

BASE DE DONNÉE
AVANCÉE



2022-2023

COMPTE RENDU PROJET 1

Participants

Cheikh BOYE

Oumy DIANKHA

Fatou SYLLA





objectif du projet

Avec la montée des utilisateurs entraînant des problèmes de connexions et la mise en attente de certains clients, une entreprise décide d'investir dans une nouvelle infrastructure pour améliorer l'accès et la gestion de sa base. Pour cela une étude comparative entre plusieurs scénarios pour améliorer des performances sera faite .

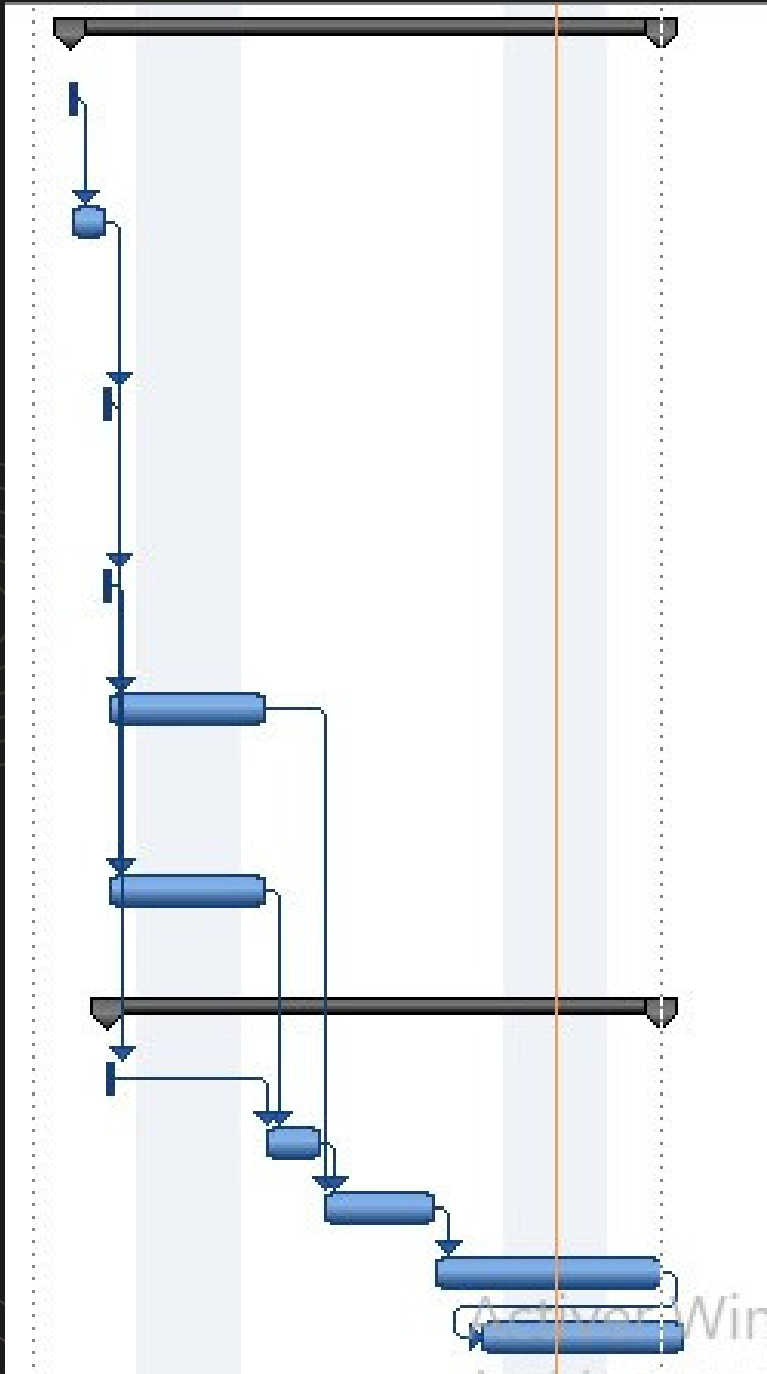
organisation de l'équipe

Cheikh BOYE: il gère la machine serveur donc il fait la configuration de APACHE PHP WORDPRESS et a le mysql qui contient la base de donnée de wordpress. Il travaille avec Ubuntu18 avec une machine virtuelle(VB) de 3Go de RAM, CP4 égale à 4 et une espace de 27 Go

Oumy DIANKHA: elle gère une machine qui mysql, fait la configuration de REDIS. Elle travaille avec Ubuntu18 avec une machine virtuelle(VB) de 3Go de RAM, CPU égale à 4 et une espace de 27 Go














Fatou SYLLA: elle gère une machine qui a mysql, fait la configuration de PROXYSQL. Elle travaille avec Ubuntu18 avec une machine virtuelle(VB) de 3Go DE RAM, CPU égale à 4 et une espace de 27 Go

Diagramme de Gant



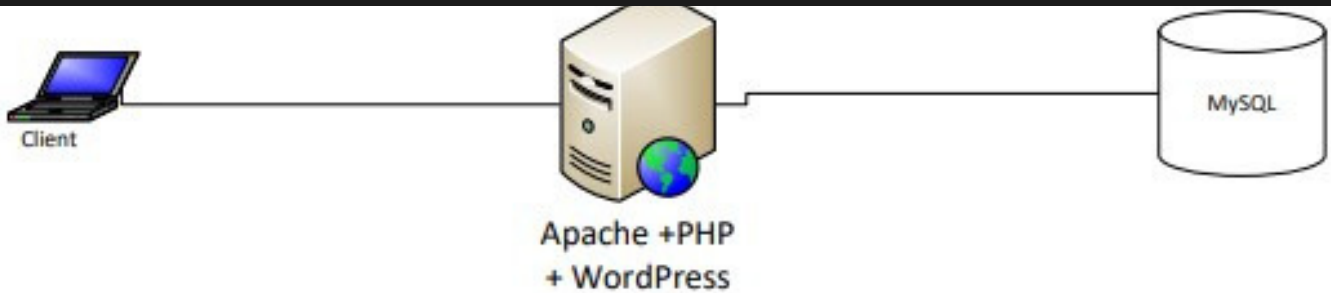
Voici le tableau des differentes taches faites avec la date de début et de la fin.

En dessus nous avons la présentation graphique (diagramme de gant) des différentes taches.

Mode Tâche	Nom de la tâche	Durée	Début	Fin	Prédécesseurs
	<input type="checkbox"/> installation et logiciel	6,21 jours	Jeu 02/03/23	Lun 13/03/23	
	installation et configuration de	30 min	Jeu 02/03/23	Jeu 02/03/23	
	installation et configuration de wordpress	30 min	Jeu 02/03/23	Ven 03/03/23	2
	installation et configuration de apach	20 min	Ven 03/03/23	Ven 03/03/23	3
	installation et configuration de	20 min	Ven 03/03/23	Ven 03/03/23	4
	installation et configuration de proxy	1 jour	Ven 03/03/23	Lun 06/03/23	3;4;5
	installatiob et configuration de	1 jour	Ven 03/03/23	Lun 06/03/23	3;4;5
	<input type="checkbox"/> differentes scenari	6 jours	Ven 03/03/23	Lun 13/03/23	
	scenario0	2 hr	Ven 03/03/23	Ven 03/03/23	5
	scenario1	1 jour	Lun 06/03/23	Mar 07/03/23	9;7
	scenario2	2 jours	Mar 07/03/23	Jeu 09/03/23	6;10
	scenario3	2 jours	Jeu 09/03/23	Lun 13/03/23	11
	rapport a rendre	12 hr	Ven 10/03/23	Mar 14/03/23	12

Paramètres de configuration

Scénario 0



Configuration de mysql

Après avoir installé mysql avec la commande ***apt-get install mysql***

Ici on a changé l'adresse du bind address qui était 127.0.0.1 en 0.0.0.0 pour que les autres machines puissent avoir accès à ce mysql. On a fait ainsi pour les deux autres machines.

```
instead of skip-networking the default is now to listen only  
on localhost which is more compatible and is not less secure.  
bind-address            = 0.0.0.0  
  
#fine Tuning
```

On se connecte à mysql en tant que root et de créer une base de données pour accueillir nos données WordPress.

mysql -u root -p.

On crée la base de donnée Wordpress avec la commande ***CREATE DATABASE Wordpress_db***
on accorde tous les privilèges et autorisations d'accès à l'utilisateur a la base de donnée



Installation des serveurs

APACHE

Apache est une option courante pour héberger un site Wordpress car il est compatible avec de nombreux système d'exploitation, offre de bonne performances, une sécurité élevée et une grande flexibilité de configuration.

On l'installe avec la commande ***apt-get install apache2***

PHP

PHP est utilisé pour traiter les requêtes de l'utilisateur, récupérer des données à partir de la base de données, générer du contenu dynamique et exécuter des scripts personnalisés.

On l'installe avec la comande ***apt-get install php php-mysql***.

Après l'intallation on entre dans le fichier de configuration et y ajouter ses lignes


```
<?php
phpinfo();
?>
```

WORDPRESS

WordPress est une plate-forme de gestion de contenu gratuite et open-source basée sur PHP et MySQL.

```
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress_db' );

/** Database username */
define( 'DB_USER', 'wp_user' );

/** Database password */
define( 'DB_PASSWORD', 'passer' );

/** Database hostname */
define( 'DB_HOST', '192.168.1.36' );

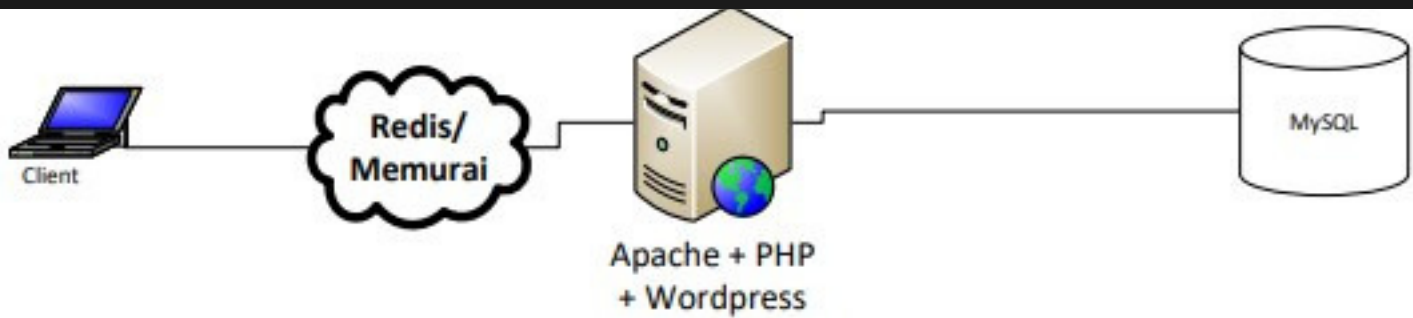
/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );
```

^G Aide	^O Écrire	^W Chercher	^K Couper	^J Justifier	^C Pos. cur.
^X Quitter	^R Lire fich.	^L Remplacer	^U Coller	^T Orthograp.	^_ Aller lig.

On renseigne les données de la base de donnée wordpress créée précédemment

Scénario 1



REDIS

l'association de Redis à WordPress permet de réduire les temps de réponse, d'améliorer la capacité de traitement des requêtes et de réduire la charge sur la base de données, ce qui peut aider à offrir une meilleure expérience utilisateur et à améliorer les performances globales du site.

Pour associer Redis à wordpress, on a téléchargé un script php puis on a déplacé le fichier dans le répertoire d'installation WordPress **/wp-content**

Ensuite on a activé les parametres de configurations de php dans **wp-config.php**.

On ajoute les lignes suivantes:

```
*/
define( 'WP_DEBUG', false );

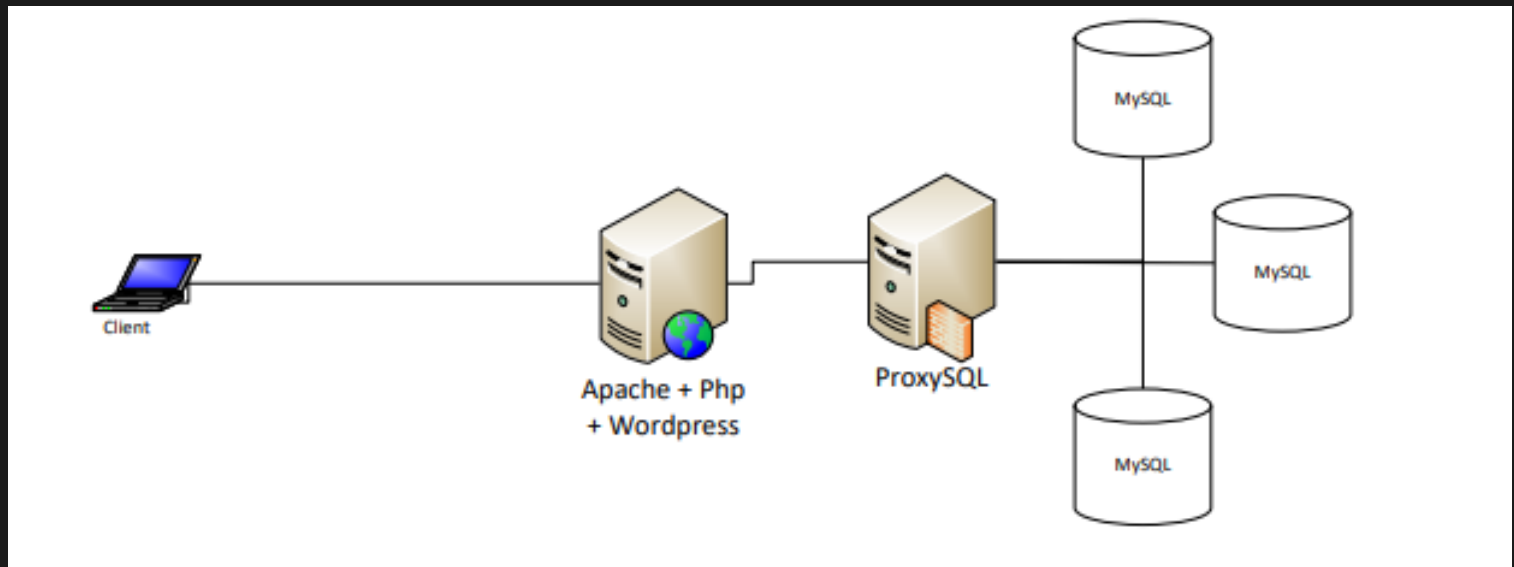
/* Add any custom values between this line and the "stop editing" line. */

define('WP_CACHE_KEY_SALT', '');
define('WP_CACHE', true);
define('WP_REDIS_HOST', '192.168.1.16');
define('WP_REDIS_PORT', '6379');
define('WP_REDIS_PASSWORD', 'passer');
define('WP_REDIS_SCHEME', 'tcp');
define('WP_ALLOW_REPAIR', true);

/* That's all, stop editing! Happy publishing. */

/** Absolute path to the WordPress directory. */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH', __DIR__ . '/' );
}
```

Scénario 2



PROXYSQL

l'association de ProxySQL à WordPress permet de distribuer la charge sur plusieurs serveurs de base de données, de réduire la charge sur chaque serveur individuel, d'améliorer les temps de réponse et de réduire la charge sur la base de données, ce qui peut aider à offrir une meilleure expérience utilisateur et à améliorer les performances globales du site.

On a téléchargé la dernière version de proxysql et l'installé à l'aide de dpkg

On se connecte à l'interface d'administration de ProxySQL avec le mot de passe par défaut 'admin'.

mysql -u admin -padmin -h 127.0.0.1 -P 6032

On crée un nouvel utilisateur 'monitoruser' avec le mot de passe 'passer' et on lui accorde les privilèges pour accéder au serveur et au la aussi creer au niveau de la base de donnees wordpress

```
mysql> select user from mysql.user;
+-----+
| user                |
+-----+
| monitoruser         |
| proxyuser           |
| replication_user    |
| wp_user             |
| debian-sys-maint    |
| mysql.session       |
| mysql.sys           |
| root                |
+-----+
8 rows in set (0,00 sec)
```

On va mettre en place une réplication mysql master slave. On entre dans le fichier de configuration mysqld.conf puis on décommente les lignes suivantes.

Pour la machine qui a la base de donnée on va mettre son serveur-id a 1 pour qu'il soit la machine master.

```
#      other settings you may need to change.
server-id                = 1
log_bin                  = /var/log/mysql/mysql-bin.log
expire_logs_days         = 10
max_binlog_size          = 100M
#binlog_do_db             = include_database_name
#binlog_ignore_db         = include_database_name
#
# * ToonDB
```

Toujours dans mysqld.conf on met le serveur-id a 2 pour que la machine soit une machine slave et l'autre machine aussi le serveur-id a 3 pour qu'elle soit slave

```
# Note: if you are setting up a replication slave, you
#       other settings you may need to change.
server-id                = 2
log_bin                  = /var/log/mysql/mysql-bin.l
expire_logs_days         = 10
max_binlog_size          = 100M
#binlog_do_db             = include_database_name
#binlog_ignore_db         = include_database_name
```

On fait un dump de nos bases de donnée pour l'injecter à la machine esclave et à noter l'index du log binaire qui servira pour la réplication.

```
mysql> show master status;
+-----+-----+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000019 | 154 | | | |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

On voit que toutes les machines sont connectées

```
mysql> select * from mysql_servers;
```

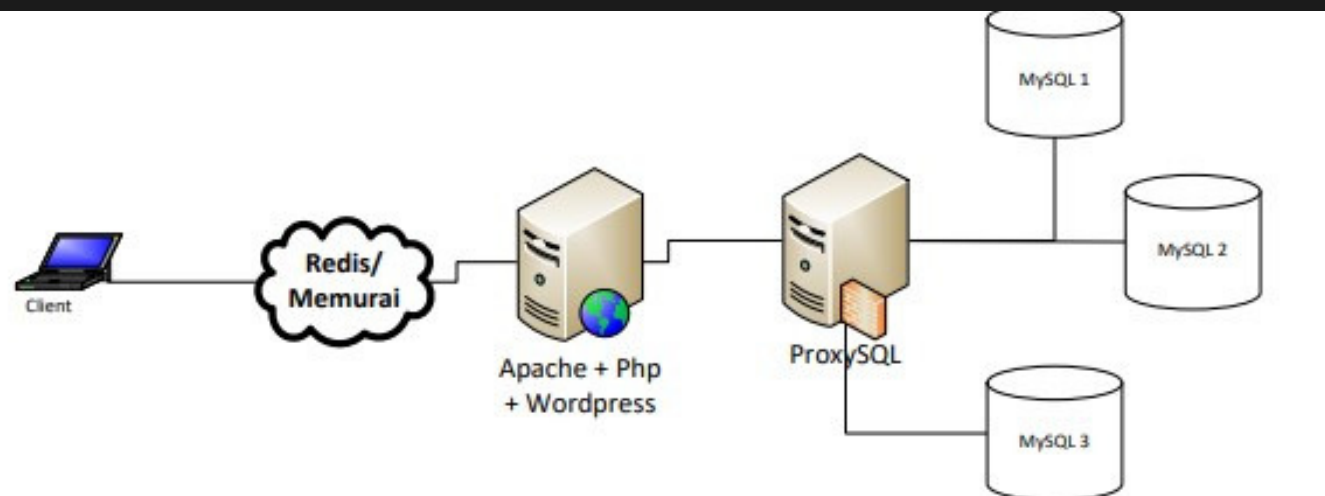
hostgroup_id	hostname	port	gtid_port	status	weight	compression	max_connections	max_replication_lag	use_ssl	max_latency
10	192.168.1.36	3306	0	ONLINE	1	0	1000	0	0	0
20	192.168.1.36	3306	0	ONLINE	1	0	1000	0	0	0
20	192.168.1.12	3306	0	ONLINE	1	0	1000	0	0	0
20	192.168.1.25	3306	0	ONLINE	1	0	1000	0	0	0
20	192.168.1.21	3306	0	ONLINE	1	0	1000	0	0	0

```
5 rows in set (0.00 sec)
```

Scenario 3

Pour cette architecture on a combiné les deux scénarios 1 et 2 pour pouvoir bénéficier leurs avantages.

Ici il n'y a pas de configuration à faire, il suffit juste de mettre en marche les deux serveurs redis et proxysql.



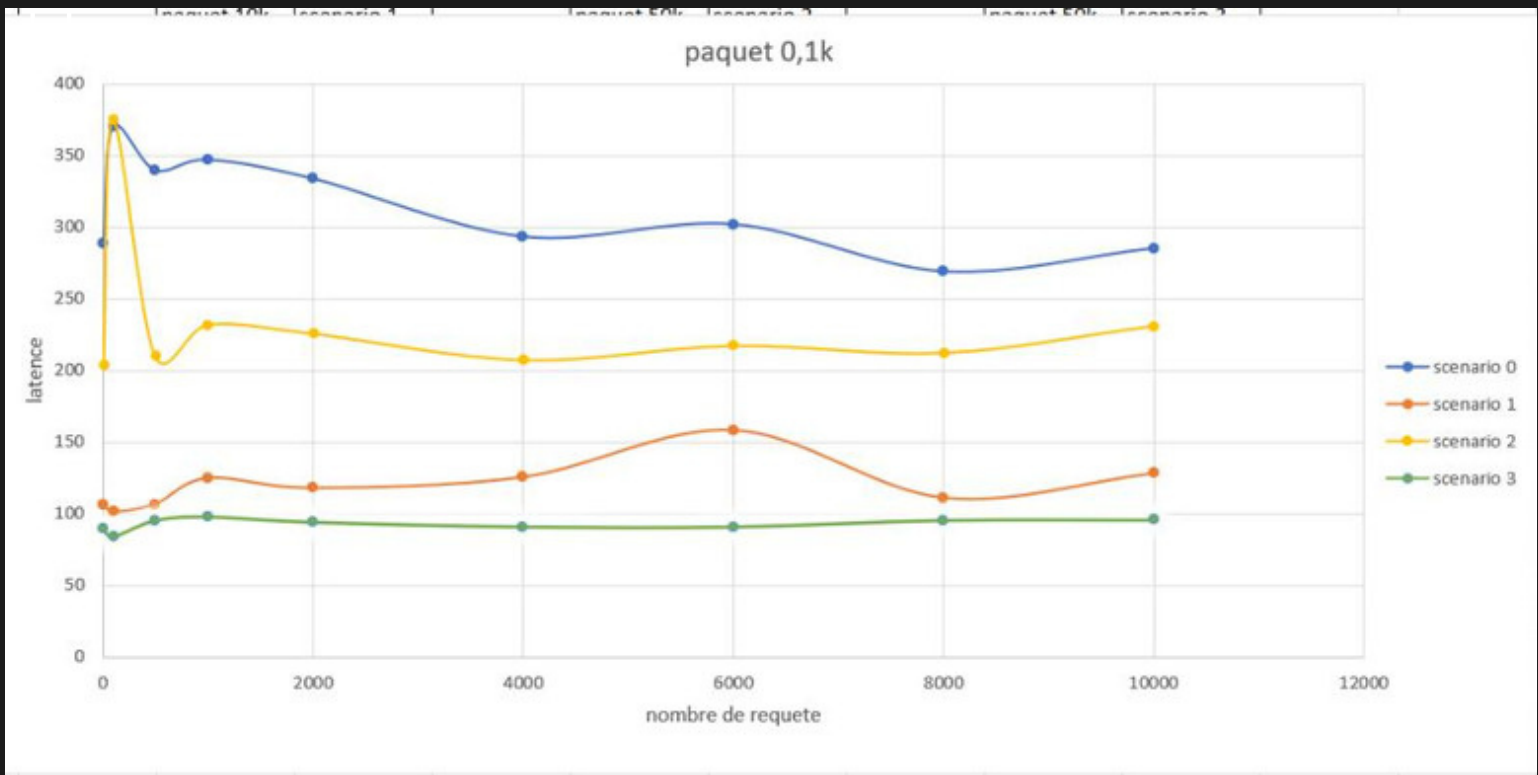
Test et diagramme des resultats

Pour tester les différents scénarios, on a fixé le nombre de paquets donnés (0.1Ko, 10Ko et 50Ko) ensuite on a envoyé différentes requêtes pour regarder la variation du temps de latence.

C'est ainsi après avoir récupérer les resultats, on a fait leurs représentations graphiques à l'aide de excel.

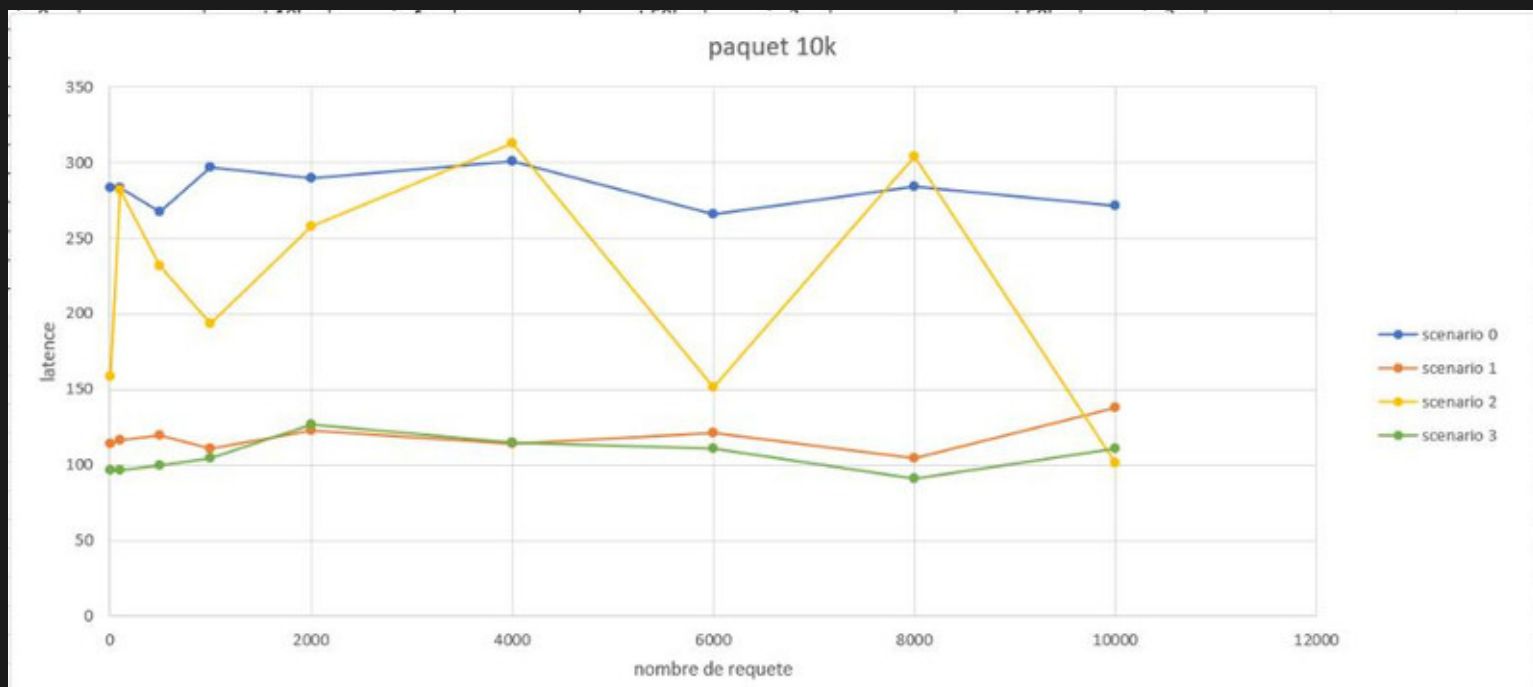
Voici ci-dessous les diagrammes obtenus :

- Pour des packets de 0.1Ko



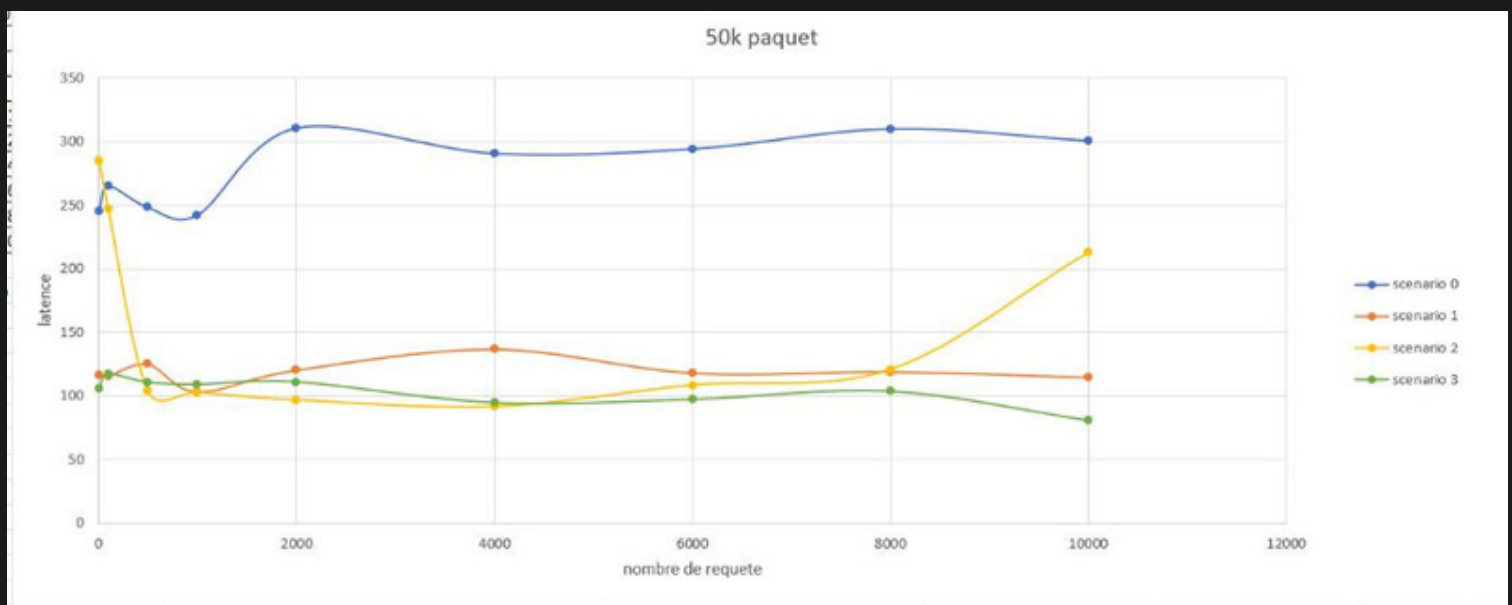
Ici on constate que le temps pour l'acheminement des requets est plus faible avec le seueur redis qu'avec le serveur proxysql mais en combinant les deux serveurs on gagne encore plus en temps

- Pour des packets de 10Ko



Si on augmente le nombre de paquets, on a les memes évolutions que le premiers test juste que le temps de latence pour le scénario 2 et le scénario 3 sont sensiblement égal mais au fur et a mesure que le nombre de requetes augmente le temps de latence est plus petit pour le scénario 3.

- Pour des packets de 50Ko



Pour 50Ko, on constate toujours les memes évolutions que les scenarios precedent, le scénario3 prend toujours le moins de temps que les autres avec l'augmentation des requetes.

CONCLUSION

Le serveur Redis est moins coûteux en terme de déploiement que le serveur proxy et en se basant sur les tests il permet des temps de réponses plus rapides en stockant les données en mémoire vive.

En somme pour répondre aux besoins d'amélioration de performance de son site web, on peut proposer à l'entreprise de choisir l'architecture du scénario 3 qui offre un temps de réponse plus rapides en stockant les données en mémoire vive et permettant aussi une meilleure gestion des données très volumineux grâce à la base de donnée distribué.