

Examen semestriel

Algorithmique et Systèmes d'exploitation distribués (Corrigé)

1 H30

Exercice 1 (10 points) : Répondre aux questions suivantes.

Question 1 : Qu'est ce qui motive le recours aux architectures parallèles ?

Réponse :

- **Besoins de certaines applications** : Certaines applications (prévisions météo, reconnaissance de séquences ADN, conception aéronautique, ...) exigent une capacité de traitement qui peut atteindre un péta-flops et plusieurs giga-octets de mémoire. Ces applications sont impossibles à faire tourner sur de simples micro-ordinateurs; elles nécessitent des architectures parallèles.
- **Parallélisme de calculs** : Certaines applications se prêtent bien aux calculs parallèles, par exemple les applications numériques et les applications de calcul matriciel . Il est naturel de vouloir faire tourner ces applications sur des machines parallèles.

(2 points)

Question 2 : Il y'a quelques années , il était impossible de faire des prévisions météorologiques de plus de 3 jours. Expliquez pourquoi.

Réponse :

Le prévisions météorologiques sont un exemple typique d' applications nécessitant une grande capacité de traitement (par exemple météo-France emploie un supercalculateur d'une capacité de 1 péta-flops). Les besoins augmentent selon la durée de prévisions souhaitée (1 jour, 2jours, 3 jours, ou plus). Il y'a quelques années, il était impossible de faire des prévisions de plus de 3 jours , car cela dépassait les capacités de traitement des ordinateurs de l'époque..

(2 points)

Question 3 : Le Web peut-il être considéré comme un système réparti ? Justifiez.

Réponse :

Oui, le Web peut-être considéré comme un "bon" système distribué, car il vérifie bien le principe de "transparence" :

- **Transparence à la localisation (désignation)** : L'utilisateur n'est pas sensé savoir où se situe physiquement les ressources existant sur le Web.
- **Transparence d'accès**. L'utilisateur accède à une ressource locale ou distante d'une façon identique.
- **Transparence aux pannes (réseaux, machines, logiciels)**. Les pannes et réincarnations sont cachées à l'utilisateur.
- **Transparence à l'extension des ressources**. Le système peut être étendu ou réduit sans occasionner de gêne pour l'utilisateur .
- **Transparence à l'hétérogénéité**. L'utilisateur n'a pas à se soucier des différences matérielles ou logicielles des ressources qu'il utilise.

(2 points)

Question 4 : Pour gérer l'allocation des ressources dans un système réparti, deux méthodes sont possibles :

- Méthode 1 : On désigne un seul site allocateur qui est chargé de gérer toutes les ressources ; qui sont dispersées sur les différents sites.
- Méthode 2 : On choisit plutôt de répartir la gestion des ressources entre plusieurs allocateurs qui peuvent être présents sur des sites différents.

Donnez les avantages et les inconvénients de chaque méthode.

Réponse :

	Un seul allocateur	Plusieurs allocateurs
Avantages	Puisque toutes les demandes et libérations de ressources parviennent à un seul allocateur ; l'obtention d'un "état global d'allocation de ressources" ne pose pas de problème	- Répartition de la charge entre les allocateurs. - La panne d'un allocateur ne compromet pas la fonction d'allocation
Inconvénients	- Possible surcharge de l'allocateur - L'éventuelle panne de l'allocateur, compromettrait la fonction d'allocation. Elle exigerait une "élection" d'un nouveau allocateur.	Difficulté d'avoir un "état global des ressources"

(2 points)

Question 5 : Le problème du consensus est un problème réputé difficile dans les systèmes répartis. Expliquez brièvement de quoi il s'agit.

Réponse :

Le problème de consensus dans un système distribué consiste à chercher qu'une certaine décision commune soit prise par tous les acteurs. Par exemple, n processus, envoie un par un une décision "je suis d'accord" pour qu'un certain traitement soit entamé. Le traitement en question commence lorsque les n décisions identiques sont envoyées.

On peut montrer que le problème du consensus est problématique en raison des contraintes suivantes sur les décisions à prendre : leur acheminement, leur évolution, leur fiabilité, ... etc.

(2 points)

Exercice 2 (10 points) : On considère l'algorithme d'élection de processus suivant.

Chaque processus possède une capacité représentée par un nombre C_i . Le processus à élire doit posséder la plus grande capacité. Chaque processus peut être dans l'un des états suivants :

- « Actif » : le processus est en activité.
- « Candidat » : le processus est candidat à l'élection.
- « Perdu » : le processus n'a aucune chance d'être élu.
- « Elu » : le processus est élu. Il doit être proclamé.

Algorithme :

Initialement, tous les Processus sont dans l'état « Actif ».

Le processus initiateur de l'élection P_i diffuse le message $Elire(C_i, i)$ à tous les processus et passe à l'état « Candidat ».

A la réception d'un message $Elire(C_k, k)$, un processus P_i :

s'il est dans l'état « Actif » :

- il répond $Ack(i)$ à P_k si sa capacité C_i est inférieure ou égale à C_k .

- il émet $Elire(C_i; i)$ en diffusion si sa capacité C_i est strictement supérieure à C_k et passe dans l'état « Candidat »

s'il est dans l'état « **Candidat** » :

- il ignore ce message si sa capacité C_i est supérieure à C_k .
- il répond $Ack(i)$ à P_k si sa capacité C_i est inférieure ou égale à C_k et passe dans l'état « Perdu » (ne peut pas être élu, il supprime alors tous les Ack qu'il a déjà reçus).

s'il est dans l'état « **Perdu** » :

- il répond $Ack(i)$ à P_k si sa capacité C_i est inférieure ou égale à C_k .

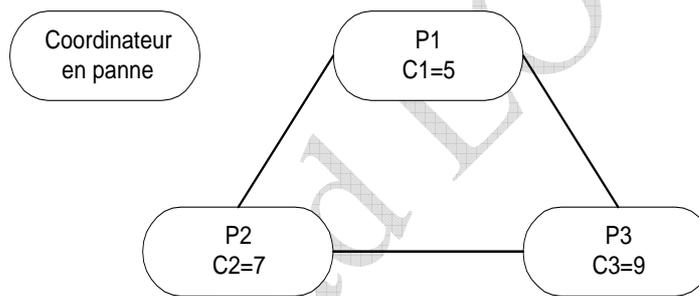
Question 1 : Avec cet algorithme, quand un processus peut-il savoir s'il est le nouvel élu ?

Réponse :

Lorsque le processus reçoit les Ack de tous les autres sites

(1 point)

Question 2 : La figure suivante représente un système où le processus P_1 a détecté que le processus coordinateur est tombé en panne .



Complétez le tableau suivant montrant le déroulement de l'algorithme pour l'élection d'un nouveau processus coordinateur.

Réponse :

Site	Etat	Message reçu	Nouvel état	Action
P1	Actif		Candidat	Diffuser $Elire(5, 1)$
P2	Actif	$Elire(5, 1)$	Candidat	Diffuser $Elire(7, 2)$
P3	Actif	$Elire(5, 1)$	Candidat	Diffuser $Elire(9, 3)$
P1	Candidat	$Elire(7, 2)$	Perdu	Envoyer $Ack(1)$ à P2
P1	Perdu	$Elire(9, 3)$	Perdu	Envoyer $Ack(1)$ à P3
P2	Candidat	$Ack(1)$	Candidat	-
P3	Candidat	$Ack(1)$	Candidat	-
P2	Candidat	$Elire(9, 3)$	Perdu	Envoyer $Ack(2)$ à P3
P3	Candidat	$Elire(7, 2)$	Candidat	-
P3	Candidat	$Ack(2)$	Elu	Proclamer

(4 point)

Question 3 : Calculer la complexité de cet algorithme, en nombre de messages. Justifiez.

Réponse :

Le pire des cas est le cas où la première demande d'élection est faite par le processus de plus petite capacité car cela peut entraîner une demande d'élection de tous les sites si ceux-ci la reçoivent avant tout autre message. Ce qui nous donne :

- N- 1 messages (diffusion initiale)
- Les N-1 autres sites émettent alors en diffusion, c'est-à-dire, chacun N-1 messages, soit au total : $(N-1)(N-1)$ messages
- Enfin il y a $0 + 1 + 2 + \dots + (N-1)$ acquittements, soit $N(N-1)/2$ messages.
- d'où au total $3N(N-1)/2$ messages

(02 points)

Question 4 : Proposez une méthode pour réaliser l'action « Proclamer ».

Réponse :

On peut proposer de diffuser un message spécial à tous les sites Proclamer(i) qui doit être émis par le processus élu (processus i).

(1 point)

Question 5 : Que doit faire un processus qui reprend son exécution après une panne ?. Quelles conséquences peuvent survenir après son action ?.

Réponse :

Tout processus P qui reprend son exécution après une défaillance doit réinitialiser le processus d'élection. Deux situations peuvent alors survenir :

- Si la capacité du processus P est inférieure à celle du coordinateur actuel , il n'y a aucun changement .
- Si la capacité du processus P est supérieure à celle du coordinateur actuel , celui-ci est relégué. Le nouveau coordinateur sera le processus P.

(2 points)