

Optimization of a Gravitational Wave Detection Pipeline

Thomas Hill Almeida (21963144)
Supervisor: Prof. Linqing Wen

University of Western Australia — OzGrav

October 6, 2020



THE UNIVERSITY OF
WESTERN
AUSTRALIA

Structure

N-detector

- Gravitational Waves

- The power of powers of 2

- The hunt for hardcoded detectors

Complexity analysis

- Background

- So why do we care about this complexity analysis?

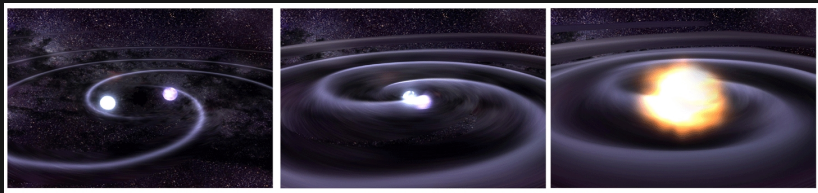
- CUDA

- Analysis

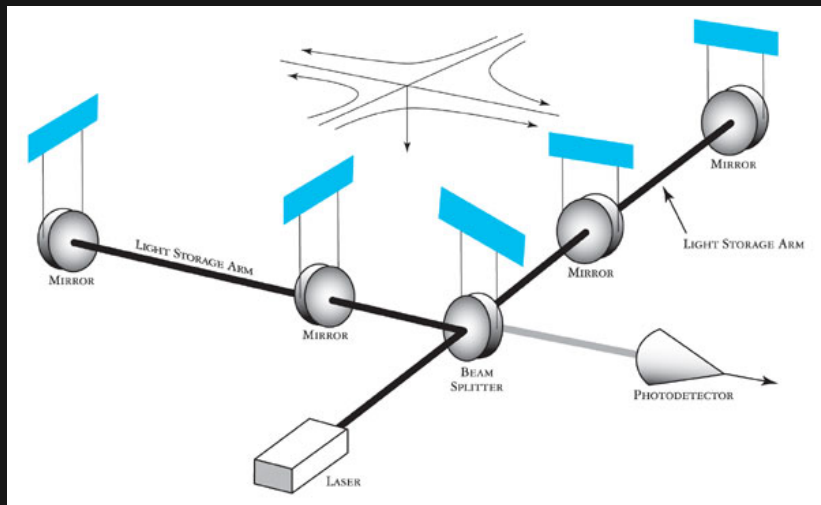
Future Work

N-detector

Gravitational Waves



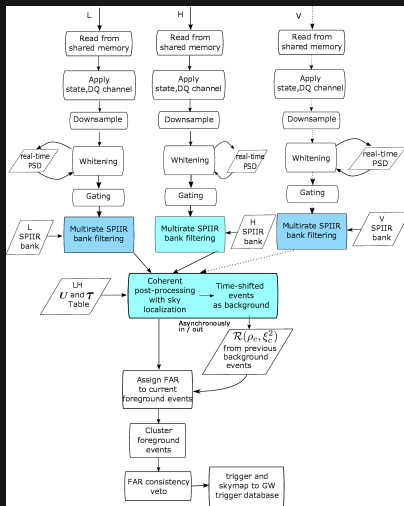
Detectors



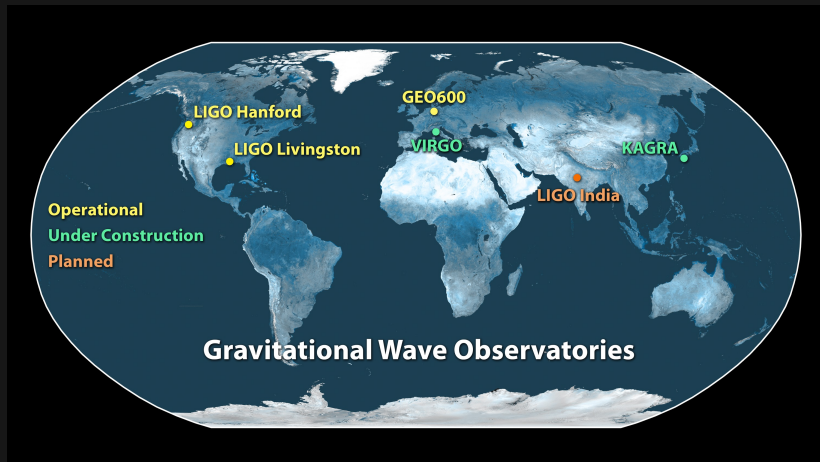
Processing Pipelines



The SPIIR Pipeline



The SPIIR Pipeline



Internal datastructures

```
#define MAX_IFO_COMBOS 7 // 2^3-1
static const IFOType IFOComboMap[MAX_IFO_COMBOS] = {
    {"H1", 0},
    {"L1", 1},
    {"V1", 2},
    {"H1L1", 3},
    {"H1V1", 4},
    {"L1V1", 5},
    {"H1L1V1", 6},
};
```

Internal datastructures

```
#define MAX_IFO_COMBOS 7 // 2^3-1
static const IFOType IFOComboMap[MAX_IFO_COMBOS] = {
    {"H1", 0},
    {"L1", 1},
    {"V1", 2},
    {"K1", 3},
    {"H1L1", 4},
    {"H1V1", 5},
    {"H1K1", 6},
    {"L1V1", 7},
    {"L1H1", 8},
    {"V1H1", 9},
    {"H1L1V1", 10},
    {"H1L1K1", 11},
    {"H1V1K1", 12},
    {"L1V1K1", 13},
    {"H1L1V1K1", 14},
};
```

Internal datastructures

```
#define MAX_IFO_COMBOS 7 // 2^3-1
static const IFOType IFOComboMap[MAX_IFO_COMBOS] = {
    {"H1", 0},
    {"L1", 1},
    {"V1", 2},
    {"H1L1", 3},
    {"H1V1", 4},
    {"L1V1", 5},
    {"H1L1V1", 6},
};
```

The power of powers of 2

0 0 0 0 0

The power of powers of 2

0 0 0 0 1

The power of powers of 2

0 0 1 0 1

The power of powers of 2

1 0 1 0 1

The power of powers of 2

```
#define MAX_IFO_COMBOS 7 //  $2^3-1$ 
static const IFOType IFOComboMap[MAX_IFO_COMBOS] = {
    { "H1", 0 },
    { "L1", 1 },
    { "H1L1", 2 },
    { "V1", 3 },
    { "H1V1", 4 },
    { "L1V1", 5 },
    { "H1L1V1", 6 },
};
```


The power of powers of 2


```
#define MAX_IFO_COMBOS 7 // 2^3-1
static const IFOType IFOComboMap[MAX_IFO_COMBOS] = {
    { "H1", 0 },
    { "L1", 1 },
    { "H1L1", 2 },
    { "V1", 3 },
    { "H1V1", 4 },
    { "L1V1", 5 },
    { "H1L1V1", 6 },
    { "K1", 7 },
    { "H1K1", 8 },
    { "L1K1", 9 },
    { "H1L1K1", 10 },
    { "V1K1", 11 },
    { "H1V1K1", 12 },
    { "L1V1K1", 13 },
    { "H1L1V1K1", 14 },
};
```

The power of powers of 2

$N_{detectors} = \text{popcount}(\text{icombo})$
Detector being used: $\text{icombo} \& 2^{ifo}$



The power of powers of 2

lscsoft > splir > Merge Requests > 19

Merged Opened 1 month ago by  Thomas Almeida Edit

Improve IFOComboMap to be powers of two and reduce size of cohfar's XML output

Overview 4 Commits 2 **Changes** 10

 Compare master ▾ and latest version ▾ 10 files +401 -415 Expand all  ▾

The hunt for hardcoded detectors

```
typedef struct tagPostcohInspiralTable {  
    struct tagPostcohInspiralTable *next;  
    long process_id;  
    long event_id;  
    LIGOTimeGPS end_time;  
    LIGOTimeGPS end_time_H;  
    LIGOTimeGPS end_time_L;  
    LIGOTimeGPS end_time_V;
```

The hunt for hardcoded detectors

```
typedef struct tagPostcohInspiralTable {  
    struct tagPostcohInspiralTable *next;  
    long process_id;  
    long event_id;  
    LIGOTimeGPS end_time;  
    LIGOTimeGPS end_time_sngl[MAX_NIFO];  
}
```

The hunt for hardcoded detectors

lscsoft > spiir > Merge Requests > 113

Merged Opened 3 weeks ago by  Thomas Almeida [Edit](#)

postcohtable: Remove detector names from table

Overview 0 Commits 1 **Changes** 22

 Compare master ▾ and latest version ▾  22 files **+860** **-1165** Expand all  ▾

Complexity analysis

Complexity Analysis

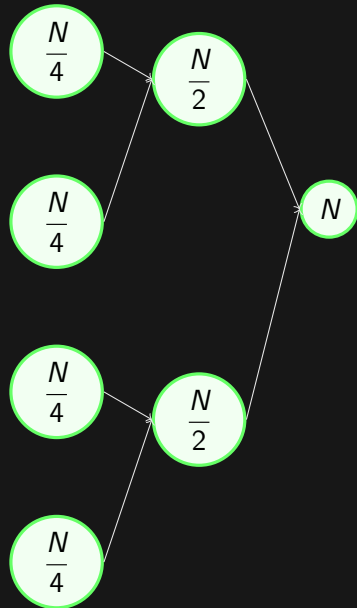
Asymptotic growth rates

- compare algorithms
- compare with inherent performance barriers
- provide simple closed form approximations
- big O — upper bounds on growth
- big Ω — lower bounds on growth

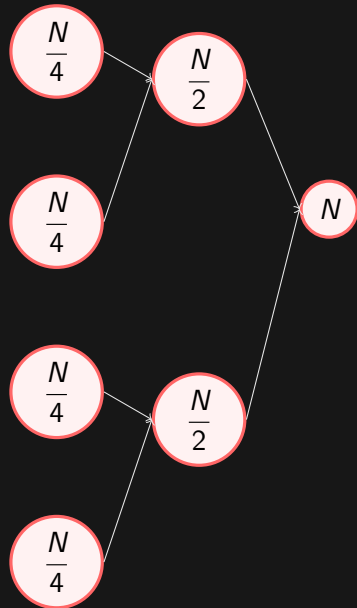
Analysis of recursive programs

- express as recurrence relation
- look for pattern to find closed form
- can then do asymptotic analysis

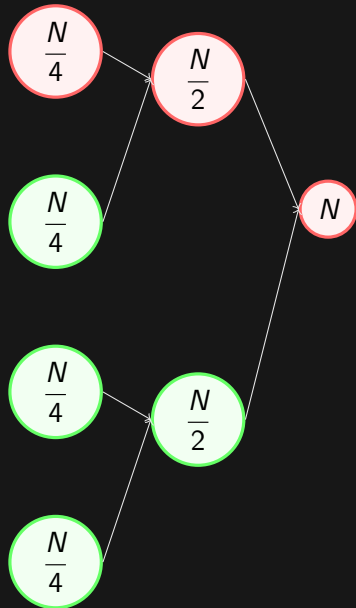
Complexity Analysis



Work



Span

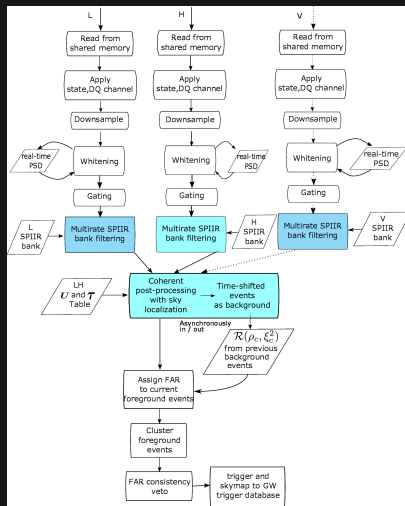


Parallel definitions

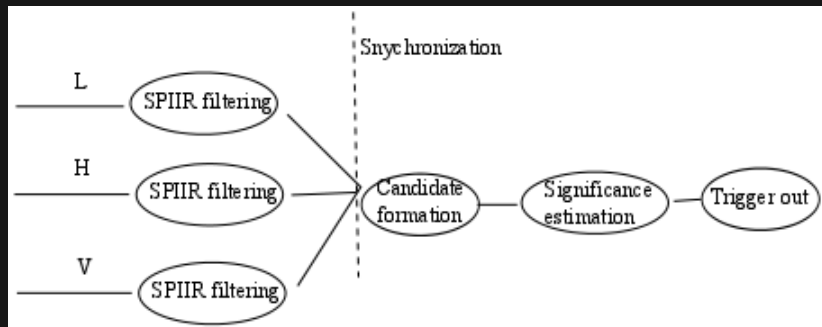
- ▶ Work = T_1
- ▶ Span = T_∞
- ▶ Brent's Theorem
 $O(\frac{T_1}{P} + T_N)$

So why do we care about this complexity analysis?

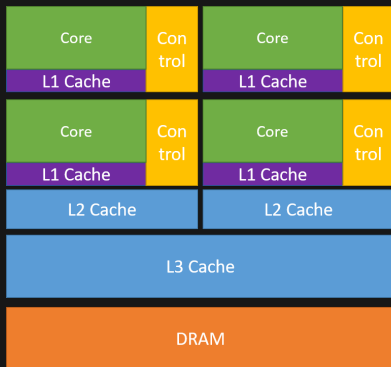
Why do we care?



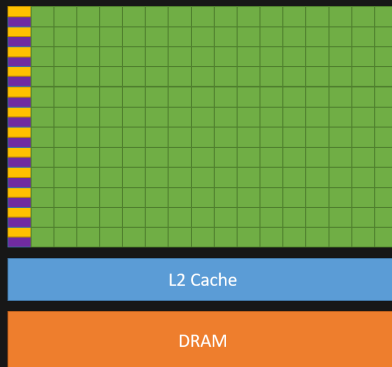
Why do we care?



CUDA

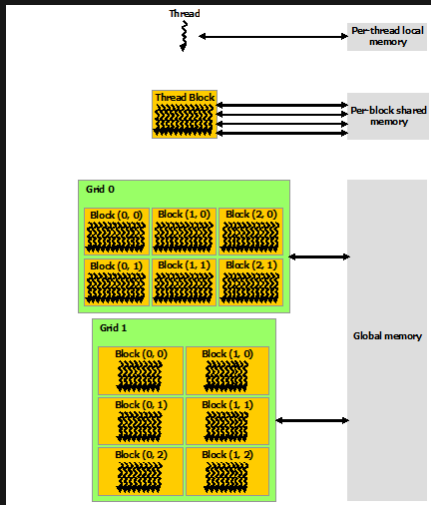


CPU



GPU

Compute model



Callgraph generation

```
tommoa:~/Documents/research# [master] python3 -m utils --help
usage: utils [-h] {combine,co,iir,dot,pipeline} ...
```

Utilities to help with Tom Almeida's GENG5551 research.

optional arguments:

-h, --help show this help message and exit

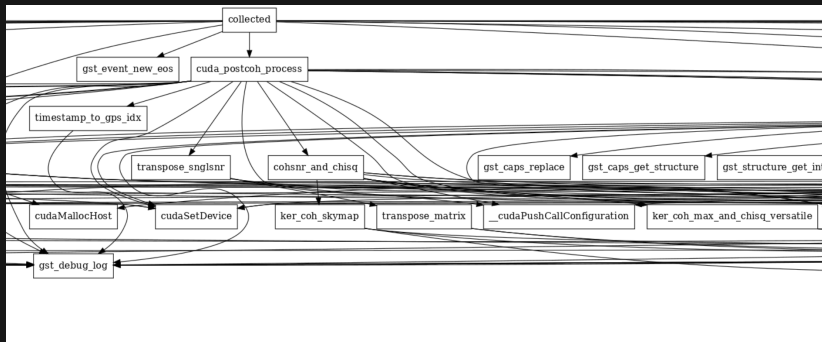
subcommands:

{combine,co,iir,dot,pipeline}

subcommand help

```
tommoa:~/Documents/research# [master]
```

Callgraph



Analysis

$$T_{\infty} = O(NT + N^2 + D^3 + D^2 \log N + D \log S + D \log P),$$

$$T_1 = O(NT + N^2 + SPD^3 + SPBD^3 + ND^2),$$

where D is the number of detectors,

S is the number of sky directions,

T is the number of templates,

N is the number of samples,

B is the number of time shifts made to background noise;

and $P = \max\{S, T\}$.

Analysis

$$T_{\infty} = O(NT + N^2 + D^3 + D^2 \log N + D \log S + D \log P),$$

$$T_1 = O(NT + N^2 + SPD^3 + SPBD^3 + ND^2),$$

where D is the number of detectors,

S is the number of sky directions,

T is the number of templates,

N is the number of samples,

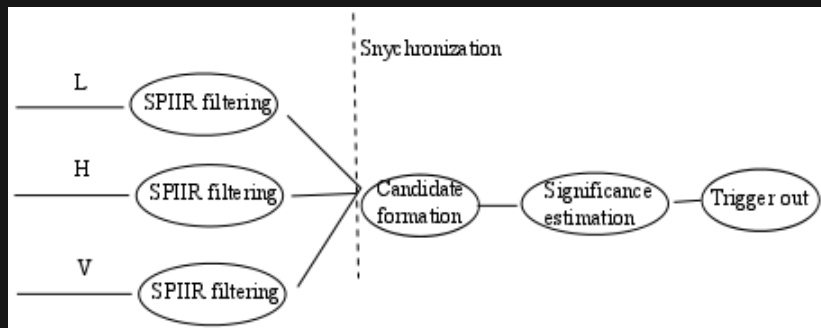
B is the number of time shifts made to background noise;

and $P = \max\{S, T\}$.

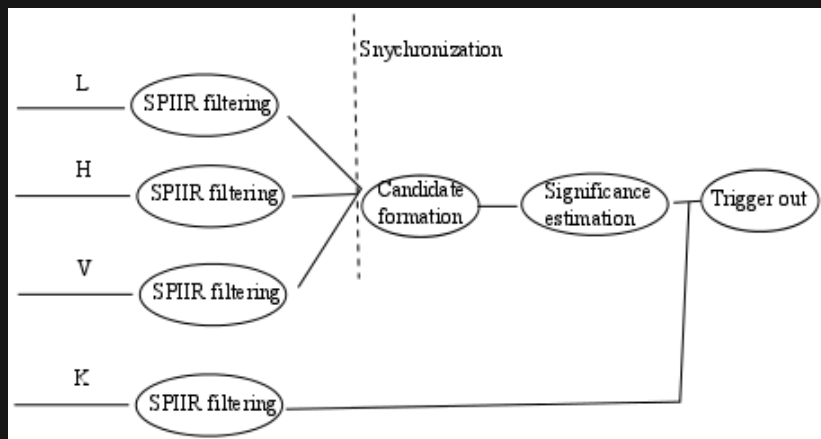
$$\frac{4^3}{3^3} = \frac{64}{27} \approx 2.37$$

Future Work

Pipeline Structure



Pipeline Structure



Algorithms

$$T_{\infty} = O(NT + N^2 + D^3 + D^2 \log N + D \log S + D \log P),$$

$$T_1 = O(NT + N^2 + SPD^3 + SPBD^3 + ND^2),$$

where D is the number of detectors,

S is the number of sky directions,

T is the number of templates,

N is the number of samples,

B is the number of time shifts made to background noise;

and $P = \max\{S, T\}$.

Algorithms

$$T_{\infty} = O(NT + N^2 + D^3 + D^2 \log N + D \log S + D \log P),$$

$$T_1 = O(NT + N^2 + SPD^3 + SPBD^3 + ND^2),$$

where D is the number of detectors,

S is the number of sky directions,

T is the number of templates,

N is the number of samples,

B is the number of time shifts made to background noise;

and $P = \max\{S, T\}$.

$$\frac{4^2}{3^2} = \frac{16}{9} \approx 1.78$$

Questions?