

Mar 02, 04 22:27

stdin

Page 1/6

Discussion of the Data Monitoring Tool Role in Data Analysis

(K. Riles and J. Zweizig)

The Data Monitoring Tool (DMT) software package (C++) is a multi-purpose environment developed primarily for real-time, online diagnostics of detector performance and sensitivity, but which can also be and is used for carrying out analysis directly relevant to astrophysical searches for gravitational waves.

The DMT is best known for monitoring programs like SenseMonitor, BLRMS and DataQual which run continuously in background on control room computers, monitoring the data, writing html summary pages, displaying graphics via the DMT viewer, generating alarms, database triggers, and writing minute trend files to disk. Each monitor has the same structure; a driver program calls a user-defined routine at fixed, user-defined time intervals. That routine carries out its computations and can call on a large variety of infrastructure classes that support signal processing, graphics, triggers, I/O to standard format files, etc. Because the language is object-oriented C++, interface and data format issues are typically trivial, leading to high transparency and little wasted coding time.

The DMT by design encourages the use of standard formats by providing I/O classes for standard formats (e.g. frame, trend frame and xml I/O plus html writing). Other non-standard formats may be used as needed. Ease of use has been paramount in its design from the beginning.

While the DMT monitors provide valuable real-time feedback to the control room operators and scientists, many also directly affect offline astrophysical analysis. The triggers they write to the database can be queried to understand odd behavior in the gravitational wave data channel (ASQ), but more important, the monitors can be rerun offline with varying thresholds and other control parameters to study and define data analysis vetoes. In addition, most of the data quality flags used in analysis to reject "bad" data are derived from these monitors. The DMT is already a critical part of our astrophysical analysis.

In addition, there has been some interest in carrying out searches directly in the DMT. One obvious benefit is improved online detector diagnostics. If the rate of neutron star inspiral candidates suddenly jumps by a factor of ten, then there is presumably a detector malfunction that should not be ignored. As time goes on, we expect to implement more and more such direct searches in the online system to improve real-time diagnostics. This plan naturally raises the issue of whether full-fledged astrophysical searches should be carried out in the DMT environment, aiming at publishable results. Presumably such analysis would be carried out primarily offline, using a large cluster of computer nodes. Our strong feeling is that such analysis is viable in the long term and should not be artificially prohibited by the LSC.

Why do we feel this way?

- 1) The DMT software builds upon the CERN ROOT library, which means it is easy to generate graphs and histograms directly in one's code. There is tremendous graphical user interface (GUI) support, and the DMT is fully integrated with other Global Diagnostics System (GDS) software, such as the familiar Diagnostic Test Tool and the DMT Viewer. Because of the ROOT basis, one can also run the DMT in interactive, interpreter mode for quick exploratory

Mar 02, 04 22:27

stdin

Page 2/6

studies.

- 2) Given the distributed nature of LSC analysis and the desire for common tools, an object-oriented language, and C++ in particular, has strong advantages. C++ is strongly typed and makes well defined interfaces the norm rather than an ideal that is attained only with difficulty. The encapsulation of data and methods also makes debugging and verification much easier. The programmer can devote a larger fraction of time to algorithm development and spend less time worrying about the "protocol" of the programming language and the details of how a routine written by another author stores and manipulates data.

At the start of DMT development, we were concerned that potential contributors would resist a change to C++. We therefore made a parallel C language development path available for these users. In fact we found that our contributors overwhelmingly chose to use the more extensive capabilities available through C++.

- 3) Because the C language is a subset of C++, one may link directly to C libraries, such as the LSC Analysis Library (LAL) and use desired routines, as needed. It seems highly likely that as we develop online astrophysical search programs for detector diagnostics, some of the routines already present in LAL will be exploited. Working in the DMT in no way precludes using routines in LAL, and if sufficient demand arises, LAL "wrappers" can be written to make LAL routines more friendly to DMT users.

There was discussion some years ago of "bundling" DMT and LAL together, but that idea was rejected since the online DMT code is mission critical, and a sudden failure of DMT compilation because of changes in LAL was considered unacceptable. But the lack of deliberate bundling into a common software framework does not prevent individual analysts from creating stand-alone DMT programs that use LAL infrastructure, just as they might call a routine from CLAPACK. This is already standard practice; for example, USGS seismograph code (written in C) was imported for use in the PTmon DMT monitor.

- 4) 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.

While there has been less emphasis on formal validation of the monitor program functionality than in some other LSC software for which realistic test vectors can be supplied, we believe the DMT codes have had considerable and more demanding vetting of performance evaluation on real, non-stationary, non-Gaussian interferometer data. For example, because the online DMT monitors must routinely process data in high-noise states and even when the interferometer is unlocked, standard tools have been developed to determine interferometer state and exclude poor data "on the fly".

Mar 02, 04 22:27

stdin

Page 3/6

Nonetheless, there is obvious virtue in low-level vetting of routines using artificially generated data. One may catch bugs that affect only small fractions of the data or appear as normalization errors and hence may have escaped notice, despite the high-level vetting in engineering and science data runs. Until recently, monitor authors and detector characterizers have had to evaluate the performance of a monitor based on its response to live (i.e. uncharacterized) data. A new set of simulation and evaluation tools exists now, however, that allows rapid and quantitative performance evaluation

of monitors as function of their own control parameters. Histograms and efficiency curves are now easy to produce in standardized form. We believe these tools will allow the finding of residual bugs in existing monitors and, more important, allow those monitors to be better tuned for performance.

It should be pointed out that debugging and validation/verification are greatly helped in the DMT environment by

- * Interactive development and testing "in the same language"
- * Support of histogramming and direct data visualization, vital to effective code development
- * Standardized low-level classes common to many monitoring programs
- * The use of base classes provides stable, uniform and well defined interfaces to shared code and makes rewriting or copying of basic code unnecessary.

Other issues:

- 1) The DMT and other GDS code is stored in CVS archives with periodic tags given to new releases, particularly those associated with engineering and science data runs.
- 2) 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 (or any!) documentation. In addition, one sorely 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.
- 3) The DMT and other GDS code was initially developed under Solaris, but is available and has been tested on linux. Various users have had varying success in porting to linux because of library/compiler compatibility issues. More work is needed before it can be claimed that the DMT is truly portable. See below under "ITR Grant".
- 4) 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. However, because the infrastructure for running DMT jobs on a cluster has not been developed yet, the process of dividing the task among several processes,

Mar 02, 04 22:27

stdin

Page 4/6

locating input data, verifying that a process has run, and collating the results can be cumbersome. This is being addressed as described below under "ITR Grant".

5) 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>

ITR Grant:

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

In addition, we are requesting from LIGO Laboratory the use of Beowulf cluster nodes at the observatory sites for running DMT jobs offline on archived data. We have in mind that with the "gridification" of the DMT, these jobs would be treated on the same footing as LAL-based jobs and would not add appreciably to the load on site personnel already managing the clusters.

Summary:

The DMT has proven to be an extremely useful software environment for online detector diagnostics and offline detector characterization affecting astrophysical data analysis. We believe it has considerable promise for an expanded role in astrophysical data analysis, thanks to its object-oriented structure, ease of use, and substantial graphical and histogramming support. It is the LSC software package that most resembles the successful analysis environments of large high energy physics experiments.

It is not that we advocate analysts now happily using LAL should adopt the DMT instead. As discussed above, there are still technical impediments to seamless offline analysis of months of data, and the installation of the DMT on non-Solaris platforms can take more time than it should. We expect that these impediments will be removed by the end of 2004. We do suggest, though, that analysts take a closer look at the DMT. Working in LAL for data manipulation and in Matlab for visualization is not necessarily a better choice than using one software package that supports both of those functions and that can be used both online and offline. Moreover, the ability to link directly to the LAL library means one need not sacrifice any existing analysis functionality.

Finally, this memo has been provoked not by a sudden desire to proselytize LAL

Mar 02, 04 22:27

stdin

Page 5/6

users into converting to the DMT way, but rather by what we consider a badly misguided recommendation by an ad hoc committee that the DMT be prohibited from use in astrophysics analysis. To those of us familiar with the DMT and the real-world vetting it has been subjected to, it seems foolish to preclude its use in astrophysical analysis, merely to ensure uniformity of tools within the collaboration. Experience has shown, we believe, that "market forces" should prevail. A software environment that is more trouble to use than it's worth will fade into disuse, and environments that prove useful will grow in use. There need not be only one such useful environment the LSC. We have already seen the perils of prematurely adopting a software environment as the one and only standard. Let us not repeat past mistakes.

Appendix:

The DASWG requested specific information on four issues. Although much of that information is contained in the above, we address these issues explicitly below and provide additional web pointers.

1) Accessibility/cost of DMT/GEO++ to entire collaboration.

The DMT belongs to the LSC and is freely available for cvs downloads. It uses the CERN ROOT library, which is also freely available.

For more information on the DMT, please see

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

For more information on ROOT, please see

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

2) Platform and other support of DMT/GEO++.

As discussed above, DMT is actively supported on Solaris and linux. DMT code has been installed and tested on both platforms and any problems found on either platform are addressed quickly by the authors. The linux support lags that for Solaris because of the demands of online use, but this situation should improve as work for the ITR grant ramps up.

3) Possibility to hook libraries (e.g. LAL, etc) into DMT/GEO++ to leverage other existing code bases.

As discussed above, we know of no reason an individual DMT program cannot link to LAL and use any desired routines. Other C libraries have been linked successfully to DMT programs. There is good reason, however, not to bundle DMT and LAL together unless individual LAL interfaces are stabilized and the release schedule for LAL is coordinated with DMT.

4) How version control, documentation and validation should be handled.

* The DMT is under CVS control. There have been a number of tagged releases over the last several years, usually associated with science and engineering runs.

* Documentation of the infrastructure and individual monitors is handled at

Mar 02, 04 22:27

stdin

Page 6/6

the coding level by doc++. In addition, there are separate documents (pdf and/or html) describing the infrastructure and monitors, but as discussed above, the degree and quality of documentation varies widely among the online monitors.

Overview documentation of the DMT can be found at

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

Documentation of online monitors can be found at

<http://blue.ligo-wa.caltech.edu/gds/dmt/Monitors/MonTable.html>

* As discussed above, validation of online DMT monitors has been extensive at a high level because of the demanding nature of data runs. 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>

It should be pointed out that debugging and validation/verification are greatly helped in the DMT environment by

- * Interactive development and testing "in the same language"
- * Support of histogramming and direct data visualization, vital to effective code development
- * Comprehensive independent class libraries for I/O, Data manipulation, Signal processing, etc. available to all monitoring programs.
- * The use of base classes provides stable, uniform and well defined interfaces to shared code and makes rewriting or copying of basic code unnecessary.