Proceedings of the 14th IASTED International Conference
PARALLEL AND DISTRIBUTED COMPUTING AND SYSTEMS
November 4-6, 2002, Cambridge, USA

Climate*prediction*.net: Design Principles for Public-Resource Modeling Research

David Stainforth[1], Jamie Kettleborough[2], Andrew Martin[3], Andrew Simpson[3], Richard Gillis[3], Ali Akkas[2], Richard Gault[3], Mat Collins[5], David Gavaghan[3] and Myles Allen[1]
[1]Atmospheric, Oceanic and Planetary Physics, University of Oxford, UK, [2]Rutherford Appleton Laboratory, Oxfordshire, UK, [3]Oxford University Computing Laboratory, [5]Dept. of Meteorology, Reading University, UK.

ABSTRACT

Climate*prediction*.net aims to harness the spare CPU cycles of a million individual users' PCs to run a massive ensemble of climate simulations using an up-to-date, full resolution, three dimensional atmosphere-ocean climate model. The project has many similarities with other public-resource high-throughput activities but is distinctive in a number of ways including the complexity of the computational task, its duration and system demands, participant interaction, and data volume and analysis procedures. These distinctive aspects, together with the documented experience of other Internet-based distributed computing projects, lead to a number of design challenges. The analysis of these, and the resulting architecture, is presented here. The initial development and testing phase is complete and roll-out will begin shortly. It is expected that the issues raised will be of relevance to an increasing number of similar projects.

KEY WORDS
Grid Computing, Modeling and Simulation, Internet Computing, Parallel and Distributed Systems.

1. Introduction

The overall objectives of the climate*prediction*.net project ([1], [2], [3]) are*:*

- To harness the power of idle home and business PCs to provide the first fully probabilistic 50-year forecast of human induced climate change based on a perturbed-physics ensemble simulation with a full-scale three-dimensional atmosphere-ocean general circulation model.
- To enhance public understanding of climate modeling and the nature of uncertainty in climate prediction, enabling secondary- and tertiary-level students to perform their own climate modeling research projects, both as individuals and in distributed research teams.
- To provide a platform for the development and demonstration of peer-to-peer data analysis and visualization software in the context of a massively distributed data archive.
- To demonstrate the potential of distributed computing for Monte Carlo ensemble simulations of chaotic systems with advanced geophysical fluid dynamics computer codes.

The quantification of uncertainty in climate predictions requires of order 1-2 million integrations of a complex climate model. This is beyond the scope of conventional supercomputing facilities but could be achieved using the Grid concept of "high-throughput computing" [4] which has been demonstrated by a number of "public-resource" projects such as SETI@home [5]. The principle is to utilize idle processing capacity on PCs volunteered by businesses and the general public. This project differs from previous projects in a number of ways including: the volume and management of the distributed data generated; the scale of the data collection problem; the complexity, granularity and duration of the computational task; the necessity and desire to provide tools to maintain long term participant interest, and the need to provide a participation package simple enough to be used by inexperienced PC users, possibly with slow Internet connections.

The initial development phase is complete and beta testing is underway with a public launch planned for early 2003. This paper describes, in section 2, the motivation for climate*prediction*.net. In section 3 the factors influencing the overall design are presented. It is anticipated that many of these factors will be similar for other modeling projects wishing to use the high-throughput, public-resource, public interest computing paradigm. Consequently it is believed that the architecture of climate*prediction*.net, described in sections 4, 5, and 6 will serve as an initial attempt at overcoming the generic problems inherent in such computationally intensive distributed computing projects.

Issues of security and experimental integrity are major elements of the project and are intrinsic to the architectural design. However, their importance and complexity are such that they will be presented in a separate future paper.

2. Motivation For Climate*prediction*.net

Recent studies have shown that atmosphere-ocean general circulation models (AOGCMs) are capable of simulating some large-scale features of present-day climate and recent climate change with remarkable accuracy (e.g. [6]). These models contain, however, large numbers of adjustable parameters many of whose values are poorly constrained by the available data on the processes they are supposed to represent and are known, individually, to have a significant impact on simulated climate. The practical question is therefore: to what extent might a different choice of parameters or parameterizations result in an equally realistic simulation of 20th century climate yet a different forecast for 21st century climate change? This is acknowledged by the Inter-Governmental Panel on Climate Change (IPCC) as one of the key outstanding priorities for climate research.

Since AOGCMs contain large numbers of under-determined parameters whose impact is non-additive [7], there is no alternative to a "brute force" exploration of high-dimensional parameter space. This requires very large numbers of long integrations of a complex climate model. Exhaustive sampling of 5 values of only 9 parameters would require ~2M simulations. Intelligent sampling based on results as they are produced would reduce this figure, but the inclusion of additional parameters and the need to include initial condition and forcing perturbations as well, easily leads to a demand for overall ensembles of 1 to 2 million independent simulations. (See [3] for experimental details. See fig 3 of [3] for results from prototype software.)

Currently available supercomputing resources would be completely inadequate for this task. However, the continuing increase in computing capacity of the average PC has meant that AOGCMs, which until recently could only be sensibly run on supercomputers, can now be run on commonly available desktop machines. This opens up the possibility of carrying out the above research by using high-throughput public-resource computing, which has been demonstrated by a number of projects, such as SETI@home [5], FightAIDS@home and Compute Against Cancer [8]. The most successful have been those that stimulate the public's imagination and interest and in such situations tremendous computing capacity has been accessed at relatively low cost.

A suitable architecture for such a project requires careful design involving a grid of servers, with one or more controlling nodes. Developing such a scalable architecture would be a significant step towards defining a generic grid configuration for similar applications.

3. Challenges Posed By The Project

Despite the experience of other public-resource distributed computing projects there are a large number of specific challenges posed by a modeling initiative of this sort. These challenges are described here, focusing on issues related to the model, the architectural restrictions posed by the size and location of the datasets, the requirements of the software package and the necessity to cater for public involvement All initial development has focused on a package suitable for Windows Operating Systems. This is justified by its ubiquity; statistics from SETI@home show that Windows participants contribute more than ten times more than users of other O/Ss. A detailed comparison of the architecture of public resource projects is planned for a future paper.

3.1. Model Preparation

The first challenge of using high-throughput public-resource computing for a complex modeling simulation, is the need to implement the model in question on personal computers running popular operating systems, in this case Windows. This will often be a non-trivial task. The codes may have been developed over a number of years and be designed to be run on parallel processor supercomputers. Furthermore, user interface and analysis tools are often UNIX based. These may be easy to port to an individual Windows machine, by installing a UNIX shell, but this is an unacceptable solution in the public-resource distributed computing context.

Climate*prediction*.net is using versions of the Hadley Centre Climate Model (HadCM3 ([9],[10]), HadSM3 [7]), the global set-up of the UK Met Office Unified Model (UM) [11]. This is one of the world's foremost climate models and although designed to be run on MPP machines, a single processor version also exists. Nevertheless, compiler limitations and oddities of the code, much of it originating as FORTRAN 77 or even FORTRAN 66, meant that several months were necessary to implement the particular version required, under Linux. To run under Windows required further effort to extract the fundamental model code from the GUI set-up and the wrapper scripts. As a result, the preparation of each specific model version is now done under a Linux implementation which includes various pre-processing modules to create the desired FORTRAN code. This is then transferred to Windows and implemented within the climate*prediction*.net package. The main difficulties encountered in porting the code from Linux to Windows were the identification of suitable compiler options and changes in the way environment variables are used.

3.2. Architectural Design Issues

A basic description of the experiments to be undertaken in climate*prediction*.net is given in [3]. The architecture must allow for a number of criteria, not present in other distributed computing projects but likely to be relevant for future modeling initiatives. In particular, the model requires a large number of input files, many of which are the same for all participants although a few are participant specific. The wide public interest in the subject of the research means that many participants will have only basic computational skills. Consequently, participation must be as simple, transparent and enjoyable as possible. Furthermore, the size of the application and the volume of data produced raises challenges of data management and analysis. These design issues are discussed below and are summarized in Table 1.

3.2.1. Participation

It is important for a project such as this to facilitate as widespread participation as possible. This is necessary both to produce the data required and to raise awareness and stimulate more informed public debate on the associated scientific issues. Furthermore, the global nature of the subject would be appropriately matched by global participation in the experiment.

The registration process personalizes the simulation being undertaken locally by defining the way in which the model physics is perturbed for the given member of the ensemble. This is done by downloading a unique version of a small ASCII input file (FORTRAN namelist) containing participant specific parameter values. However, for future flexibility, e.g. the inclusion of different models, the task of run allocation allows multiple participant-specific files of non-specific formats. Since some of these files may be large and some simulations could have greater CPU demands, the distribution server has the flexibility to build in dynamic run allocation according to the bandwidth and specification of the participant's machine.

Table 1. Design requirements and their implications for code development and structure. (Italicised points are planned for future releases.)

Design Requirement:	Implication
1. Security for participants, the experiment and other stakeholders.	• A range of issues to be described in a future paper on the subject. These include: digital certificates, secure data transfer, integrity checks etc.
2. Software suitable for use in remote areas with infrequent, low speed, modem based Internet connection.	• Main package suitable for CDROM distribution. • Run allocation requires minimal download. • Very small minimum level of data collection. • Efficient post-processing. • Prioritisation of data to be collected.
3. Flexible mechanism for perturbing run specification.	• Server/client interaction allows replacement of any of the model files.
4. Improved stability and error catching.	• Additional code written to monitor the status of the model, catch model failures and restart as appropriate.
5. Ease of participation.	• Simple installation and registration, requiring minimal user interaction.
6. Large volumes of data to be collected.	• Design must allow for multiple data collection servers distributed globally. • *Network and server management should allow for optimised choice of where data is collected.* • *Client/server should allow for "trickle back" of data.*
7. Distributed analysis.	• *Dynamic updating of post-processing to enable increased analysis to be done on the distributed PCs.* • *Development of Globus and Earth System Grid II (ESG II) software for data analysis on distributed servers. ([13])*
8. Flexibility to include other models.	• Separation of user and server interaction processes from model control.
9. Maintaining participant interest.	• Dynamic, stimulating web site. • Ability to visualize model fields while it is running. • Ongoing requirement for expanded visualization and analysis packages for the non-specialist.

A minimum recommended machine specification has been set at 400 MHz and 128 MBytes, to allow for the model's memory footprint of 55 MBytes while not interfering with normal machine operation. Most participants are expected to be connected to the Internet by modems, some with poor quality connections. Consequently, since the minimum size of the software package is 5 MBytes and may be larger for subsequent releases, the main package should be distributable by CDROM, e.g. in collaboration with magazines. Furthermore, registration (i.e. run allocation) and data upload should have low minimum data transfer levels.

3.2.2. Data

Any public-resource, high-throughput modeling initiative will effectively create a huge computational Grid with a slightly novel architecture in which the weakest link is data transfer. During the course of climate*prediction*.net several PetaBytes of data will be produced with at least 500 MBytes held on each participating machine. There are similarities with other Grid and DataGrid projects such as those related to the Large Hadron Collider (LHC) ([12]) but here the data is **produced** in a massively distributed fashion and needs to be gathered together into a distributed archive for analysis. Bandwidth limitations referred to above mean that efficient local post-processing of model data is essential, with as much analysis as possible being distributed, to minimise the volume of data collected. The package should therefore be designed to enable updates of the post-processing module during the course of the experiment. In addition, data files are prioritised so that the minimum data collected can be small (~ 1 Mbyte) while providing the facility for further files to be uploaded from those with faster, cheaper or more reliable connections. An important future development will be the facility to enable "trickle back" of data through the course of the simulation instead of waiting for run completion.

The volume of data collected will be large, particularly during later phases of the experiment when some participants might be returning up to 100 MBytes for a 100 year simulation, over a period of 3-8 months. It has therefore been necessary to design a system to allow for multiple data storage servers distributed globally. Optimisation of such a system with consideration of network traffic, storage capabilities and subsequent data analysis is anticipated to be an intriguing future research topic. Facilities to analyse and visualize this distributed data set will be based on the DataGrid software developed under the US DoE's ESG II (Earth System Grid II) pilot project. This is a powerful starting point, but there will be a need for further developments, particularly relating to more detailed metadata production and simplified user definitions. Techniques for caching, replication and optimisation of analysis location will be based on the experiences of ESG II and other DataGrid architectures, particularly The Globus Toolkit [13]. Although a minimal set of common diagnostics will be collected from all simulations, the fact that additional data files will only be held for a "random" subset of runs will provide additional statistical analysis challenges. Furthermore, visualization and analysis of subsets of the ensemble in multi-dimensional parameter space will also provide challenges. Finally, in the longer term, it is interesting to consider the possibility of analysing the **massively** distributed data set held locally on the participants' PCs.

3.2.3. Software Package

The package of software installed by each participant must fulfil a range of criteria, above and beyond those which may typically be put in place for research models. Fundamentally it must be reliable and easy to use. To this end, the installation process makes minimal changes to the PC set-up and it can be uninstalled easily and completely, to maintain the confidence of the participant community. The model itself runs as an "idle" priority process to avoid interference with other applications.

Ensuring stability and recoverability is a more complex requirement because research models of this nature are designed on the basis of an informed user who can diagnose problems and take the necessary remedial action. In climate*prediction*.net, of course, we can assume no user involvement. Consequently, it has been necessary to build an experiment control "wrapper" routine, which monitors the simulation's progress and handles failures and restarts. The main complexity here is the differentiation between model and non-model failures. The latter include, power loss, participant requested stops, full hard disks, O/S crashes etc., and can be simply restarted from the most recent checkpoint (i.e. restart dump). The former are model crashes, e.g. due to model instabilities, and although unrecoverable, are still useful results from which we would like to receive data. Indeed, in this experiment there is a high likelihood of such instabilities since the model parameters are taking values with which they have not been tested; this is, after all, the point of the experiment.

A further, desirable aspect of the software package is the flexibility to replace the Unified Model with alternative models, in the future. This involves separation of the model and the experimental control, from the user and server interaction code. Such flexibility would also simplify future integration into a generalised distributed computing platform (e.g. [14]).

It would be desirable to design the package for implementation on multiple Operating Systems. Unfortunately, aspects such as simplified installation, dynamic model monitoring and recovery, and visualization require substantial interaction with the O/S and mitigate against such a generalized design. Nevertheless the principles of the architecture will somewhat simplify the development of alternative O/S packages if support can be found for this activity.

3.3. Public Involvement

The key public involvement challenges revolve around the need to keep participants interested and therefore contributing to the project for a long period; certainly a number of months. This requires that participants are actively involved in the experiment and feel a sense of ownership of their simulations. There is therefore a need for a dynamic, personalized web site and visualization software with which participants can view the progress of their simulation, as it develops, on a timestep by timestep basis. In the medium term there are also opportunities for collaborative and peer-to-peer analysis and visualization, and possibilities for mini-research projects in primary, secondary and tertiary education, which may raise demands for additional, related tools.

4. An Internet-Based Architecture

To fulfil the requirements laid out above, an Internet-based architecture has been developed, shown in Figure 1, which we anticipate will be relevant to other similar projects. At the heart of the architecture is the climate model running within the client software. Around this is a set of servers providing a range of services as follows:

- A source for the participant package. This consists of a download server and potential mirror sites.

- A central server (eventually a collection of servers) including:
  - o Web pages providing information on the project, climate research and the status of the experiment, along with personalized details of results in the context of the overall ensemble.
  - o A database recording participants, the runs allocated and the data collected.
  - o A client interaction server with which the clients communicate.
- A number of *upload* servers, used to collect the data generated by the experimental package. The load on these servers may be high, and the communication overhead great, so it will be advantageous to have duplicates distributed around the world.

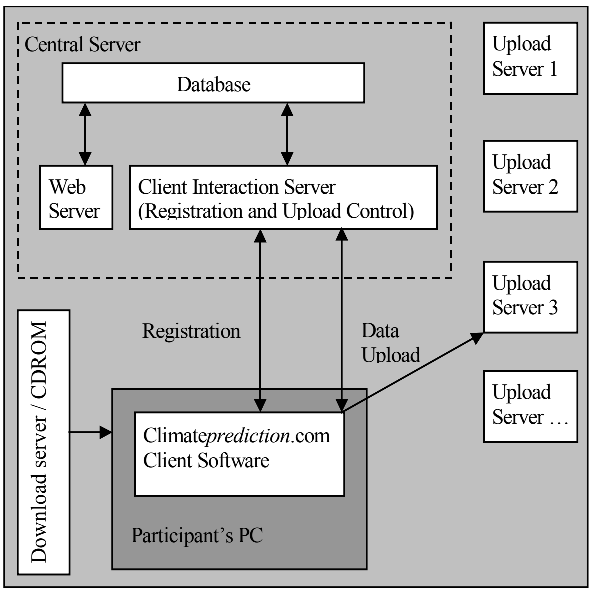

Figure 1: Overall system design.

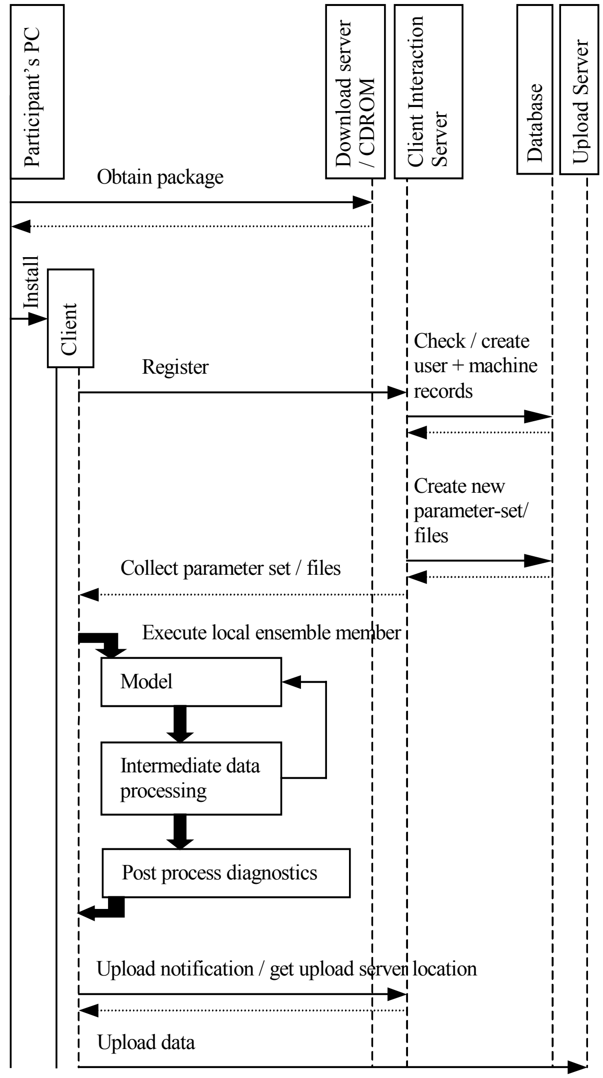

Figure 2: Sequence of events

Figure 2 shows the sequence of events and communications. The software package is self extracting and automatically launches a simple installation process, which itself launches the client software and begins the registration procedure. Details of the participant and his/her machine are collected and sent to the client interaction server. All communications use HTTP or secure HTTP, thus avoiding many firewall problems. The central server checks the database to ensure that the participant's machine has not already received an uncompleted simulation. It then extracts from the database the next set of parameters/initial conditions and provides the client with a list of files it requires to do this simulation, and the files themselves. Prioritisation of run allocations according to network connectivity and machine specifications has not been implemented at this stage but is a high priority for the future.

The client then sets the model running and no further network communication is necessary until the local simulation ends. There is no need for frequent downloads, as are required by most other public-resource high-throughput computing initiatives [5]. However, it is desirable to provide status updates to the central server to keep track of progress and provide web site feedback on the experiment's status. This functionality will be added in the near term.

Since the simulation takes at least several weeks to run, it is desirable to delay the choice of upload server until the end of the local simulation. To facilitate dynamic (de)commissioning of servers, the client reports its readiness for upload to the client interaction server and is then allocated an upload server. As the number of upload servers increases, this process will allow the load on the servers to be managed dynamically.

5. Participant Package

The package downloaded and installed by each participant contains four fundamental modules, illustrated in Fig 3: the client (i.e. the participant and server interface), the experimental control, the visualization and the model. The experimental control is incorporated in the client application to simplify coding and reduce the total package size through access to standard APIs. However, they are designed as separate modules to simplify integration in, or development of, a flexible distributed computing platform in which the client would be generic while the experiment would be a specific application.

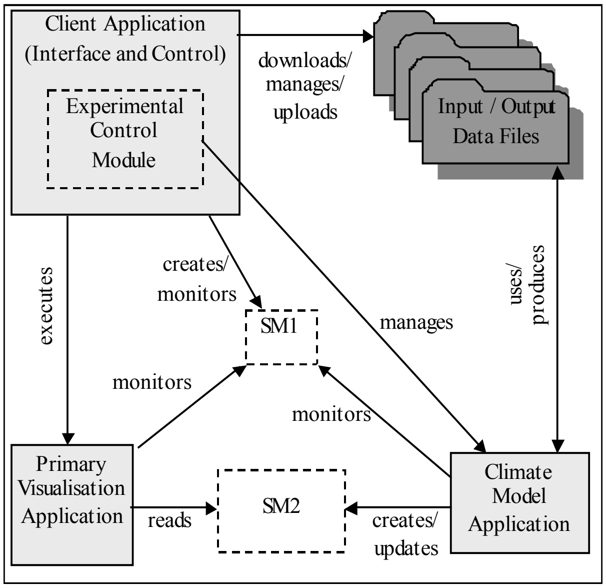

Figure 3: The fundamental processes: client, model and visualization. SM represents shared memory.

The model and visualization remain independent applications as it is anticipated that the model executable will rarely need upgrading between phases of the experiment, while the visualization and the client are anticipated to require several upgrades to incorporate new features and solve bugs. The model executable is large (~2MBytes) and there has been minimal development required beyond the code which has been widely tested by climate researchers. Keeping it separate is therefore a sensible option, as it would place a significant load on participants' patience to download a new version each time they wanted to access the latest upgrade to the client interface or the newest graphics application.

An essential requirement of the design is that participants should be able to view what their model is doing in real time. This raises the problem of how to visualize the model arrays while it is running, using a separate executable and imposing the minimum CPU and I/O demands. The solution developed involves the creation of a section of shared memory (SM2) which is accessed both by the model and the visualization. (See Fig 3) The same procedure is used for communications between the three executables (SM1) and would also simplify the separation of experimental control and client in the case of implementation within a generic distributed computing platform.

5.1. The Client

The client provides an interface between the local package, the participant and the servers. Following installation, the client is launched and begins the registration process described above. New participants are added to the database and the client downloads the necessary files and launches the experimental control thread and thus the model. The server also sends the participant an email with a password, enabling them to logon to the website. In the future this will provide facilities to see how an individual's runs compare with the distribution of results within the whole experiment. The password also enables individuals to register more than one machine under the same user name – an important feature as the competitive nature of distributed computing has been a notable aspect of other projects.

The client also provides facilities to keep track of the status of the local simulation and the files generated, as well as APIs for compression and decompression of files.

5.2. Experimental Control

The experimental control has two principle functions. Firstly, it is required as a wrapper for the model executable to ensure that model crashes are dealt with appropriately and that shut downs and restarts work effectively. Secondly it provides functionality to undertake post-processing of the model output files so that as much useful information as possible can be returned to the upload servers within files of minimal size. This code also facilitates a series of local model runs which together form a sub-experiment within the overall climate*prediction*.net experiment thus broadening the scope of possible research. (See Fig 3 of [3])

5.3. Security

Aspects of security and experimental integrity are critical to the project but are beyond the scope of this paper. They will be presented in a future paper focused on this issue.

5.4. Visualization

The ability for users to view the progress of their simulation is a primary aim and requirement for the project since it is critical to maintaining long term interest as well as raising awareness of the issues being studied. Such a visualization mechanism must be both exciting and informative. Two packages are therefore planned. The first will be distributed with the basic software package, thus requiring it be small (<= 1MByte compressed) and appealing. This includes 3D images of the earth superimposed with model fields such as clouds on different model vertical levels, temperature, rainfall etc, with the fields updating as the model simulation proceeds and allowing options to "zoom" in to areas of interest. A second package is also under development, which will allow more detailed analysis such as conventional 2D plots of surface temperature, rainfall, radiation etc. as well as regional averaging and graphs of timeseries. This package will display results from the diagnostic files. It is anticipated that it will be a crucial part in involving secondary and tertiary educational establishments.

6. Server design

As shown in Figure 1, there are two types of server associated with this project. The central server consists of a client interaction server, a web server, and the main database. The upload servers simply receive and store data and can be replicated at many locations distributed globally. Both are standard web servers and the database uses mySQL. All communication between the client and the servers is via HTTP or HTTP over a secure socket layer, and is always initiated by the client, as one of the security features. The detailed management of these servers and production of metadata is unfortunately beyond the scope of this paper. A particular requirement of the central web server is that it should be able to interrogate the database and produce summary statistics on the progress of the experiment and the input and results of individuals and groups. These could range from CPU time contributed, to years simulated, to global or regional temperature changes simulated in the context of the overall experimental distribution.

7. Future Developments

There are many opportunities to expand on the mechanism described above. In particular, other groups (e.g. [14]) are working on generic platforms for high-throughput, public-resource computing and it is anticipated that suitable collaborations will ensure that such generic platforms will be flexible enough to deal with the complex modeling and public interaction requirements described herein. There are also opportunities to develop the visualization packages within a peer-to-peer context and to provide mechanisms for dynamic updating of the post-processing. Data analysis of a distributed and massively distributed dataset will require substantial effort but should offer significant benefits. It is anticipated that the Globus and Earth System Grid II packages will form the backbone of this work. On a practical level there will be a need to provide simplified installation procedures over intranets since there have been many small and medium sized businesses who have registered their willingness to participate. Finally, dynamic assignment of simulations and upload servers, based on load monitoring and resources will help optimise the processes involved.

8. Conclusions

This paper has outlined the requirements foreseen for any large scale modeling project within a high-throughput public-resource initiative. In particular it has highlighted the demands placed on such projects by public involvement, the need to maintain public interest, and the opportunities for raising public awareness of the scientific issues. Although having a high project preparation cost, the payback is substantial in terms of both the computing power accessed and the possibilities of linking the public more closely with major research initiatives which themselves feedback on social and political decisions.

This paper has focused on the climate*prediction*.net project, but it is hoped that the approach described will be relevant to other, similar projects.

ACKNOWLEDGEMENTS

The authors wish to thank David Anderson for his insights into the experiences of the SETI@home team, and the Met Office and Tessella Support Services plc for their participation in, and support for the project.

REFERENCES

[1] M. R Allen, Do-it-yourself climate prediction, *Nature*, 401:627, 1999.

[2] J.A.Hansen, M. Allen, D. Stainforth, A. Heaps, & P. Stott, Casino-21: Climate Simulation of the 21st Century, *World Resource Review 13*, 2001.

[3] D. Stainforth, J. Kettleborough, M. Allen, M. Collins, A. Heaps & J. Murphy, Distributed Computing for Public Interest Climate Modelling Research, *Computing in Science and Engineering 4*(3): 82-89, 2002.

[4] I. Foster and C. Kesselman, (eds). *The Grid: Blueprint for a new computing infrastructure Chapter 2* (Morgan Kaufman, 1999).

[5] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, & M. Lebofsky, SETI@home - Massively distributed computing for SETI, *Computing in Science and Engineering*, 3 (1): 78-83 Jan-Feb 2001.

[6] P. Stott, S. Tett, G. Jones, M. Allen, J. Mitchell & G. Jenkins, External Control of Twentieth Century Temperature Variations by Natural and Anthropogenic Forcings, *Science 290*, 2000, 2133-2137.

[7] K.D. Williams, C.A. Senior & J.F.B. Mitchell, Transient climate change in the Hadley Centre models: The role of physical processes. *Journal of Climate, 14* (12): 2659-2674 2001.

[8] L. Hand, Computing in Cancer Research, *The Scientist* 15(1), 2001.

[9] C. Gordon, C. Cooper, C.A. Senior, H. Banks, J.M. Gregory, T.C. Johns, J.F.B Mitchell & R.A. Wood, The Simulation of SST, Sea Ice Extents and Ocean Heat Transports in a version of the Hadley Centre Coupled Model Without Flux Adjustments. *Climate Dynamics 16*, 147-168, 2000.

[10] P.M. Cox, R.A. Betts, C.D. Jones, S.A. Spall & I.J. Totterdall, Acceleration of Global Warming Due to Carbon-Cycle Feedbacks in a Coupled Climate Model. *Nature*, 408:184, 2000.

[11] M.J.P. Cullen, The Unified Forecast/Climate Model, *Meteorological Magazine*, 122 (1449), 1993.

[12] URL:http://lhcgrid.web.cern.ch/lhcgrid/

[13] I. Foster & C. Kesselman, Globus: A metacomputing infrastructure toolkit, *Intl J. Supercomputer Applications 11(2):115-128*, (1997)

[14] URL: http://boinc.berkeley.edu/
URL: http://sourceforge.net/projects/boinc/