1. COVERPAGE AND PROPOSAL SUMMARY

REPLACE THIS PAGE WITH COVER SHEET(S)
1 Coverpage and Proposal Summary i

2 Scientific/Technical/Management Section 2
  2.1 Scientific objectives and rationale .............................................. 2
  2.1.1 Technical overview ................................................................. 3
  2.1.2 Research Directions ................................................................. 5
  2.2 Technology .................................................................................... 10
  2.3 Management plan .......................................................................... 15

3 Software Engineering Plan 17
  3.1 Developing requirements ................................................................. 17
  3.2 Software development ................................................................... 18
  3.3 Documentation .............................................................................. 18
  3.4 Testing and validation .................................................................... 19
  3.5 Release and maintenance ............................................................... 19

4 References 20

5 Biographical Sketches 21
  5.1 Curriculum Vitae: Thomas Quinn ................................................... 22
  5.2 Curriculum Vitae: Derek Richardson ............................................. 23
  5.3 Curriculum Vitae: Arif Babul .......................................................... 24
  5.4 Curriculum Vitae: Julio F. Navarro ................................................ 25
  5.5 Curriculum Vitae: Matthias Steinmetz .......................................... 26
  5.6 Curriculum Vitae: Fabio Governato ............................................... 27
  5.7 Curriculum Vitae: Ben Moore ........................................................ 28
  5.8 Curriculum Vitae: Hugh Couchman .............................................. 29
  5.9 Curriculum Vitae: Rob Thacker ..................................................... 30
  5.10 Curriculum Vitae: James Wadsley ............................................... 31
  5.11 Curriculum Vitae: Andrew Markiel ................................................ 32
  5.12 Curriculum Vitae: Julian Borrill .................................................. 33
  5.13 Curriculum Vitae: Cathy Horellou ............................................... 34

6 Milestones/Schedule/Cost 35
  6.1 Performance metrics ....................................................................... 35
2. SCIENTIFIC/TECHNICAL/MANAGEMENT SECTION

2.1. Scientific objectives and rationale

In the past few years computational astrophysics has matured to the point where simulations can be used to make quantitative predictions about the origin of planets, galaxies, and the Universe. Our team has been at the forefront of these activities including:

- Simulating the clustering on the scale of the entire observable Universe.
  In Colberg et al. 1998 we demonstrated that a low density model of the Universe is needed to explain massive clusters seen at moderate redshifts.

- Making testable predictions about the density profiles of galaxies and clusters.
  In Moore et al. 1999a we established that the standard Cold Dark Matter model fails to reproduce the rotation curves of dark matter dominated galaxies, one of the key problems it was designed to resolve.

- Constraining dark matter theories based on the substructure within galaxies and clusters.
  In Moore et al. 1999 we revealed how galaxies would look like scaled versions of clusters in the standard Cold Dark Matter model. This and the previous item has lead to a flurry of activity in the field to find modifications of the standard theory to reconcile these results.

- Performing the first “full disk” direct simulation of the growth of planetesimals in the terrestrial and asteroid belt region of the Solar System.
  In Richardson et al. 2000, we demonstrate how Jupiter can suppress the formation of a planet in the asteroid belt region.

Comparisons of these simulations with observations lead to insights into fundamental issues such as: what is the amount and nature of the dark matter in the Universe? What is the likelihood of finding an Earth-like planet around another star?

As exciting as these results are, they are just stepping stones to understanding the fundamental origins questions. The simulations above were fairly simple in terms of physics (gravity only), but further progress will require accurate modeling of more complicated phenomenon. An essential ingredient in the successes above is a large dynamic range, many orders of magnitude, in time and space. Adding more sophisticated physics will only increase the resolution necessary for accurate, predictive simulations. Although Moore’s law has helped, getting several orders of magnitude improvement in the calculations requires new algorithms and massively parallel machines. Our N-Chilada framework is driven by the need to bring these together, thus enabling us to break new ground in scientific discovery:

- Understanding the formation of galaxies in their complete cosmological context.

- Disentangling the complex interaction of cluster galaxies and the intra-cluster medium.

- Creating a predictive model of Solar System formation, from proto-solar disk to fully formed planets.
Accurately performing these calculations and confronting the results with new ground and space based observations will greatly enrich our understanding of our origins. Aside from these goals, we will produce a lot of "technology transfer". We have collaborators in other scientific disciplines who recognize the strength of our approach and are very interested in using our framework to enable their science. Examples include protein folding, granular dynamics and microelectronics.

2.1.1. Technical overview

The ability to cluster commodity computers into a powerful parallel machine has been a boon to scientific computing. One can construct a dedicated, high-performance parallel computer comparable in performance to traditional supercomputers at a fraction of the cost. The national facilities have put together massively parallel machines with processor counts well into the thousands, and very impressive peak performance numbers.

Putting these parallel computers to work on scientific applications is another matter. The application at hand must be either a) "embarrassingly parallel" (e.g. Monte-Carlo simulations) b) easily expressible in an array based language, or c) rely on a software library (e.g. LAPACK) for most of the computation. If the application does not fit into one of these categories, then a large amount of time has to be invested in order for it to run well on a parallel machine.

Particle simulations, by their adaptive nature, require irregular data structures such as trees and adaptive meshes for efficient computation. Such data structures are not easily or naturally expressed as arrays and, therefore, require a significant effort to run on a parallel machine. We have had great success in this area with the PKDGRAV parallel tree code, now used by affiliated groups throughout the world, and approaching public release.

The programming effort required comes to a head when analysis of simulations is considered. This is because in contrast to the simulation, where the nature of the calculation is well defined, the analysis usually involves a number of different calculations as the scientist tries to make discoveries from the simulation data set. Furthermore, analysis methods often change rapidly as new phenomena are understood. Consequently, our widely used (serial only) visualization and analysis tool, TIPSY, is currently 60% larger in lines of code than our parallel gas dynamical simulation code, GASOLINE, reflecting the variety of analyses we perform on a simulation.

Our success at running simulations in a parallel environment has lead to a mismatch between simulation and analysis. The simulation data sets have become so large that we now need the parallel machines even for the most straightforward analysis or visualization. The number of data elements is also so large that serial software-rendered visualization is no longer practical.

N-Chilada is our proposed framework for running large simulations, and analyzing and visualizing large irregularly structured data sets using parallel computers. Using a modular design, the details of the parallel implementation will be hidden from the end user. This will allow the rapid development of new simulations and high performance analysis tools without requiring extensive knowledge of the complicated underlying parallel issues. New analysis procedures can be run in parallel by writing only serial code: either at a high level to control the flow of the computation or at the individual processor level for instructions to operate on individual particles. In its full implementation the framework will also allow the user to choose different algorithms to evolve the data set, thereby enabling the most efficient algorithm to be chosen at all times. For example, gravity calculations on weakly clustered particle distributions are most efficiently done with a grid; as the clustering gets stronger,
trees become the winner. Within the framework, we can easily switch between these two
techniques.

The concept of a modular, extensible design for parallel computation software is not new
to the members of our team. Our current production codes PKDGRAV and GASOLINE
have been designed with these concepts in mind. The success of the design was proven by
the ease with which we have taken the code originally designed for cosmological dark matter
simulations and applied it to new physical regimes such as Solar System formation, asteroid
collisions, and even collisional dynamics in grains of sand. Another benefit of our current
code structure is the ease with which it can be ported to new architectures. Originally written
for a threaded environment, it has been ported to PVM, MPI, POSIX threads, SHMEM,
MPL, NX, and SPP shared memory.

N-Chilada will build on the design of these codes, and extend them to make them useful
to a broader community. Improvements include

- a more general file format to handle a broader variety of data,
- an interactive/scripting language for quickly prototyping calculations, performing anal-
  yses, and computational steering,
- a wider variety of data structures for performing calculations on particle data sets,
  enabling interoperability of grid and tree-based algorithms,
- a real-time integrated interface from simulation data to a visualization system.

We are committed to building this new framework because we see it as the clear way for-
ward. The collaborative effort proposed here will provide the community with the following
major benefits:

- Quick prototyping and experimentation with new algorithms in parallel.

Increases in simulation quality have been as much a result of new algorithms as in-
creases in machine performance. As shown in Figure 1, changes in algorithms have
actually outpaced the phenomenal increase in computer performance over the past few
decades. Incorporating new algorithms into our simulation code is necessary as we
continue to tackle harder problems.

- Use of parallel power for analysis and visualization.

Analyzing the simulation data is currently the bottle neck: transferring and translating
huge data files, writing specialized code and laborious analysis in serial. Harnessing the
power of a parallel machine will allow us to drastically increase our scientific output.

- Rapid portability onto new platforms.

Porting code to new platforms can absorb an arbitrary amount of time. The modular
machine dependent substructure for GASOLINE/PKDGRAV is proven technology that
has effectively abolished this problem for our group.

- The application of parallel computation to a diverse set of problems with relative ease.

The last benefit may appear altruistic on our part, but it is a logical consequence of
having the first three items, and has served us well, facilitating research in diverse areas:
collisionless systems, gasdynamics, planetismsals and granular dynamics. We therefore have
a strong internal motivation to bring about this framework, which will be a great benefit to
a broader scientific community.
2.1.2. Research Directions

The implemented framework will allow us to aggressively tackle a number of fundamental problems in current astrophysics as well as enable research in totally new fields of endeavor. Progress in the following areas is currently CPU limited and the advent of \textit{N-Chilada} will allow major gains to be made.

\textsc{COSMOLOGY}

Here we make the connection between the upcoming MAP satellite data and the Universe of today. MAP will measure the small cosmological fluctuations in the microwave background temperature, essentially giving us the "initial conditions" of the Universe. Simulations of large cosmological volumes are necessary to quantify how initial fluctuations of one part in $10^5$ in the microwave background turn into the galaxies and clusters we observe. The spatial distribution of galaxies across a statistically significant region of the Universe gives clues about the early Universe, but this has to be disentangled from how these objects form according to environment. Making sense of the large scale surveys such as the Sloan Digital Sky Survey depends on understanding this process. With these kinds of simulations we aim to study how the galaxy distribution depends on the nature of the cosmological parameters and the properties of the initial density field of the Universe.

During the implementation of the \textit{N-Chilada} framework we plan to simulate a volume 100 megaparsecs on a side with at least 80 million particles and a spatial resolution of only a few kiloparsecs, a first in this field. Dumping the simulation state often we will be able to couple the simulation to advanced semi-analytical techniques (currently developed by UW postdoc Lucio Mayer) that will allow us to predict the galaxy population properties and its evolution in time.

This will require:

- A large amount of parallel computing time for the actual simulation (100k node-hours per run).

This may seem large, but we will be creating legacy simulations here. Similar simulations (50 million particles in a 1000 Mpc box) that we performed several years ago are trustworthy enough that they have been analyzed by multiple groups for several years.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{chart.png}
\caption{The evolution of algorithms and machines. The open circles (scale to the right) show the evolution of machine speeds over three decades. The evolution of algorithms for force evaluation (filled circles) is far more dramatic. Spatially and temporally adaptive algorithms have provided throughput gains that exceed the phenomenal increase in computer performance. We expect this to continue for the next five years.}
\end{figure}
A large number of outputs and hence a large amount of data (300Gb)

Extensive postprocessing of individual outputs.

Interpreting the "galaxies" requires the identification of individual halos and then computing a number of different physical quantities such as density profiles and total angular momentum. Some of the postprocessing algorithms needed currently perform \( N_{\text{halo}}^2 \) calculations and are very computationally intensive once the volume contains several individual halos with \( N_{\text{halo}} > 10^6 \). The N-Chilada framework will include a suite of parallel analysis tools to find and analyze individual dark matter halos within the simulation volume.

\[
\text{GALAXY CLUSTERS}
\]

Three stages in the formation of a galaxy cluster. The color encodes dark matter density. This is our ground-breaking simulation that first accurately resolved "halos within halos" and accurately modeled the central density profile, results that have fomented significant rethinking of the standard model of dark matter. The simulation was performed by PKDGRAV, and the visualization was done using TIPSY. However, it is a simulation of only the dark matter. Higher resolution and gas dynamics will allow us to make tighter connections with observations.

High resolution simulations of galaxy clusters are one of the hallmark achievements of this collaboration. Through extensive use of the "volume re-normalization" technique we were able to extend the dynamic range of our large cosmological simulations by several orders of magnitude. A simulation of a Coma-like cluster was able to resolve individual dwarf galaxies (10kpc in radius) in a box of 1000 Mpc across: a dynamical range of almost \( 10^5 \). The technique requires that first a large scale simulation is done at low resolution and regions of interest (e.g. clusters of galaxies) identified. The initial conditions for the high resolution simulations of the regions of interest are then generated by extracting the low-frequency initial conditions associated with these regions from the large simulation and adding high-frequency density waves and high resolution particles. It is not sufficient to just extract the initial conditions for the objects of interest. The surrounding matter distribution, because of its tidal influences on the structure under study, must also be taken into account. Note how this requires highly adaptive methods to be feasible.

These simulations resolve many thousands of subhalos within clusters, and resolution tests show we may have converged on their inner structure, putting strong constraints on the hierarchical structure formation model and the nature of dark matter. Our current highest resolution halo contains five million particles inside its virial radius. (Ghigna et al. 1999)

We now have a host of new observational tests of the hierarchical structure formation paradigm that include: the extent of galactic halos in clusters and satellites in galactic halos, the orbital distribution of cluster/satellite galaxies, the number of satellites as a function
of their circular velocity or mass, the spatial and velocity distribution of satellites, and the central density profiles of clusters and galaxies. These properties have heretofore been the subject of endless debates among cosmologist because of the lack of a predictive theory: contradictory results can be had just by turning a few “knobs” on a phenomenological model. Our accurate numerical models will allow us to unambiguously interpret these observations.

The next step to take is to perform high resolution gas dynamical simulations of these clusters. This is particularly critical in order to make reasonable comparisons with data from X-ray satellites like CHANDRA. The simulations are needed in order to interpret what constraints the X-ray data can put on physical quantities such as the size of the X-ray core and the temperature structure of the cluster. TIPSY already has the capability to generate X-ray maps from our simulated clusters, but it is very computationally intensive so this capability will be moved into our parallel framework. Similar resolution and dynamic range will be needed in order to follow galaxy intra-cluster gas interactions, but the computational challenges will be greater because of the greater dynamical ranges in timescales introduced by the gas. N-Chilada will allow us to rise to these challenges.

With GASOLINE and PKDGRAV we can study in detail the morphological evolution of galaxies as they orbit inside the gravitational potential of a galaxy cluster or a larger galaxy. The strong tidal field and rapid fly-bys with other orbiting galaxies drive a dramatic morphology evolution. Model spiral galaxies get destroyed or transformed into spheroidal systems with properties remarkably similar to those of observed galaxies in clusters or small satellites in the Local Group. Hence, we are able to construct a unified theory of the morphological diversity among these nearby galaxies.

GALAXY FORMATION

Galaxies are the most distinctive objects in the universe, containing almost all luminous material. They are remarkable dynamical systems, formed by non-linear collapse and a drawn-out series of mergers and encounters. And yet, there are tight empirical relations, drawn from observations, that seem to apply to most of them.

Members of our team have a long experience in simulating galaxy formation. Navarro was among the first to simulate galaxy formation in a proper cosmological context (Navarro, Frenk & White 1995). However, these simulations failed to create realistic spiral galaxies: the disk angular momentum was a factor of ten lower than in real spiral galaxies. After rapid cooling of gas inside the ubiquitous dense and small halos at high redshift the scene is set for a nasty surprise: gas loses too much angular momentum when it is trapped inside these halos as they spiral together to form the main body of the galaxy. The disk that forms from the assembly of gas has a scale length that is several times too small compared with real spiral galaxies. Somehow the gas has to remain in a diffuse state all the way up to the time it is accreted onto the central disk.

In the following years Quinn (Quinn, Katz & Efstathiou 1996) introduced a realistic UV background (created by forming stars) and Steinmetz & Navarro (Navarro & Steinmetz 2000) explored simple recipes for supernova feedback as possible solution. If the gas remains hot or is expelled out of the smaller halos by supernova winds its angular momentum loss should be significantly lower. Still, the problem of the “over-cooling catastrophe” remains until today, and no one has yet simulated the formation of a realistic disk galaxy from a cosmological simulation. This is a major embarrassment for the numerical cosmology community. We plan to join forces, take advantage of the large improvement in computing power afforded by the N-Chilada framework and tackle the problem again.

Galaxy formation is indeed a challenging computational problem: it requires a dynamic range of millions in each dimension, produces spatially varying timescales with a similar
dynamic range, and involves resolution-sensitive interstellar medium physics. To form a stable Milky Way-like galaxy, tens of millions of resolution elements must be simulated to the current epoch. Locally adaptive timesteps reduce the CPU work by orders of magnitude, but not evenly throughout the computational volume: a considerable challenge for parallel load balancing. No existing N-body/Hydro solver can handle this regime efficiently, however PKDGRAV/GASOLINE must be considered the state of the art. The N-Chilada design has spatial and temporal adaptivity as a core feature; thus, galaxy simulation will be a routine application that makes full use of parallel resources. This will be a watershed activity: only by thoroughly understanding galaxy formation will we be able to disentangle the light vs. mass problem that is the bane of cosmology.

Modern galaxy surveys present challenges not only in raw data analysis but also in terms of providing mock galaxy catalogues for aiding their interpretation. The simulation of large cosmological volumes at high resolution is a precursor to producing ensembles of galaxies for comparison with observational catalogues. The N-Chilada framework will be able to manage the simulation and analysis of multiple runs within a single environment with the ease of writing a driver script. The resolution of simulations made possible by the framework and the rapid development time provided by driver scripts will enable new analyses that will directly influence the understanding of galaxy catalogues. Specific examples include simulated HI emission maps, colors, absorption profiles, and radial light distribution. All of these would be calculated as simulated observations for the direct comparison with data from telescopes.

**GALAXY EVOLUTION**

A simulation of a galaxy being “harassed” by interactions with other galaxies in a cluster (see Moore et al. 1996). This simulation demonstrated how interaction with other cluster members can drive the morphological transformations of galaxies and explained the appearance of distant clusters as seen by HST. The smooth shading indicates gas density, while the points show the stellar component. This visualization was performed using TIPSY.

Powerful instruments such as Hubble Space Telescope or the Very Large Telescope make it now possible to observe directly galaxy evolution over a time scale comparable to the age of the universe. Both observations and numerical experiments seem to indicate that galaxies
may even change morphological type over their lifetimes: bars can form and dissolve, bulges and disks can grow through accretion, episodic bursts of star formation can be triggered by galaxy interactions and influence the subsequent dynamical evolution. In order to understand the quantitative importance of all these processes, and get hard results out of ambitious projects like the Hubble Deep Field, a large number of simulations with a high spatial resolution (kiloparsec scale) of galaxies in different environments has to be performed and compared with the results of large observational surveys. A framework like \textit{N-Chilada} is the natural way to couple simulations of the evolution of individual galaxies (current simulations run with $\sim 10^6$ points per galaxy) to the cosmological simulations which may currently devote only $\sim 1000$ points per galaxy.

\textit{PLANET FORMATION}

A simulation of the formation of the terrestrial planets under the influence of the outer gas giants showing how Jupiter can suppress planet formation in the asteroid belt. (See Richardson \textit{et al.} 2000) This is the first full disk simulation that self-consistently included the gravitational interactions of the planetesimals, collisions, and interactions with the outer planets. The color indicates density of planetesimals. The simulation was performed with PKDGRAV and visualized using TIPSY.

Understanding how planets form is a cornerstone of astrophysical research and one of the fundamental astrobiology and origins questions. Projects like Terrestrial Planet Finder need the theoretical groundwork to know where to point: we don’t want to be like the anecdotal drunk looking for his keys under the lamp-post just because he can’t see in the dark! The topic encompasses many complex problems, from the interaction of dust grains with a turbulent gaseous nebula to the orbital evolution of growing protoplanets. One of the goals of the \textit{N-Chilada} project is to provide the tools to tackle this diverse range of interactions.

The planetesimal hypothesis states that planets form by the pairwise accretion of smaller planetesimals. In order to model this process it is necessary to predict collisions (and near misses), and resolve the consequences of such collisions, whether they be mergers, fragmentations, or something in between. Unfortunately, encounters are relatively rare since the time between collisions is long compared to the orbital time, particularly during the late stage of
massive impacts. The computational challenge is to find an adaptive timestep scheme that is stable over millions of dynamical times yet that can handle close encounters accurately and that is suitable for simulations involving millions of planetesimals. We have algorithms in hand, but implementing them in parallel has always been a challenge absent a framework like \textit{N-Chilada}. A separate issue is handling the details of the collision physics. The framework will be extensible enough to handle the small asteroid regime where material properties play an important role. Accurate simulations of this regime will allow us to correctly parameterize collisions in the large scale simulations.

The scientific payback here will be enormous. There are a number of outstanding fundamental questions about the origins of planetary systems which these simulations will be able to answer. What is the relationship between hot Jupiters and terrestrial-like planets? Do asteroid belts form around other stars (if so, this could be trouble for the Terrestrial Planet Finder, as dust may overwhelm the planet’s signal)? How common are other small body reservoirs such as Kuiper Belts and Oort clouds? How effective are planetary systems at cleansing small bodies and thus mitigating sterilizing impacts on any terrestrial planets? What is the origin of the spin of the terrestrial planets? We are literally opening whole new worlds here.

\textit{MICROELECTRONICS}

This item is included to demonstrate the generality of the framework, and its applicability to a very wide community. The work is that of Vikram Jandhyala, a UW colleague in the Electrical Engineering department.

The high frequency of operation of micro-electronic devices necessitates a mixture of quasi-static and full-wave electromagnetic analysis. Integral equation solvers are advantageous since they only discretize the surface. However, the large number of surface basis functions ($10^5$ – $10^6$) needed necessitates the use of a tree-based algorithms for the iterative solution of the equations on an adaptive mesh. Fast multipole methods are popular for this, but faster, parallel and hierarchical methods are needed for full-chip analysis at sufficient speeds so that results can be incorporation into design cycles. A parallel adaptive tree code that can function with new inhomogeneous plane-wave operators and recursive direct solvers will be a great benefit for this application, both for research computing clusters as well as for the microelectronics industry. Producing such a code would be an inordinate amount of work unless one already has the framework in place.

Our framework will be used in other scientific disciplines for which there is not space to discuss in detail here. For example, Markiel and Quinn are currently collaborating with G. Seidler of the UW physics department on a granular dynamics project. \textit{PKDGRAV} simulations are being compared to laboratory experiments to determine the dynamics of packing spheres. Quinn has an ongoing collaboration with V. Daggett of UW Medicinal chemistry to incorporate protein folding physics into the parallel framework.

\textit{2.2. Technology}

The technological goal of our proposal is to extend the parallel programming paradigm that has been so successful in \textit{PKDGRAV}. This will enable both the enhanced performance of our high-end simulation codes as we experiment with new algorithms, and the parallelization of a whole suite of applications: both our own analysis and visualization tools, and simulation programs from other disciplines.

\textit{PKDGRAV} was initially designed in consultation with Larry Snyder’s \textit{A-ZPL} (Lin & Snyder 1994) parallel language group in the University of Washington Computer Science department, and we followed some of the design principles of the language. Ultimately, a
sufficiently powerful parallel language could address all the issues we bring up here, and it is an exciting computer science research issue to which we will continue to contribute in collaboration with the CS department. A-ZPL implements a programming model, called Phase Abstractions (Griswold et al. 1990). The Phase Abstractions model is capable of expressing task parallelism, pipelined parallelism and other parallel programming paradigms, not just data parallelism. The programming model recognizes three different “programming levels”:

- Process or “X” level – a composition of instructions,
- Phase or “Y” level – a composition of processes into a parallel algorithm,
- Problem or “Z” level – a composition of phases to solve the overall application.

However, ZPL as currently implemented only supports the “Z” programming level. Essentially, it is currently limited to an array-based language supporting data-parallel operations. As such it makes parallel programming of operations on regular data structures very simple and easy. Therefore, it is generally unsuitable for programming with irregular data structures such as trees; tree algorithms can be expressed in later versions using sparse arrays, but it is not natural. Furthermore, it is not an object oriented programming language, so modularity would require extra discipline in programming.

Nevertheless, the programming level concept proved useful in designing PKDGRAV. The program is split into a “master” level corresponding to the “Z” level where the overall flow control takes place, a “processor set tree” level which coordinates work among all the processors, and finally the individual processor level where code is executed in serial on the data in individual processors.

Finally, we have a Machine Dependent Layer, MDL, which is a function library that enables us to write “portable” parallel codes. MDL supports message passing (MPI, PVM), shared address space (shmem) and ccNUMA architectures. Codes written with MDL are entirely serial except for the functions provided by the MDL library:

- Starting and stopping threads;
- A memory swapping primitive that shuffles particles among processors;
- Client-server primitives that allow remote function calls;
- Caching primitives that allow local arrays of data to be visible to all threads (TREECACHE);
- Output diagnostics and a few functions that enable dynamic load balancing.

Three types of memory sharing are supported; the most widely used is read-only. GASOLINE modifies shared data in a highly restricted sense. AFOF (Friends-of-Friends) requires a write-once type of memory sharing. MDL is 1,200 lines of C code and has been ported to new machines in just a few hours by 4 different people (it takes days to optimize node and TREECACHE performance).

The main component of PKDGRAV is a spatially adaptive potential solver (Barnes & Hut 1986) which is critical for our “large-N” simulations.1 The parallel scaling of this code is

1It has been claimed that the Fast Multipole Method (FMM) is asymptotically faster than classical tree codes. We have proven that the lower bound for a calculation of gravity is $\Omega(N\log N)$. In one-dimensional gravity, the force is given by the number of planes to your left minus the number of planes to
Fig. 2.— Performance figures for a highly clustered 3 million particle dark matter simulation. In the left panel, the percentage of linear speedup is shown for the ARSC T3E. Note that our code maintains a high percentage of machine utilization even for large numbers of processors. The right panel shows the “science rate” (related to our milestone performance criterion), which directly measures the time a given size simulation will take. Note again the near linear speedup.

shown in figure 2. Note how we get better than 80% of linear speedup even out to 128 processors and 70% at 256 processors. If we use Amdahl’s formula to calculate the fraction of our code that is running in serial, we get 0.17%. This is amazing considering the fact that traversing the tree is considered as book-keeping overhead in these numbers. The “science rate” shown on the right hand side is in units of particle force updates per second which is a direct measure of how fast we can accomplish a given simulation assuming a fixed timestep for all particles.

However, a key feature of the layered structure is its flexibility in implementing new, faster algorithms. We see several areas where known algorithms can dramatically improve our science rate, but are too complicated to implement outside the framework. Within PKDGRAV we have already implemented adaptive timestepping such that the force on each particle is calculated only as often as needed according to the local dynamical time. In this way the cost per timestep drops from $O(N \log N)$ to $O(N_{\text{active}} \log N)$, where $N$ is the total number of particles, and $N_{\text{active}}$ is the average number of particles that need force evaluations on a given timestep. The number of force evaluations has gone down drastically ($N_{\text{active}}$ is typically small compared to $N$), but the tree builds are still $O(N \log N)$, so they dominate the calculation. Dynamic trees are a known, obvious solution, but implementing them in parallel would be tedious outside our proposed framework. Likewise, there are

your right. Hence, it is clear that the calculation of gravity has the same complexity as a sort although exactly $N$ interactions must be calculated. Many schemes that claim to be $O(N)$ start with a tree build which is a significant fraction of the work in a traditional $O(N \log N)$ scheme. Since the asymptotic scaling is $O(N \log N)$ in both cases, the only issue is which code is fastest. Furthermore, timestep adaptivity looks impossible with FMM.
algorithmic improvements borrowed from the Fast Multipole Method to be made in the gravity calculation, such as local expansions of the multipoles. In short, our colleagues in the Computer Science department are full of great ideas about known algorithms that will speed up our calculations. A parallel framework is what they need to experiment with them on hard applications.

The layered structure has also allowed us to relatively easily move other applications into the parallel code, most notably GASOLINE, a code that evolves gas dynamical equations using the technique of Smooth Particle Hydrodynamics (Lucy 1977; Gingold & Monaghan 1977), and our planet formation code which, as well as solving for gravity, detects and resolves collisions between particles. Nevertheless, moving new applications into the above structure is not completely straightforward. Firstly, a detailed knowledge of some of the inner workings of the code is needed for someone to add new functionality. Secondly, a small amount of parallel code has to be written for most new functions. This parallel code is usually straightforward, but tedious and unnecessarily complex.

Recognizing that the resources of parallel computers will be necessary for more and more of our applications, particularly visualization and analysis, we started exploring the concept of a parallel framework in which to create these applications. We first investigated the possibility of using an existing framework such as ROOT (http://root.cern.ch/) and making it parallel. Unfortunately, it has been our experience that parallelizing an already written code requires a significant re-write, and usually results in poor scaling. Another possibility is adapting an existing parallel framework such as Cactus (http://www.cactuscode.org/), DAGH (http://www.caip.rutgers.edu/~parashar/DAGH/) or PARAMESH (http://sdcd.gsfc.nasa.gov/RIB/repositories/inhouse_gsfc/Users_manual/amr.html). However all of these packages are especially designed for regular mesh calculations, and would be hard to adjust for our irregular data structures.

Figure 3 outlines the structure of our data daemon concept for N-body (and more general hierarchical) computation. It represents the first stage in our design process. Once the basic foundation is created, each of the text boxes in the figure can be considered as a more-or-less independent unit. One daemon would be running on each of the nodes of a parallel machine, and communicate with other daemons via the MDL layer. Applications would connect to one of these data daemons which would establish itself as the master of the collection of daemons involved in the calculation. At the moment we envision three separate types of data structures to organize the data and domain decomposition: binary trees, oct trees, and adaptive meshes. The latter is essentially a tree with block-adaptive meshes as the leaf nodes. Domain decomposition and the parallel aspects of the algorithms would be essentially transparent to the user who would only have to provide the upper ("Z") level application instructions and serial code for the function library (the "X" level) which would run on the individual nodes. This dramatically reduces the burden on the programmer: only serial code is needed.

The generic algorithms available would depend on the data structures used. Tree algorithms would include operations such as "Global Top Down" traversal, used to calculate gravity and "Local Proximity" for finding nearest neighbors. Grid algorithms would include operations like FFTs and Interpolation. For example, the application programmer would only have to write a routine that calculated a quantity based on values in neighboring grid cells without having to worry about which processors the cells are on.

A scripting language will be part of the framework for rapid prototyping of new applications or algorithms. We will probably choose something like CINT (also used by the ROOT package) an interpreted subset of C++. Using an interpreted language similar to C++ will
Fig. 3.— Structure of the "Data Daemon" for parallel N-body computation. One of these daemons would be running on each node of a parallel computer. The end user would interact with only the Application Interface and the Function Library/NChilada Scripts. The complexity of the parallel computation is handled by the domain decomposition and the generic algorithm level, which is in turn on top of the Machine Dependent Layer (MDL).
allow us to compile scripts directly into the function library for increased performance after the prototyping is done.

The MDL library is essentially the one we use in PKDGRAV, but it is hoped that a more object-oriented MDL++ will replace it, and actually some of this new MDL exists as test code.

A feature of the data daemon design is the separation of functionality into component pieces. As outlined in the next section, this will allow a number of authors to work independently without interference in the construction of the framework.

2.3. Management plan

The PI has experience in a number of software software projects of varying size. As a co-author of TIPSY he has experience in working out the compromises needed even within a small group. On the other end of the scale, his experience in the software side of the Sloan Digital Sky Survey has given him knowledge of the issues involved in large scale projects. These issues include such details as the strengths and limitations of various software engineering tools, but also more broader managerial problems such as communications between groups spread across the country and resolving conflicting goals. Within the current collaboration he has successfully help coordinate the efforts of people working on the same code (PKDGRAV) with wildly different scientific objectives.

The PI also has extensive cross-disciplinary experience. Starting in 1993 he participated in the University of Washington Grand Challenge team that included the departments of Astronomy, Applied Math, Physics and Computer Science. He organized a seminar series that brought together a wide variety of disciplines, including Chemistry, Materials Science, and Biology to discuss their computational techniques. He has worked closely with faculty and students in the Computer Science and Engineering department in improving algorithms for particle simulation. The PI is also a member of the Astrobiology Program at the University of Washington, where he interacts with Atmospheric Scientists, Chemists, Oceanographers, Geophysicists, Microbiologists and Aeronautical Engineers. This broad familiarity across disciplines combined with the ability to build collegiality will give the PI effective authority in making resource allocation decisions based on the needs of meeting the project milestones. Furthermore, except for a small sub-contract to the University of Maryland, all the money will be spent at UW under the PI's direction.

The group of Co-Is and collaborators that we have put together is very impressive. The core group has been working together for more than 5 years. A good number have passed through Seattle as post-docs, and have continued to benefit from collaboration as they moved on. Babul and Navarro started collaborations with UW soon after they arrived in Victoria. The group has contributed to almost all the recent major simulation milestones in cosmology: Couchman's code was used for the one billion particle "Hubble Volume" simulation (Colberg et al. 1998), PKDGRAV was used in the first simulation to accurately resolve galaxies within clusters. (Ghigna et al. 1998) Thacker and Couchman have set the standard for high resolution gas simulations of galaxy formation. (Thacker & Couchman 2000)

Our group also has a history of giving out and supporting codes in the community. Couchman's codes Hydra (Pearce & Couchman 1997) and AP$^3$M (Couchman 1991) are used by over 20 groups worldwide. Quinn is a coauthor of TIPSY with N. Katz (U. Mass.) which is used at over 46 institutions in 13 different countries. PKDGRAV and GASOLINE are currently being used by groups in Seattle, Maryland, Victoria, Ontario, Britain, Italy, France, and Germany.

On a group level, we build collegiality for cohesive interdisciplinarity as follows:
• Project meetings are held weekly.

• Postdocs and students are funded jointly by the interdisciplinary grant and a disciplinary grant to insure active interest by all stakeholders.

• Joint supervision of students provides extremely high bandwidth of communication.

• Astronomy provides a large suite of offices with an interaction area. New projects have a "fish out of water" exposure in this suite (often CS&E postdocs, during the last year protein folder Keith Laidig). Two desks are reserved for rotating use by visitors and graduate students from other departments.

• Coordination across institutions requires frequent exchange of people, team meetings and workshops. We have postdocs who divide time between UW and other institutions (U. of Victoria, Observatory of Milan). There is a significant budget for Visiting Scientists to come to Seattle for a week or a month at a time. Two workshops among the collaborators have already happened this year, and we have budgeted in our milestones travel money to keep these workshops going. In addition, UW personnel will spend time visiting the other institutions, particularly University of Victoria and McMaster University.

Finding people to do the work that we proposed is difficult in many fields. It is extremely difficult to compete for programmers and research staff against the IT industry. However, because of the draw of Seattle, the "wow" factor in the research we do and the collegial work environment, we have been very successful in recruiting and retaining skilled staff in the face of a tight labor market. For example we have been able to retain Chance Reschke, a highly skilled system administrator and one of the people on the original Beowulf team, in spite of the fact that he could make double his salary outside the University.

The large number of collaborators participating in this proposal is a reflection of past successes in the field in terms of simulation and analysis tools and the perceived need for the framework proposed. Attendance at the two Seattle workshops demonstrates the commitment these people have toward the project.

The core architects, Stadel, Wadsley and Quinn, were chosen because of their previous successes in designing and implementing efficient high performance tools that are used by the rest of the collaboration. They will be assisted by two programmers and two research staff. Particular tasks are called out for Thacker who will work on the Adaptive Mesh Refinement algorithms, and Markiel who will work on visualization aspects. As well as verifying the suitability of the framework for Solar System simulations, Richardson will also oversee work on data formats and I/O, an area in which there is other expertise at the University of Maryland. One other special area of expertise will be provided by Steinmetz who will consult on integrating the use of special purpose hardware, such as GRAPE processors, into our framework.

A key strength of our proposal is the large number of "power users" who have committed to beta-testing and using the *N-Chilada* framework as it develops. Babul, Couchman, Governato, Horellou, Borrill, Moore, and Navarro are all very experienced in the writing, running or analysis of large simulations in their respective scientific fields and will provide critical feedback both to the overall performance and usability of the framework on various platforms, and the suitability of the framework for their scientific progress. As well, all these Investigators oversee a number of postdocs and graduate students. These plus our other scientific collaborators will give our framework instantaneous exposure to a relatively large user base.
3. SOFTWARE ENGINEERING PLAN

From the development of PKDGRAV and GASOLINE, our team has demonstrated experience in the design and implementation of modular, extensible software. Although not written in an object-oriented language, these codes were designed with an object oriented flavor. The success of our design is evident in the ease with which new functionality (such as the collision detection code for Solar System simulations) was added. This experience will be invaluable for the design and implementation of the proposed framework. In achieving our current success, we have assembled individual ideas that are logical code improvements but are awkward to incorporate without substantial redesign. When combined as the N-Chilada framework, these ideas are expressed naturally and efficiently.

3.1. Developing requirements

The process of developing requirements is already well underway. Our first workshop established among the collaborators the scope of the project and how it will fit into the various needs of the Co-Investigators. At the second workshop, we introduced a very preliminary design for critique and feedback. At this workshop, we recognized the need for a more continuous collaboration, as well as the need for documenting our progress, keeping track of issues as they arise, and allowing feedback.

To this end we have established a collaboration Web site using the TWiki (http://TWiki.- SourceForge.net/) web based collaboration tool. This web site serves as a clearinghouse for a number of design issues including the following.

- Enumerating the types of scientific project we envision enabling.
- Reviews of other software packages and frameworks that may compete with or complement N-Chilada.
  This is a critical discussion area where we evaluated the merits of starting with a pre-existing framework vs. creating our own. Our conclusions are clear: nobody is doing what we want to do.
- Reviews of software that might be useful for the development of N-Chilada.
- Discussion of data formats: the design requirements of our internal and external data formats, which of the standard formats we would support, and the suitability of co-opting any of the standard formats as our own external format.
- The actual design requirements of our framework.

Face-to-face meetings will be critical in completing the design and insuring that the requirements of the Co-Investigators are met. Pending funding of this proposal, we will schedule a design review meeting in Seattle during August, 2001. The core architecture team will make a formal presentation of their design to the collaboration for critique and feedback. API's will be defined at that time so that actual software development can proceed.

2Although not strictly necessary for the evaluation of this proposal, reviewers are invited to browse our site at http://juno.astro.washington.edu/twiki/bin/view/Nchilada/WebHome. The guest user id is NchiladaCollaborator with password: lakeview
3.2. Software development

After the core design is complete, the framework as it stands is sufficiently modular to allow up to ten code authors to work on it at once without conflicts. Individual components that can be worked on independently include:

- The low-level machine dependent library
- The I/O subsystem
- Generic algorithms for binary trees
- Generic algorithms for oct-trees
- Generic algorithms for grids
- Support for the scripting language
- The application interface
- A visualization application

The first goal is to get a working prototype. The easiest route to this goal is a reworking of **PKDGRAV** by recasting the current components into the **N-Chilada** API.

To coordinate the efforts of multiple code authors, we currently are using CVS (**http://www.-cvshome.org/**) with great success. With **PKDGRAV** we were able to coordinate the efforts of four authors (Quinn, Stadel, Wadsley and Richardson) working concurrently. With the more modular nature of **N-Chilada**, we can easily double the number of authors. CVS also provides us with a mechanism for code auditing: the revision history will allow us to track the contributions from each author. Network transparency means we have easily accommodated code development happening at multiple institutions. However, we will stipulate that any code “check-ins” to our main development trunk will be first reviewed by one of the core architects.

The primary development environment will be the standard C/C++ environment on Linux PCs and DEC/Alpha clusters. This will lead to POSIX threads on SMP Linux machines and MPI on Beowulf clusters as our primary parallel environment. Note that this means that our software will be explicitly designed for these low cost clusters that are generally available to the community. However, any solution that works well on these platforms is guaranteed to work superbly on high-cost, high-bandwidth, low-latency systems. Development on more massively parallel machines will depend on availability of quick turn-around time on the Scalable Testbed. We have been very successful in using the current generation supercomputer at Goddard, and we are therefore very willing to work with the ESS applications support staff in order to speed development on the Scalable Testbed.

3.3. Documentation

Since we are building a framework for parallel computation, the key documentation is for the interfaces and classes we provide. There are a number of tools to automate this process such as **DOC++** for C++, or **Javadoc** for Java. We will evaluate and choose one of these systems as part of our design process. The documentation will be one of the main concerns when code is reviewed for “check-in”.
Another essential piece of documentation is a tutorial to get new users started and answer their questions about the system. There are a number of software tools such as Faq-O-Matic (http://faqomatic.sourceforge.net) which helps automate the process of generating such a “Frequently Asked Questions” document.

3.4. Testing and validation

We have a number of standard tests with which to validate our code. For gravity we first check force errors against a direct summation code with a small number of particles. (32,000) Then there are a number of test simulations of scientific significance for which the answer is known such as the evolution of a globular cluster and the growth of density fluctuations in the early Universe. Finally, there are full-scale simulations of moderate size (1 million particles) which we have run many times with varying parameters. This leads to a total system “stress test” of the integrated framework. Likewise, for other application areas we develop tests at all levels. For gas dynamics we run the standard shock tube tests as well as the more cosmologically interesting spherical infall test.

These tests will be broken up into to two types. The force accuracy test and similar low level tests will be run automatically as part of the software build process. The more sophisticated end-to-end tests such as full-blown simulations would be performed as part of the code release process.

In the end, it is correct scientific results that will be the ultimate verification of the framework. Therefore, tests will be performed by all members of the collaboration as they begin to use the framework to do their science. Communication back to the developers is a key issue for making these tests effective. This will be facilitated by the use of available bug tracking software such as GNATS (http://sources.redhat.com/gnats/). The available Web based front ends for this software will make it easy for the testers to provide feedback to the developers. Communication back to the users from the developers will be accomplished both on the TWiki website, and via standard e-mail lists.

In addition there will be more formal feedback. One function of the semi-annual workshops will be for the scientific Co-Is to perform a review of the framework. This will provide valuable feedback to the core developers, and will ensure that the design requirements are being met.

3.5. Release and maintenance

We plan to fully release our sources to the public, as we have done in the past with our other tools such as TIPSY and SMOOTH. We can accomplish the distribution trivially by opening our CVS repository to the world for read access. However, we will also “tag” release versions so that end users will have a stable product to work with. CVS naturally allows branching so that subsequent bug fixes can be applied to the release version without disturbing the main development.

With a release of a product there comes the difficulty of supporting a variety of platforms with minor differences. We have successfully used the GNU autoconf utility (http://sources.redhat.com/autoconf) to generate scripts to automatically configure the software for a given platform.

Even with complete documentation, new users unfamiliar with the software will have difficulty getting started because they don’t know where to start looking for information. One standard solution is to generate a “Knowledge base”: a searchable database of problem-solution pairs. Our Web collaboration software has the capability of easily constructing such a database incrementally using a Web form. Of course, the bug-tracking software and the mailing lists will be available to users outside the collaboration to provide bug reports and feedback.
4. REFERENCES

REFERENCES


Richardson, D.C., Quinn, T., Stadel, J. & Lake, G., *Direct Large-Scale N-body Simulations of Planetary Dynamics*, 2000, Icarus, 143, 45

5. BIOGRAPHICAL SKETCHES

Personnel list:

<table>
<thead>
<tr>
<th>Name</th>
<th>Core Architect</th>
<th>developer</th>
<th>power user</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thomas Quinn</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Arif Babul</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Julian Borrell</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Hugh Couchman</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Fabio Governato</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Cathy Horellou</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Andrew Markiel</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Ben Moore</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Julio Navarro</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Derek Richardson</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Joachim Stadel</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Mattias Steinmetz</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rob Thacker</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>James Wadsley</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Note: not included in the above table are numerous postdocs and graduate students working for the Investigators at their respective institutions. For example at the University of Washington Quinn has the following post-docs:

- Lucio Mayer
- Sebastiano Ghigna

and the following graduate students:

- Rory Barnes
- Zoe Leinhardt
- Graham Lufkin
- Darren Reed
- Chris Stawarz
- Andrew West

all who will be users of the *N-Chilada* framework.
5.1. Curriculum Vitae: Thomas Quinn

Education:
Ph.D. in Astrophysics, Princeton University, October, 1986.
B.S. in Engineering Physics, Lehigh University, January, 1982.

Employment History, 1986–present:
1998–present: Research Associate Professor, University of Washington.

Selected Publications relevant to proposal:
5.2. Curriculum Vitae: Derek Richardson

Curriculum Vitae for Derek Charles Richardson

Ph.D. (CANTAB) 1993, University of Cambridge

Department of Astronomy  Phone:  (301) 405-8786
University of Maryland  Fax:  (301) 314-9067
College Park, MD  E-mail: dcr@astro.umd.edu

Recent Employment

2000–  Assistant Professor, Department of Astronomy, Univ. of Maryland at College Park.

1999–00  Research Assistant Professor, Department of Astronomy, Univ. of Washington.

1996–99  Research Associate, Department of Astronomy, University of Washington.

Recent Related Publications


Recent Related Abstracts


Lake, G., T. Quinn, D. C. Richardson, and J. Stadel 2000. Speedup to virtual petaflops using adaptive potential solvers and integrators for gravitational systems. SPEEDUP 12, 53–60.


5.3. Curriculum Vitae: Arif Babul

Academic Appointments

- Visiting Associate Professor, University of Washington  July 2000–Aug. 2000
- Associate Professor of Physics and Astronomy, University of Victoria  July 1999–
- Visiting Associate Professor, University of Washington  July 1999–Aug. 1999
- Assistant Professor of Physics and Astronomy, University of Victoria  July 1997–June 1999
- Visiting Associate Professor, University of Washington  July 1998–Aug. 1998
- Assistant Professor of Physics and Astronomy, New York University  July 1994–June 1997
- Postdoctoral Fellow, Canadian Institute for Theoretical Astrophysics  Sept 1991–Aug 1994

Education

- Ph.D. (Astrophysics), Princeton University  1989
- B.A.Sc. – Dean’s Honour Roll (Engineering Science), University of Toronto  1985

Fellowships and Honors (abridged)

- Canada Foundation for Innovation Researcher  1999
- Judge Frances Bergan Career Development Award  1995
- CITA National Fellowship (declined)  1993
- NATO Science Fellowship  1989–1991
- Princeton University Pierce Prize in Astrophysics  1987–1988
- NSERC Postgraduate Scholarship  1985–1989
- Princeton University Merit Prize  1985–1986

List of Relevant publications

5.4. Curriculum Vitae: Julio F. Navarro

**Address:** Department of Physics and Astronomy, University of Victoria, Victoria, BC, V8P 1A1, CANADA. (E-mail: jfn@uvic.ca)

**Born:** October 12, 1962. Santiago del Estero, Argentina.


**Present Position:** 4/1998 - : Assistant Professor. University of Victoria

**Previous Positions:**

**Memberships:** Canadian Astronomical Society. International Astronomical Union.

**Fellowships and Awards:**

**Selected Publications:**
5.5. Curriculum Vitae: Matthias Steinmetz

- **Address**: Steward Observatory, The University of Arizona, Tucson, AZ 85721. (Email: msteinmetz@as.arizona.edu)


- **EMPLOYMENT**:
  - 8/2000 - : Associate Professor and Associate Astronomer, Steward Observatory. University of Arizona
  - 2/1997 - 7/2000: Assistant Professor and Assistant Astronomer, Steward Observatory. University of Arizona

- **Fellowships and Awards**:
  - 1987-91: University Scholarship (“Studienstiftung des Deutschen Volkes”, (German National Fellowship Foundation))
  - 1993: Otto-Hahn Medal of the Max-Planck-Gesellschaft (associated fellowship spent at: Department of Astronomy, University of California, Berkeley, 1996)
  - 1998: Alfred P. Sloan fellowship
  - 1998: David and Lucile Packard fellowship

- **Publications relevant to the proposal**:
5.6. **Curriculum Vitae: Fabio Governato**

**OFFICE ADDRESS**

Osservatorio Astronomico di Brera – Milano  
via Bianchi 46  
Merate (LC)  
23807 Italy

*Phone:* (+39) 039 9991146  
*Fax:* (+39) 039 9991160  
*Home:* (+39) 02 2365264  
*E-mail:* fabio@merate.mi.astro.it  
www.merate.mi.astro.it/~fabio/index.html

**BIOGRAPHICAL DATA**

*Date of Birth:* March 13, 1966  
*Place of Birth:* La Spezia, Italy  
*Citizenship:* Italian

**EDUCATION**

1995, Ph.D., Astronomy II University of Rome “Tor Vergata”  
1991, Laurea in Physics University of Milan

**EMPLOYMENT**

2000–present  Researcher Osservatorio Astronomico di Brera, It  
1999–00  Research Staff, Osservatorio Astronomico di Brera, It  
1997–99  Research Fellow, Dept. of Physics, Univ. of Durham, UK  
1996–97  Research Associate, Dept. of Astronomy, Univ. of Washington. Seattle, WA  
1996  Visiting Scientist, SISSA, Trieste, It

**TEACHING EXPERIENCE**

1999–  Graduate supervisor  
1998  Graduate Course on “Gravitational Instabilities and Cosmic Structure Formation”  
1997  Undergraduate supervisor

**OTHER PROFESSIONAL EXPERIENCE**

* Organizer of scientific seminars at the Observatory.  

**PUBLICATIONS**

* 15 refereed papers  
* 10 conference papers

**RESEARCH INTERESTS**

* Galaxy & Cluster Formation  
* Numerical Simulations
5.7. Curriculum Vitae: Ben Moore

PERSONAL DETAILS
Name: Ben Moore
Date of birth: 28th May 1966
Nationality: English
Address: Department of Physics
Durham University
Durham, DH1 3LE
England
Tel: +44 191 374 4613
Fax: +44 191 374 7465
Email: ben.moore@durham.ac.uk
HTTP://www.nbody.net

SCHOOL EDUCATION
1977 to 1985
Woldgate School, Pocklington, UK
1983
9 GCE ‘O’-Levels
1984
1 GCE ‘AO’-Level
1985
3 GCE ‘A’-Levels

UNDERGRADUATE STUDY
1985 to 1988
University of Newcastle Upon Tyne
B.Sc. Degree
Astronomy and Astrophysics, (First Class Honours)

GRADUATE RESEARCH
1988 to 1991
University of Durham
Ph.D. Thesis Title
Groups, Clusters and Super-Clusters of Galaxies.
Thesis Supervisor
Prof. C.S. Frenk

EMPLOYMENT HISTORY
April 1991 to March 1993
NATO Fellow, UC Berkeley
April 1993 to April 1994
Berkeley Astronomy/CfPA Post-Doc.
May 1994 to October 1995
Research Associate, University of Washington, Seattle
October 1995 to April 1996
Visiting Researcher, UC Berkeley, California
May 1996 – November 2005
Royal Society Research Fellow, Durham University, UK
5.8. Curriculum Vitae: Hugh Couchman

16th November, 2000

1. NAME: Hugh Mark Paasikivi COUCHMAN

POSITION: Professor, Department of Physics and Astronomy
McMaster University
Hamilton, ON
CANADA, L8S 4M1
(905) 525-9140 x27860; fax (905) 546-1252  couchman@mcmaster.ca

2. EDUCATION:

Ph.D. Cambridge University Institute of Astronomy 1987
M.A. Cambridge University King’s College 1984
Certificate of Advanced Study in Mathematics Cambridge University Applied Maths. and
Theoretical Physics 1982
B.A. Cambridge University King’s College (Physics) 1981

3. EMPLOYMENT HISTORY:

Jul. 1999 — 1999  Professor  Physics & Astronomy  McMaster University
Jul. 1994 — Jun. 1999  Associate Professor  Astronomy†  University of Western Ontario
Jul. 1991 — Jun. 1994  Assistant Professor  Astronomy  University of Western Ontario
Sep. 1988 — Jun. 1991  Assistant Professor  Astronomy  University of Toronto
Oct. 1986 — Aug. 1988  Postdoctoral Fellow  CITA  University of Toronto
†Physics & Astronomy since 1996

4. HONOURS, AWARDS, FELLOWSHIPS & PROFESSIONAL ACTIVITIES:

Fellow, Canadian Institute for Advanced Research, Cosmology and Gravitation, 1999–2002
Florence Bucke Science prize, 1999 (awarded annually by the Faculty of Science, UWO,
to recognize excellence in research)
Specialist Editor (Gravitational, Relativistic and Astrophysics), Computer Physics
Communications, Elsevier, Amsterdam, 1996–
Chair, CASCA ad hoc computing sub-committee of the NRC Long-Range Planning,
Panel on Astronomy, 1998–99

5. GRADUATE SUPERVISION:

Ph.D.  T. Fuller  Sep 1996–
R. Thacker  May 1997–Sep 1999  Postdoc, Berkeley

6. RESEARCH FUNDING:

Over $350,000 in direct support and $4,000,000 in collaborative computer support (last 5 years).
One of three principal authors of recently funded $17,000,000 SHARC-Net grid computing initiative in S. Ontario.

7. PUBLICATIONS:

Over 40 publication in refereed journals; two book chapters; “Hydra” and “AP³M” simulation
software used by over 20 groups worldwide.
5.9. Curriculum Vitae: Rob Thacker

Academic Details

Degres
1992: University of Nottingham, BSc in Mathematical Physics – 1.
1993: University of London, MSc in Mathematics – Pass with distinction.
   Advisor: Dr David Robinson
1999: University of Alberta, PhD, ‘Simulation of Galaxy Formation’
   Advisors: Dr Don Page & Dr Hugh Couchman (McMaster University).

Employment
1999-01: Postdoctoral Research Associate, University of California at Berkeley.

Current Institution
Department of Astronomy,
601 Campbell Hall,
University of California at Berkeley,
Berkeley, CA,
94720.
tel: 510-643-1499  email: thacker@astro.berkeley.edu

Project-Related Publications

1. ‘Simulating Galaxy Formation on SMPs’, Robert J. Thacker, Hugh M. P. Couchman,
   and Frazer R. Pearce, in High Performance Computing Systems and Applications 1998,

2. ‘Smoothed Particle Hydrodynamics in Cosmology: A Comparison of Implementations’,
   Robert J. Thacker, Eric R. Tittley, Frazer R. Pearce, Hugh M. P. Couchman and Peter

3. ‘Structure Formation from Numerical Simulations’, Robert J. Thacker and Marc Davis,
   in The Hy-Redshift Universe: Galaxy Formation and Evolution at High Redshift, ASP

   (2000)

Recent Collaborators

- Hugh Couchman, Department of Physics, McMaster University.
- Marc Davis, Department of Astronomy, University of California at Berkeley.
- Frazer Pearce, Department of Physics, University of Durham.
5.10. **Curriculum Vitae: James Wadsley**

**Current Address:** Department of Physics & Astronomy, McMaster University
1280 Main Street West, Hamilton, Ontario
L8S 4M1, CANADA

**email:**
wadsley@physics.mcmaster.ca

**Education:**

Ph.D. Astronomy, February 1998, J.R. Bond, CITA, University of Toronto, Canada

B.Sc. Hons. Applied Mathematics, 1991, Monash University, Melbourne, Australia

**Employment**

September 2000 - present: Postdoctoral Research Associate (CITA National Fellow), Department of Physics & Astronomy, McMaster University

March 1998 - August 2000: Postdoctoral Research Associate, Department of Astronomy, University of Washington, Seattle, USA

**Recent Presentations**

“Shock Tracking with SPH and WDM Galaxy Formation”, Kingston 2000 Meeting, August 26-30, 2000, Toronto

“Galaxy Formation in the Cosmological Context with GASOLINE”, Victoria Computational Cosmology Conference, Victoria, BC, August 21-25, 2000


“Baryons, Helium and Warm Dark Matter”, Seminar, University of California San Diego, 2000

“Importance Sampling applied to the Lyman-α Forest”, Colloquium, University of Milan, 2000

**Recent Papers**


5.11. Curriculum Vitae: Andrew Markiel

CONTACT INFORMATION

Department of Astronomy  phone: (206)-543-2604
University of Washington  fax: (206)-685-0403
Box 351580  email: markiel@astro.washington.edu
Seattle, WA 98195-1580  www: www-hpcc.astro.washington.edu/staff/markiel

EDUCATION

Ph.D. Physics and Astronomy  University of Rochester, 1999
  Thesis: Solar and Stellar Dynamo Models
  Advisors: John H. Thomas and Hugh M. Van Horn

M.A. Physics and Astronomy  University of Rochester, 1993

B.S. Physics  Carnegie Mellon University, 1991

CURRENT POSITION

1999-present  Research Staff, Department of Astronomy, University of Washington

RESEARCH INTERESTS

software for scientific data analysis and visualization; N-body simulations; magnetohydrodynamics

Principal Current Project:

N-Vision, an object oriented 3D visualization package for analyzing very large particle simulations. Prototype to be released November 2000.


5.12. Curriculum Vitae: Julian Borrill

Degrees
1984: University of Cambridge, BA in Mathematics & Political Science.
1990: University of London, MSc in Astrophysics.
1990: University of London, MSc in Information Technology.
1993: University of Sussex, DPhil in Theoretical Physics.

Employment
1993 - 95: Postdoctoral Research Associate, Imperial College, University of London.
1995 - 97: Postdoctoral Research Associate, Dartmouth College.
1997 - 99: Visiting Postdoctoral Research Fellow, Lawrence Berkeley National Laboratory.
1999 - : Staff Scientist, NERSC, Lawrence Berkeley National Laboratory
          & Center for Particle Astrophysics, UC Berkeley.

Current Institutions
Center for Particle Astrophysics,  Scientific Computing Group,
Le Conte Hall,                   National Energy Research Scientific Computing Center,
University of California at Berkeley, Lawrence Berkeley National Laboratory,
Berkeley, CA 94720.            Berkeley, CA 94720.
tel: 510 642 8932               510 486 7308

email: borrill@cfpa.berkeley.edu
www: http://cfpa.berkeley.edu/~borrill/

Selected Publications


5.13. Curriculum Vitae: Cathy Horellou

Curriculum Vitae - Cathy Horellou

Onsala Space Observatory Email: horellou@oso.chalmers.se
Chalmers University of Technology Tel: +46 31 772 5504
S-439 92 Onsala Fax:+46 31 772 5590
Sweden Internet: http://www.oso.chalmers.se/~horellou

Identity

Date and place of birth: 04-02-1967 in Quimper (France). Citizenship: French.
Languages: French, german, english, breton, swedish.

Education

1994 Ph.D. in Astrophysics and Space Techniques. The Interstellar Medium of Ring Galaxies.
        Thesis advisors: F. Casoli and F. Combes.
        Université de Paris VII, Obs. de Paris-Meudon, France.
        Max-Planck Inst. für Radioastronomie, Bonn, Germany.
1988 "Diplomprüfung" in Physics.

Employment

Since July 1998 Research associate at Onsala Space Observatory.
August 1995-December 1995 Astronomer on duty at SEST (Chile) on the Swedish staff.

Research interests

Galactic dynamics. Numerical simulations of colliding galaxies.
Gas content, distribution and kinematics, relationship with star-formation activity.
Magnetic fields in galaxies.

Supercomputers and telescopes used

Cray Supercomputer at Orsay (France) and SGI-2000 at the National Supercomputer Center in Linköping (Sweden).
Optical telescope: ESO 1.52m (Chile).
Infrared and millimeter band: ISO, SEST (Chile), Onsala 20m radiotelescope (Sweden), IRAM 30m radiotelescope (Spain).
Centimeter band: Effelsberg 100m (Germany), Nançay (France), Parkes (Australia), Very Large Array (USA), Australian Telescope Compact Array.
6. MILESTONES/SCHEDULE/COST

6.1. Performance metrics

In the table below we quote our performance metrics in terms of “effective particles per second”, which we believe is an accurate way of describing how fast we can accomplish a science goal with our application. In a gravitational simulation, the measure of performance is how fast one can evaluated the forces on the particles in order to update their positions and velocities. However, the timescales in a gravitational system scale as $1/\sqrt{\rho}$, so that a simulation with a large dynamic range in density (that is, the type of simulation our algorithms are designed to handle) also has a large range in timescale. This implies that some particles do not have to have their forces evaluated as often as others in order to accurately follow their orbits. If the performance metric is simply “Gigaflops” or the speed at which force evaluations can be accomplished then an algorithm that does a better job at multiple timestepping, and therefore accomplished the simulation faster, would show no improvement, and indeed, probably a performance drop (ON FIXED HARDWARE; of course we will take full advantage of faster and more massively parallel hardware.). We therefore propose a metric where we integrate our system for a time comparable to the longest dynamical time, but count every particle update as a full force evaluation. This type of criteria is crucial to the science we wish to accomplish, as we wish to use our framework not only to bring our parallel algorithms to new applications, but also for the rapid improvement of those same algorithms. As shown in figure 1 better algorithms will go far in increasing the science rate.
<table>
<thead>
<tr>
<th>Milestone Label #</th>
<th>Milestone Title</th>
<th>Expected completion date</th>
<th>Advance Payment Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 A</td>
<td>Deliver software engineering plan for developing N-Chilada. This will be a modification of the plan described in this proposal.</td>
<td>Jul. 2001</td>
<td>$30,000</td>
</tr>
<tr>
<td>2 E</td>
<td>Baseline performance evaluation of PKDGRAV and GASOLINE. The metric will be “effective particles per second”. Scaling curves will be produced out to 512 processors (depending on availability of the Teraflops Scalable Testbed) and posted to the Web. First public release of PKDGRAV and GASOLINE via submission to the HPCC/ESS software repository, and via public access to our CVS repository.</td>
<td>Aug. 2001</td>
<td>$200,000</td>
</tr>
<tr>
<td>3 -</td>
<td>Install 16-node 32 processor Intel Itanium + 16 node 64 processor Intel Xeon Linux cluster at University of Washington. The processors will be donated by Intel; the value of this milestone is set by the purchase and installation of a low-latency gigabit switch.</td>
<td>Sept. 2001</td>
<td>$35,956</td>
</tr>
<tr>
<td>4 H</td>
<td>Complete preliminary design of N-Chilada framework. Design and APIs distributed to community members for review and feedback. Community members include: Babul, Navarro, Moore, Governato, Horellou, Steinmetz, Borrill, Jandhyala and Couchman, along with their coworkers.</td>
<td>Sept. 2001</td>
<td>$300,000</td>
</tr>
<tr>
<td>5 B</td>
<td>Submit Year 1 Annual Report to ESS via Web.</td>
<td>Jun. 2002</td>
<td>$90,000</td>
</tr>
<tr>
<td>6 F</td>
<td>Demonstrate 3x improved performance of PKDGRAV and GASOLINE over “E” baseline in terms of effective particles per second. Improved code will be available from our CVS repository and submitted to the HPCC/ESS software repository.</td>
<td>Sept. 2002</td>
<td>$200,000</td>
</tr>
<tr>
<td>7 -</td>
<td>Demonstrate performance of PKDGRAV and GASOLINE on installed Beowulf.</td>
<td>Sept. 2002</td>
<td>$100,000</td>
</tr>
<tr>
<td>8 I</td>
<td>Complete prototype of N-Chilada framework. Implement PKDGRAV and GASOLINE within this framework. Make framework source available from our CVS repository, and submit to HPCC/ESS software repository.</td>
<td>Sept. 2002</td>
<td>$300,000</td>
</tr>
<tr>
<td>9 C</td>
<td>Submit Year 2 Annual Report to ESS via Web.</td>
<td>Jun. 2003</td>
<td>$90,000</td>
</tr>
<tr>
<td>Milestone Label #</td>
<td><strong>Milestone Title</strong></td>
<td>Expected completion date</td>
<td>Advance Payment Amount</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>10 G</td>
<td>Demonstrate 9x improved performance of PKDGRAV and GASOLINE over “E” baseline in terms of effective particles per second. Improved code will be available from our CVS repository and submitted to the HPCC/ESS software repository.</td>
<td>Aug. 2003</td>
<td>$200,000</td>
</tr>
<tr>
<td>11 J</td>
<td>Demonstrate full interoperability of PKDGRAV, GASOLINE, and HYDRA within the N-Chilada framework. Significant portions of TIPSY will also be ported to the framework.</td>
<td>Aug. 2003</td>
<td>$300,000</td>
</tr>
<tr>
<td>12 K</td>
<td>Port N-Chilada to a number of different architectures. These may include Alpha-based Beowulfs, IBM-SP machines, and SGI machines. Demonstrate performance on these architectures. Deliver N-Chilada to groups at U. Victoria, McMaster University, University of Arizona, University of Durham, U.C. Berkeley, Onsala Space Observatory and Milan Observatory. Establish e-mail list, web site and knowledge base for user feedback and support.</td>
<td>Jun. 2004</td>
<td>$100,000</td>
</tr>
</tbody>
</table>