Consistance Faible Byzantine pour le Cloud

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

Contexte

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 terme 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 *consistence 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 que des compromis de fonctionnement sont souvent nécessaires. En particulier, si c'est la consistence forte qui est recherchée, le temps de calcul est proportionnel à la latence de tout le réseau.

En outre, pour proposer des solutions véritablement sécurisées, 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, assurer une telle propriété de consistence 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 considèrer des données avec des propriétés de *consistences faibles*.

Applications Collaboratives

Le besoin d'applications collaboratives au fonctionnement aussi horizontal que possible n'est plus à démontrer. Dans le contexte zero-trust que nous venons de décrire, les spécificités de ces applications sont les suivantes. Dans le cadre de l'édition collaborative temps-réel, le retour des utilisateurs peut être exploité afin de travailler effectivement avec certaines consistences faibles. Dans le cadre de groupe de collaboration de taille importante, les solutions doivent minimiser en outre les impacts de faille sur la sécurité opérationnelle, en particulier si des solutions cryptographiques sont utilisées.

Objectifs

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

— la sérialisabilité
— la consistence causale
— la consistence ultime forte

Si la consistence ultime 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.

Il existe actuellement des études et solutions partielles pour la consistence causale[vdLLP20]. Les objectifs de cette thèse sont à la fois d'étudier ces trois types de consistence faible en situation byzantine et de définir des algorithmes byzantins efficaces pour pouvoir les implémenter.

Objectif du stage

L'objectif du stgae est d'étudier des solutions byzantines sans primitives cryptographiques, ou avec des primitives peu coûteuses.

La première étape consistera à évaluer les solutions cryptographiques de l'état de l'art (comme les signatures cryptographiques) dans le contexte d'utilisation de l'édition collaborative.

Une seconde étape consistera à proposer des algorithmes byzantins pour l'une des consistances faibles considérées.

Enfin, il s'agira, en collaboration avec la société partenaire, de développer une preuve de concept des premiers algorithmes.

Poursuite en Thèse : Une poursuite en thèse est possible pour ce sujet sous la forme d'une thèse CIFRE (thèse en entreprise, société SCILLE).

References

Références

[Per17] Matthieu Perrin. *Distributed Systems - Concurrency and Consistency*. Elsevier and ISTE Press, 2017.

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

[vdLLP20] Albert van der Linde, João Leitão, and Nuno M. Preguiça. Practical client-side replication : Weak consistency semantics for insecure settings. *Proc. VLDB Endow.*, 13(11) :2590–2605, 2020.

Supervisors

Emmanuel Godard is professeur (full professor) at Aix-Marseille University. His research interests are focussed on understanding and maximizing decentralization in distributed systems.

Corentin Travers is Maître de Conférences (assistant professor) at Aix-Marseille University. His research interests are efficient distributed algorithms for both share memory and distributed networks.

Distributed Computing Team

The Distributed Computing team (head Jérémie Chalopin – DR CNRS) is a part of Laboratoire d'Informatique et Systèmes (LIS CNRS UMR 7020). It is an international level research team, with 8 permanent members whose interest range from reliable distributed algorithms, privacy in distributed systems, to communication networks, graph algorithms, mobile agents, and IoT.

Société SCILLE

La société Scille est la startup éditrice du logiciel primé "parsec" (https://parsec.cloud/).