Spoke 2 use cases

Definition and workflow document

Frequency Hough (FH) Transform analysis on GW continuous sources

Spoke	2
WP	3
Use case short name	Frequency Hough (FH) Transform analysis on GW continuous sources
Use case ID	UC2.3.1
Expected Completion	9/2025

Approval workflow

Status	**Version**	**Date**	**Submitter**	**Note**	**Signature**
Submitted to Spoke Leaders	1.1	25/7/2023	WP Leader	First version	
Final version	1.2	1/9/2023	WP Leader		
Approved	1.2	11/09/2023	Spoke Leaders		

Acronyms

- SP(s): Spoke Leader(s)
- WP(s): Work Package Leader(s)
- UC: Use Case
- KPI: Key Performance Indicator

Use case definition

Introduction:

The Virgo Rome data analysis group has engaged for many years the analysis of gravitational wave signals of different characteristics (bursts, long transients, continuous signals). The analysis for All Sky searches of continuous signals (CW), which represents the scientific case we are proposing, requires a large computing power. The reason is that in order to explore the parameter space (position in the sky, frequency, spin-down) it is necessary to recalculate the core step (i.e. the Hough transforms) of the analysis procedure. In the last ten years the algorithms have been continuously optimized in order to increase the sensitivity for the signals, considering the available computing resources.

The limitation of computing power has also been tackled by concretely exploring the possibility of using GPUs. Thanks to the latter, the most significant results in the case of All-Sky searches were obtained in the analysis of the frequency bands of the data of the third scientific run (O3) of the LIGO/Virgo collaboration, analyzed in 2021. For the analysis of the data of the next run (O4) which has just started in May 2023, the algorithms will have to be further optimized.

The nature of CW is such that the analysis needs to be done integrating on reasonably long observing times. We have typically done the analysis on an entire run (O(months, 1 year)). For this reason these analyses can be done after at least months of data have been collected.

The activity we propose involves the optimization of the algorithms for the best use of the computing power. Moreover we plan to use all types of available resources, CPU and GPU, so the two versions of the code will need a specific optimization. Finally the code will need to be reorganized to be used transparently for both types of resources too.

We plan to perform on the same dataset (or on data of equivalent format) a comparative analysis of the various versions of the code, on the different resources that will be made available.

At the current status, the FH procedure already has some degree of parallelization itself, since the analysis on different frequency bands and different sky positions can be done independently. At the moment, the available versions of the code use as input a time-frequency representation of the data obtained with a C code. They can be summarized as:

(A)

Matlab-C version:

Works on CPU, with an optimized code in C and a specific procedure to fasten the computation. Used for official searches so far.

(B)

Python version (using TensorFlow libraries to access GPU):

It is not an end-to-end implementation. This becomes the input of the python FH transform, computed on GPU via TensorFlow 2.0.

This procedure was used to analyze part of the O3 data on the CINECA GPU.

The tasks we propose for this project are (detailed in the KPI's table below):

Task1: Test of an alternative CPU implementation in MATLAB.

The idea is to compare the performances of the old version of the MATLAB-C code (A) with a new algorithm. The new one uses a FH algorithm with a different computation loop structure.

Task2: Optimization of a GPU code in MATLAB, using different implementations and algorithms. This will be done in part by porting the experience of the version (B), but different computing strategies will be explored.

Task3: New implementation for GPU code in Python.

This requires rewriting the FH algorithm and the GPU access via CuPy. The final goal is to verify if it is possible to obtain a more efficient implementation with respect to (B).

The final outcome will be an assessment of the "best" options for the FH for a CPU implementation and a GPU implementation.
To obtain the final result we plan to factorize out data access problems. For the first assessment we will use as input the same data format. For a second stage we will test different GW data formats to spot possible bottlenecks in the access. Finally we will consider the impact on the search sensitivity. In order to do that we plan a test on 6/12 months of data, accordingly with the computational resources that will be available for this activity.
We will use public O3 data to perform our analysis.

References
[1] P. Astone, A. Colla, S. D'Antonio et al: PRD, 90, 042002. "Method for all-sky searches of continuous gravitational wave signals using the frequency-Hough transform" (2014)
[2] I. La Rosa, P. Astone, S. D'Antonio et al, "Continuous Gravitational-Wave Data Analysis with General Purpose Computing on Graphic Processing Units", Universe, 7, 218 (2021)
[3] O1 analysis: LV coll. PRD, 96,062002 "All-sky Search for Periodic Gravitational Waves in the O1 LIGO Data" (2017)
[4] O2 analysis: LV coll. PRD, 100,024004 "All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO O2 data" (2019)
[5] O3 analysis: LVK coll. PRD, 106,102008 "All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO and Advanced Virgo O3 data" (2022)

Participating Institutions

- Leader: Pia Astone (INFN Roma1)
- Participants: Lorenzo Pierini (INFN Roma1)
- External collaborators: Marco Serra (INFN Roma1), Cristiano Palomba (INFN Roma1), Stefano Dal Pra (INFN CNAF, Roma1)

Use Case Expected Activities

The activities on the use case will include:

- 1st period (ICSC 1 - 12 m) – MS6: Study of the different available algorithms

- 2nd period (ICSC 12 - 18 m) – MS7: Test of an alternative CPU implementation in MATLAB
  Deliverable:
  - 1- A report on the code with the description of the performances obtained through the tests done;
  - 2- Code released on an private repository (cannot be public)
- 3rd period (ICSC 18 - 22 m) – MS8: Optimization of a GPU code in MATLAB, using different implementations and algorithms:
  Deliverable (algorithm 1):
  - 1- Code released on an internal repository (cannot be public).
  - 2- A report describing the new algorithms and performances, pointing at the clarifying the sensitivity improvement
- 4th period (ICSC 22 - 26 m) – MS9: Optimization of a GPU code in MATLAB, using different implementations and algorithms:
  Deliverable (algorithm 2):
  - 1- Code released on an internal repository (cannot be public).
  - 2- A report describing the new algorithms and performances, compare it to Algorithm 1, pointing at the clarifying the sensitivity improvement
- 5th period (ICSC 26 - 31 m) – MS10-intermediate: New implementation for GPU code in Python
  Deliverable:
  - 1- Code released on an internal repository (cannot be public).
  - 2- A report describing the new algorithms CPU/GPU and performances
- 6th period (ICSC 31 - 36 m) – MS10: Extensive testing with log runs and different algorithms. Final results comparison
  Deliverable: Final report
  - 1- Code released on an internal repository (cannot be public).
  - 2- A report that will describe the results of all the tests done with the various algorithms, using an extended sample of data to verify the behavior of the software in a realistic analysis situation.

KPIs

KPI ID	Description	Acceptance threshold	To be obtained by
KPI 1	Task 1. New MATLAB CPU code Note: If we get that the new code requires the same computing power the result is successful. In fact, the new implementation allows the length of the frequency segment used to vary, improving sensitivity in each case.	**Figure of merit:** Analysis of the performances of the new code compared to the (A) implementation. We will use the same input data. We will consider the outcome successful if the required computing power for the new task is not greater than in (A).	MS7
KPI 2.1	Task 2. Algorithm 1	**Figure of merit:** Analysis of the performance of Algorithm 1 compared to Task 1 (KPI 1). We will use the same input data. We will consider the outcome successful if the effective time to perform the task on a single GPU is at least ⅕ the effective time on a single CPU core.	MS8

KPI 2.2	Task 2. Algorithm 2	**Figure of merit:** Analysis of the performance of Algorithm 2 compared to Task 1 (KPI 1) . We will use the same input data. We will consider the outcome successful if the effective time to perform the task on a single GPU is at least ⅕ the effective time on a single CPU.	MS9
KPI 3	Task 3	**Figure of merit:** Analysis of the performances of the new code compared to the (A) implementation. We will consider the outcome successful if the required computing power for the new task is at least 10% less than in (A) and that it is not higher than in case (B).	MS10
KPI 4	extensive testing. A single trial can take more than a week so we plan to start as early as 2024.	**Figure of merit:** Stability of the computational infrastructure (running jobs+data access) for the analysis for more than 90 percent of the time we will be given (fraction of “availability time” for CPU/GPU AND storage in the given time slot)	MS10

Risk Analysis

Identifier	Description	Remedy
R1 (KPI 1,2,3)	Unforeseen difficulties in achieving better algorithms	In each case we will take the best possible algorithm and try to optimize that
R2 (KPI 4)	The CN is unable to provide the needed resources, or the needed stability	The program of work and the intermediate and final tests are scaled to what is available.

Resource Needs

Computing resources

- Shared file system between computational nodes (CPU and GPU) to access input data files. 20TB to store 1 year of data. A shared file system is needed also for MATLAB runtime libraries. Each single job needs ~ 20GB of disk space.
- It must be possible to run MATLAB compiled code and Python code
- It must be possible to access GPUs using TensorFlow code or CuPy library
- RAM: for CPU: >= 4 GB/core
- RAM: for GPU: >= 24 GB/card
- It is necessary to have available a test machine, equipped with CUDA architecture GPU, equal to the final compute nodes to prepare the necessary executables and do the preparatory work. Interactive access to the machine via ssh must be possible.
- For testing purposes of this use case we need: ~10^5 core-hours (type of CPU: 10-14 HS06).

- To do an extensive test with 1 year of data (to be processed in a few weeks (2-4)) (CPU ~ 10^7 core-hours): 150~180 GPUs for 2-4 weeks. In any case we will do the test with the largest amount of available resources aiming to check stability over a long period.

Periodic reports during implementation

Milestone 7 (February 2024)

Intermediate report End of October 2023

TAR3.5: UC2.3.1 is the WP3 Flagship named "Frequency Hough (FH) Transform analysis on GW
continuous sources".
During the period September - October 2023:

- We are now running different versions of the FH loops, in order to define the best option for the CPU execution.
- We are now working at fixed sensitivity, to optimise the CPU time needed.
- We are testing different options for the time/spin-down loops.
- We are also verifying the possibility of including a C compiled loop, to further speed up the execution.

MS7 Final Report

For MS7, the planned activities are aligned with TAR3.5, and include:

- Test of an alternative CPU implementation in MATLAB
  Deliverable:
  - 1- A report on the code with the description of the performances obtained through the tests done;
  - 2- Code released on an private repository (cannot be public)

- The KPI described below.

KPI 1	Task 1. New MATLAB CPU code Note: If we get that the new code requires the same computing power the result is successful. In fact, the new implementation allows the length of the frequency segment used to vary, improving sensitivity in each case.	**Figure of merit:** Analysis of the performances of the new code compared to the (A) implementation. We will use the same input data. We will consider the outcome successful if the required computing power for the new task is not greater than in (A).	MS7

1. **Test of an alternative CPU implementation in MATLAB**:
   a. **Activity report**: mettere qui la descrizione del lavoro fatto in modo descrittivo, una specie di riassunto delle parti specifiche sotto, con possibilmente anche plots o comunque materiale dimostrativo. Nel caso si siano fatti talks a conferenze, proceedings, articoli etc metterli.
   b. **A report on the code with the description of the performances obtained through the tests done**: va bene sia farlo qui inlined, con un buon livello di dettaglio, oppure fare un documento esterno e fare riferimento qui. Nel caso di documento esterno un formato a piacere, ma sempre con header e footer e il malefico Titilium Web
   c. Code released on an private repository (cannot be public): schermata che mostra la repo, per esempio, e suo link? Se privata, i reviewers al limite chiederanno accesso
2. **KPI**: New MATLAB CPU code: reached as demonstrated. FOM: Analysis of the performances of the new code compared to the (A) implementation. We will use the same input data. We will consider the outcome successful if the required computing power for the new task is not greater than in (A).
   a. Status: reached as demonstrated in (b)

TAR3.5 declared: ✅

SE LA FLAGSHIP NON HA ATTIVITA' IN MS7 → abbiamo comunque promesso un intermediate report.
Per cui la parte seguente e' la proposta

Milestone 7 (February 2024)

Intermediate report End of October 2023

TAR3.5: UC2.3.1 is the WP3 Flagship named "Frequency Hough (FH) Transform analysis on GW
continuous sources".
During the period September - October 2023:

- We are now running different versions of the FH loops, in order to define the best option for the CPU execution.
- We are now working at fixed sensitivity, to optimise the CPU time needed.
- We are testing different options for the time/spin-down loops.
- We are also verifying the possibility of including a C compiled loop, to further speed up the execution.

MS7 Final Report

The Flagship UC2.3.1 does not have an explicit Milestone or target for MS7; an intermediate report describing the activities so far is included.

Intermediate Report for MS7: mettere almeno mezza pagina / una pagina che descriva le attivita' correnti, con manpower, meeting, e primi risultati (compresi eventuali talk, conferenze, articoli). Sarebbe importante avere un plot / un qualcosa che non sa solo testuale. Il report puo' essere qui inlined oppure in un doc esterno, nel qual caso da linkare e realizzare usando header e footer corretti come al solito.