

Mar 02, 04 22:10

stdin

Page 1/4

INTRODUCTION/OVERVIEW

The Data Monitoring Tool (DMT) software package is a multi-purpose environment developed primarily for real-time, online diagnostics of detector performance and sensitivity. It is also used for carrying out offline analysis directly relevant to astrophysical searches for gravitational waves. It is best known for providing the real-time figures of merit projected on the control room walls during the science runs.

The web pages for the DMT are at

<http://www.ligo.caltech.edu/~jzweizig/dmt/DMTProject/>

ACCESSIBILITY/DEVELOPMENT/CVS/VERSION CONTROL

The DMT is written in C++. It takes full advantage of the object-oriented aspects of C++ with data structures and methods being tightly integrated. Inheritance is also exploited in order to define a common Application Interface for many classes, such as signal processing classes, and to share and reuse a great deal of common code. The heavy reuse of code makes the monitor programs relatively compact and therefore simple to write, debug and verify.

The more esoteric features of C++ are not leveraged in DMT, and experience has shown that monitor developers can be successful with only a rudimentary knowledge of C++ above and beyond C. This is illustrated by the fact that several Surf students (and even high school students) have been able to make significant contributions to the DMT code base during their 10 week stays.

A broad cross section of the LSC membership has written DMT monitors or other DMT-based analysis code including Stefan Ballmer, Tim Bodiya, Shourov Chatterji,

David Chin, Ed Daw, Nathan Hindman, Masahiro Ito, Sergey Klimenko, Szabi Marka, Benoit Mours, Tim Olson, Adrian Ottewill, Steve Penn, Rauha Rahkola, Keith Riles

, Daniel Sigg, Brian Stubbs, Patrick Sutton, Julien Sylvestre, and Natalia Zotov

The DMT source code is maintained in a CVS repository that shares the same server as the LDAS and CDS repositories:

```
:pserver:user@ldas-sw.ligo.caltech.edu:/ldcg_server/common/repository_gds
```

with a web cgi interface at

<http://www.ldas-sw.ligo.caltech.edu/cgi-bin/cvsweb.cgi/gds?cvsroot=GDS>

The web interface makes the CVS archive for DMT open to anyone for read access. A read-only account on the cvs server further facilitates down-loading and installation by anyone in the LSC. Those DMT contributors who prefer to submit their code directly to CVS are also given write access.

John Zweizig acts as the coordinator for DMT development. The DMT source code is managed in cvs by monitor authors updating their own codes and John doing all the version tagging and installation on the online machines. Typically, a deadline for inclusion in a given release is announced about 1 week before a run. Authors with cvs access commit their new code before the deadline. Those without access send modifications to John and he commits them. After making sure that everything compiles and works together, the resulting version is

Mar 02, 04 22:10

stdin

Page 2/4

tagged and down-loaded to the sites. At any given time there are a few revisions installed at the sites so recent versions can be used if need be.

The DMT relies on several open source or LSC written packages including ROOT, FFTW, FrameCPP, ligotools and expat (an xml parser). ROOT is most heavily relied on for graphics, GUI construction and interactive visualization, testing, and development. Outside of the graphics, the majority of the code that goes into a monitor is DMT specific.

ROOT is an object-oriented data analysis framework developed primarily at CERN. The web pages are at

<http://root.cern.ch/>

ROOT has a very high commitment from the HEP community. It is open source and available for many platforms. Bugs are patched within days and feedback is welcomed from anyone. Both Daniel Sigg and John Zweizig have asked for specific features to be added to ROOT and have received quick and positive responses from the ROOT team.

DMT does not depend on any commercial packages.

PLATFORM SUPPORT

The DMT was initially developed under Solaris. It has been tested on Linux with various users having varying success due to library and compiler compatibility issues. Some issues which have been encountered and mostly solved include functions appearing in different header files, functions not appearing when compiling in posix compatibility mode, big-endian/little-endian problems, and simple makefile issues. Those problems that remain seem related to environments at individual sites (the DMT software package is regularly and successfully installed on GDS Linux systems) and should be addressed by the ITR work described below.

The DMT is compiled with gcc, though Solaris CC is used by some developers so the source code is kept compatible with both compilers. DMT source code apart from FrameCPP will compile on gcc 2.95.3 or later. Online installations are compiled under gcc 3.3.0, and backward compatibility to gcc 2.95 has been tested since moving to more recent compilers.

LEVERAGING EXISTING CODE BASES

A DMT program or monitor can link to LAL or other C libraries easily since C is a subset of C++, and this has already been demonstrated a number of times. Taking existing routines in the DMT and moving them back into LAL, however, would be very difficult and time intensive because features of C++ such as inheritance, automatic initialization and destruction, and the standard template library cannot be translated to C without considerably more code.

Online access of data via the DMT is extremely straightforward. One merely starts the program and the data streams into the monitor via the local shared memory partition. Offline data access uses the same mechanism, but reads data from a list of frame files rather than the shared memory. The list is just a flat list with one path

Mar 02, 04 22:10

stdin

Page 3/4

per line, and limited globbing is allowed. It would be relatively trivial to change the format should a LSC standard become available.

DOCUMENTATION

The documentation of DMT infrastructure and monitors is uneven. There is ample infrastructure for documentation via doc++, automated links for online monitors, automated links for online alarms, an online troubleshooting guide, some web overview material, and a DMT home page with descriptions of code infrastructure and templates. Unfortunately, not all DMT monitor authors have provided adequate documentation. In addition, one missing document is a "Getting Started" manual for first-time monitor writers, one that would provide a standard template. As it is, most authors simply copy an existing monitor program and modify it for their needs. No FTE effort is identified at this time to write a Getting Started document.

BUG REPORTING

Bug reporting for the DMT infrastructure and monitors is handled via GNATS:

<https://ldas-sw.ligo.caltech.edu/cgi-bin/gnatsweb.cgi?database=gds>

There is an e-mail distribution list for discussion of DMT issues:

<https://tenaya.physics.lsa.umich.edu/mailman/listinfo/dmt-discuss>

VALIDATION

The DMT infrastructure and its monitoring programs have been subjected to rigorous high-level performance evaluation from the start. The low-level code has been tested and is literally in constant use (~40 monitor processes have been running online for several years). Some of the code (e.g. the filter design classes) are vital for the operation of the LIGO interferometers, and many other pieces of the DMT code are used daily for IFO commissioning and operations. Much of the low level code common to many monitors has been validated to the extent that many monitors have been using the common code for many years. Some of the filtering code is also used directly in the interferometer controls.

Nonetheless, more validation would be desirable. A new set of simulation and evaluation tools exists now that should facilitate systematic validation and performance evaluation. Information on these tools and example outputs can be found at

<http://www.ligo.caltech.edu/~jzweizig/dmt/DMTProject/DMT-DataSimulation.html>

<http://www.ligo.caltech.edu/~jzweizig/dmt/DMTProject/MatchTrig1.pdf>

FUTURE

Recently the National Science Foundation (NSF) awarded a consortium of LSC institutes (PI: Eric Katsavounidis of MIT) an Information

Mar 02, 04 22:10

stdin

Page 4/4

Technology Research (ITR) grant to support advanced LSC data analysis work using GRID technologies. Part of that grant is dedicated to DMT support, with DMT FTE's allocated to Penn State, MIT, and UWM. MIT has just hired a programmer / physicist to spend part time on "gridifying" the DMT, to allow for much better transparency in data access and efficient remote submission and control of jobs. Penn State has an advertisement out for a programmer / physicist to work part time on improved porting of the DMT to unix and linux platforms. It is envisioned that once this work is complete, Penn State would provide a server for automated nightly updates of DMT code via RPM's, making the DMT more widely available in the collaboration with better uniformity.

SUMMARY

The DMT offers an object-oriented analysis environment that can be used online or offline. A large base of DMT authors, many of them novices at C++ programming, have shown that only modest effort is required to write useful detector diagnostic code in the DMT environment. Substantial ROOT-based graphical support, in-house signal processing tools, and standardized classes for manipulating data, spectra, histograms, etc., make the DMT attractive for rapid, high-level analysis code development. The ability to incorporate C code into DMT programs allows building upon special-purpose libraries, such as LAL and FFTW. In addition, much of the code that has been developed for online detector diagnostics and interferometer signal conditioning is automatically available to analysts working in the DMT environment. New infrastructure has been added recently to standardize performance evaluation of individual monitors, making it easier to vet code at low and high levels.

There are issues, however, that need to be addressed. Monitor documentation is uneven and a comprehensive "Getting Started" manual would attract new users. DMT code (developed under Solaris) has been ported to linux and is routinely maintained for linux machines at the observatory sites, but users of other flavors of linux and compilers have often run into a variety of installation problems. Broader support for linux, including a facility for automated updates, is highly desirable. In addition, the present DMT process control infrastructure addresses only online needs, leaving offline parallel processing somewhat cumbersome. These issues will be addressed as part of preparing the DMT for grid computing. Fortunately, a recent ITR grant provides explicit funding for gridifying the DMT and strengthening its portability.