Rapport de Projet : Système de Calcul Distribué avec RabbitMQ

Introduction

Dans ce projet, j'ai implémenté un système de calcul distribué utilisant RabbitMQ pour effectuer des opérations mathématiques de manière distribuée. Ce système répond aux besoins de l'institut de physique nucléaire NGI qui souhaite mettre en place un système de calcul distribué pour effectuer des opérations mathématiques complexes.

Architecture du Système

Le système est composé de plusieurs composants :

1. **Client d'envoi** : Génère des opérations mathématiques aléatoires et les envoie à RabbitMQ
2. **Workers** : Quatre workers spécialisés (add, sub, mul, div) qui effectuent les calculs
3. **Client de résultats** : Affiche les résultats des calculs effectués

Implémentation

Technologies Utilisées

- Node.js v22
- RabbitMQ (via Docker)
- AMQP (amqplib)
- Docker et Docker Compose

Points Forts de l'Implémentation

1. **Architecture Distribuée** :
    - Utilisation de RabbitMQ comme message broker
    - Workers spécialisés pour chaque type d'opération
    - Support de l'opération "all" qui distribue le calcul à tous les workers
2. **Robustesse** :
    - Gestion des erreurs (division par zéro, etc.)
    - Files d'attente durables
    - Reconnexion automatique en cas de perte de connexion
3. **Simulation de Calculs Complexes** :
    - Délais aléatoires (5-15 secondes) pour simuler des calculs complexes
    - Distribution de charge entre les workers

Améliorations Réalisées

1. **Support Multi-opérations** :

- Implémentation des 4 types d'opérations (add, sub, mul, div)
- Support de l'opération "all"
- Génération aléatoire des opérations

2. **Gestion des Résultats** :

    - Affichage en temps réel des résultats
    - Format JSON structuré pour les résultats

Difficultés Rencontrées

1. **Configuration de RabbitMQ** :

    - Mise en place initiale de l'environnement Docker
    - Configuration des files d'attente durables

2. **Gestion des Connexions** :

    - Gestion des déconnexions/reconnexions
    - Persistance des messages

Conclusion

Ce projet m'a permis de mettre en pratique les concepts de systèmes distribués et de message queuing. L'utilisation de RabbitMQ a été particulièrement intéressante pour comprendre comment implémenter un système de calcul distribué robuste et scalable.

Perspectives d'Amélioration

1. **Interface Utilisateur** :

    - Développement d'une interface web pour l'envoi des opérations
    - Dashboard de monitoring des workers

2. **Monitoring** :

    - Ajout de métriques de performance
    - Logging plus détaillé

3. **Tests** :

    - Implémentation de tests unitaires
    - Tests de charge et de performance