



CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

Grid-enabled tools to build halo merger trees

Deliverable ID: D-JRA2-2.3
Due date: **October, 31, 2005**
Actual delivery date: **November 28, 2005**
Lead contractor for this deliverable: **EPCC, UK**

Project start date: May 1st, 2004
Duration: 4 years

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document Keywords and Abstract

Keywords:	Grid-enabling, pre- and post-processing cosmology tools, group finders, Friends-of-Friends, FoF, SubFind, MergerTrees, Power, Correl, Virgo, GADGET, GADGET2
Abstract:	<p>This report describes the processes involved in porting, Grid-enabling and, where necessary, parallelising a number of pre- and post-processing tools employed by the Virgo Consortium. These are the three codes that make up the MergerTrees suite, namely: TraceSubgroups, SplitHalos and BuildTrees. The process of Grid-enabling these tools involved the incorporation of a portable binary data format. The format employed is HDF5.</p> <p>Specifically, this report is D-JRA2-2.2, and describes the work undertaken on the MergerTrees codes, in terms of the Specifications Document, D-JRA2-2.1</p>

Table of Content

1 INTRODUCTION	2
1.1 Executive Summary	2
1.2 References and Applicable Documents.....	3
1.3 List of Acronyms and Abbreviations	3
2 VIRGO’S COSMOLOGICAL SIMULATION CODES	4
2.1 GADGET.....	4
3 DESCRIPTION OF THE MERGERTREES TOOLS	4
3.1.1 TraceSubgroups	4
3.1.2 SplitHalos	5
3.1.3 BuildTrees	5
4 WORK DONE ON THE MERGERTREES SUITE.....	5
5 CONCLUSIONS	7

1 Introduction

1.1 Executive Summary

This project is DEISA's JRA2 – Cosmology Applications, and is a joint effort between EPCC and the UK's Virgo Consortium [1]. Funding for JRA2 comes from both DEISA and VirtU [7], where VirtU is Virgo's e-Science Virtual Universe project, funded by the UK Particle Physics and Astronomy Research Council (PPARC). VirtU started on the 1st October 2004 and will run for 36 months. It will form the foundations of the Theoretical Virtual Observatory.

All the tasks in WP2 are jointly funded by both DEISA and VirtU and, as such, all deliverables are reported to both DEISA and VirtU.

This report is the third deliverable of WP2, D-JRA2-2.3, "Grid-enabled tools build halo merger trees".

The input and output of one of Virgo's largest cosmological simulations codes, namely GADGET2 [8], are created and manipulated by a number of pre- and post-processing tools. GADGET2 has been ported to the DEISA infrastructure by JRA2's WP1 [1].

The main motivation of this work is that, once GADGET has been run on DEISA, albeit on a single platform [1], the post-processing tools can be run anywhere on the DEISA infrastructure, provided one has access to the data. This is possible due to the MC-GPFS, however, the data files must be stored in a portable data format.

The post-processing tools have been ported to the DEISA infrastructure, Grid-enabled and parallelised where necessary. Specifically, the tools ported were Friends-of-Friends (FoF), SubFind, MergerTrees (TraceSubgroups, SplitHalos and BuildTrees), Power and Correl. This report considers only the so-called MergerTrees suite, namely TraceSubgroups, SplitHalos and BuildTrees. The FoF and SubFind tools are considered in [3] whilst the Power and Correl tools are considered in [4]. The User Guide for all these tools is given in [5].

In this context, by Grid-enabling we mean providing these tools with the capability to use data sets generated within and outwith the DEISA infrastructure by introducing a portable binary data format. Thus members of the Virgo consortium can generate data outwith DEISA, transfer it to DEISA, run large simulations and then take the data back to their home institution and distribute it amongst collaborators without having to worry about how the binary is encoded. After a preliminary investigation, see below, the format chosen to do this was HDF5 [12]. The I/O of the tools previously described was changed to use HDF5.

HDF5 was chosen after the performance and usability of other portable binary data formats [3]. It was found that HDF5 performed the I/O faster than the other portable binary data formats, but also faster than the standard C `fwrite()` routine on a Sun and an IBM cluster.

The MergerTrees suite of codes, namely TraceSubgroups, SplitHalos and BuildTrees, have all been parallelised using a 2-dimensional slab decomposition, instead of the

embarrassingly parallel method as described in the Specifications Document [2]. In addition, their I/O routines were optimised, as described in detail for FoF in [3]. This new MergerTrees suite can only read and write using HDF5.

1.2 References and Applicable Documents

- [1] D-JRA2-1.2, Grid-enabled implementation of GADGET for metacomputing environments.
- [2] D-JRA2-2.1, Specification Document for D-JRA2-2.2, D-JRA2-2.3, D-JRA2-2.4.
- [3] D-JRA2-2.2, Grid-enabled “Group finders”.
- [4] D-JRA2-2.4, Grid-enabled tools for evaluating the basic properties of the dark matter.
- [5] D-JRA2-2.5, Documentation for D-JRA2-2.2, D-JRA2-2.3, D-JRA2-2.4.
- [6] Virgo: <http://www.virgo.dur.ac.uk>.
- [7] VirtU: <http://star-www.dur.ac.uk/~csf/virtU/virtU-final.pdf>.
- [8] GADGET: <http://www.mpa-garching.mpg.de/galform/gadget>.
- [9] DEISA: <http://www.deisa.org>.
- [10] Barnes Hut Treecode: Barnes, J., and P. Hut, *Nature*, 324, 1986.
- [11] Monaghan, J.J., *ARA&A*, 30, 543, 1992.
- [12] HDF5: <http://hdf.ncsa.uiuc.edu/HDF5>.

1.3 List of Acronyms and Abbreviations

GADGET	GA laxies with D ark matter and G as intEracT
MPI	M essage P assing I nterface
HDF5	H ierarchical D ata F ormat 5
BinX	B inary X ML D escription L anguage
VirtU	V irtual U niverse

2 Virgo's Cosmological Simulation Codes

The codes described in this document are all used as pre- and/or post-processing tools to the Virgo [6] Consortium's main Cosmological Simulation Code, namely: GADGET2 [8].

2.1 GADGET

GADGET is a code to simulate the evolution of the universe, modelling the motion of both dark matter and gas and, essentially, is a very large N-body simulation code. The code calculates the gravitational forces acting on both the dark matter particles and the gas particles in two separate calculations. Long range interactions are computed using an FFT, whilst a Treecode, similar to the Barnes-Hut Treecode [10], is employed for the short-range interactions. Further hydrodynamic forces for the gas component are evaluated using Smooth Particle Hydrodynamics (SPH) - see [11] for a good overview of SPH methods.

As part of JRA2, WP1, GADGET2 has been ported to the DEISA infrastructure. GADGET2 is the latest version of GADGET.

3 Description of the MergerTrees tools

GADGET2's output contains the positions, velocities, ID numbers and may contain other additional information (masses, density, star formation rate etc) for all of the particles in the simulation for a particular output time.

Output for a particular simulation time can be stored in a single file or the data can be split across several files, in which case each file contains the particles from some subsection of the simulation volume. These individual file(s) are usually referred to as 'snapshot files' and a complete set of files for one output time as a 'snapshot'.

To use the MergerTrees suite a large number (usually at least 50) of snapshots are needed.

Once the N-body simulation has been run, using GADGET2, and after the Friend-of-Friends (FoF) and SubFind codes have found the Friends-of-Friends halos and *subhalos* for each snapshot, the merger history of each dark matter halo need to be determined. This is achieved by taking the output of the FoF and SubFind codes and running the MergerTrees code suite.

The MergerTrees suite consists of three codes, namely: *TraceSubgroups*, *SplitHalos* and *BuildTrees*, which are run in sequence.

These three codes are parallel Fortran90 codes, use MPI, with some additional serial ANSI C codes used for I/O. *TraceSubgroups* has around 3000 lines, *SplitHalos* has 1500 lines, and *BuildTrees* has around 2000 lines.

3.1.1 *TraceSubgroups*

For each subhalo, at each output time, *TraceSubgroups* attempts to find the 'same' subhalo at the next output time by following the particles that belong to the original

subhalo. If the subhalo cannot be found at the next output time, the code may be able to find it at later output times.

For each subhalo, at each snapshot, this code outputs the ID number of the descendant subhalo and the associated snapshot (which is usually simply the next snapshot).

3.1.2 *SplitHalos*

The FoF code often merges objects that probably should be treated as independent halos in the merger trees, so certain criteria are employed to decide if any of the subhalos in a particular halo should be split off and treated as halos in their own right. These criteria are:

- 1) Whether the subhalo is within twice the half mass radius of the halo and
- 2) Whether the subhalo has been stripped of some fraction of its mass since it was last identified as a separate halo.

The output from SplitHalos is a data structure that defines a 'cleaned' halo catalogue.

3.1.3 *BuildTrees*

This code uses the 'cleaned' halo catalogue, as produced by SplitHalos, and the descendant IDs, as calculated by TraceSubgroups, to build the merger trees. Each halo at the final time is the 'root' of a merger tree. The program starts at the final output time and works backwards in time, adding each halo to whichever merger tree this descendant belongs to.

4 Work Done on the MergerTrees Suite

The Specifications Document, [1], stated that the three MergerTrees codes would be parallelised, if the codes can be parallelised in an embarrassingly parallel fashion, however, it was decided that this was too limiting, thus a more complex parallelisation was undertaken, and is now described in this Section.

The MergerTrees codes were designed to work with a version of SubFind not referenced before in either [1] or [3], namely L-SubFind. This is an embarrassingly parallel code and is intended to be used with L-GADGET2, which is a specialised version of GADGET2 intended for large, dark matter only simulations. L-GADGET2 has a built in Friends-of-Friends group finder. In this case each instance of L-SubFind reads one snapshot file and identifies all of the groups in the corresponding volume. So if the snapshot is split into n files, say, then L-SubFind may employ up to n processors to work on that single snapshot.

There is a complication here because a Friends-of-Friends group can span the boundary between snapshot files. The output from L-GADGET2 includes a hash table which allows one to quickly read in the particles from a particular region without reading the entire snapshot file. L-SubFind uses this to find the extra particles it needs to process groups which are split between snapshot files. However, the hash table is not included in the output from the standard GADGET2.

Now, FoF_Special_SubFind and P-GroupFinder, [3], generate one group catalogue per snapshot, irrespective of how many files the snapshot is spread across. L-SubFind, on the other hand, produces one group catalogue for each snapshot file.

The MergerTrees codes were designed to be used in a highly particular manner, namely they operate on snapshots generated by L-GADGET2 and must have more than one file for each snapshot. They do not use the hash table information, but they do expect one Group Catalogue, [3], per snapshot file so that the each processor can read in its share of the Group Catalogues and work on those. In the original code, the number of processors had to be a factor of the number of files per snapshot but this constraint was removed in this work.

Thus, extensive changes had to be introduced into the three MergerTrees codes.

As described in [3], the new HDF5 version of SubFind writes a single Groups file, however, the original version of the MergerTrees codes expect multiple files. Thus, new Fortran code was written to allow each of the three MergerTrees codes to read in data from a single file and distribute the data around the processors.

The simulation volume is split into slabs of equal thickness, with one slab per processor. Each processor receives all of the halos whose centre of mass lies in its slab, along with the subhalos contained in those halos. In order to facilitate this, code was added to the HDF5 version of SubFind to calculate the centre of mass of each halo and add it to the group catalogue files. This avoids the need to read all of the particle data just to determine which processor each halo or subhalo should be sent to.

In order to find the descendant of each subhalo, the TraceSubgroups code requires a list of the IDs of all of the particles in each subhalo at two consecutive snapshots. This information is read from the group catalogue produced by SubFind. The list of IDs can be quite large, so the IDs are read in blocks of 10,000. This avoids the HDF5 performance problems encountered with the FoF code while still allowing the code to be used on simulations where the group catalogue cannot be stored on a single processor.

Note that the new MergerTrees suite can only read and write using HDF5.

For testing purposes, we needed to form two simulations, one for the old MergerTrees suite and one for this new HDF5 version. This was done by running a simulation using L-Gadget2 and identifying halos and subhalos in the simulation with L-SubFind. The resulting snapshot files and group catalogues were then converted into an HDF5 format compatible with the HDF5 MergerTrees codes. Both MergerTrees suites could then be run on the same simulation. After extensive testing, it was found that the results were exactly the same.

This new MergerTrees suite is now more flexible in terms of how many processors you can use, since the number of processors is no longer required to be a factor of the number of snapshot files. This is especially useful for simulations which have only one file per snapshot which could not have been processed using the original codes.

The performance of the suite was measured due to time constraints; however, the main motivation for parallelising the codes was to allow for simulations to be larger than previously possible.

The MergerTrees suite has been installed, run and checked for correctness on *HPCx* and two of the core DEISA supercomputers, namely the platforms at IDRIS and RZG. This work was performed by the authors. This was a straightforward process, as the codes adhere to standard language features. The fact that the I/O routines now employ

HDF5 means that the binary datasets did not have to be manipulated when porting the datafiles between platforms.

5 Conclusions

The MergerTrees suite of codes, namely TraceSubgroups, SplitHalos and BuildTrees, have all been parallelised using a 2-dimensional slab decomposition, instead of the embarrassingly parallel method as described in the Specifications Document [2]. This parallelisation allows for much larger simulations to be considered than previously possible.

In addition, their I/O routines were optimised, as described in detail for FoF in [3]. This new MergerTrees suite can only read and write using HDF5.

A User Guide to this new MergerTrees suite can be found in [5].

The purpose of the DEISA research infrastructure is to enable scientific discovery across a broad spectrum of science and technology. The Virgo Consortium are typical of new users of the infrastructure requiring support and modest development of their software applications to take advantage of the new infrastructure. This work has been done under the auspices of the DEISA and VirtU projects and as a result a new community of scientific researchers have been brought to the infrastructure. The software applications are available for other users under the existing terms of the Virgo Consortium who are the application owners.

The objective of this work has been achieved, in that the output from GADGET cosmological simulations, along with output from FoF and SubFind, can now be post-processed from any of the DEISA sites, given that the MergerTrees suite has been ported to those sites.