Storm tracking with remote data and distributed computing

Lizzie S.R. Froude*

Environmental Systems Science Centre (ESSC), University of Reading, Harry Pitt Building, Whiteknights, Reading RG6 6AL, UK

Received 13 January 2007; received in revised form 29 October 2007; accepted 2 November 2007

Abstract

Meteorological datasets are continuously getting larger, as a result of increasing computer power allowing models to be run at higher resolutions and for longer time periods. This is making it increasingly difficult to analyse these datasets at a single location. A web application has been developed to address this problem. The web application allows storm identification and tracking software to be executed from a web browser. It accesses remote datasets using the OPeNDAP protocol and makes use of distributed computing techniques with Condor. The web application currently enables users to compute storm tracks from the National Centers for Environmental Prediction (NCEP) re-analysis and ensemble prediction datasets, which are both archived in the USA. A list of jobs can be constructed and executed across multiple computers to reduce computation time. The progress of each job can be monitored and once completed, the computed storm tracks can be downloaded and plotted in a web browser. Applications of the web application are also discussed. © 2008 Elsevier Ltd. All rights reserved.

Keywords: OPeNDAP; Condor; eScience; Meteorology; Ensemble prediction; Re-analysis

1. Introduction

Meteorological datasets such as those obtained from Numerical Weather Prediction (NWP) or re-analysis are continuously getting larger. This is mainly due to increased computer power, which allows models to be integrated at higher resolutions and for longer time periods. In the past NWP involved the integration of a single model from a single initial state. More recently ensemble prediction (e.g. Molteni et al., 1996; Toth and Kalnay, 1993) techniques have been introduced, in which multiple integrations of a model (or multiple models) are performed. It is clear that the output from Ensemble Prediction Systems (EPS) will constitute dramatically larger datasets than those obtained from the older deterministic approach. Re-analysis datasets (e.g. Uppala et al., 2005; Kalnay et al., 1996) have always been large, but since they are now being generated for longer time periods and at higher resolutions they are also increasing in size.

The increase in size of such datasets has resulted in more distributed archiving and it is consequently becoming more difficult to analyse these datasets at a single location. Storing the required data locally may not be possible because of the enormous amounts of disk space required and transferring the data from its remote source to a local resource can be incredibly time consuming. The vast amount of CPU required to process and analyse such large amounts of data presents another difficulty and the use of one computer for such a task may be completely infeasible.

---

*Tel.: + 44 118 378 5216; fax: + 44 118 931 6413.
E-mail address: lsrf@mail.nerc-essc.ac.uk

0098-3004/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cageo.2007.11.004
Such difficulties arose recently in a series of studies (Bengtsson et al., 2005; Froude et al., 2007a, b), which investigated the prediction of extratropical cyclones (or storms) by NWP. These studies used the objective identification and tracking software TRACK (Hodges, 1995, 1999a) to identify and track forecast storms. Statistics were then generated to assess the ability of the different models to predict the position, intensity, growth and propagation speed of the storms. These studies analysed some very large datasets; particularly Froude et al. (2007b), which evaluated the performance of the ECMWF (Buizza and Palmer, 1995; Molteni et al., 1996) and National Centers for Environmental Prediction (NCEP, Toth and Kalnay, 1993, 1997) EPS. The difficulties associated with the analysis of these large datasets motivated the development of a TRACK web application, which is described in this paper.

The TRACK program itself (as opposed to the TRACK web application) runs under the UNIX operating system. It identifies cyclones as extrema in a time series of data and then links these points together to form trajectories of the storm tracks (see Hodges, 1994, for further details). Extratropical cyclones are extremely important to the day-to-day weather of the midlatitudes; being associated with unsettled, stormy, wet and windy weather. They provide essential rainfall, but can also cause large amounts of damage via flooding and strong winds. It is therefore important to study both the prediction and climatology of these weather systems. The TRACK program provides useful diagnostics for such studies (Froude et al., 2007a, b; Hoskins and Hodges, 2002, 2005).

The TRACK web application allows the TRACK program to be executed from a web browser, with remotely stored datasets, using distributed computing. The service currently enables users to compute storm tracks from the NCEP re-analysis (Kalnay et al., 1996) and NCEP EPS (Toth and Kalnay, 1993, 1997) datasets, which are both archived in the USA and made freely available via the Internet. A list of jobs can be constructed and executed across multiple computers to reduce computation time. The progress of each job can be monitored and once completed, the computed storm tracks can be downloaded and plotted in a web browser. The web application makes use of existing technologies to access the remote datasets and to perform distributed computing, but links these technologies together in an original way.

The analysis methodology of Froude et al. (2007b) required very large amounts of data, but the methodology did enable detailed information about the prediction of storms to be determined that cannot be determined by other more conventional analysis methodologies which do not require such large amounts of data. Without the use of the eScience methodologies of the web application it would not have been possible to analyse such vast amounts of data and hence determine this new information about the prediction of storms.

There have been numerous other tools developed to make local programs (such as TRACK) accessible via the Internet. Recent trends have included the use of portlets and web services. For example GridSphere\(^1\) provides a portlet-based web portal, which allows web applications to be developed quickly. GridSphere implements the JSR 168 portlet API standard\(^2\) and allows users to make use of Grid technologies. Examples of projects that have used GridSphere include the Telescience TM Project,\(^3\) which provides a portlet-based user interface to biomedical image analysis with a Grid-based infrastructure, and the Tasmanian Partnership for Advanced Computing (TPAC) Digital Library,\(^4\) which allows users to search and navigate Open-source Project for a Network Data Access Protocol (OPeNDAP) (see Section 2.3) datasets.

The other currently popular technology is web services,\(^5\) which allow programs/services to be run remotely across the Internet. They have been used for a number of eScience projects, such as DEWS,\(^6\) which uses web services to display a combination of health, weather and marine information in a web browser. During the initial stages of development of the TRACK web application the use of web services was explored. However, after some investigation we decided not to use web services; this is discussed in Section 2.1.

---


\(^3\)GridSphere portal framework, Telescience Portal Development at the University of California, San Diego. http://www.gridsphere.org/gridsphere/guest/telescience/


There are other general tools, like GridSphere or web services, designed to help developers convert locally run applications to Internet accessible applications. The TRACK web application differs to such tools, as it was written specifically for the TRACK program. This means that the web application is very simple to use as the functions are specifically for just the TRACK program. Other general tools, such as GridSphere, are more complex because they must cater for a wide variety of different applications. Web applications analogous to the TRACK web application, using the same technologies, could easily be developed for other local applications. This paper continues with a detailed description of the web application in Section 2 and a discussion of its application and possible future developments are given in Section 3.

2. Description of web application

In this section the individual components of the TRACK web application are described in turn. Fig. 1 shows a flow chart illustrating how the components fit together and will be referred to throughout the discussion of this section. An overall summary of the web application and description of how the individual components fit together is given at the end of the section.

2.1. Job submission

A web application has been created to allow the TRACK program to be run from a web browser (labelled [1] in Fig. 1) using Java Servlets/Java Server Pages (JSP). The web application allows users to run the TRACK program with NCEP re-analysis (Kalnay et al., 1996) or NCEP EPS (Toth and Kalnay, 1993, 1997) data. If the user chooses the NCEP re-analysis dataset they are presented with an HTML form allowing them to select a time period lying within the period the re-analysis data spans. Since TRACK can identify the cyclones with a variety of meteorological fields, the HTML form allows the user to choose between the fields of mean sea level pressure (MSLP), relative vorticity at the 850-hPa ($\zeta_{850}$), 500-hPa ($\zeta_{500}$) and 250-hPa ($\zeta_{250}$) pressure levels, and geopotential also at the 850-, 500- and 250-hPa pressure levels. They can also choose between the northern and southern hemispheres.

Only a limited amount of NCEP EPS data are available from the remote source at any one time. During 2005 (when the web application was developed), NCEP ran an ensemble forecast of 10 members every 6h (41 members per day) at 0000, 0600, 1200 and 1800 UTC. The output from these integrations were available in grid form at two horizontal resolutions for the past 1 or 2 weeks. A list of the forecast start days available was provided at http://nomad5.ncep.noaa.gov:9090/OPeNDAP/ens/archive and http://nomad5.ncep.noaa.gov:9090/OPeNDAP/enshires/archive for the lower and higher resolution data, respectively. If the user chooses

---

to run TRACK with the EPS dataset, the import tag of the Java Server Pages Standard Tag Library (JSTL)\(^8\) is used to import the HTML from these web pages so that the forecast start dates currently available can be extracted and then displayed to the user as a list of hyperlinks.

When the user selects a start date from the list displayed, another JSP page is executed, to determine which ensemble members are available at which start times. This will usually be all 41 forecasts; however, if the current day is selected then not all of the forecasts may be available. For example if the user selects the current day, and the time is currently 1100 UTC, then clearly the forecasts started from 1200 and 1800 UTC will not be available yet. This JSP page works in the same way as the previous one, using JSTL to import information from the NCEP web pages. All the forecasts available are then listed in a HTML form with check boxes for the user to select. The user can choose between the fields of MSLP, \(\xi_{850}\), \(\xi_{500}\) and \(\xi_{250}\). They can again also choose between the northern and southern hemispheres.

The user is able to construct a list of multiple jobs they wish to run. These are independent jobs, such as different data periods from the re-analysis data or different ensemble members from the EPS, rather than dependent jobs for parallelisation. Fig. 2 shows a screen-shot of the NCEP re-analysis job submission page. Jobs can be added and removed from the job list as required using the “Add to Job List” and “Remove” buttons. This required a technique called session tracking (Hall, 2000, Chapter 9). Each time a client makes a HTTP request (e.g. a user adds a job to their job list) it opens a separate connection to the server (labelled [2] in Fig. 1). A separate record of each users job list therefore needs to be kept and updated as each user adds/removes jobs from their list. Servlets have an HTTP Session API (Application Program Interface), which the web application makes use of by creating a new session object for each user that stores all the jobs currently in their job list. Since a new session object will be created for each new session, each users job list will be stored in a separate session object and it is therefore not necessary for the user to login or identify themselves. The users session objects are updated as each user adds/removes jobs from their list. This session tracking technique is also used.

frequently by online shopping websites for users to add/remove items from their shopping basket. Once the user has finished constructing their job list they can submit their list for processing by clicking the “Run Track” button (see Fig. 2).

As mentioned previously in the introduction, the technology of web services was also considered initially in the development of the web application. During these early stages of development, the TRACK program was deployed as a web service using the Axis servlet.\(^9\) This made it possible to run TRACK across the Internet from a program (not necessarily a web browser) on a remote computer. Since for this particular application it was a requirement that TRACK could be run from a web browser, a java web application was written, which allowed a user to run the TRACK web service from their web browser. To clarify, this web application ran the web service which then ran the TRACK program, whereas the web application described previously just runs the TRACK program directly. This extra layer made the web application slower and the question of whether web services were required was considered. Using web services would offer two possible advantages. Firstly TRACK would not necessarily have to be run from a web browser. Instead it could be run as a web service from another type of program on a remote machine. Secondly, since web services can be linked together, it would be possible to link TRACK with another program such as a data visualisation program (provided this visualisation program was deployed as a web service). Since neither of these possible advantages were likely to be a requirement in the foreseeable future, and computation speed was more of an issue, the web application was developed without using web services.

2.2. Executing multiple jobs with Condor

When a user submits a completed job list it is submitted (by the server, labelled [2] in Fig. 1) for processing to the Condor (Thain et al., 2005) pool (labelled [3] in Fig. 1) in the Environmental Systems Science Centre (ESSC), University of Reading, UK. Condor is a software system that manages a collection of jobs by making use of the computational power of machines over a network. Users can submit a list of multiple jobs to Condor, which chooses where and when to run them. For each job Condor determines if there is a suitable machine available and if there is it begins to run the job on that machine (labelled [4] in Fig. 1). Each job in the users job list is submitted as a separate job to the Condor pool and is run on a different machine. This allows a faster throughput than using just a single machine.

The ESSC Condor pool consists of approximately 15 Sun Solaris machines and 10 Linux machines. The TRACK web application currently runs jobs on the Sun Solaris machines. It was only necessary to install the TRACK program on one of the Sun Solaris machines in the network, rather than every machine. When a list of TRACK jobs are submitted to the Condor pool they can then be run on any of the 15 Sun Solaris machines. It is also possible to run the TRACK program under the Linux operating system. In the future it would be possible to install TRACK on one of the Linux machines in the Condor pool and then modify the web application so that jobs could be run on machines running either operating system. This would require a heterogeneous job submission to Condor, in which two executables are specified so that jobs run on the Sun Solaris machines use the executable compiled for Sun Solaris and the jobs run on the Linux machines use the executable compiled for Linux. Condor is very easy to use, the user just submits a list of jobs and Condor handles all the data transfer between individual machines. For further details see Thain et al. (2005).

2.3. Accessing remote datasets with OPeNDAP

The remote datasets (labelled [6] in Fig. 1) are accessed using the OPeNDAP.\(^{10}\) OPeNDAP allows data to be accessed, in several file formats, over the Internet by using a client-server model in which the client requests some data from an OPeNDAP server and the server replies by returning the data. Data analysis programs, which use data access APIs such as netCDF,\(^{11}\) can be modified so that they can access remotely stored datasets in effectively the same way as locally stored datasets by using a URL instead of a filename (using CURL).\(^{12}\) Such data


\(^{11}\)NetCDF. http://ftp.unidata.ucar.edu/software/netcdf/.

\(^{12}\)CURL. http://curl.haxx.se/.
analysis programs can be converted into OPeNDAP clients by linking them with OPeNDAP versions of the API libraries. OPeNDAP also has a sub-sampling facility, so that a specific part of the data can be requested by appending information to the end of the URL that references the data. This allows the user to download just the parts of the data they require, rather than downloading the entire data file (see below).

The NCEP re-analysis dataset is provided by NOAA-CIRES Climate Diagnostics Center (CDC), Boulder, Colorado, USA from the OPeNDAP server http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/ncep.reanalysis/ and, during the time the web application was developed, limited amounts of the NCEP EPS dataset were available via OPeNDAP from http://nomad5.ncep.noaa.gov/ncepdata/index.html. The TRACK program uses the netCDF API and was re-linked with the OPeNDAP versions of the libraries. It can now be used to compute storm tracks from remote datasets in the same way as locally stored data, but using URLs instead of filenames. The OPeNDAP sub-sampling facility is used to request the meteorological fields and time periods requested by the user from the dataset. For example the URL, http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/ncep.reanalysis/surface/slp.2001.nc points to the NCEP re-analysis MSLP data for 2001. The data file consists of a time array (denoted time) of 1460 (= 365 × 6, i.e. six-hourly time steps from 0000 UTC 1 January 2001 to 1800 UTC 31 December 2001) elements, a latitude array (denoted lat) of 73 elements, a longitude array (denoted lon) of 144 elements and a MSLP grid (denoted slp), of values of MSLP at each latitude–longitude point for each time step. To select just the data for January (i.e. from 0000 UTC 1 January 2001 to 1800 UTC 31 January 2001) the URL, http://www.cdc.noaa.gov/cgi-bin/nph-nc/Datasets/ncep.reanalysis/surface/slp.2001.nc?time[0:123],lat,lon,slp[0:123][0:72][0:143] would be used. This selects the first 124 (= 31 × 4) elements of the time array, all of the elements of the lat and lon arrays and the values of the slp grid at each latitude–longitude point for the first 124 elements of the time array. Each field (such as MSLP) of the NCEP re-analysis data are in a different file and subsetting is therefore not required to select the required field. The NCEP EPS data, on the other hand, is archived with all the fields together in the same file and subsetting is therefore required to select specific fields.

2.4. Aggregation of NCEP re-analysis dataset

The NCEP re-analysis data at CDC is stored in yearly files (January–December). The OPeNDAP sub-sampling facility allows users to select a time period within a given year (i.e. in the same file). It is not, however, possible to select a period that begins in one year and ends in another (e.g. December–February season) because the data for such a period is split across two files. To overcome this problem the OPeNDAP Aggregation Server\(^\text{13}\) (labelled [6] in Fig. 1) was used. This is a Java servlet that can be installed in Tomcat and implements an OPeNDAP server for netCDF files and aggregation datasets. It can be used to create aggregated datasets by effectively merging individual files so they appear as one large file. These individual files do not have to be local files, they can also be remote files that are provided by an OPeNDAP server. Once a number of files have been aggregated to appear as one large file the OPeNDAP sub-sampling facility can be used to access a section of data that overlaps multiple files.

The OPeNDAP Aggregation Server was already installed at ESSC. The NCEP re-analysis dataset was aggregated so that it could be treated as one large 50 year file rather than 50 smaller individual 1 year files. This involved adding some Extensible Mark-up Language (XML) declarations to the Aggregation servers Thematic Realtime Environmental Data Distributed Services (THREDDS) catalog. These declarations specify where the NCEP re-analysis data are located and how to aggregate them. The aggregation of the re-analysis dataset means that the user is able to run TRACK with NCEP re-analysis data from any time period between 1943 and the present.

2.5. Monitoring progress of jobs

The time taken for a TRACK job to run can vary considerably. It depends on many factors, such as the amount of data to be processed and the computer it is running on. It is therefore useful to the user to know how far each of their jobs has progressed and how much longer it may take. Once the user has submitted a list of jobs to Condor they are given a user ID number. This ID number

identifies the session object which stores the users job list (see Section 2.1) and is unique to each user. They can use this number to access a progress page and monitor how each of their jobs is progressing. The page displays a list of the jobs the user has submitted and a description of the stage the job is at. A list of the possible stages is given below:

(i) Job is in the queue.
(ii) Converting winds to vorticity.
(iii) Filtering the data.
(iv) Tracking is running: x\% complete.
(v) Job completed.
(vi) An error occurred when trying to access the data.

Stage (i) is the initial stage when a job has been submitted to Condor and is in Condor’s job queue. The amount of time that the job will sit in the queue depends on how many other jobs have been submitted at that time and the current availability of machines. Stage (ii) only occurs when the user has requested the vorticity field. The NCEP data does not include vorticity and so TRACK converts the wind field to vorticity before it can begin to compute the storm tracks. Before TRACK identifies the cyclones the data are first filtered to remove the large-scale background field (Hoskins and Hodges, 2002). Stage (iii) informs the user that this filtering is taking place. The fourth stage is displayed when the cyclones are being identified and tracked. A percentage of how far the tracking part of the program has got is also given. Stage (v) tells the user when a job has completed. Stage (vi) tells the user when there is an error due to a problem with the server that the data are provided from. The progress page was written as a Java servlet, which checks how far a job has progressed by searching the output files that the job has produced so far. Fig. 3 shows a screen-shot of the progress page. There is a “check progress” button on the progress page that the user can use to update the progress page.

In the future the web application could be modified so that the user could be sent an email notification of when their jobs have completed. This would, however, require the user to enter their email address, which is not necessary at the moment.

---

Fig. 3. Screen-shot of progress page of TRACK web application.
2.6. Downloading and plotting storm tracks

The TRACK program has a graphical facility, based on the UNIRAS API,14 allowing users to plot sets of storm tracks. This facility was incorporated into the web application. Once a job has completed a “Plot Tracks” button appears next to the “Job Completed” statement on the progress page [see job (1) of Fig. 3]. This button executes a Java servlet which runs the graphical facility of TRACK to plot the tracks and produce a JPEG file, which can then be displayed to the user. When the user clicks the “Plot Tracks” button the servlet first checks to see whether the JPEG already exists and if so it will send it straight to the user rather than creating the file again. This reduces the load on the server when a user selects the “Plot Tracks” function, for the same set of storm tracks, more than once. Fig. 4 shows a screen-shot of the plotted tracks in the web browser obtained from the NCEP re-analysis MSLP data for the 1 December 2002–28 February 2003 season (the first job in the job list of Fig. 2). Once all of the jobs in the users job list have finished the user is able to download the output. The output from TRACK is initially stored on a data disk, but when the user requests to download the output (storm tracks) the required files are moved to the server (labelled [7] in Fig. 1). These storm tracks can then be used to generate various types of statistics (e.g. Hoskins and Hodges, 2002; Froude et al., 2007a, b). The storm track data are significantly smaller, in terms of disk space, than the original data the storm tracks were computed from. Accessing the original data remotely rather than downloading it and accessing it locally therefore considerably reduces the amount of disk space required for storage (see Section 3 for further discussion).

We note here that currently if two users submit identical job lists then the same jobs will be run twice and the same output will be stored twice under two different user IDs. This is unlikely to happen at the moment due to the limited number of users. However, if the web application is used by a larger number of users in the future then it would be useful to modify it so that when a user submits a job list it checks first whether any of the jobs have been run before by other users. If they have, then the web

application could inform the user of the user ID associated with the particular jobs and the user could then download the output straight away. This would stop unnecessary computer processing and would also save the user some time.

2.7. Summary of web application

All the individual components of the TRACK web application have now been discussed. In summary, the user constructs a list of TRACK jobs in their web browser (labelled [1] in Fig. 1), which is then sent to the server (labelled [2] in Fig. 1). The server then submits this list of jobs to the Condor pool (labelled [3] in Fig. 1) in ESSC. Condor puts the jobs into a queue and then sends the jobs to different computers (labelled [4] in Fig. 1) as and when they become available. The TRACK program accesses the data using the OPeNDAP protocol. The NCEP EPS data are accessed directly from the remote OPeNDAP server (labelled [6] in Fig. 1), whereas the re-analysis data are accessed via the aggregation server (labelled [5] in Fig. 1) at ESSC. Once all the jobs have finished running the output from TRACK (labelled [7] in Fig. 1) is put onto the server for the user to download or plot. Whilst a set of jobs are still running, the user is able to check the progress of each individual job from their web browser.

3. Application and future development of the TRACK web application

The TRACK web application was used to compute the storm tracks from the NCEP EPS for the Froude et al. (2007b) study. The use of the web application made the processing of this large amount of data considerably easier. By running the TRACK program across multiple computers the processing time was significantly reduced. Accessing the data directly using OPeNDAP dramatically reduced the amount of data that needed to be stored locally. The NCEP EPS data files include a large number of meteorological fields at a large number of different pressure levels. For the analysis presented in Froude et al. (2007b) only the MSLP and $\xi_{850}$ fields were required. By using OPeNDAP and its subsetting functionality, the TRACK software was able to download small parts of the data as and when required. As mentioned previously in the introduction, the analysis methodology of Froude et al. (2007b) required very large samples of data. Without the use of the eScience methodologies of the web application it would not have been possible to store and analyse such large amounts and obtain new and detailed information about the prediction of storms.

The main problem experienced with the web application was caused by the NCEP EPS data server, which is a test server and not operational. Problems with the server meant the rate at which the data could be processed was considerably reduced. However, this dependence on the data provider would be a problem of any such web application. Another issue concerning the web application is that the OPeNDAP protocol used to access the remote data offers very little security. This is a problem for datasets, such as the ECMWF EPS dataset, that require a higher level of security. If the security of OPeNDAP was improved then the web application could potentially be modified to include such datasets.

Although it was not possible to use the web application to compute the storm tracks from the ECMWF EPS data, Condor was used to perform the storm tracking across multiple computers. It was also used for the storm tracking of Bengtsson et al. (2005) and Froude et al. (2007a). The importance of Condor to the analysis presented in these studies cannot be over emphasised. Without Condor the data processing would have been extremely difficult with the facilities available. Condor was particularly well suited to the processing of the EPS data, since the storm tracking for each ensemble member could be performed on a different computer.

In the future it is hoped that the TRACK web application and services like it will be used as a tool for other areas of scientific research. The service has already been used by scientists from the US Navy to study past storms using the NCEP re-analysis data. eScience methodologies, such as those used by the web application, could be useful for operational NWP. For example, distributed computing techniques such as Condor would be ideal for ensemble prediction. If the security issues of OPeNDAP were addressed then the protocol could be very useful for providing data to Scientific researchers. Projects such as the THORPEX Interactive Grand Global Ensemble (TIGGE),\textsuperscript{15} which aims to create a multi-model ensemble database by the international collaboration of Centres/Institutions such as

The TRACK web application could be developed considerably in the future. Other datasets provided via OPeNDAP could be included. These need not necessarily be atmospheric datasets. They could for example be ocean datasets, since the TRACK program can also be used to track ocean eddies (Hodges, 1999b). At the moment it is possible to download storm tracks and plot them in a web browser. Since this is rather limited, the service could be extended to allow statistics to be generated from the computed storm tracks. Statistics such as those of Hodges (1996), Froude et al. (2007a, b) could potentially be generated from a web browser. The service currently runs on computers within ESSC using Condor, which limits the number of users we are able to offer the service to. The service could potentially be extended so that the user was able to run jobs on other Condor pools (such as the Reading University pool, which currently consists of approximately 150 machines) or even the National Grid Service (NGS),16 which is the UK’s largest operational grid with clusters of computers located at Rutherford Appleton Laboratory (RAL) and the Universities of Manchester, Oxford, Leeds, Sheffield and York. The use of NGS would allow a much larger number of users and would enable larger amounts of data to be processed with TRACK.

Acknowledgements

The author would like to thank Kevin Hodges for advice about the use of the TRACK program and NCEP for making the re-analysis and EPS data available via the OPeNDAP protocol. Thanks also go to the reviewers whose thoughtful comments helped to improve this paper considerably.

References
