

SI4 Polytech'Nice – UNS

UE « Applications Réparties », Partie « Objets » - Epreuve écrite individuelle

25 Mai 2012, durée 1h

Aucun document autorisé

F. Baude

Exercice 1 (barème approximatif : 10 points)

C'est vous qui jouez au prof ! Toutes les affirmations ci-dessous (provenant de vos étudiants) contiennent des erreurs. Pour chaque point, il vous est demandé de relever ces erreurs, et évidemment de donner, de manière très succincte, la bonne explication à destination des étudiants, leur permettant ainsi de se corriger.

1. Un RMI registry sert à stocker les objets serveurs RMI eux-mêmes (en les sérialisant)
2. Faire un lookup dans le RMI registry est le seul moyen disponible pour obtenir un objet permettant d'accéder aux méthodes implémentées par un objet serveur RMI
3. Un client RMI a besoin d'implémenter une interface du package RMI, afin de pouvoir invoquer les méthodes de l'objet serveur RMI
4. Comme pour le cas d'un client CORBA écrit en Java, un client RMI a besoin d'avoir dans son CLASSPATH les fichiers .class correspondant à tous les paramètres de type objet qui peuvent lui être renvoyés lors des appels de méthodes distantes
5. Le stub RMI d'un objet serveur RMI étendant UnicastRemoteObject doit forcément être généré à l'avance avec l'outil *rmic*
6. On a besoin de générer le stub RMI de l'objet RMI serveur, en se servant de *rmic -iio*, si on a comme client uniquement des clients CORBA écrits en Java (qui interagissent avec le serveur RMI en se fiant à l'interface IDL correspondant à l'interface RMI implémentée par l'objet serveur RMI)
7. JAAS offre une API exposant une méthode *login()* via laquelle on se donne les moyens de vérifier l'authenticité de l'appelant. C'est à l'intérieur de l'exécution de *login* qu'on va crypter les informations d'authentification, pour éviter qu'elles passent en clair sur le réseau dans le cas où l'appelant et l'appelé ne sont pas sur la même machine.
8. JAAS est prétendument portable, mais, si je modifie la manière d'authentifier un appelant (c'est-à-dire, si je veux prendre une autre classe qui contienne le code de la méthode *login()*) alors, je dois forcément recompiler toute mon application.
9. Un IDL CORBA n'a aucune ressemblance avec une interface RMI. C'est tellement différent qu'on ne peut d'ailleurs pas facilement générer l'un à partir de l'autre.
10. La sérialisation des paramètres des méthodes RMI repose sur la sérialisation Java, impliquant que les attributs de type objet (sous classes de Object) de ces paramètres (quand il y en a) sont automatiquement sérialisés (qu'ils annoncent ou non implémenter l'interface Java *Serializable*)

Exercice 2 (barème approximatif : 1 point)

```
Exception in thread "main" java.security.AccessControlException: access denied
(java.net.SocketPermission 127.0.0.1:1099 connect,resolve)
    at java.security.AccessControlContext.checkPermission(Unknown Source)
    at java.security.AccessController.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkPermission(Unknown Source)
    at java.lang.SecurityManager.checkConnect(Unknown Source)
    at java.net.Socket.connect(Unknown Source)
    at java.net.Socket.connect(Unknown Source)
    at java.net.Socket.<init>(Unknown Source)
    at java.net.Socket.<init>(Unknown Source)
    at sun.rmi.transport.proxy.RMIDirectSocketFactory.createSocket(Unknown Source)
    at sun.rmi.transport.proxy.RMIMasterSocketFactory.createSocket(Unknown Source)
    at sun.rmi.transport.tcp.TCPEndpoint.newSocket(Unknown Source)
    at sun.rmi.transport.tcp.TCPChannel.createConnection(Unknown Source)
    at sun.rmi.transport.tcp.TCPChannel.newConnection(Unknown Source)
    at sun.rmi.server.UnicastRef.newCall(Unknown Source)
    at sun.rmi.registry.RegistryImpl_Stub.rebind(Unknown Source)
    at java.rmi.Naming.rebind(Unknown Source)
    at BrokerImpl.main(BrokerImpl.java:24)
```

Vous obtenez cette exception, quelle est la cause probable du problème ? Que faites vous exactement pour y remédier (notez que votre programme utilise un SecurityManager)?

Exercice 3 (barème approximatif : 4 points)

On suppose que vous utilisez une vieille version de Java RMI, impliquant que le stub n'est pas généré à la volée, mais, a besoin d'être disponible dans un fichier .class. Supposons que dans votre application, ce stub est donc généré côté serveur, se nomme InterfaceDistantImpl_Stub.class. L'interface nommée InterfaceDistante se trouve donc implémentée par la classe InterfaceDistantImpl.class.

Par ailleurs, vous voudriez utiliser le téléchargement de classes, et donc, vous avez déposé le fichier InterfaceDistantImpl_Stub.class sur une machine de votre réseau, disponible en téléchargement pour vos clients, via un serveur http.

1. Quand on lance la JVM correspondant à l'application côté serveur, comment faire pour que le téléchargement de classes soit « activé » et donc, utilisable automatiquement coté client ?
2. Par malchance aujourd'hui, le comportement de votre réseau ne permet pas au client d'ouvrir des sockets vers la machine qui met à disposition les classes en téléchargement. Par contre, le client est tout de même en mesure d'ouvrir des sockets vers la machine où s'exécute le serveur RMI. Par contre, il vous est impossible d'aller récupérer manuellement le fichier .class correspondant au stub, qui est disponible sur la machine serveur.
Mais aujourd'hui vous voulez vraiment pouvoir exécuter l'application cliente, qui a besoin de se connecter au serveur RMI. Comment pourriez-vous vous en sortir ? C'est-à-dire, trouvez un moyen de quand même disposer coté client d'un fichier .class correspondant au stub.

Exercice 4 (barème approximatif : 5 points)

La façon de détecter si un nombre entier positif n est premier est de tester si les entiers dans l'intervalle $[2, \lceil \sqrt{n} \rceil]$ divisent ou non n . On veut mettre en place un objet RMI serveur, qui offre une méthode dont voici la signature `boolean isPrime(long candidate, long begin, long end)`. Voici un code d'implémentation de cette méthode.

```
for (long divider = begin ; divider < end ; divider++){
    if ((candidate % divider) == 0) {
        return false;
    }
}
return true;
}
```

1. Fournir le code du serveur RMI et celui de l'interface RMI.

2. Fournir le code d'un client RMI dont le but est d'utiliser cette méthode isPrime offerte par le serveur
3. Donner les lignes de commande permettant de lancer ce qu'il faut sur les 2 machines, la machine serveur notée **M**, et la machine du client notée **C1**.

Les deux prochaines questions sont volontairement un peu « ouvertes ».

Pour rendre votre serveur RMI encore plus utile, vous aimeriez pouvoir le rendre également disponible sous forme d'un service Web, afin que des clients Web puissent profiter de cette méthode isPrime. Il n'est pas question de dupliquer un code similaire à celui de isPrime pour implémenter le service Web, mais, il vous est demandé de réfléchir à une solution qui permette au service Web de profiter de la méthode isPrime implémentée par le serveur RMI.

4. Expliquer comment procéder, plus précisément quelles sont les étapes majeures à suivre. Vous pouvez supposer que vous utilisez Axis. Votre objectif est donc de proposer un service Web que des clients Web pourront utiliser. Ce service Web pourrait exposer une méthode nommée estPremier et permettant de passer en paramètre le n , et récupérer en retour, la réponse. Donner l'implémentation de ce service Web. En particulier, quelles sont les .class issus de l'application RMI, qu'il faut rendre disponible pour Axis (et donc recopier au sein d'un des sous répertoires de AXIS_HOME, dont on ne vous demande évidemment pas le nom) ?

On suppose à présent que plusieurs instances du même serveur RMI sont disponibles. Elles ont chacune été lancées sur des machines nommées **M**, **M2**, **M3**. Sachant ceci, notre client RMI de la question 2 va être modifié, dans l'objectif de profiter de ces 3 serveurs RMI **en parallèle**

5. Expliquer pourquoi, de base, le système RMI ne permet pas d'engendrer des appels à distance qui s'exécuteront en parallèle ? Comment du coup faire en sorte de profiter quand même de ces 3 serveurs RMI exposant isPrime? Expliquer également comment le client découpera le travail en 3.