



# GWTools

*A many-core gravitational wave data analysis toolkit*

A project status report

*Debreczeni Gergely  
on behalf of GWTools developers*

*Wigner Research Centre for Physics  
([Gergely.Debreczeni@wigner.mta.hu](mailto:Gergely.Debreczeni@wigner.mta.hu))*

# Content

- This is not a complete presentation just a reminder and a short status update.
- What is GWTools, what this project is about ?
- The GWTools web site: <http://gwtools.org>
- A gadget app - gwt-inspiral-spectrum
- A performance app - GWTools's findchirp implementation
- Future plans



# The GWTools project

This presentation is an extract of the web site containing much more information:

<http://gwtools.org>

You can find much more documentation, explanation, description, installation instructions and even a tutorial there.

## Developers:

Gergely Debreczeni

Tito dal Canton

Badris Krishnan

Karsten Wiesner



**wigner** Max-Planck-Institut für Gravitationsphysik (Albert-Einstein-Institut)

Main page | Apps | Docs | Downloads | Links | News | Career | Devel | Contacts

GWTools is a Gravitational Wave data analysis **Toolkit** which seamlessly brings the immense computing power provided by the emerging many-core architectures - such as GPUs, APUs, MICs - to the service of scientists of the era of advanced detectors...

[...more](#)

**VIRGO**

**The GWtools project**  
The GWTools - Gravitational Wave data analysis Toolkit is jointly developed by the Virgo group of the Wigner Research Centre for Physics of the Hungarian Academy of Sciences, Budapest and the Ligo Group of Max Planck Institute for Gravitational Physics, Hannover. The goal of the project...

[...more](#)

**LIGO**

**Technical description**  
GWTools is an implementation of many data analysis algorithms widely used within the LigoVirgo gravitational wave research community. Based on C++ and OpenCL, it is modular, portable across operating systems and hardware platforms and compatible with all major batch systems. The OpenCL implementation inherently allows the automated and invisible parallelisation of the various routines on CPUs as well as on GPUs...

[...more](#)

**People**  
We have already received many contributions from people with expertise in various fields, such as software engineers, data analysts, theoretical and experimental physicist together with new requests for various analysis groups for applications. Current team members are ...

[...more](#)

**Welcome to the GWTools web site ! Here you will find quite some interesting and/or useful information about GWTools, the C++/OpenCL based Gravitational Wave data analysis Toolkit.**

The website contains all the necessary information about GWTools library starting from [general documentation](#), through [architectural overview](#) and [download resources](#), together with [installation instructions](#), or the answers so [frequently asked questions](#). You can also consult with the page of the [already implemented application](#) or check out some of the [interesting links](#), or in the worst case you can reconsider [your career](#). And if you have anything to share do not hesitate to [contact us](#).

[learn more](#) ➔

October 30, 2012  
[CBC focus session](#)  
An introductory presentation about GWTools will be given on the CBC focus session...

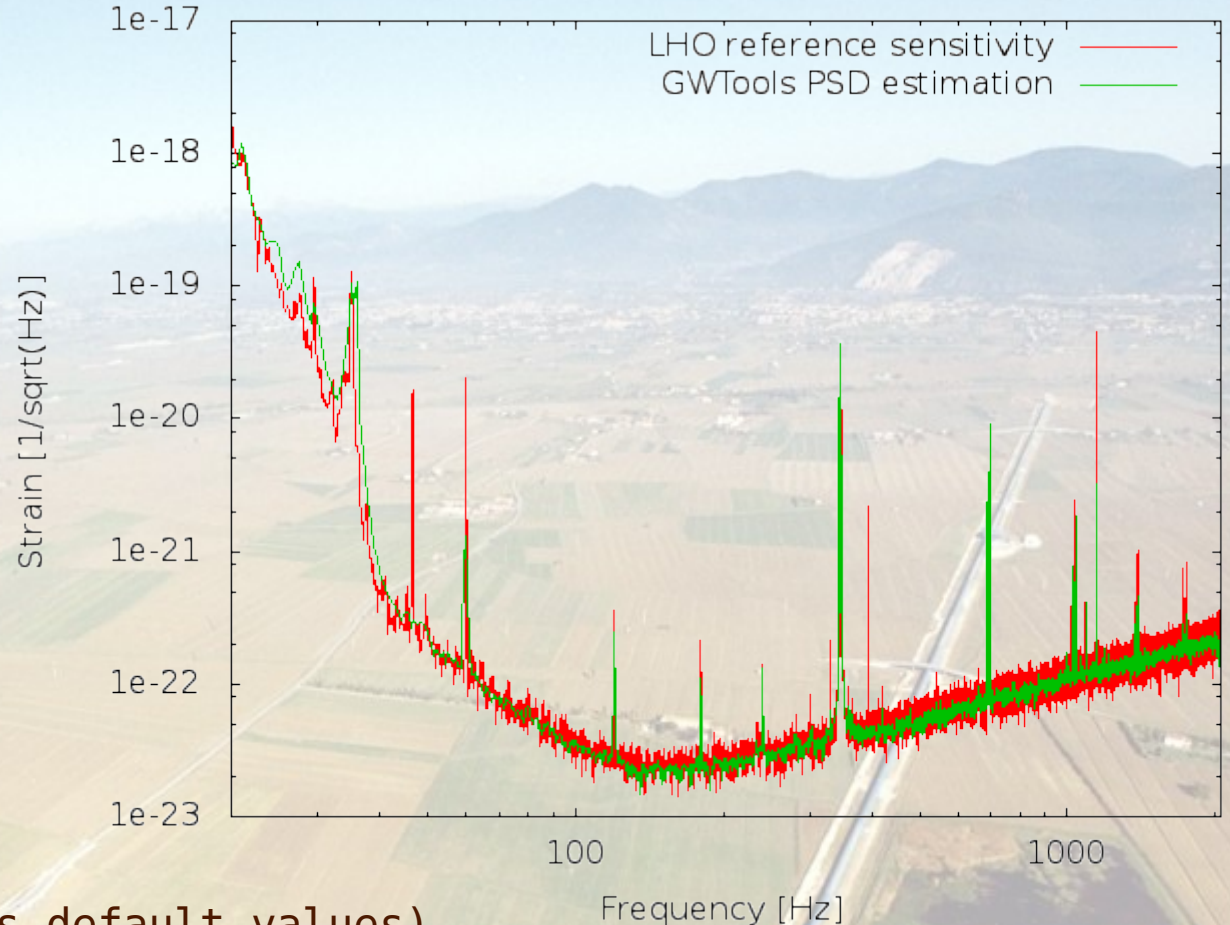
October 17, 2012  
[EGO signal processing workshop](#)  
A short presentation about GWTools will be given on the 1st Signal Processing Workshop organized by EGO...

[view all news](#) ➔

©2012 Ligo - Virgo Collaboration • [Disclaimer](#)

# A gadget app : gwt-inspiral-spectrum

- Small utility for obtaining spectra
- Part of gwtools-inspiral package
- Read a frame-cache file and outputs the PSD on the required sampling rate
- Useful for quick testing (see Stephen's mail last week on DASWG)



- Usage (many of the flags has default values)

```
gdebrecz@gpudev6: gwt-inspiral-spectrum --debug-level 5 --frame-cache  
/home/gdebrecz/testcache.cache --spectrum-file mypsd.dat --spectrum-type average  
--channel-name H1:LSC-STRAIN --dynamic-range-exponent 16 --segment-length 128  
--gps-start-time 865543903 --gps-end-time 865544927 --sampling-rate 4096
```



# A performance app: GWTool's „findchirp” application

## GWTool's „findchirp”:

- Implements the standard findchirp (matched-filter) algorithm
- Data preconditioning
- Template generation
- SNR time series
- 16-band chi<sup>2</sup> time series of SNR is above the threshold
- Clustering and maximum finding
- Input and output is compatible with `lalapps_inspiral`
- Can be plugged into any `ihope` pipeline

### Configuration file:

```
frame-cache = ./ninja2.cache
channel-name = H1:LDAS-STRAIN
gps-start-time = 871148452
gps-end-time = 871150500
#bank-file = ./bank_11678.xml.gz
bank-file = ./bank_100.xml.gz
snr-threshold = 5.5
chisq-threshold = 10.0
chisq-bins = 16
chisq-delta = 0.2
dynamic-range-exponent = 69.0
pad-data = 8
sample-rate = 4096
enable-high-pass = 30.0
low-frequency-cutoff = 40.0
spectrum-type = median
segment-length = 1048576
segment-overlap = 524288
inverse-spec-length = 16
ifo-tag = FIRST
trig-start-time = 0
trig-end-time = 0
cluster-method = template
```

# Findchirp current results

## Configuration file:

```
frame-cache = ./ninja2.cache
channel-name = H1:LDAS-STRAIN
gps-start-time = 871148452
gps-end-time = 871150500
#bank-file = ./bank_11678.xml.gz
bank-file = ./bank_100.xml.gz
snr-threshold = 5.5
chisq-threshold = 10.0
chisq-bins = 16
chisq-delta = 0.2
dynamic-range-exponent = 69.0
pad-data = 8
sample-rate = 4096
enable-high-pass = 30.0
low-frequency-cutoff = 40.0
spectrum-type = median
segment-length = 1048576
segment-overlap = 524288
inverse-spec-length = 16
ifo-tag = FIRST
trig-start-time = 0
trig-end-time = 0
cluster-method = template
```

**Compared to lalapps\_inspiral it  
is c.c. x120 times faster ->**

## Timing numbers:

```
11/18 16:19:46 DEBUG [main] Template 98 sigma 17.980
11/18 16:19:46 DEBUG [main] Template 99 sigma 17.136
11/18 16:19:46 INFO [main] Found 860 triggers
11/18 16:19:46 INFO [main] Writing output XML file
11/18 16:19:46 INFO [main] Total execution time: 36.4 s, 100%
11/18 16:19:46 INFO [main] Data conditioning: 27.5 s, 75.6%
11/18 16:19:46 INFO [main] TIMING INFO WITHOUT PRECONDITIONING:
11/18 16:19:46 INFO [main] Waveform generation: 0.2 s, 2.1%
11/18 16:19:46 INFO [main] Variance calculation: 0.2 s, 2.4%
11/18 16:19:46 INFO [main] Matched filter: 2.2 s, 25.2%
11/18 16:19:46 INFO [main] Overlap calculation: 0.6 s, 6.8%
11/18 16:19:46 INFO [main] Inverse FFT: 1.6 s, 18.3%
11/18 16:19:46 INFO [main] Normalisation: 0.1 s, 0.7%
11/18 16:19:46 INFO [main] (Re)initialisation: 0.4 s, 5.0%
11/18 16:19:46 INFO [main] Chi^2 band calculation: 0.0 s, 0.5%
11/18 16:19:46 INFO [main] Chi^2 test: 4.1 s, 46.2%
11/18 16:19:46 INFO [main] Array operations: 1.6 s, 17.7%
11/18 16:19:46 INFO [main] Data copy around: 0.0 s, 0.0%
11/18 16:19:46 INFO [main] FFT: 2.0 s, 22.8%
11/18 16:19:46 INFO [main] Chi^2 calculation: 0.5 s, 5.7%
11/18 16:19:46 INFO [main] Thresholding: 1.0 s, 11.0%
11/18 16:19:46 INFO [main] Copying data: 0.1 s, 0.6%
11/18 16:19:46 INFO [main] Clustering: 0.5 s, 5.9%
11/18 16:19:46 INFO [main] Maximize over bank: 0.0 s, 0.0%
11/18 16:19:46 INFO [main] Disk I/O: 0.0 s, 0.4%
11/18 16:19:46 INFO [main] OTHER:
11/18 16:19:46 INFO [main] Segments filtered: 1500, 100%
11/18 16:19:46 INFO [main] Chi2 filtered segments: 173, 11.5%
11/18 16:19:46 INFO [main] Performance: 11.3 template/sec
```



# Status, future plans:

- Currently the fine tuning of the algorithms are happening
- Thanks to its uniq OpenCL code base, GWTools **does run on**
  - Nvidia & AMD GPUs,
  - any CPUs,
  - Intel Phi,
  - Altera FPGA,
  - HSA Architectures such as AMD's Kaveri
- Aligned spinning template waveforms are implemented
- Analysis for aligned spin binaries are on the way...
- More unit and stress testing to be done
- First scientific results to be expected in the following month....