Visual Exploration of Large Scale Particle-Based Cosmological Data Sets

1 Introduction

Recent advances in the area of Cosmological simulations are allowing application scientists to enhance their knowledge about the Universe and its continuous evolution through space and time. In order to model the phenomena with greater accuracy, scientists are designing their simulations with higher precision and resolution. Numerically accurate results obtained from such simulations enable the scientists to look into the details of specific phenomena and develop a deeper understanding. The simulations attempt to model the structures in the universe by modeling dark matter, a collisionless fluid that only interacts with other objects gravitationally. In the simulation, dark matter is modeled as a very large number of discrete particles to fulfill the constraints of adequate mass. For efficient identification of the dark matter structures, a technique called halo finding is used. A halo is a cluster of dark matter particles where the clustering is based on several domain specific constraints. By finding the halos, the structures and sub-structures of the galaxies and their hierarchies can be studied in great detail. However, it is to be noted that, the benefits of these high precision simulations come with several non-trivial challenges requiring novel algorithmic solutions. In this write-up, we address two problems and propose possible solutions to efficiently tackle each of them.

The existence of the large number of particles in the data inevitably poses significant challenges in visualizing the useful data effectively without occlusion. The first task in the contest acknowledges the basic need of designing an interactive visualization system. This system will take the extracted meaningful information from the data and present them efciently at the user's request. The primary aim of such a system is to provide the scientists a centralized tool where they can intuitively hypothesize and refine their queries and quickly visualize the filtered data in great detail. Therefore, we choose task 1 as our first goal. Next, to study the temporal nature of the data, we will delve into the task 2 which is associated with the extraction and tracking of the halos over time. Tracking halos presents some unique challenges. The halos in the cosmological data usually follow a hierarchical structure, i.e. bigger halos can contain multiple small halos and as time progresses, this hierarchy can change significantly. The existence of the large number of halos makes this problem more difficult. Furthermore, these halos can merge, split, appear and disappear. Keeping track of all of these evolutionary events is imperative and non-trivial.

In this work, we aim to build an interactive visualization system for the application scientists for efficiently exploring the cosmological data. Keeping in mind the large number of halos, we are proposing to categorize all the halos based on several domain specific attributes. A refinement based strategy can be used where the halos which show similar temporal trend based on certain criteria will be grouped together. This will allow scientists to have a reduced search space and they will be able to interactively investigate each group of halos separately. We would also like to create an abstract visualization interface of all the halos in a group where the users can visualize the high level temporal trend of the halos and pick interesting halos for a more detailed analysis. For efficient tracking of the halos, we will use the information given in the form of a merger tree and devise additional attributes which will make tracking the halos more accurate. Finally, we can visualize the halos using appropriate visualization techniques. Since identification of the sub-halos is important, we would like to first identify the particles that form each of such sub-halos and then create a 3D convex hull which will allow us to estimate the shape of the halos. Finally, we can measure the intersection of the convex hulls across consecutive time steps to detect events like merge, split etc. If this technique gives us a robust method for tracking the halos, we would also like to make an animation based visualization where the temporal evolution of specific halos can be visualized effectively through the changes in their convex hulls.

2 Data Description

The cosmological data in general are represented by their volume and the large number of discrete particles. The particles are used to discretize the mass in the simulation. These simulations try to model the gravitional interactions among dark matter particles by simulating the interaction using *gravitational N-body integrators*. The data provided in this contest are derived from the **Dark Sky Simulations** [4]. The data set consists of three primary type of data: (1) raw particle data, (2) a halo catalog, and (3) A merge tree database which links all the halo catalogs.

1. **The raw particle data:** The particle data set provided here is stored using SDF files (self describing format). SDF files are binary files which contain a header followed by the raw data values. The data contains a unique id, position in 3D space, velocity, and an acceleration for each particle. The header also contains information about a particle's mass, the total number of particles, and units for different quantities.

2. **The halo catalog:** The halo catalog files describe the properties of each halo for individual time steps. It contains an id for each halo which is unique across all time steps. The halo catalog is

stored in ASCII format and each line in the halo uniquely describes a halo. Different attributes are listed for each halo, such as position, size, and acceleration.

3. **The merge tree database:** The merge tree database contains information regaring the temporal evolution of the halos. These files are also stored in ASCII format and is a product of combining all the halo catalog files. From the merge tree database, the spatial and temporal hierarchy of individual halos can be identified. The information obtained from this tree database will be critical in devising an efficient tracking algorithm.

3 Detailed Description of the Tasks

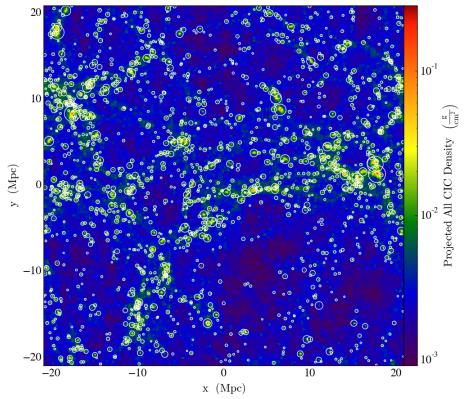


Figure 1: 2D projected halos in X-Y plane.

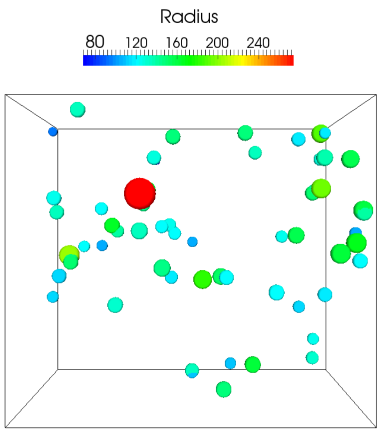


Figure 2: Detected Halos in a time step. Size and color reflects radius of the halos.

In this section we elaborate the tasks in more detail and discuss the approaches that we are going to take to find solutions. The tasks we have chosen are listed as follows: (1) Initial data integration and browsing, (2) Halo identification and visualization.

3.1 Initial Data Integration and Browsing

Since the data set consists of three separate formats, the first aim of this task is to make a link among these three types of information and provide a visualization tool which will allow flexible and efficient browsing of the data. The tool should enable users to query for the particles and halos based on user specified the values for certain attributes, such as velocity, mass, acceleration, halo id, etc. Since the total number of particles and halos are significantly large, visualizing everything together will create visual clutter and the visualization will be inefficient. In Figure 1, generated using the YT package[5], we show the 2D projection of all the halos into the X-Y plane and the halos are identified by white circles. The more dense halos appear in yellow to red. Even though this image shows the overall halo distributions, they lose the 3D context and we can not investigate specific halos. Therefore, we want to provide functionality to filter and create subsets of the data so that the exploration remains tractable and meaningful. The users should be able to tune their search space based on the results they get from the initial query. We are planning to use glyph-based visualizations for the filtered halos which will help to visualize their shape in 3D.

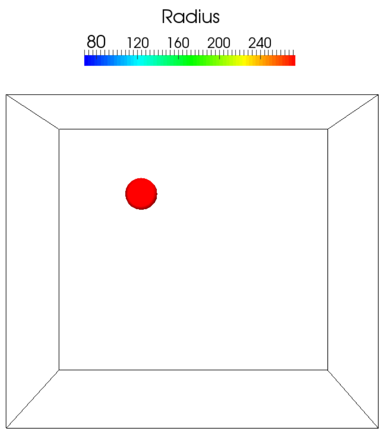
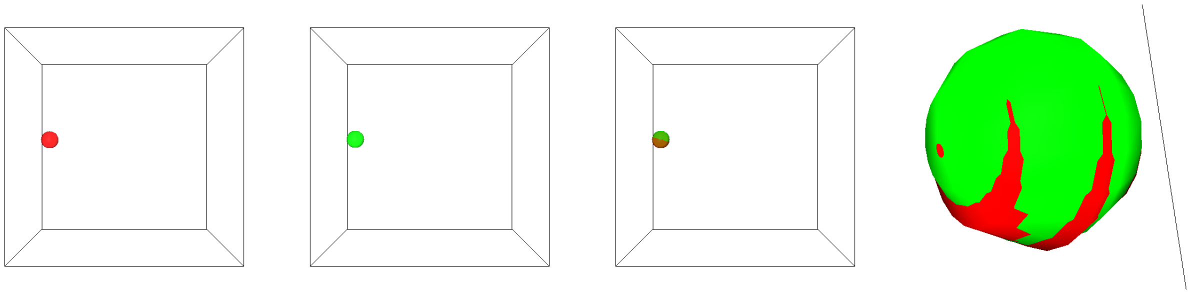


Figure 3: A specific halo is picked after applying thresholding on the radius values.

Furthermore, the size and color of the halos can be modulated based on user specified attributes which will convey more information. In Figure 2 we show the halos of a time step where the size and color of the halos are modulated by their radius. However, we are showing the halo here by a spherical glyph which may not be the actual shape of the halo. Therefore, we want to find the particle set of that formed this halo and visualize the convex hull of the particles which will be a more accurate visualization. In Figure 3, we show a specific halo picked based on thresholding on the radius values. This will allow users to focus on a specific halo and investigate its structure and time-varying nature.

3.2 Halo Identification and Visualization

From the given data in halo catalog and merger tree database, we can easily find out the total number of halos in each time step. Also, we can query the halo radius, the velocity of the halo and other attributes.



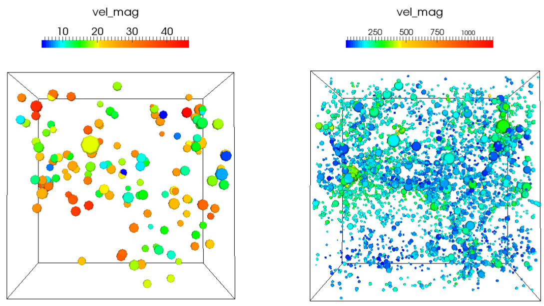
(a) Isolated halo at T=50. (b) Isolated halo at T=51. (c) Overlaped view of the isolated halo. (d) Zoomed in view of the overlapped halo.

Figure 4: Isolated halo in two consecutive time steps.

Figure 3 shows the result of a query based on the halo's radius, which yielded a single halo. However, to analyze the individual halos in detail we need more detailed information. We need to find which particles fall inside a halo and detect the sub-halos as well. This can be achieved by finding all the particles that fall inside the sphere of the halo. The halo catalog provided with the data was originally generated by a *friends-of-friends* (FOF) based algorithm called **ROCKSTAR** (Robust Over-density Calculation using K-Space Topologically Adaptive Refinement) [6]. First, the algorithm applies a 3D FOF method for finding the dense regions in the data based on some constraints. Next, the algorithm gradually builds the hierarchy information among those detected regions. Finally, when the algorithm is finished, the halo attributes are computed and stored for later use. In this work, we want to investigate this FOF based halo finding algorithm and attempt to enhance its capabilities to extract more information regarding the halos and enrich the analysis.

The temporal evolution of these halos are also important to analyze. Since the halos undergo complex evolutionary events such as merging, splitting, forming, and dissipating, tracking their behavior over time poses significant challenges. To track a halo over time, we have to find the same halo in each of the consecutive time steps. This specific problem is known as the *correspondence problem* in tracking literature. Previous studies in the literature have provided several methods for solving this problem [8, 9, 12] and we would like to adapt a similar method that can be used to accurately track the halos over time. A recent work by Sauer et al. proposed a method for tracking particle data sets [10]. We would like to enhance their technique for efficient halo tracking. The information given in the form of a merger tree database will be to be used in this context to make the tracking more accurate.

From a conceptual viewpoint, merger trees simply represent many halos and track which time steps these halos existed during the simulation. Typically, when a halo is first identified, this halo is called a progenitor [7]. In successive time steps, if a halo contains mostly the same particles as a halo in the previous time step, this halo is called a descendant and is linked to the progenitor in the merger tree [7]. Now, we can schematically visualize a representation of how this data might be stored. Figure 6 describes a way that one could visualize a merger tree. In Figure 6, we can see that the initial Progenitor_1 node is a halo that was identified as a unique halo in one of the earlier time steps. Each Descendant node is then appended to the list that holds the Progenitor node that this Descendant is associated with. This is the idea that Figure 6 is attempting to capture by having an arrow point to the next descendant over time.



(a) halos at T = 13. (b) halos at T = 50.

Figure 5: Halos at an initial and a later time step. The number of halos has increased significantly. The halos are colored by their velocity magnitude.

However, it is important to note that ROCKSTAR Halo Finder may have some issues when detecting halos due to the simulation and behavior of the particles [7]. For example, halos may exist in one time step and then move closer to a larger halo. The algorithm may mistake two halos as one large halo, resulting in the loss of one of the halos for this time step. The halos may then be distinct again in a later time step, resulting in two progenitor halos when there should have only been one progenitor. Other issues with the halo finders include misidentifying particles as part of the incorrect halo, large halos seemingly appearing and then disappearing, and duplication of halos through other methods. Behroozi et al. in [7] proposed an algorithm that would help decrease the problems with the ROCKSTAR Halo Finders. For example, in one time step, a halo may be present. In the next time step, the halo might disappear and actually be incorrectly identified as a new halo or be merged with another halo. The work suggests that the halo can be predicted from the previous time step. Thus, halos would no longer disappear and reappear. We would like to investigate this approach and try using it in this work for achieving better accuracy.

4 Software

For building our exploration system, we are planning to use both C++ and python. In Python specifically we are planning to use the package Yt [5] which has a specific module dedicated to cosmological data analysis. Yt allows efficient loading of the data and querying on them. For preliminary tests, Paraview [1] will be used as a visualization tool

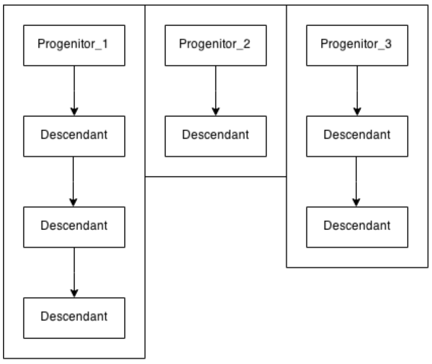


Figure 6: A conceptual view of a Merger Tree.

as well. A study of cosmological data sets using Paraview was presented earlier in [13]. We are also going to use a C++ library called Self-Describing File (SDF) Library (libSDF) [3] for processing the particle data in C++. We are expecting the end product to be a C++ based interactive visualization system. For rendering purpose we are going to use the open source visualization library The Visualization Toolkit (VTK) [11] and for creating the user interface, we will be using QT [2].

5 Visualization Design

The goal of our visualization tool is to provide the users an intuitive interface which can be used to interactively query the data and visualize it effectively. Since, we have mentioned earlier that for browsing the halos in general, we want to separate them into groups of similar behavior, we want to have an appropriate visualization where the users will be able to pick any group they want and then pick specific halos from the group. An abstract view of all the halos in the form of an interactive chart will be devised where the users will be able to interactively select important halos. The importance of the halos will be quantified based on the domain specific halo attributes such as radius, velocity, acceleration etc. The shape of the selected halos in the data domain can be visualized either by using glyphs or using the convex hulls. We believe that the use of convex hull will reflect a more accurate shape of the halos. Also, once all the tracking information is computed from the merger tree, we would like to present an animated view of the tracked halo in the data domain using their convex hulls. This will accurately show how the halos merge and split in space.

6 Expected Results

So far we have made some progress towards achieving our goals. We have downloaded the data from the contest website and studied the three different types of formats provided in the data. We have also written a few small programs which allows querying of specific particles and their specific attributes from the raw sdf files. In order to visualize the halo using spherical glyphs, we have converted the given file into a csv format and used VTK to render them as shown in Figure 2 and Figure 3. Preliminary visualization results have helped us to better understand the nature of the halos and the particle data. In Figure 4 we show another preliminary test result where we are focusing on a specific halo. In Figure 4a and Figure 4b we show the halo at time step 50 and 51 respectively. In Figure 4c we overlaid two halos to study thier changed in position. The halo in time step 51 demonstrates a small shift which is shown in Figure 4d by zooming into the halo. In Figure 5 we present the visualization of all the detected halos of two time steps. One from the beginning of the simuation at T = 13 (Figure 5a) and another (Figure 5b) from T = 50. The size of the halos reflect their radius and the color reflects the velocity magnitude. It can be observed that the number of halos has increased significantly in the later time steps.

After the completion of this project we are expecting to have an interactive visualization system. The halos are expected to be visualized efficiently with both an abstract view of all the halos and then specific halos in the data domain. Also, the tracking of halos will be able to be observed with an animated view where either glyphs or convex hulls will be used to show the detected merge and split phenomena.

7 Conclusion

In summary, we have proposed an outline for exploring the particle-based cosmological data sets. Two specific tasks (1) and (2) are discussed in this description and some possible directions of solving them have been discussed. We would like to continue work on this project and make necessary enhancements to this proposal to make our project better and effective. Even though we have not talked about the scope of parallelization of this work, however, we are keeping this point in mind and will try to devise our algorithm in such a way that the part of the method can be parallelized and we will exploit the parallel processing in those steps.

References

[1] ParaView: Parallel Visualization Application. http://www.paraview.org/.
[2] Qt Graphical User Interface Framework. http://qt-project.org/.
[3] Self-Describing File Library. https://bitbucket.org/JohnSalmon/sdf.
[4] The Darksky Simulations. http://darksky.slac.stanford.edu/.
[5] The Yt Project. http://yt-project.org/.
[6] P. S. Behroozi, R. H. Wechsler, and H.-Y. Wu. The rockstar phase-space temporal halo finder and the velocity offsets of cluster cores. *The Astrophysical Journal*, 762(2):109, 2013.
[7] P. S. Behroozi, R. H. Wechsler, H.-Y. Wu, M. T. Busha, A. A. Klypin, and J. R. Primack. Gravitationally consistent halo catalogs and merger trees for precision cosmology. *The Astrophysical Journal*, 763(1):18, 2013.
[8] F. Reinders, F. H. Post, and H. J. W. Spoelder. Attribute-based feature tracking. In *Data Visualization 99*, pages 63–72. Springer Verlag, 1999.
[9] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. Visualizing features and tracking their evolution. *Computer*, 27:20–27, 1994.
[10] F. Sauer, H. Yu, and K.-L. Ma. Trajectory-based flow feature tracking in joint particle/volume datasets. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints):1, 2014.
[11] W. Schroeder, K. Martin, and B. Lorensen. The Visualization Toolkit. *Kitware*, 2006.
[12] J. Takle, D. Silver, E. Kovacs, and K. Heitmann. Visualization of multivariate dark matter halos in cosmology simulations. In *Large-Scale Data Analysis and Visualization (LDAV), 2013 IEEE Symposium on*, pages 131–132, Oct 2013.
[13] J. Woodring, K. Heitmann, J. Ahrens, P. Fasel, C.-H. Hsu, S. Habib, and A. Pope. Analyzing and visualizing cosmological simulations with paraview. *The Astrophysical Journal Supplement Series*, 195(1):11, 2011.