

Architectures Distribuées

Question d'examens

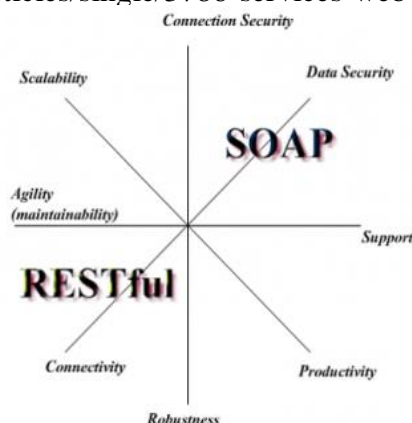
1. Dites pourquoi dans un accès distant, on a besoin d'utiliser des interfaces ?
 - C'est pour spécifier les méthodes qui sont accessibles au client
2. En programmation distribuée, expliquez comment un objet distant est appelé par un client (cas de RMI)
 - Extraire une référence de l'objet visé
 - Localiser les objets grâce à leur URL (classe Naming)
 - Extraire l'objet du registre
 - Convertir l'objet vers l'interface distante
 - Faire appel aux méthodes
3. Pourquoi les messages échangés dans un système distribué doivent être typés ?
 - Pour éviter les erreurs possibles liées aux types de données qui peuvent être différents d'un serveur à l'autre ou d'un client à l'autre, en fonction des technologies utilisées et des configurations de localisation.
 - Dans le cas de SOAP, ce problème est résolu par WSDL et DDL pour CORBA
4. Donner la définition d'une relation d'ordre causale

Ce sont les événements qui se sont déclenchés et qui sont causes de l'évènement considéré. Elle permet d'ordonner des événements de manière cohérente. Exemple : relation de précédence
5. Présenter les services qu'un Middleware propose pour les entités communicantes
6. On désire mettre en place une application, implémentée par un service web. On suppose que l'accès à celle-ci est conditionné par une **authentification** des utilisateurs. Laquelle des architectures SOAP ou REST peut-on recommander ? Justifier votre réponse

Nous recommandons SOAP pour les raisons ci-après :

- SOAP est un standard tandis que REST est une architecture
- SOAP est avec état (statfull) tandis que REST est sans état (statless)
- SOAP utilise HTTP, HTTPS, SMTP, FTP, SSH, ... tandis que REST utilise HTTP basé sur URL
- Avec SOAP on est avec un fort couplage entre le client et le serveur contrairement à REST
- La sécurité de SOAP repose sur WS-Security (signature et chiffrement) qui est plus sécurisé que le TLS/SSH de REST

Référence : <https://www.supinfo.com/articles/single/3786-services-web-rest-soap>



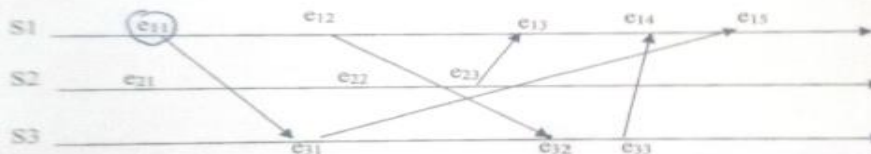
7. Quel est le rôle d'un service de nommage dans une architecture distribuée ?
 - Un service de nommage permet de retrouver les objets servants pour les clients
8. Qu'est-ce qu'un système de publication-souscription ?
9. Quelle est la différence entre une communication asynchrone et synchrone ?
10. Quel est le rôle du proxy dans une communication basée sur SOAP ?
 - Le proxy, placé du côté client, implémente l'interface du service et se charge des appels au service en utilisant le protocole SOAP lors de l'invocation de ses méthodes.
 - Ce proxy est généré par un compilateur dédié qui va utiliser le WSDL, notamment le portType pour définir l'interface de l'objet et les binding et port pour connaître les paramètres d'appel du service.
 - Le proxy généré est responsable de la transformation des invocations de méthodes en requêtes Soap et de la transformation des messages Soap en objets selon les indications fournies dans le WSDL.
11. Dans une communication basée sur REST, a-t-on la notion de proxy ?
 - Non, pas du tout
12. Expliquer pourquoi dans un accès distant on a besoin d'utiliser des interfaces
13. Rappeler le fonctionnement de l'horloge vectorielle

Examen : Architectures distribuées

Durée : 1H

Documents non autorisés

1. Quelle est la différence entre architecture répartie et distribuée ?
2. Quelles sont les deux caractéristiques fondamentales que doivent respecter les événements déclenchés dans un système distribué asynchrone ?
3. Quels sont les avantages d'une horloge logique par rapport à celle physique ?
4. On considère le diagramme suivant décrivant le début d'un calcul réparti qui comporte trois processus :



- a. Le diagramme respecte-t-il la dépendance causale entre message ? justifiez votre réponse.
- b. Donnez le passé de l'évènement e_{33} ?

5. Quelles sont les propriétés assurées par un middleware
6. En programmation distribuée, expliquez comment un objet distribué est appelé par un client
7. Dites pourquoi pour dans un accès distant on a besoin d'utiliser des interfaces ?
8. Quelles sont les limites du service de nommage de Java RMI ?
9. Que signifie l'erreur "address already in use" au lancement du service RMI-registry ? et que doit-on faire pour la corriger ?
10. Expliquez comment les EJB3 supportent l'asynchronisme ?
11. Quelles sont les étapes de l'appel d'un service Web décrit en WSDL, depuis un client écrit en Java.
12. Citez quatre inconvénients de REST par rapport à SOAP.

Scanned by CamScanner

Question de cours

1. Définir un système distribué

- Un système distribué est un système composé de plusieurs ressources (ordinateurs, smartphones, capteurs) reliées par un réseau, qui collaborent pour réaliser une tâche commune.
- Ces ressources sont autonomes et indépendantes

2. Quelle est la différence entre coopérer et collaborer ?

- Coopérer : je travaille tout seul et je suis seul responsable du résultat
- Collaborer : on travaille ensemble et nous sommes tous responsables du résultat

3. Donner 3 exemples de systèmes distribués

- Système de fichiers distribués : NFS
- Bases de données réparties

- Web services
4. Citer quelques intérêts des systèmes distribués
- Haute disponibilité
 - Extensibilité des ressources sans perturbation du système
 - Puissance de calcul démultipliée
 - Problème de capacité mémoire dépassée
5. Citer deux différences entre « Cluster computing » et « Grid computing », tout en donnant une brève explication
- Un **cluster** ou **grappe** est un ensemble de ressources homogènes et localisées
 - Un **grid** ou **grille** est ensemble de ressources hétérogènes et délocalisées
6. Quelle est la différence entre un **système distribué** et un **système parallèle** ?
- Dans un système parallèle, on a une seule et unique mémoire
 - Dans un système distribué (ou repart), on a plusieurs mémoires
7. Quels sont les caractéristiques des ressources dans un système distribués (spécificités) ?
- Pas de mémoire commune
 - Pas d'horloge commune
 - Asynchronisme des communications et des traitements
 - Pas d'état global instantané mais on essaie de l'approcher à travers des algorithmes
8. Quels sont les défis ou problèmes auxquels sont confrontés les systèmes distribués ?
- Pas d'horloge global
 - Pas de visibilité sur l'état global instantané
 - Tolérance aux fautes
 - Problème d'évolutivité
 - Qualité de service
 - Transparence
 - Sécurité
9. Donner les 8 types de transparences offertes par les systèmes distribués
- Transparence d'accès
 - Transparence de localisation
 - Transparence de réplication
 - Transparence à la mobilité
 - Transparence aux pannes
 - Transparence à la concurrence
 - Transparence à l'échelle
 - Transparence au parallélisme
10. Comment peut-on définir l'état global d'un programme ?
- Valeurs des variables

- Niveau d'instruction ou on s'est limité

11. Comment peut-on définir l'état global d'un système ?

- Les messages échangés sur le réseau (état des canaux de communication)
- L'état de chaque machine (ensemble des états des sites)

12. Comment peut-on définir l'état global sur un site ?

- Etat initial du site
- Suite des évènements sur le site

13. Comment peut-on définir l'état global d'un canal de communication entre les sites S1 et S2 ?

- Ensemble des messages émis par S1 et non encore reçus par S2

14. Quels sont les trois types possibles d'évènements ?

- Evènement local
- Evènement d'émission
- Evènement de réception

15. Citer une des solutions permettant de définir la cohérence de l'état global d'un système distribué ?

- La relation de précédence causale

16. Quelles sont les conditions nécessaires pour qu'un état global d'un système distribué soit cohérent ?

Un état global est dit cohérent s'il vérifie les deux conditions :

- a) Si une **émission de message** est captée dans un état local, alors la **réception de ce message** est soit dans un autre état local soit encore dans le canal
- b) Si une **émission de message** n'est pas captée dans un état local, alors la **réception de ce message** ne sera pas non plus dans un état local

17. Quels sont les modes de fonctionnement possibles d'un canal ?

- Canaux FIFO
- Canaux non FIFO

18. Définir un middleware

- Un middleware est un logiciel médiateur ou intergiciel qui va fournir des services à une application au-delà des services fournis par l'OS. Son rôle est de fournir des services spécifiques récurrents tels que :
 - Traitement des données
 - Stockage
 - Echange de message

19. Citer les quatre types de middleware

- Transaction Processing
- Appel de procédure/méthode distante (RPC/RMI)
- MOM

- ORB

20. Dire ce que c'est q'un MOM et expliquer brièvement son mode de fonctionnement

- Un MOM est un type de middleware qui permet aux applications de communiquer par message.
- Il permet d'éviter d'implémenter un service spécifique pour chaque type de communication entre les applications. Ainsi, chaque application ne s'adresse qu'au serveur de message (qui implémente une file d'attente) et utilise donc toujours les API de ce dernier.

21. Donner le rôle de chacune des commandes :

- `rmic -classpath .fichier_avec_bind`
 - `start rmiregistry`
 - `java fichier_avec_bind`
- `rmic -classpath .fichier_avec_bind` : génération du stub
 - `start rmiregistry` : démarrage du serveur de nom (**port par défaut 1099**) permet de faire correspondre un objet à un nom et inversement
 - `java fichier_avec_bind` : lancement du serveur

22. Définir un web service

- Un service web est un composant logiciel distribué exposant les fonctionnalités à forte valeur ajoutée d'un domaine métier.
- Il traduit le niveau logique d'accès aux traitements plutôt que le niveau physique d'implémentation

23. Citer quelques exemples de services web

- SOAP et REST

24. Citer quelques intérêts d'un service web

- Interopérabilité
- Réutilisation de l'infrastructure existante : HTTP, HTML, XML
- Augmentation du time to market
- Administration simple des composants

25. Citer 4 principes d'un service web

- Contrat standardisé
- Couplage lâche
- Abstraction
- Réutilisabilité
- Autonomie

26. Quelles sont les caractéristiques d'un web service ?

- Fournir à d'autres applications des services de traitements et données à travers une interface
- Protocole SOAP
- Répertoire de services UDDI

- Langage WSDL

27. Consommation d'un service web

- Recherche du service dans un annuaire UDDI
- Connexion au fournisseur pour récupérer l'interface du composant
- Invocation du service

28. Donner les 3 méthodes d'identification des services en décrivant brièvement les avantages et inconvénients de chacun

- Approche top-down :

Elle consiste à définir d'abord les processus métiers puis ensuite les services métiers nécessaires au fonctionnement de ces processus. Ainsi, on réduit la redondance des services mais les coûts sont élevés.

- Approche bottom-top :

On recense les fonctions existantes du SI dans le but de réutiliser en tant que service. Ainsi, on réduit les coût d'implémentation mais les services à développer rester très dépendants à leur système de base.

- Approche outside ou in the middle

Il s'agit d'une combinaison des deux premières approches

29. Quelle est la différence entre une application web et un service web ?

- Une application web utilise le protocole HTTP
- Un service web utilise comme protocole SOAP, REST qui sont camouflés à l'intérieur du protocole HTTP

30. Citer les 3 parties de l'annuaire UDDI

- Pages blanches : contient les informations générales (nom, adresse, contacts, ...)
- Pages jaunes : contient les informations sur le métier
- Pages vertes : contient les informations techniques

31. C'est quoi une architecture n-tiers ?

- Une architecture n-tiers est une architecture dans laquelle n entités collaborent pour réaliser une tâche

32. Quelles sont les 3 informations contenues dans un PBDU ?

- Id du pont racine
- Coût
- Id du pont émetteur

33. Quelle est la différence entre une communication synchrone et asynchrone ?

- Dans une communication synchrone, on ne tolère pas l'attente
- Dans une communication asynchrone, on tolère l'attente

34. Quelles sont les qualités d'un algorithme distribué ?

- Degré de répartition de contrôle
- Résistance aux pannes
- Trafic engendré

--système distribué

https://www.youtube.com/watch?v=MgyV_WJ4KhA

algorithme BFS:

<https://www.youtube.com/watch?v=DZb3ycSxCVE>

<https://www.youtube.com/watch?v=ONkUtuczgo4>

--middleware

<https://www.youtube.com/watch?v=8zaxoQOysBc>

--cluster et grid

<https://www.youtube.com/watch?v=9EoqLdmZCTU&t=2s>