

Université du Québec à Chicoutimi
Département d'informatique et de mathématique
8INF843 - Systèmes répartis
TP2

Professeur : Hamid Mcheick

Groupe : deux étudiant(e)s au maximum

Session : Automne 2020

Date de distribution : 28 septembre 2020

Pondération : 15 points

Date de remise : 9 Novembre 2020

Vous devez réaliser UN SEUL choix parmi les quatre choix donnés dans ce TP.

Choix 1 :

Afin de contribuer à la protection de l'environnement, une entreprise TravelExpress.com (semblable à amigoexpress : <http://www.amigoexpress.com/>) désire développer une application distribuée (Web) de covoiturage. L'application permettra aux gens qui partagent les mêmes parcours de se connaître et de s'organiser pour faire du covoiturage.

Les fonctions principales de l'application sont :

- ***L'inscription des usagers*** : Cette fonction permet aux usagers de s'inscrire sur le site de covoiturage. L'utilisateur introduit ses informations personnelles ainsi qu'un nom d'utilisateur et un mot de passe qui lui permettront d'ouvrir une session.
- ***Publication*** : L'utilisateur dont la session est ouverte doit être en mesure de publier les itinéraires de ses voyages. Il indiquera aussi l'horaire, les dates et les fréquences de chacun de ses voyages.
- ***Préférences*** : L'utilisateur indique dans son profil ses préférences en indiquant ce qui tolère aux passagers et ce qui sera interdit. Si l'utilisateur est comblé, une indication dans son profil doit permettre de savoir.
- ***Recherche*** : Cette fonction permettra aux usagers de faire des recherches selon certains critères et d'obtenir les coordonnées des personnes intéressées.
- ***Réservation*** : cette fonction permet de réserver une ou plusieurs places sur le bord de véhicule (plus petit que le nombre de places du véhicule).
- ***Payment*** : cette fonction permet au client (passager) de payer 5\$ par voyage. Vous pouvez simuler cette fonction d'une façon virtuelle sans vérifier les crédits des passagers. Vous simplifiez la méthode de paiement pour finaliser ce TP le plutôt possible.

Travail à faire :

1. (80%) Implémenter les fonctionnalités de l'application et donner le code source bien documenté. Je vous laisse le choix de sélectionner les technologies (ou combinaison) pour développer cette application distribuée:
 - a. Application distribuée côté serveur, SocketWeb,
 - b. RMI, RPC,
 - c. Java EE (EJB),
 - d. HTML, XML,
 - e. JSP et/ou Servlets, Des classes beans, ...
 - f. SOAP, REST,
 - g. NodeJS, Django (Python), Ruby on Rails,
 - h. Angular JS, React, Javascript, Ajax, Bootstrap, ...
 - i. etc.
2. (20%) Produire une documentation du système incluant :
 - a. Introduction et description du système
 - b. Schémas et diagrammes de conception
 - c. Le guide de l'utilisateur
 - d. Le test (trace d'exécution)

Conditions exigées:

- L'application peut être réalisée en utilisant REST, RPC, RMI-Java, Java EE (EJB), servlet-JSP, NodeJS, Ruby, PHP, C++, C#, React, Angular JS, ou autres technologies et middlewares des systèmes répartis.
- Notez le mapping entre les objets (Java, C++, ...) et la base de données Oracle ou MySQL pour la gestion de la persistance.
- Le travail doit être exécuté sur plusieurs plateformes (Windows, Linux, Android, iOS, etc.) et l'application doit être accessible de l'extérieur par l'Internet ou Telnet ou localement (serveur web local sur la même machine). Il faut fournir l'URL pour accéder à votre application.
- Le rapport et les fichiers de votre application doivent être remis au professeur directement ou par email. Des pénalités de 10% par jour seront appliquées pour tout travail remis après cette date.

Quoi remettre :

- Documentation du système et
- Code complet

Choix 2 :

Concevoir et développer une application (exemple de communication) en utilisant une technologie (WIFI-Direct, Wifi, Lora, SigFox, ...). L'idée est de faire passer un message M entre un émetteur et un récepteur dans un réseau de n nœuds en tenant compte des paramètres de la technologie utilisée (distance, consommation d'énergie, etc.). Dans cette application, il faut assurer les tâches suivantes en choisissant une solution pour simuler l'application distribuée et la communication en utilisant un appareil (Téléphone cellulaire, tablette, Arduino, ESP32, Lopy4, ou autre) comme outil:

- 1- Algorithme ou technique de connectivité lors d'un extrême événement pour passer un message d'un émetteur à un récepteur (étudier les algorithmes et les techniques de routage et choisir celles/ceux qui fonctionnent le mieux pour connecter ces appareils et faire passer les messages entre eux).
- 2- Gestion des modes de communication (client-serveur, multi-tiers, Peer-to-Peer, etc.).
- 3- Représentation des données (XDR, CDR, sérialisation, Json, XML, etc.).

Quoi remettre :

- Documentation du système décrivant comment vous allez concevoir ton application pour compléter les tâches décrites ici-haut : 20%
 - o avec un Schémas et diagrammes de conception (différentes classes de l'application)
 - o avec un guide de l'utilisateur (comment utiliser, installer et exécuter cette application?)
- Code complet de l'application 80%

Choix 3 : Considérons le réseau Mesh Network pour Lopy4 (Pycom)

- Donnez et expliquez (motiver) l'architecture de Mesh network (illustrez la réponse avec images) 25%
- Pourquoi Mesh Network pour l'IoT utilisant Lopy4 (Pycom) devient important ? 25%
- Montrez une étude de cas (exemple de code complet) pour illustrer l'utilité de ce réseau 25%
- Discutez les avantages et les inconvénients de Mesh Network pour l'IoT 25%

À remettre : document Word/pdf avec code complet de l'étude de cas.

Choix 4 : Considérons les algorithmes d'ordonnement suivants : FCFS, LCFP, SCFP, MCT

Les algorithmes sont: First come First Serve (FCFS) [1], Minimum Completion Time (MCT) [2], Longest Cloudlet Fastest Processing Element (LCFP) and Shortest Cloudlet Fastest Processing Element (SCFP) proposed by Sindhu et al. [3].

[1] N. Bansal, A. Awasthi, S. Bansal, "Task Scheduling Algorithms with Multiple Factor in Cloud Computing Environment," Springer India, pp. 619-627, 2016.

[2] S. Sheikh, A. Nagaraju, “A Comparative Study of Task Scheduling and Load Balancing Techniques with MCT using ETC on Computational Grids 2017,” Indian Journal of Science and Technology, Vol 10(32), DOI: 10.17485/ijst/2017/v10i32/110751, August 2017.

[3] S. Sindhu, S. Mukherjee, “Efficient Task Scheduling Algorithms for Cloud Computing Environment,” Springer-Verlag Berlin Heidelberg, pp. 79-83, 2011.

- Donnez et expliquez (motiver) l’importance de ces algorithmes (illustrez la réponse avec exemples) 25%
- Identifier des critères de comparaison de ces algorithmes 25%
- Montrez une étude de cas (exemple de code complet de deux algorithmes parmi les quatre donnés ci-haut) pour illustrer l’utilité de ces deux algorithmes. Vous pouvez choisir un langage de programmation et outil pour la partie développement 25%
- Discutez les avantages et les inconvénients de chacun de deux algorithmes choisis 25%

À remettre : document Word/pdf avec code complet de l’étude de cas.

Bon courage !