R301 - Développement efficace

Ibrahim Ahamada, Noan LISSILLOUR, Maxime PIAU, Rafaël BAUDRY

Lien vers le dépot Git

1. Mise en place de méthodes pour montrer que l'application est correcte :

justifier à partir de vos choix de conception que vos fonctionnalités sont celles attendues
1/ indiquer les fonctionnalités attendues et leur précondition (conditions nécessaires d'emploi)

Fonctionnalités :

- **Création de produits :**
  Précondition :
  Pour créer un produit, il faut que l'administrateur fournisse un nom, une description, une image (au format png ou jpg), l'identifiant de la collection auquel il appartient, des informations sur le modèle : la couleur et les mot-clés associés au produit pour la recherche, un prix et la quantité d'exemplaires du produit présent en stock.

  Les informations fournis doivent respecter des contraintes :
  - contraintes de types de données, les données doivent être du bon type, ex : l'administrateur doit entrer un entier pour la quantité, (On peut modifier le formulaire au niveau html pour envoyer autre chose si on le veut donc on a mis en place des vérifications au niveau backend du site).
  - Accès limité, pour pouvoir créer un produit, l'utilisateur doit posséder le rôle d'Admin

  Résultat obtenu :
  Le produit est ajouté dans la base de données

- **Affichages des produits**
  Préconditions:
  Pour afficher les produits il n'y a pas besoin de préconditions sauf dans le cas où l'utilisateur veut trier les produits. Dans ce cas, il doit compléter le formulaire de tri.
  Résultats obtenus :
  L'utilisateur obtient un ensemble de produits triés par ses choix dans la cas ou il en a fait. Sinon il a accès au catalogue entier

- **Catalogue des produits**
  Précondition : Aucune
  Résultat : Affiche les coques disponibles.

- **Recherche et filtrage**

Précondition : L'utilisateur fournit les informations pour sa recherche tel que le nom du produit, le prix en entrant un minimum et un maximum, la collection, etc…

Objectif : Permettre aux utilisateurs de trouver rapidement une coque spécifique en fonction de leurs critères (le nom, le prix, la collection, …).

- **Fiche produit détaillée**
  Précondition : Les données des produits doivent être complètes et fiables.
  Objectif : Fournir des détails sur chaque coque (image, collection, description).

- **Panier d'achat**
  Précondition : Les utilisateurs doivent être connectés à leur compte pour accéder et ajouter des produits au panier et il faut que le produit ne soit pas en rupture de stock.
  Objectif : Permettre aux utilisateurs de stocker plusieurs articles avant de passer la commande.

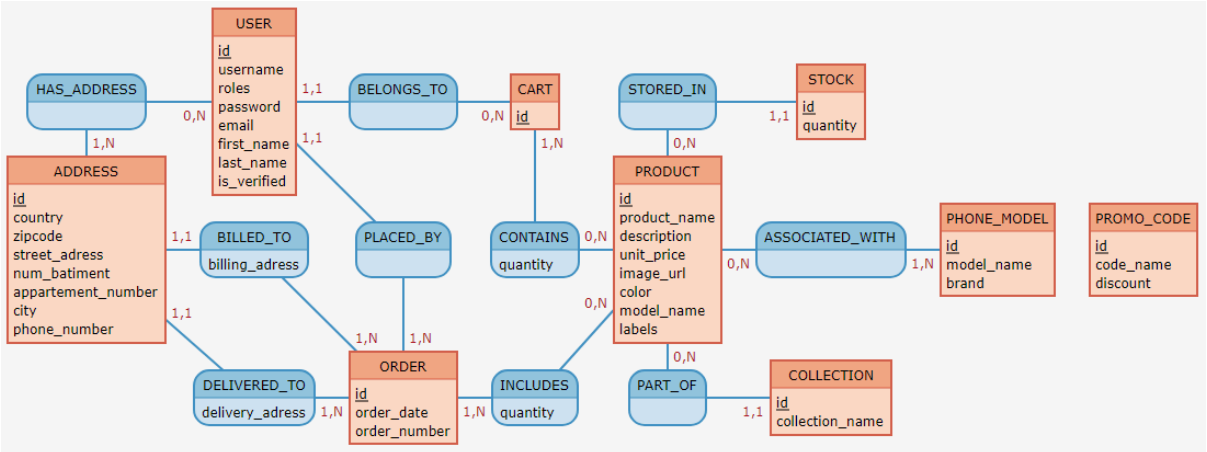
- **Gestion des comptes utilisateurs**
  Précondition : Une base de données pour gérer les utilisateurs (enregistrement, connexion, mot de passe) doit être disponible.
  Objectif : Permettre
- **Interface d'administration**
  Précondition : Il faut implémenter un accès sécurisé pour les administrateurs.
  Objectif : Permettre aux administrateurs de gérer les produits, les commandes, les utilisateurs et les codes de promotions.

- **Page web ergonomique**
  Précondition : Le site doit être accessible avec tout type d'appareils mobiles et optimisé pour un chargement rapide.
  Objectif : Permettre aux utilisateurs de naviguer facilement et d'avoir une vue d'ensemble des produits disponibles.

2. Montrer que l'application n'est pas coûteuse en temps d'exécution et en mémoire :

1/ Justifier le traitement optimal des données, lors de la construction des données (normalisation 3NF), par le SGBD choisi, par des procédures ad hoc.

Pour le traitement optimal de données, nous utilisons le SGBD MariaDB qui offre de très bonnes performances, une gestion efficace des requêtes tout en étant entièrement open source et sécurisé. Nos tables et relations respectent la normalisation 3NF.



Nous avons mis en place des déclencheurs afin de :

- **Gérer les données entrant dans les tables** : Les déclencheurs permettent de valider et de contrôler les données insérées ou mises à jour, garantissant ainsi la cohérence des informations dans le système.
- **Faciliter les mises à jour dans les tables** : Par exemple, lorsqu'un utilisateur est supprimé, les déclencheurs assurent automatiquement la suppression des données associées dans les autres tables (comme les paniers, commandes, et adresses).

Cela réduit considérablement la charge sur le serveur PHP, car il n'a plus besoin d'exécuter plusieurs requêtes distinctes pour effectuer ces actions. Le système devient ainsi plus efficace et mieux optimisé, tout en minimisant le risque d'erreurs ou d'incohérences dans la gestion des données.

Déclencheurs :

Supprimer un utilisateur : Supprime les entrées liées à un utilisateur dans les tables `cart`, `order`, et `adress` après sa suppression.

Limitation à 10 exemplaires d'un produit dans le panier) : Empêche l'insertion d'une quantité supérieure à 10 ou inférieure à 1 d'un produit dans le panier.

Limite de 20 articles différents par panier : Vérifie que le panier d'un client ne contient pas plus de 20 produits différents.

Limitation à 5 adresses par utilisateur : Empêche l'ajout d'une adresse si l'utilisateur a déjà 5 adresses enregistrées.

Ces limitations et mécanismes sont mis en place pour garantir l'intégrité des données dans la base de données tout en évitant des comportements non désirés ou des abus, tels que :

- **Surcharge du panier** : Empêcher un utilisateur de stocker un nombre excessif d'articles ou de quantités dans un panier, ce qui pourrait nuire à la performance du système.
- **Multiplication d'adresses** : Éviter qu'un utilisateur n'enregistre un nombre disproportionné d'adresses inutiles, préservant ainsi l'espace de stockage et simplifiant la gestion des données.
- **Cohérence des données liées** : Assurer que lorsqu'un utilisateur est supprimé, toutes les informations associées (paniers, commandes, et adresses) soient automatiquement nettoyées pour éviter les données orphelines.

Ces mesures contribuent à maintenir un système efficace, sécurisé et optimisé pour les utilisateurs, tout en respectant les bonnes pratiques de gestion des bases de données.

Procédures :

Création d'une commande : Crée une nouvelle commande en transférant les articles du panier vers la table `order` et en supprimant ces articles du panier.

Cette procédure nous garantit un processus fluide et cohérent de la création d'une commande effectuée par un utilisateur, en veillant à ce que les articles du panier soient bien enregistrés comme partie intégrante de la commande. Elle réduit également le risque de doublons ou d'erreurs en supprimant automatiquement les articles du panier une fois la commande validée.

Ajout d'un produit : Ajoute un nouveau produit dans la table `product` et gère la quantité correspondante dans la table `stock`.

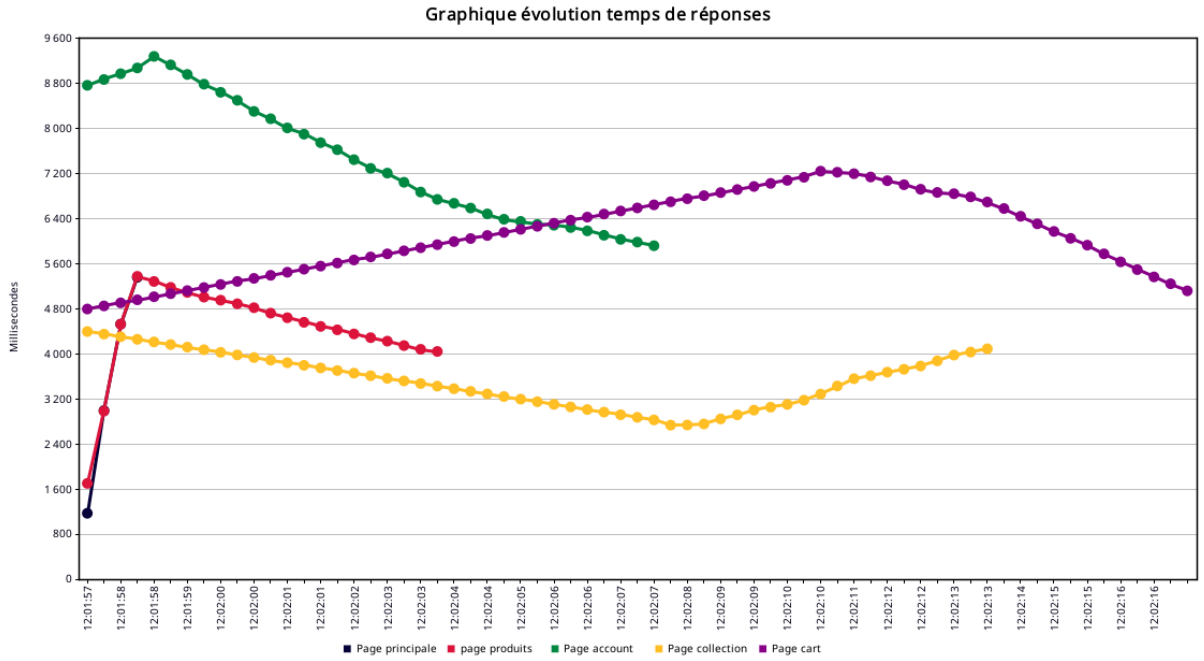
Cette procédure assure que chaque nouveau produit est correctement enregistré avec ses détails (nom, description, prix, etc.) et synchronisé avec les informations du stock du produit (le stock d'un produit, étant séparé de la table produit). Cela permet de maintenir une gestion cohérente et à jour des produits disponibles à la vente, tout en évitant des incohérences dans les données.

2/ Justifier le cas échéant, que le serveur absorbe bien la montée en charge (ou des pics) des ventes sans dégrader les temps de réponse aux clients ;

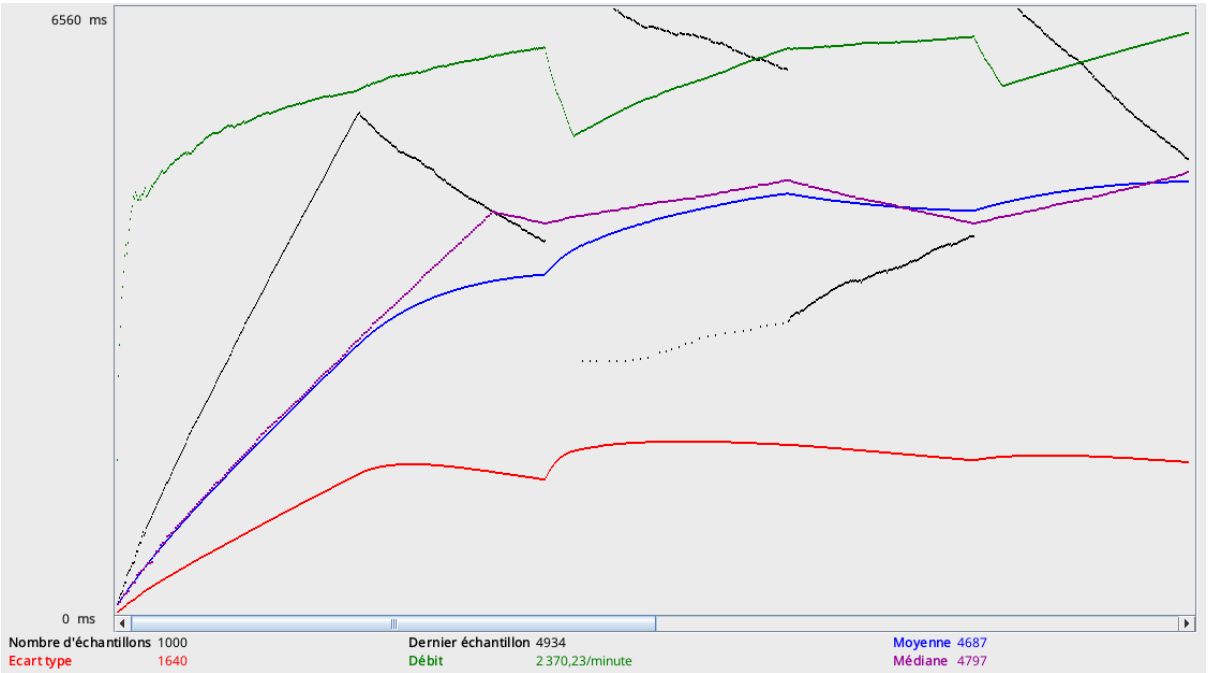
Les test sont réalisé vers une machines possédant 8 Gib de ram, 1 physical CPU et 4 virtuals CPU (serveur hébergeant le site web)

Nous avons donc fait des tests avec JMeter :

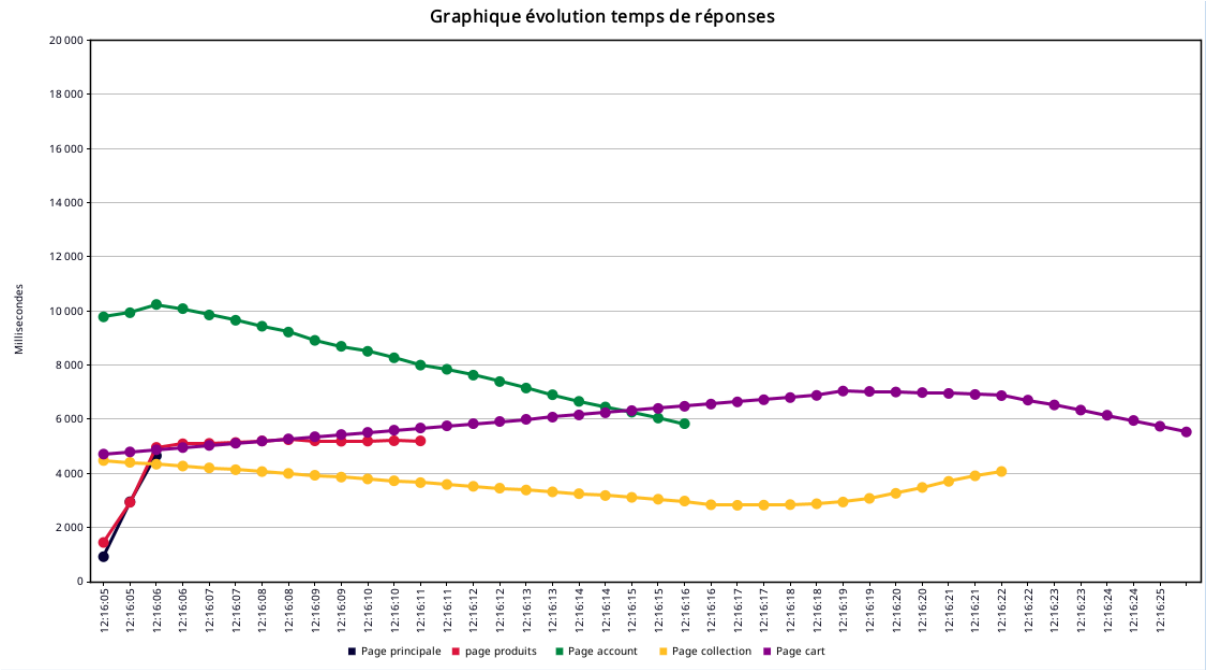
2.1 / Test avec 200 unités, durée de monté 1s, 1 itération et 20 produits dans la BDD



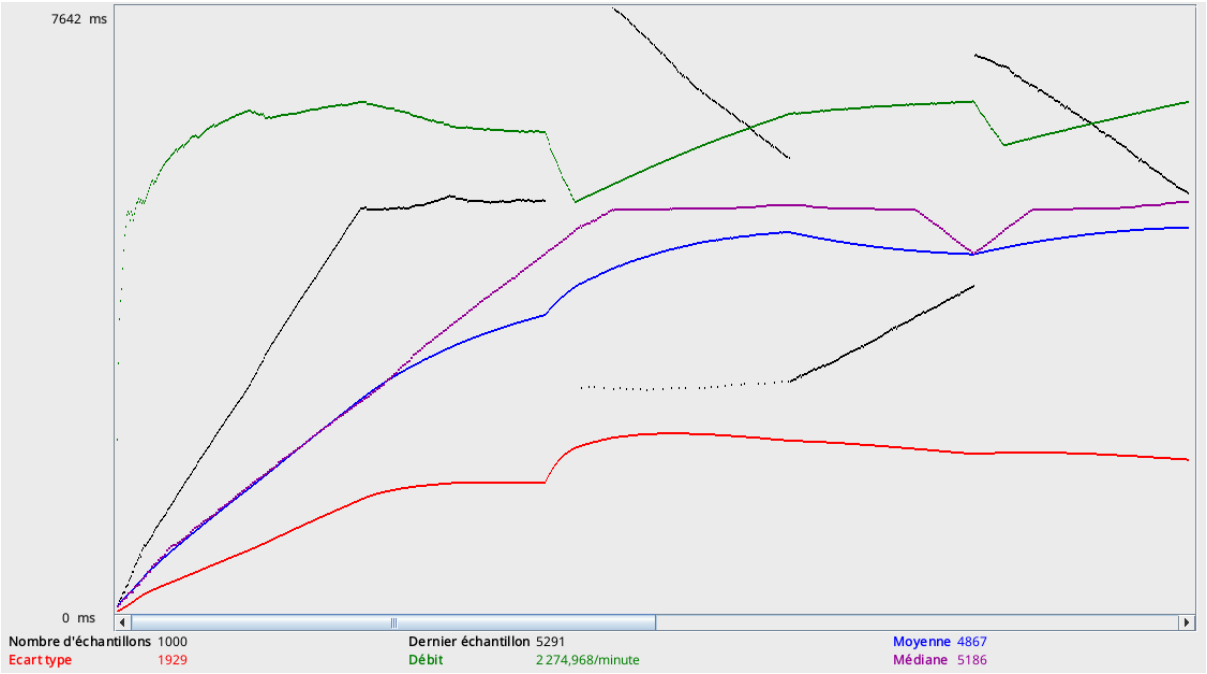
Pour toutes ces requêtes voici le données relatives aux temps de réponse(en ms) en ordonnée par rapport au temps(en abscisse)



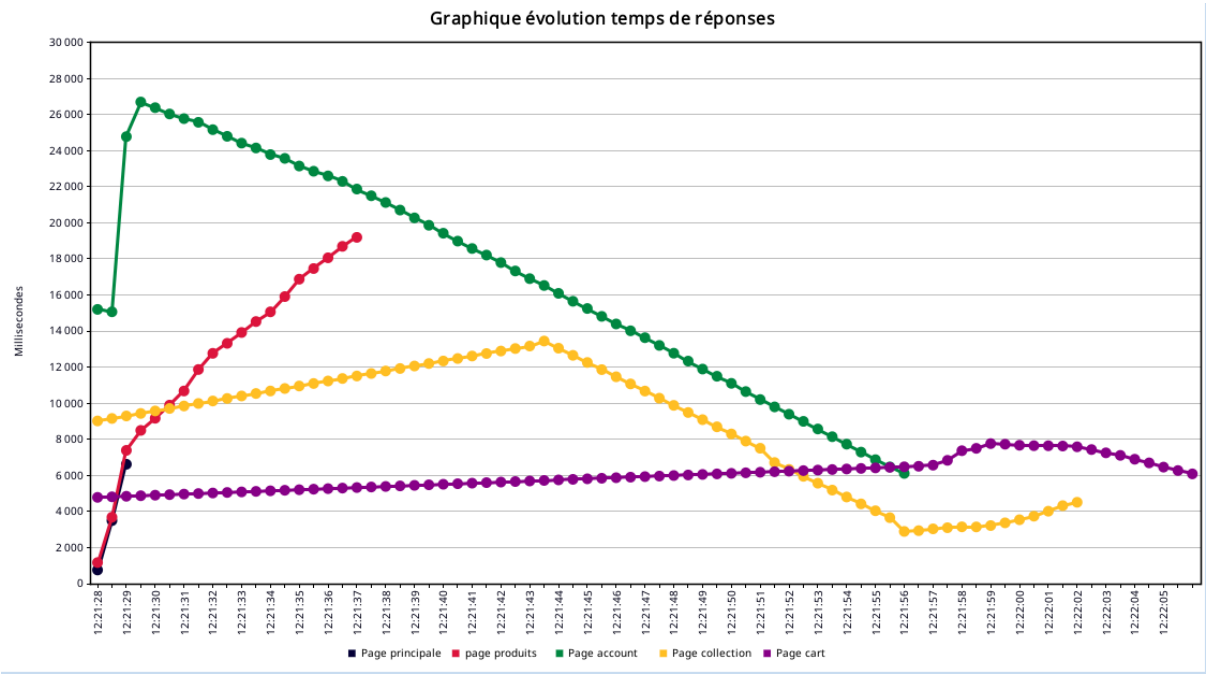
2.2 / test avec 200 unités, durée de monté 1s, 1 itération et 500 produits dans la BDD



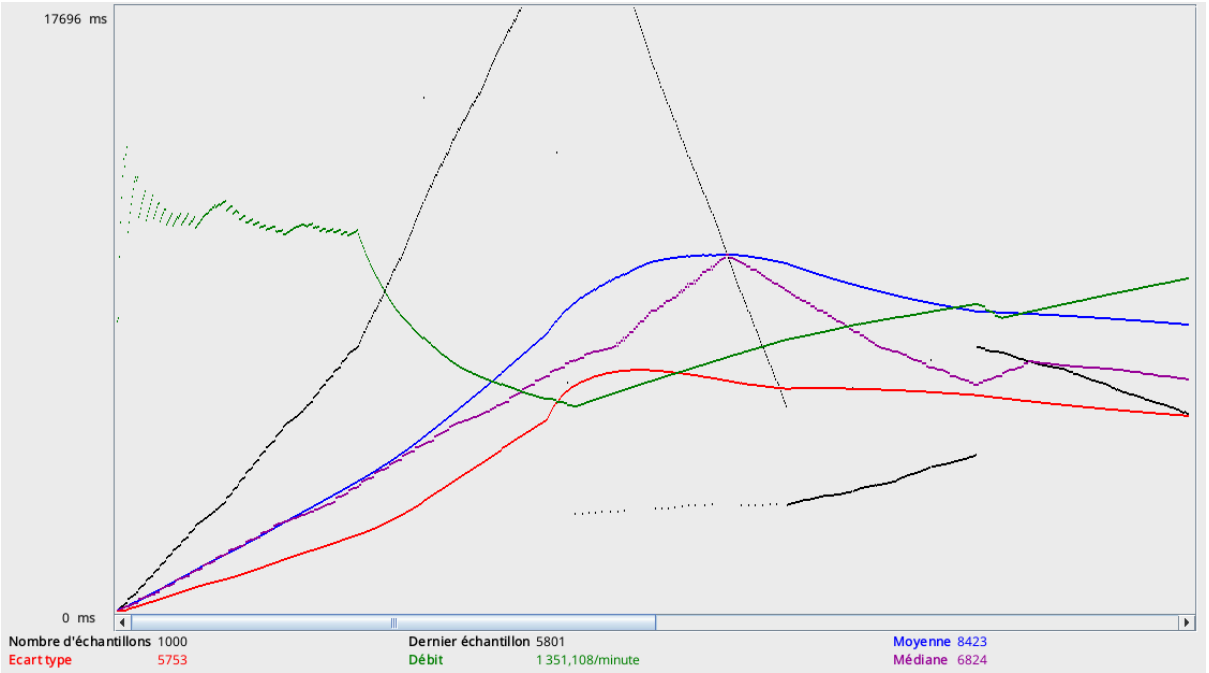
Pour toutes ces requêtes voici le données relatives aux temps de réponse(en ms) en ordonnée par rapport au temps(en abscisse)



2.3 / test avec 200 unités, durée de monté 1s, 1 itération et 10000 produits dans la BDD



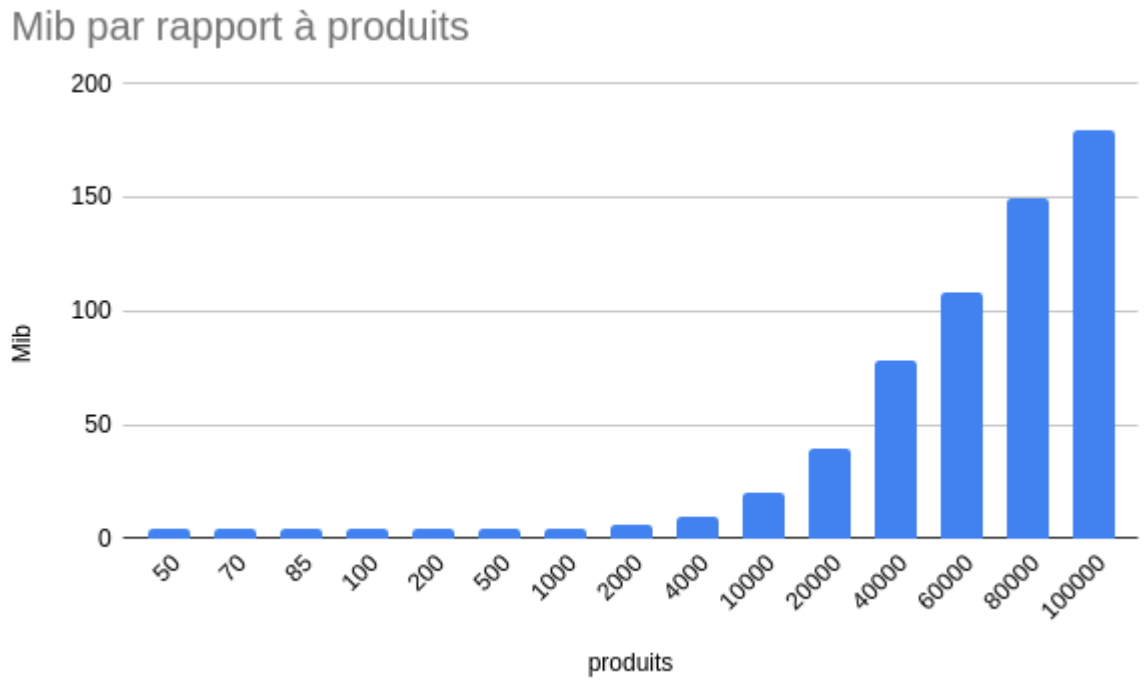
Pour toutes ces requêtes voici le données relatives aux temps de réponse(en ms) en ordonnée par rapport au temps(en abscisse)



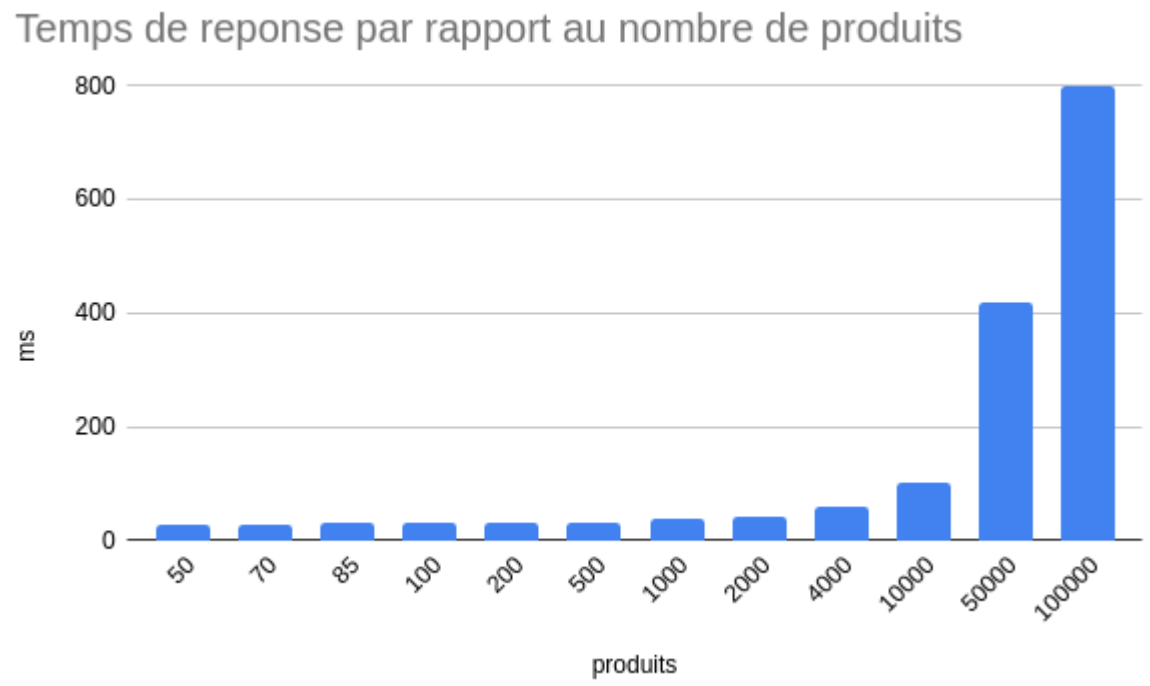
Via ces graphiques, on peut constater que plus il y a de produits de la base de données, plus les temps de réponse deviennent importants. Il existe aussi la contrainte du nombre d'utilisateurs qui peut saturer la machine et augmenter le temps de réponse, ce qui est directement lié aux performances de la machine hébergeant le site web.

Nous avons aussi analysé uniquement la page products en variant le nombre de produit présent dans la base de donnée avec les information fournies par symfony :

La consommation de RAM par rapport au nombre de produits présents :



Le temps de réponse par rapport au nombre de produits présents :



identifier et lister les points de votre application qui sont susceptibles de consommer du temps (selon les données et leur traitements), les affichages des vues et leurs enchaînements, (par exemple des images lourdes/légères)

La partie de notre site qui est susceptible de consommer le plus de ressources est lors du chargement des produits dans la page ou tous les produits sont listés (affichage de la vue produit/index.html.twig). Pour limiter ce problème, nous utilisons de la pagination afin de limiter le nombre de produits affichés et donc le chargement d'images pour permettre de réduire le temps de traitement côté client.
La deuxième partie qui est susceptible de consommer le plus de ressources est la recherche ou le filtrage sur un grand nombre de produits. Cela pourrait solliciter fortement la base de données. Pour éviter ce problème, nous limitons les recherches sur un grand nombre de produits, avec des résultats précis.

Identifier des situations éventuelles de montée en charge - justifier.

Les situations éventuelles de montée en charge peuvent survenir lors d'une augmentation de la fréquentation du site durant les périodes de Noël par exemple, ou encore l'arrivée d'une nouvelle collection. Une augmentation du nombre de requêtes pourrait entraîner un ralentissement du site.
Un ajout important de produit pourrait créer un ralentissement de la base de données comme le prouve les analyses effectué précédemment avec JMeter

Montrer que le plan de dimensionnement (des serveurs application et données) permet de stocker convenablement les données dans le temps.

Nous avons 2 machines virtuelles (une pour le site web et une pour la base de données). Nous sommes alors sur une plateforme "distribuée". C'est-à-dire que les ressources allouées à chacune des machines leur sont intégralement réservées.
Cela apporte donc plusieurs avantages. Chaque machine dispose de ses propres ressources (CPU, RAM, stockage…) et donc si une des 2 machines ralenties, l'autre ne subira pas les conséquences de ce ralentissement.
Chaque machine étant indépendante, on peut les maintenir à jour séparément.
Il est également très facile d'augmenter les ressources de chacune des machines (par exemple si une machine a besoin de plus de RAM).
En utilisant la base de données sur une machine virtuelle dédiée, nous bénéficions de fonctionnalités comme les sauvegardes régulières, essentielles pour garantir la disponibilité des données, la gestion optimale des données à mesure que la base s'agrandit et une sécurité accrue des informations de nos utilisateurs : en cas de compromission de la VM hébergeant le site web, les données des utilisateurs restent protégées sur la machine dédiée.

Expliquer l'organisation des données et le déploiement sur le ou les serveurs en rapport avec les SGBD.

Pour l'hébergement du SGBD, il se fait sur une seule machine externe à celle qui héberge le site afin d'avoir le maximum de performance et de sécurité. La BDD est hébergée sur une machine possédant 4Gib de ram, 1 physical CPU et 4 virtuals CPU

Expliquer et montrer ce que vous avez prévu ou comment vous prévoyez le dimensionnement afin d'avoir une application bien dimensionnée pour les données et en conséquence optimale en temps

Pour prévoir le dimensionnement de notre site web, il nous a fallu faire une estimation du nombre de client que notre site accueillerait
Nous avons pensé à mettre en place plus de serveur dans des périodes ou l'on pourrait rencontrer une plus grande fréquentation de notre site web (lors des périodes de fêtes ou de promotions).

Pour prévoir le dimensionnement de notre site web, nous avons commencé par estimer le nombre de clients susceptibles de visiter notre plateforme. Cette analyse nous a permis d'identifier des périodes de forte affluence, comme les fêtes ou les promotions. Afin de garantir des performances optimales durant ces moments critiques, nous allons déployer des serveurs supplémentaires pour anticiper cet afflux de personnes et maintenir une expérience utilisateur optimale.

Le choix d'hébergement éventuel du serveur des données (pour un déploiement final) impacte-t-il et comment, les temps de réponse des fonctionnalités ?

Oui, le choix d'hébergement du serveur des données impacte directement les temps de réponse des fonctionnalités. Pour garantir des temps de réponse optimaux, il nous faut choisir un hébergement adapté à nos besoins : localisation proche des utilisateurs (le moins loin possible), infrastructure performante, bande passante suffisante et capacité à gérer la montée en charge ou des pics

3. Evaluation : Votre rapport de la SAE doit contenir explicitement les éléments permettant d'évaluer/noter les points précédents liés, au paragraphe R3.02 du sujet qui vous a été donné.

Sources:

- Comment dimensionner une plateforme informatique
- Guide de dimensionnement du serveur - MyWorkDrive