



BIG DATA : UNE NOUVELLE FORME D'INTELLIGENCE COLLECTIVE

Mini-série de Billets #3 : Des moteurs de recherche au Big Data. Et la résolution des conflits ?

Billet #3B - Hadoop – une architecture distribuée, un nouveau paradigme infrastructurel

La genèse du Big Data se trouve dans la nécessité initiale d'améliorer l'efficacité des moteurs de recherche ; c'est ce lien qui porte le nom d'*Hadoop* que je vous fais découvrir dans cette mini-série de Billets #3 qui poursuit un double objectif : cerner la notion de Big Data, d'une part et clarifier l'impact de ces innovations sur les modes résolution des conflits, d'autre part.

Architecture distribuée vs architecture centralisée

Hadoop s'installe exclusivement sur un *cluster computing*ⁱ. Cette expression vient de l'anglais *cluster* qui signifie grappe, régime, groupe, communauté, amasⁱⁱ. D'où sa traduction française : grappe de calcul. On parle aussi de cluster d'ordinateurs.

Un cluster c'est d'abord une forme d'architecture distribuée, par opposition à une architecture centralisée autour d'un serveur. A cet endroit se situe précisément le changement de paradigme. On change de modèle d'architecture, on quitte les architectures centralisées pour aller vers des architectures distribuées.

Plusieurs raisons expliquent la nécessité de ce changement. Notamment, le fait que dans les architectures centralisées les goulots d'étranglement dans le réseau sont fréquents. En outre, ces architectures ne conviennent pas aux traitements analytiques. La croissance du volume des données augmente la charge de calcul au niveau du serveur central, et donc le temps de latence, à savoir le temps mis par le traitement d'une tâche pour s'achever ; cette dernière s'avère préjudiciable pour les usages reposant sur des réponses en quasi temps réelⁱⁱⁱ.

Architectures distribuées

Juvénal CHOKOGOUE, notre auteur de référence, opère une distinction parmi les types d'architectures distribuées. Il considère qu'il existe des « fausses » architectures distribuées : les architectures client/serveur (ou architectures délocalisées). Elles ne représentent pas de « vraies » architectures distribuées car elles « transfèrent simplement la charge de traitement sur un autre poste, tandis que, dans une architecture distribuée, le traitement est divisé en sous-tâches, ensuite ces sous-tâches sont distribuées pour le traitement à un ensemble d'ordinateurs qui sont vus comme un seul serveur central. La philosophie n'est vraiment pas la même »^{iv}. Il estime que le *cluster computing* (grappe de calcul ; plusieurs ordinateurs traitent le même problème) et le *grid computing* (grille de calcul ; un/plusieurs problèmes différents sont partagés entre plusieurs ordinateurs qui les traitent de manière indépendante) forment les « vraies » architectures distribuées^v. Hadoop devant être installé sur une architecture de cluster computing, nous concentrerons notre attention sur ce modèle et ses spécificités pour comprendre l'amélioration des moteurs de recherche, son lien avec le Big Data, et finalement l'éventuel impact sociologique de cette invention, en particulier sur les modes de résolution des conflits.

Un cluster d'ordinateurs se définit comme le regroupement de dizaines, de centaines, voire même des milliers de PC classiques interconnectés par des câbles réseau au moyen de commutateurs (*switch* en anglais ; lien physique). Plusieurs casiers qui contiennent chacun entre 8 et 64 nœuds (ordinateurs) le constituent. Les ressources nœuds sont les suivantes, partagées différemment selon le cluster : la mémoire RAM, le disque dur, le processeur CPU. Dans le cluster Hadoop, en termes de partage des ressources, chaque nœud possède son propre disque dur, sa propre mémoire, son propre processeur. On parle d'architecture *cluster shared-nothing*. Comme son nom l'indique, « rien n'est partagé », chaque nœud est autosuffisant et utilise ses propres ressources pour accomplir le travail qui lui incombe. Il existe d'autres possibilités de partager les ressources au sein d'un cluster, mais les présenter excéderait les objectifs de cette mini-série de Billets #3, d'autant plus que le *shared-nothing* constitue le mode de partage par défaut d'un cluster, mode utilisé dans de nombreuses problématiques Big Data^{vi}.

Si j'ose la comparaison, dans les cafétérias il est fréquent de rencontrer des casiers où déposer la vaisselle utilisée. Les casiers représentent les casiers, les plateaux seraient les nœuds, et l'ensemble des casiers formerait le cluster.

Quelle communication à l'intérieur du cluster : y a-t-il un pilote dans l'avion Hadoop ?

Shared-nothing signifie que les nœuds du cluster ne partagent pas leurs ressources. Ils sont indépendants les uns des autres. Puisque le cluster est vu par l'utilisateur comme un seul ordinateur, comment les nœuds communiquent-ils entre eux ? Il existe un chef d'orchestre, un orchestrateur de tâches qui planifie et gère les activités du cluster, l'ensemble de la communication qui s'y déroule. Cette couche logicielle (lien virtuel) placée au-dessus des nœuds permet effectivement à l'utilisateur de travailler avec le cluster comme s'il s'agissait d'un seul ordinateur. Ingénieux ! Dans le modèle Hadoop, un nœud maître (le nœud de référence) répartit les tâches entre les nœuds du cluster (esclaves) et gère la redondance/réplication du stockage des données via le Hadoop Distributed File System

(HDFS)^{vii}. Voilà pourquoi ce modèle de communication s'appelle maître/esclave (master-slave), différent du modèle peer-to-peer dans lequel la charge de l'exécution de la requête dans le cluster appartient à tous les nœuds^{viii}.

En résumé, l'architecture Hadoop adéquate est un cluster shared-nothing maître/esclave^{ix}.

Voici en image à quoi peut ressembler un cluster. Impressionnant !



Source : Grappe de serveurs, https://fr.wikipedia.org/wiki/Grappe_de_serveurs

Les quatre caractéristiques et avantages des infrastructures distribuées de type cluster par rapport aux architectures centralisées.

1) La scalabilité horizontale

La scalabilité d'un système informatique se définit comme étant sa capacité à passer à l'échelle ou à monter en charge, autrement dit à maintenir en cas de forte demande ses fonctionnalités et ses performances. En application de la désormais fameuse loi de Moore selon laquelle « le coût des ordinateurs se réduit de moitié tous les 18 mois, en raison de l'augmentation du nombre de transistors dans les microprocesseurs »^x, ajouter un nœud à un cluster (scalabilité horizontale) est moins coûteux que d'augmenter les capacités d'un serveur central (scalabilité verticale, *uprizing*). Et ainsi, *Wikipédia* de conclure : « En conséquence, les machines électroniques sont devenues de moins en moins coûteuses et de plus en plus puissantes »^{xi}. On comprend donc qu'opter pour une architecture de type cluster ne constitue pas un choix anodin pour une entreprise. A la fois architecture qui sert de support au traitement massivement parallèle^{xii} car possédant les atouts du calcul intensif permet en outre de profiter des rendements d'échelle engendrés par la baisse des coûts des ordinateurs ; cela participe à l'adoption d'Hadoop en entreprise. Ainsi, gérer le volume croissant de données, se fait simplement en augmentant les nœuds dans le cluster d'ordinateurs^{xiii}.

« Passer à l'échelle, monter en charge » ; concept central pour comprendre l'évolution de l'amélioration des moteurs de recherche au Big Data ainsi que la popularité et le choix des clusters pour traiter la déferlante de données à laquelle nous assistons aujourd'hui. « Constituer un data center de 1000, 10000 machines ou plus revient aujourd'hui bien moins cher que faire passer à l'échelle un serveur qui centralise la gestion des données »^{xiv}.

2) La scalabilité linéaire

Un système informatique est dit scalable linéairement lorsque sa performance augmente de façon proportionnelle à l'ajout de nouveaux composants. Cette définition vaut également pour un cluster. Linéaire qualifie la performance des architectures linéairement scalables qui augmente à un taux constant avec l'augmentation du nombre de processeurs. L'ajout progressif de nœuds supplémentaires dans le cluster (scalabilité horizontale) permet de stabiliser, dans le temps, le temps de traitement induit par la croissance du volume des données. La scalabilité linéaire du cluster computing, modèle unique de regroupement d'ordinateurs qui permet d'obtenir la scalabilité linéaire, possède un avantage concurrentiel indiscutable par rapport aux architectures non-scalables ou à scalabilité non-linéaire dont la performance finit par plafonner après l'ajout d'un certain nombre de processeurs^{xv}.

C'est le partage des ressources de type shared-nothing qui permet au cluster sur lequel Hadoop s'appuie de faire du calcul intensif car il garantit la scalabilité horizontale et linéaire quasi illimitées du cluster, ainsi que la tolérance aux pannes et la haute disponibilité^{xvi}.

3) La tolérance aux pannes

La tolérance d'un cluster aux pannes ou aux défaillances correspond à sa capacité à fonctionner malgré les pannes, spécialement les pannes de nœuds. On parle aussi de résilience. Elle est obtenue grâce à la redondance des données dans le cluster : chaque bloc de fichier traité par chaque nœud est répliqué dans le cluster ; dans le cas d'Hadoop, la réplication a lieu trois fois par défaut, par le biais du Hadoop Distributed File System (HDFS). Ainsi, lorsqu'un nœud tombe en panne, l'astuce est la suivante : le système passe le relais à un autre nœud qui contient une copie des données identiques à celles traitées par le nœud défaillant, en cas de manque de ressources, un autre ordinateur doté de plus de ressources prend le relais^{xvii}.

4) La haute disponibilité

La disponibilité d'un système signifie qu'il est opérationnel et capable de répondre aux requêtes des utilisateurs. A la différence des architectures centralisées dans lesquelles la disponibilité de tout le système repose entièrement sur le serveur central (point de défaillance unique, *single point of failure*), les architectures distribuées offrent la capacité au système de continuer à fonctionner malgré les défaillances. Nous avons vu ci-dessus le mécanisme de relais qui s'enclenche en cas de panne. Tous les systèmes étant en soi un peu disponibles, on l'espère, la haute disponibilité ne représente pas une caractéristique per se d'un cluster, il s'agit plutôt de la mesure d'une caractéristique. Hadoop constitue un système à haute disponibilité. Dans le cas d'un cluster, la haute disponibilité consiste à minimiser son niveau d'interruption, mais minimiser jusqu'à quel point ? La réponse varie selon les domaines. Evidemment il est des contextes dans lesquels la disponibilité peut revêtir une dimension de vie ou de mort : les hôpitaux, l'aviation, par exemple. Ces exigences-métier sont formalisées dans un SLA (*Service Level Agreement*). Les deux principes suivant régissent l'obtention de la haute disponibilité d'un système : l'élimination des points de défaillance uniques dans le système, déjà vue ci-dessus, d'une part, et la détection automatique des défaillances, d'autre part^{xviii}. Si vous souhaitez approfondir ce sujet, êtes intéressés par les aspects techniques et les cinq manières différentes de construire un cluster pour le rendre hautement disponible, je

vous invite à lire les pages 41 à 43 de l'ouvrage de Juvénal CHOKOGOUE, *Hadoop : devenez opérationnel dans le monde du Big Data*^{xix}. Ce sujet excédant largement les questions posées dans le Billet #3, je ne m'y arrêterai pas.

Impact sociétal et résolution des conflits ?

Le traitement de la déferlante de données impose un type d'architecture, notamment pour bénéficier de la scalabilité linéaire : un cluster computing, forme d'architecture distribuée. Il s'agit d'un changement de paradigme architectural. Les anciennes architectures centralisées sont dépassées par le traitement des données massives. Compte tenu des usages croissants et variés faits de ces données, les technologies s'appuyant sur une architecture distribuée sous-tendent notre société désormais numérique. Le changement est technologique certes, mais il est également profondément sociétal. Alors, ces nouvelles technologies constituent-elles des forces horizontalisantes, décentralisantes ? En quoi les modes de résolution des conflits, en particulier le mode judiciaire vertical, centralisé, asymétrique, par définition, pourrait-il se trouver affecté par ces forces décentralisantes ? Comment le principe sociologique reliant justice et société - un modèle de société engendrant un mode de régulation correspondant^{xx}- trouvera-t-il application à l'ère numérique ? Début de réponse dans le dernier Billet de cette mini-série #3. Avançons pas à pas vers cette discussion.

Il existe aussi une définition logicielle d'un cluster Hadoop

Dans ce Billet #3B, j'ai défini un cluster Hadoop d'un point de vue principalement architectural, physique. Il est possible par ailleurs de donner une définition logicielle du cluster Hadoop ; la voici : un cluster Hadoop est simplement un cluster sur lequel Hadoop a été installé, soit deux éléments : premièrement, l'ensemble des classes JAVA d'implémentation de l'algorithme MapReduce et, deuxièmement, le Hadoop Distributed System File (HDSF) comme système de fichiers sur l'ensemble des disques durs des nœuds du cluster^{xxi}. Ces notions feront l'objet du billet suivant.

A bientôt pour le Billet #3C : *Hadoop – des logiciels : MapReduce et le Hadoop Distributed File system – un nouveau paradigme de programmation !*

Anne-Sylvie Weinmann

Nota bene :

Tous les billets de cette série seront publiés sur LinkedIn mais également disponibles en format pdf avec les références détaillées sur le blog de mon site (www.medialien.ch).

An English version of this series of posts - Big Data: a new form of collective intelligence - will follow in a while.

Références & Notes :

-
- i CHOKOGOUE Juvénal, *Hadoop : devenez opérationnel dans le monde du Big Data*, St-Herblain, ENI, 2017, p. 63 (<https://m.editions-eni.fr/livre/hadoop-devenez-operationnel-dans-le-monde-du-big-data-9782409007613#>).
- ii *Cluster*, <http://www.wordreference.com/enfr/cluster>
- iii CHOKOGOUE Juvénal, *op. cit.*, pp. 31, 33, 34, 117.
- iv CHOKOGOUE Juvénal, *op. cit.*, p. 36.
- v CHOKOGOUE Juvénal, *op. cit.*, pp. 28, 31, 37, 63, 65, 71.
- vi CHOKOGOUE Juvénal, *op. cit.*, pp. 31, 35, 45ss (notamment 45, 46, 49), 72.
- vii Voir Billet #3C (à venir).
- viii CHOKOGOUE Juvénal, *op. cit.*, pp. 34, 50, 51, 63.
- ix CHOKOGOUE Juvénal, *op. cit.*, p. 65.
- x CHOKOGOUE Juvénal, *op. cit.*, p. 39.
- xi *Loi de Moore*, https://fr.wikipedia.org/wiki/Loi_de_Moore
- xii Voir Billet #3C.
- xiii CHOKOGOUE Juvénal, *op. cit.*, pp. 21, 37-39, 63.
- xiv CHOKOGOUE Juvénal, *op. cit.*, p. 39.
- xv CHOKOGOUE Juvénal, *op. cit.*, pp. 38-39.
- xvi CHOKOGOUE Juvénal, *op. cit.*, p. 63.
- xvii CHOKOGOUE Juvénal, *op. cit.*, pp. 33, 40, 41, 111.
- xviii CHOKOGOUE Juvénal, *op. cit.*, pp. 33, 41, 43.
- xix CHOKOGOUE Juvénal, *Hadoop : devenez opérationnel dans le monde du Big Data*, St-Herblain, ENI, 2017, p. 19 (<https://m.editions-eni.fr/livre/hadoop-devenez-operationnel-dans-le-monde-du-big-data-9782409007613#>).
- xx BONAFE-SCHMITT Jean-Pierre, *La médiation : une justice douce*. Paris, Syros-Alternatives, 1992, p. 180.
- xxi CHOKOGOUE Juvénal, *op. cit.*, p. 63.