LIGO Data Analysis System
Preliminary Design Review (PDR)

0830 - 1230 PST **11 March 1999**

Agenda

- Review of DRR action items/issues
- Preliminary Design
  - Software
  - Hardware
  - Archive
  - Interfaces
  - Networks
- Development strategy / schedule
- Budget

LIGO Data Analysis System (LDAS)
Review of DRR Report Issues

- Consider using a web-based browser GUI vs. an X11 GUI
  - ✓ Being implemented -- Tclets and Tcl plugins
- Priorities & schedules need better definition
  - ✓ Discussed below
- Data distribution server should inform user of “cost of service” being request
  - ✓ Can be implemented at any time - will be a managerAPI function
- Layered SW design does not identify different components as “production” or “optional”
  - ✓ No distinction exists in software -- all layers shown (except specific filters/wrappers) are production.
- Need to specify data types produced by LDAS
  - ✓ - Issue has been addressed extensively; still some TBDs, however -- discussed below
- LDAS-DAQS interface needs better definition
  - ✓ Interface definition has been completed -- discussed below
- Tape playback at sites needed
  - Requirement noted; will be implemented at additional cost beyond originally scoped budget.
- Data flow/storage hierarchy needs further definition
  - Still being addressed -- we still have the volume-of-data issue
  - LSC has since formed three working groups to deal with such issues.

LIGO Data Analysis System (LDAS)
Review of DRR Report Issues

- LDAS support for diagnostics (GDS) needs to be addressed
  - ✓ Discussed below
- LDAS component redundancy needs to be addressed
  - ✓ Being addressed by extensible, network-based model for LDAS architecture.
- 99+% of data not science[astrophysical] data - determine what is needed
  - Issue being worked on; GDS has down-scoped acquisition rates; everyone recognizes desirability of data summaries for ancillary channels; how issue is resolved will likely have to wait for experience base to build with actual working IFOs.
- On-line system goals not clearly defined
  - Needs LSC inputs; will try again at this review....
- $h[t]_{calibrated}$ (read: w[f]*R[f]*h[f]) needs to be on spinning media for ease of access
  - Working with CDS, have identified need for full, analysis and trend frames. Analysis frames are reduced-volume objects corresponding to h[t]. Moreover, single-channel h[t] can also be kept if deemed reasonable. Note however, that h[t] for 1 year from 3 IFOs corresponds to ~ 3TB/year at 16384 samples/s. Sampling may be reduced to ~ 2048 without loss of astrophysical value of data since shot noise starts to dominate beyond ~ 2kHz. at 2048 sample/s, volume = 360 GB per year, which is tractable.
- APIs not adequately defined
  - ✓ Work since DRR has focused on this definition (see genericAPI specification); however, it should be noted that the conceptual design document actually provided clear definition of functionalities.

LIGO Data Analysis System (LDAS) Review of DRR Report Issues

- Media definition not clearly stated; also differences between archive and sites could cause incompatibilities
  - ✓ LDAS and CDS have redefined the media interface; CDS provides the disk cache (on-line) and LDAS provides backup media/hardware so that it can be compatible with archive.
- Need central librarian, server to track metadata
  - A commercial DBMS will be implemented to handle metadata; some TBDs remain, however.
- Use of hierarchical binary inspiral schema could reduce CPU requirements for template-based searches
  - ✓ True. However, no validated, accepted algorithm presently available. Needs to come from LSC.
  - ✓ Reconfigurable hardware can be used to implement such a search schema when it is available; design anticipates such changes/improvements.
  - ✓ Beowulf compute cluster is presently affordable to do the flat version of the job.
- Need to provide user-driven archival resources
  - There will be some (TBD) disk cache available for user archiving; however large (what is “large is TBD) volumes of data will need to be downloaded or requested via FedEx for local archival on user’s local resources at his/her institution.
- Data reduction requirements not well defined
  - Observation is true. Subject of a previous issue (see above)
  - We plan to use lossless compression as much as possible
  - Debate unresolved on whether/what to dismiss permanently
  - LSC working groups involved in this issue.

LIGO Data Analysis System (LDAS)
Review of DRR Report Issues

- CACR role within LIGO needs definition
  - ✓ Since DRR a formal MOU has been written and signed
  - LSC has some issues with MOU; this was addressed at LIGO-LSC meeting on 26 February 1999.
  - MOU is being revised to reflect long term archival needs by LIGO.

LIGO Data Analysis System (LDAS) Overview of Top Level Requirements *(from 12/97 review)*

- Assumptions/Dependencies
  - Detector delivers a fully functional DAQS. Data are written in frame format to a disc cache system available to the on-line LDAS. Includes the availability from DAQS of data-valid logic flags to identify saturated or aliased waveforms.
  - Detector implements interferometer diagnostics system. LDAS does not need to provide real-time (signal) feedback information to the LIGO interferometers. LDAS - Diagnostics System interface shall be primarily through the operator or scientist. Data or parameters derived by diagnostics routines will be done ~~through~~ using frame-based data.
  - LDAS, together with DAQS, will provide for an on-line (volatile) data storage system capable of accommodating a volume of data sufficient to provide overlap between shifts.
  - LDAS goal shall be to process datastream in real-time and on-line. This includes providing for the exchange of detection event lists between LIGO sites.
  - The off-line system does not directly interface to the on-line system. Connectivity is not precluded; not part of network topology at present.
  - Data reduction shall be accomplished as far upstream in the data acquisition process as possible in order to enable LIGO to archive reduced datasets for at least ~~5~~ 2 years. As a target, a minimum volume reduction of 10X is assumed. As a minimum, the GW channel, calibrated in strain, shall be archived permanently.

LIGO Data Analysis System (LDAS) Requirements *(from 12/97 review)*

- Assumptions/Dependencies
  - Specifically not considered to be within the scope of the LDAS are:
    - Data analysis functions performed at centers other than the LIGO Laboratory Facilities.
    - The on-line diagnostics system used for stimulus-response characterization, transfer function determination, and calibration functions. However, it is expected that software developed for the LDAS will find utility within the diagnostics system (e.g., numerical algorithm libraries, database tools).
  - Simulations shall be provided separately from, but coordinated with, the LDAS. The interface shall be using frame-based representations of simulation outputs.

LIGO Data Analysis System (LDAS) Requirements

- Mission-critical services:

1. Provide on-line analysis at the observatories.

- ›› Physical strain extraction possibly using relevant ancillary channels (e.g., PEM) to remove instrumental or environmental signatures.
- ›› Processing of strain data through real-time detection algorithms for both performance monitoring and scientific purposes.
- ›› A means to cross-correlate data (either time series or event lists) from multiple interferometers.
- ›› A means to store data frames and analysis results (local to the Observatory LAN) to short term storage media. This functionality will be provided by the LIGO DAQS resources, with augmentation by LDAS.
- ›› A means to access both “live” and short term archived data via the Observatory LAN and the LIGO WAN. Access shall be subject to available bandwidth and demand.
- ›› Means to retrieve, concatenate and extract specific channels of recent data from the on-line storage system.
- ›› Sufficient automation to run continuously and autonomously during periods of normal operation.
- ›› *A means to display and visualize results of analyses over the Observatory LAN.*
  - *Extensive graphics available with Matlab*
  - *Primitive graphics with LDAS userAPIs built on Tcl/Tk*

LIGO Data Analysis System (LDAS) Requirements *(from 12/97 review)*

- Mission-critical services:

2. Provide for extended off-line processing capabilities:

- » A means to reduce the raw data to science data representing calibrated GW strain data and a reduced subset of ancillary data and a data quality descriptor (will also be available on-line).
- » A means to archive, retrieve and distribute reduced datasets acquired over a period of time at least ~~5~~ 2 years in duration.
- » A means for duplicating reduced datasets either for backup or for distribution.
- » Sufficient computing margin to enable multiple analyses to be conducted in parallel.

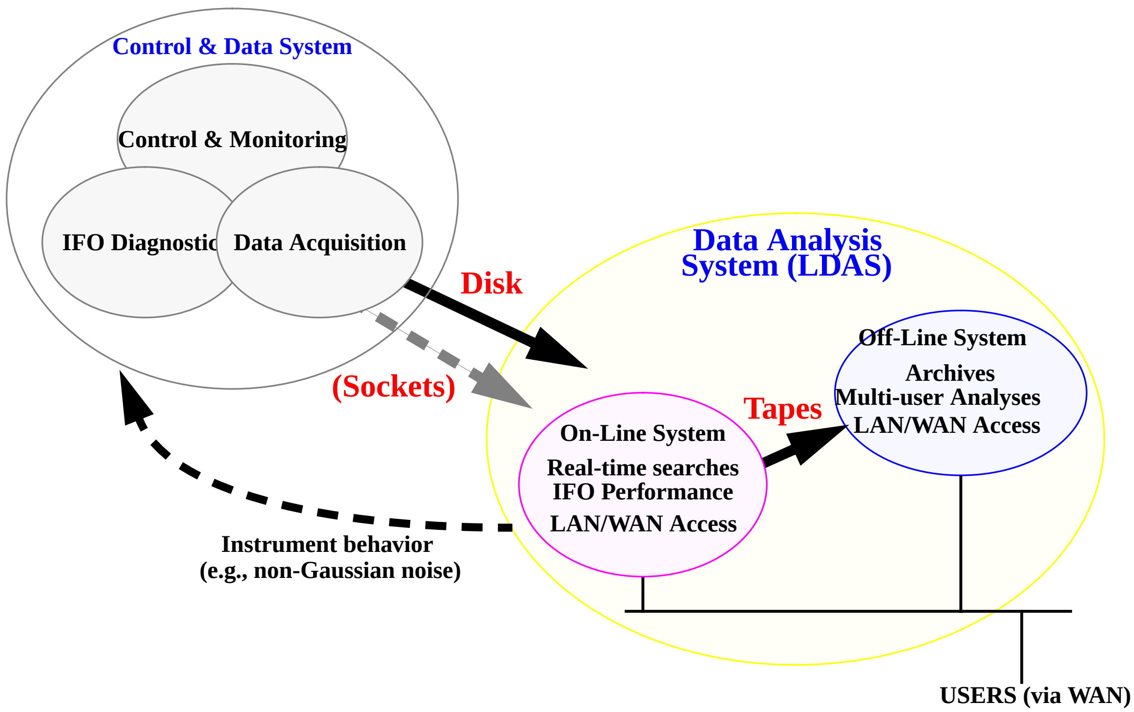
3. Provide a means to access the data archive via the LIGO WAN by the LIGO Laboratory and LIGO Scientific Collaboration to support ~~database~~ data products (frames; dBs; LigoLW objects) manipulation at the off-line site by remote users.

4. Provide a flexible design which can be reconfigured to reflect new analysis or computational requirements as they evolve.

5. Implementation goals:

- » Flexibility => No (or very little) custom hardware with custom software interfaces
- » Extensibility => Modular (not function specific) component design
- » Portability => Upgradable hardware under same software or vice-versa => POSIX compliance, software standards, etc.
- » Maintainability => Object oriented programming design ("reusable software components")
- » Reliability => Distributed (redundant) components, independent of SW components
  - – implied by previous design features

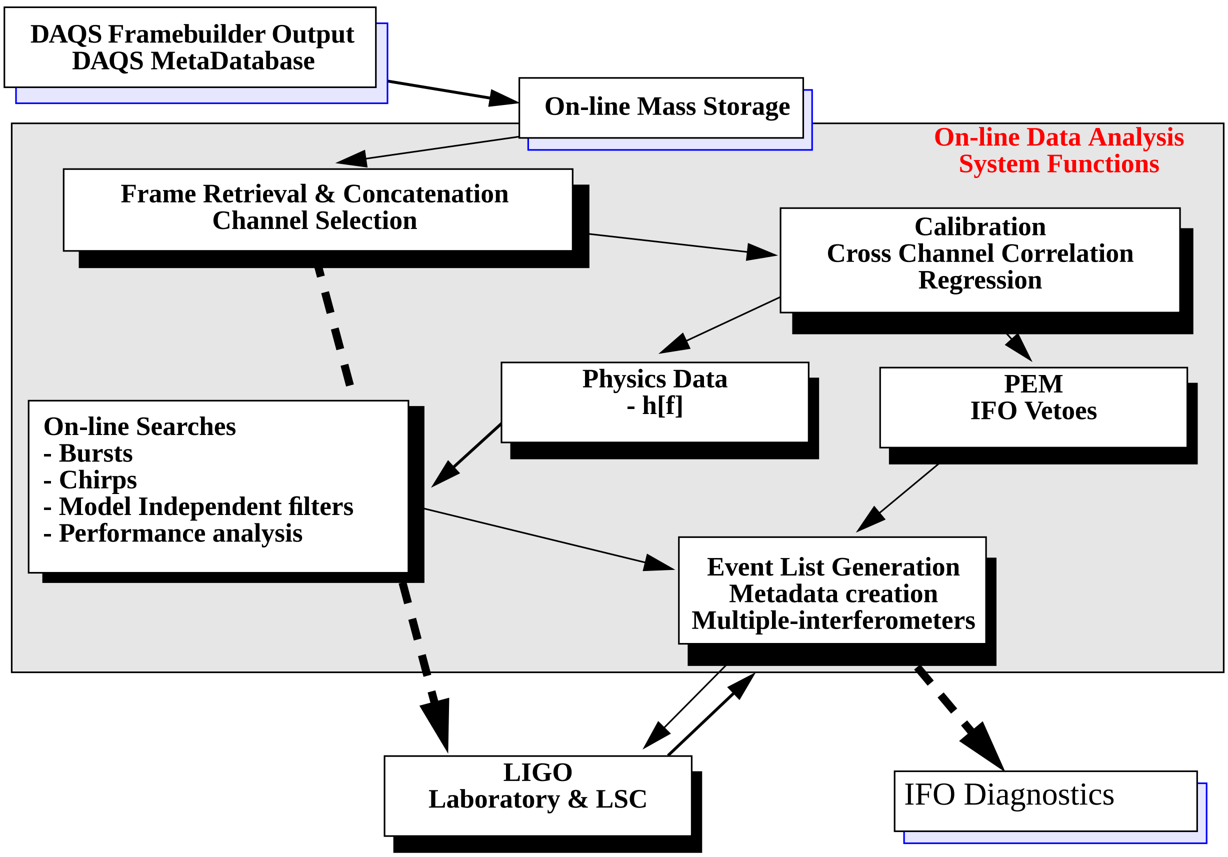
LIGO Data Analysis System (LDAS) Functional Units



LIGO Data Analysis System (LDAS)
Design

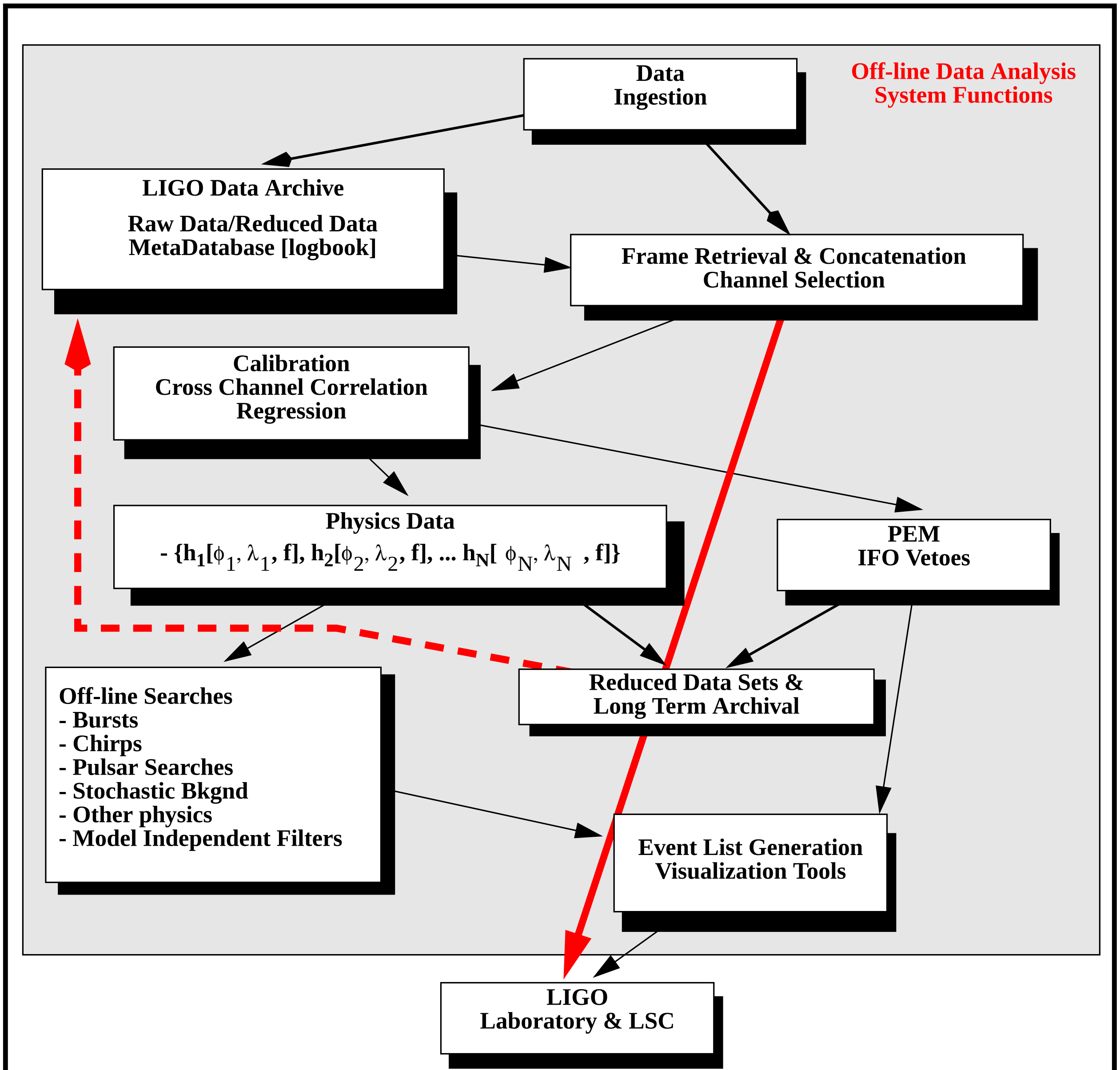
- Two LDAS components
  - » On-line LDAS
    - – Two systems, one for Hanford, and one for Livingston
    - – Hanford system handles 2 interferometers
    - – Provide computational power at the observatories to support diagnostics, detection, expansion/growth,...
  - » Off-line LDAS
    - – Collaborative arrangement with CACR
      - – Dedicated LIGO hardware within CACR on scale of observatory systems
      - – Database archive
      - – Strategic use of other CACR facilities as available
    - – Transparent access for off-line analysis of archived data
      - – LIGO Laboratory
      - – LIGO Scientific Collaboration
- Wide area network (WAN) to enable inter-site communications
  - » University scientific and engineering support to Observatories
  - » Access to archive database
  - » Access to real-time data from observatories
  - » Inter-observatory event sharing

LIGO Data Analysis System (LDAS)
On-line Functions



LDAS Flow Block Diagram: On-line Analysis

LIGO Data Analysis System (LDAS)
Off-Line Functions



LDAS Flow Block Diagram: Off-line Analysis

The LIGO Data Analysis System: LDAS

"Preliminary Design Review"

LIGO LABORATORY

MARCH 11TH, 1999

LIGO-G990XXX-00-E

Primary Purpose:

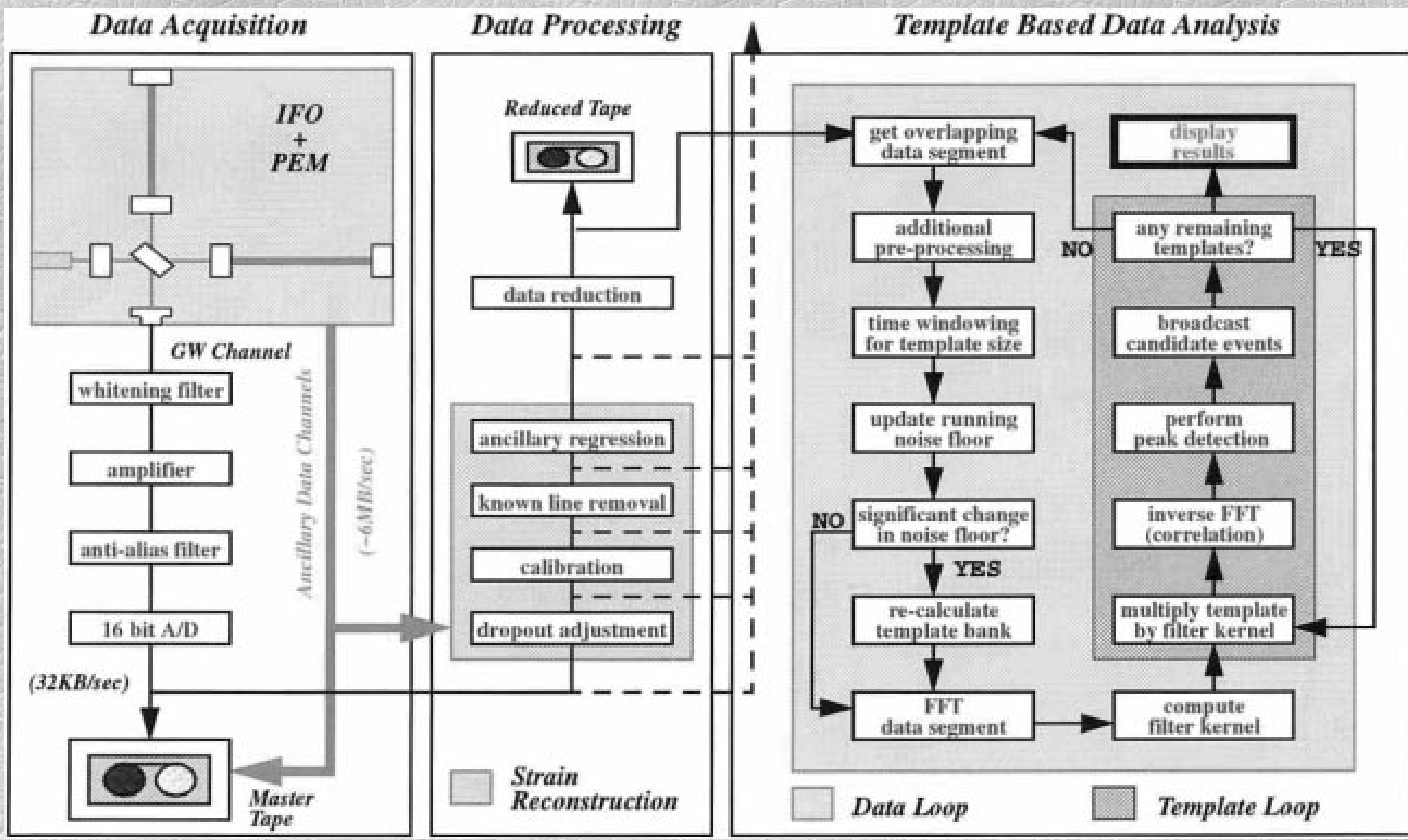
- First and foremost, the LIGO Data Analysis System is being implemented to detect and characterize gravitational waves from astrophysical sources.
- In addition, the LDAS will perform:
  1. raw frame data archival,
  2. database management functions for
     - raw frame data descriptors,
     - diagnostic trigger descriptors,
     - and astrophysical filter (template) event descriptors,
  3. data & metadata distribution services.

Principal Sources:

- Binary Inspiral of Compact Stellar Objects
  1. neutron star - neutron star
  2. black hole - neutron star
  3. black hole - black hole

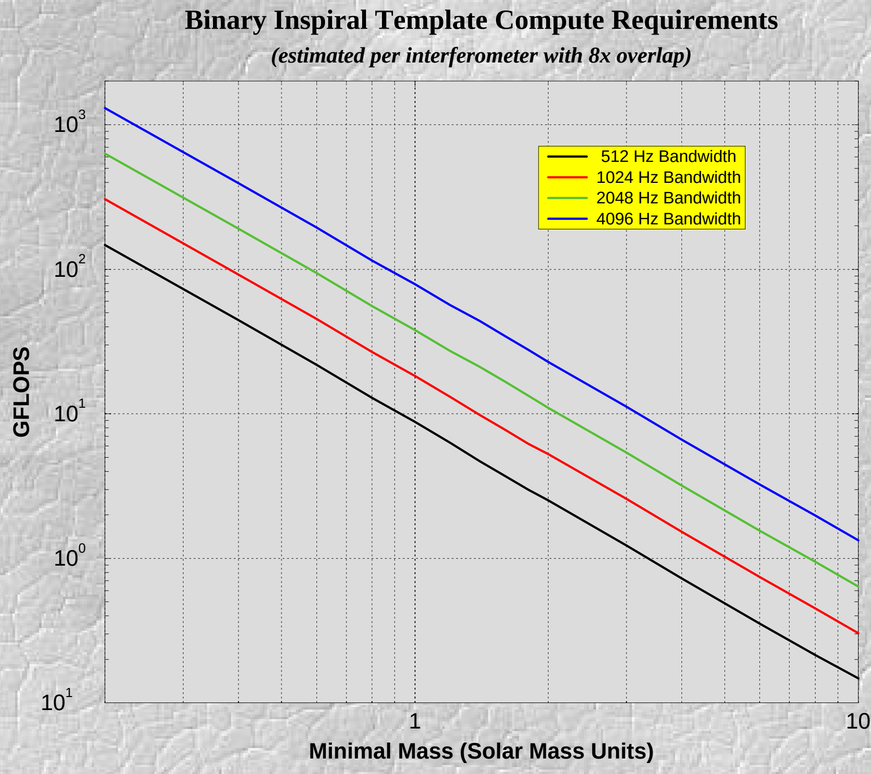
  } dominate LDAS requirements
- Burst Events
  4. supernovae (requiring coincidence between sites)
- Periodic Sources
  5. pulsars (all sky unlikely, but targeted searches easily carried out)
- Others
  6. black hole ring-downs
  7. black hole mergers
  8. stochastic background
  9. serendipitous discoveries!

Data Flow Model:

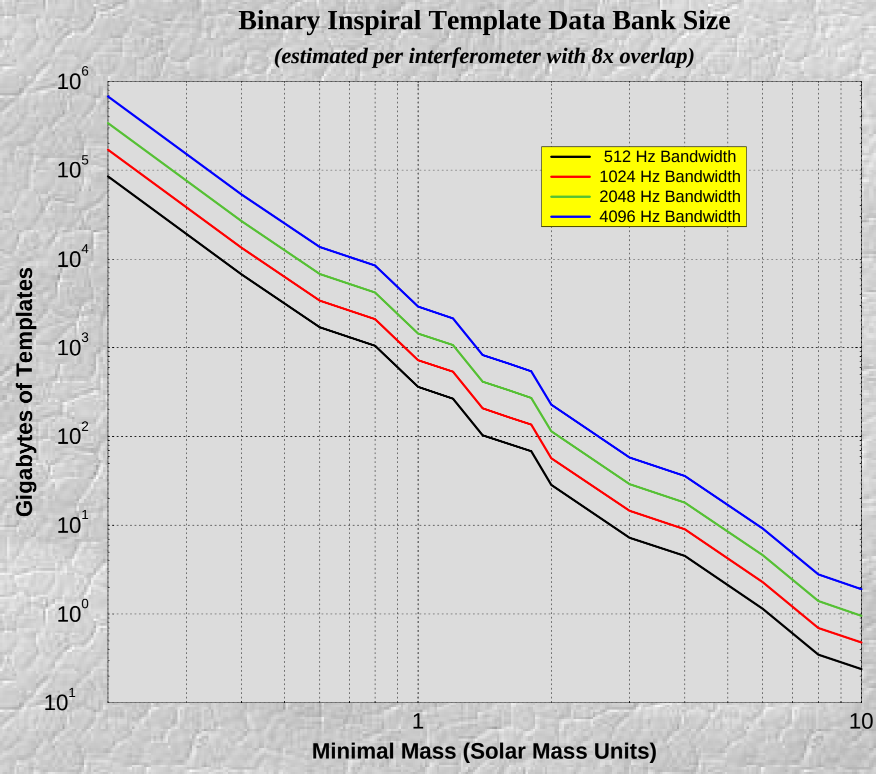


Optimal Filtering Demands:

- **Computation:**



- **Templates:**

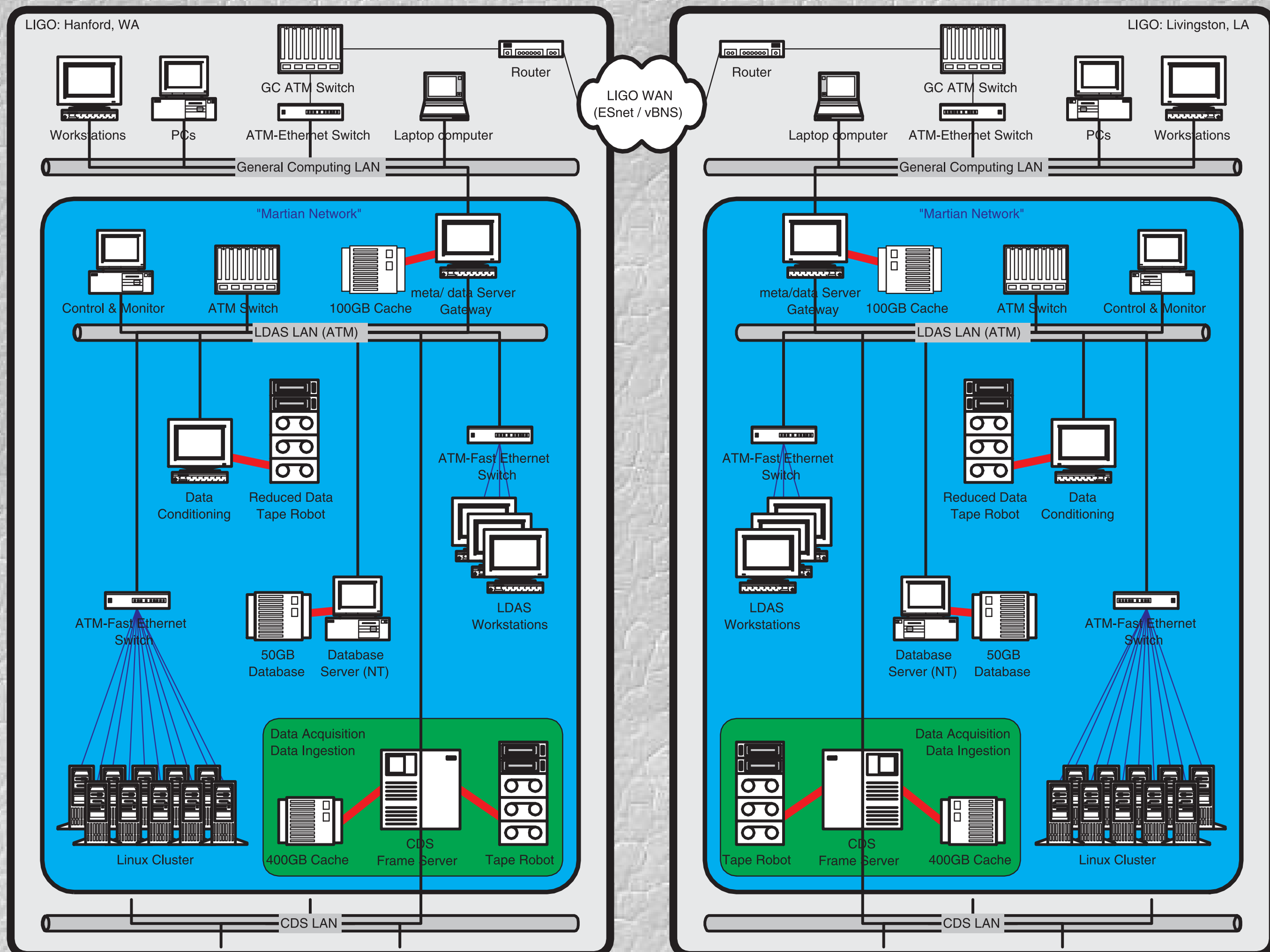


Hardware Design Needs:

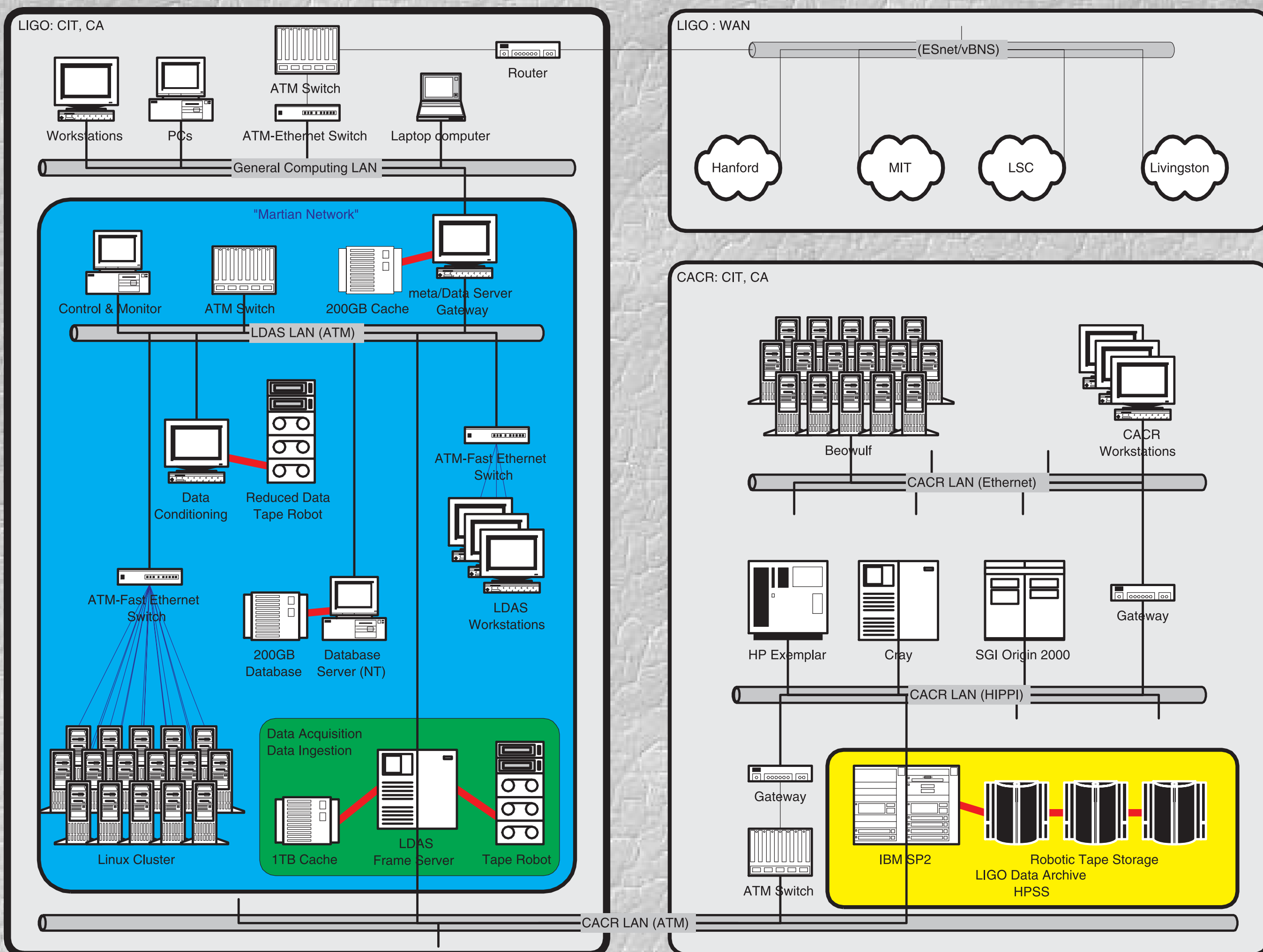
- **Requirements:**
  1. *10-100 GFLOPS of Compute Performance & 0.5-5 TB of uncompressed Templates*
  2. *100-620 megabits per second point-to-point*
  3. *500 GB of On-Line Disk Cache*
  4. *50-500 TB Archived Data per Year*
  5. *50-500 GB of metadata per Year*
  6. *LAN & WAN Networks*

- **Technologies:**
  1. *Beowulf (PC-Linux Clusters)*
  2. *Switched ATM & Fast-Ethernet for LANs*
  3. *SCSI Hard Disk Storage*
  4. *HPSS, Tapes (optical?)*
  5. *Database Capable of handling this Volume*
  6. *LIGO ATM & Fast-Ethernet LANs with ESNet and vBNS connectivity between the Observatories & Archive*

Hardware Architecture(online):



Hardware Architecture(offline):



Software Design Needs:

Requirements:

1. *Portability* -
   - POSIX for UNIX I/O
   - ANSI C/C++ for Compilers
   - TCL/TK for Steering
   - MPI for Parallel Computing
   - ODBC for Database Clients
2. *Extensibility* -
   - Modular/Reusable Code
   - OOP Design
3. *Maintainability* -
   - OO Languages Where Practical
   - CVS Source Code Management
4. *Flexibility* -
   - Class/Object Design
   - Modular Libraries
   - Distributed Processing
5. *Reliability* !

Components:

1. *Data Formats* -
   - IGWD Frames
   - Lightweight (Metadata, Events, Templates, Communications ...)
2. *Libraries* -
   - Supported Data Format I/O
   - Numerical (FFT's, filters, etc.)
   - POSIX & Socket interfaces
3. *LDAS API's* -
   - Supervisors to Data I/O
   - Control, Monitor, & Management
   - MPI Level Communications
   - Filtering and Analysis
4. *UI's* -
   - TCL/TK (Wish Shell) GUI
   - TCL Interpreter Command Line
   - TCLet Plug-ins to Web Browsers

Standardized Data Formats:

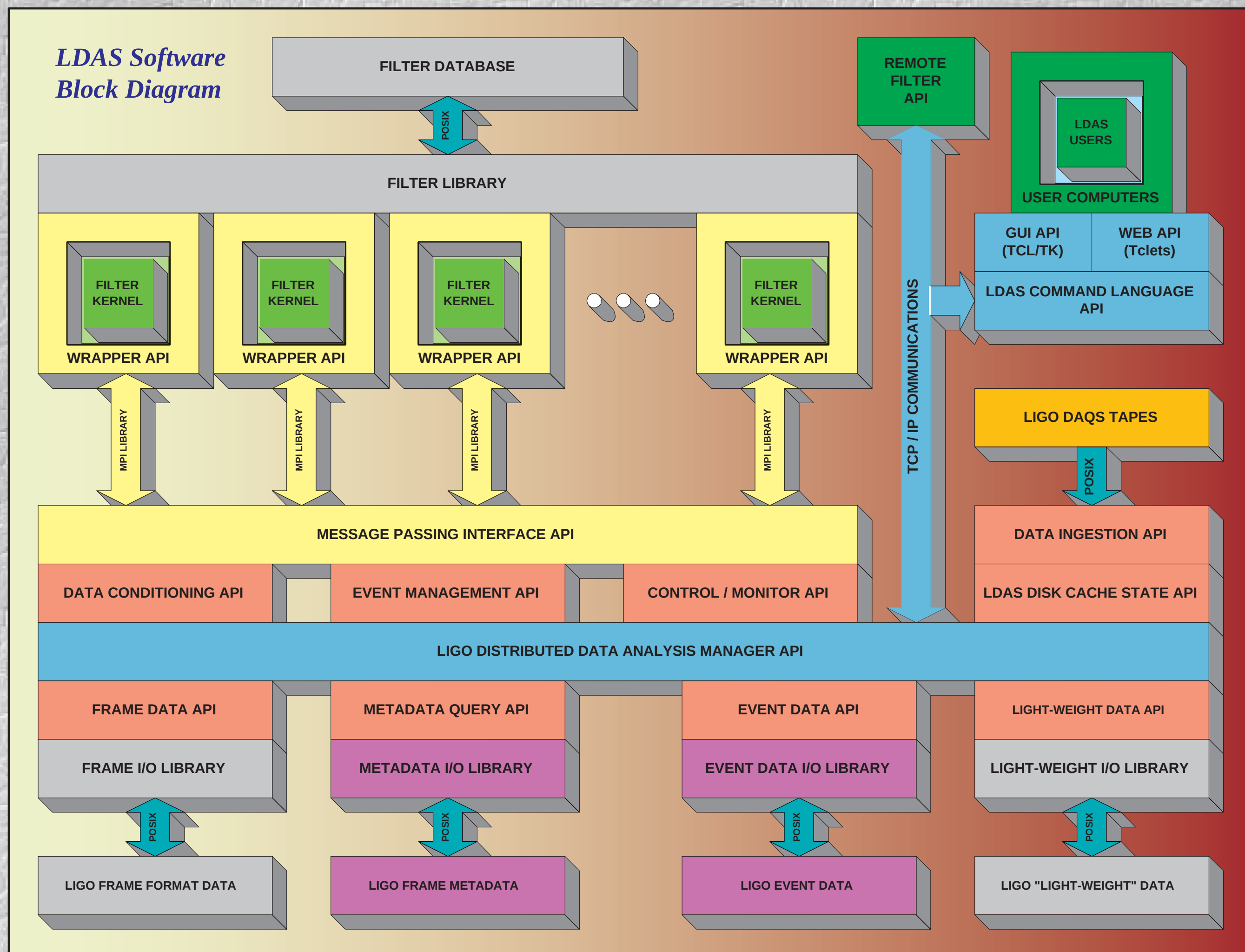
Frame:

1. Structured (C-like) Format
   - Samples of these structures:
     - ✓ ADC, Static, Detector, Trigger, ...
     - ✓ Simulated, History, Logs, ...
2. Jointly Developed with VIRGO
   - LIGO-T970130-B-E
   - VIRGO-SPE-LAP-5400-102
3. Original I/O Library in C
4. C++ Class I/O Library
5. Primary Uses:
   - Data Acquisition
   - Data Archival
   - *Subsystem for Diagnostics*

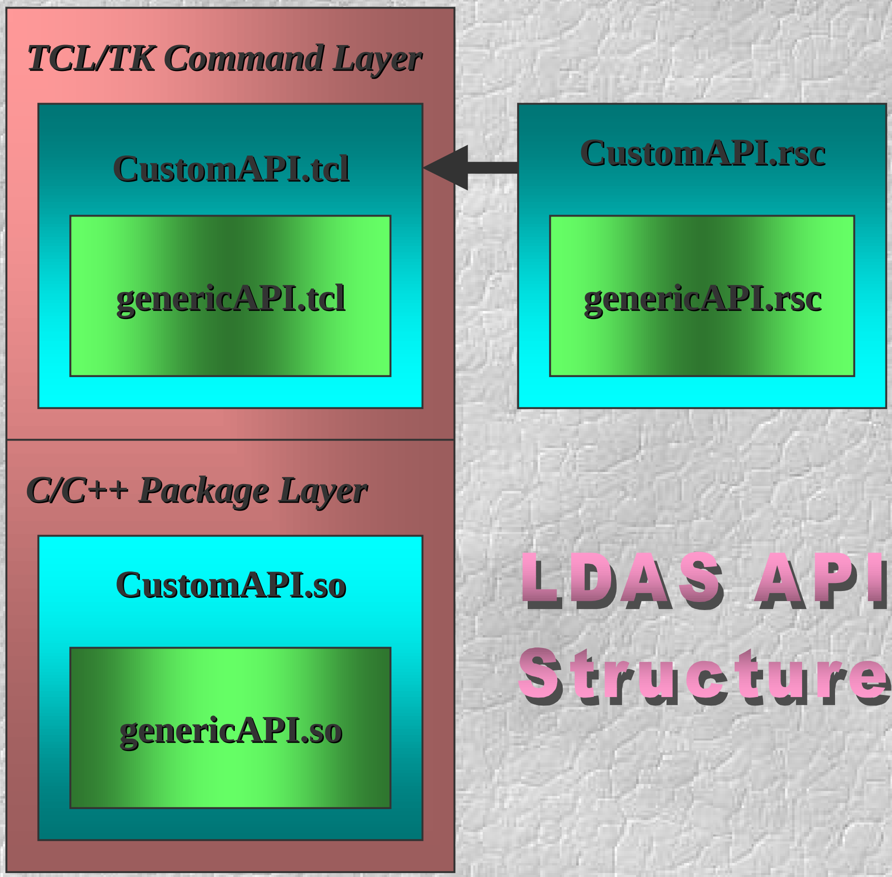
Lightweight:

1. Tagged (XML-based) Format
2. Easily Parsed *(and written)*
   - <int_s format="ascii">57, 7, 15, 12, 15, 31, 755</int_s>
3. LIGO Defined Objects
   - tables (n-tuplets),
   - arrays (matrix, vector),
   - vectors (time-series, power-spectra)
   - Some Revisions expected
4. Use Complement Frames!
   - LIGO Event data
   - LIGO Metadata
   - Spectra & Time-series data
   - Inter-process Communications

Software Architecture(block):



Layered LDAS API Design:



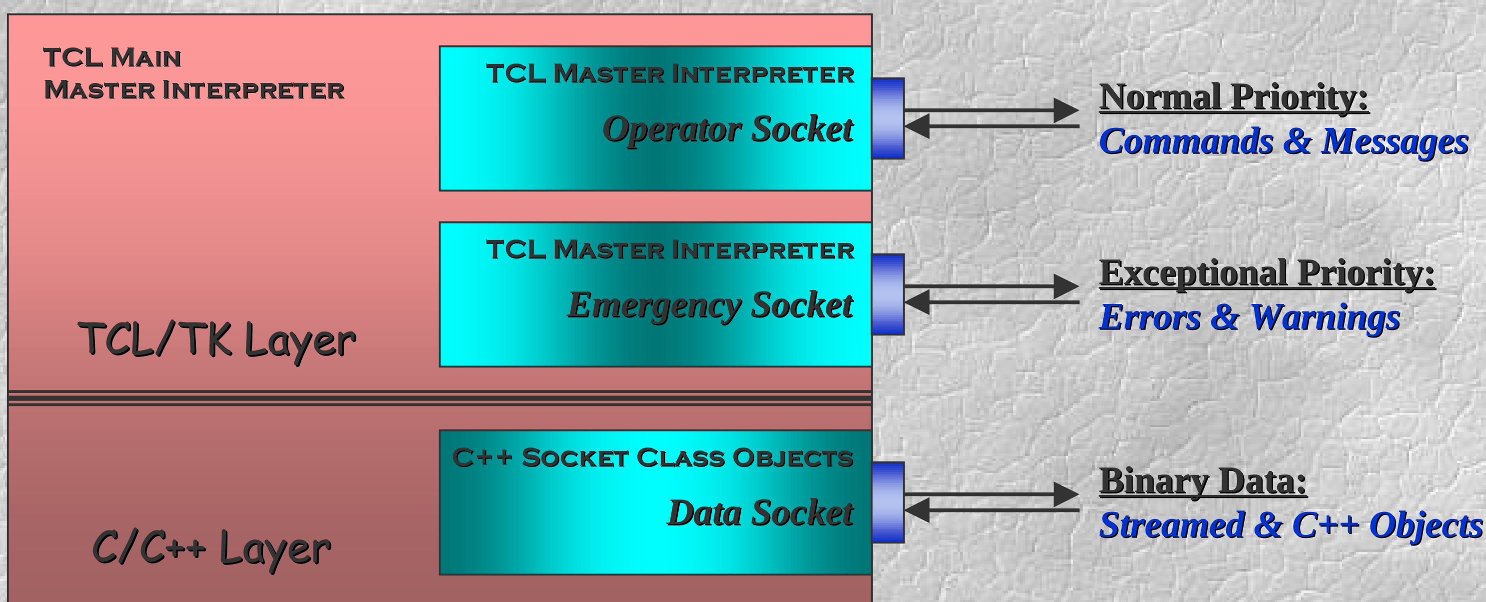
LDAS API Structure

LDAS API's:

1. Two Layers:
   - TCL/TK
   - C/C++ *(extends TCL Language)*
   - SWIG Unifies Layers
2. GenericAPI *(core)* Module:
   - Communications
     - ✓ *TCL <-> C++*
     - ✓ *API <-> API*
   - Common TCL proc's:
     - ✓ *Help*
     - ✓ *Logging*
     - ✓ *Command Socket Management*
     - ✓ *Resource Management*
   - Common C/C++ methods:
     - ✓ *Data Socket Management*
     - ✓ *Internal Data Management*
     - ✓ *Class Save & Restore*
3. *Custom (specialization) Module*

LDAS API Communications:

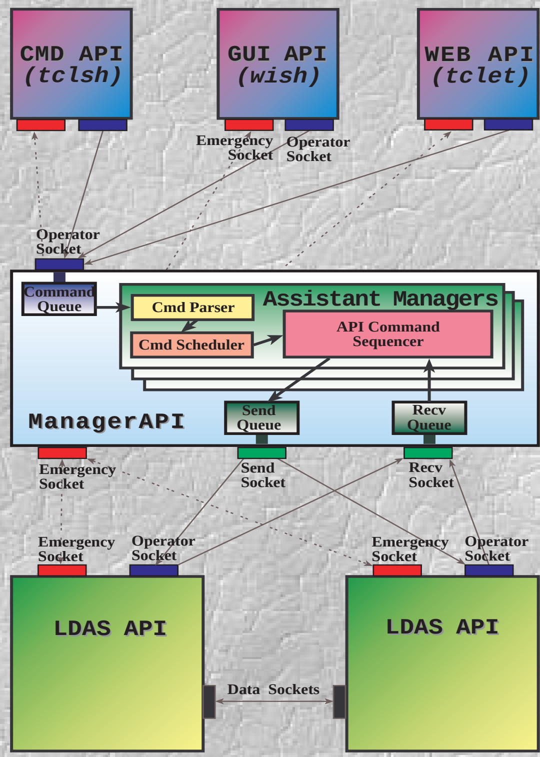
- 3 Types of Socket Communications in API's:
  1. Operator Sockets - Normal Inter-process Commands & Messages
  2. Emergency Sockets - Error & Warning Commands & Messages
  3. Data Sockets - Binary Data in either Raw Streams or C++ Objects



Distributed API Supervisor:

Manager API:

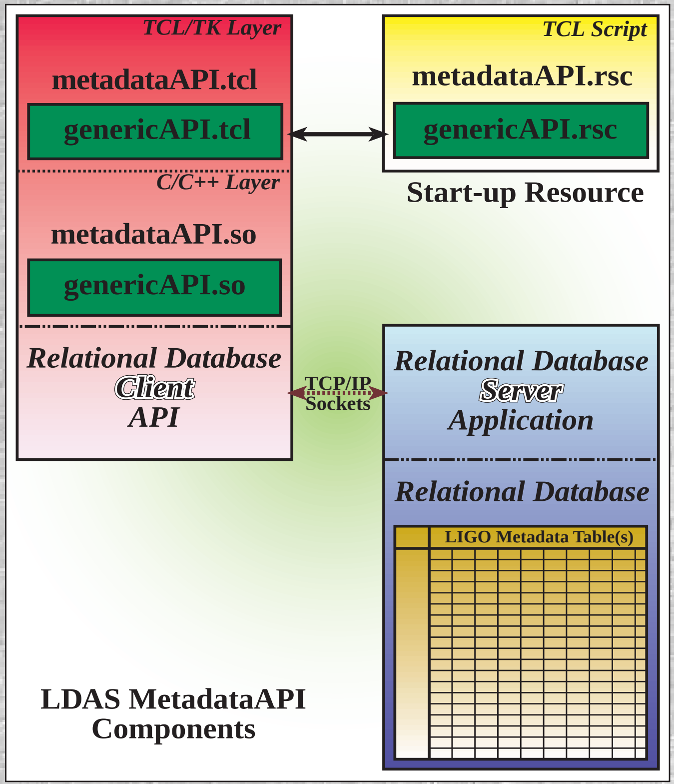
- Two Supervisory Layers:
  - Single "top-level" Manager
  - Multiple (3-10) Assistant Managers
- Manager Layer:
  - Communications
    - *Manages sockets & queues*
    - *Interface for User API's*
    - *Point of contact for All API's*
  - Exception Management
  - Log File Management
  - System Operator Interface
- Assistant Manager Layer:
  - Command Execution:
    - *Parsing*
    - *Schedule Building*
    - *Command Sequencing*
  - Reports to Manager Layer



Distributed Database API:

Metadata API:

1. Relational Database:
   - Using ODBC Standard for Calls
   - Currently Developing with DB2
   - Server can be Unix or NT Based
2. Table Contents:
   - Frame Characterization:
     - ✓ *Descriptors*
     - ✓ *Simple Statistical Summary*
   - Event Characterization:
     - ✓ *Optimal Filter Results*
     - ✓ *Astrophysical parameters*
   - GDS Trigger Results
   - CDS State Vectors
   - LDAS Logs
3. Metadata Query Sevices
   - LDAS Processing Queries
   - Data-Mining Queries



Database Event Tables:

Binary Inspiral Table

Event ID	Start Time	Delta Time	IFO Site	Number Frames	Mass 1	Mass 2	SNR	Confidence	Ringdown ID

Black Hole Ringdown Table

Event ID	Start Time	Delta Time	IFO Site	Number Frames	Quality Factor	Frequency	SNR	Confidence	Inspiral ID

Periodic Source Table

Event ID	Start Time	Delta Time	IFO Site	Number Frames	Sky Location	Frequency	Source Name	SNR	Confidence

Burst Table

Event ID	IFOWA4 Time	IFOWA2 Time	IFOLA4 Time	Duration	Number Frames	Sky Central	Sky Radius	SNR	Frequency	Bandwidth

Source Independent Table

Event ID	Start Time	IFO Site	SNR	Filter Type	Par 1 Name	Par 1 Type	Par 1 Value	Par 2 Name	Par 2 Type	Par 2 Value	…	Par N Name	Par N Type	Par N Value

see URL: http://www.ligo.caltech.edu/~xhu/index.html for Event, Frame, GDS, Log, CDS Table Descriptions

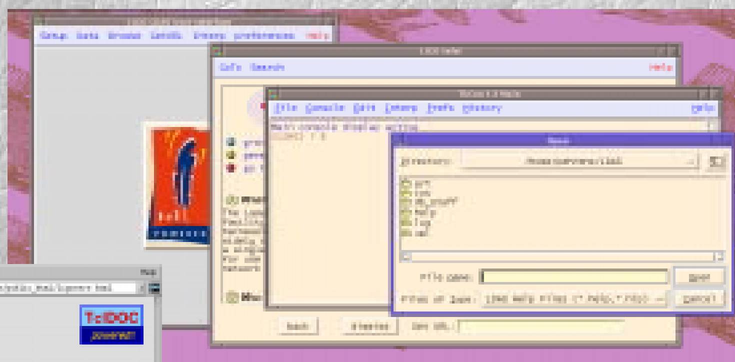
LDAS User Interfaces:

- **User Interface model based on TCL/TK**
  1. Doesn't preclude other software languages:
     - Any environment that can send LDAS Commands, properly formatted, to the ManagerAPI's Command Socket and receive LDAS Light-Weight Data Format can be made to work with LDAS!
  2. Portable between Unix / Windows / Mac Operating Systems
- **3 Types of User Interfaces Based on TCL/TK to be provided:**
  1. Command Line Interfaces - TCL shell scripts which communicate with the LDAS ManagerAPI sockets
  2. Graphical User Interfaces - TK Wish shell widgets which communicate with the LDAS ManagerAPI's sockets
  3. Web Browser Interfaces - TCLet plug-ins that display widgets in web browsers & communicate with the LDAS ManagerAPI's sockets

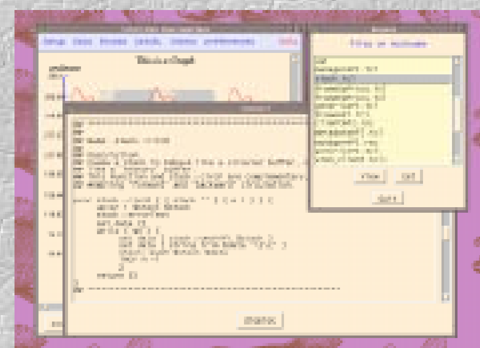
"Smart" User Interfaces:

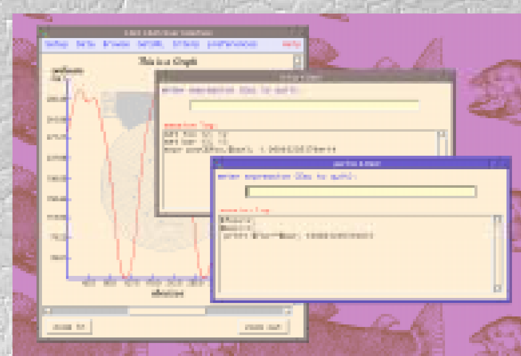
- **Initial Limited UI**
  1. Client/Server Socket Links
  2. Server sends TCL/TK Code

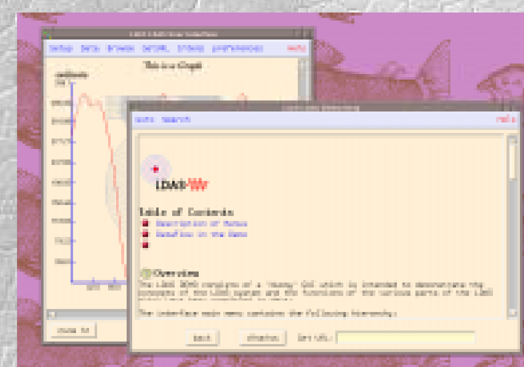
- **Final Custom UI**
  1. Appended Functionality to UI
  2. Provideds centralized UI Code Management



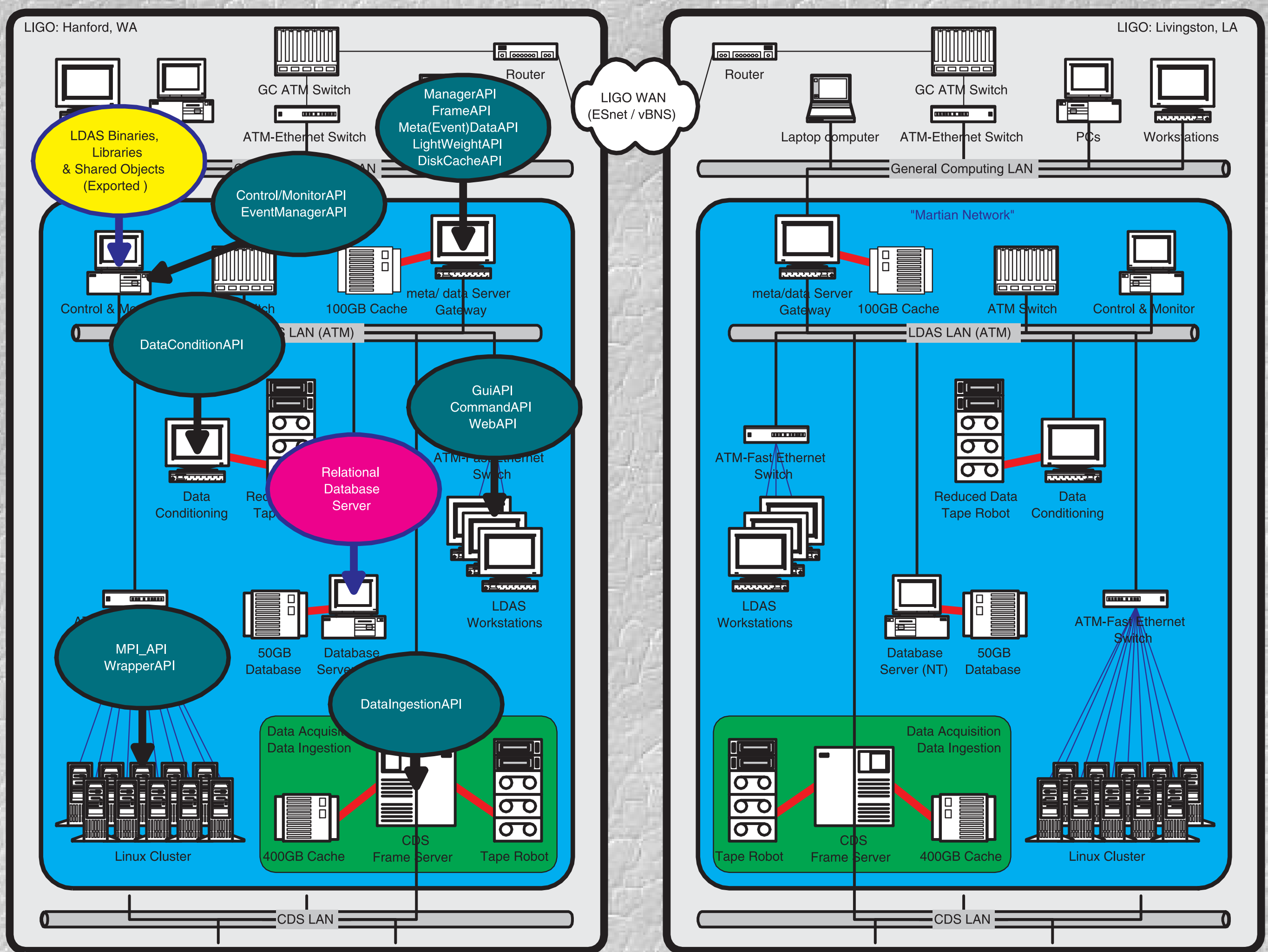
Socket Based Communications



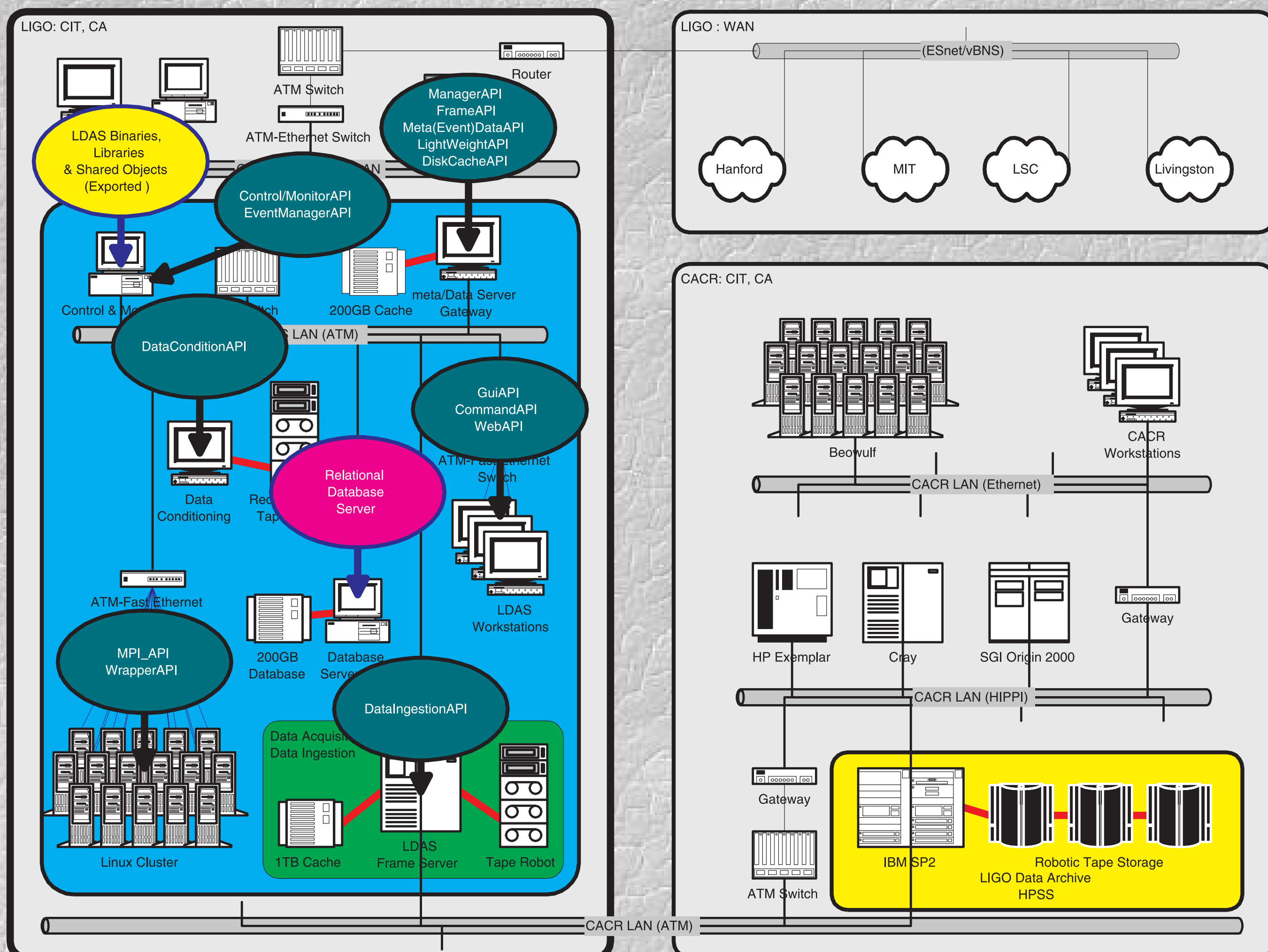




Software Mapping(online):

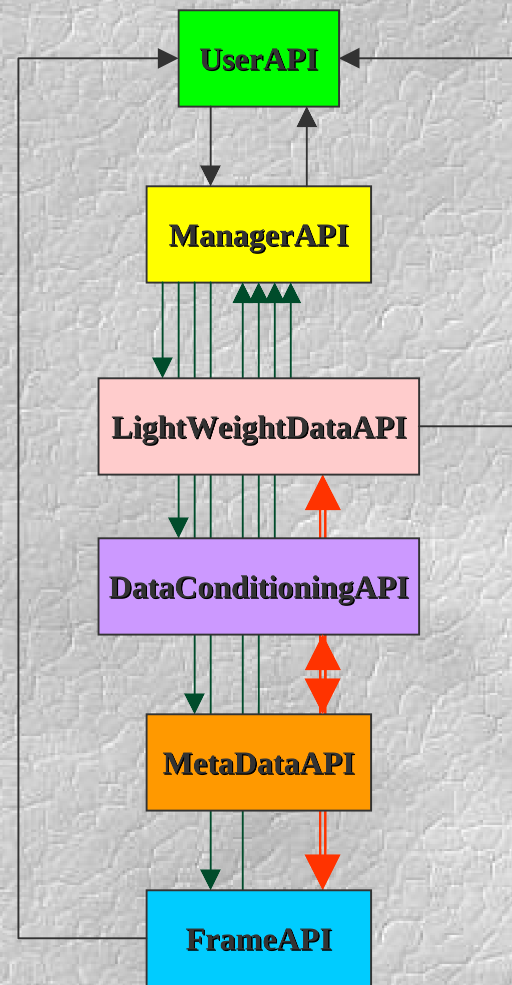


Software Mapping(offline):



First Software Deliverable:

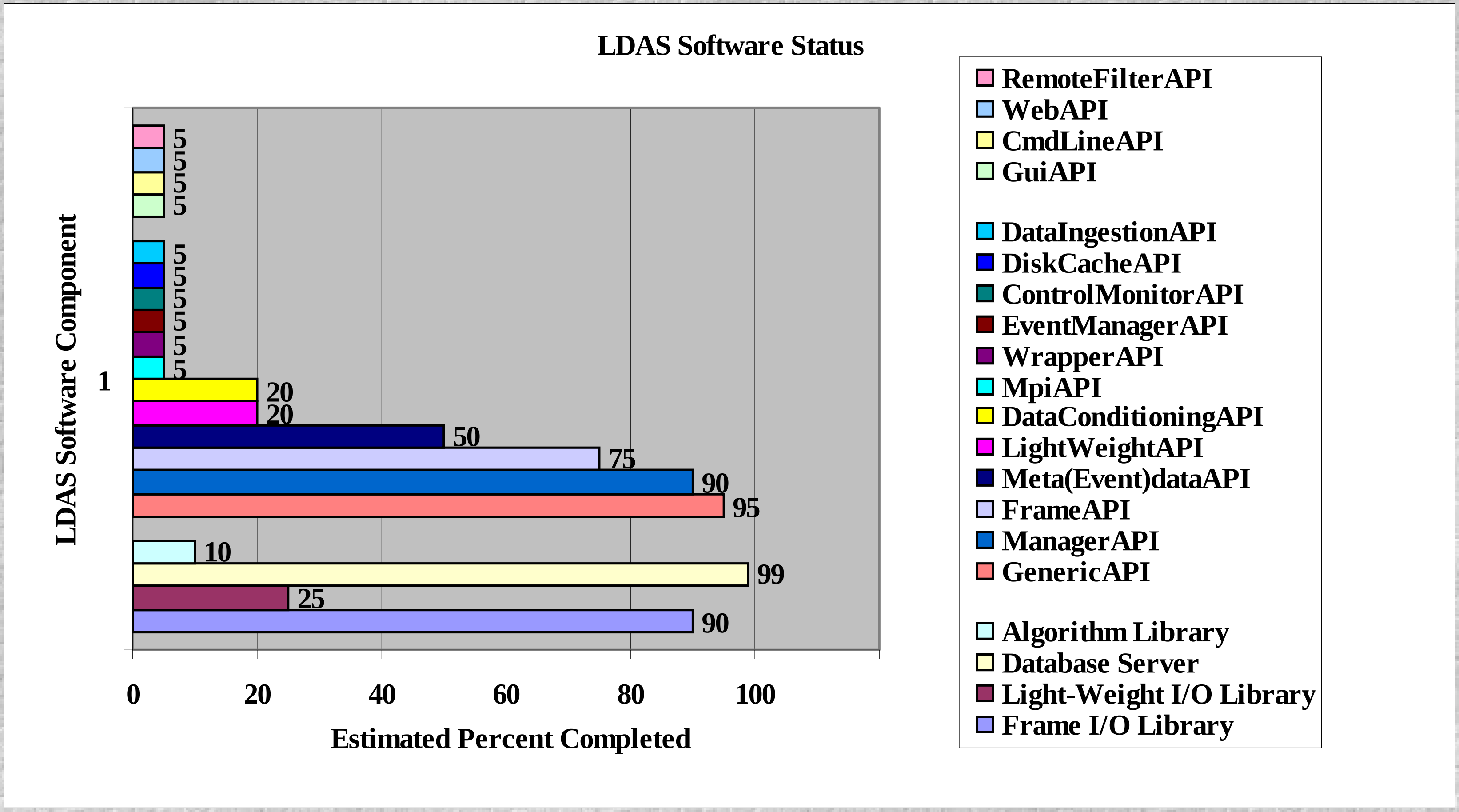
- α-release Summer 99
  1. LDAS Components to be included:
     - GenericAPI
     - FrameAPI
     - Meta(Event)dataAPI
     - LightWeightDataAPI
     - DataConditioningAPI
     - ManagerAPI
     - Simple UserAPI
  2. Some "Quick Look" Capabilities
  3. Support of Site Installation Activities
  4. Provide User Base for Testing and Debugging System



LDAS Documentation:

- On the Web: LIGO Data Working Group Bulletin Board
  - http://www.ligo.caltech.edu/~prince/LDCG_lsc/LDCG.html
    - LDAS Technical Review Documents
    - LDAS Software Guidelines
    - LDAS Software Requirements
    - LDAS Software Specifications
    - LDAS Code Formatted as HTML
    - LDAS Presentations
    - Other Useful Documentation
  - LDAS Source Code Presentation
    - C++ to HTML using DOC++
    - TCL/TK to HTML using TCLDoc
    - CVS Server to LDAS Software Repository (password protected)

Software Status:



Hardware Status:

- LDAS Hardware Design Relatively Complete
  1. Procurement being delayed/staged as much as possible
  2. PDR tentatively scheduled for March 11th
- WAN to Sites Established
  1. T1 links at both Hanford, WA & Livingston, LA
- First LAN Hardware (ATM Switch) Arrived at Hanford
- Compact Beowulf Cluster Received from ALTA Technologies for Prototyping
- meta/Data Server Hardware Identified in conjunction with CDS Frame-Builder
- Establishing Archival Storage Specifications in collaboration with CACR

Data Server:

- FUNCTIONS:
  - Access to frame data and metadata.Connected via to the to the CDS framebuilder.
  - Performs data decimation and channel selection from the raw frame data.
  - Users subscribe to distribution services on this server.
    - Data in the form of frames or lightweight data sets
  - ON-LINE SERVER:
    Gateway between the LDAS LAN (private) and the General Computing LANs at LHO, LLO and other LIGO Laboratory sites.
  - OFF-LINE SERVER:
    Server is a gateway between the LDAS LAN (private) and the local (CACR) LAN.
  - Provides access to clients for LIGO metadata created by GDS, CDS/DAQS and LDAS at the observatory.
- CPU:
  - UNIX SUN Ultra 60 class server with 1 GB RAM.
    - Frame data to multiple users at total bandwidth of ~20 MB/s over OC12 ATM.
- DISK:
  - ON-LINE/OFF-LINE:
    Server configured with disk cache of ~100GB, independent of the on-line disk farm. Permits data staging from the on-line cache if required to provide sustained throughput.

Data Server:

- NETWORK CONNECTIONS:
  - ON-LINE:
    Server connected to DAQS framebuilder.
  - OFF-LINE:
    Serve connected to HPSS archive over ATM.
- SOFTWARE:
  - managerAPI handles data requests from users
  - frameAPI provides interface to/from raw framed data
  - metadataAPI provides ingestion and retrieval for metadata
  - lightweightDataAPI provides interface to/from data stored as LigoLW data objects
  - diskCacheAPI manages data retrieved from the frame archive and stored to local (to the data distribution server) disk space
- **LIBRARIES**:
  - Frame data library for frame data I/O
  - Lightweight data library for LigoLW I/O
  - POSIX library

Database Server:

- FUNCTIONS:
  - Access to metadatabase.
  - Requests in ODBC standardized SQL submitted by the data server.
  - Metadata Database Management System (DBMS) server.
- CPU:
  - CPU will be an NT server machine with 512 MB RAM
- DISK:
  - 50GB
- NETWORK CONNECTIONS:
  - Connected to LDAS LAN at 100BT. Capable of serving metadata to multiple clients at total bandwidth of 10 MB/s
- SOFTWARE:
  - Commercial database management system (DBMS) server software
- LIBRARIES:
  - DBMS library
  - POSIX library

LDAS Control & Monitoring Server:

- FUNCTIONS:
  - Configures, starts and synchronizes operation of LDAS parallel computing resources.
  - Controls the LDAS on-line.
  - Summary performance statistics for LDAS functions.
  - Logs exceptions, errors, etc.
  - Logs event metadatabase created as part of the on-line detection algorithm data processing.
- CPU:
  - Linux PC with 256MB RAM, UWSCSI interface to disk.
- DISK:
  - 20 GB
- NETWORK CONNECTIONS:
  - 100BT ethernet via the LDAS ATM LAN.
- SOFTWARE:
  - eventManagerAPI; provides access into the event metadatabase
  - eventDataAPI; processes and creates LIGO event objects for ingestion into the metadatabase.
  - controlMonitorAPI; provides statistics, reports, and performance overview of LDAS. Fault, recovery and start-up management
- LIBRARIES:
  - Event library for event data I/O
  - Lightweight data library for LigoLW I/O
  - POSIX library

Compute Server (linux cluster):

- FUNCTIONS:
  - MPI-based multi-processor parallel computation for a number of analysis processes, including signal processing and detection algorithms
  - Manages template database (if needed) for optimal filtering
  - Generates events for further processing
- CPU:
  - Multiple nodes connected via high-speed switch network dedicated to inter-node communications.
  - PCs operating under linux. The number of nodes is extensible. Present estimates indicate ~48 nodes per interferometer;1
  - 256MB RAM per CPU (node)
- DISK:
  - Each node will have 4GB disk.
- NETWORK CONNECTIONS:
  - Internode connection @ 100BT or faster on a dedicated switch.
  - External gateway provided by master node, ethernet 100BT to LDAS LAN.

Compute Server (linux cluster):

- SOFTWARE:
  - messagePassingAPI to interface with MPI for parallelized computation.
  - wrapperAPI to provide C++ interface into procedural numerical algorithms written in other compiled languages (e.g., C, FORTRAN90).
- LIBRARIES:
  - MPI library.
  - Lightweight data library for LigoLW I/O.
  - Numerical filter library.
  - POSIX library.

Data Conditioning & Regression Processor:

» FUNCTIONS:

- Preprocessing of raw data for analysis/data reduction
- Reduced datasets for off-line data analysis
- User requests for specific datasets

» CPU:

- 1 or more PC linux machines with 512 MB RAM
- This function could be migrated into the linux compute cluster by allocating a group of dedicated nodes to this service

» DISK:

- UWSCSI, 20 GB per CPU

» **TAPES**:

- Tape robot capable of handling 5 - 10 cassettes. Exabyte 5 - 10 GB/cassette.

» NETWORK CONNECTIONS:

- UWSCSI to the CPUs for tape robot and disks.
- 100BT to LDAS LAN

» SOFTWARE:

- dataConditioningAPI for filter algorithms for a specific subset of signal processing technique
- frameAPI
- lightweightAPI

Data Conditioning & Regression Processor:

» LIBRARIES:

- Frame data library for frame data I/O
- Lightweight data library for LigoLW I/O
- POSIX library
- LIGO numerical algorithm library

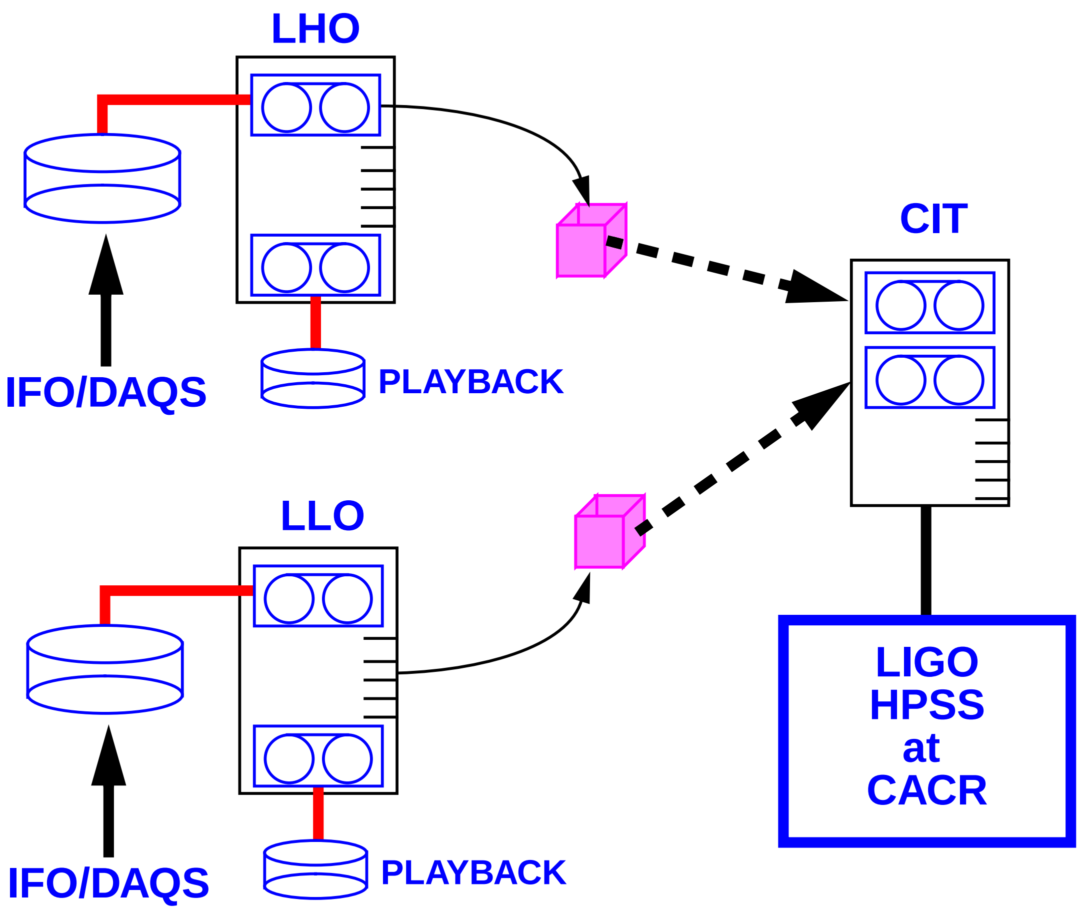
Data Ingestion:

- FUNCTIONS:
  - Ingests media written at observatories for incorporation into the data archive
  - Generates reduced datasets for long-term archiving
  - Access locally stored (backup) data at observatory
- CPU:
  - linux PC with 128MB RAM; UWSCSI interface for peripherals
- DISK:
  - 50 - 100 GB
- TAPE DRIVE:
  - Tape drive system identical with recording mechanisms installed at the observatories. Tape handling capability to accommodate >24 hours of data media (from both observatories and all interferometers) without requiring human intervention.
- NETWORK CONNECTIONS:
  - ON-LINE:
    UWSCSI interface for peripherals. TCP/IP over ATM running 100BT or 1000BT ethernet (OC3/OC12)
  - OFF-LINE:
    UWSCSI interface for peripherals. TCP/IP over ATM running 100BT or 1000BT ethernet (OC3/OC12)

Data Ingestion:

- SOFTWARE:
  - dataIngestionAPI to manipulate tape media and transfer data between tapes and disk cache
  - frameAPI
- LIBRARIES:
  - Frame data library for frame data I/O
  - Lightweight data library for LigoLW I/O
  - POSIX library

TAPE FLOW WITHIN LDAS



LIGO Data Archive -- On-line at Observatories

- Stage similar media technology on a (much) smaller scale
  - » Provides local access to recent data
  - » Archive is a small scale version of off-line system
    - – Storage Tek robot with Redwood drives
    - – 10 TB storage
    - – One or two drives used to create raw data tapes from DAQS on-line cache
    - – One drive available as spare/ingestion unit for playback.
  - » Minimizes media-to-media transfers
- Same LDAS software components can be used to ingest/retrieve data as for off-line systems
- Can use HPSS software if cost is justifiable

LIGO on-site Storage

- StorageTek Robotics (TimberWolf 9740):
  - I/O: up to $4 \times 11\,$MB/s.
  - Capacity: up to 25TB.
  - Footprint: $1.2\,\mathrm{m} \times 1.75\,\mathrm{m}$.
  - Media exchange: 14 cartridge access port.
- Baseline Configuration:
  - 2 Redwood drives with separate disk caches.
    * Create dual-copy master tapes.
    * Playback and master drive spare.
  - Joint tape pool (~ 50 days @ LHO).

What is HPSS?

CENTER FOR ADVANCED COMPUTING RESEARCH

- HPSS - High Performance Storage System
- Provides Hierarchical Storage Management (HSM)
- A collaboration between Department of Energy Supercomputing Labs and IBM Government Systems.
  - Includes Los Alamos, Lawrence Livermore, Sandia, Lawrence Berkeley (NERSC), and Oak Ridge

CALIFORNIA INSTITUTE OF TECHNOLOGY

CENTER FOR ADVANCED COMPUTING RESEARCH

HPSS - What's the big deal?

- Scalability
  - Files: billions
  - File sizes: gigabytes - terabytes
  - Total storage: petabytes - exabytes
  - Transfer rate: gigabytes per second

CALIFORNIA INSTITUTE OF TECHNOLOGY

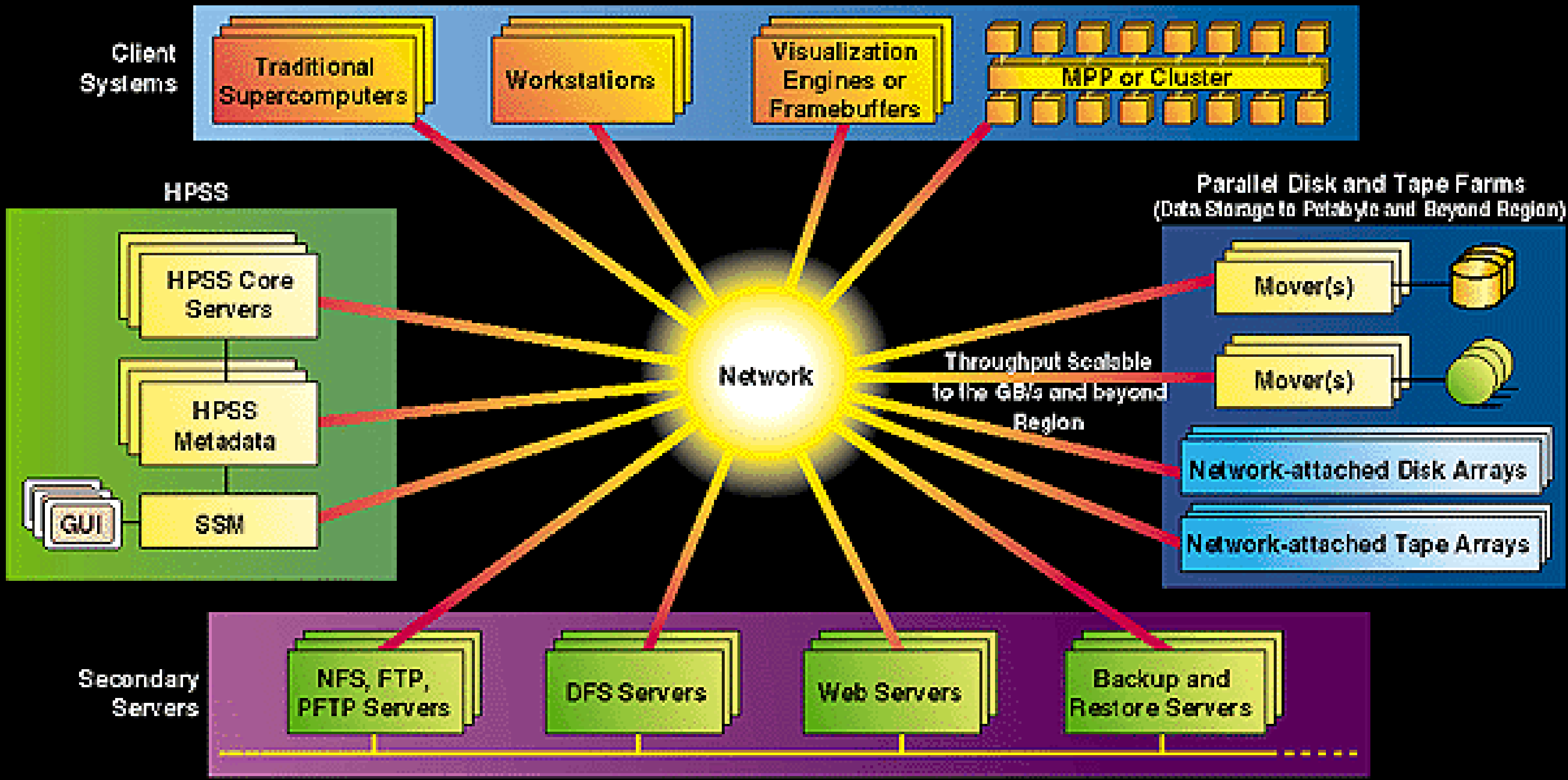
HPSS - What's the big deal?

CENTER FOR ADVANCED COMPUTING RESEARCH

- Reliability
  - Transactional design
    - Consistency maintained in the distributed system via commit / abort and rollback semantics
    - Implemented using Transarc's Encina.
    - Utilizes journaling which allows transactions to be backed out upon failure.

CALIFORNIA INSTITUTE OF TECHNOLOGY

System Architecture Supported by HPSS

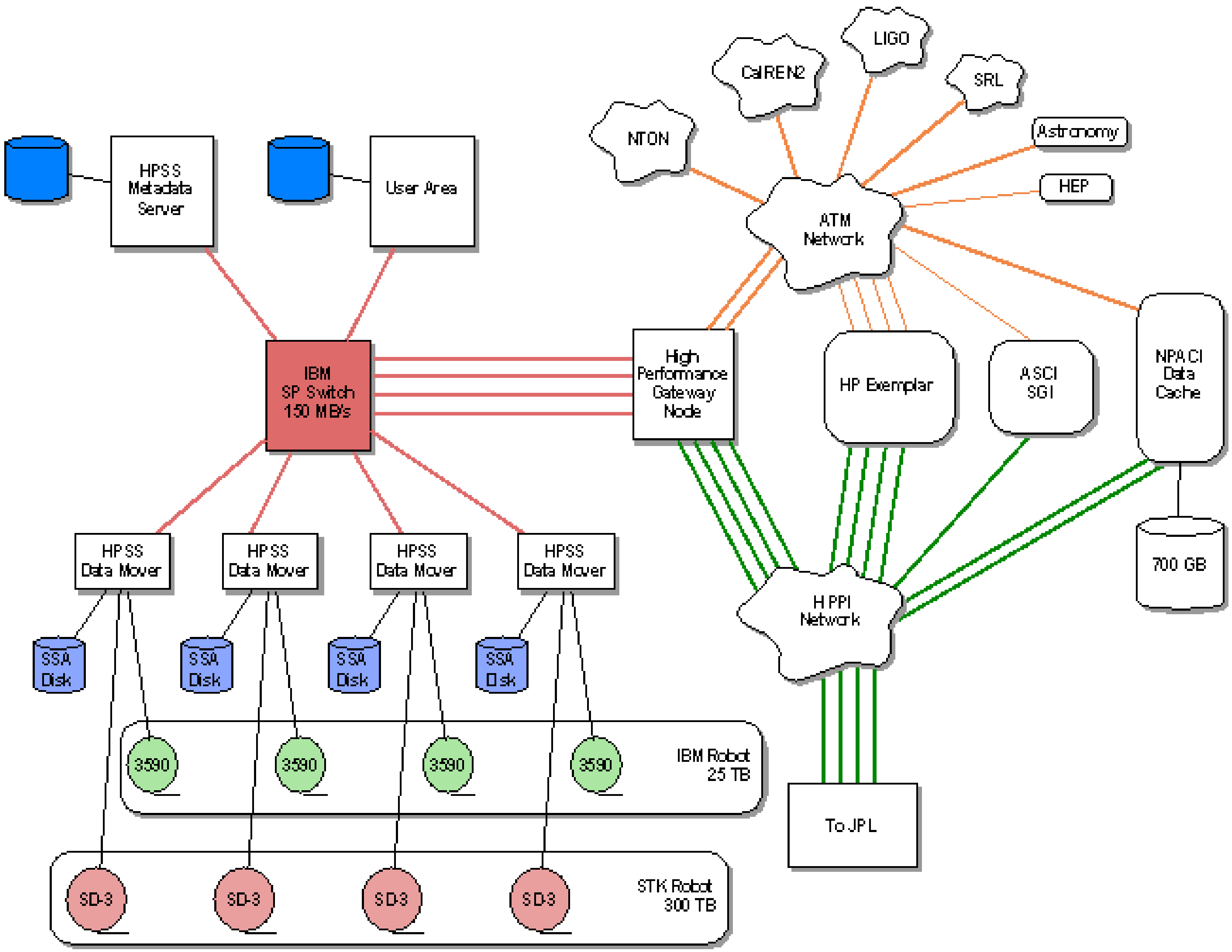


LIGO HPSS Capability

- Storage and access for ~ 100TB of LIGO data for indefinite period of time
- Support 4 independent users seeking data
- Tape drive throughput to disk cache ~ 11 MB/s (per unit)
- Use SP2 rack backplane as a high speed/low latency network to communicate between HPSS processes.
- One 4-way PCI node would support metadata services (HPSS metadata, not LIGO metadata)
- Four other nodes operate as parallel data movers
  - » 4 independent users, or
  - » 4-way striped tapes (data striped across 4 tape cassettes) for throughput
- High performance gateway node (HPGN) would interface directly to SP2 backplane and allow high speed access to HPSS via ATM, HiPPI, gigabit or other high perf. network technologies

CENTER FOR ADVANCED COMPUTING RESEARCH

Networking to HPSS



CALIFORNIA INSTITUTE OF TECHNOLOGY

LDAS Interfaces to CACR Systems - HPSS

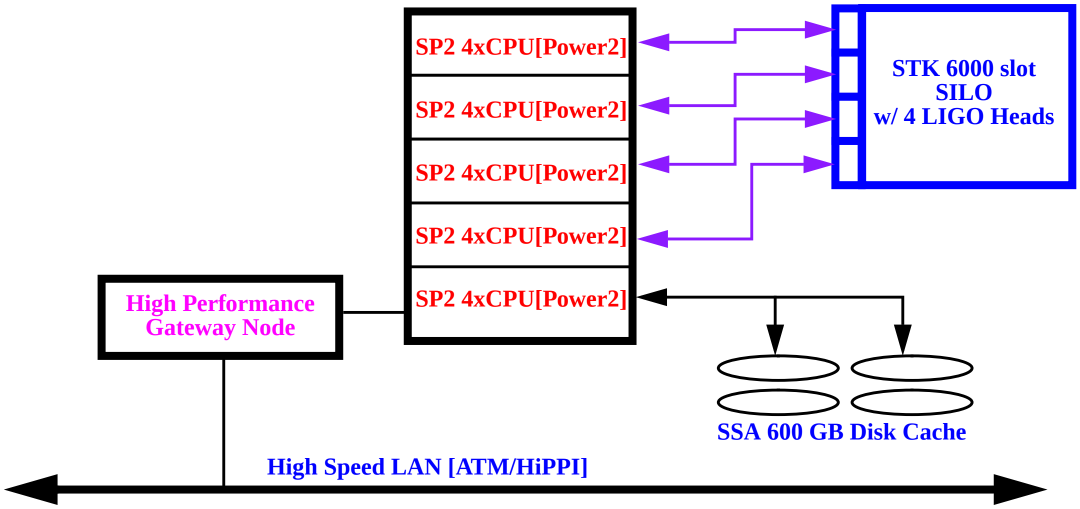
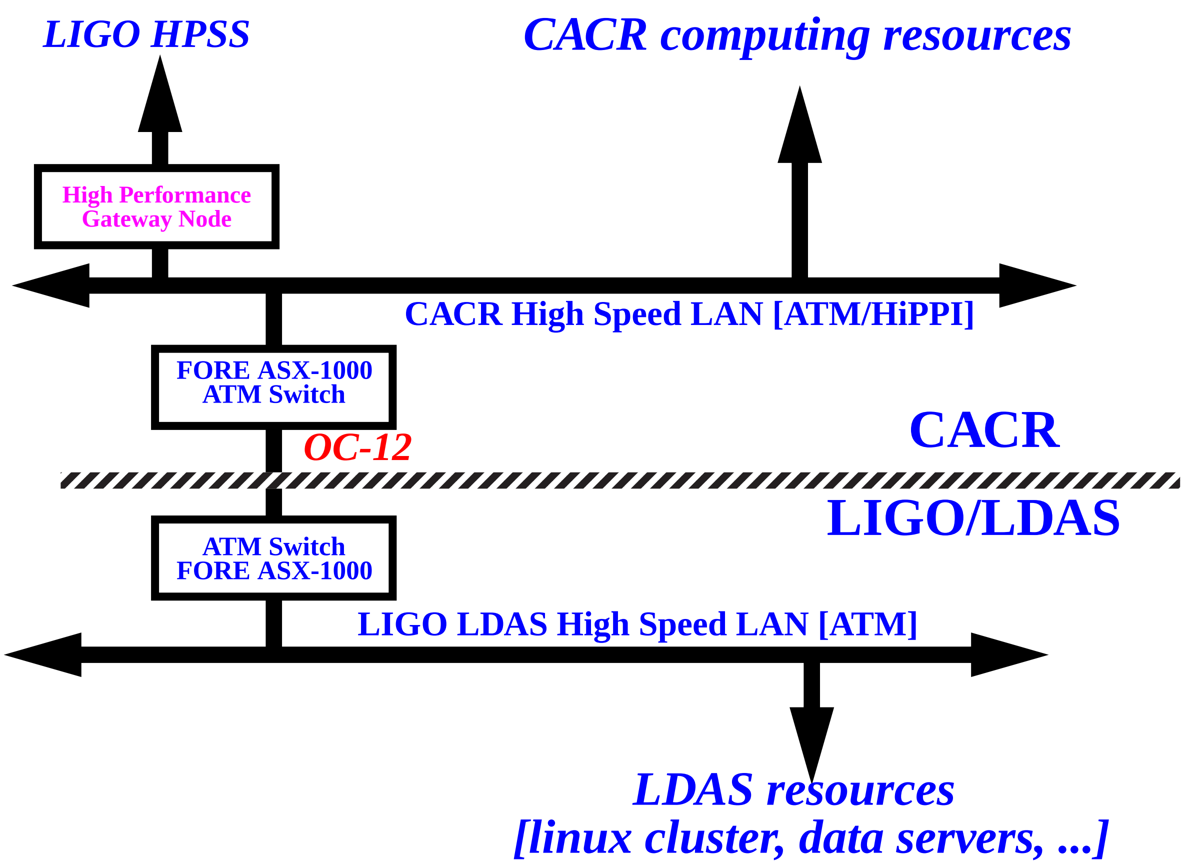


Figure 1 HPSS Configuration for LIGO

LDAS Interfaces to CACR Systems



CENTER FOR ADVANCED COMPUTING RESEARCH

HPSS Access Methods

- FTP
  - Compatible with all standard FTP clients
  - Not the most efficient since HPSS doesn't know the file size until it already has all the data - must reshuffle after file close
  - Data must be written to the FTP server, which then passes it to the data movers.
- PFTP - Parallel FTP
  - FTP server only sees control connection
  - Data flows directly to the mover that will be storing it
  - Parallel streams possible
- HSI
  - Command-line and scripting interface to HPSS
  - Supports recursive actions, stdin/stdout, chmod, find.
  - See: http://www.sdsc.edu/Storage/hsi
- Client API
  - HPSS provides a POSIX like interface (e.g. hpss_Open, hpss_Close, hpss_Read, hpss_Write, etc)

CALIFORNIA INSTITUTE OF TECHNOLOGY

New HPSS Versions

CENTER FOR ADVANCED COMPUTING RESEARCH

CALIFORNIA INSTITUTE OF TECHNOLOGY

- HPSS 4.1 - already released
  - DFS
  - MPI-IO
  - File Families
- HPSS 4.1.1 - later this year
  - Shelf tape support
  - non-DCE Client API for AIX and SGI IRIX
  - SGI based data movers
  - Better interface for managing movement of files throughout the hierarchy
    - lock/unlock to prevent purging
    - manual migration and purging capabilities
- HPSS 4.2 - mid 2000
  - Storage subsystems
  - Multiple distributed movers
  - Real-time monitoring
  - Multiple migration/purge policies per Storage Class
  - Import / Export of tapes

LIGO Data Archive -- Off-line at CACR
HPSS (High Performance Storage System)

- IBM System
  - » Used by SDSC, FLNL, others
  - » IBM OS/SW/CPUs; 3rd party robotics
  - » scalable, starting from ~ few TB to 300+TB
- LIGO HPSS specification:
  - » IBM SP2 rack with:
    - – backplane switch
    - – five 4-way PCI nodes in rack
  - » ~500+ GB SSA RAID disk cache to dump tapes
  - » 1 high performance gateway node (HPGN)
  - » 4 Redwood tape drives (=> 4 users)
  - » few X 100GB disk storage for non-HPSS data (metadata on spinning media)
  - » buy tapes as needed
- Use CACR tape robotics/silo for cassette storage
  - » focus LIGO funds on “smart” hardware: tapes drives; disk drives; CPUs; switches;...

HPSS Issues

- New: 20 sites world wide.
  - Centralized code support from IBM ($\sim \$1$M/yr).
  - Development support from DOE labs (~ 25FTEs).
  - Caltech has been running HPSS for 2 years.
- Stability: 95% uptime.
  - Currently precludes operation at sites (Fermilab Run II).
  - Sufficient for off-line operations.
- Complexity: built on top of other large software packages.
  - DCE (distributed computing)
  - Ensina (robust transactions)
- Maintenance: most HPSS sites operate with 2 FTEs.
  - CACR has been successful with 1 FTE.
- Licensing: $\sim \$100$K/yr
  - Sole reason for Cornell Theory Center dropping HPSS.
  - CACR is currently paying all license fees.
- Scope: general solution.
  - Designed for a large number of large files.
- Collaboration: work with CACR.
  - LIGO is actively prototyping LIGO problems on the $1M CACR production system at no cost.
  - The current CACR licensing agreement explicitly allows for a LIGO system at Caltech without a second license fee.

CENTER FOR ADVANCED COMPUTING RESEARCH

HPSS - Who's using it?

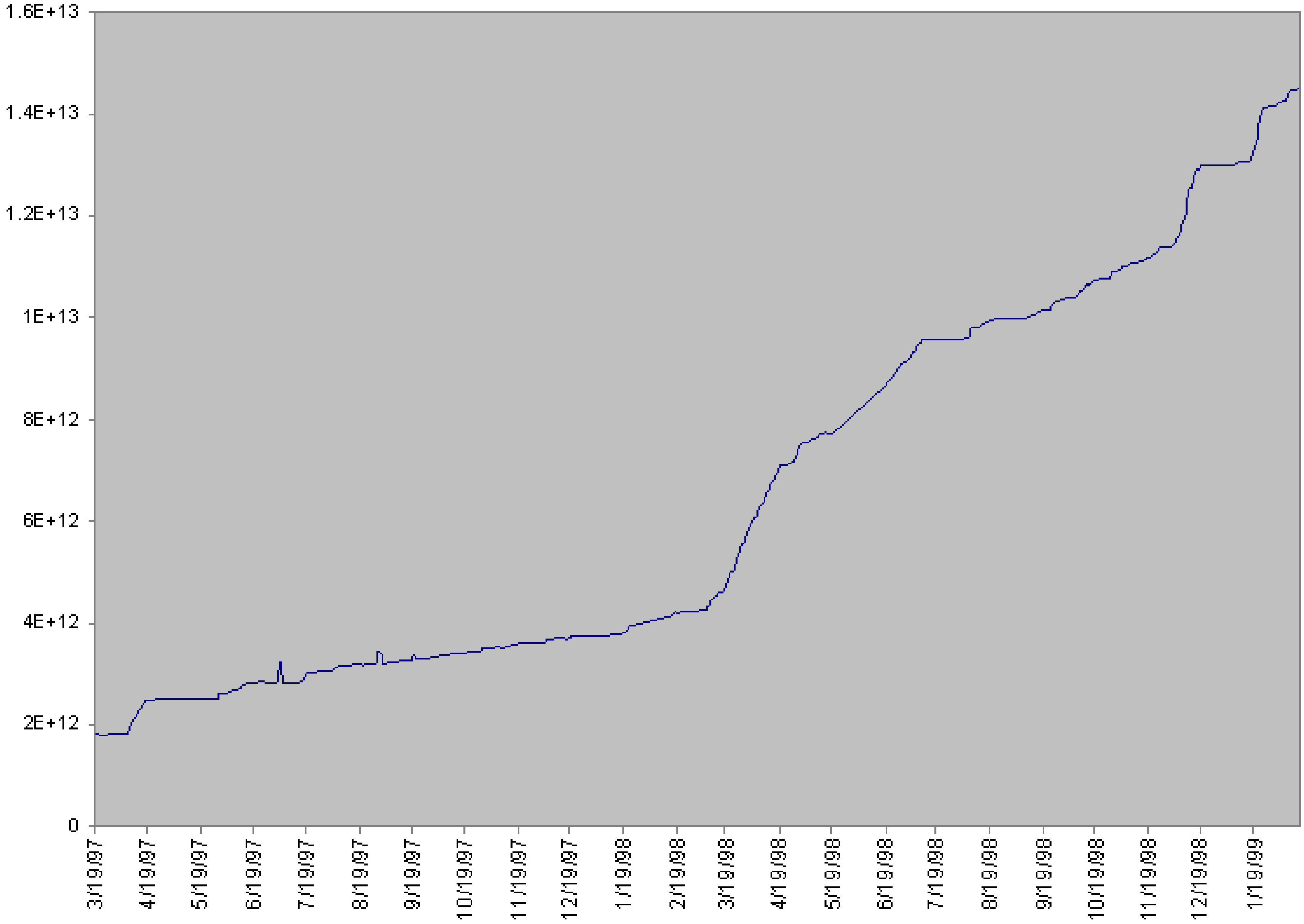
- Lawrence Livermore National Lab
- Los Alamos National Lab
- Sandia National Lab
- Lawrence Berkeley Lab - NERSC
- Oak Ridge National Lab
- Fermi National Accelerator Laboratory
- NASA Langley
- Maui High Performance Computing Center
- Caltech / JPL
- San Diego Supercomputer Center
- University of Washington
- Stanford Linear Accelerator (SLAC)
- University of Maryland
- Commissariat a l'Energie Atomique (CEA)
- Institut National de Physique Nuclèaire et de Physique des Particules (IN2P3
- Brookhaven National Laboratory
- NOAA National Climatic Data Center
- Rechenzentrum University of Stuttgart
- CERN

CALIFORNIA INSTITUTE OF TECHNOLOGY

HPSS Current Usage

CENTER FOR ADVANCED COMPUTING RESEARCH

- 14.5 Terabytes



CALIFORNIA INSTITUTE OF TECHNOLOGY

CENTER FOR ADVANCED COMPUTING RESEARCH

New HPSS Equipment

- Acquired from IBM Grant along with CACR purchase
- Totals approximately $850,000 list
- Includes
  - 5 new nodes for SP2
  - Twenty-two 18 GB disks
  - High Performance Gateway Node
    - high-speed router into the SP-Switch fabric
    - 4 SP-Switch adapters
    - 4 HiPPI adapters
    - 2 OC-12 ATM adapters

CALIFORNIA INSTITUTE OF TECHNOLOGY

New Configuration

CENTER FOR ADVANCED COMPUTING RESEARCH

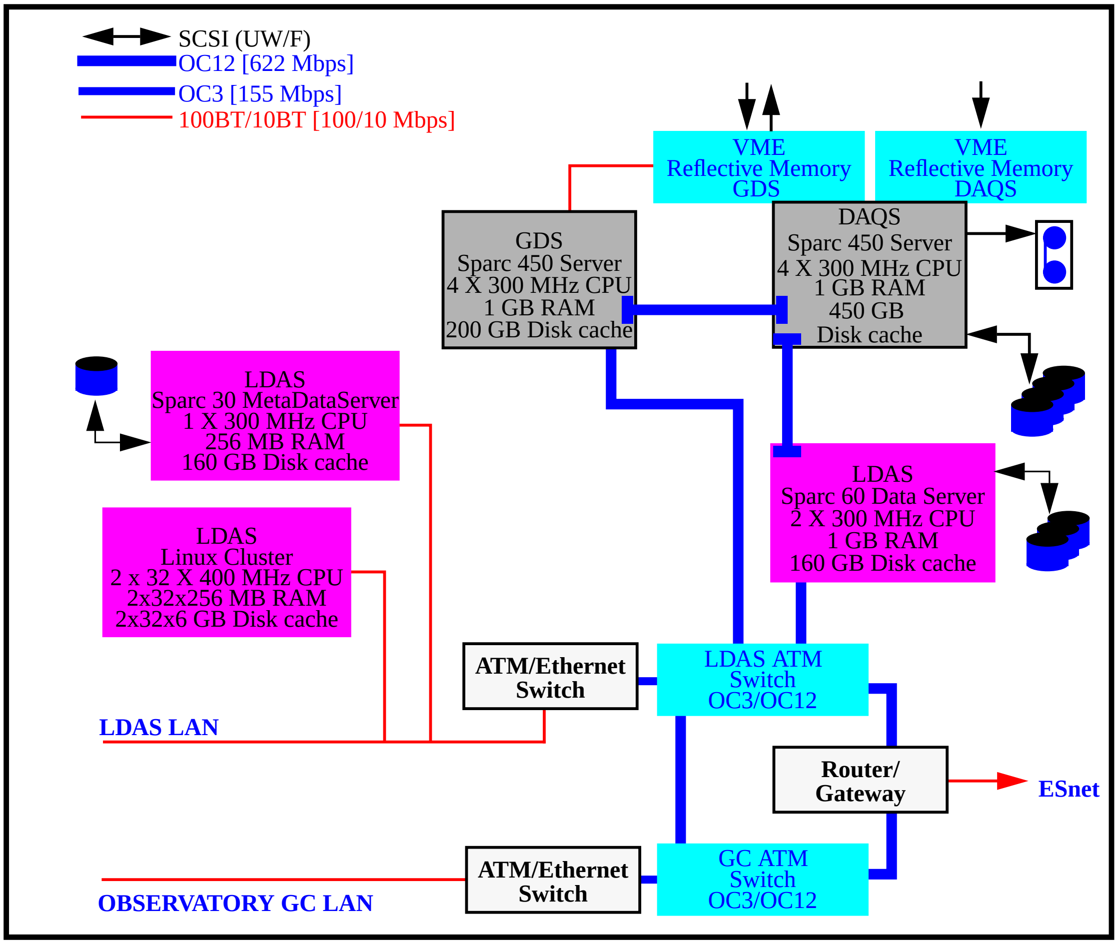
- One of the new nodes will handle all metadata transactions
  - Plan to use ¼ of the existing SSA disk to store metadata
- The remaining four nodes will provide data mover services
  - SSA Disk cache will be configured as RAID 5 sets for protection against disk failure.
- Adding 4410 STK Silo with 4 Redwood drives.
  - Around 6000 cartridge capacity, 50 GB each

CALIFORNIA INSTITUTE OF TECHNOLOGY

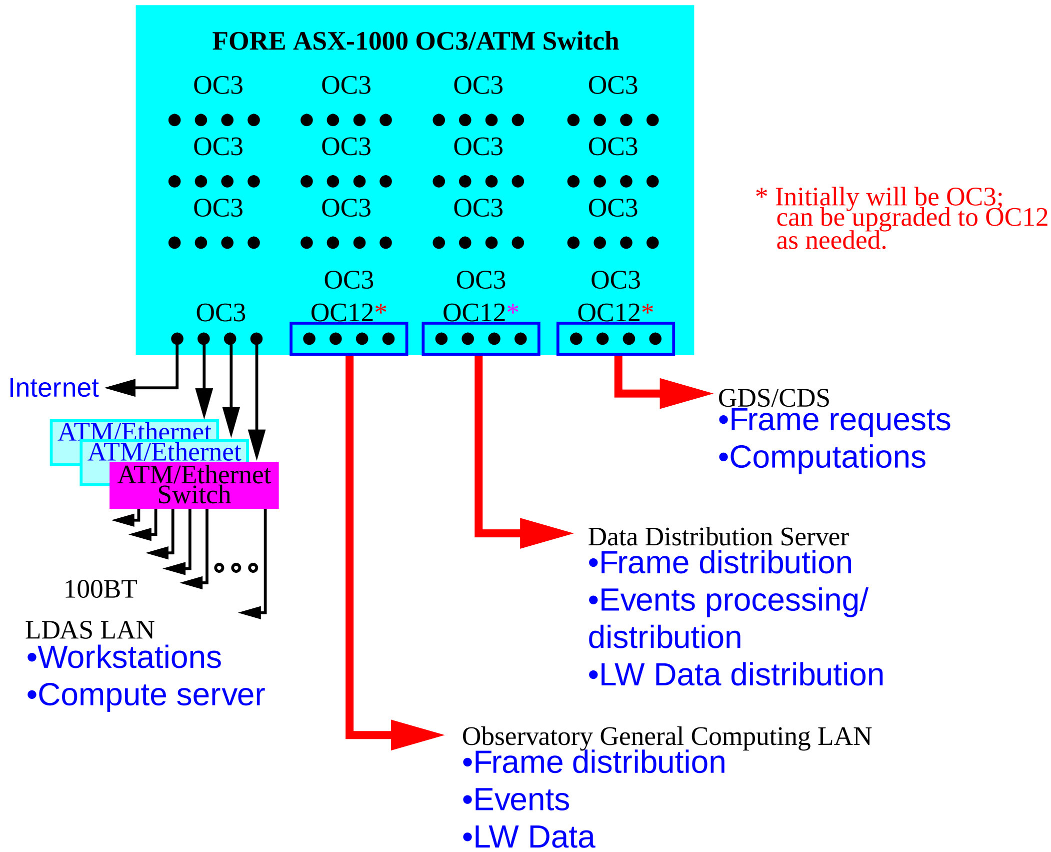
New Equipment - SP2 nodes

- 5 “Silver” wide nodes
- PCI based PowerPC multiprocessors
  - 4 PowerPC CPU’s running at 332 MHz.
  - 1 GB RAM
  - 3 Ultra-SCSI adapters
  - dual SSA RAID 5 adapter
  - SP-Switch adapter (150 MB/s peak)

LDAS Interfaces to Detector Systems



LDAS Interfaces to Detector Systems (cont)



LDAS User Interface

- “LIGO Command Language” (or LCL) based on Tcl scripts (“steering language”);
  - “re-sample channel X at rate R”;
  - “heterodyne channel X with mixer frequency f0”
  - “regress channels {Y,Z,...} from channel X, pass resultant to socket S or process P”;
  - “calibrate channel X (using the calibration data valid at the time channel X was acquired and which are available in the framed data)”
  - ; create a subframe for channels {X,Y,Z,...} from time interval {T1,T2} and re-sampled at rate R”;
  - etc.
- Understood by the managerAPI.
- Commands can themselves be concatenated into more complex sequences.
- Inputs can be via a GUI, browser and command line interfaces

Initial LDAS Functionality requests from a user's perspective (D. Sigg)

- 1. An easy way to take a data run of a few user defined channels and store it in a file:
  - ✓ select channels by name
    - – **define data rate by specifying decimation factors**
    - – **(data rates slower than 1Hz should be supported!)**
  - ✓ multiple users can make multiple requests for their own sets of data(only has to support a modest throughput)
  - ✓ may look at past data
    - – mainly works on current data coming out of the machine
  - ✓ ability to handle data over long time periods; **support gaps in the data stream**
  - ✓ ability to handle system reboots smoothly, i.e. automatic restart
  - ✓ write ascii, binary or frame format
  - ✓ easy to use, i.e. should not require the user to know C/C++, root, Matlab, etc.
- 2. Support of one analysis tool (root) to work on data streams measured *(via frames)*
  - ✓ with the above tool.
    - – **tY and fY graphs**
    - – **simple statistical meausres such as average, mean, etc.**
    - – **FFT cabability**
    - – **low and high pass filters**
  - ✓ simple histograms
  - – **correlations**

LDAS Networks(s):

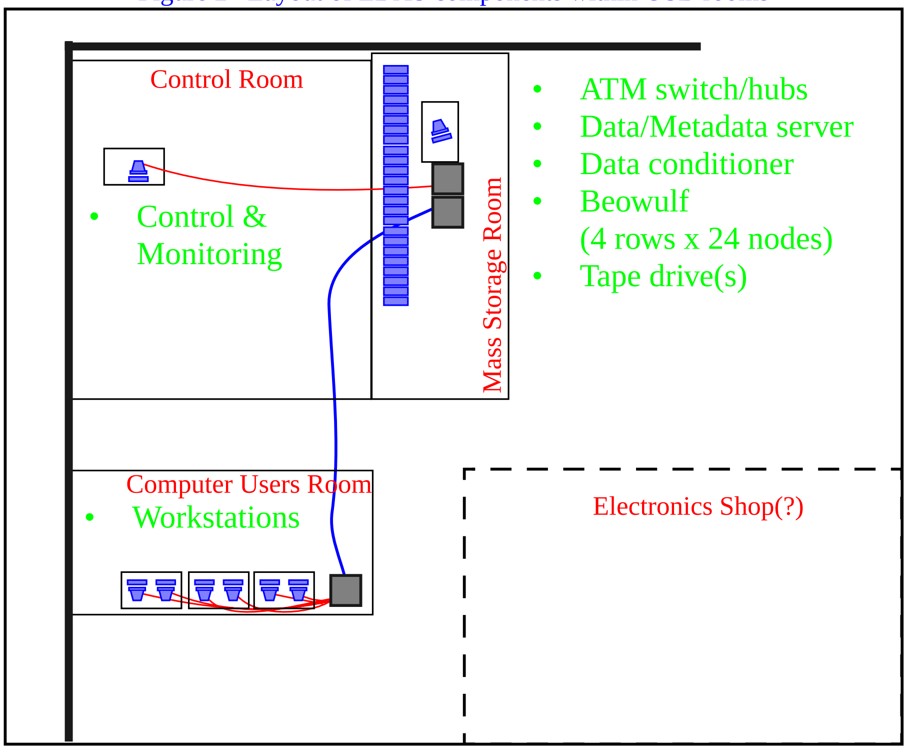
- LANS
  - Each site (LHO, LLO, Caltech, MIT) will have an LDAS LAN, based on ATM, to provide for high bandwidth data transmission between processors.
- WAN:

Table 1 WAN/LAN Connectivity among LIGO Laboratory Sites

Site	***Livingston, LA***	***Hanford, WA***	***MIT***	***Caltech***
Caltech	vBNS/OC3	ESnet (4 X T1) <-> vBNS/OC3	vBNS/ OC3	OC3/ ATM 100BT
MIT	vBNS/OC3	ESnet (4 X T1) <-> vBNS/OC3	100BT OC3(?)	
Hanford, WA	ESnet (4 X T1) <-> vBNS/OC3	OC3 100BT		
Livingston, LA	OC3 100BT			

LDAS Layout at Observatories

Figure 2 Layout of LDAS components within OSB rooms



» Question: If Mass Storage Room is too crowded, can LDAS expand into Electronics room?

LIGO Development Timeline
Major Detector Milestones

LIGO Installation Major Milestones

Milestone	Date	Comment
Vacuum Equipment Complete	**1998.12.08**	**Both sites**
BT Bakeout Complete	**2000.02.21**	**Both sites**
LHO 2km Start	**1998.07.01**	**Begun**
Power Recycled	**1999.10.18**	
Vertex Michelson complete		
LHO 2km IFO complete	**2000.08.28**	**h[f] < 10^{-20}**
LLO 4km Start	**1999.06.01**	
Power Recycled	**2000.02.28**	
Vertex Michelson complete		
LLO 4km IFO complete	**2000.11.20**	**h[f] < 10^{-20}**
LHO 4km Start	**1998.07.01**	**Begun**
Power Recycled	**2000.03.20**	
Vertex Michelson complete		
LHO 4km IFO complete	**2000.10.16**	**h[f] < 10^{-20}**
Design sensitivity	**2001.11.05**	**h[f] < 10^{-21}**
First science run	**2002.01.01**	**3X operation**

LDAS Development Strategy

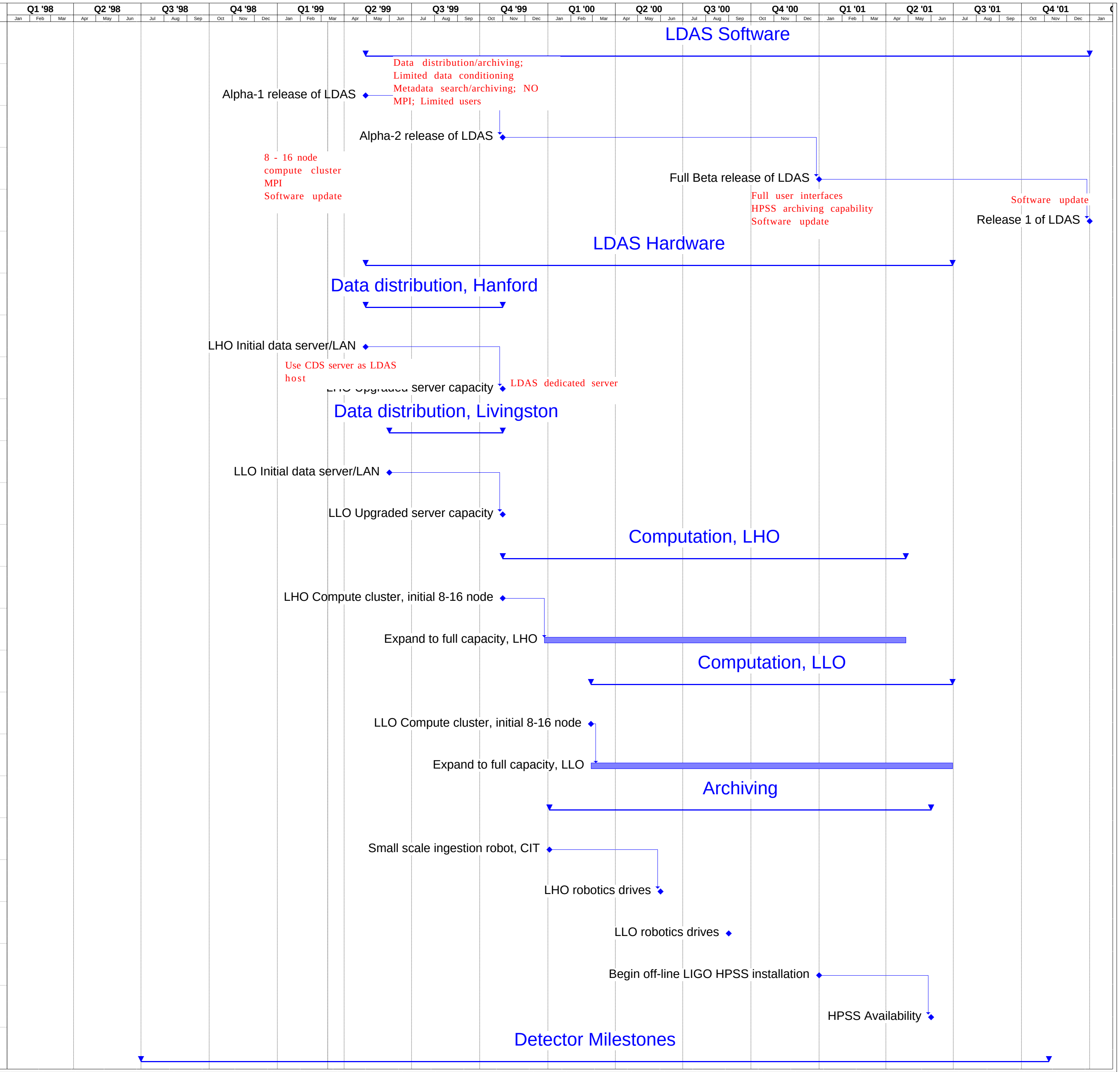
- Focus on initial installation for 2km IFO at Hanford
  - **Initial α release (partial LDAS) in late spring/summer 1999**
    - data distribution/archival
    - limited data conditioning
    - metadata search/archival
    - *no MPI*
    - *limited user base to test system*
  - **Supports initial commissioning of PSL + IO**
- Secondary α release (full LDAS) in late fall 1999:
    - 8 - 16 node beowulf for data analysis (diagnostics)
    - software update, includes MPI
- Full LDAS β release at end of 2000
    - full user interfaces
    - includes HPSS archiving capability
- First official release concurrent with LIGO I science run
- Replicate design at each stage of development as needed for other interferometers.

LDAS Development Strategy

- Full off-line system availability contemporaneous with full LDAS release
  - large linux cluster
  - HPSS + data servers
  - data conditoning
  - data ingestion

CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

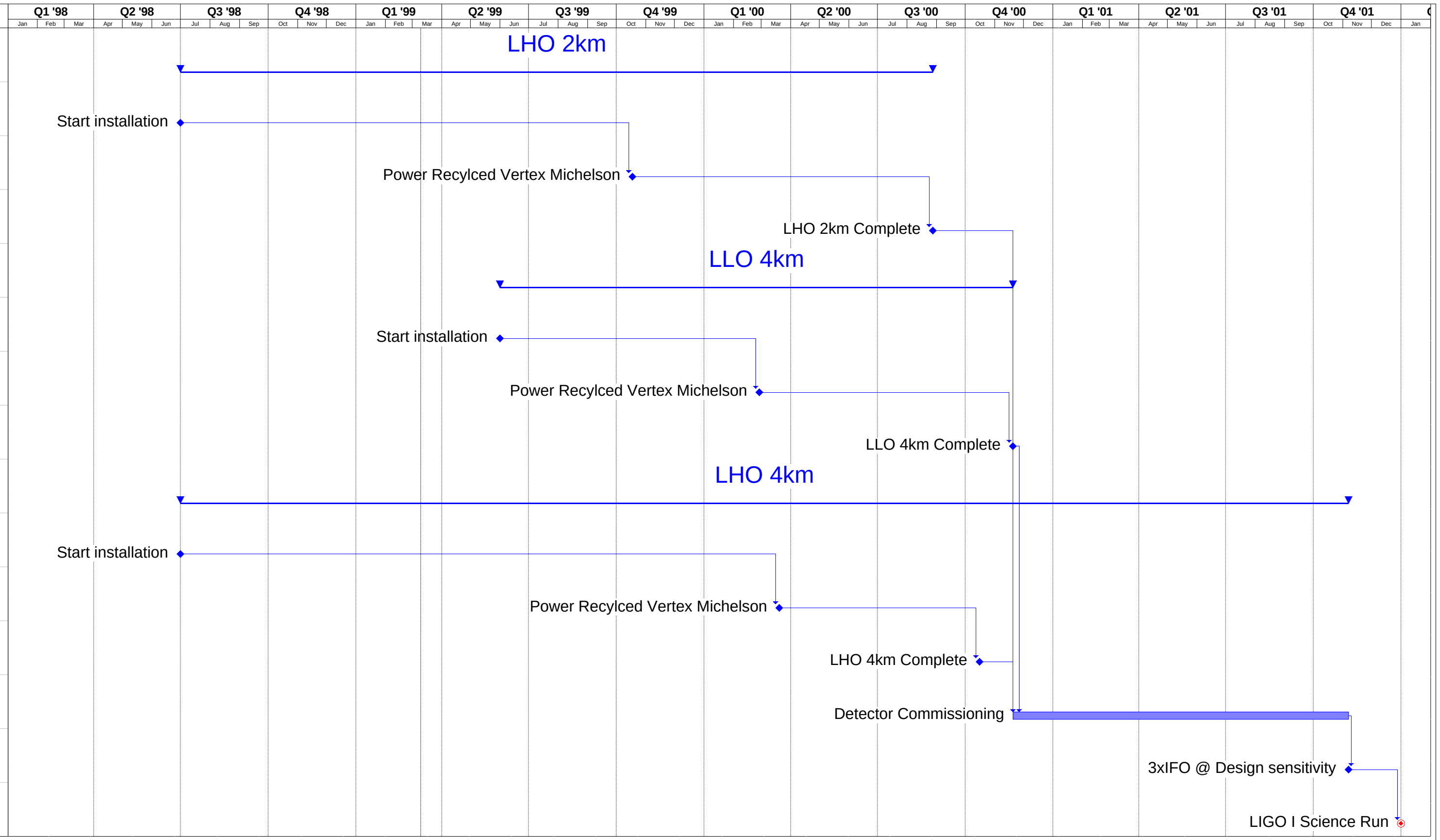
ID	Task Name
1	LDAS Software
2	Alpha-1 release of LDAS
3	Alpha-2 release of LDAS
4	Full Beta release of LDAS
5	Release 1 of LDAS
6	LDAS Hardware
7	Data distribution, Hanford
8	LHO Initial data server/LAN
9	LHO Upgraded server capacity
10	Data distribution, Livingston
11	LLO Initial data server/LAN
12	LLO Upgraded server capacity
13	Computation, LHO
14	LHO Compute cluster, initial 8-16 node
15	Expand to full capacity, LHO
16	Computation, LLO
17	LLO Compute cluster, initial 8-16 node
18	Expand to full capacity, LLO
19	Archiving
20	Small scale ingestion robot, CIT
21	LHO robotics drives
22	LLO robotics drives
23	Begin off-line LIGO HPSS installation
24	HPSS Availability
25	Detector Milestones



Project:
Date: 3/10/99

Task | Task Progress | Critical Task | Critical Task Progress | Milestone | Summary | Rolled Up Task | Rolled Up Critical Task | Rolled Up Milestone | Rolled Up Progress

ID	Task Name
26	LHO 2km
27	Start installation
28	Power Recylced Vertex Michelson
29	LHO 2km Complete
30	LLO 4km
31	Start installation
32	Power Recylced Vertex Michelson
33	LLO 4km Complete
34	LHO 4km
35	Start installation
36	Power Recylced Vertex Michelson
37	LHO 4km Complete
38	Detector Commissioning
39	3xIFO @ Design sensitivity
40	LIGO I Science Run



	Component	Desctription	$K	
1	5N515: On-Line, LHO Hardware - 2 IFOs			
	Data server	SUN 450 & NT for DBMS	60	
	Disk cache, data	500 GB	250	
	Metadata storage	50 GB	25	
	Signal Condtioning System	8 CPUs	80	
	BEOWULF, 2 x 10 GFLOPS	96 nodes ; local disks; RAM; 100BT switches	300	
	Control & Monitoring System	1 CPU	10	
	Network -ATM	OC3/OC12	50	
	On-site tape robotics	10TB, Redwood drives (2x), disk cache	360	*new scope*
1a	Users, LHO			
	Analysis Workstations	10 CPUs	100	
	Misc. SW/Peripheral	Licenses, printers, plotters, scanners, tape drives, local SCSI disks,...	50	
		LHO Subtotal ->		1285
2	5N516: On-Line, LLO Hardware - 1 IFO			
	Data server	SUN 60 & NT for DBMS	40	
	Disk cache, data	250 GB	125	
	Metadata storage	50 GB	25	
	Signal Condtioning System	4 CPUs	40	
	BEOWULF, 10 GFLOPS	48 nodes; local disks; RAM; 100BT switches	165	
	Control & Monitoring System	1@200MFLOPS	10	
	Network -- ATM	OC3/OC12	50	
	On-site tape robotics	10TB, Redwood drives (2x), disk cache	240	*new scope*
2a	Users, LLO			
	Analysis Workstations	7 CPUs	70	
	Misc. SW/Peripheral	Licenses, printers, plotters, scanners, tape drives, local SCSI disks,...	50	
		LLO Subtotal ->		815

Component	Desctription	$K	
3 5N513: Off-Line, CIT			
LIGO Laboratory/LSC			
Hardware - HPSS			
SP2	Frame + backplane	60	
4-way PCI Power3 nodes	5 nodes	240	
SSA Disk Cache	1000 GB	192	
Gateway node	HPGN	86	
Tape heads	Redwood, 4 heads	480	
Hardware - Metadata			
DBMS server	NT CU	8	
Disk	350GB	350	
Hardware- computing			
Template disk cache	500 GB	250	
BEOWULF 3 x 10 GFLOPS	128 nodes; local disks; RAM; 100BT switches	450	
Postprocessing Workstations	4 CPUs	40	
Network -- ATM	OC3/OC12	60	
	Subtotal, LIGO Lab. ->		2216
CIT-specific			
Analysis Workstations	12	120	
Misc. SW/Peripheral	Licenses, printers, plotters, scanners, tape drives, local SCSI disks,...	100	
Prototyping hardware	RAID	42	
	Servers, 2 CPUs	24	
	BEOWULF, 8 node	25	
	Subtotal, Prototyping + CIT-Specific ->		311
	CIT Subtotal ->		2527

	Component	Desctription	$K	
4	5N514: Off-line, MIT			
	Analysis Workstations	10 CPUs	100	
	Disk cache	200	150	
	Network -- ATM	OC3/OC12	30	
	Misc. Networking @ MIT	ATM/OC3	40	
	Misc. SW/Peripheral	Licenses, printers, plotters, scanners, tape drives, local SCSI disks,...	100	
		MIT Subtotal ->		420
		Grand Total, Hdwre ->		5047