Optimal dataset transformations for ultrasound simulations

Gabriel Bordovský

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 612 66 Brno - Královo Pole

ibordovsky@fit.vutbr.cz

December 7, 2017

Introduction (1/2)

Ultrasound simulations are used as a modern tool for medical treatment or imaging.

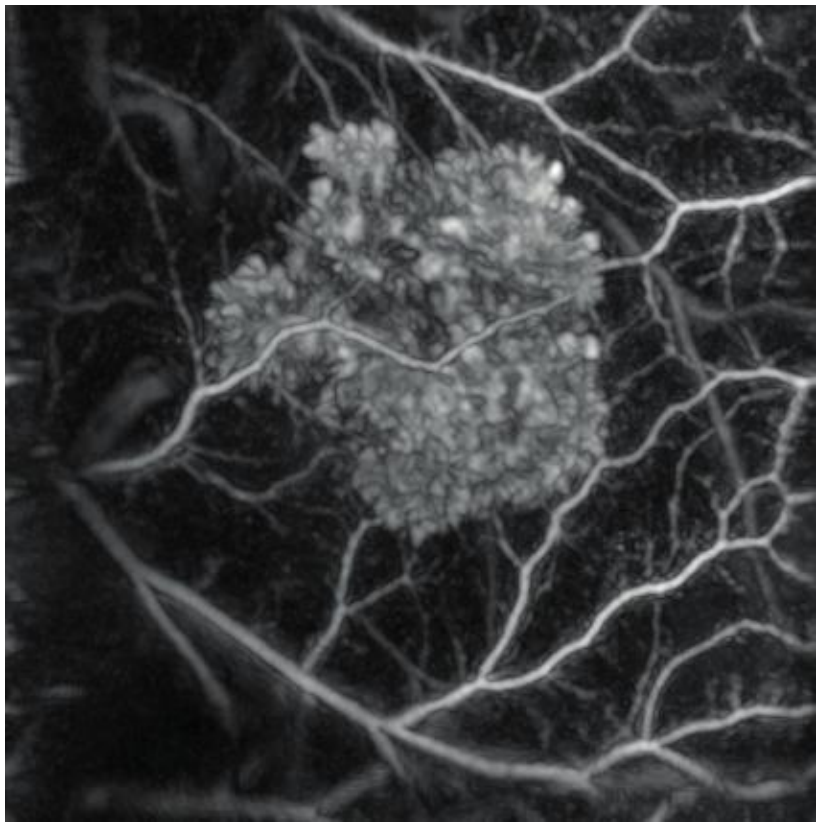
Positives:

- No radiation

Negatives:

- Big data (up to tens of TB)
- Simulation runtime
- Strongly heterogeneous tissue

Introduction (2/2)

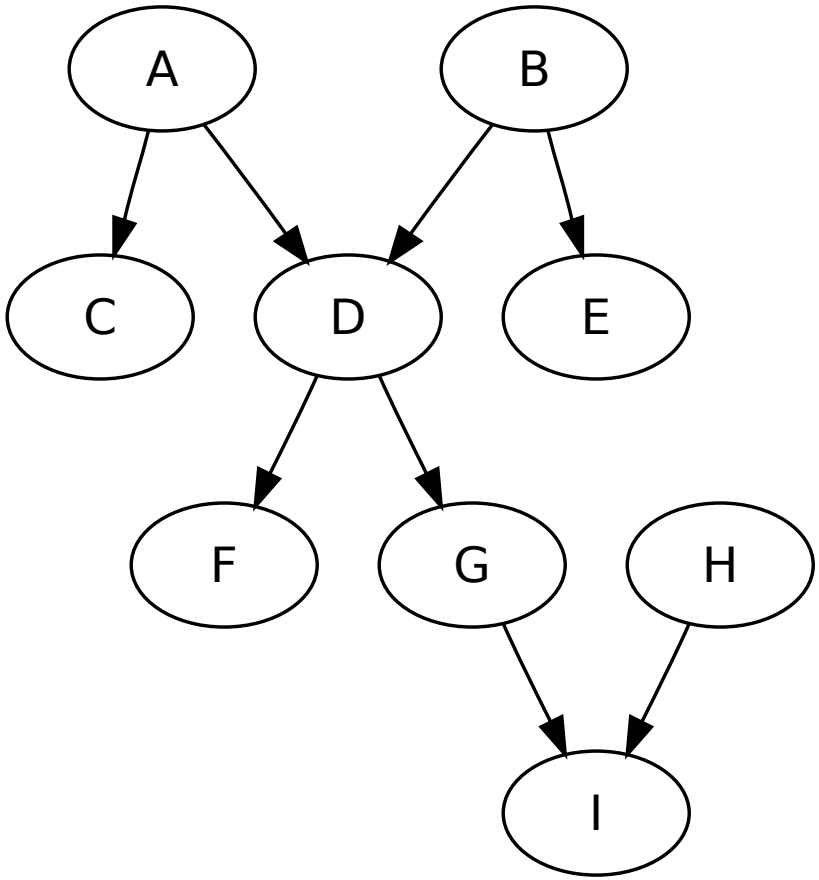


Need for effective parallel algorithms and schedulers that consider:

- Available memory
- Communication time
- Cache locality
- Heterogeneous hardware
- Data dependency

Possible solution

The whole problem may be described as the directed acyclic graph (DAG) where the verticle are tasks and the edges are dependencies. Order in which are the tasks completed is equivalent to the pebbling strategy for the DAG.



Sources for the solution

- Pebble game
- Uniform memory sharing pebble game[1]
- Two-color pebble game for heterogenous computation[2]

[1] Lionel Eyraud-Dubois et al. “Parallel scheduling of task trees with limited memory”. In: ACM Transactions on Parallel Computing 2.2 (2015), p. 13.

[2] Julien Herrmann. “Memory-aware Algorithms and Scheduling Techniques for Matrix Computattions”. PhD thesis. Ecole normale supérieure de lyon-ENS LYON, 2015.

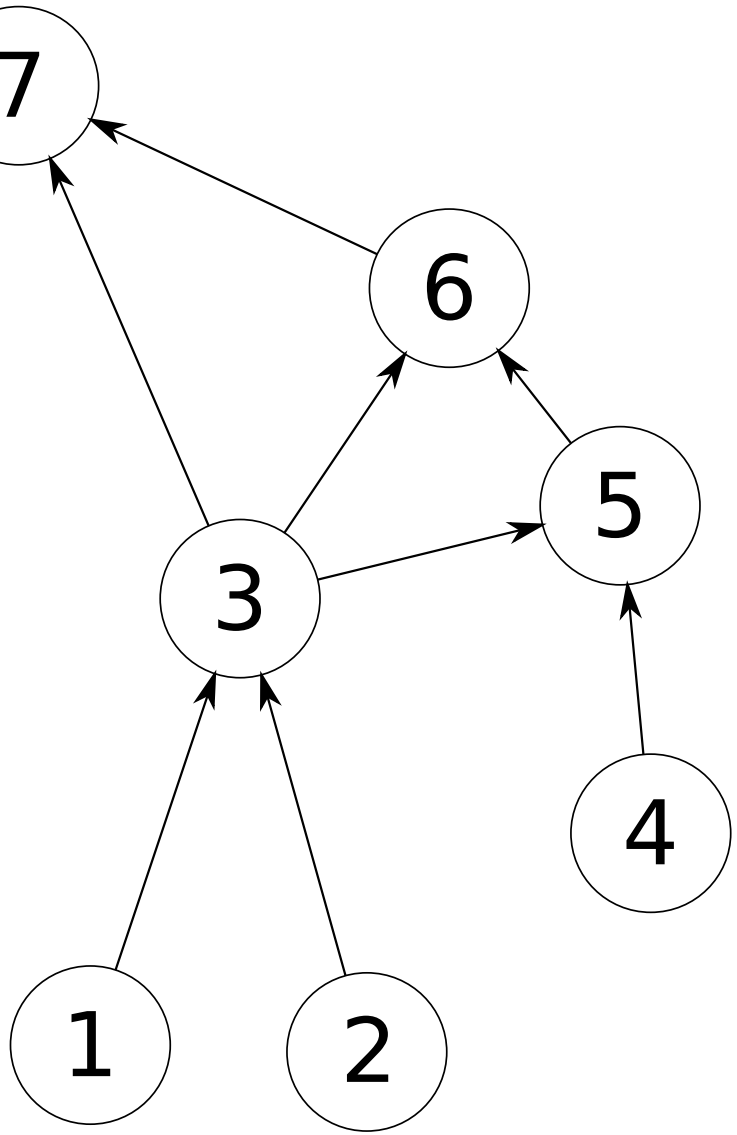
Pebble game

Characteristics:

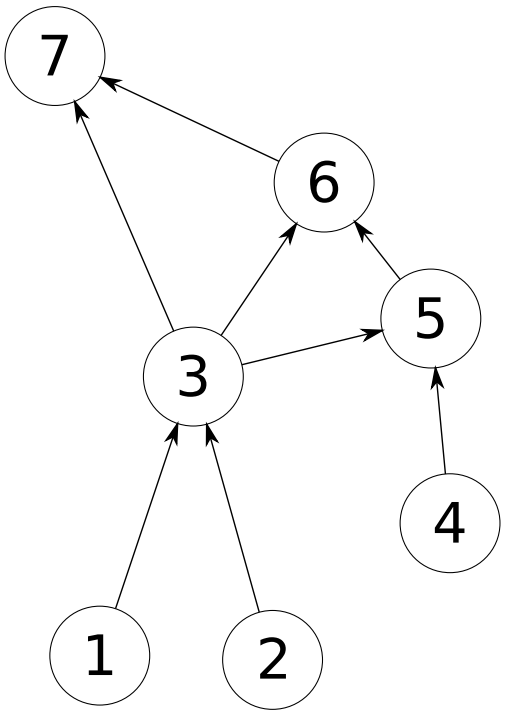
- Simple representation
- Close-enough abstraction
- Easy to understand
- Extensible
- Well-known (in sequence)

Use:

- Proof of optimal execution order
- Analyze of space-time trade-offs
- Minimize algorithm memory footprint



Pebble game example



Timestep	1	2	3	4	5	6
Put a pebble on	1	2	3			4
Take a pebble from				1	2	
Number of pebbles	1	2	**3**	2	1	2

Timestep	7	8	9	10	11
Put pebble on	5		6		7
Take pebble from		4		5	
Number of pebbles	**3**	2	**3**	2	**3**

Pebble game variations

By the source topology:

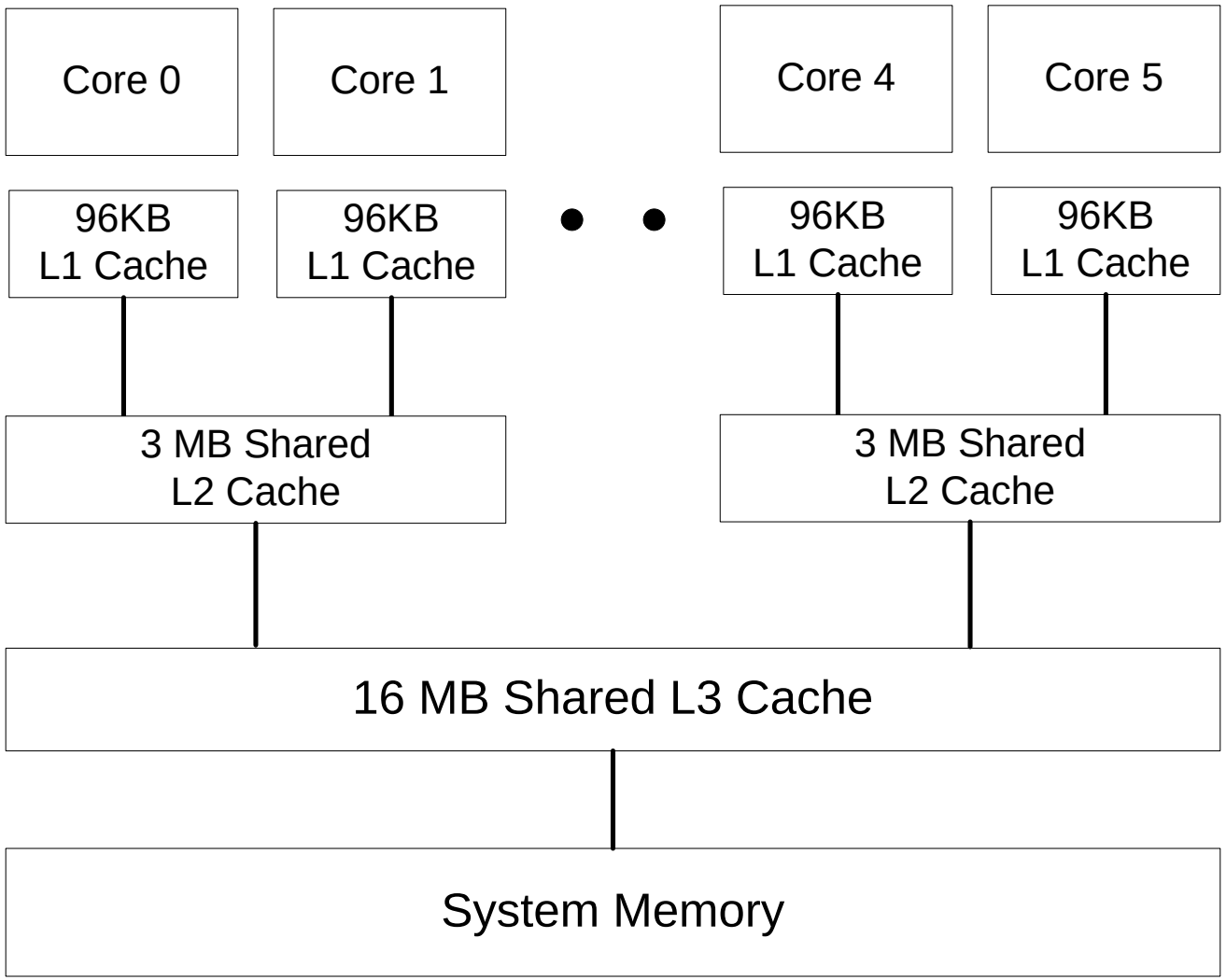
- Complete binary tree
- Pyramid
- Butterfly
- (Un-) limited outputs

By rules modification:

- Limited number of pebbles
- More players (pebble colors)
- Process removes sources
- (Platform specific)
- Allow or forbid re-computation

Cache game

Close to real processors with UNIFORM memory access and cache hierarchy.



Each processor may during one time step do:

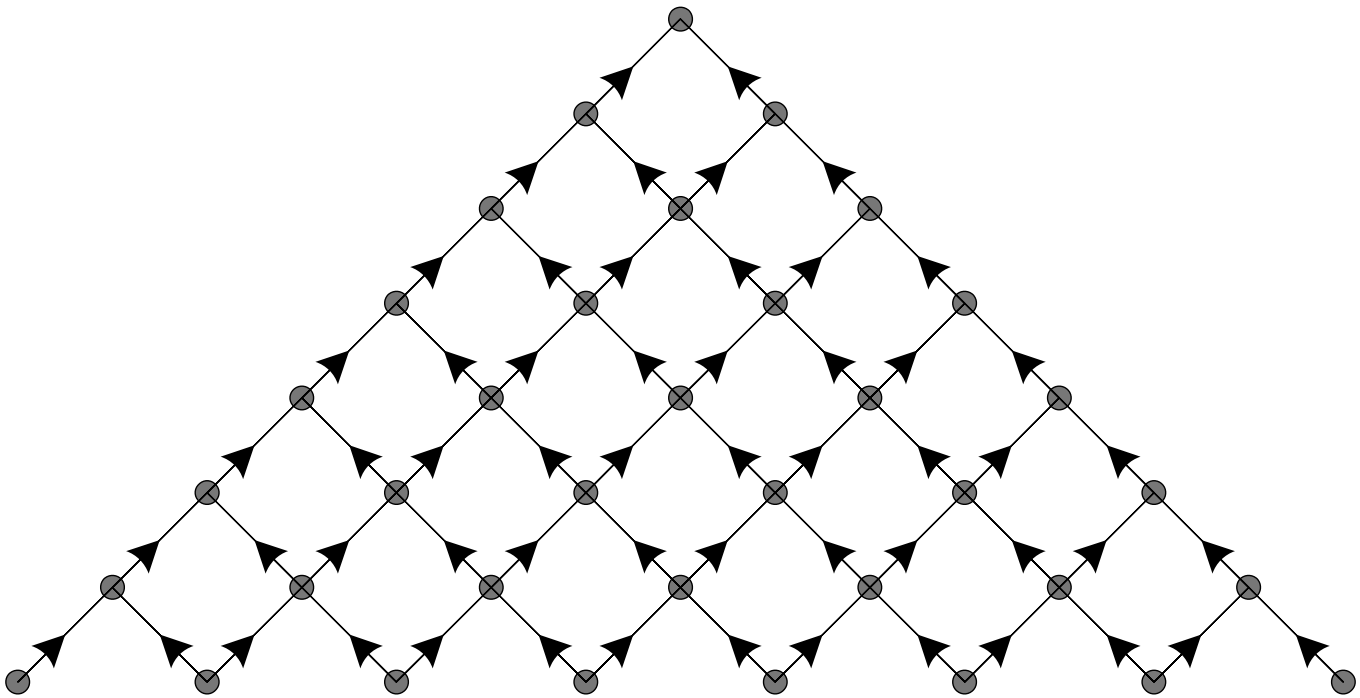
1) Computation
2) Load (from higher level)
3) Store (to higher level)
4) Deletion
5) Initialization
6) Output has to have top level pebble

All cores shares the last level cache. Caches on same level have same size and share the same number of processors. Cores can **not** share a verticle.

Cache game: Strategy 1/2

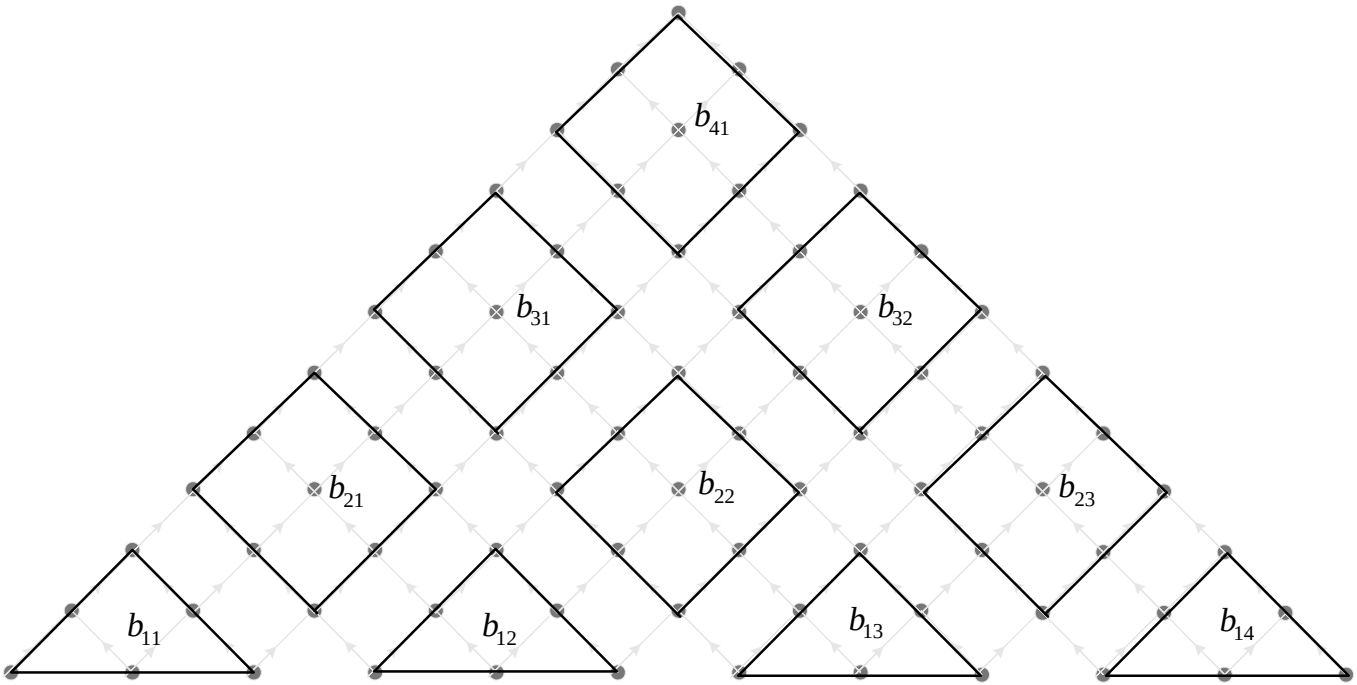
As there are p processors for the given DAG G, it should be split into n tasks. Ideally $p \leq n$ or for the best $n = p \times N$.

Each task is subgraph of the G and can be processed without L/S instruction.



Binomial DAG

As the subgraphs are chosen the blocks that each fits the Lowest cache(L1).

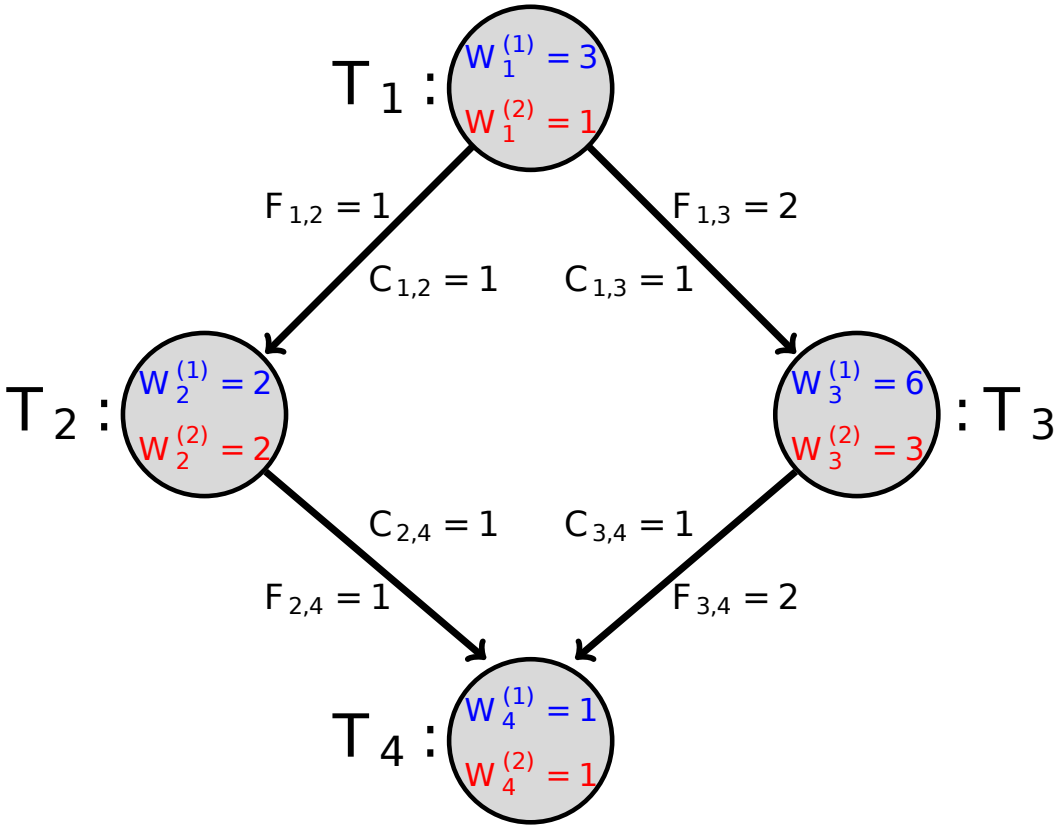


Binomial DAG for L1

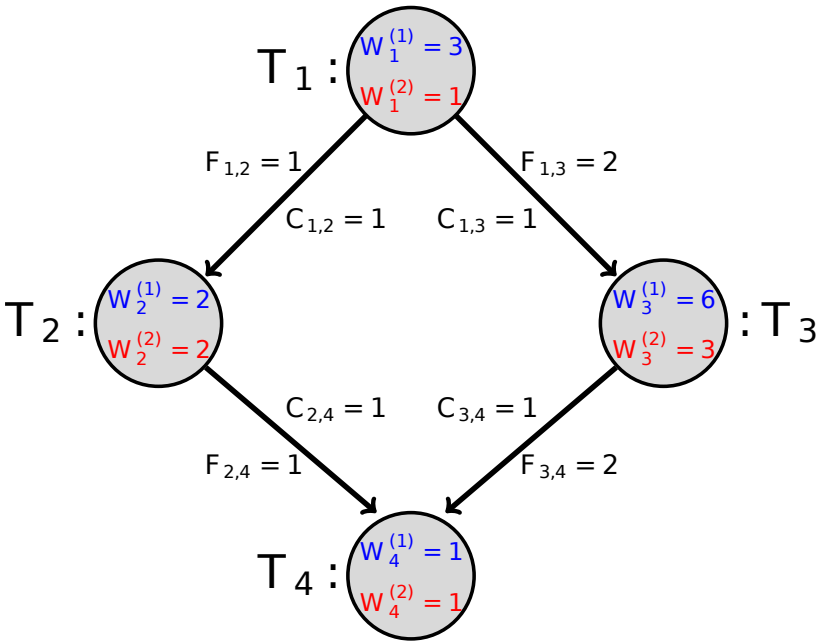
Core does not reuse the data!
Proposition: The task may be re-scheduled for L2.

Hybrid pebble game

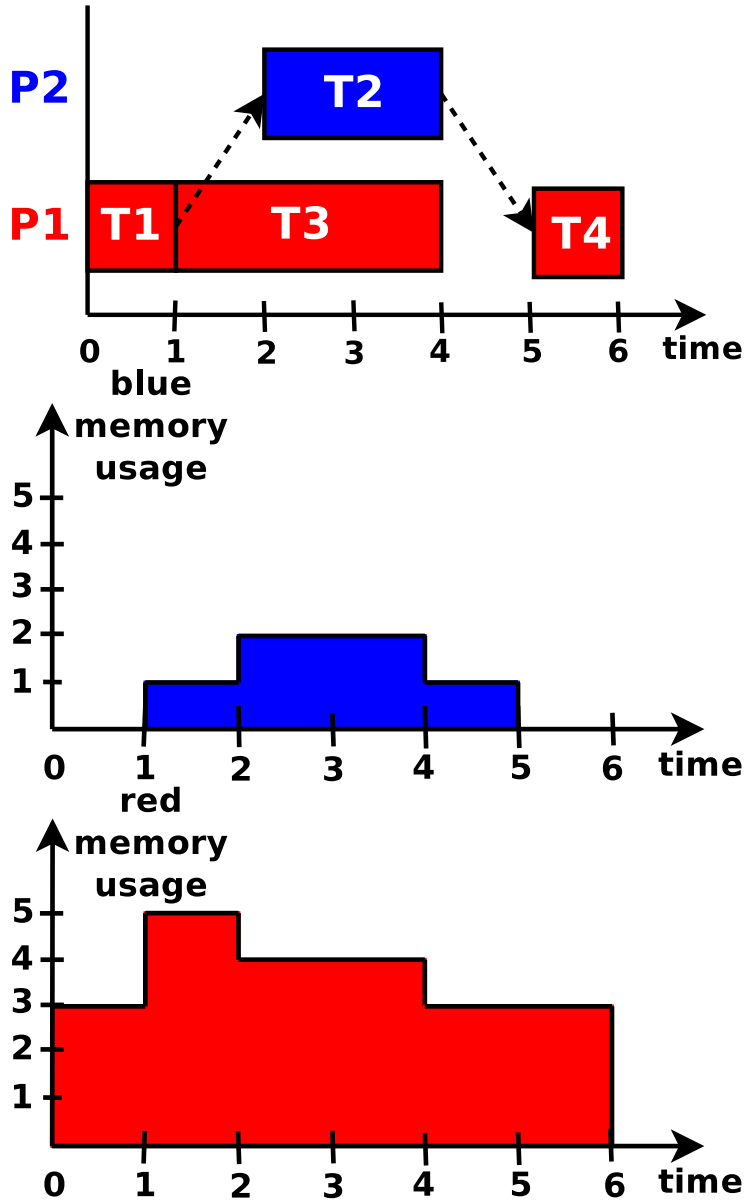
- Different computation speed
- Data transfer latency
- Based on Red-Blue variant of the game



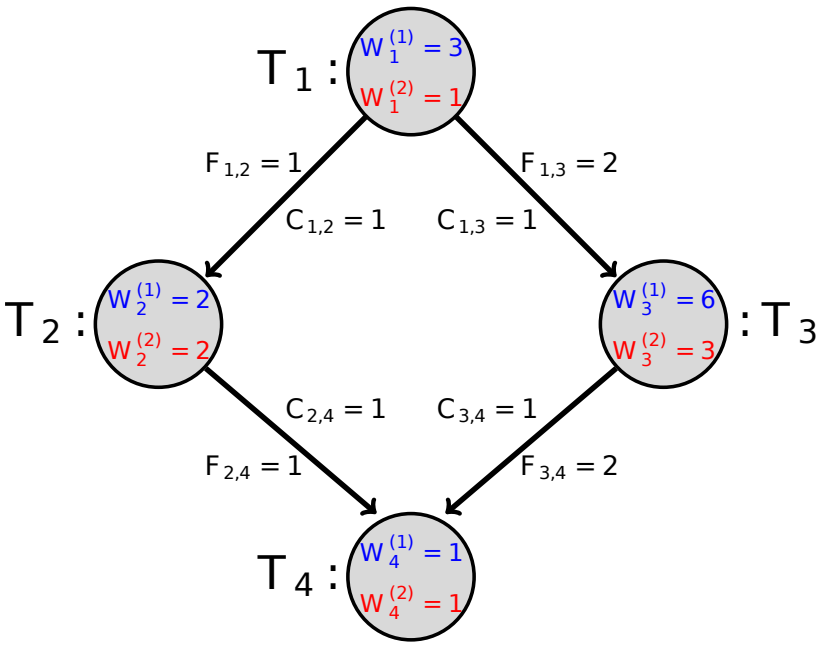
Hybrid pebble game: Strategy 1/2



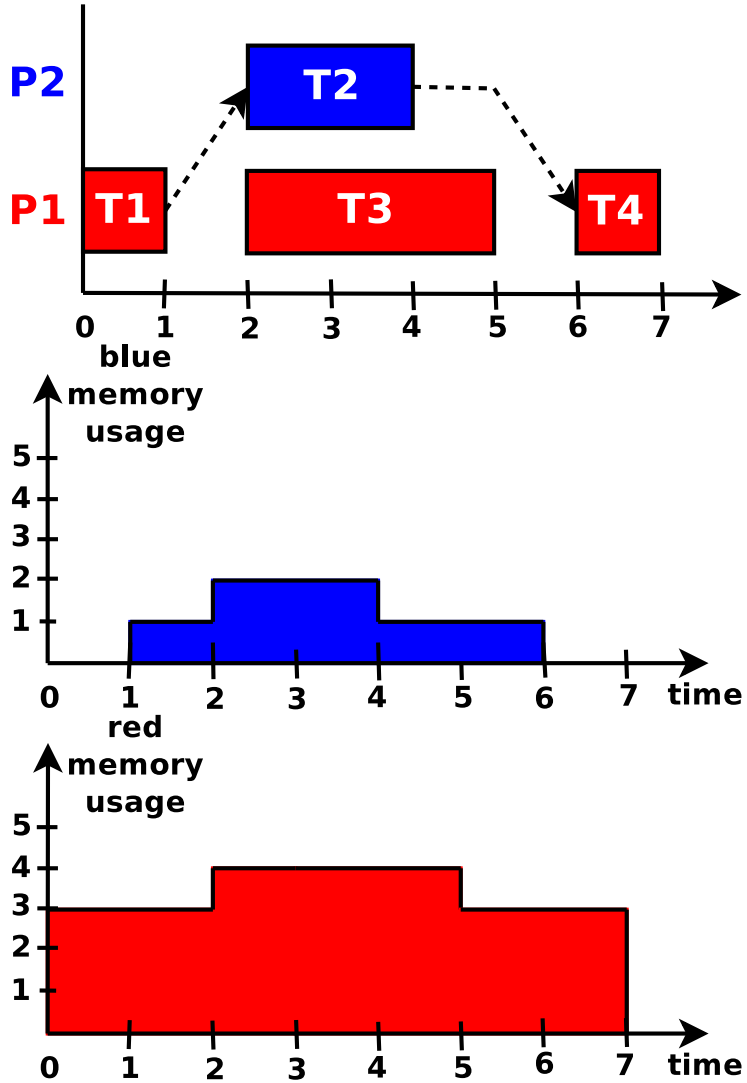
Memory = $F_{in} + F_{out}$



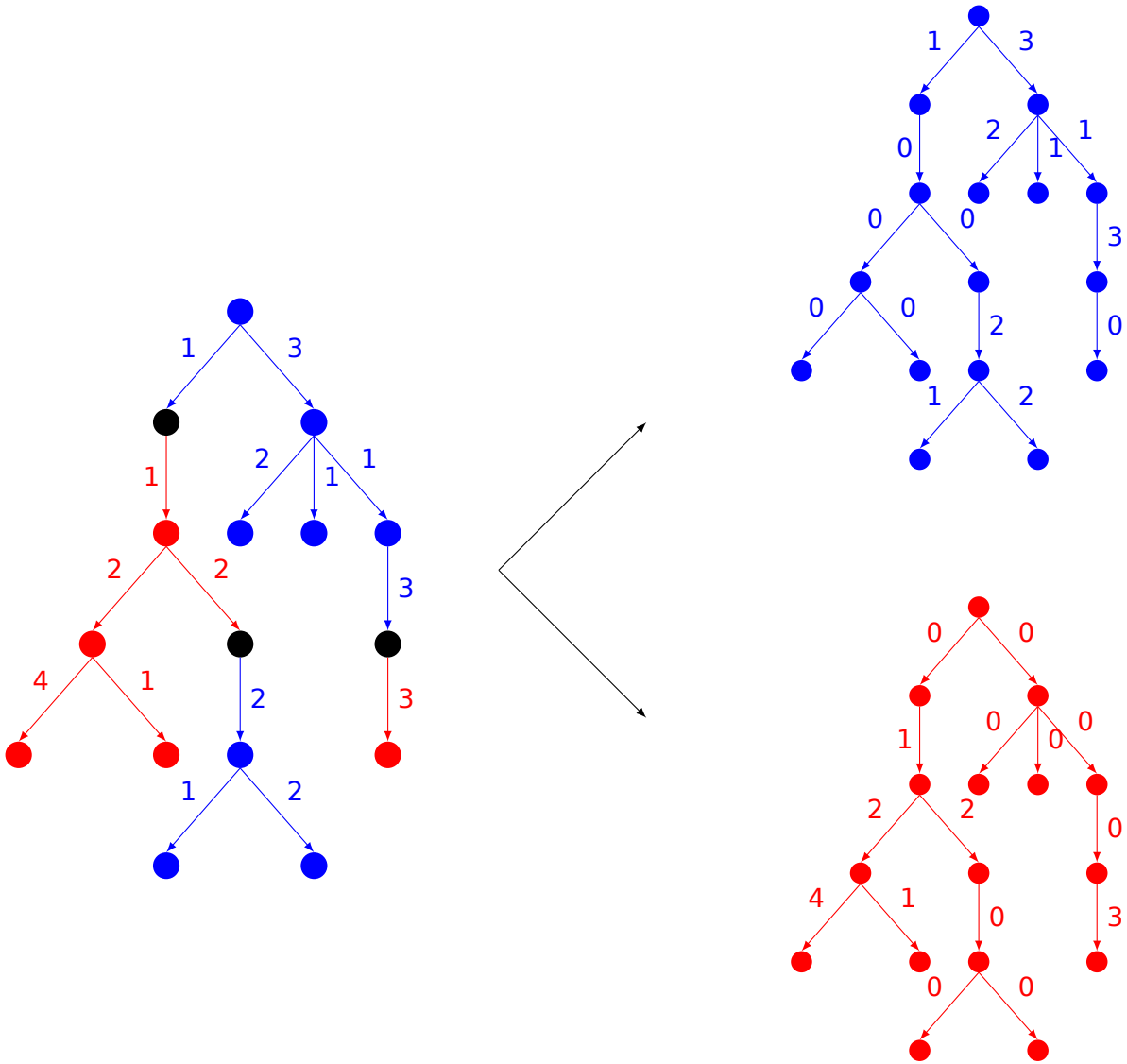
Hybrid pebble game: Strategy 2/2



Max memory = 4



Target specific tasks



Combination = NUMA

Something is missing.

The real computing clusters are often HETEROGENEOUS and NOT-UNIFORM.

The first part provides formal and simple to understand tool for scheduling on different memory levels.

The second part presents communication scheme for not uniform architecture together with different computation architectures.

Possible future work

- Specify custom program measurement for DAG
- Implement memory binding scheduler
- Implement communication heterogeneous scheduler
- Combine them together

References

- J. E. Savage. Models of computation, exploring the power of computing. 2008
- Lionel Eyraud-Dubois et al. “Parallel scheduling of task trees with limited memory”. In: ACM Transactions on Parallel Computing 2.2 (2015), p. 13
- Julien Herrmann. “Memory-aware Algorithms and Scheduling Techniques for Matrix Computattions”. PhD thesis. Ecole normale supérieure de lyon-ENS LYON, 2015