Streaming Algorithms for Astronomic Simulations

Vladimir Braverman, Alex Szalay
Johns Hopkins University

The goal of astrophysics is to explain the observed properties of the universe we live in. In cosmology in particular, one tries to understand how matter is distributed on the largest scales we can observe. In this effort, advanced computer simulations play an ever more important role. Simulations are currently the only way to accurately understand the nonlinear processes that produce cosmic structures such as galaxies and patterns of galaxies. Hence a large amount of effort is spent on running simulations modelling representative parts of the universe in ever greater detail. A necessary step in the analysis of such simulations involves locating mass concentrations, called "haloes", where galaxies would be expected to form. This step is crucial to connect theory to observations – galaxies are the most observable objects that trace the large-scale structure, but their precise spatial distribution is only established through these simulations.

Many algorithms have been developed to find these haloes in simulations. The algorithms vary widely, even conceptually. There is no absolutely agreed-upon physical definition of a halo, although all algorithms give density peaks, i.e. clusters of particles. Galaxies are thought to form at these concentrations of matter. Some codes find regions inside an effective high-density contour, such as Friends-of-Friends (FoF). In FoF, particles closer to each other than a specified linking length are gathered together into haloes. Other algorithms directly incorporate velocity information as well. FoF is often considered to be a standard approach, if only because it was among the first used, and is simple conceptually. The drawbacks of FoF include that the simple density estimate can artificially link physically separate haloes together, and the arbitrariness of the linking length.

Halo-finding algorithms are generally computationally intensive, often requiring all particle positions and velocities to be loaded in memory simultaneously. In fact most are executed during the execution of the simulation itself, requiring comparable computational resources. However, in order to understand the systematic errors in such algorithms, it is often necessary to run multiple halo-finders, often well after the original simulation has been run. Also, many of the newest simulations have several hundred billion to a trillion particles, with a very large memory footprint, making such posterior computations quite difficult. Here, we investigate a way to apply streaming algorithms as halo finders, and compare the results to those of other algorithms participating in the Halo-Finding Comparison Project.

Recently, streaming algorithms have become a popular way to process massive data sets. In the streaming model, the input is given as a sequence of items and the algorithm is allowed to make a single or constant number of passes over the input data while using sub-linear, usually poly-logarithmic space compared to the storage of the data. We propose applying streaming algorithms to the area of cosmological simulations and provide space and time efficient solutions. We already demonstrated a relation between the problem of finding haloes in the simulation data and the well-known problem of finding "heavy hitters" in the streaming data. This connection allows us to employ efficient heavy hitter algorithms, such as Count-Sketch and Pick-and-Drop Sampling. By equating heavy hitters to haloes, we are implicitly defining haloes as positions exceeding some high density threshold. In our case, these usually turn out to be density peaks, but only because of the very spiky nature of the particle distributions in cosmology. Conceptually, FoF haloes are also regions enclosed by high density contours, but in practice, the FoF implementation is very different from ours. We plan to extend the usage of streaming algorithms to other important problems in astronomic simulations of large scale.