Synchronisation d'horloge dans un système multi-agents

Mohamed Limame, Julien Henriet, Christophe Lang, Nicolas Marilleau

Synchronisation d'horloge dans un système multi-agents

M. Limame[1] J. Henriet[2] C. Lang[2] N. Marilleau[1]

[1] IRD/UMMISCO – {prenom.nom@ird.fr}

[2] Univ. Bourgogne Franche-Comté FEMTO-ST Institute, CNRS {prenom.nom@univ-fcomte.fr}

Résumé

Nous présentons dans cette publication un algorithme permettant au sein d'un système multi agents (SMA) d'instaurer une cohérence au niveau des données recueillies par chacun des agents à travers un nouveau mode de synchronisation. Nous nous focalisons essentiellement sur une flotte de drones comme application de l'algorithme de synchronisation. Notre approche est fondée sur l'assimilation de données.

Mots-clés : Synchronisation, Cohérence des données, assimilation de données, systèmes multi-agents.

Abstract

We present in this publication an algorithm allowing within a multi-agent system (SMA) to establish coherence at the level of the data collected by each agent through a new synchronization mode. We focus essentially on a fleet of drones as application of the synchronization algorithm. Our approach is based on data assimilation.

Keywords: Synchronization, data consistency, data assimilation, multi-agent systems.

1 Introduction

Un système distribué est constitué d'un ensemble d'entités autonomes interconnectées, pouvant communiquer ensemble et dont chacune dispose d'une horloge locale. Néanmoins, il est souvent nécessaire que ces entités obtiennent une notion commune du temps via une synchronisation. En effet, les systèmes distribués sont majoritairement synchronisés afin d'assurer le service pour lesquels ils ont été conçus, notamment dans dans le domaine de supervision utilisant des systèmes distribuées avec entités mobiles.

1.1 L'algorithme de synchronisation un domaine exploré

Sur la base des horloges physiques, trois orientations majeures sont proposées dans la littérature pour la synchronisation :

(i) les architectures centralisées dans lesquelles il existe une hiérarchie distinguant un noeud par une unique horloge permettant de synchroniser l'ensemble du système distribué. Comme détaillé dans [1], parmi les techniques de synchronisation utilisant cette architecture, nous pouvons citer le système de positionnement global (GPS);

(ii) les architectures distribuées de synchronisation dans lesquelles il y a réplication d'horloge sur l'ensemble des nœuds du système ce qui rend cette architecture non hiérarchique puisque donnant à chaque agent la capacité de calculer l'horloge. Parmi les techniques de synchronisation utilisant cette architecture, nous pouvons citer l'algorithme de Cristian dans lequel, comme présenté dans [2], un agent du système envoie une demande au serveur de temps pour recevoir l'horloge actuelle. Lorsqu'il reçoit la réponse, il trouve le délai de transmission (délai entre l'envoi de la demande et la réception de la réponse), le divise par deux et l'ajoute au délai reçu du serveur. Une limite de cet algorithme se situe dans ses implémentations qui se basent sur un serveur unique, le rendant impropre à une utilisation dans les applications distribuées où la redondance peut s'avérer critique. Il existe aussi l'algorithme de Berkeley qui, comme précisé dans [3], contient un serveur temps actif qui interroge périodiquement chaque agent du système distribué pour lui demander l'heure. Sur la base des réponses, il calcule une durée moyenne et demande à tous les agents du système d'avancer leurs horloges vers la nouvelle

heure ou de ralentir leur horloge jusqu'à ce qu'une réduction spécifiée soit atteinte. Avec cette approche il y a un risque fort d'avoir un goulot d'étranglement au niveau du serveur temps.

(iii) les architectures distribuées hybrides de synchronisation dans lesquelles il y a réplication d'horloge uniquement sur certains nœuds du système ce qui fait que cette architecture est un mixte des deux architectures (i) et (ii). Parmi les techniques de synchronisation pouvant être appliquées à cette architecture, nous pouvons citer le protocole Network Time Protocol (NTP) qui, selon [2], utilise un système hiérarchique stratifié de sources de temps se basant sur l'UTC comme temps de référence. Lorsqu'un serveur est défini comme horloge maître, un message est envoyé à tous les esclaves (clients) pour synchroniser l'horloge. Ensuite, les esclaves calculent leur heure locale et la dérive de l'horloge maître. Mais le problème dans cette approche est qu'il existe un délai de propagation. Le décalage est égal à la différence entre l'horloge de l'esclave et l'horloge du serveur maître.

Il existe une approche alternative à la synchronisation d'horloges physiques qui se base sur le concept de l'horloge logique utilisant des algorithmes de synchronisation. Une horloge logique est un dispositif logiciel qui sert à établir et mesurer une notion de temps établie selon la relation de causalité arrivé-avant dans un système réparti asynchrone. Différents méthodes de synchronisation d'horloges logiques existent. Parmi ces méthodes : l'horloge de Lamport qui, comme détaillé dans [4], attribue une horloge logique ou estampille à tous les événements d'un système distribué de manière à ce que si un événement E1 précède un événement E2 passé sur un même agent du système, alors H(E1) < H(E2). Néanmoins, quand deux événements sont concurrents, on ne peut rien conclure quant à leurs horloges logiques respectives;la seule certitude est que Si H(E1) = H(E2) alors E1 et E2 sont concurrents. Il y a aussi l'horloge de Mattern avec laquelle chaque agent *e* possède un vecteur d'entiers appelé estampille dans lequel chaque composant estampille[i] est l'estimation par *e* de la valeur de l'horloge de Lamport de l'agent *i*. En particulier, estampille[*e*] est exactement l'horloge de Lamport de *e*. Les horloges de Mattern donnent une information plus précise que les horloges logiques de lamport pour un coût plus élevé en mémoire.

1.2 Les limites des algorithmes de synchronisation dans les systèmes autonomes

Quelle que soit la technique de synchronisation d'horloges physiques utilisée, une synchronisation ne peut être obtenue qu'au travers d'une approche algorithmique évoluée et de protocoles de synchronisation dont la complexité est intimement liée à la précision souhaitée de l'horloge. Plus le degré de précision est important plus le système a besoin de capacités énergétique et calculatoire pour : (i) exécuter les algorithmes ; (ii) communiquer en vue d'échanger de l'information d'horodatage ; (iii) et utiliser les horloges physique. Ceci constitue un véritable frein pour un usage dans les systèmes embarqués où l'énergie, les communications et la puissance de calcul sont limitées. En conséquence, la synchronisation d'horloges physiques est inadaptée pour un système multi-agents mobile riche en capteurs à l'instar d'une flotte de drones.

Quelle que soit la technique de synchronisation d'horloges logiques utilisée, elles se caractérisent toutes par un coût élevé en mémoire pour pouvoir gérer la logique d'ordonnancement des événements survenant dans un système distribué. En conséquence, la synchronisation d'horloges logiques constitue une difficulté dans un système multi-agents mobile riche en capteurs dont la capacité en mémoire est limitée.

1.3 Tirer parti des concepts de connaissance dans les SMA pour synchroniser des entités autonomes

D'une manière générale, en dehors des domaines d'application critiques, la synchronisation oeuvre pour que les agents du système multi-agents soient en accord et puissent prendre des décisions cohérentes en se basant sur des ressources et événements du système. En conséquence, la synchronisation des horloges logiques est nécessaire pour garantir le bon fonctionnement du système et une synchronisation des horloges physiques n'est pas indispensable.

Les méthodes de synchronisation d'horloges physiques et logiques étant inadaptées (non optimisées) pour un SMA mobile, nous proposons dans cet article de décrire une nouvelle approche de synchronisation respectant l'architecture de synchronisation

présentée dans la partie 1.1, paragraphe (ii) et s'appuyant sur le principe d'ordonnancement des événements et faisant appel à la technique d'assimilation de données. Dans ce qui suit, nous faisons d'abord un état des lieux des mécanismes de synchronisation dans les systèmes distribués pour passer ensuite à une description d'une nouvelle approche synchronisation appelée SMASDEV.

2 Nouvelle approche de synchronisation

2.1 Principe

La nouvelle approche apportée s'inspire de l'approche de Lamport basée sur les événements pouvant survenir dans un SMA sauf qu'elle ne s'intéresse pas à leurs ordres de survenance mais elle s'intéresse plutôt au contenu d'un événement et précisément à la donnée qui lui est associée.

N'étant pas gourmande en ressources, nous utiliserons aux techniques d'assimilation des données pour les analyser et les ré-organiser dans un chronographe. Ainsi, cette nouvelle approche a pour objectif d'instaurer une cohérence globale au niveau d'un SMA en permettant à chaque agent de rétablir un ordre chronologique des données qu'il reçoit des autres agents, sur la base de ses connaissances et s ans faire appel à une horloge (physique ou logique).

Le principe consiste à ce que chaque agent du système enregistre en mémoire sa perception personnelle de l'évolution d'une donnée qu'il recueille à une fréquence donnée par rapport à son horloge locale ce qui va lui permettre, grâce au principe de l'assimilation, de prévoir son évolution dans le futur. Pour cela, l'agent doit disposer du modèle d'évolution en adéquation avec la donnée en question. Pour passer d'une perception personnelle vers une perception globale, chaque agent du système doit, dans un premier temps, demander aux autres agents de lui transmettre la donnée qu'ils ont recueillie puis dans un deuxième temps il doit les positionner par rapport à sa perception personnelle. Ainsi l'agent sera en mesure d'identifier le positionnement de l'ensemble des données reçues de la part des autres agents par rapport à son horloge. En conséquence, les agents du système seront synchrones et en phase par rapport à l'évolution de la donnée dans le temps.

2.2 Algorithme SMASDEV

Soit :

- A^d un agent distant
- A^l l'agent local
- H^{Ad} horloge de l'agent distant (A^d)
- H^{Al} horloge de l'agent local (A^l)
- t^{ε} erre ur maximale entre deux horloges du système
- $M^{Ad} = (p_1, p_2,\ldots,p_i,\ldots, p_n)$ une mesure réelle de l'agent A^d des paramètres p_i à la date t selon l'horloge H^{Ad} (t^{test} selon l'horloge H^{Al}, valeur estimée après exécution de l'algorithme)
- $M^{est} = (p^{est}_1, p^{est}_2,\ldots,p^{est}_i,\ldots, p^{est}_n)$ une mesure estimée de l'agent A^l des paramètres p_i à la date t selon l'horloge H^{Al}
- $K^{Al} = (k_1, k_2,\ldots, k_i,\ldots, k_n)$ un tuple ordonné de connaissances de l'agent local (A^l), tel que $k_i = \{(p^{k}_i,t_1), (p^{k}_i,t_2),\ldots, (p^{k}_i ,t_j),\ldots , (p^{k}_i,t_m))\}$ et (p^{k}_i ,t_j) une mesure admise du paramètre p_i à une date t_j selon l'horloge H^{Al}
- $F = (f_1, f_2,\ldots , f_i,\ldots , f_n)$ un tuple ordonnées de fonctions d'estimation $f_i(k_i,t)$ permettant d'estimer la valeur p^{est}_i d'un paramètre p_i, à un instant t selon une base de connaissances k_i et l'horloge H^{Al}
- f^{eval} une fonction d'évaluation multicritère $f^{eval}(M^{est}, M^{Ad})$ qui permet de mesurer la distance entre un tuple de paramètres estimés $M^{est}{}_t$ et un tuple de paramètres mesurés M^{Ad}

Avec :

- $t_{\varepsilon} > | t_{Ad} - t_{Al} |$

L'algorithme de la fonction baptisé SMASDEV (Multi Agent System Synchronisation based on Data Evolution) est décrit ci-dessous :

```
var eval ← +inf
M^est ← tableau(n)
var t^est ← 0
POUR TOUT t_id ∈ [t_Al - t_ε, t_Al + t_ε] FAIRE
        var M^temp ← tableau(n)
        POUR TOUT i ∈ [1,n] FAIRE
                M^temp[i] ← F[i](K^Al[i], t_id)
        FIN POUR
        var eval^temp ← f^eval(M^temp, M^Ad)
        SI eval^temp < eval ALORS
                eval ← eval^temp
                M^est ← M^temp
                t^est ← t_id
        FIN SI
FIN POUR
RETOURNE t^est
```

Algorithme SMASDEV

Dans un premier temps, l'algorithme définit la plage de temps sur laquelle va se baser la fonction d'estimation $f_i(k_i,t)$ et ceci en prenant en compte le taux d'erreur maximale entre deux horloges en l'occurrence entre l'horloge de l'agent distant et l'horloge de l'agent local. Ce taux d'erreur varie en fonction des caractéristiques techniques des horloges utilisées.

Dans un deuxième temps, après avoir obtenu un ensemble de tuples de données estimées par rapport au contexte de l'agent local, l'objectif de l'algorithme est d'identifier le tuple dont les données estimées sont les plus proches par rapport aux tuples des données mesurées par l'agent distant d'où l'utilisation de la fonction f^{eval} qui sert pour calculer les différences entre les tuples de données estimés et les tuples de données mesurées. La variable eval est utilisée pour stocker le tuple de données estimées dont la différence avec le tuple de données mesurées est la moins importante. L'obtention de ce tuple permettra d'identifier l'instant t^{est}. Nous pouvons en conséquence identifier le tuple de données le plus récent en faisant une comparaison entre les instants t^{est} selon l'horloge H^{Al} et t selon l'horloge H^{Ad}. Le tuple dont l'instant t de prise est le plus important correspond au tuple dont les données sont les plus récentes.

La fonction $f_i(k_i,t)$ permettant d'obtenir un tuple de données estimées au cours d'une plage temps est une fonction qui dépend de la nature de la donnée et précisément de son modèle d'évolution dans le temps. Par exemple dans le cas où la donnée objet de l'estimation porte sur la température la fonction d'estimation peut se baser sur une fonction affine. En revanche, une fonction estimant la densité des particules dans l'air suit un modèle plus complexe.

3 Conclusion & Perspectives

Notre objectif est d'introduire un protocole basé sur le contenu des observations d'un agent pour rétablir une chronologie des événements observés et valeurs recueillies. Cette nouvelle approche permettrait en particulier d'apporter une solution à la synchronisation des horloges d'un système multi-agents tels qu'une flotte de drones de surveillance d'un territoire. En perspective, nous souhaitons également explorer la possibilité d'introduire des outils de raisonnement par analogie ou de classification proposés par le paradigme de l'intelligence artificielle dans ce nouveau protocole.

Références

[1] Hofmann¬Wellenhof, B., Lichtenegger, H., and Collins, J., 2001, "Global Positioning System: Theory and Practice," Text Book, Fifth edition

[2] M. Leela, D. Manoj Kumar, G. Bhavana, 2018, Clock Synchronisation in Distributed Systems: A Review

[3] D.Adithya Chandra Varma, Praveen Kumar Reddy.M, Prof.Gopinath, 2013, Performance Comparison of Physical Clock Synchronization Algorithms

[4] Lamport, L., 1978, Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 21(7):558–565