Friki

(Free dIstributed wIKI)
Projet Développement de Systèmes Distribués
Année 2004-2005

Bram De Wachter

March 8, 2005

1 Introduction

Les dévelopeurs de Wikipedia[1] sont intéressés par une version distribuée de leur base de données constituant la plus grande encyclopédie disponible gratuitement sur le web. Ils vous demandent de faire un *proof of concept* qui pourrait être intégré dans leur moteur de gestion couplé à leur serveur web. Puisqu'on parle d'un proof of concept, il ne faut pas leur montrer des interfaces utilisateurs poussées (un programme avec interface simple *ligne de commande* ou graphique très réduite est amplement suffisant). Ce qui vous est demandé devra être implémenté en utilisant Java 2 RMI. En résumé, il s'agit de trois blocs fonctionnels qui seront détaillés plus loins :

- Gestion des utilisateurs
- Gestion de la modération
- Gestion de la mise à jour des pages disponibles

L'aspect distribution est détaillé à la fin de ce document.

2 Gestion des utilisateurs

Les utilisateurs sont principalement divisés en deux catégories :

- Les modérateurs (aussi appelés *super*-utilisateurs) sont capable de juger sur le contenu des pages en ligne. Quand ils se connectent au système, ils recevront une liste d'articles modifiés, et pourront accepter ou refuser les changements
- Les utilisateurs, sont les clients du système, ils peuvent
  - Consulter les pages disponibles en donnant un mot clé
  - Créer des pages
  - Modifier les pages disponibles, ce qui veut dire qu'un utilisateur peut dans l'ordre
    1. Demander la permission de modifier une page
    2. Si la permission est accordé, apporter des changements
    3. Valider ces changements

Une page sera constituée d'un texte, une liste de mots clés, et une liste de références vers d'autres pages dans l'encyclopédie.

[1] `http://en.wikipedia.org/wiki/Main_Page`

3 Gestion de la modération

Quand un utilisateur modifie une page, la modification ne sera pas visible directement. La raison pour cela est que les modifications déplacées de certains utilisateurs pourrait faire descendre considérablement la qualité des pages en ligne. Pour cela, les modifications sont stockées, et c'est aux modérateurs de juger si les modifications apportées à une certaine page sont acceptables.

Si tel est le cas, le modérateur peut décider d'accepter les modifications, et intégrer les changements pour la version en ligne. La page visible sera dans ce cas-là tout simplement remplacé avec la nouvelle version.

4 Gestion de la mise à jour des pages

Un problème majeur avec ce système de modération est que plusieurs utilisateurs sont susceptibles de changer la même page en même temps, ce qui demanderait un effort considérable au modérateur de réunir les multiples changements pour obtenir un document cohérent. En effet, dans ce cas il ne peut plus simplement remplacer la page avec une version modifiée par un utilisateur car les changements dus aux autres utilisateurs se perdraient.

Pour éviter cela, une page ne peut avoir qu'un seul éditeur (utilisateur) à la fois. Une fois qu'un utilisateur décide de modifier une page, il faut donc s'assurer qu'il n'existe pas déjà un autre utilisateur qui est en train de modifier la page. Si tel est le cas, on signale à cet utilisateur que la page est *bloquée* est qu'il doit attendre que le modérateur ait accepté ou refusé les modifications de l'autre utilisateur.

Un deuxième problème est causé par les dépendances entre les pages. Imaginez qu'un certain utilisateur fait dans la page A, une référence vers une page B, tandis que la page B est en train d'être modifiée. Il est évident que les modifications apportés sur la page B peuvent rendre inconsistant le contenu de la page A.

Pour remédier à ce problème, il ne faut pas seulement bloquer la page X quand elle est modifiée, mais aussi toutes les pages sur qui X dépend et ceux dont X dépend. On peut formaliser cette idée à l'aide d'un graphe G où les noeuds sont les pages et les arcs sont les dépendances. Quand un certain noeud n est modifié par un utilisateur U, on bloquera toutes les pages $\in \{n' | n \rightarrow n' \vee n' \rightarrow n\}$ pour toutes les utilisateurs autre que U. Remarquons qu'une page bloquée reste disponible (avec son ancien contenu) pour la lecture.

5 La distribution

Ces opérations doivent évidemment se faire de manière distribuée, c'est à dire que chaque page peut se trouver sur un autre serveur. Il vous est spécifiquement demandé de ne pas centraliser la gestion de la mise à jour. Plus particulièrement, pour des raisons de robustesse, on ne veut pas que le graphe de dépendance des pages existe dans son entièreté sur un des serveurs. Chaque serveur aura uniquement la liste des dépendances directes des pages qu'il possède.

6 En pratique

Il faut que votre proposition fonctionne sur les machines du NO 4^e étage (c'est là que vous défendrez votre proposition). Il faut impérativement utiliser Java RMI et fournir un code bien commenté. On vous demande aussi un rapport (de 5 à 10 pages) comprenant vos choix d'implémentation ainsi que les problèmes rencontrés, résolus et/ou pas résolus. Le travail doit être terminé pour fin avril.

Vous pouvez travailler seul, ou en groupe de deux. N'hésitez pas à poser des questions pendant ou en dehors du TP.

Bon travail.