

# Base de données Open Source : état de l'art et méthodologie de choix....



## Nous avons segmenté ces solutions en 3 catégories :

- Les bases de données relationnelles (MySQL, PostgreSQL, MaxDB, Interbase/Firebird...)
- Les bases de données natives XML (Apache Xindice, eXIST....)
- Les bases de données multidimensionnelles (Encore peu d'offres abouties en Open Source)
- Nous allons par la suite nous concentrer sur l'étude des SGBDR, de loin le marché le plus important et le plus mature.

# SGBDR open source: les questions à se poser pour faire un choix

- Pouvez-vous en profiter pour remplacer vos bases de données d'éditeurs par des alternatives gratuites ?
- Quel est la base de données la mieux adaptée à votre contexte ?
- Comment migrer et comment assurer le support technique ?
- Que peut-on attendre comme évolution et comme pérennité ?

## Les SGBD open source viennent d'horizons divers et n'évoluent pas de la même façon.

- Les bases de données issues en ligne directe du mouvement open source : MySQL et PostgreSQL
- Les bases de données qui viennent du monde des éditeurs : FireBird (ex-Interbase de Borland) et SAPdb (dérivé d'Adabas)
- Public/audience/popularité de chaque projet ?
- Le type de communauté derrière chaque projet (projet indépendant ou appuyé par un éditeur) ?
- Forces et faiblesses de chaque projet ?

# Les caractéristiques techniques

- Les différentes bases de données open source ne sont pas basées sur les mêmes choix techniques et sont donc plus ou moins adaptées à tel ou tel contexte (support des standards SQL (ANSI et 92), support des fonctions ODBC/JDBC, support des types de données, limites physiques (tailles des bases, nb de tables, etc.).)
- Le spectre de fonctionnalités, les plates-formes supportées, la connectivité et les outils de développement/administration sont les différenciateurs clés dans ce domaine (les middlewares d'intégration qui tiennent compte des bases de données open source, les pilotes ODBC/JDBC disponibles pour chaque projet, outils de développement et d'administration disponibles pour chaque projet, les outils de développement (éditeur de procédures stockées, triggers, etc.), les outils d'administration (création/gestion des objets de la base, gestion des permissions-utilisateurs, suivi du fonctionnement, etc.)

# Migration et support technique

- Choisir une base de données open source pour un nouveau projet c'est facile et intelligent mais quid d'une migration ?
- Quels sont les projets qui offrent les moyens de migrer depuis Oracle, DB2 ou SQL Server et comment ?
- De plus, migrer est une chose mais, après, il faut assurer la continuité de service ou MCO ... quelles sont les offres de support technique autour des bases de données étudiées ?

# Perspectives d'évolution et pérennité

- Quand et comment les bases de données les moins avancées techniquement vont-elles rattrapper leurs retards de fonctionnalités ? (les perspectives d'évolutions techniques des différents projets (roadmap))
- Quels sont les projets qui paraissent les plus solides, les plus pérennes ? (analyse des « chances de survie » des différents projets en fonction de leurs « sponsors » et de leur implantation dans le marché...)

# Le témoignage d'un industriel: F. Aime – CTO d'Arthémys Portail

- Arthémys Portail
  - Une solution d'E-Procurement
    - Workflows
    - Gestion des cycles de recrutement
  - Un outil d'aide à la décision
  - Une faculté de traçabilité

# Critères de choix

- Les raisons du choix MySQL 4
  - Des besoins de stockage de données simples
  - Une intelligence basée sur un moteur C++ maison communiquant en XML (XML-rpc/Soap)
    - Une couche d'abstraction BD simple
      - (SELECT, INSERT, UPDATE, DELETE)
    - Matching de données
    - Débruitage de séries numériques (scoring)

# Problèmes rencontrés et limitations de MySQL

- Peu de fonctions évoluées
  - Pas de procédures stockées
  - Pas de triggers
  - Peu d'outils de statistiques (au moment du développement)

# Bénéfices retirés de l'utilisation de MySQL

- Bénéfices majeurs :
  - Outil approprié au besoin d'Arthémis Portail
  - Performances très élevées
- Bénéfice mineur:
  - Coût réduit (dans le cas Arthémis)

# Conclusion

- Le choix serait d'autant plus à ré-itérer aujourd'hui, que MySQL arrive à maturité, sans pour autant (trop) perdre en performances
- Déléguer à MySQL un ensemble de tâches complexe devient possible et remets plus que jamais en cause l'utilisation d'outils propriétaires.