



UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
USTO - MOHAMED BOUDAF-
FACULTÉ DES MATHÉMATIQUES ET INFORMATIQUE

module : GRILLES INFORMATIQUES

Membres du groupe

BELHOUARI FAROUK BENYAMINA
HABBOUCHI ABDELFAH
FARDEHEB SALIM

Produit matriciel avec le
simulateur de grille
GRIDSIM

2023-2024

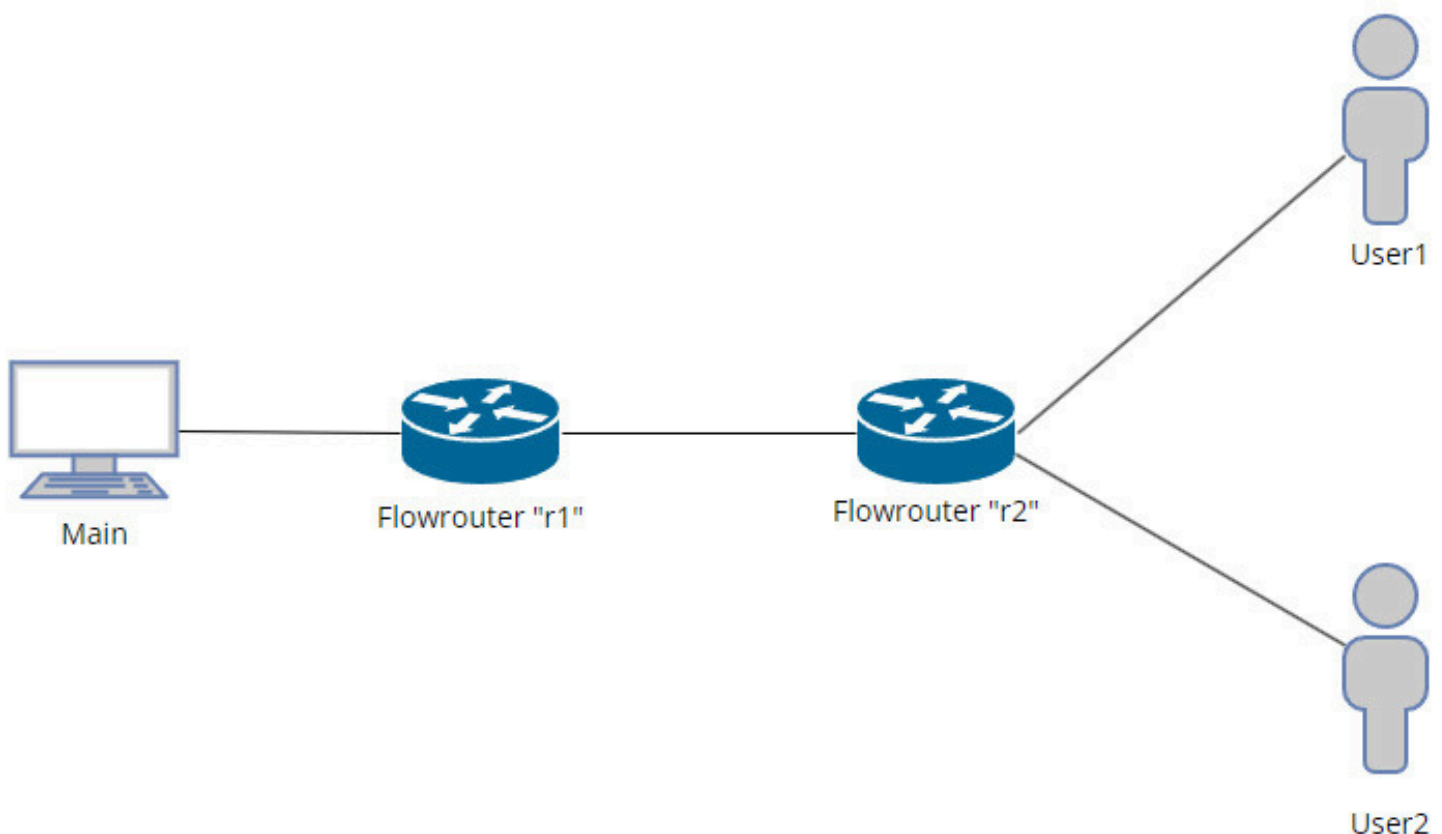
Sommaire

- 01.** Topologie
- 02.** Les entités et leur role
- 03.** approche
- 04.** les classes
- 05.** Déroulement
- 06.** Interprétation

Topologie

Dans notre Topologie , nous avons deux utilisateurs, chacun connecté à un routeur nommé "r1". En parallèle, nous avons un deuxième routeur nommé "r2" qui est connecté à une machine "M1" et également connecté au routeur "r1". Au cours du déroulement du programme, une fois que les deux utilisateurs ont calculé leurs parties respectives, ils envoient chacun leurs résultats à travers "r1" vers "r2", afin que ces données parviennent à leur destination finale, la machine "M1".

le type " flowrouteur" a été choisi car il est plus développé que le RIP-routeur dans la particularité de choix de chemin , a cause de ça il pose moins de problèmes indésirables lors de l'échange des messages



Les entités et leurs rôles



Main distribue les taches et reçoit les resultats d'apres les utilisateurs , elle affiche la matrice qui resulte de la multiplication



les 2 utilisateurs

chacun d'eux reçoit sa partie à calculer , après il la calcule et envoie son résultat a l'entité main



flowrouteur r2 liée avec "Main" et avec l'autre routeur .



flowrouteur r1 Il lie entre les 2 utilisateurs , et est liée aussi avec "r2" pour assurer la transmission des messages (résultats) vers main .

approche

1

Distribution

Dans notre approche pour le produit matriciel distribué, nous avons choisi d'impliquer deux utilisateurs dans le partage du travail.

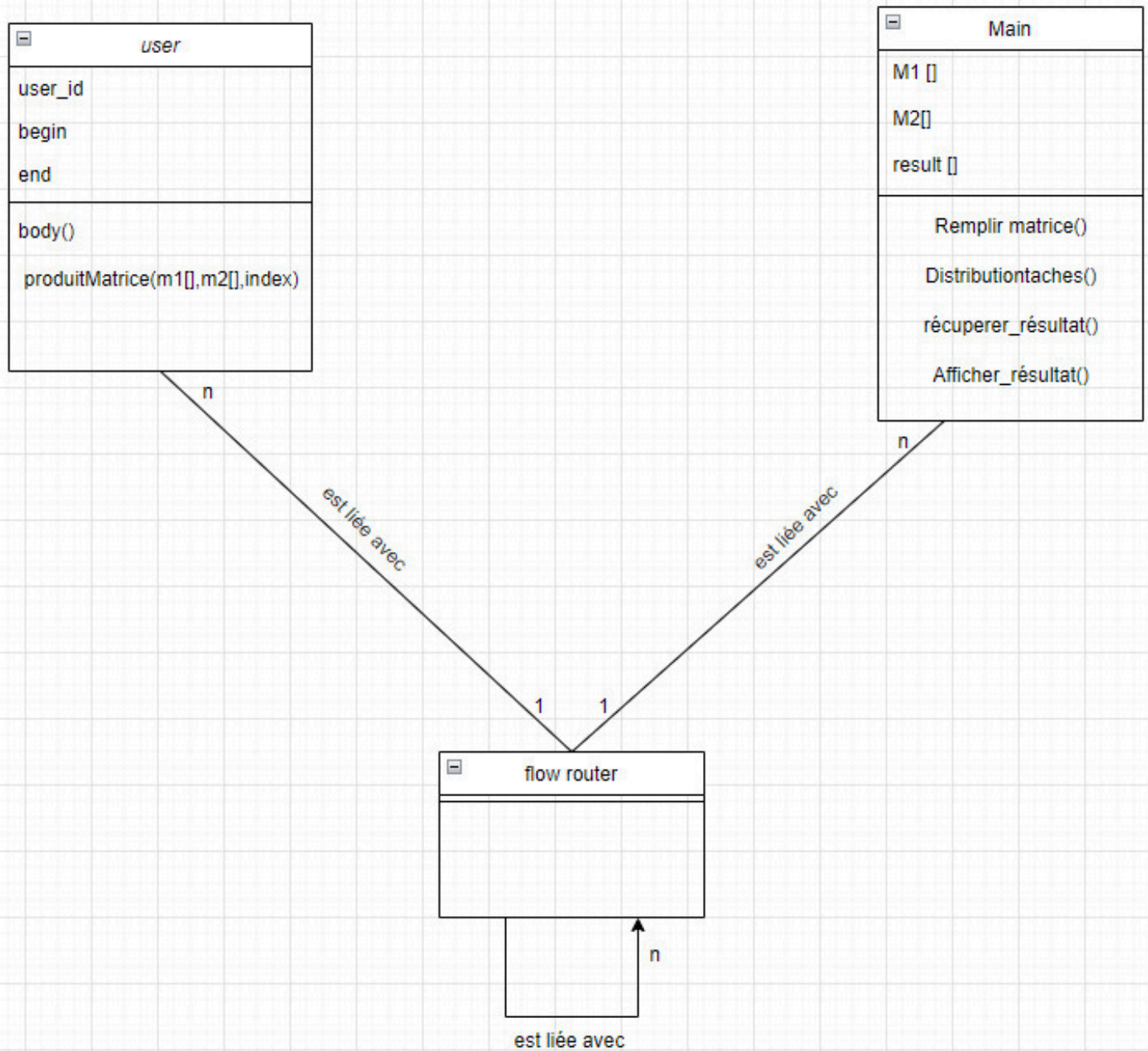
Voici comment nous procédons :
nous disposons de deux matrices, et le résultat du produit de ces dernières est une matrice dont le nombre de lignes correspond à celui de la première et le nombre de colonnes à celui de la deuxième.

Ensuite, une fois ce nombre de colonnes est déterminé, nous répartissons le travail entre les deux utilisateurs. Si ce nombre est pair, la répartition est équitable, chacun des utilisateurs se voyant attribuer un nombre égal de valeurs à calculer.

Cependant, s'il est impair, l'un des utilisateurs aura une colonne supplémentaire à calculer.

Ce processus de distribution est géré par l'entité "machine" à l'aide de sa méthode "distributionTaches()".

les classes



les classes

1_user

les deux utilisateurs qui s'occupent des calcul sont des instances de cette classe .

1_1: ses attributs

id_user :c'est l'identifiant qui permet d'identifier l'utilisateur a l'Entité main lors de la récupération des resultats pour qu'elle sait d'ou ils viennent et leur renvoi l'accusé de reception

begin et end : ces deux attributs sont les indices de début et fin de calcul , ça veut dire les colonnes que cet utilisateur va calculer (du début vers end)

1_2: ses méthodes

body():c'est une methode qui se declenchent lors de debut de la simulation
Gridsim()

produitMatrice(m1[],m2[],index): cette méthode calcule la colonne de la matrice de résultat qui a l'indice "index" qui est passé comme argument aux alongs des des matrices a multiplier .

2_main

c'est l'entité qui représente la machine qui est la destination que les utilisateurs envoient les résultats vers

2_1: ses attributs

M1[] et M2[]: ce sont les deux matrices qu'on veut multiplier

result[]:c'est la matrice qui contient le résultat de multiplication

2_2 ses méthodes

remplir_matrice []: remplit les matrices a multiplier, dans le cas ou le nombre de colonne et lignes de la matrice est supérieur à 2 , elle va être rempli avec des valeurs générés aléatoirement de 1 à 10 .

Distributiontaches[]: cette methode partitionne le travail entre les deux utilisateurs en fonction de nombre de colonnes de la matrice "result" et assignent a chacun sa partie de calcul

recuperer_resultat[]: récupere les resultats envoyés par les utilisateurs et l'affecte a la matrice resultat pour la remplir

afficher_resultat[]: affiche la matrice resultat remplis .

3_routeur

Déroulement

1

Au début, les deux matrices à multiplier sont remplies, soit manuellement soit de manière aléatoire si leur taille dépasse 2x2. Ensuite, ces matrices sont attribuées aux utilisateurs. L'entité main déduit la taille de la matrice résultante et l'initialise. Elle appelle ensuite sa méthode "DistributionTaches()" et attribue à chaque utilisateur sa partie de calcul en modifiant les valeurs de leurs attributs "begin" et "end".

2

Après le lancement de la simulation GridSim, chaque utilisateur appelle la méthode "body()" dès le début de la simulation. Ils invoquent ensuite la méthode "produitMatrice(m1[], m2[], index)" qui calcule la colonne "index" de la matrice résultante. Une fois calculée, la colonne est envoyée par valeur vers l'entité main dans un triplet (un tuple à trois valeurs). La première valeur du triplet représente le résultat calculé, la deuxième son indice de ligne, et la troisième son indice de colonne.

3

Cet appel se répète en boucle, où l'indice "index" prend des valeurs de "begin" à "end". En sortant de la boucle, chaque utilisateur a terminé son calcul et son envoi. Ensuite, ils ferment leurs sorties d'entrée/sortie. Les valeurs envoyées suivent leur chemin défini par la topologie des utilisateurs vers "r1", puis vers "r2", pour aboutir à l'entité main.

4

L'entité main, à son tour, après avoir reçu chaque valeur, envoie un accusé de réception à l'expéditeur, récupère la valeur ainsi que ses coordonnées, puis les affecte à la matrice "result" à leur emplacement spécifique. Une fois toutes les valeurs reçues et après un bref délai sans message, l'entité main ferme également ses sorties d'entrée/sortie et clôt la connexion.

5

La dernière étape consiste à afficher la matrice "result".

Interprétation

voila un exemple concret :

1

les deux matrices A et B qu'on veut multiplier :

- une matrice A de taille (2x4)
- une matrice B de taille (4x3)

2

le travail que chaque utilisateur a fait :

- on voit que l'utilisateur 1 a envoyé la première et la deuxième colonne après les calculer (dans notre programme c'est les valeurs (0,0) , (1,0) ,(0,1),(1,1))
- l'utilisateur 2 a envoyé la 3eme colonne (les valeurs (0,2) et (1,2)) , ce qui est tout a fait logique car on a 3 colonnes a calculer .

3

- **Resultat :**

la matrice résultante est de taille (2x3) , ce qui est juste ,et la matrice obtenue est juste par rapport au produit matricielle $A*B$ donc on conclut que les valeurs calculés par chaque utilisateur est juste et dans son bon emplacement , Alors le produit matricielle

Fin.