

TP1 - Appels de méthodes à distance

INF4410 - Systèmes répartis et infonuagique

Nom de l'étudiant : Félix La Rocque Carrier

Matricule : 1621348

Nom de l'étudiant : Mathieu Gamache

Matricule : 1626377

Polytechnique Montréal

19 septembre 2016

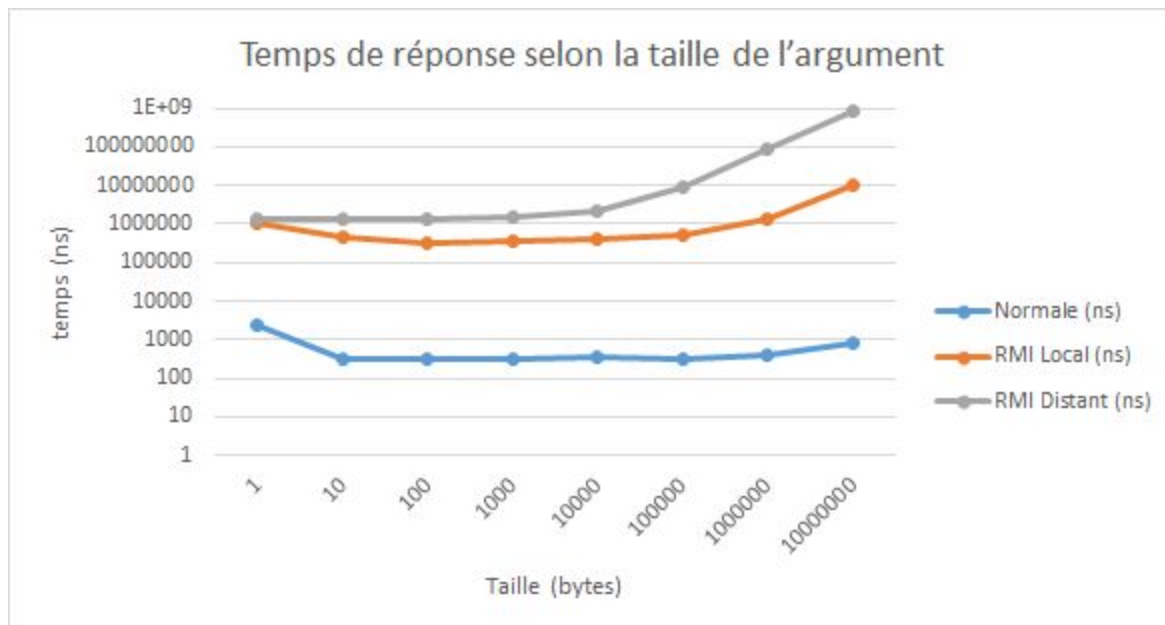
Partie 1

Question 1: On se propose d'étudier l'impact de la taille des arguments sur la performance des appels RMI

Tableau 1: Temps de réponse selon la taille de l'argument.

Size	Normale (ns)	RMI Local (ns)	RMI Distant (ns)
10^0	2 516	1 087 499	1 403 790
10^1	318	485 544	1 274 057
10^2	302	338 114	1 343 090
10^3	324	378 772	1 500 925
10^4	362	417 809	2 096 771
10^5	302	492 643	9 562 655
10^6	410	1 322 484	86 373 973
10^7	812	9 716 770	863 468 899

Nous pouvons voir que plus l'argument est grand, plus le temps de réponse est lent. Nous avons toutefois une donnée bizarre pour la première taille de 1 qui prend un temps largement supérieur à celle escompté.



Question 2: Expliquez l'interaction entre les différents acteurs (client, serveur et registre RMI) à partir du tout début de l'exécution.

Dans notre système, nous avons 3 acteurs principaux:

- Le Client, possédant l'interface utilisateur et un dossier contenant des documents.
- Le Registre RMI, permettant de faire le lien pour enregistrer et récupérer un
- Le Serveur, possédant une interface et des méthodes pouvant être appelé en plus d'un dossier de documents.

Pour permettre l'appel et le retour d'information entre le client et le serveur, le registre RMI abstrait le lien entre ceux-ci et permet l'appel de fonction peu importe où se situe le serveur.

Du côté serveur, la première étape est de définir les fonctions pouvant être appelé à distance sur le serveur dans un interface (ServerInterface.java). Une fois cette interface définit, on doit implémenter les fonctions (server.java).

Dans la fonction Run de la classe "Server", on utilise LocateRegistry.getRegistry() pour obtenir le Registre RMI puis la fonction bind pour attribuer le nom "server" dans le registre de nom RMI avec la valeur du hostname défini dans le fichier de script server.

Du côté client, dans la fonction Run, on lui passe la référence vers le hostname voulu pour serveur et le client peut récupérer le stub dans le registre RMI. Le client peut ensuite appeler des méthodes de l'interface du serveur sur le stub.