4D DATACUBE IN SUPPORT OF OPERATIONS PLANNING

Ryan Hofschneider* and Oleg Kiselyov, Technology Advancement Group,
Fleet Numerical and Oceanography Center, 7 Grace Hopper Avenue, Monterey, CA, 93943-5501

Abstract

This paper presents a 1D-4D grid data distribution system and its use in support of tactical operations. The system, which is a part of Metcast, has been tested during Fleet Battle Experiment-Juliet and is being used operationally. We store NOGAPS, COAMPS, WW3, SWAN and other model grids. The built-in fine-grained access control allows the system to be used for coalition support and joint operations.

The system incorporates the Grid DataBlade (Barrodale Computing Services) — an extension to an IBM Informix RDBMS. The DataBlade stores tiles of scalar and vector grids arranged in time and the vertical dimension. The DataBlade can compute a subgrid, select a vertical post, re-project and interpolate in any dimension. Because these computations are performed within the database engine, they are highly efficient. A flexible query system lets the user select a 1D-4D (sub)grid based on a model, geographical region, valid time and other criteria.

1 METCAST

1.1 Overview

Metcast is a request-reply and a subscription system developed by the Fleet Numerical Meteorology and Oceanography Center (FNMOC)[1] for distributing, disseminating, publishing and broadcasting of real-time weather information [4]. It is a data-oriented web service, acting as an intermediate between clients communicating over HTTP and several meteorological databases. These databases contain weather observation reports, forecasts, advisories, gridded data produced by weather models, as well as satellite imagery and plain text messages and discussions. A sub-system of Metcast includes a comprehensive set of message decoders which convert complex (and often ill-formed) World Meteorological Organization (WMO) bulletins into well-tagged XML documents [6].

Metcast has built-in fine-grained access control that allows the system to be used for coalition support and joint operations. The system has been used operationally at FNMOC since 1999 and clients are deployed at a great many sites throughout the U.S. Navy, as well as the U.S. Air Force, NATO, NOAA, and other government agencies.

[1] http://www.fnmoc.navy.mil

1.2 The Metcast Request Language

To retrieve a subset of data from a Metcast database, a Metcast client uses a domain-specific, flexible request language called MBL [2]. MBL is designed to express requests for meteorological information for a specific area of interest, which may be the entire globe. Besides geographical localization, meteorological products are also characterized by timestamps, layers (height, isobaric pressure, depth), and type (grids, satellite imagery, advisories, etc.). Each separate type is described by its own set of parameters and therefore, requires its own set of request specifications. Given below are two sample requests:

```
(request1 (products
  (grid "Air Temperature"
    (bounding-box 90 -180 -90 180)
    (model NOGAPS "global_720x361")
    (valid-time 1021917600)
    (layer isbr_lvl 1000))))

(request2 (bounding-box 63 -129 14 -63)
  (products (AC-advisory ((SIGMET))
    (mime-type text/plain))))
```

A client communicates the request(s) via an HTTP POST to a Metcast server. Data fulfilling the request is MIME-encoded and returned in the message body of the HTTP response. If an error occurs (e.g. MBL request is incorrectly structured), or no data in the Metcast databases matches the request(s), an appropriate HTTP status code is indicated.

Despite the technical advantages of using an S-expression[2] based request language [5], the rise of XML as the de facto standard for data interchange on the web in the years since Metcast's design in 1998, has occasionally lead to criticisms of Metcast. FNMOC is currently participating in the efforts lead by the Department of Defense Meteorology and Oceanography Data Administration[3] to define the Joint METOC Broker Language (JMBL). The goal of JMBL is to provide a unified XML Schema and semantic to promote interoperability between MetOc data clients and servers.

1.3 Metcast Table of Contents

As described above, a Metcast server can be asked to deliver a wide variety of MetOc products. To formulate a

[2] http://www.wikipedia.org/wiki/S-expression
[3] https://www.cnmoc.navy.mil/da/jmcdm.htm

request however, a client needs to know what is available, and how to ask for it. In particular, the client has to know the names and categories of products as well as their request attributes – levels, taus, bounding boxes, etc. A "menu" of data types, let alone the list of the currently available products, is specific to each particular Metcast server, and is subject to change.

Every Metcast server therefore offers a reflexive facility: an ability to describe what it can serve, both in principle and in reality. A server can also tell how to build a request language phrase to ask for a particular piece of information.

In response to a *Describe* MBL query, a server sends an XML document – a Metcast Table of Contents (MTOC or TOC). Depending on the parameters of the Describe query, Metcast will return a *potential* TOC or an *actual* TOC. A potential TOC lists the products that the server is registered to receive and store in the database. The potential TOC shows what could potentially be in the database – although not everything will actually be (e.g. a product for a particular time or area may not have been ingested into the Metcast databases yet). The potential TOC is *not* dynamic and is relatively static. It only changes when new data is registered to be stored within the Metcast databases, or existing products no longer need to be distributed. The potential TOC is most useful for scheduling of product deliveries from a Metcast server.

The actual TOC is dynamically generated and represents the actual contents of the Metcast databases at the time of the request. It is most useful for ad hoc and mission-planning requests.

An MTOC only lists the products a user has authorization to download; no meta-data is leaked (e.g. geographic location of a high-resolution forecast grid).

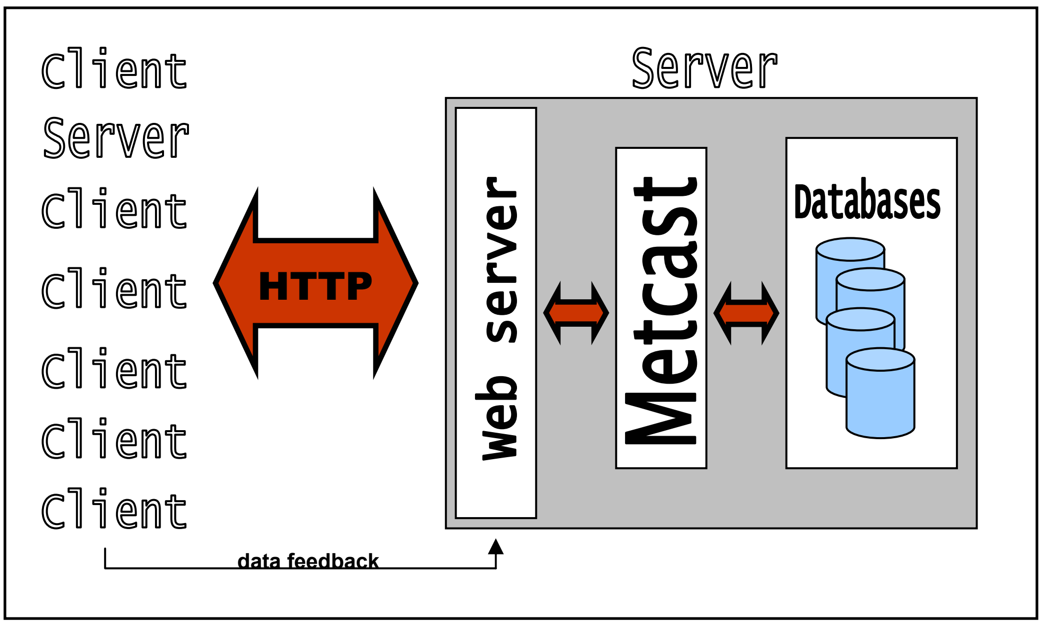

Figure 1: Simplified diagram of the Metcast server components

1.4 Metcast architecture

The Metcast application server, implemented using the Scheme programming language, has a highly modular structure. The main program is responsible for receiving and parsing of a request and packing replies; execution of a particular product request is delegated to separate modules.

The application server does not directly communicate with clients, but leverages a separate HTTP server to provide authentication and other infrastructure. A FastCGI[4] module controls the communication between the http server and Metcast; UNIX sockets are used to communicate with the database server(s) (Fig. 1).

Metcast is typically hosted on RedHat Linux or Sun Microsystems Solaris. The Apache HTTP server, version 1.3.x, and the IBM Informix Dynamic Server database system[5] (IDS), version 9.x, currently implement the other components of a Metcast server.

An IDS instance can host several different databases for Metcast: one storing satellite imagery; another for observations, warnings, and advisories; a SingleTEDS[6] database for storing forecast grids for legacy applications; and a database leveraging the Grid DataBlade[7], also for storing grids, but in a manner

[4] http://www.fastcgi.com

[5] http://www-3.ibm.com/software/data/informix/

[6] https://teds.navy.mil

[7] http://www.barrodale.com/grid_Demo/index.html

supporting sophisticated extractions and efficient sub-gridding.

2 APPLICATIONS USING METCAST

Over the four years Metcast server has been deployed operationally at FNMOC, a number of client applications have been developed. The following sections highlight a select few of these tools.

2.1 Joint Metoc Viewer

The *Joint Metoc Viewer* (JMV)[8] is the oldest and most widely distributed Metcast client. JMV is a Microsoft Windows application for displaying, annotating, and looping meteorological and oceanographic data and satellite imagery. It is deployed throughout the U.S. Navy and other DOD agencies; and is one of the main tools used by the MetOc officer within the OA division aboard a U.S. Navy ship.

2.2 XiS ViewPoint

XiS ViewPoint[9] is a Java-based, information analysis and visualization system written by Polexis that integrates real-time data and presents it in graphical overlays. Metcast forecast grid and satellite imagery support was integrated into XiS during the past year and was tested during Fleet Battle Experiment-Juliet. Naval Pacific Meteorology and Oceanography Center, San Diego (NPMOC-SD) uses XiS to produce fused products for SOCAL and the Fallon Strike Range. Naval Special Warfare has used XiS for mission planning and INTEL requests.

2.3 MyWxMap

MyWxMap[10] is a web application developed by FNMOC that gives users the ability to define an area of interest on the globe and receive well-crafted, near real-time weather charts, using forecast grids extracted from a Metcast server. The web application utilizes a cluster of 32 personal-computer-class machines and has a production capacity of 150,000 charts per hour. MyWxMap is used by professional weather forecasters within the U.S. Navy as well as TV meteorologists, scientists and the general public.

2.4 Chemical Downwind Forecast

FNMOC's *Chemical Downwind Forecast* (CDF) web application is designed to allow military planners to predict relevant weather conditions for specified geo-located points in the event of a nuclear, chemical, or biological attack. The CDF fetches forecast grid data from Metcast and generates any one of a number of different reports that can be used to calculate downwind effects. It was used by joint forces during Operation Iraqi Freedom as well as for operations in Afghanistan.

2.5 Flight Simulator with Future Weather

Flight Simulator with Future Weather (FSFW) is a Microsoft Flight Simulator 2002 (FS2002) add-on that inserts predicted weather values into the FS2002 environment [8]. FSFW retrieves gridded data for several different weather parameters (air temperature, relative humidity, total cloud cover, etc.), performs processing to map these parameters to FS2002 equivalents, and updates the FS2002 settings as the simulated flight progresses. The Metcast server is queried periodically, retrieving new predicted weather values for the aircraft's new position.

3 THE GRID DATABLADE

The IBM Informix Dynamic Server (IDS) that Metcast uses for its database backend has a convenient facility for extending its base functionality through the use of dynamically loadable shared libraries called *DataBlades*. A DataBlade can define new data types and routines which can be used in SQL just as like any built-in data type or procedure. The Grid DataBlade, developed by Barrodale Computing Services (BCS)[11], defines data types and routines that allow the storage of 1-4D gridded volumes within IDS in a manner allowing sophisticated extractions and efficient sub-gridding. A Grid DataBlade-equipped database is one of the back-ends used by Metcast server to store forecast grids. The power of the Grid DataBlade is exposed to Metcast clients through the Metcast request language, MBL.

The heart of the Grid DataBlade is the *GRDValue* data type, which leverages the smart large object technology built into Informix to store gridded data. Internally, the GRDValue tiles the data, similar in spirit to how a file is stored in multiple blocks on a file system. The tiling scheme employed by the Grid DataBlade allows for very efficient extractions, since only the tiles necessary to fulfill a data request are extracted from a GRDValue table column. In comparison to the traditional method used by Metcast to store forecast grids, which requires entire 2D slices along the x and y grid bounds to be extracted (no matter how small the area of interest of a client request), the tiling scheme of the Grid DataBlade can result in extraction times orders of magnitude faster.

Further, by storing gridded data for a particular weather parameter as a 4D volume rather than a set of disassociated 2D slices, the Grid DataBlade is able to interpolate between grid points on the fly. For example if an ocean forecast model generates data for depths of

[8] http://www.metnet.navy.mil/metcast/METCAST-JMV.htm
[9] http://www.polexis.com/viewpoint.html
[10] http://aipswebu1.fnmoc.navy.mil/MyWxmap/
[11] http://www.barrodale.com

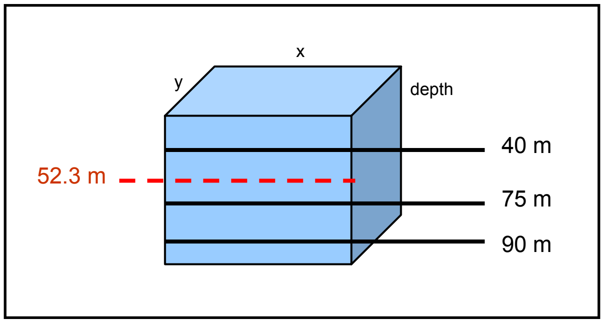

Figure 2: Interpolating through a GRDValue, on the fly

40m, 75m, and 90m, but the client requires a best guess for a depth of 52.3, the Metcast server, through the functionality of the Grid DataBlade, can provide the answer. Because the interpolation is done inside the database, where the data resides, the calculations are done in an efficient manner. Basic interpolation methods are built-in to the Grid DataBlade; custom interpolations schemes can be linked in using a provided API.

Gridded data can be extracted from the 4D volume organized by the Grid DataBlade from any angle, leading to many potential slant-range visibility applications not yet fully explored by current Metcast clients. Of course, users are not limited to extract 2D planes as Figure 2 might imply, but are free to request individual points, "soundings", 3D cubes or 4D subvolumes.

NetCDF[12] is the native import/export format used by the Grid DataBlade; the Grid DataBlade provides a convenient mechanism for mapping global netCDF attributes to database table columns. A NetCDF convention named GIEF-F was developed by FNMOC to capture the meta-data typically desired by forecast grid users [1]; it roughly corresponds to the meta-data found in a WMO GRIB file, except descriptive strings are favored instead of integer ids that require look-up tables under the GRIB paradigm.

Lastly, the Grid DataBlade leverages the IBM Informix Spatial DataBlade[13] (freely available for use with IDS) to permit the storage and extraction of grids using over a hundred different map projections. This ability given to Metcast server, through the Grid DataBlade, to spatially transform grids on the fly is a key feature – forecast grids are often overlaid in GIS applications with other data sources which may have particular map projection requirements.

4 FUTURE DIRECTIONS

There are several Metcast server enhancements the Technology Advancement Group (TAG) at FNMOC are currently investigating:

Radial extraction support

This would extend the Metcast and Grid DataBlade to allow a client to specify a horizontal vector in order to return a vertical planar slice of data from a 3D grid, a *radial*.

PostgreSQL

While IDS has served FNMOC well as the database backend for Metcast, the advantages of a freely available, fully-featured, open-source relational database system like PostgreSQL are obvious. Initial trials using PostgreSQL instead of IDS for portions of the Metcast backend will be performed during the coming year.

JMBL

As mentioned previously, FNMOC is participating in a DOD effort to define the Joint Metoc Broker Language. The standard is still in development, but hopefully should mature enough in the next year to allow for at least an initial, perhaps partial, implementation in Metcast.

REFERENCES

[1] Fleet Numerical Meteorology and Oceanography Center, "FNMOC GIEF-F Conventions", http://www.metnet.navy.mil/~hofschnr/GIEF, Jun. 2002.
[2] Kiselyov, O., "A delegation language to request weather products and a scheme of its interpretation", Proc. Third ACM SIGPLAN Int. Conf. on Functional Programming (ICFP'98), Baltimore, Maryland, Sep. 27-29, 1998, p. 343.
[3] Kiselyov, O., "Distributing Weather Products through an HTTP pipe", http://www.metnet.navy.mil/Metcast, Mar. 10, 2000.
[4] Kiselyov, O., "Implementing Metcast in Scheme", Principles, Logics, and Implementations of high-level programming languages (PLI) Conf., Montreal, Canada, Sep. 17-22, 2000.
[5] Kiselyov, O., "SXML Specification Revision 2.1", http://pobox.com/~oleg/ftp/Scheme/SXML.html, Mar. 1, 2002.
[6] Latendresse, M., "A persistent web application for distribution of weather information in Scheme", International Lisp Conf., San Francisco, California, Oct. 29, 2002.
[7] Prescod, P., "XML is not S-Expressions", http://www.prescod.net/xml/s-exprs.html, Feb. 24, 2003.
[8] Williams, A. B., and R. Hofschneider, "FNMOC Weather forecast into Microsoft Flight Simulator 2002", Battlespace Atmospheric and Cloud Impacts on Military Operations (BACIMO) Conf., Monterey, California, Sep. 9-11, 2003.

[12] http://www.unidata.ucar.edu/packages/netcdf/
[13] http://www.ibm.com/software/data/informix/blades/spatial/