

1. Algorithmes de contrôle réparti

Rappelons que nous nous adressons à un environnement distribué

" **Absence de mémoire commune** (support habituel de l'état)

" **Absence d'horloge commune** (séquencement des événements)

" **Asynchronisme des communications**

Pas de borne supérieure sur le temps de transit d'un message

Conséquence du mode de fonctionnement de la plupart des réseaux (dont l'Internet)

Asynchronisme des traitements

Pas de bornes sur le rapport relatif des vitesses d'exécution sur deux sites

Conséquence de la variabilité de la charge et de l'absence (en général) de garanties sur l'allocation des ressources

1. Algorithmes de contrôle réparti

Lorsque des processus parallèles sont contrôlés par des variables partagées et qu'ils disposent de mémoires communes, on dit que le contrôle est centralisé.

Notre cas d'étude : contrôle de mécanismes de communication et de synchronisation de processus qui s'exécutent en parallèle sur un support réparti. i.e les processus participants s'exécutent sur différents sites (LAN ou WAN) dans le cadre d'une coopération à un même service et communiquent par l'échange de messages.

1. Algorithmes de contrôle réparti

Un algorithme de contrôle est dit distribué s'il n'utilise pas un même espace d'adressage dans lequel différents processus peuvent lire et écrire: son but est généralement d'assurer la synchronisation et la communication de ces processus:

En grande majorité, les ARs sont basés sur l'échange de messages.

Pour notre cas d'étude, nous considérons que la communication se fait par demande d'informations ou diffusion de l'information d'un processus vers les autres processus.

1. Algorithmes de contrôle réparti : hypothèses

.On suppose pour simplifier que **l'ensemble des processus du SR** sont placés sur des sites distincts, **un processus par site**.

.On rappelle le problème de base d'un SR est que **chaque site** ne connaît à un instant donné que **son propre état**, les informations qu'il obtient sur l'état des autres processus ne lui parviennent pas simultanément avec les changements d'état de ceux-ci, la réception d'une information venant d'un autre site l'informe sur un état antérieur mais pas sur l'état actuel : il n'y a **pas de vision globale de l'état du système**.

195

1. Algorithmes de contrôle réparti : Maillage dans les SRs

Une caractéristique importante d'un système distribué où l'on emploie des Algorithmes basés sur l'échange de messages, est le **maillage du réseau de communication**: l'ensemble des liaisons directes entre les sites :C'est un **maillage logique** ; (il est évidemment nécessaire que le **réseau physique soit connexe**.)

196

1. Algorithmes de contrôle réparti : Maillage dans les SRs

.Les principaux maillages utilisés dans un SR sont :

.**Le maillage en anneau** : le réseau est un cycle, tout processus possède deux voisins et il ne peut communiquer qu'avec ceux-ci.

.**Le maillage en étoile** : un processus particulier peut communiquer avec tous les autres qui, réciproquement, ne communiquent qu'avec le processus particulier. Ce maillage présente l'inconvénient de particulariser un processeur et de rendre fâcheuse la panne du site central.

197

1. Algorithmes de contrôle réparti : Maillage dans les SRs

Et aussi

Le maillage complet : tout processus peut communiquer avec tout processus.

Le maillage en arbre : la panne de tout processus non terminal déconnecte le système logique.

Dans les trois derniers maillages, il est possible de prévoir une reconfiguration lorsque l'un des processeurs tombe en panne (si les connexions physiques le permettent).

198

1. Algorithmes de contrôle réparti : qualités souhaitées

- Minimiser le temps d'exécution.
- Que le nombre de messages échangés soit peu important.
- Que le trafic sur les voies de communication soit bien réparti entre les lignes de communication et qu'il n'y ait pas de site trop sollicité par la réception ou l'envoi de messages.
- Qu'il ne nécessite pas certaines hypothèses sur les voies de communication, par exemple qu'il admette le déséquencelement des messages.
- Qu'il ait une bonne résistance aux pannes, c'est-à-dire que le système contrôlé ne soit pas irrémédiablement interrompu si l'un des processeurs qui assurent le contrôle devient défectueux.

199

1. Algorithmique de contrôle réparti : qualités souhaitées

Sûreté (safety): un événement indésirable n'arrivera jamais

Exemples : violation de l'exclusion mutuelle, incohérence dans les données.

Vivacité (liveness) : un événement désirable finira par arriver

Exemples: un message sera délivré à son destinataire, une ressource demandée sera rendue disponible, un algorithme se termine. Il est souvent impossible de fixer une borne supérieure à l'attente. Les propriétés de vivacité sont plus difficiles à établir que celles de sûreté.