SBC
Sociedade Brasileira de Computação

WSCAD 2023

WSCAD 2023

XXIV Simpósio em Sistemas Computacionais de Alto Desempenho

17 a 20 de outubro, 2023 — Porto Alegre, Brasil

Avaliação de estilos de código para árvore de decisão em GPU com Microbenchmarks

Jeronimo Penha, Alysson Kelvim, Olavo Silva, Icaro Moreira, José A. Nacif, **Ricardo Ferreira**- Universidade Federal de Viçosa

CNPq

FAPEMIG

CAPES

Summary

- Random Forest and GPU
- Three Implementation: IF, **without IF**, Memory
- Which is the best ? Depends on Depth and Trees
- Conclusions

Random Forest

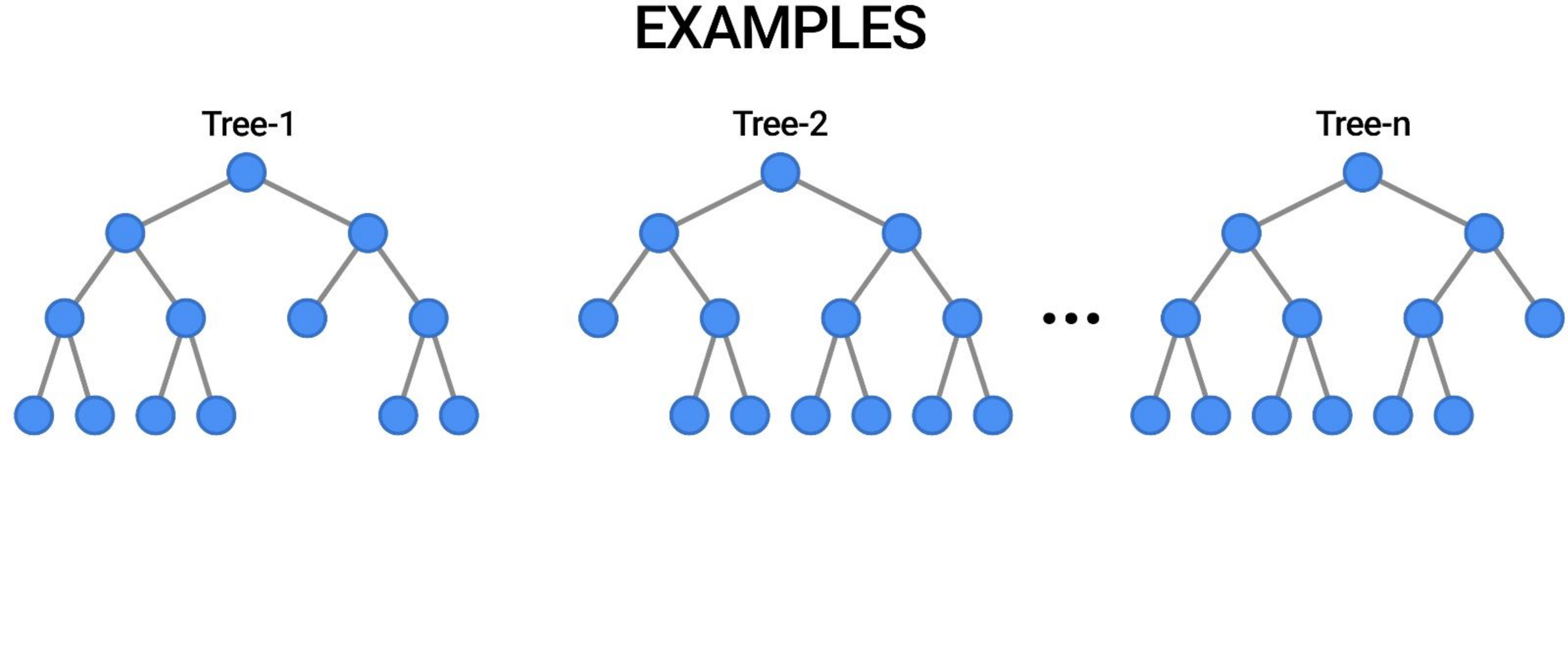
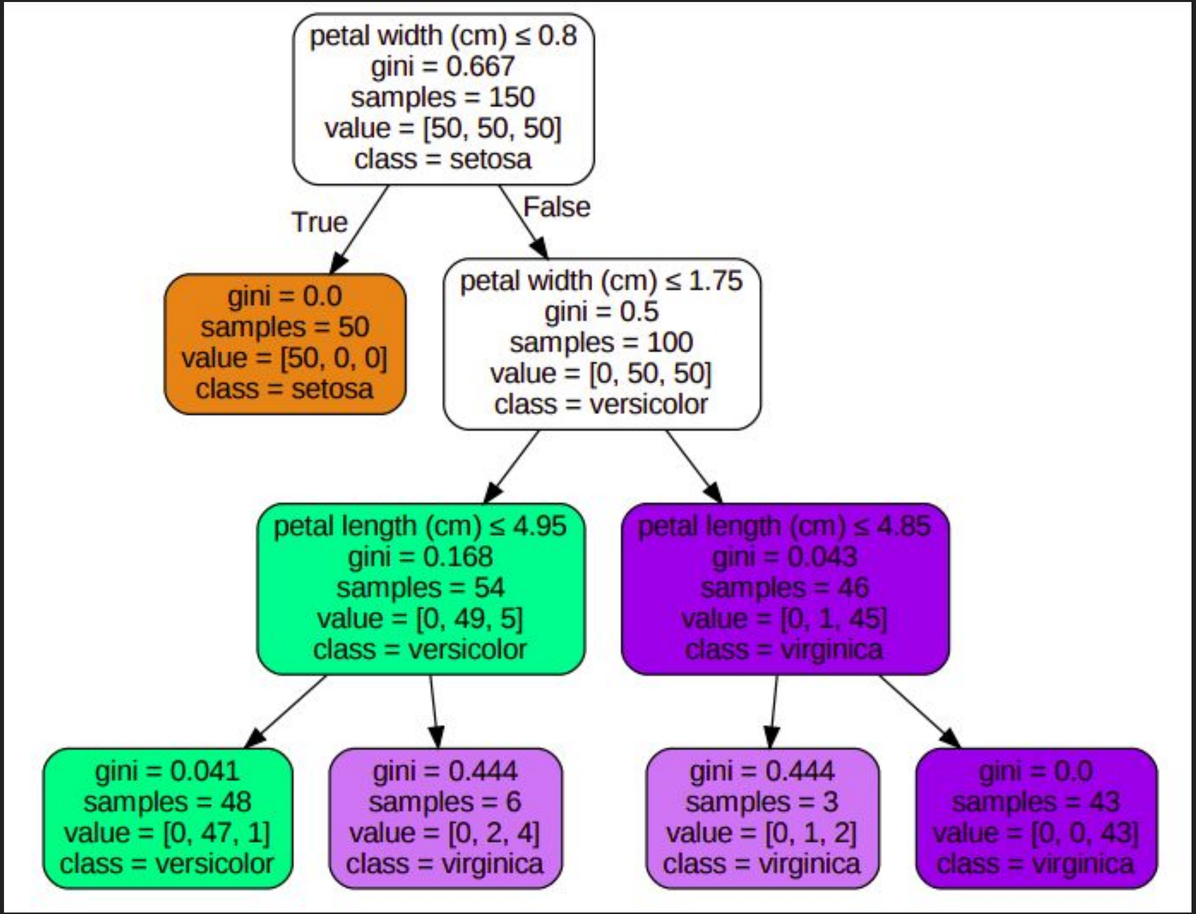


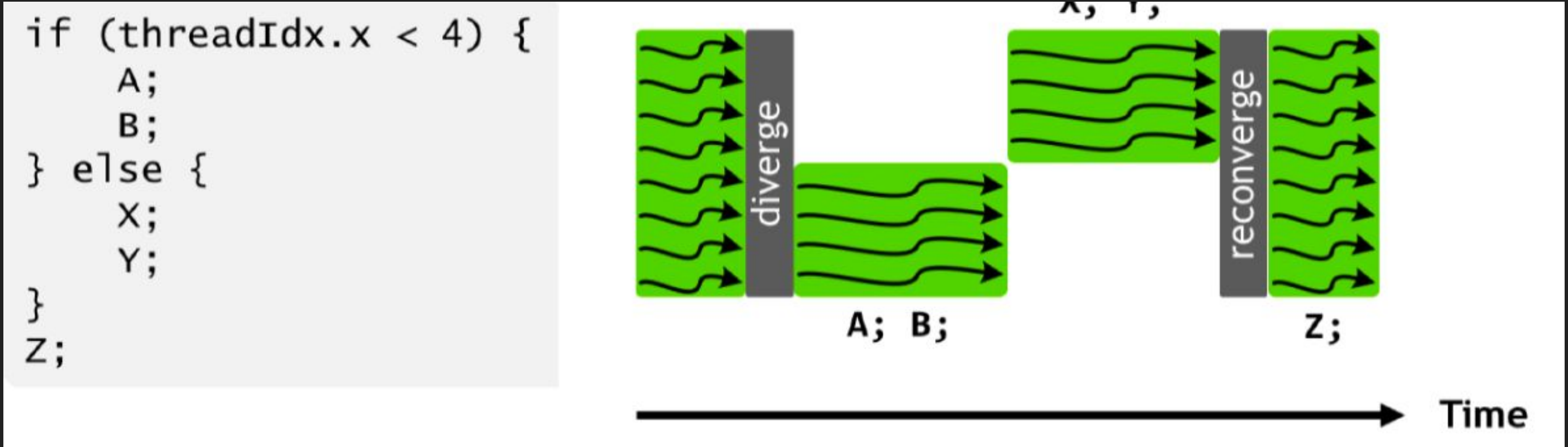
Figure from Tensor Flow Blog

Random Forest



```
 if (x[3] <= 0.80)
      return 0;
else {
      if (x[3] <= 1.75) {
            if (x[2] <= 4.95)
                  return 1;
            else
                  return 2;
      } else {
            if (x[2] <= 4.850)
                  return 2;
            else
                  return 2;
      }
```

GPU and Branches



```
if (threadIdx.x < 4) {
      A;
      B;
} else {
      X;
      Y;
}
Z;
```

Figure extracted from here

GPU and Branches

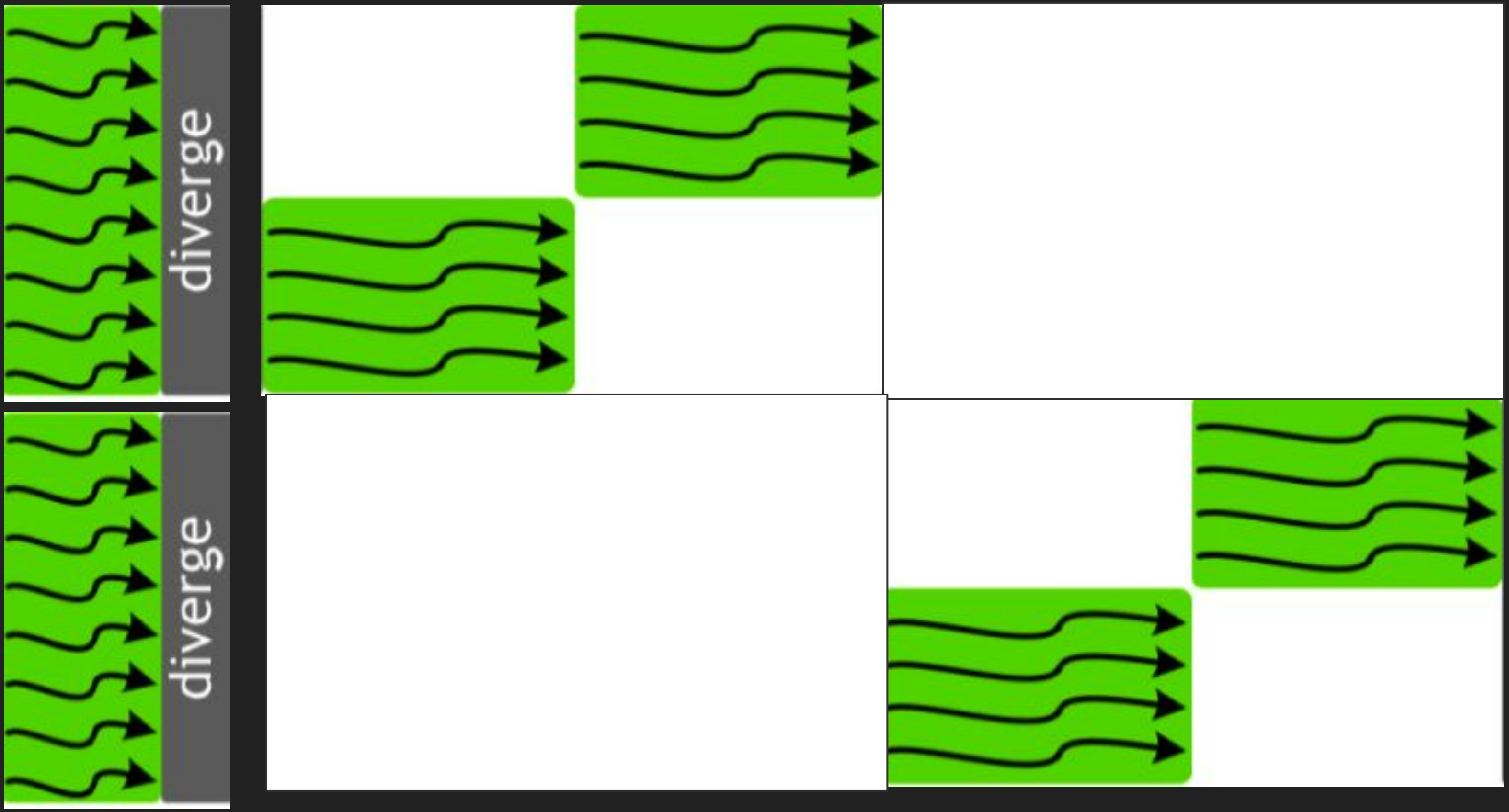
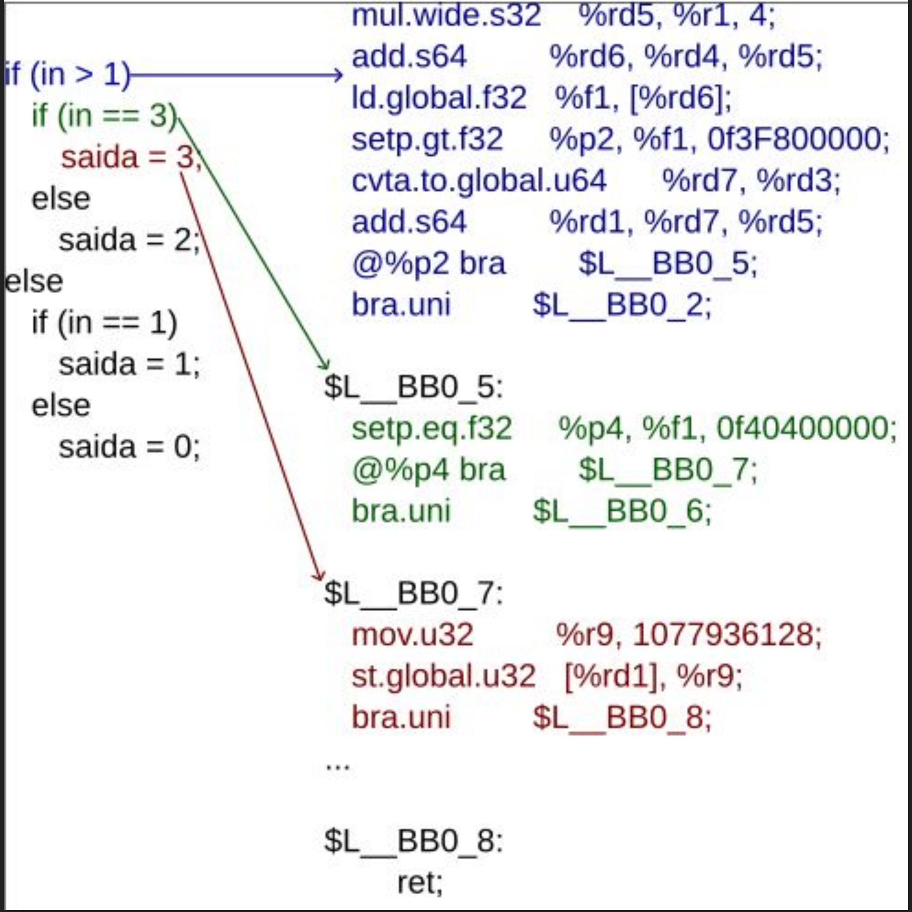


Figure adapted from here

PTX IF



```
if (in > 1)
  if (in == 3)
    saida = 3;
  else
    saida = 2;
else
  if (in == 1)
    saida = 1;
  else
    saida = 0;
```

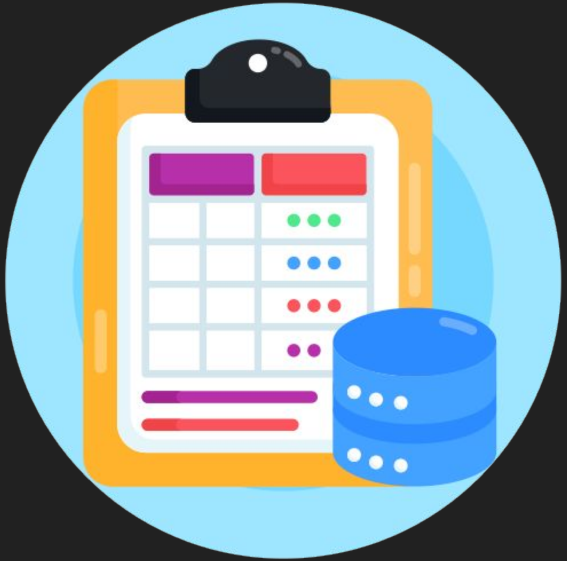
```
  mul.wide.s32    %rd5, %r1, 4;
  add.s64         %rd6, %rd4, %rd5;
  ld.global.f32   %f1, [%rd6];
  setp.gt.f32     %p2, %f1, 0f3F800000;
  cvta.to.global.u64      %rd7, %rd3;
  add.s64         %rd1, %rd7, %rd5;
  @%p2 bra        $L__BB0_5;
  bra.uni         $L__BB0_2;

$L__BB0_5:
  setp.eq.f32     %p4, %f1, 0f40400000;
  @%p4 bra        $L__BB0_7;
  bra.uni         $L__BB0_6;

$L__BB0_7:
  mov.u32         %r9, 1077936128;
  st.global.u32   [%rd1], %r9;
  bra.uni         $L__BB0_8;
...

$L__BB0_8:
    ret;
```

Faster approach for more than 7 levels

- no indirection
- simple branch

Research Question

- Is there an alternative approach to implementing trees without branching?

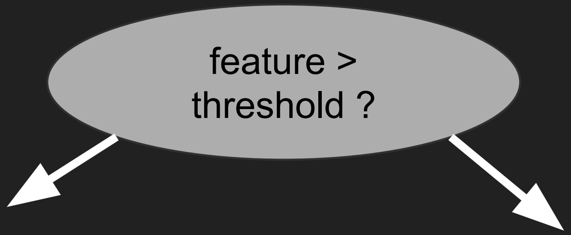


When is it advantageous to utilize a GPU?

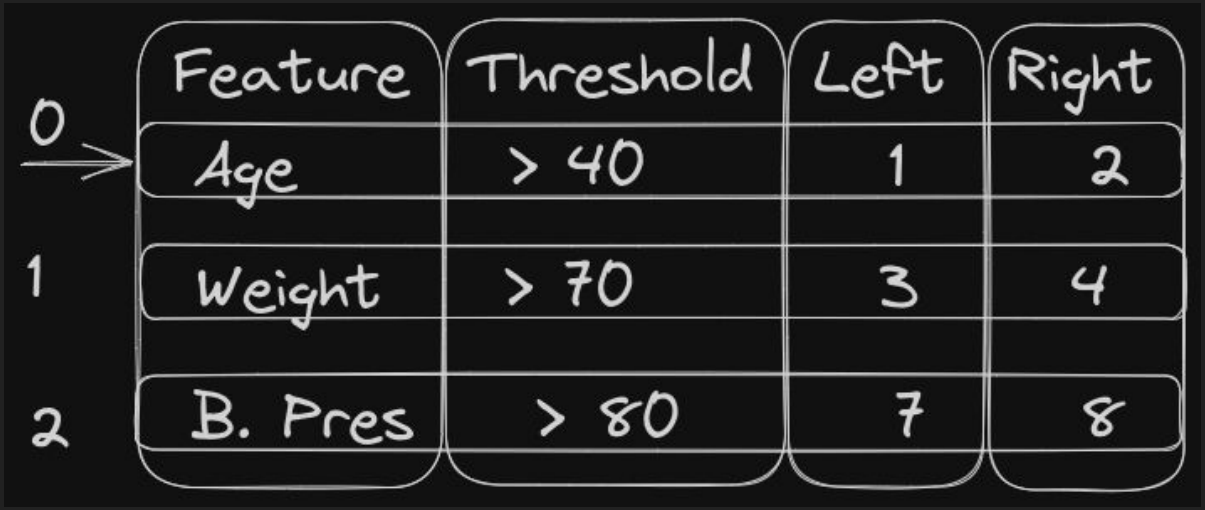
- Over 5 million registers
- Numerous memory types and units
- Thousands of computing units

Decision Tree as a table

- Used in FPGAs, No divergence, same code for all threads



feature > threshold ?

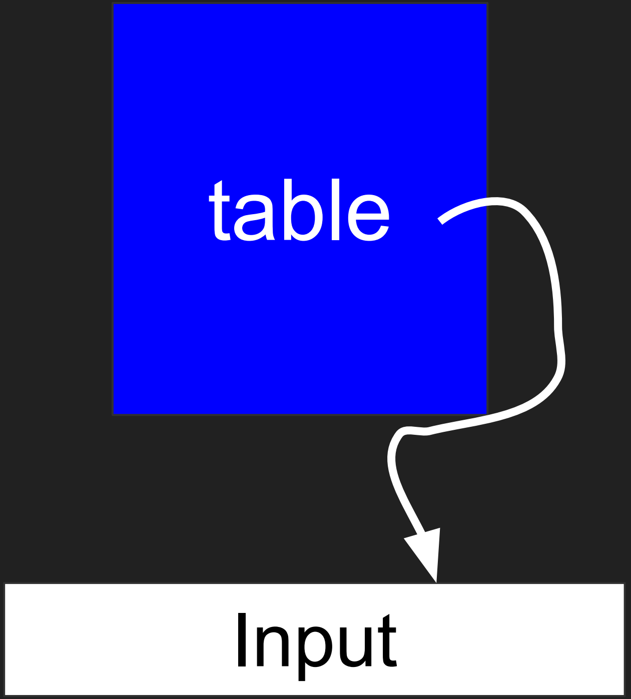


	Feature	Threshold	Left	Right
0 →	Age	> 40	1	2
1	Weight	> 70	3	4
2	B. Pres	> 80	7	8

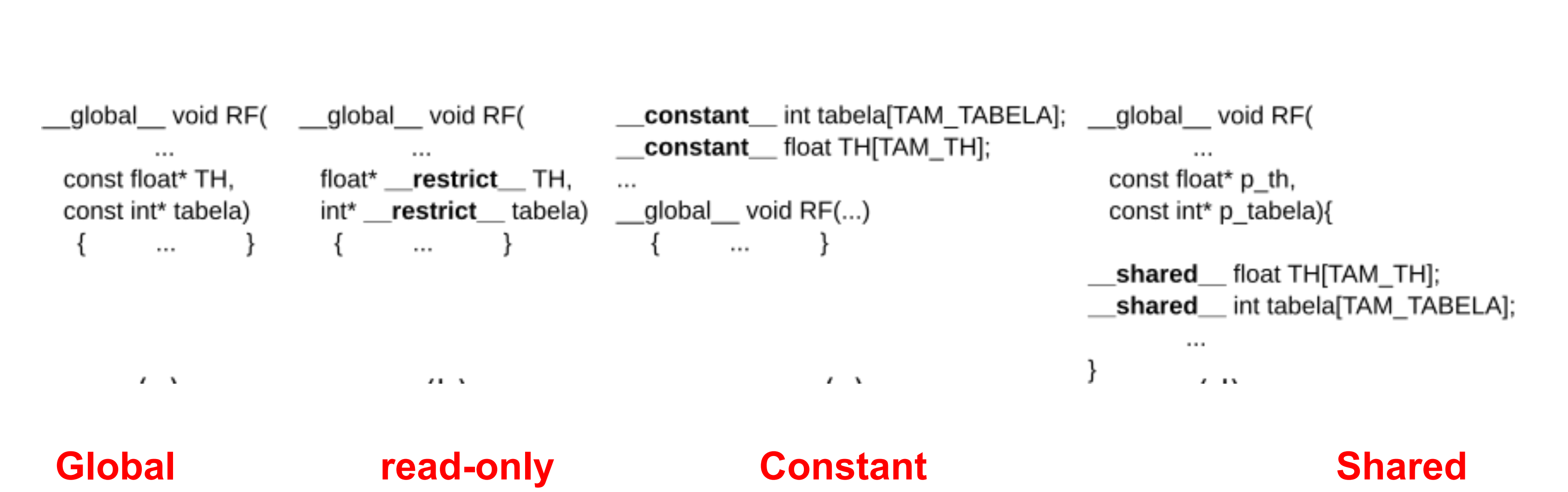
```
Index = 0
while not leaf do
        F= table.feature(index)
        data = Input(F)
        T = table.Threshold(index)
        Index = ( data > T) ?
                    table.left(index):
                    table.right(index);
```

Decision Tree as a table

```
Index = 0
while not leaf do
    F= table.feature(index)
    data = Input(F)
    T = table.Threshold(index)
    Index = ( data > T) ?
            table.left(index):
            table.right(index);
```



Decision Tree as a table



```
__global__ void RF(
          ...
   const float* TH,
   const int* tabela)
    {      ...      }
```

(a)

Global

```
__global__ void RF(
          ...
   float* __restrict__ TH,
   int* __restrict__ tabela)
    {      ...      }
```

(b)

read-only

```
__constant__ int tabela[TAM_TABELA];
__constant__ float TH[TAM_TH];
...
__global__ void RF(...)
    {      ...      }
```

(c)

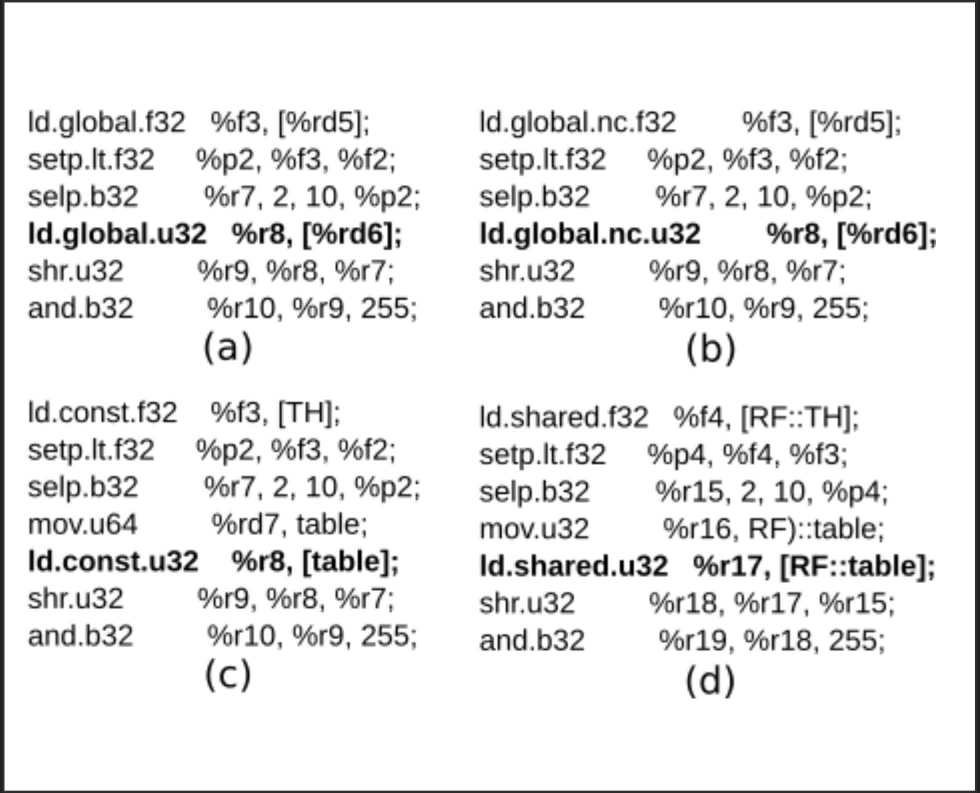
Constant

```
__global__ void RF(
          ...
   const float* p_th,
   const int* p_tabela){

__shared__ float TH[TAM_TH];
__shared__ int tabela[TAM_TABELA];
         ...
}
```

(d)

Shared

PTX Table



```
ld.global.f32   %f3, [%rd5];
setp.lt.f32     %p2, %f3, %f2;
selp.b32        %r7, 2, 10, %p2;
ld.global.u32   %r8, [%rd6];
shr.u32         %r9, %r8, %r7;
and.b32         %r10, %r9, 255;
```

(a)

```
ld.global.nc.f32      %f3, [%rd5];
setp.lt.f32      %p2, %f3, %f2;
selp.b32         %r7, 2, 10, %p2;
ld.global.nc.u32      %r8, [%rd6];
shr.u32          %r9, %r8, %r7;
and.b32          %r10, %r9, 255;
```

(b)

```
ld.const.f32    %f3, [TH];
setp.lt.f32     %p2, %f3, %f2;
selp.b32        %r7, 2, 10, %p2;
mov.u64         %rd7, table;
ld.const.u32    %r8, [table];
shr.u32         %r9, %r8, %r7;
and.b32         %r10, %r9, 255;
```

(c)

```
ld.shared.f32   %f4, [RF::TH];
setp.lt.f32     %p4, %f4, %f3;
selp.b32        %r15, 2, 10, %p4;
mov.u32         %r16, RF)::table;
ld.shared.u32   %r17, [RF::table];
shr.u32         %r18, %r17, %r15;
and.b32         %r19, %r18, 255;
```

(d)

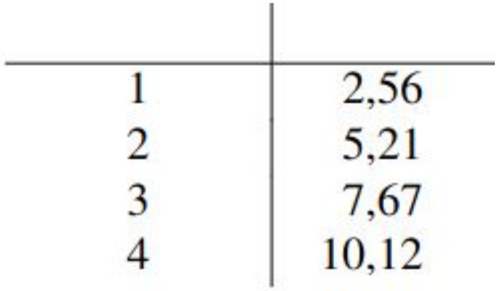
High latency 30 cycles

- indirection
- no divergence
- Slow approach (good for FPGA)

Three level - Global

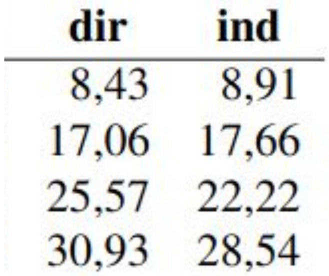
Table in MEMORY

Trees IF



1	2,56
2	5,21
3	7,67
4	10,12

Global



dir	ind
8,43	8,91
17,06	17,66
25,57	22,22
30,93	28,54

Time in Milliseconds - 50 million samples

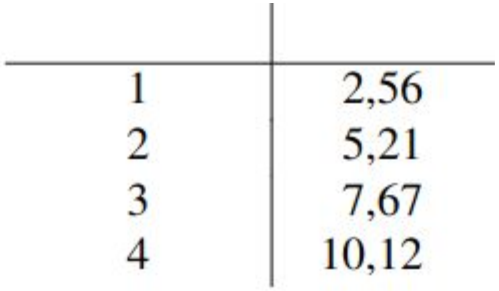
Nvidia GTX 1070

Three level - Read Only

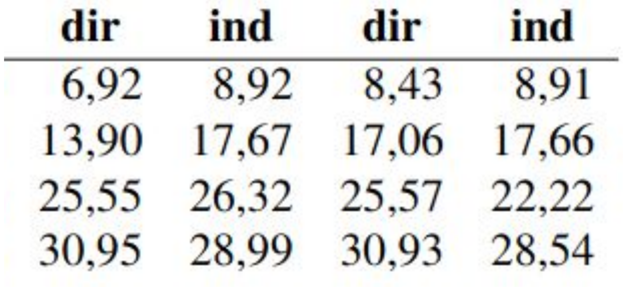
Table in MEMORY

Trees IF

Read-only Global



1	2,56
2	5,21
3	7,67
4	10,12

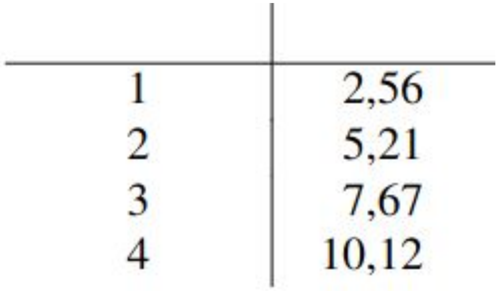


dir	ind	dir	ind
6,92	8,92	8,43	8,91
13,90	17,67	17,06	17,66
25,55	26,32	25,57	22,22
30,95	28,99	30,93	28,54

Time in Milliseconds - 50 million samples

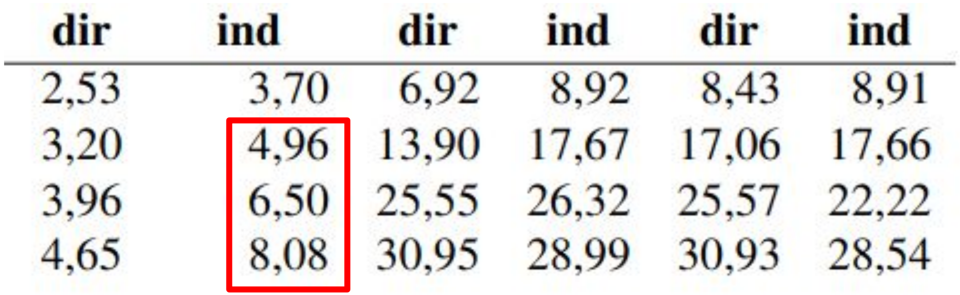
Nvidia GTX 1070

Three level   Shared



Trees	IF
1	2,56
2	5,21
3	7,67
4	10,12

Table in MEMORY



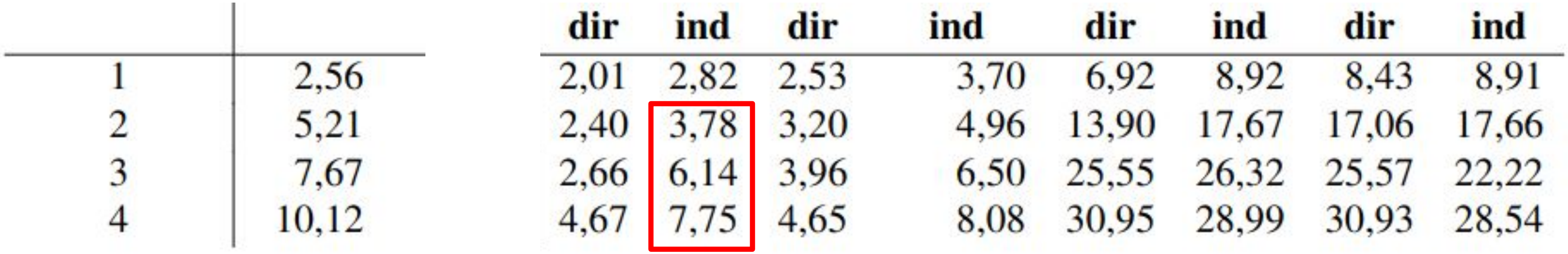
Shared		Read-only		Global	
dir	ind	dir	ind	dir	ind
2,53	3,70	6,92	8,92	8,43	8,91
3,20	4,96	13,90	17,67	17,06	17,66
3,96	6,50	25,55	26,32	25,57	22,22
4,65	8,08	30,95	28,99	30,93	28,54

Time in Milliseconds   - 50 million samples

Nvidia GTX 1070

Three level Const

Table in MEMORY



Trees	IF	Const		Shared		Read-only		Global	
		dir	ind	dir	ind	dir	ind	dir	ind
1	2,56	2,01	2,82	2,53	3,70	6,92	8,92	8,43	8,91
2	5,21	2,40	3,78	3,20	4,96	13,90	17,67	17,06	17,66
3	7,67	2,66	6,14	3,96	6,50	25,55	26,32	25,57	22,22
4	10,12	4,67	7,75	4,65	8,08	30,95	28,99	30,93	28,54

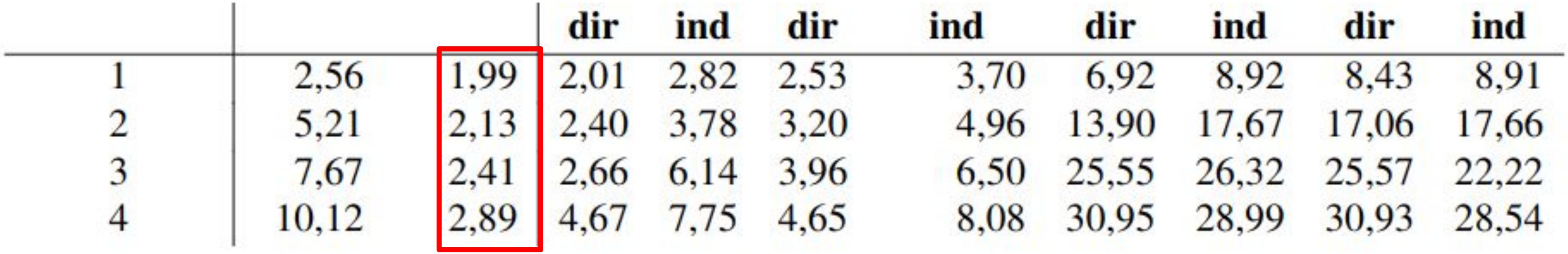
Time in Milliseconds - 50 million samples

Nvidia GTX 1070

Three level   NoIF

WSCAD 2023

Table in MEMORY

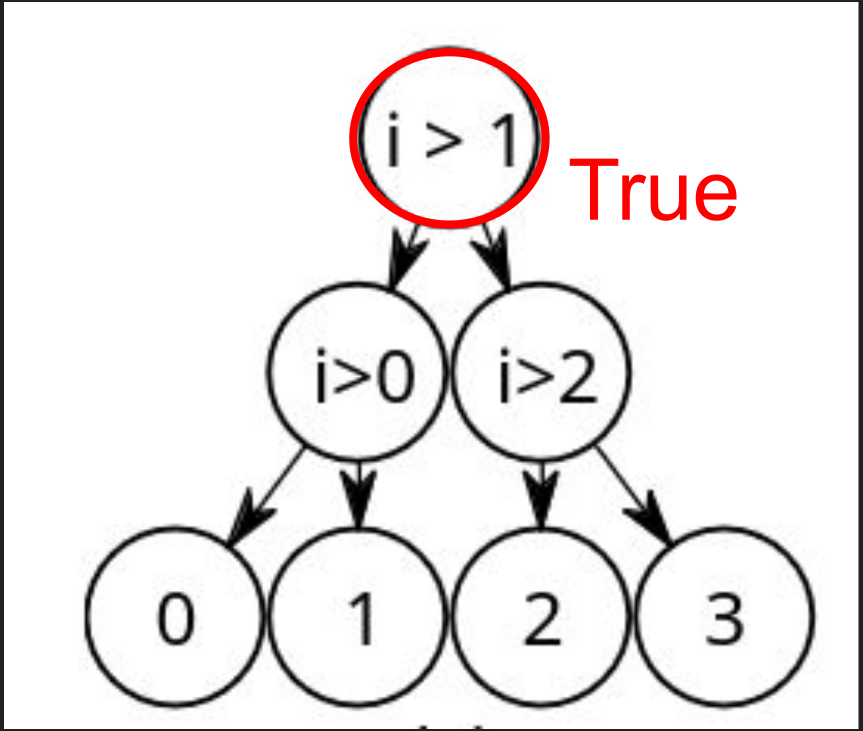


Trees	IF	noIF	Const		Shared		Read-only		Global	
			dir	ind	dir	ind	dir	ind	dir	ind
1	2,56	1,99	2,01	2,82	2,53	3,70	6,92	8,92	8,43	8,91
2	5,21	2,13	2,40	3,78	3,20	4,96	13,90	17,67	17,06	17,66
3	7,67	2,41	2,66	6,14	3,96	6,50	25,55	26,32	25,57	22,22
4	10,12	2,89	4,67	7,75	4,65	8,08	30,95	28,99	30,93	28,54

Time in Milliseconds   - 50 million samples

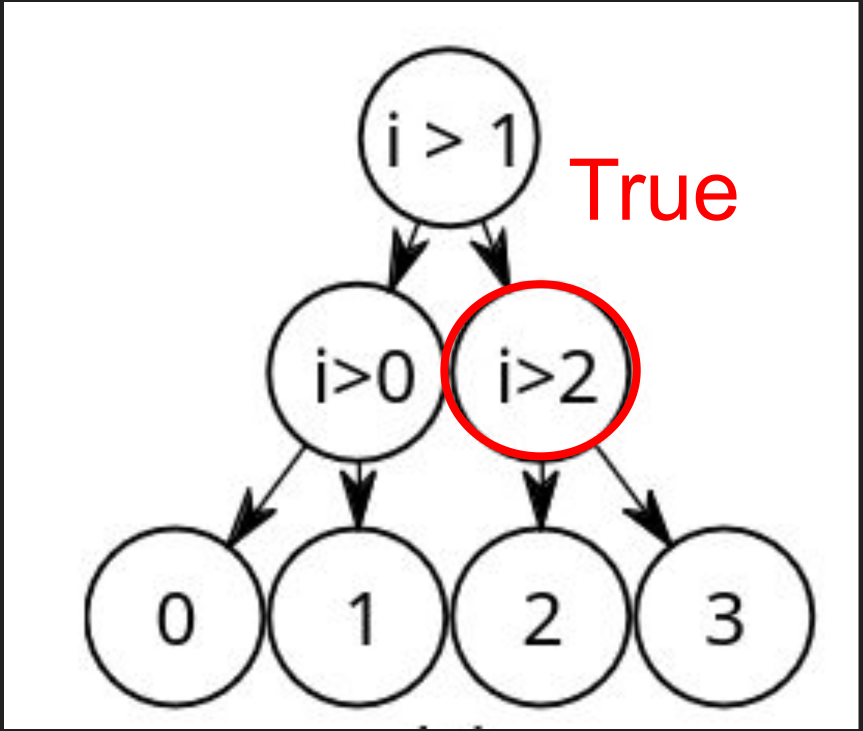
Nvidia GTX 1070

Proposal to implement Without IF



```
Root = ( in > 1 )

leaf = root * ( 2 + (in >2))

leaf += ~~(1-root)~~*(in>0)

Output = leaf
```

Proposal to implement Without IF



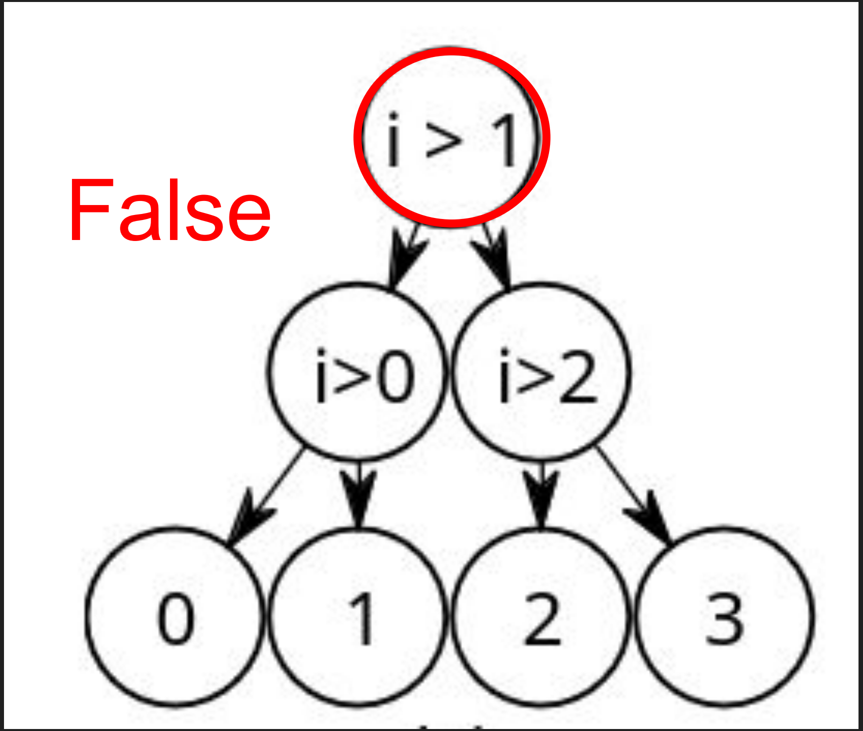
Root = (in > 1)

leaf = root * (2 + (in >2))

leaf += ~~(1-root)*~~(in>0)

Output = leaf

Proposal to implement Without IF



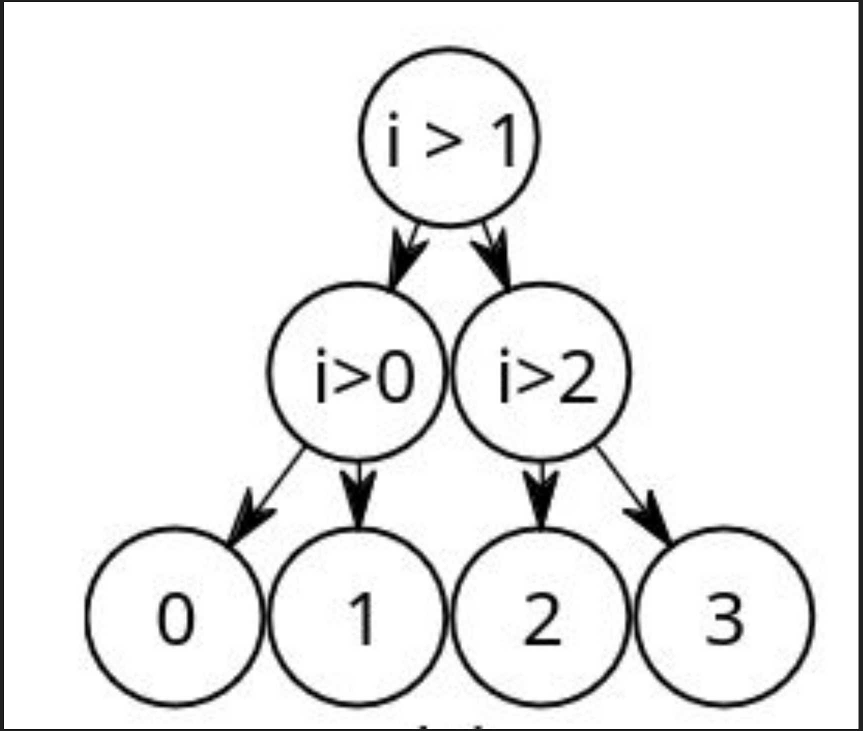
Root = (in > 1)

leaf = ~~root * (2 + (in >2))~~

leaf += (1-root)*(in>0)

Output = leaf

Three Comparisons + 5 add/mult !!!



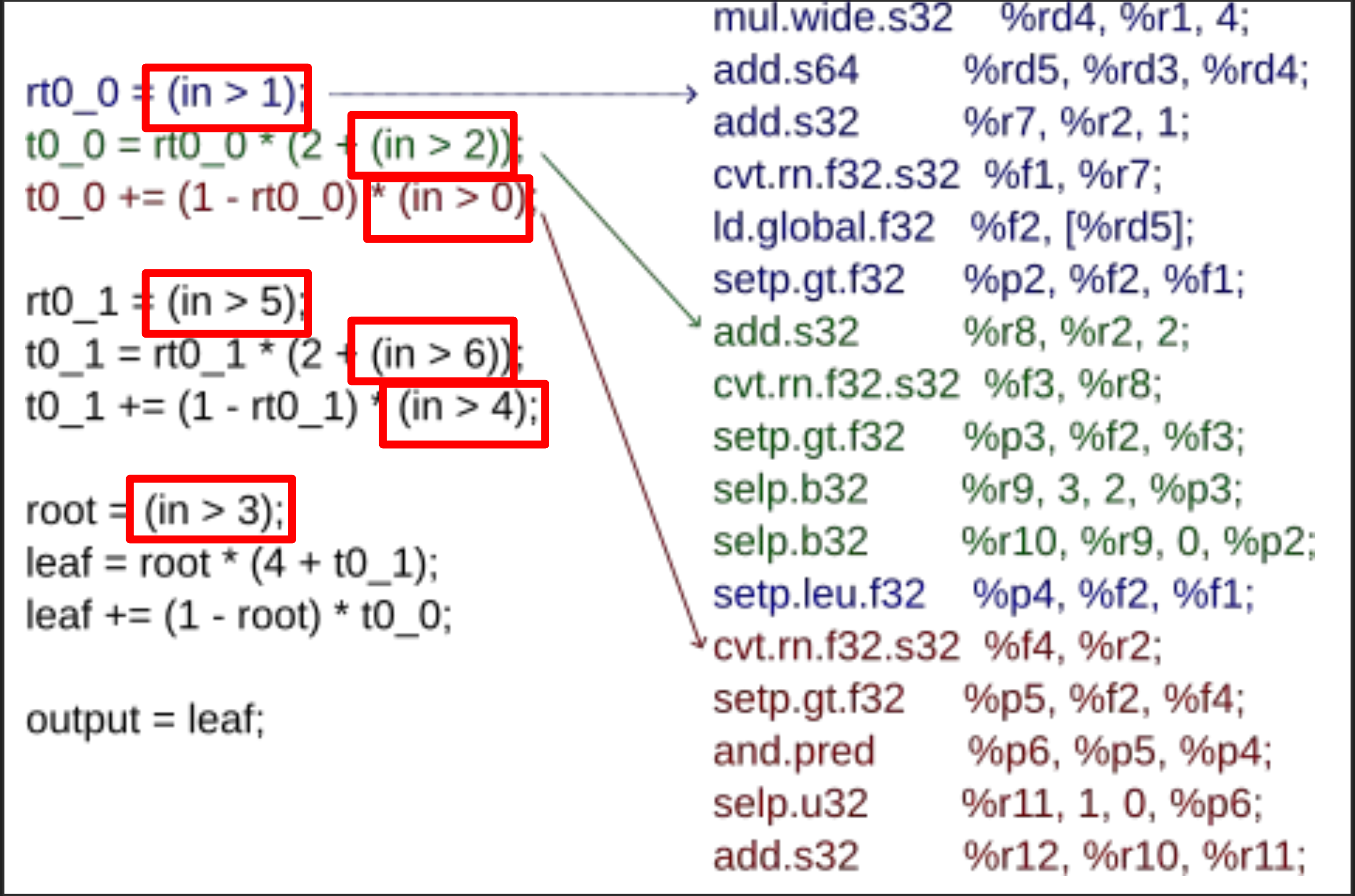
Root = (in > 1)

leaf = root * (2 + (in >2))

leaf += (1-root)*(in>0)

Output = leaf

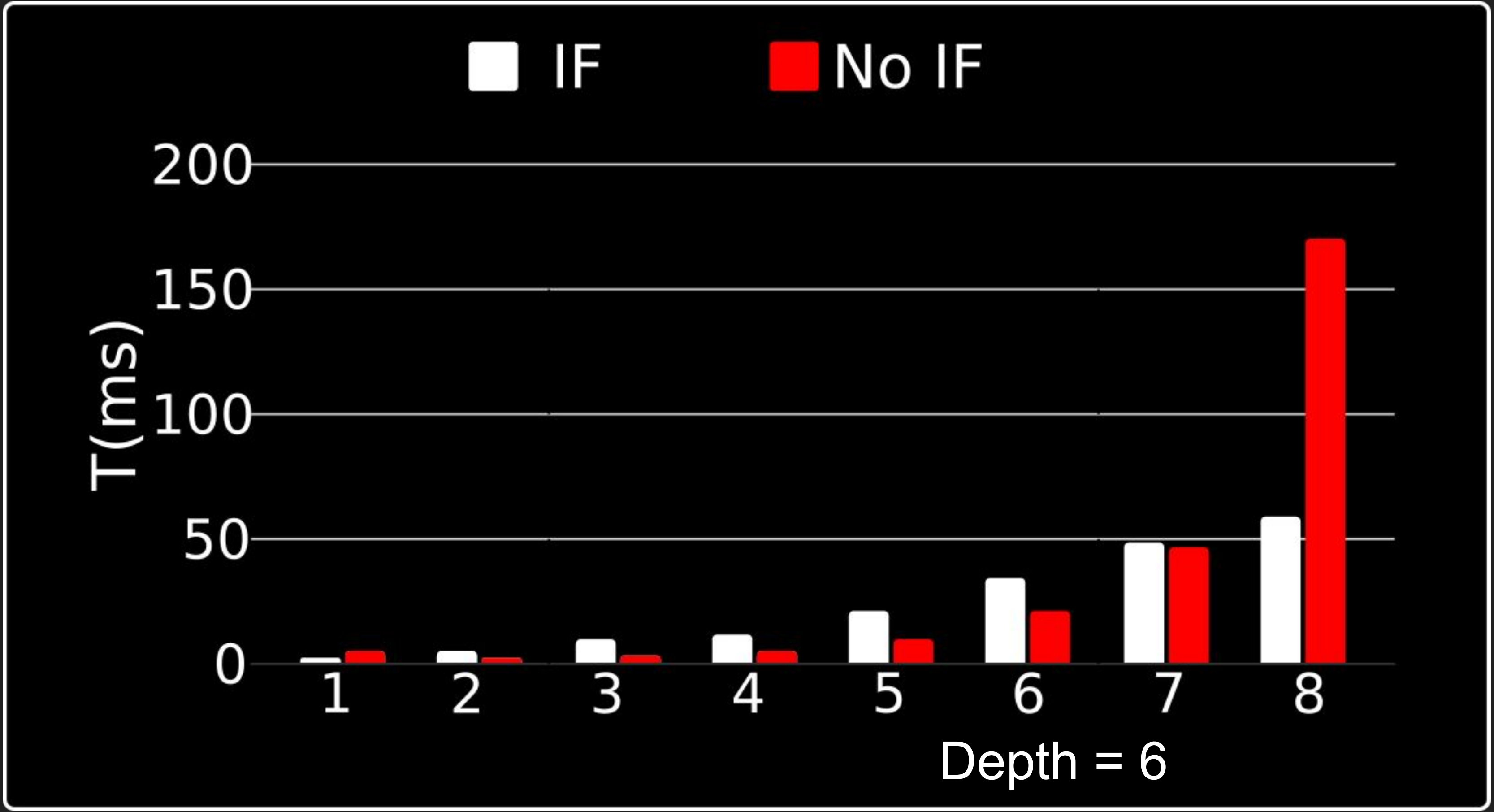
Seven Comparisons…



```
rt0_0 = (in > 1);
t0_0 = rt0_0 * (2 + (in > 2));
t0_0 += (1 - rt0_0) * (in > 0);

rt0_1 = (in > 5);
t0_1 = rt0_1 * (2 + (in > 6));
t0_1 += (1 - rt0_1) * (in > 4);

root = (in > 3);
leaf = root * (4 + t0_1);
leaf += (1 - root) * t0_0;

output = leaf;
```

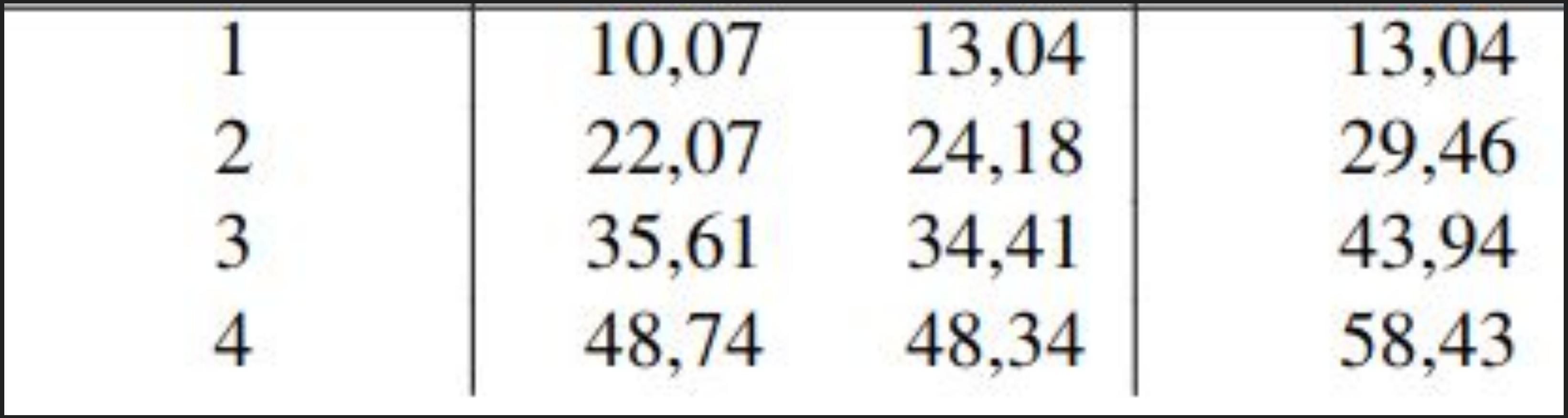
```
mul.wide.s32    %rd4, %r1, 4;
add.s64         %rd5, %rd3, %rd4;
add.s32         %r7, %r2, 1;
cvt.rn.f32.s32  %f1, %r7;
ld.global.f32   %f2, [%rd5];
setp.gt.f32     %p2, %f2, %f1;
add.s32         %r8, %r2, 2;
cvt.rn.f32.s32  %f3, %r8;
setp.gt.f32     %p3, %f2, %f3;
selp.b32        %r9, 3, 2, %p3;
selp.b32        %r10, %r9, 0, %p2;
setp.leu.f32    %p4, %f2, %f1;
cvt.rn.f32.s32  %f4, %r2;
setp.gt.f32     %p5, %f2, %f4;
and.pred        %p6, %p5, %p4;
selp.u32        %r11, 1, 0, %p6;
add.s32         %r12, %r10, %r11;
```

Which is better IF or No IF ?



Nvidia GTX 1070

Depth == 7 ? Time in ms



Trees	IF	No IF	memory
1	10,07	13,04	13,04
2	22,07	24,18	29,46
3	35,61	34,41	43,94
4	48,74	48,34	58,43

Nvidia GTX 1070

Which is better:
a few deep trees or numerous shadow trees?

IF implementation

10 Trees Depth 5 = 53.31ms 2.2x faster

5 Tress Depth 10 = 127.31

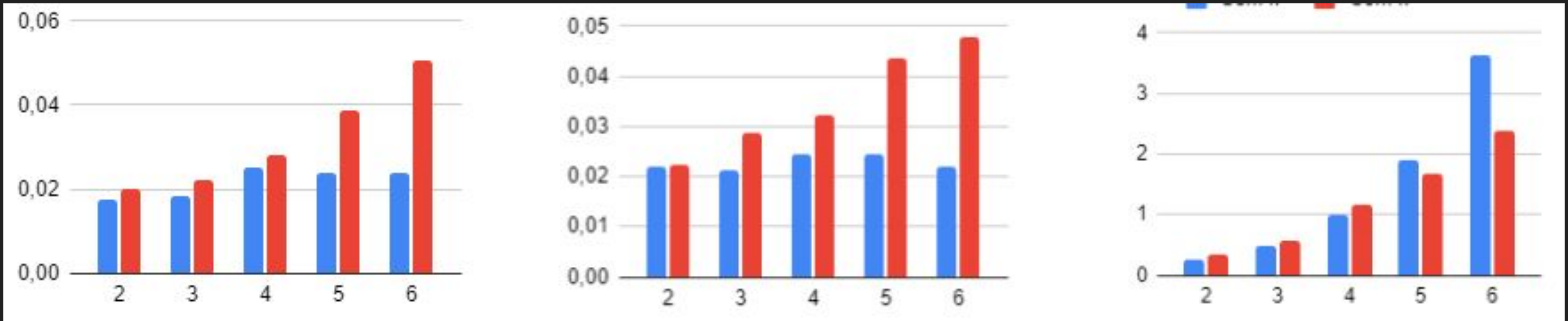
No IF

10 Trees Depth 5 = 33.4ms 3.8x faster !

Real Datasets depth=6

🟦 No IF

🟥 IF

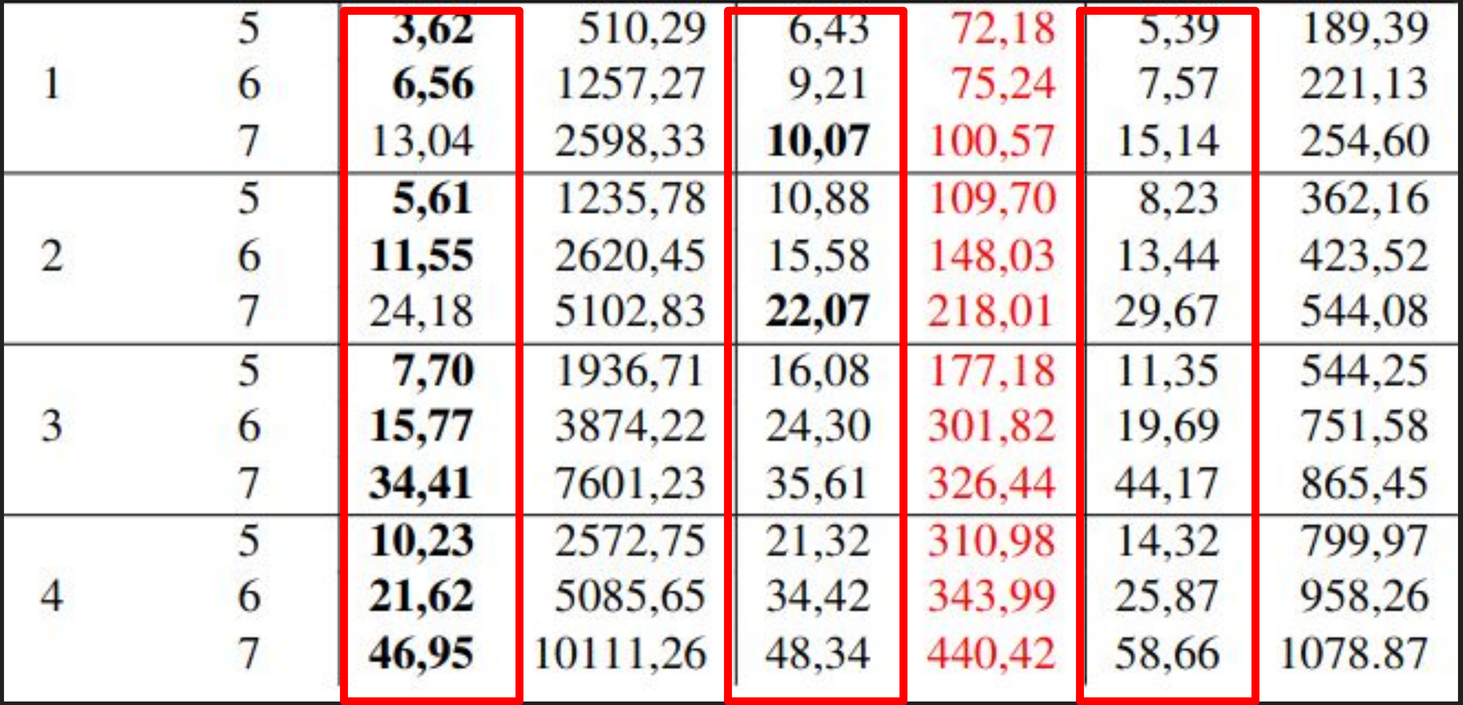


	Hospital	Adult	Susy
features	18	14	18
size	59.557	32.561	50.00.000

WSCAD 2023

GPU versus OpenMP CPU

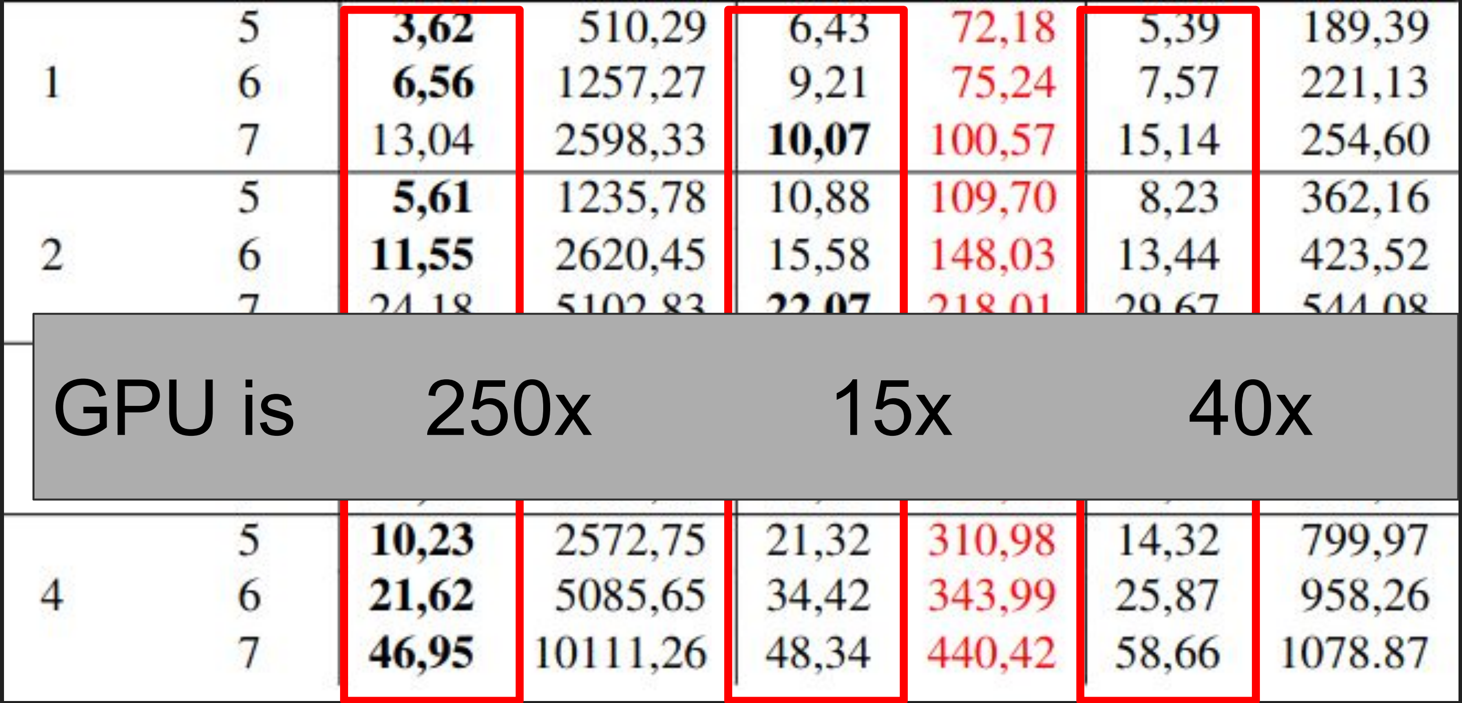
i7-7 3.6G 8 cores 16 Ths



Trees	D	NoIF		IF		Table	
1	5	**3,62**	510,29	6,43	72,18	5,39	189,39
	6	**6,56**	1257,27	9,21	75,24	7,57	221,13
	7	13,04	2598,33	**10,07**	100,57	15,14	254,60
2	5	**5,61**	1235,78	10,88	109,70	8,23	362,16
	6	**11,55**	2620,45	15,58	148,03	13,44	423,52
	7	24,18	5102,83	**22,07**	218,01	29,67	544,08
3	5	**7,70**	1936,71	16,08	177,18	11,35	544,25
	6	**15,77**	3874,22	24,30	301,82	19,69	751,58
	7	**34,41**	7601,23	35,61	326,44	44,17	865,45
4	5	**10,23**	2572,75	21,32	310,98	14,32	799,97
	6	**21,62**	5085,65	34,42	343,99	25,87	958,26
	7	**46,95**	10111,26	48,34	440,42	58,66	1078.87

GPU versus OpenMP CPU

30x i7-7 3.6G 8 cores 16 Ths



Trees	D	NoIF		IF		Table	
1	5	**3,62**	510,29	6,43	72,18	5,39	189,39
	6	**6,56**	1257,27	9,21	75,24	7,57	221,13
	7	13,04	2598,33	**10,07**	100,57	15,14	254,60
2	5	**5,61**	1235,78	10,88	109,70	8,23	362,16
	6	**11,55**	2620,45	15,58	148,03	13,44	423,52
	7	24,18	5102,83	**22,07**	218,01	29,67	544,08
4	5	**10,23**	2572,75	21,32	310,98	14,32	799,97
	6	**21,62**	5085,65	34,42	343,99	25,87	958,26
	7	**46,95**	10111,26	48,34	440,42	58,66	1078.87

GPU is 250x 15x 40x

Conclusions

- No IF up to Depth 6
- Better many shadow trees than a few deep tree
- GPU 30x faster than CPU
- Future Work: FPGAs and compare with Real Dataset and Tree Depth+Number

WSCAD 2023

XXIV Simpósio em Sistemas Computacionais de Alto Desempenho

17 a 20 de outubro, 2023 — Porto Alegre, Brasil

Avaliação de estilos de código para árvore de decisão em GPU com Microbenchmarks

ricardo@ufv.br

Questions ?

FAPEMIG APQ-01577-22

CNPq

CAPES