

Jordan ELLAPIN
Abdelaziz FOUNAS
Taquiéddine ZEGAOU

Compte rendu du TP BAM :

Objectif n°1 :

Afin d'accomplir cet objectif, nous nous sommes aidé de l'ancien TP RMI.
Nous avons trois parties : une commune aux clients et aux serveurs, et deux autres spécifiques aux clients et serveurs.

Objectif n°2 :

Pour cet objectif nous avons relue plusieurs fois le TP afin de bien comprendre le but de chacune des classes de l'infrastructure BAM. De plus, la lecture des fichiers XML utilisés par la classe Starter a été très utile pour comprendre comment les éléments s'enchaînent entre eux.

Objectif n°3 :

Après avoir compris un maximum d'éléments sur l'infrastructure BAM, nous nous sommes attaqué au codage de cette dernière. Nous avons trois packages : un pour toutes les classes spécifiques à l'étape de l'agent Hello plus un un service ServiceShow que l'on a ajouté, un autre pour toutes les classes spécifiques à l'étape de l'agent LookForHotel et les services destiné à son bon fonctionnement, et un dernier se nommant kernel contenant les classes permettant les fonctionnement des serveurs accueillant tous ces différents types d'agents.

Objectif n°4 :

Pour cet objectif, on s'est grandement inspiré du code que l'on avait déjà développé pour l'infrastructure RMI. En effet, lors de la création du service on load toutes les informations utiles - du XML contenant les hotels ou les numéros - dans l'instance du service. Lors d'un appel à ce service on cherche les informations demandées sans repasser par le XML. Exactement comme ce qui a été fait pour RMI.

Objectif n°5 :

Pour cet objectif, on pense que les infrastructure BAM et RMI peuvent être comparé sur plusieurs points déterminant pour la performance globale du système.

Ces points sont les suivant :

- Le temps mis pour obtenir le même montant d'information

- Le montant de données transmises pour le même montant d'information

Lors de nos exécution sur les 40000 numéros, BAM et RMI ont mis tous deux approximativement 10 secondes à obtenir toutes les informations sur les clients demandant. Par contre sur 20000 numéros, BAM fait 2 secondes tandis que RMI en fait 3.

Sur le montant de données transmises, RMI fait nécessairement mieux que BAM, car il n'y a quasiment que des données utiles qui transitent. Alors que sur BAM, il y a les mêmes données que RMI, plus les agents, et les jars contenant les classes nécessaires au fonctionnement de l'agent.

Objectif n°6 :

Pour l'objectif 6, nous avons créé une classe `AnnuaireServices` étendant `UnicastRemoteObject`. Cet objet aura pour but de s'enregistrer auprès d'un serveur RMI afin d'offrir deux primitives. Une permettant d'ajouter un nouveau service de la part d'un serveur BAM, l'autre permettant de récupérer la liste des serveurs BAM offrant un service donné.