TD Tolérance aux fautes
INF722

auteur : Thomas Robert

Consignes

- Il vous est demandé de répondre aux questions en priorité sur le sujet de TD.

1 Rapide évaluation de vos connaissances (10 min maxi)

Voici un QCM dont l'objectif est de vérifier le niveau de votre maîtrise du vocabulaire du domaine. Indiquez la ou les réponses correctes pour chaque question qui suit.

Q.1. **QCM** ()

(a) La réplication passive permet de tolérer a priori

1. des défaillances byzantines
2. des défaillances en valeur
3. des défaillances par crash

Réponse :

(b) Le N versions programming est une manière de garantir

1. la reprise d'une exécution à partir d'une version précédente de l'état d'exécution
2. l'indépendance de l'activation des fautes par l'utilisation d'implémentations différentes d'une même fonction
3. la qualité du code produit grâce à l'usage d'un gestionnaire de versions (type git, svn ...)

Réponse :

(c) Dans le cadre d'un mécanisme de recouvrement avant, il est important de sauvegarder régulièrement l'état d'exécution (afin de le recharger plus tard)

1. vrai
2. faux

Réponse :

(d) On suppose que l'on utilise une bibliothèque dont on ne dispose pas du code source. A l'exécution, une instruction du code de la bibliothèque engendre de manière systématique une exception qui déclenche le signal SIGSEGV. On souhaiterait ajouter un mécanisme au projet pour tolérer cette situation et garantir un service minimum. Quel mécanisme n'améliorera en rien le fonctionnement de l'application ?

1. recouvrement avant
2. recouvrement arrière
3. N version programming

Réponse :

(e) Peut on implémenter le modèle de réplication active même si l'on ne connaît pas à priori le temps d'exécution maximum de la fonction à répliquer ?

1. oui
2. non

Réponse :

(f) Le masquage d'erreur consiste à écraser l'état erronée par une valeur par défaut

1. vrai
2. faux

Réponse :

(g) En supposant que l'on souhaite évaluer la fiabilité et la disponibilité d'un système. pour lequel de ces attributs est il a priori nécessaire de connaître le temps moyen passé dans l'état défaillant ?

1. fiabilité
2. disponibilité

Réponse :

(h) Un codage par blocs de type Hamming (7,4) (7 bits au total dont 4 de données) est-il adapté à la tolérance de plages de bits contigus altérés (taille ¿1):

1. vrai
2. faux

Réponse :

(i) Pour la réplication active à 5 répliques, supposons que seules les répliques peuvent défaillir et que l'état de chaque réplique est soit fonctionnel ou défaillant byzantin. En supposant que l'état des répliques est modélisé par des variables entières $(r_i),\ i \in \{1..5\}$ (1 si fonctionnel, 0 sinon). Quelle expression définit les états de fonctionnement fiable de l'architecture complète ?

1. $(\sum_{1 \leq i \leq 5} r_i) > 0$
2. $(\sum_{1 \leq i \leq 5} r_i) \geq 3$
3. $(\sum_{1 \leq i \leq 5} r_i) < 3$

Réponse :

Q.2. **Faute, erreur, défaillance** ()

Une faute d'interaction pour un système logiciel correspond souvent à l'utilisation du système dans un contexte non prévu ou avec des données ne correspondant pas au domaine de valeur pouvant être traitées normalement par le système.

Illustrez par un exemple concret, comment dans un système complexe (composés de plusieurs sous systèmes) la défaillance d'une partie du système peut être considérée comme une faute d'interaction par une autre partie du système ?

Réponse :

Q.3. **Réflexion sur les architectures répliquées** ()

Un ingénieur doit déployer une fonction de calcul numérique qui traite chaque entrée indépendamment mais nécessite une longue période de calcul avant de retourner son résultat. Quelle architecture de réplication est la plus adaptée (justifiez) si :

- l'on veut maîtriser le temps de réponse de l'architecture dans son ensemble
- l'on souhaite limiter la puissance de calcul consommée (imaginez que vous payez pour chaque unité de temps cpu consommée).

Réponse :

2 Exercices

Le but de cette partie est de vous faire raisonner sur des problèmes dont la solution repose sur l'usage de mécanismes de tolérance aux fautes ou sur la compréhension des fautes et de leurs conséquences.

2.1 Topologie réseau et tolérance

Cet exercice est un cas d'école. Supposons que l'on souhaite connecter N sites distants via un réseau filaire. Supposons que le réseau n'est pas déjà en place et que l'on doit décider des connexions physiques à déployer entre sites. Le but est que chaque paire de site soit capable de communiquer. De plus, nous allons considérer différentes exigences du point de vue de la tolérance aux fautes.

Les nœuds et les connexions physiques les reliant forment un graphe. Nous utiliserons ce modèle pour raisonner. La structure du graphe (i.e. quels sites sont directement reliés) est appelée topologie du réseau. Il est possible de communiquer entre 2 sites dès lors qu'il existe un chemin les reliant. Supposons que les sites sont numérotés de 1 à N et désignés respectivement s_1, ..,s_N. L'architecture nécessitant le moins de connexions pour relier tous les nœuds est une organisation linéaire : le nœud s_i est connecté à s_{i+1} pour i de 1 à $N-1$.

Le problème de cette organisation est qu'il suffit qu'un câble soit rompu pour "couper" le réseau en deux. Nous allons considérer deux alternatives de topologies : l'architecture complète où l'on connecte par un câble chaque paire de sites, et l'architecture en anneau qui étend l'architecture en ligne en connectant s_N à s_1.

Q.4. **Anneaux et plus** ()

L'architecture en anneau permet de tolérer exactement une rupture de câble. Un ingénieur prétend avoir trouvé un moyen de tolérer 2 fautes en ajoutant strictement moins de "partie entière de N/2 câbles supplémentaires". Illustrez que cette affirmation est incorrecte pour N=4. (i.e. l'architecture correspond donc à un anneau + un câble de plus).

Réponse :

Q.5. **Rupture de câble** ()

Q.6. **Rupture de câble** ()

Considérez le réseau complètement maillé à 4 nœds. Chaque nœd est connecté aux autres via 3 câbles. En supposant, que 3 câbles sont rompus sur l'ensemble de l'architecture, identifiez le scénario qui fait qu'au moins un nœds est déconnecté du réseau. Généralisez ce raisonnement au cas du réseau complet à N nœds et donnez le nombre de ruptures tolérables pour ce réseau ?

Réponse :

Q.7. **Maillage pour N=6** ()

Trouver un maillage qui utilise exactement 9 connexions et permette de tolérer deux ruptures de câbles pour N=6. Notez que le pire cas du réseau complet s'applique toujours.
Astuce : vous pouvez partir de la structure en anneau et la modifier (ajouter ou enlever des liens).

Réponse :

2.2 Etats de fiabilité / disponibilité d'une architecture

On suppose que l'on étudie une architecture qui implémente la réplication active à 3 répliques telle que un résultat n'est produit que si le voteur reçoit au moins 2 réponses identiques. Dans le cas où un nœud est identifié défaillant, il est mis en quarantaine et remplacé par un nœud de remplacement qui était jusque là inactif. Ceci est fait durant le traitement de la requête dans la limite des nœuds de remplacement disponibles et des capacités de détection de la réplication active. Le système démarre avec 2 nœuds de remplacement. Si aucune majorité ne peut être dégagée, la requête de calcul échoue. On supposera deux modes de défaillances distincts pour les répliques : en valeur, et par crash.

Q.8. ()

En introduisant la ou les variable(s) nécessaire(s) pour modéliser le système, définissez une formule identifiant les états de fonctionnement du système qui garantissent la fiabilité des résultats

Réponse :

Q.9. ()

Procédez de même pour les états de disponibilité

Réponse :

Q.10. ()

Proposez un état non fiable mais pour lequel le système reste disponible

Réponse :

Q.11. ()

Le système peut il être temporairement non fiable ? (en supposant que la défaillance en valeur peut être transitoire mais que le crash est permanent).

Réponse :

2.3 Tolérance aux fautes sur les données

Nous souhaitons mettre en application les principes vus en cours sur la tolérance aux donnée.

2.4 Tolérance aux fautes isolées

Q.12. **Code de Hamming (7,4)** ()

Le principe du code de Hamming (7,4) est de transformer un mot de 4 bits en un mot de 7 bits tel que la distance de Hamming entre chaque mot du code résultant soit maximale. Quelle est cette distance au mieux ?

Réponse :

Q.13. **Code de Hamming (7,4) et tolérance à l'effacement** ()

Supposons que l'on connaisse les bits qui sont altérés, que peut on dire de la capacité de tolérance aux fautes d'un tel code ?

Réponse :

2.5 Tolérance aux "Burst"

Un burst est une suite de bits altérés de manière aléatoire. Nous ne nous attarderons pas sur l'aspect aléatoire. Nous retiendrons le pire des cas : "*si N bits peuvent être altérés de manière contiguë, alors cela signifie que ces N bits peuvent se voir inversés (ou laissés identiques) sans que l'on sache a priori ce qui a été réalisé.*"

Il est possible d'améliorer les capacité de tolérance aux faute d'un code sur mot de taille fixe en tirant parti du fait que l'on ait une séquence de mots à émettre. Il suffit de :

- découper chaque mot du code en un nombre identique de sous blocs
- émettre le premier sous bloc d'un ensemble de K mots de la séquence de mots à transmettre. Puis de transmettre le second sous bloc de ce même ensemble et ainsi de suite jusqu'à ce que l'ensemble des sous blocs d'un mot du code soient transmis pour chaque message.

Ainsi, si vos mots sont composés de deux blocs : $M_0 = B_0^1 B_0^2$, et $M_1 = B_1^1 B_1^2$. L'entrelacement serait d'émettre : $B_0^1 B_1^1 B_0^2 B_1^2$.

Q.14. **Entrelacement pour tolérance aux ”burst”** ()

On suppose que l'on utilise un codage capable de tolérer sur chaque bloc r fautes, et que chaque message a une taille de 64 bits. Proposez un entrelacement qui permette de tolérer un burst de 16 bits altérés consécutifs.

Réponse :

Q.15. **Entrelacement, le cas extrème** ()

Expliquez pourquoi on peut dire que l'on peut toujours compenser le taux de tolérance du codage par bloc ? (i.e. il existe toujours un entrelacement permettant de tolérer k bits altérés consécutifs)

Réponse :