

Thomas MADRANGE-MAIRE
Julie LEROY
Valérian LAMBERT
Clément BERARD
Arsène JEROME

INF3-FI



IUT de Vélizy-Rambouillet
CAMPUS DE VÉLIZY-VILLACOUBLAY
CAMPUS DE RAMBOUILLET

SAE5

Recueil des besoins

2025 / 2026

Chapitre 1 Objectif et portée

(a) Les objectifs généraux et la portée du système

Objectifs généraux du système

Dans ce projet le principal objectif est de concevoir et déployer une plateforme permettant l'exécution de programmes de calcul distribués ou parallèle sur un cluster de Raspberry Pi.

La porté du système

- Mise en place du Cluster : Installer et configurer les nœuds du cluster Raspberry Pi pour qu'ils puissent communiquer et exécuter des tâches distribuées.
- Développement d'une Interface web sécurisé accessible aux utilisateurs authentifiés
- Gestion de plusieurs tâches : Implémenter un programme permettant d'envoyer et d'exécuter des tâches dans les différents RPI du cluster et de récupérer les résultats obtenus.
- Affichage des résultats : Afficher sur la page web les résultats obtenus après calcul

(b) Les intervenants

Acteurs	Rôle
Client	Valider les livrables de sorte à contrôler la bonne réalisation du projet
Utilisateur Enregistré	Lancer des calculs et regarder le résultat sur une interface facile d'utilisation, fiable et rapide
Équipe de développement	Conçoit et développe le système en suivant les spécifications et en faisant des choix sur les technologies adaptées tout en respectant les délais

(c) Ce qui est et ce qui n'est pas dans la portée

Dans la portée	Hors portée
Configuration du Cluster et paramétrage des RPIs	Gestion de rôle utilisateur (administrateur)
Interface Web	Confirmation d'inscription par email ou SMS.

Lancement de calcul pour les utilisateurs connecté	Récupération de mot de passe.
Affichage des résultats	journaux d'activité administrateur
Historique des calculs

Chapitre 2 Glossaire

Terme	Définition
Calcul Distribué	Méthode d'exécution d'un programme sur plusieurs machines connectées, où l'état et les données sont partagés et synchronisés.
Calcul Parallèle	Exécution simultanée de plusieurs instructions de programme sur différents processeurs/cœurs pour réduire le temps de calcul.
Cluster Raspberry Pi	Ensemble d'ordinateurs monocartes Raspberry Pi interconnectés et configurés pour fonctionner comme un seul système de calcul haute performance (HPC).
Interface Web	Il s'agit de la partie front-end d'une application (c'est-à-dire la partie graphique visible généralement sur le navigateur). Cette interface est utilisée par l'utilisateur pour interagir avec le système.
Nœud (Node)	Il s'agit d'une des cartes Raspberry Pi dans le cluster. On distingue souvent le Nœud Maître (qui gère les tâches) des Nœuds Esclaves (qui effectuent les calculs).
Utilisateur inscrit	Utilisateur ayant un compte lui permettant d'accéder au tableau de bord et aux modules de calcul.
Visiteur	Utilisateur non inscrit qui accède à la page d'accueil de la plateforme.

Chapitre 3 Les cas d'utilisation

(a) Les acteurs principaux et leur objectifs généraux

Acteur	Objectif général
Utilisateur Enregistré	Lancer des calculs et obtenir des résultats fiable

Système Cluster	Exécuter les différentes tâches de calcul demandées par l'interface web et distribuer la charge
Visiteur	Visualiser les informations disponibles et une vidéo explicative.

(b) Les cas d'utilisation

Cas d'utilisation n°1 : Vidéo explicative

Nom : Visionner la vidéo explicative

Contexte d'utilisation : Le visiteur découvre la page d'accueil du site et souhaite comprendre les fonctionnalités avant de s'inscrire.

Portée : Site web de la plateforme (front-end).

Niveau : Utilisateur.

Acteur principal : Visiteur.

Intervenants et intérêts :

- **Visiteur** : souhaite évaluer l'intérêt du service avant de créer un compte.
- **Système** : doit afficher la vidéo sans erreur et permettre une lecture fluide.
Précondition : Le site est accessible et la vidéo hébergée correctement.
Garantie minimale : Le visiteur voit une miniature ou une zone vidéo, même si la lecture échoue.
Garantie en cas de succès : La vidéo s'affiche et se lit jusqu'à la fin.
Déclencheur : Le visiteur accède à la page d'accueil.

Scénario nominal :

1. Le visiteur ouvre la page d'accueil.
2. Le site affiche une vidéo explicative visible sans défilement.
3. Le visiteur clique sur le bouton de lecture.
4. La vidéo démarre et se lit jusqu'à la fin.

Extension : Le visiteur peut mettre en pause ou relancer la vidéo.

Liste de variantes : Vidéo en plein écran, lecture automatique (optionnelle).

Informations connexes : Peut être hébergée sur une plateforme externe (YouTube, Vimeo).

Cas d'utilisation n°2 : Formulaire d'inscription

Nom : S'inscrire sur la plateforme

Contexte d'utilisation : Le visiteur souhaite créer un compte utilisateur pour accéder aux

fonctionnalités avancées.

Portée : Système d'authentification du site.

Niveau : Utilisateur.

Acteur principal : Visiteur.

Intervenants et intérêts :

- **Visiteur** : souhaite s'enregistrer pour accéder aux modules de calcul.
- **Système** : enregistre les données et crée un compte sécurisé.
Précondition : L'utilisateur est sur le site et clique sur le bouton "S'inscrire" dans la barre de navigation.
Garantie minimale : Formulaire visible mais affiche message d'erreur.
Garantie en cas de succès : Le compte est créé et un message de confirmation s'affiche. L'utilisateur est redirigé vers la page d'accueil et est connecté à son profil, il peut accéder aux modules de calculs.
Déclencheur : Clic sur le bouton "S'inscrire".
Scénario nominal :

1. Le visiteur clique sur "S'inscrire".
2. Le site affiche le formulaire d'inscription.
3. Le visiteur saisit ses informations (nom, email, mot de passe).
4. Le système affiche un **captcha** à résoudre.
5. Le visiteur remplit le captcha et soumet le formulaire.
6. Le système vérifie la validité des données et crée le compte.
7. Un message confirme l'inscription et propose de se connecter.

Extension :

- Si le captcha est incorrect → message d'erreur.
- Si un champ obligatoire est manquant → avertissement et re-soumission.

Liste de variantes : Aucune de prévu

Informations connexes : Lié au cas d'utilisation "Formulaire de connexion".

Cas d'utilisation n°3 : Formulaire de connexion

Nom : Se connecter à la plateforme

Contexte d'utilisation : L'utilisateur inscrit souhaite accéder à son espace personnel.

Portée : Système d'authentification.

Niveau : Utilisateur.

Acteur principal : Utilisateur inscrit.

Intervenants et intérêts :

- **Utilisateur inscrit** : veut accéder à son tableau de bord.
- **Système** : vérifie les identifiants et ouvre la session utilisateur.
Précondition : L'utilisateur dispose déjà d'un compte.
Garantie minimale : Le formulaire s'affiche même si la connexion échoue + message d'erreur ("nom ou mot de passe incorrect")
Garantie en cas de succès : L'utilisateur est redirigé vers son tableau de bord.
Déclencheur : L'utilisateur clique sur "Connexion" dans la barre de navigation.
Scénario nominal :

1. L'utilisateur clique sur "Connexion".
2. Le site affiche le formulaire de connexion.
3. L'utilisateur saisit ses identifiants.
4. Le système vérifie les informations.
5. L'utilisateur est redirigé vers le tableau de bord.

Extension :

- Identifiants incorrects → message d'erreur.
- Clic sur "Mot de passe oublié" → redirection vers la page de réinitialisation.
Informations connexes : Lien direct avec le cas "Mot de passe oublié".

Cas d'utilisation n°5 : Tableau de bord

Nom : Accéder au tableau de bord

Contexte d'utilisation : L'utilisateur souhaite visualiser les modules disponibles après connexion.

Portée : Interface principale après authentification.

Niveau : Utilisateur.

Acteur principal : Utilisateur inscrit.

Intervenants et intérêts :

- **Utilisateur** : veut choisir un module de calcul.
- **Système** : affiche les modules et leurs accès.
Précondition : L'utilisateur est connecté.
Garantie minimale : La page du tableau de bord s'affiche.
Garantie en cas de succès : Les modules disponibles sont affichés et cliquables.
Déclencheur : Connexion réussie ou clic sur "Tableau de bord".

Scénario nominal :

1. L'utilisateur se connecte ou clique sur "Tableau de bord".
2. Le système affiche les modules de calcul disponibles.
3. L'utilisateur clique sur un module pour y accéder.

Extension : Aucun module disponible → le bouton tableau bord ne s'affiche pas si l'utilisateur n'est pas connecté.

Liste de variantes : Accès rapide à l'historique ou au profil via des raccourcis.

Informations connexes : Lien avec les cas "Profil utilisateur" et "Historique des résultats".

Cas d'utilisation n°6 : Profil utilisateur

Nom : Modifier son mot de passe

Contexte d'utilisation : L'utilisateur souhaite changer son mot de passe depuis son espace personnel.

Portée : Système d'authentification.

Niveau : Utilisateur.

Acteur principal : Utilisateur inscrit.

Intervenants et intérêts :

- **Utilisateur :** veut renforcer la sécurité de son compte.
- **Système :** enregistre et vérifie la cohérence du nouveau mot de passe.
Précondition : L'utilisateur est connecté.
Garantie minimale : Le formulaire est affiché.
Garantie en cas de succès : Le nouveau mot de passe est enregistré.
Déclencheur : Clic sur "Profil" depuis le tableau de bord.

Scénario nominal :

1. L'utilisateur clique sur "Profil".
2. Le système affiche les informations de compte.
3. L'utilisateur saisit son ancien et son nouveau mot de passe.
4. Le système vérifie la validité et enregistre le nouveau mot de passe.
5. Message de confirmation affiché.

Extension : Ancien mot de passe incorrect → message d'erreur.

Liste de variantes : Option de déconnexion depuis la même page.

Informations connexes : Lien avec "Formulaire de connexion" et "Réinitialisation du mot de passe".

Cas d'utilisation n°7 : Historique des résultats

Nom : Consulter et gérer l'historique des résultats

Contexte d'utilisation : L'utilisateur souhaite retrouver les résultats précédents de ses calculs.

Portée : Espace utilisateur.

Niveau : Utilisateur.

Acteur principal : Utilisateur inscrit.

Intervenants et intérêts :

- **Utilisateur :** veut visualiser, supprimer ou télécharger ses résultats.
- **Système :** conserve et met à disposition les historiques.
 - Précondition :** L'utilisateur est connecté et dispose de résultats enregistrés.
 - Garantie minimale :** La page s'affiche, même vide.
 - Garantie en cas de succès :** Les résultats sont listés avec les actions disponibles.
 - Déclencheur :** Clic sur "Historique" depuis le tableau de bord.

Scénario nominal :

1. L'utilisateur clique sur "Historique".
2. Le système affiche la liste des résultats enregistrés.
3. L'utilisateur peut :
 - a. Télécharger un résultat.
 - b. Supprimer un résultat précis.
 - c. Supprimer tout l'historique.
4. Le système met à jour la page selon les actions effectuées.
 - Extension :** Aucun résultat → message "Aucun historique disponible".
 - Liste de variantes :** Export CSV/PDF.
 - Informations connexes :** Relié au cas "Tableau de bord".

Chapitre 4 La technologie employée

(a) Les exigences technologiques pour ce système

Domaine	Exigence	Justification
Matériel	Kit cluster Raspberry PI (1 RPI 4 (1.8 GHz) et 4 RPis Zéro(1.0 GHz))	Matériel fourni par les commanditaire pour mener à bien le projet

OS du Cluster	(à ajouter)	(à ajouter)
Langage de script	Python	Langage donnée dans l'exemple et assez facile d'utilisation pour l'équipe
Serveur web	Apache	Serveur pour héberger le site web
Web	HTML, PHP, CSS	Langages basiques pour la plateforme web.
Gestion des tâches	MPI	Nécessaire pour coordonner l'exécution du code Python sur plusieurs nœuds
Base de donnée	MySQL/MariaDB	Pour stocker les données des utilisateurs

(b) Avec quels systèmes ce système s'interfacerait-il et avec quelles exigences ?

Chapitre 5 Autre Exigence

(a) Processus de développement

i) Les participants au projet

- Un chef d'équipe : Il est responsable de la gestion des tâches à faire (il qualifie leur priorité et le temps accordé), et de la communication.
- Développeurs Frontend : Ils sont responsables de l'interface Web. Ils doivent veiller à créer une interface accessible et ergonomique.
- Développeurs Backend : Ils sont responsables de la logique côté serveur et de la configuration du cluster

ii) Les valeurs à privilégier

- Simplicité d'utilisation : L'interface utilisateur doit être le plus simple à utiliser
- Souplesse : La plateforme doit être capable d'intégrer de nouveaux scripts sans modification majeure de l'infrastructure
- Rapidité : Le système doit être le plus rapide possible
- Fiabilité : Le système doit fournir des résultats exactes

iii) Quels retours ou quelle visibilité sur le projet les utilisateurs et commanditaires souhaitent-ils ?

- Utilisateur : Il veut voir les résultats obtenus par son script
- Commanditaires : Il veut connaître l'avancement du projet de manière régulière avec éventuellement des démonstrations des nouvelles fonctionnalités implémentées

iv) Que doit-on fournir ? Que doit-on construire ? Qui sont nos concurrents ?

- A fournir : Le cluster Raspberry Pi, les librairies et logiciels
- Construire : L'intégralité de l'interface web, la système de gestion des tâches
- Concurrent : Aucun

v) Les autres exigences

- Tests : Des tests unitaire pour les fonctions backend et des test d'intégration (Test de la chaîne : Soumission Web -> Exécution Cluster -> Affichage Résultat)
- Installation/Déploiement : Une procédure de déploiement claire pour réinstaller la plateforme rapidement sur un nouveau cluster si nécessaire.

vi) Les dépendances auxquels le projet est soumis

- Dépendance au kit cluster Raspberry Pi au niveau de sa stabilité
- Dépendance au script donné par l'utilisateur

(b) Règle métier

1/ Seul les utilisateurs connectés peuvent utiliser les fonctionnalités du système

2/ Les scripts de calcul soumis devront avoir un type spécifique (certainement Python)

3/ Un tâche utilisera autant de ressource qu'une autre tâche

4/ Les mots de passe des utilisateurs seront stockés de manière sécurisée

5/ L'interface respectera les règles d'accessibilité

6/ Les utilisateurs inscrits peuvent uniquement modifier leur mot de passe, effectuer des opérations sur les différents modules, gérer leurs historiques (consultation, suppression d'entrées, suppression totale), et supprimer leurs comptes

(c) Performances

(ici c'est surtout le temps de réponse type le temps pour afficher, le temps pour lancer une tâche...)

(d) Opération, sécurité, documentation

- Opérations : Mise en place d'un système de logging centralisé sur le noeud maître
- Sécurité : Mise en place de sécurité contre les attaques par injections. Contrôle au niveau du stockage des données, c'est-à-dire que seul les données essentiels sont gardés
- Documentation : Une documentation technique destinée au développeur et éventuellement au commanditaire décrivant l'architecture et les choix de design. Une documentation utilisateur décrivant l'utilisation de la plateforme

(e) Utilisation et utilisabilité

- Ergonomie : L'interface doit avoir un design clair pour les utilisateurs

- Accessibilité : L'interface doit être responsive.
- Les résultats : L'utilisateur doit avoir un retour visuel clair de ses actions sur l'interface web

(f) Maintenance et portabilité

- Le code doit être modulaire pour être facilement mis à jour ou remplacer

(g) Question non résolues ou reporté à plus tard

- Y aura-t-il la possibilité de stopper une tâche ?
- Y aura-t-il une limite de taille de fichier ?

chapitre 6 Recours humain, questions juridiques, politiques, organisationnelles.

(a) Quel est le recours humains au fonctionnement du système ?

- Développeurs chargés du développement et des mises à jour des modules.

(b) Quelles sont les exigences juridiques et politiques ?

- La plateforme doit respecter les lois en vigueur dans l'Union Européenne et la France, telle que le RGPD en matière de protection des données.

(c) Quelles sont les conséquences humaines de la réalisation du système ?

Aucune.

(d) Quels sont les besoins en formation ?

- Formation des développeurs de niveau Bac +3, avec connaissance en base de données, HTML, CSS, PHP, services réseaux et connaissance MPI niveau bac +5.
- Se former sur calcul parallèle et distribué
- Se renseigner sur MonteCarlo