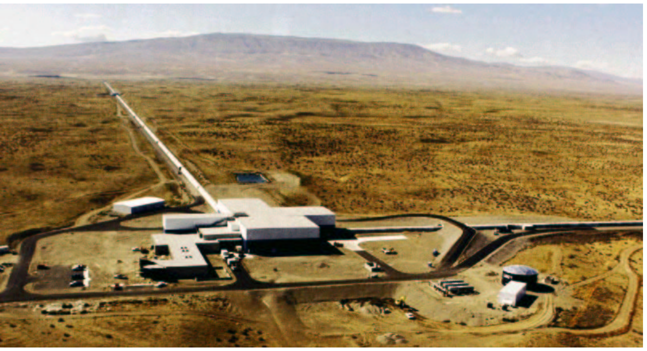
LIGO Scientific Collaboration Grid

Patrick Brady

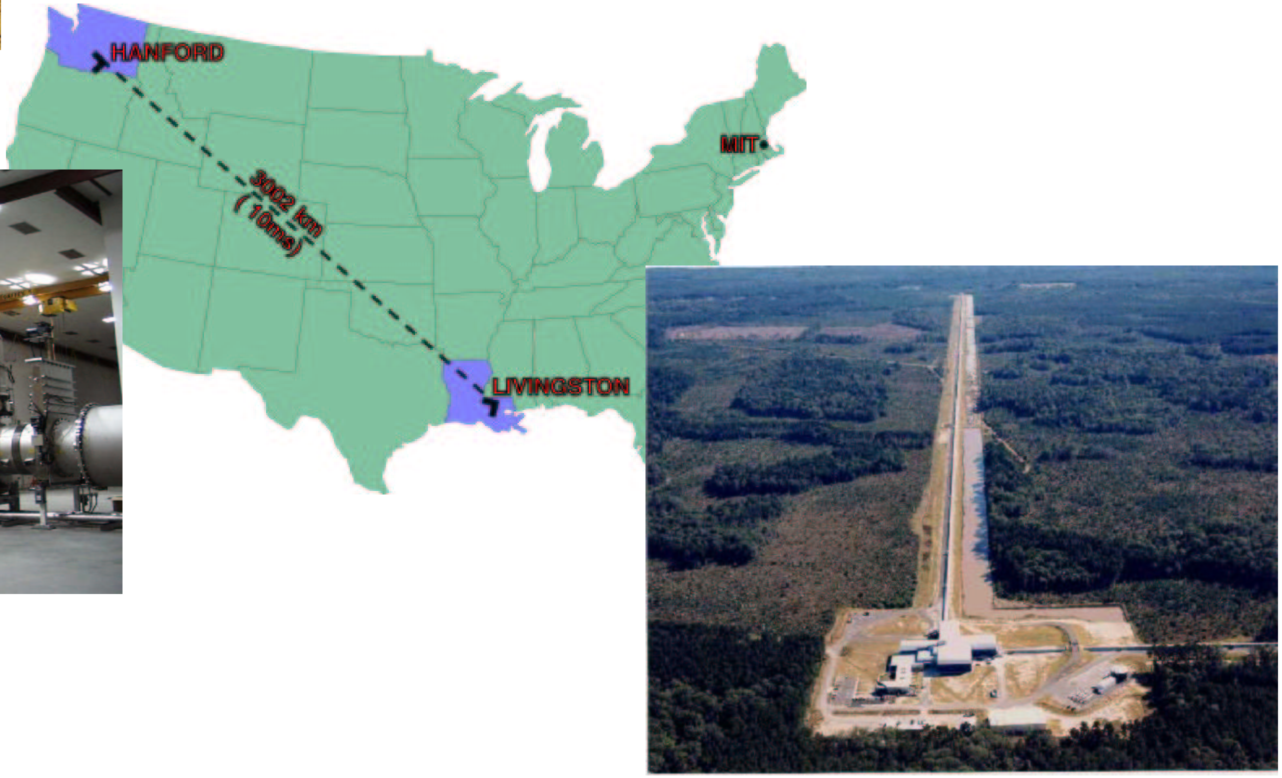
University of Wisconsin-Milwaukee

LIGO Scientific Collaboration

LIGO Project



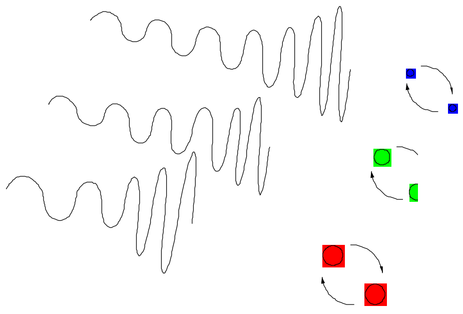
LIGO is a National Science Foundation funded project to detect gravitational waves and initiate gravitational-wave astronomy

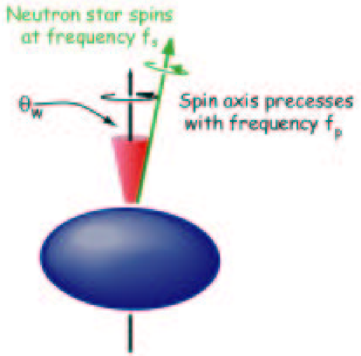


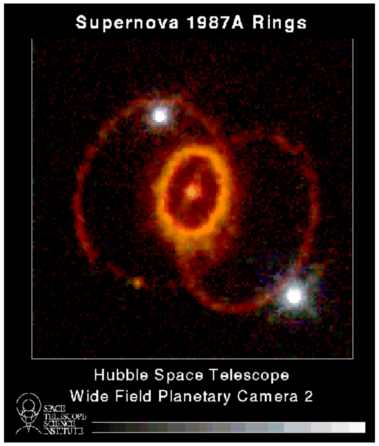
G040235-00-Z

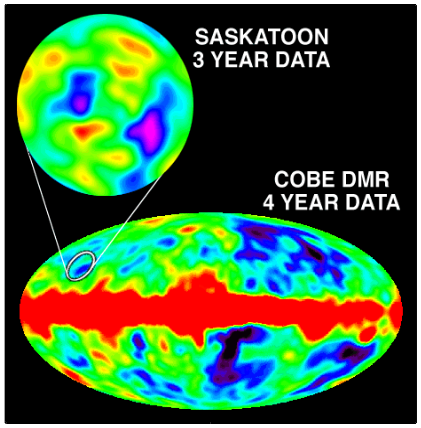
Gravitational wave sources

- Compact binary systems
  - Neutron star inspiral
  - Black hole inspiral/merger
  - Large computational burden
  - On the fly triggers to astronomers
- Neutron star birth
  - Supernova explosions
  - Easy computation
  - On the fly triggers to astronomers
- Spinning neutron stars
  - Need months of integration time
  - Infinite computational burden
- Stochastic background
  - Big bang & other early universe

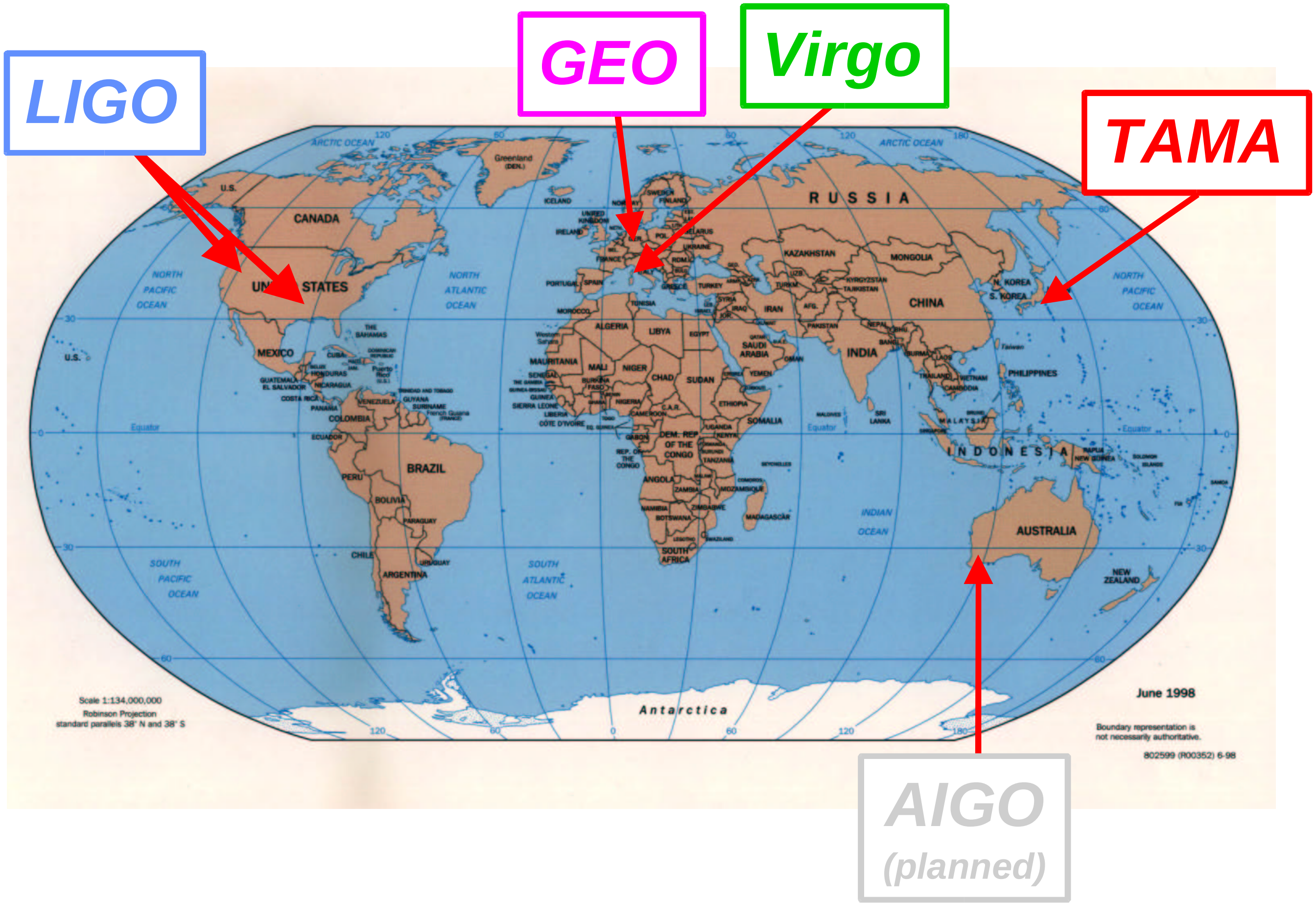



Neutron star spins at frequency f_s
θ_W
Spin axis precesses with frequency f_p


Supernova 1987A Rings
Hubble Space Telescope
Wide Field Planetary Camera 2
SPACE TELESCOPE SCIENCE INSTITUTE



International Network of Detectors



LIGO

GEO

Virgo

TAMA

AIGO
(planned)

Analysis is computation and data intensive

- Revealing the full science content of LIGO data is a computationally and data intensive challenge
  - » LIGO interferometers generate about 10 MB/s or almost 1 TB/day
- Several classes of data analysis challenges require large-scale computational resources
  - » FFT data segment
  - » Choose template (use physical parameters)
  - » Filter data segment using FFT
  - » Repeat again and again and again…
  - » ..... can go to 100's of TFlops
- Realizing the full science content of LIGO data is computationally and data intensive

LSC Data Grid Tier I/II sites

Tier I @ CALTECH

- 2548 GHz CPU
- 60 TB RAID storage
- 780 TB Tape Storage

MIT

UNIVERSITY of WISCONSIN UWMILWAUKEE

- 296 GHz CPU
- 64 TB storage (commodity IDE)
- OC-12 (622 Mbps) to Abilene

PENNSTATE

- 889 GHz CPU
- 34 TB RAID 5 storage
- OC-12 (622 Mbps) to Abilene

Max-Planck-Institut für Gravitationsphysik Albert-Einstein-Institut

- 670 GHz CPU
- 40 TB storage (commodity IDE)
- Fast Ethernet to G-WiN (... Abilene)

CARDIFF UNIVERSITY PRIFYSGOL

- 272 GHz CPU
- 18 TB storage (RAID 5 & IDE)
- GigE to SuperJANET (… Abilene)

THE UNIVERSITY OF BIRMINGHAM

- 508 GHz CPU
- 18 TB storage (RAID5 & IDE)
- 10 Gbps to SuperJANET (… to Abilene)

In a nutshell…

- Hardware at 9 sites on two continents (and growing…)
- LIGO data at Hanford and Livingston sites
  - » Replicate data sets for analysis to the centers where hardware & storage available using Lightweight Data Replicator (LDR)
- LSC scientists at 41 institutions
  - » need rational, scalable, secure way for people to leverage available hardware
- People resources for data analysis are geographically distributed
- Emerging Grid Computing technology helps put data + hardware + people together for more science

Lightweight Data Replicator (LDR)

- Replicate data files quickly, securely, robustly to sites in a DataGrid
  - » Python – glue to hold it all together and make it robust
  - » PyGlobus
    - – Python API for Globus toolkit (2.x)
    - – Keith Jackson’s group at LBL
- To date > 50 TB replicated to 9 sites in 3 continents
  - » Rates of 20 MB/s sustained for CIT to UWM transfers
- LDR 0.6.0 aims to be free of LSC specific code
  - » http://www.lsc-group.phys.uwm.edu/LDR
  - » Scott Koranda and Brian Moe

LSC Requirements for Data Analysis System

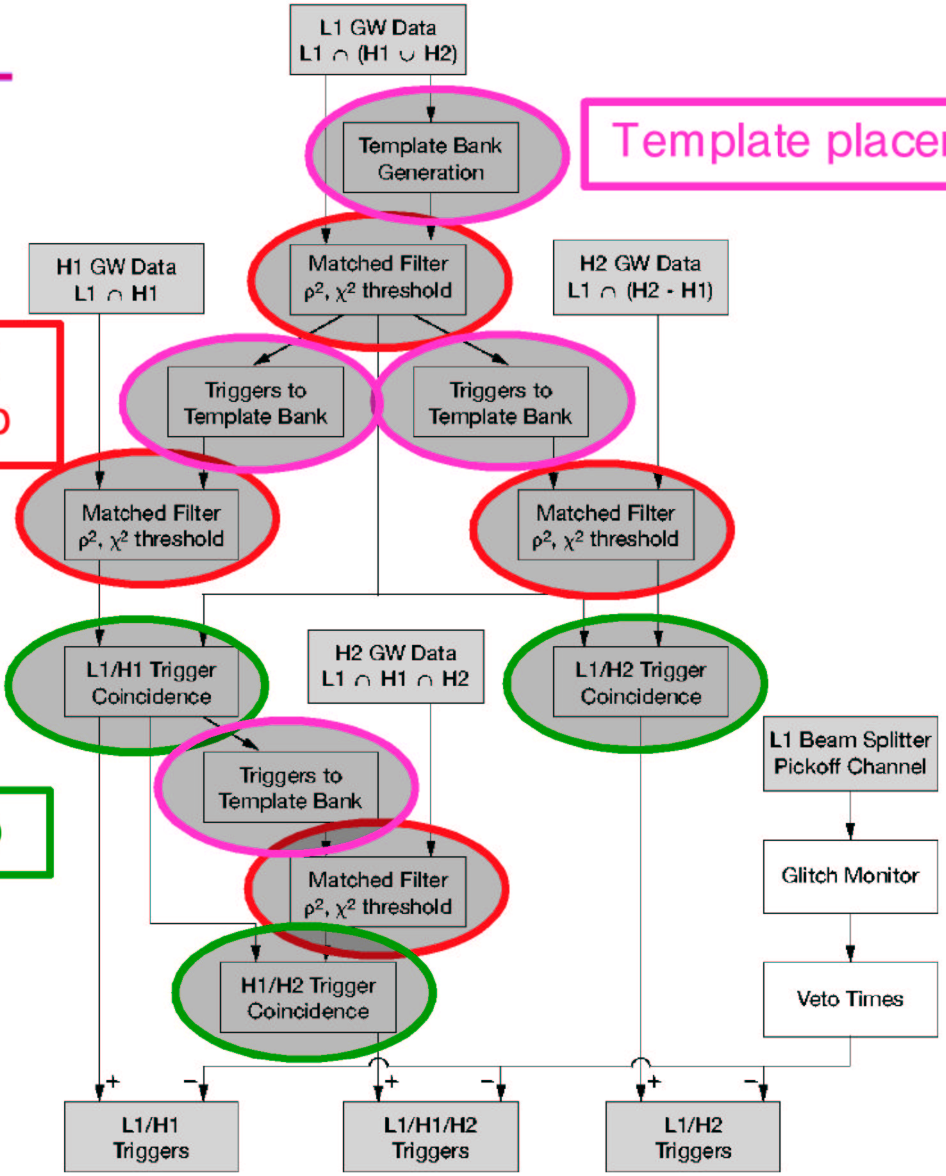
- Goal is to automate entire analysis chain
  - Analysis in similar time as data acquisition
  - Remove error prone user interaction
- Requirements
  - Allow easy construction of complex work flow
  - Simple reusable infrastructure
  - Easy to debug
  - Provide flexible pipeline for testing and tuning
    - computational cost of simulation and tuning
- Other
  - Need to implement a real-time system .....
  - .... alongside the downstream analysis system

LIGO Example: Inspiral search pipeline

Template placement step

Data preparation and matched Filtering Step

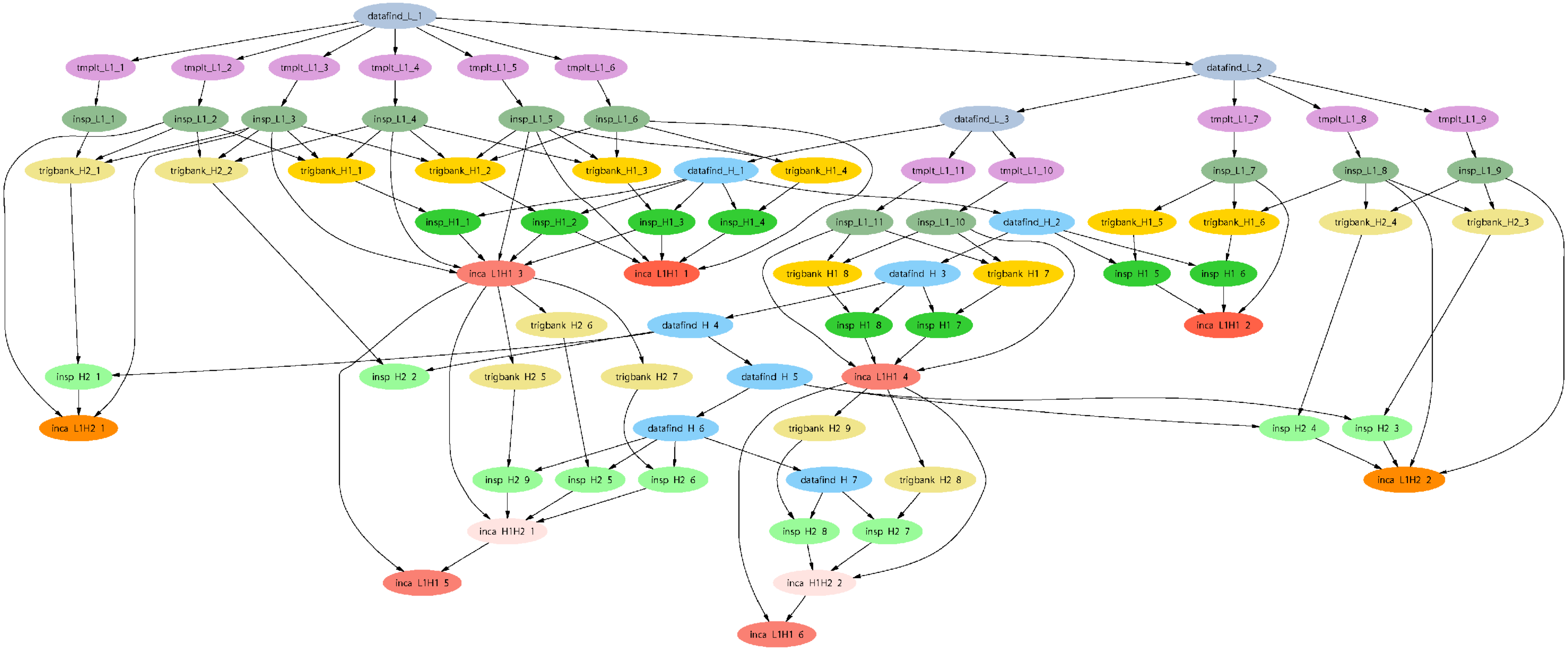
Coincidence Step



Pipeline Implementation

- Condor to manage job submission
  - » Works very well
- Data management is critical
  - » Data discovery and movement
- LALApps code to execute components of pipeline
  - » LALApps contains programs to do simple tasks
  - » Use L(SC) A(lgorithm) L(ibrary) for gravitational-wave functions
  - » Simple modules to construct pipeline scripts
- Pipeline → Directed Acyclic Graph (DAG)
  - » Condor DAGman to manage execution of pipeline

Inspiral DAG



- Inspiral search node runs for about 1 hour on 30 mins of data
- 30 days of data analyzed about 30 times during past 3 months
- Inspiral Analysis Group is very happy with Condor & Condor DagMan
- Other LIGO Apps are being ported

Novel Use of Condor Checkpointing to exploit LSC Grid

- Idea: use Condor's checkpoint mechanism to farm the CPU intensive part out to the LSC Grid by creating a checkpoint file after the data is read in.
  - » The executable and checkpoint image can be sent out onto the grid using the condor Globus universe.
  - » All the data that is needed is contained in the checkpoint file so the remote cluster does not need access to the LIGO data.
  - » The code performs the filtering and writes out the trigger file on the remote file system which can fetched back ``home`` for post-processing.

Novel Use of Condor ....

- Proof of principle (6 April '04) – Duncan Brown, UWM
  - Run the inspiral code at UWM in the vanilla universe so Condor doesn't interpret the checkpoint as a standard universe eviction.
  - After the inspiral code call ckpt_and_exit() Condor reports an abnormal termination with code 12, run a post script that checks for the existence of the checkpoint file. If it's there the post script returns sucess so the DAG can continue.
  - Submit a job to the Globus universe that schedules the job at PSU. The executable and checkpoint file are staged as input & the job runs.
  - The job writes output file to a directory at PSU. Since this isn't in the gram scratch directory, Condor can't retrieve it, so have to run a post script to fetch it (using globus-url-copy) and then delete the output file by globus-running rm.

Conclusions

- The LSC Grid is coming along
  - » Hardware is deployed at 9 computational centers on two continents
  - » Lightweight Data Replicator (Koranda et al) moves data
  - » Individual centers running Condor as batch scheduler
- End to end analyses have been completed
  - » Condor and Condor DAGMan indispensable for inspiral analysis
  - » Developed LSC specific infrastructure to build LSC analysis DAGs
  - » Other analyses are being ported to use this infrastructure
  - » Still have some teething problems educating users to Grid
- Coming soon
  - » Use of Pegasus/Chimera to LSC Grid exploitation
  - » Condor checkpoint method to simplify aspects of data management
  - » Need to develop real-time analysis (COD or Bologna Batch System)