



CONTRACT NUMBER 508830

DEISA
**DISTRIBUTED EUROPEAN INFRASTRUCTURE FOR
SUPERCOMPUTING APPLICATIONS**

European Community Sixth Framework Programme
RESEARCH INFRASTRUCTURES
Integrated Infrastructure Initiative

Specification document for D-JRA2-2.2,
D-JRA2-2.3, D-JRA2-2.4

Deliverable ID: D-JRA2-2.1
Due date: November, 30, 2004
Actual delivery date: December, 21, 2004
Lead contractor for this deliverable: EPCC, UK

Project start date: May 1st, 2004
Duration: 5 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	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Content

1 INTRODUCTION	2
1.1 Executive Summary	2
1.2 References and Applicable Documents	2
1.3 Document Amendment Procedure	Erreur ! Signet non défini.
1.4 List of Acronyms and Abbreviations	3
2 VIRGO'S COSMOLOGICAL SIMULATION CODES	4
2.1 GADGET	4
2.2 Hydra_MPI.....	4
2.3 Enzo and FLASH.....	4
3 TYPICAL WORK FLOW	4
4 DESCRIPTION OF THE TOOLS	5
4.1 FoF	5
4.2 SubFind	5
4.3 MergerTrees	6
4.3.1 TraceSubgroups.....	6
4.3.2 SplitHalos	6
4.3.3 BuildTrees	6
4.4 Power and Correl	7
5 DEVELOPMENT OF A GRID-ENABLED TOOLSET	7
5.1 Porting	7
5.2 Parallelising	7
5.3 Grid-enabling.....	7
6 DESCRIPTION OF D-JRA2-2.3, D-JRA2-2.4 AND D-JRA2-2.5.	9
6.1 D-JRA2-2.3 Grid-enabled "Group Finders".....	9
6.2 D-JRA2-2.4 Grid-enabled tools to build halo merger trees	9
6.3 D-JRA2-2.5 Grid-enabled tools for evaluating basic properties of the dark matter	9
7 APPENDIX	10
7.1 Quote from the DEISA Technical Annex, describing WP2 of JRA2	10

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 [2], 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 for 36 months and will form the foundations of the Theoretical Virtual Observatory.

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

This report is the first deliverable of WP2, "Development of a Grid-enabled toolset". This report describes the work to be carried out over the coming 10 months, which constitutes the remainder of JRA2's WP2.

The input and output of Virgo's large cosmological simulation codes (GADGET [3], Hydra_MPI [4], Enzo [5] and FLASH [6]) are created and manipulated by a number of pre- and post-processing tools.

These tools are to be ported to the DEISA infrastructure, Grid-enabled and parallelised if necessary. Specifically, these tools to be ported are FoF, SubFind, MergerTrees (TraceSubgroups, SplitHalos and BuildTrees), Power and Correl.

In this context, Grid-enabling means to allow the tools to utilise distributed data sets within the DEISA infrastructure, at least, and by introducing the data standards HDF5 [11] or BinX [12] into the tool's I/O routines. One of the first tasks of WP2 will be to investigate what Grid-enabling entails in practice on the DEISA infrastructure.

Two of the simulation codes, GADGET and either Hydra_MPI, Enzo or FLASH, will be ported to the DEISA infrastructure by JRA2's WP1 and WP3 respectively.

NB: this 'specification' document briefly describes each of the tools and how we intend to port, Grid-enable and, potentially, parallelise each one in turn. It also describes how much time will be spent on each code. As such, this is not a *technical* specifications document.

1.2 References and Applicable Documents

- [1] Virgo: <http://www.virgo.dur.ac.uk>
- [2] VirtU: <http://star-www.dur.ac.uk/~csf/virtU/virtU-final.pdf>
- [3] GADGET: <http://www.mpa-garching.mpg.de/galform/gadget>
- [4] Hydra_MPI: <http://www.epcc.ed.ac.uk/t3e/virgo>
- [5] Enzo: <http://cosmos.ucsd.edu/enzo>
- [6] FLASH: <http://flash.uchicago.edu>
- [7] DEISA: <http://www.deisa.org>

- [8] Barnes Hut Treecode: Barnes, J., and P. Hut, *Nature*, 324, 1986.
- [9] Monaghan, J.J., *ARA&A*, 30, 543, 1992.
- [10] Couchman, H.M.P., *ApJ*, 368, 23, 1991.
- [11] HDF5: <http://hdf.ncsa.uiuc.edu/HDF5>
- [12] BinX: <http://www.edikt.org/binx>

1.3 List of Acronyms and Abbreviations

EPCC	E dinburgh P arallel C omputing C entre
GADGET	G Alaxies with D ark matter and G as int E rac T
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 [1] Consortium's four main Cosmological Simulation Codes, namely GADGET [3], Hydra_MPI [4], Enzo [5] and FLASH [6].

These 4 codes are now briefly described. Note that *two* of these codes, namely GADGET and either Hydra_MPI, Enzo or FLASH, will be ported to the DEISA [7] infrastructure, as part of JRA2's WP1 and WP3 respectively.

2.1 GADGET

GADGET is a code to simulate the evolution of the universe, modelling the motion of both dark matter and gas and is, essentially, a very large N-body simulation code. The code calculates the gravitational evolution of the dark matter and gas in two separate calculations. The dark matter is computed using an FFT for long-range interactions, whilst a Treecode, similar to the Barnes-Hut Treecode [8], is employed for the short-range interactions. The gas component is computed using Smooth Particle Hydrodynamics, or SPH (See [9] for a good overview of SPH methods)

As part of JRA2, WP1, GADGET2 will be ported to the DEISA infrastructure, where GADGET2 is the most up-to-date, general version of GADGET that currently exists.

2.2 Hydra_MPI

Hydra_MPI is a very similar code to GADGET, in that it models very similar science, simulating both dark matter and gas, however, the N-body algorithm utilised for the dark matter component is different to the one used in GADGET, specifically Hydra_MPI employs a parallel Adaptive P3M method [10]. Furthermore, Hydra_MPI uses features of MPI-2 that GADGET does not require. Moreover, GADGET2 models more physical processes and thus offers more than Hydra_MPI. As such, the Virgo Consortium is considering replacing the Hydra_MPI code, which is to be Grid-enabled as part of JRA2's WP3, with an Adaptive Mesh Refinement, or AMR code. This AMR code will be either Enzo [4] or FLASH [5].

2.3 Enzo and FLASH

Enzo and FLASH are two AMR, or Adaptive Mesh Refinement codes, which are alternatives to using GADGET and Hydra_MPI for large cosmological simulations. Unlike GADGET and Hydra_MPI, these two AMR code do not employ SPH to model the gas phase. Unlike SPH, this alternative technique (mesh based as opposed to Lagrangian) is able to capture shocks in the gas component more accurately.

3 Typical Work Flow

This section gives a brief outline of how the tools are currently employed to post process data produced by a large cosmological simulation.

1. Firstly, one must run a large cosmological simulation using GADGET2, say. This code then produces a number of snapshots (output files) of the simulation at

-
- different times in the evolution. One can also employ Hydra_MPI, Enzo or FLASH to produce these snapshots.
2. Two codes, namely *Power* and *Correl*, can be employed to compute the power spectrum and correlation function, respectively, once the snapshot files have been produced. (Note, GADGET contains code to do these calculations already).
 3. Use *FoF* and *SubFind* to identify dark matter halos and subhalos for each of the snapshots produced.
 4. Use *MergerTrees* to find the formation history (referred to as merger trees) of each halo in the final simulation output frame. MergerTrees takes snapshot files and the output from FoF and SubFind as input.
 5. *Galform* then uses the output from MergerTrees to predict what galaxies would form in each halo in the simulation.
 6. *GRASIL* then takes the information regarding the galaxies predicted by Galform (such as their star formation histories and metallicity) and models the stellar population in each identified galaxy and the effects of extinction caused by dust in order to calculate the spectral energy distribution of each galaxy.
 7. Another code, *Zoom*, can be run to determine the parameters for another simulation, at a higher resolution, of smaller areas containing objects of interest produced by the original simulation. GADGET2 can then be used for this simulation, using the parameters from Zoom as input.

4 Description of the tools

The tools under consideration within WP2 are FoF, SubFind, MergerTrees, Power and Correl. The remaining tools, namely Galform, GRASIL and Zoom, will be considered outwith this work package.

4.1 FoF

Once GADGET2 has been run, the first task is to locate groups within the check-pointed datafiles: these are known as *snapshots*. These snapshots are then employed to determine gravitationally bound dark matter groups.

The Friends-of-Friends, or FoF, codes are a collection of *group finder* codes. For WP2, we shall be employing a code called *P-GroupFinder_Special*, which caters for both dark matter and gas. This is a parallel C/MPI code with around 3000 lines of code.

4.2 SubFind

Once a FoF group finder code has been run, which locates the Friends-of-Friends dark matter halos, SubFind can be used to locate the *subhalos* in the given snapshot.

After loading the particle data and the group catalogue of the snapshot, the dark matter density of all particles are bound in groups. The search of neighbours is not restricted to particles of the FoF group, but SubFind considers all dark matter particles. After this step, the code is applied to each FoF halo in turn, including an unbinding procedure, where the gravitational potential is computed with a tree algorithm. The net result of SubFind is that each FoF halo is decomposed into a set of substructures that are gravitationally bound. There can also be a number of particles left over ("fuzz") that are not bound to any of the subhalos

We intend to create a parallel SubFind for use with GADGET2 for DEISA, using the serial code *FoF_Special_SubFind* as a basis. This serial code has around 2500 lines of C code.

4.3 MergerTrees

Once the N-body simulation has been run, using GADGET2 say, and after the FoF and SubFind codes have found the Friends-of-Friends halos and *subhalos* for each snapshot, we then need to determine the merger history of each dark matter halo. 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, employing MPI, plus some serial ANSI C codes used for I/O. *TraceSubgroups* has around 3000 lines, *SplitHalos* has 1500 lines, and *BuildTrees* has around 200 lines.

4.3.1 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).

4.3.2 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.

4.3.3 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.

The output from *BuildTrees* can be used as input for *Galform*, where each merger tree is processed individually.

4.4 Power and Correl

The codes Power and Correl are used to calculate the power spectrum and correlation function of a dark matter distribution, respectively.

The Correl code is quite sophisticated, in that the correlation function can be measured accurately, even for large scales, in a relatively fast time, when compared to more traditional methods.

Power is a parallel C/MPI code with around 1600 lines, and Correl is a serial C code with around 1200 lines.

GADGET2 actually contains parallel code similar to both Power and Correl and it will be *this* code that will be extracted and ported to the DEISA infrastructure to form the DEISA versions of Power and Correl.

5 Development of a Grid-enabled toolset

As described in the DEISA technical annex for JRA2, WP2 (See Appendix), we intend to take the tools, described in the preceding section, and make them available on the DEISA infrastructure, by porting, parallelising and Grid-enabling them. The section now describes what we mean by the terms porting, parallelising and Grid-enabling.

5.1 Porting

Each code will, in turn, be installed, run and checked for correctness on each supercomputer currently participating in the DEISA infrastructure. This should be a straightforward process, provided that the codes adhere to standard language features, however, any binary datasets may have to be manipulated, in the first instance, to check the codes are ported correctly.

5.2 Parallelising

Three of the tools are currently execute as serial codes, namely SubFind, Galform and GRASIL. These codes will be parallelised, when the codes can be run in an embarrassingly parallel fashion, i.e. multiple instances of the same code running concurrently on different (distributed) datasets. More involved parallelisation strategies will not be considered due to time restrictions.

5.3 Grid-enabling

Once WP2 is complete, users will be able to run each of the tools on distributed datasets, where each dataset may reside on a different site to that running the tool. These tools will take advantage of the DEISA global file system as well as operating within a "conventional" distributed environment outside of the DEISA framework. To ensure this it will be necessary to introduce a portable data format for the I/O produced by each of these codes, such as HDF5 [11] or BinX [12]. Alternatives, or additional, solutions will also be investigated as part of D-JRA2-2.2. The result of this investigation will be used as a basis for JRA2's WP4: "Distributed data management and information extraction".

6 Description of D-JRA2-2.3, D-JRA2-2.4 and D-JRA2-2.5.

This section describes the projected work plan for the remainder of WP2.

6.1 D-JRA2-2.3 Grid-enabled “Group Finders”

Firstly, we shall investigate what it means to allow all the tools to access distributed datasets, both within and outside the DEISA infrastructure. Initially, we consider introducing HDF5 (or BinX may be a possible alternative) to ensure that the codes can operate in heterogeneous platforms, however, a general investigation will be carried out (see Section 5.3).

For this deliverable, only two codes will be considered:

- ? The parallel FoF code, *P-GroupFinder_Special*, will be ported to work under the DEISA infrastructure and Grid-enabled.
- ? The SubFind code, *FoF_Special_SubFind*, will also port be ported to operate under the DEISA infrastructure, parallelised¹ and Grid-enabled.

In this context, Grid-enabled means the codes are able to access data both inside and outside the DEISA infrastructure, using HDF5 or BinX (See Section 5.3).

This task will last 5 months, starting November 2004. The associated deliverable is due by the end of March 2005.

6.2 D-JRA2-2.4 Grid-enabled tools to build halo merger trees

This deliverable will consist of porting and Grid-enabling the three parallel codes within MergerTrees, namely *TraceSubgroups*, *SplitHalos* and *BuildTrees*.

In this context, Grid-enabled means the codes are able to access data both inside and outside the DEISA infrastructure, using HDF5 or BinX (See Section 5.3).

This task will last 4 months, starting April 2005. The associated deliverable is due by the end of July 2005.

6.3 D-JRA2-2.5 Grid-enabled tools for evaluating basic properties of the dark matter

Whilst *Power* is already parallel, a parallel version of the *Correl* code will be produced. Further, the I/O statements of both these codes will have to be replaced to function with GADGET’s snapshot files.

To these ends, we intend to extract code from GADGET2 itself, which includes advanced forms of both *Power* and *Correl*, to create new, stand-alone, parallel versions. These new versions will then be ported and Grid-enabled.

¹ A parallel version of SubFind that caters for dark matter only exists which we will use as a template for the parallelisation of *FoF_Special_SubFind*.

In this context, Grid-enabled means the codes are able to access data both inside and outside the DEISA infrastructure, using HDF5 or BinX (See Section 5.3).

This task will last 2 months, starting August 2005. The associated deliverable is due by the end of September 2005.

Appendix

This Appendix contains a direct excerpt from the DEISA Technical Annex, *Annex I – “Description of Work”*, describing WP2 of JRA2, and is repeated here for the sake of completeness.

6.4 Quote from the DEISA Technical Annex, describing WP2 of JRA2

WP 2: Development of a Grid-enabled toolset: a number of serial pre- and post-processing tools have been developed for GADGET and Hydra-MPI. Grid-enabled versions of these tools will be developed to cater for distributed datasets in heterogeneous environments, hence laying the foundations for WP 4. These tools will be able to take advantage of the DEISA global file system (see SA2) as well as operating within a “conventional” distributed environment for interoperability outside of the DEISA framework. More specifically this work has three objectives:

- Grid-enablement of group finders: we propose to develop and Grid-enable a series of tools known as “group finders”. Their purpose is to identify gravitationally bound structures in the dark matter. A code called SubFind has already been developed by Volker Springel to identify bound structures (or substructures) that are located inside larger bound structures. The work undertaken as part of this WP will enable users to apply the SubFind code to large distributed datasets in order to identify dark matter halos and their substructure. Once found, a number of existing analysis tools could be applied to calculate properties of the groups.
- Parallelisation and Grid-enablement of codes for making halo merger trees: the primary objective is to build structures called merger trees for each halo in the final output of the simulation. This task requires us to find all the groups at all times. The Virgo consortium has codes that allow one to determine for each halo at the final time - which halos in the past merged together to form it. Having found the merger trees for all the halos one can then use these trees to model galaxy formation. The codes for building merger trees already exist, but generally are serial codes, and would need to be adapted to work in a Grid environment.
- Development and Grid-enablement of tools for evaluating basic properties of the dark matter (e.g., density field, correlation functions, power spectra).