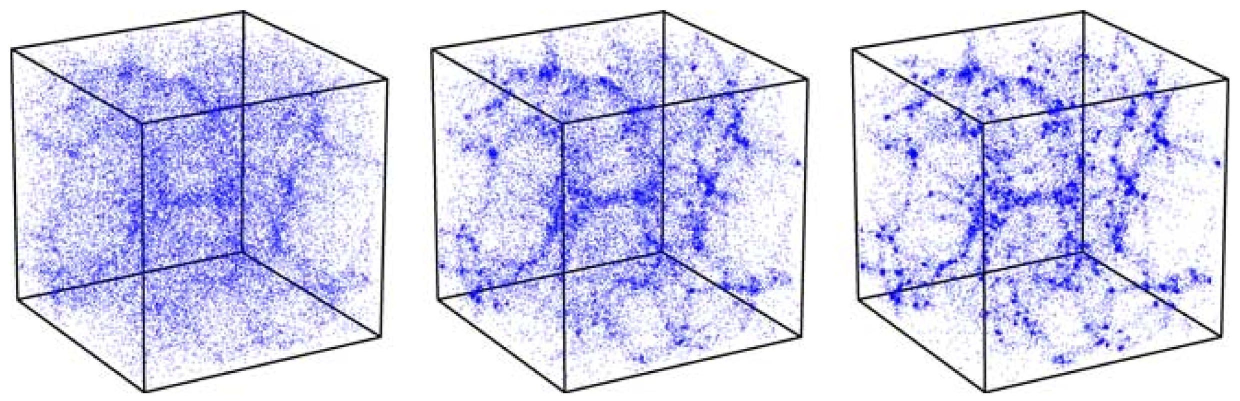
Sakura: Recursive Construction and Traversal of FMM Skeleton

Nikos Sismanis[1] Alexandros-Stavros Iliopoulos[2] Rio Yokota[3] Nikos P. Pitsianis[1,2] Xiaobai Sun[2]

[1] ECE, Aristotle University of Thessaloniki [2] CS, Duke University [3] GSIC, Tokyo Institute of Technology

Motivation

- Tree building and graph traversal for interaction challenge parallel architectures, slowing down parallel FMM execution.
- Updating the tree and interaction lists is critical in time-dependent computations.
- Infrequent updates lead to loss of accuracy.



Evolution of galaxy distribution in Astrophysics simulation; figure taken from [2].

→ Sakura takes a fully recursive approach for processing the FMM skeleton (tree partitions + interaction lists), ensuring the efficient storage and traversal of the data structure.

Adaptive Tree Building

- Fast spatial partitioning of source and target sets via adaptive geometric binning.
- Hierarchically local mapping of particle records to memory.

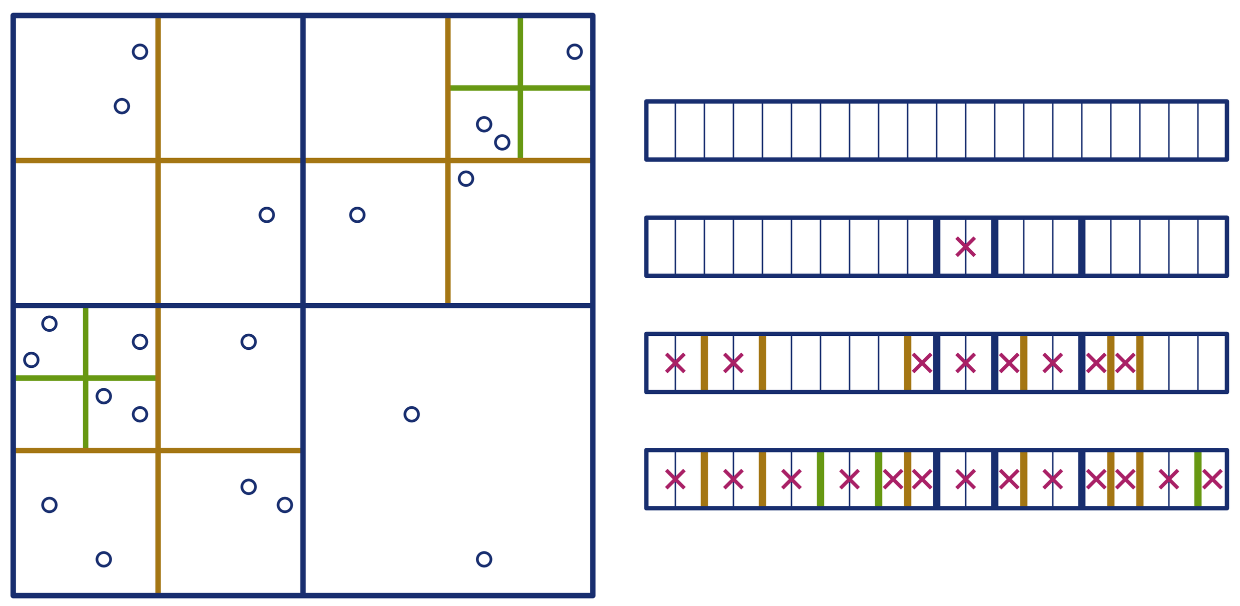


Illustration of geometric partitioning and corresponding memory rearrangement of a 2D particle-set. The particle records are recursively binned in memory, following the spatial partition hierarchy. The recursion terminates for partitions/bins whose population is below a threshold, while empty partitions/bins are discarded.

References

[1] L. F. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, 1987.

[2] E. Platen, R. van de Weygaert, and B. J. T. Jones. A cosmic watershed: the WVF void detection technique. *Monthly Notices of the Royal Astronomical Society*, 380(2):551–570, Sept. 2007.

[3] H. C. Plummer. On the problem of distribution in globular star clusters. *Monthly Notices of the Royal Astronomical Society*, 71(1):460–470, 1911.

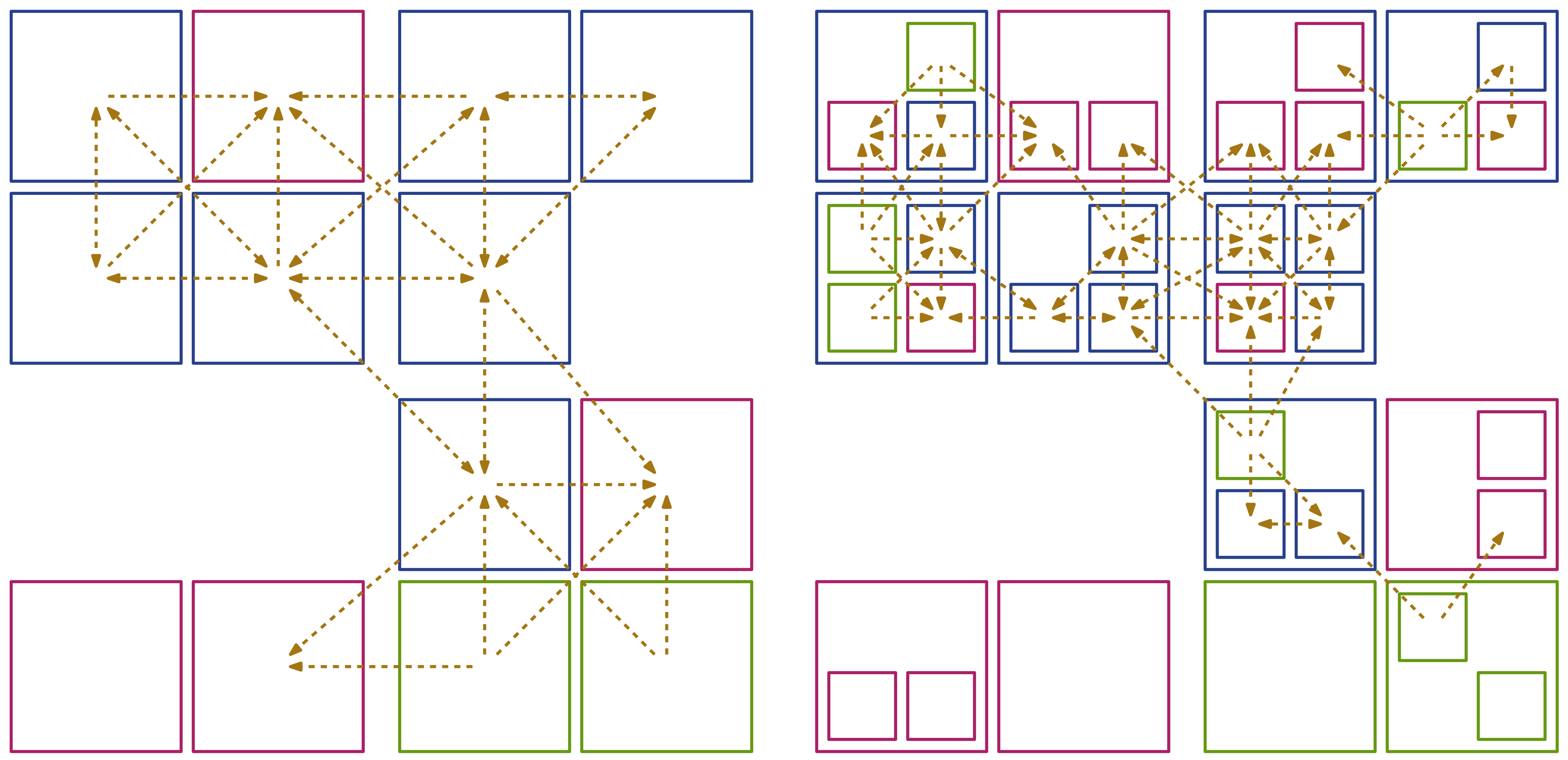
[4] R. Yokota. An FMM based on dual tree traversal for many-core architectures. *Journal of Algorithms and Computational Technology*, 7(3):301–324, 2013.

FMM Interaction Lists

▶ *Explicit formation of interaction lists*

1. Fast counting of interaction links (dual-tree traversal; pass 1)
2. Tight memory allocation for all links with a single OS call
3. Instantiation of interaction lists (dual-tree traversal and memory stuffing; pass 2)

- Minimal number of node-pair visits during the counting and instantiation passes.



(level 2) (level 3)

Near-neighbor links in two resolution levels of a sample dual tree (source + target trees). **Green** boxes contain target particles, **magenta** boxes contain source particles, while **blue** boxes contain both target and source particles. **Orange** arrows indicate target-to-source links. Interaction lists (not shown) are induced by multi-level neighboring relationships; hence, only near-neighboring pairs of nodes need to be visited for interaction list formation.

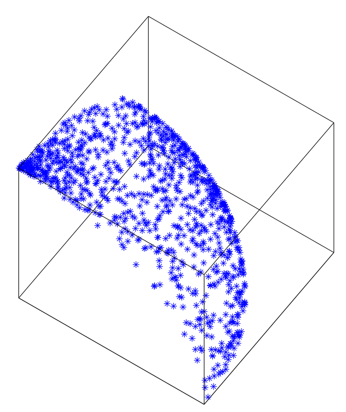
- Original, compressed, and cross-level interaction lists are supported.

▶ *Implicit-list interactions*

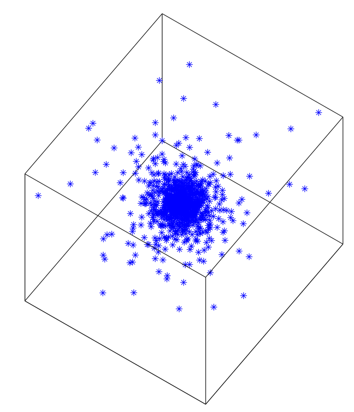
– The fast traversal algorithm is also applicable to implicit interaction list approaches.

Experimental Set-up

- Particle distributions for simulations:



uniform distribution on sphere octant surface



Plummer distribution [3]

- Low- and high-accuracy simulations:
  – expansion order $p = 4$ and $p = 9$
- Simulation platform:

CPU Type	CPUs	Cores	Thr.	CPU clock	Cache levels L1	L2	L3
Xeon E5-2650	2	8	16	2.4GHz	32KB	256KB	30MB

Experiments

Sakura performance (comparison with the widely-used ExaFMM package [4]):

ratio of combinatorial construction to total FMM execution time (t_c/t_c+t_n)

	$p=4$				$p=9$			
	Octant		**Plum.**		**Octant**		**Plum.**	
N	*skr*	*exa*	*skr*	*exa*	*skr*	*exa*	*skr*	*exa*
10	20	48	22	38	6	16	4	11
20	18	50	22	43	5	18	4	12
40	18	53	22	45	5	21	4	13
80	18	50	20	45	5	18	3	14
160	18	54	21	50	5	19	3	15
320	19	54	21	52	5	19	3	31

total FMM execution time in seconds $(t_c + t_n)$

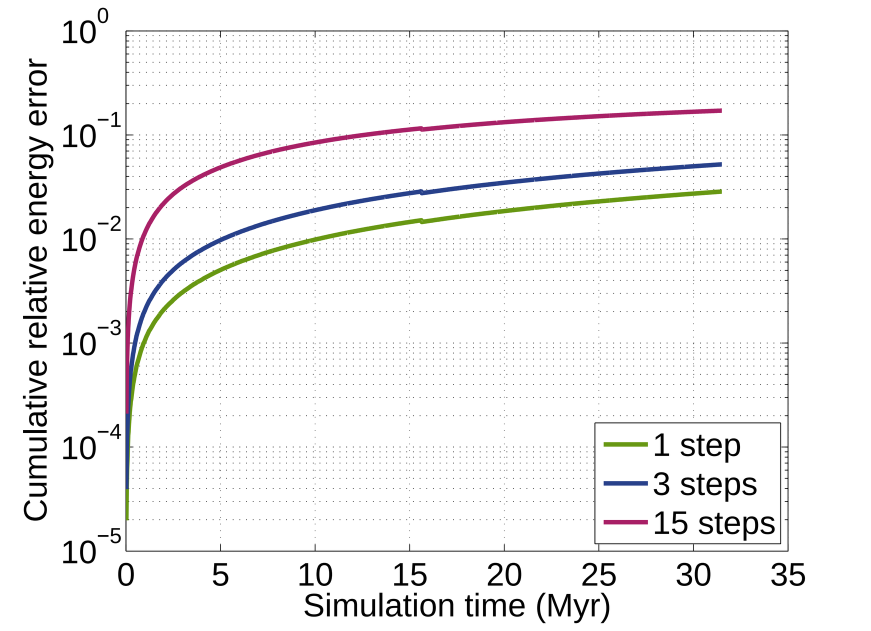
	$p=4$				$p=9$			
	Octant		**Plum.**		**Octant**		**Plum.**	
N	*skr*	*exa*	*skr*	*exa*	*skr*	*exa*	*skr*	*exa*
10	2	3	5	6	5	7	10	14
20	4	6	10	13	11	13	20	27
40	8	13	19	25	21	26	40	52
80	14	23	30	43	40	51	75	101
160	28	49	58	90	82	102	151	201
320	49	103	107	189	159	214	299	502

N: # of particles (M); **skr**: Sakura execution; **exa**: ExaFMM execution
t_c: time for combinatorial construction; t_n: time for numerical evaluation of FMM

– FMM skeleton construction cost ratio reduced by 2.5–3.5×
– total FMM time reduced up to 2× for larger data-sets

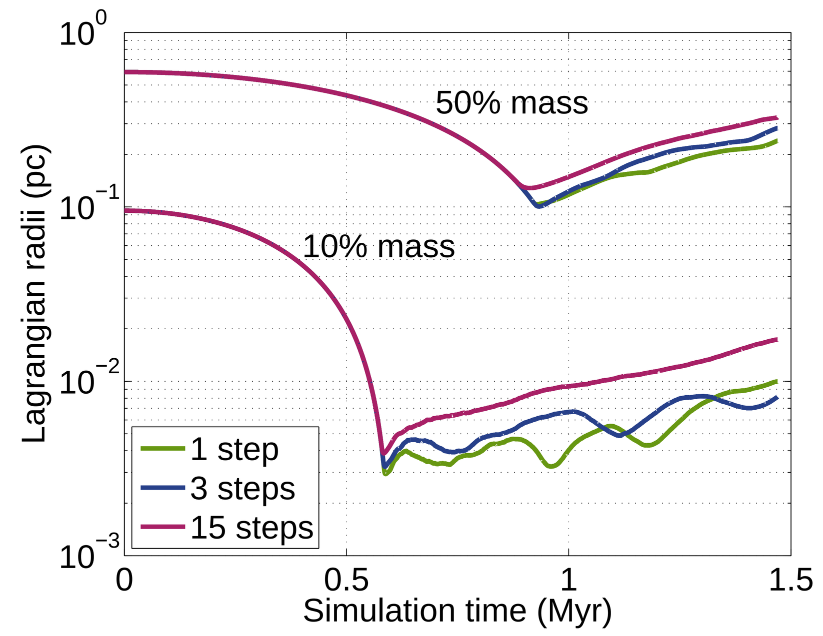
Plummer sphere evolution

– cumulative relative energy error: $\Delta E_t = \frac{E_t - E_0}{E_0}$
– E_t: total (kinetic + potential) energy at time t
– # particles: $10M$; # time steps: 3000; $\Delta t = 1kyr$



Plummer sphere core collapse

– Lagrangian radii: minimal radii which enclose a certain portion of the total system mass
– # particles: $10M$; # time steps: 25000; $\Delta t = 100yr$

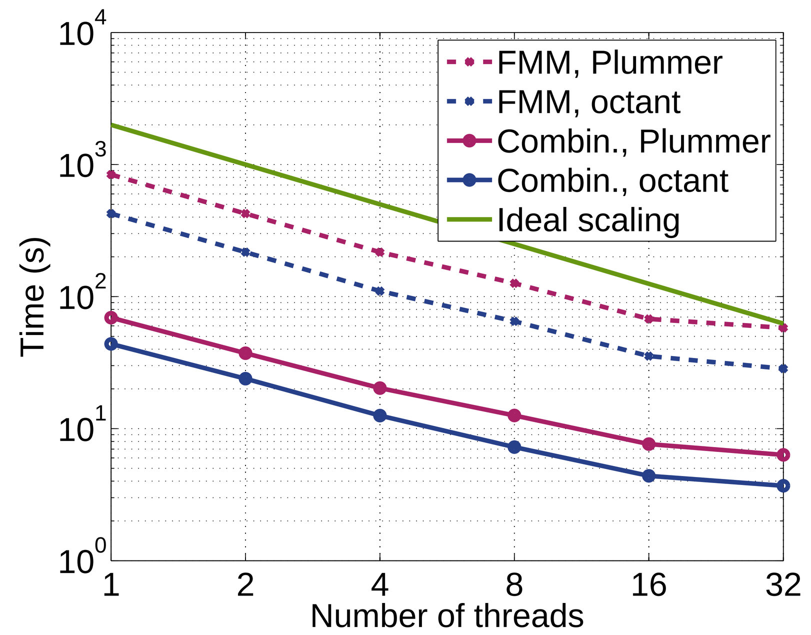


– error increases with less frequent FMM skeleton updates
– execution time: 3-step updates with Sakura ≈ 15-step updates with ExaFMM

Strong thread scalability

– # particles: $160M$
– $p = 4$

– FMM skeleton construction scales just as well as the FMM computations
– both scale almost ideally up to 16 threads (# of cores)



LaTeX TikZposter