Pilot Systems

SeSQL : un moteur de recherche puissant et efficace

Gaël LE MIGNOT
Bruno DUPUIS
Pilot Systems

5 avril 2011

Plan

Introduction

Contexte original

- Quotidien Libération
- Libération actuellement en PHP, migration Django en cours
- Pour le front-office comme pour le back-office

Solution précédente

- Propriétaire, sous Windows (TextML)
- Problèmes de performances et de stabilité

Volumétrie

- 685 331 articles
- 165 551 pages
- plus de 100 000 contenus divers

Architecture

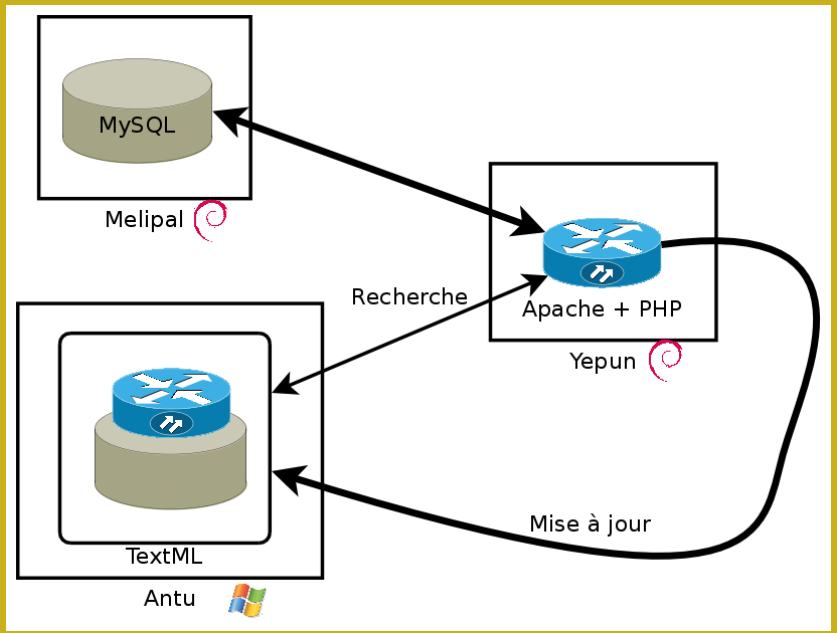
Architecture

Architecture initiale

Fonctionnement

- Solution propriétaire sous Windows
- Utilisée par des appels XML depuis le PHP

Schéma

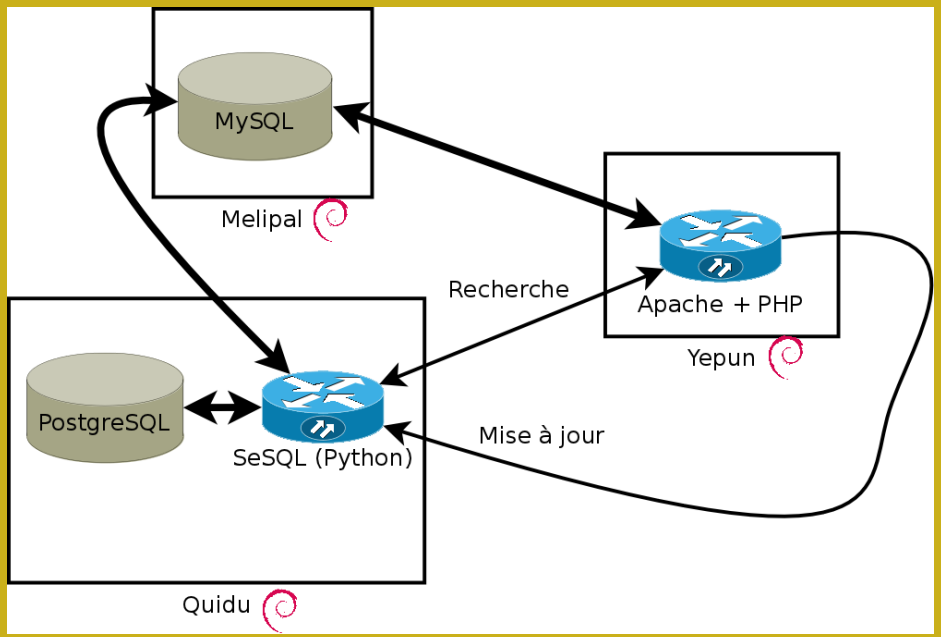


Architecture de la v1 de SeSQL

Fonctionnement

- Webservice en Python pur
- Compatible niveau API avec TextML

Schéma

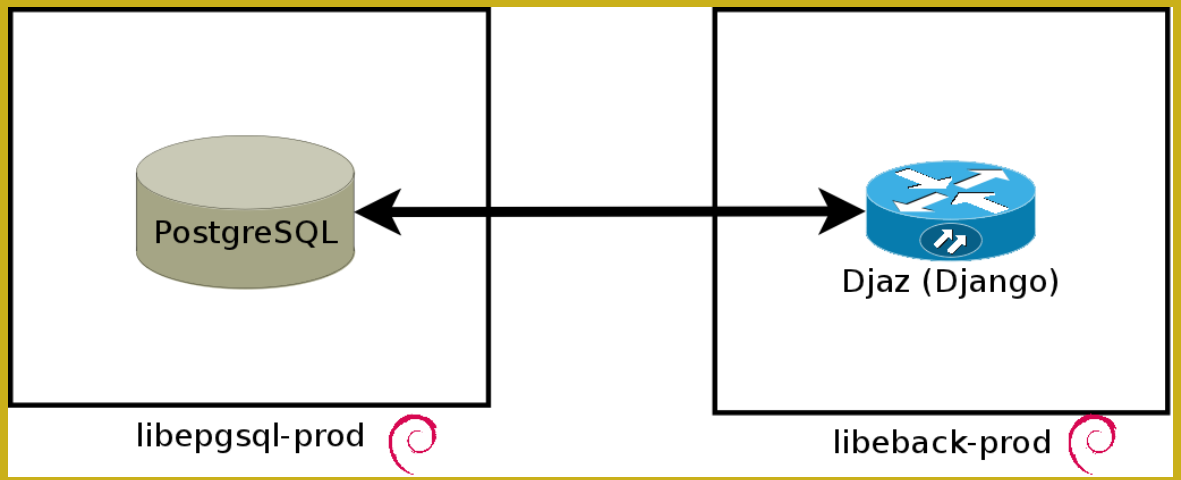


Architecture de la v2 de SeSQL

Fonctionnement

- Application Django
- Nécessite PostgreSQL
- Utilise Q et renvoie des modèles Django

Schéma



Fonctionnalités

Recherche multi-critères

Définition des index

- Définis dans un fichier spécifique, `sesql_config.py`
- Possibilité de définir plusieurs index (privés, publics, ...)
- Gestion des index composites, pouvant suivre les relations
- Gestion d'index fulltextes, multivalués, ...
- Gestion des dictionnaires (stopwords, recherche de racine, ...)

Utilisation de l'objet Q

- Support de critères comme `__containswords` ou `__containsexact`
- Support des connecteurs logiques (AND, OR, NOT)
- Renvoi des objets Django

Recherches courtes et longues

Recherches courtes

- Retournent un petit nombre d'éléments (une cinquantaine)
- Utilisent des heuristiques pour aller très vite pour renvoyer du contenu récent
- Pas de décompte des résultats, ni de pagination
- Idéal pour des portlets, requêtes de navigation, ...

Recherches longues

- Gèrent la pagination de manière stable (même si du contenu est inséré)
- Décomptent le nombre exact de résultat
- Supportent le tri par pertinence

Gestion des dépendances

Utilité

- L'indexation suit les relations
- Par exemple on indexe le nom de l'auteur de l'article
- Mais l'auteur change de nom
- ...il faut réindexer tous les articles !

Fonctionnement

- Une méthode à implémenter sur les modèles
- Ensuite, automatiquement, les objets à réindexer sont ajoutés dans une table
- Un deamon les reindexe en tâche de fond

À venir

Suggestions de recherches

- On stock les recherches effectuées
- On garde des informations comme la date et le nombre de résultats
- Quand une recherche est faite, on calcule un score à partir de tout ça
- Et on suggère des recherches proches.

Utilisation dans l'admin Django

- Fonctionnel, mais *quick and dirty*
- On écrit `sesql:index` dans les modèles de l'admin
- Sera bientôt intégré

Performances

Performances

Contexte du bench

Précautions d'usage

There are three kind of lies : lies, damn lies and benchmarks.

Requêtes

- Proviennent de la prod du back-office Libération
- Effectuées sur la v1 de SeSQL

Tests

- Requêtes séquentielles
- Requêtes concurrentes, dans des threads

Contexte du bench

	TextML	Sesql	Sesql	Sesql	Sesql	Sesql	Sesql
Nombre de recherches	19280	19280	19280	19280	19280	19280	19280
Recherches simultanées	n/a	1	2	4	1	2	4
Nombre d'insertions	n/a	0	0	0	19306	12157	15657
Insertions simultanées	n/a	0	0	0	1	1	3
Temps total	n/a	329,49	259,47	244,88	1594,88	1345,59	1519,24
Temps moyen	0,955	0,017	0,022	0,041	0,049	0,074	0,171
>10s	14	0	0	0	0	2	4
>5s	154	0	0	0	9	11	56
>2s	1002	0	0	0	70	104	310
>1s	7221	0	0	8	168	235	643
5 plus lentes requêtes							
1	14,46	0,61	0,73	1,51	6,26	17,4	17,27
2	14,2	0,54	0,71	1,41	6,21	11,81	13,37
3	13,75	0,48	0,64	1,38	6,18	9,87	11,47
4	13,61	0,46	0,63	1,18	5,81	8,17	10,28
5	13,12	0,44	0,62	1,12	5,77	7,27	8,38

Conclusion

Conclusion

Conclusion

Situation actuelle

- SeSQL disponible en GPL sur bitbucket
- Utilisé en production sur Libération (front et back)
- Utilisé sur d'autres projets Django

Évolutions futures

- Rendre le processus d'installation plus aisé
- Gestion des suggestions de recherche
- Intégration à `haystack` ?

En attendant

Remerciements

- aux communautés PostgreSQL, Python et Django
- à Libération de nous avoir fait confiance
- à Jérôme Petazzoni qui a contribué à la conception

La page de pub

- Pilot Systems, société de services en logiciels libres : `http://www.pilotsystems.net`
- Slides en licence CC-By-Sa 
- `http://contributions.pilotsystems.net/`

Des questions ?