Streaming Algorithms for Halo Finders

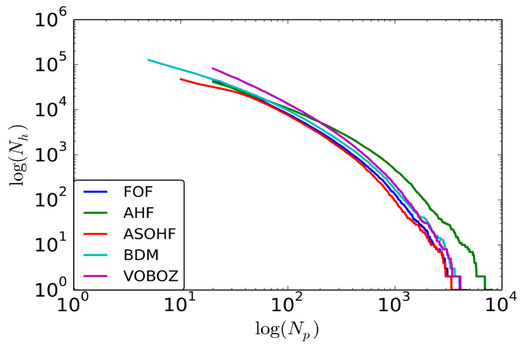
Zaoxing Liu , Nikita Ivkin , Lin F. Yang , Mark Neyrinck , Gerard Lemson, Alexander S. Szalay,
Vladimir Braverman, Tamas Budavari, Randal Burns, Xin Wang
Johns Hopkins University, Baltimore MD

Data

- Cosmological Simulation provides positions of particles.
- Halo – macro structure with high mass concentration.
- Finding haloes is crucial to connect theory to observation.
- Some facts about data:

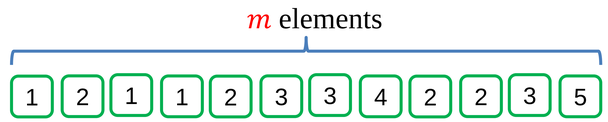
# of particles	# of haloes	# of particles in haloes
~ 10^{12}	~ 10^{9}	~10%

Distribution of Halo sizes



N_h – number of haloes
N_p – number of particles in halo

Streaming Model



Stream:

- m elements from dictionary of size n
  e.g. D=$\{x_1,x_2,\dots,x_m\}$ = 3 5 3 7 5 4 …

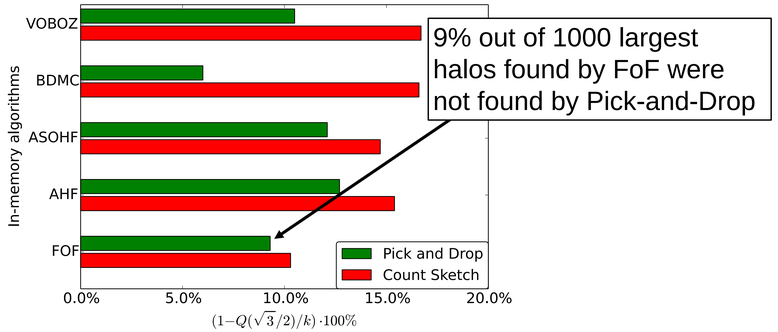
Goal:

- Compute the function of stream (e.g. k most frequent items) in sublinear memory.
- Approximate answer with high probability is OK.

Heavy Hitter Algorithms

Algorithms implemented:

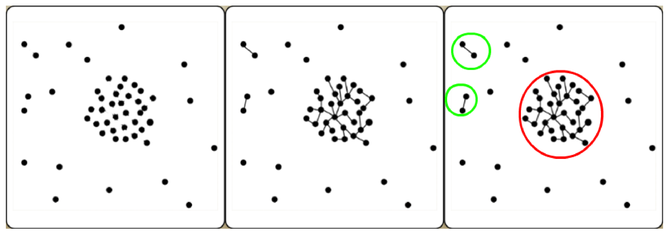
- Count Sketch algorithm: ~ 1GB
- Pick-and-Drop algorithm: ~ 30MB
- Friends-of-Friends: ~12GB



Previous Methods to find Haloes

- Current solutions require to load all the data into memory (~12TB).

Huge memory!



Close particles are friends.
A halo is a group of friends.

Friends-of-Friends Algorithm (FOF)

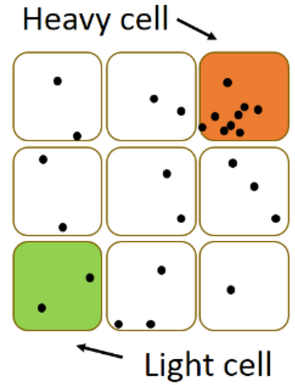
Streaming Solutions

haloes ≈ heavy hitters?

Heavy hitters – most frequent items in the stream.
Naïve solution is to use 3D mesh:

1. Particle ⟶ Cell ID.
2. Heavy cells ≈ Haloes.

Use any heavy hitter algorithm as black box



Results

- Connection between haloes and heavy hitters.
- Two streaming algorithms for finding top-k haloes with 90% accuracy.
- Sublinear memory provides scalability.

Future Directions

- Extend result for large k.
- Consider 6-dimensional space (position + velocity).

Supported by NSF BigData #1447639