

# Base de données distribuées

## QCM et tests de connaissance

MaDSI 2019

\*\*\*

### A. Interblocage

Un interblocage est une situation dans laquelle deux ou plusieurs transactions attendent les éléments de données détenus et libérés par les autres transactions.

Un blocage peut survenir si les 4 conditions suivantes sont réunies simultanément dans un système;

1. Exclusion mutuelle: au moins une ressource est détenue dans un mode non partageable. Par exemple, parmi les transactions s'il existe une demande de verrouillage exclusif (verrouillage en écriture), cet élément de données ne peut pas être partagé avec d'autres.
2. Mise en attente et attente: Il y a une transaction qui a acquis et maintenu verrouillé un élément de données et attend un autre élément de données.
3. Pas de préemption: situation dans laquelle une transaction ne libère le verrouillage des éléments de données qu'elle détient qu'après l'achèvement de la transaction.
4. Attente circulaire: situation dans laquelle une transaction dit que T1 attend qu'une autre transaction T2 libère le verrou sur certains éléments de données, à son tour T2 attend qu'une autre transaction T3 libère le verrou, etc.

Lors du traitement de la transaction, si toutes les conditions susmentionnées sont remplies, il se produit un blocage.

### B. Approche centralisée de détection des blocages

C'est la technique utilisée dans le système de base de données distribuée pour gérer la détection d'interblocage. Selon cette approche, le système conserve un graphe d'attente global dans un seul site choisi, nommé coordonnateur de la détection des blocages. Le graphe d'attente global est mis à jour dans les conditions suivantes;

- chaque fois qu'un nouveau noeud est inséré ou supprimé dans l'attente locale pour les graphes de tous les sites,
- périodiquement
- chaque fois que le coordinateur appelle l'algorithme de détection.

### Comment ça marche?

- Lorsque le coordinateur de détection d'interblocage démarre l'algorithme de détection d'interblocage, il recherche des cycles. Si le coordinateur trouve un cycle, alors ce qui suit se produira.
- Le coordinateur sélectionne une transaction victime qui doit être annulée.
- Le coordinateur informe de la transaction victime sur tous les sites de la base de données distribuée.
- Les sites annulent la transaction.

Cette approche de détection centralisée peut entraîner des retours en arrière inutiles pour l'un des motifs suivants: (la cause principale étant le délai de communication)

1. Faux cycles
2. L'annulation d'une transaction individuelle pendant une impasse et une victime est choisie - par exemple; supposons qu'un blocage a eu lieu dans une base de données distribuée. Ensuite, le coordinateur choisit une transaction victime et informe les sites de la victime à annuler. Dans le même temps, pour d'autres raisons, une transaction  $T_i$  est annulée elle-même. Maintenant, tout le système effectuait des annulations inutiles.

## C. Gestion des blocages dans la base de données distribuée

### Blocage dans les bases de données distribuées

Comme dans le cas des systèmes de base de données centralisés, les systèmes de base de données distribués sont également sujets aux blocages. Dans les systèmes de bases de données distribuées, nous devons gérer les transactions différemment. Dans chaque site faisant partie de la base de données distribuée, nous avons les composants spécifiques à la transaction - coordinateurs de transaction, gestionnaires de transaction, gestionnaires de verrouillage, etc. Les données peuvent avant tout être détenues par de nombreux sites ou répliquées dans de nombreux sites. Pour ces raisons, la gestion des interblocages est un travail difficile dans la base de données distribuée.

L'interblocage peut être traité de deux manières;

1. **Prévention de l'interblocage** - il s'agit d'empêcher l'interblocage avant qu'elle ne se produise. C'est une tâche plus difficile dans le système de base de données centralisée car cela implique plus de restaurations et ralentit les transactions. Dans une base de données distribuée, cela causerait plus de problèmes car cela annulerait

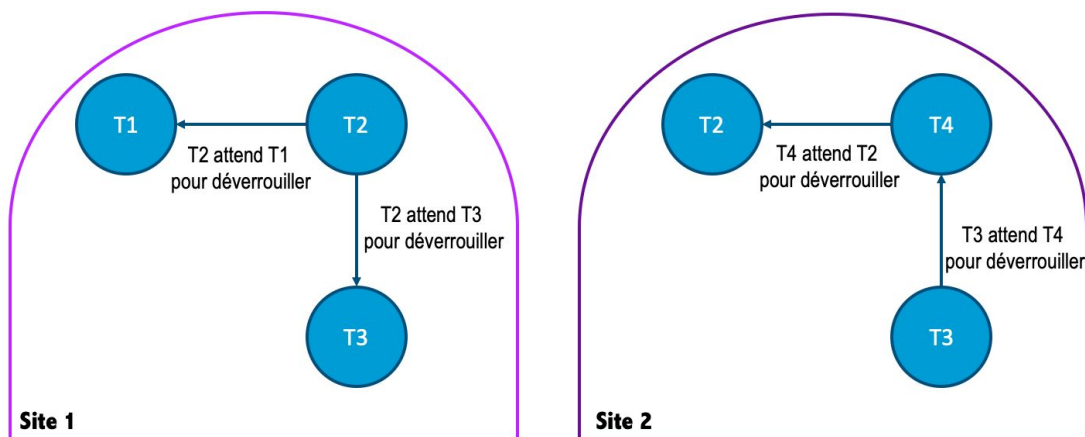
plus de transactions qui se produisent dans plus de sites (pas sur un seul serveur mais éventuellement sur plusieurs serveurs).

2. **Détection d'interblocage** - il traite de la détection d'interblocage, le cas échéant. Dans les systèmes de base de données centralisés, la détection est plus facile que la prévention. Nous avons géré la détection en utilisant des graphes d'attente. Dans le cas d'une base de données distribuée, le principal problème est de savoir où et comment gérer les graphes d'attente.

### Technique de détection de blocage dans une base de données distribuée

Nous avons géré la détection des blocages dans un système de base de données centralisé à l'aide du graphe d'attente. La même chose peut être utilisée dans une base de données distribuée. Autrement dit, nous pouvons conserver des graphes d'attente locaux sur chaque site. (Comment construire un graphique d'attente peut être référé ici). Si le graphe d'attente local de n'importe quel site forme un cycle, alors nous dirions qu'un blocage a eu lieu.

D'autre part, aucun cycle dans le graphe d'attente local ne signifie pas qu'il n'y a pas d'interblocage. Laissez-nous discuter de ce point avec des exemples de graphiques d'attente locaux, comme indiqué ci-dessous;



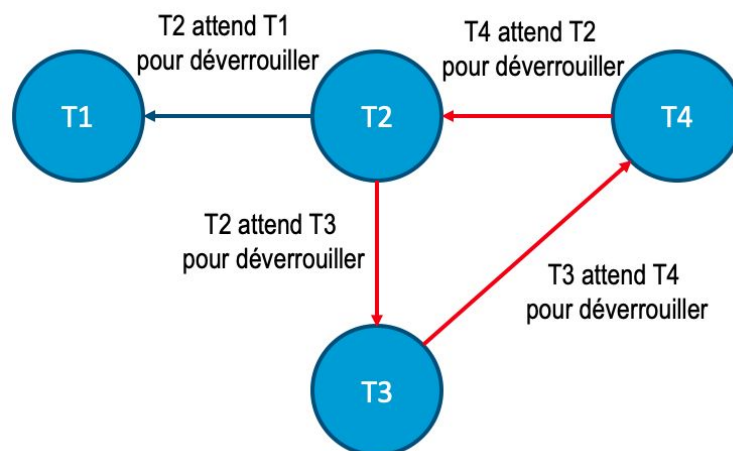
**Figure 1.** Graphes d'attente globaux

La figure 1 montre l'état de la demande de verrouillage pour les transactions T1, T2, T3 et T4 dans un système de base de données répartie. Dans le graphe d'attente local du SITE 1, la transaction T2 attend la fin des transactions T1 et T3. Sur le SITE 2, les transactions T3 attendent T4 et T4 attend T2. Sur les graphes d'attente locaux des sites SITE 1 et 2, il est clair que les transactions T2 et T3 sont impliquées dans les deux sites.

Comment cela a-t-il pu arrivé? Par exemple, la transaction T2 lancée sur le SITE 2 peut nécessiter certaines données conservées dans les transactions T1 et T3 du SITE 1. Par conséquent, le SITE 2 transfère la demande au SITE 1. Si les transactions sont occupées, le SITE 1 insère les noeuds T2 à T1 et T2 à T3 dans son graphe d'attente local.

Autre exemple, la transaction T3 initiée sur le SITE 1 peut nécessiter des données conservées par la transaction T4 sur le SITE 2. Ainsi, le SITE 1 transfère la demande au SITE 2. Sur la base du statut de T4, le SITE 2 insère un noeud T3 à T4 dans son graphe d'attente local.

Vous pouvez observer, à partir de l'attente locale des graphes des sites 1 et 2, qu'il n'y a aucun symptôme de cycle. Si nous fusionnons ces deux graphes d'attente locaux en un seul graphe d'attente, nous obtiendrions alors le graphe présenté à la figure 2 ci-dessous. D'après la figure 2, il est clair que l'union de deux graphes d'attente locaux a formé un cycle, ce qui signifie qu'un interblocage s'est produit. Ce graphe d'attente fusionné est appelé graphe d'attente global.



**Figure 2.** Graphe d'attente global