première expérience fut prometteur, mais non décisif, en partie parce qu'ils n'ont pu observé la réaction du plancton qu'une semaine durant. Le même groupe a répété l'expérience en 1995 et a pu poursuivre ses observations pendant quatre semaines : ils ont constaté sans ambiguïté que le fer ajouté accroît énormément la photosynthèse du phytoplancton, ce qui entraîne une «efflorescence» de ces micro-organismes, colorant les eaux en vert.

La fertilisation de l'océan

Depuis, trois autres groupes (en Nouvelle-Zélande, en Allemagne et aux États-Unis) ont démontré sans équivoque que l'ajout de petites quantités de fer dans l'océan Austral stimule la productivité du phytoplancton. L'expérience de fertilisation réalisée à plus grande échelle date des mois de janvier et de février 2002. Au cours du projet SOFeX, mené par l'Institut de recherche aquatique de la baie de Monterey et par le Laboratoire *Moss Landing Marine*, on a montré qu'une tonne de solution de fer dispersée sur environ 300 kilomètres carrés a décuplé la productivité primaire, en huit semaines.

Ces résultats ont convaincu la plupart des biologistes que le fer accélère la croissance du phytoplancton aux hautes latitudes, mais on ignore encore si cette productivité accrue renforce la pompe biologique et augmente le stockage de dioxyde de carbone dans les profondeurs de l'océan. Toutefois, selon les modèles mathématiques les plus récents, même si le phytoplancton incorporait tout l'azote et tout le phosphore disponibles dans les eaux de surface de l'océan Austral pendant les 100 prochaines années, 15 pour cent seulement du dioxyde de carbone libéré par la combustion des combustibles fossiles serait recapturé.

Malgré toutes les incertitudes concernant la fertilisation des océans, quelques groupes de laboratoires privés et publics travaillent sur des projets à grande échelle. Une entreprise a proposé de faire déverser par les navires de commerce qui traversent régulièrement l'océan Pacifique Sud de petites quantités d'un mélange d'«engrais». D'autres groupes étudient la possibilité de déclencher la floraison de phytoplancton en déversant directement dans les eaux côtières, par des tuyaux, des nutriments, dont le fer et l'ammoniaque.

Il n'est pas encore parfaitement établi que ces stratégies de fertilisation des océans soient un jour applicables. Pour être efficace, la fertilisation devra être conduite chaque année pendant des décennies. Comme la circulation océanique finit par remonter en surface les eaux profondes, tout le dioxyde de carbone supplémentaire stocké par la pompe biologique regagnerait l'atmosphère, quelques centaines d'années après la dernière campagne de fertilisation. En outre, quelles seraient les conséquences de telles opérations ? Les fermiers ne parviennent pas à confiner les engrais dans un champ, alors que dire de fertilisants déversés dans des océans turbulents ? Pour cette raison, de nombreux experts craignent que cette fertilisation à grande échelle cause des dommages à long terme qui seraient difficiles, voire impossibles, à endiguer.

Ainsi, on risquerait de perturber la chaîne alimentaire marine. Les simulations informatiques et les études des efflorescences naturelles de phytoplancton montrent que l'augmentation de la productivité primaire entraînerait un déficit en oxygène. Les micro-organismes qui consomment les cellules de phytoplancton mortes lorsqu'elles sombrent dans les profondeurs consomment parfois l'oxygène plus vite que la circulation océanique ne le remplace. Dans ces conditions, les animaux incapables de s'échapper vers des eaux plus riches en oxygène suffoqueraient.

Solution providentielle ou boîte de Pandore ?

De telles conditions favorisent également la croissance de microbes qui produisent du méthane et de l'oxyde d'azote, deux gaz à effet de serre dont la capacité à piéger la chaleur est supérieure à celle du dioxyde de carbone. D'après l'Office américain des océans et de l'atmosphère, un déficit en oxygène et d'autres difficultés dues à l'excès de nutriments ont déjà dégradé la qualité de l'eau sur plus de la moitié des côtes des États-Unis ; c'est notamment le cas de la «zone morte» dans la partie Nord du golfe du Mexique. Des difficultés similaires surgissent dans diverses régions du monde.

Même si les éventuelles conséquences néfastes de la fertilisation étaient jugées acceptables, encore faudrait-il s'assurer que de telles mesures compenseraient les réactions des plantes et des océans à un monde plus chaud. En comparant des images obtenues par satellites au début des années 1980 et dans les années 1990, on constate que l'océan a tendance à verdir, mais plusieurs biologistes ont souligné qu'une augmentation de la productivité primaire ne garantit pas un stockage plus important du carbone dans les profondeurs océaniques. Ce pourrait même être l'inverse qui se produit. Selon des simulations numériques, le réchauffement augmenterait la stratification de l'océan, l'eau douce venant de la fonte des glaciers et des icebergs restant au-dessus de l'eau de mer salée, plus dense. Une telle stratification ralentirait la capacité de la pompe biologique à transporter le carbone de la surface vers les profondeurs.

De nouveaux satellites observent quotidiennement le phytoplancton, et les futures expériences de fertilisation à petite échelle nous permettront de mieux comprendre le comportement du phytoplancton. L'idée de développer de grands projets de fertilisation des océans pour modifier le climat reste débattue au sein de la communauté scientifique et parmi les autorités politiques. Pour beaucoup, le bénéfice temporaire des projets de fertilisation ne justifie pas les dégâts, imprévisibles, mais inévitables, que subiront les écosystèmes marins. Aujourd'hui, la société voudrait en appeler au phytoplancton actuel pour l'aider à résoudre un problème lié en partie à la combustion du phytoplancton fossile !

Paul FALKOWSKI est professeur à l'Institut de sciences marines et côtières et au Département de géologie de l'Université Rutgers.

Paul FALKOWSKI et John RAVEN, *Aquatic Photosynthesis, Blackwell Scientific*, 1997.

Paul FALKOWSKI et al., *The Global Carbon Cycle : A Test of Our Knowledge of the Earth as a System*, in *Science*, vol. 290, pp. 291-294, 13 octobre 2000.

P. CHISHOLM, *Stirring times in the Southern Ocean*, in *Nature*, vol. 407, pp. 685-687, 2001.

Z. NEUFELD, P. HAYNES, V. GARÇON et J. SUDRE, *Ocean fertilization experiments may initiate a large scale phytoplancton bloom*, in *Geophysical Research Letters*, n°1129, 10.1029/2001GL013677, 2002.

Ordinateurs asynchrones

IVAN SUTHERLAND • JO EBERGEN

Des puces qui ne sont pas contrôlées par une horloge centrale améliorent les performances des ordinateurs en permettant à chaque circuit de travailler au maximum de ses possibilités.

Quelle est la vitesse de votre ordinateur ? Si son acquisition date de moins de six mois, vous répondriez sans doute : deux gigahertz. Cette fréquence est celle d'une minuscule horloge – il s'agit, comme dans une montre, d'un oscillateur à quartz – contenue dans l'ordinateur qui détermine le rythme de fonctionnement de l'ensemble de la machine. Dans un ordinateur cadencé à un gigahertz, par exemple, le cristal fait «tic» un milliard de fois par seconde, c'est-à-dire que les tâches effectuées par l'ordinateur sont décomposées en opérations élémentaires qui durent chacune un milliardième de seconde. Un transfert de données simple s'effectue en une seule étape, des calculs complexes peuvent en nécessiter de nombreuses, mais, au final, la puissance d'un ordinateur est approximativement proportionnelle à la fréquence de son horloge.

La plupart des ordinateurs modernes, qualifiés de synchrones, utilisent une fréquence unique pour l'ensemble des circuits du microprocesseur. La synchronisation est assurée par un système qui distribue les impulsions de l'oscillateur à quartz à tous les circuits. Si la synchronisation générale des circuits a certains avantages, la distribution d'impulsions de plus en plus rapides présente une difficulté croissante. De surcroît, les transistors actuels traitent les données si vite qu'ils peuvent accomplir plusieurs opérations élémentaires pendant le temps que met une impulsion à être acheminée d'un bout à l'autre de la puce. La conservation de la synchronisation dans un tel circuit nécessite des puces parfaitement conçues (surtout quand elles sont de grande taille), et entraîne une consommation importante d'électricité.

Différentes équipes, dont la nôtre, aux Laboratoires *Sun Microsystems*, cherchent comment s'affranchir du principe de synchronisme. Nous explorons divers systèmes informatiques où chaque élément fonctionnerait à son propre rythme, sans avoir à obéir à une horloge centrale. Ces systèmes sont nommés asynchrones. Chaque élément d'un système asynchrone peut réduire ou augmenter, selon les besoins, sa propre fréquence d'horloge, tout comme un coureur allonge ou raccourcit sa foulée en fonction du terrain. Certains des pionniers de l'ère informatique, tel le mathématicien Alan Turing, ont tenté de concevoir des machines asynchrones dès les années 1950. Les ingénieurs avaient rapidement écarté cette approche au profit des systèmes synchrones, plus simples à développer.

L'informatique asynchrone connaît aujourd'hui une renaissance. Des informaticiens de l'Université de Manchester, de l'Université de Tokyo et de l'Institut de technologie de Californie ont mis au point des microprocesseurs asynchrones. Quelques puces asynchrones sont d'ores et déjà produites en masse à des fins commerciales. À la fin des années 1990, la société d'électronique japonaise *Sharp* a construit sur un concept asynchrone un processeur multimédia – une puce d'édition graphique, vidéo et audio. *Philips Electronics*, de son côté, a produit un microcontrôleur asynchrone pour deux de ses messageries de poche. Des composants asynchrones intégrés dans des systèmes synchrones commencent également à apparaître, tels ceux qu'a conçus notre groupe et qui équipent l'un des nouveaux processeurs de *Sun Microsystems*. Nous sommes persuadés que les systèmes asynchrones se généraliseront à mesure que les informaticiens apprendront à tirer parti de leurs avantages et à simplifier leur construction. Les fabricants de puces asynchrones ont déjà réalisé de belles prouesses techniques, mais le succès commercial n'est pas encore au rendez-vous. Il reste beaucoup de chemin à parcourir avant que toutes les promesses des ordinateurs asynchrones ne se concrétisent.

Contre la montre

Quels sont les avantages des systèmes asynchrones ? En premier lieu, les systèmes asynchrones apportent un gain de vitesse. Dans une puce synchrone, la cadence de l'horloge doit être au moins aussi lente que le plus lent des circuits de la puce. S'il faut un milliardième de seconde à l'un des circuits pour effectuer une opération, la puce ne peut pas fonctionner à plus d'un gigahertz. Même si la plupart des autres circuits de cette puce sont capables de réaliser leurs opérations en un temps plus court, ils doivent attendre le signal de l'horloge pour passer à l'étape logique suivante. À l'inverse, dans un système asynchrone, les éléments autonomes du processeur travaillent à leur rythme : il dispose exactement du temps qui leur est nécessaire pour exécuter une tâche. Ainsi les opérations simples sont effectuées plus rapidement que la moyenne, le gain de temps pouvant être attribué aux opérations complexes, plus longues. Les tâches de calcul commencent dès que les actions préalables ont été réalisées, sans attendre le signal suivant de l'horloge. De cette façon, la vitesse d'un système asynchrone dépend de la durée d'action moyenne, plutôt que de la plus grande durée d'action.

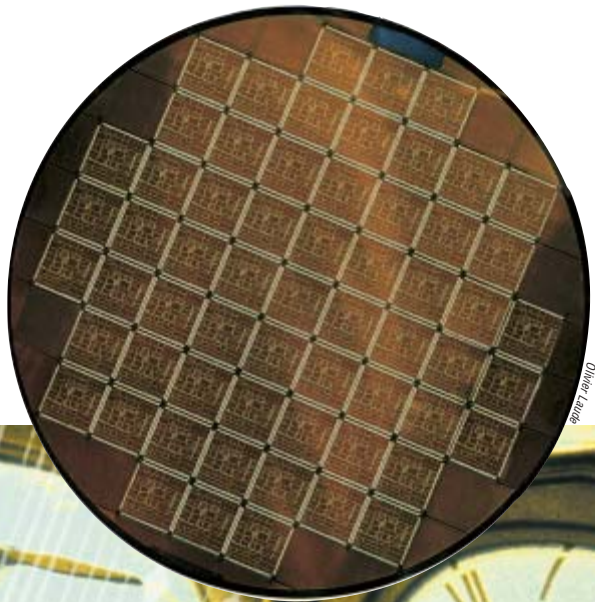
Cependant, la coordination d'actions asynchrones nécessite, elle aussi, du temps et de la place sur les puces. Dans le cas où les contraintes imposées par une coordination locale sont faibles, un système asynchrone peut être plus rapide en moyenne qu'un système synchrone. Ainsi,

l'asynchronisme est particulièrement indiqué pour les puces où les tâches lentes interviennent de façon irrégulière.

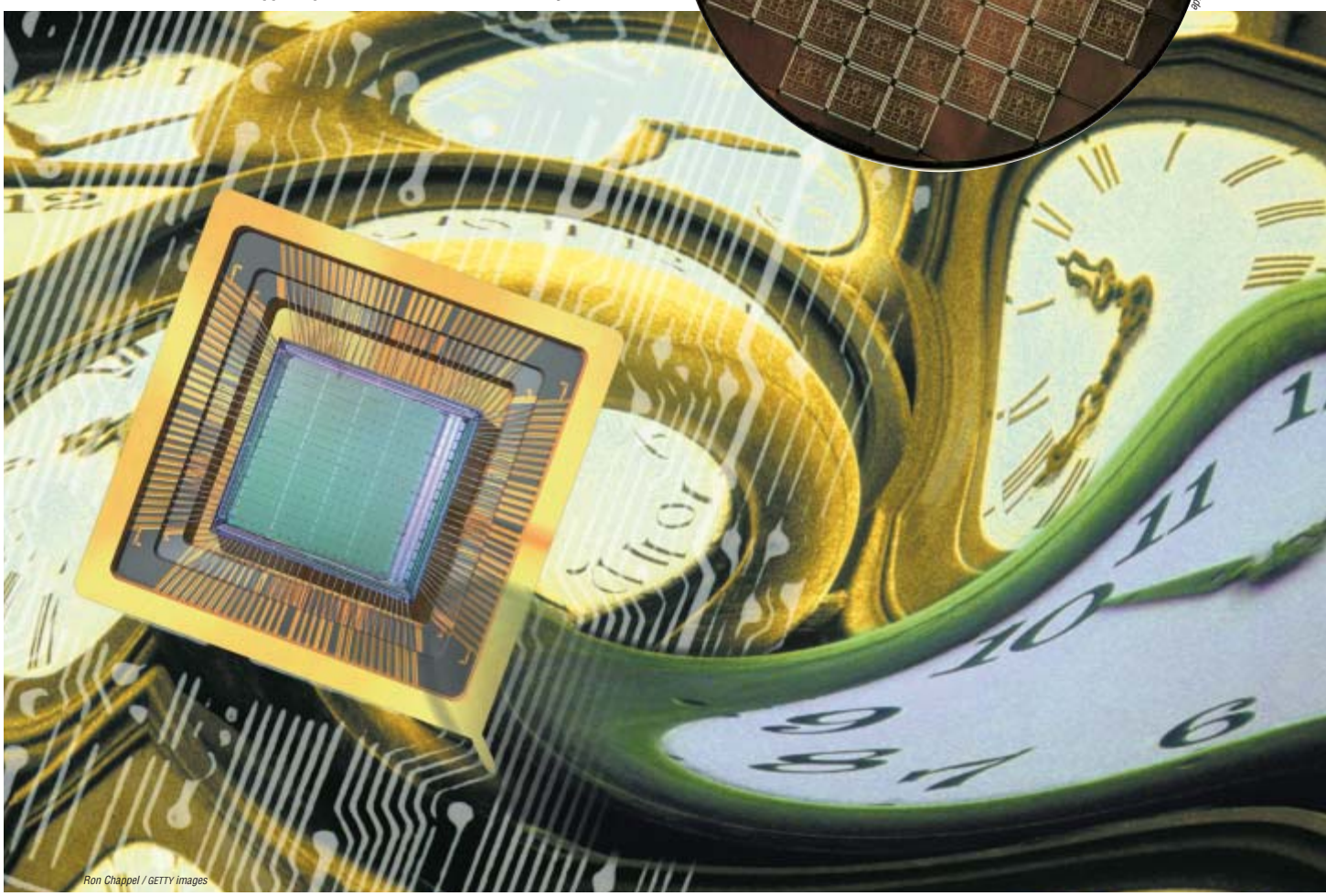
Une conception asynchrone peut également permettre de réduire la consommation électrique des puces. Dans les puces massives actuelles, les circuits qui émettent les signaux d'horloge occupent une bonne partie de la surface : environ 30 pour cent de l'électricité consommée par la puce est allouée à l'horloge et à son système de distribution. De surcroît, comme l'horloge est constamment en activité, elle produit de la chaleur même quand la puce ne fonctionne pas.

Dans les systèmes asynchrones, les parties inactives de la puce ne consomment quasiment rien. Cette caractéristique est particulièrement intéressante dans les équipements fonctionnant sur piles, mais elle peut également contribuer à réduire les coûts de fonctionnement de machines plus volumineuses qui nécessitent d'être refroidies par des dispositifs de ventilation et de climatisation, qui évitent toute surchauffe du système. La quantité d'électricité ainsi économisée dépend du type d'activité de la machine. Les systèmes dont certaines parties sont souvent au repos y gagnent plus que les systèmes en fonctionnement continu. La plupart des ordinateurs comportent des composants inactifs pendant de longues périodes, telle l'unité de calcul en virgule flottante.

Par ailleurs, les systèmes asynchrones produisent moins d'interférences radio que les machines synchrones. Comme les systèmes cadencés fonctionnent à un rythme fixe, ils émettent un fort signal radio correspondant à leur fréquence de fonctionnement et à ses harmoniques (des ondes dont la

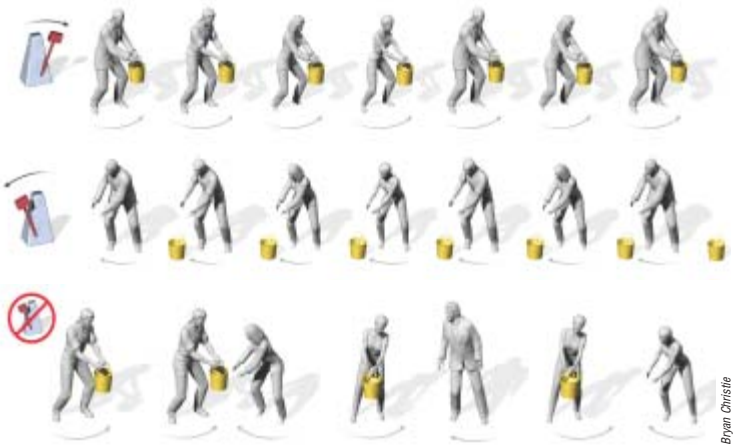


Olivier Laude



Ron Chappel / GETTY images

1. LES PROCESSEURS MODERNES, pour la plupart, sont synchrones : les circuits fonctionnent au même rythme, imposé par une horloge centrale. Les informaticiens construisent désormais des systèmes asynchrones qui traitent des données sans être soumis à la cadence d'une horloge. Les avantages potentiels des circuits asynchrones sont notamment des vitesses de fonctionnement supérieures, une consommation électrique réduite et des interférences radio négligeables. Ci-contre, une tranche de silicium contenant des puces électroniques asynchrones UltraSPARC IIIi développées par les Laboratoires *Sun Microsystems*.



Bryan Christie

2. LES CHAÎNES DE PASSEURS DE SEAUX : on peut utiliser cette métaphore pour décrire le flux des données dans un ordinateur. Un système informatique synchrone est comparable à une chaîne de passeurs où chaque personne reçoit et passe les seaux au rythme du tic-tac d'une horloge. Quand l'horloge fait «tic», chaque passeur avance son seau vers la personne suivante *(en haut)*. Quand l'horloge fait «tac», chaque passeur attrape le seau posé par la personne précédente *(au milieu)*. Au contraire, un système asynchrone ressemble à une chaîne de passeurs ordinaire : chaque personne tenant un seau peut le passer à la suivante dès qu'elle a les mains libres *(en bas)*. Le désordre engendré par cette perte de cohésion complique certes la conception des puces, mais le flux de données traitées augmente : la vitesse de la chaîne n'est plus limitée par le circuit le plus lent.

fréquence est un multiple de la fréquence originelle). De tels signaux peuvent interférer avec les téléphones portables, la télévision et les systèmes de navigation des avions qui emploient les mêmes fréquences. Comme les systèmes asynchrones n'ont pas de cadence fixe, ils répartissent leur énergie de rayonnement sur l'ensemble du spectre radio, de sorte que l'émission dans chacune des fréquences est faible.

La conception asynchrone présente un avantage supplémentaire : elle permet la construction de ponts entre des ordinateurs de fréquences différentes. Les réseaux d'ordinateurs hétérogènes – on les nomme des grappes – qui prolifèrent depuis quelques années, mettent en relation des ordinateurs rapides avec des machines plus lentes, qui, sinon, auraient été mises au rebut. Ces grappes permettent de traiter des problèmes complexes en répartissant les tâches de calcul entre les différentes machines. Un tel système est par essence asynchrone : les divers constituants ont des cadences différentes. Le transfert de données contrôlées par une horloge vers une machine contrôlée par une autre horloge nécessite des ponts asynchrones, car les données peuvent être, au cours des calculs, «désynchronisées» par rapport à l'horloge de l'ordinateur qui reçoit l'information.

En dernier lieu, bien que la conception asynchrone présente des difficultés, elle offre une grande flexibilité. Étant donné que les circuits d'un système asynchrone ne sont pas tenus de suivre tous le même rythme, les concepteurs ont plus de liberté pour choisir quelles parties du système participeront au calcul, et pour décider comment elles interagiront. En outre, le remplacement de n'importe quelle partie par une version plus rapide augmente la vitesse de l'ensemble du système, alors que dans un système cadencé, l'augmentation de la vitesse nécessite généralement la mise à jour de tous les constituants.

Afin de décrire le fonctionnement des systèmes asynchrones, nous utilisons souvent la métaphore d'une chaîne de passeurs de seaux. Un système cadencé est comparable à une chaîne de passeurs de seaux où chaque participant doit passer et recevoir les seaux au rythme d'une horloge. Au premier «tic», chacun pose son seau devant son voisin de gauche, par exemple ; quand l'horloge fait «tac», chacun attrape le seau déposé par le voisin *(voir la figure 2)*. Le rythme de cette chaîne est défini par le temps que met la personne la plus lente de la chaîne à déplacer le seau le plus lourd. Même si les seaux sont légers, tout le monde doit attendre le battement de l'horloge avant de passer le seau suivant.

Coopération locale

Dans une chaîne de passeurs de seaux asynchrones, c'est la coopération locale qui détermine le rythme, et non une horloge commune. Chaque personne tenant un seau peut le passer à la suivante dès que cette dernière a les mains libres. Quand les seaux sont légers, ils font rapidement le trajet d'un bout à l'autre de la chaîne. De surcroît, quand il n'y a pas d'eau à transporter, tout le monde peut se reposer en attendant. Si une personne lente ralentit l'ensemble de la chaîne, on la remplace et l'ensemble retrouve des performances optimales.

Dans les ordinateurs, les chaînes de passeurs de seaux sont nommées des «pipelines». Un pipeline ordinaire comporte environ une demi-douzaine de relais, ayant chacun un rôle analogue à celui d'un des passeurs de seaux. Au lieu de manipuler des seaux d'eau, chaque relais effectue l'une des tâches nécessaires à l'exécution de l'instruction. Par exemple, un processeur exécutant l'instruction $C = A + B$ doit aller chercher la signification de l'instruction «+» dans la mémoire, récupérer les nombres correspondant à A et à B, en faire la somme et stocker le résultat dans la mémoire C.

Un pipeline cadencé effectue ces opérations à un rythme ne dépendant ni de l'opération à effectuer ni de la taille des nombres. Additionner *1* et *1* demande autant de temps qu'additionner deux nombres de 30 chiffres. En revanche, dans un pipeline asynchrone, la durée de chaque action dépend de l'opération effectuée, de la taille des nombres ainsi que de l'emplacement des données dans la mémoire (de la même manière que dans la chaîne de passeurs, la quantité d'eau dans un seau peut influer sur le temps qu'il faut pour le passer). Sans horloge pour réguler ses actions, un circuit asynchrone recourt à des circuits de coordination locale. Ces circuits échangent des signaux de fin d'exécution afin de s'assurer qu'à chaque étape les actions ne commencent que lorsque le circuit dispose de toutes les données nécessaires. Les deux circuits de coordination les plus importants sont nommés le *Rendez-vous* et l'*Arbitre*.

Un élément *Rendez-vous* indique le moment où toutes les données requises par un circuit pour effectuer une tâche sont parvenues à destination. Par exemple, un circuit de division arithmétique doit disposer à la fois du dividende (par exemple 16) et du diviseur (par exemple 2) avant de pouvoir diviser l'un par l'autre (et donner le résultat 8).

Un type de circuit *Rendez-vous* est nommé élément C de Muller, du nom de David Muller, qui fut professeur à l'Université de l'Illinois. Un élément C de Muller est un circuit logique avec deux entrées et une sortie *(voir l'encadré page 39)*. Quand les deux entrées d'un élément C de Muller sont VRAI, sa sortie devient VRAI. Quand les deux entrées

sont FAUX, la sortie devient FAUX. Dans les autres cas, la sortie ne change pas. Les éléments C de Muller servent notamment à contrôler le flux des données. Par un agencement astucieux de plusieurs d'entre eux, ils permettent de gérer l'avancée de données le long d'une chaîne de passeurs électronique sans le besoin d'une horloge.

Notre groupe de recherche a récemment introduit un nouveau type de circuit *Rendez-vous*, nommé GasP *(voir l'encadré page 41)*. Le GasP est dérivé d'une famille plus ancienne de circuits mise au point par Charles Molnar, qui travailla aux Laboratoires *Sun Microsystems*. Ch. Molnar avait baptisé sa création asP* (pour protocole de pulsation asynchrone et symétrique). Nous y avons ajouté le *G*, pour obtenir l'onomatopée *(gasp)* qui, en anglais, traduit la surprise, surprise créée ici par la rapidité des nouveaux circuits. Nous avons établi que les modules GasP ont une rapidité et une efficacité énergétique aussi bonnes que les éléments C de Muller, qu'ils s'assemblent mieux avec les circuits ordinaires et qu'ils sont beaucoup plus faciles à utiliser dans des circuits complexes.

Les circuits *Arbitre* remplissent, quant à eux, une autre fonction essentielle pour les ordinateurs asynchrones. Un *Arbitre*, tel un agent de la circulation posté à une intersection qui décide quel véhicule passera le premier, orchestre le traitement des données. S'il n'y a qu'une requête, l'*Arbitre* donne immédiatement le feu vert à l'action correspondante, et met en attente une éventuelle seconde requête jusqu'à ce que la première soit terminée. Quand l'*Arbitre* reçoit deux requêtes quasi simultanément, il doit décider à laquelle accorder la priorité. Par exemple, quand deux processeurs réclament l'accès à une mémoire au même moment, le rôle de l'*Arbitre* est de n'accorder l'accès qu'à un processeur à la fois. L'*Arbitre* garantit que deux actions ne se déroulent jamais en même temps.

L'âne de Buridan

Bien que les circuits *Arbitre* n'accèdent jamais à plus d'une requête à la fois, il n'existe aucun *Arbitre* qui tranche entre les diverses requêtes en un temps limité. Les *Arbitre* actuels prennent leurs décisions très rapidement en moyenne, généralement en quelques picosecondes (un millième de milliardième de seconde). Cependant, quand les demandes sont très rapprochées, les circuits mettent parfois deux fois plus longtemps à trancher, voire, dans quelques rares cas, dix fois plus.

Les difficultés auxquelles est confronté l'*Arbitre* pour prendre ses décisions rappellent celles de l'âne de Buridan. Selon cette parabole attribuée à Buridan, philosophe français du XIV^{e} siècle, un âne affamé et assoiffé, se trouvant à égale distance d'un picotin d'avoine et d'un seau d'eau, mourrait de faim et de soif parce qu'il serait incapable de choisir entre l'un ou l'autre. Nous sommes confrontés tous les jours à ce type de dilemmes, par exemple lorsque deux

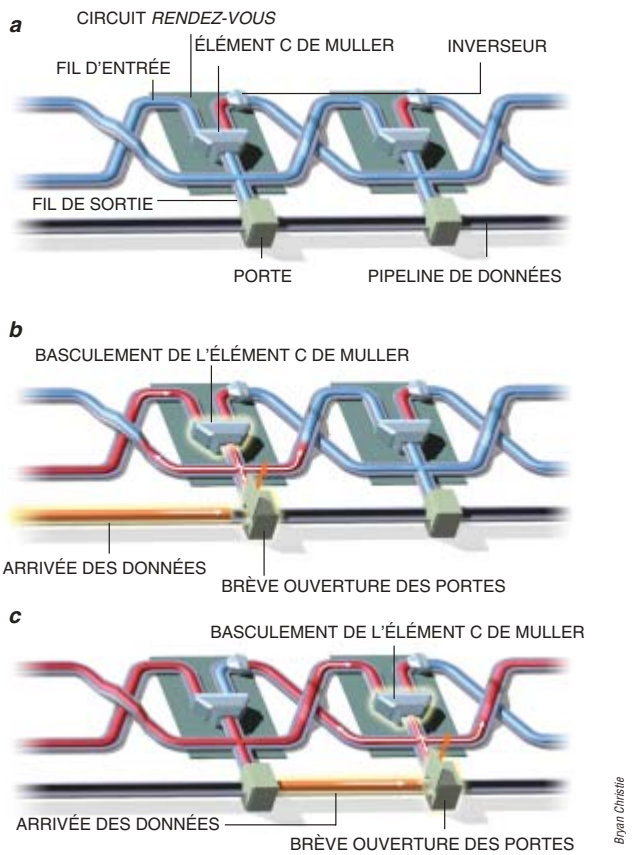
Fonctionnement d'un circuit *Rendez-vous*

Les circuits *Rendez-vous* gèrent le flux des données à travers des circuits sans horloge centrale. Ici un pipeline électronique est contrôlé par une chaîne de circuits *Rendez-vous*, nommés éléments C de Muller. Chacun de ces éléments autorise le passage des données *(en orange)* seulement quand l'étage précédent est «plein» (les données sont prêtes à poursuivre leur chemin) et que l'étage suivant est «vide».

Chaque élément C de Muller a deux entrées et une sortie. La sortie devient FAUX quand les deux entrées sont FAUX, et redevient VRAI quand les deux entrées sont VRAI (dans le diagramme, les signaux VRAI sont en bleu, et les signaux FAUX en rouge). L'inverseur fait en sorte que les entrées initiales diffèrent de sorte qu'au départ, tous les étages sont vides.

Supposons que l'entrée de gauche soit VRAI et celle de droite FAUX *(a)*

Un changement de l'entrée de gauche de VRAI à FAUX *(b)* indique que l'étage de gauche est plein, c'est-à-dire que des données sont arrivées. Comme les deux entrées de l'élément sont maintenant toutes deux FAUX, l'élément bascule sa sortie sur FAUX. Cette modification a trois conséquences : des données sont transmises dans le pipeline pour poursuivre leur trajet (les portes s'ouvrent), un signal FAUX est renvoyé vers l'élément C de Muller précédent pour que l'étage soit vidé et un signal FAUX est expédié vers l'élément C de Muller suivant, de sorte que l'étage suivant est plein *(c)*.



personnes arrivant ensemble devant une porte marquent une pause avant de décider qui passera en premier. Dans ces situations, l'ordre de passage n'a pas d'importance, la seule difficulté consiste à prendre la décision.

Le rôle de l'*Arbitre* est précisément de prendre cette décision. Un *Arbitre* a deux états stables correspondant aux deux choix. On peut se représenter ces deux états, l'un comme la France, l'autre comme l'Espagne. Chaque requête exprimée auprès d'un *Arbitre* pousse le circuit vers l'un ou l'autre de ces états stables, comme un grêlon, tombé au sommet des Pyrénées, pourrait rouler aussi bien vers la France que vers l'Espagne. Entre les deux états stables, il doit néanmoins y avoir une ligne métastable, équivalente à la ligne de crête des Pyrénées. Si le grêlon tombe précisément sur cette ligne de partage, il peut rester en équilibre quelques instants sur la crête avant de basculer vers l'Espagne ou

Processeurs asynchrones et relativité

Les questions soulevées par I. Sutherland et J. Ebergen, *a priori* de nature technologique, sont aussi le reflet de questions fondamentales sur le temps. Pour connaître le temps, il ne suffit malheureusement pas de le mesurer. La difficulté rencontrée pour synchroniser tous les calculs au sein d'une puce électronique résulte aussi de ce fait : le temps n'existe pas sous la forme d'une grandeur physique unique qu'il suffirait de mesurer à l'endroit où l'on se trouve. C'est cette constatation qui a conduit à l'introduction de la théorie de la relativité. Certes, le temps est obtenu à l'aide des indications des horloges, mais ce n'est pas suffisant. Comme le dit Einstein en 1905 : «Il nous faut garder à l'esprit que tous les jugements dans lesquels le temps joue un rôle sont toujours des événements simultanés. Lorsque, par exemple, je dis : «Tel train arrive ici à 7 heures», cela signifie à peu près : «Le passage de la petite aiguille de ma montre sur le 7 et l'arrivée du train sont des événements simultanés.» Einstein ajoute plus loin : «Cette définition ne suffit cependant plus dès lors qu'il s'agit de relier temporellement des séries d'événements qui se produisent à des endroits différents.» En fait, pour disposer d'un temps coordonné en des lieux différents, il faut synchroniser les horloges qui s'y trouvent.

Pour ce faire, il suffit d'échanger des signaux. Malheureusement, il existe une vitesse maximale à laquelle peuvent se propager toutes formes de signaux, fixée par la vitesse de la lumière dans le vide, de l'ordre de 300 000 kilomètres par seconde. Il en résulte que la simultanéité de deux événements se produisant en des lieux différents ne peut plus être définie de façon absolue, mais prend un caractère conventionnel. De ce point de vue, les concepteurs des microprocesseurs actuels sont confrontés au même problème que les ingénieurs de la fin du XIX^{e} siècle : ils voulaient accorder les horaires des trains, mais se heurtaient à la difficulté de synchroniser les horloges des gares sur un territoire large de plusieurs centaines de kilomètres. Einstein, alors employé du bureau des brevets à Berne, voyait passer des demandes de brevets pour des systèmes de synchronisation électromagnétique des horloges. La théorie de la relativité qui en a résulté permet aujourd'hui de disposer, sur toute la Terre, d'une référence de temps, et ainsi de systèmes de repérage, tel le GPS.

Le problème de synchronisation se pose également dans les processeurs, où les signaux électriques n'ont pas le temps de se propager d'un bout à l'autre de la puce entre deux bips de l'horloge. L'horloge d'un ordinateur à 2 gigahertz a une période de 0,5 nanoseconde (un demi milliardième de seconde), c'est-à-dire qu'en une période, les signaux au sein des puces ne parcourent que quelques millimètres, alors que la largeur des puces est couramment de l'ordre de deux centimètres. Tout comme la simultanéité, l'ordre dans le temps de deux événements n'est clairement défini que dans le cas où les événements se produisent au même endroit. Pour des événements *A* et *B* distants dans l'espace, l'ordre temporel «*A* avant *B*» n'est définissable que si un signal peut voyager de *A* vers *B*, car dans ce cas la réception du signal est toujours postérieure à son émission. On dit encore que *A* et *B* sont en relation causale. Comme les signaux se propagent à vitesse finie, tous les couples d'événements ne satisfont pas nécessairement cette condition. Les couples d'événements qui ne peuvent pas être reliés de cette façon ne sont pas ordonnés dans le temps, et ne sont donc pas liés causalement.

Ainsi, au sein d'une puce, il n'existe pas d'ordre temporel total pour tous les événements qui s'y produisent. Dans les circuits synchrones, l'horloge maîtresse distribue un temps périodique à tous les endroits de la puce, selon un arbre dont toutes les branches sont parcourues en des temps approximativement égaux. Les événements du calcul sont ainsi sélectionnés de façon telle que leur ordre temporel et, par conséquent, leur relation causale sont définis sans ambiguïté au niveau global.

En revanche, dans les circuits asynchrones, l'activité d'un opérateur logique est déclenchée par l'arrivée de données sur ses entrées. Cela ne permet de définir un ordre temporel que pour une partie des événements du calcul seulement. Un calcul dépend en général de l'ordre global dans lequel les opérations s'effectuent. Or, dans certains cas, l'ordre relatif où arrivent les données sur différents opérateurs peut dépendre des durées de propagation. Afin de maîtriser le calcul effectué, une méthode consiste à restreindre les calculs possibles et à concevoir la puce de façon telle que ne subsistent que des calculs indépendants des retards dus aux propagations (on qualifie alors le calcul de «DI», c'est-à-dire insensible au délai). Certes, les performances en vitesse dépendent de ces retards, mais le résultat du calcul, lui, n'en dépend pas. La condition DI conduit à un ensemble de contraintes logiques, dites règles de Udding, qui restreignent le comportement des opérateurs logiques élémentaires (comme le *Rendez-vous* ou l'*Arbitre*). En fait, ces règles traduisent de façon logique l'existence ou l'absence d'une relation causale physique entre les événements du calcul.

La théorie de la relativité et les circuits asynchrones semblent deux domaines bien éloignés. Pourtant, ils se rejoignent sur une même question centrale : celle de l'existence d'une référence de temps et d'un ordre temporel. Cette convergence permet de mieux cerner les problèmes liés au calcul asynchrone, et d'envisager de concevoir des puces encore plus performantes. En retour, la physique doit aussi y gagner : la forte miniaturisation des circuits logiques (l'épaisseur d'une grille de transistor se rapproche des distances interatomiques) ne rend-elle pas indispensable de maîtriser l'espace-temps jusqu'au domaine microscopique ?

Philippe MATHERAT, Laboratoire traitement et communication de l'information, CNRS, École nationale supérieure des télécommunications et Marc-Thierry JAEKEL, Laboratoire de physique théorique, CNRS, École normale supérieure

vers la France. De même, si deux requêtes arrivent sur un *Arbitre* à quelques picosecondes d'écart, le circuit peut faire une pause dans son état métastable avant d'atteindre un de ses états stables et de résoudre le dilemme.

Les concepteurs débutants d'*Arbitre* cherchent souvent à éviter ces délais prolongés occasionnels en développant des circuits complexes. Ils commettent fréquemment l'erreur d'introduire un circuit qui pousse l'*Arbitre* vers une décision particulière si jamais le circuit se trouve dans l'état métastable. Cela revient à dresser l'âne de Buridan à aller vers la gauche (ou vers la droite) lorsqu'il est confronté à un choix difficile. Cependant un tel dressage a pour seul effet de présenter à l'âne trois choix plutôt que deux : aller à gauche, aller à droite, ou s'apercevoir qu'il y a un choix difficile et aller à gauche. Même un âne dressé mourra de faim s'il est incapable de trancher entre les deux dernières possibilités. En d'autres termes, pour reprendre la métaphore géographique, vous pouvez déplacer la ligne de crêtes des Pyrénées, mais vous ne pouvez pas vous en débarrasser. Bien qu'il n'existe aucun moyen d'éliminer la métastabilité, des circuits *Arbitre* simples et bien conçus peuvent garantir des pauses très courtes dans tous les cas. Typiquement, les *Arbitre* actuels ont des pauses normales de 100 picosecondes, et, exceptionnellement (une fois sur dix heures de fonctionnement), de 400 picosecondes. La probabilité qu'une pause survienne diminue exponentiellement avec la durée de la pause : une pause de 800 picosecondes se produit moins d'une fois pour un milliard d'années de fonctionnement.

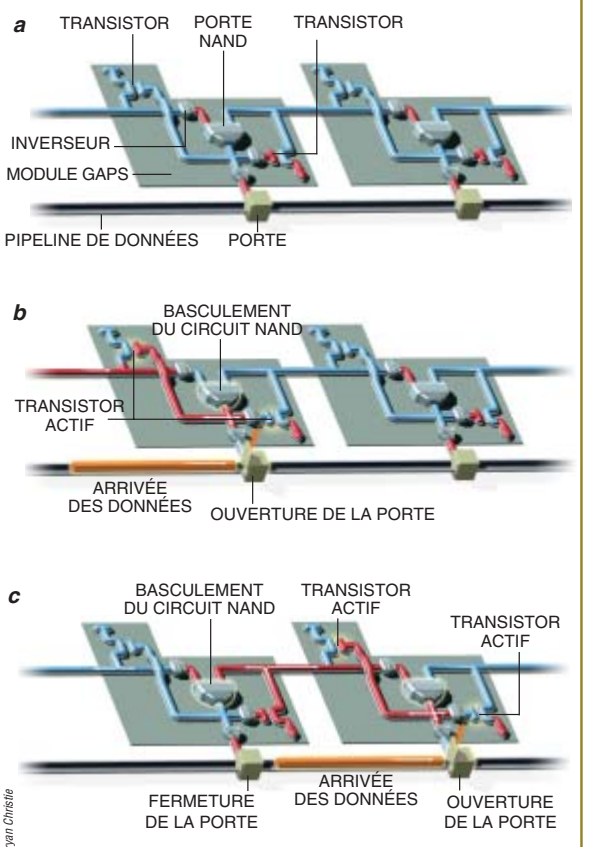
Une nécessaire rapidité

Notre équipe étudie surtout le développement de systèmes asynchrones rapides. Nous avons observé que la simplicité est souvent garante de rapidité. Nous voulions construire un pipeline à double sens de circulation, comportant deux courants de données opposés (comme deux chaînes de passeurs de seaux parallèles, mais acheminant l'eau en sens contraire). Nous voulions que les données des deux flux interagissent à chaque étape ; la principale difficulté était de faire en sorte que chaque élément de données allant dans un sens interagisse avec chaque élément de données allant dans l'autre sens. L'arbitrage s'est révélé un facteur essentiel. À chaque jonction entre étapes successives, un circuit *Arbitre* autorisait le passage d'un seul élément à la fois. Nous avons beaucoup appris grâce à ce projet à propos de la coordination et de l'arbitrage. Nous avons ensuite construit des puces expérimentales pour vérifier la validité et la fiabilité de nos circuits *Arbitre*.

Plus récemment, nous nous sommes intéressés à une structure nommée FLEET. Ce nom fait référence à la rapidité de la structure, mais aussi à l'ensemble des composants qui la constituent, autant de «navires» de la flotte (*fleet*, en anglais) voguant à leur propre allure. Chaque navire effectue sa tâche en même temps que les autres ; le système FLEET les contrôle en faisant circuler des données parmi eux par l'intermédiaire d'un réseau de commutateurs asynchrones. Les recherches menées pour le projet FLEET nous ont conduits à de nombreuses découvertes. Nous voulions de la vitesse, et c'est pourquoi nous avons créé les circuits GasP de base. Nous voulions aiguiller des données d'un pipeline à un autre, comme des voitures aux jonctions autoroutières, et cela nous a amenés à concevoir une famille de circuits GasP plus vaste, agissant comme un système dense de voies rapides. Ces circuits peuvent transporter les données environ deux fois plus vite que les systèmes cadencés. Pour évaluer la vitesse de nos réseaux de commutateurs, nous construisons souvent sur nos puces expérimentales des circuits en anneau où des données tournent. Nous mesurons le temps qu'il faut à une donnée pour effectuer un tour complet, ce qui est bien plus simple que de mesurer le temps – notablement plus court – nécessaire aux données pour avancer d'une étape.

Fonctionnement d'un module GasP

Les modules GasP servent également d'éléments *Rendez-vous* pour un pipeline asynchrone de données. Au centre de chaque module se trouve un circuit NAND, qui produit une sortie FAUX si les entrées sont toutes deux VRAI. Sinon le circuit NAND produit une sortie VRAI. Au départ, tous les signaux dans les fils de connexion des modules sont VRAI *(en bleu, a)*. L'arrivée de données dans le pipeline *(b)* change le signal d'arrivée en FAUX *(en rouge)*. Un inverseur bascule le signal sur VRAI et l'envoie au circuit NAND, qui change sa sortie en FAUX. Ceci entraîne l'ouverture de la porte, permettant aux données d'avancer dans le pipeline. La sortie FAUX ramène également le fil d'entrée à son état VRAI (c'est-à-dire vide) par l'intermédiaire d'un transistor, et met le fil de sortie dans l'état FAUX (c'est-à-dire plein) par le biais d'un inverseur et d'un transistor. La sortie de la porte NAND redevient alors VRAI *(c)*, ce qui ferme la porte. Pendant ce temps, le signal FAUX du fil de sortie déclenche le même processus dans le module GasP suivant.



La puce processeur Ultra PARC IIIi *(voir la figure 1)* contient des circuits asynchrones situés au carrefour de puces de mémoire. Ces circuits sont le moyen le plus simple et le plus rapide pour compenser les décalages des temps d'arrivée des signaux issus de puces de mémoire situées à différentes distances du processeur. Les concepteurs de *Sun Microsystems* sont convaincus que l'asynchronisme présente des avantages importants par rapport aux systèmes cadencés, et que d'ores et déjà de nombreux circuits asynchrones sont réalisables. À mesure que la confiance gagnera d'autres concepteurs, le nombre de produits comportant des composants asynchrones augmentera, apportant des améliorations en termes de vitesse, de flexibilité, de consommation électrique et de réduction des interférences radio. Par ailleurs, les microprocesseurs asynchrones sont compatibles avec leurs homologues cadencés, et peuvent se connecter à des machines périphériques sans programmes ou circuits d'interface spéciaux.

D'autres équipes travaillent sur les circuits asynchrones. Un groupe de la Société *Philips*, aux Pays-Bas, a développé un correcteur d'erreurs asynchrone pour un lecteur de bande magnétique numérique compact, ainsi qu'une version asynchrone d'un microcontrôleur utilisé dans les appareils portables. Ce microcontrôleur asynchrone a été également incorporé à des messageries de poches commercialisées par *Philips*, augmentant l'autonomie électrique des appareils. La réduction des interférences radio a permis l'inclusion d'un ordinateur et d'un récepteur radio dans des dispositifs minuscules. Le langage utilisé est le Tangram.

Bien que la liberté permise dans l'architecture des systèmes asynchrones présente des avantages notables, elle est aussi source de difficultés. Du fait que chaque partie décide de son propre rythme, cette allure varie par moment au sein d'un même système et d'un système à un autre. Si plusieurs actions sont simultanées, elles peuvent se terminer dans des ordres très variés. L'énumération de toutes les séquences possibles d'actions dans une puce asynchrone complexe est aussi difficile que de prédire la suite des activités des enfants dans une cour d'école. Ce dilemme est le problème d'explosion du nombre des états. Les concepteurs de puces peuvent-ils mettre de l'ordre dans le chaos potentiel des actions simultanées ?

L'heure des défis

Des informaticiens cherchent à élaborer des théories permettant de résoudre ce problème. Les concepteurs n'ont pas à se soucier de toutes les séquences d'actions possibles à condition de poser certaines limites au comportement de communication de chaque circuit. Pour prolonger la métaphore de la cour d'école, un enseignant assurera la sécurité des enfants lors des jeux en apprenant à chacun d'eux à éviter le danger.

Nous sommes aussi confrontés au manque d'outils de conception, de méthodes d'expérimentation reconnues, et d'un enseignement généralisé de la conception asynchrone. Même si aujourd'hui notre communauté de recherche s'agrandit et progresse, l'investissement total dans le domaine de l'informatique asynchrone est dérisoire par rapport aux sommes consacrées aux ordinateurs cadencés. Néanmoins, nous sommes persuadés que la course effrénée aux performances et la complexité croissante des circuits intégrés amèneront les concepteurs à se tourner vers les techniques asynchrones. Nous ne savons pas encore si les systèmes asynchrones seront davantage promus par les grands groupes d'électronique et d'informatique, ou plutôt par de «jeunes pousses» motivées par le développement d'idées nouvelles, mais il est certain que la tendance est irréversible, et que la conception asynchrone s'imposera dans les décennies à venir. Il n'y aura bientôt plus de réponse simple à la question : quelle est la vitesse de votre ordinateur ?

Ivan SUTHERLAND est vice-président des Laboratoires *Sun Microsystems*, et y dirige le Groupe de conception asynchrone, dont fait partie Jo EBERGEN.

Asynchronous Circuits and Systems, numéro spécial des *Proceedings of the IEEE*, février 1999.

Principles of Asynchronous Circuit Design : A Systems Perspective, sous la direction de Jens Sparsø et Steve Furber, Kluwer Academic Publisher, 2001.

P. MATHERAT et M.-T. JAEKEL, *Concurrent Computing Machines and Physical Space-time*, in *Mathematical Structures in Computing Science*, numéro spécial *The Difference Between Concurrent and Sequential Computation*, à paraître.