

Bases de données  
documentaires et distribuées,  
<http://b3d.bdpedia.fr>

Introduction du cours

# Le sujet, en bref

## Documents, collections massives, stockage et calcul distribués

Comprendre le lien entre la représentation d'entités sous forme de **documents** pour la gestion de **collections** à très grande échelle dans des environnements distribués.

Les grands thèmes :

- modélisation sous forme de **document**, structuré, semi-structuré, non structuré ;
- gestion de **très grandes collections**, et notamment recherche d'information ;
- **distribution du stockage** pour maintenir de très grands volumes de données ;
- **distribution des calculs**, notamment pour les méthodes analytiques.

Des mots-clés à comprendre (et à critiquer) : données structurées/non structurées, bases NoSQL, Cloud, BigData, moteurs de recherche, etc.

# Bases relationnelles

Les **bases relationnelles** = applications gérant des informations **structurées** et **régulières** : applications de "gestion", Web, mobiles.

- Une modélisation normalisée.
- Un langage (SQL) très bien défini, normalisé lui aussi.
- De très bonnes performances, obtenues **automatiquement**.
- Une gestion robuste de la concurrence d'accès.

**Attention à bien apprécier ce qu'on gagne / perd en passant au "NoSQL"** (on gagne marginalement, on perd beaucoup)

Révisions nécessaires? Mieux vaut s'en apercevoir **maintenant**. Révisions sur <http://www.bdpedia.fr>

# Problématique 1 : La notion de “document”

Document = unité d'information autonome ou quasi-autonome.

- Peu ou pas de référence à d'autres documents.
- Peu ou pas de structure ; ou une structure très flexible.
- Un contenu souvent à orientation multimédia.

**Exemples (1)** : documents textuels, types documents Web.

**Exemples (2)** : images, documents audios, vidéos ; pas de structure explicite, production de descripteurs synthétiques pour tenter de les indexer.

**Exemples (3)** : jeux en ligne : artifacts graphiques, actions utilisateur.

**Exemples (4)** : tous les fichiers de votre ordinateur...

Impose de repenser la notion de **modèle**, schéma et représentation des données.

## Problématique 2 : recherche d'information

Dans les collections documentaires, peu ou pas de structure fixe,

- SQL inadapté.
- Recherche « exacte » souvent insatisfaisante.

La recherche s'effectue souvent **par similarité**

- on fournit un document “requête”.
- le système recherche les documents **proches** du document-requête.

Par exemple, quand on recherche sur le Web :

- on fournit un ensemble de mots-clés : c'est le document requête ;
- le moteur de recherche trouve les documents les plus proches (on verra comment) ;

# Problématique 3 : stockage à très grande échelle

On atteint facilement des **volumes** extrêmement importants (Téraoctets+)

- les moteurs de recherche qui collectent des **documents** disponibles sur le Web.
- les applications utilisées à l'échelle du Web ; commerce électronique (Amazon) ; réseaux sociaux (Facebook).
- données gérées par les jeux en ligne.

Nouveaux systèmes, dits “NoSQL” sont conçus pour gérer de vastes collections de documents de manière scalable

- pour les accès temps réel ;
- pour les traitements analytiques (MapReduce).

# Plan du cours

- **Représentation de documents textuels** : les formats XML et JSON ; la modélisation de documents, les langages de manipulation.
- **Recherche d'information dans les bases documentaires** : moteurs de recherche, index, algorithmes.
- **Stockage distribué**. Systèmes distribués ; NoSQL.
- **Calcul distribué**. MapReduce, et au-delà

## Connaissances pratiques

- Des systèmes "NoSQL" ; (MongoDB, CouchDB, Cassandra)
- Des moteurs de recherche (Solr, ElasticSearch).
- Systèmes distribués : MongoDB, ElasticSearch, Cassandra.
- Traitement massif : Hadoop, Spark, Flink

# Prérequis

- **Compréhension des bases relationnelles**, soit au moins la conception d'un schéma, SQL, ce qu'est un index et des notions de base sur les transactions.  
⇒ si nécessaire, voir <http://www.bdpedia.fr>
- **Une aisance minimale dans un environnement de développement**. Editer un fichier, lancer une commande, ne pas paniquer devant un nouvel outil, savoir résoudre un problème avec un minimum de tenacité, etc.  
⇒ Vous devez reproduire les exemples donnés.

La connaissance de langage de programmation comme Java ou Python est un plus pour consolider vos nouvelles connaissances.



# Environnement matériel et logiciel

Le cours repose beaucoup sur la mise en pratique.

- au Cnam, tout vous est fourni ;
- Cnam à distance, ou en auditeur libre, il vous faut une machine dotée d'au moins 8 GO de mémoire (Linux, Mac/OS ou Windows).

**Tous les logiciels utilisés sont libres de droits.** Nous travaillons essentiellement avec

- Solr, et Elasticsearch pour l'indexation de documents.
- MongoDB, Cassandra, et plusieurs autres outils NoSQL.
- Flink et Spark.

L'installation de ces outils (et leur utilisation) peut se faire avec **Docker**, un gestionnaire de systèmes distribués **virtuels**.

# Organisation du cours

Le cours est découpé en **chapitres**, couvrant un sujet bien déterminé, et en **sessions**.

**L'unité de travail est la session**. Nous essayons de structurer les sessions pour qu'elles ne demandent environ 2 heures de travail personnel (bien sûr, cela dépend également de vous).

Pour assimiler une session vous pouvez combiner les ressources suivantes :

- **La lecture** du support en ligne, également disponible en PDF ou en ePub.
- **Le suivi du cours**, en vidéo ou en présentiel.
- **Le test des exemples de code** fournis dans chaque session.
- **La réalisation des exercices** proposés en fin de session.

**La réalisation des exercices est essentielle** pour vérifier que vous maîtrisez le contenu.

# Mener votre travail personnel

Vous devez maîtriser le contenu des sessions **dans l'ordre où elles sont proposées**.

**Commencez par lire le support**, jusqu'à ce que les principes vous paraissent clairs.

**Reproduisez les exemples de code**. Tous les exemples donnés sont testés et doivent donc fonctionner.

**Cherchez à résoudre les problèmes par vous-mêmes** au besoin : c'est le meilleur moyen de comprendre.

**Finissez par les exercices**. Faites les vous-mêmes, avant de regarder la solution.

**Répondez au Quiz** proposé en fin des principaux chapitres.

# Les projets

**Objectif** : un sujet qui vous intéresse, en liaison avec les techniques avancées de gestion de données, et vous préparez un rapport (app. 20 pages).

Exemple : vous étudiez le dernier système NoSQL à la mode (pas celui vu en cours!)

- Une partie **pratique** : choix d'un jeu de données, insertion, manipulations
- Une partie **étude** d'un aspect technique (interrogation, distribution, tolérance aux pannes, ...)

Vous êtes invité(e) à proposer quelque chose ! **Et faites ce qui vous intéresse avant tout.**

**Travail PERSONNEL, plagiat éliminatoire**

# Evaluation

Pour les auditeurs du Cnam, l'évaluation de l'UE combine :

- une note portant sur le **rapport écrit** du projet (coeff. 1/3).
- la note d'un examen de contrôle (coeff. 2/3).

Le projet est suivi et évalué par les enseignants du Cnam.

Voir support en ligne pour les annales, et les instructions détaillées.

**Bon travail !**