Rust sur GPU avec Vulkan

Hadrien Grasland 2023-07-10

Audience

- Vous voulez **coder pour GPU en optimisant beaucoup**
  - Besoin d'accès fin aux fonctionnalités matérielles
  - Les couches d'abstraction ne doivent pas les cacher
- Vous avez besoin de **portabilité** entre OSs, matériels…
- Vous exigez une certaine **qualité/maturité** de l'implémentation
  - Installation/maintenance facile, en dev et en prod
  - Pas trop de bugs, d'APIs brisées tous les quelques mois…
- Une **API moins ergonomique** est un compromis acceptable

Pourquoi une API graphique ?

- Conçues pour permettre de **tirer le maximum** d'un GPU
  - Expose tout ce qu'on trouve dans une API calcul, et plus
- **Audience** jeux vidéos, CAO >> Calculs GPGPU
  - Installation facile, moins de bugs, API rarement changée
  - Outillage abondant (mais parfois trop orienté 3D)
- Certaines APIs conçues pour la **portabilité** OS/matériel
  - Bonnes perfs possibles à >95 % de code partagé
  - Grande audience → Les fabricants *doivent* les supporter

Pourquoi pas OpenGL ?

- **Conception obsolète** héritée de IrisGL de SGI (1992)
  - Pensé pour du matériel très différent des GPU modernes
  - Faible support du parallélisme (etat global implicite)
- Evolution limitée par la **rétro-compatibilité**
  - N façons de faire la même chose, dont beaucoup à éviter
  - Implémentations « futées » → Imprévisibles et buggées
- **Perte de vitesse** des évolutions en faveur de Vulkan

Vulkan vs OpenGL

- Maintenu par les mêmes auteurs (groupe Khronos)
- **Suppression** de nombreuses fonctionnalités obsolètes
- Pas d'état global → **Parallélisme** grandement facilité
- **Meilleur contrôle** de l'ordonnancement et des allocations
- Meilleur support de **mémoire unifiée**, **dallage**, **multi-GPU**
- **Pilote simplifié** → Moins de bugs, moins de surprises
- Contrepartie : **Très lourd** → Abstraire selon ses besoins !

Pourquoi pas WebGPU ?

- API **graphique bas niveau** (type Vulkan, Metal, Direct3D 12)
- Backends pour les 3 grandes APIs → **Support HW/OS optimal**
- Prévue pour le Web → **API simplifiée, pas d'UB possible**
- MAIS très jeune → **Fonctionnalités réduites, API mouvante**
- Ma recommandation future quand elle aura maturé ?
  - En attendant, Vulkan + un peu d'abstraction fait le travail

En résumé

	Support OS / Installation			Support matériel			API	
	Windows	macOS	Linux	Nvidia	AMD	Intel	Facile	Complet
CUDA	✔	✘	⚠	✔	✘	✘	✔	✔
HIP	?	✘	⚠	⚠	✔	✘	✔	✔
SYCL	⚠	✘	⚠	⚠	⚠	✔	✔	⚠
OpenCL	⚠	💀	⚠	💀	⚠	⚠	⚠	✘
Direct3D	✔	✘	⚠	✔	✔	✔	✘	✔
Metal	✘	✔	✘	💀	💀	💀	⚠	✔
OpenGL	⚠	💀	✔	✔	✔	✔	⚠	⚠
Vulkan	⚠	⚠	✔	✔	✔	✔	✘	✔
WebGPU	✔	✔	✔	✔	✔	✔	⚠	⚠

Conclusion générale

- D’un côté, on a une API graphique bas niveau
  - Support matériel excellent
  - Utilisation difficile car riche en UB
- De l’autre, un langage qui rend le bas niveau ergonomique
- Essayons de combiner les deux !

Passons au TP !