Cohérences faibles pour le cloud zero-trust

SUJET DE RECHERCHE

Emmanuel Godard (LIS) – Corentin Travers (LIS)
emmanuel.godard@lis-lab.fr et corentin.travers@lis-lab.fr

Mots-clefs : Cloud, Sécurité par conception, Structures et algorithmes distribués, Cohérences faibles, Systèmes byzantins

Résumé

Les applications collaboratives en temps réel sont de plus en plus utilisées dans le cadre de la mise en place de systèmes de travail à distance. Ces applications sont souvent basées sur des architectures client-serveur centralisées, ce qui pose des problèmes de sécurité et de confidentialité. Les données sont stockées sur un serveur centralisé, ce qui implique que les utilisateurs doivent faire confiance à un tiers pour la gestion de leurs données. De plus, ces architectures sont souvent vulnérables aux attaques par déni de service, et ne permettent pas de garantir la confidentialité des données.

Pour répondre à ces problématiques, nous proposons d'explorer des solutions d'échange de l'information basées sur des architectures sans tiers de confiances à travers des approches dites zero-trust et/ou pair à pair. Ces solutions nous permettraient de proposer de solutions à haut niveau de sécurité tout en garantissant une certaine résilience du système. Pour conserver des performances fortes notamment en haute disponibilité, les cohérences faibles sont fréquemment utilisées.

Dans ce contexte, nous proposons d'étudier les propriétés de cohérences faibles appliquée aux problématiques liées au cloud. Dans un premier temps sera réalisé un état de l'art sur les solutions byzantines sans primitives cryptographiques, ainsi que sur les différentes implémentations existantes (WP1). Une deuxième étape consistera à proposer des solutions plus efficaces mais utilisant des primitives cryptographiques (WP2). Enfin, une dernière étape consistera en la production d'une preuve de concept de solution de stockage clef/valeurs utilisant les algorithmes retenus aux étapes précédentes (WP3).

Problématique

Depuis les travaux pionniers des années 80, par Lamport [Lam86] et Misra [Mis86] notamment, la gestion de la réplication est au cœur des développements du numérique en terme de haute disponibilité. L'une des problématiques fondamentales est d'offrir aux développeurs d'applications une abstraction de la mémoire répliquée qui soit à la fois simple à utiliser et permette de mobiliser de manière souple et résistante aux défaillances l'intégralité des ressources distribuées. Cette voie de recherche a produit la notion de *cohérence des données* dont les nombreuses déclinaisons permette de s'adapter aux meilleurs compromis d'usage et spécificités de chaque application.

La tendance actuelle autour de la mise en Cloud des applications informatiques implique des modifications importantes dans les usages et les modes de développement des nouvelles applications. Dans le cadre de nouvelles facilités d'usage, où la maintenance de l'infrastructure est déléguée à un prestataire, cela a conduit à une centralisation des ressources. Cela ré-introduit des problématiques classiques en termes de sécurité : nécessité de confiance/souveraineté ou bien *point central de défaillance* (SPOF).

De nouvelles approches dites *sans-confiance* (zero-trust) ont donc été proposées pour continuer à utiliser ces ressources cloud sans dépendre d'un prestataire particulier. Elles nécessitent à la fois des architectures multi-fournisseurs et des approches cryptographiques avancées.

Du point de vue des programmeurs, il est souvent avantageux de considérer de telles applications sur le nuage comme un seul système centralisé. Cela nécessite que les structures de données utilisées aient une propriété dite de *cohérence forte*.

En conditions réelles, les serveurs peuvent avoir à supporter des conditions de fonctionnement très difficiles. Il est bien connu, à la fois des théoriciens et des praticiens, par le théorème CAP (Consistency, Availability, Partition tolerance) que des compromis de fonctionnement sont souvent nécessaires. En particulier, si c'est la cohérence forte qui est recherchée, le temps de calcul est proportionnel à la latence de **tout** le réseau. Ce qui diminue en pratique la disponibilité.

Si l'on se réfère au théorème CAP, en appliquant la cohérence forte il est impossible de mettre en place un système hautement résilient, tout en fournissant une application hautement disponible. Ces deux points pouvant néanmoins se retrouver être essentiels dans la réalisation d'une application collaborative.

L'approche pair-à-pair implique en effet une grande résistance du système face à la panne. Les répliques sont emmenées à se déconnecter les uns des autres et à avoir des différences de latences importantes et inégales. La non-maitrise du poste et de l'environnement d'exécution de l'application nous pousse à imaginer des systèmes pouvant résister aux pires situations possibles.

Dans le même temps, la nature de l'application recherchée, qui est la collaboration en temps réel, est liée à la question de la haute disponibilité. Le but étant de permettre à des répliques différentes d'accéder à la même donnée partagée pour un travail en temps réel. Il ne serait donc pas acceptable de proposer des temps de latences trop conséquents entre deux modifications.

Etant donnée l'impossibilité de satisfaire ces deux aspects nous nous tournons vers l'étude des cohérences faibles, et notamment de la convergence. On peut ainsi définir comme convergent les systèmes respectant la propriété suivante :

Si les répliques arrêtent de proposer des modifications, alors ces mêmes répliques doivent éventuellement atteindre un état cohérent.

La convergence (ou Eventual Consistency) est particulièrement étudiée. Ainsi un certains nombres de structures de données distribuées proposant de respecter la convergence ont vu le jour. Néan-

moins à elles seules, celles-ci ne permettent pas de résoudre notre problématique. En effet cette propriété n'offre pas de garantie sur les comportements durant l'exécution, là exactement où l'incohérence au sein du système est permise par la convergence. Or il ne suffit pas qu'un document converge à terme pour en faire une application d'édition collaborative satisfaisante. Mais il faut aussi proposer des mécanismes pour résoudre les conflits, qui sont inévitables dans l'approche collaborative. Cette résolution doit être réalisée de la manière la plus optimale pour maximiser la préservation du sens donné à chaque modification par la réplique qui l'a émise.

Ces questions ont bien entendu été très étudiées et les différentes solutions proposées particulièrement adaptées dans notre contexte sont les *types des données répliqués* (ou Replicated Data Type). Il en existe deux classes, les types de données répliquées commutatives (CmRDT), dont les opérations donnent le même résultat, peu importe leurs ordres d'exécutions locales. Et les structures de données convergentes (CvRDT), par exemple un système où la donnée viserait à croitre continuellement convergeant ainsi vers une structure maximale. Ces deux classes sont regroupées sous la dénomination de type de données sans conflit (CRDT) et sont en réalité équivalentes l'une à l'autre [SPBZ11].

En outre, pour proposer des solutions véritablement sécurisées dans un contexte zéro-trust, les conditions de fonctionnement les plus difficiles à considérer sont lorsque des serveurs ou des clients participants ont été compromis et ne respectent pas strictement le protocole. Dans la littérature, cela s'appelle un fonctionnement byzantin.

Etant données ces contraintes difficiles de disponibilité et de sécurité, assurer une propriété de cohérence forte peut être très coûteux en calcul et en temps. Les exigences applicatives ne sont parfois pas compatibles avec de telles conditions de fonctionnement. On peut alors considérer des données avec des propriétés dites de *cohérences faibles*.

État de l'art

Le paysage des propriétés de *cohérences faibles* est relativement complexe. On peut distinguer trois grandes familles de cohérences faibles [Ray18], [Per17] :

— la sérialisabilité
— la cohérence causale
— la cohérence éventuellement forte

Si la cohérence éventuellement forte est en général recherchée pour les applications collaboratives, elle est particulièrement coûteuse. La sérialisabilité est plus simple à implémenter mais produit parfois des transactions qui ne terminent pas. Ces situations d'erreur doivent alors être gérées par l'application. La cohérence causale maintient l'ordre causal perçu par chaque processus et permet en général d'implémenter des structures de données de plus haut niveau de manière efficace. Le lecteur pourra se référer à la cartographie assez exhaustive de M. Perrin [Per17].

Résultats Algorithmiques

Les premiers travaux sur des outils collaboratifs sécurisés dans un contexte de haute disponibilité datent de 2009, cependant les recherches plus systématiques concernant la sécurité des cohérences dites faibles sont en fait très récentes. En 2009, Sing *et al.* propose le système Zeno qui est le premier à proposer un algorithme byzantin qui privilégie la disponibilité sur la cohérence (forte). Il offre une robustesse byzantine à la cohérence éventuellement forte [SFK+09]. L'algorithme montre

de manière expérimentale de meilleures performances de disponibilité que les algorithmes byzantins classiques.

Il existe actuellement essentiellement des études et solutions partielles pour la cohérence causale [TWZP19] et [VDLLP20]. Tseng *et al.* présentent des bornes exactes de calculabilité dans un cadre byzantin d'un côté et donnent un algorithme dont les performances sont comparées avec ceux de la plateforme Google Compute. Van Der Linde *et al.* présentent un système pair-à-pair résistant aux attaques byzantines qui offre des garanties de cohérence causale. Leur évaluation considère que malgré une architecture pair-à-pair, les performances, notamment en termes de latence sont très bonnes en comparaison avec une architecture client-serveur classique.

En complément de ces algorithmes, Misra et Kshemkalyani ont montré dans [MK23] que dans un contexte asynchrone, il n'est pas possible de proposer de la consistance causale même avec un seul participant byzantin.

L'une des particularités de [VDLLP20] est de proposer également une réflexion sur les défaillances byzantines dans un contexte de cohérences faibles. Un système pair-à-pair tel que celui de [MK23] justifie de proposer de nouvelles attaques où un participant exploite les informations des couches basses de réplication pour créer des attaques au niveau applicatif.

L'application de critères de cohérences faibles ne suffit pas à satisfaire le cadre de notre problématique. Le contexte du cloud pose notamment de grande questions en termes de centralisation et de gouvernance des données, avec un marché dominé par quelques acteurs majeurs auxquels les utilisateurs doivent faire confiance de manière aveugle. Posant ainsi de grande question sur la confidentialité et la souveraineté de leurs informations.

C'est dans ce contexte qu'intégrer la notion d'un cloud zero-trust est essentiel en ancrant nos réflexions dans une approche pertinente d'un point de vue industriel et réglementaire. Le zero-trust comme défini par le NIST dans la SP 800-207 [RBMC20] est un modèle de sécurité qui ne fait confiance à aucun tiers, et qui ne fait aucune hypothèse sur la sécurité du réseau. Il permet ainsi de se préserver des comportements malveillants émis par les intermédiaires diminuant la surface d'attaque et limitant les comportements byzantins aux seuls clients qui eux ont accès aux données.

Evidement ce dernier point est aussi à considérer. C'est pourquoi une approche de sécurité centrée sur la donnée en plus des communications peut aussi être envisagé en adoptant des approches dites "Data Centric". C'est-à-dire de considérer la donnée elle-même comme un acteur vivant du système en lui attribuant des processus de contrôle d'accès et de suivie [Bay09]. Ces questions représentent des enjeux grandissants et sont considérés par les acteurs étatique et inter-étatique à l'image de l'OTAN qui statut sur ces problématiques à travers les STANAG 4774 et 4778. Ces questions sont largement étudiées depuis les années 2010 avec des travaux comme [GPSW06, MKE09] qui définisse des solutions pour mettre en place du chiffrement par attribut. Consistant à émettre des clés de chiffrements dépendantes de droits, et donc de permettre de définir des politiques de sécurité. Des travaux comme [YLWV17] propose des solutions plus adaptées au cloud en se basant sur des architectures plus flexibles et avec une plus grande granularité dans la définition des droits.

Néanmoins sur les aspects du zero-trust et de la sécurité centrée sur la donnée, il n'existe pas encore de travaux académiques concernant une formalisation consensuelle de ces notions. Et ces termes sont soumis à de nombreuses interprétations. Il reste donc à spécifier formellement ces différents termes pour comprendre quelles propriétés sont à satisfaire pour réaliser de la cohérence faible dans un contexte zero-trust.

Implémentations Existantes

Des projets actuels tentent d'implémenter des protocoles de cohérences faibles pour la mise en place d'applications collaboratives en temps réel. Parmi ces projets on peut citer yjs [Yjs23] qui implémente le protocole YATA [NJDK16] et qui permet d'assurer une convergence forte (ou SEC d'après le référentiel de Perrin) à travers un système de type CRDT. D'autres projets plus anciens tel qu'Etherpad utilise des solutions plus simples à base de résolution de conflit continue, assurant aussi une convergence forte mais réalisant des opérations algorithmiques plus complexes en termes de mémoire et de temps de calcul vis-à-vis des CRDTs [App11].

Objectifs

Les objectifs de cette thèse sont à la fois d'étudier les trois types de cohérence faible en situation byzantine et de définir des algorithmes byzantins efficaces pour pouvoir les implémenter. Puisque la cohérence causale est déjà bien étudiée, ce sont les deux autres cohérences qui seront les principaux axes de recherche de cette thèse.

La première étape (WP1) consistera à étudier des solutions byzantines sans primitives cryptographiques, ou avec des primitives raisonnablement coûteuses, c'est-à-dire notamment sans calcul homomorphe. Une étude des implémentations existantes sera réalisée pour notamment déterminer les garanties offertes par ces solutions dans le vocabulaire des cohérences faibles.

La deuxième étape (WP2) consistera à produire des solutions plus efficaces mais qui utilisent des primitives cryptographiques nécessitant des primitives de partage de secret avancées et/ou de calcul homomorphe.

Une dernière étape (WP3) consistera en la production d'une preuve de concept de solution de stockage *clef/valeurs* utilisant les algorithmes retenus aux étapes précédentes.

Méthodologie et Planning

Une revue précise des modèles de calcul distribué pour lesquels des solutions (principalement de consistance causale) ont été proposées sera établie dans le but de déterminer l'ensemble des hypothèses, théoriques et pratiques, de validité de ces solutions. En parallèle de cetté étude, en relation avec l'entreprise Scille, une liste d'attaques sur les architectures pair-à-pairs à cohérence faible sera établie. L'accent sera mis sur la production de connaissances nouvelles (nouvelles solutions par rapport à l'état de l'art mais également nouvelles attaques).

Les algorithmes seront tout d'abord validé de manière formelle avant de voir une preuve de concept développée.

Le WP1 se déroulera en 2024, le WP2 en 2025, et le WP3 en ZO26.

Modalités de Suivi et d'Échange

Le doctorant participe aux réunions hebdomadaires de suivi de l'entreprise Scille. Les partenaires se rencontreront tous les trois mois pour un point d'avancée sur les travaux.

Il participera également aux réunions physiques de l'entreprise tous les 6 mois.

Moyens Matériels

Le doctorant sera hébergé au Laboratoire d'Informatique et des Systèmes. Il bénéficiera de l'environnement scientifique et technique d'un laboratoire UMR CNRS de 800 personnes, dont environ 400 personnels permanents .

Du côté de l'entreprise Scille, qui fonctionne en *full remote*, le doctorant aura accès à un banc d'essai cloud hébergé par l'entreprise.

Retombées Attendues

Du côté du laboratoire LIS, les retombées attendues sont les publications scientifiques suivantes :

— état de l'art et synthèse concernant les consistances faibles byzantines
— propositions et preuves de nouveaux algorithmes dans le contexte zéro-trust

Du côté de l'entreprise Scille, il est attendu une mini-maquette de synchronisation et collaboration cloud, une preuve de concept des algorithmes sus-cités ainsi que du conseil et de l'expertise dans le domaine du "développement scientifique" des produits développés par Scille, notamment `parsec`.

Équipe

Équipe Algorithmique Distribuée (DALGO)

L'équipe Algorithmique Distribuée (responsable Arnaud Labourel) fait partie du Laboratoire d'Informatique et Systèmes (LIS CNRS UMR 7020). du Laboratoire d'Informatique et Systèmes (LIS CNRS UMR 7020). C'est une équipe de recherche reconnue au plus haut niveau international, avec 8 membres permanents dont les centres d'intérêt vont des algorithmes distribués fiables, de la confidentialité dans les systèmes distribués aux réseaux de communication, ainsi qu'aux algorithmes de graphes, aux agents mobiles et à l'IoT,

Encadrants

Emmanuel Godard est professeur à l'Université Aix-Marseille. Ses intérêts de recherche portent principalement sur la compréhension et la maximisation de la décentralisation (en un sens large) dans les systèmes distribués. Il est expert en algorithmique et calculabilité distribuées.

Corentin Travers est Maître de Conférences à l'Université Aix-Marseille. Ses intérêts de recherche portent sur les algorithmes distributés robustes et efficaces pour les systèmes à mémoire partagée ou les réseaux distribués. Il est expert en algorithmique et complexité distribuées.

Marcos Medrano est ingénieur R&D chez Scille. Diplômé d'un master de recherche en sciences de l'informatique et mathématique appliqué. Il est en charge de la stratégie de développement du produit Parsec et réalise le lien entre les ingénieurs et les intervenants académiques.

Choix du Candidat

L'équipe DALGO est partie prenante du Master "Fiabilité et Sécurité Informatique" de l'Université Aix-Marseille. Ce parcours de master est labellisé *SecNumEdu* par l'ANSSI. À l'automne 2022,

le sujet proposé avec l'entreprise Scille a été présenté à l'ensemble des étudiants de master. Suite à cet appel à candidature, M. Amaury Joly a été retenu pour un stage de recherche préliminaire de 6 mois sur le thème des consistances faibles au laboratoire LIS.

Les notes de M. Amaury Joly sont très bonnes, il obtient une mention bien au master. Il présente en outre un très bon double profil à la fois théorique et technique, sa motivation pour les activités de recherche en lien avec la sécurité du Cloud est très forte, il est le candidat parfait pour un tel sujet de recherche.

Références

[App11] AppJet. Etherpad and EasySync Technical Manual. https ://raw.githubusercontent.com/ether/etherpad-lite/master/doc/easysync/easysync-full-description.pdf, 2011.

[Bay09] Jennifer Bayuk. Data-centric security. *Computer Fraud & Security*, 2009(3) :7–11, March 2009.

[BGYZ14] Sebastian Burckhardt, Alexey Gotsman, Hongseok Yang, and Marek Zawirski. Replicated data types : Specification, verification, optimality. In *Proceedings of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 271–284, San Diego California USA, January 2014. ACM.

[DHJ+07] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo : amazon's highly available key-value store. In Thomas C. Bressoud and M. Frans Kaashoek, editors, *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP)*, pages 205–220. ACM, 2007.

[GPSW06] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 89–98. ACM, October 2006.

[HA90] P. W. Hutto and M. Ahamad. Slow memory : Weakening consistency to enhance concurrency in distributed shared memories. In *Proceedings.,10th International Conference on Distributed Computing Systems*, pages 302–309. IEEE Computer Society, January 1990.

[KB17] Martin Kleppmann and Alastair R. Beresford. A Conflict-Free Replicated JSON Datatype. *IEEE Transactions on Parallel and Distributed Systems*, 28(10) :2733–2746, October 2017.

[Kum19] Saptaparni Kumar. *Fault-Tolerant Distributed Services in Message-Passing Systems*. PhD thesis, Texas A&M University, 2019.

[Lam79] Lamport. How to Make a Multiprocessor Computer That Correctly Executes Multiprocess Programs. *IEEE Transactions on Computers*, C-28(9) :690–691, September 1979.

[Lam86] Leslie Lamport. On interprocess communication. *Distributed Computing*, 1(2) :86–101, June 1986.

[LS88] Richard J. Lipton and Jonathan S. Sandberg. PRAM : A Scalable Shared Memory. Technical report, Princeton University, Department of Computer Science, 1988.

[Mis86] J. Misra. Axioms for memory access in asynchronous hardware systems. *ACM Transactions on Programming Languages and Systems*, 8(1) :142–153, January 1986.

[MK23] Anshuman Misra and Ajay D. Kshemkalyani. Byzantine fault-tolerant causal ordering. In *24th International Conference on Distributed Computing and Networking, (ICDCN)*, pages 100–109. ACM, 2023.

[MKE09] Sascha Müller, Stefan Katzenbeisser, and Claudia Eckert. Distributed Attribute-Based Encryption. In Pil Joong Lee and Jung Hee Cheon, editors, *Information Security and Cryptology – ICISC 2008*, volume 5461, pages 20–36. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[Mos93] David Mosberger. Memory consistency models. *ACM SIGOPS Operating Systems Review*, 27(1) :18–26, January 1993.

[NJDK16] Petru Nicolaescu, Kevin Jahns, Michael Derntl, and Ralf Klamma. Near Real-Time Peer-to-Peer Shared Editing on Extensible Data Types. In *Proceedings of the 19th International Conference on Supporting Group Work*, pages 39–49. ACM, November 2016.

[Per17] Matthieu Perrin. *Concurrence et cohérence dans les systèmes répartis*. ISTE Group, September 2017.

[Ray18] Michel Raynal. *Fault-Tolerant Message-Passing Distributed Systems : An Algorithmic Approach*. Springer, September 2018.

[RBMC20] Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. Zero Trust Architecture. Technical report, National Institute of Standards and Technology, August 2020.

[RS95] Michel Raynal and André Schiper. From causal consistency to sequential consistency in shared memory systems. In Gerhard Goos, Juris Hartmanis, Jan Leeuwen, and P. S. Thiagarajan, editors, *Foundations of Software Technology and Theoretical Computer Science*, volume 1026, pages 180–194. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995.

[SCB22] Premathas Somasekaram, Radu Calinescu, and Rajkumar Buyya. High-Availability Clusters : A Taxonomy, Survey, and Future Directions. *Journal of Systems and Software*, 187 :111208, May 2022.

[SFK+09] Atul Singh, Pedro Fonseca, Petr Kuznetsov, Rodrigo Rodrigues, and Petros Maniatis. Zeno : Eventually consistent byzantine-fault tolerance. In Jennifer Rexford and Emin Gün Sirer, editors, *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 169–184. USENIX Association, 2009.

[SPBZ11] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-Free Replicated Data Types. In *Stabilization, Safety, and Security of Distributed Systems*, volume 6976, pages 386–400. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[SS05] Yasushi Saito and Marc Shapiro. Optimistic Replication. *ACM Computing Surveys*, 37(1) :42, 2005.

[SS19] Mehrnoosh Shakarami and Ravi Sandhu. Refresh Instead of Revoke Enhances Safety and Availability : A Formal Analysis. In *33th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec)*, volume LNCS-11559, page 301. Springer International Publishing, July 2019.

[TWZP19] Lewis Tseng, Zezhi Wang, Yajie Zhao, and Haochen Pan. Distributed Causal Memory in the Presence of Byzantine Servers. In *IEEE 18th International Symposium on Network Computing and Applications (NCA)*, pages 1–8, September 2019.

[VDLLP20] Albert Van Der Linde, João Leitão, and Nuno Preguiça. Practical client-side replication : Weak consistency semantics for insecure settings. *Proceedings of the VLDB Endowment*, 13(12) :2590–2605, August 2020.

[Yjs23] Yjs/yjs : Shared data types for building collaborative software. https ://github.com/yjs/yjs, December 2023.

[YLWV17] Zheng Yan, Xueyun Li, Mingjun Wang, and Athanasios V. Vasilakos. Flexible Data Access Control Based on Trust and Reputation in Cloud Computing. *IEEE Transactions on Cloud Computing*, 5(3) :485–498, July 2017.