The ***STACK-SLIDE*** Search

Initial Report: PULG F2F UWM June 2003

Gregory Mendell, Mike Landry

LIGO Hanford Observatory

Initial ***STACK-SLIDE*** Proposal

- Mendell/Landry at LHO to code stack slide algorithm based on Brady/Creighton gr-qc/9812014
- We are in the very preliminary stages of this project, trying to answer questions like
  - Is anyone else planning on working on this (emailed Creighton/Brady/Riles/Chin)?
  - Is there any known problem with stack slide? Is Hough all we need?
  - Do we need to modify B/C algorithm? Stack slide SFTs, DeFTs, or F-stat?
- Algorithm will be coded under LAL, and search run under LDAS
  - Driver code will be available to all (detailed understanding of LDAS not required)
  - Parallel search; exploit ~THz LDAS computing power
  - Efficient I/O of SFTs, results database (or frame output) inherent in LDAS
- We will have a more complete proposal/progress report at the June F2F
- Code written and working by the August LSC meeting

Definitions

- T_0 = observation time.
- M = number of data segments (number of stacks).
- N = number of data points in one segment.
- N_T = total number of data points in T_0.
- A = signal amplitude.
- σ = square root variance of the noise.

Coherent Search

$$x = s + n = A\cos(2pft + f_0) + n$$

$$\tilde{x} = DFT(x)$$

$$\tilde{x}^*\tilde{x} = \left(\frac{ANM}{2}\right)^2 + \left(NMs \pm NMs\right) + cross - term$$

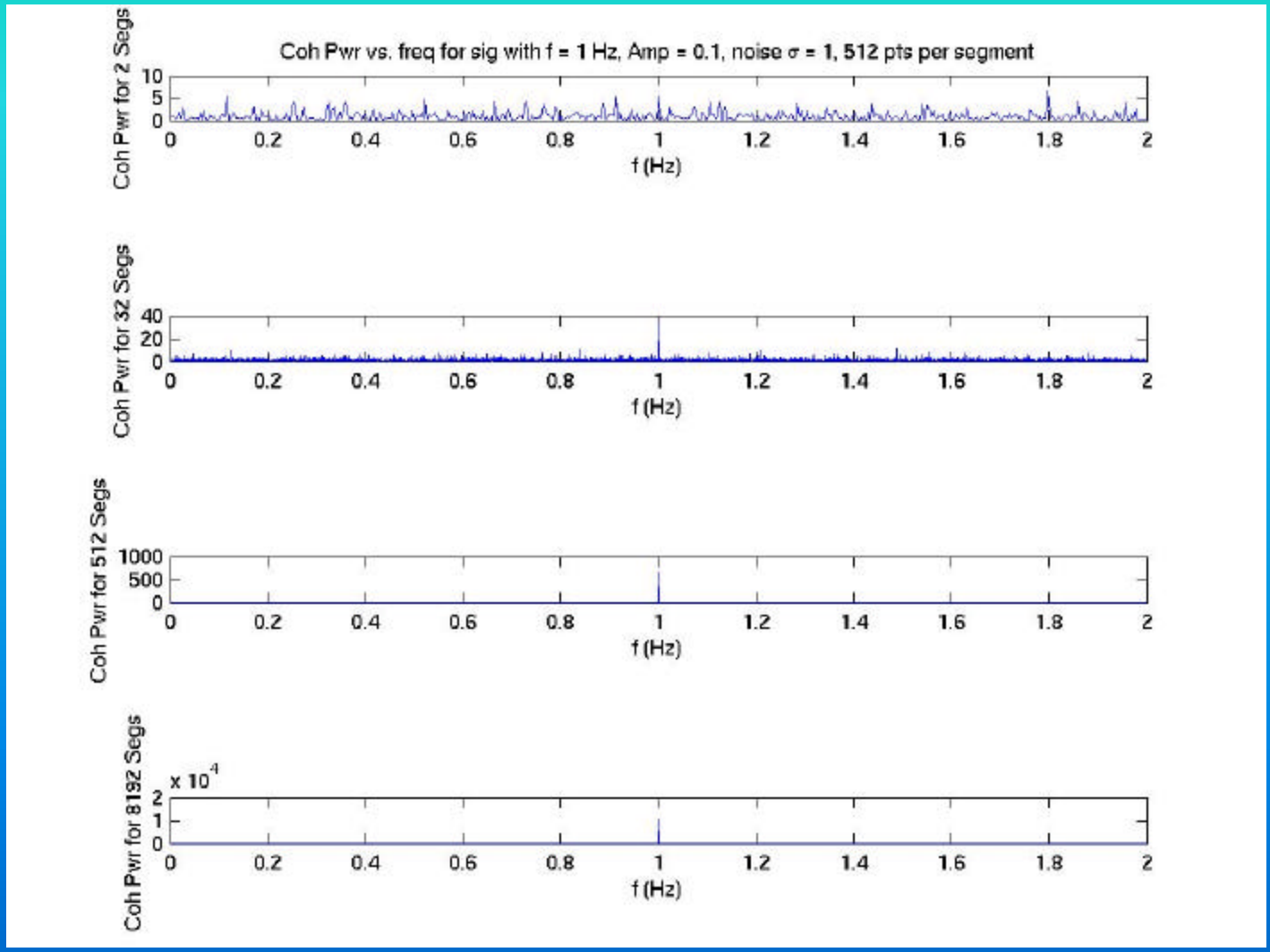
$$\left\langle \tilde{n}^*\tilde{n} \right\rangle = NMs^2$$

For $A^2NM > 4s^2$: $SNR \sim \sqrt{\tilde{x}^*\tilde{x} / \left\langle \tilde{n}^* n \right\rangle} = \frac{A}{2s}\sqrt{NM} = \frac{A}{2s}\sqrt{N_T}$

Detection if: $\frac{A}{2s}\sqrt{NM} >> 1$

False Alarm Rate: $C_{false} = e^{-\frac{A^2}{4s^2}NM}$

Coherent Power vs. Freq



STACK-SLIDE Search

$$x = s + n = A\cos(2pft + f_0) + n$$

$$\tilde{x} = DFT(x)$$

One stack: $$\tilde{x}^*\tilde{x} = \left(\frac{AN}{2}\right)^2 + \left(Ns \pm Ns\right) + cross - term$$

$$\left\langle \tilde{n}^*\tilde{n} \right\rangle = Ns^2$$

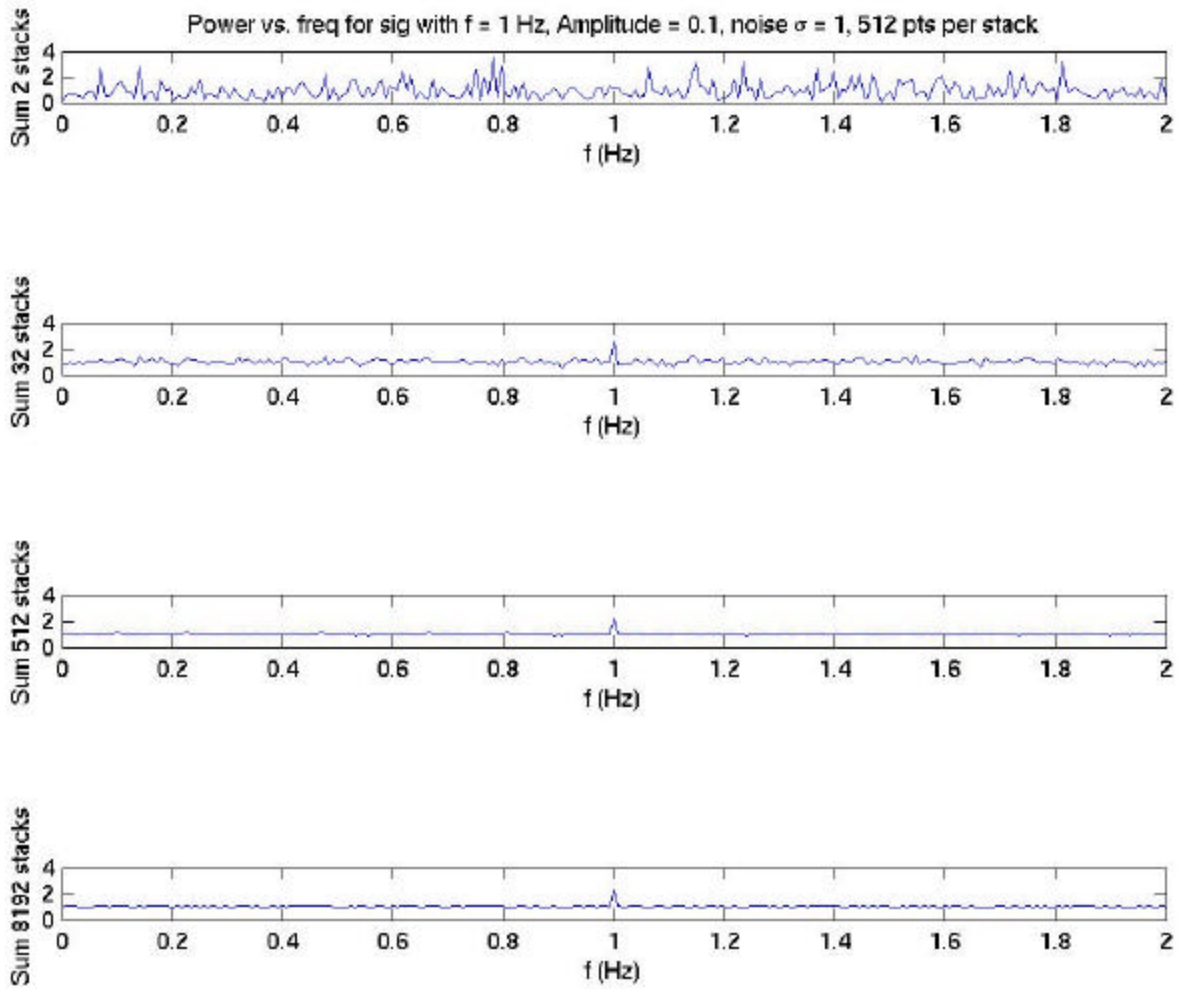
For $A^2N < s^2$: $$SNR \sim (1 \pm 1/\sqrt{M}) + \frac{1}{8}\frac{A^2N}{s^2}$$ **(Ave M stacks)**

Detection if: $$\frac{A}{\sqrt{8}s}\sqrt{N}M^{\frac{1}{4}} >> 1$$

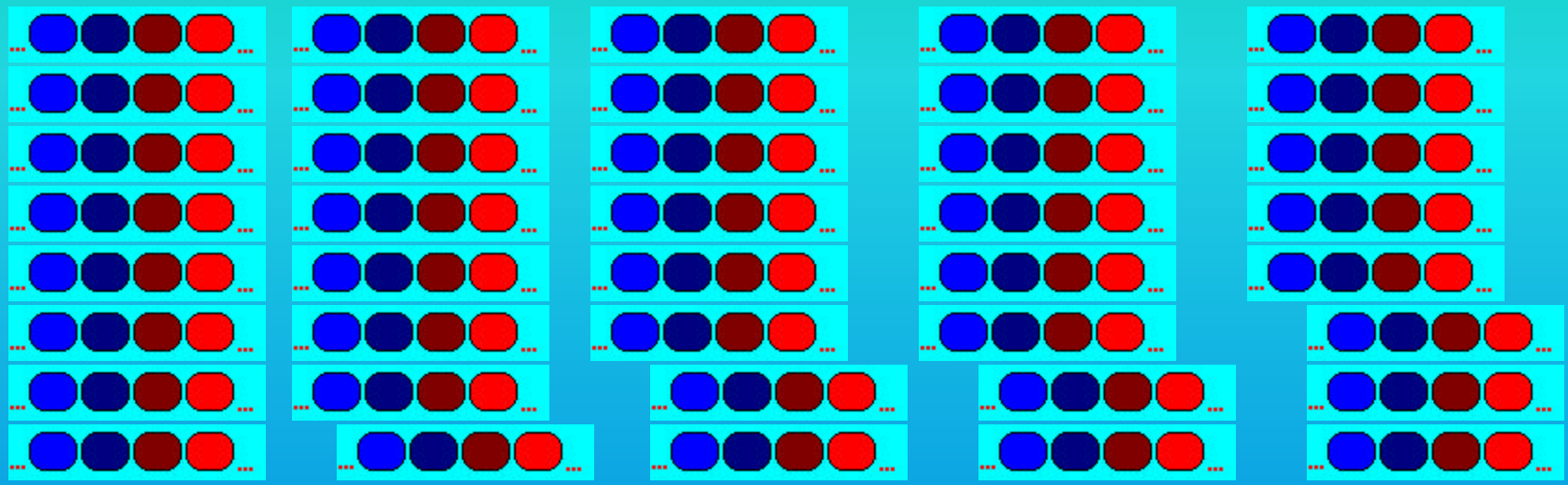
False Alarm Rate: $$C_{false} = 1 - \frac{g(M, SNR)}{(M-1)!}$$ *(Brady/Creighton gr-qc/9812014)*

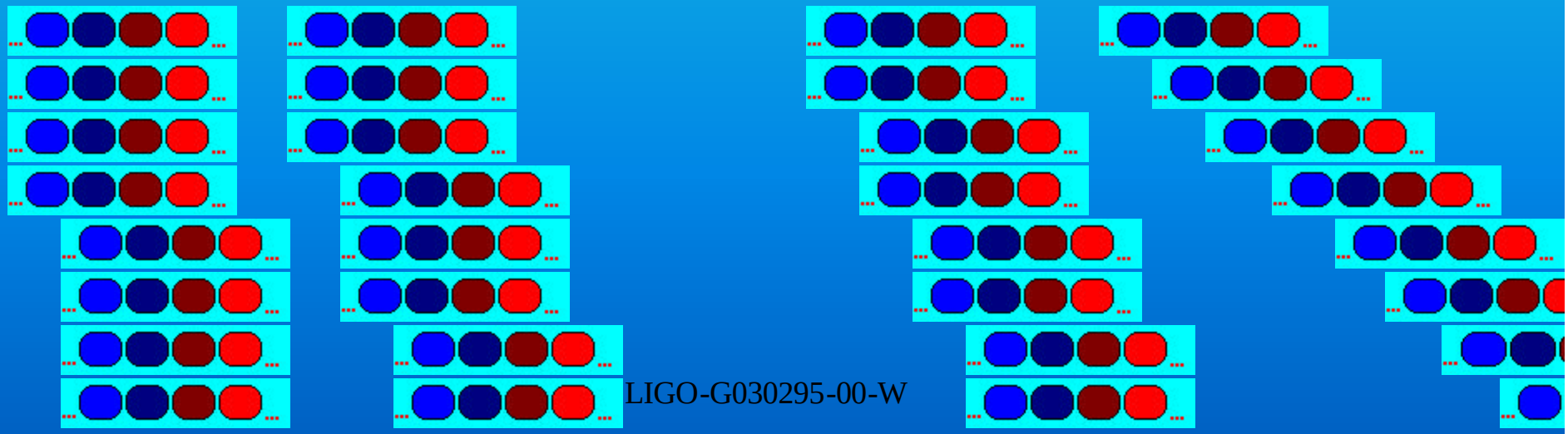
g = Incomplete Gamma Function

STACK-SLIDE Power vs. Freq



Stacking and Sliding…





Complexity?

Complexity STACK-SLIDE = $O(MN\log_2 N) + O(NM^2)$?

Complexity SFT Coherent Per Sky Position = $O(MN\log_2 N) + O(DkNM^2)$?

Complexity Coherent FFT Resampled Time Series Per Sky Position = $O(MN\log_2 MN)$?

We are still working to…

- Understand the Algorithm.
- Understand the Computational Complexity.
- Understand the Statistics.

Brady/Creighton algorithm and potential modifications

Resample?
Or use SFTs,
DeFTs, F-stat
From LALDemod

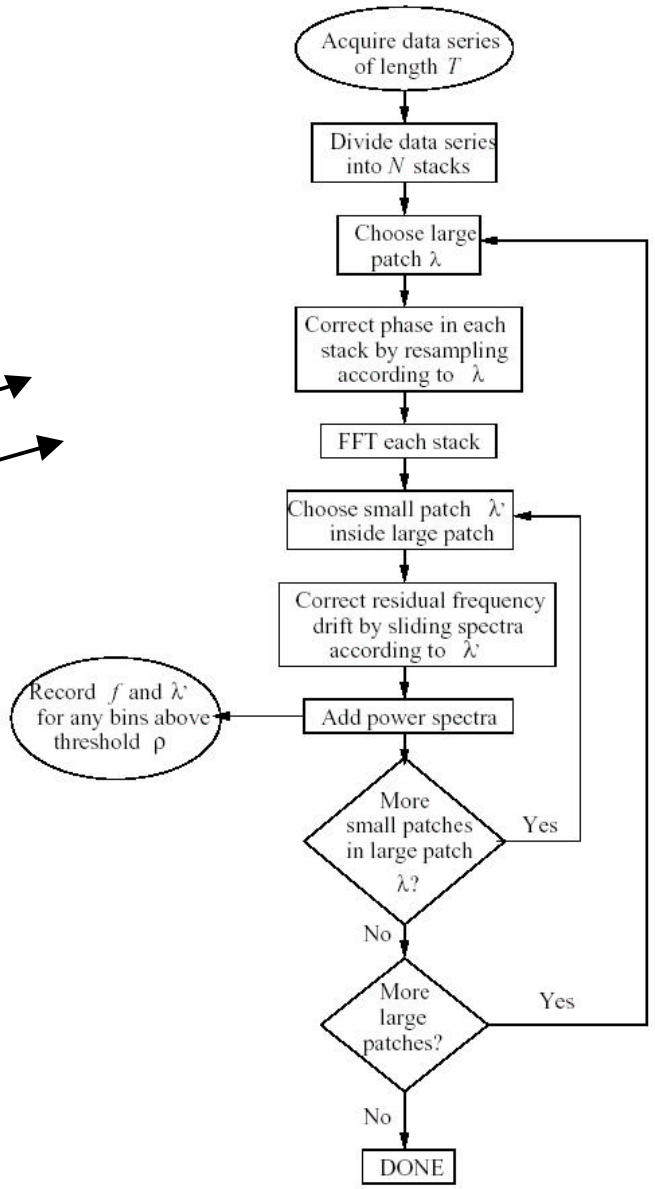


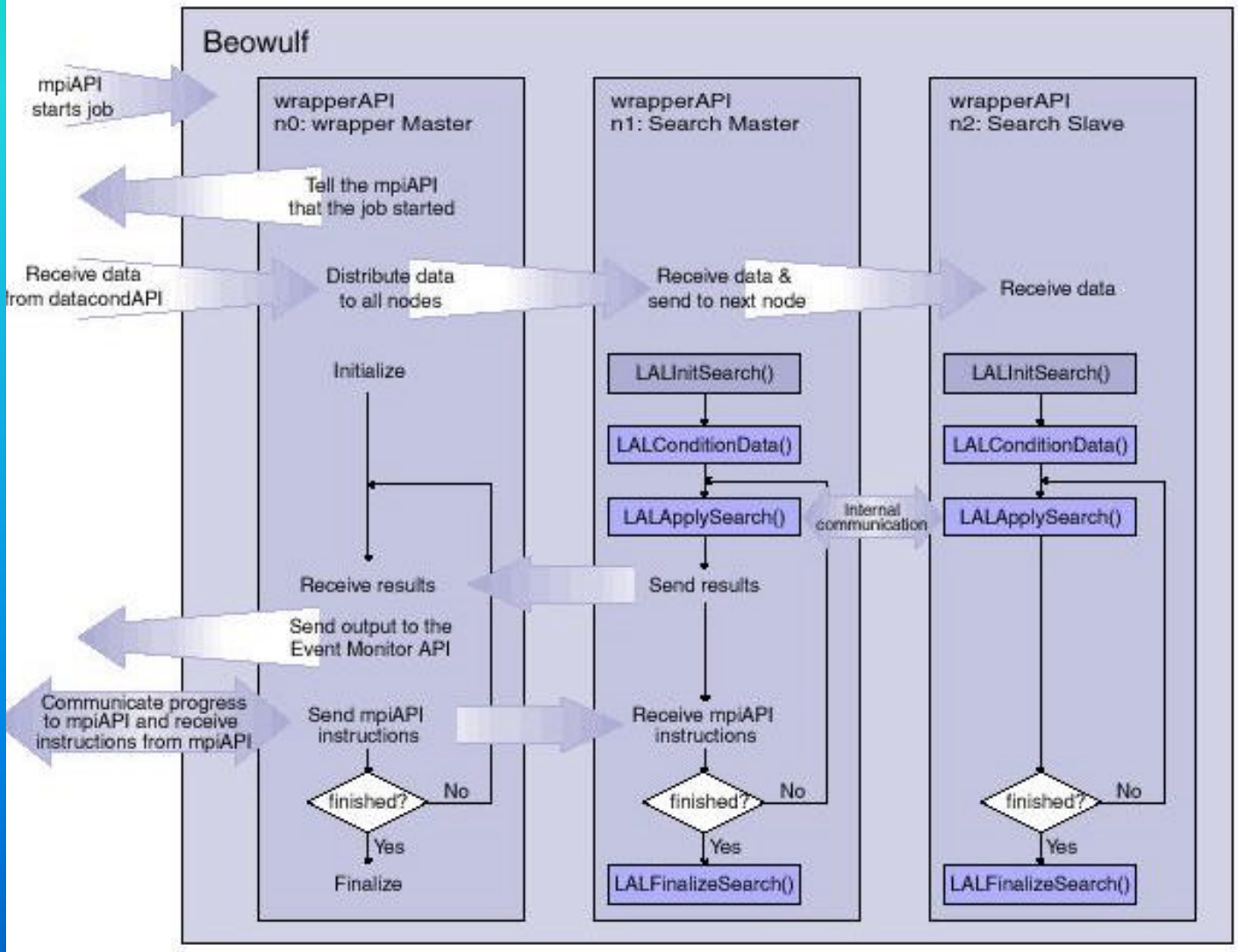
FIG. 1. A flowchart representation of the *stacked slide* algorithm to search for sources of continuous gravitational waves. Notice that the computational cost of sampling the fine grid is reduced by sliding the power spectra, rather than re-computing an FFT for each point on the fine grid.

- The algorithm shown in the Flowchart is iterated in a hierarchical approach

Will Provide LDAS, LAL, & Driver Code

- LDAS code will be based on knownpulsardemod DSO.
- LAL parameter space metric code exists.
- Need LAL STACK-SLIDE function.
- User-friendly driver code that anyone can run.

LDAS Code To Write



Example Driver getFstat

```
Usage: ./getFstat_v1.tclsh <site> <ifo> <start_time> <duration>
<templateFreq0> <band_width> <sft_time> <framecachefile> <templatefile> <outputdatabase>
<outfile> [<run>]


Example:
./getFstat_v1.tclsh lho H1 729976096 8192 1283.5 2 2048
SFTFrameCache_H1_040203.txt testJ1939.params ldas_tst testFstat.xml run
Starting Thu Jun 12 12:51:54 PDT 2003 LDAS job with LJrun at lho;
User: gmendell; Job id: LDAS-WA890236.
LDAS job succeeded.
Total time to run job = 76 seconds.


[gmendell@vulcan scripts]$ less testFstat.xml
...
<LIGO_LW Name=":data:Container(Vect):Frame">
<Dim Scale="1.2207031250000000e-04"
Start="1.2825000000000000e+03" Unit="hz">16388</Dim>
Name="data" Type="Local">
2.6003204903121175e+00,2.3834808600241506e+00,2.7261332694330522e+00,...
```