GPUs in Gravitational-Wave Data Analysis

Drew Keppel[1,2]
for the LIGO Scientific Collaboration
& Virgo collaboration

3RD ASPERA Computing and Astroparticle Physics Workshop
Hannover, Germany
2012-05-03

1: Albert-Einstein-Institut, Max-Planck-Institut für Gravitationsphysik
2: Leibniz Universität Hannover

LIGO-G1200381

LSC And Virgo GPU Contributors

- Abilene Christian University
  - Josh Willis
- AEI
  - Carsten Aulbert, Oliver Bock, Tito Dal Canton, Heinz-Bernd Eggenstein, Drew Keppel, Badri Krishnan, Bernd Machenschalk, Karsten Wiesner
- Canadian Institute of Theoretical Astrophysics
  - Kipp Cannon
- Eötvös Loránd University
  - Máté Nagy
- Syracuse University
  - Duncan Brown, Alexander Nitz
- Tshinghua University
  - Zhihui Du, Yuan Liu
- Università degli Studi “Carlo Bo” di Urbino
  - Riccardo Sturani
- The University of Birmingham
  - Ben Aylott
- University of Western Australia
  - David BLair, Shin Kee Chung, Amitava Datta, Shaun Hooper, Linqing Wen
- Wigner Research Centre for Physics
  - Debreczeni Gergely

GW Interferometers

LIGO Hanford Observatory

GEO600

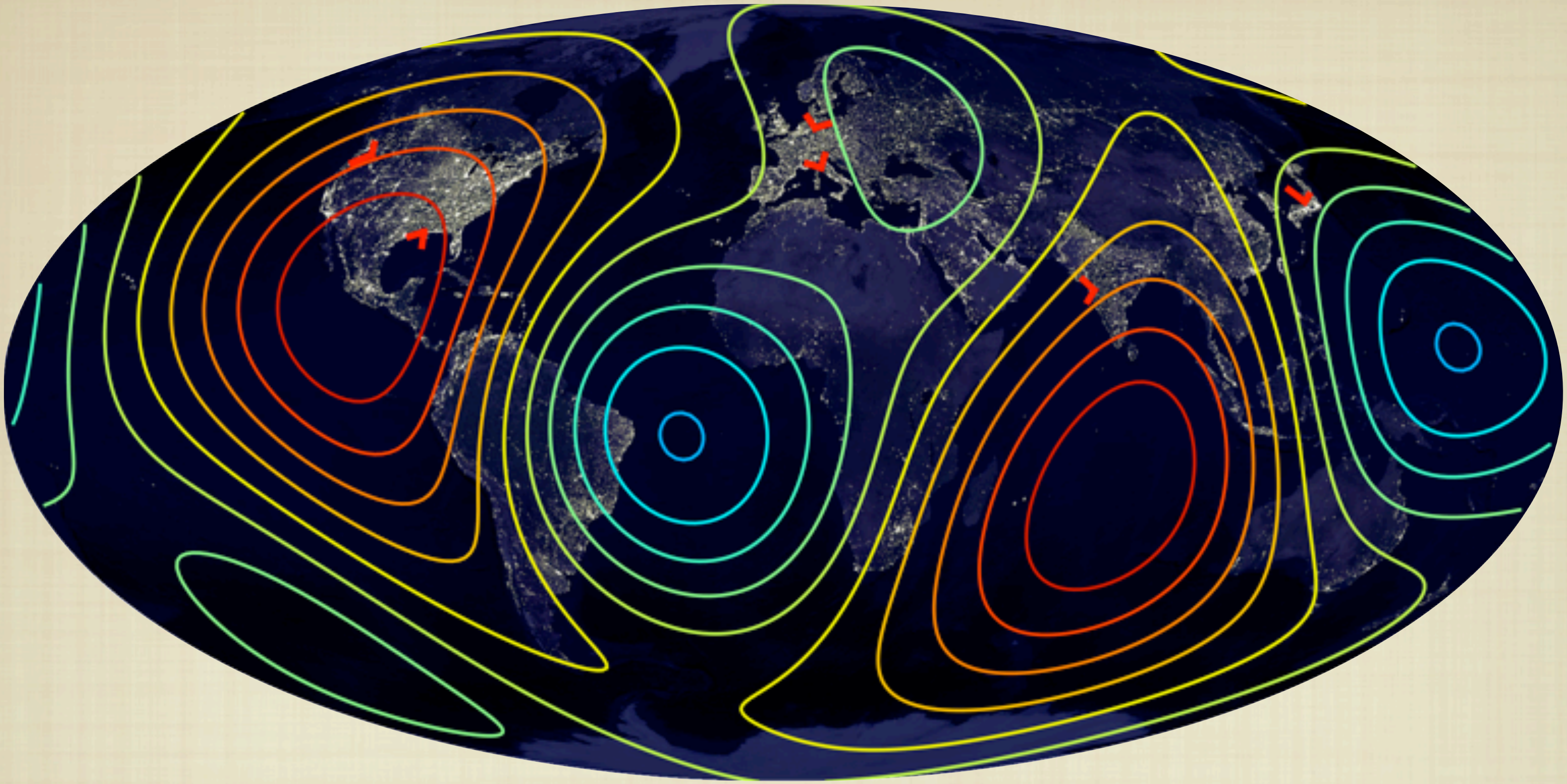
LIGO Livingston Observatory

Virgo

- Pumped up Michelson Interferometers at their core
  - sensitive to differential displacements

Photo Credits: LHO and LLO images courtesy of LIGO Lab. Virgo image courtesy of Eurelios . GEO600 image courtesy of Albert Einstein Institute Hannover and Deutsche Luftbild, Hamburg.

GW Detector Network

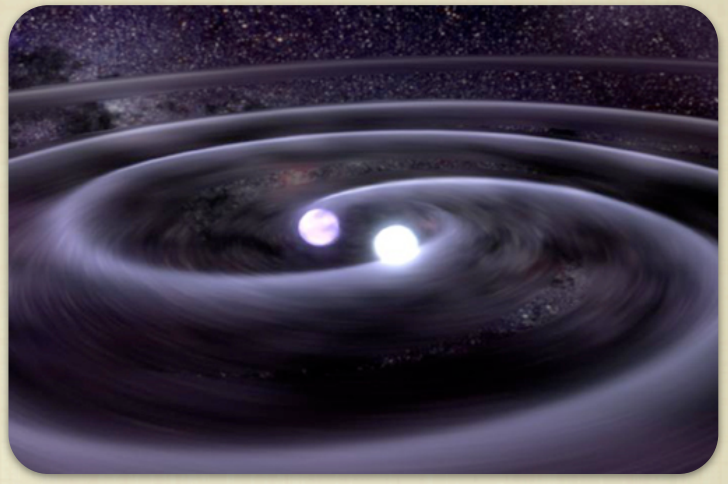


- Era of the first generation of interferometric detectors has ended

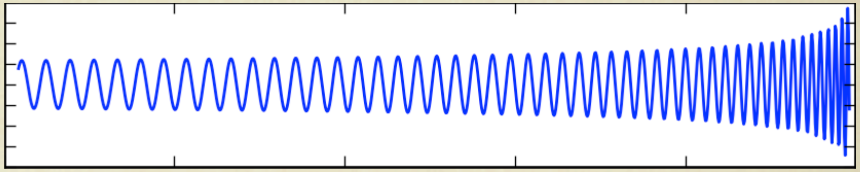
- upgrades underway to second generation designs
    - Exception: GEO600 operating in "astrowatch" mode

Photo Credit: Satellite data courtesy Marc Imhoff of NASA GSFC and Christopher Elvidge of NOAA NGDC. Earth image by Craig Mayhew and Robert Simmon, NASA GSFC.

GW Signals From Inspiraling Compact Binaries



- neutron star and/or black hole binaries
- Most promising source for second generation detectors
  - 10s of detections expected each year
- Characterized by a "chirping" signal



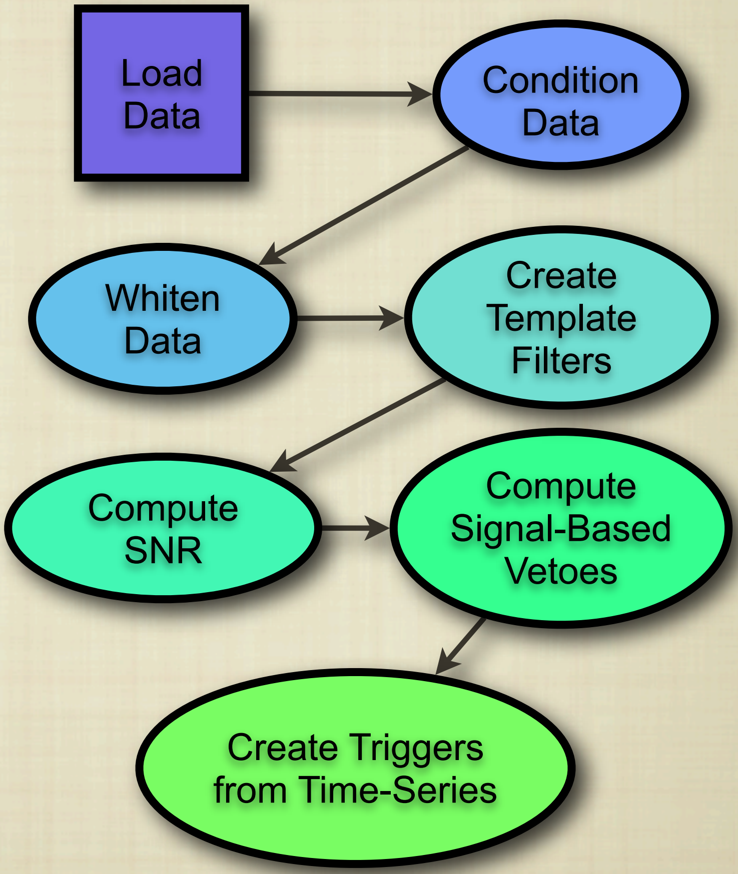
$$\Delta t = (5\mathcal{M})^{-5/3}\left(\frac{5}{8\pi f}\right)^{8/3}$$

Frequency Band (Hz)	Duration (s)
40 - 2048	~25
10 - 2048	~975

Illustration Credit: Dana Berry of NASA GSFC.

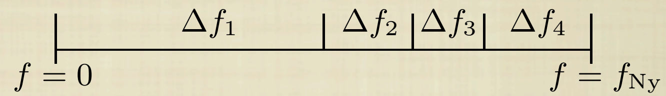
Searching For Inspiral Signals: Single Detector Method

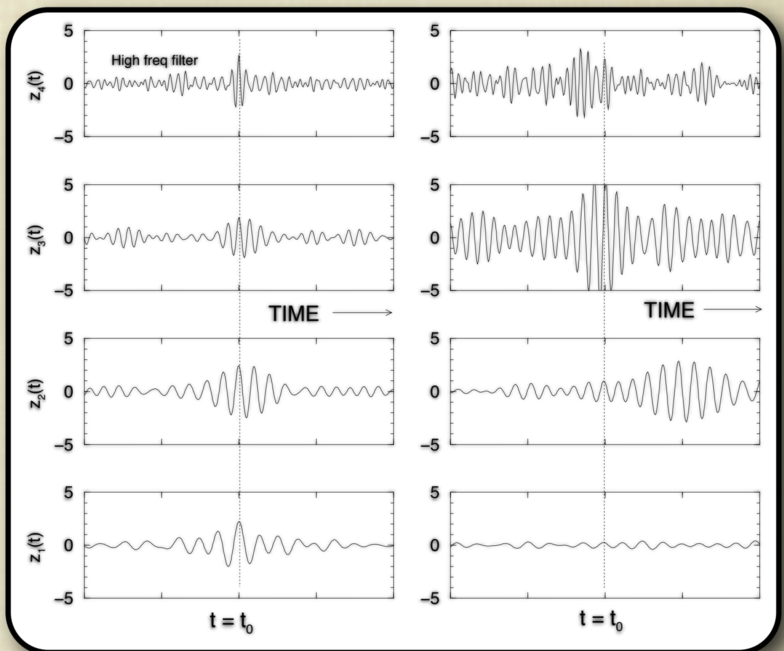
- Matched filter used to search for known waveforms
  - Overlap-Save algorithm used to reduce computational cost at expense of latency
  - SNR = 1 IFFT
  - Non-gaussian data requires something more
- Signal-based vetoes reduce effects of non-Gaussian detector glitches
  - Chisq = 16 IFFTs



Searching For Inspiral Signals: Single Detector Method

- Matched filter used to search for known waveforms
    - Overlap-Save algorithm used to reduce computational cost at expense of latency
    - SNR = 1 IFFT
    - Non-gaussian data requires something more
- Signal-based vetoes reduce effects of non-Gaussian detector glitches
    - Chisq = 16 IFFTs



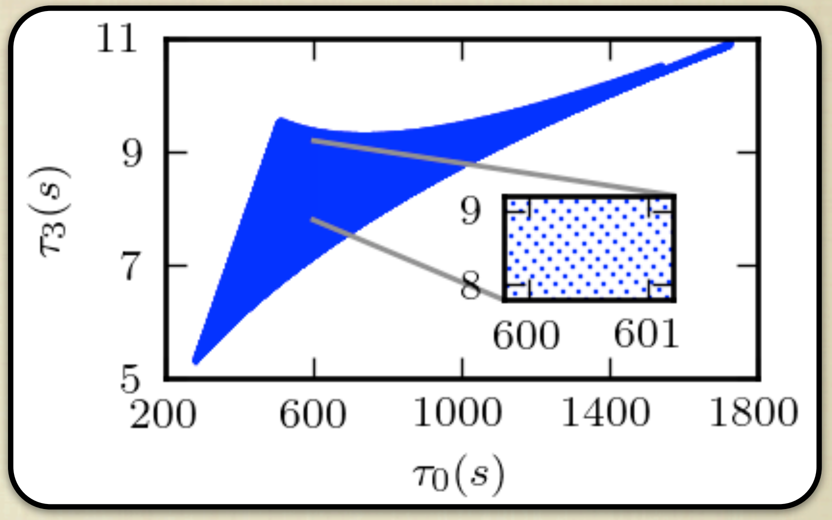


$$\chi^2 = \sum_{i=1}^{16} |\rho_i - \rho/16|^2$$

Figure Credit: B. Allen, Phys. Rev. D **71**, 062001 (2005)

Searching For Inspiral Signals: Single Detector Cost

- Full BNS Search in Second Generation Detectors
  - template banks made up of few x 10^5 waveforms
  - waveforms have few x 10^6 samples
  - ~100 GFlops needed to produce SNR data in real time (with a latency of ~15 minutes)
  - ~10x more including Chisq veto calculation
- Smart algorithm designs can reduce computational cost and memory footprint



Speed of 2^{21}-length FFTs	# of units for real time processing
ATLAS CPU Core @3GFlops	~10
NVIDIA C2050s @200GFlops	~1 *

*Neglecting memory restrictions

GPU Acceleration Efforts: The FFT & Chisq Calculation

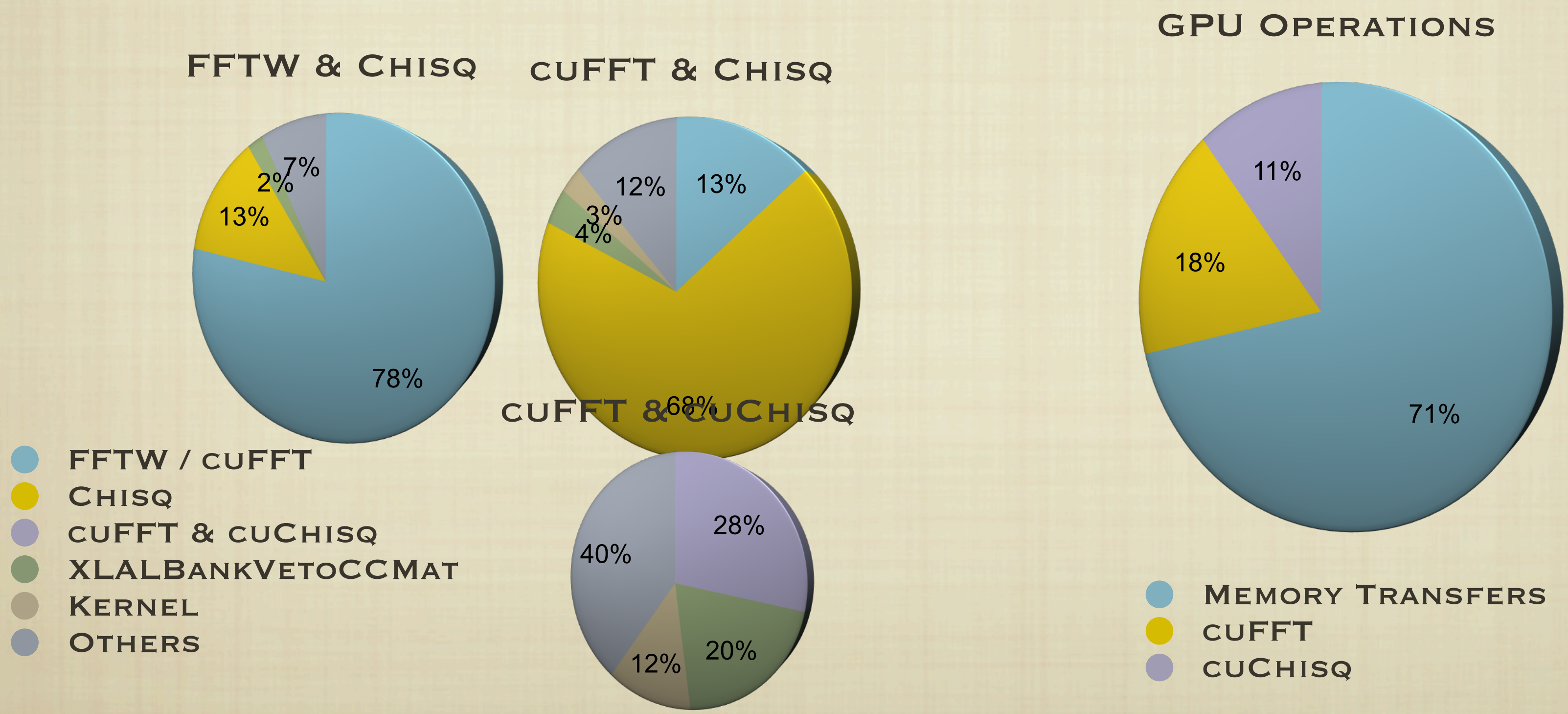
- Porting just the FFT routines of "lalapps_inspiral"
  - Limited by the transferring data to and from GPU for each FFT
- Porting also the Chisq calculation of "lalapps_inspiral"
  - reduces number of transfers to and from GPU by 8x

lalapps_inspiral w/	Execution Time (seconds)
CPU FFT	480
CUDA FFT	164
CPU FFT and CPU CHISQ	1530
CUDA FFT and CPU CHISQ	895
CUDA FFT and CUDA CHISQ	188

GPU Acceleration Efforts:

"lalapps_inspiral" Limitations

- Limited by:
  - other CPU operations
  - memory transfers between to and from GPU
- Restructuring code necessary for obtaining full GPU benefits

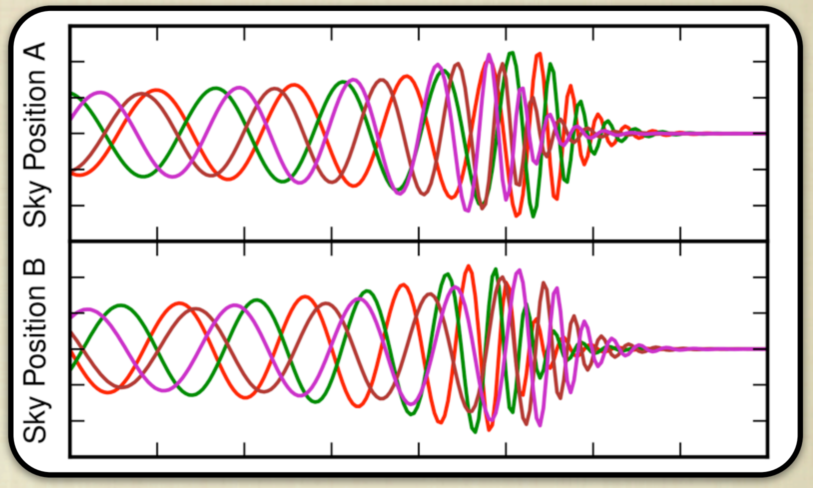


GPGPU Acceleration Efforts: GWtools

- Based on OpenCL for portability
- Achieved theoretical speedup for SNR using Overlap-Save algorithm
- Prototyping code for new algorithms
  - waveform generation
  - SNR calculation
  - signal-based vetoes
  - maximization and trigger production
  - trigger clustering
- Kernels developed will be incorporated into other toolkits

Searching For Inspiral Signals: Coherent Search

- Coherently combine data from detector network
- Each sky location requires:
  - N time-shifts and IFFTs
  - recombination
- ~10^3-10^4 sky locations for network of N advanced detectors
- Hundreds of GFlops become few PFlops
  - A hierarchical approach could reduce cost



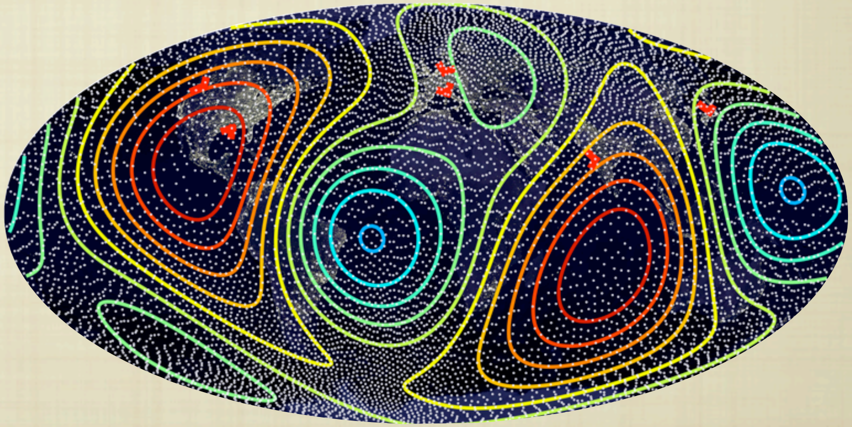
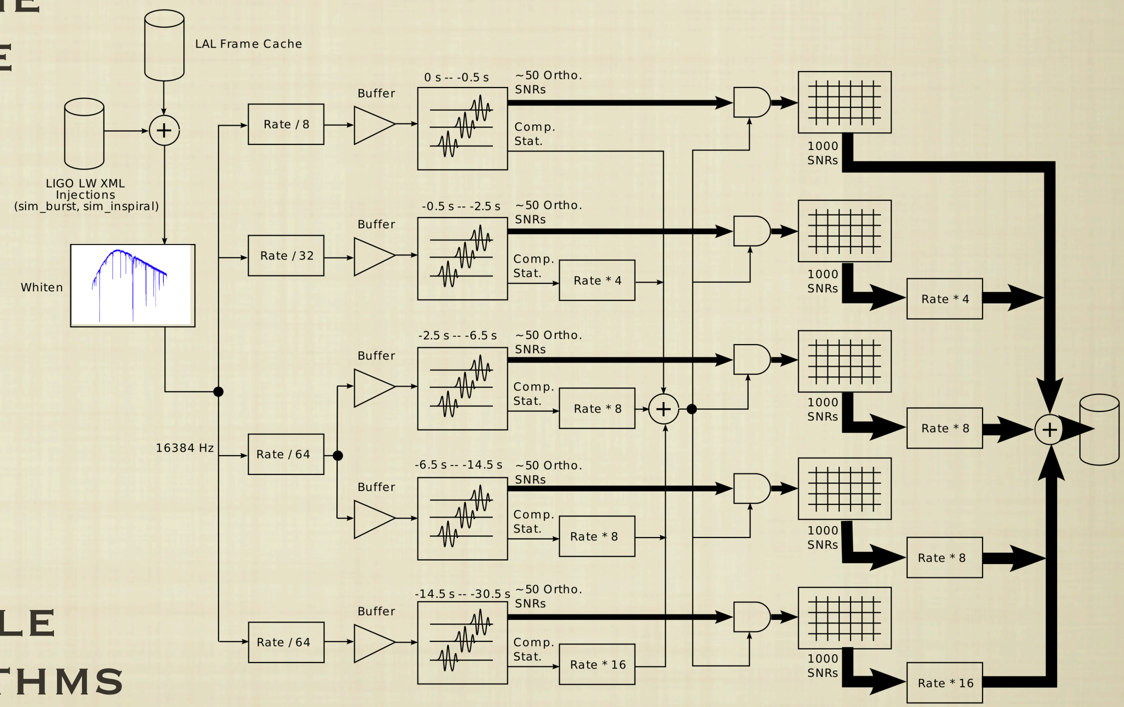


Photo Credit: Satellite data courtesy Marc Imhoff of NASA GSFC and Christopher Elvidge of NOAA NGDC. Earth image by Craig Mayhew and Robert Simmon, NASA GSFC.

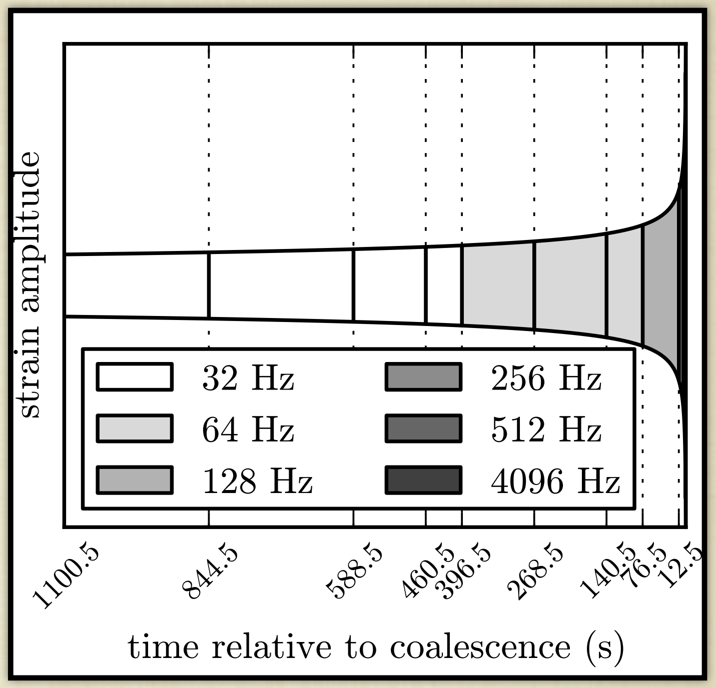
Low-Latency Pipelines

- High payoff could come from prompt telescope pointing
  - reduce latency as much as possible
- Ex: GstLAL based on Gstreamer multimedia framework and LSC's Algorithms Library
- filter engine swappable with different algorithms



GPU Acceleration Efforts: Low Latency With LLOID

- Decreased latency means increased computational cost
- Computational cost reductions:
  - Multirate filtering
    - also implemented within MBTA
  - Significance-Based filtering
- investigating benefit of porting individual tools to GPU



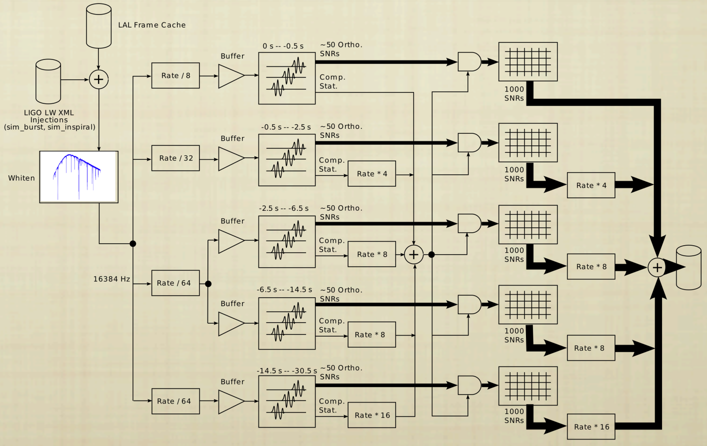
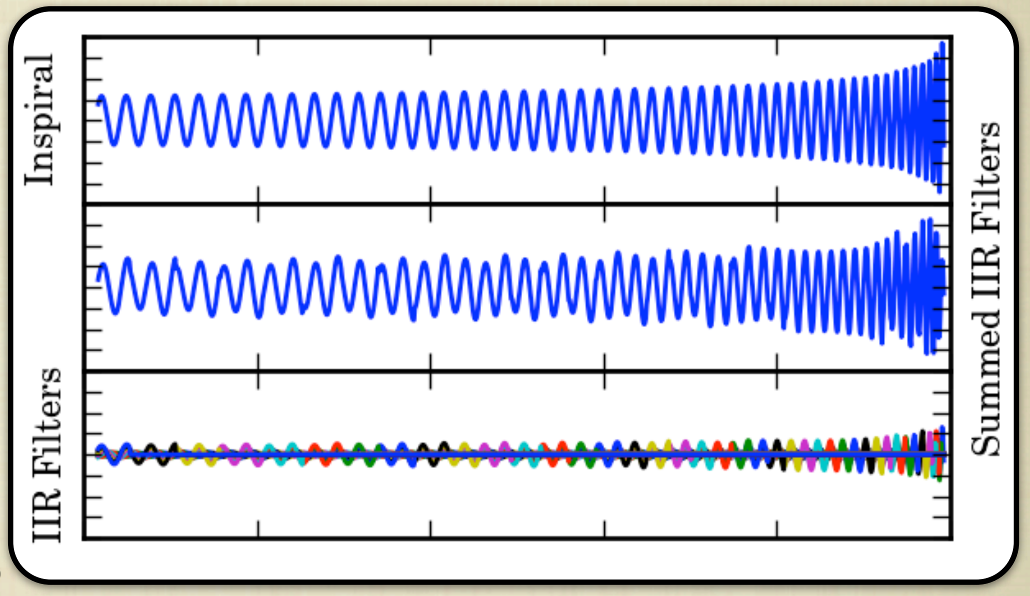


Figure Credit: K. Cannon, ..., **DK**, et al, ApJ **748**, 136 (2012)

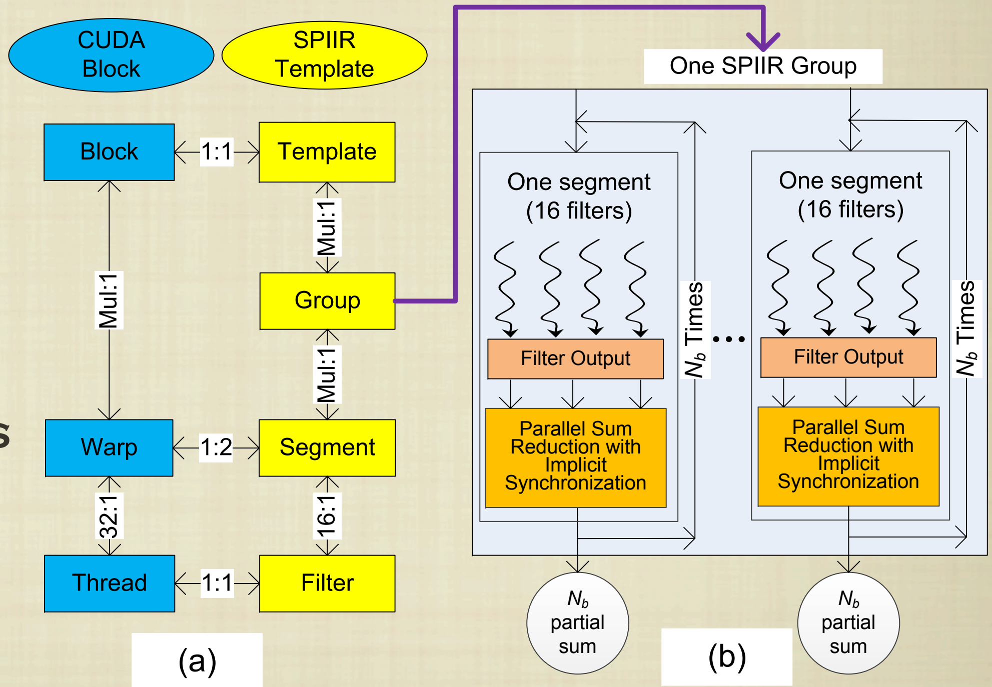
GPU Acceleration Efforts: Low Latency With SPIIR

- Waveforms approximated by Summed Parallel IIR filters
- Computationally less expensive than FIR representation
  - 1(complex) Add + 1(complex) Mul per sample per IIR filter (each is a single pole)
  - Hundreds of IIR filters per waveform



GPU Acceleration Efforts: Low Latency With SPIIR

- Good target for acceleration
  - PRO: Algorithm highly parallelized
  - CON: Requires synchronized outputs
- Sub-calculations regrouped for performance enhancements
  - Synch events reduced by number of samples per batch
  - Align number of filters per segment with size of warp



GPU Acceleration Efforts: Low Latency Limitations

- Choice of implementation
  - make GPU kernels for small tools or large blocks?
  - Where is the line between CPU and GPU computations?
- Observed limits in SPIIR
  - few operations per data sample
  - more adds than muls
  - comparable to inefficiencies in FFT algorithms

GPU Acceleration Efforts: Using Pyfft, Pycuda, Pyopencl

- Build a framework with transparent acceleration for simple algorithms
- Some investigations compute filter output 10^3-10^6 times with various parameters
  - Template bank covering studies
  - parameter estimation searches
- Amount of acceleration algorithm dependent
  - memory transfers
  - generation of waveforms
- Could be used to investigate and implement new pipelines

The End

Questions?