Rapport de projet de LP25

CORNET Nicolas CHIEM Romain HAJJAR Mathieu
Automne 2024

Contexte du projet

Le projet consiste à construire une solution de sauvegarde incrémentale d'un répertoire source (le répertoire à sauvegarder) vers un répertoire cible (le répertoire de sauvegarde). Il s'agit de développer un programme en langage C pour créer un outil de sauvegarde inspiré de *Borg Backup*, avec un accent sur la déduplication des données et la possibilité de réaliser des sauvegardes sur un serveur distant via des sockets.

État des lieux des compétences du groupe

Nous avons tous les trois des compétences similaires au début de ce projet avec une même familiarité du langage C. Même à l'aide des compétences acquises tout au long de l'année, ce projet nous a poussé à apprendre à utiliser de nouvelles librairies. Nous nous attendions a de potentielles difficultés vis-à-vis de la taille du projet.

Répartition des tâches : planification

En découvrant le sujet, nous avons constaté que le projet avait plusieurs parties presques indépendantes les unes des autres. Ces différentes parties correspondent aux fichiers .c/.h.

Nous avions prévu de nous répartir les tâches comme ceci :

- Romain : backup_manager
- Nicolas : déduplication
- Mathieu : file_handler et main

Répartition des tâches : réalisation

Nous avons globalement réussi à réaliser le travail planifié, même si nous avons dû revenir sur certaines parties au fur et à mesure pour répondre aux attentes du projet.
La répartition finale des taches a été :

- Romain : backup_manager ,déduplication ,file_handler et rapport
- Nicolas : déduplication (première itération), network et rapport
- Mathieu : file_handler et main

Difficultés rencontrées

Compréhension des objectifs

Lorsque nous avons examiné le sujet, nous avons rencontré des difficultés à comprendre exactement de quoi les fichiers dédupliqués étaient constitués. Nous avons aussi rencontré des difficultés à comprendre et à incorporer le log dans les sauvegardes. Notre vision de la chose étant pour l'utilisateur de pouvoir faire plusieurs sauvegardes et de pouvoir restore depuis n'importe quelle sauvegarde. La logique lors de la restauration étant tout d'abord de restaurer l'entièreté de la toute première backup entière (copié par hard link lors de la backup original) et ensuite d'utiliser le log représentant l'ensemble des changements appliqué à cette backup pour reconstituer une sauvegarde précise. Le fichier log, unique, comporte ainsi l'ensemble des modifications apportés à la première sauvegarde au fil des backup.
Ainsi lorsque l'on veut restore à une date précise, il suffit de copier la première sauvegarde et de suivre chaque instruction du log jusqu'à la date précise pour obtenir le dossier.

Ce choix d'implémentation et de direction du projet permet alors de favoriser les projets nécessitant de modifier souvent et peu les fichiers avec en plus une grande répétition de chunks. Chaque nouvelle backup aura alors un poids quasi négligeable et c'est uniquement lors de l'ajout de nouveau chunks, ou de nouvelles données que la taille des backups augmente.

Finalement, dans le dossier de sauvegarde, il y a la première backup original constitué de hard link donc sans poids, les dossiers de toutes les backups, ces dossiers ne contenant que des fichiers dédupliqués sans poids, le log retraçant les modifications avec un poids négligeable et les tables de chunk contenant les données réels, uniques fichiers ayant un poids.

Réalisation des objectifs

Décrire pour chaque difficulté rencontrée dans l'implémentation :

- Les « symptômes » de cette difficulté (par ex. : des résultats incohérents, des données perdues dans les communications, etc.)
- Comment le problème a été résolu (architecture du code dans la fonctionnalité concernée par exemple)
- Si le problème aurait pu être anticipé.
- Ce qui aurait pu faciliter la résolution/éviter le problème

Dans la réalisation, la full backup et la génération des logs sont bien implémenté, la sauvegarde incrémentale également même s'il existe toujours des bugs dans la gestion des suppressions des fichiers dans les sous dossiers.

J'ai aussi rencontré des bugs dont j'ai abandonné la résolution (des sauts de lignes hasardeux illogiques dans le log notamment).

Dans le debug, il y a souvent eu des problèmes de path passé en paramètre, de boucle infinie.

Cela a été fastidieux car je ne suis pas familier avec le développement sur linux et j'ai donc eu du mal, j'aurai pu chercher et utilisé des outils de débug, mais j'ai surtout utilisé des messages de débug printf directement dans le code. J'aurais dû anticiper cela et me familiariser avec le développement sur linux.

Proposition d'une amélioration

Les hashs n'ont pas été utilisés dans notre code, cela aurait facilité le parcours des tables de chunks.

Le principe des tables de chunks globales n'est pas adapté à des dossiers volumineux ayant peu de chunk répétitif.

Le code aurait pu être plus modulaire. Le jeu des noms de fichier et path ont rendu tout cela assez compliqué.

Sur le plan personnel, il est clair que je n'ai pas utilisé assez d'outils externes pour me faciliter la tâche comme un débugger.

Analyse des résultats

Proposer votre analyse des résultats

Les résultats sur les backup sont plutôt satisfaisant, les fichiers sont bien dédupliqués et sauvegardés en chunk, la structure est gardé et le fichier log est bien construit.

Retour d'expérience

Le déroulement de ce projet nous a permis d'identifier plusieurs points critiques où des ajustements auraient pu être faits.

Une partie des problèmes a été sur la déduplication et gestion des logs. Cela a entrainé du retard dans l'implémentation des fonctionnalités liées aux sauvegardes incrémentales, entraînant un décalage sur les étapes suivantes. Cela est dû à une incompréhension initiale de la structure des fichiers dédupliqués et de leur interaction avec le fichier log. La documentation fournie est sur de nombreux points incomplète sur des aspects techniques du sujet. Nous aurions pu réaliser un prototype minimal pour confirmer nos hypothèses avant de se lancer sur une implémentation complète. Nous avons donc appris qu'il est important d'investir davantage de temps en phase de préparation pour comprendre pleinement les exigences fonctionnelles et anticiper les zones d'ambiguïté.

L’autre partie des problèmes est apparu sous la forme de la coordination au sein de l’équipe. Cela a mené à une perte de temps lors de l'intégration des différents modules, nécessitant des ajustements et corrections de dernière minute.

Faible synchronisation entre les membres de l’équipe pendant les phases intermédiaires.

Documentation insuffisante sur les interfaces entre les modules.

Solutions rétrospectives :

Mettre en place un plan d’intégration progressive dès le début du projet, avec des points réguliers pour tester l’interopérabilité des modules.

Adopter des outils de gestion de version (Git) avec une organisation stricte des branches pour éviter les conflits.

Enseignement pour l’avenir :

Prioriser une communication régulière entre les membres et documenter rigoureusement les interfaces et protocoles utilisés.

Mettre en œuvre une intégration continue pour détecter les problèmes dès qu’ils surviennent.

Conclusion

Le projet de création d'une solution de sauvegarde incrémentale a représenté un défi significatif tant sur le plan technique qu'organisationnel. En travaillant sur ce projet, nous avons non seulement approfondi nos connaissances en langage C et en manipulation des structures de données, mais nous avons également développé des compétences essentielles en gestion de projet et en résolution de problèmes.

La répartition initiale des tâches nous a permis de travailler en parallèle sur des fonctionnalités indépendantes, mais la collaboration et les révisions communes ont été nécessaires pour garantir la cohérence globale du programme. Malgré les imprévus et les ajustements au cours de la réalisation, nous avons réussi à atteindre les objectifs principaux.

Ce projet nous a sensibilisés à l'importance de bien comprendre les exigences dès le départ et d'adopter une planification réaliste. Nous avons également appris que les solutions techniques choisies doivent être adaptées à la problématique spécifique, comme l'ont montré nos réflexions sur l'utilisation des tables de chunks ou l'optimisation du système de log.

Enfin, cette expérience nous a permis de mieux appréhender les complexités inhérentes aux projets d'ingénierie logicielle, renforçant ainsi notre capacité à concevoir des solutions robustes et adaptées aux besoins des utilisateurs. Si nous devions recommencer ce projet avec nos connaissances actuelles, nous serions mieux préparés à anticiper les difficultés, à optimiser nos choix technologiques et à améliorer l'efficacité globale du processus.