

APPLICATION PARALLELE DE GESTION DES MOYENNES

RÉALISÉ PAR :

MELLOULI MOHAMED ALI

TEBOUB ISLEM

TAKHERIST AYOUB

PLAN

2

- ▶ Introduction
- ▶ Cas d'utilisation d'une application parallèle
- ▶ Conception
- ▶ Technologies et outils à utiliser
- ▶ Développement de l'Application
- ▶ Discussions et critiques
- ▶ Conclusion et perspectives

Introduction

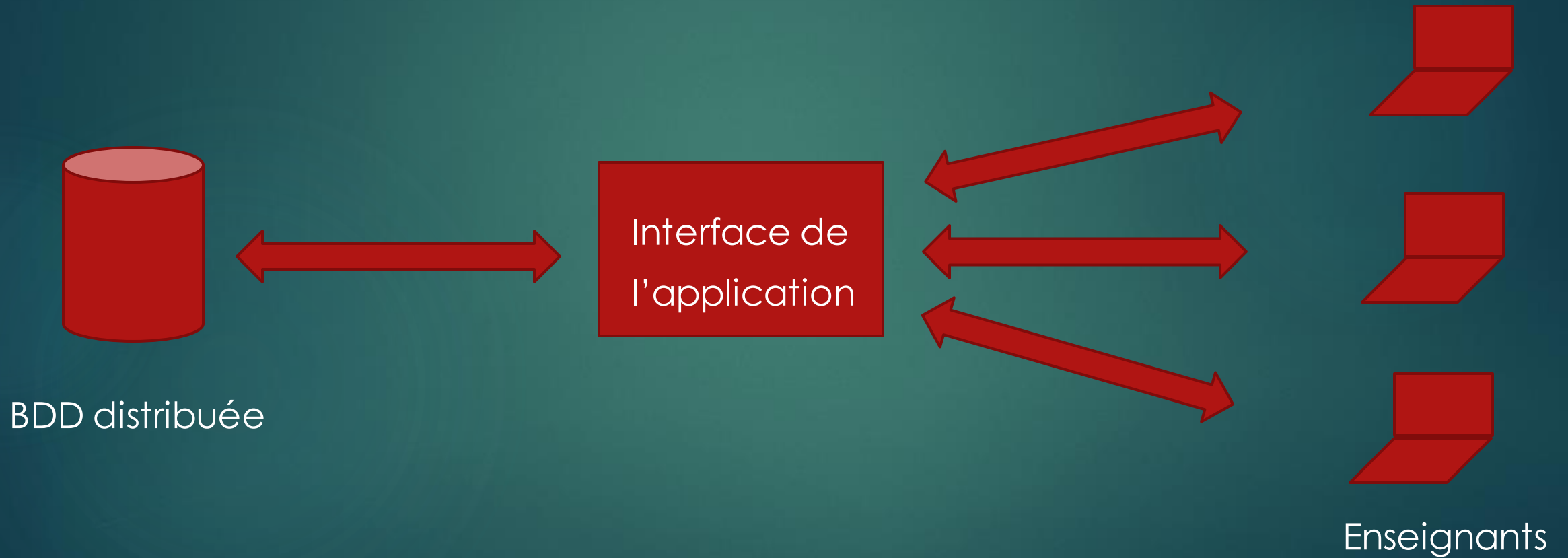
- ▶ De nos jours, les systèmes informatiques sont de plus en plus distribués, d'où la nécessité d'applications sous-jacentes assurant cette distributivité.
- ▶ Pour maintenir l'accessibilité, la répartition de charge et un temps de calcul optimal, on est obligé d'utiliser des applications parallèles.
- ▶ Cependant, il est nécessaire de penser aux problèmes issus de ce parallélisme tels que : la gestion d'accès, de cohérence et des synchronisations

Cas d'utilisation d'une application parallèle

- ▶ L'application qu'on propose est destinée à la gestion des moyennes des étudiants.
- ▶ L'application permet de:
 - ❑ Ajouter les informations d'un étudiant et sa moyenne respective.
 - ❑ Modifier les informations d'un étudiant ou sa moyenne respective.
 - ❑ Rechercher les étudiants ayant obtenue telle et telle moyenne.
 - ❑ Suppression des informations d'un étudiant et sa moyenne .
- ▶ On prend en considération les points suivants:
 - ❑ Les étudiants appartiennent à différentes filières.
 - ❑ Plusieurs enseignants peuvent accéder à l'application en même temps.

Conception

5



Technologies et outils à utiliser

6

- ▶ MySQL est choisit comme système de gestion de base de données relationnelles.
- ▶ WampServer pour l'utilisation de la base de données.
- ▶ Les requêtes SQL seront combinées avec le langage JAVA.
- ▶ Netbeans est choisit comme Platform de travail.

Développement de l'Application(1)

- ▶ Vu que l'application est accessible par plusieurs enseignants, il est nécessaire qu'elle supporte la distributivité des entités et/ou le partitionnement du code.
- ▶ Pour assurer ces caractéristiques, il est nécessaire de classer les fonctionnalités fournies en deux classes :
 - ❑ Fonction pouvant être accessible par plusieurs enseignants en même temps : telles que l'ajout, la suppression ou la recherche des étudiants
 - ❑ Fonction ne pouvant pas être accessible par plusieurs à la fois; dont la nécessité d'introduire une section critique, telle que la modification des étudiants.

Développement de l'Application(2)

- L'appel des fonctions d'affichage ou d'interrogation de donnée se fait par simple exécution de threads permettant un accès souple et multiple à ces mm fonctionnalités.

Développement de l'Application(3)

9

- Fonction d'insertion pour un accès simultanée et parallèle à la base de donnée.

```
new Thread(new Runnable() {
    @Override
    public void run() {
        start=System.currentTimeMillis();
        String id = txtid.getText();
        String nom = txtno.getText();
        String prenom = txtpr.getText();
        String branche = txtbr.getSelectedItem().toString();
        String note = txtnot.getText();
        String requete = "insert into etudiant(id,Nom,Prenom,branche,note)VALUES ('"
            + id + "','" + nom + "','" + prenom + "','" + branche + "','" + note + "')";
        try {
            stm.executeUpdate(requete);
            JOptionPane.showMessageDialog(null, "l'etudiant est bien ajouter");
            txtno.setText("");
            txtpr.setText("");
            txtbr.setSelectedItem(2);
            txtnot.setText("");
            afficher();
            clearFiled();
        } catch (Exception ex) {
            //JOptionPane.showMessageDialog(null, ex.getMessage());
        }
        end=System.currentTimeMillis();
        double t=end-start;
        System.out.println("temps d'ajoute etudiant a BDD+::" +t+"ms");
    }
}).start();
```

Développement de l'Application(4)

10

- ▶ L'exécution des fonctions de mise à jour (requêtes de modification de donnée) se fait via des threads synchronisés.
- ▶ L'utilisation de ces derniers permet une manipulation atomique de la donnée tout en assurant une accessibilité et une transparence d'accès.

Développement de l'Application(5)

11

- Fonction de modification pour un accès simultanée et synchronisé à la base de donnée.

```
private void afficher() {
    try {
        //-----//
        ///thread 1111111111

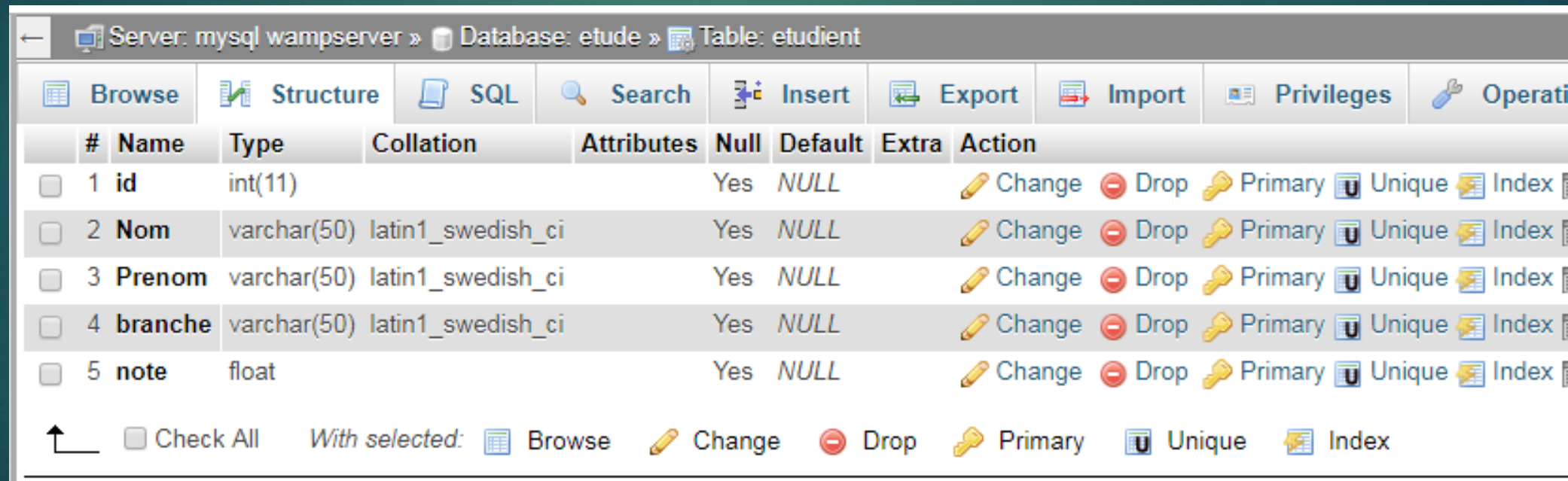
        Thread T1= new Thread(new Runnable() {
            @Override
            public void run() {
                try {

                    ///code affichage
                    model.setRowCount(0);
                    stm = conn.obtenirconnexion().createStatement();
                    ResultSet Rs = stm.executeQuery("Select * from etudiant");
                    synchronized(model)
                    {
                        while ((Rs.next())) {
                            model.addRow(new Object[]{Rs.getString("id"), Rs.getString("Nom"), Rs.getString("Prenom"),Rs.getString("br
                                , Rs.getString("note")});
                        }
                    }
                } catch (Exception e) {
                    //JOptionPane.showMessageDialog(null, "erreur de deplacement " + e.getLocalizedMessage());
                }
            }
        });
        T1.start();
    }
```

Développement de l'Application(6)

12

- La base de donnée est une simple table composée de 5 colonnes comme suit:



The screenshot shows the phpMyAdmin interface for a MySQL database named 'etude'. The 'Structure' tab is selected, displaying the table 'etudiant'. The table has 5 columns: 'id' (int(11)), 'Nom' (varchar(50)), 'Prenom' (varchar(50)), 'branche' (varchar(50)), and 'note' (float). Each column is marked as 'Yes' for 'Null' and 'NULL' for 'Default'. The 'Action' column for each row contains icons for 'Change', 'Drop', 'Primary', 'Unique', and 'Index'.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/>	2 Nom	varchar(50)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/>	3 Prenom	varchar(50)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/>	4 branche	varchar(50)	latin1_swedish_ci		Yes	NULL		Change Drop Primary Unique Index
<input type="checkbox"/>	5 note	float			Yes	NULL		Change Drop Primary Unique Index

At the bottom, there is a 'Check All' checkbox and a 'With selected:' section containing icons for 'Browse', 'Change', 'Drop', 'Primary', 'Unique', and 'Index'.

Développement de l'Application(7)

13

- L'interface applicative

Gestion des moyenneG

Matricule:

Nom :

Prenom:

Branche :

Moy :

matricule	nom	prenom	branche	MoyenneG
201402	teboub	islem	INFO	17.5
201403	mellouli	mohamed	INFO	17
201404	takherist	ayoub	INFO	17
201405	kebaili	khadija	SM	15
201303	mohamed	amine	INFO	14
201307	karim	mohamed	INFO	13
201209	benzarwel	walid	INFO	16
201210	kqrimi	abdellah	LGE	12
201214	mellel	mohamed	INFO	13.5
201215	abdelkader	amine	IGE	10.75
201245	smaïli	amina	INFO	11.75

Ajouter **actualiser** **modifier** **Supprimer**

recherche

Discussions et critiques

14

- ▶ Les besoins de cette application a mener de distinguer 2 types de parallélisme:
 - ▣ parallélisme d'exécution intra-ordinateur : permettant des exécutions de fonctionnalités indépendantes les unes des autres
 - ▣ parallélisme d'exécution inter-ordinateur : permettant l'accès simultané à une ou plusieurs fonctionnalités par différents tiers.
- ▶ En effet, l'interrogation simultanée de la base de donnée par différents tiers a réduit la complexité en temps d'accès.
- ▶ on a mesuré le temps d'accès pour les requêtes d'affichage, les résultats sont comme suit :
 $600\text{ms} < T(n) < 800\text{ms} \implies O(n) = 800\text{ms}$ où $1 \leq n \leq 12$
où n = nombre d'ordinateur accédant à la BDD simultanément
 $\text{Speedup} = T_{\text{seq}} / T_{\text{par}} = n \implies \text{efficacité} = 100\%$

Conclusion et perspectives

15

- ▶ L'objectif de ce travail été de réaliser une application orientée calcul parallèle, le cas d'utilisation été une application de gestion de moyenne, cette dernière est implémentée dans un environnement N-tiers distribué.
- ▶ La parallélisme de l'application est reflété par l'exécution simultanée de fonctionnalités de manipulation et d'interrogation relatives à la base de données implémentée.
- ▶ En terme d'application , on peut ajouter un processus d'identification des enseignants.
- ▶ En terme de fonctionnalité, on peut ajouter des fonctions pour le calcul de pourcentage de réussite ou pour la calcul de la moyennes générale en parallèle pour telle ou telle filière .

Merci pour votre attention