

Presenting a multi-terabyte dataset via the web

Ag Stephens, Kevin P. Marsh, Bryan N. Lawrence

British Atmospheric Data Centre, Rutherford Appleton Laboratory, Chilton, OX11 0QX, U.K.

The British Atmospheric Data Centre (BADC) holds a 4TB copy of the ERA-40 Re-analysis dataset. Combining open-source data analysis and presentation software packages with our own caching system we have allowed users to create their own outputs from an 18TB virtual dataset. Regular gridded files are created on-the-fly in response to user requests and data subsets distributed via a secure web-based data server.

1. Introduction

In recent years the potential volume of atmospheric science datasets has risen greatly and as a data archivist and distributor, the BADC has a key role in delivering data to users in a useful form. Climate and Numerical Weather Prediction datasets such as those generated by the ECMWF and the UK Meteorological Office present the problem of how to present Terabytes of data to the user in a way that (a) their resources can cope with, (b) the user can understand and (c) avoids the need to download enormous files.

The ERA-40 dataset has proved a test case for how the BADC will present other datasets and we are just about to release our Live Access Server (LAS) providing access to subsetting, plotting and file conversion tools via a single web-interface. This will also provide data comparison in future although we have not implemented this yet. We provide around 18TB of NetCDF data to our users via a virtual dataset and perform dynamic conversion of individual files whilst only storing 4TB of data.

2. Project Design

In building our own ERA-40 archive we chose to get all available parameters in the analyses on model, pressure, potential temperature levels as well as the surface/single level parameters and those on the potential vorticity levels. The BADC has a policy of archiving data in the original format so we opted to archive spectral upper air data in T159 resolution with the surface parameters on the reduced N80 Gaussian Grid. We kept the ECMWF Grib format because it is compact and well known to our users.

The total volume of this archive was 4TB, small enough to store on disk so we could avoid shunting data onto tapes. The downside to using spectral and gaussian gridded data is that most standard software products used by atmospheric scientists cannot read non-rectilinear grids. Many users would want regular gridded products, most commonly at 1.0°, 1.125° and 2.5° resolutions, so we decided to create a system for dynamically converting the standard archive into a cache of regular gridded 'virtual' data.

This regular 1° gridded *virtual* dataset would be available in Grib (8TB) and NetCDF (15-20TB) but would never all exist at any one time. Figure 1 shows the various archives, formats used and processes involved in converting the files.

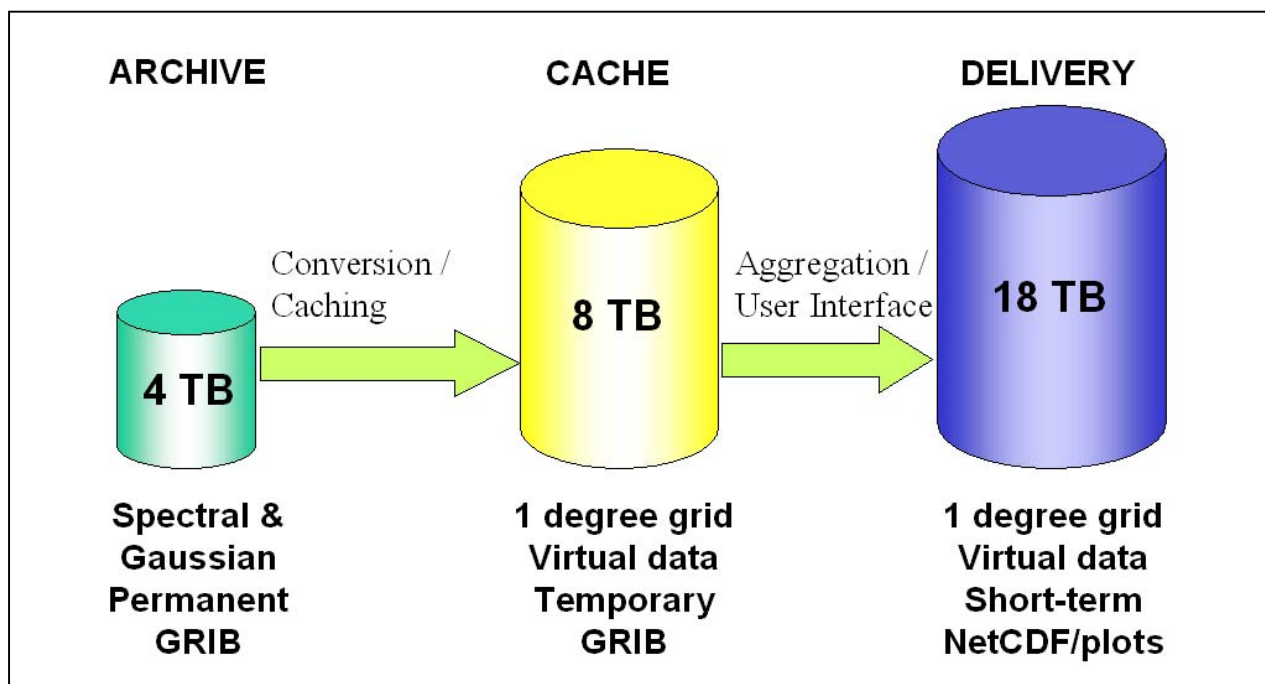


Figure 1. An outline of the design and datasets involved in the project.

3. Conversion Tools

The main conversion tool chosen for use was CDAT (Climate Data Analysis Tools). CDAT is a data processing software package developed by the Program for Climate Model Diagnosis and Intercomparison (PCMDI) at the Lawrence Livermore National Lab in the US. The BADC has invested significant time into CDAT for a number of reasons as listed below:

- It is designed for atmospheric and earth science data so is applicable to our field.
- It is scriptable directly from the Python interface so is easy to automate via scripts or use interactively.
- It allows data analysis, conversion, sub-setting and numerical operation of arrays (using Numerical Python).
- Python can be bound to Fortran and C code using various freely available utilities. Since we wanted to use CDAT to read and manipulate data we chose a Python-based route to control the conversion of irregular to regular gridded data. Conversion code was developed in Fortran using the ECMWF EMOS library and this was 'glued' to Python via the *Pyfort* utility, allowing Fortran subroutines to be called directly from Python.
- CDAT also allows visualisation via its Visualization and Control System.
- It ships with a Graphical User Interface, Visual CDAT (VCDAT).
- It has an XML representation (Climate Data Markup Language) that can be used to aggregate large datasets into a single file handle.
- It is open-source so users can add their own functionality and feed it into future releases.
- It is freely available via the Internet.
- PCMDI has integrated other popular software components, such as the LAS which is relevant to this particular application.

4. Caching

We created a directory structure in the cache that exactly mirrored that in the standard archive. However, the cache directories would only contain the G^rADS/GRIB control files (".ctl") which are very short ASCII descriptions of the metadata only. This would give CDAT something to aim at when the actual data files were not in place yet.

The caching algorithm was written in Python for compatibility with CDAT and was designed to control data volumes, keep track of what files are in the cache and record how long they have been

there. Each request is analysed to calculate the volume requested and check if there is enough room in the cache. We decided on an initial cache volume of 1TB.

This design would lead to files that are frequently requested remaining in the cache without being removed, thereby increasing access time to commonly requested data.

5. File Aggregation

CDAT provides a tool for aggregating spatiotemporal files with a similar domain by describing the metadata in an XML file. These Climate Data Markup Language (CDML) files, created by the CDAT's *cdscan* command-line utility, allow an entire dataset to be represented by a single XML file. The CDML file contains the following main elements:

<dataset> - including a template for pointing to the appropriate directories and files, and a file format description of the aggregated files.

<axis> - information on the dimensions of the axes, units, name, start, step, number of steps, etc.

<variable> - information on each variable including name, units, long name, and dimensions along each axis.

This structure is highly suited to uniform datasets that aggregate on regular grids in time and space, such as ERA-40. After slightly modifying the contents of our CDML files we now have the entire ERA-40 dataset represented by 7 files. The surface analysis CDML file links to 2,955,040 files and is only 21 KB in size. It is parsed within 1 second and provides access to any files within that dataset.

6. Live Access Server

When looking for a web front-end to our system the most suitable product at the time appeared to be the LAS, developed by National Oceanic and Atmospheric Administration, Pacific Marine Environmental Laboratory, Seattle, USA. The LAS is a configurable web server designed to provide flexible access to geo-referenced scientific data (Hankin *et al.*, 1998). It requires a data analysis and visualisation system underneath so we are using CDAT (other possibilities include Matlab, IDL and G^rADS).

The following software required configuration for our LAS (version 6.1):

- Apache web server (on Linux RedHat 7.3)
- TomCat java servlet engine (to deliver java-based servlets running behind the LAS).
- Connection to a MySQL database (to store information on the datasets, variables, axes etc.)
- Slight recoding of the CDAT/LAS interface (Python and Perl).
- Ingesting the CDML data into the LAS XML representation. Since LAS has its own XML representation, Python-based tools are provided to convert the CDML representation into the LAS form. Once the LAS XML files have been generated they are ingested into the MySQL database and can then be viewed via the web interface.
- Interfacing with BADC user authentication system to restrict data to bona fide registered ERA-40 users. Users applying for access to ECMWF datasets must sign a declaration on how they intend to use the data. We already have a standard system for authenticating users and providing a listing of data directories via the web for them to download. We therefore had to layer this security code in front of the LAS to restrict access.

7. The Operational System

Figure 2 shows how the various parts of the LAS/CDAT/Caching project fit together.

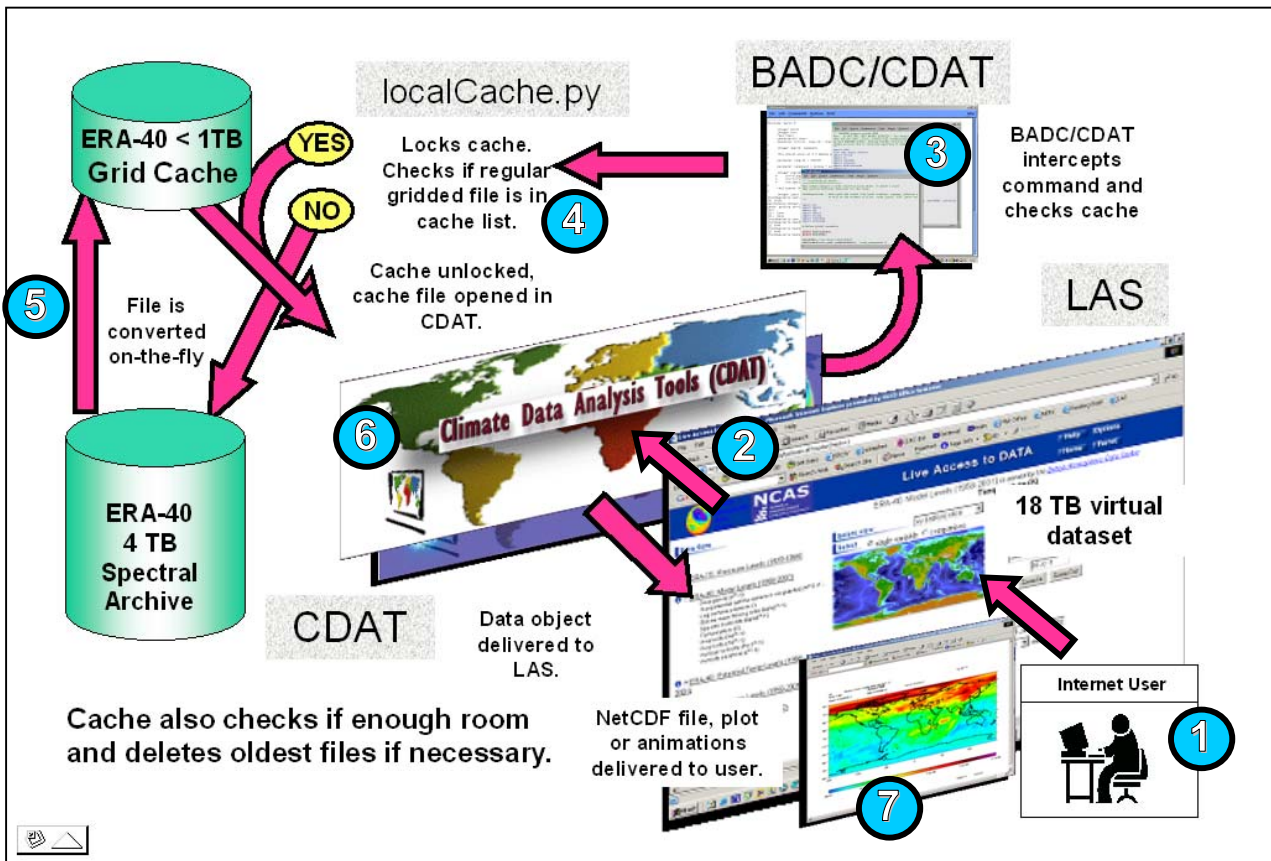
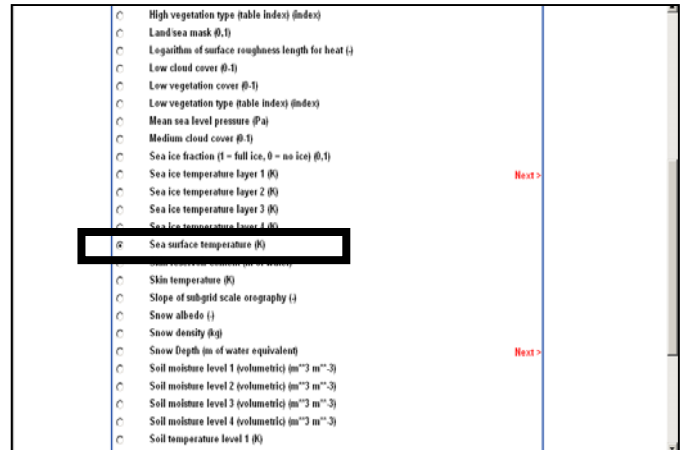


Figure 2. An overview of the LAS/ CDAT/Caching system. (1). A user logs into the BADC LAS with a username and password; (2) She makes a request for a product (such as a GIF plot or NetCDF file) and a request is sent to CDAT; (3) Our own Python code then intercepts the CDAT *open file* command and links to the caching algorithm; (4) The cache is locked whilst the request is made and the Python database checks which files are already in the cache and calculates if there is enough space for the request. If files need deleting the caching algorithm will do this. (5) All required files will then be converted to the regular gridded cache files and the (6) CDAT request then continues to extract the required data and convert to the output format; (7) The LAS then delivers the product to the user via the web.

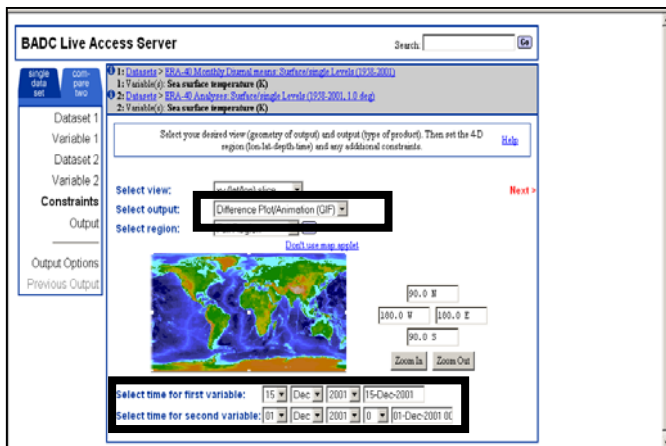
An example of how the system can be used is to plot the difference between two variables. In figure 3, the LAS is used to generate departures from a monthly mean field. For 1st December 2001, the global 6-hourly sea surface temperature field is subtracted from the background monthly mean field to generate an anomaly field on-the-fly.



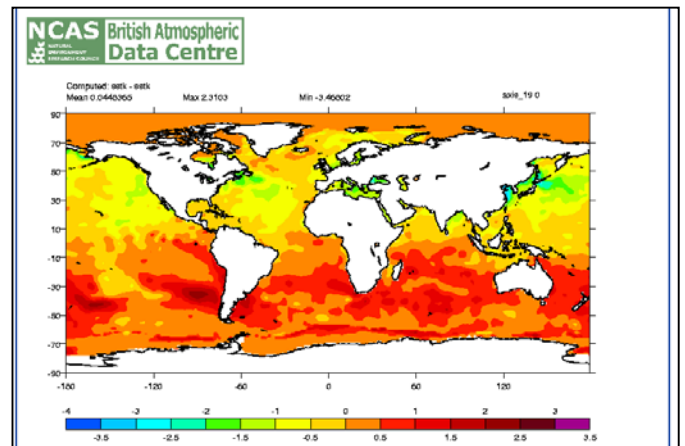
(3a)



(3b)



(3c)



(3d)

Figure 3. An example use of the LAS/CDAT/Caching system to generate a departure plot. (3a) Select two datasets: 6-hourly surface data and monthly mean surface data; (3b) Select a meteorological variable; (3c) Select the output type and times for each dataset; (3d) The LAS returns a departure plot generated by CDAT.

8. Reviewing the System

From our experience so far, not yet including user feedback, the system has the following advantages:

- We can present multi-terabyte virtual datasets to users via one interface, creating data on request.
- Users can get to data without having to interact with files and formats.
- New options are now available to users in sub-setting and extracting data.
- We can control and monitor requests and usage of data automatically.
- The caching system can be applied to other real and virtual datasets with relative ease.
- Our acquired knowledge of CDAT and LAS will be applicable to further datasets.

There are also some disadvantages in our design:

- The system does not yet deal with enormous requests.
- There are limits on the configuration of plots etc that need to be extended for user-defined settings.
- The caching system is slow due to dependence on the native Python database system.
- At present, the LAS only presents one dataset to users.

These issues have highlighted problems to address in the next stage of the project. Key areas that we aim to extend and improve our LAS/CDAT/Caching infrastructure are as follows:

- We plan to develop parallel LAS servers for other datasets (ECMWF Operational and ERA-15 and Met Office Unified Model data).
- We plan to implement a timing algorithm to inform users how long a request will take.
- Comparison between different datasets, letting the user difference, for example, the Unified Model data with the ECMWF Operational data.
- Moving the caching database to MySQL. This should improve performance and scalability.
- We are also considering parallelising the background file conversions on a Beowulf cluster to massively improve performance.

9. Conclusion

Software tools now exist within the open-source community to provide access to terabyte spatiotemporal datasets via a web interface. By combining a Python-based caching system with the CDAT and LAS software packages the BADC has developed secure access to the ECMWF ERA-40 virtual dataset, allowing registered users to extract data subsets, plots, animations and NetCDF files directly from the web.

References and Links

Hankin S., J. Davison, J. Callahan, D. E. Harrison, K. O'Brien. (1998). A configurable web server for gridded data: a framework for collaboration. In *14th International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*, AMS, 417–418.

British Atmospheric Data Centre: <http://badc.nerc.ac.uk>

Climate Data Analysis Tools: <http://esg.llnl.gov/cdat>

Climate Data Markup Language: http://esg.llnl.gov/cdat/cdms_html/cdms-6.htm

Live Access Server: http://ferret.pmel.noaa.gov/Ferret/LAS/ferret_LAS.html

Pyfort: <http://pyfortran.sourceforge.net>