UWM Milestone Report

Binary Inspiral Hierarchical Search Code

Bruce Allen, Duncan Brown, Jolien Creighton

1. Overview of Status

Hierarchical Search Engine

- coding complete
- fully tested

Binary Inspiral Filtering code

- coding complete
- currently in testing phase

Integration of Inspiral Search into LDAS

- good progress made
- under active development

2. Hierarchical Search Engine

MPI based hierarchical search engine built from LAL functions

Slave driven master/slave design described at last LSC meeting

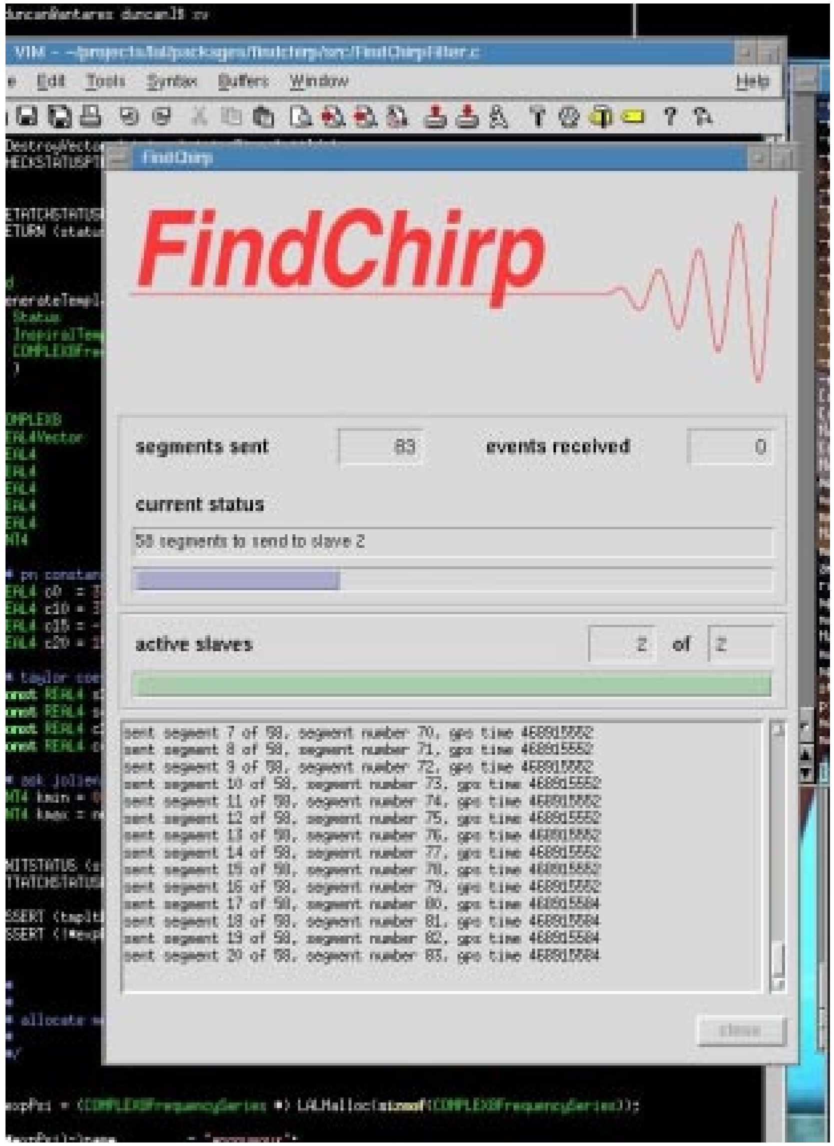
Flexible n level hierarchical engine implemented using template linked list

Filtering code modularised to allow other types of search algorithm

Tck/Tk GUI added to master

Testing of engine with dummy filtering code and template placement routines

- single processor Intel machine
- 48 node DEC Alpha beowulf


FindChirp
segments sent
83
events received
0
current status
58 segments to send to slave 2
active slaves
2
of
2
close

Code divided into two parts:

LAL functions

- linked list manipulation functions
- communications functions
- etc.

Has been documented using LSD and will be released in LAL v0.6

Non-LAL Code

- `int main()`
- master/slave functions
- etc.

Will be available as a "stand alone" package in CVS archive for download

3. Binary Inspiral Search Code

LAL code written to implement matched filtering for inspiral chirps

Uses stationary phase approximation to the inspiral chirp.

Functions coded and documented to LAL standard in package `findchirp`

Will be released in LAL v0.6

To increase efficiency algorithm is divided into

- independent of template
- template dependent

Stationary Phase Approximation to Binary Inspiral Chirp

$$\tilde{h}_c(f) = A_0(M,\eta) f^{-7/6} e^{i\Psi(f;M,\eta)} \tag{1}$$

$$\tilde{h}_s(f) = i\tilde{h}_c(f) \tag{2}$$

Normalisation of the Filter

$$\begin{aligned} \left(\frac{\tilde{h}_c(f;M,\eta)}{S_h(|f|)}, \frac{\tilde{h}_c(f;M,\eta)}{S_h(|f|)}\right) &= A_0^2(M,\eta) \int df \, \frac{f^{-7/3}}{S_h(|f|)} \\ &= \frac{\alpha^2}{2}. \end{aligned} \tag{3}$$

Construction of the Filter

Want to compute the filter outputs:

$$S_c(t; M, \eta) = \frac{2}{\alpha} \int_{-\infty}^{\infty} df\, e^{-2\pi i f t} \frac{\tilde{h}(f) \tilde{h}_c^*(f; M, \eta)}{S_h(f)}, \tag{4}$$

$$S_s(t; M, \eta) = \frac{2}{\alpha} \int_{-\infty}^{\infty} df\, e^{-2\pi i f t} \frac{\tilde{h}(f) \tilde{h}_s^*(f; M, \eta)}{S_h(f)}. \tag{5}$$

Define the signal to noise squared to be

$$\rho^2(t; M, \eta) = S_c^2(t; M, \eta) + S_s^2(t; M, \eta) \tag{6}$$

Then if the signal is white Gaussian noise

$$\langle \rho^2(t; M, \eta) \rangle = 2 \tag{7}$$

Can write

$$\rho^2(t;M,\eta) = \frac{16}{\alpha^2}|z(t;M,\eta)|^2 \tag{8}$$

where

$$z(t;M,\eta) = \int_0^\infty df\, e^{-2\pi i f t}\frac{\tilde{h}(f)\tilde{h}_c^*(f;M,\eta)}{S_h(f)} \tag{9}$$

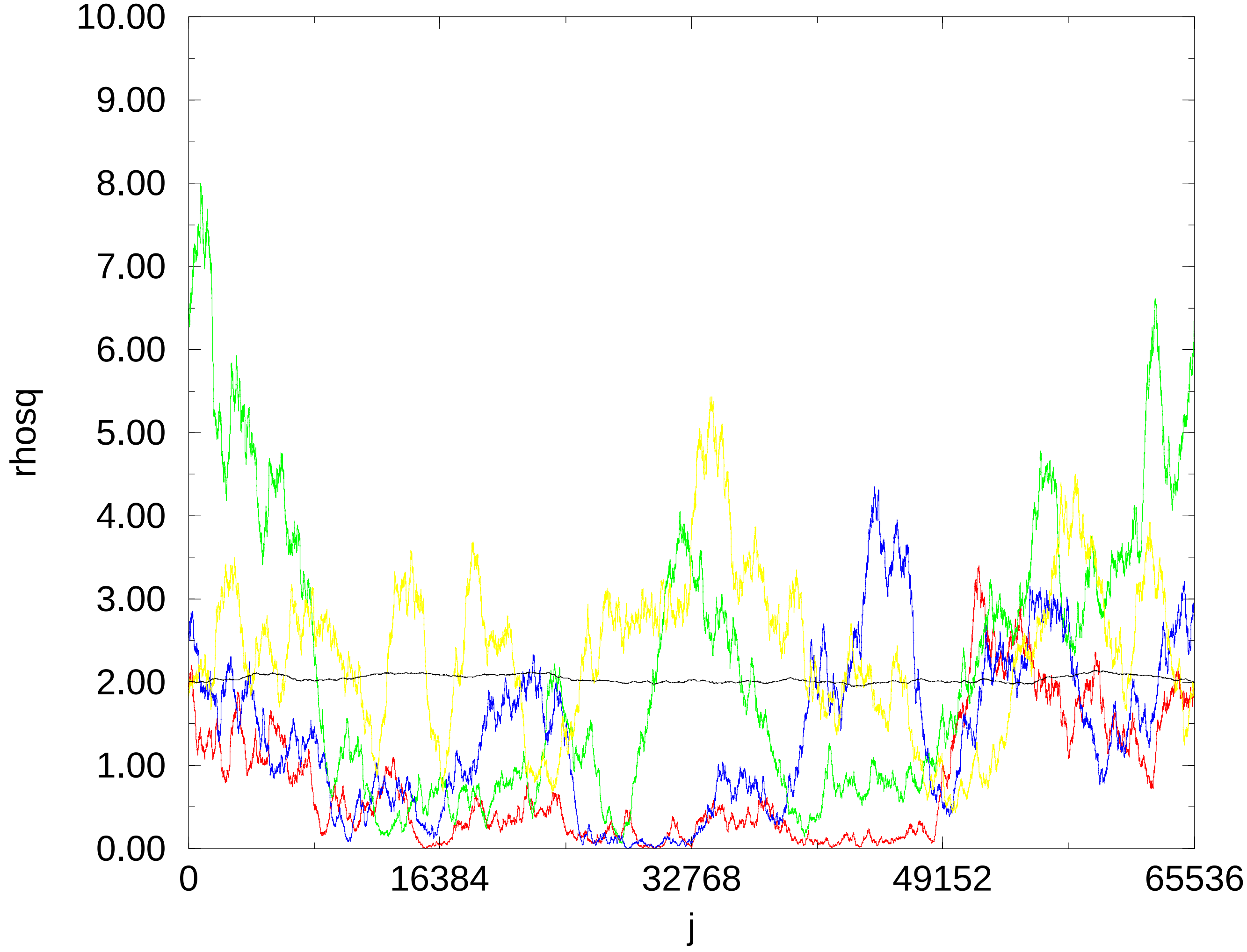
Define

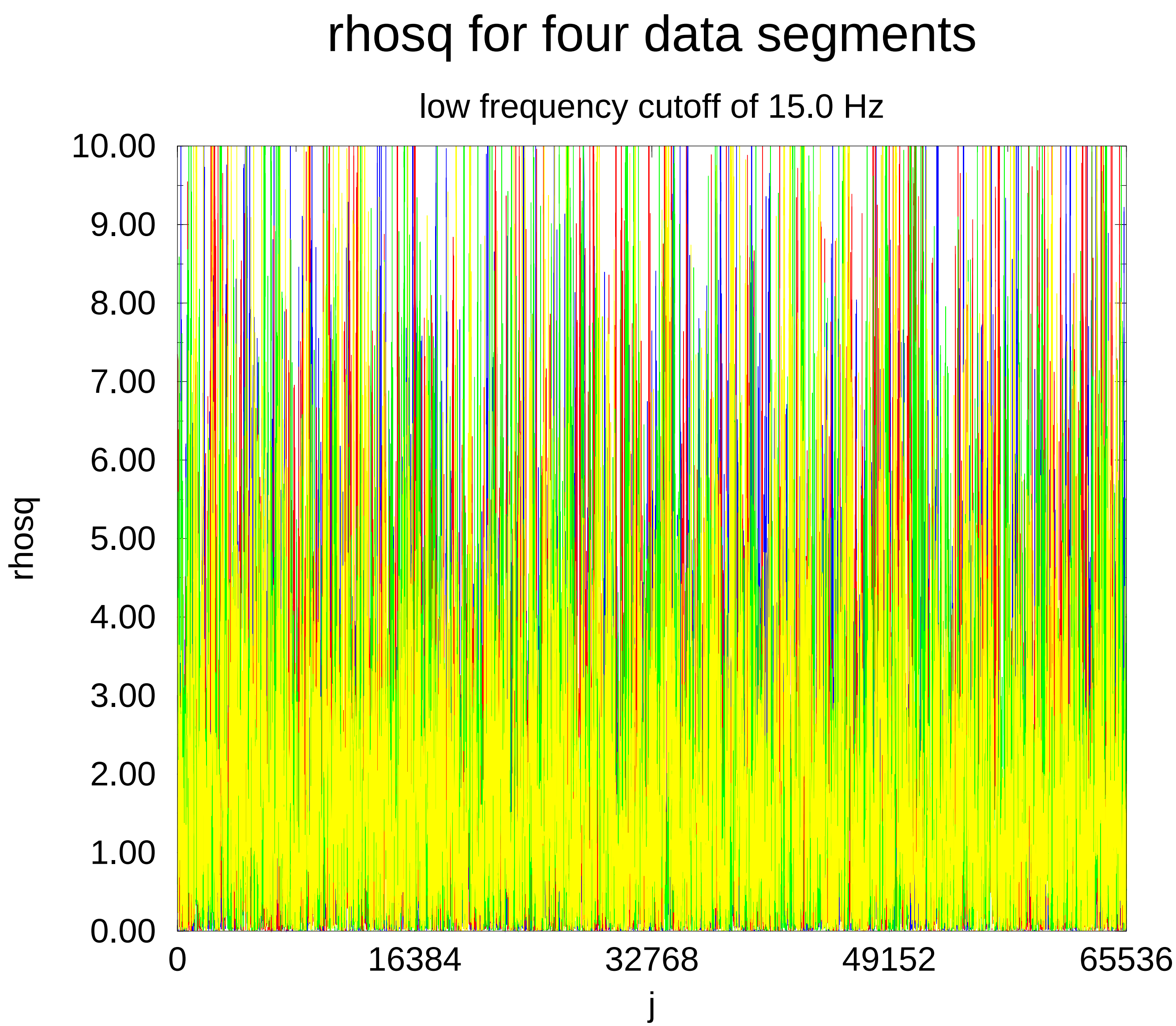
$$\tilde{z}(f;M,\eta) = \begin{cases} 0 & f<0 \\ \frac{\tilde{h}(f)\tilde{h}_c^*(f)}{S_h(f)} & f\geq 0 \end{cases} \tag{10}$$

then can compute $\rho^2(t;M,\eta)$ by IFT of $\tilde{z}(t;M,\eta)$

rhosq averaged over 1024 segments

4 filter outputs shown for comparison




rhosq for four data segments
low frequency cutoff of 15.0 Hz
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
rhosq
0
16384
32768
49152
65536
j

4. Integration of Inspiral Code into LDAS

wrapperAPI shared object has written using search code

Implements flat inspiral search

Converts LDAS standard data types into LAL data structures

Calls LAL filtering code to perform search

Tested on LDAS beowulf

Currently under further development

5. Future Work

Test filtering code with injected signals and 40m data

Integrate Cardiff Template bank placement routines

Test filtering code in hierarchical search engine

Incorporate χ^2 veto code(Warren Anderson)

Continue development of LAL/LDAS interface for MPI MDC