Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

GWTBoost: boosting the efficiency of gravitational wave transients

Giacomo Principe on behalf of the Univ. Trieste GW group: *Edoardo Milotti, Agata Trovato, Giacomo Principe, Davide De Piero, Panaiotis Iosif, Leigh Smith, Giuseppe Troian, Andrea Virtuoso*;
and collaborating with *Francesco Salemi* and *Marco Drago* (Uni. Roma 1)

ICSC Spoke 2 Annual Meeting – 10-12 December 2024 Catania

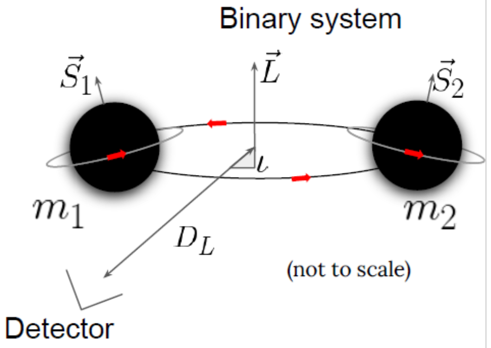
Unmodelled GW events

GW universe is not only Binary systems

(only Compact Binary Coalescence detected so far)

- GW burst expected also for other sources:

  (Supernovae, GRB/FRB, Magnetar flares,?)
- We do not have models for any possible mechanism
- Must be ready for the unexpected
- Search for unknown transients / **Burst**

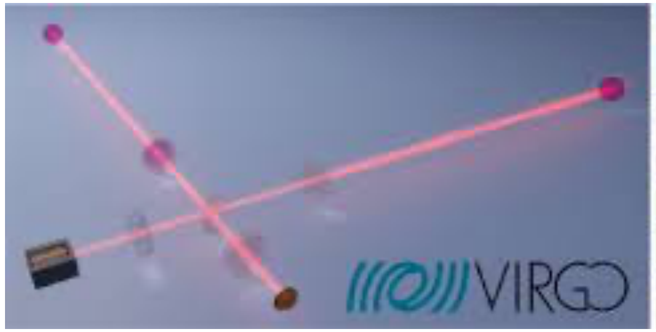


All sky unmodelled search	**model–informed search**	**matched-filter search**

assumptions on the GW waveform

GW Burst search Challenges:

1. Detection
   - Lack of models, open to all possible signals
   - Distinguish the real signal from detector noise
2. Source identification
   - Waveform shape depends on generating process
3. Sky localization
   - Search of counterparts



cWB: Search for unmodelled GW events

cWB is an open source software (written in C/C++ and ROOT) for GW data analysis

cWB is weakly-modelled algorithm used to search for GW bursts and compact object coalescences, and to reconstruct GW waveforms with minimal assumptions.

cWB detects the excess of signal power that is coherent in the network of detectors.

https://gwburst.gitlab.io

Abbott, R., et al. "All-sky search for short gravitational-wave bursts in the third Advanced LIGO and Advanced Virgo run." (2021)

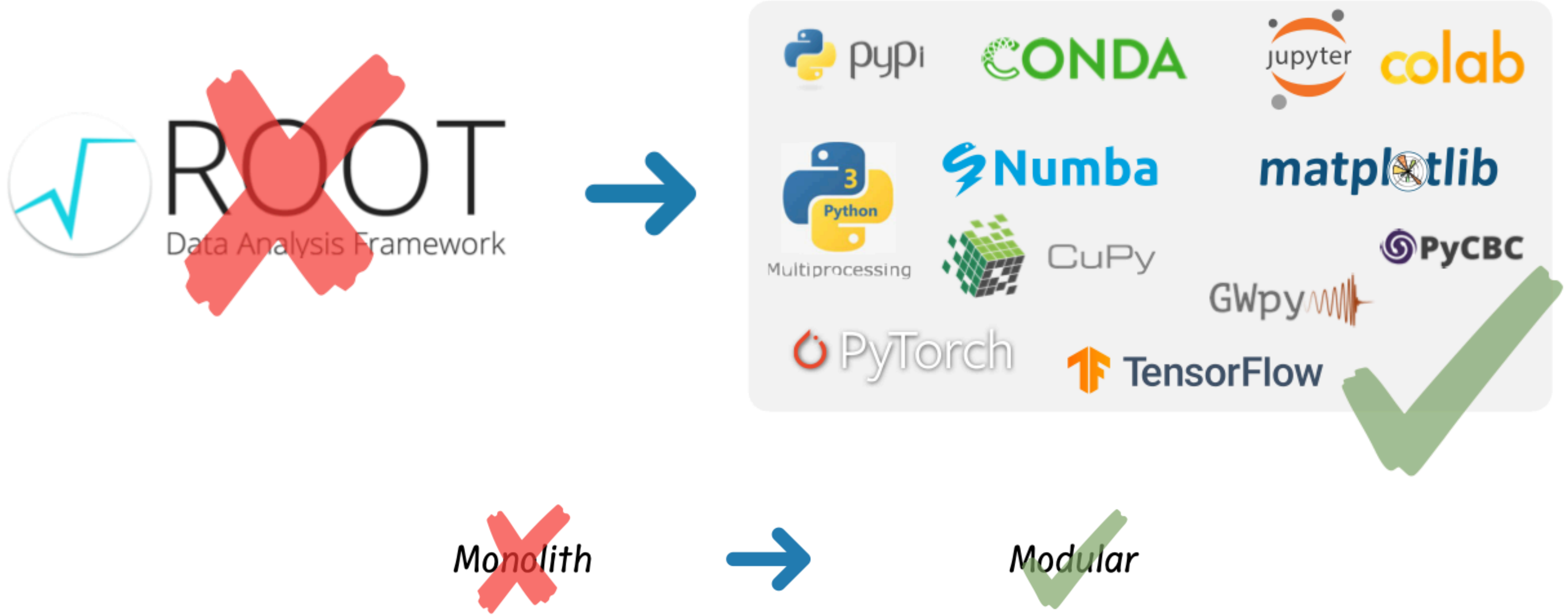
Salemi, F., et al. "Wider look at the gravitational-wave transients from GWTC-1 using an unmodeled reconstruction method." (2019).

pycWB: a *under-development* user-friendly, modularized Python package for GW burst search based on the core function of cWB.

https://arxiv.org/abs/2308.08639 - https://pypi.org/project/PycWB/0.18.2/

Motivation for PycWB

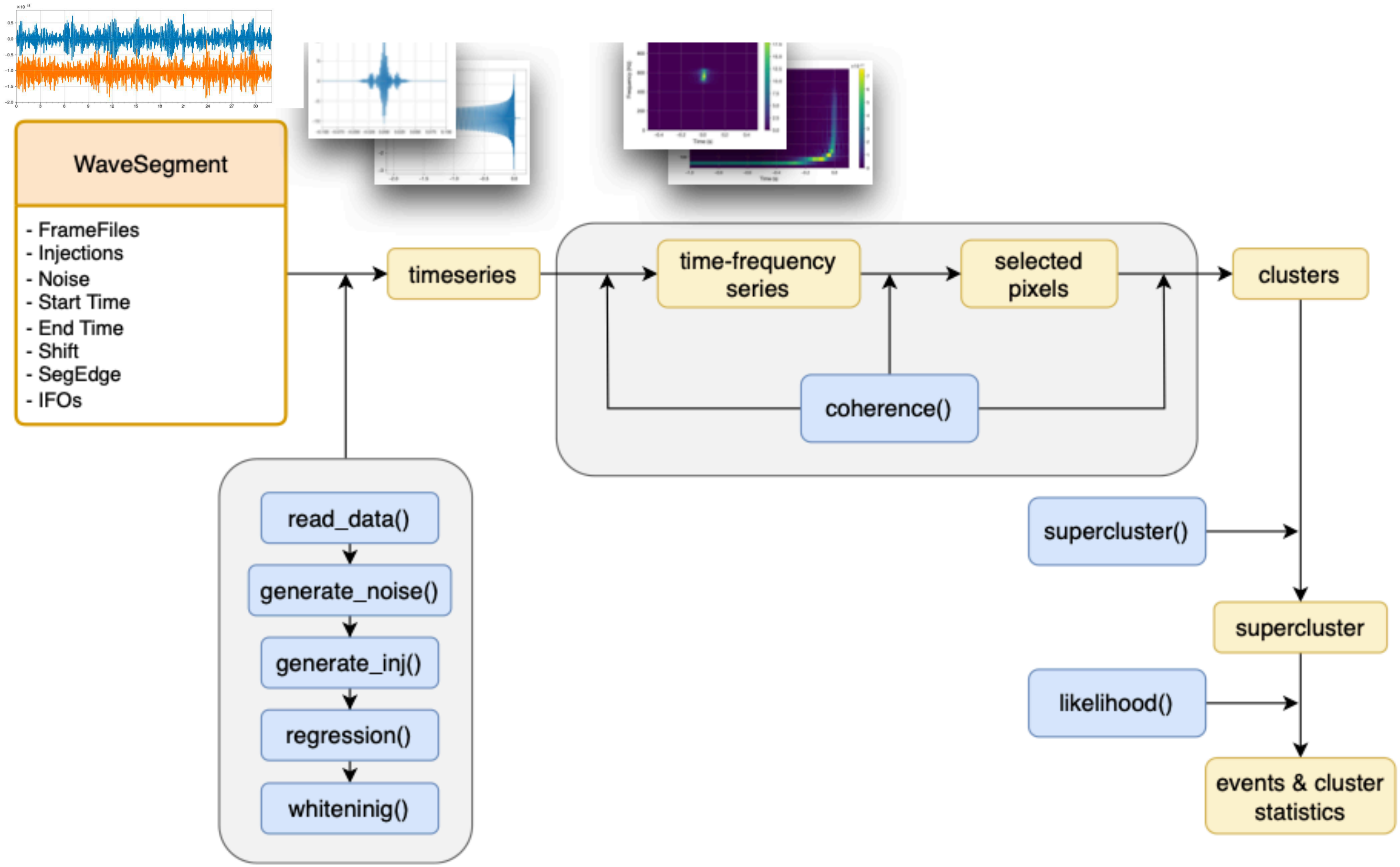
pycWB



cWB and pyCWB workflow

Pipeline workflow:

1. *Time-frequency representation*
2. *Whitening*
3. *Selection of most energetic pixels*
4. *Coherent analysis (maximum likelihood)*
5. *Compute ranking statistic to each trigger*

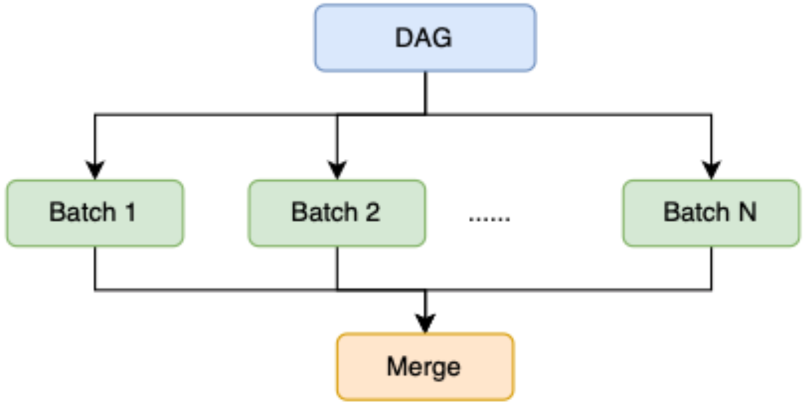


cWB almost exclusively runs on CIT (HPC center at the California Institute of Technology) which is often overburdened by the wider GW community.

cWB/pyCWB and computations

Because of its non-stationarity and non-Gaussianity, the *background noise must be empirically estimated* (i.e. repeating the search using *time slides*). The computational burden of this process scales linearly with the level of accuracy needed at a given *false alarm rate*.

In **pyCWB** job control done with *HTCondor*



Monitoring the jobs with *Prefect+Dask*

- splitting each stage into task
- Dask can manage and scale with cluster
- can run each task in parallel

Porting cWB and pyCWB on LEONARDO (+ machine learning features)

We are working on superseding the computing burden limitations (which are particularly relevant for the likelihood analysis) by porting and adapting the pipeline code to the GPUs available at the LEONARDO.

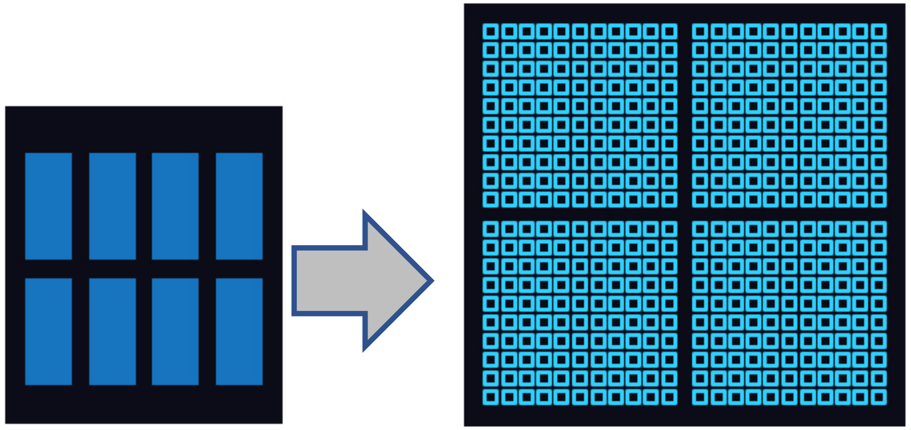
cWB is coded in C/C++ for efficiency, with vectorizations wherever possible.

- parallelization, partial rewriting using CUDA for *GPU exploitation* and (possibly) MPI to distribute load over several nodes.

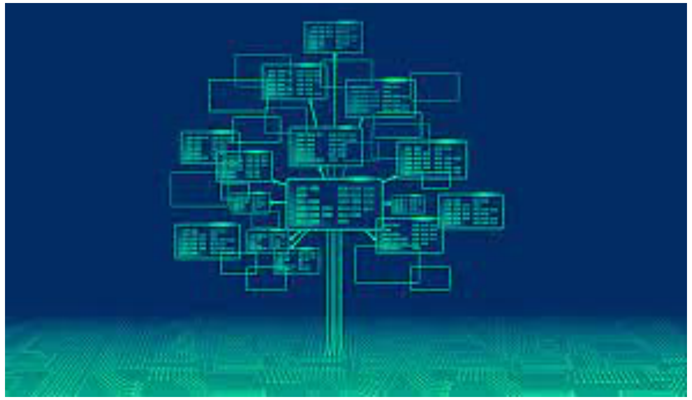
PycWB

- boost the efficiency of the Python wrapper by moving to GPU-enhanced versions of the Python libraries.

+ Inclusion of new features using *machine learning (Boosted Decision Trees)*



CPU GPU



Workplan GWTBoost (WP3)

- Dec. 2023: Submitted GWTBoost proposal to Spoke2
- April 2024: Proposal accepted
- June 2024 Obtained resources at LEONARDO

Status resources
- 400k hours allocated at LEONARDO
- 1-2% used (for initial tests)

	2024		2025			
Task title	**Months 7-9**	**Months 10-12**	**Months 1-3**	**Months 4-6**	**Months 7-9**	**Months 10-12**
Setup computing environment	COMPLETED					
Import O3 LVK public data		ONGOING				
Benchmark existing cWB-2G on LEONARDO		ONGOING				
TIming tests to detect time-consuming code in cWB-2G						
Timing tests on PycWB		ONGOING				
Developing new functionalities in PyCWB		ONGOING				
cWB-2G and PycWB optimization						
Algorithmic mods in cWB-2G and PycWB						
cWB-2G and PycWB + mods: tests and optimization						
communication of results at meeting/conferences						
paper writing						

Thanks for your attention

Backup slides

Future cWB implementations: Wavelet and Qp transform

Implementing on cWB (or a future pipeline) the new results obtained by Virtuoso et al. (in preparation)

1. Implement a wavelet version of the Q-transform which does have a non-standard inversion formula
   - effective and computationally fast denoising formula

$$s(\tau) = \frac{2}{\left(\frac{2}{\pi Q^2}\right)^{1/4} \mathrm{erf}\left(\frac{Q}{2}\right)} \mathrm{Re}\left[\int_0^{+\infty} d\phi \, \frac{1}{i\sqrt{2\pi\phi}2\pi\phi} \frac{\partial}{\partial\tau} T(\tau, \phi, Q)\right]$$

2. Extend the Q-transform (Qp) to better follow frequency-evolving / chirp-like GW signals
   - very effective noise filtering
   - more compact representation of chirps, improving likelihood, SNR and waveform reconstruction

Virtuoso A. & Milotti E. (2024) https://arxiv.org/abs/2404.18781

Future implementations: Wavelet and Qp transform

Virtuoso A. & Milotti E. (2024)
https://arxiv.org/abs/2404.18781

Q-transform

Qp-transform