Data Analysis and Science from S1 and S2

Overview of Data Analysis Systems and Simulations

(*LIGO Data Analysis System Software)*

14th Meeting of the LIGO Laboratory PAC

June 4th - 5th, 2003

California Institute of Technology

Kent Blackburn

LIGO Laboratory at Caltech

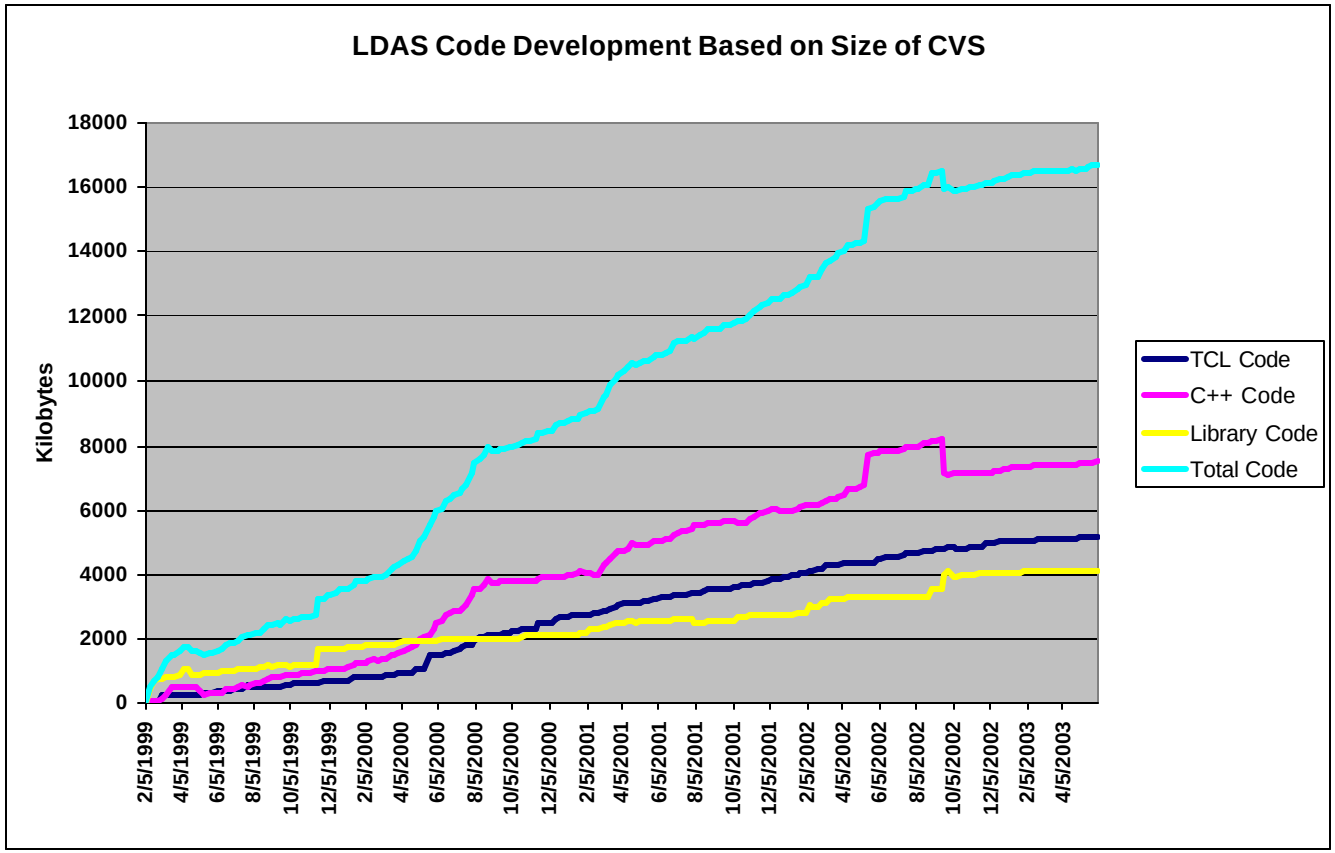
LIGO Data Analysis Software

Status Update

Software Status

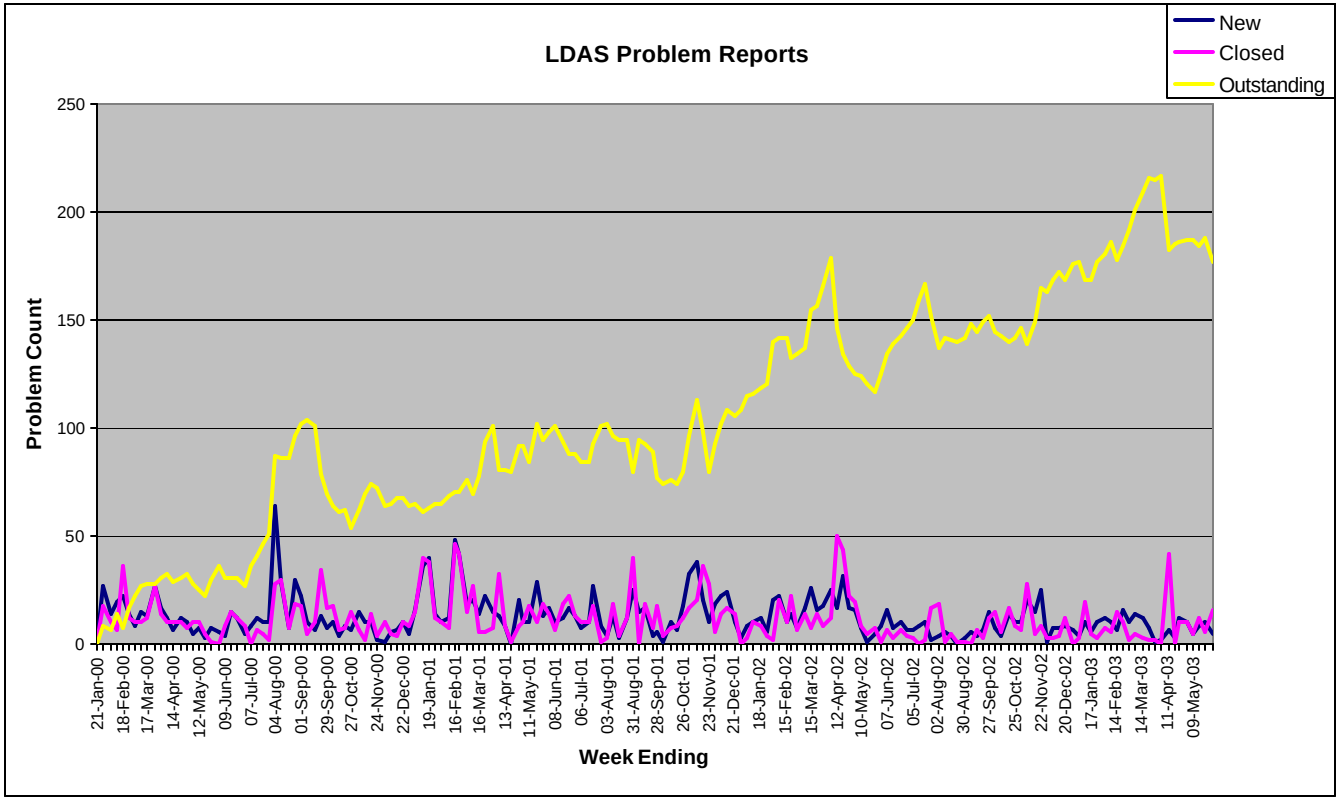
- First new release since S2 Run this week (0.7.0).
- Resolves major issues seen in version used in the S2 Run.
- Will support *merged* LLO/LHO Reduced Data Set Frame generation.
- Nearly all (~95%) functional requirements in place.
  - Usage continues to introduce new ideas requiring new functionality.
  - Grid support and interfaced are an *emerging* arena for new requirements and code development.
    - *Anticipate hiring Grid software developer this summer.*
- User support, platform ports, and performance improvements now dominate level of effort going into development.
- Expect 0.8.0 release prior to S3 with 1.0.0 release early next year.

Code Development Trends



New code development starting to show signs of tapering off.

Problem Report Trends



Steady stream of new problem reports as software usage grows.
91% of the 2000+ problem reports filed on LDAS have been closed!

Remaining Development

- **FrameAPI**: missing support for several compression modes from spec; better support for other IFO Frames, *especially w.r.t. to RDS generation.*
- **metaDataAPI**: not yet multithreaded; need to update ODBC interface for IBM's DB2 version 8.1; significantly improve performance, *support database federation!*
- **diskCacheAPI**: needs threaded C++ new frame file discovery layer.
- **dataConditionAPI**: new signal conditioning and fundamental algorithms constantly being requested as Upper Limits Groups refine analysis methods.
- **controlMonitorAPI**: log file parsers require excessive time to process.
- **dataIngestionAPI**: currently not an internal LDAS API; this may need to change as we migrate to a more unified data archival and grid enable system.
- **All APIs**: boost performance; close out problem reports; migrate to new compilers and new operating systems in a timely manor to avoid software dead-ends – *with code base is so large, this is significant development activity for us!*
- **Grid Technologies**: continuing to develop new interfaces based on GLOBUS for Grid security (certificates), grid utilization, data product publication and virtual data cataloging.

LIGO Data Analysis Software

GriPhyN Collaboration Update

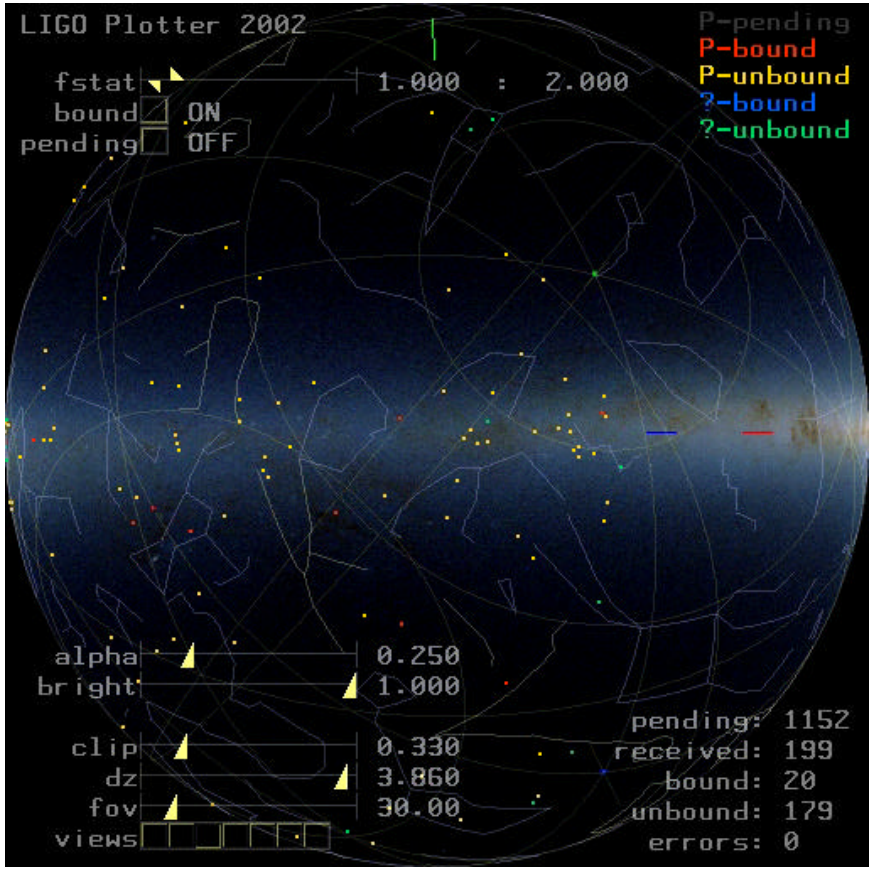
Grid Related Activities

- Began issuing Department of Energy Certificates to provide Grid Security protocols for specific activities (such as data replication).
- Integrated Globus (GridFTP) support into all Laboratory LDAS systems.
- Used Grid Technologies (LDR –lightweight data replicator) to replicate data from Caltech to Tier II centers.
- Added two new user commands (*dataStandAlone* and *putStandAlone*) to allow running the *remoteAPI* on Grid resources and then putting results back into LDAS databases and frame storage.
  - This new technology used for a limited “directed pulsar search” on an hour of S1 data at the Super Computing 2003 Conference…demo!

SC2003 Conference Demo

(LIGO Pulsar Search)

- Combined LDAS technology with Grid technology.
  - Submission interface and all compute/data management used Globus GSI (Grid Security Infrastructure) with DOE X509 certificates.
- Collaboration on project include LDAS team here at Caltech, UWM and USC-ISI.
- Targeted over 1300 know pulsars.
  - 200 targets search during allotted demo period at conference.
- Complete search required 100 hours of CPU time to complete.
- Results stored in LIGO database.



LIGO Data Analysis Software

LDAS Overview – S2 Run

S2 LDAS Overview

- All sites started out with LDAS release 0.6.0
  - Two interim (patched) version 0.6.20 & 0.6.61 pushed to LHO and LLO to support run.
- Pre-release testing revealed about a 0.2% to 2% failure rate
- During S2 Run 2.1% of jobs failed at LHO & 3.9% failed at LLO.
  - Approximately 80% of failures due to user errors.
  - Total job failure rate about twice as high as seen in S1 Run.
  - Averaged about one job every 6 seconds - *slightly less than in S1.*
  - Averaged about 13 rows inserted into the databases each second - *2.6 times higher than in S1*.
- CIT archive (SAM/QFS) came online after start of Run/Release.
  - diskCacheAPI patched as we learned "how to interface" on LDAS-CIT but mostly worked.
- RDS frames generated at site but not shipped – CIT responsible for generation of RDS frames shipped to Tier II centers.
  - LDAS software developer used to support this task to generate RDS frames at CIT.
  - LDAS unable to generate RDS frames in real time at LHO; not a problem at LLO.
  - RDS generation at CIT started out slow but greatly improved as new tape drives came on line!

Job Submission Summary

Science Run II	*02/14/03 10:00 AM PST – 4/14/03 10:00AM PST*															
		LHO				LLO				CIT				MIT		
User Command	**Submitted**	**Passed**	**Failed**	**Percent**	**Submitted**	**Passed**	**Failed**	**Percent**	**Submitted**	**Passes**	**Failed**	**Percent**	**Submitted**	**Passed**	**Failed**	**Percent**
CreateRDS	47868	46736	1132	2.36%	29348	24563	4785	16.30%	7441	4395	3046	40.94%	1407	1270	137	9.74%
DataPipeline	101901	94578	7323	7.19%	44725	40670	4055	9.07%	33145	30478	2667	8.05%	108890	102288	6602	6.06%
Inspiral	12301	6958	6958	43.44%	3042	1774	1268	41.68%	29	10	19	65.52%	9	0	9	100.00%
Power	22707	21991	716	3.15%	6607	6493	114	1.73%	755	745	10	1.32%	18381	18120	261	1.42%
Cohere	0	0	0	0.00%	0	0	0	0.00%	4837	4750	87	1.80%	5225	4645	580	11.10%
Slope	18868	18211	657	3.48%	5203	5033	170	3.27%	763	746	17	2.23%	23300	22760	540	2.32%
Tfcluster	9408	9063	345	3.67%	3013	2492	521	17.29%	810	752	58	7.16%	26971	22739	4232	15.69%
Stochastic	0	0	0	0.00%	0	0	0	0.00%	14006	12564	1442	10.30%	11	2	9	81.82%
Waveburst	35811	35702	109	0.30%	26846	24875	1971	7.34%	3646	2725	921	25.26%	34981	34020	961	2.75%
KnownPulsar	2802	2653	149	5.32%	0	0	0	0.00%	13	13	0	0.00%	0	0	0	0.00%
GetMetaData	135708	135650	58	0.04%	82935	82922	13	0.02%	1501	1493	8	0.53%	64337	64232	105	0.16%
PutMetaData	127034	127018	16	0.01%	70508	70471	37	0.05%	5	3	2	40.00%	1	0	1	100.00%
GetFrameData	58	29	29	50.00%	1	1	0	0.00%	17	11	6	35.29%	3	3	0	0.00%
GetChannels	31	20	11	35.48%	52	30	22	42.31%	85	53	32	37.65%	24	17	7	29.17%
ConcatFrame	43	29	14	32.56%	17	9	8	47.06%	0	0	0	0.00%	0	0	0	0.00%
ConditionData	69	51	18	26.09%	48	44	4	8.33%	20	5	15	75.00%	46	26	20	43.48%
All Jobs	**417870**	**409274**	**8596**	**2.06%**	**228136**	**219206**	**8930**	**3.91%**	**42881**	**37099**	**5782**	**13.48%**	**174999**	**168125**	**6874**	**3.93%**

Database	LHO	LLO	CIT	MIT	Total
Rows Inserted	24378362	15215169	1995041	26205456	67794028
Rows Queried	1357913	767033	411761	16967773	19504480
S1 DB Size (GB)	0.95	0.85	0.07	5.1	6.97
S2 DB Size (GB)	5.8	5	0.01	0.04	10.85

LDAS S2 Performance

- LDAS versions 0.6.x were all based on GCC 3.2.1 and GCC 3.2.2.
  - Fixes thread safety issues that plagued earlier LDAS versions based on GCC 2.95.x – *also removed several memory leaks as a bonus.*
    - In particular the three year struggle to stabilize the dataConditionAPI!
  - Unfortunately, GCC 3.2.x has known issues with optimization under Sun Solaris: no optimization beyond ***"–O2 –no-inline"*** possible!
    - Slowed LDAS down by factor of two!
- Overall, the rate that jobs went through was about the same and rows per second inserted into database increased slightly in S2 over S1.
  - Mostly due to LDAS being used throughout the S2 run in contrast to it being used only during the second half of S1 run.
- RDS frame generation probably *hit the hardest* by lose of performance.

Response to S2 Run Experiences

- Reliability of LDAS and its components had to be significantly improved during the S2 Run.
  - Didn't have realistic examples of the types of jobs that would be submitted to the LDAS systems prior to start of S2 Run.
  - LDAS was "vulnerable" to ill-behaved LALwrapper codes impacting system level functionality – better recognition and isolation needed.
  - Integrated testing directly into controlMonitorAPI's GUI for 0.7.0!
- In the two most severe issues, making LDAS more reliable took on order a month once the problem was identified and isolated through reproducible tests.
- Need to improve performance of LDAS as soon as compiler technologies allow.
  - Frame I/O seen to be bottleneck for 25% by volume RDS generation.
  - Large number of candidate events causing occasional pile-up at database.

LIGO Data Analysis Software

The End