NMRF/TR/03/2016

सत्यमेव जयते

TECHNICAL REPORT

Implementation and Up-gradation of NCUM in Bhaskara HPC

R. Rakhi, A. Jayakumar , M. N. R. Sreevathsa
and E. N. Rajagopal

May 2016

National Centre for Medium Range Weather Forecasting
Ministry of Earth Sciences, Government of India
A-50, Sector-62, NOIDA-201309, INDIA

Implementation and Up-gradation of NCUM in Bhaskara HPC

R. Rakhi, A. Jayakumar, M. N. R. Sreevathsa and E. N. Rajagopal

National Centre for Medium Range Weather Forecasting
Ministry of Earth Sciences
A-50, Sector 62, NOIDA-201309, INDIA

May 2016

Ministry of Earth Sciences
National Centre for Medium Range Weather Forecasting
Document Control Data Sheet

1	Name of the Institute	National Centre for Medium Range Weather Forecasting
2	Document Number	NMRF/TR/03/2016
3	Date of publication	May 2016
4	Title of the document	Implementation and Up-gradation of NCUM in Bhaskara HPC
5	Type of Document	Technical Report
6	No. of pages & Figures	22 Pages and 9 Figures
7	Number of References	7
8	Author (s)	R. Rakhi, A. Jayakumar, M. N. R. Sreevathsa and E. N. Rajagopal
9	Originating Unit	NCMRWF
10	Abstract	The Unified Model based global forecast system at NCMRWF (NCUM) was upgraded to version 8.5 in the new IBM *iDataPlex* HPC system named "*Bhaskara*". The major model upgrade includes an increased resolution to 17-km (at mid-latitudes), ENDGame dynamics and physics changes to Global Atmosphere (GA) 6.1. The scaling of the model was done at a number of processor decompositions, and Open Multi-Processing (OpenMP) application program interface settings to get the least wall clock time for 10-day forecast. The details of dynamics and physics up-gradation along with scaling of the model are described in this report. Subsequently a later version of NCUM (version 10.1) based on a new model operating framework called Rose/Cylc was also implemented in *Bhaskara HPC.* From version 10.0, the UM and its associated systems are hosted on the Met Office Science Repository Service (MOSRS). The implementation strategy adopted for this up-gradation is also detailed in this report.
11	Security classification	Non-Secure
12	Distribution	Unrestricted Distribution
13	Key Words	

Table of Contents

Abstract

The Unified Model based global forecast system at NCMRWF (NCUM) was upgraded to version 8.5 in the new IBM *iDataPlex* HPC system named "*Bhaskara*". The major model upgrade includes an increased resolution to 17-km (at mid-latitudes), ENDGame dynamics and physics changes to Global Atmosphere (GA) 6.1. The scaling of the model was done at a number of processor decompositions, and Open Multi-Processing (OpenMP) application program interface settings to get the least wall clock time for 10-day forecast. The details of dynamics and physics up-gradation along with scaling of the model are described in this report. Subsequently a later version of NCUM (version 10.1) based on a new model operating framework called Rose/Cylc was also implemented in *Bhaskara HPC.* From version 10.0, the UM and its associated systems are hosted on the Met Office Science Repository Service (MOSRS). The implementation strategy adopted for this up-gradation is also detailed in this report.

1. Introduction

The Unified Model (UM) system at NCMRWF (NCUM) is the numerical modelling system developed at the United Kingdom's Met Office (UKMO). It employs *seamless modelling* approach, wherein different configurations of the same model can be used across all time and space scales. The Met Office UM is continually evolving, taking advantage of improved understanding of atmospheric processes and steadily increasing supercomputer power. Hence, NCUM is also upgraded frequently to incorporate these developments. In June 2015, UM was upgraded to version 8.5 (with ENDGame dynamical core) with a spatial resolution of ~17 km at mid-latitudes. ENDGame stands for Even Newer Dynamics for General atmospheric modelling of the environment and improves upon the New-Dynamics (ND) dynamical core. This latest version uses non-hydrostatic dynamics with semi-Lagrangian advection and semi-implicit time stepping. It is again a grid point model with ability to run with a rotated pole and variable horizontal grid. A number of sub-grid scale processes are also represented, including convection, boundary layer turbulence, radiation, cloud, microphysics and orographic drag. This version was implemented at NCMRWF on its new 350TF IBM *iDataPlex* HPC system named "*Bhaskara*".

UM version 8.5 uses "Unified Model User Interface" (UMUI), a Tcl/Tk based Graphical User Interface (GUI) to set up and run the tasks of model compilation, model run and reconfiguration jobs. However, this software infrastructure for running scientific suites and applications has many issues including complexity, duplication, no version control etc. To resolve these issues, Met Office introduced a new framework, called *Rose* as a replacement for the UMUI from version 9.0 onwards. *Rose* uses an open-source meta-scheduler called *cylc* as a replacement for the older Suite Control System User Interface (SCSUI) for scheduling various tasks in an operational suite (A brief introduction to Rose/cylc is given Appendix-1). In accordance with these changes, UM version 10.1 under Rose/cylc framework was implemented on Bhaskara HPC, the details of which are presented in the following sections of this report.

2. Up-gradation of NCUM in Bhaskara HPC

As aforementioned, NCUM version 8.5 was upgraded from NCUM version 7.9 (New Dynamics dynamical core; Davies et al. [2005]). Compared to earlier versions there are major changes in the dynamics and physics options, which are detailed below.

2.1 Model Dynamics

The major change in the implementation of this version of NCUM system was the up-gradation of its dynamical core from the New Dynamics to “ENDGame”. Both of them are based on the Non-Hydrostatic Deep formulation (NHD). The cores both use a latitude-longitude grid with a terrain following height-based vertical coordinate. The cores also have the same underlying horizontal (i.e. an Arakawa-C grid), and vertical (Charney-Phillips grid) grid structure, and both are semi-implicit and semi-Lagrangian. Although the equation set and grid staggering are the same in ENDGame and ND, the development of the ENDGame dynamical core includes a large number of changes. The most important of these improvements are: (i) better handling of flow across the poles of the latitude-longitude coordinate system;(ii) an iterated semi-implicit scheme, providing reduced temporal truncation error; (iii) better scaling on multiple processor computer architecture; and (iv) an overall improvement of model stability and robustness. The code now includes a set of ‘switchable’ physical assumptions (for instance it can run a non-hydrostatic and a hydrostatic formulation; deep-atmosphere and shallow-atmosphere formulations, and use of spheroidal, spherical or Cartesian co-ordinates). Additionally, a novel mass conserving transport scheme, Semi-Lagrangian Inherently Conserving and Efficient (SLICE) has been developed (Wood et al. [2013]).

The most significant differences between ENDGame and New Dynamics are:

a) ENDGame uses a nested iterative time step structure (more implicit, approaching Crank-Nicolson), which improves its numerical stability.

b) Iterative time stepping allows a simpler approach to solving the Helmholtz equation, which reduces one of the bottlenecks in model scalability.

c) Increased stability allows the time-weights in the semi-implicit time stepping to be much closer to the time-centred value of 0.5 (alpha time-weights, all equal to 0.55), which improves accuracy and reduces the damping in the model.

d) Same Semi-Lagrangian (SL) advection for all variables (cf. Eulerian continuity equation + SL in New Dynamics) and removal of non-interpolating in the vertical for theta advection.

e) Increased stability also allows us to remove almost all explicit numerical diffusion. The only damping applied is a sponge-layer that damps the vertical velocity near the top of the model and also extends to the surface very close to the poles i.e., no polar filtering or horizontal diffusion, control near lid and poles achieved by implicit damping of vertical velocity (ω) giving improved scalability and accuracy.

f) The horizontal grid is shifted half a grid length in both directions so that scalars are no longer held at the poles (*v* is stored at the poles, rather than *u*, *w* and all scalars, as in the New Dynamics) which means not solving Helmholtz problem at singular point of grid. *v* at poles is only required for the evaluation of departure points and interpolation of *v* near to the poles.

g) There are subtle changes to many of the prognostic variables, e.g. virtual dry potential temperature is used as the thermodynamic prognostic, whilst all moist prognostics move from specific quantities to mixing ratios. In addition, potential temperature advection now utilises a fully three-dimensional semi-Lagrangian scheme.

h) Compared to New Dynamics, ENDGame has been designed to allow the code to be switchable between various options: a non-hydrostatic and a hydrostatic formulation, deep-atmosphere and shallow-atmosphere formulations, and use of spheroidal, spherical or Cartesian co-ordinates (as appropriate).

2.2 Model Physics

The other significant development in the new version of the UM in addition to the ENDGame dynamical core is the changes in the model physics. The model physics are based on Global Atmosphere (GA) version 6.1 and Global Land (GL) version 6.1 as described below.

Radiation scheme:

- Solar constant reduced to the latest estimated value of 1361W/m^2.
- Improved CO_2 and O_3 long wave absorption using method of Zhong and Haigh [2003], which improves heating/cooling in the stratosphere.
- Radiation time-step reduced to 1 hour from 3hour, which improves the accuracy of the radiation scheme.
- Removal of the "Delta" aerosol climatology.
- An update to sea ice albedo.

Boundary layer scheme:

- Shorter "Mes" tails in stability functions and other changes to mixing in stable boundary layers over land and "Sharpest" over sea (Lock, [2001]).
- Revised stability functions for unstable boundary layers, to use the "conventional" functions from the Met Office large-eddy model.

- Revised diagnosis of shear-dominated boundary layers to improve cloud fields in cold air outbreaks.

A prognostic cloud fraction and condensation (PC2) scheme:

- Improved cloud erosion method and numerical definition for mixed-phase cloud.
- New cirrus term and use of the model winds in the shear term in the treatment of falling ice cloud fraction.
- Smoother phase change for cloud condensate detrained from convective plumes.
- Apply cloud optical depth filter and diagnostic convective cores to improve the consistency of the standard cloud diagnostics.

Large-scale precipitation scheme:

- Implement improved drizzle size distribution to better match observations and further reduce spurious light rain.
- Loop microphysics sub-stepping over the column rather than in each level.
- Use aerosol climatologies for CCN in second indirect effect, again to replace the use of an inappropriate land/sea split.

Convection scheme:

- Introduction of the 5A convection scheme, with the removal of an ill-formulated convective energy correction.
- An increased entrainment rate in deep convection (and related modified detrainment rates). This follows the work of Klingaman and Woolnough [2014], which shows that increasing entrainment parameter significantly improves modes of tropical variability such as the Indian Monsoon, tropical cyclones and Madden–Julian Oscillation (MJO).
- Smoothed adaptive detrainment of qcl, qcf and tracers.

Gravity wave drag:

- Introduction of the 5A orographic gravity wave drag scheme, which includes a ”cut-off mountain” approach to diagnosing mountain-wave drag and the distribution of the applied drag from the breaking of waves over a layer representing their vertical wavelength.

2.3 Scaling of UM8.5 (N768L70) with ENDGame Dynamical core

The current operational NCUM global model operates at an effective resolution of N768 (~17 km at mid-latitudes), which translates into a 1536 x 1152 horizontal grid and runs with a GA6.1 science package. The scaling of the model was tested at a number of processor decompositions, and Open Multi-Processing (OMP) application program interface settings to determine the best combination for each processor count.

The run had full Storage Handling and Diagnostic System (STASH) diagnostics and dumping, which was set to 5-days (since the writing of dump (*_da files) every day consumes more wall clock time). The model was run for 10-day forecast with and without I/O servers and simultaneous multithreading (SMT) turned off. A description on the concept of using I/O servers in UM is given in Selwood (2014). A set of MPI parameters are additionally specified by environment variables (given in Appendix-2) for further optimising the model. The summary of various attempts are summarized in Table 1. Table 1 shows that there is a strong scaling for a processor combination of 64x48 with 4 I/O servers, resulting in a total of 196 nodes. These nodes are dedicated for the Operational NCUM forecasts to run well within the time constraints of the operational requirements.

Table 1: Computational performance statistics of UM8.5 (N768L70) with various processor combinations and job submission methods

Processor Combination	No. of OMP threads	No of I/O Servers	Job Submission (LSF / HYDRA)	Total Tasks (No. of nodes)	Wall clock Time (in sec)
64x48	2	4	LSF	3136 (196)	4156.90
64x48	2	0	LSF	3072 (192)	8560.21
66x48	2	0	LSF	3168 (198)	8059.89
32x48	4	0	HYDRA	1536 (384)	6706.85
16x32	2	0	HYDRA	512 (64)	21692.58

3. Implementation of UM version 10.1 in Bhaskara HPC

Current operational NCUM is UM8.5 at N768L70 resolution with ENDGame dynamical core, which utilises UMUI support. However, from UM 9.0 onwards the only supported method of running the UM is through Rose/cylc. Under Rose/cylc framework, UM "Jobs" are called as "suites" and are version controlled. The UM makes use of Flexible Configuration Management (FCM) system, which itself is based on Subversion and further makes extensive use of the UNIX "make" utility for compilation of Fortran and C source codes. The UM is controlled by top level UNIX scripts, which take input from *namelists* file and environment variables. These inputs are configured using the UM's Rose GUI and the UM configurations are run via Rose using cylc for scheduling. To run the UM on any system, compatible versions of FCM, Subversion, Rose and cylc are required.

To meet the pre-requisites to run UM10.1 at NCMRWF, the FCM, Rose and Cylc were upgraded to FCM 2015.05.0, rose-2015.06.0 and cylc-6.4.1 respectively. As UM 10.1 supports the use of General Communication (GCOM) Module version 5.1 (GCOM5.1), the same was also installed in Bhaskara HPC. For installing the UM on a platform for the first time, a new set of *platform configuration* files have to be created for use with the fcm-make system. The development of machine configuration files is likely to be an iterative process, particularly if compiler options are being explored for optimisation.

As mentioned earlier, compiling the UM via Rose makes use of *fcm-make* build system, which was introduced in FCM-2. The NCMRWF platform configuration files are set up in the directory **fcm-make/ncm-ibm-ifort.** These site configurations contain all the site-specific information for a build task: a top-level file that sets the build type and the optimisation level, and below this more specific compiler and pre-processing declarations for each build, along with compiler specific files for each type of build (parallel, serial, 32-bit serial). These are used in conjunction with the files in *fcm-make/inc*, which contains generic build instructions common to all sites. A detailed description of the installation procedure of UM10.1 is given in Appendix-3.

As of version 10.0, the UM and its associated systems (e.g. JULES, GCOM etc.) and documentation are now hosted on the Met Office Science Repository Service (MOSRS). MOSRS of UM and *roses-u* are referred with the following URL: https://code.metoffice.gov.uk/svn/um/main and https://code.metoffice.gov.uk/svn/roses-u respectively, wherein the MOSRS-UM contains the UM trunk, while the latter stores various Rose suites. At NCMRWF, the corresponding mirror repositories are located at: svn://ncmlogin3/SREPOS/um and svn://ncmlogin3/SREPOS/roses-u respectively.

3.1 Creating branches in MOSRS

There are several ways to create a branch in a Subversion repository and the general method used is based on user preference (For more details, see Mancell [2015]). Three common methods would be:

a) Using FCM via command line

This is the method adopted by NCMRWF for creating branch in MOSRS. For example,

fcm branch-create --type DEV::USER branch_name fcm:um.x/trunk@vn10.1

where,

- --type is optional as DEV::USER is the default if not present.
- branch_name is the name of the branch and should to be a useful description of the branch's purpose.
- Finally the target URL. This could be any valid Subversion URL, using keywords where convenient. The branch will be created at the specified revision number/keyword (e.g. "@vn10.1", "@2765") or at the *head* of the given URL if this is no supplied. The revision number/keyword will then be used as a prefix for the branch name.

b) Using the "create branch script" located in the UM's admin directory

In the utilities directory of the Admin project there is a script called create branch. This script runs in the Korn shell (*ksh*); if the DISPLAY environment variable is defined, it will use "kdialog" to produce pop-up window, otherwise it will run as a text-based process in the current terminal window. The script asks the user a series of simple questions to construct a command to create a branch; only the most commonly used options are accessible this way. To run the script, extract a copy of it from your repository, e.g.

fcm export fcm:um.x/trunk/admin/branch management/create branch

The above command will extract a copy of the create branch script from the admin directory. If many users wish to use this script then it is advisable to copy it to a centrally available location for executables in the $PATH.

c) Using the fcm gui package.

This is a self describing graphical option and hence is not discussed further.

NOTE: Only those users who have an account in MOSRS are permitted to create a branch in repository. Other users have to check out the working copy of the UM suite from the local mirror repositories.

3.2 Unified Model within Rose/Cylc:

A release of the UM includes a limited number of suites containing standard science settings. These cover the major configurations (global, limited-area, and climate) in which the UM is used. A typical UM suite contains the following files and sub-directories:

a) ***rose-suite.conf*** file: This is used to define suite configurations outside of cylc. E.g. inputs to the cylc suite configuration file, files to install at run time, etc.

b) ***rose-suite.info*** file: This is the suite discovery information file which you saw when you created the suite.

c) ***suite.rc*** file: This is used to define configurations in a cylc suite.

d) ***app/*** directory: This directory contains the configurations of the applications run by the tasks of the suite.

e) ***src/***: This directory contains the source file of a simple program that we shall run in this suite. Note that this directory *will not* normally be present in a real scientific suite, as source files are typically held in separate locations (for e.g. Met Office Science Repository Service).

Building and running the UM under Rose requires (at least) two separate apps: an *fcm_make* app to build the model executable and a *UM* app to configure the runtime *namelists and environment variables*. Coupled models may require additional `fcm_make` apps, one for each executable to be built.

3.2.1 *fcm_make* app

a) Adding branches

This is a special Rose application to build the executable from the UM source file. The options for adding additional branches and adjusting the versions of the UM and JULES can be found in the "Sources" section of the `fcm_make` app. Branches are added as a space separated list as shown in Fig.1.

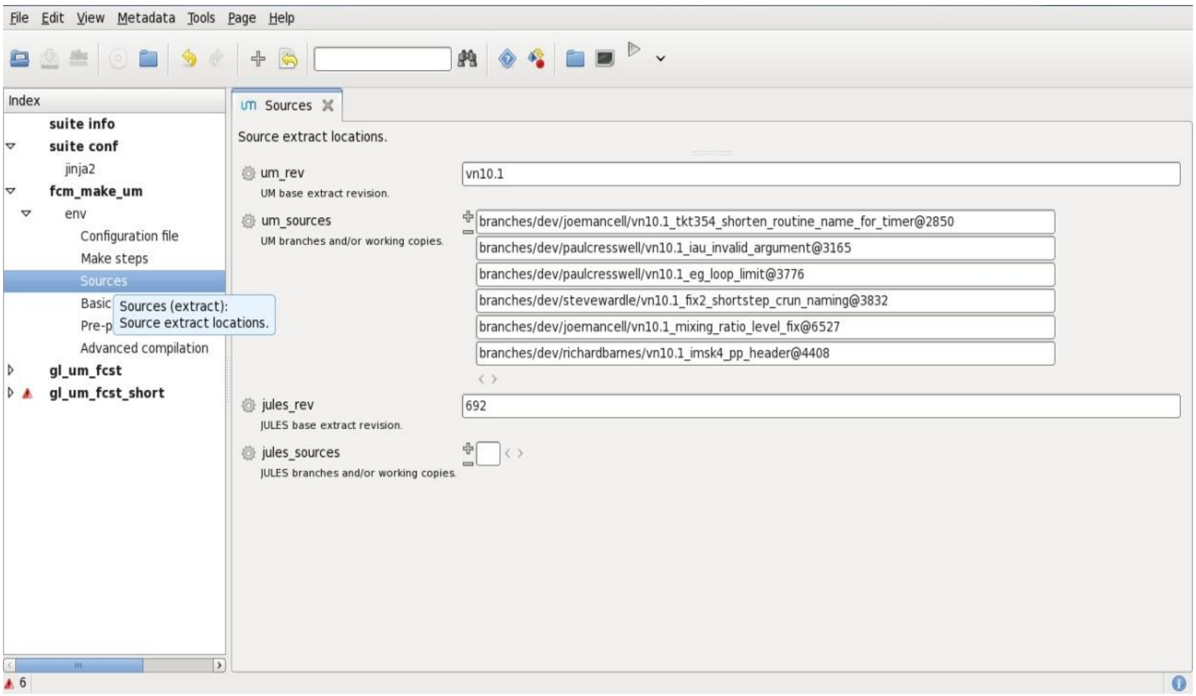

Fig.1: Rose GUI showing the addition of various sources as branches.

b) Configuring builds

The fcm_make application expects a file file/fcm-make.cfg in its application configuration. It runs fcm make using this configuration file. The "Configuration file" contains options to change the type of executable being built and select which revision of the configuration files are to be used (e.g. app/fcm_make/rose-app.conf). The "make steps" section allows the step list of the build to be modified. A schematic work flow diagram of NCMRWF's UM Build System is depicted in Fig.2. The optimisation level of the build, whether or not to compile with OpenMP and the prebuild location can all be found in the "Basic compilation" section. More advanced settings such as compiler flags/overrides and library paths can be found in the "External libraries" and "Advanced compilation" sections. These sections may be hidden as many of the fcm_make environment variables are not set to compulsory.

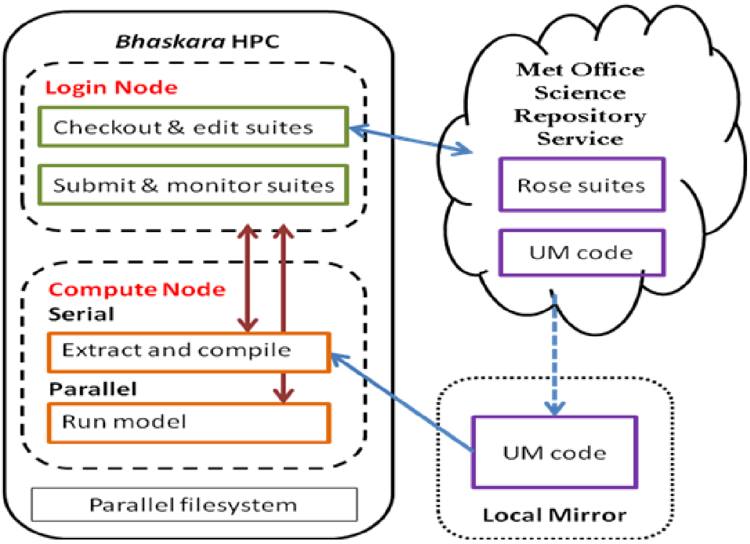

Fig 2: Schematic work flow diagram of NCMRWF's UM Build System

3.2.2 UM app

a) Namelists

All app namelists are contained in the namelist section. One of the first differences compared to a UMUI panel is that the UM namelist item names are visible in addition to the more familiar descriptions underneath. Range and type checking of variables is done as soon as the user enters a new value.

b) STASH

The four STASH namelists: *streq* (STASH Requests), *domain* (Domain Profiles), *time* (Time Profiles) and *use* (Usage Profiles) can all be found in the STASH section as shown in Fig.3. By default STASHmaster entries are grouped together by Section code.

It is possible to group items by any of the STASHmaster codes using the Group drop down list.

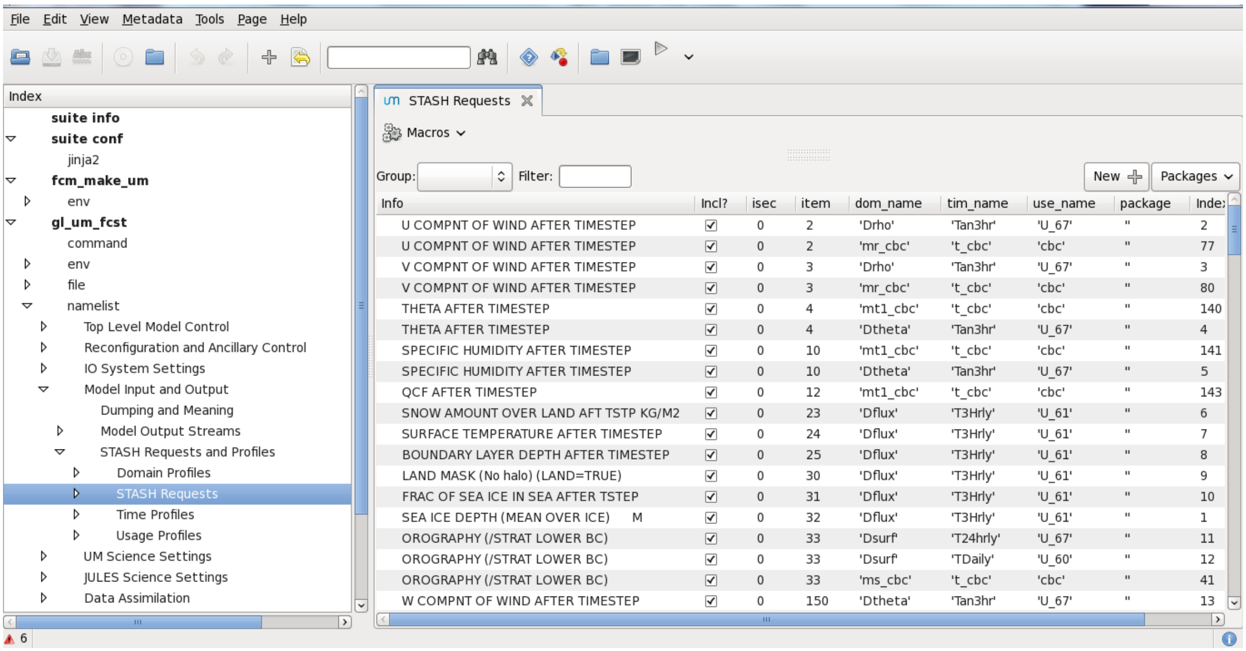

Fig.3: STASH Window in Rose GUI

3.3 Suite Outputs

The cylc keeps a runtime directory for each suite in *$HOME/cylc-run/*, which is updated when you run the suite. It is a mirror of our suite directory structure and content, but with some added cylc directories and symbolic links (see Fig 4.). The directory holds items like installed files, current task status and task output.

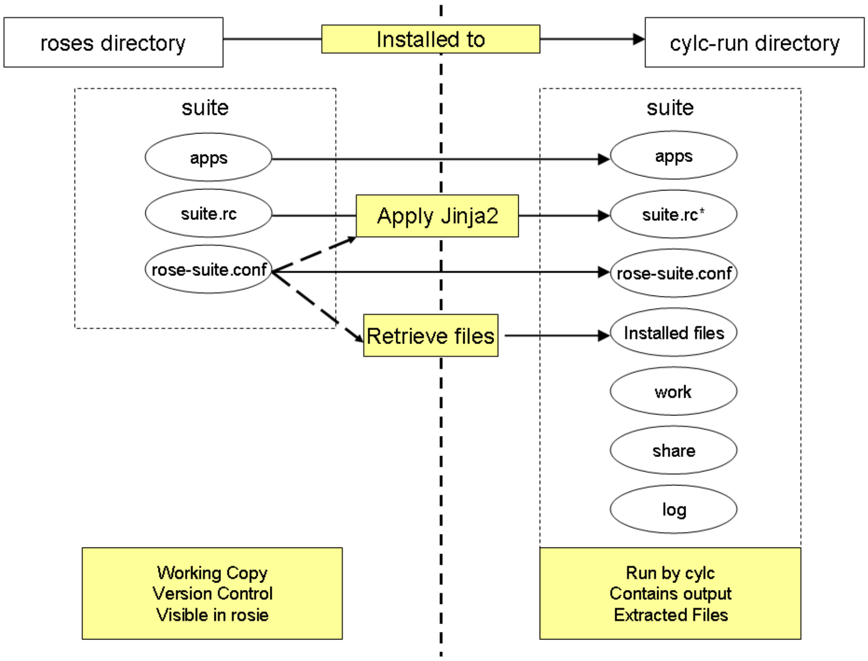

Fig. 4: Mapping of suite and output directories

a) Rose Bush

Rose Bush is a simple service to browse suite logs over a web browser. The output from a standard suite goes to a variety of places, depending on the type of file. The standard output and errors from the suite can be viewed (together with other information about the suite) using *Rose Bush*. At NCMRWF, Rose Bush is set up on *Bhaskara* HPC at http://192.168.0.222:8080/ (see Fig. 5).

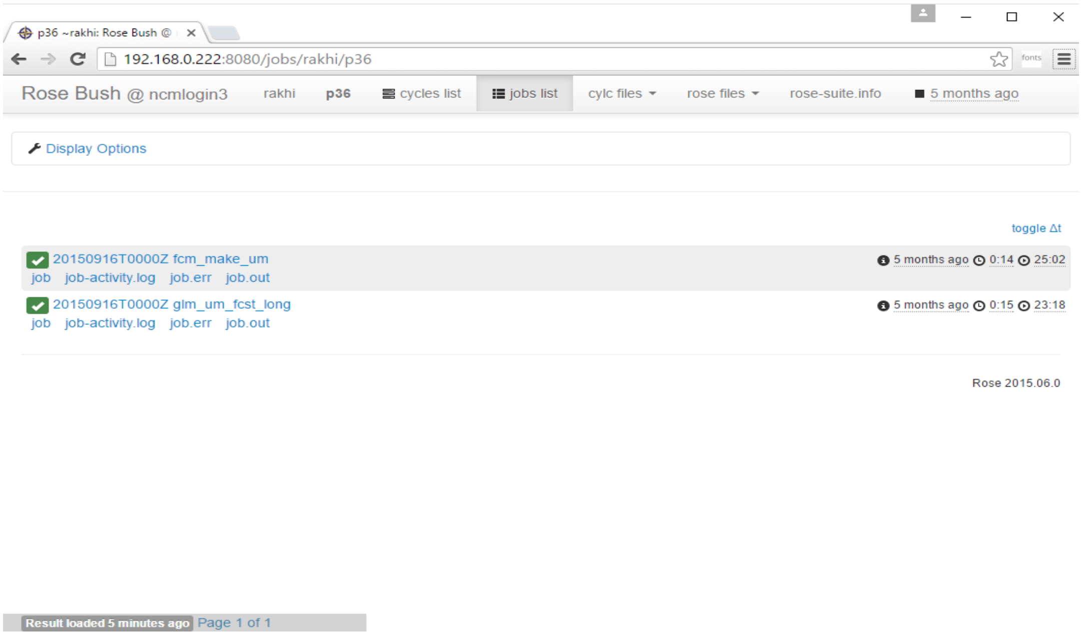

Fig. 5: Rose suite log viewer (Rose Bush) over a web browser showing current and past suites and their tasks.

b) Compilation Output

The output from the compilation is stored on the host upon which the compilation was performed, inside the directory *~/cylc-run/<suitename>*, where *suitename* is the name of the suite. The output from *fcm_make* is symbolically linked inside the directory containing the build, which is stored inside the *~/cylc-run/<suitename>/share* subdirectory.

c) Standard Output

The UM scripts which run the executable makes use of UNIX command cat to copy the output from PE0 to standard out. This is captured by the queue system, reported to Rose, and then copied to the *log* subdirectory, e.g.

~/cylc-run/<suitename>/log/job/1/<appname>/NN/job.out.

This has a corresponding error file:

~/cylc-run/<suitename>/log/job/1/<appname>/NN/job.err.

Here *NN* is a symbolic link created by Rose to the output of the most recently run task of that name.

d) Binary output - share and work

The output files of the UM are controlled by two environment variables, DATAM and DATAW. These are typically set to be somewhere in the share or work directories; if they are not explicitly specified within the suite they default to: *~/cylc-run/<suitename>/share/data.* Some suites direct most task output to work instead. This ensures that tasks cannot interfere with each other at the expense of functionality such as restarts (which require tasks to access the same data). The output goes to the task's work directory: *~/cylc-run/<suitename>/work/<appname>*.

3.4 UM10.1 (N768L70) ENDGame Forecast run

At NCMRWF, UM10.1 (N768L70) ENDGame suite has been developed based on UK-Met Office's Parallel Suite (PS36), which has the Rose suite-id: u-aa437. This suite will build and run the UM10.1 global model at N768L70 resolution up to 10-days. The suite can be started using the rose command *rose suite-run* (some useful rose commands are listed in Appendix-1). Once the suite has started, the cylc GUI (*gcylc*) will appear which allows us to track the progress of the suite graphically (see Fig. 6). Sample plots of the model's Orography at N768 resolution and predicted rainfall (Day-1 and Day-3) are given in Appendix-4.

3.5 Scaling of UM10.1

In order to get a better wall clock time with UM10.1 (ENDGame) at N768L70 resolution, lots of experimentations were performed by varying I/O and processor/node combinations, compiler options (while building GCOM and UM) and dump frequency settings. Additional set of platform specific parameters (see, Appendix-2) were also included as environment variables in rose-app-run.conf. The lowest timing of ~4500 seconds for a 10-day forecast was obtained with a processor combination of 64 x 48 and 4 I/O servers (equivalent to 196 nodes).

Acknowledgments

The authors also thankfully acknowledge the support provided by K. Bangarusamy of M/s IBM India Ltd. during the implementation and up-gradation activities.

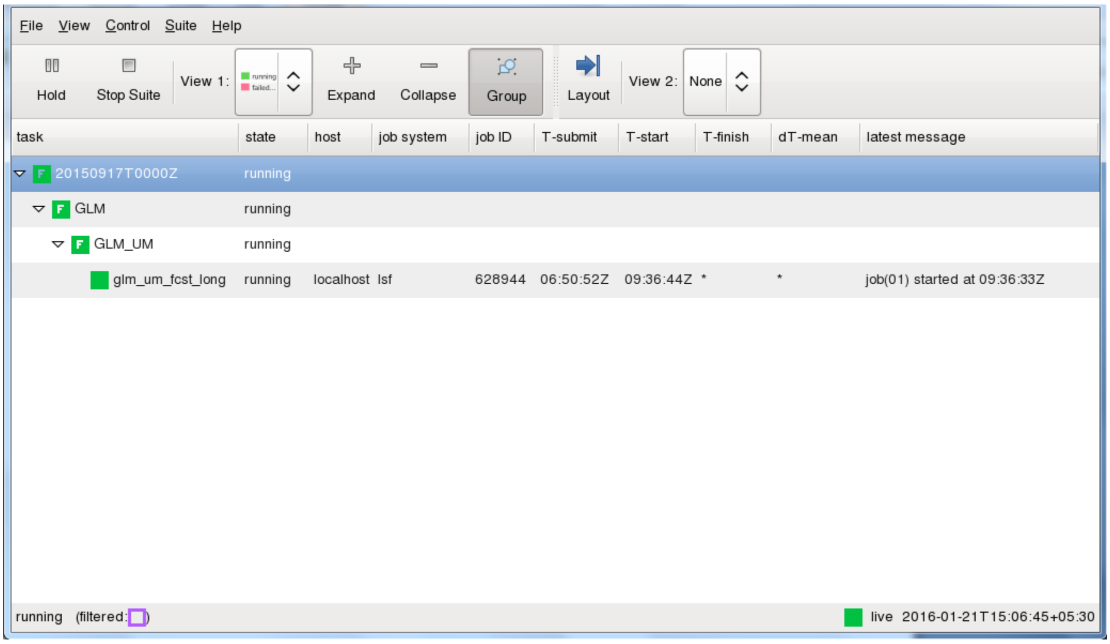

Fig 6: The cylc GUI (*gcylc*) depicting the running status of UM Global Model

References

Davies, T., M. J. P. Cullen, A. J. Malcolm, M. H. Mawson, A. Staniforth, A. A. White, and N. Wood, 2005: A new dynamical core for the Met Offices global and regional modelling of the atmosphere, *Q. J. R. Meteorol. Soc.*, Vol. 131 (608), 1759-1782.

Klingaman, N. P., and S. J. Woolnough, 2014: Using a case-study approach to improve the Madden-Julian oscillation in the Hadley Centre model, *Q. J. R. Meteorol. Soc.*, Vol. 140 (685), 2491–2505.

Lock, A. P., 2001: The numerical representation of entrainment in parameterizations of boundary layer turbulent mixing, *Mon. Wea. Rev.*, Vol. 129 (5), 1148–1163.

Mancell, J., 2015: Unified Model Installation Guide, *Unified Model Documentation Paper (UMDP) X04, UM Version 10.1*, 28 p.

Selwood, P., 2014: Input/Output Services in the Unified Model, *Unified Model Documentation Paper C10*, 35 p.

Wood, N., Staniforth, A., White, A., Allen, T., Diamantakis, M., Gross, M., Melvin, T., Smith, C., Vosper, S., Zerroukat, M., and Thuburn, J., 2013: An inherently mass-conserving semi-implicit semi-Lagrangian discretization of the deep-atmosphere global non-hydrostatic equations, *Q.J.R. Meteorol. Soc.*, *140*, 1505–1520.

Zhong, W., and J. D. Haigh, 2003: Shortwave radiative forcing by stratospheric water vapour, *Geophys. Res. Lett.*, Vol. 30(3), pp. 4, note. 1113, doi: 10.1029/2002GL016042.

Rose documentation: [http://metomi.github.io/rose/doc/rose.html.](http://metomi.github.io/rose/doc/rose.html)

Appendix -1

1. Introduction to Rose/cylc:

Rose is a group of utilities and specifications which aim to provide a common way to manage the development and running of scientific application suites in both research and production environments. The suite and the applications it runs are driven mainly by configurations that can easily be understood and modified by a human user as well as a computer.

1.1 Components of Rose

Rose has the following components:

a) *Application configuration and runner*: Rose provides a simple and portable configuration driven application runner to run individual scientific applications in suite tasks.
b) *Configuration editor GUI (Rose GUI*): The Rose config editor in combination with the meta-data file which describes UM inputs is the new GUI for editing UM apps. Configuring the UM under Rose aims to have greater transparency between the options presented to the user in the GUI and the underlying namelist inputs.
c) *Suite engine*: Rose uses *cylc* (a workflow engine and meta-scheduler) for its suite engine, which is developed by Hilary Oliver at National Institute of Water and Atmospheric Research (NIWA).
d) *Suite installer*: Rose provides a configuration driven installer to install and start a cylc suite. This can install additional files from locations such as version control systems and mirror the suite to any remote platforms where tasks are to be run.
e) *Common suite utilities*: Rose provides various utilities to allow suites to be written in a common way. This includes a task runner, which makes it easy to run Rose applications and utilities within suite tasks, a task hook to notify task failures and interact with the suite log viewer, plus utilities for file installation, housekeeping, archiving, etc.
f) *Rose Bush*: Rose Bush is a simple service for users to browse their suite logs over HTTP on their intranet.
g) *Suite storage and discovery*: Rose suites are just files in a directory hierarchy, so they can be stored anywhere. Each centrally hosted suite will provide an identity file, which the system can use to populate a database. Users can discover relevant suites by querying the database.

1.2 *Useful Commands*

rose suite-run --name=<suitename>	Run a suite
rose suite-run –new	Equivalent to "rose suite-clean && rose suite-run"
rose suite-run --restart	Restart a suite from the point it had got up to. Most useful when an app fails. In that case, you can edit the app and then restart the suite, which will bring up the gcylc window.

rose suite-scan	Returns a list of the suites you have which are currently running
rose suite-shutdown --name= =<suitename>	Shut down a suite
rose suite-gcontrol --name=<suitename> OR rose sgc –name=<suitename>	Re-open a cylc GUI
rose suite-clean --name= =<suitename>	Removes items created by rose suite-run, including items in $HOME/.cylc/, items on task job hosts and actual run locations under alternate disks.
rose config-edit OR rose edit	Evokes rose GUI

Appendix-2

Set of platform specific parameters specified by environment variable

KMP_AFFINITY=granularity=fine,compact,verbose

KMP_STACKSIZE=1024m

LD_LIBRARY_PATH=/gpfs1/home/Libs/INTEL/GRIB_API/lib:$LD_LIBRARY_PATH

MP_CLOCK_SOURCE=OS

MP_COREFILE_FORMAT=light_core

MP_CSS_INTERRUPT=no

MP_DEVTYPE=ib

MP_EUIDEVELOP=min

MP_EUIDEVICE=sn_all

MP_EUILIB=us

MP_INFOLEVEL=0

MP_LABELIO=yes

MP_MPILIB=mpich2

MP_MSG_API=mpi

MP_PE_AFFINITY=yes

MP_POLLING_INTERVAL=100000000

MP_PRINTENV=yes

MP_SHARED_MEMORY=yes

MP_SINGLE_THREAD=yes

MP_STDOUTMODE=unordered

MP_TASK_AFFINITY=cpu

MP_USE_BULK_XFER=yes

MP_WAIT_MODE=poll

Appendix-3

Installation Procedure of UM10.1

cd $HOME

##Check out the working copy

fcm co svn://ncmlogin3/SREPOS/roses-u /a/a/4/3/7 UM10.1N768e

cd UM10.1N768e

Setting up platform configuration files

mkdir fcm-make

cd fcm-make

fcm cp fcm:um.xm/branches/dev/paulcresswell/vn10.1_xc40_oasis_library/fcm-make/inc inc

fcm cp fcm:um.xm/branches/dev/paulcresswell/vn10.1_xc40_oasis_library/fcm-make/meto-x86-ifort ncm-ibm-ifort

Make NCMRWF specific configuration changes in ncm-ifort-mpich.cfg & um-atmos-high.cfg

cd ../ UM10.1N768e

Make the suite suitable for Global UM N768e Forecast

vi suite.rc

Set up the suite configuration file to be compatible with suite.rc

vi rose-suite.conf

rose suite-run

Appendix -4

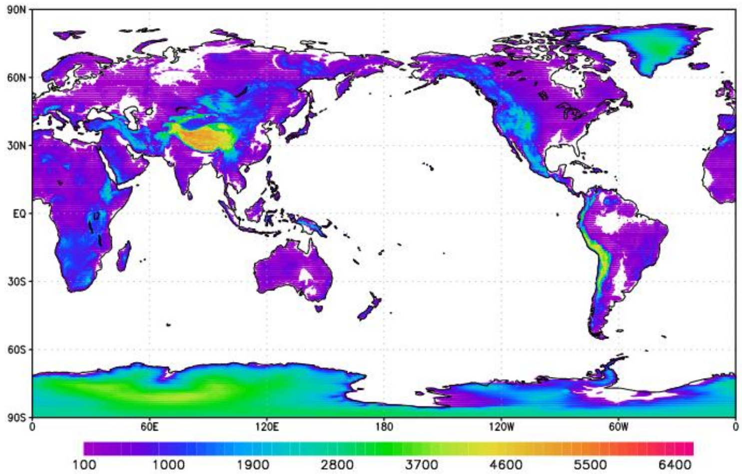

Fig 7: The model's global orography at N768 horizontal resolution

(a) **(b)**

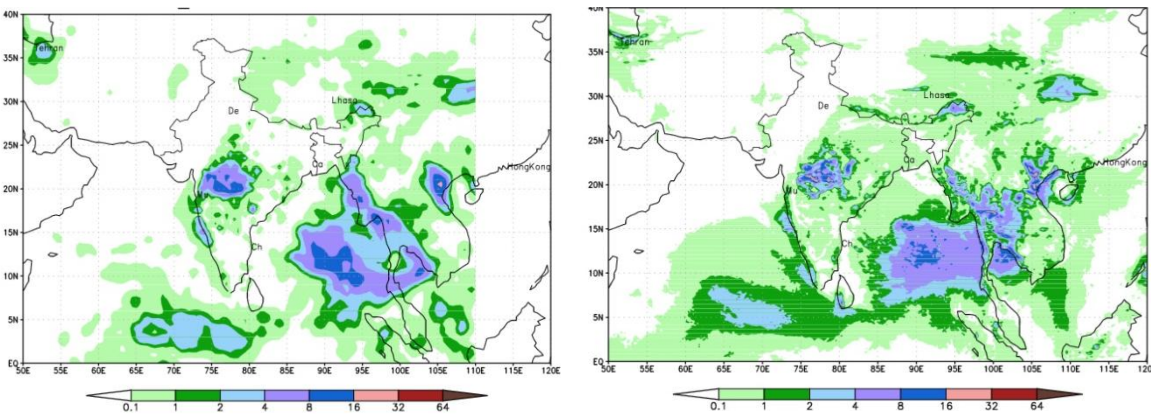

Fig 8: (a) IMD-NCMRWF Observed Rainfall valid on 18th September 2015
(b) Day-1 Model Rainfall forecast valid on 18th September 2015

(a) **(b)**

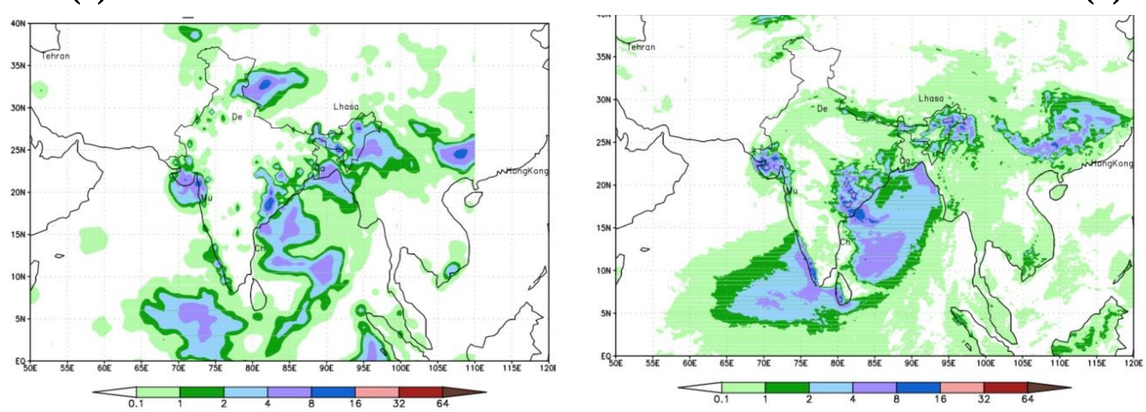

Fig 9: (a) IMD-NCMRWF Observed Rainfall valid on 20th September 2015
(b) Day-3 Model Rainfall forecast valid on 20th September 2015