

SAUVEGARDE COOPÉRATIVE POUR DISPOSITIFS MOBILES

Ludovic Courtès*

Directeur de thèse : David Powell

Co-directeur de thèse : Marc-Olivier Killijian

Laboratoire d'accueil :

Laboratoire d'Analyse et d'Architecture du CNRS
7, avenue du Colonel Roche
31077 Toulouse CEDEX 4

Établissement d'inscription :

Institut National Polytechnique de Toulouse
6 allée Émile Monso - ZAC du Palays
BP 34038
31029 Toulouse CEDEX 4

Résumé

Nous présentons les fonctionnalités d'un système de sauvegarde coopérative pour dispositifs mobiles. Ce système repose sur la collaboration entre dispositifs pour assurer la sauvegarde et le recouvrement des données de chaque dispositif. Nous identifions les défis qui devront être relevés, en particulier dans le domaine du stockage réparti. Nous proposons une analyse de différents algorithmes de dissémination de données. Enfin, nous concluons sur nos premiers résultats et les axes de recherche à explorer.

Mots-clés

tolérance aux fautes, sauvegarde, mobilité, réseaux ad hoc, pair-à-pair

1 INTRODUCTION

Nous abordons la problématique liée à la conception d'un outil fournissant des mécanismes de sûreté de fonctionnement à des dispositifs mobiles dotés de moyens de communication sans fil. Nous considérons des dispositifs mobiles (ordinateurs portables, assistants personnels, etc.) n'ayant accès à une infrastructure fixe de communication (un réseau local ou Internet) que par intermittence. Ces dispositifs mobiles doivent être capables de communiquer entre eux, lorsqu'ils sont à proximité physique, en utilisant des moyens de communication sans fil. Cependant, nous souhaitons que l'outil développé, MoSAIC¹ soit utile à une large palette de systèmes, allant aussi bien de dispositifs très mobiles n'ayant que rarement accès à Internet, à l'autre extrême représenté par des machines connectées en permanence à Internet et peu ou pas mobiles. En outre, nous faisons l'hypothèse que les participants au service n'ont aucune relation de confiance au préalable.

MoSAIC a pour objectif de permettre aux dispositifs sur lesquels il s'exécute de tolérer les fautes logicielles ou matérielles pouvant entraîner la perte de données. Ces fautes sont le plus souvent permanentes : perte ou vol du dispositif mobile, effacement accidentel de données par l'utilisateur. À l'heure actuelle, les utilisateurs de systèmes mobiles effectuent le plus souvent la sauvegarde de leurs données uniquement lorsqu'ils ont accès à leur machine de bureau. Entre temps, il serait théoriquement possible d'utiliser l'accès à une infrastructure (par exemple UMTS ou GPRS) pour ce faire mais cette solution est très rarement utilisée, pour des raisons pratiques ou de coût.

* <ludovic.courtes@laas.fr>

¹Mobile System Availability, Integrity and Confidentiality, <http://www.laas.fr/mosaic/>.

En revanche, nous pensons que l'avènement de dispositifs mobiles équipés de moyens de communication sans fil de courte portée offre des possibilités d'interactions *entre pairs* dont pourrait profiter un système de sauvegarde *coopératif*. En effet, l'utilisation répandue de systèmes mobiles communicant va permettre des interactions fréquentes mais de courte durée. Avec MoSAIC nous souhaitons tirer profit de ces interactions : à chaque rencontre de deux systèmes mobiles, le système de sauvegarde va automatiquement initier une demande de sauvegarde pour une partie de ses données, de manière *transparente*.

Nous présentons dans la section suivante les objectifs que devra atteindre MoSAIC, notamment en termes de sûreté de fonctionnement, en fonction des problèmes nouveaux qu'il pose. La section 3 présente nos travaux actuels portant sur l'évaluation analytique de la disponibilité des données obtenue grâce au service de sauvegarde. Enfin, dans la section 4, nous résumons nos résultats et présentons nos perspectives de recherche.

2 OBJECTIFS ET PROBLÉMATIQUES

Dans cette section, nous présentons les problèmes que nous souhaitons résoudre avec notre service de sauvegarde coopérative, en termes de sûreté de fonctionnement des dispositifs mobiles. Nous décrivons alors les fonctions fournies par le service. Enfin, nous montrons les nouveaux défis de sûreté de fonctionnement que doit résoudre ce service *coopératif*.

2.1 TOLÉRANCE AUX FAUTES DES DISPOSITIFS MOBILES

Les dispositifs mobiles, de par leur utilisation, sont sujets à des fautes permanentes (perte, vol, casse) et à des fautes transitoires (effacement accidentel, corruption). L'occurrence de ces fautes peut entraîner une perte des données stockées sur le dispositif. Or de tels dispositifs sont de plus en plus utilisés pour la saisie ou la capture de données nouvelles, dans un contexte où il est difficile voire impossible de réaliser des sauvegardes avec une fréquence raisonnable. Ce sont donc ces fautes que nous souhaitons pouvoir tolérer par le développement d'un service de sauvegarde.

L'objectif premier du service est donc d'améliorer la *disponibilité* des données stockées sur ces dispositifs mobiles. Chaque dispositif mobile peut sauvegarder ses données sur les dispositifs qui l'entourent, grâce à des moyens de communication sans fil, et de manière *opportuniste* (rôle d'utilisateur, *propriétaire* de données). En contrepartie, chaque dispositif doit également dédier un certain nombre de ses ressources de stockage au service pour que d'autres puissent en profiter de la même manière (rôle de *contributeur*). Nous faisons l'hypothèse que, dès qu'un contributeur a accès à Internet (c'est-à-dire dans une situation où cet accès lui est peu coûteux), il transmet les données qu'il a acquises à leurs propriétaires. Ces derniers pourront alors les restaurer le cas échéant [3].

Il s'agit donc d'une approche semblable à celle des réseaux de partage de données *pair-à-pair* largement utilisés aujourd'hui sur Internet. Nous allons voir que cette approche, en particulier dans l'environnement mobile, lance des défis en termes de sûreté de fonctionnement.

2.2 DÉFIS

Nous faisons l'hypothèse que les participants à un tel service de sauvegarde n'ont *a priori* aucune relation de confiance entre eux. Par conséquent, nous devons prendre en compte la possibilité que des participants soient *malveillants* et cherchent à causer du tort à des participants individuels voire au service dans son ensemble. Par conséquent, un tel service doit pouvoir garantir la confidentialité et l'intégrité des données des utilisateurs, et doit aussi se prémunir contre les attaques en déni de services pouvant porter atteinte à son fonctionnement (rétention de données, inondation, égoïsme, etc.).

D'autres défis touchent au stockage et à la restauration des données : la dissémination des données sur des contributeurs multiples ainsi que le fait que les rencontres de contributeurs soient *imprévisibles* et *éphémères*. Enfin, le fait que les contributeurs soient potentiellement difficiles d'accès (par connexion directe ou *via* Internet) et suspects implique, de la part des propriétaires, de stocker les données avec un niveau de redondance approprié. Toutefois, compte-tenu du coût énergétique des communications sur de tels dispositifs, il est nécessaire de limiter autant que possible la quantité de données à échanger.

Dans cet article, nous nous focalisons sur les aspects ayant trait au stockage en nous intéressant à la conception d'un algorithme de dissémination des données.

2.3 SOLUTIONS

Le fait que les rencontres entre participants soient imprévisibles et potentiellement éphémères rend nécessaire la *fragmentation* des données à sauvegarder. En outre, ces fragments de données seront nécessairement disséminés, comme nous l'avons vu, avant, finalement, d'être rendus accessibles à leur propriétaire *via* Internet [3]. La dissémination peut par ailleurs être vue comme bénéfique du point de vue de la confidentialité des données [4].

Le niveau de redondance souhaité peut s'obtenir en stockant chaque fragment sur un certain nombre de contributeurs différents. En stockant n fois un fragment donné, il est alors possible de tolérer la défaillance (ou malveillance) de $n - 1$ contributeurs. Cependant, pour un nombre de défaillances tolérées donné, il est possible de réduire la quantité de stockage nécessaire grâce à l'utilisation de *codes d'effacement* [11]. Schématiquement, il s'agit d'algorithmes qui, à partir d'une donnée d'entrée de taille k produisent $n > k$ fragments parmi lesquels n'importe quels k fragments suffisent pour reconstituer la donnée d'origine¹. On tolère alors $n - k$ défaillances pour un coût de stockage de $\frac{n}{k}$. La réplication de fragments entiers peut donc être vue comme un cas particulier où $k = 1$. En pratique, les valeurs de k et n seront dépendantes de l'algorithme choisi (voir annexe).

Il faut cependant noter que le bénéfice en termes de disponibilité apporté par les codes d'effacement, pour un nombre de défaillances tolérées données, est fortement dépendant de la disponibilité des nœuds constituant le support de stockage [7]. En deçà d'une certaine disponibilité, il est préférable de faire appel à de la simple duplication. Par conséquent, il pourra être nécessaire de procéder à une adaptation dynamique du codage des copies en fonction de la disponibilité des contributeurs [1].

Plusieurs choix sont donc possibles pour paramétrer la fragmentation des données et plusieurs algorithmes de dissémination sont envisageables. Dans la section suivante, nous présentons différentes politiques possibles et proposons une méthode pour évaluer l'impact de ces algorithmes sur la disponibilité des données.

3 ÉVALUATION D'ALGORITHMES DE DISSÉMINATION DES DONNÉES

La section précédente a montré différents paramètres pouvant être pris en compte pour la dissémination et la redondance des données à sauvegarder. Dans cette section, nous présentons l'évaluation que nous souhaitons faire des algorithmes de dissémination envisageables ainsi que la méthodologie que nous souhaitons employer.

3.1 OBJECTIFS

S'agissant d'un service de sauvegarde, le principal critère d'évaluation sera bien entendu l'impact de l'algorithme sur la disponibilité des données à sauvegarder. Un *modèle* général du processus de sauvegarde doit donc être défini. Afin de simplifier ce modèle, on supposera que

¹Cette description s'applique aux codes d'effacement *optimaux*.

dès lors que l'un des intervenants (propriétaire ou contributeur) a accès à Internet, alors les données sont « mises en sécurité ». En outre, nous considérons que dès lors qu'un propriétaire rencontre un contributeur il peut lui demander de stocker ses données. Ce modèle est composé de trois processus stochastiques Poissoniens à taux constants :

- le processus de rencontre d'un contributeur par le propriétaire de données, permettant à ce dernier de stocker une partie de ses données; ce processus a pour taux α ;
- le processus de connexion à Internet de chaque participant, de taux β ;
- le processus de défaillance des participants, de taux λ ; dans un premier temps, on fait donc abstraction des malveillances en les assimilant à des défaillances aléatoires.

À ce modèle nous ajoutons l'hypothèse suivante : un propriétaire de données n'est pas notifié de la défaillance de contributeurs disposant de ses données. C'est le cas le plus vraisemblable compte-tenu de la connectivité observée en environnement mobile. Par conséquent, nous considérons qu'un propriétaire de données ne pourra pas adapter sa politique de réplication et dissémination en fonction de la défaillance de ses contributeurs.

Ce modèle pourra bien sûr être raffiné par la suite. Pour chaque algorithme de dissémination donné, il pourra nous servir à évaluer analytiquement, en fonction de ces paramètres, les informations suivantes :

- la probabilité asymptotique que les données soient mises en sécurité (c'est-à-dire qu'elles atteignent Internet);
- le temps nécessaire aux données pour être rendues sûres (c'est-à-dire pour atteindre Internet);
- la disponibilité des données créées sur un dispositif mobile participant en fonction du temps.

Nous envisageons également d'utiliser ce modèle pour comparer différents algorithmes de dissémination en fonction de ces critères.

3.2 MÉTHODOLOGIE

Le processus de sauvegarde des données, dans le cadre du modèle simplificateur que nous venons d'évoquer, peut être modélisé sous la forme d'un graphe de Markov, c'est-à-dire un automate à états finis dont les branches sont étiquetées par les paramètres α , β et λ .

Supposons un algorithme de dissémination simple donnant une copie complète des données à chaque contributeur rencontré et se donnant pour objectif d'en distribuer N . La figure 1 représente les différents scénarios possibles dans l'exécution de cet algorithme, pour $N = 2$. On remarque deux états puits correspondant à la perte définitive des données (tous les dispositifs en disposant ont défailli) et à la « mise en lieu sûr » des données (un des dispositifs a accédé à Internet). Dans l'état initial noté 1/2, un seul exemplaire des données est disponibles (celui du propriétaire) et 2 copies restent donc à faire.

Dans l'état 3/0, trois exemplaires sont disponibles (dont celui du propriétaire) et il ne reste plus aucune copie à faire. À partir de l'état 2/1, il suffit qu'un dispositif parmi les 2 disposant des données accède à Internet pour passer dans l'état « sûr » (d'où le taux $2 \times \beta$); si le propriétaire défaille, alors on passe dans l'état 1/0 signifiant qu'un exemplaire est toujours disponible mais qu'aucune nouvelle copie ne sera faite à l'avenir; enfin, si le contributeur défaille, alors le propriétaire n'augmentera pas pour autant son nombre de copies souhaité, d'où le passage dans l'état 1/1.

Il s'agit là d'un modèle stochastique que nous pouvons généraliser à d'autres nombres de copies, à d'autres politiques, mais aussi à l'utilisation de codes d'effacement. La reformulation de ce modèle sous forme d'un réseau de Petri stochastique doit permettre de généraliser la modélisation à différentes politiques et paramètres pour le protocole de sauvegarde [2].

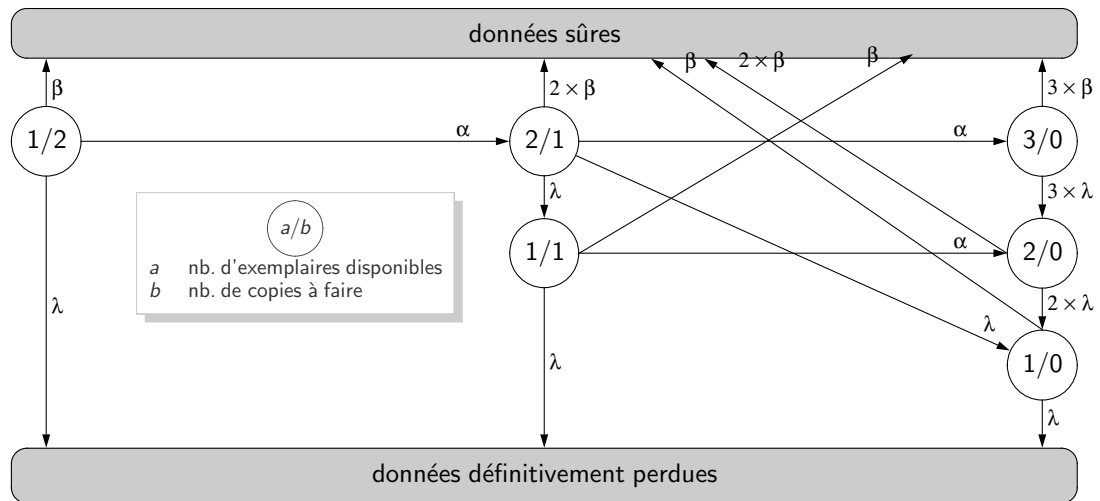


Fig. 1. Algorithme de dissémination réalisant 2 copies complètes des données originales.

4 CONCLUSION

Nous avons présenté le cadre dans lequel s'inscrit notre thèse, la sauvegarde coopérative pour dispositifs mobiles, ainsi que les défis nouveaux à relever dans ce contexte. Nous avons présenté, en particulier, le contour architectural du service que nous envisageons.

Nos travaux actuels portent sur l'évaluation analytique de la disponibilité des données offerte par un tel service de sauvegarde. Notre objectif est d'évaluer, pour des taux donnés de rencontre entre participants, d'accès à Internet, et de défaillance, la disponibilité atteinte en fonction du temps. Cette évaluation devrait nous permettre de considérer différentes politiques de dissémination des données.

RÉFÉRENCES

- [1] R. BHAGWAN, K. TATI, Y-C. CHENG, S. SAVAGE, G. M. VOELKER. Total Recall : System Support for Automated Availability Management. *Proc. of the ACM/USENIX Symp. on Networked Systems Design and Implementation*, 2004.
- [2] C. BÉOUNES, M. AGUÉRA, J. ARLAT, K. KANOUN, J-C. LAPRIE, S. METGE, S. BACHMANN, C. BOURDEAU., J-E. DOUCET, D. POWELL, P. SPIESSER. SURF-2 : A Program for Dependability Evaluation of Complex Hardware and Software Systems. *Proc. of the Twenty-Third IEEE Annual Int. Symp. on Fault-Tolerant Computing*, pages 668–673, 1993.
- [3] L. COURTÈS, M-O. KILLIJIAN, D. POWELL, M. ROY. Sauvegarde coopérative entre pairs pour dispositifs mobiles. *Actes des deuxièmes journées francophones Mobilité et Ubiquité (UbiMob)*, pages 97–104, 2005.
- [4] Y. DESWARTE, L. BLAIN, J-C. FABRE. Intrusion Tolerance in Distributed Computing Systems. *Proc. of the IEEE Symp. on Research in Security and Privacy*, pages 110–121, 1991.
- [5] C. HUANG, L. XU. STAR : An Efficient Coding Scheme for Correcting Triple Storage Node Failures. *Proc. of the Fourth USENIX FAST*, pages 197–210, 2005.
- [6] R. KATTI, X. RUAN. S-Code : New Distance-3 MDS Array Codes. *IEEE Int. Symp. on Circuits and Systems*, 2005.
- [7] W. K. LIN, D. M. CHIU, Y. B. LEE. Erasure Code Replication Revisited. *Proc. of the Fourth P2P*, pages 90–97, 2004.

- [8] M. G. LUBY, M. MITZENMACHER, M. A. SHOKROLLAHI, D. A. SPIELMAN. Efficient Erasure Correcting Codes. *IEEE Transactions on Information Theory*, 47(2), February 2001, pages 569–584.
- [9] P. MAYMOUNKOV. Online Codes. TR2002-833, Secure Computer Systems Group, New York University, NY, USA, November 2002.
- [10] M. MITZENMACHER. Digital Fountains : A Survey and Look Forward. *Proc. of the IEEE Information Theory Workshop*, pages 271–276, 2004.
- [11] L. XU. Hydra : A Platform for Survivable and Secure Data Storage Systems. *Proc. of the ACM Workshop on Storage Security and Survivability*, pages 108–114, 2005.
- [12] L. XU, V. BOHOSSIAN, J. BRUCK, D. G. WAGNER. Low Density MDS Codes and Factors of Complete Graphs. *IEEE Transactions on Information Theory*, 45(1), November 1999, pages 1817–1826.

ANNEXE : ALGORITHMES DE CODES D'EFFACEMENT

On utilise la notation (n,k) pour désigner un code d'effacement qui, à partir d'une donnée partitionnée en k fragments, produit n fragments, avec $n > k$. Un code d'effacement est dit *optimal* si le nombre de fragments nécessaire pour recouvrer la donnée d'origine est égal à k ; on dit qu'il est quasi-optimal si il suffit de $k + \epsilon$ fragments, où ϵ est petit par rapport à k [12]. On appelle *taux* d'un code le rapport entre le nombre de fragments nécessaires pour reconstituer la donnée initiale et le nombre n de fragments produits. On appelle *distance* d'un code la différence entre le nombre de fragments produits n et le nombre de fragments nécessaires, plus 1. Par exemple, dans le cas d'un code optimal, la distance est $d = n - k + 1$ [5].

En pratique, un certain nombre de codes d'effacement sont prévus pour un taux fixe. On notera par exemple les codes optimaux suivants :

Nom	Distance	Taux
B-Code [12]	3	$\frac{k}{k+2}$
STAR [5]	4	$\frac{k}{k+3}$
S-Code [6]	3	$\frac{k}{k+2}$
Tornado [8]	au choix, choisi <i>a priori</i>	

Comme on le voit, les codes Tornado définissent tout une famille de codes dont le taux doit être choisi avant utilisation. Par ailleurs, il existe des codes dits « sans taux » capables de produire potentiellement une infinité de fragments codés parmi lesquels seuls $k + \epsilon$ suffisent pour restaurer la donnée d'origine [9,10].