

TD 10 - État global

Mai 2020

L'état global d'un système est constitué d'un état local de chacun des processus, ainsi que de l'état des canaux de communication. Un algorithme de calcul d'état global doit coordonner les enregistrements effectués sur les différents sites de manière à garantir que l'état obtenu en rassemblant les informations locales est cohérent : si cet état ne s'est pas produit effectivement lors de l'exécution de l'application, il pourrait se produire lors d'une autre exécution.

Contexte

- ★ Il y a N processus communiquant par messages au travers d'un réseau fortement connexe.
- ★ Les liaisons sont fiables : les messages transitent en un temps arbitraire mais fini et non nul.
- ★ Les liaisons sont FIFO.

Exercice(s)

Exercice 1 – Exécution de l'algorithme de Chandy & Lamport

Question 1

On considère 3 processus P_1 , P_2 et P_3 , reliés comme indiqué dans la figure 1, et le schéma d'exécution correspondant.

P_1 et P_2 enregistrent spontanément leur état local aux instants représentés par les triangles. Complétez la figure 1 en ajoutant les marqueurs manquants et en indiquant l'instant où P_3 enregistre son état local. Donnez l'état local enregistré par chaque processus ainsi que l'état enregistré pour les liaisons.

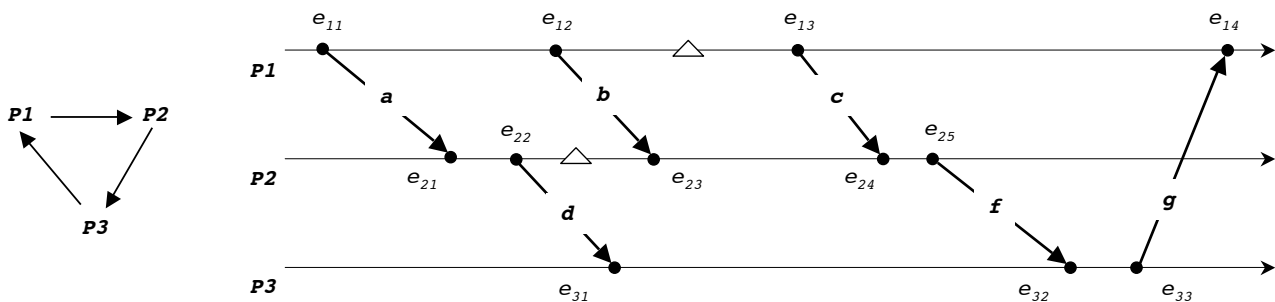


FIGURE 1 – Construction d'un état global

Solution:

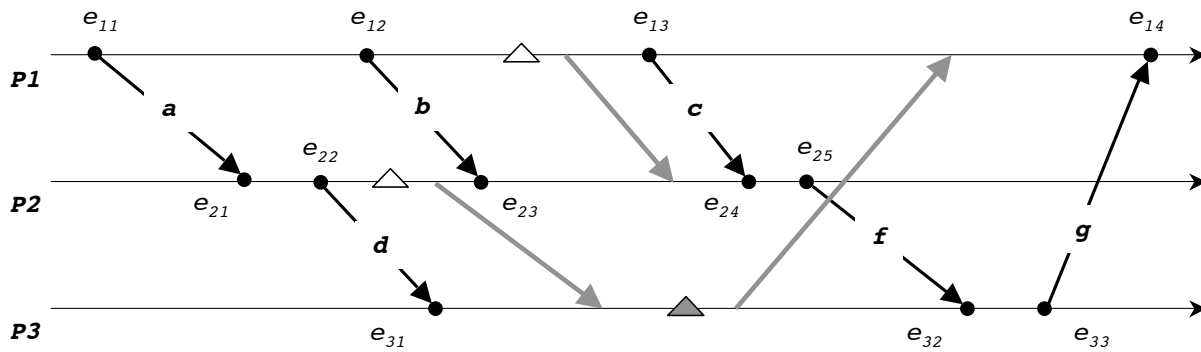


FIGURE 2 – Exécution complète de l'algorithme de Chandy & Lamport

La solution est donnée dans la figure 2. Les flèches grises représentent la circulation des marqueurs : celui de P_1 est nécessairement envoyé avant e_{13} , donc reçu avant e_{24} . Le marqueur de P_2 est envoyé avant e_{25} , donc reçu avant e_{32} . Le marqueur de P_3 est envoyé avant e_{33} , donc reçu avant e_{14} .

L'état local d'un processus est le dernier état avant l'enregistrement, donc :

e_{12} pour P_1 , e_{22} pour P_2 et e_{31} pour P_3 .

P_1 ne reçoit pas de message entre l'enregistrement de son état local et la réception du marqueur : l'état de la liaison (P_3, P_1) est vide. P_2 reçoit le message b : l'état de la liaison (P_1, P_2) est b . P_3 enregistre son état local après la réception du marqueur : l'état de la liaison (P_2, P_3) est donc vide.

L'état d'une liaison c est l'ensemble des messages reçus sur c entre l'enregistrement de l'état local et la réception du marqueur. Il correspond en fait aux messages qui seraient en transit sur c si on ramenait tous les enregistrements à un même instant physique.

Fin solution

Exercice 2 – Pertinence de l'algorithme de Chandy & Lamport

De manière générale, un état global calculé ne correspond à aucun des états atteints par l'application au cours de l'exécution sur laquelle le calcul d'état global a été lancé.

Nous allons montrer qu'il s'agit cependant d'un état que l'application pourrait atteindre. Pour éviter la confusion, nous appelons configurations les états effectivement atteints lors de l'exécution (qui correspondent à un trait vertical dans une représentation telle que celle donnée dans la figure 3).

Soit C_τ la configuration du système au moment où l'algorithme de calcul d'état global est initialisé, C_ϕ la configuration au moment où l'algorithme se termine, et S^* l'état global calculé par l'algorithme.

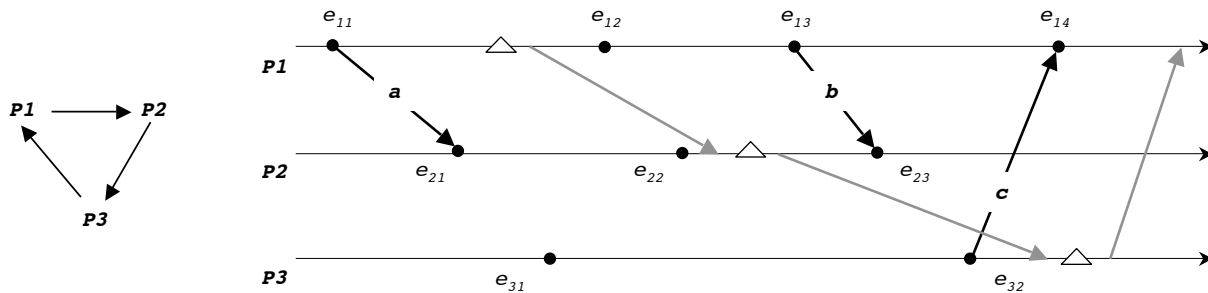


FIGURE 3 – Exécution de l'algorithme de Chandy & Lamport

Question 1

Donnez les valeurs de C_τ , S^* et C_ϕ pour l'exemple de la figure 3. S^* correspond-il à une configuration de cette exécution ?

Solution:

C_τ est obtenu en tirant une ligne sur le premier triangle, C_ϕ est obtenu en tirant une ligne sur la réception du dernier marqueur, et S^* en reliant les différents triangles entre eux.

Une configuration est donnée par la liste des événements ayant eu lieu sur chaque site. On peut ajouter l'état des liaisons mais ce n'est pas indispensable si on connaît la sémantique des événements : un message est en transit si son émission a été exécutée et pas sa réception.

C_τ : $(e_{11}, e_{21}, -)$, toutes les liaisons sont vides.

C_ϕ : (e_{14}, e_{23}, e_{32}) , toutes les liaisons sont vides.

S^* : (e_{11}, e_{22}, e_{32}) , $[P_3P_1] = \{c\}$, les autres liaisons sont vides.

S^* ne correspond pas à une configuration de cette exécution : lorsque e_{32} est exécuté, e_{12} , e_{13} et e_{23} ont déjà eu lieu.

Fin solution

Ici, nous allons appeler exécution répartie la liste classée *par ordre d'occurrence* de tous les événements de l'application (dans la pratique, cette information est inaccessible puisqu'il faut connaître instantanément l'état de tous les sites).

Question 2

Donnez l'exécution répartie *seq* correspondant à l'exemple de la figure 3.

Solution:

L'exécution répartie est donnée par la séquence : $(e_{11}, e_{21}, e_{31}, e_{12}, e_{22}, e_{13}, e_{23}, e_{32}, e_{14})$

Fin solution

Le calcul d'un état global nécessite que chaque site enregistre son état local. On appelle événement pré-enregistrement (resp. post-enregistrement) un événement ayant lieu sur le site p_i avant (resp. après) l'enregistrement par p_i de son état local.

Question 3

Donnez la liste des événements pré-enregistrement et la liste des événements post-enregistrement pour l'exemple de la figure 3.

Solution:

Liste des événements pré-enregistrement : $(e_{11}, e_{21}, e_{22}, e_{31}, e_{32})$

Liste des événements post-enregistrement : $(e_{12}, e_{13}, e_{14}, e_{23})$

Fin solution

On rappelle la définition de la relation de causalité entre deux événements : Soit E un ensemble d'événements, e_1 et e_2 deux éléments de E . La relation de causalité sur E , notée " \rightarrow " est telle que, pour une exécution donnée, $e_1 \rightarrow e_2$ si l'une des 3 conditions suivantes est vérifiée :

1. e_1 et e_2 sont deux événements du même processus et e_1 précède e_2 ,
2. e_1 est l'événement "envoi d'un message" et e_2 est l'événement "réception de message" correspondant,
3. il existe un événement e_3 tel que $e_1 \rightarrow e_3$ et $e_3 \rightarrow e_2$ (transitivité).

Lorsque deux événements ne sont pas reliés causalement (par exemple, e_{12} et e_{22}), l'ordre dans lequel ils se produisent n'influence pas réellement le comportement de l'application répartie. On peut donc construire une relation d'équivalence entre exécutions en se basant sur la causalité :

$$\text{seq} \sim \text{seq}' \Leftrightarrow \text{seq et seq' ont même relation de causalité.}$$

Autrement dit, $e \rightarrow e'$ dans $\text{seq} \Leftrightarrow e \rightarrow e'$ dans seq' .

Question 4

Donnez une exécution seq' équivalente à seq pour l'exemple de la figure 3, dans laquelle les événements post-enregistrement ont tous lieu après les événements pré-enregistrement.

Solution:

On commence par construire les contraintes de causalité entre événements n'ayant pas lieu sur le même site (les contraintes locales sont triviales) :

$$e_{11} \rightarrow \{e_{21}, e_{22}, e_{23}\} \quad e_{13} \rightarrow \{e_{23}\} \quad e_{32} \rightarrow \{e_{14}\}$$

On peut noter qu'aucune de ces contraintes n'est en contradiction avec l'ordonnancement recherché (et ce n'est pas un hasard, comme on le verra à la question suivante). On a beaucoup de solutions : e_{31} et e_{32} peuvent être insérés dans cet ordre n'importe où dans la séquence (e_{11}, e_{21}, e_{22}) . e_{14} peut avoir lieu avant ou après e_{23} . Donc on a par exemple les séquences :

$$(e_{11}, e_{21}, e_{22}, e_{31}, e_{32}, e_{12}, e_{13}, e_{14}, e_{23})$$

$$(e_{31}, e_{11}, e_{21}, e_{32}, e_{22}, e_{12}, e_{13}, e_{23}, e_{14})$$

Fin solution

Nous voulons maintenant montrer que le résultat de la question précédente est général. Pour cela, nous considérons une exécution répartie quelconque seq et une application de l'algorithme de Chandy et Lamport sur cette exécution.

Question 5

Montrez qu'il existe une exécution seq' équivalente à seq dans laquelle les événements post-enregistrement ont tous lieu après les événements pré-enregistrement.

Solution:

Soit seq vérifie déjà la condition et il n'y a rien à faire. Soit il existe un événement pré-enregistrement e_{i+1} ayant lieu immédiatement après un événement post-enregistrement e_i dans seq (NB : le premier événement pré-enregistrement "mal placé" a forcément lieu immédiatement après un événement post-enregistrement). Nous allons montrer qu'on peut permuter e_i et e_{i+1} sans perturber la relation de causalité.

1. e_i et e_{i+1} ne peuvent pas avoir lieu sur le même site : si e_i est un événement post-enregistrement sur p et si e_{i+1} a lieu après e_i sur p , alors e_{i+1} est un événement post-enregistrement : contradiction.

2. e_{i+1} ne peut pas être la réception d'un message envoyé en e_i : supposons que e_i soit une émission. Puisque e_i est un événement post-enregistrement, le site sur lequel il a lieu a enregistré son état avant e_i , donc il a envoyé un marqueur avant e_i . Puisqu'il n'y a pas de déséquence, ce marqueur arrive avant e_{i+1} . L'arrivée du marqueur provoque l'enregistrement sur le site destinataire, donc e_{i+1} est un événement post-enregistrement : contradiction.
3. Puisqu'on suppose que e_i et e_{i+1} sont consécutifs, ils ne peut y avoir un événement intermédiaire les reliant causalement.

En conclusion, e_i et e_{i+1} ne sont pas reliés par la relation de causalité. On peut donc les permuter dans la séquence sans modifier cette relation. En répétant cette permutation de proche en proche, on peut amener tous les événements pré-enregistrement en tête de liste. On a alors une séquence seq' équivalente à seq et qui respecte la condition demandée.

Fin solution

Question 6

En déduire que pour toute exécution seq , il existe une exécution équivalente dont C_τ , C_ϕ et l'état global S^* enregistré par l'algorithme de Chandy et Lamport sont des configurations. Représentez cette exécution pour l'exemple de la figure 3.

Solution:

Parmi les séquences équivalentes à seq , et qui sont telles que tous les événements pré-enregistrement ont lieu avant les événements post-enregistrement, il y en a une qui conserve l'ordre entre les événements pré-enregistrement de seq et l'ordre entre les événements post-enregistrement de seq . Soit seq' cette séquence. Dans l'exemple : $seq' = (e_{11}, e_{21}, e_{31}, e_{22}, e_{32}, e_{12}, e_{13}, e_{23}, e_{14})$.

La configuration du système à l'instant t est donnée par l'ensemble ordonné (ordre local à chaque site) des événements ayant eu lieu avant t sur chaque site. Soit τ (respectivement ϕ) l'instant de la configuration C_τ (respectivement C_ϕ) dans seq .

Tous les événements antérieurs à τ dans seq sont pré-enregistrement. seq' conserve l'ordre des événements pré-enregistrement. Donc seq' a même préfixe que seq et C_τ est une configuration de seq' .

Soit t un instant situé entre le dernier événement pré-enregistrement e_i et le premier événement post-enregistrement e_{i+1} dans seq' . La configuration de seq' à l'instant t contient tous les événements pré-enregistrement de seq et aucun événement post-enregistrement : c'est la définition de S^* . S^* est une configuration de seq' .

Tous les événements postérieurs à ϕ dans seq sont post-enregistrement. seq' conserve l'ordre des événements post-enregistrement, donc seq et seq' ont même terminaison. Ce qui veut dire aussi que le préfixe de seq' contient tous les événements ayant eu lieu avant ϕ dans seq , dans un ordre éventuellement différent.

Puisque les deux exécutions ont même relation de causalité, l'ordre local des événements sur chaque site n'est pas modifié, donc C_ϕ est une configuration de seq' .

L'obtention de seq' à partir de seq , c'est l'élastique de Mattern : on étire les exécutions pour aligner les instants d'enregistrement... Le résultat est donné dans la figure 4 : les triangles gris représentent les instants d'enregistrement dans l'exécution initiale. Dans cette nouvelle exécution, ils ont été décalés sans que la relation de causalité ait été perturbée.

Fin solution

