

 **SYNACKTIV**



Spatial Frinet

SSTIC 2026 - Théo EMERIAU

- **Combiner l'analyse statique & dynamique**
 - Traces d'exécution → IDA
- **gaasedelen/tenet (2021)**
- **synacktiv/frinet (SSTIC 2024)**
 - Traceur basé sur Frida
 - Vue « Callgraph »
 - Recherche dans la mémoire

Trace d'exécution

```
$ python3 trace.py spawn [BIN_PATH] [MODULE_NAME] [FUNC_ADDR] -a [ARGS]
```

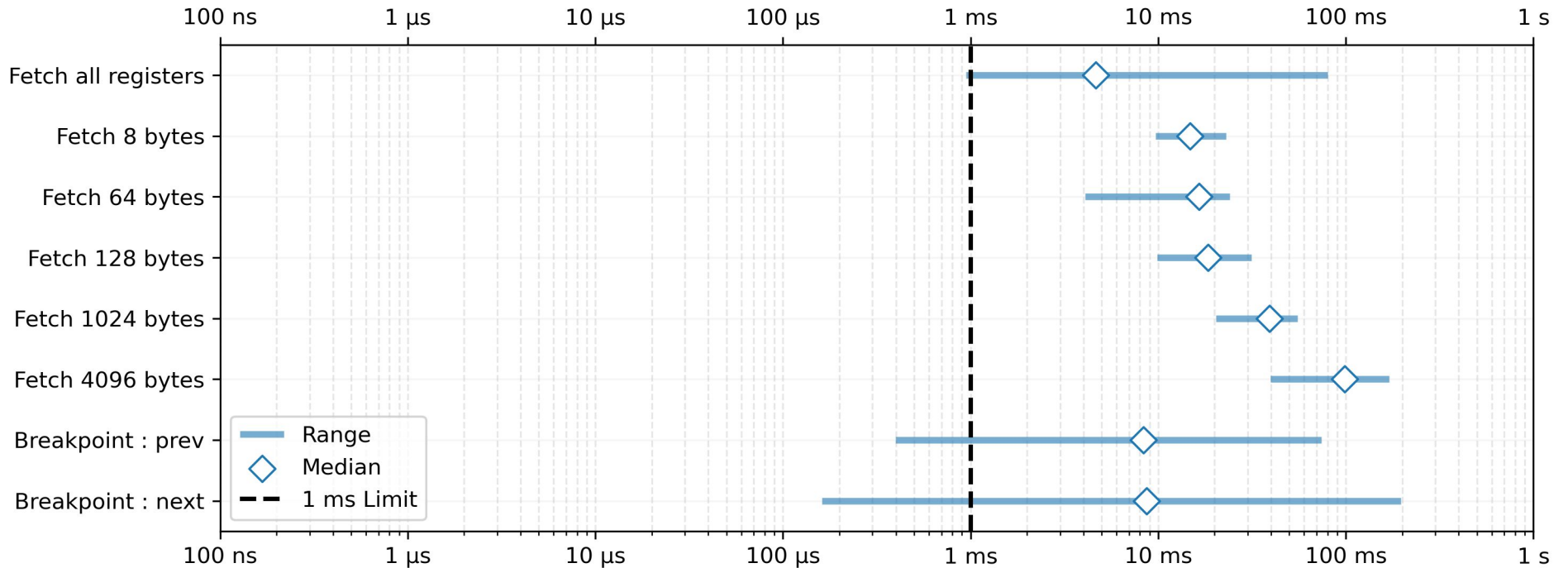
```
rip=0x7fdf256e2494, rdi=0x7fdf24291df0, mr=0x7ffc857caeb8:f01d2924df7f0000  
rip=0x7fdf256e249b, mw=0x7ffc857caea8:00000000  
rip=0x7fdf256e24a5, mw=0x7ffc857caeb8:0000000000000000  
rip=0x7fdf256e24b0  
rip=0x7fdf256e24b3  
rip=0x7fdf256e24b5  
rip=0x7fdf256e24ba, rdi=0x7fdf2428a890, mr=0x7ffc857caeb0:90a82824df7f0000  
rip=0x7fdf256e24c1, mw=0x7ffc857caeb0:0000000000000000  
rip=0x7fdf256e24cc  
...
```


- **Identifier le code lié à un log particulier**
 - Rechercher la construction du message dans la mémoire
 - Puis remonter le callgraph
- **Suivre l'évolution d'une allocation**
 - Breakpoint en R/W sur la zone mémoire
 - Ctrl-scroll
- **Time Travel Debugger**

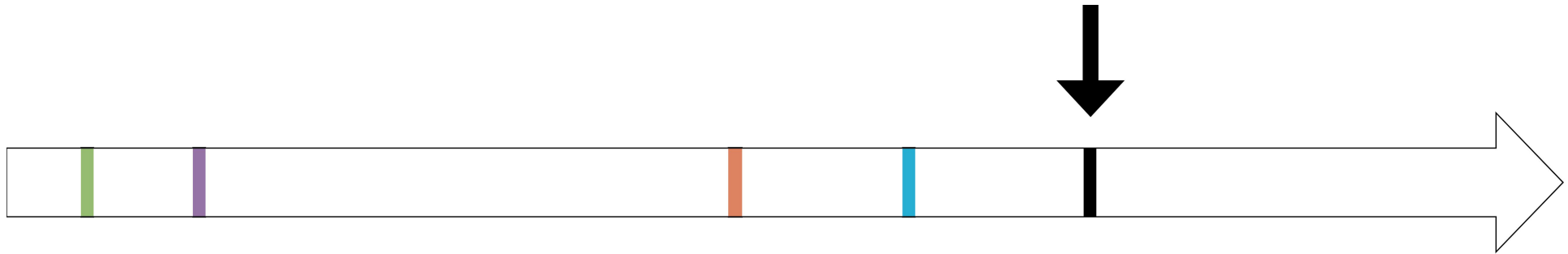
- **Un obstacle persiste ...**
- **200 millions d'instructions (10 Go)**
 - Chargement initial : 23 minutes / 20 Go de RAM
 - Fluidité de l'interface dégradée
- **Besoin communs des équipes Synacktiv**
 - Entre 50 et 500 millions d'instructions

Benchmark : Frinet Legacy

Frinet legacy : 200 millions d'instructions (10 Go)

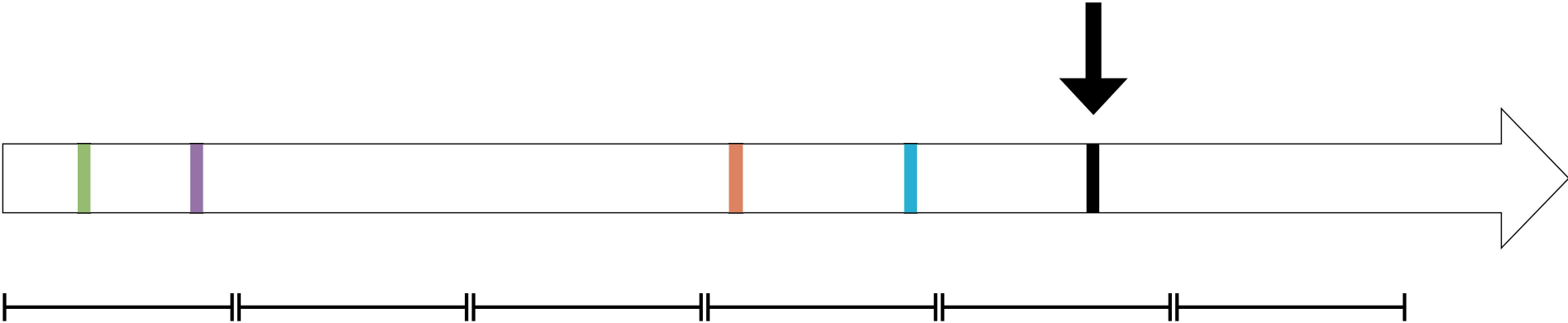


Informations dispersées

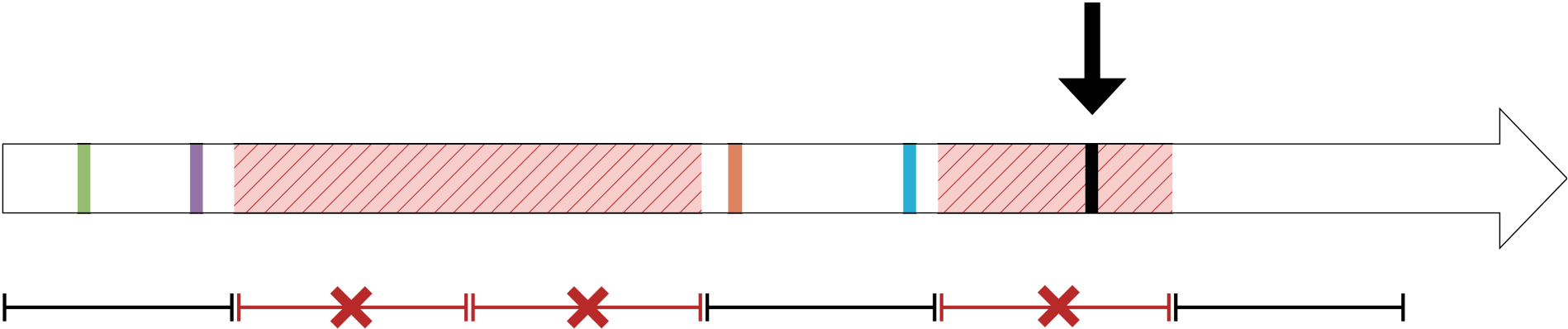


00007FFCB2C30B50	E8 03 00 00 00 00 00 00	00 00 00 97 9D FC C9 80 16 A8
00007FFCB2C30B60	80 0B C3 B2 FC 7F 00 00	60 71 82 BF 51 56 00 00`q..QV..
00007FFCB2C30B70	B9 B9 82 BF 51 56 00 00	68 51 44 AB 51 56 00 00QV..hQD.QV..
00007FFCB2C30B80	00 00 00 00 00 00 00 00	FF FF FF FF 00 00 00 00
00007FFCB2C30B90	00 00 00 00 00 00 00 00	58 08 46 AB 51 56 00 00X.F.QV..
00007FFCB2C30BA0	68 51 44 AB 51 56 00 00	67 0C C3 B2 FC 7F 00 00	hQD.QV..g.....
00007FFCB2C30BB0	F0 CA 80 BF 51 56 00 00	B0 0D C3 B2 FC 7F 00 00QV.....
00007FFCB2C30BC0	02 00 00 00 01 00 00 00	58 0F C3 00 00 7F 00 00X.....
00007FFCB2C30BD0	F0 CA 80 BF 51 56 00 00	B0 0D C3 B2 FC 7F 00 00QV.....

Pré-calculs par segments



Pré-calculs par segments



■ 3 Composants

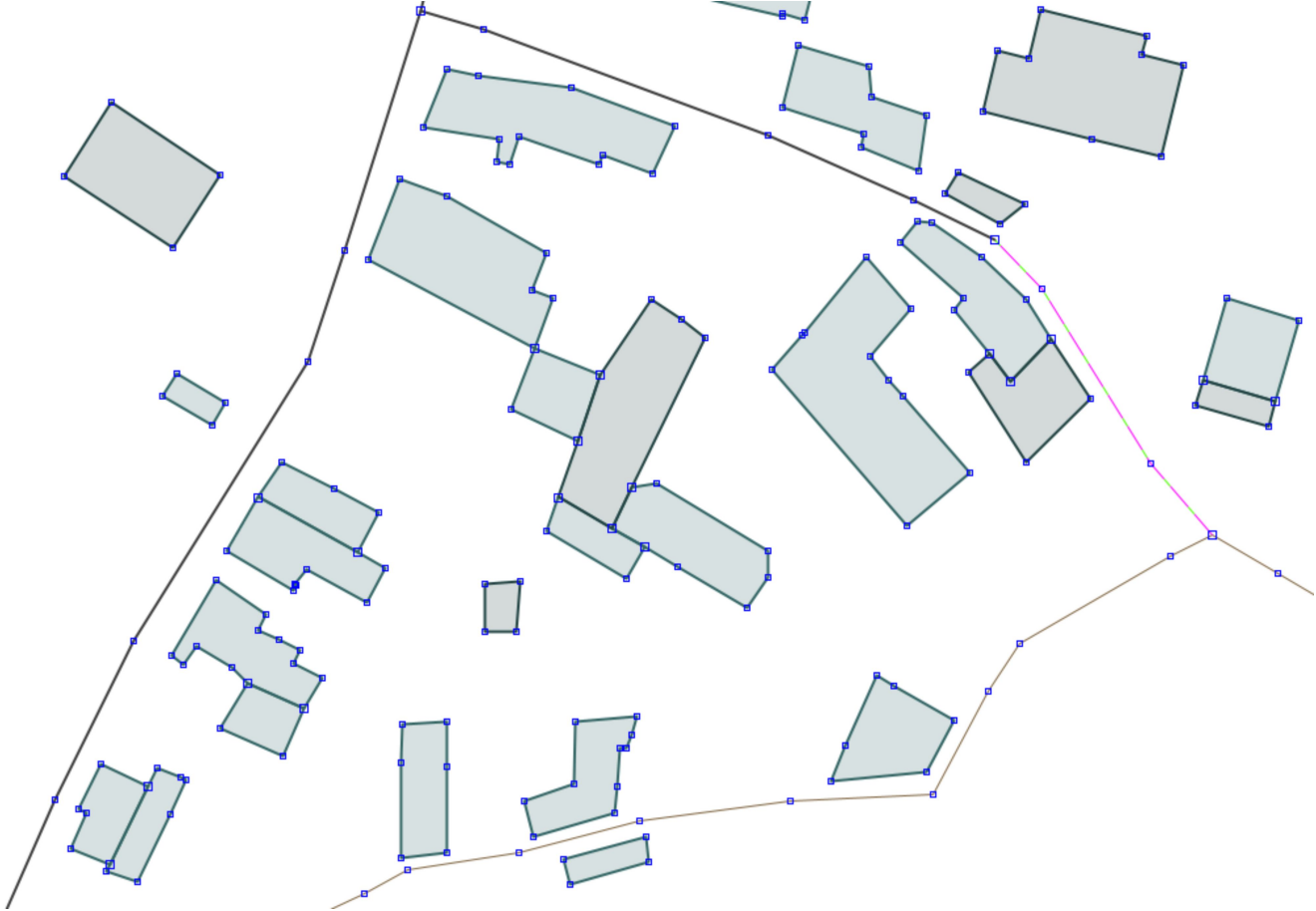
- Indexeur (Rust) : CLI externe
- Backend (Rust) : algorithmes de recherches
- Frontend (Python) : plugin IDA

■ Objectif cible : minimiser la latence

- Idéalement ~20ms par frame / ~1ms par requête

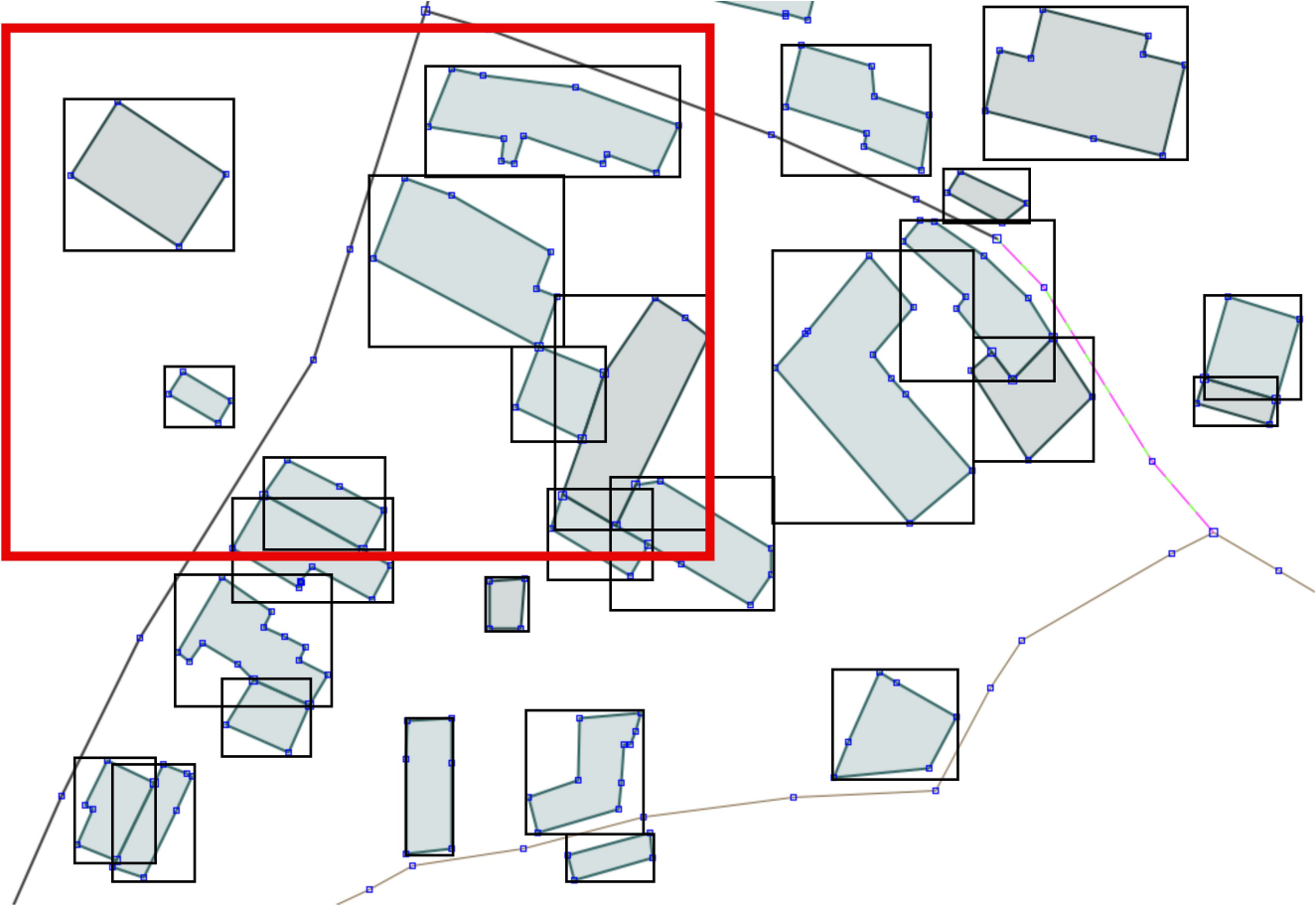
- **Multi-dimensionnel**
 - Mémoire : Temps + Adresse
 - Registre : Temps + Valeur
- **Construction sur un PC portable lambda**
 - Temps & RAM
- **Stockage**
 - Fichier auto-porteur
 - Chargement instantané (zero-copy)

OpenStreetMap

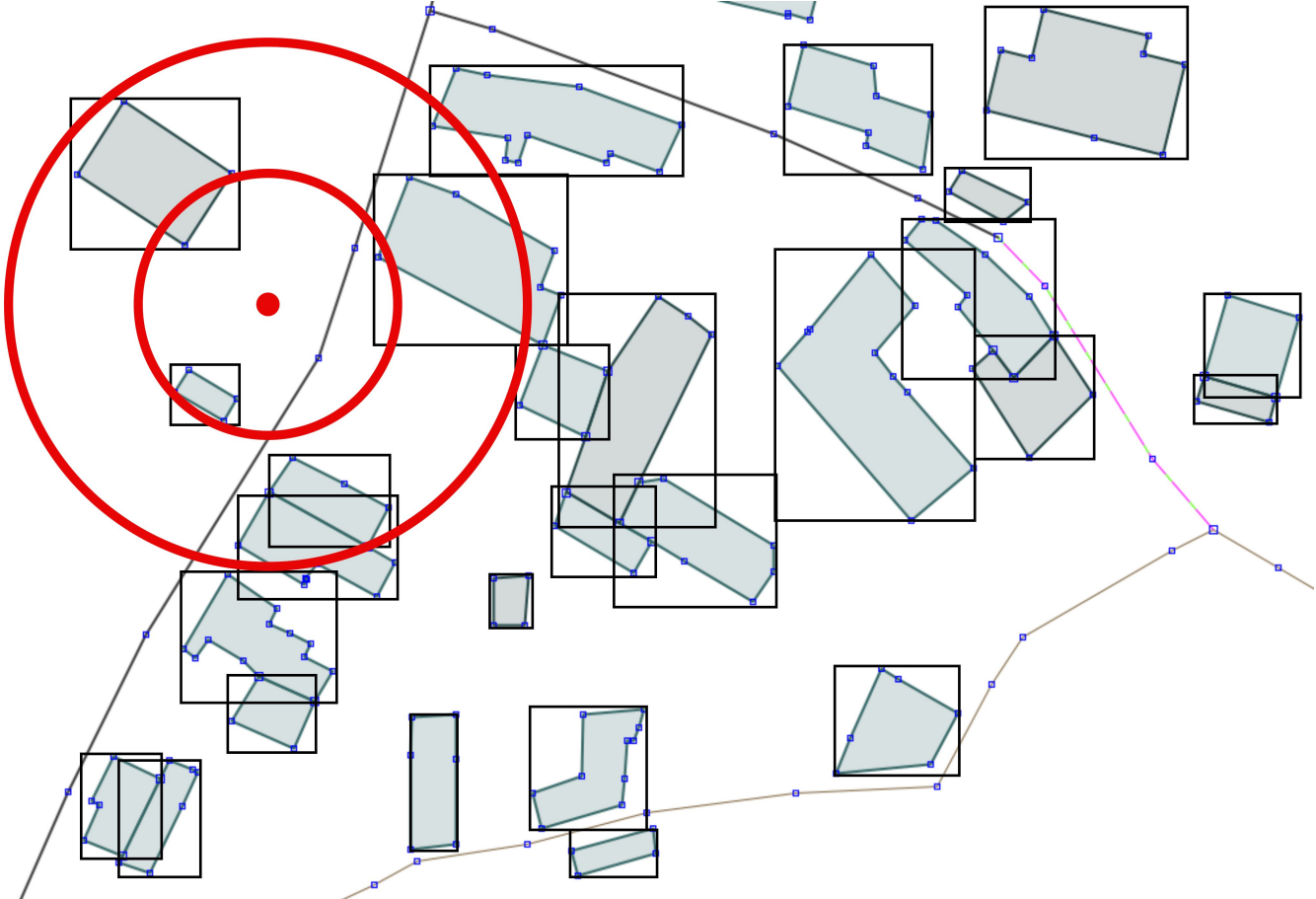




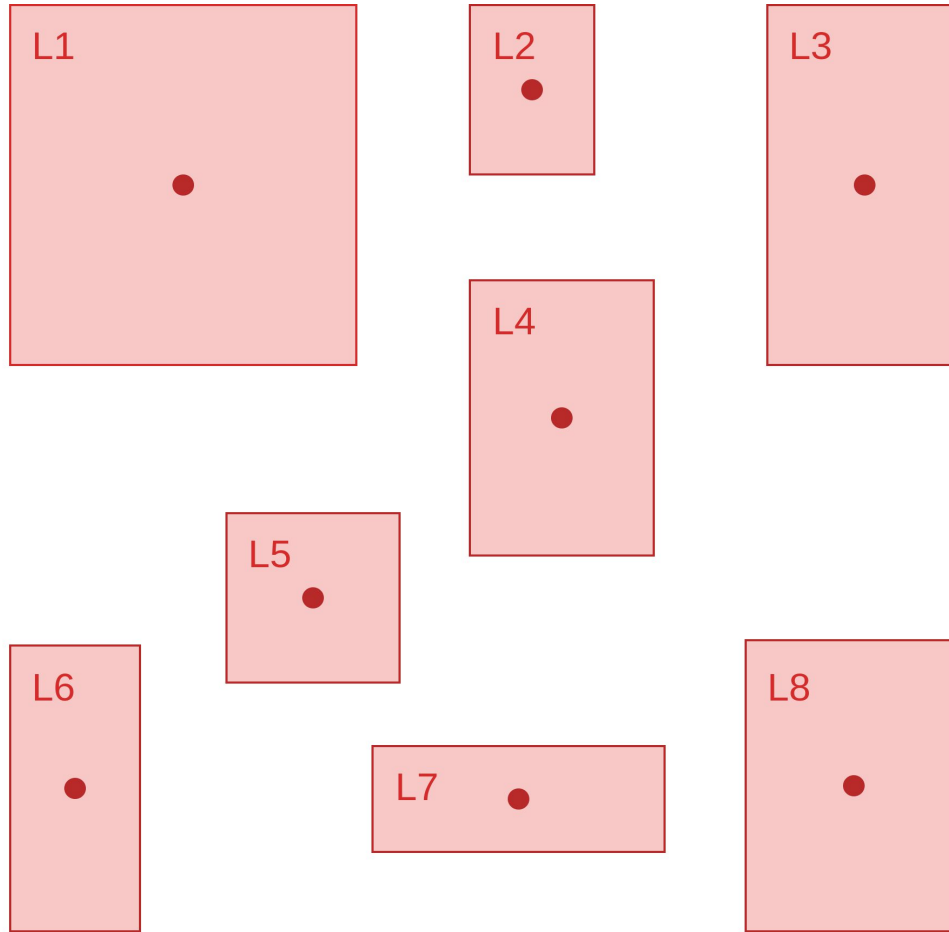
OpenStreetMap : intersection

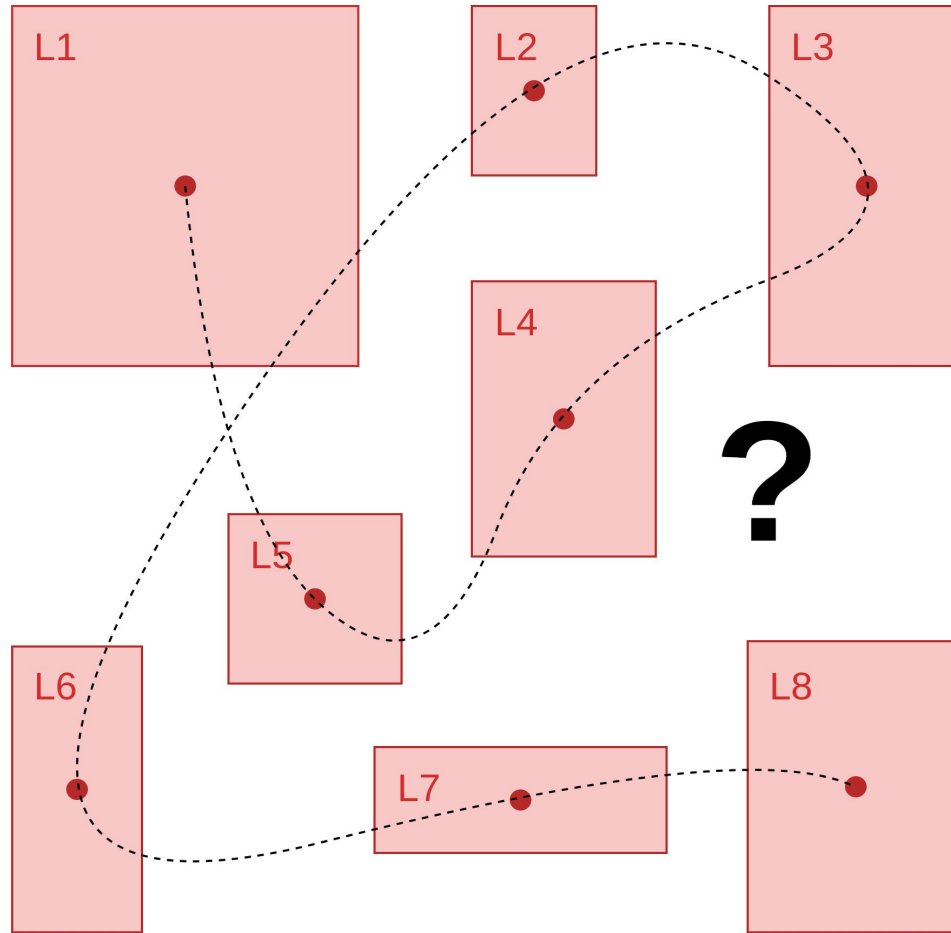


OpenStreetMap : nearest neighbour

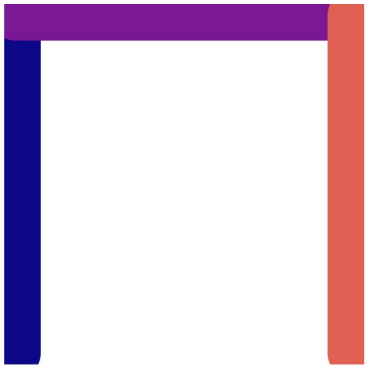


Hilbert Packed R-Tree

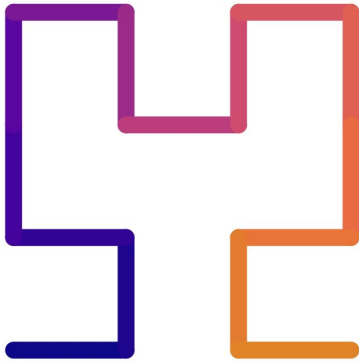




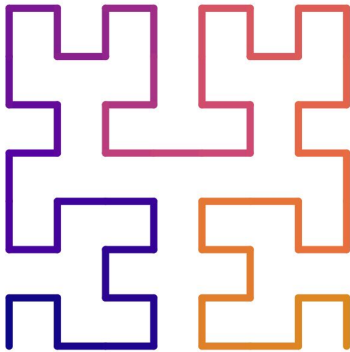
Courbe de Hilbert



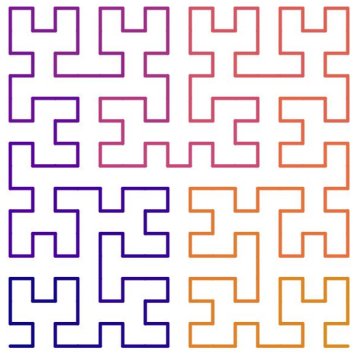
2x2



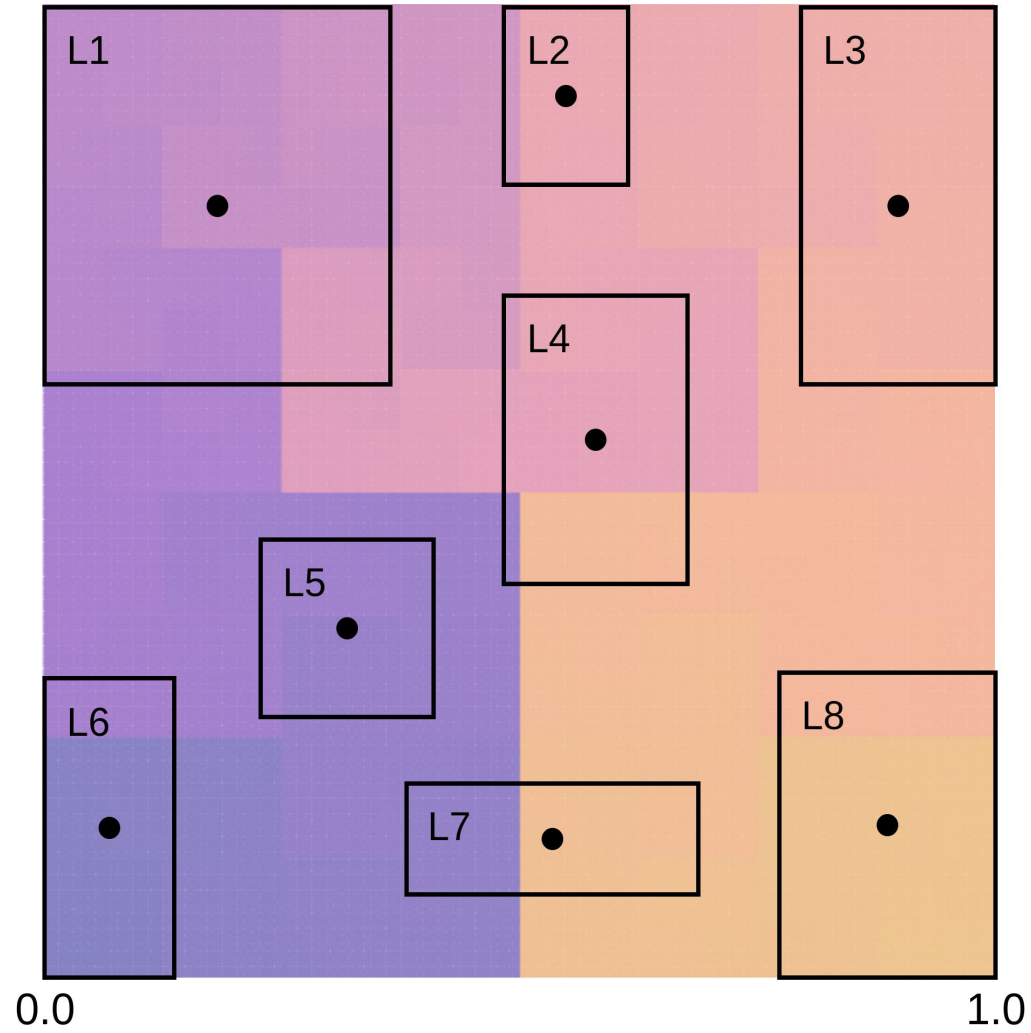
4x4

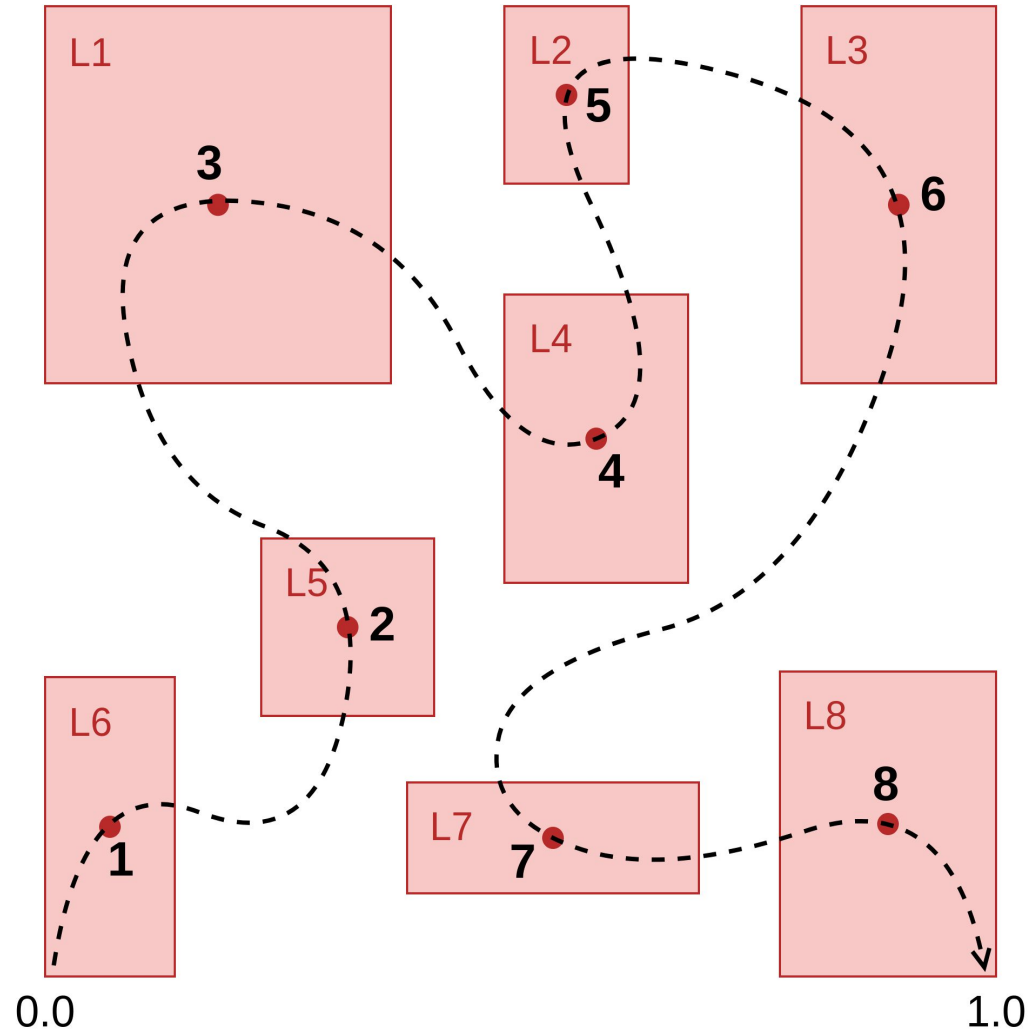


8x8

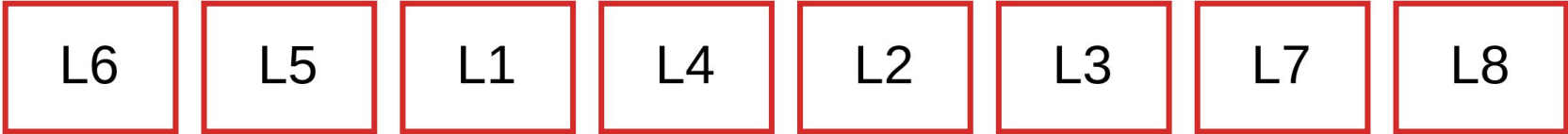


16x16



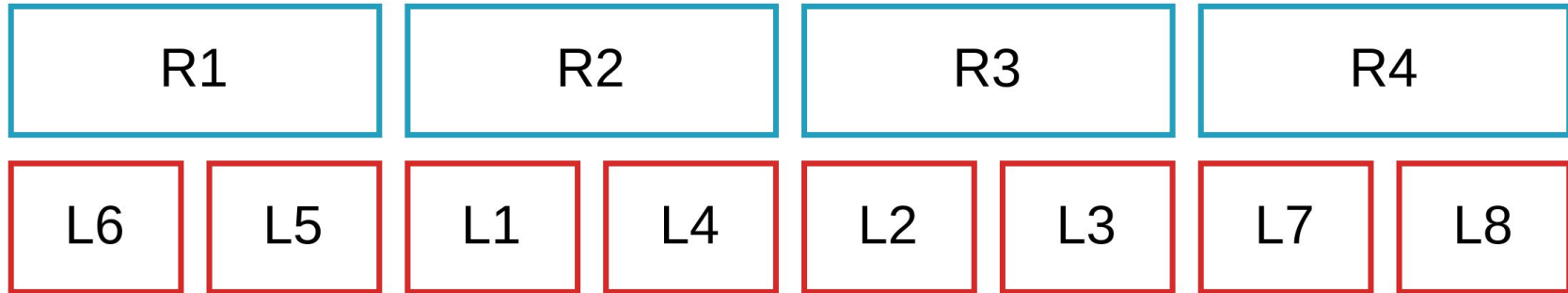


Hilbert Packed RTree



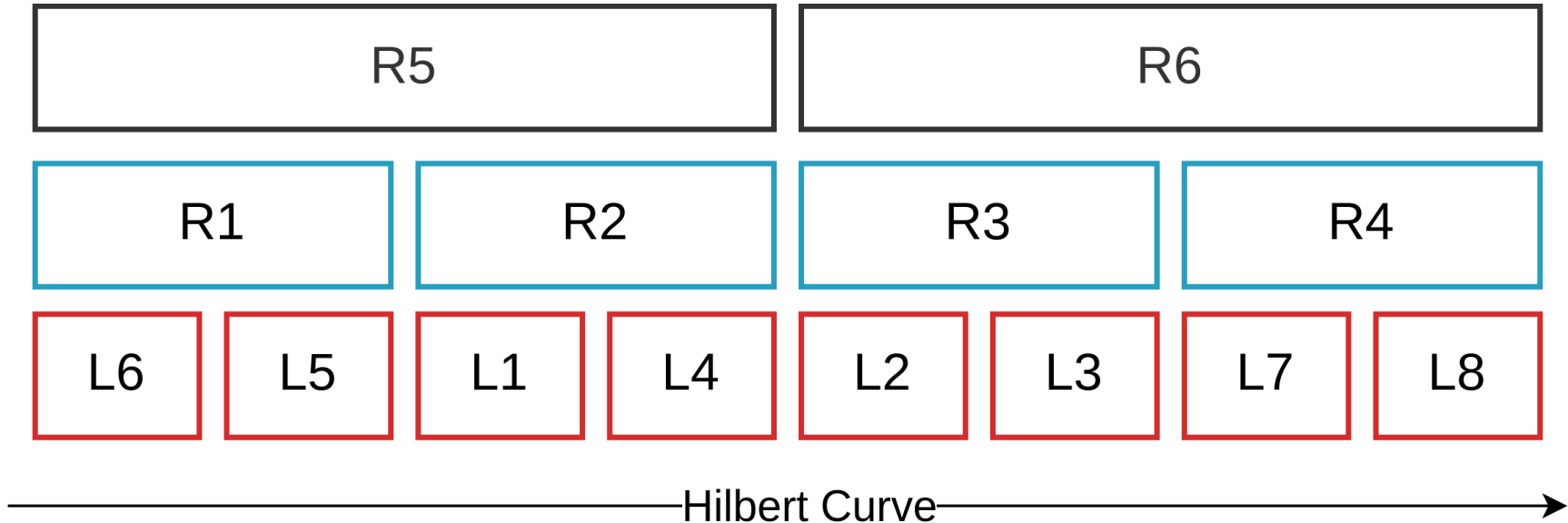
————— Hilbert Curve —————>

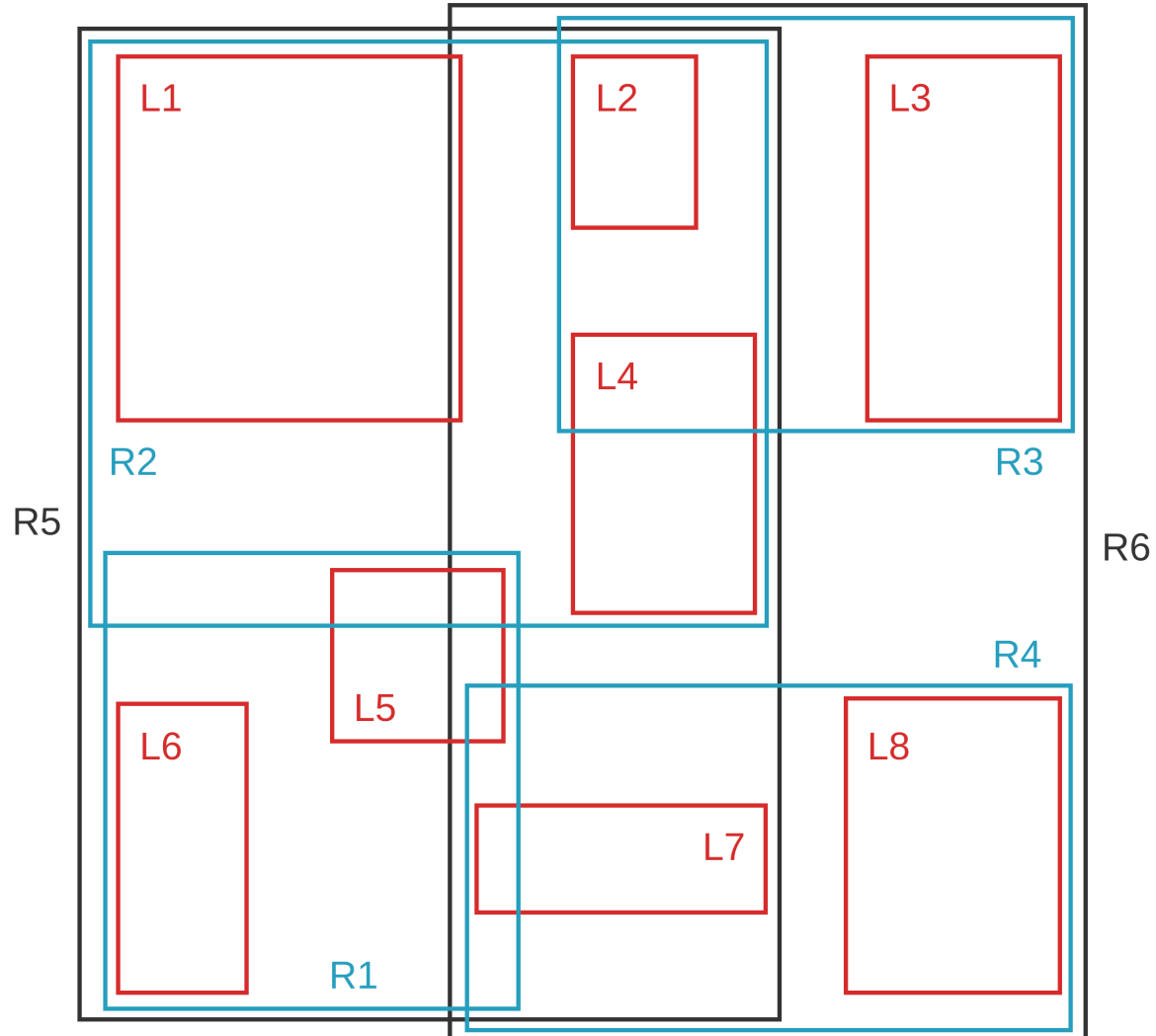
Hilbert Packed RTree

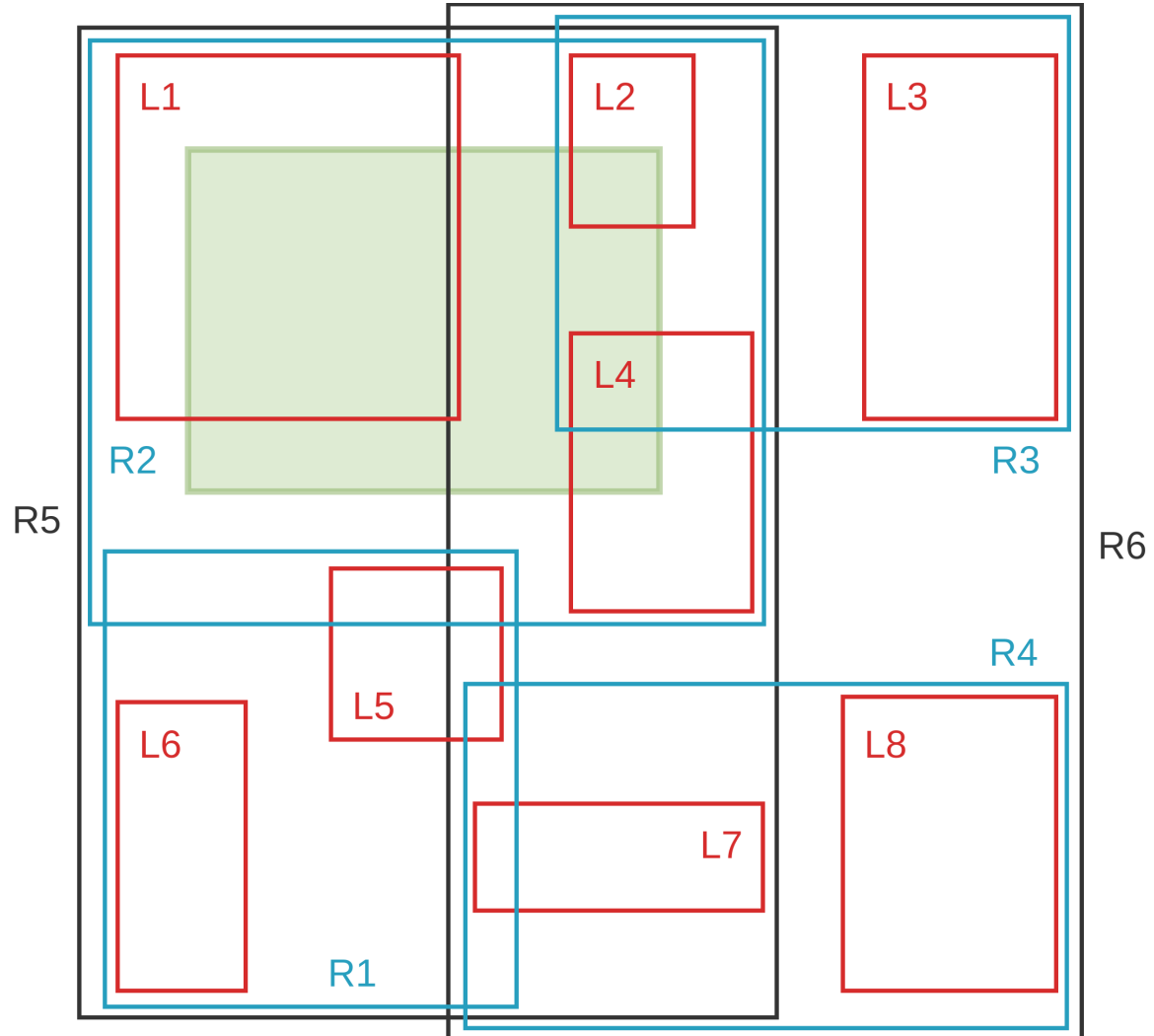


————— Hilbert Curve —————>

Hilbert Packed RTree

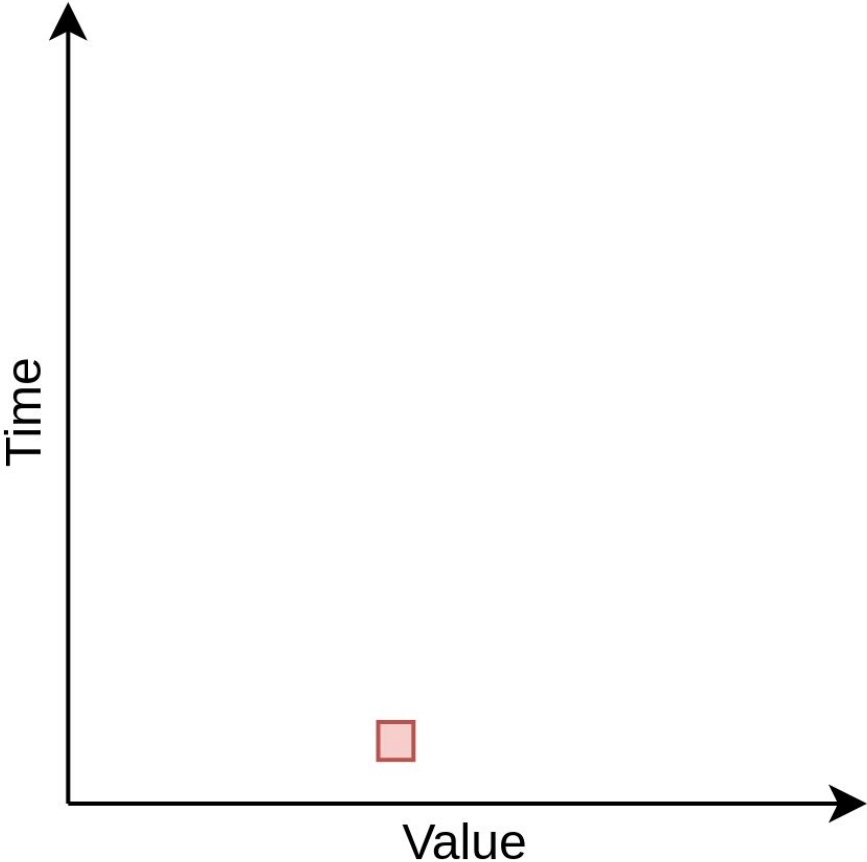




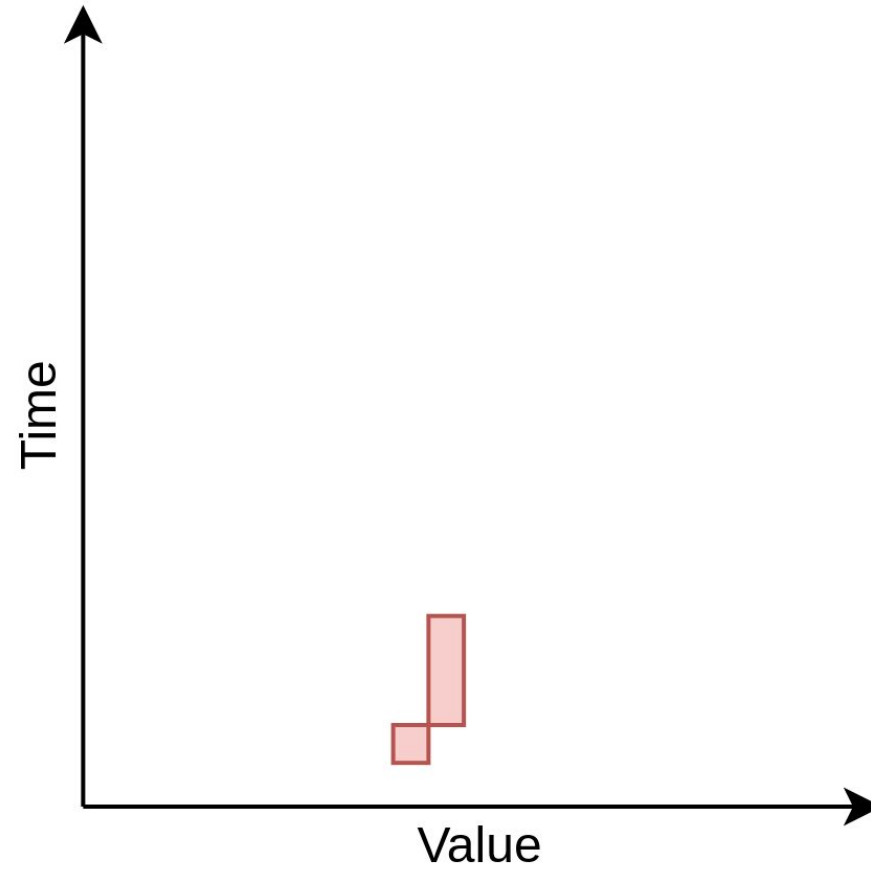


Trace d'exécution → Rectangles

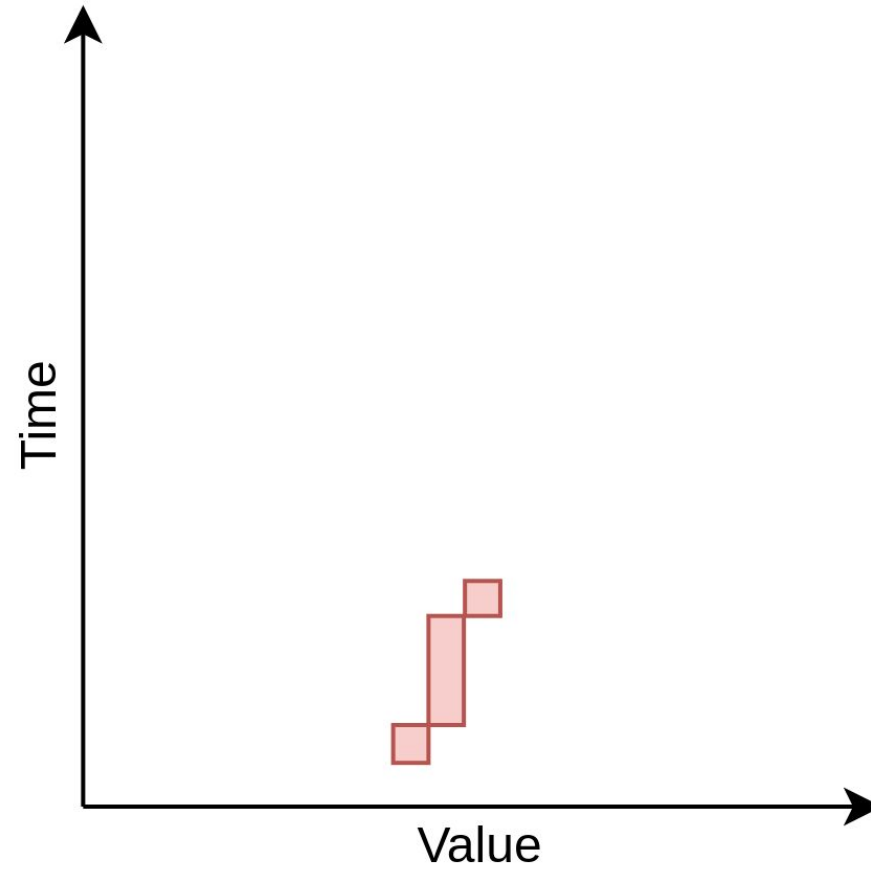
Registres : RAX



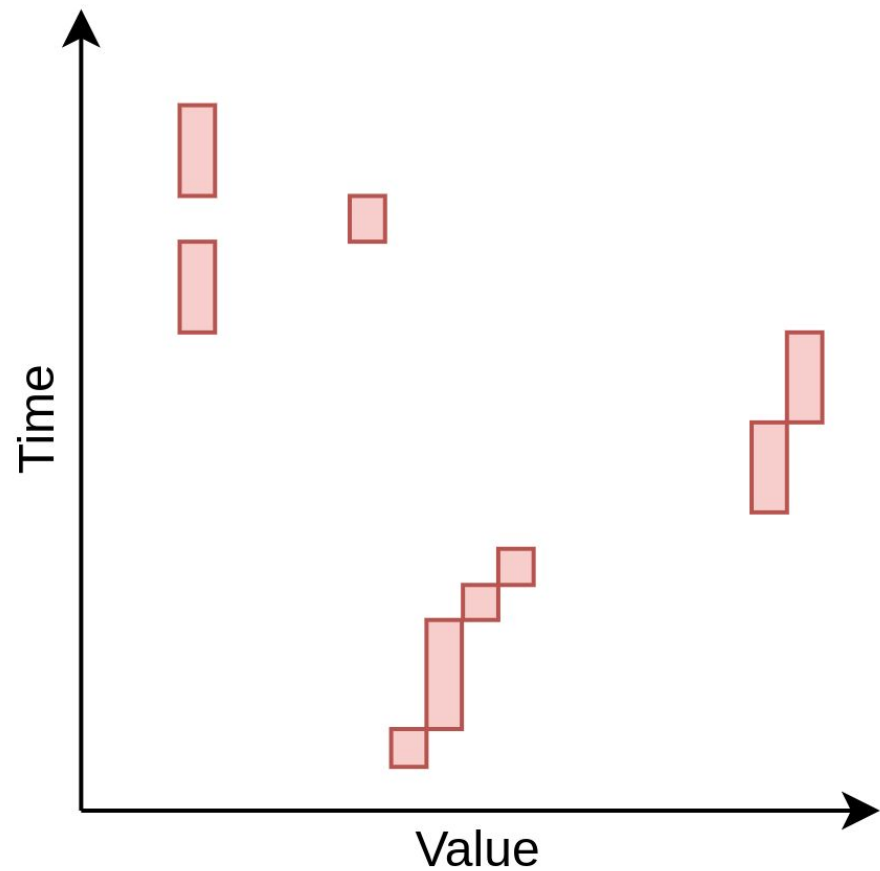
Registres : RAX



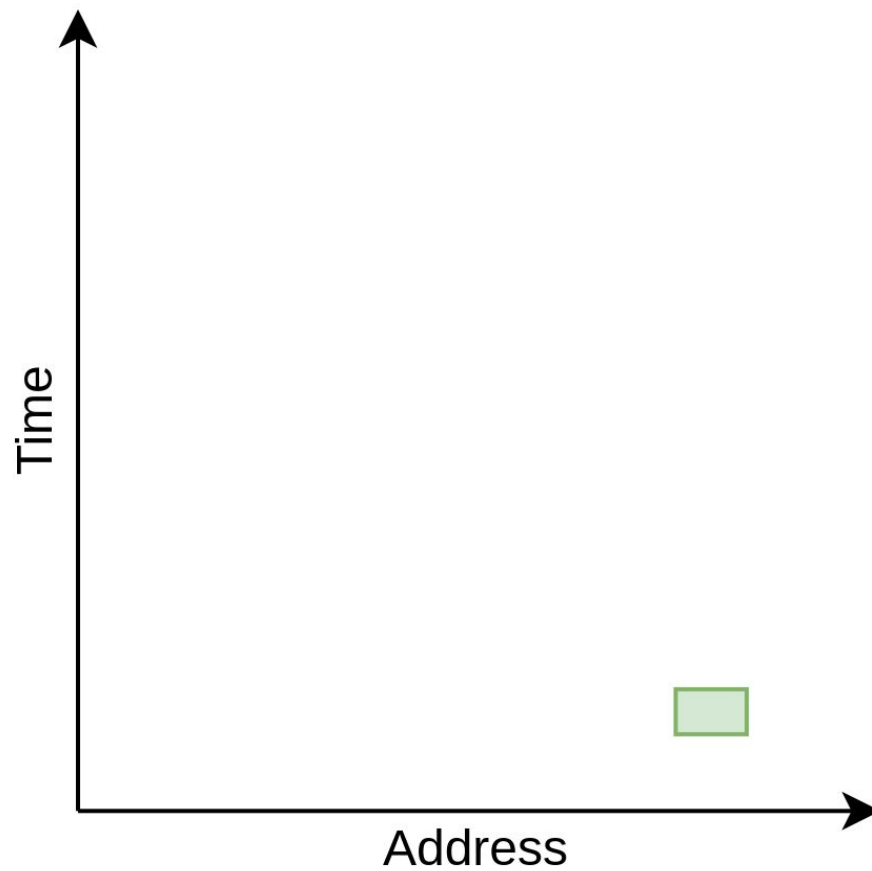
Registres : RAX



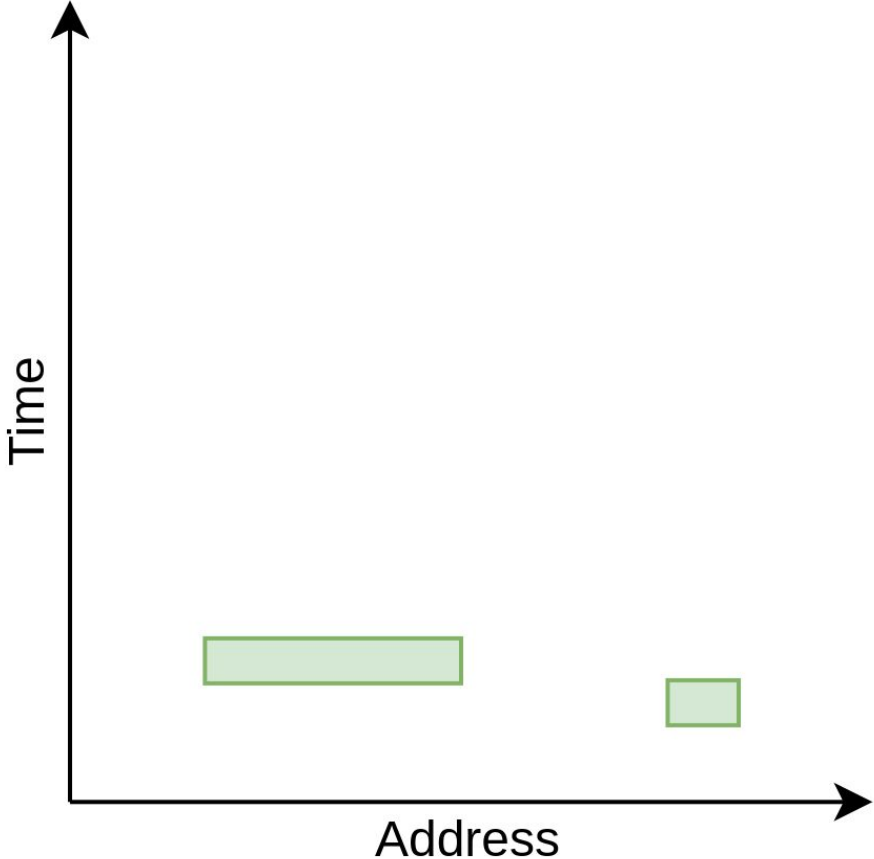
Registres : RAX



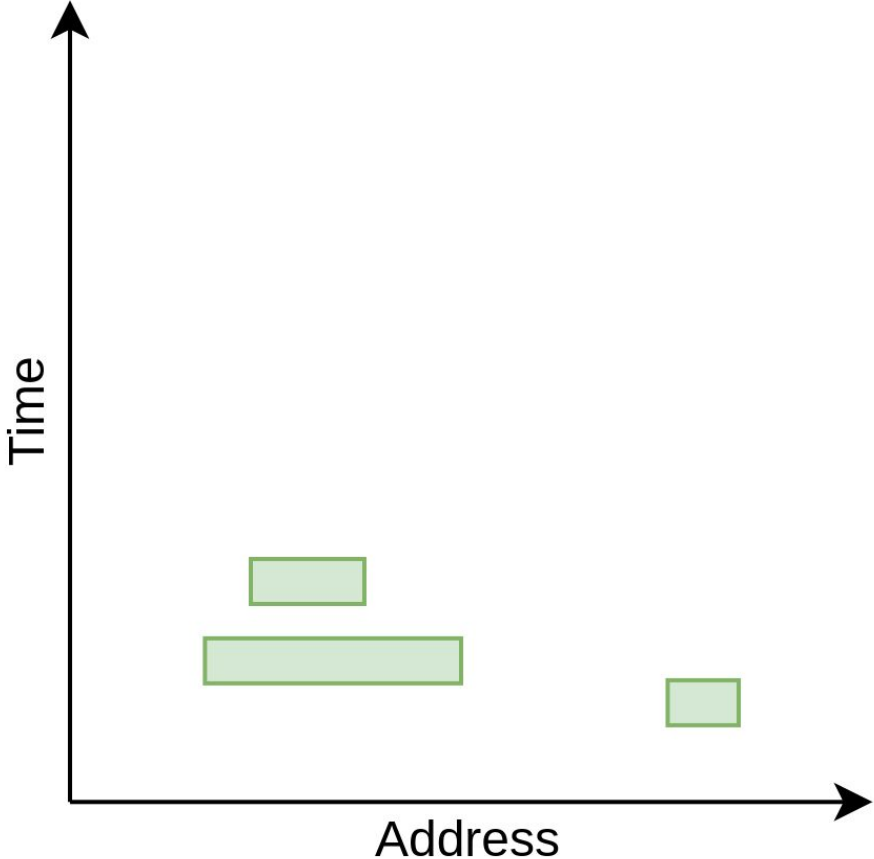
Lectures mémoire



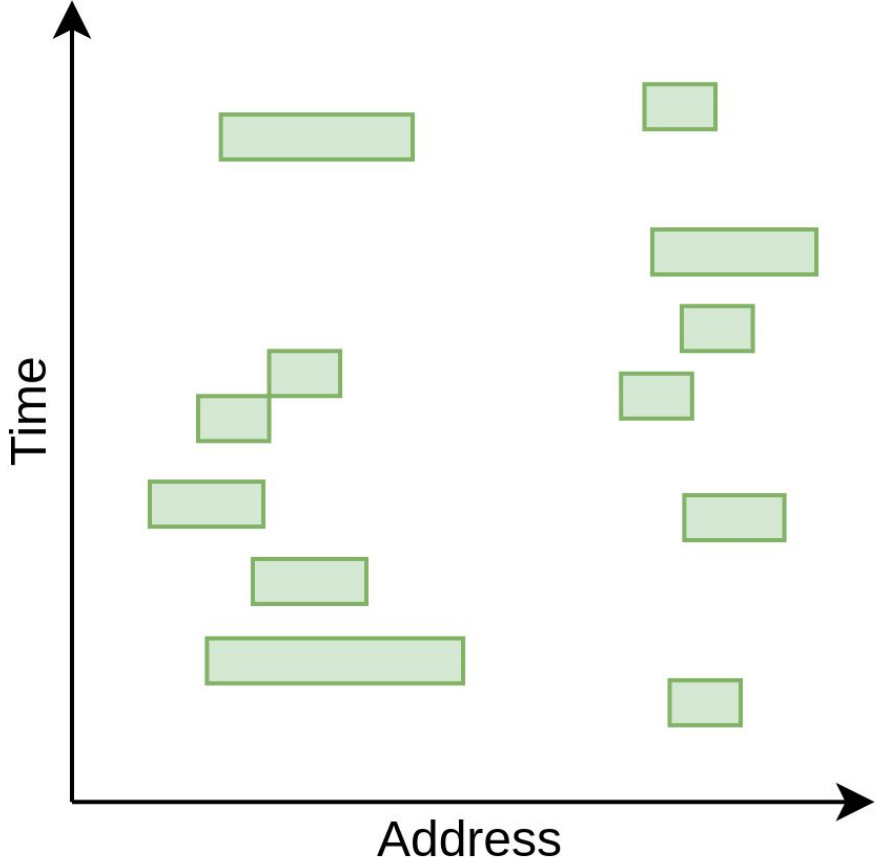
Lectures mémoire



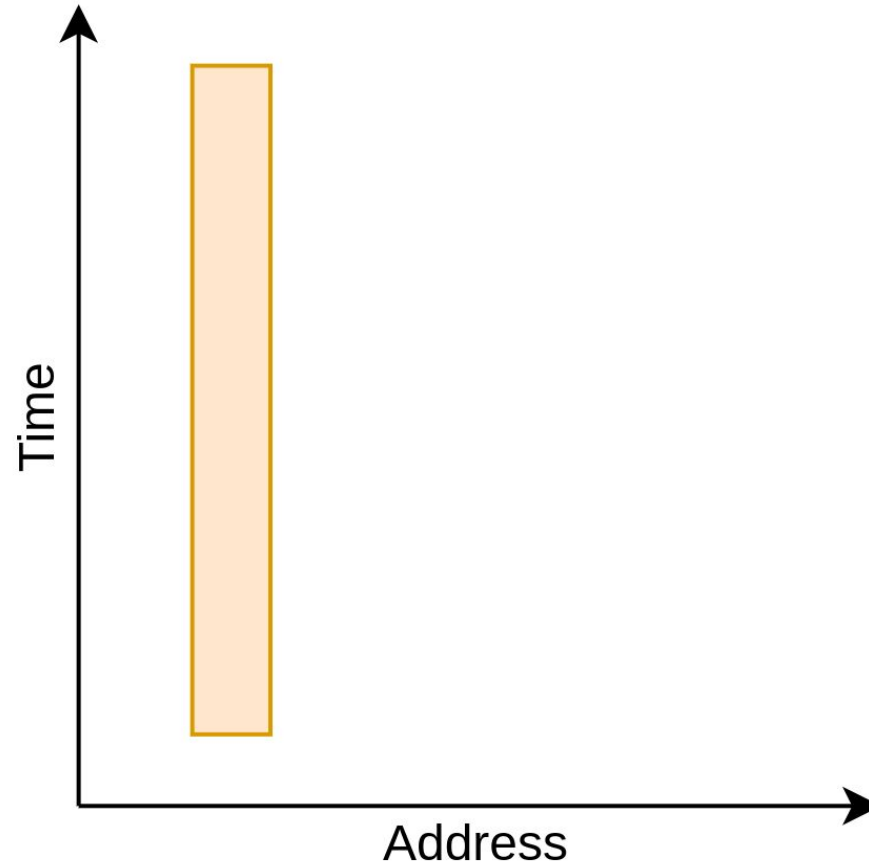
Lectures mémoire



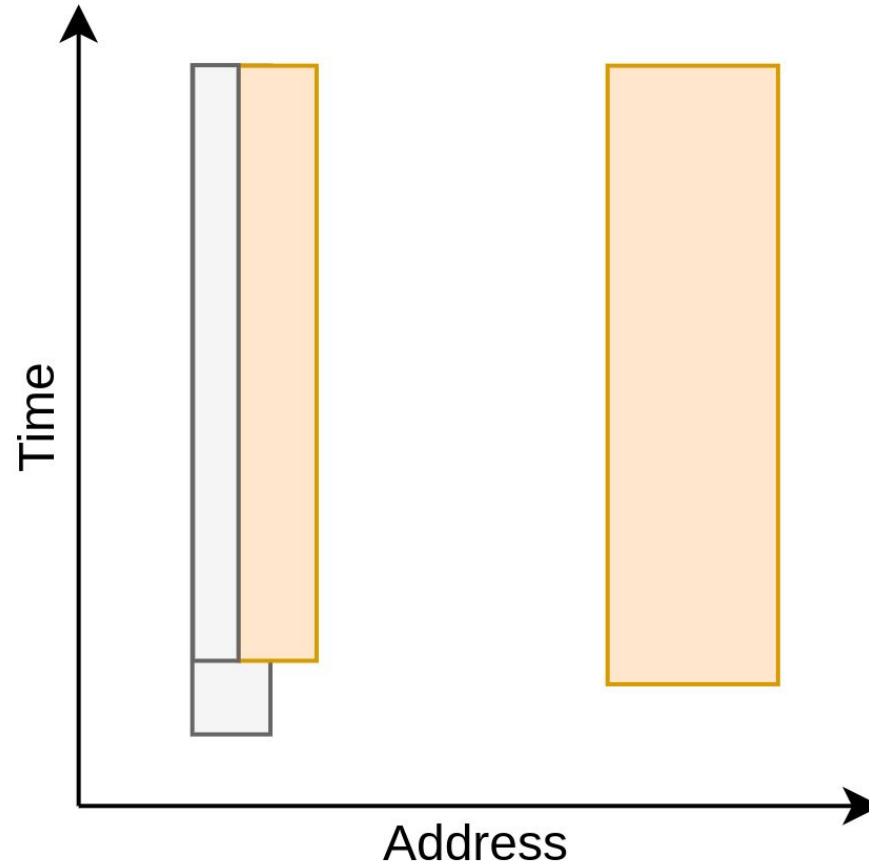
Lectures mémoire



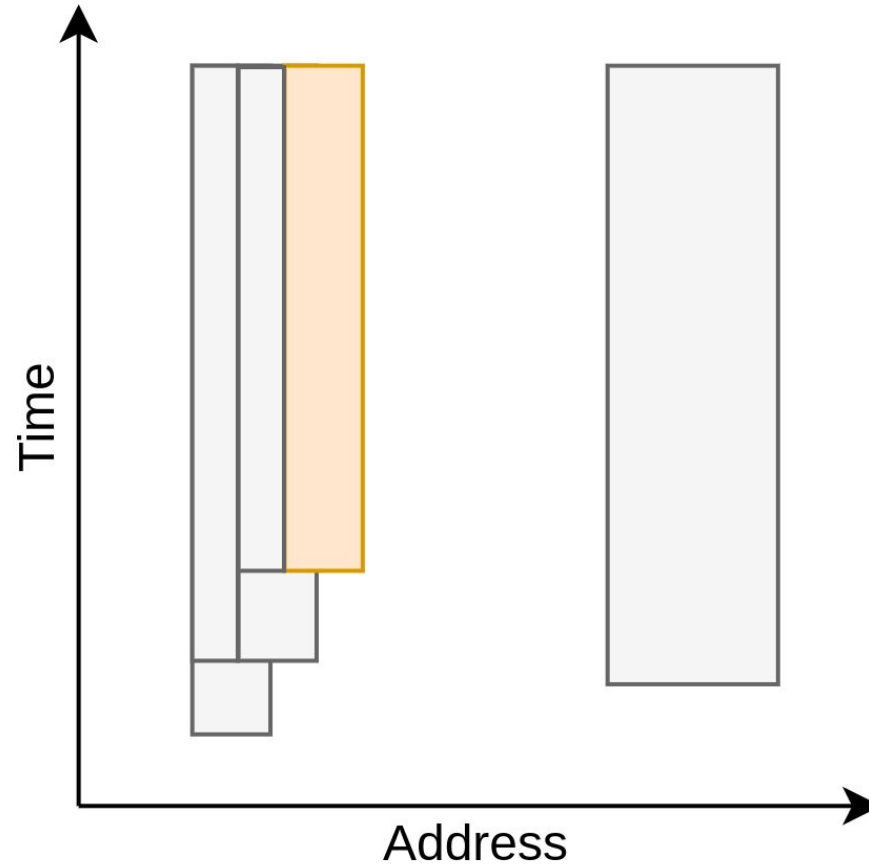
Écritures mémoire



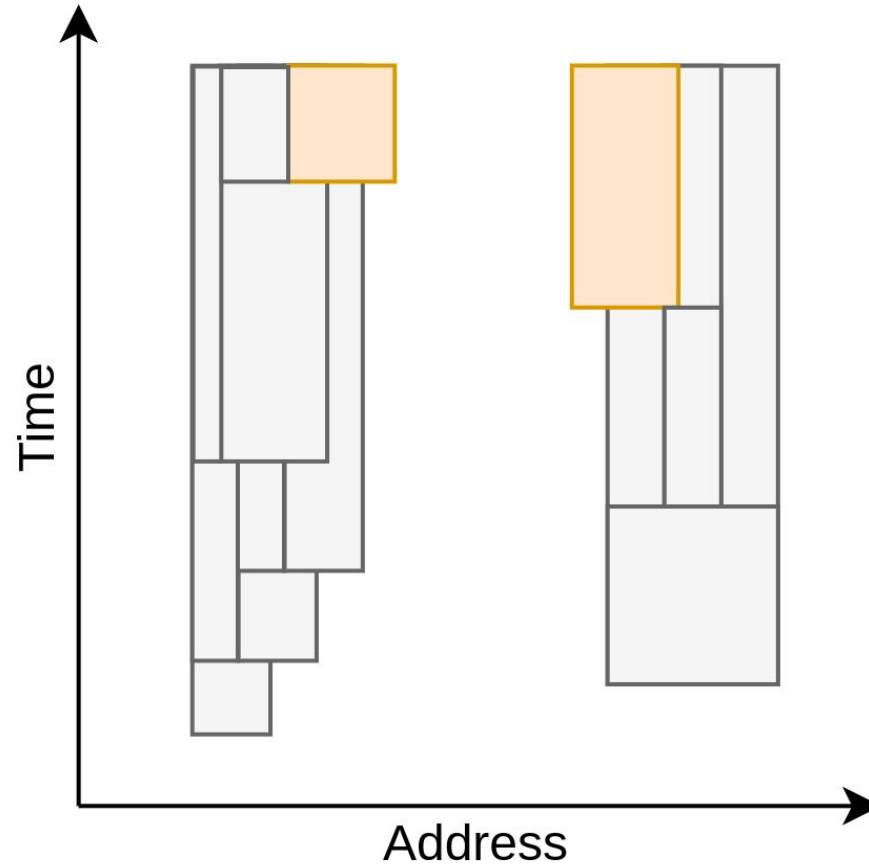
Écritures mémoire



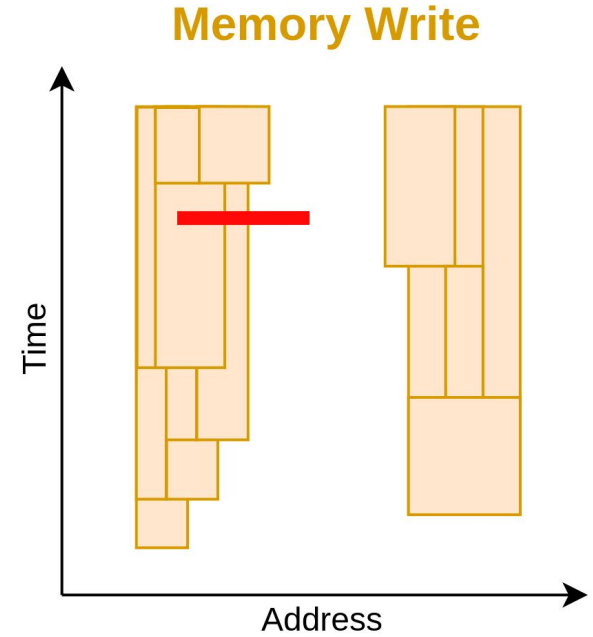
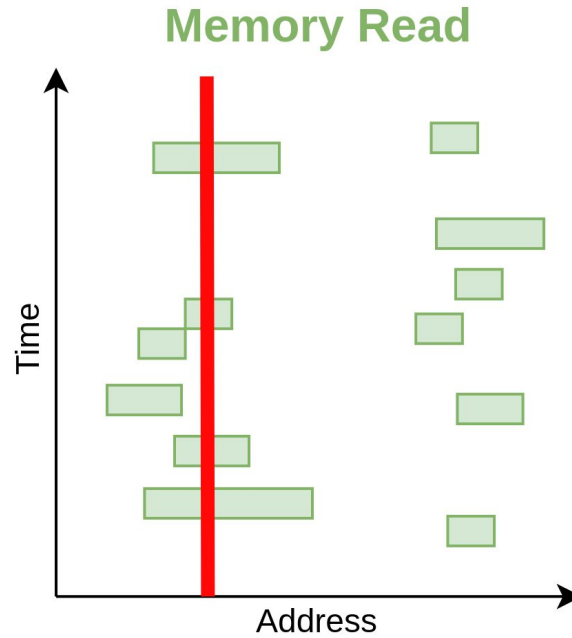
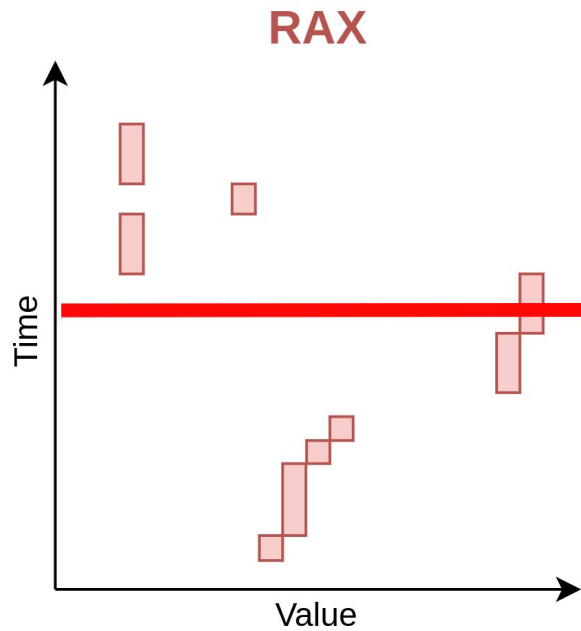
Écritures mémoire



Écritures mémoire



Requêtes → Rectangles



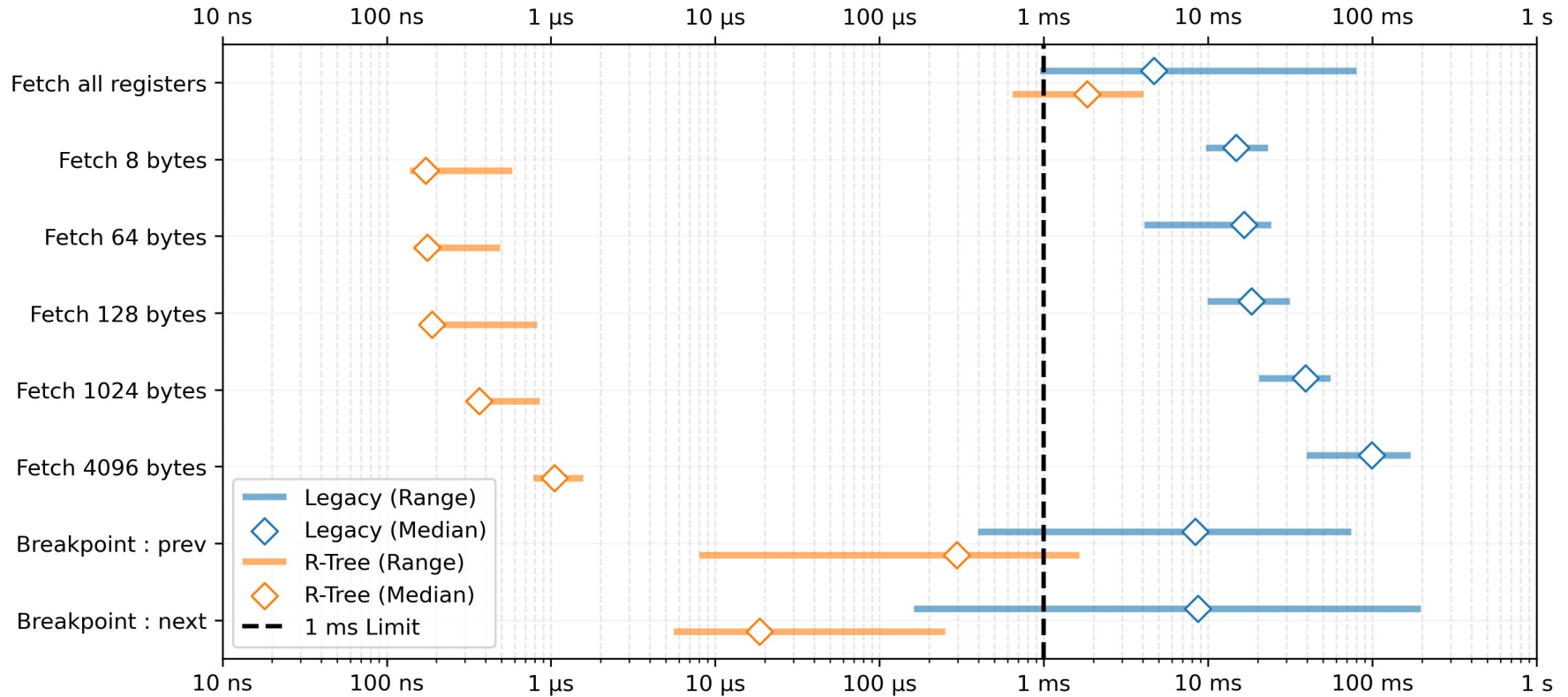
```
$ frinet index --format tenet little_wget.tenet little_wget.db
  Scout [00:01:02] ##### 10.08 GiB/10.08 GiB
Partitions [00:01:01] ##### 10.08 GiB/10.08 GiB
  Fill [00:01:24] ##### 10.08 GiB/10.08 GiB
[INFO frinet] Total time : 3 minutes
[INFO frinet] Maximum memory usage : 2.89 GiB
```

■ Parsing & Indexing

- 23 min → 3 min
- 20 Go → 3 Go

Benchmark : comparaison

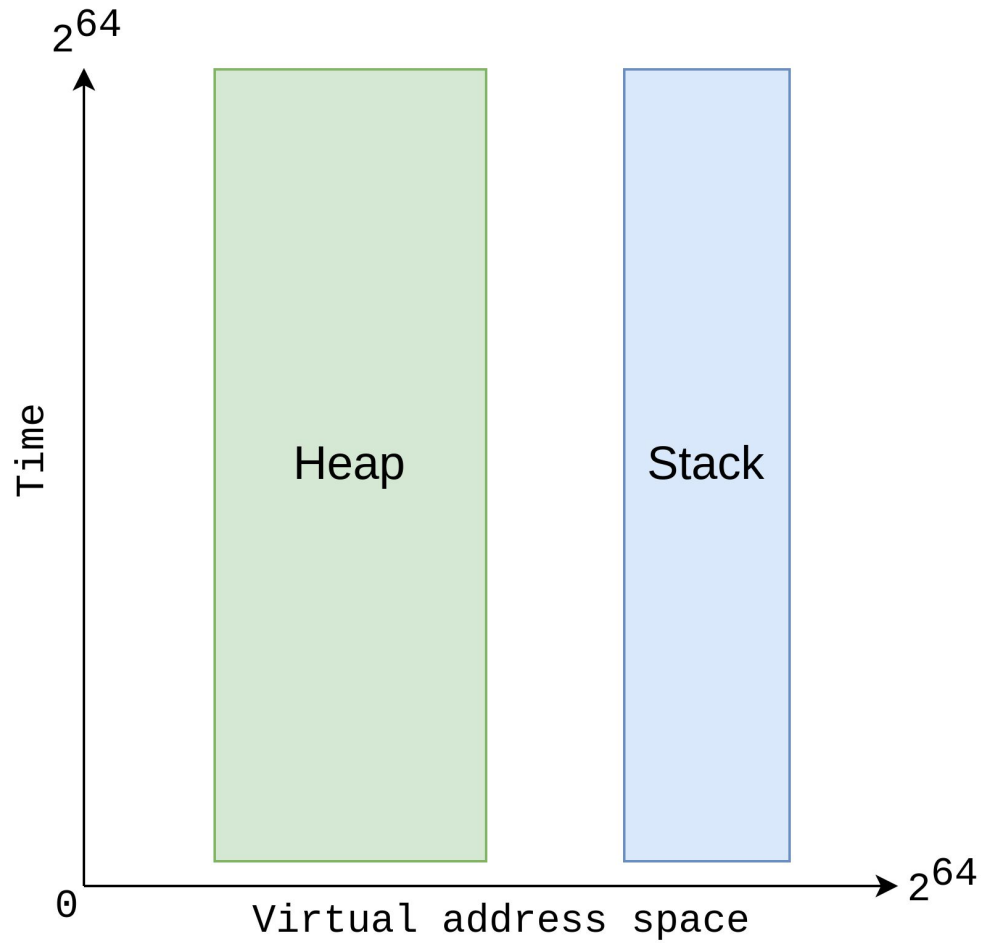
Frinet legacy vs spatial : 200 millions d'instructions (10 Go)



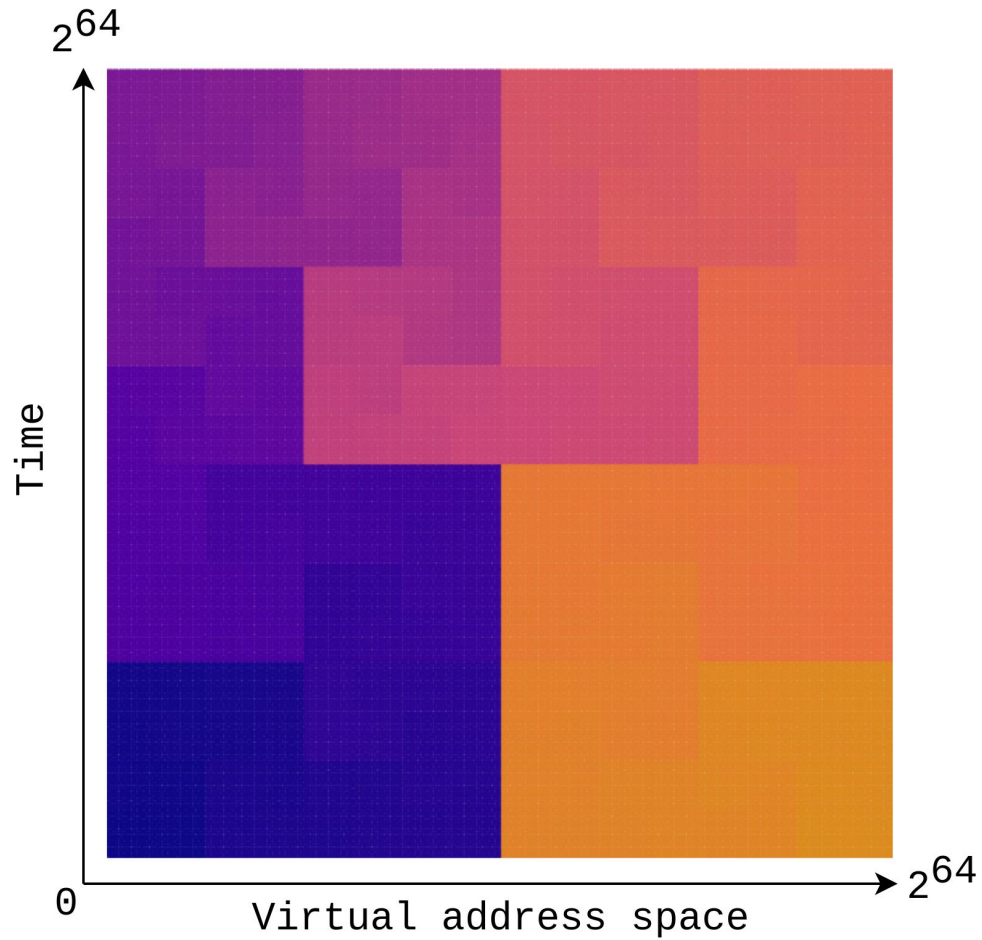
Passage à l'échelle

- **>500 millions d'instructions**
- **Tri en mémoire**
 - 32 octets par feuille
 - → Tri en 2 phases
- **Index inefficace**
 - Latence se dégrade très rapidement
 - → Densité des feuilles trop hétérogène
- **1 → 3 passes**

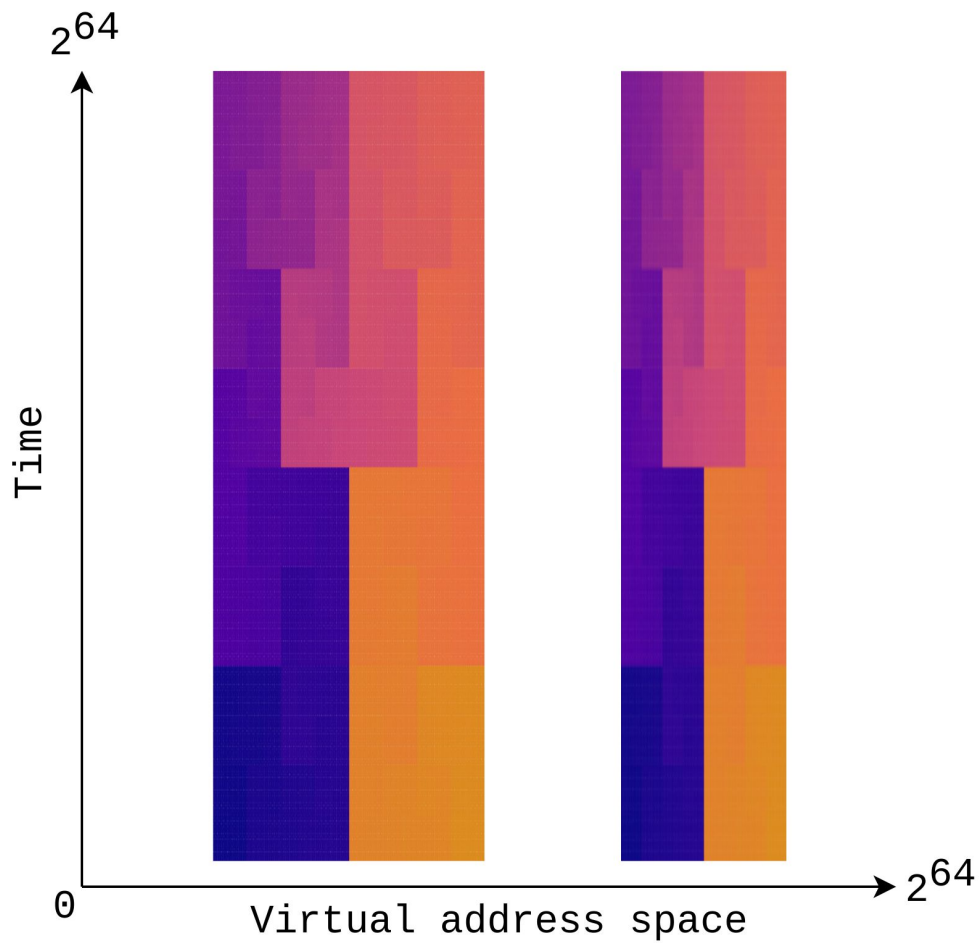
1^{ère} passe : régions



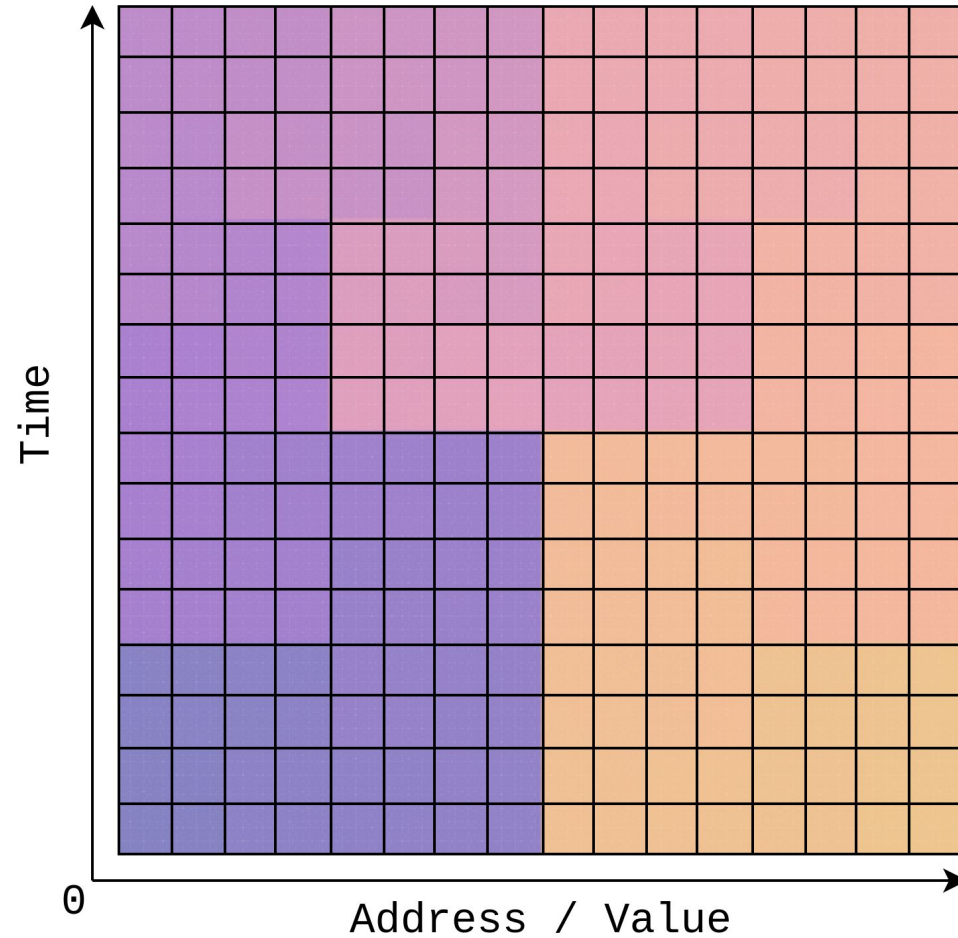
1^{ère} passe : régions



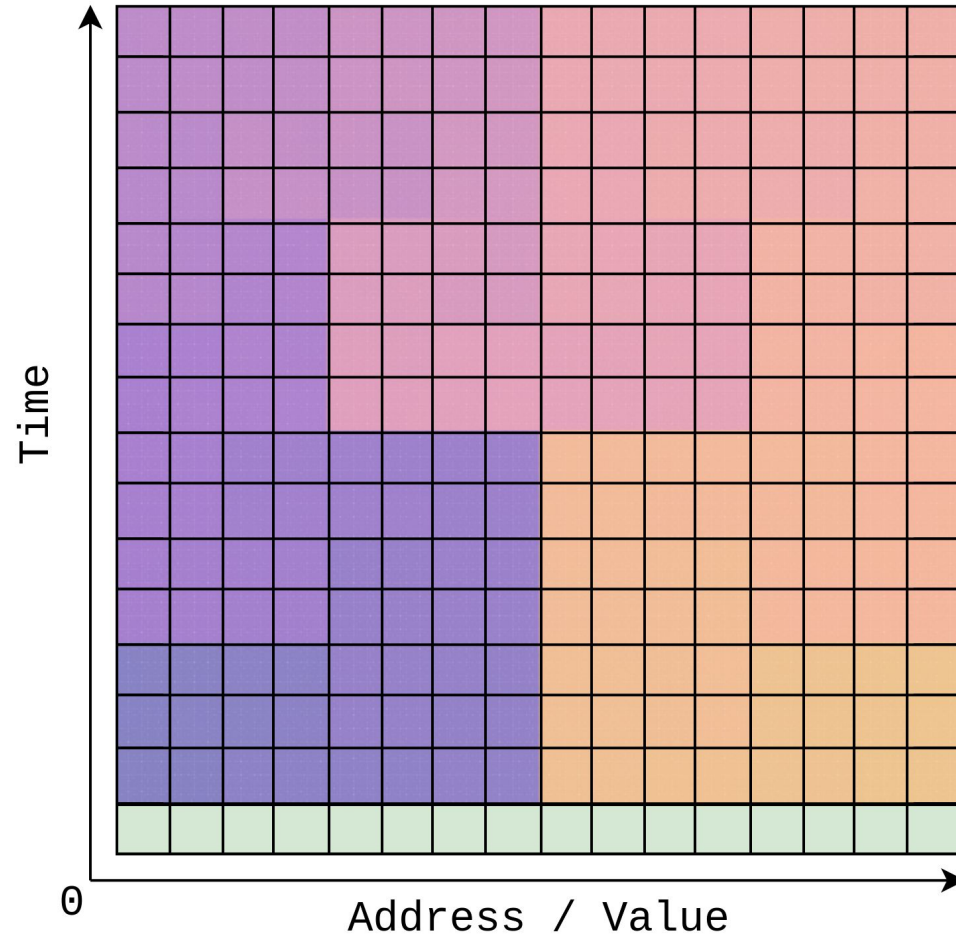
1^{ère} passe : régions



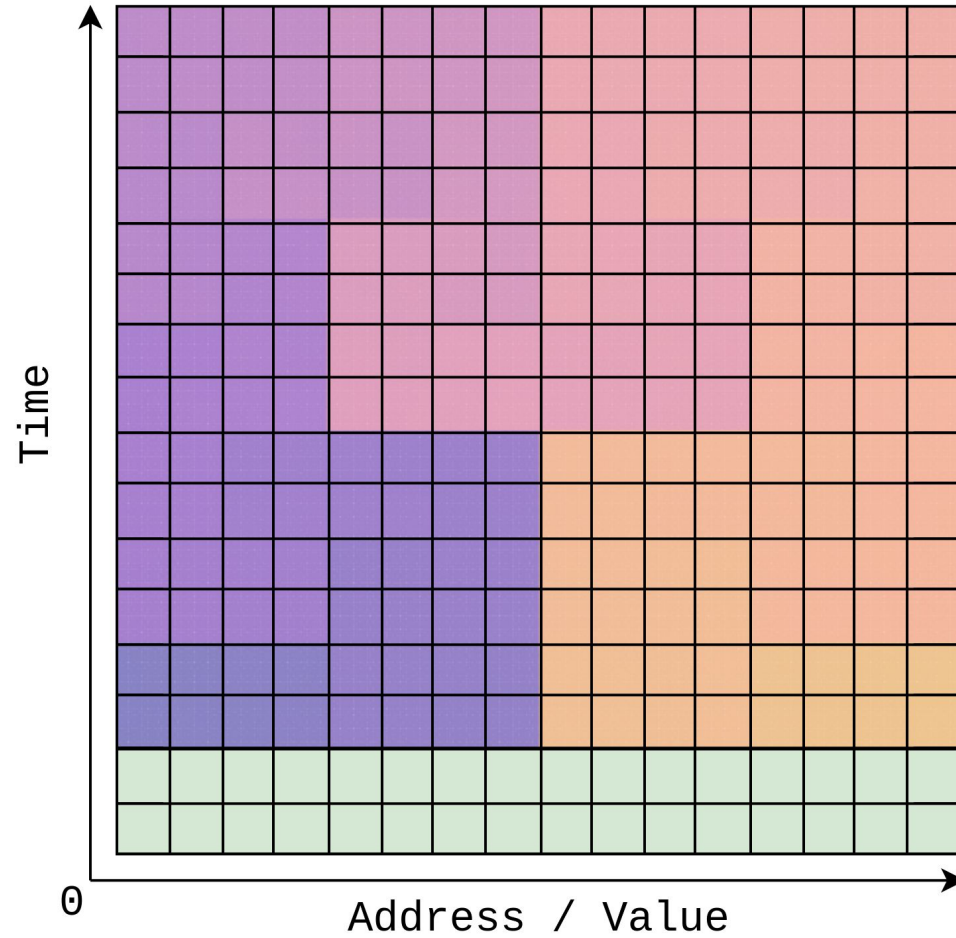
2^{ème} passe : partitions



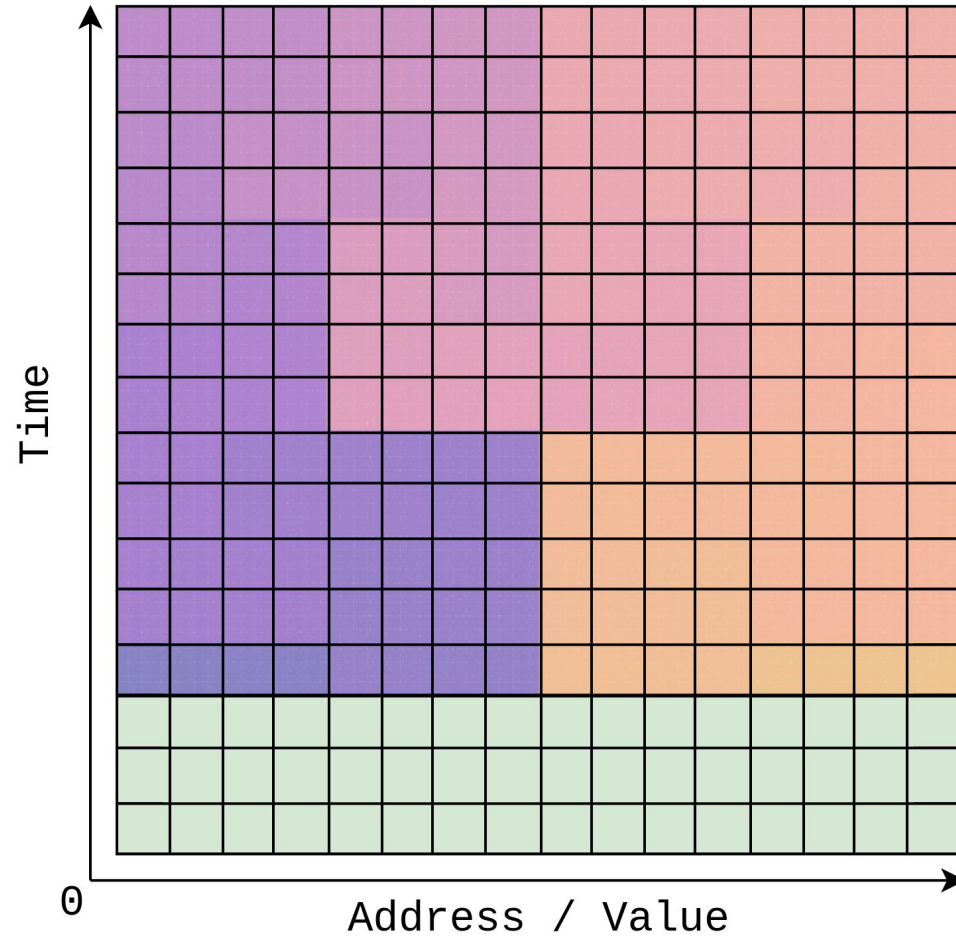
3^{ème} passe : fill



3^{ème} passe : fill



3^{ème} passe : fill



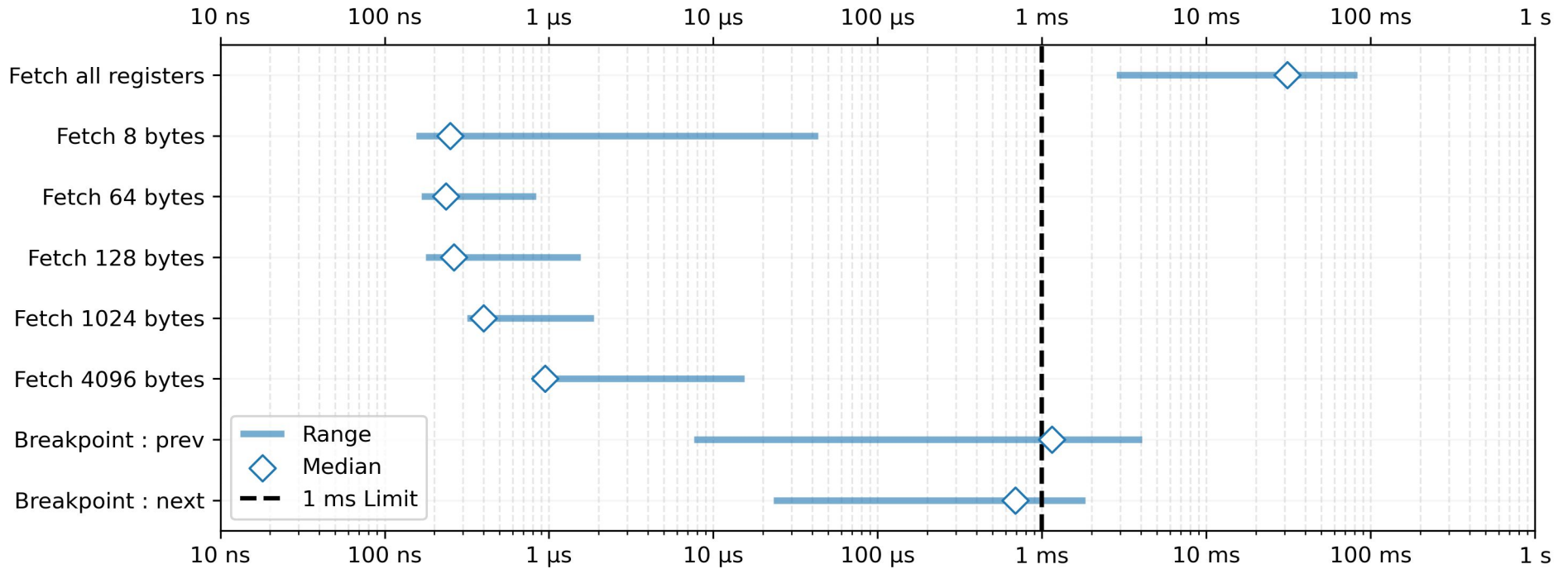
Benchmark : Spatial Frinet

- 1,7 milliards d'instructions / 90 Go

```
$ frinet index --format tenet wget.tenet wget.db
  Scout [00:08:45] ##### 90.98 GiB/90.98 GiB
Partitions [00:09:03] ##### 90.98 GiB/90.98 GiB
  Fill [00:12:47] ##### 90.98 GiB/90.98 GiB
[INFO frinet] Total time : 31 minutes
[INFO frinet] Maximum memory usage : 22.16 GiB
$ █
```

Benchmark : passage à l'échelle

Frinet spatial : 1.7 milliards d'instructions (98Go)



- **2 ans de recherche et d'expérimentation**
- **Réécriture complète du backend & frontend**
 - Vue « Callgraph » pas encore implémentée
 - L'interface mérite plus de travail
- **Open-source : <https://github.com/synacktiv/frinet>**

The logo for SYNACKTIV features a stylized icon on the left consisting of a 3x3 grid of squares. The top-left square is white, the top-middle square is white with a red dot, and the top-right square is white. The remaining squares are black. To the right of this icon, the word "SYNACKTIV" is written in a bold, sans-serif font. "SYN" is in white, and "ACKTIV" is in red.

SYNACKTIV



<https://www.linkedin.com/company/synacktiv>



<https://twitter.com/synacktiv>



<https://synacktiv.com>