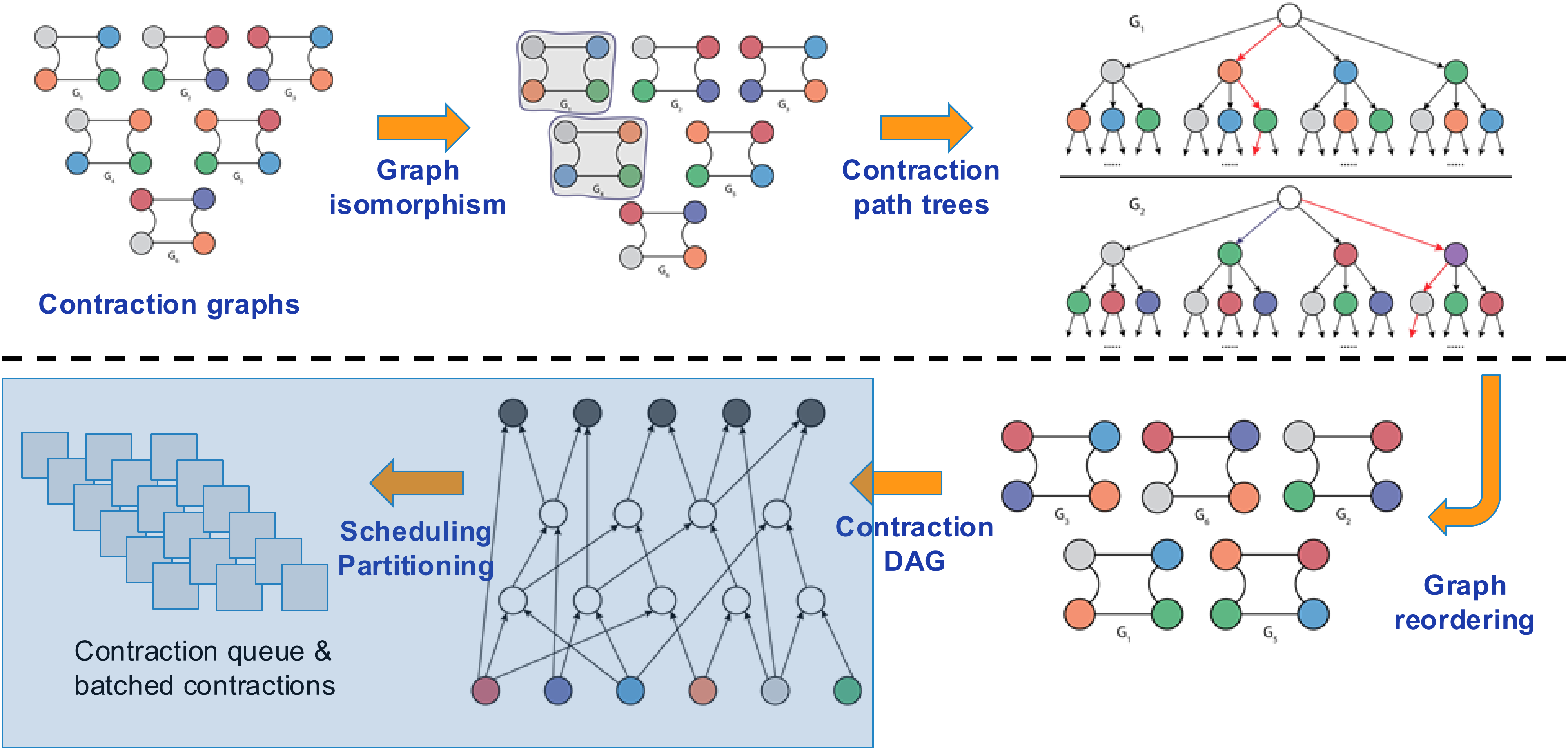
Graph Contractions in Redstar

Oguz Selvitopi, Aydın Buluç (LBNL)
Emin Ozturk, P. (Saday) Sadayappan (Utah)
Jie Chen, Robert Edwards (JLab)

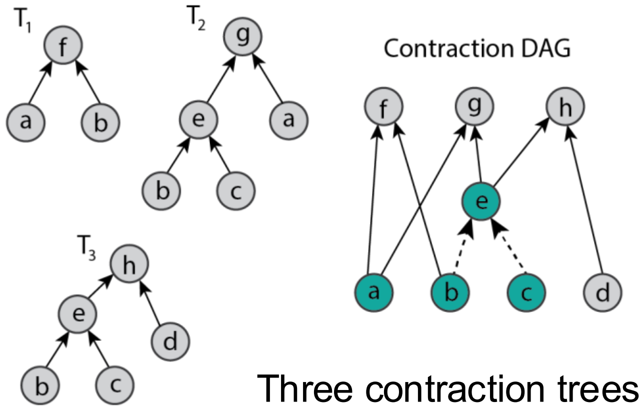
Computational phases & graphs



Scheduling

- **Scheduling**
  - Optimize memory utilization on **single** GPU
  - **Goal**: Increase tensor reuse
    - Reduce evictions, data transfer between host and device
  - **How**: Reorder contractions (contraction DAG)
- **Three heuristics**
  - Sibling-based
  - Node-based
  - Tree-based

Memory model and scheduling order



Three contraction trees and the corresponding contraction DAG

n contractions $c_1, c_2, \ldots, c_n$
Memory in use after contraction c_i: M_i
Peak memory: $\mathbf{max_i\ M_i}$

Order #1
Peak memory: 3

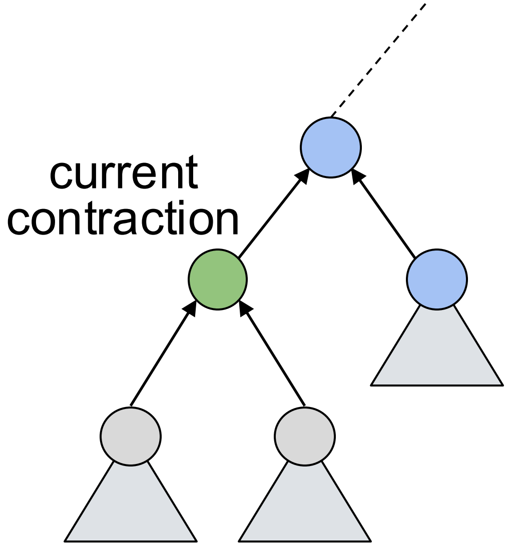
Order	Contents	Size
e	{b, e}	**2**
g	{a, b, e}	**3**
h	{a, b}	**2**
f	{}	**0**

Order #2
Peak memory: 2

Order	Contents	Size
f	{a, b}	**2**
e	{a, e}	**2**
g	{a}	**1**
h	{}	**0**

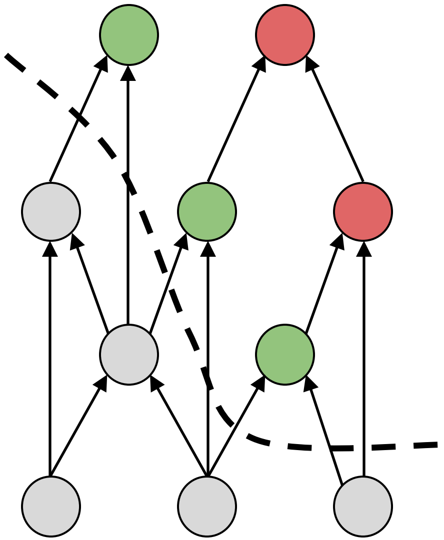
Sibling-based scheduler

- Exploit the specific property that **each contraction is binary**
  - Each node → two children
- **Idea**: After completing a contraction, process its **sibling** contractions
- **Motivation**: Enable contractions higher in the contraction DAG
  - Higher priority for the contractions higher in the DAG
  - Achieve a DFS scheduling of contractions to reduce memory footprint



Node-based scheduler

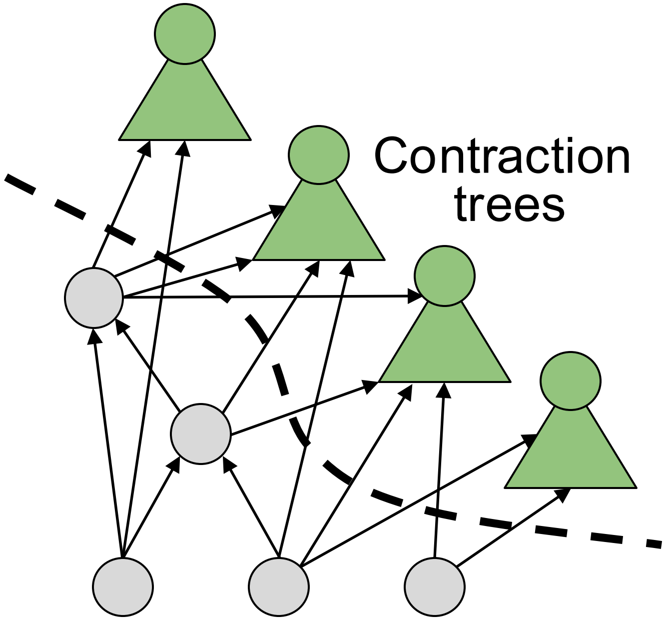
- More **general** scheduling choices
  - Do not depend on specific structure of the DAG
- **Idea**: Choose the contraction that causes **least amount of increase in memory**
  - Most recent state of the memory
- **Motivation**: Scheduling decisions based on **an objective to reduce memory footprint**
  - For each node **u**, maintain
    - **u.δ**: change in utilized memory
  - Global view of the DAG
    - Choose a contraction among all that can be scheduled



GRAY Completed contractions / ready tensors
GREEN Can be scheduled
RED Depend on tensors not yet available

Tree-based scheduler

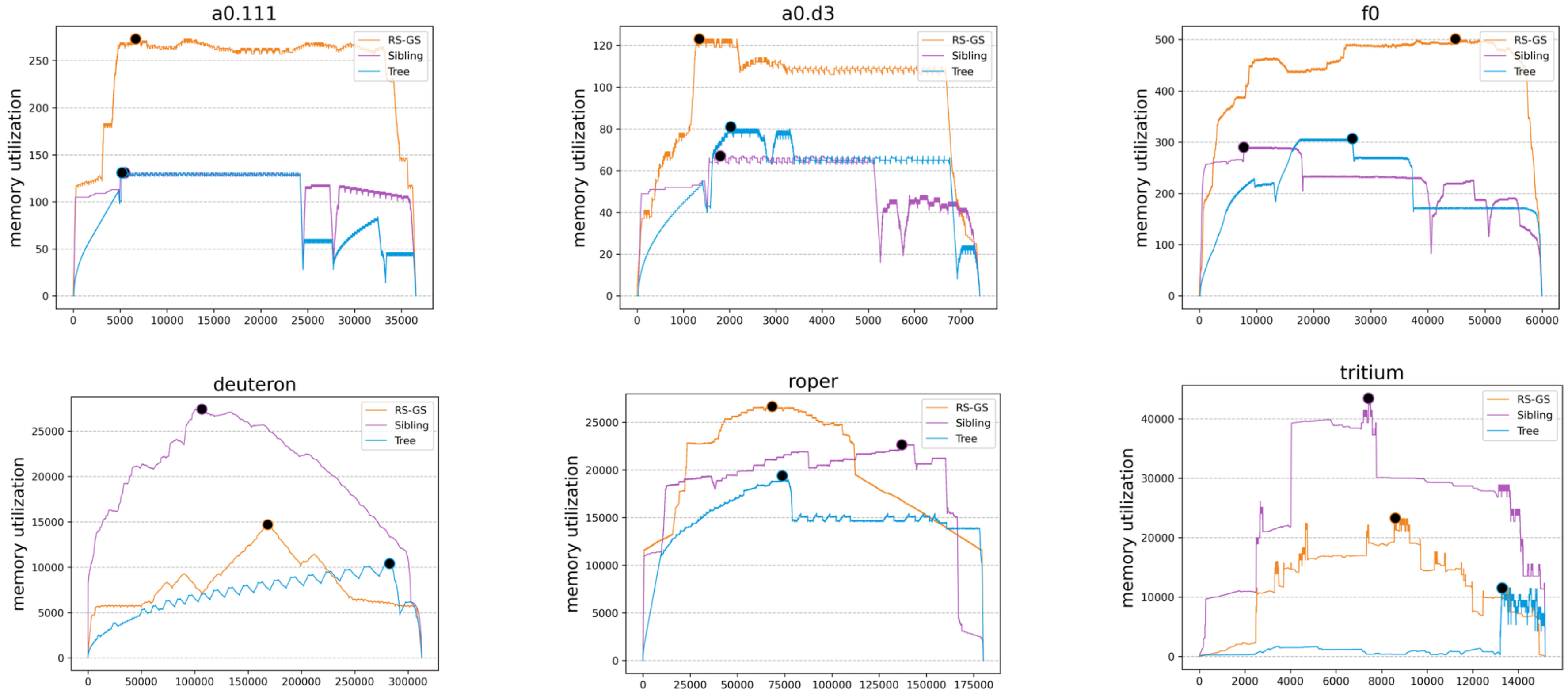
- Similar to node-based scheduler but:
  - Schedule **a subset of nodes** instead of a single node
  - Subset of nodes
    - ≡ nodes in contraction trees
- **Motivation**
  - Nodes in contraction trees are connected
  - Inherent locality
- For each tree T_i
  - **gain(T_i):** Change in memory if contractions in T_i were performed
- Select T_i with **highest gain among all trees**
  - ≡ smallest increase in memory



Peak memory

a0.111: MxM (18K vertices, 36K edges)
a0.d3: MxM (3.8K vertices, 7.2K edges)
f0: MxMxM (30K vertices, 59K vertices)

roper: BxM (90K vertices, 180K edges)
deuteron: BxB (156K vertices, 312K edges)
tritium: BxBxB (7.5K vertices, 15K vertices)



Data movement

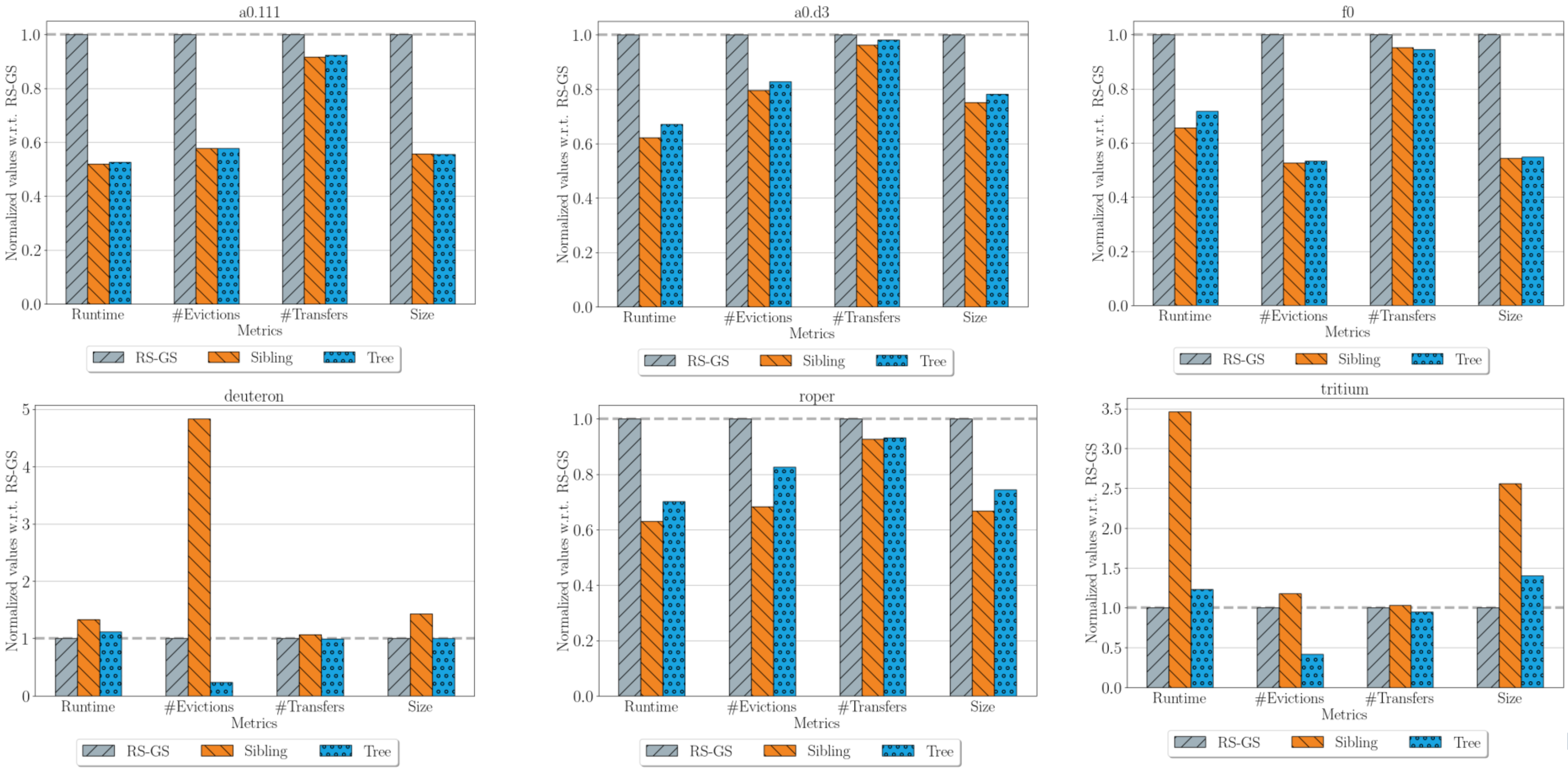
Corr. Func.	RS-GS	Sibling	Tree
a0-111	2.00	1.12	**1.11**
a0-d3	1.12	**0.84**	0.88
f0	1.01	**0.55**	**0.55**
roper	7.35	**4.91**	5.47
deuteron	0.21	0.29	**0.20**
tritium	**0.53**	1.13	0.74

Size in TBs

Corr. Func.	RS-GS	Sibling	Tree
a0-111	3.5	29.5	189.1
a0-d3	0.5	3.8	19.2
f0	5.9	50.0	295.2
roper	26.0	234.3	3095.9
deuteron	38.4	451.0	17005.5
tritium	1.5	25.0	212.5

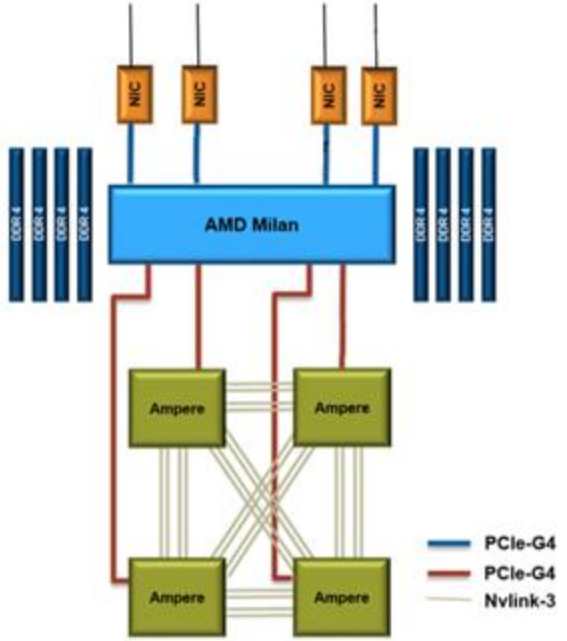
Time in msec

Redstar runtime



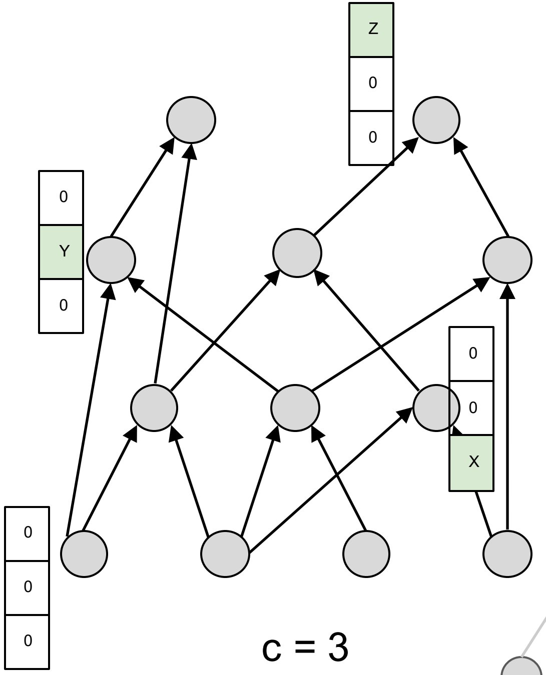
Partitioning

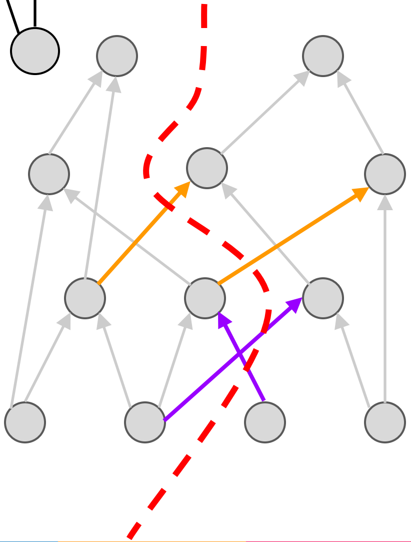
- Distribute contractions among GPUs
  - **Goal:** Reduce data transfers
    - **h2d**: slow (PCIe)
    - **d2d**: fast (NVLink)
  - Balance GPU loads
- Contraction DAG
  - **Leaves** → h2d
  - **Non-leaves** → d2d



Partitioning model

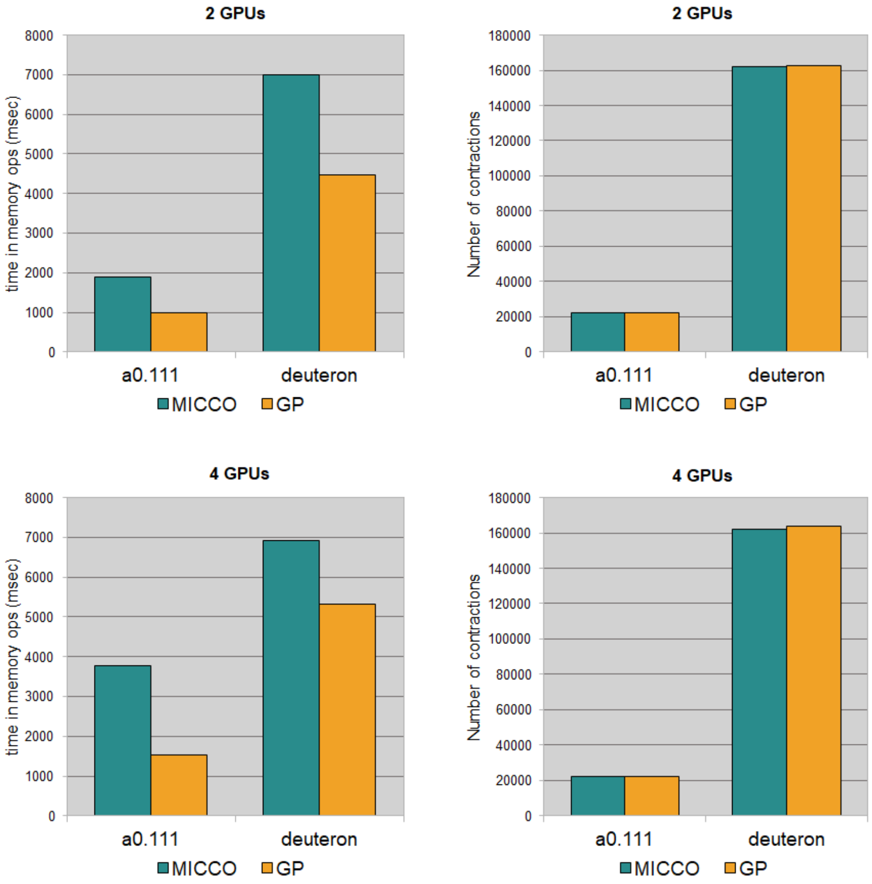
- ▷ **Desired**
  - **#1** Roughly equal amount of contractions from each level
  - **#2** Reduction of h2d is more important than reduction of d2d
- ▷ **Model**
  - Each contraction/tensor → vertex
  - Edge between vertices → dependency of a contraction on a tensor
- ▷ **Vertex weights**
  - **c = number of levels-1** ≡ number of constraints
  - vertex at level i
    - $w_i(v)$ = contraction cost, $w_{1 \leq j \neq i \leq c}(v) = 0$
  - leaves → no computation, no weight





Boundary replication

No communication among devices
= zero d2d operations **(GP)**
Still communication between host and device



Replicate contractions on the boundary
= extra work on GPUs **(GP)**