

SYNCHRONISATION DES PROCESSUS DANS UN ENVIRONNEMENT DISTRIBUÉ

SAIDOUNI Djamel Eddine

**Université Constantine 2 - Abdelhamid Mehri
Faculté des Nouvelles Technologies de l'Information et de la Communication
Département d'Informatique Fondamentale et ses Applications**

Laboratoire de Modélisation et d'Implémentation des Systèmes Complexes

Djamel.saidouni@univ-constantine2.dz

Tel: 0559082425

INTRODUCTION

LA SYNCHRONISATION PEUT SE FAIRE SELON DIFFÉRENTES APPROCHES QUI SONT FONCTIONNELLEMENT CORRECTES, MAIS NON ACCEPTABLES.

EXEMPLE: SOLUTIONS BASÉES CLIENT-SERVEUR.. CONDUIT À DES SOLUTIONS SIMPLES MAIS AVEC LES INCONVÉNIENTS SUIVANTS:

- 1) DÉSÉQUILIBRE ENTRE LES SITES POINT DE VUE RESPONSABILITÉ DANS L'ASSURANCE DE LA SYNCHRONISATION.
- 2) LE TEMPS SIGNIFICATIF MIS EN ŒUVRE LORS DE LA SYNCHRONISATION DES PROCESSUS.
- 3) LA FRAGILITÉ DE CES SOLUTIONS DUE AU RISQUE DE PANNES DES SERVEURS. A CET ÉGARD, LE DÉVELOPPEMENT D'APPLICATIONS DISTRIBUÉES NÉCESSITE UN NOUVEAU MODE DE PENSÉE.

LE CONTRÔLE DISTRIBUÉ

INGRÉDIENTS POUR LE DÉVELOPPEMENT D'UNE APPLICATION DISTRIBUÉE

hypothèses sur les liaisons de communication:

H1 : Les liaisons sont fiables (pas de pertes de messages).

H2 : Les liaisons sont régulières ou FIFO (les messages sont reçus dans l'ordre de leur émission).

H3 : Le délai de transmission d'un message est fini mais indéterminé.

H4 : Le délai de transmission d'un message est borné. Dans ce cas, un message qui n'est pas reçu dans les temps est perdu.

LE CONTRÔLE DISTRIBUÉ

INGRÉDIENTS POUR LE DÉVELOPPEMENT D'UNE APPLICATION DISTRIBUÉE (SUITE)

Démarche à suivre :

Mise en évidence des propriétés fonctionnelles que doit vérifier toute solution au problème posé.

Recensement des processus intervenants dans l'application.

Pour chaque processus, mettre en évidence deux catégories d'événements :

- Les événements qui sont à l'initiative du processus.
- Les événements externes auxquels le processus doit réagir.

Association à chacun des événements, le traitement adéquat.

Vérification de la satisfaction des propriétés fonctionnelles.

Étude des propriétés de : Performance, Distribution du contrôle, Passage à l'échelle, etc.

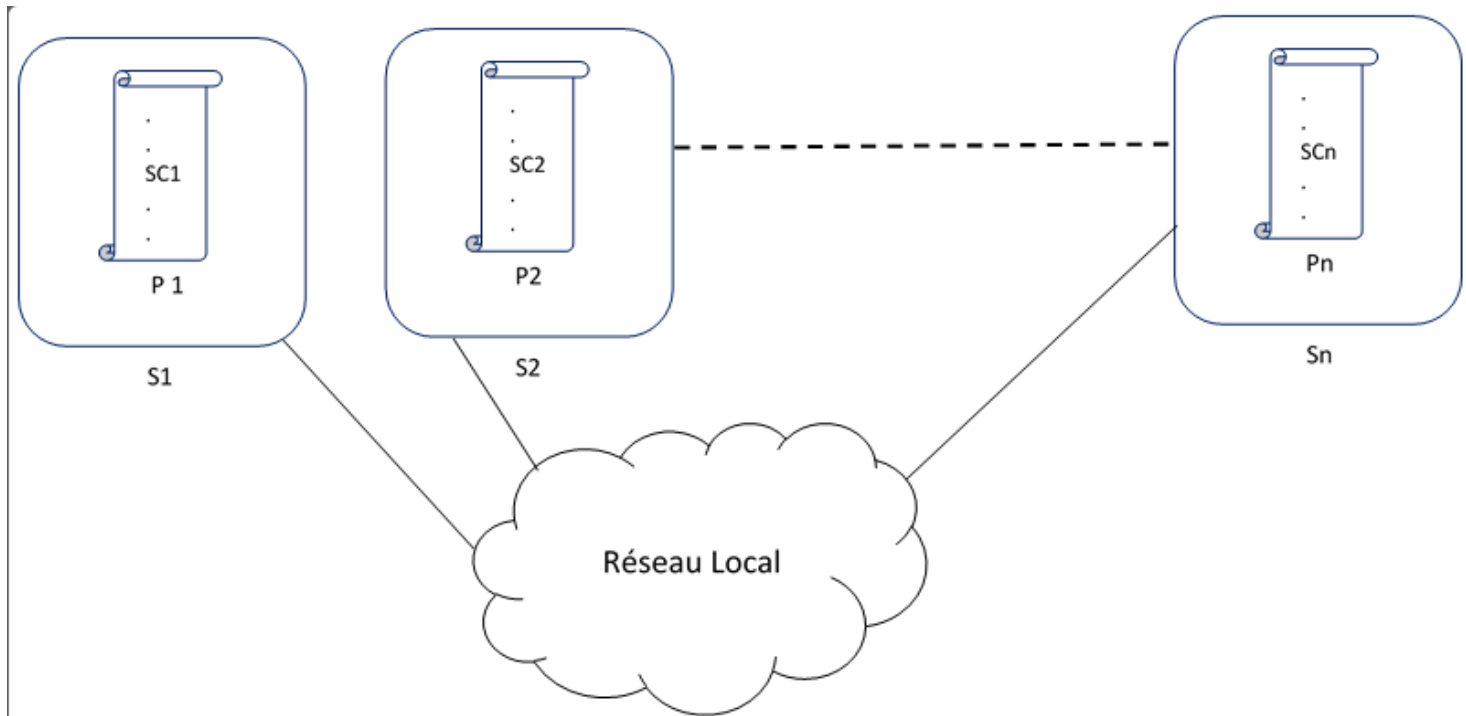
LE CONTRÔLE DISTRIBUÉ

EXEMPLE ILLUSTRATIF

Application constituée de n processus (**P1**, **P2**,, **Pn**) résidant chacun dans un site.

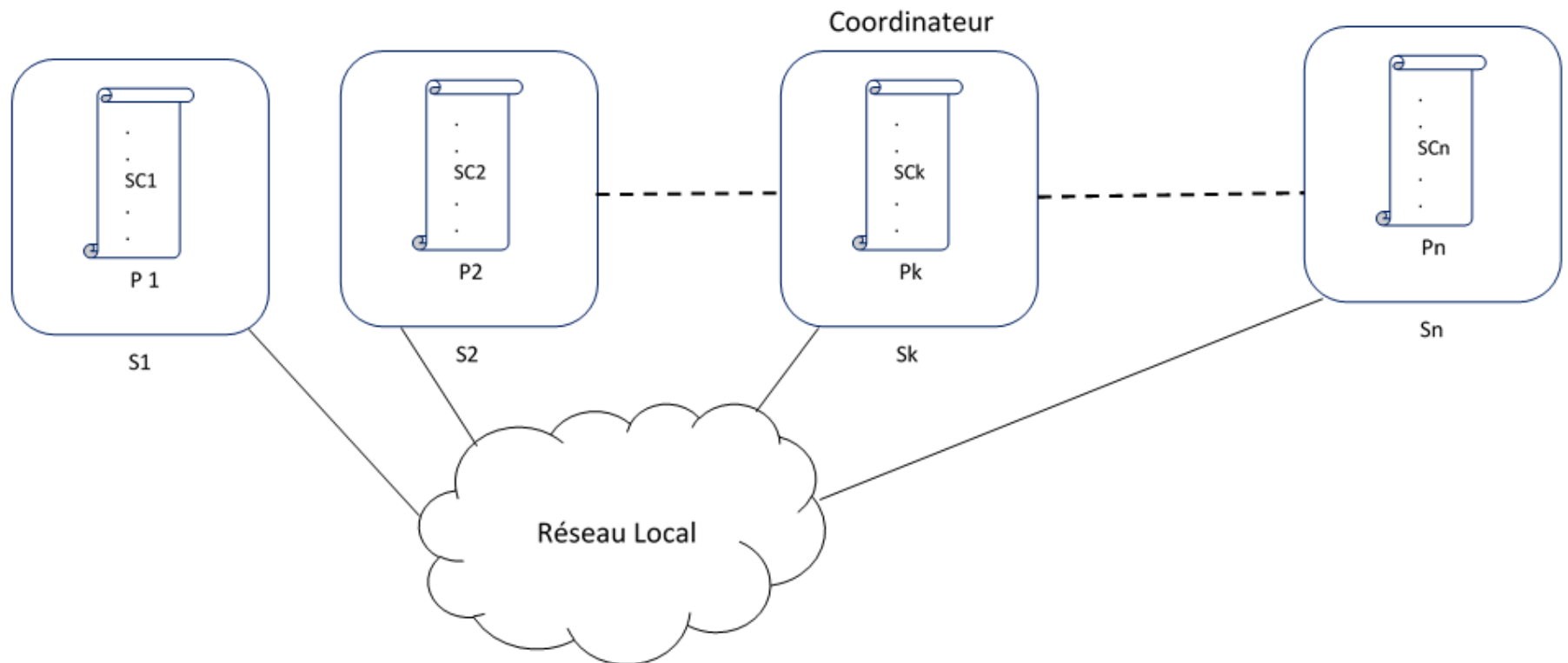
Les n sites (**S1**, **S2**,, **Sn**) sont interconnectés par un réseau local.

Chaque processus englobe une section de code particulière (**SC**) devant s'exécuter en exclusion mutuelle avec les sections de code des autres processus.



SOLUTION NAÏVE

Une solution consiste à désigner un site particulier qui jouera le rôle de coordinateur (**contrôleur**) pour l'accès aux sections critiques des processus.



SOLUTION NAÏVE

PROPRIÉTÉS FONCTIONNELLES:

Toute solution, au problème de la section critique, doit vérifier deux types de propriétés :

- ❖ **Propriétés de sûreté** : Expriment le fait qu'une mauvaise chose ne se produira jamais. Dans ce cas, cette propriété se traduit par l'assurance de l'exclusion mutuelle pour l'accès à la section critique par les processus.
- ❖ **Propriétés de vivacité** : Expriment le fait que la bonne chose attendue arrivera sûrement. Dans ce cas, un processus demandant l'accès à sa section critique finira par y accéder au bout de d'un temps fini.

SOLUTION NAÏVE

PROCESSUS INTERVENANT DANS L'APPLICATION:

Dans ce cas, ce sont :

- ❖ Les n processus qui résident dans les n sites représentent les processus de l'application.
- ❖ Un de ces processus jouera en plus le rôle du contrôleur d'accès à la section critique.

SOLUTION NAÏVE

EVÈNEMENTS:

Evènements attachés à chaque processus de l'application :

- **Evènements à l'initiative du processus**
 - **Demande d'accès à la section critique** : Cet événement déclenchera l'exécution d'une procédure dans laquelle une requête sera envoyée au processus contrôleur.
 - **Libération de la section critique** : Cet événement déclenchera l'exécution d'une procédure dans laquelle un message retour-permission sera envoyé au processus contrôleur.
- **Evènements subis par le processus :**
 - Réception d'un message permission d'accès à la section en critique en provenance du processus contrôleur.

Evènements attachés au processus contrôleur : Pour ce processus tous les événements sont externes (subis par le contrôleur).

- Réception d'un message requête en provenance d'un processus de l'application.
- Réception d'un message en provenance du processus de l'application qui vient de libérer sa SC.

SOLUTION NAÏVE

PROCÉDURES LIÉES AUX ÉVÉNEMENTS:

Procédure 1 : Procédure liée à la Demande d'accès à la section critique du processus P_i

Debut

Etati :=Demandeur ;

Send(requête(P_i)) to Contrôleur ;

Attendre(Etati=dedans)

Fin

Procédure 2 : Procédure liée à la Libération de la section critique du processus P_i

Debut

Etati :=Dehors ;

Send(retour-permission(P_i)) to Contrôleur ;

Fin

SOLUTION NAÏVE

PROCÉDURES LIÉES AUX ÉVÉNEMENTS (SUITE):

Procédure 3 : Procédure liée à la réception du message Permission par le processus Pi

Debut

Etati :=Dedans

Fin

Procédure 4 : Procédure liée à la réception du message requête(Pi) par le processus Contrôleur

Debut

Si file.vide() **alors**

Envoyer Permission à Pi

finsi

File.Enfiler(Pi) ;

Fin

SOLUTION NAÏVE

PROCÉDURES LIÉES AUX ÉVÉNEMENTS (SUITE):

Procédure 5 : Procédure liée à la réception du message retour-permission(P_i) par le processus Contrôleur

Debut

File.Defiler() ;

Si not file.vide() **alors**

P := File.tete() ;

Envoyer Permission à P ;

finsi

Fin

SOLUTION NAÏVE

VÉRIFICATION DE LA SATISFACTION DES PROPRIÉTÉS FONCTIONNELLES

La vérification dans ce cas est évidente. En effet :

- ❖ **Propriété de sûreté** : Le contrôleur est l'unique processus qui donne la permission d'accès à la section critique. Une fois que le contrôleur donne la permission à un processus, *Procédures 4 et 5*, il ne donnera une permission à un autre processus qu'après la réception d'un message retour-permission de la part du processus qui vient de quitter sa section critique. D'où l'assurance de la propriété d'exclusion mutuelle.
- ❖ **Propriété de vivacité** : Les processus, demandant la permission d'accès à la section critique au contrôleur, sont autorisés d'y accéder un à la fois selon une politique FIFO. Donc, tout processus demandeur finira par être en tête de file, donc autorisé à y accéder. La vivacité est garantie sous l'hypothèse que la résidence en section critique est de temps fini d'une part. D'autre part il est supposé que les liaisons de communication sont fiables (**H1**). Sans cette hypothèse, les messages requête, permission et retour-permission peuvent se perdre, ce qui compromet l'assurance de cette propriété.

SOLUTION NAÏVE

ETUDE DE CERTAINES PROPRIÉTÉS

Performance : Les performances de l'algorithme sont mesurées par:

- Le nombre de messages de contrôle mis en œuvre.
- Le temps durant lequel la section critique est libre en présence de processus demandeurs ainsi que la robustesse de la solution.
 - L'accès à la section critique par un processus nécessite 3 messages.
 - En supposant que le temps moyen de transit d'un message entre deux sites est égal à T , le temps moyen durant lequel la section critique est libre en présence de sites demandeurs d'accès à la section critique est de $2 \cdot T$. Le temps moyen est fonction de la topologie du réseau.

Robustesse: Cette solution est fragile dans le sens où la défaillance du contrôleur mettra en péril tout le système.

Distribution du contrôle : Cette solution ne peut être acceptée vu que le contrôle est centralisé au niveau d'un seul processus. On dit que le contrôleur assure toute la responsabilité alors que les autres sites n'ont n'assure aucune.

Passage à l'échelle : Dans le cas où le nombre de sites augmente, le site contrôleur peut se trouver submergé par la gestion de l'accès à la section critique de l'ensemble des processus.

SOLUTION BASÉE SUR LE JETON

Développer cette solution avec la participation des étudiants