

VITE

PFA ENSEIRB-MATMECA

Membre de l'équipe

BAHHOU HOUSSAM

BENMENDIL HAMZA

BOULLIT MOHAMED FAYÇAL

BREDEL NOLAN

ELOMARI ALAOUI ISMAIL

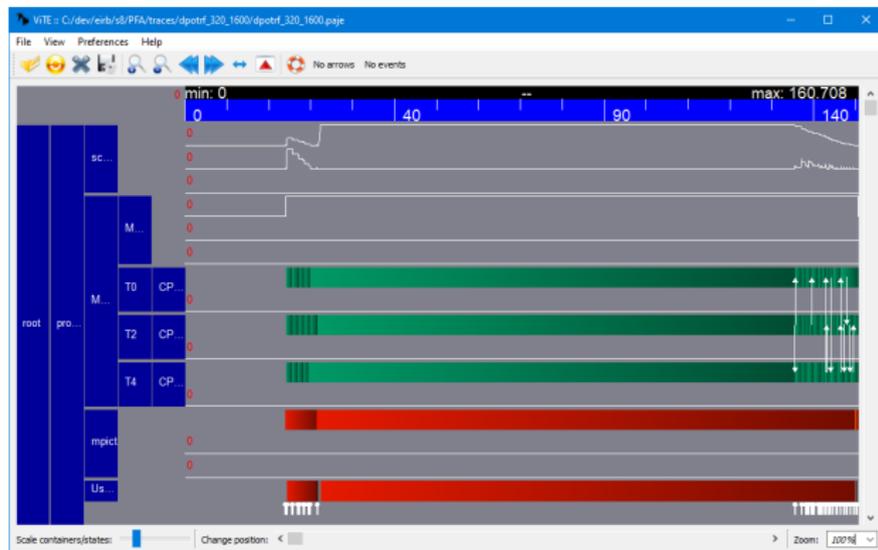
GAUCHET AUGUSTIN

GUEDON LUCAS





Responsable
pédagogique :
M. Rollet
Client :
M. Faverge



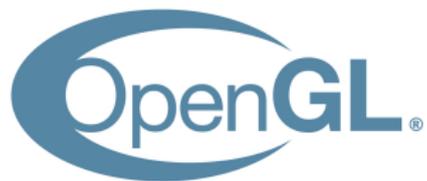
- 1 Module de rendu
 - Refactoring du code existant
 - Comparaison entre OpenGL et Vulkan
- 2 Plugin d'affichage de matrice (creuse)
 - Ancienne méthode
 - Méthode des QuadTree
 - Comparaison d'affichage
- 3 Plugin de statistiques
 - Parsing des fichiers donnés en entrée
 - Chemin critique
 - Niveau de parallélisation possible
 - Affichage du chemin critique et sa taille sur le diagramme du Gantt
- 4 Conclusion



- 1 Module de rendu
 - Refactoring du code existant
 - Comparaison entre OpenGL et Vulkan
- 2 Plugin d'affichage de matrice (creuse)
 - Ancienne méthode
 - Méthode des QuadTree
 - Comparaison d'affichage
- 3 Plugin de statistiques
 - Parsing des fichiers donnés en entrée
 - Chemin critique
 - Niveau de parallélisation possible
 - Affichage du chemin critique et sa taille sur le diagramme du Gantt
- 4 Conclusion



But de l'implémentation en Vulkan



But de l'implémentation en Vulkan



 Facile à prendre en main



But de l'implémentation en Vulkan



- 😊 Facile à prendre en main
- 😐 Performance moyenne



But de l'implémentation en Vulkan



- 😊 Facile à prendre en main
- 😐 Performance moyenne
- 😡 Désuet (Voir déprécié sous certain OS)

But de l'implémentation en Vulkan



- 😊 Facile à prendre en main
- 😐 Performance moyenne
- 😡 Désuet (Voir déprécié sous certain OS)



- 😡 Difficile à prendre en mains



But de l'implémentation en Vulkan



- 😊 Facile à prendre en main
- 😐 Performance moyenne
- 😡 Désuet (Voir déprécié sous certain OS)



- 😡 Difficile à prendre en mains
- 😊 Bonne performance

But de l'implémentation en Vulkan



- 😊 Facile à prendre en main
- 😐 Performance moyenne
- 😡 Désuet (Voir déprécié sous certain OS)



- 😡 Difficile à prendre en mains
- 😊 Bonne performance
- 😊 Moderne

Refactoring du code existant

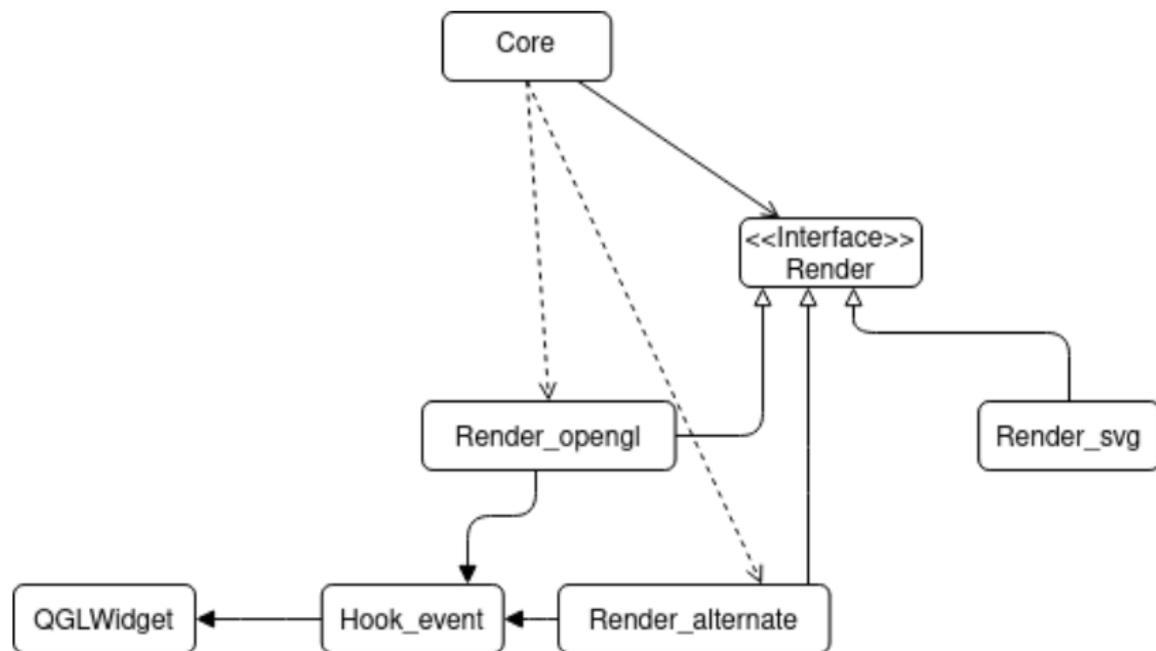


Figure: Diagramme de classes de l'ancien module de rendu

Refactoring du code existant

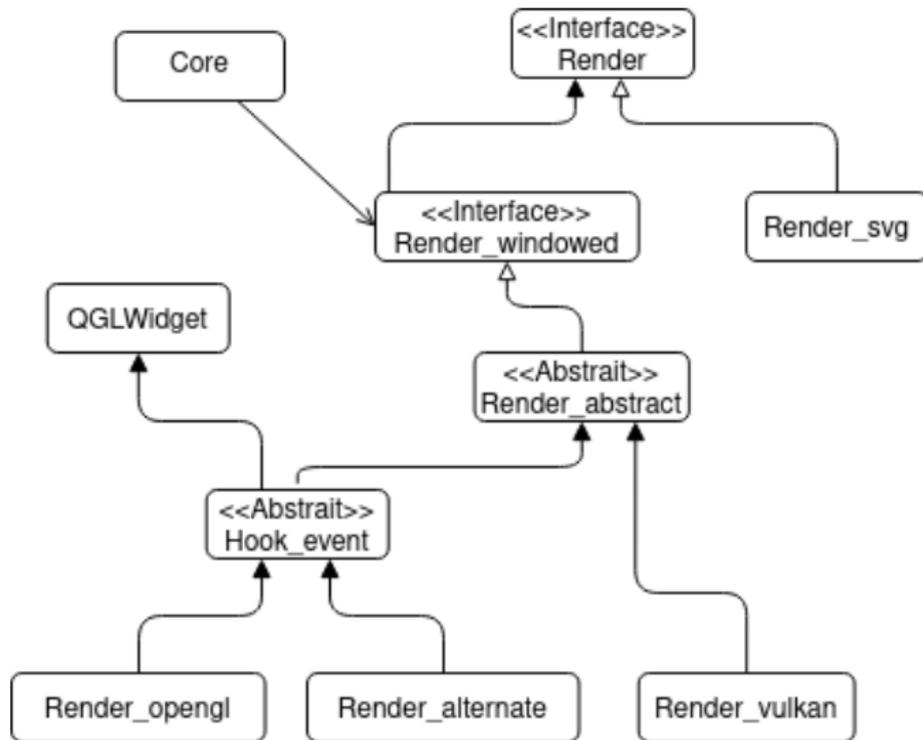


Figure: Diagramme de classes après notre remaniement



Comparaison entre OpenGL et Vulkan



Figure: Trace obtenue avec OpenGL



Figure: Même trace avec Vulkan

Présent :

- Rectangle des tâches
- Compteurs

Non présent :

- Flèches d'interaction
- Règle graduée

- 1 Module de rendu
 - Refactoring du code existant
 - Comparaison entre OpenGL et Vulkan
- 2 Plugin d'affichage de matrice (creuse)
 - Ancienne méthode
 - Méthode des QuadTree
 - Comparaison d'affichage
- 3 Plugin de statistiques
 - Parsing des fichiers donnés en entrée
 - Chemin critique
 - Niveau de parallélisation possible
 - Affichage du chemin critique et sa taille sur le diagramme du Gantt
- 4 Conclusion



Le but est de mettre à jour le plugin déjà existant qui a un résultat peu satisfaisant. On part de la matrice symbolique utilisée par PaStiX.

On cherche à mettre en avant :

- Dans quelle partie de la matrice on travail
- Visualiser le rang des blocs de la matrice en low-rank
- Visualiser la distribution sur les différents noeuds



Affichage de base dans ViTE

La compression des données dans cet affichage présente une énorme perte d'information:

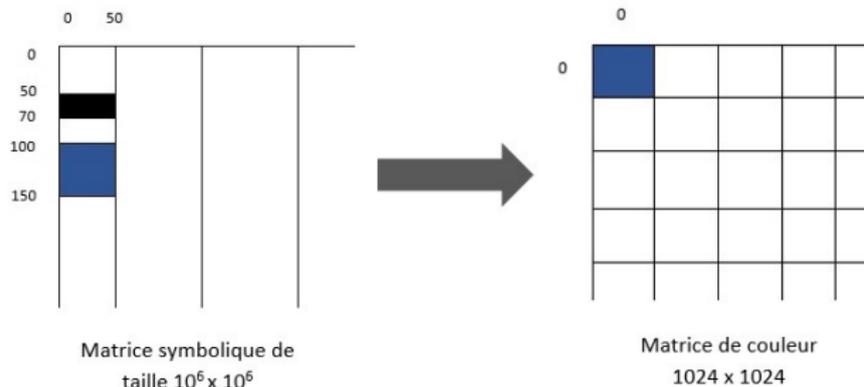
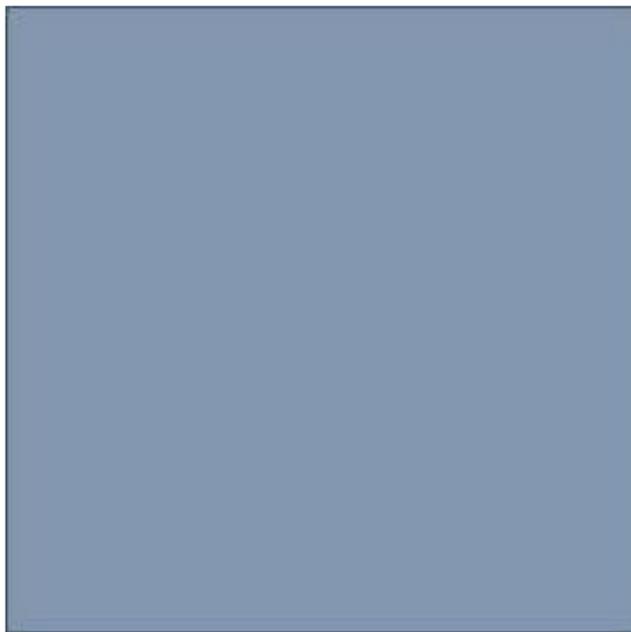
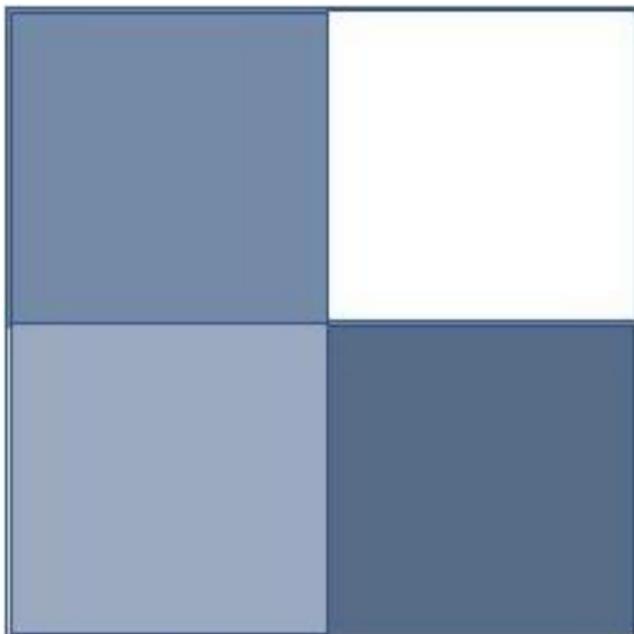


Figure: Résultat montrant le problème de l'affichage existant

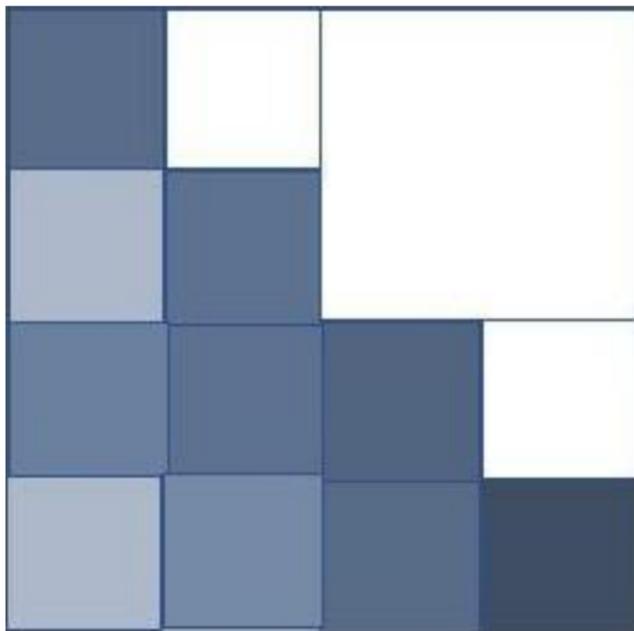
Nouvelle approche: Quadtree



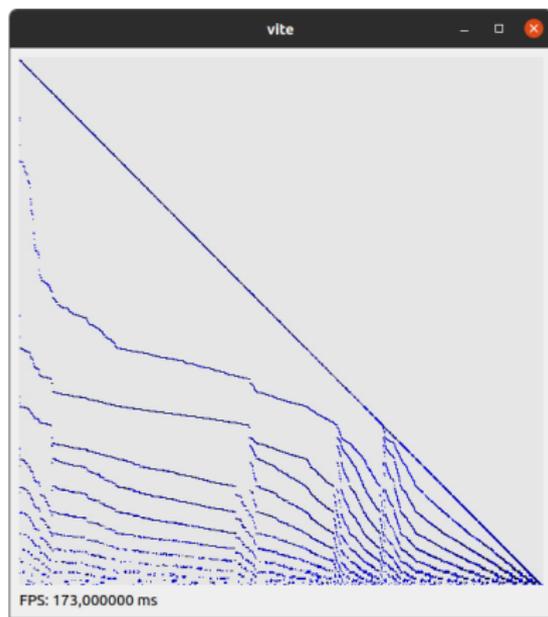
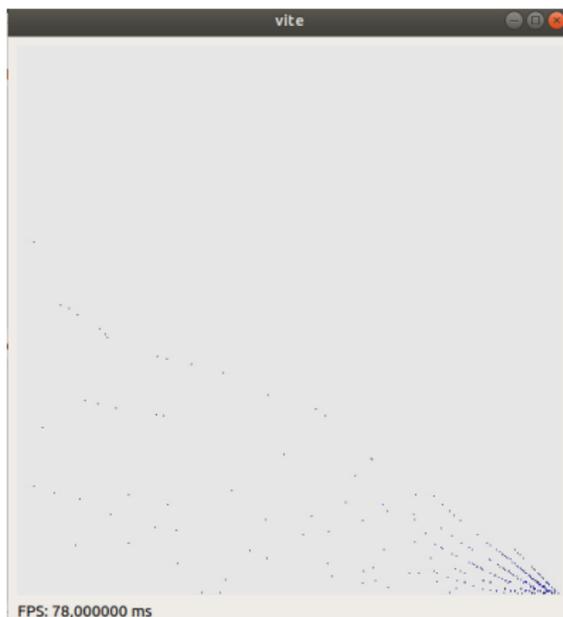
Nouvelle approche: Quadtree



Nouvelle approche: Quadtree



Comparaison d'affichage



- 1 Module de rendu
 - Refactoring du code existant
 - Comparaison entre OpenGL et Vulkan
- 2 Plugin d'affichage de matrice (creuse)
 - Ancienne méthode
 - Méthode des QuadTree
 - Comparaison d'affichage
- 3 Plugin de statistiques
 - Parsing des fichiers donnés en entrée
 - Chemin critique
 - Niveau de parallélisation possible
 - Affichage du chemin critique et sa taille sur le diagramme du Gantt
- 4 Conclusion

Plugin pour effectuer automatiquement différentes statistiques sur la trace.

- Le chemin critique (temps et nombre de tâche)
- Niveau de parallélisation possible

Parsing des fichiers donnés en entrée (.dot)

Utilisation de *BOOST* et de la fonction *read_graphviz()*.

Champs initialisés à l'aide des parsing des 2 fichiers:

- executionTime
- timeElapsed

```
using namespace boost;

struct Vertex {
    std::string task_name, label, fillcolor, style;
    int not_max_breadth;
    int task_id;
    double execution_time;
    double time_elapsed;
};

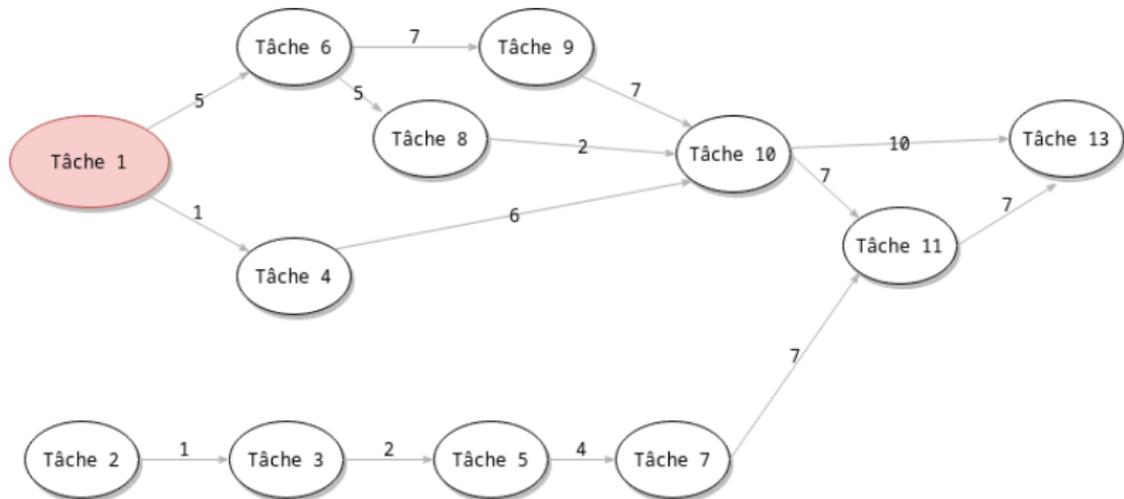
struct Edge {
    std::string color;
};
```

Figure: Structure de graphe dans notre module

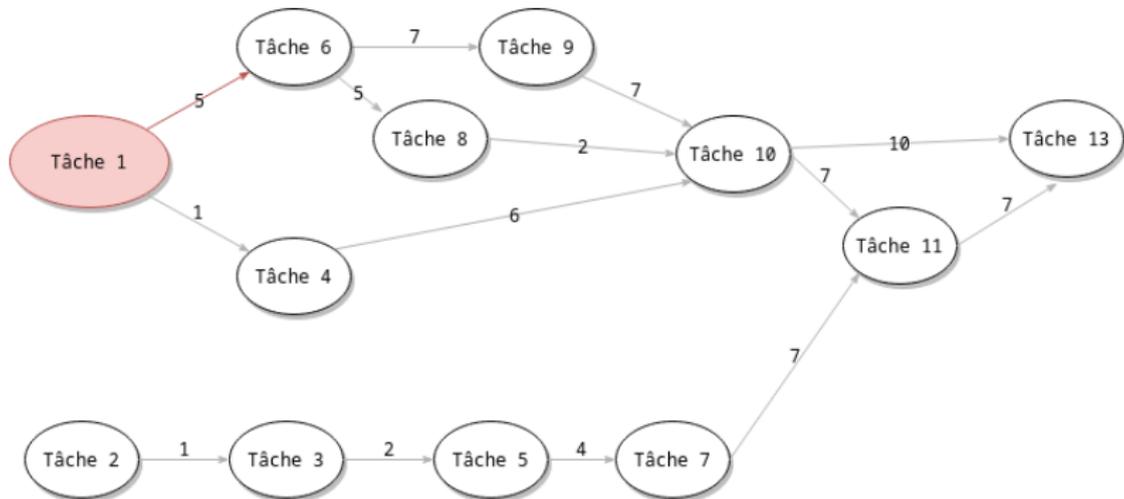
```
dynamic_properties dp(ignore_other_properties);
dp.property("node_id", get(&Vertex::name, graph));
dp.property("label", get(&Vertex::label, graph));
dp.property("fillcolor", get(&Vertex::fillcolor, graph));
dp.property("style", get(&Vertex::style, graph));
dp.property("color", get(&Edge::color, graph));
```

Figure: Propriétés initialisées dynamiquement

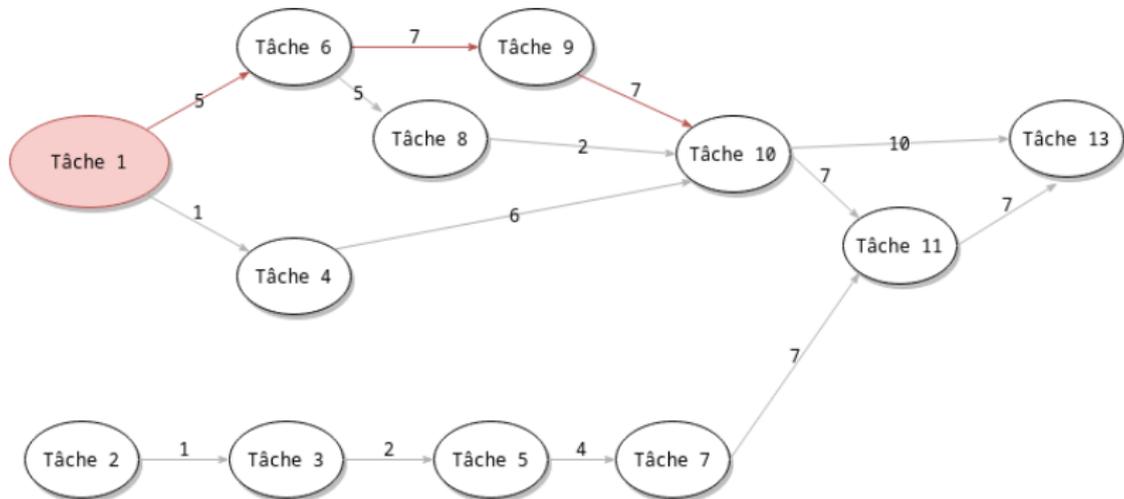
Longueur du chemin critique



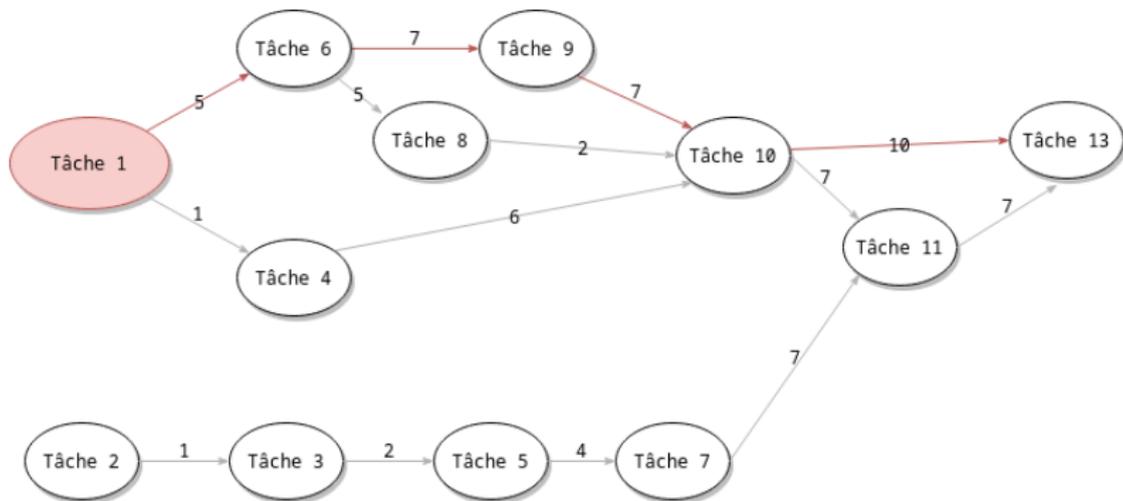
Longueur du chemin critique



Longueur du chemin critique



Longueur du chemin critique



Niveau de parallélisation possible



Affichage du chemin critique et sa taille sur le diagramme du Gantt

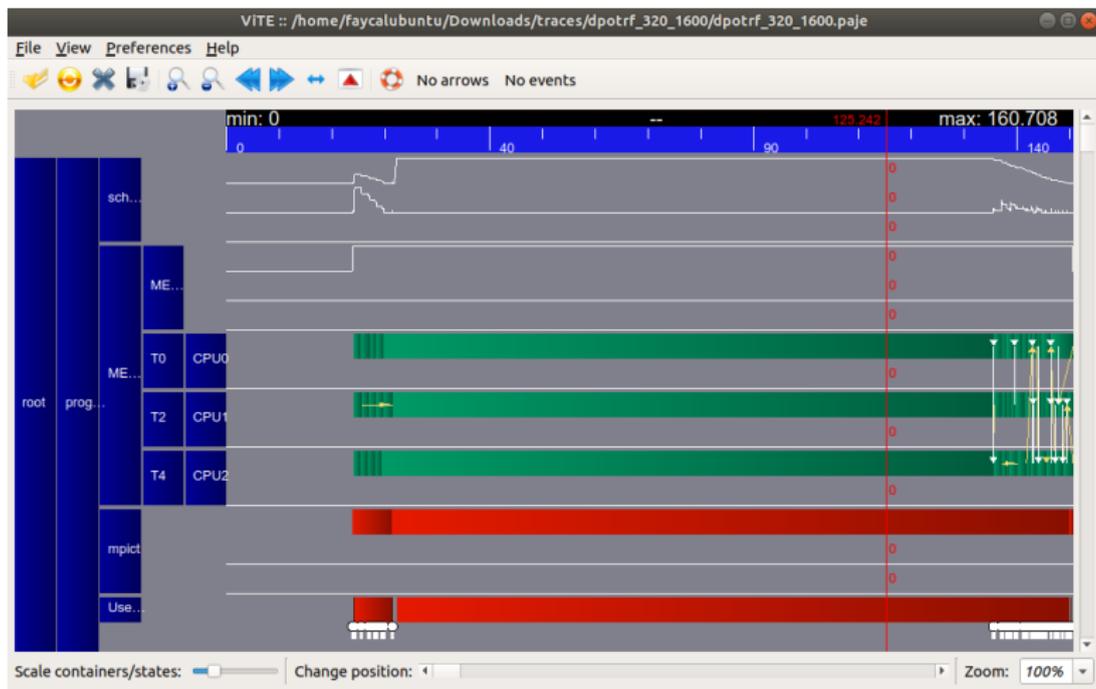


Figure: Aperçu sur l'affichage du chemin critique et sa taille sur le diagramme du Gantt

Aperçu sur la fenêtre du plugin

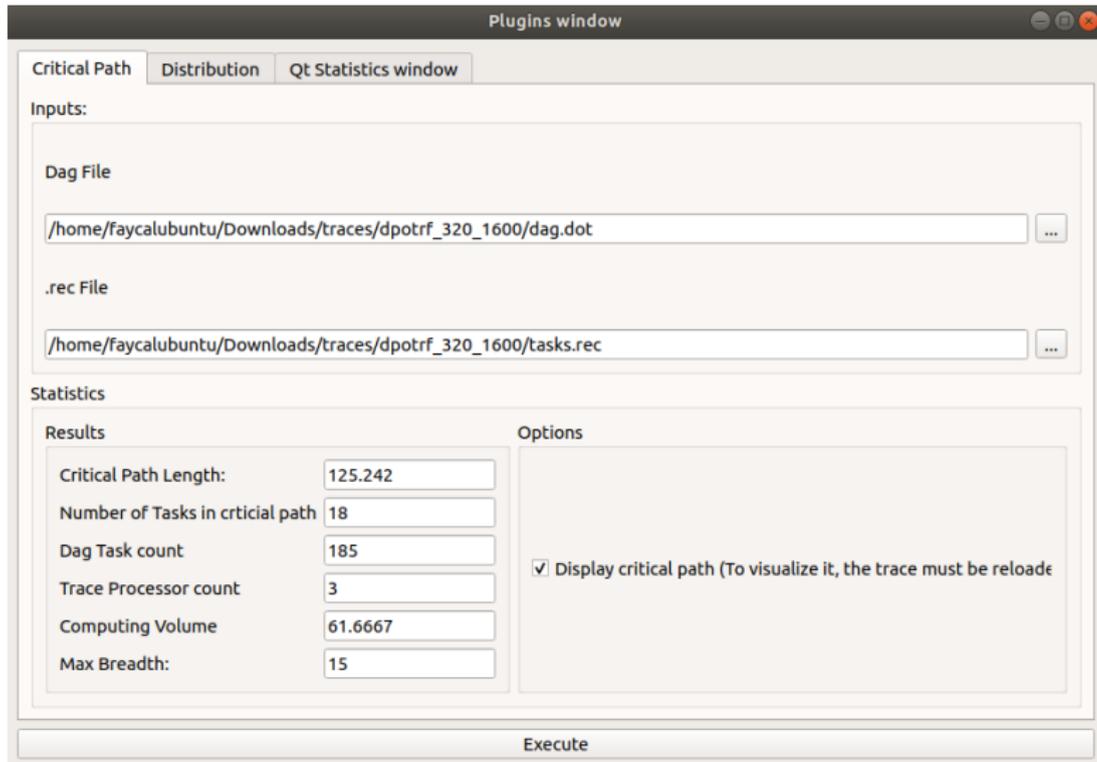


Figure: Aperçu sur la fenêtre du plugin

- 1 Module de rendu
 - Refactoring du code existant
 - Comparaison entre OpenGL et Vulkan
- 2 Plugin d'affichage de matrice (creuse)
 - Ancienne méthode
 - Méthode des QuadTree
 - Comparaison d'affichage
- 3 Plugin de statistiques
 - Parsing des fichiers donnés en entrée
 - Chemin critique
 - Niveau de parallélisation possible
 - Affichage du chemin critique et sa taille sur le diagramme du Gantt
- 4 Conclusion



- Les deux plugins sont achevés et conformes aux spécifications
- Bien que le module de rendu ne soit pas achevé, l'architecture adoptée permettra de continuer facilement ce qu'il reste à faire.

Site: <http://vite.gforge.inria.fr>

Merci pour votre attention

