

Groupe 9

INFO5

Cahier des charges

Dashboard en technologie egui/rust pour plateforme HPC

ARLE Alexandre
DEL MEDICO Rémi
GOGUILLOT Amaury
MIRAS Romain

2024/2025
27 Janvier 2025 - 14 Mars 2025

Table des matières

I. Cahier des charges	1
A. Contexte	1
B. Équipe projet	1
C. Objectifs.....	1
D. Périmètre fonctionnel du projet.....	1
E. Livrables attendus	2
F. Planning prévisionnel	2
G. Ressources nécessaires.....	3
H. Risques identifiés et solution	3
I. Conclusion	3

I. Cahier des charges

A. Contexte

Le gestionnaire de tâches et de ressources **OAR** ([lien](#)) est utilisé dans des plateformes HPC (High Performance Computing) pour orchestrer et gérer les tâches et les ressources. Cependant, les interfaces graphiques associées sont souvent limitées ou absentes. Ce projet vise à développer une interface utilisateur graphique moderne et performante à l'aide de la bibliothèque **Egui** ([lien](#)), un framework basé sur **Rust** ([lien](#)) et utilisant une **GUI en mode immédiat** ([lien](#)).

B. Équipe projet

Nous sommes une équipe composée de 4 étudiants en 5ème année d'école d'ingénieur en informatique. Rôles proposés (à préciser) :

- **Chef de projet** : Gestion du suivi, planification et coordination de l'équipe.
- **Développeurs principaux** : Conception et implémentation de la solution en Rust/Egui.
- **Responsable des tests** : Conception des scénarios de test et validation du démonstrateur.
- **Responsable documentation** : Rédaction des livrables et supports techniques.

C. Objectifs

- **Document sur l'architecture de la solution** :
 - Élaboration d'un diagramme d'architecture décrivant les interactions principales entre les composants du projet.
 - Création d'un diagramme de séquence illustrant les flux d'informations dans les principaux cas d'utilisation.
- **Démonstrateur fonctionnel** :
 - Conception de la structure d'un dashboard complet qui inclut les modules suivants :
 - * Diagramme de Gantt : Visualisation des tâches planifiées, inspirée du module flamegraph de Puffin ([lien](#)).
 - * État des ressources : Affichage en temps réel ou simulé de l'utilisation des ressources HPC (CPU, mémoire, réseau, etc.).
 - Développement d'une procédure de démonstration et de tests, reposant sur des données fabriquées ou simulées.

D. Périmètre fonctionnel du projet

- Fonctionnalités principales :
 - Tableau de bord principal :
 - * Menu de navigation (Accueil, Ressources, Gantt, etc.).
 - * Résumé de l'état des ressources sous forme de graphiques simples (barres, jauges).
 - Diagramme de Gantt :
 - * Affichage des tâches avec leurs dates de début, fin, et état d'avancement.
 - * Visualisation graphique des dépendances ou chevauchements entre tâches.
 - Visualisation des ressources :
 - * Représentation graphique en temps réel ou simulée (CPU, RAM, réseau, stockage).
 - * Indicateurs de performance globaux (e.g., taux d'utilisation, disponibilité).

- Contraintes techniques :
 - Technologies utilisées :
 - * Langage principal : Rust
 - * Bibliothèque GUI : Egui/eframe
 - Compatibilité :
 - * Application native avec support multi-plateforme (Linux, Windows, macOS).
- Limites :
 - Le projet ne prévoit pas, dans sa version initiale, d'intégration avec un backend réel de gestion des tâches OAR (simulation uniquement).

E. Livrables attendus

- Documentation technique :
 - Diagramme d'architecture.
 - Diagrammes de séquence pour les cas d'utilisation principaux (e.g., ajout d'une tâche, mise à jour des ressources).
 - Manuel utilisateur pour le démonstrateur.
- Code source :
 - Dépôt Git contenant un code documenté et organisé.
 - Instructions pour compiler et exécuter l'application.
- Démonstrateur :
 - Application GUI fonctionnelle avec des données simulées.
- Rapport final :
 - Résumé des choix techniques et méthodologiques.
 - Analyse des résultats obtenus et des perspectives d'évolution.

F. Planning prévisionnel

Étape	Description	Durée estimée
Phase 1 : Analyse	Analyse des besoins, état de l'art et préparation	1 semaine
Phase 2 : Conception	Définition de l'architecture et des diagrammes	1 semaine
Phase 3 : Développement	Implémentation du dashboard et des modules	2-4 semaines
Phase 4 : Documentation	Rédaction du rapport et manuel utilisateur	1 semaine

Table 1 : Planning

G. Ressources nécessaires

- Compétences techniques :
 - Connaissance du langage Rust.
 - Familiarité avec la bibliothèque Egui/eframe.
 - Compréhension des concepts de HPC et des gestionnaires de tâches comme OAR (formation ou documentation).
- Environnement de travail :
 - IDE recommandé : Visual Studio Code avec extensions Rust ou JetBrains CLion.
 - Outils de versionnement : Git (hébergement sur GitHub ou GitLab).
 - Documentation Rust et Egui pour référence.
- Données :
 - À produire : Données de simulation pour les tâches et ressources.
 - Des jobs OAR et les informations qui leurs sont liée
 - Récupérer via des call API

H. Risques identifiés et solution

Risque	Impact	Solution
Difficulté à maîtriser Egui/Rust	Élevé	Prévoir du temps pour monter en compétence (tutoriels, exemples).
Retards dans la conception	Moyen	Organisation claire des tâches (GitHub issues et tableau Kanban).
Problème d'intégration des modules	Moyen	Tests réguliers pour chaque composant.

Table 2 : Risques identifiés et solutions

I. Conclusion

Ce projet vise à fournir une interface graphique moderne et performante pour visualiser et gérer les tâches sur des plateformes HPC. Grâce à la bibliothèque Rust/Egui, l'équipe développera un démonstrateur fonctionnel basé sur des données simulées, tout en documentant l'architecture et les processus clés. Des étapes de tests et de validation garantiront la qualité du produit livré.