

La *softwarization* des réseaux et son impact technico-économique

Par Emmanuel PUIG
Amarisoft

La *softwarization*, ou migration des fonctions informatiques du matériel vers le logiciel, marque une évolution technologique clé où les CPU de plus en plus puissants permettent des solutions *software* autrefois impossibles. Cette transition, visible dans des domaines variés comme les jeux vidéo et les télécoms, est motivée par des avantages économiques et opérationnels tels que la flexibilité, la réduction des coûts, et l'accès à une gamme étendue d'outils et de ressources *open source*. Dans les réseaux télécoms, bien que les cœurs de réseau soient déjà largement « *softwarisés* », l'accès radio reste majoritairement *hardware*, mais la tendance change avec l'essor des réseaux privés et des technologies comme la 5G, favorisant des solutions plus flexibles et adaptées aux petits volumes. Les entreprises qui adoptent cette transformation au moment opportun peuvent gagner un avantage concurrentiel significatif.

LA *SOFTWARIZATION*, UNE PREUVE DE MATURITÉ

Le débat entre le *software* et le *hardware* est un débat presque aussi ancien que l'informatique. Si la distinction entre le logiciel et le matériel informatique est discutable et que sa frontière est parfois floue, nous nous placerons dans le cadre où une implémentation *software* sera du code destiné à être exécuté sur un CPU (Processeur à Usage Général) et une implémentation *hardware* sera sur du silicium dédié à la tâche.

Ce choix a toujours été une affaire de compromis, donc soumis aux conditions technologiques à un moment donné. Cela implique par conséquent que ce choix peut être remis en cause avec l'évolution de ces conditions.

Lorsque l'on s'intéresse à l'évolution de ces conditions, on remarque que pour une nouvelle technologie, le premier choix est bien souvent le choix *hardware* pour la simple raison que la puissance de calcul disponible sur les CPUs est largement insuffisante et qu'il est donc fonctionnellement impossible de faire avec.

La loi de Moore aidant, il arrive un moment où la capacité des CPUs rend possible une implémentation *software*.

À ce stade, le facteur économique penche en général largement vers la solution *hardware* : les machines à base de CPUs capables de fournir le même service étant en général beaucoup plus chères.

Avec un peu plus de temps, ce frein finit par ne plus en être un et d'autres critères peuvent alors rentrer en jeu. Ainsi, les avantages du développement *software* finissent par l'emporter sur ceux du *hardware*. La contamination se fait par l'entrée de gamme et monte au fur et à mesure que les CPU rattrapent leur retard.

Les exemples de ce phénomène de fond ne manquent pas : dans le jeu vidéo, on est passé des bornes d'arcades, assemblage de puces dédiées aux consoles d'aujourd'hui qui ne sont plus finalement que des PCs. De la même manière, le monde de l'audio vidéo est passé d'encodeur et de décodeurs dédiés à des suites logicielles représentées par les très français FFmpeg et VLC.

Dans le domaine des télécoms, l'exemple le plus marquant a été l'avènement du logiciel Asterisk reléguant de gros et dispendieux équipements au passé.

Bien évidemment, pour qu'une telle transition ait lieu, il est indispensable que les solutions *software* soient fonctionnelles (sinon aucun débat n'est possible) et aient un niveau de maturité suffisant, auquel cas l'adoption d'une approche *software* devient une vraie question.

On notera que cette description doit bien évidemment être soumise aux subtilités de chaque technologie et qu'en pratique, il n'y a pas de critère absolu.

LES AVANTAGES DE L'APPROCHE *SOFTWARE*

Pourquoi vouloir adopter une approche *software* plutôt qu'*hardware* ?

Il y a d'abord les promesses, réelles et parfaitement connues et documentées que sont les possibilités de virtualiser, d'automatiser simplement, de désagréger, de mutualiser dans des *data centers*, d'offrir de nouveaux *business models*...

Ici, nous resterons pragmatiques et opérationnels en nous focalisant sur l'aspect coût du cycle de vie d'un produit, et en raisonnant toutes choses égales par ailleurs, sans avoir à mettre dans la balance des telles promesses, nécessitant des modifications importantes et débordant l'aspect technique.

Dans le développement d'un nouveau produit, on distinguera : les coûts de développement initiaux, principalement de R&D, les coûts de production du produit (usine, logistique...), ceux de son exploitation (maintenance, opérations) et ceux liés à son évolution (nouvelles fonctionnalités, améliorations).

Toutes ces phases vont bénéficier de la *softwarization*, en les accélérant, les simplifiant, les rendant moins dépendantes...

Comme dans tout processus d'ingénierie, dans une démarche empirique, il faut itérer un certain nombre de fois. Chaque itération sera coûteuse en temps et en ressources. Par la nature même du *software*, une itération peut être plus rapide de plusieurs ordres de grandeur : une compilation sera toujours plus rapide que produire un prototype. À l'heure de l'agilité, cela est fortement appréciable.

Dans un monde où il y a une tension sur la disponibilité d'ingénieurs qualifiés, les ressources disponibles seront plus accessibles en *software*, car plus génériques et même s'il sera toujours nécessaire d'avoir une connaissance du domaine, une expertise *software* sera plus facilement adaptable.

Travailler sur des CPUs donne accès à tous les outils possibles et imaginables disponibles en *open source* et permettant d'accroître la productivité et la qualité des produits. Parlez-en à vos équipes, vous pourriez voir des étoiles dans leurs yeux.

Utiliser des composants standards donne accès à une gamme incroyablement étendue de fournisseurs, permettant de réduire les coûts, la dépendance et les possibilités. On appréciera la possibilité de fonctionner en temps non réel en n'étant plus tributaire d'une horloge physique par exemple.

Les mises à jour *hardware* étant coûteuses (rétrofit) contrairement au *software*, plus cette dernière sera prépondérante, plus les possibilités seront importantes. On appréciera par exemple la possibilité de répondre aux problèmes de sécurité.

Cette liste n'est évidemment pas exhaustive, et le calcul économique variera en fonction des situations, et si les avantages ne sont pas suffisants, on restera sur une approche *hardware*, le propos n'étant pas de dire que le *software* est meilleur dans l'absolu. Des notions de performance ou énergétique pouvant être dans certains cas rédhibitoires.

Cela étant dit, il faut bien comprendre qu'au-delà de toutes ces considérations, penser en termes de *software* impose un changement de paradigme :

- dans l'approche *hardware*, les spécifications du produit doivent être définies *a priori* et fixeront un cadre indépassable. Une fois celui-ci planté, les fonctionnalités devront rentrer dans ce cadre ;
- lorsque l'on adopte une approche entièrement *software*, ce cadre saute dans la mesure où le matériel acquiert une forme de flexibilité. Celui-ci peut être modifié *a posteriori*, décliné en différentes versions, mutualisé... Cela est particulièrement prégnant dès que l'on veut bénéficier des économies d'échelle ou pouvoir décliner simplement un produit en gamme.

En définitive, tôt ou tard, l'équilibre finira par pencher en faveur du *software*. La question est donc quand ?

LA MIGRATION, UN PROCESSUS LENT MAIS INEXORABLE

La migration du *hardware* vers le *software* ne se fait que très rarement en une seule fois mais découle plutôt du glissement progressif des différentes fonctions d'un produit.

Nous sommes très loin de l'époque où le *hardware* était entièrement dédié à la tâche à réaliser. Aujourd'hui, la grande majorité des produits repose sur des SoC¹ implémentant des CPUs. Ainsi, de plus en plus de fonctions terminent sur le CPU et ne sont plus réalisées par des macros *hardware*.

Lentement, la balance entre les CPUs et le *hardware* spécialisé se déséquilibre en faveur du premier.

Et dans certains cas, tout peut être fait sur le CPU.

SOFTWAREZATION DES RÉSEAUX TÉLÉCOMS

Ce phénomène de fond est à l'œuvre dans le domaine des réseaux filaires depuis plusieurs décennies. Le transport de l'information par paquets a remplacé la commutation de circuit, l'écrasante majorité des fonctions sont l'œuvre de logiciels tournant sur des serveurs Linux dans des data center, et même les composants les plus critiques en termes de performance, comme les *switchs*, exécutent une grande partie de leurs fonctions sur des CPUs. Le *software* a fini par totalement envahir ce secteur et relègue petit à petit le *hardware* à des cas bien spécifiques² [1].

¹ *System on a chip* : il s'agit de puces intégrées, dont les versions récentes embarquent le CPU, le GPU, la puce dédiée au réseau, au wifi et au bluetooth, la RAM et la mémoire de stockage, sur une seule et même puce.

² <https://theses.hal.science/tel-03752344/>

QUEL AVENIR POUR LA *SOFTWARE*IZATION DES RÉSEAUX TÉLÉCOMS ?

Comme expliqué précédemment, le choix du *software* est circonstanciel, il impose donc par essence, afin d'être au plus près de l'innovation, d'être capable de mettre à jour les éléments ayant abouti à ce choix afin de saisir le moment où la balance s'inverse. Et cela ne dépend pas seulement de facteurs techniques, une grande part du processus décisionnel est dépendante du facteur humain, de ses biais, de ses croyances. Il est donc particulièrement important dans ce contexte de ne pas seulement connaître les choix mais de ne jamais perdre de vue les conditions de ce choix, sous peine de croire en cette fameuse phrase faussement attribuée à Mark Twain : « ils ne savaient pas que c'était impossible, alors ils l'ont fait ».

Étant donné les avantages du *soft*, il est clair que les acteurs qui franchissent le pas au moment opportun, c'est-à-dire au moment où l'inéquation s'inverse, en tirent un avantage certain, car non seulement ils seront présents plus tôt, mais ils bénéficieront de l'accélération inhérente au développement *software*.

Il est donc très important pour toute organisation de fournir une veille et même des expérimentations, appelons cela de la recherche, en permanence. Les équipes techniques doivent être challengées et sorties de leur zone de confort afin d'être prêtes à anticiper le mouvement plutôt que de le suivre.

Dans le cadre des réseaux télécoms, et en particulier mobile, les cœurs de réseau sont déjà fortement « *softwarisé* ». En témoigne les architectures 3GPP qui utilisent de manière intensive la désagrégation en éclatant les différents éléments fonctionnels en autant de machines et de logiciels.

Il reste cependant un bastion encore largement dominé par le *hardware* : l'accès radio.

Si le processus a déjà été initié par des projets comme OpenBTS, il relevait encore du bricolage. Mais aujourd'hui nous voyons poindre des solutions matures et compétitives pour de nombreux cas d'usage.

Par exemple, les stations de base sont aujourd'hui largement spécifiées pour le marché des grands réseaux publics, seuls à même de justifier de lourds investissements de par les volumes mis en jeu.

À ce titre on remarquera l'échec du déploiement des *femtocell* et autre dérivés, coincées par les faibles marges.

Avec l'avènement des réseaux privés, par essence parfaitement hétérogènes, le nombre de cas d'usage et leur spécificité explosent. Un réseau sur une plateforme pétrolière ne présente pas du tout les mêmes caractéristiques et besoins qu'un réseau de diffusion itinérant sur un événement sportif.

Si l'on ajoute à cela l'incroyable flexibilité des technologies telle que la 5G, cette problématique devient vertigineuse.

Afin d'exploiter tout ce potentiel et ainsi permettre l'avènement de nouveaux marchés, ce qui fera la différence est la capacité des acteurs à concevoir des produits adaptés pour des petits volumes. Ainsi, l'approche *hardware*, parce que son développement initial est long et coûteux n'est pas adapté, car on ne peut envisager un *hardware* dédié qui serait optimisé pour tous les cas. À partir du moment où l'approche *software* est utilisée, le changement de *hardware* (le PC) n'est plus un problème.

L'adoption récente [2] par les références de l'industrie de CPU Intel x86 dans leurs produits d'accès radio est un indicateur significatif que ce changement de paradigme est déjà à l'œuvre.

RÉFÉRENCES

[1] DLAMINI, T., ROSSI M. & MUNARETTO D. (2017), “Softwarization of mobile network functions towards agile and energy efficient 5G architectures: A survey”, *Wireless Communications and Mobile Computing*, 8618364, 21.

[2] <https://www.ericsson.com/en/press-releases/2023/6/ericsson-first-cloud-ran-call-with-intel-latest-generation-processor-on-hpe-server> ; <https://www.nokia.com/about-us/news/releases/2024/02/21/nokia-collaborates-with-intel-to-deliver-high-performance-cost-efficient-private-5g-networks/> ; <https://news.samsung.com/global/samsung-expands-collaboration-with-intel-to-advance-vran-innovation>