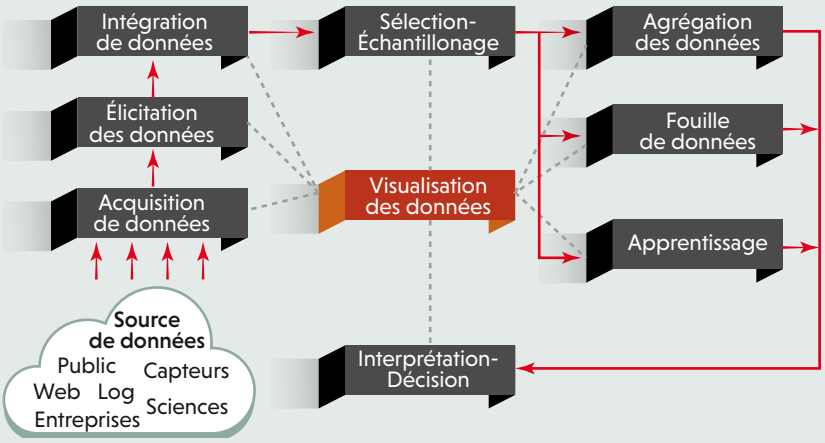
LES MILLE ET UNE VIES DES DONNÉES



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 >

PROJECTION

MAGASIN	PRODUIT	PRIX	DATE	SITE
Auchan	Iphone	700	12/9/2017	Paris Est
Auchan	Iphone	760	12/9/2017	Paris Est
Leclerc	IPad	480	27/9/2017	Essonne
Fnac	IPad	870	30/9/2017	La Défense
Carrefour	PC Dell	600	24/9/2017	Paris Ouest
Leclerc	Galaxy	400	28/9/2017	Essonne
Boulanger	Galaxy	310	4/10/2017	Plaisir
Carrefour	Galaxy	470	4/11/2017	St. Quentin
Leclerc	MacBookAir	970	7/10/2017	Yvelines
Auchan	MacBookAir	1100	7/10/2017	Yvelines
Darty	PC Dell	860	19/10/2017	Velizy
Darty	PC Dell	640	23/10/2017	Parly
Darty	iPad	480	27/10/2017	Parly

RESTRICTION

MAGASIN	PRODUIT	PRIX	DATE	SITE
Auchan	Iphone	700	12/9/2017	Paris Est
Auchan	Iphone	760	23/9/2017	Paris Est
Carrefour	PC Dell	600	24/9/2017	Paris Ouest
Leclerc	IPad	400	27/9/2017	Essonne
Leclerc	Galaxy	400	28/9/2017	Essonne
Fnac	IPad	870	30/9/2017	La Défense
Fnac	iMac	1220	1/10/2017	La Défense
Fnac	iMac	2100	3/10/2017	Montparnasse
Boulanger	Galaxy	310	4/10/2017	Plaisir
...	...	...	...	...
...	...	...	...	...

PRODUIT	MARQUE	BREVET
Iphone	Apple	AP2056XY
PC Dell	Dell	DL3467GH
Galaxy	Samsung	SM734DG
IPad	Apple	AP205VY

JOINTURE

MAGASIN	PRODUIT	PRIX	DATE	SITE	MARQUE	BREVET
Auchan	Iphone	700	12/9/2017	Paris Est	Apple	AP2056XY
Auchan	Iphone	760	23/9/2017	Paris Est	Apple	AP2056XY
Carrefour	PC Dell	600	24/9/2017	Paris Ouest	Dell	DL3467GH
Leclerc	IPad	400	27/9/2017	Essonne	Apple	AP205VY
Leclerc	Galaxy	400	28/9/2017	Essonne	Samsung	SM734DG
Fnac	IPad	870	30/9/2017	La Défense	Apple	AP205VY
Boulanger	Galaxy	310	4/10/2017	Plaisir	Samsung	SM734DG
...	...	...	...	...		
...	...	...	...	...		

L'algèbre relationnelle offre cinq opérateurs de base grâce auxquels on peut exprimer un grand nombre de requêtes sur des données. À partir d'une table relationnelle *(le tableau bleu)*, la projection réduit le nombre de colonnes, la restriction celui des lignes, l'union fusionne deux tables de même structure, l'intersection calcule les lignes communes à deux tables de même structure, la jointure apparie deux tables sur un ou plusieurs critères.

> d'exploration de données, les tables seront rangées par lignes alors que pour les requêtes d'analyse agrégeant les valeurs, la représentation par colonne est plus adaptée.

Dans le même élan de définition de ce modèle élégant, des extensions ont été apportées pour traiter des objets plus complexes que les tables relationnelles, notamment les tables imbriquées et les tables à plus de deux dimensions. L'algèbre relationnelle a également été étendue à d'autres domaines spécialisés grâce à l'introduction, par exemple, d'opérations géométriques ou d'autres sur des données temporelles. Le champ d'investigation du modèle relationnel est loin d'être couvert !

DES ARBRES ET DES GRAPHES

Avec le Web, les bases de données ont évolué pour intégrer des données XML dont la structure est formalisée par des arbres. Une algèbre est aussi associée à ces arbres pour effectuer des recherches d'éléments et pour transformer leurs structures. La sélection et la projection permettent d'extraire des sous-arbres vérifiant certaines conditions portant soit sur les données (les valeurs d'un élément), soit sur les métadonnées (les éléments descriptifs de ces valeurs). Il est possible également de restructurer un arbre, ou de l'aplatir, pour obtenir une organisation différente des données et les visualiser sous des angles variés.

Enfin, l'appariement permet de comparer deux structures, une opération très utile pour de nombreuses applications, comme la bioinformatique, l'analyse d'images, les bases de documents... On associe à ce type d'opération une mesure de similarité qui peut être le nombre de transformations à effectuer pour obtenir un arbre à partir d'un autre. Plus la distance est petite, plus les arbres sont similaires. Cette notion est cruciale dans le contexte des *big data*. En effet, le calcul de ces distances est un problème difficile, dont la complexité augmente fortement avec le volume des données et avec la diversité structurelle des graphes à comparer. Les données dont nous avons parlé sont durables, mais qu'en est-il des flux de données, qui disparaissent juste après leur observation ?

LES FLUX DE DONNÉES

La logique de gestion est ici bien différente. Cette fois, ce sont les requêtes qui sont persistantes. Elles sont définies une fois pour toutes et agissent comme des filtres, déterminant les données observées qu'on souhaite conserver. Le premier opérateur indispensable sur un flux définit la taille de la fenêtre d'observation et un pas de glissement pour suivre le flux. Ces paramètres peuvent être définis par des durées, par le nombre d'événements attendus ou la combinaison des deux. Les données observées peuvent être stockées temporairement dans des tables relationnelles et filtrées par des requêtes définies en amont.

Pour faciliter l'interprétation des flux, il est souvent nécessaire de les corréler avec des données de référence persistantes dans des catalogues, ce qui impose des opérations de jointures potentiellement coûteuses si ces catalogues sont gigantesques. Par exemple, l'observation d'une température de −110 °C dans un bâtiment est sans doute liée à la défaillance d'un capteur. En revanche, une forte température trahit sans

doute l'explosion d'une chaudière située dans ce bâtiment. Cette interprétation n'est possible qu'avec un catalogue des immeubles avec leurs caractéristiques. Avec des millions de capteurs, le passage à l'échelle du traitement des flux envoyés est un défi. Le déploiement des compteurs Linky est un exemple.

LES LIMITES DES ALGÈBRES

Malgré tous ses avantages, les modèles de calcul «à la relationnelle» ne couvrent pas les besoins de l'ensemble des applications, et notamment ceux de deux grands domaines. Dans le premier, on trouve des applications dont les données sont conceptuellement relationnelles, mais dont les performances et certaines contraintes (telle la représentation en ligne des tables) ne sont pas satisfaisantes. C'est le cas des applications nécessitant une représentation des tables par colonnes plus adaptée aux calculs d'agrégats, ou dont la taille des tables, gigantesques, conduit à des performances médiocres de la part des systèmes de gestion de base de données.

Ce domaine est illustré par les applications décisionnelles faisant de gros calculs sur les colonnes et par des applications en astrophysique faisant des calculs sur des tables immenses. Dans les deux cas, même si conceptuellement on manipule des tables, on développe des algorithmes *ad hoc* opérant sur des données massivement distribuées et exploitant des architectures parallèles et des *clusters* de machines.

MapReduce permet une parallélisation des tâches en distribuant les données plutôt que le code

Le second domaine d'applications est illustré, au-delà des volumes de données importants, soit par des données non relationnelles (documents, images…), donc sans connaissance préalable d'une structure, soit par des logiques de traitement qui ne s'expriment pas aisément avec l'algèbre relationnelle. C'est le cas des algorithmes de classification ou d'apprentissage. Là encore, on a recours à une programmation *ad hoc*.

Notons que dans les deux domaines d'applications, le relationnel n'est pas exclu, mais seulement limité à quelques tâches. Par exemple, dans un projet de *machine learning*, la phase de prétraitement peut s'appuyer sur une technologie relationnelle, alors que la phase d'apprentissage proprement dite sera faite par un algorithme approprié.

Cette approche, notée NoSQL, n'est pas nouvelle, des interfaces entre les langages de requêtes et les langages de programmation ont toujours existé pour développer des applications complexes, sous-traitant aux systèmes de gestion de base de données ce qu'ils peuvent faire de mieux et laissant le programmeur réaliser les autres tâches. On la retrouve également dans les systèmes à base de *workflows* où l'on coordonne un ensemble d'activités pouvant être réalisées par des technologies différentes.

Aujourd'hui, ce mode d'ingénierie est toujours pratiqué, avec un regard plus spécifique sur ces tâches. Il s'agit en général d'opérations complexes dont l'optimisation nécessite la mobilisation de ressources de calcul particulières. Ce type de développement est généralement guidé par des patrons (ou *patterns*) de programmation qui intègrent l'invocation de services de données existant et le code de l'utilisateur.

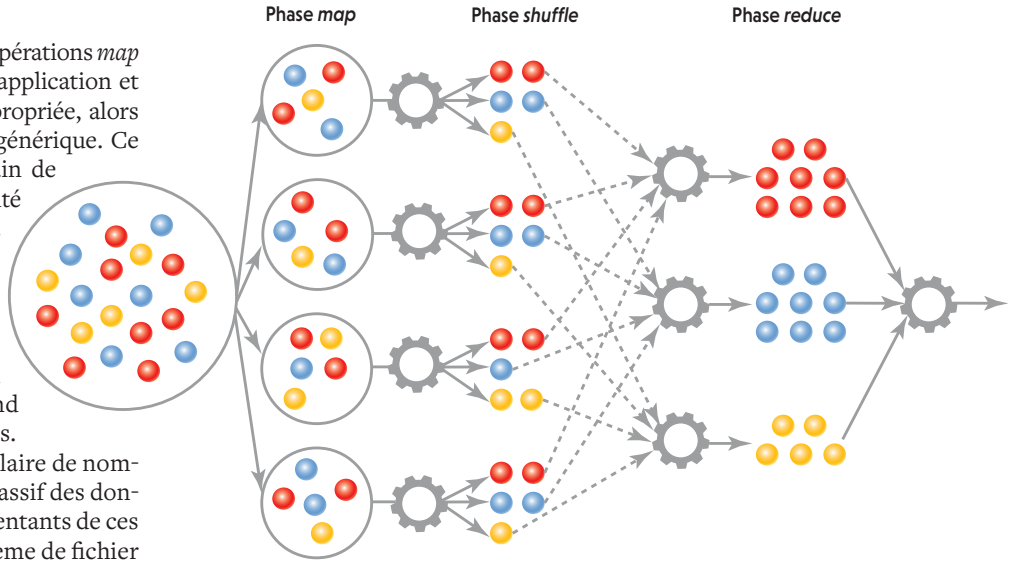
Un de ces *patterns*, connu depuis longtemps en programmation fonctionnelle, est MapReduce (*voir la figure page suivante*). Il permet une parallélisation des tâches à grande échelle en distribuant les données au lieu de distribuer le code. Pour illustrer ce *pattern*, imaginons une opération visant à classifier un sac de billes en fonction de leurs couleurs et à évaluer ensuite la taille de chaque classe et le nombre total de billes. Cette opération peut être réalisée par une seule personne, mais elle sera longue, surtout si le sac de billes est très grand. On recrute alors dix enfants entre qui on partage les billes. En augmentant le parallélisme, on accélère le processus, mais on a à la fin dix classements, chacun produisant des paquets de billes selon leurs couleurs.

Après un travail intermédiaire de regroupement des paquets par couleur, on recrute un nouveau groupe d'enfants et on leur demande de compter le nombre de billes de chaque couleur. À la fin, une seule personne fait la somme des résultats intermédiaires.

Dans ce principe général du modèle de calcul MapReduce, la phase *map* répartit les données (les billes) sur plusieurs serveurs (les enfants), chacun réalisant la même tâche (reconnaissance de la couleur et classement des billes). La phase *reduce* prend les paquets de données intermédiaires et fait la tâche suivante (le comptage). Entre les deux, une «boîte noire» (le *shuffle*) regroupe les paquets par couleur, avant de les acheminer aux serveurs >

> suivants qui feront le *reduce*. Les opérations *map* et *reduce* sont dépendantes de l'application et donc programmées de façon appropriée, alors que le *shuffle* est une opération générique. Ce modèle de calcul permet un gain de temps significatif grâce à sa capacité de mobilisation de ressources à chacune de ses phases d'exécution, mais le prix à payer est au niveau du développement qui nécessite une compétence pointue et au niveau de la mise en œuvre qui dépend d'un grand nombre de paramètres techniques.

Ce modèle est la pierre angulaire de nombreux systèmes de traitement massif des données. Hadoop est l'un des représentants de ces systèmes, combiné avec un système de fichier massivement distribué, de nombreuses extensions ont été réalisées soit au cœur du système pour améliorer ses performances soit au-dessus du système (on parle de surcouches) pour offrir des interfaces de développement plus riches et plus simples.



MapReduce accélère le processus de traitement des données grâce au parallélisme, c'est-à-dire à une distribution des tâches. Ici, elle consiste en une répartition des données pour les classer. D'abord, le paquet initial est divisé en quatre parties *(à gauche)*, chacune étant ensuite classée *(au centre)*. Les résultats partiels sont enfin réunis en un seul *(à droite)*.

AU-DELÀ DU CALCUL

Si les opérations de calcul sur les données sont importantes et cristallisent à juste titre l'attention des spécialistes et des programmeurs, la gestion des données ne se limite pas à ces opérations. L'indexation des données et les techniques de hachage, c'est-à-dire la répartition des données sur divers disques, sont cruciales pour la gestion des données, quel que soit le modèle de calcul choisi. Indexer des tables comportant des milliards de lignes ou des documents de plusieurs centaines de millions de pages est une tâche coûteuse. Les techniques mises en œuvre rivalisent d'ingéniosité à la fois dans la façon de construire l'index pour accélérer les accès aux données et dans les performances de cette construction.

Outre l'indexation et le hachage, la réplication des données est aussi un bon moyen d'augmenter la disponibilité de l'information en multipliant les exemplaires et les points d'accès. La réplication est très importante dans un système distribué afin de réduire les transferts de données, souvent coûteux, sur le réseau. Cependant, les avantages de l'indexation et de la réplication sont vite anéantis quand les données sont fréquemment mises à jour, ce qui n'est heureusement pas le cas dans de nombreuses applications des *big data*.

La protection des données reste un sujet sensible lorsqu'il s'agit de données personnelles, de santé ou concernant des processus industriels. De nombreux systèmes intégrés de gestion de données sont dotés de mécanismes d'anonymisation (*voir L'art de préserver l'anonymat, par T. Allard, page 98*), de cryptage, ou de restriction d'accès à certaines données. Des hiérarchies de droits sont souvent associées à des hiérarchies d'utilisateurs contrôlant ainsi l'accès aux données sensibles. Les algorithmes associés à ces garde-fous sont souvent complexes.

Dans le registre des risques, ajoutons que la complexité des algorithmes mis en œuvre dans l'exploitation des *big data* peut entraîner des effets de bord qui, si l'on n'y prend pas garde, peuvent poser des problèmes d'éthique, violer la vie privée, réduire la liberté et avoir un impact fortement intrusif dans la vie des personnes. Ce sont des questions essentielles mises en débat récemment (*voir l'entretien avec N. Boujemaa, page 104*).

La sensibilité des données n'est pas seulement liée à leur usage, elle peut être liée à la technologie. Les erreurs de matériel ou logicielles ne sont pas rares. Ces incidents ne doivent pas mettre le système de données dans un état incohérent, avec des risques de perte d'informations. Des mécanismes de fiabilisation sont développés, notamment au niveau des systèmes de gestion de base de données pour protéger ces systèmes et permettre un redémarrage à chaud ou à froid, sans perte d'informations et sans incohérence sur les données globales. Ces algorithmes de protection des données à tous les niveaux occupent une place stratégique. Ils sont complexes et leur mise en œuvre n'est pas sans incidence sur les coûts de calcul.

En fin de compte, les algorithmes destinés à la gestion des *big data* constituent une vaste faune, toujours plus diversifiée et riche à mesure des développements. C'est que, pour filer la métaphore, leur monde, celui des *big data*, ne cesse de croître. D'ailleurs, la comparaison avec le pétrole trouve ici ses limites. Les ressources en cette énergie fossile sont inéluctablement appelées à se tarir, alors que celles en données sont en augmentation permanente. Et une donnée peut être utilisée autant de fois que nécessaire sans se consumer. ■

BIBLIOGRAPHIE

M. BOUZEGHOUB ET R. MOSSERI (DIR.), ***Les Big Data à découvert,*** **CNRS Éditions, 2017.**

S. ABITEBOUL, «Sciences des données: de la logique du premier ordre à la Toile», Leçon inaugurale de la Chaire d'Informatique et sciences numériques au Collège de France, 2012.

ABONNEZ-VOUS À POUR LA SCIENCE

NOUVELLE FORMULE

FORMULE PASSION -27%

79€

FORMULE DÉCOUVERTE -25%

59€

FORMULE INTÉGRALE -41%

99€

BULLETIN D'ABONNEMENT

À renvoyer accompagné de votre règlement à : Pour la Science - Service abonnements - 19 rue de l'industrie - BP 90 053 - 67 402 Illkirch cedex

❑ OUI, JE M'ABONNE À *POUR LA SCIENCE* POUR 1 AN, JE CHOISIS MA FORMULE CI-DESSOUS :

❑ FORMULE **INTÉGRALE**
12 n° de *Pour la Science* + 4 Hors-Séries
+ Accès illimité aux archives en ligne depuis 1996
I1A99E

99€ Au lieu de ~~168,45€~~*

❑ FORMULE **PASSION**
12 n° de *Pour la Science*
+ 4 Hors-Séries
P1A79E

79€ Au lieu de ~~108,45€~~*

❑ FORMULE **DÉCOUVERTE**
12 n° de *Pour la Science*
D1A59E

59€ Au lieu de ~~78,45€~~*

J'INDIQUE MES COORDONNÉES

❑ M. ❑ Mme

Nom : _____ Prénom : _____

Adresse : _____

_____

Code postal |_|_|_|_|_| Ville : _____

Téléphone |_|_|_|_|_|_|_|_|_|_|

• Mon e-mail : (à remplir en majuscule)

_____@_____

J'accepte de recevoir les informations de *Pour la Science* ❑ OUI ❑ NON

et de ses partenaires ❑ OUI ❑ NON

• Je choisis mon mode de règlement :

❑ Par chèque à l'ordre de *Pour la Science*

❑ Par carte bancaire

N° |_|_|_|_| |_|_|_|_| |_|_|_|_| |_|_|_|_|

Date d'expiration : |_|_| |_|_| Clé (Les 3 chiffres au dos de votre CB) : |_|_|_|

Signature obligatoire

PAG18098STDC

* Par rapport au prix de vente en kiosque, l'accès numérique aux archives. Délai de livraison : dans le mois suivant l'enregistrement de votre règlement. Offre réservée aux nouveaux abonnés, valable jusqu'au 30/06/2018 en France métropolitaine uniquement. Pour un abonnement à l'étranger, merci de consulter notre site www.pourlascience.fr. Conformément à la loi "Informatique et libertés" du 6 janvier 1978, vous disposez d'un droit d'accès et de rectification aux données vous concernant en adressant un courrier à *Pour la Science*. Photos non contractuelles.

Groupe Pour la Science - Siège social : 170 bis, boulevard du Montparnasse, CS20012, 75680 Paris cedex 14 - Sarl au capital de 32 000€ - RCS Paris B 311 797 393 - Siret : 311 797 393 000 23 - APE 58.14 Z

CALCULER plus vite,

© Shutterstock.com/chombosan