ESP Classes : Master I GLSI Soir
Intervenant : M SONKO

Evaluation Finale

Module : Systèmes Répartis
Etudiant : Ousseynou Seck
Année Académique : 2024-2025

Répondons brièvement aux questions suivantes :

Exercice I (Contrôle)

1/ Quelles sont les composantes d'un système réparti

Un système réparti comprend des nœuds/machines interconnectées, un réseau de communication, des middlewares pour l'abstraction, des applications distribuées et des données réparties. Ces composantes travaillent ensemble via des protocoles de communication standardisés pour former un système cohérent.

2/ Quels sont les avantages et inconvénients d'un système réparti

Avantages : Partage de ressources, tolérance aux pannes, scalabilité, performance par parallélisme et économies d'échelle. **Inconvénients :** Complexité de développement élevée, problèmes de sécurité, latence réseau, gestion difficile de la cohérence et débogage complexe.

3/ Selon vous comment des applications écrites dans des langages différents arrivent-elles à communiquer ?

Les applications de langages différents communiquent via des protocoles standards (HTTP, TCP/IP), des formats d'échange universels (JSON, XML) et des interfaces standardisées (REST, Web Services). Les middlewares comme RabbitMQ, gRPC facilitent cette interopérabilité en abstrayant les différences de langages.

4/ Quelle est la différence entre duplication et réplication de données ?

La **duplication** crée des copies statiques des données principalement pour la sauvegarde et la récupération en cas de panne. La **réplication** maintient des copies cohérentes et synchronisées des données sur plusieurs sites pour améliorer les performances et la disponibilité.

5/ Citez quelques propriétés fondamentales que doit posséder un système réparti

Un système réparti doit posséder la transparence (masquer la complexité), la fiabilité (fonctionnement correct malgré les pannes), la disponibilité (accessibilité continue). Il doit aussi garantir la sécurité des données, la scalabilité pour gérer la montée en charge et la cohérence des données distribuées.

6/ Expliquez brièvement les notions de socket, port et le déroulement du dialogue entre un client et un serveur

Un **socket** est un point de communication bidirectionnel entre processus. Un **port** (0-65535) identifie un service sur une machine. Le **dialogue** suit : serveur (socket, bind, listen) → client (socket, connect) → serveur (accept) → échange de données → fermeture.

Exercice II (D.S)

1/ Le développeur d'applications réparties peut utiliser 2 techniques :

• La programmation directe de la couche réseau du système d'exploitation via les API fournies par les langages de programmation

• L'utilisation d'intergiciels (middlewares)

Décrivez brièvement les avantages et inconvénients de chaque solution

Programmation directe : Contrôle total et performance optimale mais complexité élevée et temps de développement long. **Middlewares :** Simplicité d'usage et réutilisabilité mais overhead de performance et dépendances externes. Le choix dépend du compromis entre contrôle et facilité de développement.

2/ Pourquoi un système centralisé est-il nécessaire dans la conception d'un système distribué ? Techniques utilisées pour mettre en place un système centralisé ?

Un système centralisé est nécessaire pour la coordination globale, la gestion des ressources partagées et le maintien de la cohérence. **Techniques :** serveurs de noms/annuaire, serveurs de configuration, coordinateurs de transactions, load balancers centraux et services de découverte.

3/ Expliquez les notions suivantes : serveur virtuel, machine virtuelle, conteneur et réseau virtuel

Serveur virtuel : Partition logique d'un serveur physique. **Machine virtuelle :** Émulation complète d'un système avec son OS. **Conteneur :** Virtualisation légère partageant le noyau de l'hôte. **Réseau virtuel :** Infrastructure réseau créée par logiciel, indépendante du matériel.

4/ Différences entre communication synchrone et asynchrone

Synchrone : L'émetteur bloque son exécution en attendant la réponse (ex: appel de fonction, RPC). Ordre prévisible mais peut créer des goulots. **Asynchrone :** L'émetteur continue sans attendre (ex: messages, callbacks). Meilleures performances mais gestion plus complexe.

5/ Expliquez la notion de passage à l'échelle (scalability)

Capacité d'un système à maintenir ses performances lors de l'augmentation de la charge. Trois types : **horizontale** (ajouter des machines), **verticale** (améliorer les ressources existantes), **fonctionnelle** (distribuer les fonctions sur des serveurs spécialisés).

6/ Que représente le load-balancing (équilibrage de charges) ?

Répartition équitable de la charge entre plusieurs serveurs pour optimiser les performances et éviter la surcharge. Techniques principales : Round-robin (rotation), Least connections (serveur le moins chargé), Weighted (pondération), IP Hash (basé sur l'adresse client).

7/ Donnez quelques exemples de systèmes distribués réels

Web : Google, Facebook, Amazon avec centres de données mondiaux. **Cloud :** AWS, Azure, Google Cloud. **Bases de données :** MongoDB, Cassandra. **Messaging :** WhatsApp, Telegram. **Streaming :** Netflix, YouTube. **Blockchain :** Bitcoin, Ethereum.
