Tree traversals with task-memory affinities on hybrid platforms

Julien Herrmann, Loris Marchal, Yves Robert

Friday Lunch Talk, ICL – May 17, 2013

Outline

Outline

Motivation

- Scientific computing: workflows with large data files
- Bad evolution of processing power vs. communication cost: $1/\text{Flops} \ll 1/\text{bandwidth} \ll \text{latency}$
- Gap increases exponentially

	annual improvements
Flops rate	59%
mem. bandwidth	26%
mem. latency	5%

Solutions:

- Communication-avoiding algorithms
- Restrict to in-core memory (out-of-core is expensive), and minimize memory peak

Motivation

- Scientific computing: workflows with large data files
- Bad evolution of processing power vs. communication cost: $1/\text{Flops} \ll 1/\text{bandwidth} \ll \text{latency}$
- Gap increases exponentially

	annual improvements
Flops rate	59%
mem. bandwidth	26%
mem. latency	5%

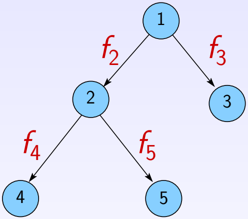
Solutions:

- Communication-avoiding algorithms
- Restrict to in-core memory (out-of-core is expensive), and minimize memory peak

Tree-shaped workflows

Sparse-matrix factorization with multifrontal methods:

- Elimination tree (task graph)
- Large memory peak: memory usage becomes a bottleneck



- Out-tree of tasks
- Dependencies: files with different sizes
- When processing a node, input and output files must fit in memory
- After processing a node, input file is deallocated

- Node schedule (=tree traversal) impacts memory peak
- Schedule for corresponding in-tree: mirror of schedule for out-tree

Memory minimizing traversals: state of the art

General problem on DAGs, with unit costs (pebble game):

- P-Space complete [Gilbert, Lengauer & Tarjan, 1980]
- Without re-computation: NP-complete [Sethi, 1973]

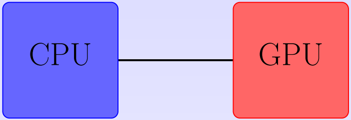
For tree-shaped task graphs, with arbitrary costs:

- Best depth-first traversal [Liu, 1986]
- Best traversal [Liu, 1987]

Previous studies:

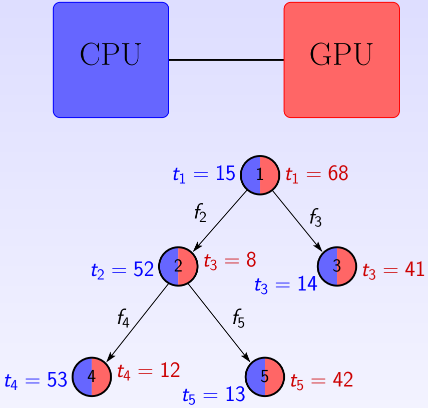
- Comparison of optimal and post-order traversals
- Complexity study of parallel tree traversals

Target hybrid platform



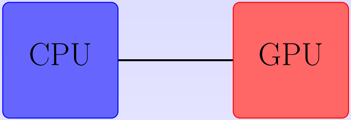
- Realistic model: unrelated computation times
- Simpler model: strong affinities
- Preliminary complexity study:
  only memory matters (sequential execution)

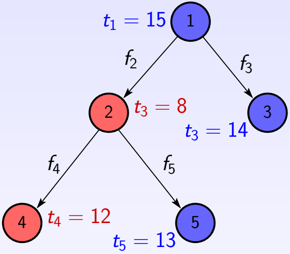
Target hybrid platform



- Realistic model: unrelated computation times
- Simpler model: strong affinities
- Preliminary complexity study:
  only memory matters (sequential execution)

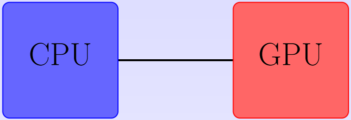
Target hybrid platform

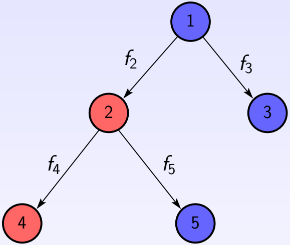




- Realistic model: unrelated computation times
- Simpler model: strong affinities
- Preliminary complexity study:
  only memory matters (sequential execution)

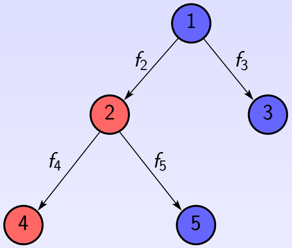
Target hybrid platform





- Realistic model: unrelated computation times
- Simpler model: strong affinities
- Preliminary complexity study:
  only memory matters (sequential execution)

Model

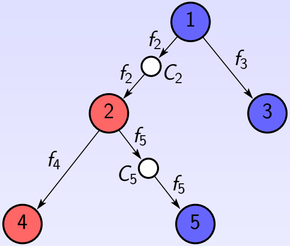


- n colored nodes (tasks)
- f_i: size of input file of task i
- No output file for leaves
- Communication nodes
- No unavoidable communications

- Tree traversal: ordering of computation and communication nodes (which enforces dependencies)
- When a node is processed, its input/output files must fit in the corresponding memory

- Objective: minimize both memory peaks M_{blue} and M_{red}

Model

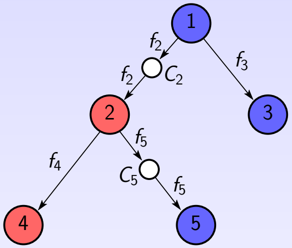


- n colored nodes (tasks)
- f_i: size of input file of task i
- No output file for leaves
- Communication nodes
- No unavoidable communications

- Tree traversal: ordering of computation and communication nodes (which enforces dependencies)
- When a node is processed, its input/output files must fit in the corresponding memory

- Objective: minimize both memory peaks M_{blue} and M_{red}

Model



- n colored nodes (tasks)
- f_i: size of input file of task i
- No output file for leaves
- Communication nodes
- No unavoidable communications

- Tree traversal: ordering of computation and communication nodes (which enforces dependencies)
- When a node is processed, its input/output files must fit in the corresponding memory

- Objective: minimize both memory peaks M_{blue} and M_{red}

Outline

Complexity

TwoMemoryTraversal

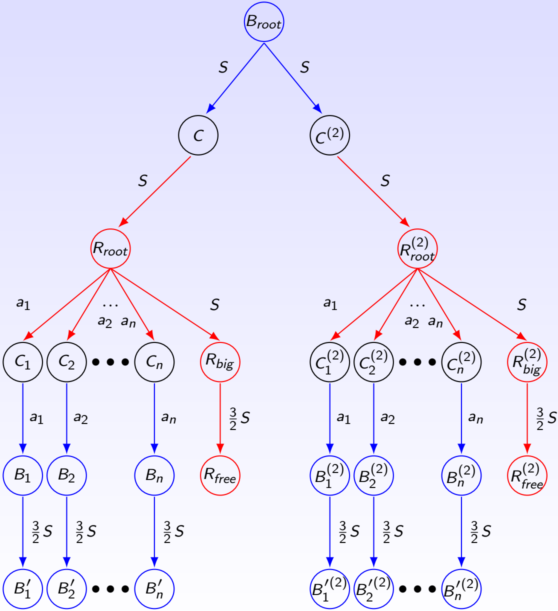
Given a tree $\mathcal{T}$, and two bounds M_{red} and M_{blue}, is there a traversal σ of the tree that uses less than M_{red} red memory and M_{blue} blue memory?

Theorem: NP-completeness

TwoMemoryTraversal is NP-complete.

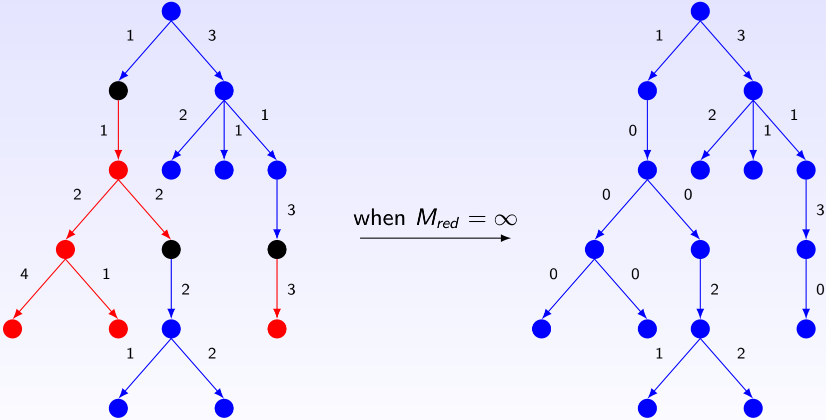
NP-completeness proof

- The problem belongs to NP
- Reduction from the 2-Partition problem
- Instance of the 2-Partition problem: $\begin{cases} a_1, a_2, \ldots, a_n \\ \sum_{i=1}^{n} a_i = S \end{cases}$
- Instance of the TwoMemoryTraversal problem: $\begin{cases} M_{red} = 3S \\ M_{blue} = 2S \end{cases}$



When one memory is unbounded

- Minimization of the second memory usage can be reduced to the uncolored problem.



$$M^{\text{opt}}_{\text{blue}}(\mathcal{T}) = M^{opt}(\mathcal{T}_{blue})$$

Joint minimization of both objectives

Zenith: optimal point for both memories (not a feasible solution a priori)

Theorem: No Zenith approximation

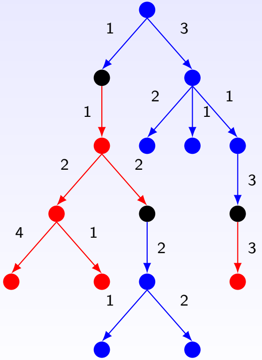
There exists no algorithm that is both an α-approximation for blue memory peak minimization and a β-approximation for red memory peak minimization, when scheduling bi-colored trees.

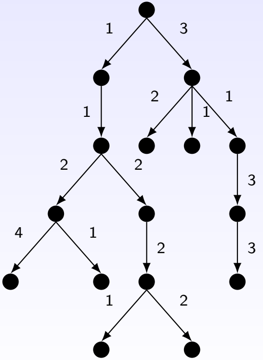
Joint minimization of both objectives

Definition

Given a bi-colored tree $\mathcal{T}$, we note:

- $\mathcal{T}_{\mathrm{unco}}$ the corresponding uncolored tree
- M^{opt}_{unco} the minimal amount of memory needed to process it.



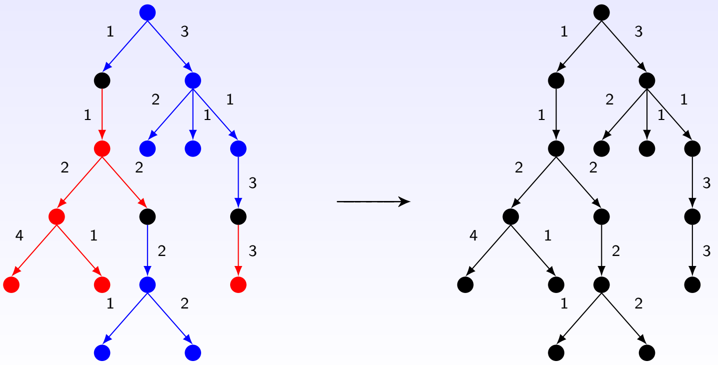


Joint minimization of both objectives

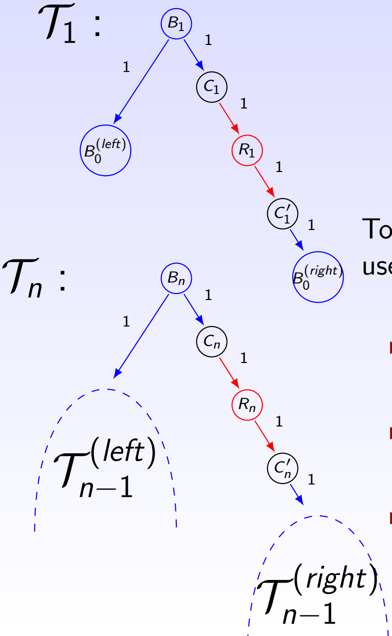
Lemma 1

Given a bi-colored tree $\mathcal{T}$ and an arbitrary traversal σ of $\mathcal{T}$ that requires M^{σ}_{red} units of red memory and M^{σ}_{blue} of blue memory. Then necessarily:

$$M^{\sigma}_{red} + M^{\sigma}_{blue} \geq M^{opt}_{unco}$$



Inapproximability proof



To derive the contradiction, we use the family of tree $(\mathcal{T}_n)_{n\in\mathbb{N}}$

- $\forall \mathbf{n} \geq \mathbf{1}, \mathbf{M}_{\mathrm{red}}^{\mathrm{opt}}(\mathcal{T}_\mathbf{n}) = \mathbf{2}$
- $\forall \mathbf{n} \geq \mathbf{2}, \mathbf{M}_{\mathrm{blue}}^{\mathrm{opt}}(\mathcal{T}_\mathbf{n}) = \mathbf{3}$
- $\forall \mathbf{n} \geq \mathbf{2}, \mathbf{M}_{\mathrm{unco}}^{\mathrm{opt}}(\mathcal{T}_\mathbf{n}) = \mathbf{n} + \mathbf{2}$

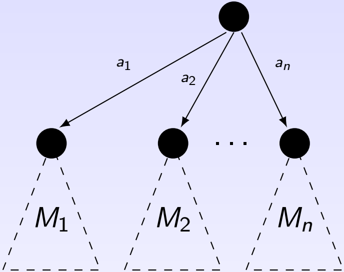
Inapproximability proof

- Two integers: α and β
- An algorithm: $\mathcal{A}$ being an (α,β)-approximation of the Zenith.
- $n_0 = \lceil 3\alpha + 2\beta \rceil$

$$\begin{aligned} M^{\mathcal{A}}_{\mathrm{blue}}(\mathcal{T}_{n_0}) + M^{\mathcal{A}}_{\mathrm{red}}(\mathcal{T}_{n_0}) &\leq 3\alpha + 2\beta \\ &< \lceil 3\alpha + 2\beta \rceil + 2 \\ &= M^{\mathrm{opt}}_{\mathrm{unco}}(\mathcal{T}^{\mathrm{unco}}_{n_0}) \end{aligned}$$

Outline

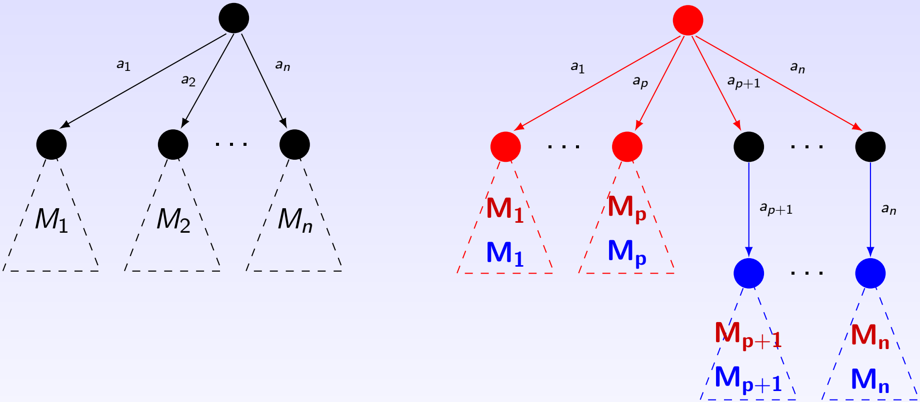
Best Depth First traversal



Liu's Best Depth-First Traversal:

- Subtrees in non-decreasing order of $M_i - a_i$

Best Depth First traversal



Liu's Best Depth-First Traversal:

- Subtrees in non-decreasing order of $M_i - a_i$

Adaptation to bi-colored trees:

- Subtrees in non-decreasing order of $\mathbf{M_i} - a_i$

Heuristics

- BestDepthFirst
- LiuUncolored: optimal algorithm for the single memory problem [Liu, 1987] (optimal for the sum)
- LiuWeightedSum: Liu's algorithm on the tree with normalized weight for the edges (optimal for the weighted sum)
  - $f_i \leftarrow f_i / M^{opt}_{color(i)}$
- LiuWeightedMax: Liu's algorithm with the maximum relative overhead as criterium for combination (not optimal for the maximum)

Data sets

- RealTrees: assembly trees resulting of a multi-frontal factorization of sparse matrices [University of Florida Sparse Matrix Collection]
- ColoredTrees: assembly trees with a CPU/GPU coloring
- RandColoredTrees: assembly trees with a random coloring
- RandWeightedTrees: assembly trees with a random coloring and random edges weight
- RandomTrees: generated random trees

Results

- Bi-criteria optimization (two equivalent memories)
- Criterion: maximum relative overhead

$$max \left(\frac{M_{red}^{used} - M_{red}^{opt}}{M_{red}^{opt}}, \frac{M_{blue}^{used} - M_{blue}^{opt}}{M_{blue}^{opt}} \right)$$

Data set	Algorithm	Avg.	Max.	Std. Dev.	Frac. of Opt.	$\leq$ 10%
ColoredTrees	**Depth-first**	**6.3%**	**64.4%**	**8.0%**	**55.6%**	**73.7%**
	LiuUncolored	6.6%	60.0%	8.3%	55.0%	73.8%
	LiuWeightedSum	7.5%	76.0%	9.1%	52.8%	70.6%
	LiuWeightedMax	**8.4%**	**116.5%**	**9.9%**	**49.8%**	**68.3%**
RandColoredTrees	**Depth-first**	**3.8%**	**44.0%**	**5.4%**	**67.2%**	**83.9%**
	LiuUncolored	5.2%	52.6%	6.9%	59.7%	78.0%
	LiuWeightedSum	5.9%	52.6%	7.3%	54.1%	75.8%
	LiuWeightedMax	**6.0%**	**52.3%**	**7.2%**	**51.4%**	**75.5%**
RandWeightedTrees	**Depth-first**	**20.9%**	**90.3%**	**18.6%**	**28.3%**	**44.6%**
	LiuUncolored	15.4%	413.1%	17.0%	26.5%	60.2%
	LiuWeightedSum	13.4%	107.5%	16.3%	37.7%	65.2%
	LiuWeightedMax	**10.2%**	**88.2%**	**13.6%**	**39.8%**	**72.7%**
RandomTrees	Depth-first	4.5%	28.2%	4.3%	33.4%	83.4%
	LiuUncolored	**6.8%**	**32.9%**	**4.8%**	**14.6%**	**72.6%**
	LiuWeightedSum	4.4%	21.4%	3.7%	20.6%	86.0%
	LiuWeightedMax	3.4%	**23.5%**	**3.2%**	**26.0%**	**92.0%**

Outline

Conclusion and perspectives

- Model for memory-aware hybrid computations
- NP-completeness and inapproximations results
- Optimal depth-first search traversal
- Design of heuristics, experimentally compared on real trees

Perspectives:

- Refine the model
  - Include task computation times on both resources
  - Minimize both makespan and memory peaks
  - Model data movement (CPU-GPU communications)
  - Consider several CPUs/GPUs
- Consider general DAGs