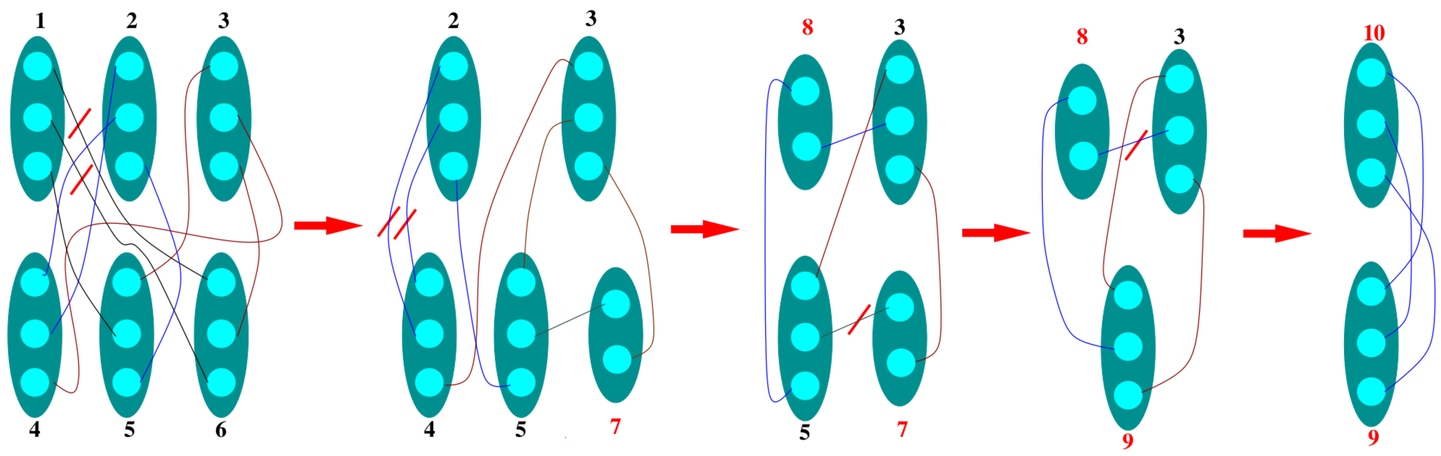
- The correlation functions f_i are the result of the addition of tensor contractions that results in a single complex number

$$f_i = \sum_j f_{i,j}, \text{ where } f_{i,j} = \prod_k M^{(i,j,k)}_{I(i,j,k)} \in \mathbb{C}$$

- Examples:

$$f_0 = M^{(0,0,0)}_{0,1} M^{(0,0,1)}_{0,1} + M^{(0,1,0)}_{0,1} M^{(0,1,1)}_{1,2} M^{(0,1,2)}_{2,0}$$
$$f_1 = M^{(1,0,0)}_{0,1,2} M^{(1,0,0)}_{0,3} M^{(1,0,0)}_{1,2,3}$$

- The $M^{(i,j,k)}_{I(i,j,k)}$ are distillation objects, that is, mesons, baryons, perambulators and generalized perambulators
- Mesons, perambulators and generalized perambulators have two distillation dimensions and two spin dimensions, while baryons have three of each



Implementation description

- Many multi-tensor contractions with common sub-expressions
- 2D and 3D tensors of dimension size $\approx$ 100s
- Main GPU kernel:
  - Permutation of tensors
  - Contract two tensors at a time with batched GEMM
- All input tensors and the temporary tensors may not fit on a single GPU device
- Developed heuristics to:
  - Plan the contractions minimizing the maximum memory footprint
  - Eliminate redundant CPU/GPU memory transfers
  - Handle GPU memory oversubscription via smart eviction
  - Split the contractions among several GPUs minimizing communications
- Multiplatform implementation: CPU, NVIDIA GPUs, AMD GPUs, Intel GPUs (in progress)

Recent GPU optimizations

- Remove many host-device synchronizations
- Reduce by orders of magnitude the amount and volume of memory copies between device and host
- Incorporate heuristics to batch the contractions in a way that minimizes memory footprint
- Optimize functions managing baryons, the largest tensors on our runs

Test

- roper:
  - Main product: $M_{0,1,2}\ M_{3,4,5}\ M_{0,3}\ M_{1,5}\ M_{2,4}$
  - Tensor dimensions: $0, \ldots, 5$ have size 128×2
- nucleon 3 pt:
  - Main product: $M_{0,1,2}\ M_{3,4}\ M_{5,6,7}\ M_{0,5}\ M_{1,3}\ M_{2,6}\ M_{4,7}$
  - Tensor dimensions: $0, 1, 2$ and $5, 6, 7$ have size 64×2, and $3, 4$ has size 64×4
- Timings on a single AMD M250 device:

Test	Number of $M^{(i,j,k)}_{I(i,j,k)}$ Total	Unique	Read from disk	TFLOPs	Time	Speedup from prev. version
roper	1k	334	1 TiB	514	207 s	100x
nucleon 3 pt	71M	2k	1.8 TiB	1,100	900 s	200x