Making the ECMWF tools, including METVIEW, available in a restricted version of the PAIPIX scientific live Linux.

António Amorim[1] and Lígia Amorim[2]
[1]SIM/IDL and Fac. Ciências da Universidade de Lisboa, Ed. C8, Campo Grande
[2]IM - Instituto de Meteorologia, Rua do Aeroporto, Lisboa

Abstract

Setting up a platform for meteorological data analysis and application development, that fully supports the ECMWF tools and libraries can be quite challenging for institutes whithout a large team devoted to local support with the required expertise. At *Instituto de Meteorologia* – IM, this challenge was addressed by developing a special version of the scientific Gnu/Linux live distribution, PAIPIX [1], including the main ECMWF software packages. The work also consisted in creating the appropriate Debian packages and patches for the ECMWF applications using the Debian Sarge distribution and the Linux kernel 2.6. The platform can be directly booted from a DVD (live) or installed on the hard disk,becoming easy to run by new users. The work also integrated the database tools currently being developed at the institute, including our tidb2[2] grib[3] and bufr[4] enabled database interface, making it readily installed. Compilation rules are made available to allow the users to link both using the free and proprietary compilers. The special PAIPIX version with meteorological tools supports several languages, following the pattern of the general (http://www.paipix.org) distribution.

1 Introduction and Motivation

Running a meteorological operational system, including the typical client/server applications, usually implies facing the challenges of maintaining and supporting several distinct computing platforms. Moreover the typical number of problems that have to be solved scales more with the number of platforms and applications than with the number of users and the amount of data that has to be processed. This makes the present multi-platform multi-application situation specially difficult to manage for organizations/institutes of medium or small size like the Portuguese *Instituto de Meteorologia* – IM that have a rather small team of people devoted to these type of problems. This difficulties are the close motivation of the present work.

While better tools and new developments have continuously been made available within the international community, the combination of old platforms that are no longer supported by original internal or external code developers has stacked over time obsolete computing applications that continue to run operationally providing the "usual" information for the forecasters. While it is relatively easy to motivate the management structures to allocate resources to brand new developments aimed at new products it is harder to justify the work of bringing up to date the operational software or hardware infrastructure merely to be able to accommodate generic future developments. The presence at IM of a first generation of applications that fulfil the basic operational requirements constitutes, in this way, a blocking factor, not only to the investment in new applications but more importantly in keeping the operational software and hardware infrastructure relatively up to date in order to enable that future developments can be feasible within a reasonable effort. Another important factor that disables innovation results from insufficient training in recent computing technics for meteorologists at universities considering, in particular, that they will in practice carry out this type or work instead of regular computer engineers.

Computational systems have evolved from a typical mainframe infrastructure, where all users run applications on the same centrally maintained machine, to a distributed client/server framework with each computing node, or group of nodes, providing a given service to other computing hosts including the user desktop machines. While this new approach provides a clearer separation of responsibilities among different nodes, the spread of applications and code must be carefully managed to be able to cope with the extra complexity of the system. The minimal requirements to manage the new type of distributed infrastructure include a central software repository (using CVS[5] for example) managing all the source code and a central task manager using a control infrastructure like SMS[6] or similar. A further consequence of the distributed nature of the present computer infrastructure is the need to cope with heterogeneity since one can no longer aim at a simultaneous upgrade of all different platforms. Due to historical reasons the distributed computer infrastructure at IM, for example, is composed from a supperposition of different platforms, that were the "default" development environment at a given time, running the applications that were actively developed at that time. The present diversity includes a Micro VAX/VMS machine, supporting many important applications, several Digital Alpha based workstations that could never fully support the meteorological applications imported from ECMWF, and several Linux and windows PC's. The maintenance costs of many of these obsolete platforms is presently prohibitive.

The tremendous development of the GNU/Linux based PC software and hardware architecture over the recent years provides an unique opportunity to drastically change the present picture that was depicted above. In fact, not only does the low price of the very powerful commodity hardware enables the acquisition of a large number of machines, that can support all the data processing now spread over different platforms, but also the nature of free software has lead GNU/Linux to a singular position making as a platform where an enormous amount of computing applications and development libraries is available. This constitutes an opportunity, for the first time, to carry out the migration of existent applications and the development of new ones concentrating only on the technical issues that can be solved without the complex and slow acquisition of individual specific software packages.

The development of a dedicated release of PAIPIX Linux including the IM specific tools was aimed at easing the migration of the institute computing infrastructure to a more coherent and updated status while minimizing both the cost and the diversity of the final systems. The same hardware and software platform supports and installs both the client and the server applications that could eventually run the entire institute production system.

2 PAIPIX live Linux

This Section describes the free version of PAIPIX that is available on the web and does not include the ECMWF tools due to license restrictions. The way how these tools have been ported to a special restricted version of PAIPIX is described in Sec. 3.

The evolution of GNU/Linux over time included several attempts to diversify the user community by making the installation of a Linux system easier and more automatic. Installing the GNU/Linux operating system on a PC typically involves understanding the best way to configure the packages supporting the particular computer hardware including configuring display drives, media controllers, network and so on. The PC open architecture enables a enormous range of different compatible boards and devices to be connected in a given PC making it is generally hard to know in advance how the system will be able to cope with a particular hardware. This is one reason why the main commercial Linux distributions include a live CD or DVD system that is essentially meant for a one-time-test and is not really exploiting the full potential of the computer where it is running. This limited approach results also from the fact that the bandwith for CD or DVD reading, used while starting any application, is rather limited.

In parallel, the evolution of Linux distributions has gone trough several stages starting from small groups of maintainers that would pick the different source releases for many different applications, including the Linux kernel, and compile them in an integrated environment to be distributed. At a later stage, larger teams were gathered to support commercial distributions and a package managing system was introduced that consisted of a database of all files and applications installed in the system. This development enabled both the installation and the removal of applications that were distributed in special compiled format. In parallel, non-commercial teams like the Debian developers were systematically growing at a slower pace. Finally the expansion of the internet brought an invertion of the effort spent on commercial versus non-commercial distributions by making non-commercial free distributions like Debian or Gentoo include a much wider development community that any singly company could afford. This resulted also in the development of new free package managing systems that are better than their counterpart created in the commercial world. The distributions managed by these open communities are strongly network based and generally can be installed using a small kernel system and repository of individual package files available on the network. Arguably, these systems may not be very encouraging for absolute beginners but anyhow provided a common environment for developers to either install their machines or assemble an enlarged set of the commonly used software and make it ready to run.

The KNOPPIX[8] system resulted from exploiting the opportunities created by the Debian distribution and the particular development of an highly compressed filesystem driver (cloop and later squashfs) that enabled not only including several gigabytes of applications in one single CD but also made its usage much faster due to the fact that the size of the data that is actually red from the low bandwith CD is much less than the size of the actual application. The kudzu[13] automatic hadware configuration mechanism, initially developed by a commercial maintainer, was adapted to provide easy startup of all the hardware services.

The free nature of the KNOPPIX live CD project has enabled the development of a large amount of different “remastering” distributions, using the Debian distribution to create integrated computing environments to suit a wide variety of usage patterns. The amount of effort left to the user is basically to download the live CD/DVD and boot his PC using it. One recent very interesting development was the usage of the unionfs[11] file system, already included in PAIPIX, that allows to merge the ramdisk and the readonly CD/DVD filesystem to create a writable disk interface allowing the installation of any set of packages while running live from the CD/DVD.

Once the PAIPIX live Linux is running, a rather simple command automates all the tasks associated with the installation on hard disk. Even after having the system installed on disk, the DVD can be of use if one needs to modify the root filesystem in such away that can not be accommodated with running the system from disk.

The free PAIPIX DVD project is a KNOPPIX remastering specially suitable for the science and technology domains, including four new main sets of features, namely:

- An extended set of scientific applications in physics, mathematics, engineering, astrophysics, statistics, scientific typesetting, numerical and algebraic processing and many other;
- The most important server software for shared filesystems (NFS, AFS, samba), web (apache), relational databases (MySQL, Postgresql), email (exim and cyrus), the server side of many content management systems and an extended “terminalserver” package enabling diskless cluster setup;
- Multi-Language support. Most PAIPIX applications are installed to automatically support both English, German, Italian, French, Spanish and Portuguese;
- An extremely complete code development environment for many different computing languages.

The complete and readily available nature of PAIPIX makes it ideal to enable the work of science and engineering students on their personal computers or laptops. Since many scientific applications make use of 3D enabled display drivers, an extensive support for free and proprietary drivers is included. With good 3D graphics support and the student population in mind, it was unavoidable to include also a wide range of the best free Linux games.

The PAIPIX system can be downloaded from http://www.paipix.org and was tested both to be run as a live system and as a support for the Debian disk installation, a mechanism that was reused from KNOPPIX and extended to cope with the new PAIPIX features. Once installed on disk the, regular updates can be performed using the Debian standard applications like “aptget” and “dpkg” that also allow tuning the set applications installed on disk. The main PAIPIX information is kept on a set of wiki pages enabling user contribution and improvements in any of the several supported languages.

There were three types of developments required to create PAIPIX:

- Selecting the relevant applications available on the Debian distribution to include;
- Creating and configuring new applications not available from Debian packages;
- Final tuning and configuration of special files.

App. A summarizes the main packages and fixes included in PAIPIX.

Different approaches can be followed in pursuing the tasks mentioned above but one rapidly realises that very special care has to be put on choosing a methodology that can cope with the enormous challenge of reusing and keeping the work synchronized with all developments of an extended diversified community. In fact, allowing development reuse and staying synchronized is probably the most demanding challenge involved in developing and maintaining the PAIPIX live DVD. It can only be accomplished because of the experience incorporated in the free software development community methods also incorporated in the debian GNU/Linux distribution framework. If one compares all difficulties that our organizations usually face in maintaining, in a coherent state, a small number of applications, one must realise the absolutely fantastic nature of all the procedures of an international free organization like Debian that has to maintain thousands of packages working nicely together while continuously upgrading them. This is accomplished by a sophisticated dependence mechanism that includes automatic dependency checking on package creation and package installation.

The PAIPIX remastering uses this mechanism follows the policy of creating Debian packages even if they were not available already. These packages are also made available on the PAIPIX web site both in source and binary formats using the debian package management tools.

The final tuning item is related to the configuration and general changes in different files. This process usually involves localized modifications in several dozen files, already belonging to other debian packages, that are spread over the entire file system. Furthermore, these files will be updated as the corresponding debian files get updated and the modifications will have to be reissued. The default debian process to achieve this goal is called “diversion” and essentially blocks the modified files from being changed while the package is being upgraded. In PAIPIX we have introduced a special paipix-patches package that keeps all modifications together as “diff” files. By removing this package all modifications in the system are removed while by issuing “dpkg-reconfigure paipix-patch” all modifications are first dropped and than applied from scratch. This procedure tries to reconfigure the system even if some file in some underlying debian package has been modified in the upgrade process. The way the patches are applied is one of the most delicate issues in PAIPIX that is under revision for version 5.0.

2.1 Dealing with the persistence of user files

As noted before the PAIPIX system can either start live from the DVD read-only filesystem or be installed in the hard disk drive, becoming a Debian based installation. While the installation in the hard disk presents the optimum configuration and tuning possibilities, in some cases, it might be preferable to use the live system for regular usage since it can always be reset to its initial state and does not have to be managed at any level. This feature could be of use at the operational centers where many users share the same machines. Users served by this type of installation would however be limited to writing their files in the ramdisk that would be erased upon reboot.

The PAIPIX system includes several solutions to address the problem of making the user files persistent in a live installation environment. One way is to use a special file in one of the local hard disks or user USB drives and the other possibility is to distribute the home directories over a network based filesystem. The local disk approach is activated by including in the boot prompt a statement like "home=scan" that scans hard disks or USB storage devices for a special file that is then used as a persistent repository for all user files. The client/server solution is implemented using the Andrews File System -AFS -approach and can be activated by including in the boot prompt "afs=<<cell>> afsuser=<<user>>". This procedure will login the user to an AFS fileserver and mount the home directory from the server. The AFS filesystem solution implements local caching of the shared files and is therefore extremely tolerant on the network performance. The PAIPIX system includes also all the software necessary to implement the local AFS server infrastructure.

2.2 *A PAIPIX terminal server facility*

A utility that was picked form KNOPPIX live CD and further extended is the "terminalserver" that starts all necessary services to allow remote nodes to boot over the network from a server using the live CD/DVD. This facility was revised and extended by the clusterknoppix[10] project. ThePAIPIX distribution included about 25 corrections/improvements in this utility ranging from ssh access to farm nodes to proper DNS configuration, MPI configuration, etc. The terminal server model is presently being used at IM in a diskless farm running a mesoscale meteorological model using MPI.

The terminalserver services include:

- PXEboot and miniroot image creation;
- sharing the compressed file system trough NFS;
- optional local DNS server and iptables rooting;
- ssh services and validation.

3 APAIPIX release including the ECMWF tools

This Section describes the version of PAIPIX that includes the licensed ECMWF tools.

Following the general PAIPIX policy, creating a special release for IM that includes the ECMWF meteorological data analysis, visualization and data processing applications, implied the development of separated Debian packages that can be installed both for the live DVD remastering and in a generic PAIPIX disk installation. The applications that have been ported were EMOSlib, MAGICS, METVIEW and SMS as well as the separate GRIB and BUFR tools that can be freely downloaded from the ECMWF site. The EMOSlib and MAGICS standalone libraries were created using the special compilation procedure departing from the general metview source release. The IM operational forecast meteorological chart creation application was also included.

While the separate GRIB and BUFR libraries where compiled using the gnu g77FORTRAN compiler, the METVIEW package and associated libraries could only be compiled on Linux using the Portlang Group FORTRAN compiler. This results from the use of CRAY type of FORTRAN pointers that was a natural requirement as an extension to the FORTRAN language but was made obsolete by the FORTRAN 90 standard and has not been implemented in both gfortran and g95 free FORTRAN compiler implementations that adhere to the new standard. The use of the Portland family of compilers to build METVIEW is not a very limiting requirement since the compiler is not required to run the application. On the other side it would be an important restriction if user applications, that have to link to the EMOSlib and MAGICS libraries would have also to be compiled with the commercial compiler. Fortunately this situation can be avoided and the user applications compiled using gcc, g++, gfortran or f95 can be linked to the previous libraries by following the rules detailed in Appendix B.

Building METVIEW version 3.7 in the Debian framework required developing the debian make rules and the control files. A special subdirectory was created to build simultaneously the standalone libraries. One of the important steps required in building these packages was to establish the proper development environment. The debian control file reflects this knowledge and requires at building time the development packages, g++, flex, pdksh, libgl-dev, xlibs-dev, netcdfg-dev, libmotif-dev, libgd2-dev, libdb1-compat, vis5d-tcl and libgdbm-dev.

As a subproduct of the debian building process the resulting packages for METVIEW EMOSlib and MAGICS automatically require the installation of pdksh, vis5d-tcl, libc6, libfreetype6, libgcc1, libgd2-xpm, (libgdbm3 or libice6), xlibs, libjpeg62, libmotif3, libpng12-0, libsm6, libstdc++5, libx11-6, libxext6, libxmu6, libxp6, libxpm4, netcdfg3, xlibmesa-gl, mesag3, mesag3+ggi, (mesag3glide2 or libgl1-mesa-glide3), (xlibmesa-glu, libglu1-mesa or libglu1-xorg or xlibmesaglu) and zlib1g.

Knowing these external dependencies both for building and for installing the particular packages is important, not only to be able to run the applications in any Debian environment but also to support future developments.

An important change was required in Metview code when Linux kernel was upgraded. Kernel upgrade from version 2.4 to version 2.6 allows systems to be configured with the maximum available memory space up to4GB. When maximum memory space is configured above2 GB Metview/Magics memory management stops working. Magics drivers' 32 bit C++ memory pointers (which correspond to 'unsigned int' in C++) are in the calling FORTRAN code stored in signed FORTRAN integers. When a pointer value is greater than 2GB,the most significant bit is taken as a sign bit and the value inside FORTRAN code is interpreted to be negative. Once the cause of the problem was identified the fix was an easy one.

The METVIEW application is available trough the user desktop menu system and requires no extra configuration, making use of the OpenGL features supported by PAIPIX.

4 Integrating the TIDB2 database interface

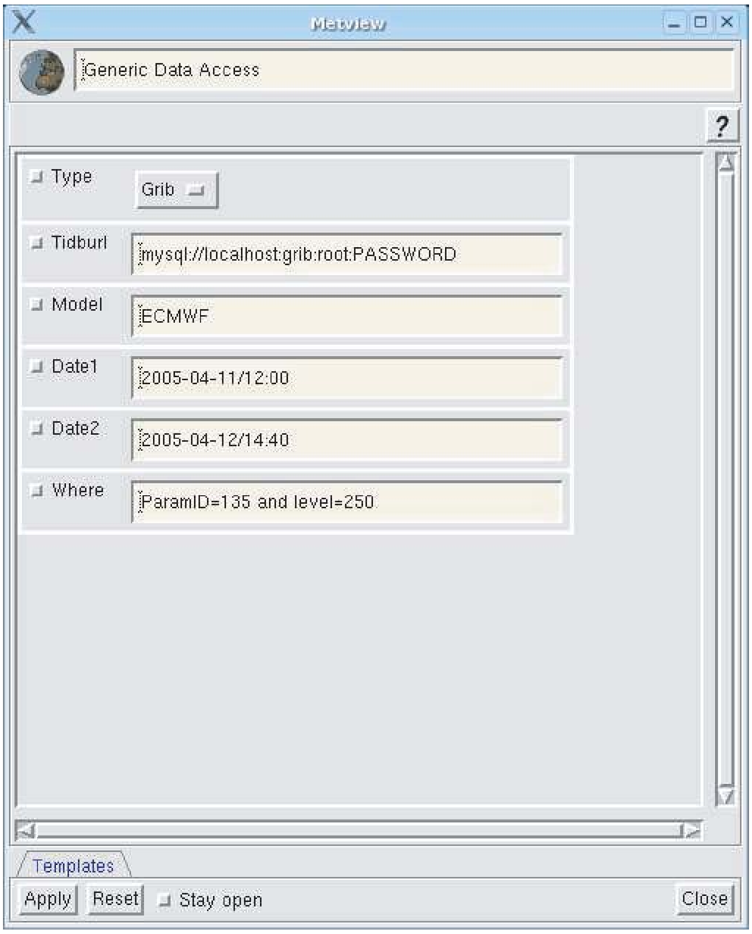
The tidb2 database interface[2] is being developed as a multi-disciplinary free-software project at the University of Lisboa. It was initially motivated to support the calibration and alignment data of the particle physics ATLAS experiment at the Large Hadron Collider at CERN. This project has recently been extended to support meteorological data at IM. The main purpose of the tidb2 project is to provide an unified interface to relational database management systems to support time dependent scientific and technical data, including the most important scientific data formats, through an unified object oriented interface. Presently both MySQL and Post gresql DBMS are supported while the ORACLE plugin is being finalized. The tidb2 interface addresses the time evolution of data trough automatic time interval and versioning mechanisms in the relational database management systems. It also provides an hierarchical folder like structure to accommodate the data in the database systems and enlarges the relational model by defining attributes of many different scientific types like arbitrary arrays, ROOT[7]objects and standard WMOGRIB[3] and BUFR[4] data, trough the use of run-time plugins.

The tidb2 main packages were included in the PAIPIX live DVD by creating the appropriate debian packages that include the tidb2 libraries, the binaries and the *ktidbexplorer*, a QT based unified database browser.

Making use of the server software included in the live DVD and the METVIEW interface to tidb2 that was developed specially for PAIPIX, one can follow in any single PC the entire tidb2 data processing and visualization chain. The main steps of this chain include:

- Starting the MySQL database server:
  /etc/init.d/mysql start
  you will be prompted for a new MySQL root password. Use "leno" in this
  example or modify bellow according to your choice.
- Defining the database connection string:
  `export TIDBURL=protocol://server:database:username:password`
  in general and
  `export TIDBURL=mysql://localhost:tidbexample:root:leno`
  for this particular example.
- Storing GRIB or BUFR data in the database. The tidbexample database
  will be automatically created:
  storegrib<CENTRE><gribfile>
- Browsing the tidb2 databases by connecting to the MySQL server tidb example database.
  Open in *ktidbexplorer* a database connection of the form:
  `mysql://localhost:tidbexample:root:leno`
- Visualizing fields in METVIEW. After starting METVIEW, create a new icon of the *Generic Data Access* type. Edit the created icon according to Figure1 and modify the parameters including the password, the dates and the parameter types in the SQL where clause. Close the tab.
- Visualize the *Generic Data Access* icon that was edited before.

The previous example illustrates the use of tidb2 in PAIPIX as an online database client/server system storing all data in a relational database. The extension of the usage of tidb2 to the offline case and managing historical data involves scaling the tidb2 database interface to historical dataset repositories is presently being investigated.



Generic Data Access

Type: Grib

Tidburl: mysql://localhost:grib:root:PASSWORD

Model: ECMWF

Date1: 2005-04-11/12:00

Date2: 2005-04-12/14:40

Where: ParamID=135 and level=250

Fig. 1 The METVIEW Generic Data Access Options

5 Possible PAIPIX usage in a meteorological information system.

As already discussed, the evolution of the IM information system provides an example of a highly complex and heterogeneous environment resulting from keeping the platforms used at a given time and from maintaining the set of applications developed at that time that are still bound to the corresponding platform. The migration of individual applications running in obsolete platforms has been extremely painful due to the fact that it has to cope with the interfaces to other obsolete solutions that either supply or require the data provided from the particular application that is being migrated.

The data paths, interfaces, and set of hardware and software solutions used in production are complex and difficult to summarize in a simple diagram.

The PAIPIX client/server Debian based approach, including the tidb2 interface, presents a potential for recovering from the present very fragile web of interdependent solutions to a distributed client/server infrastructure that can be extended to a large number of similar machines using only one single software and hardware platform. One possible simplified solution is presented in Figure 2. The work on the interfaces to GTS and to the surface, satelite and radar observation systems presents special challenges that would have to be addressed in order to implement a this overall solution.

6 Conclusions and further developments

The PAIPIX project has addressed the complexity associated with maintaining a complex scientific computing environment framework in a way that minimizes the required human intervention by making use of the GNU, Linux, Debian and KNOPPIX technologies. A large amount of effort was put to develop the procedures to enable the PAIPIX evolution synchronized with all developments that are happening at a fast pace in the Free Software world.

The special restricted version of PAIPIX for IM illustrates the possibilities of this approach to provide an alternative to the long and diverse stacking of historical applications and platforms that make the evolution of even maintenance extremely difficult.

Presently the new free PAIPIX version 5.0 is being finalized including besides a new amd64 release, the upgraded packages OpenOffice 2.0, KDE 3.4.3, Xorg 6.8, Mysql 5.0, Apache 2.0 and many more. The new version is remastered form KANOTIX [12] including a much better mechanism for hard disk installation and upgrade.

Concerning the meteorological systems, the new developments being investigated are focused on the offline high volume solutions for tidb2 including an interface to the MARS server archiving system from ECMWF.

7 Acknowledgments

The PAIPIX project is a very small contribution to the generous developments of an huge free software community gathered around the GNU, Linux and Debian projects. We would like to acknowledge the decision of the data processing division at IM to use PAIPIX at all desktops as a very helpful development to encourage testing and user feedback. The PAIPIX documentation pages have been translated to German by Helmut Wolters, to Italian by Vincenzo Vangoni, to French by Pedro Ferreia and to Spanish by Oscar Diaz Fouces.

References

[1] http://www.paipix.org

[2] TIDB2 – Implementation and Testing Object Extensions of Open Source RDBMS for Meteorological Data, João Simões, António Amorim and Maria Monteiro, at this Workshop and references therein.

[3] http://www.wmo.ch/web/www/WDM/Guides/Guide-binary-2.html

[4] http://www.ecmwf.int/products/data/software/bufr.html

[5] http://www.nongnu.org/cvs/

[6] http://www.ecmwf.int/products/data/software/sms.html

[7] http://root.cern.ch

[8] http://www.knoppix.org

[9] http://www.openafs.org

[10] http://bofh.be/clusterknoppix

[11] http://www.fsl.cs.sunysb.edu/pro ject-unionfs.html

[12] http://www.kanotix.com

[13] http://rhLinux.redhat.com/kudzu/

APPENDIX

A Main PAIPIX application classes

The scientific and technical packages that have been selected from the Debian distribution are presented in Table 1. The list of Debian packages specially created for PAIPIX is described in Table 2 and the set of files, associated to either the Debian distribution package, that are modified in the PAIPIX distribution are presented in Table 3.

Domain	**Applications**
Development/Programming	gcc; g++; gcc-4.0; g++-4.0;; .. Kdevelop
Debugger and profiler	ddd valgrind kdbg dbg
Development/FORTRAN	g77 gfortran-4.0; g95
Development/Java	jikes kaffe-pthreads ;gcj-4.0
Software testing	qmtest
Numerical Mathematics	Gnu Scientific Library, LAPACK
Algebraic Mathematics	Maxima (xmaxima); ...
Calculus	Octave , scilab
Distributed System	(CORBA,rpc...) OmniOrbi4
Parallel Computation	(MPI, PVM) lam4, pvm, mpich
Scientific Documentation	Latex; Kile; TeXmacs
UML case tool	Umbrello
Electronics	kicad; pcb; geda; ngspice; verilog
3dModel/2D/FEA	kpovmodeler gmsh FreeFem FreeFem3D
Animation 3D	blender, k3d
Database (DBMS)	mysql postgresql
Data Analysis	Root
Simulation of radiation	geant3 isajetPythia herwig
Earth Science charts	gmt
Meteorological charts	Grads, vis5d
GIS	Grass
Statistics	R
Medical images	(DICOM) amide, xmedcon, medcon
AFS file-system client/server	openafs

Table 1 Selected Scientific and technical Packages in PAIPIX.

etcskel-paipix	paipix-bookmarks	paipix-patches
sermenu-paipix		
care2x	amide	
bufr	gribex	grads
tidb2	ktidbexplorer	vis5d-tcl
ng-spice	kicad	labjack
tikiwiki	phpmybibli	kernel-hs-paipix
myspell-pt-paipix	kile-templates-paipix	
evaristo	g95	

Table 2 List of Debian packages created specially for PAIPIX.

```
etc/
     apt/sources.list
     skel/.kde/share/config/kickerrc
                                      konsolerc
                                     profilerc
                                     kcmnspluginrc
     X11/XF86Config-4.in
           Xsession.d/45xsession
     apache/httpd.conf
     php4/apache/php.ini
     init.d/knoppix-autoconfig
              mysql
              postgresql
              openafs-client
              xsession
              knoppix-hd-config
              knoppix-reboot
     postgresql/pg_hba.conf
     openoffice/psprint.conf
     menu-methods/zzz-update-menus-knoppix
     kde3/kdevelop3rc
     motd
     exim4/exim4.conf.template
     squirrelmail/config.php
     imapd.conf
     default/ntpdate
     mysql/my.cnf
     default/mpd usr/sbin/netcardconfig
     modprobe.d/nvnet
     mkinitrd/mkinitrd.conf
usr/share/hwdata/pci.ids
                          pcitable
              applnk/OpenOffice.org1.1/.directory
                                                   calc.deskto
                                                   draw.desktop
                                                   impress.desktop
                                                   math.desktop
                                                   printeradmin.desktop
                                                   template.desktop
                                                   web.desktop
                                                   writer.desktop
               applications/kde/kdevelop_c_cpp.desktop
                                        kdevelop_ruby.desktop
                                        kdevelop_kde_cpp.desktop
                                        kdevelop_scripting.desktop
 Knoppix/Config/.directory
               Knoppix/.directory
                            Services/.directory
               knoppix-installer/modules/install/02-install-tools.bm
               knoppix-terminalserver/templates/miniroot/Linuxrc
               wine/skel/config
   bin/openoffice
   sbin/update-rc.d
           knoppix-terminalserver
          mkxf86config
   lib/zope2.7/skel/etc/zope.conf.in
       menu/tuxmath
```

Table 3 List of files modified in the PAIPIX distribution to solve problems associated to either the Debian packages or the KNOPPIX live CD. 14

B Example make rules to compile user code with the ECMWF libraries in PAIPIX.

For user applications using EMOSlib the following rules can be used:

```
# export  LOCAL_DEFINITION_TEMPLATES=/usr/local/apps/METVIEW/etc_emos/gribex_templates
# Using the Portland compiler:
exemple.o: exemple.f
        pgf90 -c   -tp px -Ktrap=fp -i4 -r8  -g exemple.f
exemple: exemple.o
        g++ -o exemple /usr/pgi/Linux86/6.0/lib/f90main.o exemplo.o  -L/usr/local/apps/METVIEW/lib/ \
-lMagEmos -L/usr/pgi/Linux86/6.0/lib -lpgf90 -lpgf90_rpm1 -lpgf902 -lpgf90rtl       \
-lpgftnrtl -lpgc -lpgf90 -lgd -Wl,-rpath,/usr/pgi/Linux86/6.0/lib/

# Using the  g77 gnu compiler
# Do not forget to use IMPLICIT DOUBLE since there is no -r8 option
example.o: example.f
        g77 -c -fugly-logint -g example.f

example: example.o
        g++ -o example /usr/lib/libfrtbegin.a example.o  -L/usr/local/apps/METVIEW/lib/ \
-lMagEmos -L/usr/pgi/Linux86/6.0/lib -lpgf90 -lpgf90_rpm1 -lpgf902 -lpgf90rtl \
-lpgftnrtl -lpgc -lpgf90 -lgd /usr/lib/libg2c.a -Wl,-rpath,/usr/pgi/Linux86/6.0/lib/

# Usig the g95 free compiler
# Do not forget to use IMPLICIT DOUBLE since there is no -r8 option
example.o: example.f
        g95 -c -i4 -r8  -g exemplo.f

example: example.o
        g++ -o example /usr/lib/gcc-lib/i686-pc-Linux-gnu/4.0.0/libf95.a exemplo.o \
 -L/usr/local/apps/METVIEW/lib/-lMagEmos -L/usr/pgi/Linux86/6.0/lib -lpgf90 \
 -lpgf90_rpm1 -lpgf902 -lpgf90rtl -lpgftnrtl -lpgc -lpgf90 -lgd \
 /usr/lib/gcc-lib/i686-pc-Linux-gnu/4.0.0/libf95.a \
 -Wl,-rpath,/usr/pgi/Linux86/6.0/lib/
```

For applications that require both EMOSlib and the magics library one can use:

```
# export MAG_HOME=/usr/local/apps/METVIEW/etc_magics
# export LOCAL_DEFINITION_TEMPLATES=/usr/local/apps/METVIEW/etc_emos/gribex_templates
# Using the Portland compiler:
example.o: example.f
        pgf90 -c   -tp px -Ktrap=fp -i4 -r8  -g exmplo.f

example: example.o
        g++ -o example /usr/pgi/Linux86/6.0/lib/f90main.o example.o  -L/usr/local/apps/METVIEW/lib/ \
-lMagEmos -lMagics -lMagDrivers -lMagics -lMagEmos -L/usr/pgi/Linux86/6.0/lib -lpgf90 \
-lpgf90_rpm1 -lpgf902 -lpgf90rtl -lpgftnrtl -lpgc -lpgf90 -lgd \
-Wl,-rpath,/usr/pgi/Linux86/6.0/lib/

# Using the  g77 gnu compiler
# Do not forget to use IMPLICIT DOUBLE since there is no -r8 option

example.o: example.f
        g77 -c -fugly-logint -g example.f


example: example.o
        g++ -o example /usr/lib/libfrtbegin.a example.o  -L/usr/local/apps/METVIEW/lib/ -lMagEmos \
-lMagics -lMagDrivers -lMagics -lMagEmos -L/usr/pgi/Linux86/6.0/lib -lpgf90 -lpgf90_rpm1 \
-lpgf902 -lpgf90rtl -lpgftnrtl -lpgc -lpgf90 -lgd /usr/lib/libg2c.a  \
-Wl,-rpath,/usr/pgi/Linux86/6.0/lib/

# Usig the g95 free compiler


# Do not forget to use IMPLICIT DOUBLE since there is no -r8 option

example.o: example.f
        g95 -c -i4 -r8  -g example.f

example: example.o
        g++ -o example /usr/lib/gcc-lib/i686-pc-Linux-gnu/4.0.0/libf95.a example.o \
-L/usr/local/apps/METVIEW/lib/ -lMagEmos -lMagics -lMagDrivers -lMagics -lMagEmos \
-L/usr/pgi/Linux86/6.0/lib -lpgf90 -lpgf90_rpm1 -lpgf902 -lpgf90rtl -lpgftnrtl -lpgc \
-lpgf90 -lgd /usr/lib/gcc-lib/i686-pc-Linux-gnu/4.0.0/libf95.a \
-Wl,-rpath,/usr/pgi/Linux86/6.0/lib/
```

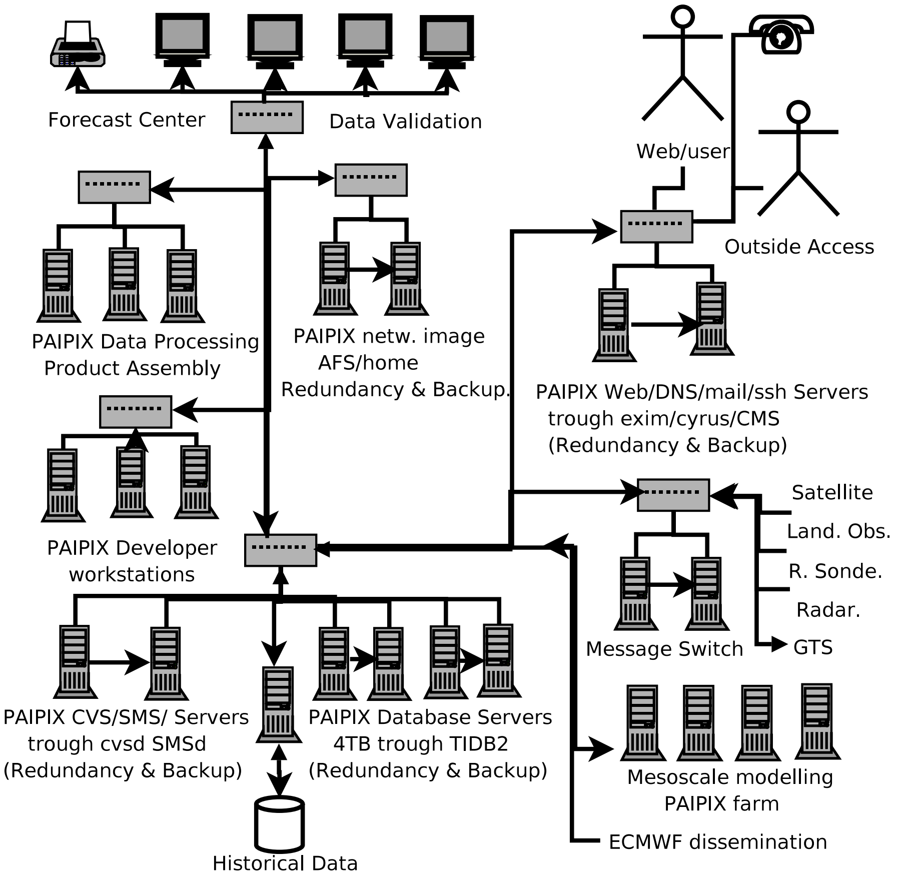


Fig. 2 A simplified view of PAIPIX usage in a possible homogeneous local Meteorological Information System