Examen de DESS Génie Logiciel Le 14 janvier 2002
Durée : 2 heures
Documents autorisés : notes de cours et documents de travail sauf les livres

Exercices

Exercice 1. IHM, X Window

a. Donner les principales étapes d'une application qui utilise un Toolkit X Window.
b. Donner les principales étapes d'une application X Window qui utilise Wcl.
c. Schématisez par un ou plusieurs dessins le fonctionnement réseau du système X Window.
d. On vous demande de porter le système X Window sur une nouvelle architecture. Quel est le travail effectif à réaliser ?

Exercice 2. Les Threads

On considère la classe *Compte* suivante.

```
public class Compte{

        private int valeurCourante

        public Compte(int montantInitial){

                valeurCourante=montantInitial;
        }

        void déposer(int montant){

                valeurCourante=valeurCourante+montant;
        }

        int retirer (int retraitDemandé){

                if (valeurCourante < retraitDemandé)
                        retraitEffectif=valeurCourante;
                else
                        retraitEffectif=retraitDemandé;

                valeurCourante =valeurCourante-retraitEffectif;

                return retraitEffectif;
        }
}
```

a. Pourquoi l'utilisation de cette classe dans une application multi-*threadée* est elle une source potentielle de problèmes ? Donner au moins un exemple significatif qui porterait préjudice à la banque gestionnaire du compte.
b. Proposer une solution.

Exercice 3. Programmation avancée Java

a. On se place dans l'hypothèse ou vous devez développer très vite le prototype d'une application de manipulation de graphes, offrant en particulier une fonctionnalité de sauvegarde/restauration des graphes manipulés. Quelle solution nécessitant très peu d'implémentation proposez vous pour réaliser cette fonctionnalité de sauvegarde/restauration ?

b. Qu'est-ce que Jini ?

Exercice 4. CORBA

a. Que signifie IDL ? A quoi cela sert-il ?

b. Que signifie IIOP ? A quoi cela sert-il ?

c. On vous demande de choisir un ou plusieurs ORB pour développer une application. Quels sont les principaux critères à considérer ?

Problème

L'entreprise QuickComp est spécialisée dans la fourniture de services de calcul au travers du Web. Ses clients sont des sociétés qui ont besoin d'effectuer des calculs qui nécessitent une machine puissante mais qui ne souhaitent pas investir dans une telle machine.

Pour assurer ce service QuickComp dispose actuellement de deux machines : un serveur Web www.quickcomp.com et une machine de calcul puissante appelée comp-1.

L'objectif de ce problème est de comprendre le fonctionnement de l'environnement logiciel actuel de la société QuickComp puis de lui proposer une architecture qui permette d'assurer sa montée en charge, c'est à dire d'assurer l'accueil d'un nombre croissant de clients.

Question 1. Architecture générale

Le fonctionnement actuel du service est le suivant.

Coté client. Le client écrit le code de son calcul en utilisant le langage Java. Nous verrons précisément sous quelle forme et avec quelles contraintes dans la question 2. Il se connecte au serveur Web de QuickComp *http://www.quickcomp.com/* et remplit un formulaire pour soumettre son code Java compilé. Les résultats de son calcul sont automatiquement intégrés dans une page HTML dont l'URL lui est ensuite communiquée par courrier électronique.

Coté QuickComp. Le serveur Web *www.quickcomp.com* transmet le code soumis par le client à l'application appelée *ServeurDeCalcul* tournant sur la machine puissante *comp-1*, et ce en utilisant RMI. Cette application exécute le code en question, puis transmet les résultats au client sous la forme d'un courrier électronique contenant l'URL d'une page HTML générée pendant l'exécution.

Question. Réaliser un dessin de cette architecture générale montrant en particulier l'utilisateur, les machines, les processus, et mettant en évidence les différentes phases de la fourniture de service.

Question 2. Exemple de demande de service

L'interface *Travail* est définie de la façon suivante :

```
import java.io.OutputStream;

public interface Travail{

        void travailAEffectuer(OutputStream out);

}
```

Pour décrire un calcul à effectuer, le client écrit une classe qui implémente l'interface *Travail*. Dans une implémentation de cette interface, c'est la méthode *travailAEffectuer* qui doit contenir le code du calcul effectif à réaliser. Le paramètre reçu par cette méthode est le flot de données à utiliser pour écrire les résultats, qui seront intégrés dans la page HTML dont l'URL sera transmise au client par la suite.

Le code Java transmis à QuickComp via le serveur Web www.quickcomp.com pour une demande de réalisation de calcul est une classe implémentant cette interface *Travail*.

Question. Ecrivez le code nécessaire pour décrire le travail suivant : calcul du nombre de pixels blancs dans une image. Cette image prend la forme d'une matrice d'entiers de taille 10x10 ou un pixel blanc est représenté par la valeur 0.

Question 3. Service de calcul

Question. Ecrivez le code du service de calcul *ServeurDeCalcul* fonctionnant sur *comp-1*.

Question 4. Réalisation d'une version réseau

Pour assurer la montée en charge, on souhaite pouvoir augmenter la puissance du système en ajoutant à *comp-1* des machines supplémentaires *comp-2* à *comp-n.*

Pour ce faire, on interpose entre le serveur Web www.quickcomp.com et les machines *comp-1* à *comp-n* une machine appelée *répartiteur*. On obtient ainsi une architecture 3-tiers.

Question. Présenter l'architecture résultante et écrire les codes des services devant fonctionner dans *répartiteur* et dans *comp-1* à *comp-n.*

Question 5. Généralisation.

Dans la configuration actuelle du système, les clients doivent soumettre leurs travaux via le serveur Web de QuikComp et les résultats des calcul effectués ne peuvent qu'être intégrés dans une page HTML. On souhaite maintenant permettre à un client :

- d'utiliser la puissance de calcul de QuickComp directement depuis une application Java fonctionnant chez lui ;
- de récupérer le résultat de son code de calcul sous la forme d'un objet Java.

Par exemple :

```
        …
        Resultat monResultat = (Resultat) quickcomp.calculer(travailAEffectuer);
        …
```

Question. Proposer une architecture supportant cette fonctionnalité.