The sub-committee was asked to address the question of what should happen to LDAS development after the release of LDAS 1.0 (which happened on 2/14/04), and to consider the "charges" listed below. The sub-committee exchanged emails, met once on Feb 27, and reviewed a document prepared by Tom Nash (at the request of Albert Lazzarini) on "LIGO Computing Environments and Analysis Quality Assurance", LIGO-L030155.

Charge 1) Accessibility of LDAS to the LSC. Please consider, among other things, ease of use.

The committee interprets this charge to mean that we should gather information on who uses LDAS, what they use it for, whether they find it easy to use, what their plans are for future use, what features of LDAS they find to be essential, and what additional features they would like (or need) to see.

LDAS logs all users' requests, and the CMON utility provides reports on LDAS use. It reports that several dozen members of the LSC have used LDAS in the last few months, and over 5 million jobs have been submitted to the production clusters since S1. LDAS is used for many different tasks:

- Short data or metadata queries using GUILD or LIGOTools.
- Metadata database insertion or extraction.
- Acquiring frame data, delivery as frame files or LIGO LW data files.
- Conditioning frame data prior to delivery.
- Generation of reduced datasets via createRDS for LSC-wide use.
- Creation of custom RDS's for specific uses.
- Providing location of online data for online processing.
- Analyzing data with LAL-based GW search software via LALWrapper DSOs.
- Generate custom data products necessary for the continuing development of DSOs, including new DSO search codes.

In the context of the DASWG, information on the last two task categories may be of greatest interest, although we do not mean to underestimate the importance of the other task categories, such as the createRDS facility or the use of the database by online DetChar software. The subcommittee is aware of significant use of LDAS by the Burst group (Tfclusters, Waveburst DSOs) as well as some evidence of significant use by members of the Stochastic, Pulsar, and Inspiral groups. The subcommittee decided that the best approach to addressing this charge would be to distribute a short questionnaire to the entire LSC, and compiling the results. This questionnaire is appended to this report, and we entreat all LIGO data analysts to respond to it ASAP.
Charge 2) Platform and other support of LDAS.

Charge 3) Personnel requirements if LDAS goes into maintenance mode with the release of LDAS 1.0. Please compare to requirements for development of LDAS 1.0 or continued development of all components of LDAS.

These charges were combined because much of the maintenance requirements for LDAS depend on which platforms are supported. Specifically, several major proposed development tasks for the LDAS team involve extending support for different compilers and platform architectures. LDAS is built on top of a very large collection of third-party software packages collectively referred to as LDCG. As directed by LIGO Lab and the LSC, both LDCG and LDAS software proper are currently supported on Sun Solaris 9 and Intel RedHat Linux 9; on both platforms, the GCC 3.3.1 compiler is used.

Following the release of LDAS v. 1.0.0, Kent has compiled a long list of potential future development projects. We summarize them here without any supporting details; just ask. All estimates of time required to implement these goals are very approximate.

- Port to the Solaris C++ compiler - 2-4 months for one programmer.
- Port to the Intel C++ compiler - 1-2 months for one programmer.
- Implement fully federated LDAS database - 4-6 months for 1.5 people.
- TCL/Globus software development - 4-6 months of a new hire to begin real work on this long-term project.
- Maintenance of underlying tool sets (LDCG) - 1-2 FTE's, ongoing.
- Further development of data conditioning tools - 1-2 FTE's, ongoing.
- Maintenance of LDAS proper, including modernizing socket communication technology, and more - at least 2 experienced FTE's, ongoing.
- Port to 64bit INTEL or AMD CPUs - 1-2 FTE's for 6 months.
- RPM based distributions of LDAS - 1 FTE for 6-9 months.
- Multi-FrameAPI topology - group effort, 6-9 months.

The lab-based LDAS team maintains 6 clusters (LDAS-WA, LA, CIT, TEST, DEV, MIT) which run LDAS. There are similar linux-based beowulf clusters maintained at and by UWM, PSU, AEI, and UTB, although some of these may not be currently running LDAS (LDAS was designed as a software environment primarily for the lab-based clusters). The linux-based beowulf clusters can be configured to run condor; currently, only LDAS-CIT amongst the lab-based clusters is so configured. However, the front-end servers that LDAS APIs run on are specialized hardware (Sun Servers, Multi-CPU boxes that are not supported or practical for a condor environment. They would most likely have little role outside of LDAS.

It should be noted that LDAS software development is tested using a very mature set of test suites that include nightly builds, nightly testing of LDAS against every single change in the code, a problem tracking system, etc. This type of QA infrastructure is in general (and was for LDAS) expensive to establish.
Charge 4) Possibility to extract pieces of LDAS 1.0 as components of a revised on-line analysis system.

LDAS currently provides certain services that are central to near-real-time data processing:

- Data discovery – LADAS maintains a table of locations of frame files produced by the online data acquisition system, and is made available for use by any job that inputs frame data, both near-real-time and off-line. This information is used by createRDS, and by near-real-time LDAS/LAL-based GW search software.
- The LDAS-based createRDS facility runs near-real-time, creating RDSs for distribution to other computer clusters for use by LSC data analysts.
- In the first three science runs, LADAS/LAL-based GW search software was run on the site LDAS clusters near-real-time, in an attempt to provide useful feedback to operators and scimons. There is no consensus that these efforts were particularly useful.

It is certainly possible to extract pieces of LDAS code. LDAS software is organized as stand-alone c++ libraries (such as the frame or datacond libraries), and code in those libraries can be called from TCL scripts; this is how most of LDAS is implemented. This code is available for use in other applications. Of course, revised systems for replacing the role of LDAS for the first two services listed above would require considerable effort.

With regard to the datacond library: this is also a stand-alone c++ library that can be used in other applications. It should be noted that LSC data analysts make use of at least four different extensive software libraries for data conditioning (signal processing): MATLAB, DMT, LAL, and LDAS/datacond. This is a lamentable duplication of effort, creates confusion, and makes validation of all software more laborious.
LDAS Questionnaire, March 2004

The LSC Data Analysis Software Working Group (DASWG) sub-committee on LDAS requests your input: do you use LDAS, what do you use it for, do you find it easy to use, what are your plans are for future use, what features of LDAS do you find to be essential, and what additional features would you like (or need) to see.
This is a short questionnaire; please respond in the next few days. Many thanks!

Q1: Do you use LDAS for any/all of the functions (Q1.1-6) listed below?
   [ ] YES
   [ ] NO

For each function (Q1.1-6), please specify your level of "Usage":
0=not at all, 1=light, 2=average, 3=heavy

For each function (Q1.1-6) that you use, please give a "rating":
0=don't know, 1=useful/functional, 2=needs work, 3=difficult to use

Q1.1 Short data or metadata queries using GUILD or LIGOTools
Q1.2 Metadata database insertion or extraction
Q1.3 Acquiring frame data, delivery as frame files or LIGO LWfiles
Q1.4 Conditioning frame data prior to delivery
Q1.5 Creation of custom RDS's for specific uses
Q1.6 Analyzing data with LAL-based GW search software via LALWrapper DSOs
Q1.7 Which LAL-based GW search DSO(s) do you use under LDAS

Q2.1 Do you envision continuing the use of LDAS through the next 12 month?
   [ ] Not at all, [ ] Lightly, [ ] Heavily, [ ] I'm phasing out

Q2.2 If you are moving to other tools/platforms, which are you considering:
   [ ] I'm sticking with LDAS, [ ] Condor, [ ] MATLAB, [ ] DMT,
   [ ] Other(specify)
What is your primary reason for considering another platform? [.........]

Q3: What is your opinion of the ease of use of LDAS, apart from issues of ease of use of LAL software?
   0=don't know, 1=useful/functional, 2=needs work, 3=difficult to use
   Please add specific comments.

Q4.1 What features of LDAS do you find to be essential?

Q4.2 What additional features would you like (or need) to see?