Evolving N-Body Simulations to Determine the Origin and Structure of the Milky Way Galaxy’s Halo using Volunteer Computing

Travis Desell, Malik Magdon-Ismail, Boleslaw Szymanski, Carlos A. Varela  
Department of Computer Science  
Rensselaer Polytechnic Institute  
Troy, NY, USA  
deselt,magdon,szymansk,cvarela@cs.rpi.edu

Benjamin A. Willett, Matthew Arsenault, Heidi Newberg  
Department of Physics, Applied Physics and Astronomy  
Rensselaer Polytechnic Institute  
Troy, NY, USA  
willeb,arsenn2,heidi@rpi.edu

Abstract—This work describes research done by the Milky-Way@Home project to use N-Body simulations to model the formation of the Milky Way Galaxy’s halo. While there have been previous efforts to use N-Body simulations to perform astronomical modeling, to our knowledge this is the first to use evolutionary algorithms to discover the initial parameters to the N-Body simulations so that they accurately model astronomical data. Performing a single 32,000 body simulation can take up to 200 hours on a typical processor, with an average of 15 hours. As optimizing the input parameters to these N-Body simulations typically takes at least 30,000 or more simulations, this work is made possible by utilizing the computing power of the 35,000 volunteered hosts at the Milky-Way@Home project, which are currently providing around 800 teraFLOPS. This work also describes improvements to an open-source framework for generic distributed optimization (FGDO), which provide more efficient validation in performing these evolutionary algorithms in conjunction the Berkeley Open Infrastructure for Network Computing (BOINC).

I. INTRODUCTION

MilkyWay@Home has used a probabilistic sampling method to measure the shape of stellar substructure in the Milky Way, primarily from tidal streams, stars that have been tidally stripped from dwarf galaxies as they are pulled apart by the Milky Way’s gravity [1], [2]. This method simultaneously fits a smooth component of the Milky Way’s stellar halo that is presumably the result of galaxy mergers that occurred early in the formation of the Milky Way, along with these disrupted dwarf galaxy stars around the entire galaxy. Since the Milky Way galaxy is the only galaxy for which it is possible to measure the positions and velocities of stars in three dimensions, our galaxy provides important clues to the mechanisms through which galaxies form and the nature of dark matter.

However, this approach has some limitations. As an accurate model of the smooth component (or the background model) is unknown, models of the disrupted dwarf galaxy stars can end up fitting errors in the background model. Additionally, the models generated from this approach only provide information about where these tidal streams are, not why they are there. This work addresses these deficiencies by using N-body simulations to model this tidal disruption of dwarf galaxies and their interaction with the Milky Way. The disruption depends on initial properties of the dwarf galaxy and on the gravitational potential of the Milky Way, which is primarily due to dark matter. Asynchronous evolutionary algorithms (AEAs) have been used successfully to find the optimal input parameters for the probabilistic sampling method [3], [4], [3], [5], [6], and this work expands these AEAs to find the optimal input parameters for the N-body simulations, which will in turn provide new understanding of the origin and structure of the Milky Way.

Successfully evolving these N-body simulations using a volunteer computing system such as MilkyWay@Home involved three main challenges. First, as volunteer computing systems consist of highly heterogeneous hosts with unreliable availability, that are potentially malicious, existing N-body simulation code had to be modified to enable checkpointing so applications can stop and restart and provide the same results across multiple platforms. Second, as the N-body simulation code is the second real scientific application being optimized using FGDO, this work has made steps to improve the generality and usability of that framework. Further, the validation strategy used by FGDO has been updated, improving its robustness while reducing validation overhead. Lastly, a method for determining how accurately an N-body simulation represents astronomical data gathered by various sky surveys, such as the Two Micron All Sky Survey (2MASS) [7] and the Sloan Digital Sky Survey (SDSS) [8], was required so that their fitness could be optimized by the AEAs.

These N-body simulations are extremely computationally expensive as a single 32,000 particle simulation can take up to 200 hours on a standard processor. The massive amounts of computing power produced by the volunteer computing hosts at MilkyWay@Home provide one of the few computing systems where performing such optimization in a realistic amount of time is possible. Preliminary results show that for a test data set with known optimal parameters, the
35,000 volunteered computing hosts at MilkyWay@Home can be successfully harnessed to evolve 4,096 and 32,768 particle N-body simulations to accurately model the Milky Way Galaxy and the formation of structure in its halo. Further, the FGDO framework is now successfully being used to optimize 32,768 particle N-body simulations to real observed data from the SDSS.

II. RELATED WORK

A. Distributed Evolutionary Algorithms

Evolutionary algorithms (EAs) are a popular approach for parameter optimization where the search space contains many local optima that traditional search methods such as gradient descent and simplex get trapped in. As the search space for these N-body simulations is highly complex, EAs are an ideal candidate for performing the parameter optimization.

Common EAs for continuous search spaces include differential evolution (DE) [9], particle swarm optimization (PSO) [10] and genetic search (GS). In general, an EA keeps track of an population of potential solutions, where each individual in the population represents a set of parameters in the search space and has a fitness that represents how good of a solution that individual is. As the EA progresses, new individuals are generated by recombining individuals in the current population, and those with higher fitnesses are kept while those with lower fitnesses are discarded. As the generation of new individuals involves random elements, newly generated individuals have the potential to both explore new regions of the search space, and exploit areas of the search space that are known to have good fitness. This results in the population of solutions evolving towards an optimal solution.

There have been many different approaches to making EAs work on different distributed computing systems. In general these approaches are either sequential [11], [12], with distinct synchronization points; partially asynchronous, with few distinct synchronization points [13], [14]; or hybrid methods [15], [16], [17], [18], [19]. FGDO supports fully asynchronous versions of differential evolution, particle swarm optimization and genetic search, which differ from previous approaches as they remove all explicit synchronization points. This allows these optimization methods to scale to hundreds of thousands or more computing hosts, as shown by Desell [4], making them ideally suited for volunteer computing.

B. N-Body Simulations in Astroinformatics

N-body simulations are a well established tool for modeling tidal disruption in the Milky Way. The Sagittarius Dwarf Tidal Stream was initially modeled by Johnston et al. [20], and followed up by Law et al. [7]. While this placed some constraints on the kinematics of the Sagittarius dwarf, Law et al. were unable to simultaneously fit the kinematics and sky positions of the Sagittarius stream within an axisymmetric dark matter halo. Only by expanding to a triaxial halo did Law & Majewski [21] satisfy all constraints. Predating this work, Dehnen et al. [22] modeled the Palomar 5 globular cluster tidal stream via N-body simulations and showed that the majority of its properties were results of its orbital kinematics. Similar studies of the GD-1 (Grillmair & Dionatos [23] stellar stream by Willett et al. [24] and Koposov et al. [25] were able to determine orbital kinematics, but did not perform N-body simulations.

While these studies were groundbreaking in their ability to constrain tidal streams, they did not address the interesting research question: can N-body simulations be used to rigorously fit the stellar density along a tidal stream? Newberg et al. [26] published a re-analysis of the Orphan Stream (Belokurov et al. [27], Grillmair [28]), and extracted the density of Orphan Stream F-turnoff stars as a function of Orphan Stream longitude $\lambda_{\text{Orphan}}$, which is shown in Figure 1. Newberg et al. [26] were able to reproduce the overall form of the Orphan density using a Plummer model with mass $M_P = 2 \times 10^6 \ M_\odot$ (where $M_\odot$ is the mass of the sun), scale length $r_a = 0.2 \ kpc$ (kiloparsecs), orbit time $t_{\text{back}} = 4 \ Gyr$ (gigayears) and evolution time $t_{\text{back}} = 3.945 \ Gyr$, evolved along the best fit orbit within a Galactic potential. While these parameters produce a model that broadly reproduces the Orphan Stream density, an interesting research question emerges: can an N-body model of the Orphan Stream progenitor actually be fit to the Orphan stream density?

III. MODELING THE MILKY WAY GALAXY USING N-BODY SIMULATIONS

The scientific purpose of this work is to utilize the BOINC volunteer computing environment to perform distributed gravitational N-body simulations of dwarf galaxies orbiting the Milky Way. A dwarf galaxy is a small spherical galaxy that typically possesses millions of stars and has a mass on the order of one ten-thousandth of the Milky Way’s mass. As it orbits our Galaxy, it becomes disrupted by gravity and forms tidal streams: long arms of stars that can span the entire sky. Utilizing massive and well calibrated photometric surveys such as the Two Micron All Sky Survey (2MASS) [7] and the Sloan Digital Sky Survey (SDSS) [8], astronomers have identified tens of streams orbiting the Milky Way. Figure 1 shows a stellar density map of SDSS F-turnoff stars in the Milky Way halo from [26]. Darker areas indicate higher stellar density. There are two tidal streams in this Figure. The first runs nearly vertically from $l = 200^\circ$ to $l = 240^\circ$. This is the Sagittarius Dwarf Tidal Stream. The other, running horizontally at $b \approx 50^\circ$, is the Orphan Stream.

The physical problem in understanding tidal streams is that they represent the disordered state of the original dwarf galaxy: they have already been disrupted. How can
we determine the properties of the original dwarf galaxy that created the stream? The simplest way to resolve this difficulty is to understand the kinematics of the stream, propagate an orbit back in time to a previously ordered state, and propagate a collection of particles forward in time to the present day. We can understand the kinematic properties of the stream by determining its velocity and distance at various points along the sky. For most purposes, the radial velocity (velocity along the line of sight) is the only knowable velocity component. Some stars have known proper motions, which would allow other velocity components to be determined, but their errors are often so large as to preclude their use. Knowing the line of sight kinematics of and distance to the stream, we can use search algorithms to find the best fit three dimensional kinematics and background Galactic model [24].

With the orbit of the stream understood, we can now create a group of particles at some time in the past, place it on this orbit, and propagate it forward to the present day. The model from the group of particles is a Plummer Sphere, which is an energetically stable three dimensional spherical distribution [29]. This model has two parameters: its total mass $M$ and scale length $a$. In addition, we will consider two more parameters: $t_{\text{orbit}}$, the amount of time the orbit is evolved back in time, and $t_{\text{dwarf}}$, the amount of time the dwarf is evolved forward in time. In order to determine the parameters of the best fit model dwarf galaxy, we need some measure of how the stars are distributed in the stream. The density of stars along the stream as a function of angle on the sky provides this very measure.

The hashed histogram in Figure 2 shows the Orphan stream stellar density as a function of Orphan stream longitude ($\Lambda_{\text{Orphan}}$, a spherical coordinate system whose equator is along the stream). Note that the gap in the histogram around $\Lambda_{\text{Orphan}} = 25^\circ$ is the same gap in Figure 1 at $l = 260^\circ$, and thus is not a true absence of stars. We also see that the high density of stars near $(l, b) = (255^\circ, 49^\circ)$ corresponds with a peak in stellar density at $\Lambda_{\text{Orphan}} \approx +23^\circ$. We wish to determine the four parameters of the model dwarf galaxy that best fits the density profile given in Figure 2. Our metric for determining the goodness of fit to the density profile is given in Equation 1:

$$\chi^2 = \sum_i \left( \frac{\eta_i,\text{model} - \eta_i,\text{data}}{\sigma_i} \right)^2$$  \hspace{1cm} (1)

where $\eta_i = N_i/N_{\text{total}}$ is the normalized bin height, $\sigma_i = \sqrt{N_i/N_{\text{total}}}$ is the normalized error of a data bin, $N_i$ is the number of stars in bin $i$, and $N_{\text{total}}$ is the total number of stars in the histogram.

A. Proof of Concept

A proof of concept simulation can be generated by selecting a dwarf mass of $M = 1 \times 10^6 \, M_\odot$ (where
the cell is accepted for force evaluation if:

d > \frac{l}{\theta}.

(2)

Smaller values of \( \theta \) therefore give rise to more precise force evaluations. A \( \theta \) value of 1 typically results in acceleration errors of one percent compared to the full \( N^2 \) force algorithm [32].

Checkpointing has been implemented which allows the N-body simulations to be restarted when volunteers stop or pause the BOINC client that runs the N-body simulation. As the hosts at MilkyWay@Home are volunteered, this checkpointing minimizes the amount of work lost by volunteers using their computers.Checkpointing can be done after each timestep. Typically the BOINC client determines checkpointing is required every few minutes. The positions and velocities of the particles, as well as the simulation time are saved in a binary format. This information is sufficient to resume the simulation. The binary format ensures this is a lossless process, avoiding inconsistencies in string to floating point conversions present in nearly every compiler.

In addition, the treecode has been adapted to make it easier to add and use different initial distributions of particles for the dwarf model, such as Plummer models [29] as well as a selection of different components for an external acceleration due to the Milky Way, such as spherical bulges [33], exponential disks [34], and dark matter halos [7], [35]. This modified code has also been made freely available as a public repository on GitHub\(^2\).

IV. A FRAMEWORK FOR GENERIC DISTRIBUTED OPTIMIZATION (FGDO)

FGDO has made a series of improvements enabling its use for the optimization of N-body simulations in addition to the MilkyWay@Home’s probabilistic sampling application, making it more generic and easier to use with other computing projects. It is also available as public repository on GitHub for public use\(^3\). The new implementation has been written in Java, which allows for easier extension of the search methods being used because of Java’s Object oriented nature. In addition, Java has made it much simpler to plug in different credit and validation implementations for the different applications being used.

The previous implementation of FGDO could use either optimistic or pessimistic validation to reduce the amount of duplicate work done by volunteered hosts [36]. This previous implementation suffers from some drawbacks. First, it uses a fixed verification rate to determine the rate workunits are generated for validation. This has a significant impact on the speed that the search progresses, and a fixed rate is not an optimal solution. Additionally, many results are simply not validated as they will not potentially improve the population. This makes it easier for malicious hosts to cheat by reporting

\(^1\)http://www.ifa.hawaii.edu/~barnes/treecode/treeguide.html

\(^2\)http://github.com/Milkyway-at-home/milkywayathome

\(^3\)http://github.com/Milkyway-at-home/fgdo_java
bad results as there is a decent chance they will still receive credit for them when they do not need to be validated.

The new implementation solves these problems by combining optimistic and pessimistic validation with BOINC’s quorum and adaptive replication schemes (which are described in detail in [37]), as follows:

- When the queue of available work is low, new individuals are generated through recombination from the unvalidated population if the optimistic approach is used, otherwise they are generated from the validated population. Newly generated work has an initial quorum of one.
- When a result is reported for work with a quorum of size one and it cannot improve the validated population, it is validated with a chance equal to a host’s error rate. A host’s error rate is initialized to 0.1 (meaning 10% of its results will be validated). When a host returns a result that is validated successfully against another result, the error rate is multiplied by 0.95, to a minimum of 0.1. When a host returns a result that is invalidated against other results that match each other, its error rate is increased by 0.1 to a maximum of 1.0.
- When a result is reported for work with a quorum of size one and it can improve the validated population, FGDO will try to insert the individual into its unvalidated population. In addition, the quorum for this piece of work is increased to the amount specified by the project, which will cause the BOINC scheduler to send out copies of this work for verification.
- When results have been reported that potentially complete a quorum and enough results match to successfully determine a canonical result, the canonical result is inserted into the validation population. Any of these results that do not match the canonical result (and thus are invalid) are removed from the unvalidated population.
- When results have been reported that potentially complete a quorum, but not enough match to successfully determine a canonical result, the quorum size is again increased to allow the BOINC scheduler to generate more copies of this work for validation.

In this way, the user defined verification rate no longer required, as the BOINC scheduler will take care of the frequency in which duplicate work is sent out to hosts for verification and will try and generate it in a manner that verifies the most important results first (the ones with the shortest deadline). This allows the BOINC computing project to spend more time on verification when it is needed, and more time on exploration when not many results require verification. Additionally, it significantly reduces the amount of credits a malicious or broken host can gain by returning bad results by adaptively modifying how frequently a host’s results are verified based on their previous performance.

![Figure 4](image_url)

**Figure 4.** Progress of the best, average, median and worst validated individuals for an asynchronous differential evolution search over the search space $M_P = 0.22 \ldots 11.11 \times 10^6 M_{\odot}$, $r_s = 0.05 \ldots 1.0$ kpc, $t_{\text{orbit}} = 1 \ldots 5$ Gyr and $t_{\text{dwarf}} = 1 \ldots 5$ Gyr, with a population of 300 individuals. The fitness of these 4096 particle N-body simulations is calculated by comparing their stellar density histogram to the histogram of an N-body simulation with dwarf parameters $(M_P, r_s, t_{\text{orbit}}, t_{\text{dwarf}}) = (3.6 \times 10^6 M_{\odot}, 0.2$ kpc, 4 Gyr, 3.945 Gyr).

![Figure 5](image_url)

**Figure 5.** Simulated Orphan Stream stellar density modeled via Barnes & Hut treecode. The solid red histogram is the n-body simulation of the Orphan Stream orbit was performed using the parameters from Newberg et al. [26], dwarf parameters $(M_P, r_s, t_{\text{orbit}}, t_{\text{dwarf}}) = (3.6 \times 10^6 M_{\odot}, 0.2$ kpc, 4 Gyr, 3.945 Gyr). The dotted green histogram is the best fit found to this histogram using FGDO on MilkyWay@Home. $(M_P, r_s, t_{\text{orbit}}, t_{\text{dwarf}}) = (3.591 \times 10^6 M_{\odot}, 0.22$ kpc, 3.97 Gyr, 3.91 Gyr).

V. Results

A. Comparison to Known Test Data

As a test to see the potential for using AEs to optimize the parameters to these N-body simulations, a 4096 particle N-body simulation was performed using the parameters from Newberg et al. [26], dwarf parameters $(M_P, r_s, t_{\text{orbit}}, t_{\text{dwarf}}) = (3.6 \times 10^6 M_{\odot}, 0.2$ kpc, 4 Gyr, 3.945 Gyr). Asynchronous
The N-body simulations using only 4096 particles the initial seed to generate the initial particle distribution and with best fit individual found at the end of the search. For this search and Figure 5 compares the stellar density the progress of the individuals in the validated population distribution played a large factor in the final stellar density anywhere between -30 to -1200. However, in spite of this, the search space ended up being highly noisy. For example, using the best fit parameters found by generated seeds, the search space was generated from:

\[ M_p = 0.22 \ldots 11.11 \times 10^6 \, M_{\odot}, \quad r_s = 0.05 \ldots 1.0 \, \text{kpc}, \quad t_{\text{orbit}} = 1 \ldots 5 \, \text{Gyr} \text{ and } t_{\text{dwarf}} = 1 \ldots 5 \, \text{Gyr}, \]

with a population of 300 individuals. The N-body simulations consist of 32,768 particles. Model 1 utilizes an exponential disk and NFW halo profile with an enclosed mass of \( M_{60} = 40 \times 10^{10} M_{\odot}. \)

Differential evolution was then used with a population of size 300, best parent selection, and binary recombination with a crossover rate of 0.5, a pair weight of 0.5, or DE/best/1/bin (for more detail on differential evolution variants, see Mezura-Montes et al. [38]). The search space given was \( M_p = 0.22 \ldots 11.11 \times 10^6 \, M_{\odot}, \quad r_s = 0.05 \ldots 1.0 \, \text{kpc}, \quad t_{\text{orbit}} = 1 \ldots 5 \, \text{Gyr} \text{ and } t_{\text{dwarf}} = 1 \ldots 5 \, \text{Gyr}. \) Figure 4 shows the progress of the individuals in the validated population for this search and Figure 5 compares the stellar density histograms of the known test data to the parameters of the best fit individual found at the end of the search.

Some discrepancies arose because clients used a random seed to generate the initial particle distribution and with the N-body simulations using only 4096 particles the initial distribution played a large factor in the final stellar density model. As this initial distribution was due to randomly generated seeds, the search space ended up being highly noisy. For example, using the best fit parameters found by the search, different seeds resulted in fitness values from anywhere between -30 to -1200. However, in spite of this noisy search space, asynchronous differential evolution was able to find parameters quite similar to what the test data was generated from: \((M_p, r_s, t_{\text{orbit}}, t_{\text{dwarf}}) = (3.591 \times 10^6 \, M_{\odot}, 0.22 \, \text{kpc}, 3.97 \, \text{Gyr}, 3.91 \, \text{Gyr}),\) which also had very similar histogram to the test data (as shown in Figure 5). The authors feel that this shows that this approach is not only feasible, but highly robust.

B. Comparison to Actual Data

With the test simulations producing results with consistent parameters, fits are currently being run on the true Orphan Stream stellar density obtained from the SDSS sky survey. N-body simulations are being run within seven models of the Orphan Stream kinematics and Galactic potential from [26]. These models consist of the best fit orbits to the Orphan Stream kinematics in a variety of Galactic potentials. The aim of this work is to test the dependence of the Orphan Stream progenitor parameters (mass, scale length, and evolution time) on the various Galactic potential models. Dwarf parameters that depend on the Galactic potential could be a powerful probe into the structure of the Milky Way.

Figures 6 and 7 show the progress of a selection of these models after approximately a week being run on MilkyWay@Home. As the average N-body simulation time is around 15 hours on a typical computer, to achieve this amount of progress for single one of these searches would take over 5 years on a single computer. These results show that not only can MilkyWay@Home perform multiple N-Body simulation optimizations concurrently (while also computing it’s other optimization problem), it can also provide results in a reasonable amount of time for scientific progress.

These searches are being run with the same search space and search parameters as the test data, except the population size has been lowered to 100, as 300 was high for only 4 optimization parameters. Additionally, the size of the N-body simulations has been increased to 32,768 particles, which provides more accurate results with less variance based on the seed of the initial distribution. The various models interchange disk and halo gravitational potentials to find the best fit combination. They are characterized by their mass enclosed within 60 kpc of the Galactic center, \( M_{60}. \)
Model 1 utilizes an exponential disk and NFW halo profile with an enclosed mass of $M_{60} = 40 \times 10^{10} M_{\text{Sun}}$. Model 4 is a standard Galactic model, using a Miyamoto-Nagai disk and logarithmic halo and having an enclosed mass of $M_{60} = 47 \times 10^{10} M_{\text{Sun}}$.

The progress of these different searches shows that reducing the population size along with a larger number of particles in the N-body simulation has a dramatic effect on the convergence rates of the searches. In less than 4,000 evaluations the populations have already reached similar or better fitnesses to the 4096 particle test data N-body simulations after 150,000 evaluations. This means that the histogram made by the final state of the best N-Body simulations found matches the histogram of observed stars as well as or better than the histograms in Figure 5. The authors feel that this provides evidence that larger N-body simulations will be able to provide even better models of the Milky Way galaxy’s halo, and increasing the number of particles can potentially improve the number of evaluations required to reach a good fit as the noise due to the random seeding of the initial particle distribution in the search space decreases.

VI. Conclusion

This work presents preliminary results showing that a large scale volunteer computing project such as MilkyWay@Home can successfully evolve N-body simulations to model the formation of debris in the Milky Way galaxy’s halo. This was made successful by modifying existing N-body simulation code to allow for different galactic models and initial particle distributions, and using the CR-libm math library so that results are uniform across the over 35,000 heterogeneous volunteered computing hosts at MilkyWay@Home. In addition, this work has involved improvements to a framework for generic distributed optimization (FGDO), which can scalably run asynchronous evolutionary algorithms using BOINC with minimal validation overhead, and provides various tools for displaying the progress of these searches and finding errors in the applications run on volunteered hosts.

This work also opens up many avenues for future research. For example, the parameters of the different models being tested against the actual data could be optimized as well. Further, the type of initial particle distribution and type of model of the Milky Way could also become optimization parameters. This could lead to even more accurate representations of the formation of the Milky Way galaxy. In addition to providing a solution to the immediate problem of tidal stream modeling, this method can also be extended to other disciplines. Some examples include models of electromagnetic phenomena (namely charges in external fields), as well as molecules which bond to create organic compounds. The external potentials currently used are Galaxy specific but others can be easily added. Ultimately, any process that models the interactions of particles in an attempt to minimize or maximize a quantity can benefit from this method.

REFERENCES


