> qui alimente les applications de décision ou d'apprentissage. Qu'il s'agisse d'algorithmes simples ou complexes, déterministes ou prédictifs, opérant sur des données exactes ou bruitées, leur compréhension est indispensable pour entrevoir en quoi consiste l'exploitation des *big data*.

Occultés par les succès récents de l'apprentissage machine et autres programmes de jeux, ces algorithmes enfouis sont donnés pour acquis, avec un coût nul pour leur mise en œuvre et leur exécution. Nous verrons que ce n'est pas le cas, un travail permanent de recherche et d'ingénierie est nécessaire pour les adapter aux nouveaux contextes applicatifs, les étendre, les optimiser et en inventer de nouveaux.

Il y a autant de processus de gestion et de valorisation des données que de raisons de valoriser ces données. On y retrouve cependant de grandes activités génériques (*voir l'encadré page ci-contre*), sur chacune desquelles existent des verrous importants.

Pour réaliser ces grandes activités, il est important de spécifier les tâches constitutives en fonction des applications visées. La consultation intensive des données n'a pas les mêmes besoins qu'un traitement des flux de données en temps réel. Une application d'aide à la décision n'a pas les mêmes besoins qu'une application de prédiction du panier d'un consommateur ou d'aide au diagnostic d'une tumeur. La nature de l'application et des tâches algorithmiques qui en découlent doivent être mises en perspective avec le modèle de calcul le plus adapté.

DEUX MODÈLES DE CALCUL

La gestion des données n'est pas constituée d'algorithmes *ad hoc* réalisés au coup par coup, dans un langage donné, en fonction des applications. C'est une science qui a ses fondements théoriques, ses propres objets d'étude, ses propres abstractions et ses propres méthodes. Les technologies produites ont pour vocation de réduire l'effort nécessaire à la gestion des données, de contrôler l'accès à ces données et de faciliter l'évolution des systèmes sans réécriture de leur code. Pour simplifier, il y a deux modèles de calcul sur les grands volumes de données: celui dédié à la recherche d'information et celui adapté à l'analyse de données. Même si les deux approches partagent de nombreux concepts et techniques et peuvent se combiner, elles diffèrent d'un point de vue sémantique.

Dans le premier cas, le sens est porté par la requête de l'utilisateur; si la requête est pertinente, les données délivrées seront bien conformes aux critères de recherche spécifiés. Par exemple, «rechercher les références de produits vendus en quantité supérieure à 200 dans chacun des magasins de la région parisienne durant la période de Noël», est une requête complexe, mais qui définit clairement le calcul à faire et les critères de sélection de données. Ainsi spécifié, le système de gestion de données se chargera de générer le code nécessaire à l'exécution de cette requête. D'autres requêtes sont plus approximatives, leurs critères étant spécifiés de façon lâche. C'est le cas des recherches sur le Web. Ainsi, «rechercher les magasins parisiens pratiquant les prix les moins chers pour le chocolat et le champagne durant la période de Noël» est une requête floue. Elle impose un traitement approché du critère «moins cher» obligeant à présenter les résultats dans un certain ordre.

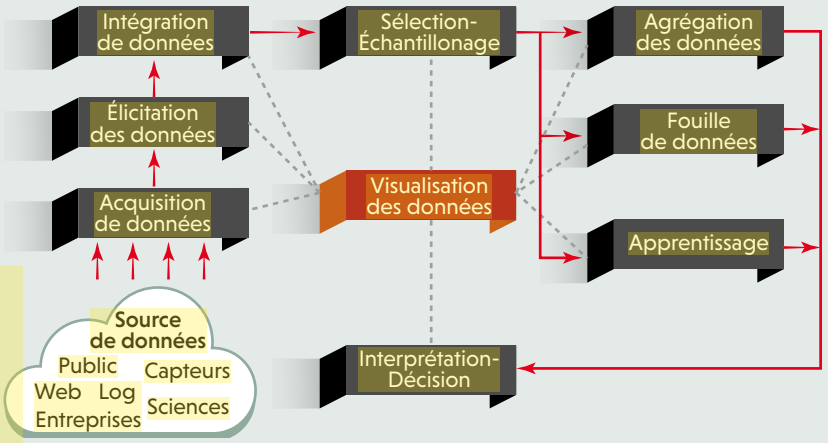
Dans le second cas, la question du sens est plus complexe, car elle interroge à la fois la méthode utilisée pour l'analyse des données et l'interprétation par l'utilisateur des résultats produits. Pour une requête d'analyse de l'évolution des cours de bourse des entreprises du CAC40 ou de prédiction des ventes de smartphones par région et par période de l'année, les résultats ne sont pas interprétables sans connaissance du métier et des modèles de prédiction sous-jacents.

Ce problème peut devenir plus aigu dans des algorithmes encore moins transparents. Par exemple, dans un système de reconnaissance fondé sur l'apprentissage, la confusion entre un chat et un caniche fait sourire, mais celle d'un honnête citoyen avec un criminel peut être dramatique. Les algorithmes sous-jacents à ces systèmes posent des défis à la société.

Il y a autant de processus de valorisation des données que de raisons de valoriser ces données

D'un point de vue conceptuel, un modèle de calcul est défini par les structures de données qu'il reconnaît et les opérations qu'il offre sur ces structures de données. Les structures de données peuvent être des tables (ou matrices), des arbres, des graphes ou des séquences finies ou infinies. Les opérations sont associées à chaque structure: les opérations de tri, projection, produit et union sur

LES MILLE ET UNE VIES DES DONNÉES



Le cycle de vie des données au terme duquel elles peuvent livrer des informations.

Le traitement des *big data* implique des activités que l'on retrouve quelle que soit l'application. Passons-les en revue. Le stockage pose la question du choix des données brutes à enregistrer, de leur acquisition et des données à résumer, ainsi que les méthodes d'indexation pour accélérer leur recherche ultérieure. La validation des données, c'est-à-dire leur nettoyage et leur annotation (on parle d'élicitation), nécessite souvent une interaction avec un expert, ce qui est hors de portée d'un humain si le volume de données est considérable. L'intégration de données est une phase très complexe car elle concerne des sources de données distribuées à grande échelle où l'hétérogénéité est très élevée. Le modèle d'intégration dépendra des objectifs assignés au système : la recherche d'information, la prise de décision, l'extraction de connaissances… C'est là que se situent les principaux verrous liés aux performances des systèmes et à la pertinence des données produites. La sélection et l'échantillonnage constituent des vues sur les données intégrées, c'est-à-dire une façon de réduire les données aux sous-ensembles nécessaires à un type de requête ou un type d'analyse. Un mauvais choix de ce sous-ensemble hypothèque la qualité de la tâche ultérieure. Quant à l'analyse elle-même, elle peut être réalisée par différentes méthodes : l'agrégation pour générer des indicateurs statistiques (somme, moyenne, tendance…), la fouille de données pour identifier des règles d'association ou des motifs fréquents, l'apprentissage automatique (profond ou par renforcement) pour définir un modèle de prédiction permettant de décider ultérieurement si un nouvel objet peut être classé de façon pertinente. Enfin, la visualisation des données est utile à toutes les étapes. Elle offre souvent une vue synthétique des grandes données et aide à leur compréhension.

les tables ; la recherche d'éléments ou le parcours de chemin sur un arbre ; l'appariement ou l'extraction de sous-graphes ; une copie instantanée d'une séquence. Ces structures et ces opérations ont été étudiées depuis des décennies pour formaliser leurs sémantiques, expliciter leurs propriétés et trouver les meilleurs programmes qui leur correspondent dans différents environnements.

Plusieurs facteurs interviennent dans leur mise en œuvre : la hiérarchie et la taille des mémoires (mémoire principale, mémoire cache, disque…), la localisation des données (un site centralisé, de multiples serveurs…), les ressources de calcul mobilisables (une ou plusieurs machines…) et les caractéristiques techniques de ces ressources (puissance de calcul, partage de mémoire…).

DES TABLEAUX MALLÉABLES

L'algèbre relationnelle a représenté un saut technologique et qualitatif dans la gestion des données. En quoi consiste-t-elle ? Rappelons d'abord qu'une table relationnelle est décrite par un ensemble de variables (ou attributs) désignant les colonnes et un ensemble de valeurs à ces variables organisées en lignes, chacune représentant un objet du réel. En d'autres termes, il s'agit de tableaux à deux dimensions, on parle aussi de tableau à double entrée. Or avec cinq opérateurs de base (*voir la figure page suivante*), on sait aujourd'hui transformer ces tableaux et en dériver d'autres.

C'est d'une simplicité confondante, et pourtant, la combinaison de ces opérations permet d'exprimer une large palette de requêtes, sans programmer une seule ligne de code, puisque le code correspondant à chacune des opérations est défini une fois pour toutes et pour toutes les applications. Cette façon de représenter le réel par un ensemble de tableaux indépendants et d'exprimer une requête sur ces tableaux au moyen d'une expression algébrique a notablement modifié notre rapport à la programmation.

Parmi ces opérateurs, certains sont plus sensibles que d'autres au volume des données. Ainsi la jointure de deux tables comportant des milliards de lignes pose de réels problèmes de calcul pour comparer ces milliards de lignes. De nombreux travaux de recherche proposent des algorithmes de jointure tenant compte ou non du tri des données, de la taille de la mémoire centrale, de l'indexation des données, de la distribution des données sur différents disques, du parallélisme des machines et du débit sur les réseaux.

Ce coût dépend aussi du choix initial de la représentation d'une table selon ses lignes ou ses colonnes. Par exemple, pour les requêtes >