Rapport de projet LexInnova : passage à l'échelle

Mathieu Mangeot

Interlots:LexInnova/Passage à l'échelle

Auteur : Mathieu Mangeot

Date : 6 avril 2016

Cette page regroupe des tests pour évaluer le travail et l'architecture nécessaires au passage à l'échelle du prototype LexInnova codé en PHP (et actuellement déployé à cette adresse : http://totoro.imag.fr/Lexinnova/) pour obtenir un module intégré dans la plate-forme Claroline Connect fonctionnel en situation réelle d'apprentissage d'une langue. Nous remercions en premier lieu Donovan pour son aide précieuse concernant les outils d'évaluation et les possibilités techniques pour le passage à l'échelle (voir Interlots:LexInnova/C.R.Informels#CR_r.C3.A9union_Mathieu_M_et_Donovan) ainsi que Mathieu Loiseau pour sa relecture attentive.

Nombre d'utilisateurs

Utilisateurs enregistrés sur la plate-forme (espace de stockage)

Pour conduire nos évaluations, nous avons pris le chiffre de 10 000 utilisateurs maximum enregistrés sur une instance de l'ENPA. La plate-forme Jibiki nécessite pour chaque utilisateur et chaque langue de créer 3 tables. Nous faisons l'hypothèse que les utilisateurs apprendront en moyenne deux langues sur l'ENPA en plus de leur langue maternelle. Cela nécessite au total 10 000 × 3 × (2+1) = 90 000 tables dans une base de données.

La plate-forme Jibiki utilise la base de données Postgres pour sa couche données. Nous avons trouvé sur un forum les chiffres suivants d'utilisation de Postgres pour une grande base de données :

```
Taille de la base de données : 3500 Go
Taille du catalogue pg_catalog : 2Go
select count(*) from pg_tables;
120'884
select count(*) from pg_indexes;
219'082
select count(*) from pg_attribute;
2'779'199
Serveur : 48 Go de RAM & 12 cœurs
```

La taille d'une base de données de Jibiki comprenant un volume avec 154 000 entrées est de 1 500 Mo (1,5 Go) ce qui fait approximativement 10 Mo par entrée (même si la taille de la base n'est pas tout à fait proportionnelle au nombre d'enregistrements). Prenons l'hypothèse de 20 langues pour les dictionnaires de référence : 1,5 Go × 20 = 30 Go.

Prenons l'hypothèse d'une taille de 1 000 entrées pour chacun des 10 000 utilisateurs soit 10 Mo × 10 000 = 100 Go. Nous estimons la taille totale d'une telle base à 100 Go + 30 Go = 130 Go.

Les chiffres cités plus haut correspondent à une base de données avec 1/4 de tables en plus et 20 fois plus de données que pour notre hypothèse.

Le serveur actuel sur lequel est installé l'instance du prototype LexInnova est équipé de 32 Go de RAM et 4 cœurs, soit 1/3 de RAM et 2/3 de cœurs en moins. Nous faisons l'hypothèse que ce serveur aura la capacité suffisante pour gérer une telle base de données.

Utilisateurs simultanés (traitement des requêtes)

Pour conduire nos évaluations, nous avons pris le chiffre de 1000 utilisateurs connectés en simultané sur la même instance de l'ENPA. Une analyse de l'application SELF a montré que pour 25 personnes, on obtient 5 requêtes/seconde, soit 5 fois moins de requêtes/secondes que d'utilisateurs. Le serveur doit être capable de répondre à 200 requêtes/seconde.

Les tests de connexion à la plate-forme jibiki via son API REST ont été réalisés avec l'outil wrk depuis une machine sur le même sous-réseau. wrk lance 200 connexions simultanées sur une durée de 10 secondes.

Test sans cache http de consultation d'un mot dans un lexique :

```
wrk -c 200 -d 10 http://totoro.imag.fr/lexinnova/api/Lexinnova/esp/cdm-headword/ser/
Running 10s test @ http://totoro.imag.fr/lexinnova/api/Lexinnova/esp/cdm-headword/ser/
 2 threads and 200 connections
 Thread Stats   Avg      Stdev     Max   +/- Stdev
   Latency   134.16ms  143.75ms   1.87s    83.29%
   Req/Sec   270.79    108.13   500.00     66.48%
 4848 requests in 10.07s, 4.96MB read
 Socket errors: connect 0, read 0, write 0, timeout 154
Requests/sec:    481.39
Transfer/sec:    503.99KB
```

Test avec cache http de consultation d'un mot dans un lexique :

```
wrk -c 200 -d 10 http://totoro.imag.fr/lexinnova/api/Lexinnova/esp/cdm-headword/ser/
Running 10s test @ http://totoro.imag.fr/lexinnova/api/Lexinnova/esp/cdm-headword/ser/
 2 threads and 200 connections
 Thread Stats   Avg      Stdev     Max   +/- Stdev
   Latency   179.78ms  310.38ms   1.81s    83.09%
   Req/Sec     5.36k     1.54k   10.74k    84.26%
 105037 requests in 10.00s, 108.11MB read
 Socket errors: connect 0, read 0, write 0, timeout 60
Requests/sec:  10499.68
Transfer/sec:     10.81MB
```

Résultat : même sans cache, le serveur est capable de répondre à 4800 requêtes en 10 secondes soit 480 requêtes/seconde, ce qui est au dessus de nos besoins.

Quantité de Données

Les tests effectués plus haut l'ont été sur des lexiques de 10 entrées. Les lexiques du prototype actuel LexInnova étant limités en nombre d'entrées, nous avons mené nos tests sur un volume contenant 154 000 entrées.

Test avec nomenclature de 154 000 entrées sans cache http :

```
wrk -c 200 -d 10 http://jibiki.fr/jibiki/api/Cesselin/jpn/cdm-writing/tesuto/?strategy=EQUAL
Running 10s test @ http://jibiki.fr/jibiki/api/Cesselin/jpn/cdm-writing/tesuto/?strategy=EQUAL
 2 threads and 200 connections
 Thread Stats   Avg      Stdev     Max   +/- Stdev
   Latency   109.32ms  131.17ms   1.99s    84.06%
   Req/Sec   252.31     76.85   450.00     72.73%
 4702 requests in 10.01s, 3.36MB read
```

```
 Socket errors: connect 0, read 0, write 0, timeout 155
Requests/sec:    469.63
Transfer/sec:    343.54KB
```

Test avec nomenclature de 154 000 entrées et cache http :

```
wrk -c 200 -d 10 http://jibiki.fr/jibiki/api/Cesselin/jpn/cdm-writing/tesuto/?strategy=EQUAL
Running 10s test @ http://jibiki.fr/jibiki/api/Cesselin/jpn/cdm-writing/tesuto/?strategy=EQUAL
 2 threads and 200 connections
 Thread Stats   Avg      Stdev     Max   +/- Stdev
   Latency    45.86ms   88.86ms   1.91s    85.06%
   Req/Sec    20.59k     6.23k   35.49k    70.50%
 411599 requests in 10.07s, 296.83MB read
 Socket errors: connect 0, read 0, write 0, timeout 56
Requests/sec:  40890.80
Transfer/sec:     29.49MB
```

Résultat : même sans cache, nous obtenons une moyenne de 470 requêtes/seconde, ce qui est au dessus de nos besoins. *De plus nous pouvons remarquer qu'en multipliant le nombre d'entrées par 15 000, le temps de traitement n'augmente que de 2%, ce qui **suggère** que les temps de traitement des requêtes sont **peu** affectés par la quantité de données.*

Lecture active

Le module de lecture active est programmé de manière spécifique. Il fait actuellement appel à des analyseurs morphologiques installés sur le serveur totoro ainsi qu'à l'API REST de Jibiki. Nous avons testé sa réactivité dans un premier temps sans cache http en augmentant le temps de réponse à 20 secondes et en diminuant le nombre de connexions à 100.

Test sans cache http (100 connexions et 20 secondes de test) :

```
wrk -c 100 -d 20 "http://totoro.imag.fr/Lexinnova/Pages/lectureActive.php?text=Es+el+centro+de+Madrid.&langtext=esp&submit=Analyser"
Running 20s test @ http://totoro.imag.fr/Lexinnova/Pages/lectureActive.php?text=Es+el+centro+de+Madrid.&langtext=esp&submit=Analyser
 2 threads and 100 connections
 Thread Stats   Avg      Stdev     Max   +/- Stdev
   Latency     0.00us    0.00us   0.00us     nan%
   Req/Sec    87.22    150.23   380.00     77.78%
 100 requests in 20.09s, 1.64MB read
 Socket errors: connect 0, read 0, write 0, timeout 100
Requests/sec:      4.98
Transfer/sec:     83.57KB
```

Test avec cache http (200 connexions et 10 secondes de test) :

```
wrk -c 200 -d 10 "http://totoro.imag.fr/Lexinnova/Pages/lectureActive.php?text=Es+el+centro+de+Madrid.&langtext=esp&submit=Analyser"
Running 10s test @ http://totoro.imag.fr/Lexinnova/Pages/lectureActive.php?text=Es+el+centro+de+Madrid.&langtext=esp&submit=Analyser
 2 threads and 200 connections
 Thread Stats   Avg      Stdev     Max   +/- Stdev
   Latency   225.29ms  266.47ms   1.98s    87.29%
   Req/Sec   333.29     48.09   454.00     75.00%
 6640 requests in 10.01s, 109.78MB read
 Socket errors: connect 0, read 0, write 0, timeout 50
```

```
Requests/sec:    663.60

Transfer/sec:     10.97MB
```

Résultats : L'utilisation d'analyseurs morphologiques est très gourmande en temps de calcul (CPU). Sans cache, il est possible de répondre au maximum à 25 utilisateurs simultanés. Avec cache, nous obtenons plus de 660 requêtes/seconde, ce qui est au dessus de nos besoins.

Remarques :

- Dans une situation d'apprentissage par classes, nous pouvons espérer que le cache jouera un rôle non négligeable. En effet, nous pouvons faire l'hypothèse que les étudiants vont chercher à lire des textes similaires.
- Au niveau de l'infrastructure, si le cache ne fait pas son effet, il pourrait être pertinent de ne pas héberger ce service sur le même serveur que le reste de LexInnova afin que les ralentissements dûs à une sur-utilisation de la lecture active n'affectent que la lecture active et non le reste des usages de LexInnova.

Macrostructure

La macrostructure des ressources actuellement modélisée dans le prototype LexInnova est très simplifiée par rapport aux premières spécifications. En effet, celle-ci se compose d'un simple volume par langue sans lien entre eux.

Nous avons donc conduit des tests sur une macrostructure Pivax (pivot avec 2 niveaux de lien et 9 langues). Le serveur utilisé est différent du premier. Il est équipée de 8 Go de RAM (contre 32 pour les premiers tests) et de 4 cœurs.

```
wrk -c 200 -d 10 "http://takara.imag.fr:8998/pivax/Home.
po?FACET.0=cdm-headword&OPERATOR.0=0&FACETVALUE.
0=test&SOURCE.0=eng&showType=ShowClassique&
TARGETS=*ALL*&VOLUMES=CommonUNLDict_eng&
VOLUMES=iToldU_eng&VOLUMES=iToldU_eng_axeme&XSL=Default&
LIMIT=5&action=lookup"
Running 10s test @ http://takara.imag.fr:8998/pivax/Home.
po?FACET.0=cdm-headword&OPERATOR.0=0&FACETVALUE.
0=test&SOURCE.0=eng&showType=ShowClassique&
TARGETS=*ALL*&VOLUMES=CommonUNLDict_eng&
VOLUMES=iToldU_eng&VOLUMES=iToldU_eng_axeme&XSL=Default&
LIMIT=5&action=lookup
 2 threads and 200 connections
 Thread Stats   Avg      Stdev     Max   +/- Stdev
   Latency   164.23ms   94.85ms 631.87ms   66.73%
   Req/Sec   556.39    191.44     0.98k    64.00%
 11086 requests in 10.01s, 47.18MB read
Requests/sec:   1107.77
Transfer/sec:      4.71MB
```

Résultat : le serveur répond à 1100 requêtes/seconde, ce qui est largement au dessus de nos besoins.

Microstructure

L'ajout de nouvelles informations (comme par exemple la prononciation) dans une microstructure existante n'influant quasiment pas sur la taille de la base finale, nous n'avons pas conduit de test spécifique sur cette partie.

Infrastructure

L'infrastructure actuelle de la plate-forme jibiki est regroupée sur un seul serveur. Les tests effectués précédemment montrent que celui-ci peut très bien supporter la charge d'un passage à l'échelle conséquent. Néanmoins, nous pouvons également imaginer de distribuer l'infrastructure de la plate-forme sur plusieurs serveurs afin de mieux répartir la charge.

- une première étape serait d'installer la base de données sur un serveur dédié
- une deuxième étape consisterait à installer la partie applicative en java de la plate-forme sur plusieurs serveurs (au moins deux en cas de panne d'un serveur) avec un serveur frontal qui gère la répartition de charge (load balancing).
- une troisième étape consisterait à créer plusieurs bases de données pour le projet. Les dictionnaires de référence pourraient être séparés des autres lexiques car ils sont fréquemment consultés mais rarement mis à jour.

Intégration du prototype LexInnova PHP dans Claroline Connect

Le prototype actuel LexInnova a été développé intentionnellement en PHP. Il intègre la charge graphique de l'ENPA mais n'est pas encore développé comme un module intégrable dans la plate-forme Claroline Connect.

Le prototype se compose de :

- 6 pages frontales :
  - consultation du dictionnaire de référence
  - import d'un article dans un lexique personnel
  - consultation du lexique personnel
  - lecture active
  - liste des lexiques (administration)
  - liste des utilisateurs (administration)
- 3 classes :
  - dictionnaire
  - volume
  - utilisateur
- 1 fichier de fonctionnalités d'interaction avec l'API REST de Jibiki :
  - get_user
  - post_user
  - delete_user
  - get_userlist
  - get_dictionnaire
  - post_dictionnaire
  - put_dictionnaire
  - delete_dictionnaire
  - get_volume
  - post_volume
  - put_volume
  - delete_volume

- get_article
- post_article
- put_article
- delete_article
- put_article_part

Le temps de développement nécessaire à l'intégration du prototype dans Claroline Connect a été évalué par Donovan à environ 3 personnes/semaine. (3 personnes durant une semaine ou une personne pendant 3 semaines ou...) Cette intégration peut se faire soit :

- par l'auteur principal du prototype (Mathieu Mangeot), ce qui nécessitera de se familiariser avec les méthodes de développement de Claroline Connect et notamment l'environnement de développement Symfony
- par l'équipe de développement du projet Innovalangues
- en externalisant à un prestataire du consortium Claroline connect

Intégration du module de lecture active

Le module de lecture active nécessite un développement spécifique à l'extérieur de Claroline Connect concernant l'appel aux analyseurs morphologiques. Pour l'instant, ceux-ci sont programmés directement en PHP à l'intérieur du prototype. Il sera nécessaire d'utiliser ensuite une API du type de LEXTOH, un outil développé dans l'équipe GETALP.

Conclusion

En conclusion, moyennant 3 semaines de travail, le prototype actuel LexInnova peut être intégré à la plate-forme Claroline Connect avec les fonctionnalités actuelles mais sans la lecture active : 1 dictionnaire de référence pour chaque langue et la possibilité pour chaque utilisateur de créer un lexique personnel pour chacune des langues qu'il étudie. Le prototype peut passer à l'échelle sans problèmes en utilisant l'instance de Jibiki installée sur le serveur totoro hébergé au Laboratoire d'Informatique de Grenoble.

Pour le passage en production, il restera à constituer pour chaque langue un dictionnaire de référence de départ et à négocier avec le LIG les conditions d'utilisation du serveur totoro.

Le temps nécessaire de développement pour reprogrammer le module de lecture active est estimé également à 3 personnes/semaine.

Il est également possible d'ajouter de nouvelles fonctionnalités au prototype comme la gestion de lexiques de groupe sans changer l'installation actuelle de la plate-forme Jibiki. Cela nécessitera des développements supplémentaires au niveau du prototype en PHP. Ces développements n'ont pas encore été chiffrés car ils n'ont pas encore fait l'objet de discussions sur les priorités et surtout sur la main d'œuvre disponible pour ce travail.

Sources et contributeurs de l'article

Interlots:LexInnova/Passage à l'échelle *Source*: http://wiki.innovalangues.net/index.php?oldid=17870 *Contributeurs*: Eggerse, Loizbek, Mangeot

Licence