

Les réseaux informatiques

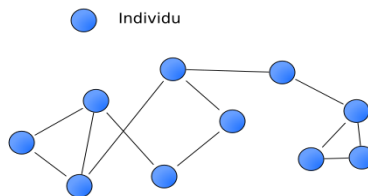
Serge Abiteboul & Pierre Sennelart

RÉSUMÉ : En informatique et donc dans le monde numérique, les réseaux jouent un rôle central. Mathématiquement, un réseau (ou graphe) est un ensemble d'objets (ou nœuds) reliés les uns aux autres par des relations (ou connexions ou arcs ou fils ou arêtes). En particulier, les réseaux de communications, tels qu'internet, ont déplacé la science informatique loin du modèle classique des machines de Turing. Le calcul peut être distribué entre une multitude de processeurs. Cela change tout : la puissance des logiciels, leur vérification, les raisonnements qu'ils réalisent... jusqu'à nos utilisations individuelles, ou collectives. L'essence même de la pensée informatique est transformée.

MOTS CLÉS : Réseau, informatique, toile, internet, communs.

Introduction

En informatique et donc dans le monde numérique, les réseaux jouent un rôle central. Mathématiquement, un réseau (ou graphe) est un ensemble d'objets (ou nœuds) reliés les uns aux autres par des relations (ou connexions ou arcs ou fils ou arêtes).



Exemple de réseau reliant des individus

Les « réseaux sociaux numériques »¹ connectent par exemple les humains entre eux, leur permettant d'échanger du texte, des images, des vidéos. Ces réseaux sociaux enrichissent et redéfinissent les liens entre ces humains.

¹ Serge ABITEBOUL & Jean CATTAN, *Nous sommes les réseaux sociaux*, Paris, Odile Jacob, 2022.

L'informatique permet également de construire des liens entre des textes pour proposer un réseau hypertexte, le *World Wide Web*², la toile en français. Sur le *Web*, les humains consultent des pages *web*, regroupées en sites. Les pages sont ainsi les nœuds de la toile, et les hyperliens qui relient les pages *web* entre elles en constituent les fils.

Enfin, avec l'informatique, il est devenu possible de constituer des réseaux d'ordinateurs, et au-delà, le réseau Internet³, un réseau de réseaux sur lequel fonctionnent des services comme le courrier électronique ou la toile. Internet se déploie entre des millions d'ordinateurs couvrant de ses mailles toute la planète.

Ce chapitre s'intéresse à cette dernière classe de réseaux, les réseaux entre ordinateurs, et aux transformations radicales qu'ils apportent à l'informatique, à nos vies et également à notre société.

Une vision moderne de l'informatique

L'ordinateur⁴, tel qu'envisagé et conçu au milieu du XX^e siècle par les pionniers de l'informatique comme Alan Turing et John von Neumann, est simplement composé d'un processeur qui calcule et d'une mémoire qui garde les données. Cela est illustré par le schéma de l'architecture de von Neumann, un modèle primitif de l'ordinateur.

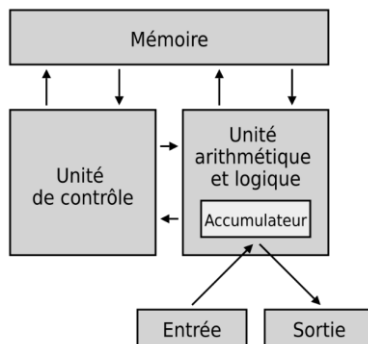


Schéma simplifié de l'architecture de von Neumann, CC-BY-SA Chris-martin & Aeroid

² Tim BERNERS-LEE & Mark FISCHETTI, *Weaving the Web. The Past, Present and Future of the World Wide Web by its Inventor*, Londres, Texere, 2000.

³ Christian HUITEMA, *Et Dieu créa l'Internet...*, Paris, Eyrolles, 1996.

⁴ Gilles DOWEK et al., *Informatique et sciences du numérique. Spécialité ISN en terminale S*, Paris, Eyrolles, 2012.

Une telle machine fonctionne suivant une unité de lieu (organisée autour de sa mémoire), de temps (le temps du calcul séquentiel), et d'action (la tâche réalisée).

L'informatique moderne a battu en brèche ce modèle. Un système informatique est typiquement composé aujourd'hui de nombreux composants logiciels ou matériels (distribués géographiquement), qui coopèrent de manière asynchrone (sans disposer de signal d'horloge global) pour réaliser simultanément plusieurs tâches (multitâches). Ce sont les comportements de chacun des composants et leurs interactions entre eux et avec le monde réel, qui lui permettent de réaliser ses objectifs. Considérons quelques exemples typiques de tels systèmes.

L'ordinateur ou le téléphone intelligent

Quand on parle d'un ordinateur, cela semble indiquer qu'il s'agit d'un objet, avec une certaine centralisation. Cela n'est qu'illusoire. On y trouve typiquement plusieurs processeurs, de la mémoire rapide pour les calculs, de la mémoire de masse (disque dur ou SSD) pour stocker des données, peut-être un processeur graphique séparé et une carte réseau (pour interagir avec le reste du monde). Même au sein du processeur de calcul central ou du processeur graphique, il n'y a pas de centralisation : ces processeurs sont en général formés de cœurs ou d'unités de calcul indépendantes qui permettent d'exécuter en parallèle un grand nombre d'opérations. Par ailleurs, nombre d'applications tournant sur un tel ordinateur dépendent de calculs effectués sur d'autres machines, accessibles grâce à Internet. C'est encore plus vrai sur un *smartphone*, qui n'est rien d'autre qu'un ordinateur de petit format prévu pour être toujours connecté. L'architecture d'un ordinateur aujourd'hui, comme un téléphone intelligent, peut être aussi complexe que celle des ordinateurs qui ont permis d'envoyer des astronautes sur la lune, et avoir plus de puissance de calcul que ces ordinateurs-là.

Au Moyen Âge, c'est le travail simultané d'un nombre impressionnant d'humains et la combinaison de leurs talents qui a conduit aux cathédrales. Dans ce que l'on peut voir comme une analogie moderne, le travail simultané de masses d'unités de calcul (à l'intérieur d'un ordinateur ou au sein d'un *data center*) et leurs performances individuelles conduisent à des ordinateurs ou des smartphones, ces cathédrales des temps modernes, témoignages des progrès technologiques de notre temps.

Data centers

Prenons maintenant le cas d'un *data center* (centre de données en français) qui réalise typiquement des millions de milliards d'opérations par seconde. Son nom suggère une certaine centralité, celle du bâtiment qui l'héberge. Là encore, la centralisation est illusoire. On y trouve un très grand nombre de machines avec différentes spécialisations :

des serveurs composés d'un nombre considérable de processeurs qui réalisent des calculs,

des baies de stockage pour permettre la persistance dans le temps de certaines données,

des équipements réseaux pour connecter les machines entre elles, et au réseau internet et

des serveurs responsables de gérer de gros volumes d'entrées et sorties de données.

Ces millions de processeurs ne sont pas synchronisés par une horloge unique et les *data centers* ne sont pas souvent dirigés par un serveur « maître » qui orchestre le travail de tous les autres ordinateurs. Dans ce genre d'architecture basée sur un serveur « maître », ce maître devient un point de faiblesse : s'il est en panne, tout s'arrête.

Internet

Le système Internet qui supporte les communications à l'échelle de la planète est constitué d'un réseau de millions de réseaux universitaires, commerciaux et gouvernementaux. Pour faire fonctionner le courrier électronique, le Web, etc., les données, découpées en petits paquets, sont échangées sur Internet entre les serveurs en s'appuyant sur des règles standardisées, des « protocoles de transfert de données ». Une caractéristique essentielle de ce système est son absence totale de centralisation, ce qui constitue une grande différence entre la toile informatique et celle de l'araignée. L'absence de centre névralgique qui en résulte lui procure une résilience considérable qui a séduit les militaires qui ont accompagné sa naissance (l'agence ARPA⁵).

⁵ Une agence du Département de la Défense des États-Unis chargée de la recherche et développement des nouvelles technologies destinées à un usage militaire, connue maintenant sous le nom de DARPA.

Le réseau Internet est comme une structure vivante avec des machines qui naissent et d'autres qui s'effacent. Les connexions s'étendent, se ramifient, proposant de multiples chemins où les données se côtoient et se croisent sans cesse. Prenons une simple recherche sur la toile. Votre question traverse le réseau, déclenchant des réactions sur plusieurs machines. Vous choisissez une des propositions de réponses, vous surfez ensuite sur le réseau, initiant des échanges de fragments narratifs, des flux de questions-réponses, comme le récit d'un roman, une intrigue littéraire en construction.

Ce que cela change dans la science informatique

L'informatique théorique (une science récente qui date comme l'informatique du milieu du XX^e siècle) a évolué très vite et a obtenu des résultats superbes. La calculabilité permet par exemple de distinguer les problèmes qui peuvent être résolus informatiquement (les problèmes calculables) de ceux qui ne le peuvent pas. On connaît également des classes de problèmes pour lesquels on peut montrer qu'en l'état actuel des connaissances scientifiques, aucune solution informatique ne peut fonctionner dans des temps raisonnables.

Pourtant, ces résultats parlent de systèmes qui ne collent plus à la réalité des systèmes informatiques car la technologie a, elle aussi, évolué très vite. Pour répondre aux évolutions techniques, des efforts sont réalisés pour introduire la notion de calcul parallèle et distribué dans les théories existantes, pour construire de nouvelles théories plus proches de la réalité, plus complexes. Il faut revisiter complètement certains problèmes. De plus, la distribution soulève également des questions nouvelles passionnantes.

Nous illustrons ces aspects par des exemples.

Tester les logiciels

Quand on conçoit un logiciel, il faut le tester, vérifier qu'il se comporte bien comme on le souhaite. Dans le modèle original, on peut tester un programme en l'exécutant pas à pas. Le test d'un logiciel qui s'exécute sur plusieurs machines tient beaucoup plus de l'art. Les erreurs sont difficiles à corriger, en particulier parce qu'elles ne sont pas simples à reproduire. L'absence d'horloge globale complique considérablement le problème.

Communiquer

Un autre défi pour un contexte distribué est de communiquer. Il faut améliorer les performances de ces communications, par exemple avec des techniques de compression qui savent s'adapter aux terminaux d'accès et à la charge du réseau. On a appris à « router » des messages dans des réseaux gigantesques comme internet, et à diffuser des informations vers un très grand nombre de nœuds. Les exigences en confidentialité ont conduit à des avancées fantastiques en cryptographie pour l'authentification, le chiffrement, et la signature, etc., et des protocoles comme RSA⁶.

Distribuer

On peut « prouver » des propriétés d'un code qui s'exécute séquentiellement sur une machine. Il est infiniment plus complexe de raisonner sur un code qui s'exécute dans un environnement distribué. Quand on a plusieurs machines, la question se pose par exemple de se mettre collectivement d'accord sur une décision particulière, comme de savoir par exemple si une place particulière dans un train précis est affectée à Mme Dupond ou à M. Martin qui en ont fait la demande au même instant. Comment faire que des programmes qui fonctionnent sur des machines différentes arrivent à se mettre d'accord ? C'est compliqué dans ce contexte parce que tout se passe de façon asynchrone entre les machines. On doit donc supposer que rien n'est fiable, ni les machines ni les réseaux ; tout peut tomber en panne ou se retrouver sous la coupe d'un hacker malfaisant. Ce problème, connu sous le nom de « problème du consensus distribué », est l'un des problèmes d'informatique les plus étudiés.

Distribuer pour la puissance de calcul

Une raison essentielle pour la distribution, à l'origine notamment des *data centers*, est d'augmenter la puissance de calcul. Pour certains problèmes, le parallélisme « marche » bien. Plus on dispose de machines, plus on peut accélérer la recherche de résultats (pour d'autres problèmes, c'est moins vrai).

Voyons un exemple où cela marche bien, la mise à disposition de films. Dans une architecture classique, les films sont stockés sur une machine unique qu'on appelle un « serveur » (il « sert » des films). Les utilisateurs du service qui veulent

⁶ Ronald RIVEST, Adi SHAMIR & Leonard ADLEMAN, « A method for obtaining digital signatures and public-key cryptosystems », *Communications of the ACM*, vol. 21, n° 2, 1978, p. 120-126.

visualiser un film sont les « clients ». Quand le nombre de clients augmente rapidement, le serveur est vite saturé. Il faut alors s'équiper d'un serveur plus puissant, avec une puissance de communication plus grande ; on peut aussi répliquer la collection de films sur un deuxième serveur qui sera vite à son tour saturé, etc. Une autre architecture est possible, dite en « pair-à-pair ». Un client qui visualise un film en stocke des morceaux et devient à son tour serveur pour ces morceaux. Son ordinateur est à la fois client et serveur, on parle de « pair ». Quand quelqu'un demande un film, il va chercher ses morceaux chez plusieurs pairs où ils sont stockés. Quand un film est très populaire, il est répliqué un peu partout et sans doute dans un voisinage proche de clients éventuels. Ce système est bien meilleur, en particulier pour les communications : quand on veut regarder un film très populaire, les morceaux de ce film se retrouvent tout près sur le réseau.

Distribuer pour la confidentialité des données

En centralisant, on centralise les données et donc on perd en confidentialité.

Un exemple est celui des réseaux sociaux. Le modèle dominant est celui de grands systèmes comme Facebook, TikTok, LinkedIn ou X, qui cherchent à capter tout le marché et qui ne permettent que des liens entre les utilisateurs de leur propre réseau. Des alternatives existent, comme Mastodon⁷ qui permet la collaboration de nombreux systèmes informatiques indépendants. Vous pouvez rejoindre n'importe lequel de ces systèmes et échanger avec des utilisateurs de n'importe quel autre. Cela fonctionne parce que ces systèmes utilisent tous le même protocole de communication, ActivityPub.

Un autre exemple est celui des grands modèles de langue des IA génératives telles que Copilot ou ChatGPT. Pour entraîner ces modèles, le système aspire des quantités considérables de données dans un *data center*. Tout le monde doit donc confier ses données à ce *data center*. Il existe une alternative : l'apprentissage fédéré⁸. Chacun garde ses données et on fait tourner le système d'apprentissage collectivement. Le résultat de l'entraînement, le modèle, est partagé mais pas les données. Bien entendu, la distribution (donc ici la protection des données privées) a un coût : l'apprentissage fédéré est plus complexe, donne potentiellement des résultats moins précis, et peut être difficile à protéger contre les adversaires.

⁷ <https://joinmastodon.org/>

⁸ Rachid GUERRAOUI, Nirupam GUPTA & Rafael PINOT, *Robust Machine Learning. Distributed Methods for Safe Ai*, Berlin, Springer Verlag, 2024.

Distribuer pour s'organiser autrement

L'informatique permet des communications et du partage de données. On peut collaborer pour résoudre ensemble des problèmes spécifiques. Comme chercheurs, nous écrivons des articles scientifiques avec des collaborateurs du monde entier que nous n'avons parfois jamais rencontrés, en partageant des fichiers ou des logiciels sur des systèmes d'édition distribuée, en discutant en visioconférence.

Au XVIII^e siècle, Diderot et D'Alembert ont fait collaborer plus de 160 spécialistes pour écrire leur encyclopédie. L'ambition était considérable. Aujourd'hui, Wikipédia est alimentée chaque jour par plus de cent mille contributeurs. Pour faire collaborer toutes ces personnes, l'informatique et la distribution sont indispensables.

De manière générale, l'informatique distribuée permet des organisations moins centralisées, dans lesquelles les participants bénéficient de plus d'autonomie. Elle met en avant des travaux plus coopératifs. Par exemple, le développement de communautés comme Wikipédia ou OpenStreetMap est facilité par l'informatique distribuée ; le développement collaboratif de logiciels open-source comme Linux également.

Distribuer ou pas

Distribuer ou ne pas distribuer, telle est (souvent) la question. À l'origine, on a développé de gros ordinateurs comme l'ENIAC. D'ailleurs, pendant un temps, on ne savait rien faire d'autre que de centraliser les calculs. On avait des serveurs de plus en plus puissants et on les pilotait d'abord à coup de cartes perforées, puis à partir de terminaux peu évolués. On a ensuite conçu des ordinateurs personnels puissants, puis des téléphones intelligents très puissants également. On pouvait penser que la distribution des calculs sur les machines personnelles des utilisateurs avait gagné. Mais sont arrivés les *data centers* qui d'une certaine manière recentralisent les calculs et les données (dans des entrepôts avec des millions de processeurs). Cela s'accompagne d'une centralisation capitaliste énorme autour d'une poignée de compagnies, notamment les célèbres GAFAM (Google, Apple, Facebook, Amazon, et Microsoft).

Pourtant, cette victoire de la centralisation n'est pas si évidente quand nos téléphones possèdent de plus en plus de puissance de calcul, quand le pair-à-pair continue à se développer, quand on parle de ramener une partie du traitement plus près de l'utilisateur avec l'informatique de périphérie (*edge computing*), quand des

voix s'élèvent contre la toute-puissance de quelques entreprises. La messe n'est peut-être pas dite.

Cette tension entre distribuer ou pas n'est pas que technique. Un des aspects clés de la centralisation est commercial : plus de centralisation permet d'augmenter les profits en fidélisant (de force) le client. Un autre aspect clé de la centralisation est qu'elle permet plus de simplification. Par exemple, on peut déployer en une seule commande la toute nouvelle version d'un logiciel sur des millions de processeurs dans un service centralisé. La même maintenance peut exiger des masses d'interventions dans un service distribué.

Les communs numériques

L'information, au début de l'informatique, résidait dans des îlots avec des formats et des applications fermés. Avec la distribution, on a pu faire collaborer des systèmes. Mais, pour que cela fonctionne, il faut souligner l'importance de protocoles et de formats ouverts. On doit faire interagir des programmes, ce qui conduit à se mettre d'accord sur des protocoles ouverts. Pour que les programmes puissent partager des données de différentes sources, il est nécessaire de s'appuyer également sur des formats de données ouverts. Ces protocoles et ces formats sont des communs numériques⁹ indispensables, typiquement promus par des organismes comme le W3C (World Wide Web Consortium). En poussant le trait, les logiciels sont eux-mêmes ouverts et les données en accès ouvert.

Bien sûr, le recours aux communs ne se fait pas sans des tentatives d'*enclosures*. Le terme, un anglicisme, fait référence à un mouvement du XVI^e siècle en Angleterre (« mouvement des enclosures ») qui a vu des champs communaux être enclos de haies ou de murs pour être privatisés. De manière assez semblable, on voit aujourd'hui des entreprises privées mettre la main sur des « communs numériques » (web, logiciels libres, données ouvertes, etc.) et chercher à cantonner les utilisateurs dans des silos. On assiste aussi à l'enfermement de portions d'Internet par des États totalitaires.

⁹ Serge ABITEBOUL & François BANCILHON, *Vive les communs numériques*, Paris, Odile Jacob, 2024.

Conclusion

L'histoire de l'informatique est encore jeune. De nouvelles technologies comme la technologie quantique pourraient complètement rebattre les cartes. Pourtant, il est déjà clair que l'ordinateur séquentiel des débuts, s'il reste un outil conceptuel essentiel notamment pour parler de calculabilité ou de complexité, est bien derrière nous.

Reliés aux réseaux numériques et aux machines intelligentes par nos téléphones, nos ordinateurs, peut-être un jour des neuroprothèses, nous allons de plus en plus cohabiter avec les intelligences artificielles, interconnectées par des fils invisibles. Nous tisserons ensemble le monde complexe de demain. Nous pouvons nous demander comment nous allons vivre avec ces systèmes informatiques de plus en plus nombreux, de plus en plus intelligents et comment ils vont transformer nos façons de vivre, nous transformer nous-mêmes.

Bibliographie

- ABITEBOUL, Serge & BANCILHON, François, *Vive les communs numériques*, Paris, Odile Jacob, 2024.
- ABITEBOUL, Serge & CATTAN, Jean, *Nous sommes les réseaux sociaux*, Paris, Odile Jacob, 2022.
- BERNERS-LEE, Tim & FISCHETTI, Mark, *Weaving the Web. The Past, Present and Future of the World Wide Web by its Inventor*, Londres, Texere, 2000.
- DOWEK, Gilles *et al.*, *Informatique et sciences du numérique. Spécialité ISN en terminale S*, Paris, Eyrolles, 2012.
- GUERRAOUI, Rachid, GUPTA, Nirupam & PINOT, Rafael, *Robust Machine Learning. Distributed Methods for Safe Ai*, Berlin, Springer Verlag, 2024.
- HUITEMA, Christian, *Et Dieu créa l'Internet...*, Paris, Eyrolles, 1996.
- RIVEST, Ronald, SHAMIR, Adi & ADLEMAN, Leonard, « A method for obtaining digital signatures and public-key cryptosystems », *Communications of the ACM*, vol. 21, n° 2, 1978