H24 :

Question 1 :
Un ordinateur A envoie un message à B à 11h11m11.150s pour obtenir le temps et reçoit une réponse à 11h11m11.600s, ces deux temps étant mesurés avec l'horloge de A. L'ordinateur B reçoit la requête de A à 11h11m05.200s et retourne sa réponse à A à 11h11m05.500s, ces deux temps étant mesurés avec l'horloge de B. Quel est le décalage (ajustement) à appliquer sur A? Donnez la réponse en secondes (avec trois décimales). Utilisez la virgule (,) pour séparer la partie entière de la partie décimale.
T1 = 11:11:11.150 → = 11.150 s (temps local A - envoi)
T4 = 11:11:11.600 → = 11.600 s (temps local A - réception)
T2 = 11:11:05.200 → = 5.200 s (temps local B - réception)
T3 = 11:11:05.500 → = 5.500 s (temps local B - envoi)
1. Temps aller-retour (RTT) selon A :
RTT=T4−T1=11.600−11.150=0.450 s
2. Temps de traitement à B :
Durée chez B=T3−T2=5.500−5.200=0.300 s
3. Temps de transmission (aller + retour) :
Transmission=RTT−Durée chez B=0.450−0.300=0.150 s
4. Supposant transmission symétrique → aller = retour :
Temps aller=Temps retour=0.150/2=0.075 s
5. Décalage (offset) :
Offset=T2−(T1+Temps aller)=5.200−(11.150+0.075)=5.200−11.225=−6.025 s

Question 2 :
Quel est la précision associé? Donnez la réponse en secondes (avec trois décimales). Utilisez la virgule (,) pour séparer la partie entière de la partie décimale.
Hypothèse de symétrie → précision = ± (temps de transmission / 2) :
Précision = 0,150 / 2 = 0,075 s

Question 3 :
Un service de base de données réparti est offert par 3 serveurs redondants. Au moins un serveur doit être disponible pour que le service soit disponible. Chaque serveur est constitué d'un boîtier et son électronique, avec une probabilité de disponibilité de 0.65, ainsi que d'un ensemble de disques en RAID, 5 disques dont au moins 3 doivent être fonctionnels. La probabilité d'être fonctionnel pour un disque est de 0.75. Quelle est la probabilité qu'un ensemble de disques en RAID soit fonctionnel? Donnez la réponse avec cinq décimales. Utilisez la virgule (,) pour séparer la partie entière de la partie décimale.
Nombre total de disques : n = 5
Seuil minimum de disques fonctionnels : k = 3
Probabilité de fonctionnement d'un disque : p = 0,75
Probabilité d'échec : q = 1 − p = 0,25
Calculer la probabilité qu'au moins 3 disques soient fonctionnels :
P=P(3)+P(4)+P(5)
Formule de probabilité binomiale
P(k)=(n k)·p ^(k)·q^(n−k)
• P(3)=(5 3)·0,75^(3)·0,25^(5-3) = 10·0,421875·0,0625 = 0,263672
• P(4)=(5 4)·0,75^(4)·0,25^(5-4) = 5·0,31640625·0,25 = 0,395508
• P(5)=(5 5)·0,75^(5)·0,25^(5-5) = 1·0,2373046875 = 0,237305
P_RAID OK = 0,263672+0,395508+0,237305 = 0,89649

Question 4 :
Quelle est la probabilité qu'un serveur de base de données soit fonctionnel? Donnez la réponse avec cinq décimales. Utilisez la virgule (,) pour séparer la partie entière de la partie décimale.
Probabilité que le boîtier soit fonctionnel : P_boîtier = 0,65
Probabilité que l'ensemble de disques RAID soit fonctionnel (de Q3) : P_RAID = 0,896484376
P_serveur OK = P_boîtier * P_RAID = 0,65 * 0,89649 = 0,582714844

Question 5 :
Quelle est la probabilité que le service de base de données réparti soit fonctionnel? Donnez la réponse avec cinq décimales. Utilisez la virgule (,) pour séparer la partie entière de la partie décimale.
Probabilité qu'un serveur soit fonctionnel : p = 0,582714844 (de Q4)
Nombre de serveurs : 3
Le service fonctionne si au moins 1 des 3 serveurs fonctionne
Probabilité qu'un serveur soit en panne :
q = 1−p=1−0,582714844=0,417285156
Probabilité que les 3 serveurs soient en panne :
q^3 = (0,417285156)^3 = 0,072660571
Probabilité que le service fonctionne (au moins un serveur OK) :
1−q^3 =1−0,072660571= 0,92733

Question 6 :
Les transactions T1, T2, T3 et T4 s'exécutent en même temps et leurs opérations de lecture et d'écriture sur des variables (x1, x2, ... x6) sont entrelacées. Les lectures d'une transaction sont effectuées sur les versions courantes des variables, et les écritures d'une transaction sont effectuées sur une version provisoire des variables pour la transaction. Lorsque la transaction se termine et est acceptée, la version provisoire des variables écrites par la transaction devient la version courante. Une validation de la cohérence par contrôle optimiste de la concurrence est effectuée pour accepter ou non chaque transaction. Une transaction est acceptée s'il n'y a aucun conflit sur les variables accédées (selon les critères de la méthode choisie) et est refusée s'il y a conflit sur certaines variables. Il faut tenir compte des transactions précédentes qui ont été validées (et ignorer celles qui ne l'ont pas été) dans le calcul des conflits pour savoir si chacune des transactions est acceptée ou non.

1	T1:	Begin	15	T4:	Write x2
2	T1:	Read x2	16	T1:	Write x1
3	T1:	Write x5	17	T1:	End
4	T2:	Begin	18	T2:	Read x6
5	T1:	Read x5	19	T3:	Read x3
6	T1:	Write x1	20	T2:	Write x3
7	T3:	Begin	21	T2:	Write x5
8	T1:	Read x4	22	T2:	End
9	T1:	Read x6	23	T4:	Read x5
10	T3:	Write x3	24	T3:	Write x4
11	T1:	Write x2	25	T3:	End
12	T4:	Begin	26	T4:	Read x6
13	T4:	Read x3	27	T4:	Write x3
14	T4:	Read x2	28	T4:	End

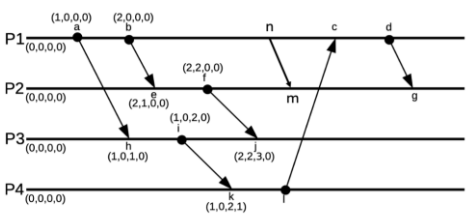
Si une validation en reculant pour vérifier la cohérence de la transaction T1 est utilisée, pourrait-elle être validée ? Si la transaction T1 n'est pas validée, quelles sont les variables en conflit.
a. T1 est validée, il n'y a pas de conflit pour les variables utilisées.
b. T1 est validée, il y a un conflit pour la variable x6.
c. T1 n'est pas validée, il y a un conflit pour la variable x6.
d. T1 n'est pas validée, il y a un conflit pour les variable x2 et x6.
Justification : Une transaction T peut être validée si aucune transaction validée avant elle n'a écrit sur une variable que T a lue.
T1 lit : x2 (ligne 2), x5 (ligne 5), x4 (ligne 8), x6 (ligne 9)
T1 écrit (provisoirement jusqu'à validation) : x5 (ligne 3), x1 (ligne 6, 16), x2 (ligne 11)
T1 se termine ligne 17.
Transactions à considérer pour conflits : seulement celles validées avant T1 : Aucune !, T2 commence à la ligne 4, mais se termine à la ligne 22 → après T1., T3 commence à la ligne 7, se termine à la ligne 25 → après T1., T4 commence à la ligne 12, se termine à la ligne 28 → après T1.

Question 7 :
Si une validation en reculant pour vérifier la cohérence de la transaction T2 est utilisée, pourrait-elle être validée ? Si la transaction T2 n'est pas validée, quelles sont les variables en conflit.
a. T2 est validée, il n'y a pas de conflit pour les variables utilisées.
b. T2 est validée, il y a un conflit pour la variable x6.
c. T2 n'est pas validée, il y a un conflit pour la variable x6
d. T2 n'est pas validée, il y a un conflit pour les variables x1 et x5.
Justification : T2 lit : x6 (ligne 18)
T2 écrit (provisoirement jusqu'à validation) : x3 (ligne 20), x5 (ligne 21)
T2 se termine ligne 22
Transactions à considérer (validées avant la fin de T2) :
• T1, validée ligne 17 (donc avant T2)
• T1 écrit : x5 (ligne 3), x1 (ligne 6/16), x2 (ligne 11)
Comparer écritures de T1 avec lectures de T2 :
T2 lit x6 → pas de conflit, car T1 n'écrit pas x6

Question 8 :
Si une validation en reculant pour vérifier la cohérence de la transaction T3 est utilisée, pourrait-elle être validée ? Si la transaction T3 n'est pas validée, quelles sont les variables en conflit
a. T3 est validée, il n'y a pas de conflit pour les variables utilisées.
b. T3 est validée, il y a un conflit pour la variable x3.
c. T3 n'est pas validée, il y a un conflit pour la variable x3
d. T3 n'est pas validée, il y a un conflit pour les variables x2 et x5.
Justification : T3 lit : x3 (ligne 19)
T3 écrit (provisoirement jusqu'à validation) : x3 (ligne 10), x4 (ligne 24)
T3 se termine ligne 25
Transactions validées avant la fin de T3 :
• T1 (fin ligne 17)
• T2 (fin ligne 22)
Comparer les écritures de T1 et T2 avec les lectures de T3 :
• T3 lit x3 (ligne 19)
• T1 n'écrit pas x3
• T2 écrit x3 (ligne 20) → donc conflit

Question 9 :
Si une validation en reculant pour vérifier la cohérence de la transaction T4 est utilisée, pourrait-elle être validée ? Si la transaction T4 n'est pas validée, quelles sont les variables en conflit
a. T4 est validée, il n'y a pas de conflit pour les variables utilisées.
b. T4 est validée, il y a un conflit pour la variable x3.
c. T3 n'est pas validée, il y a un conflit pour la variable x2, x3 et x5.
d. T3 n'est pas validée, il y a un conflit pour la variable x2 et x3
Justification : T4 lit : x3 (ligne 13), x2 (ligne 14), x5 (ligne 23), x6 (ligne 26)
T4 écrit : x2 (ligne 15), x3 (ligne 27)
T4 se termine ligne 28
Transactions validées avant la fin de T4 :
• T1 (ligne 17)
• T2 (ligne 22)
• T3 (ligne 25)
Comparer écritures précédentes avec les lectures de T4 :
• x3 lu par T4 (ligne 13)
• T2 écrit x3 (ligne 20) → conflit
• T3 écrit x3 (ligne 10) → conflit
• x2 lu par T4 (ligne 14)
• T1 écrit x2 (ligne 11) → conflit
• x5 lu par T4 (ligne 23)
• T1 écrit x5 (ligne 3)
• T2 écrit x5 (ligne 21) → conflit

Question 10 :
Quatre processus, P1, P2, P3 et P4, démarrent à peu près en même temps et s'envoient des messages. Chaque processus maintient un vecteur de 4 compteurs d'événements, avec une entrée correspondante à chaque processus. Chaque processus, au moment d'envoyer un message, incrémente son compteur d'événements dans son vecteur et joint le vecteur au message. Chaque processus, lorsqu'il reçoit un message, incrémente son compteur d'événements dans son vecteur et fusionne son vecteur avec celui reçu dans le message. Donnez finalement les vecteurs de compteurs d'événements pour les points c et g.



Quel est le vecteur pour le point c
Donnez la réponse comme : (valeur1,valeur2,valeur3,valeur4)
Méthode générale : Pour trouver le vecteur à un point donné :
1. On part du dernier événement connu sur le même processus.
2. À chaque événement local, on incrémente la case correspondant au processus.
3. Quand un message est reçu, on fusionne le vecteur reçu avec celui local (on prend le maximum valeur par valeur), puis on incrémente sa propre case.
Réponse : (4,0,2,2) : On fusionne (1,0,2,2) avec (4,0,0,0)

Question 11 :
Quel est le vecteur pour le point g.
Donnez la réponse comme : (valeur1,valeur2,valeur3,valeur4).
Réponse : (5,4,2,2) : La 3ème transaction de P3 n'est pas comtablilisé.

Question 12 :
Trois ordinateurs offrent un service. Chaque ordinateur fait en moyenne une panne après 10 jours d'opération et cette panne prend 10 heures à réparer. Quelle est la disponibilité de ce système? Donnez la réponse avec 5 chiffres décimales et utilisez la virgule pour séparer (exemple: 0,12345)
Réponse : Temps de fonctionnement moyen : 10 jours = 10 × 24 = 240 h
Temps total de cycle = 240h (fonctionnement) + 10h (réparation) = 250h
Indisponibilité d'un ordinateur = 10 / 250 = 0,04
Indisponibilité du système = 0,04³ = 0,000064
(car le système est indisponible seulement si les 3 tombent en panne en même temps)
Disponibilité du système = 1 − 0,000064 = 0,999936

A24 :

Question 1 :
a) Un ordinateur A envoie un message à B à 9h30m00.000s pour obtenir le temps et reçoit une réponse à 9h30m00.800s, ces deux temps étant mesurés avec l'horloge de A. L'ordinateur B reçoit la requête de A à 9h30m02.250s et retourne sa réponse à A à 9h30m02.650s, ces deux temps étant mesurés avec l'horloge de B. Quel est le décalage à appliquer sur A? Quel est l'intervalle d'incertitude associé?
a = 9h30m02.250s − 9h30m00.000s = 2.250s
b = 9h30m02.650s − 9h30m00.800s = 1.850s
Ajustement = (a + b) / 2 = (2.250s + 1.850s) / 2 = 2.050s
Précision = (a − b) / 2 = (2.250 − 1.850s) / 2 = 0.200s
Le décalage à appliquer à A est de 2.050s et l'incertitude est de +/- 0.200s.
b) Un groupe de 36 processus sont reliés entre eux sous la forme d'un anneau. Si on déclenche une élection en anneau, combien de messages doivent être envoyés autour de l'anneau i) au minimum et ii) au maximum avant d'identifier le processus élu? iii) Combien de messages s'ajoutent pour que l'élection soit complétée et le résultat communiqué à tous les processus?
Si le processus qui déclenche l'élection est celui le plus prioritaire, au bout de 36 messages le vote sera conclu et il saura qu'il a gagné. Par contre, si le processus qui suit le plus prioritaire est celui qui déclenche l'élection, il faudra 35 messages avant que celui plus prioritaire ne vote et ensuite 36 messages pour qu'il se sache élu.
Le minimum est donc i) 36 et le maximum ii) 35 + 36 = 71. Ensuite, il faut iii) 36 messages pour faire le tour et annoncer à tous le résultat de l'élection.
c) Un système réparti avec plusieurs processus utilise des horloges logiques. À chaque événement dans un processus, son horloge est incrémentée de 1. Lorsqu'un message est envoyé, la valeur de l'horloge est jointe au message. Lorsqu'un message est reçu, l'horloge logique du processus en réception prend le maximum de son horloge et de la valeur jointe avec le message. Si on compare la valeur de l'horloge logique pour deux événements sur deux processus différents, on se demande ce qu'on peut dire de l'ordre relatif de ces événements. i) Si un événement est postérieur à un autre avec un lien de causalité qui démontre la postériorité, quelle sera la relation entre les valeurs d'horloges logiques? ii) Si un événement est postérieur à un autre sans lien de causalité qui démontre la postériorité, est-ce que son horloge logique sera nécessairement plus grande?
Pour démontrer causalement qu'un événement d'un premier processus est postérieur à un autre sur un second processus, il faut qu'il soit postérieur sur le même processus, ou qu'il soit sur un deuxième processus après la réception d'un message venant du premier processus. L'incrémentation des horloges logiques suit ces liens de causalité (incrément à chaque événement sur le même processus et incrément par maximum lors d'un envoi de message). Ainsi, pour un message qui a un lien de causalité qui en démontre la postériorité, l'horloge de l'événement postérieur sera plus grande que celle du premier événement. S'il n'y a pas de lien de causalité entre deux événements, la valeur de l'horloge logique de l'événement postérieur peut être inférieure, égale ou plus grande que celle du premier événement, selon comment l'horloge logique est incrémentée dans chaque processus séparé.

Question 2 :
a) Un nouveau conte de Noël, intitulé Le sapin de Pollux, vient d'être rendu disponible à l'adresse www.pollux.org. L'adresse de www.pollux.org est disponible du serveur de noms pour le domaine pollux.org avec un TTL (Time To Live) de 2 minutes. L'adresse du serveur de noms pour pollux.org est disponible sur les 13 serveurs racine de l'Internet avec un TTL de 24h. De nombreux clients des 39 fournisseurs Internet officiels de Noël, hohoho.ca, hohoho.fr, hohoho.it..., veulent accéder le conte à cette adresse, www.pollux.org. Ces clients passent toujours par le serveur de noms de leur fournisseur Internet pour obtenir l'adresse cherchée, ils n'ont pas de cache local. Le serveur de noms de chaque fournisseur Internet fait des accès récursifs, de manière à maintenir en cache, pour toute leur période de validité, toutes les entrées recherchées par ses clients. On suppose que les requêtes de ces fournisseurs Internet sont uniformément réparties entre les serveurs de noms racine. Chacun de ces 39 fournisseurs Internet possède exactement 2880 clients, et chaque client accède au conte une fois le jour de sa sortie. Les requêtes sont réparties uniformément sur les 24 heures du jour de sortie de ce conte.
i) Combien de ces requêtes pour le serveur de noms de pollux.org est-ce que chaque serveur racine recevra en moyenne pour ce jour de sortie du conte? ii) Combien de requêtes par minute est-ce que chaque serveur de nom d'un de ces fournisseurs Internet recevra pour www.pollux.org pendant cette journée de sortie du conte? iii) Combien de requêtes pour www.pollux.org est-ce que le serveur de noms du domaine pollux.org recevra pendant cette journée de sortie du conte?
i) Le serveur de noms de chacun des 39 fournisseurs Internet demandera une fois l'adresse du serveur de noms pour le domaine pollux.org.
Ces requêtes seront réparties entre les 13 serveurs racine, ce qui donnera 39 / 13 = 3 requêtes pour chaque serveur racine.
ii) Puisque chacun des 2880 clients d'un fournisseur fera cette requête une fois pendant la journée, cela donne 2880 / (24 heures x 60 minutes) = 2 requêtes par minute.
iii) Chaque serveur de noms d'un fournisseur Internet fera une requête par 2 minutes, soit 30 requêtes / heure au serveur de noms du domaine pollux.org. Le nombre de requêtes pour la journée sera donc de 39 x 24 heures x 30 requêtes / heure = 28080 requêtes.

b) Une entreprise veut installer un service LDAP redondant et très sécuritaire. Pour ce faire, chaque client peut interroger un ou plusieurs de 6 serveurs redondants. Supposons que chaque serveur a une probabilité d'être disponible de 0.45. i) Si ces serveurs présentent des pannes par omission, combien de serveurs en panne peut-on tolérer et quelle est la probabilité que le service soit non disponible au client? ii) Si les pannes sont de type réponse aléatoire et le client prend un vote parmi les réponses des 6 serveurs, combien de serveurs en panne peut-on tolérer, et que devient la probabilité que le service soit non disponible?

i) Avec des pannes par omission, on peut tolérer jusqu'à 5 serveurs en panne. La probabilité que les 6 serveurs soient en panne est de (1−0.45)^6 = 0.027680641.

ii) Avec des pannes de réponse aléatoire, il faut avoir au moins 2 serveurs disponibles.

Nous avons une probabilité de (1−0.45)^6 = 0.027680641 d'avoir les 6 serveurs en panne, et de 6 × (1−0.45)^5 × 0.45^1 = 0.135886781 d'avoir exactement 5 serveurs en panne et 1 disponible.

Ceci représente les cas où le service sera non disponible, une probabilité totale de 0.027680641 + 0.135886781 = 0.163567422.

Les autres cas, 2 à 6 serveurs disponibles, font que nous avons au moins 2 serveurs disponibles et le service fonctionne.

Question 3 :

a) Les transactions T, U et V s'exécutent en même temps et leurs opérations de lecture et d'écriture sur des variables (a, b, c, d, e) sont entrelacées. Les lectures d'une transaction sont effectuées sur les versions courantes des variables, et les écritures d'une transaction sont effectuées sur une version provisoire des variables pour la transaction. Lorsque la transaction se termine et est acceptée, la version provisoire des variables écrites par la transaction devient la version courante. Une validation de la cohérence par contrôle optimiste de la concurrence est effectuée pour accepter ou non chaque transaction. Il faut tenir compte des transactions précédentes qui ont été validées (et ignorer celles qui ne l'ont pas été) pour savoir si chacune des transactions est acceptée ou non. Lesquelles des transactions T, U et V pourraient être validées, i) Si une validation en reculant était utilisée pour vérifier la cohérence des transactions? ii) Une validation en avançant? Pour chaque transaction non validée, donnez-la ou les variables en conflit.

```
 1 T: Début                    10 T: Compléter
 2 T: Read(a)                  11 U: Read(b)
 3 T: Write(b,1)               12 V: Read(a)
 4 T: Write(d,3)               13 V: Read(d)
 5 U: Début                    14 V: Read(c)
 6 U: Read(a)                  15 V: Write(c,3)
 7 U: Read(e)                  16 V: Write(e,7)
 8 U: Write(c,2)               17 V: Compléter
 9 V: Début                    18 U: Compléter
```

Pour la validation en reculant i), au moment où T termine, il ne peut y avoir de problème et T est accepté.

Pour compléter V, on examine ce qui est lu (a, c, d) versus ce qui a été écrit par T (b, d) et V ne peut compléter (conflit sur d).

Rendu à compléter U, ce qui est lu (a, b, e) intersecte avec T (b, d) et U doit aussi être abandonné (conflit sur b).

Pour la validation en avançant ii), au moment où T termine, ce que T a écrit (b, d) n'intersecte pas avec ce que U a lu (a, e) ni V (rien encore) et T peut compléter.

Au moment de compléter V, les écritures (c, e) intersectent avec les lectures de U (a, b, e) et V ne peut compléter (conflit sur e).

La dernière transaction, U, peut nécessairement compléter.

b) Une transaction répartie T, gérée par le coordonnateur C, veut écrire les variables a=2 et b=3 sur le serveur s1, c=22 et d=33 sur le serveur s2, et e=12 et f=24 sur le serveur s3.

Cette transaction répartie est commise en utilisant le protocole de fin de transaction atomique à deux phases.

i) Quels seront les messages échangés entre le coordonnateur C et le serveur s3 pour ce protocole de fin de transaction à 2 phases?

ii) Quelles sont les entrées qui seront ajoutées au journal du serveur s3 pour cette transaction, et à quel moment par rapport aux messages échangés avec le coordonnateur?

Le coordonnateur C va envoyer un message à tous les participants, incluant s3, pour savoir s'ils sont prêts à accepter la transaction (canCommit?). Le serveur s3 répondra qu'il est prêt après avoir écrit dans son journal les nouvelles valeurs écrites et le fait qu'il est prêt à commettre cette transaction (P1: write(e, 12); P2: write(f, 24); P3: Prepare T(P1, P2) P0;). Si tous les serveurs répondent positivement au coordonnateur, celui-ci procédera avec la seconde phase et annoncera à tous, incluant s3, que la transaction est acceptée (doCommit). Le serveur s3 écrira dans son journal que la transaction est acceptée (P4: Completer T, P3) et répondra au coordonnateur C que c'est fait (haveCommitted).

Question 4 :

a) Un service de base de données réparti est offert par 3 serveurs redondants. Au moins un serveur doit être disponible pour que le service soit disponible. Chaque serveur est constitué d'un boîtier et son électronique, avec une probabilité de disponibilité de 0.6, ainsi que d'un ensemble de disques en RAID, 5 disques dont au moins 3 doivent être fonctionnels. La probabilité d'être fonctionnel pour un disque est de 0.7. i) Quelle est la probabilité qu'un ensemble de disques en RAID soit fonctionnel? ii) Un serveur de base de données? iii) Le service de base de données réparti?

i) L'ensemble de disques RAID a une probabilité de fonctionner de 0.7^5 = 0.16807 (5 disques fonctionnels) plus 5! / ((5−4)! × 4!) × 0.7^4 × (1−0.7)^1 = 0.36015 (exactement 4 disques fonctionnels) plus 5! / ((5−3)! × 3!) × 0.7^3 × (1−0.7)^2 = 0.3087 (exactement 3 disques fonctionnels), pour un total de 0.16807 + 0.36015 + 0.3087 = 0.83692.

ii) Un serveur est opérationnel si le boîtier est fonctionnel de même que son ensemble de disques RAID, ce qui donne 0.6 × 0.83692 = 0.502152.

iii) Le service sera disponible sauf si les 3 serveurs sont en panne, une probabilité de 1 − (1 − 0.502152)^3 = 0.876607063.

b) Sur un réseau qui ne supporte pas la multi-diffusion, un message de groupe atomique est envoyé d'un membre M aux autres. Le groupe compte 28 membres. i) Si aucun message n'est perdu, combien de messages envoyés au total sur le réseau seront requis pour compléter ce message atomique?

ii) Si un des messages envoyés par M à un autre participant est perdu, comment est-ce que cela peut être détecté?

i) M envoie un message aux 27 autres membres et attend les 27 réponses positives. Ensuite, il envoie la confirmation aux 27 autres membres.

Le total est donc de 3 × 27 = 81 messages.

ii) Si un message de préparation est perdu, M détectera un problème lorsqu'il ne recevra pas de réponse positive ou négative. Si la réponse est perdue, la détection se fera de même. Si un message de confirmation est perdu, le membre qui ne reçoit pas détectera qu'il a reçu un message de préparation, mais pas de message de confirmation correspondant.

Question 5 :

Vous devez planifier l'installation d'un nouveau centre de données au Québec. Ce centre regroupera des milliers de serveurs afin d'offrir des services infonuagiques.

a) En tant qu'ingénieur, quelles sont les lois applicables pour un tel projet, en ce qui concerne les aspects de développement durable?

L'ingénieur doit tenir compte des conséquences de l'exécution de ses travaux sur l'environnement dans une perspective de développement durable.

Ceci est prescrit par la loi canadienne de 2008 sur le développement durable, la loi québécoise de 2006 sur le développement durable, et le code de déontologie de l'Ordre des Ingénieurs du Québec.

La procédure d'évaluation environnementale du BAPE ne s'applique normalement pas à un centre de données.

b) Outre le volet purement financier, dans une approche de développement durable et d'analyse du cycle de vie, quels sont les quatre niveaux pour lesquels on doit évaluer l'impact environnemental d'un tel projet?

Pour toutes les activités qui se retrouvent dans les différentes phases du cycle de vie d'un projet, il faut examiner l'impact sur la santé humaine, sur l'écologie, sur les changements climatiques et sur l'appauvrissement des ressources.

c) Pour un projet de centre de données, quels sont typiquement les impacts environnementaux les plus importants pour chacune des phases du cycle de vie?

Les phases du cycle de vie sont la fabrication (construction du site et fabrication des équipements), le transport (des matériaux pour la construction et des équipements), l'opération du site (entretien, alimentation électrique...) et finalement la fin de vie (la démolition ou conversion du bâtiment, la restauration du site, et le recyclage ou la mise aux rebuts des matériaux et équipements). Pour un centre de données typique, une grande partie de l'impact est reliée à la consommation d'énergie pendant l'opération de l'équipement informatique et de la climatisation. Pour un centre au Québec, étant donné la disponibilité de l'énergie hydroélectrique, l'impact de la consommation d'électricité, sans être négligeable, est moindre que dans la plupart des autres endroits. Un autre impact important est la fabrication des équipements informatiques et électriques, en particulier le raffinage de l'or et du cuivre requis pour ces équipements. La construction du site, le transport et le démantèlement du site à la fin ont normalement un impact beaucoup moins grand que l'opération et que la fabrication des équipements.

A23 :

Question 1 :

a) Un client utilise le service DNS et peut contacter 4 serveurs différents. Chaque serveur a une probabilité d'être disponible de 0.6.

i) Si ces serveurs présentent des pannes par omission, combien de serveurs en panne peut-on tolérer et quelle est la probabilité que le service soit disponible au client?

ii) Si les pannes sont de type réponse aléatoire et le client prend un vote parmi les réponses des 4 serveurs, combien de serveurs en panne peut-on tolérer, et que devient la disponibilité du service?

i) Avec des pannes par omission, on peut tolérer jusqu'à 3 serveurs en panne. La probabilité que les 4 serveurs soient en panne est de 0.4^4 = 0.0256.

La probabilité de disponibilité est donc 1 − 0.0256 = 0.9744.

ii) Avec des pannes de réponse aléatoire, il faut avoir au moins 2 serveurs disponibles. Nous avons une probabilité de 0.6^4 = 0.1296 d'avoir les 4 serveurs disponibles, de 4 × 0.6^3 × (1−0.6)^1 = 0.3456 d'avoir exactement 3 serveurs disponibles, et de 4! / (2!(4−2)!) × 0.6^2 × (1−0.6)^2 = 0.3456 d'avoir exactement 2 serveurs fonctionnels. Le total est de 0.1296 + 0.3456 + 0.3456 = 0.8208. On peut aussi calculer la probabilité d'avoir exactement 3 serveurs en panne : 4 × 0.4^3 × (1−0.4)^1 = 0.1536, et les 4 serveurs en panne : 0.4^4 = 0.0256. La probabilité d'avoir moins de 2 serveurs fonctionnels est donc 0.0256 + 0.1536 = 0.1792. La probabilité d'avoir 2 ou plus serveurs disponibles est donc de 1 − 0.1792 = 0.8208. Cette autre manière de faire le calcul donne exactement le même résultat.

b) i) Quelles sont les différences entre les services de noms DNS et LDAP?

ii) Quelle est la relation entre les services de noms LDAP et X.500, lequel est le plus populaire?

Le service de nom DNS a été conçu spécifiquement pour convertir les adresses textuelles hiérarchiques de noms de domaines en adresses IP. C'est un service un peu ancien, très spécifique, mais qui a été conçu pour offrir une bonne tolérance aux pannes et une bonne performance en raison des caches. Il est uniquement utilisé pour la conversion d'adresses IP.

Le service LDAP est un service de noms beaucoup plus générique, qui accepte des schémas de données flexibles et permet des opérations de recherche avancées. Il est beaucoup utilisé pour accéder aux données des utilisateurs des systèmes informatiques (code usager, mot de passe, paramètres...), mais peut contenir des informations très variées.

LDAP s'inspire de X.500 mais se veut beaucoup moins compliqué.

Pratiquement personne n'utilise X.500.

Question 2 :

a) Un client C envoie un message au serveur S pour lui demander l'heure exacte. Ce message est envoyé à 16h05m00.200s et la réponse est reçue à 16h05m00.662s, heure du client. La réponse indique que l'heure exacte du serveur, au moment où il a répondu, était de 16h05m06.222s.

i) En utilisant l'algorithme de Christian, calculez le décalage à appliquer à l'heure du client et donnez l'intervalle d'incertitude sur cette valeur de décalage?

ii) Si le serveur avait fourni l'information additionnelle suivante, réception de la demande à 16h05m06.111 et envoi de la réponse à 16h05m06.333s, en utilisant l'algorithme de NTP qui tire parti de cette information, que deviendrait le calcul de décalage et l'intervalle d'incertitude associé?

On suppose que la moitié de l'intervalle, entre les deux temps sur le client, est pour que le message se rende au serveur et l'autre moitié est prise pour acheminer la réponse. L'incertitude est l'intervalle : 16h05m00.662s − 16h05m00.200s = 0.462s, soit +/- 0.231s. La nouvelle heure devrait être augmentée du temps pour acheminer la réponse : 16h05m06.222s + 0.231s = 16h05m06.453s. Le décalage à appliquer est donc de 16h05m06.453s − 16h05m00.662s = 5.791s.

Avec la méthode NTP, on a plutôt :

a = 16h05m06.333s − 16h05m00.662s = 5.671s

b = 16h05m06.111s − 16h05m00.200s = 5.911s

Ajustement = (a + b) / 2 = (5.671s + 5.911s) / 2 = 5.791s

Précision = (a − b) / 2 = (5.671 − 5.911s) / 2 = +/- 0.12s

Le décalage à appliquer au client est de 5.791s et l'incert. est de +/- 0.12s.

b) Un groupe de 21 processus, p1 à p21, utilisent un serveur central d'exclusion mutuelle. Les 21 processus demandent en même temps (e.g., à 16h00m00.000s) un même verrou v1. i) Que peut-on dire de l'ordre dans lequel ces 21 processus obtiendront le verrou? ii) Combien de messages seront échangés au total pour que tous ces processus obtiennent éventuellement le verrou demandé? On suppose qu'il n'y a pas de message perdu.

i) Si les horloges ne sont pas bien synchronisées, les demandes ne seront pas émises exactement en même temps, et l'ordre dans lequel elles sont envoyées dépendra de leur décalage de temps.

Même si tous les processus demandent le verrou en même temps, ces requêtes seront sérialisées sur le réseau et arriveront dans un certain ordre, un peu aléatoire, au serveur qui pourra les traiter séquentiellement.

ii) Un message est requis pour qu'un processus demande le verrou, il reçoit un message de réponse du serveur qui le lui accorde, et le client enverra finalement un message lorsqu'il en a terminé avec le verrou.

Le processus suivant en ligne, qui avait déjà envoyé une demande, recevra alors un message du serveur qui lui accorde le verrou.

On voit donc qu'il faut 3 messages pour chaque client (demande, obtention, cession). Le nombre total de messages échangés est donc de 21 × 3 = 63 messages. Si des accusés de réception sont utilisés, car le délai n'est pas facilement prévisible, soit pour obtenir la réponse suite à une demande, ou soit pour que le client relâche le verrou après son obtention, alors le nombre de messages pourrait être doublé à 126.

c) Le soleil est à son zénith à 12h00, temps universel coordonné, au centre du fuseau horaire, et cela ne varie pas vraiment d'une année à l'autre. Est-ce parce que la durée d'une rotation est exactement de 24h × 60min × 60s et ne varie jamais? Est-ce parce que la durée des secondes est ajustée pour compenser les variations dans la durée des rotations? Est-ce une autre raison?

La durée des jours varie légèrement, notamment en fonction des marées et du déplacement du noyau liquide au centre de la Terre.

La durée des secondes est fixe. L'ajustement se fait en ajoutant ou en retranchant une seconde au besoin, au 30 juin ou au 31 décembre de chaque année.

Question 3 :

a) Les transactions T1, T2, T3 et T4 s'exécutent en même temps et leurs opérations de lecture et d'écriture sur des variables (x1, x2, x3, x4, x5 et x6) sont entrelacées. Les lectures d'une transaction sont effectuées sur les versions courantes des variables, et les écritures d'une transaction sont effectuées sur une version provisoire des variables pour la transaction. Lorsque la transaction se termine et est acceptée, la version provisoire des variables écrites par la transaction devient la version courante.

Une validation de la cohérence par contrôle optimiste de la concurrence est effectuée pour accepter ou non chaque transaction. Il faut tenir compte des transactions précédentes qui ont été validées (et ignorer celles qui ne l'ont pas été) pour savoir si chacune des transactions est acceptée ou non. Lesquelles des transactions T1, T2, T3 et T4 pourraient être validées, si une validation en reculant était utilisée pour vérifier la cohérence des transactions? Pour chaque transaction non validée, donnez la ou les variables en conflit.

```
 1 T1: Begin                   13 T3: Write(x4)
 2 T1: Read(x5)                14 T1: End
 3 T1: Write(x4)               15 T3: Write(x6)
 4 T1: Read(x1)                16 T4: Read(x4)
 5 T2: Begin                   17 T2: Read(x5)
 6 T1: Read(x2)                18 T2: End
 7 T2: Write(x6)               19 T4: Read(x5)
 8 T3: Begin                   20 T4: Write(x6)
 9 T3: Write(x4)               21 T3: End
10 T3: Write(x3)               22 T4: Read(x3)
11 T4: Begin                   23 T4: Write(x5)
12 T3: Read(x4)                24 T4: End
```

T1: End - read (x1, x2, x5), write (x4), reculant → validé

T2: End - read (x5), write (x6), reculant → validé

T3: End - read (x4), write (x3, x4, x6), reculant → non validé (conflit x4)

T4: End - read (x3, x4, x5), write (x5, x6), reculant → non validé (conflit x4)

b) Une transaction répartie T, gérée par le coordonnateur s0, veut écrire les variables a = 25 sur le serveur s1, b = 33 sur le serveur s2, c = 14 sur le serveur s3 et d = 9 sur le serveur s4. Cette transaction répartie est commise en utilisant le protocole de fin de transaction atomique à deux phases.

i) Quelles seront les entrées ajoutées au journal du coordonnateur et des serveurs s1 et s2, en lien avec cette transaction T?

ii) Qu'est-ce qui arrive si s1 plante et redémarre avant la première phase de fin de transaction? Après la première phase?

s0 (coordonnateur)	s1	s2
P0: ... P1: Prepare T(s1, s2, s3, s4), P0 P2: Completer T, P1	P0: ... P1: write(a, 25) P2: Prepare T(a: P1), P0 P3: Completer T, P2	P0: ... P1: write(b, 33) P2: Prepare T(b: P1), P0 P3: Completer T, P2

ii) Le coordonnateur s0 va écrire les participants à la transaction puis va écrire que la transaction est complétée après avoir reçu un vote positif de chaque participant pendant la première phase.

Il pourra ensuite annoncer que la transaction est complétée à chacun des participants et au client.

Chacun des participants écrit les variables modifiées, ensuite écrit "prépare" lorsqu'on lui demande s'il est prêt à accepter la transaction, et finalement écrit "complété" lorsqu'il reçoit la confirmation du coordonnateur.

Question 4 :

a) Une compagnie, qui offre des services infonuagiques, met gratuitement à la disposition des usagers un service de fichiers (semblable à Google Drive). Cette compagnie opère 30 000 serveurs, constitués en 10 000 groupes de 3 serveurs en redondance. En effet, les fichiers de chaque usager sont dupliqués sur 3 serveurs différents, les 3 serveurs d'un groupe, pour assurer une bonne tolérance aux pannes. Chaque serveur contient une unité RAID de 3 disques. Pour chaque serveur, la probabilité de panne, hormis les disques, est de 0.2. Pour chaque disque, la probabilité de panne est de 0.3, et l'unité RAID peut tolérer un disque en panne sur les 3.

i) Quelle est la probabilité qu'un serveur donné soit disponible?

ii) Quelle est la probabilité, pour un usager donné, que ses fichiers soient disponibles?
iii) Quelle est l'équation (par exemple en fonction de la réponse à i ou ii) qui donne la probabilité que le service soit entièrement disponible pour tous les usagers?
i) L'unité de disques RAID a une probabilité de fonctionner de 0.7^3 = 0.343 (3 disques fonctionnels) plus 3! / ((3−2)! × 2!) × 0.7^2 × (1−0.7)^1 = 0.441 (exactement 2 disques fonctionnels), pour un total de 0.343 + 0.441 = 0.784.
Un serveur est opérationnel si l'unité RAID fonctionne et le reste du serveur aussi, ce qui donne 0.8 × 0.784 = 0.6272.
Le service sera disponible pour un utilisateur sauf si les 3 serveurs du groupe qui le sert sont en panne, ce qui donne une probabilité de disponibilité de 1 − (1 − 0.6272)^3 = 0.948188316 que le groupe de l'usager soit fonctionnel.
Le service sera disponible pour tous les usagers si tous les groupes de 3 serveurs sont fonctionnels, ce qui donne une probabilité de 0.948188316^10000, qui est pratiquement nulle.
b) Lors du TP4, vous avez été amené à configurer un serveur DNS pour la gestion de la zone "polymtl.ca". Dans les TPs précédents, le fichier "hosts" a été souvent utilisé pour faciliter la communication entre les différents conteneurs. Pouvez-vous décrire de manière détaillée comment la mise en place d'un service DNS est plus avantageuse que l'utilisation du fichier hosts? Quelles sont les limites de l'utilisation du fichier "hosts"?
Si on utilise le fichier /etc/hosts, beaucoup d'informations sur la configuration réseau doit s'y trouver, venant possiblement de plusieurs domaines, et peut devoir être ajustée d'un conteneur à l'autre. Si le service DNS est plutôt utilisé, chaque conteneur sera plus générique (il se connecte simplement au serveur DNS), et son image est donc plus simple à générer. Les paramètres spécifiques seront centralisés au niveau du serveur DNS de chaque domaine.
Question 5 :
a) Qu'est-ce que le développement durable? Dans une approche d'étude du cycle de vie, quelles sont les phases pour lesquelles faire l'étude et quels sont les 4 principaux aspects à évaluer?
Le développement durable est : "un développement qui répond aux besoins du présent sans compromettre la capacité des générations futures à répondre à leurs propres besoins".
Il faut évaluer l'impact d'un projet ou d'un produit sur l'ensemble de son cycle de vie, du début à la fin : fabrication à partir des matières premières, transport / distribution, opération, fin de vie / démantèlement / disposition / recyclage.
Les 4 principaux aspects à évaluer sont : les impacts sur la santé humaine,l'écologie, l'appauvrissement des ressources, les changements climatiques.
b) Quelles sont les responsabilités de l'ingénieur concernant les aspects de développement durable d'un projet?
Quels sont les lois et règlements applicables au développement durable pour un ingénieur?
Le Québec a voté une loi sur le développement durable en 2006, et le Canada en 2008.
De plus, le BAPE est appelé à évaluer beaucoup de gros projets comme les centrales, les barrages ou les aéroports.
La loi sur la responsabilité élargie des producteurs a aussi un impact sur la production de certains types de biens comme les produits électroniques, les batteries et les peintures.
Finalement, le code de déontologie de l'Ordre des Ingénieurs du Québec stipule clairement que l'ingénieur doit tenir compte du développement durable dans l'exercice de ses fonctions.
c) Pour un projet de centre de données, quels sont les deux éléments qui en général ont le plus gros impact sur l'environnement?
Les deux éléments qui ont le plus gros impact au niveau du développement durable pour un centre de données sont généralement : la consommation d'électricité pendant son opération, la fabrication des équipements informatiques et électriques, particulièrement les équipements informatiques qui doivent être renouvelés aux 3 à 5 ans.

A22 :

Question 1 :
a) Le service de noms de domaines (DNS) et le service de répertoire de noms par le protocole LDAP sont deux services de noms fréquemment utilisés. Quelles sont les informations qui sont typiquement placées dans chacun, dans les différentes organisations comme Polytechnique? Dans le protocole LDAP, lors d'une recherche, un temps de traitement et un volume maximal de données sont spécifiés, ce qui n'est pas le cas pour le DNS.
Quelle est l'utilité de ces valeurs maximales? Pourquoi ce n'est pas aussi utile pour le DNS?
Le service de noms de domaines permet essentiellement de convertir les noms de domaines en adresses IP. Il permet aussi de faire la conversion inverse (nom pour une adresse IP). Quelques autres informations peuvent aussi être contenues comme l'adresse du serveur de courriel pour un domaine ou le type d'un ordinateur.
LDAP contient la plupart des autres informations requises pour l'accès à un système, comme : les noms d'usagers et leur mot de passe (encrypté), les groupes, les privilèges de chaque usager, la localisation du répertoire maison de chaque usager, et potentiellement beaucoup d'autres informations.
Étant donné que LDAP permet de mettre toutes sortes d'informations, certaines requêtes pourraient retourner un grand volume d'information ou prendre beaucoup de temps à chercher dans une grande arborescence, même si après filtrage le volume retourné est petit. Dans ce contexte, mettre un quota sur le temps et sur le volume est utile. Avec le DNS, il n'y a pas de recherche dans une arborescence aussi étoffée, ni de filtre, et le résultat retourné est normalement très court (une adresse IP de quelques octets). C'est donc peu utile d'avoir un quota sur le temps ou le volume.
b) Un serveur DNS local s'exécute sur votre poste de travail afin de maintenir un cache local. En effet, de nombreux processus sur votre poste de travail font des requêtes DNS qui peuvent souvent se répéter. Lorsque ce serveur local reçoit une requête, il prend 3 ms de cœur de CPU et peut la servir dans 40% des cas. Autrement, il doit en plus faire une demande récursive au serveur DNS de votre fournisseur Internet, ce qui ajoute le temps de cette requête récursive à son temps de réponse.
Le serveur DNS de votre fournisseur répond en 10 ms dans 65% des cas et en 20 ms dans 35% des cas. Quel est le nombre maximal de requêtes par seconde que peut soutenir le serveur local, s'il n'utilise qu'un seul fil d'exécution? S'il utilise de nombreux fils d'exécution et que 2 cœurs de CPU sont dédiés à ce processus serveur local?
Dans tous les cas, une requête prend 3 ms de cœur de CPU.
Dans 0.6 × 0.65 des cas, 10 ms s'ajoutent, alors que dans 0.6 × 0.35 des cas, l'ajout est de 20 ms.Le temps moyen est donc :
3 ms + 0.6 × 0.65 × 10 ms + 0.6 × 0.35 × 20 ms = 11.1 ms
Avec un seul fil d'exécution :
1000 ms/s ÷ 11.1 ms/r = 90.09 requêtes/seconde
Si plusieurs fils d'exécution et deux cœurs sont disponibles :
Chaque requête consomme 3 ms de cœur CPU → 1000 ms/s ÷ 3 ms/r = 333.33 r/s par cœur Donc avec deux cœurs : 666.66 requêtes/seconde
Question 2 :
a) Un ordinateur A envoie un message à B à 11h11m11.150s pour obtenir le temps et reçoit une réponse à 11h11m11.600s, ces deux temps étant mesurés avec l'horloge de A. L'ordinateur B reçoit la requête de A à 11h11m05.200s et retourne sa réponse à A à 11h11m05.500s, ces deux temps étant mesurés avec l'horloge de B. Quel est le décalage à appliquer sur A? Quel est l'intervalle d'incertitude associé?
a = 11h11m05.200s − 11h11m11.150s = -5.950s
b = 11h11m05.500s − 11h11m11.600s = -6.100s
Ajustement = (a + b) / 2 = (-5.950s - 6.100s) / 2 = -6.025s
Précision = (a - b) / 2 = (-5.950s + 6.100s) / 2 = 0.075s
Le décalage à appliquer à A est de -6.025s et l'incertitude est de +/- 0.075s.
b) Un groupe de 25 processus qui communiquent par message sont connectés en anneau. L'algorithme d'élection en anneau est utilisé.
Quel est le nombre minimal de messages requis pour compléter une élection? Le nombre maximal?
Le meilleur cas est lorsque le processus de plus haute priorité amorce l'élection. Après 25 messages, il saura qu'il est élu.
Ensuite, après un autre 25 messages, il saura que tous ont appris la nouvelle, pour un total de 50 messages.
Le pire cas est si le processus suivant de celui de plus haute priorité amorce l'élection.
Au bout de 24 messages, celui de haute priorité recevra le message d'élection.
Après 25 autres messages, il saura qu'il est élu.
Finalement, 25 autres messages complèteront la propagation de la nouvelle, pour un total de 74 messages.
c) L'algorithme de l'élection hiérarchique permet de réaliser efficacement une élection. Néanmoins, des algorithmes beaucoup plus complexes comme Paxos et Raft ont été développés pour réaliser des élections.
Quels sont les problèmes de l'algorithme de l'élection hiérarchique que l'algorithme Raft permet de pallier?
Le principal problème de cet algorithme est qu'il n'a pas de notion de quorum ou de majorité absolue.
En cas de partition de réseau, il est donc possible d'avoir deux processus qui sont élus, un dans chacune des deux partitions.
Pour plusieurs applications comme l'exclusion mutuelle ou les transactions, ceci n'est pas acceptable.
Raft ne peut élire un processus que si une majorité absolue accepte sa candidature.
Ceci empêche l'élection de plus d'un processus.
En plus, Raft est mieux conçu pour : les cas de processus qui tombent en panne pendant l'élection, empêcher que tous les processus ne déclenchent une élection en même temps.
Question 3 :
a) Les transactions T, U, V et W s'exécutent en même temps et leurs opérations de lecture et d'écriture sur des variables (a, b, c, d, e, f, g) sont entrelacées. Les lectures d'une transaction sont effectuées sur les versions courantes des variables, et les écritures d'une transaction sont effectuées sur une version provisoire des variables pour la transaction. Lorsque la transaction se termine et est acceptée, la version provisoire des variables écrites par la transaction devient la version courante. Une validation de la cohérence par contrôle optimiste de la concurrence est effectuée pour accepter ou non chaque transaction. Il faut tenir compte des transactions précédentes qui ont été validées (et ignorer celles qui ne l'ont pas été) pour savoir si chacune des transactions est acceptée ou non. Lesquelles des transactions T, U, V et W pourraient être validées si une validation en avançant était utilisée pour vérifier la cohérence des transactions? Pour chaque transaction non validée. donnez-la ou les variables en conflit.

```
 1 T: Début                    13 T: Compléter
 2 U: Début                    14 W: Read(f)
 3 T: Read(a)                  15 U: Write(c)
 4 T: Read(b)                  16 U: Write(d)
 5 U: Read(c)                  17 U: Compléter
 6 U: Read(d)                  18 V: Read(b)
 7 V: Début                    19 W: Write(a)
 8 V: Read(e)                  20 W: Write(b)
 9 T: Write(b)                 21 W: Compléter
10 T: Write(f)                 22 V: Read(g)
11 W: Début                    23 V: Write(e)
12 W: Read(e)                  24 V: Compléter
```

Au moment de compléter T, il faut vérifier ce que T a écrit (b, f) versus les variables lues par U (c, d), V (e) et W (e).
Il n'y a pas d'intersection → T est validée.
Au moment de compléter U, il faut vérifier ce que U a écrit (c, d) versus les variables lues par V (e) et W (e, f).
Il n'y a pas d'intersection → U est validée.
Au moment de compléter W, il faut vérifier ce que W a écrit (a, b) versus les variables lues par V (b, e).
Il y a intersection sur b → W est annulée.
La dernière transaction, V, peut compléter : il n'y a plus de transaction ultérieure → V est validée.
b) Une transaction distribuée T, sur 4 serveurs (s1, s2, s3, s4), effectue des opérations sur les variables (a, b, c, d).Chaque variable est répliquée sur deux serveurs, et est dénotée a1 pour la réplique de la variable a qui réside sur le serveur s1. Les opérations effectuées sont :
read a1; read b2; read c3; read d4; write a1; write a2; write b2; write b3; write d4; write d1. Ensuite, la transaction répartie est commise en utilisant le protocole de fin de transaction atomique à deux phases.
Quelles seront les entrées ajoutées au journal de chacun des serveurs s1, s2, s3 et s4 en lien avec cette transaction T?
Pour s1: P1: écrire a1; P2 écrire d1; P3 Préparer T(P1, P2) P0; P4 Compléter T, P3;
Pour s2: P1: écrire a2; P2 écrire b2; P3 Préparer T(P1, P2) P0; P4 Compléter T, P3;
Pour s3: P1: écrire b3; P2 Préparer T(P1) P0; P3 Compléter T, P2;
Pour s4: P1: écrire d4; P2 Préparer T(P1) P0; P3 Compléter T, P2;
Question 4 :
a) Un service de base de données réparti est offert par 3 serveurs redondants. Au moins un serveur doit être disponible pour que le service soit disponible. Chaque serveur est constitué d'un boîtier et son électronique, avec une probabilité de disponibilité de 0.65, ainsi que d'un ensemble de disques en RAID, 5 disques dont au moins 3 doivent être fonctionnels. La probabilité d'être fonctionnel pour un disque est de 0.75. Quelle est la probabilité qu'un ensemble de disques en RAID soit fonctionnel? Un serveur de base de données? Le service de base de données réparti?
L'ensemble de disques RAID a une probabilité de fonctionner de 0.755 = 0.237304688 (5 disques fonctionnels) plus 5!/((5 −4)!4!)×0.754 ×(1−0.75)5−4 = 0.395507813 (exactement 4 disques fonctionnels) plus 5!/((5− 3)!3!)×0.753×(1−0.75)5−3 =0.263671875 (exactement 3 disques fonctionnels), pour un total de 0.237304688+ 0.395507813+0.263671875 = 0.896484376. Un serveur est opérationnel si le boîtier est fonctionnel de même que son ensemble de disques RAID, ce qui donne 0.65×0.896484376 = 0.582714844. Le service sera disponible sauf si les 3 serveurs sont en panne, une probabilité de 1−(1−0.582714844)3 = 0.927339429.
b) Considérez la base de données créée avec les commandes suivantes, sur un serveur Postgres configuré de la même manière que lors du travail pratique TP5. Après les commandes pour créer la table ops2, et insérer les valeurs initiales, deux consoles sont utilisées pour entrer deux transactions concurrentes (montrées comme Transaction 1 et Transaction 2). La transaction 2 demande d'imprimer sum(amount) à deux reprises. Quelles sont les valeurs imprimées qui ont ici été remplacées par XXXX et YYYY dans la retranscription fournie?

```
postgres=#CREATE DATABASE bank;
postgres=# \connect bank;
bank=# CREATE TABLE ops2 (id int, amount float, PRIMARY KEY (id));
bank=# INSERT INTO ops2 VALUES (1,-100);
bank=# INSERT INTO ops2 VALUES (2,+150);
bank=# INSERT INTO ops2 VALUES (3,-22.2);

Temps Transaction 1     Transaction 2

1                       bank=# BEGIN TRANSACTION ISOLATION LEVEL REPEATABLE READ;
2                       bank=#  SELECT sum(amount) FROM ops2;
                        sum
                        ------
                        XXXX

3     bank=# BEGIN;
4     bank=# INSERT into ops2 VALUES (4,150);
5     bank=# COMMIT;

6                       bank=#  SELECT sum(amount) FROM ops2;
                        sum
                        ------
                        YYYY
```

Dans le moded'isolation REPEATABLE READ utilisé par la transaction 2, lorsque la même valeur est lue plus d'une fois dans la même transaction, la même valeur est obtenue à chaque fois. La mise à jour, effectuée aux temps 3 à 5 par l'autre transaction, ne peut donc pas être prise en compte dans ce mode. Le résultat pour XXXX est donc-100 + 150- 22.2 = 27.8, et il est le même pour YYYY.
Question 5 :
Vous devez planifier un nouveau centre de données et comparer différents scénarios. Vous avez déjà prévu examiner la performance des systèmes (et les revenus qu'on peut en tirer), le coût des ordinateurs, le coût du bâtiment ainsi que les coûts d'opération et de renouvellement des équipements, afin de déterminer le projet le plus rentable sur la durée de vie anticipée. Devez-vous comme ingénieur aussi vérifier les aspects du développement durable de ce projet? i) Quelles sont les lois applicables? ii) Quelles sont les différentes phases du cycle de vie? iii) Quels sont les quatre différents types d'impact sur l'environnement et ceux les plus importants typiquement pour un centre de données dans chaque phase?
L'ingénieur doit tenir compte des conséquences de l'exécution de ses travaux sur l'environnement dans une perspective de développement durable.
i) Ceci est prescrit par la loi canadienne de 2008 sur le développement durable, la loi québécoise de 2006 sur le développement durable, et le code de déontologie de l'Ordre des Ingénieurs du Québec. La procédure d'évaluation environnementale du BAPE ne s'applique normalement pas à un centre de données.
ii) Les phases du cycle de vie sont la fabrication (construction du site et fabrication des équipements), le transport (des matériaux pour la construction, et des équipements), l'opération du site (entretien, alimentation électrique...) et finalement la fin de vie (la démolition ou conversion du bâtiment, la restauration du site, et le recyclage ou la mise aux rebuts des matériaux et équipements).
iii) Pour toutes les activités qui se retrouvent dans ces différentes phases, il faut examiner l'impact sur la santé humaine, sur l'écologie, sur les changements climatiques et sur l'appauvrissement des ressources. Pour un centre de données typique, une grande partie de l'impact est reliée à la consommation d'énergie pendant l'opération de l'équipement informatique et de la climatisation (réchauffement climatique et appauvrissement des ressources, santé humaine et écologie aussi si charbon). Un autre impact non négligeable est la fabrication des équipement informatiques et électriques, en particulier le raffinage de l'or et du cuivre requis pour ces équipements (santé humaine et écologie).
Étant donné l'importance de la consommation électrique, une source d'énergie propre, et un climat froid qui ne requiert pas de climatisation, peuvent diminuer considérablement l'impact négatif d'un centre de données.

A21 :

Question 1 :
a) Il y a 13 serveurs DNS racine en redondance, qui ont tous le même contenu, pour traduire les noms de domaine en adresses IP. Un client interroge les 13 serveurs pour une requête et retient la réponse qui revient le plus souvent. Combien de serveurs en panne peut-on tolérer avant que ce service DNS racine ne soit plus en mesure de fournir une bonne réponse, si le modèle de panne est i) détectée (un serveur en panne répond avec un message d'erreur), ii) par omission (un serveur défectueux ne répond pas) iii) panne de réponse (une mauvaise réponse aléatoire), ou iv) arbitraire (mauvaise réponse concertée entre les serveurs défectueux)?
Les cas i) et ii) reviennent au même, à la différence que dans le premier cas on sait tout de suite que le serveur est défectueux, alors que dans le second cas on le déduit après un certain délai. On peut tolérer jusqu'à 12 serveurs en panne, car chaque serveur a le même contenu et,

même s'il ne reste qu'un seul serveur, il peut offrir le service. Il se peut toutefois que ce serveur devienne très chargé. Pour le cas iii) on suppose qu'il y a très peu de chances que deux mauvaises réponses aléatoires coïncident, et on peut donc obtenir la bonne réponse s'il reste deux serveurs valides. On peut alors tolérer 11 serveurs en panne. Finalement, dans le cas iv), il faut qu'une majorité de serveurs soient valides pour s'assurer d'une bonne réponse, car tous les serveurs compromis peuvent se liguer pour retourner la même mauvaise réponse. On peut ainsi tolérer 6 serveurs en panne, de sorte qu'il reste 7 serveurs valides dont la bonne réponse l'emportera.

b) Le serveur DNS d'un fournisseur de service Internet peut servir des requêtes DNS soit de manière itérative, soit de manière récursive. De manière itérative, le serveur traite la requête avec son CPU pendant 5 ms. Il peut alors retourner la réponse à ce moment dans 70% des cas. Cependant, dans 30% des cas, il doit poursuivre le travail et lire en plus son disque pendant 15 ms. Il trouve alors et retourne la réponse dans 60% des cas, mais dans 40% des cas il retourne une redirection vers un serveur plus haut dans la hiérarchie. En recevant une redirection, le client doit aller chercher la réponse sur un autre serveur, ce qui lui prend 20 ms. Si le serveur est configuré pour fonctionner de manière récursive, il prend aussi 5 ms de CPU et retourne la réponse à ce moment dans 70% des cas. Il doit aussi en plus lire son disque en 15 ms dans 30% des cas. Ensuite, il retournera la réponse demandée dans 90% des cas, mais dans 10% des cas il devra faire une requête et attendre la réponse avant de la retourner, ce qui lui demande en plus 1 ms de CPU et 20 ms d'attente. Quel sera le délai moyen vu par un client pour obtenir la réponse demandée dans chaque cas (récursif ou itératif)? Quel est le nombre de requêtes par seconde que peut soutenir le serveur, s'il n'utilise qu'un seul thread, de manière itérative? De manière récursive?

Itératif : Temps moyen de traitement côté serveur : 5 ms + 0.3 × 15 ms = 9.5 ms
Ajout du délai client dans les cas de redirection : 0.3 × 0.4 × 20 ms = 2.4 ms
Délai moyen total pour le client : 9.5 ms + 2.4 ms = 11.9 ms
Récursif : Temps de base : 5 ms + 0.3 × 15 ms = 9.5 ms
Ajout dans 10% des cas de lecture disque (sur les 30%) : 0.3 × 0.1 × (1 ms + 20 ms) = 0.63 ms
Délai moyen total pour le client : 9.5 ms + 0.63 ms = 10.13 ms
Capacité de traitement du serveur : Itératif : Le serveur est bloquant pendant 9.5 ms → 1000 ms/s ÷ 9.5 ms = 105.26 requêtes/seconde
Récursif : Le serveur est bloquant pendant 10.13 ms → 1000 ms/s ÷ 10.13 ms = 98.71 requêtes/seconde

Question 2 :

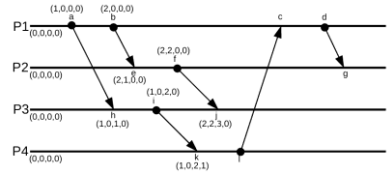
a) Un ordinateur A envoie une série de 3 messages à B pour obtenir l'heure exacte. Pour chaque envoi, l'heure d'envoi de la requête et l'heure de réception de la réponse sont notées avec l'horloge de A. La réponse de B donne la valeur de l'horloge de B au moment où il a traité la requête. Les différents messages ont été envoyés dans un intervalle très court, et on peut assumer que le décalage entre les deux horloges n'a pas changé pendant cet intervalle. La variabilité dans les réponses reçues dépend plutôt des délais variables pour l'envoi, la transmission et la réception des paquets via le réseau. Le détail des 3 envois est fourni dans le tableau suivant. Quels sont le décalage de temps et l'incertitude calculés pour chacun de ces 3 envois? Lequel doit-on retenir? Peut-on combiner les résultats des 3 requêtes pour obtenir une valeur de décalage de temps encore plus précise (avec moins d'incertitude)? Expliquez.

Envoi	envoi	réception	réponse
Envoi 1	14h00m05.200	14h00m05.450	14h00m03.300
Envoi 2	14h00m06.250	14h00m06.450	14h00m04.400
Envoi 3	14h00m07.300	14h00m07.600	14h00m05.400

L'incertitude est donnée par le délai de réponse (réception- envoi), soit 14h00m05.450- 14h00m05.200 = 0.250s pour l'envoi 1. En supposant que le délai est le même pour l'envoi et la réception, l'incertitude sera de +/- 0.250s / 2 = +/- 0.125s. L'heure de B à la réception sera ainsi l'heure de la réponse plus la moitié du délai, soit 14h00m03.300 + 0.250 / 2 = 14h00m03.425 +/- 0.125s. Le décalage est ainsi l'heure de réception de la réponse sur A moins l'heure de B à ce moment, soit 14h00m05.450- 14h00m03.425 = 2.025s +/- 0.125s (ou l'intervalle [1.900s- 2.150s]), valeur qu'il faut soustraire à l'heure de A pour se synchroniser avec celle de B. Le même calcul est fait pour les envois 2 et 3. On obtient ainsi les intervalles résultants pour chacune des requêtes. Si on doit choisir une réponse parmi les 3, on retient celle qui donne la plus petite incertitude, soit l'envoi 2. Par contre, en combinant ces trois intervalles, on peut déduire que le décalage doit être supérieur à 1.900s et inférieur à 2.050s, soit de 1.975s +/- 0.075s, une réponse encore plus précise.

Envoi	*envoi*	*réception*	*réponse*	*incertitude*	*d min*	*d max*
Envoi 1	*14h00m05.200*	*14h00m05.450*	*14h00m03.300*	*+/- 0.125s*	*1.900s*	*2.150s*
Envoi 2	*14h00m06.250*	*14h00m06.450*	*14h00m04.400*	*+/- 0.100s*	*1.850s*	*2.050s*
Envoi 3	*14h00m07.300*	*14h00m07.600*	*14h00m05.400*	*+/- 0.150s*	*1.900s*	*2.200s*

b) Quatre processus, P1, P2, P3 et P4, démarrent à peu près en même temps et s'envoient des messages. Chaque processus maintient un vecteur de 4 compteurs d'événements, avec une entrée correspondant à chaque processus. Chaque processus, au moment d'envoyer un message, incrémente son compteur d'événements dans son vecteur et joint le vecteur au message. Chaque processus, lorsqu'il reçoit un message, incrémente son compteur d'événements dans son vecteur et fusionne son vecteur avec celui reçu dans le message. Les vecteurs sont donnés pour les points a, b, e, f, h, i, j et k. Que peut-on dire de la relation temporelle entre les points a et j, à l'aide de ces vecteurs de compteurs d'événements? De la relation entre les points b et k? Expliquez. Donnez finalement les vecteurs de compteurs d'événements pour les points c et g.



Le point j est postérieur au point a car toutes les entrées de son vecteur de compteurs d'événements sont supérieures, (2,2,3,0) vs (1,0,0,0). Par contre, les points b et k sont potentiellement concurrents car certaines valeurs dans les vecteurs sont supérieures à celles correspondantes dans l'autre vecteur, pour chacun des deux points, (2,0,0,0) vs (1,0,2,1). Pour le point c, le vecteur est (3,0,2,2). Pour le point g, le vecteur est (4,3,2,2).

c) Un serveur central d'exclusion mutuelle reçoit les requêtes de divers clients. Il accorde la ressource si elle est libre, ou met la requête en queue pour la ressource, si elle est déjà prise. Lorsqu'une ressource est libérée par un client, la ressource est accordée au prochain client en queue, ou la ressource devient libre si la queue est vide. Est-ce que ce service est sûr (un seul utilisateur de la ressource à la fois), vivace (chacun peut obtenir un tour) et respecte l'ordre (premier arrivé, premier servi)?

Oui, le serveur central d'exclusion mutuelle est sûr, il n'accorde une ressource qu'à un seul client à la fois, il est vivace car chaque requête est maintenue en queue pour être servie à son tour, et respecte l'ordre, justement en raison des queues pour chaque ressource. On peut noter qu'il y a toutefois la possibilité que des interblocages se présentent, étant donné qu'il n'y a pas de contrainte imposée par le serveur sur comment sont pris les verrous.

Question 3 :

Les transactions T1, T2, T3 et T4 s'exécutent en même temps et leurs opérations de lecture et d'écriture sur des variables (x1, x2, ... x6) sont entrelacées. Les lectures d'une transaction sont effectuées sur les versions courantes des variables, et les écritures d'une transaction sont effectuées sur une version provisoire des variables pour la transaction. Lorsque la transaction se termine et est acceptée, la version provisoire des variables écrites par la transaction devient la version courante. Une validation de la cohérence par contrôle optimiste de la concurrence est effectuée pour accepter ou non chaque transaction. Une transaction est acceptée s'il n'y a aucun conflit sur les variables accédées (selon les critères de la méthode choisie) et est refusée s'il y a conflit sur certaines variables. Il faut tenir compte des transactions précédentes qui ont été validées (et ignorer celles qui ne l'ont pas été) dans le calcul des conflits pour savoir si chacune des transactions est acceptée ou non.

```
 1  T1:  Begin                 15 T4:  Write x2
 2  T1:  Read x2               16 T1:  Write x1
 3  T1:  Write x5              17 T1:  End
 4  T2:  Begin                 18 T2:  Read x6
 5  T1:  Read x5               19 T3:  Read x3
 6  T1:  Write x1              20 T2:  Write x3
 7  T3:  Begin                 21 T2:  Write x5
 8  T1:  Read x4               22 T2:  End
 9  T1:  Read x6               23 T4:  Read x5
10 T3:  Write x3              24 T3:  Write x4
11 T1:  Write x2              25 T3:  End
12 T4:  Begin                 26 T4:  Read x6
13 T4:  Read x3               27 T4:  Write x3
14 T4:  Read x2               28 T4:  End
```

a) Lesquelles des transactions T1, T2, T3 et T4 pourraient être validées si une validation en reculant était utilisée pour vérifier la cohérence des transactions? Pour chaque transaction non validée, donnez la ou les variables en conflit.

En reculant, au moment de compléter T1, il n'y a pas de problème. Pour compléter T2, il faut vérifier ce qu'il a lu (x6) versus ce qui a été écrit par les transactions précédentes concurrentes, T1 (x5, x1, x2). Il n'y a pas d'intersection et T2 peut compléter. Pour T3, ses lectures (x3) sont vérifiées versus les écritures de T1 (x5, x1, x2) et T2 (x3, x5). Il y a intersection (x3) et la transaction T3 doit être abandonnée. Pour T4, les lectures (x3, x2, x5, x6) doivent être vérifiées versus les écritures de T1 (x5, x1, x2) et T2 (x3, x5), mais pas de T3 qui a été abandonné. Il y a intersection (x2, x3, x5) et la transaction T4 doit être abandonnée.

b) Quel serait le contenu du journal écrit pour les transactions T1 et T2 (en supposant qu'il n'y aurait pas de conflit et qu'elles pourraient compléter, afin de bien découpler les sous-questions a et b).

Les écritures des transactions peuvent être écrites dans le journal au fur et à mesure qu'elles apparaîssent dans les transactions, ou elles peuvent être regroupées et écrites au moment de commettre. Dans le premier cas, T1 sauverait x5, x1, x2 et une deuxième fois x1; au moment de la relecture, seulement la valeur de x1 la plus récente serait prise. Dans le second cas, en écrivant au moment de commettre, x5 , x2 et seule la dernière valeur de x1 seraient sauvés. Voici un contenu possible du journal.

```
P0: ...                          P5: Compléter T1, P4
P1: écrire x5                    P6: écrire x3
P2: écrire x2                    P7: écrire x5
P3: écrire x1                    P8: Préparer T2(P6, P7), P5
P4: Préparer T1(P1, P2, P3), P0  P9: Compléter T2, P8
```

Question 4 :

a) Un service de fichiers répartis est offert par 3 serveurs redondants. Au moins une majorité de serveurs, soit 2 sur 3, doit être disponible pour que le service soit disponible. Chaque serveur est constitué d'un boîtier et son électronique, avec une probabilité de disponibilité de 0.9, ainsi que d'un ensemble de disques en RAID, 5 disques dont au moins 4 doivent être fonctionnels. La probabilité d'être fonctionnel pour un disque est de 0.8. Quelle est la probabilité qu'un ensemble de disques en RAID soit fonctionnel? Un serveur de fichiers? Le service de fichiers répartis?

L'ensemble de disques RAID a une probabilité de fonctionner de 0.8^5 = 0.32768 (5 disques fonctionnels) et 5!/((5−4)!4!)×0.8^4 ×(1−0.8)^(5−4) = 0.4096 (exactement 4 disques fonctionnels) pour un total de 0.32768 + 0.4096 = 0.73728. Un serveur est opérationnel si le boîtier est fonctionnel de même que son ensemble de disques RAID, ce qui donne 0.9×0.73728 = 0.663552. Le service sera disponible si 2 ou 3 serveurs le sont. La probabilité est de 0.663552^3 = 0.292162779 (3 serveurs fonctionnels) et 3!/((3−2)!2!)×0.663552^2 ×(1−0.663552)^(3−2) = 0.444415432 (exactement 2 serveurs fonctionnels), soit un total de 0.292162779+0.444415432 = 0.736578211.

b) Expliquez la différence entre un service répliqué avec des serveurs tous actifs, versus une réplication passive avec un serveur primaire actif et un serveur secondaire passif. Quels sont les avantages de chacun?

Dans un service répliqué avec des serveurs tous actifs, chaque serveur peut accepter des requêtes. Ceci a l'avantage de permettre plus de parallélisme et une meilleure performance. De plus, en cas de panne, un autre serveur est tout de suite disponible pour prendre la relève. L'inconvénient est qu'il est plus difficile d'assurer la cohérence dans un tel environnement.

Avec un serveur primaire, celui-ci s'occupe de servir toutes les requêtes. Il communique ses changements d'état au serveur secondaire, pour que ce dernier puisse prendre la relève en cas de panne.

Dans un tel système, la cohérence est très facile à assurer, tout passe par le serveur primaire, et les mises à jour sont envoyées dans l'ordre, sérialisées, au serveur secondaire. L'inconvénient est qu'il peut y avoir un court délai en cas de panne pour que le serveur secondaire bascule et devienne le serveur primaire. De plus, le serveur secondaire est largement inutilisé, puisque les pannes sont normalement rares, et cette ressource ne contribue pas à augmenter la performance du système.

c) Lors de vos travaux pratiques, vous avez utilisé Docker et Kubernetes. Expliquez brièvement la fonction de chacun. Lequel des deux vous permet d'obtenir une résilience aux pannes? Expliquez.

Docker permet d'encapsuler dans une image une application avec tous les exécutables et librairies qui sont requis. Il est alors facile d'exécuter cette application sans avoir à se soucier si les bons exécutables ou les bonnes versions de librairies sont disponibles sur le système où on veut deployer cette application. Kubernetes fait l'orchestration d'applications, souvent contenues dans des images Docker. L'orchestration consiste en rouler les différentes applications requises pour un système, tout en connectant ces applications aux bons ports réseau. Kubernetes permet en outre de surveiller les applications, redémarrer celles qui s'arrêtent prématurément, et répartir la charge des requêtes entre plusieurs réplicats. C'est cette dernière fonctionalité de Kubernetes qui permet d'obtenir une résilience aux pannes; les requêtes ne sont pas envoyées à un réplicat en panne, et celui-ci est redémarré aussitôt.

Question 5 :

L'entreprise pour laquelle vous travaillez veut construire un nouveau centre de données. Vous faites partie d'une équipe d'ingénieurs qui doit concevoir et planifier la construction et l'opération de ce centre de données. Vous voulez mettre de l'avant un design qui optimise le rendement par rapport aux coûts. Devez-vous comme ingénieur vous limiter aux impacts financiers, ou devez-vous aussi vérifier l'impact environnemental du projet? i) Quelles sont les lois applicables qui imposent à l'ingénieur ou au promoteur de tenir compte de ces impacts et du développement durable? ii) Quelles sont les différentes phases à considérer pour évaluer dans son ensemble un tel projet dans un contexte de développement durable? iii) Quels sont les quatre différents types d'impact sur l'environnement normalement considérés dans le cadre d'une analyse de cycle de vie?

L'ingénieur doit tenir compte des conséquences de l'exécution de ses travaux sur l'environnement dans une perspective de développement durable. i) Ceci est prescrit par la loi canadienne de 2008 sur le développement durable, la loi québécoise de 2006 sur le développement durable, et le code de déontologie de l'Ordre des Ingénieurs du Québec. La procédure d'évaluation environnementale du BAPE ne s'applique normalement pas à un centre de données. ii) Les phases du cycle de vie sont la fabrication (construction du site et fabrication des équipements), le transport (des matériaux pour la construction et des équipements), l'opération du site (entretien, alimentation électrique...) et finalement la fin de vie (la démolition ou conversion du bâtiment, la restauration du site, et le recyclage ou la mise aux rebuts des matériaux et équipements). iii) Pour toutes les activités qui se retrouvent dans ces différentes phases, il faut examiner l'impact sur la santé humaine, sur l'écologie, sur les changements climatiques et sur l'appauvrissement des ressources. Pour un centre de données typique, une grande partie de l'impact est reliée à la consommation d'énergie pendant l'opération de l'équipement informatique et de la climatisation. Un autre impact non négligeable est la fabrication des équipement informatiques et électriques, en particulier le raffinage de l'or et du cuivre requis pour ces équipements. Etant donné l'importance de la consommation électrique, une source d'énergie propre, et un climat froid qui ne requiert pas de climatisation, peuvent diminuer considérablement l'impact négatif d'un centre de données.