
Contrôle des connaissances INF722

auteur : Thomas Robert

Consignes

- Il vous est demandé de répondre aux questions sur votre copie (et non pas le sujet)
- L'examen est prévu pour une durée de 2h.
- Le seul document autorisé est le compte rendu de TD corrigé qui vous aura été remis juste avant l'examen
- Les téléphones portables, ordinateurs, ou calculatrices sont interdits durant l'examen.

1 Questions rapides

Q.1. QCM

(5 points)

Voici un QCM. Il est possible que certaines questions aient pour bonne réponse plus d'un choix. Dans ce cas, il vous est demandé d'indiquer l'ensemble des bonnes réponses. Chaque question compte pour 0.5.

- (a) Une donnée incorrecte fournie en entrée d'un composant est considéré pour ce composant comme
- une défaillance en valeur.
 - une faute d'interaction.
 - une défaillance byzantine.
- (b) On suppose une topologie de réseau en anneau liant N machines différentes nommées m_0, \dots, m_{N-1} telle que pour tout i , $0 \leq i \leq N-2$, m_i est connecté à m_{i+1} , et m_{N-1} est connecté à m_0 . Sur chaque connexion, les communications sont bi-directionnelles. Cette topologie peut supporter au maximum un certain nombre de rupture de câble sans qu'aucune paire de noeuds ne se retrouve déconnectée (considérez le pire cas):
- aucune rupture de câble tolérée.
 - 1 rupture de câble tolérée.
 - 2 ruptures de câble tolérée.
- (c) (TD) La distance de Hamming est :
1. un opérateur de comparaison de vecteurs de bits.
 2. un code correcteur d'erreur.
 3. un mécanisme de détection d'erreur sur des données stockées.

- (d) Pour quel(s) mécanisme(s) le N version programming est-il utilisé afin d'assurer un bon fonctionnement du mécanisme de tolérance aux fautes
1. la réplication active.
 2. le recovery block.
 3. la réplication passive.
- (e) (TD) 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.
- (f) il est possible de combiner recouvrement arrière et avant pour tenter de tolérer des fautes de développement sur une fonction logicielle. Laquelle des combinaisons suivante fait sens pour une faute dont on ne connaît pas les conditions d'activation a priori
- application d'un recouvrement arrière, puis d'un recouvrement avant si l'erreur n'a pas été corrigée via le recouvrement arrière.
 - application d'un recouvrement avant, puis d'un recouvrement arrière si l'erreur n'a pas été corrigé via le recouvrement avant.
- (g) L'évaluation du risque correspond à l'évaluation :
1. des conséquences d'un incident.
 2. de la combinaison de la vraisemblance et de l'impact d'un incident.
 3. d'une probabilité d'occurrence d'une défaillance.
- (h) Une contrainte de sécurité innocuité a pour objectif :
- de garantir l'absence de défaillance.
 - de garantir la capacité du système à redevenir fonctionnel après défaillance.
 - de maîtriser la gravité des défaillances du système.
- (i) Est-on obligé de définir une contrainte de temps de réponse pour que le modèle de défaillance crash soit correctement défini ?
- vrai.
 - faux.
- (j) En supposant un modèle de faute byzantin, combien de défaillances peuvent être tolérées si l'on utilise 7 répliques.
- 1
 - 3
 - 5
 - 6

2 Exercices Architecture complexes (5 pts)

Un ingénieur souhaite utiliser un mécanisme de réplication active reposant sur 9 répliques pour améliorer la fiabilité de composants peu fiables. Il doit décider si il utilise l'architecture A ou B (cf figure 1). Dans un premier temps on considère que le composant "votant" stocke les réponses de toutes les répliques qui lui sont connectées avant d'émettre son résultat.

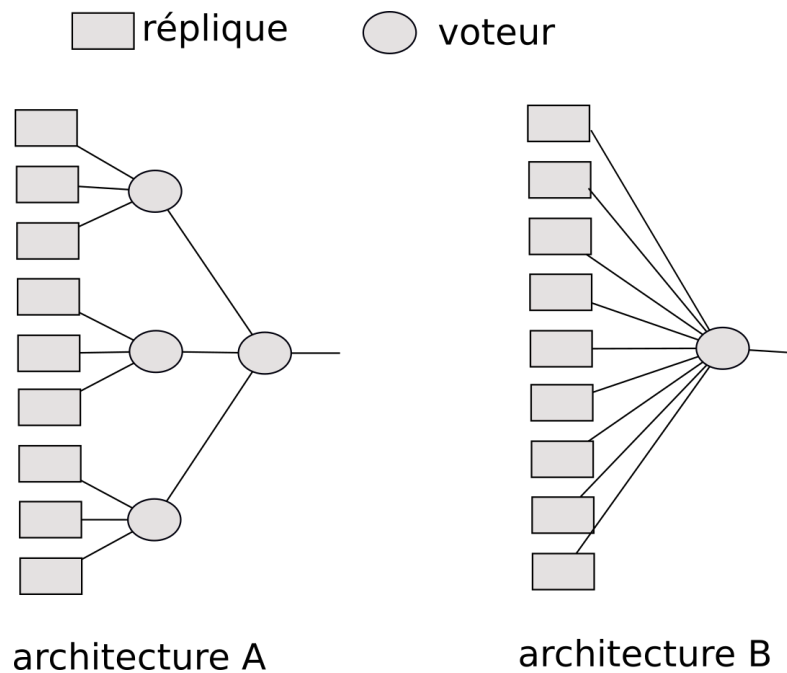


Figure 1: Architectures possibles

Le résultat émis correspond à la valeur majoritairement reçue (ou la première valeur reçue si elles ont toutes la même fréquence, e.g. 1 par exemple).

Q.2. Architecture A (1 point)

Combien de défaillance en valeur peut-on tolérer dans l'architecture A dans le pire cas ?

Q.3. Architecture B (1 point)

Combien de défaillance en valeur peut-on tolérer dans l'architecture B dans le pire cas ?

Q.4. Défaillances temporelles (1 point)

On suppose que les répliques peuvent défaillir de deux manières différentes : en valeur, ou en produisant un résultat en retard par rapport au temps de réponse de 10 ms attendu pour le calcul réalisé par une réplique. Le composant "voteur" doit maintenant traiter ces défaillances temporelles en plus du vote. Indiquez comment vous procéderiez ?

Q.5. États de fiabilité (2 points)

En supposant, que l'on dispose de deux variables indiquant le nombre de défaillances en valeur et temporelle par sollicitation de l'architecture (i.e. par calcul lancé sur l'architecture). Vous noterez ces variables n_{DeV} et n_{DeT} respectivement). Définissez une formule caractérisant les états de fiabilité (indication : la formule appartient à la logique des prédicats – à cause du type entier des variables).

3 Topologie réseau et tolérance (5 pts)

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 sites soit capable de communiquer.

Les sites 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, \dots, 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.

Q.6. (TD) Anneaux et plus (1 point)

Un ingénieur prétend avoir trouvé un moyen de tolérer 2 fautes en ajoutant à une structure en anneaux strictement moins de "partie entière de $N/2$ câbles supplémentaires". Illustrez que cette affirmation est incorrecte pour $N=4$.

Q.7. (TD) Le bon degré de connexion ($1\frac{1}{2}$ points)

Proposez une propriété sur le nombre d'arcs reliés à un site qui permette d'exprimer une condition nécessaire et suffisante permettant de tolérer k ruptures de câbles. (notez nc_i le nombre de câbles ayant le site s_i pour extrémité par exemple). Justifiez la propriété en expliquant le pire cas.

Q.8. Maillage pour $N=6$ (1 point)

Trouver un maillage qui utilise exactement 9 connexions et permette de tolérer deux ruptures de câbles pour $N=6$.

Q.9. Connexions Directionnelle et reconfiguration ($1\frac{1}{2}$ points)

Prenez $N=3$, On suppose désormais que les connexions sont directionnelles. Pour relier s_i à s_{i+1} , on utilise deux arcs : un de s_i à s_{i+1} et un de s_{i+1} vers s_i . Les messages ne peuvent circuler que dans le sens des arcs. Combien de ruptures de câbles cette architecture peut elle tolérer (en maintenant la capacité d'émettre et recevoir des messages pour chaque nœuds)? Est ce que le fait de pouvoir changer une fois le sens de communication d'un lien améliore ce nombre ? (justifiez).

4 Tolérance aux fautes sur les données (5 pts)

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

Q.10. (TD) Code de Hamming (7,4) (1 point)

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 pour ce code (7,4) ?

Un burst est une suite de bits altérés de manière aléatoire (pas nécessairement une inversion de valeur mais on remplace une séquence de bit par une autre). Le problème ici est que l'on ne connaît pas a priori la nature de l'altération réalisée (inversion ou mise à une valeur prédéfinie), ni la position à laquelle l'altération a lieu.

Il est possible d'améliorer les capacités de tolérance aux fautes d'un code sur une donnée correspondant à un message de taille fixe N en tirant parti du fait que l'on ait une séquence de P messages à émettre, M_1, \dots, M_P . Il suffit de :

- découper chaque message M_i du code en un nombre identique de sous blocs, par exemple k , $M_i(1) \dots M_i(k)$.
- Puis réaliser l'émission comme suis :
 Pour j de 1 à k , répéter :
 Pour i de 1 à P répéter :
 émettre $M_i(j)$

Q.11. Entrelacement pour tolérance aux "burst" (1 1/2 points)

On suppose que l'on utilise un codage capable de tolérer sur chaque message 9 bits de valeur erronée avant entrelacement (potentiellement altérés), et que chaque message a une taille de $N=64$ bits. Peut-on tolérer 10 bits consécutifs altérés à partir du premier bit émis pour le premier message émis si on prend $k=8$, et $P=2$? **Justifiez votre réponse**

Q.12. Entrelacement le cas extrême (2 1/2 points)

En supposant que la taille de chaque message est $N=k*s$ (i.e. on découpe le message en k blocs de taille identique), et que le codage utilisé pour chaque message possède (avant découpe et entrelacement) une capacité à tolérer r bits altérés.

(a) Illustrez par un schéma représentant la séquence de bits émis le scénario où :

- les bits sont altérés dès le premier bit émis, q bits sont altérés.
- les paramètres s, k, r, P ont les valeurs suivantes : $s=16, k=4, r=20, P=2$.

Quelle valeur peut prendre q au maximum dans ce scénario sans empêcher la correction des bits altérés ? **(on ne vous demande pas de faire un schéma où vous numéroter chaque bit mais de donner suffisamment de détails pour justifier la réponse concernant le paramètre q)**

(b) Donnez la formule générale qui permet de connaître la valeur maximale de q pouvant être tolérée en fonction de tout ou partie des paramètres suivants : s, k, r, P ? (vous pouvez utiliser les notations suivantes $\lfloor f \rfloor$ pour désigner la partie entière de la fraction f).