

Estrutura de dados da HiGtree



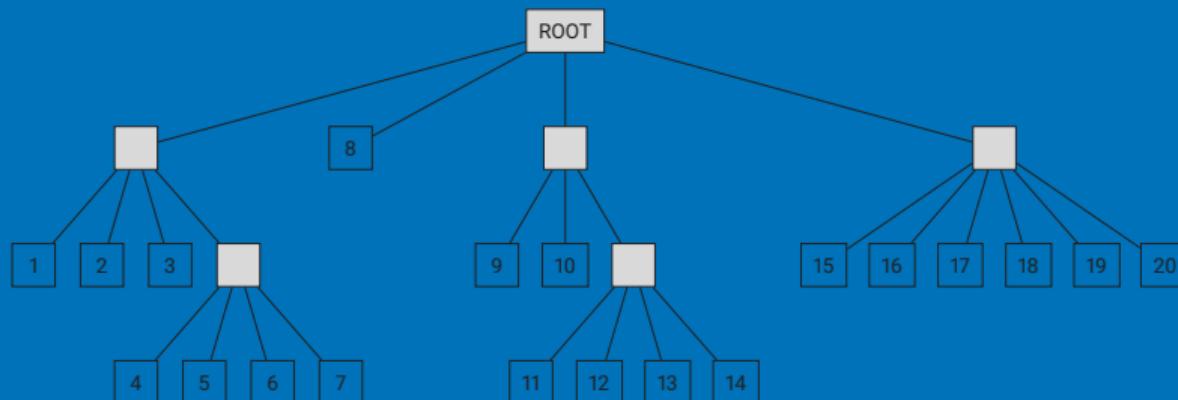
A HiG-Tree (*Hierarchical Grid Tree*) é uma estrutura de dados em árvore usada em malhas adaptativas, que permite refinamento, dividindo o espaço recursivamente, economizando memória e tempo de processamento.

Nos casos 2D e 3D, chamamos de *quadtree* e *octree*, respectivamente.



Exemplo de HiG-Tree

8	16	18	20
	15		
	17		19
2	5	7	
	11	12	13
	14		
4	6		
1	3		
			10
			9



Exemplo de malha 2D e sua quadtree associada.
Fonte: ??

lib/layouts/vertical_bg.pdf



Células da HiG-Tree

```
typedef struct hig_cell {  
    Point lowpoint;  
    Point highpoint;  
    int numcells[DIM];  
    struct hig_cell * parent;  
    int posinparent;  
    struct hig_cell **children;  
    uniqueid ids[NUMIDS];  
} hig_cell;
```

$(x_{1min}, x_{2min}, \dots, x_{mmin})$

$(x_{1max}, x_{2max}, \dots, x_{mmax})$

(n_1, n_2, \dots, n_m)

Célula-pai

Posição em relação ao pai

Células-filho

Id de cada célula/faceta



Facetas da HiG-Tree

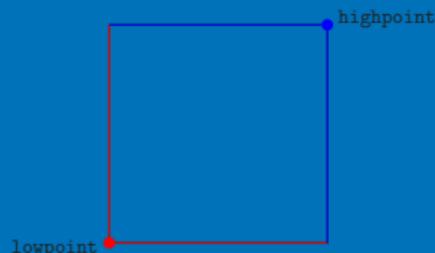
```
typedef struct hig_facet {  
    hig_cell *c;  
    int dim;  
    int dir;  
} hig_facet;
```

Célula associada

Cada faceta tem dimensão $m-1$

Direção {0,1}

dir=0: Facetas geradas pelas arestas que contém o lowpoint.
dir=1: Facetas geradas pelas arestas que contém o highpoint.



Facetas de uma célula 2D.



Visualização das malhas

Escrevemos a malha em formato AMR (*Adaptive Mesh Refinement*) como entrada, por exemplo:

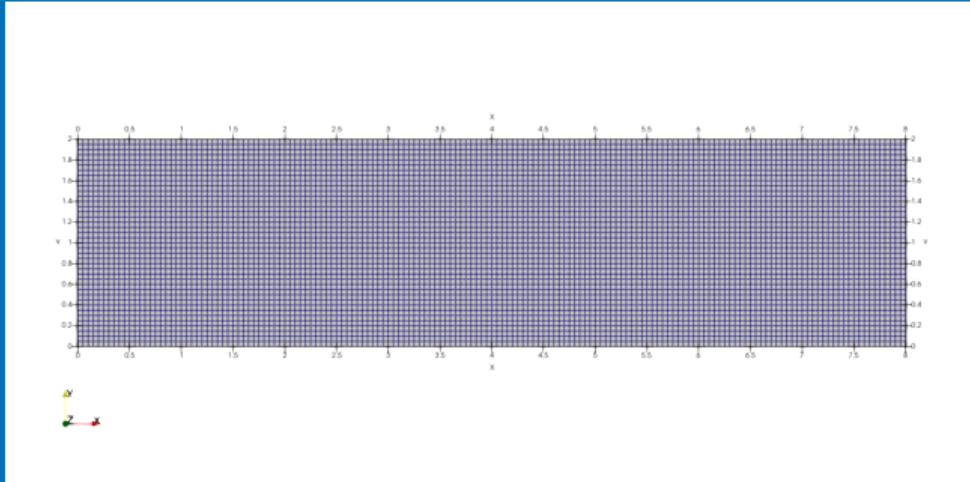
```
0.0 8.0 -1.0 1.0      lowpoint highpoint = xmin xmax ymin ymax
2                         numlevels
0.05 0.05 1            (level 1) delta numpatches = delta_x delta_y numpatches
1 1 160 40              initcell patchsize
0.025 0.025 4          (level 2) delta numpatches = delta_x delta_y numpatches
1 1 320 16              initcell patchsize
1 65 320 16             initcell patchsize
1 17 16 48              initcell patchsize
305 17 16 48            initcell patchsize
```

então, geramos um arquivo de saída em formato VTK (*Visualization Toolkit*) com `higtree/src/higtree-io.c` e visualizamos com o ParaView.



Visualização das malhas

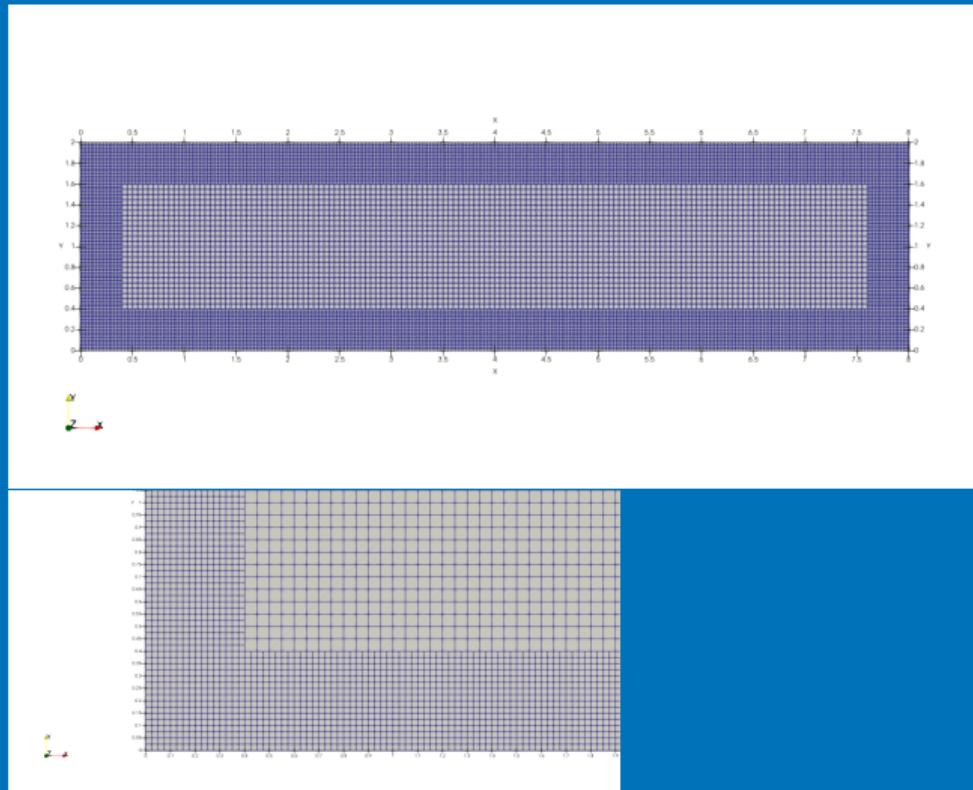
```
0.0 8.0 -1.0 1.0  
1  
0.05 0.05 1  
1 1 160 40
```



Exemplo de malha 2D com 1 nível de refinamento.



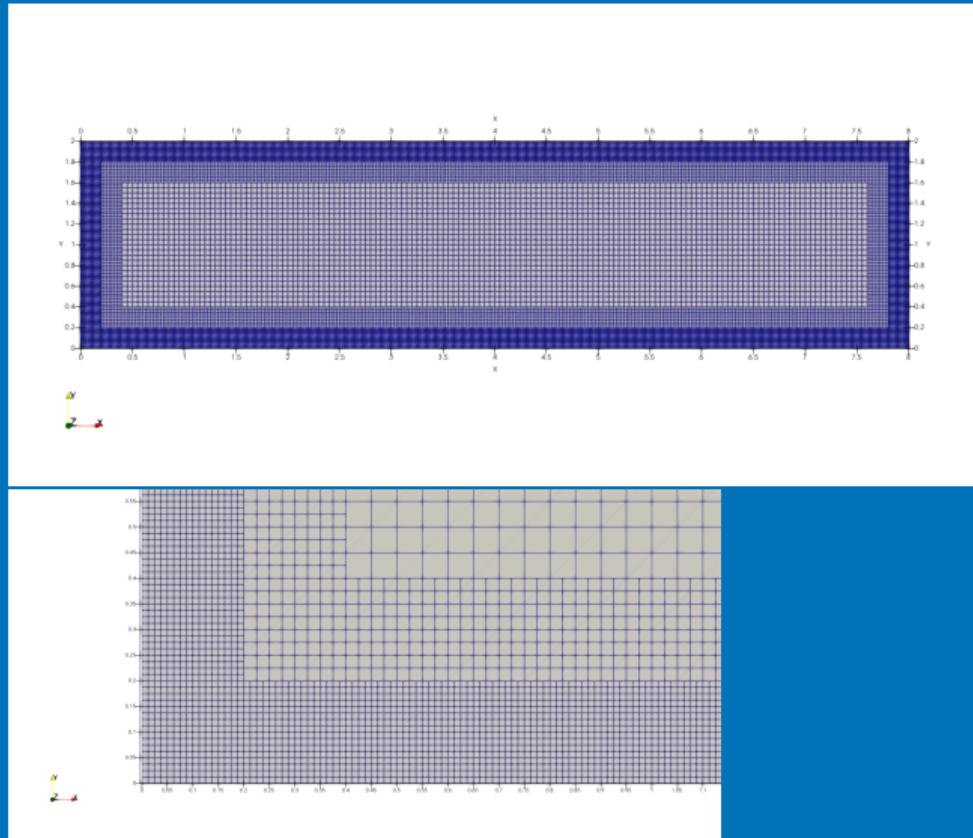
```
0.0 8.0 -1.0 1.0  
2  
0.05 0.05 1  
1 1 160 40  
0.025 0.025 4  
1 1 320 16  
1 65 320 16  
1 17 16 48  
305 17 16 48
```



Exemplo de malha 2D com 2 níveis de refinamento.



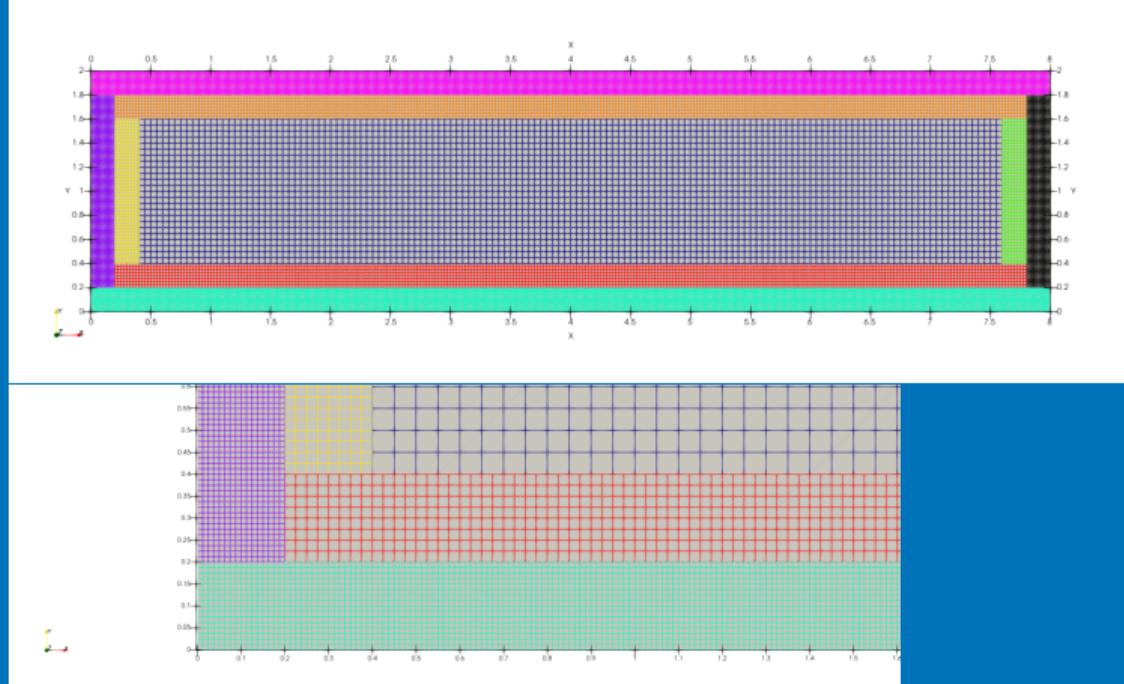
```
0.0 8.0 -1.0 1.0  
3  
0.05 0.05 1  
1 1 160 40  
0.025 0.025 4  
1 1 320 16  
1 65 320 16  
1 17 16 48  
305 17 16 48  
0.0125 0.0125 4  
1 1 640 16  
1 145 640 16  
1 17 16 128  
625 17 16 128
```



Exemplo de malha 2D com 3 níveis de refinamento.



```
0.0 8.0 -1.0 1.0
3
0.05 0.05 1
1 1 160 40
0.025 0.025 4
1 1 320 16
1 65 320 16
1 17 16 48
305 17 16 48
0.0125 0.0125 4
1 1 640 16
1 145 640 16
1 17 16 128
625 17 16 128
```



Exemplo de malha 2D com 3 níveis de refinamento.