

CODAZZI Rama
DAMOTTE Alan
GUI Kai
HAMMERER Jérémy
TORCK Quentin

RICM5

Rapport de charges

Groupe 2 – Site de vente de stickers personnalisés

My Stick It



RESUME	3
CHOIX DES SCENARIOS	4
SCENARIO 1	4
SCENARIO 2	5
SCENARIO 2 BIS	5
SCENARIO 3	6
SCENARIO 4	7
SCENARIO 5	8
SCENARIO 6	9
SCENARIO ANNEXE (APPLICATION DEPLOYEE SUR LE CLOUD)	10

Résumé

Ce rapport contient les résultats de tests de charge effectués autour de divers scénarios. Ces tests ont permis de confirmer le bon fonctionnement de notre application, ainsi que les points sensibles sur lequel il faudrait certainement retravailler afin d'améliorer les performances.

De manière globale, les tests effectués sont relativement positifs et montre que l'application se comporte bien tant que le nombre d'utilisateurs simultanés ne dépasse pas les limites que nous avons fixées à notre application.

Nous avons notamment observé l'influence de l'utilisation d'un pool de connexion (BoneCP) et montré les limites de ce dernier. En effet, vous pourrez voir sur certains tests, que la limite du nombre de connexions simultanées a une influence sur le temps de réponses moyens des différentes pages de notre application. Cette limite (à 20 connexions) est raisonnable pour un développement et des tests en local, mais serait évidemment à élever lors du déploiement sur le cloud.

A noter que ces tests ont été effectués sur une machine en local et les résultats sont moins bons que sur un serveur dédié. Pourquoi ce choix ? Car le déploiement sur le Cloud n'était pas disponible tout de suite et surtout la base de données SQL (de base) fournie par Azure n'offre que 4 connexions simultanées ce qui est relativement limité pour effectuer des tests de charges. Toutefois, vous trouverez en fin de rapport un scénario de test annexe afin de montrer le fonctionnement de JMeter avec l'application déployée sur le cloud.

Choix des scénarios

Afin de réaliser ce rapport de charge, nous avons mis en place différents scénarii qui pourraient paraître intéressants à étudier. Voici les différents scénarii et les résultats. Tous les tests de charges ont été établis avec l'outil Apache JMeter et permettent d'étudier le comportement de l'application lors de son utilisation par un nombre significatif de clients potentiels.

Scénario 1

Il s'agit ici du scénario principal permettant de tester les fonctionnalités principales de notre application.

Pour ce scénario nous avons simulé 10 achats consécutifs. Pour ce faire, nous avons établis un contrôleur enregistreur sur Apache Jmeter permettant d'enregistrer une suite d'action effectuées sur notre application.

Le scénario est le suivant : l'utilisateur est déjà présent sur le site et son panier n'est pas vide. Il consulte alors son panier, supprime le produit existant (s'il y en a un), consulte le catalogue, sélectionne un produit et l'ajoute au panier. Ensuite, en consultant son panier, il veut procéder au paiement, et pour ce faire, il doit alors se connecter et recommencer l'action de paiement. Enfin il réalise les différentes étapes de paiement.

Voici le rapport agrégé obtenu avec JMeter :

Libellé	# Echantillons	Moyenn e	Min	Max	Ecart type	% Erreur	Débit	Ko/sec	Moy. octets
27 /ecom/cartManagement	10	353	4	3479	1041,73	0,00%	0,5	5,94	12060,7
28 /ecom/removeFromCart	10	11	8	24	4,34	0,00%	0,6	7,51	12590,2
30 /ecom/catalog	10	50	10	373	107,6	0,00%	0,6	7,52	12588,1
33 /ecom/productView	10	54	14	363	102,82	0,00%	0,6	7,94	13019,1
34 /ecom/addToCart	10	16	12	24	4,08	0,00%	0,6	8,19	13131,1
38 /ecom/cartManagement	10	11	7	23	4,65	0,00%	0,6	8,21	13141,2
39 /ecom/connection	10	25	15	57	11,22	0,00%	0,6	8,23	13177
43 /ecom/cartManagement	10	11	8	16	2,59	0,00%	0,6	8,26	13188
44 /ecom/paymentManagement	10	23	2	196	57,64	0,00%	0,6	3,73	5944
47 /ecom/paymentManagement	10	21	2	186	54,81	0,00%	0,6	3,89	6122
50 /ecom/createOrder	10	1548	1217	1916	234,78	0,00%	0,6	7,41	12634
TOTAL	110	193	2	3479	546,83	0,00%	5,1	58,21	11599,6

On observe notamment les délais supérieures lors de la consultation du catalogue (/ecom/catalog) et de la vue du produit (/ecom/productView). Ceci est normal car dû au requêtage

de la base de données permettant la récupération des divers produits disponibles et de leurs informations. Ces informations sont alors stockées dans des attributs de session permettant l'affichage côté client. On remarque également qu'aucune erreur n'a été levée durant ce test.

Scénario 2

Ce scénario consiste à vérifier la charge lors du chargement d'une page de l'application par plusieurs utilisateurs. Pour ce faire, nous paramétrons JMeter afin d'effectuer une requête sur la page du catalogue. En effet, il s'agit certainement de la page qui nécessite le plus de données à récupérer. Pour ce test, nous disposons en base de 20 articles et nous simulons 50 utilisateurs en parallèle.

Libellé	# Echantillons	Moyenne	Médiane	90% centile	95% centile	99% centile	Min	Max	% Erreur	Débit	Ko/sec
Requête HTTP	101	4772	4813	7503	7840	8116	803	8133	0,00%	4	43,9
TOTAL	101	4772	4813	7503	7840	8116	803	8133	0,00%	4	43,9

On s'aperçoit que le temps de réponse est déjà élevé (les explications de ce phénomène sont dans le scénario 3) et qu'aucune erreur n'est survenu durant ce test. Des améliorations peuvent être faite quant au nombre d'article chargé dès l'affichage du catalogue. On pourrait diminuer le nombre d'articles affiché par page à 10 et l'utilisateur chargerait la suite des articles seulement si ça l'intéresse.

Scénario 2 bis

Même requête mais cette fois-ci avec 500 utilisateurs :

Libellé	# Echantillons	Moyenne	Médiane	90% centile	95% centile	99% centile	Min	Max	% Erreur	Débit	Ko/s
Requête HTTP	601	14581	13307	28029	29958	30944	452	31253	1,66%	3,1	33,9
TOTAL	601	14581	13307	28029	29958	30944	452	31253	1,66%	3,1	33,9

On peut alors voir que le nombre d'utilisateur a un impact significatif sur le temps de réponse. Toutefois ceci n'est pas réellement inquiétant. En effet, nous avons configuré un pool de connexion (avec BoneCP) permettant de gérer un nombre de connexion parallèle relativement limité pour notre projet. Ici nous avons poussé le test suffisamment loin pour se rendre compte des limites de notre configuration de BoneCP. On en conclut alors qu'il s'agit d'un point sur lequel il faudrait certainement retravailler afin d'améliorer les performances. Toutefois les taux d'erreurs sont relativement bas.

Scénario 3

Dans la continuité des tests précédents, le test qui va suivre tend à montrer le comportement du site lors de l'utilisation de la fonction de recherche. Le scénario est le suivant : l'utilisateur se rend sur la page de catalogue, effectue d'abord une recherche qui n'aboutit à rien, puis réitère le processus et obtient une page de résultats.

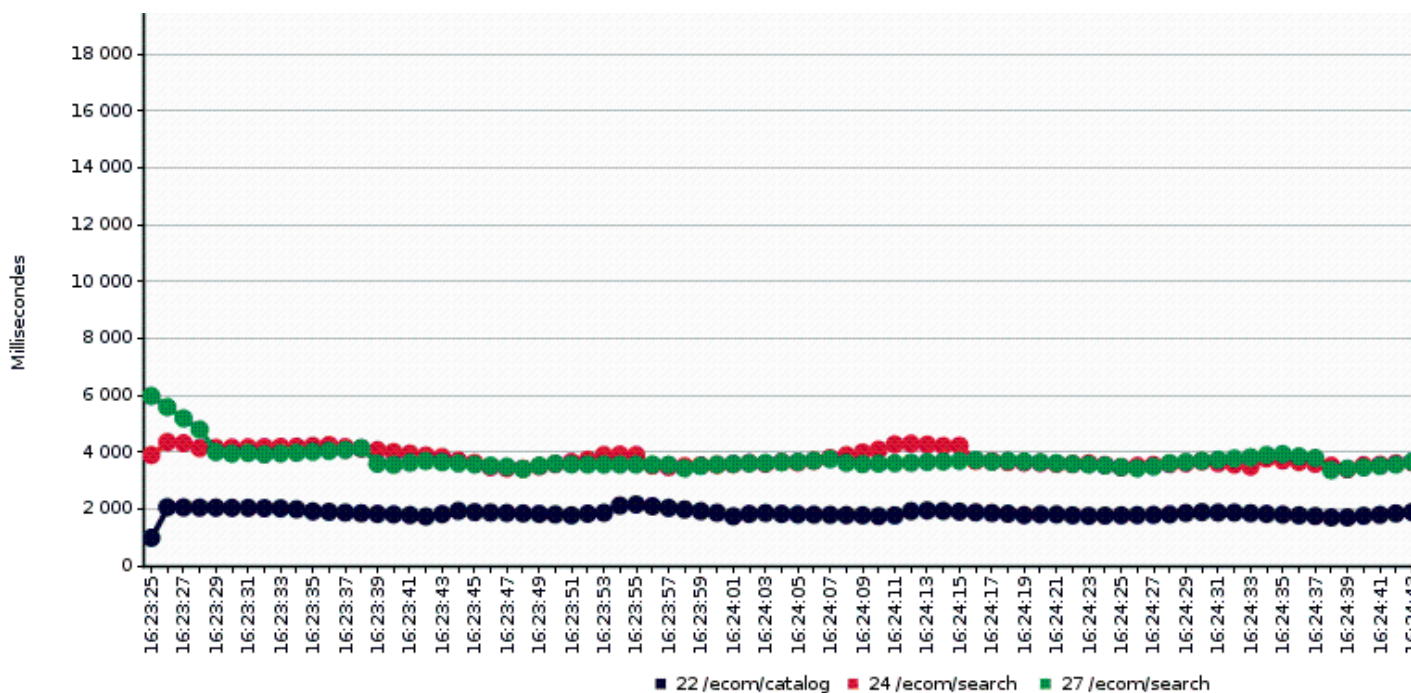


Figure 1 - Evolution du temps de réponses avec 50 utilisateur simultanés

Ce test de charge a été effectué en simulant 50 utilisateurs simultanés, qui réitèrent le processus 10 fois. Le graphe ci-dessus ne montre cependant qu'environ 20% du test. En effet, les tests étant réalisés en local, la machine de test ne pouvait supporter plus. On a cependant suffisamment d'informations pour voir que l'application se comporte bien. Le fait d'avoir ou non des résultats de recherche n'influe pas sur le temps de réponse.

Le temps de réponse est certainement plus important que sur un serveur dédié. De plus, la taille du pool de connexion (BoneCP) à la base de données ralentit les performances pour ce nombre d'utilisateur effectuant des requêtes simultanées. Dans notre cas, le nombre maximum de connexion par partition est de 20 ce qui explique les temps de réponses aussi élevés. Pour vérifier ceci nous procédons au même test que précédemment mais cette fois-ci avec 20 utilisateurs simultanés.

Voici le résultat :

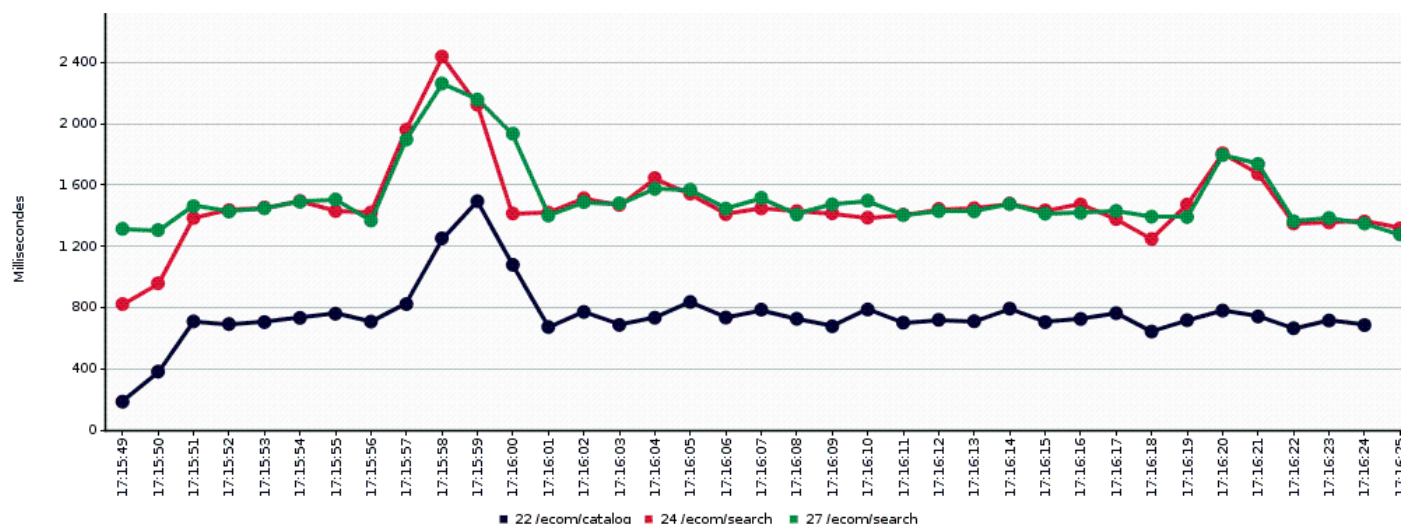


Figure 2 - Evolution du temps de réponses avec 20 utilisateurs simultanés

On s'aperçoit alors que la tendance moyenne du temps de réponse à largement diminué (quasiment divisée par 4), ce qui confirme notre hypothèse.

Scénario 4

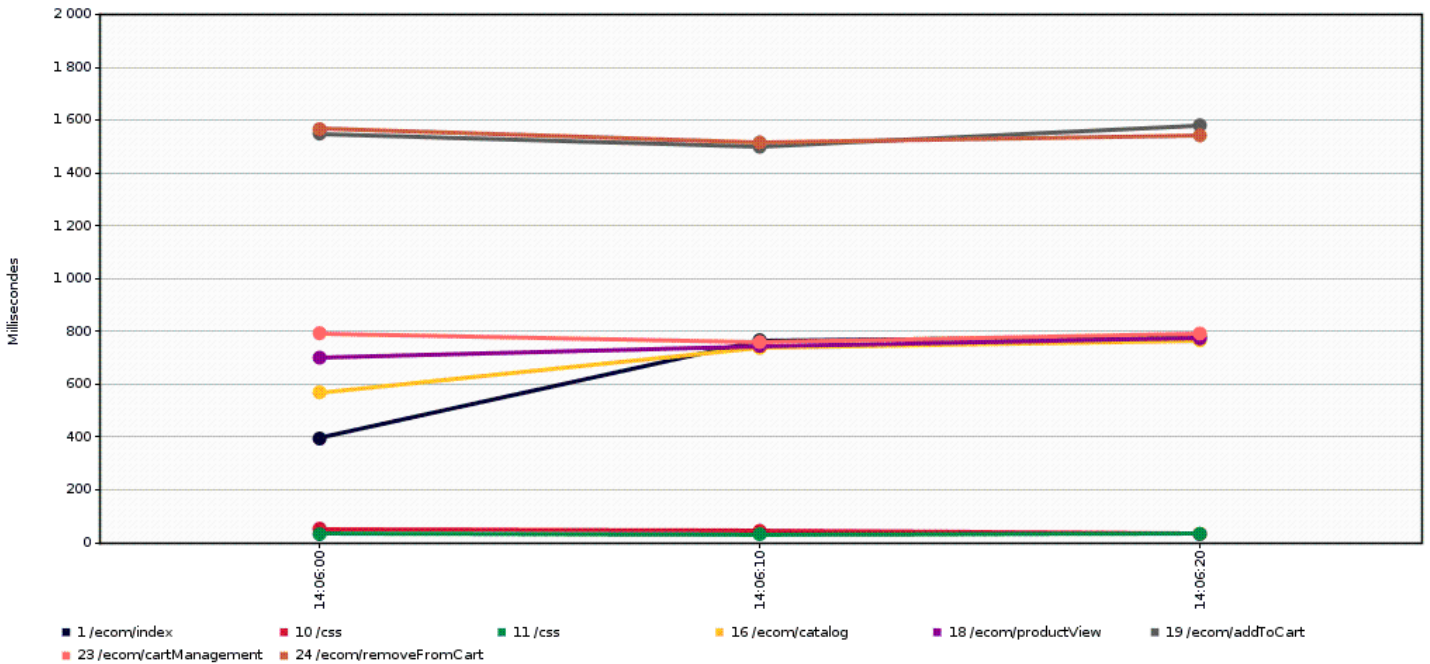
Voici le scénario pour ce test :

Nous répétons 1 fois ce test qui comprend 20 utilisateurs. Chaque utilisateur arrive sur l'index, consulte le catalogue, ajoute un produit au panier, consulte son panier et supprime l'article.

Libellé	# Echan- tillons	Moyenne	Médiane	90% cen- tile	95% cen- tile	99% cen- tile	Min	Max	% Erreur
1 /ecom/index	20	187	176	232	252	338	124	338	0,00%
10 /css	20	614	554	960	1049	1400	178	1400	0,00%
11 /css	20	456	486	687	688	690	230	690	0,00%
16 /ecom/catalog	20	409	360	711	711	732	152	732	0,00%
18 /ecom/productView	20	507	529	717	720	738	195	738	0,00%
19 /ecom/addToCart	20	1198	1354	1578	1606	1702	279	1702	0,00%
23 /ecom/cartManagement	20	610	490	915	962	967	183	967	0,00%
24 /ecom/remove-FromCart	20	991	982	1454	1457	1553	456	1553	0,00%
TOTAL	160	621	502	1314	1457	1578	124	1702	0,00%

On peut voir que pour une vingtaine d'utilisateurs simultanés l'application se comporte bien avec un taux d'erreur nul.

Graphique évolution temps de réponses



Voici le graphe représentant l'évolution du temps de réponse pour chaque page de ce test en fonction du temps (réalisé durant un second test). On peut donc voir que malgré l'utilisation du site par une vingtaine de client, les temps de réponses restent stables.

Scénario 5

Pour ce test de charge le scénario est le suivant : un utilisateur arrive sur la page d'index, se connecte, et consulte son historique. L'opération est répétée 10 fois. A noter qu'au moment du test, l'historique de l'utilisateur test contient 21 commandes contenant chacune un panier d'un 1 produit (les quantités sont variables).

Libellé	# Echantillons	Moyenne	Médiane	90% cen- tile	95% cen- tile	99% cen- tile	Min	Max	% Erreur
1 /ecom/index	10	137	128	165	165	201	112	201	0,00%
5 /css	20	75	31	133	142	603	27	603	0,00%
16 /ecom/connection	10	242	237	255	255	269	227	269	0,00%
19 /ecom/orderHis- tory	10	242	239	249	249	272	231	272	0,00%
TOTAL	50	154	133	249	255	603	27	603	0,00%

On remarque que la connexion prend un peu de temps, en effet il faut procéder à la vérification du mot de passe utilisateur. Le second point qui prend un peu de temps est la récupération de l'historique utilisateur car il faut récupérer en base de données le contenu du panier de ses dernières commandes.

Scénario 6

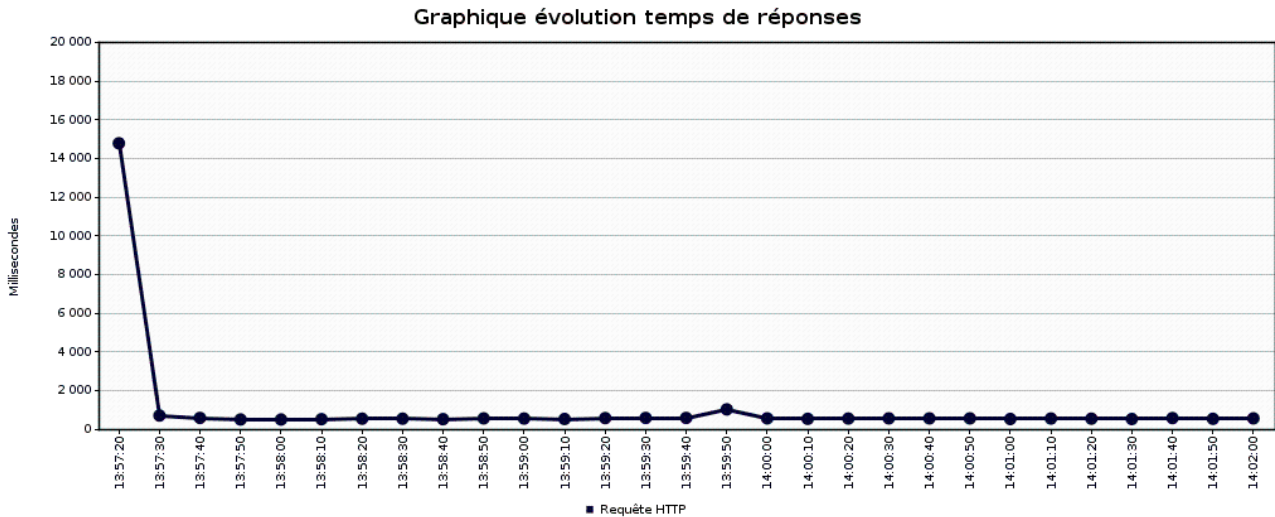
Ce test consiste à tester l'efficacité de la gestion du panier. Pour ce faire le scénario est le suivant : l'utilisateur est non connecté, il ajoute un produit à son panier, consulte son panier, se connecte, ajoute un nouveau produit à son panier, consulte son panier, se déconnecte, consulte à nouveau son panier.

Libellé	# Echantillons	Moyenne	Médiane	90% centile	95% centile	99% centile	Min	Max	% Erreur
28 /ecom/productView	50	221	189	252	641	641	129	641	0,00%
29 /ecom/addToCart	50	361	364	394	399	892	261	892	0,00%
33 /ecom/connection	50	355	352	403	406	436	257	436	0,00%
37 /ecom/cartManagement	50	170	179	202	209	225	115	225	0,00%
38 /ecom/catalog	50	170	177	204	207	225	122	225	0,00%
41 /ecom/productView	50	171	178	201	204	214	117	214	0,00%
42 /ecom/addToCart	50	337	329	398	405	474	173	474	2,00%
46 /ecom/cartManagement	50	163	163	196	201	206	116	206	0,00%
47 /ecom/disconnection	50	174	170	196	199	632	112	632	100,00%
51 /ecom/cartManagement	50	167	167	197	202	225	116	225	0,00%
52 /ecom/connection	50	358	336	398	402	896	260	896	0,00%
56 /ecom/cartManagement	50	187	189	213	251	689	109	689	0,00%
TOTAL	600	236	195	384	395	641	109	896	8,50%

Avec ce test nous nous sommes aperçus d'un problème lors de la déconnexion. On a étudié l'origine du problème et il semblerait qu'il soit lié au chargement d'une librairie Bootstrap car la déconnexion s'effectue correctement.

Scénario annexe (application déployée sur le cloud)

Il s'agit simplement d'étudier le chargement de la page de catalogue par un utilisateur et ce 500 fois de suite.



On voit que le temps de réponse est très correct (moyenne à 580 ms d'après le rapport agrégé). On observe un pic de chargement pour les premières connexions dont l'origine est inconnu (lenteur de la machine virtuelle Azure ?).