



J. F. DAVID
Ingénieur Commercial
Agence Paris Industrie de Base et de Transformation

24 Mai 1975

la virtualisation pour quoi faire ?

**"Etre transparent, c'est exister et ne pas se voir.
Etre virtuel, c'est ne pas exister et se voir.**

.....
La virtualisation est transparente à l'utilisateur"

L'utilisateur cherche souvent une cohérence dans l'ensemble de la gamme des produits IBM.

Il est parfois intéressant de formaliser quelques principes qui sous-tendent le développement de ces produits.

Les techniques de virtualisation en sont un exemple.

Le fait le plus marquant de l'environnement actuel est son aspect évolutif.

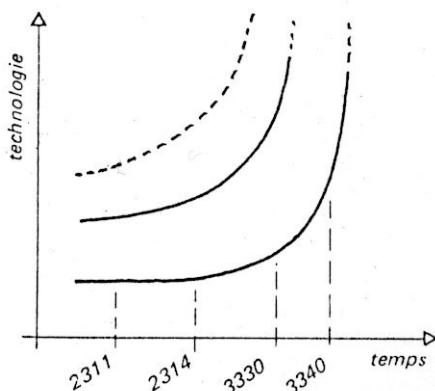
— Evolution clé de la technologie, très rapide dans certains secteurs.

— Evolution des structures de l'utilisateur qui varient au gré des pressions économiques et humaines.

Les investissements des utilisateurs de l'informatique doivent être protégés dans ce contexte mouvant.

De la mémoire virtuelle aux banques de données, des techniques mises en œuvre par la microprogrammation aux machines virtuelles en passant par les récents systèmes de stockage de l'information, la virtualisation représente peut-être une solution optimale à ces problèmes difficiles.

L'ENVIRONNEMENT NECESSITE UNE VIRTUALISATION



Les matériels évoluant de manière exponentielle il suffit de projeter n'importe quel facteur technologique (densité, débit, vitesse, capacité) en fonction du temps pour s'en apercevoir.

La courbe des coûts (de l'octet stocké! ...) est par ailleurs inverse.

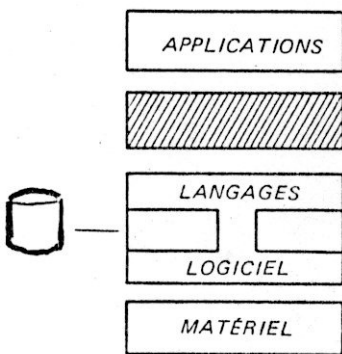
On pourrait penser que la technologie atteint aujourd'hui un pallier.

Ceci est erroné, aussi bien en vitesse de commutation (effet Josephson), en miniaturisation (techniques électroniques de fabrication...) qu'en stockage de l'information (mémoires "à bulles"...), des ratios de 1 à 200 par rapport à ce qui existe aujourd'hui sont envisagés à moyen ou long terme.

Les débuts de l'utilisation des techniques de type LSI (Large Scale Intégration) dans nos matériels laissent même à court terme présager d'une évolution rapide.

Les constructeurs sont amenés à faire également évoluer rapidement les logiciels pour faire bénéficier les utilisateurs des améliorations (fonctionnelles, prix/ performance) des matériels.

Le jour où, par exemple, un système de stockage de l'information qui a aujourd'hui un facteur de performance de 1 bénéficierait d'un facteur de 10, une évolution qualitative des méthodes d'accès devrait sans doute intervenir.

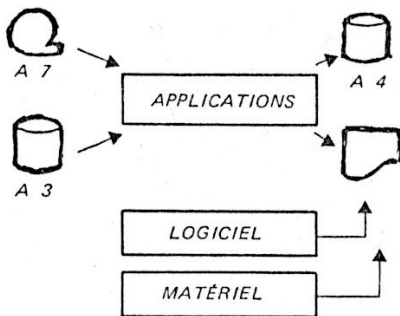


Les utilisateurs ne veulent plus voir leurs investissements humains (développement) remis en cause par ces évolutions dont ils veulent par ailleurs bénéficier,

Historiquement, l'utilisation des langages évolués apportait une certaine garantie.

Les investissements en analyse programmation semblaient protégés face à révolution des matériels (et des changements de constructeur)

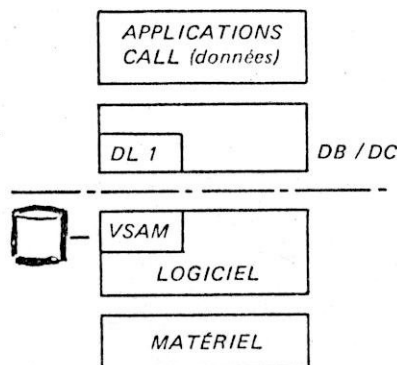
On sait aujourd'hui que ceci est faux dans deux domaines fondamentaux: le stockage et l'accès à l'information (locale ou à distance).



En exemple, considérons la conception des applications. La part d'autocensure liée à la technologie est énorme ("en entrée la bande A7 et le fichier A3, en sortie le fichier A4...").

Si demain une bande n'est plus une bande ou un disque devient autre chose, que restera-t-il de cette application ?

Quelle est la part de pure conception dans une application et la part liée aux contraintes des technologies actuelles ?



Si demain le disque devient une mémoire hiérarchisée ou la bande est remplacée par une unité à accès aléatoire, l'application sera sans doute à revoir, à repenser, à reprogrammer.

Si, par contre, l'application ne fait référence à l'extérieur que par des ordres purement conceptuels du type «appel donnée» ou «appel message», il y a de meilleures chances que l'investissement soit

préservé.

Le problème est encore plus complexe du côté du télétraitement ou la technologie permet de plus en plus d'intelligence «distribuée» et où les techniques de l'interface homme-machine évoluent très vite.

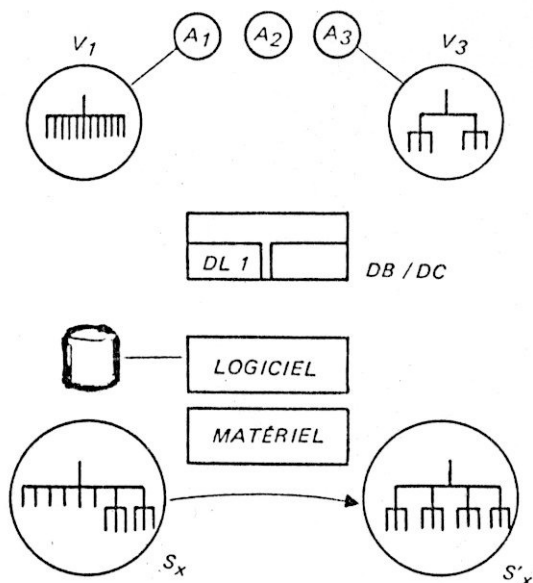
Ceci est une des raisons qui ont amené IBM à créer un nouveau type de relation «application-extérieur», le DB/DC...

DL1 OU LA BASE DE DONNÉE VIRTUELLE

On connaît bien maintenant le rôle fondamental des «bases de données» quant à la simplification des problèmes de communication dans l'entreprise et à la cohérence de ces communications (système d'information).

Le concept le plus important qui sous-tend DL1 est une virtualisation.

L'ensemble des données étant stocké de façon structurée (relationnelle) l'utilisateur (l'application) n'a pas à connaître l'ensemble de la structure mais ne connaît qu'une «base de données virtuelle»), que le système DL1 se chargera d'appliquer sur la structure complète.



Ce niveau de virtualisation permet un changement de structure du système d'information ($S_x \longrightarrow S'_x$) sans impact sur l'application de l'utilisateur qui garde la même «vision» du monde extérieur.

Il permet aussi les évolutions des matériels et logiciels de base. Même si un jour la conception de l'organisation de la base de données se modifiait (modèles relationnels, support à accès aléatoire de nouveaux types, mémoires associatives...), l'application pourrait conserver sa «vision» du monde extérieur.

D'autres niveaux de virtualisation existent en DL1.

La meilleure structuration des données au sens du système d'information de l'entreprise n'est peut être pas la façon la plus efficace de les stocker. Réfléter une dépendance par un chaînage physique sur un support à un moment donné de la technologie n'est peut être pas optimal.

En DL1, en fonction de l'utilisation prévue ou constatée des données, les informations sont stockées suivant des techniques diverses {indexés, aléatoires, chaînés, séquentielles} d'une façon transparente pour l'application.

Par ailleurs, et nous le verrons, l'utilisateur travaillera, grâce à VSAM, sur un support abstrait, virtuel.

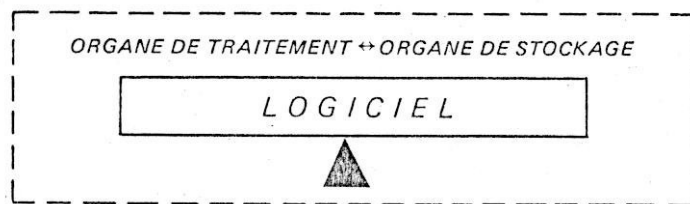
La virtualisation des données (Data Base Virtuelle) garantit l'investissement face aux changements.

Elle garantit l'indépendance des applications (l'application A4 n'a pas à s'interroger sur les modifications éventuelles de A1, A2, A3... qu'elle pourrait impliquer dans un système classique...). Elle permet une croissance sans phénomène de rétroaction. Nous retrouverons cette vertu de la virtualisation pour tous les éléments d'un système.

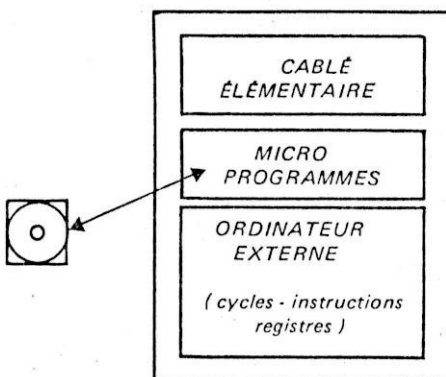
La virtualisation sous-tend l'ensemble de l'architecture 370. Nous allons parcourir rapidement les différents niveaux d'un système de base (matériel, logiciel de base, méthodes d'accès, disque et télétraitement}, sans tenter d'être exhaustifs, mais en tentant d'isoler les points fondamentaux, où l'investissement d'IBM apparaît comme majeur.

LE MATÉRIEL

Le matériel local peut être grossièrement décomposé en unités de traitement et unités de stockage de l'information, un équilibre devant être par ailleurs atteint entre ces deux parties et le logiciel qui les utilisera, problème rendu complexe par l'évolution exponentielle des possibilités des matériels.



La microprogrammation ou le traitement virtuel



La technique de microprogrammation est aujourd'hui la caractéristique principale de la conception de nos matériels de traitement.

Les années 1970 ont vu apparaître des câblés élémentaires pouvant développer des «horloges internes») rapides (50 nano-secondes). Ceci a permis de rentabiliser le principe de mémoire de contrôle rechargeable.

Les techniques précédentes (Read Only Storage) ne possédaient que partiellement la souplesse des techniques actuelles.

En effet la microprogrammation des unités est :

- une garantie d'évolution (une évolution des spécifications de la machine se fera dans la plupart des cas par simple recompilation par le constructeur d'un micro-programme),
- une technique facilitant le mode émulation (coexistence d'un code machine 370 et d'un autre Code (non 370),
- une contribution potentielle à la fiabilité et à la disponibilité globale.

En effet, l'existence de la microprogrammation permet dans la plupart des cas de concevoir un câblé très simple (building blocks), les fonctions étant développées par microprogrammes.

Une unité (disque, bande, ...) aura une mécanique simplifiée mais asservie par micro-programme («programme» de gestion des START/STOP sur bande, suivi «à la trace» d'une piste sur disque ...). Des ré-essais sophistiqués microprogrammés pourront garder le système disponible en cas de défaillance du hardware (CPU retry, CMD retry, correction «à la volée» d'informations en erreur..).

Des diagnostics microprogrammés peuvent surveiller la machine d'une façon continue ou à la demande.

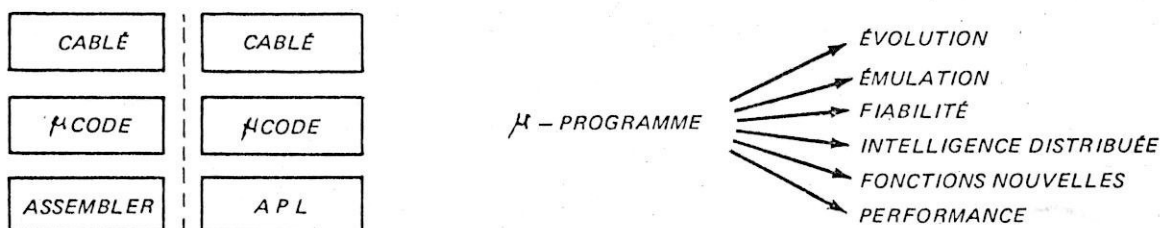
Ces matériels microprogrammés ont naturellement des chances d'être plus fiables (si la microprogrammation n'a pas d'incorrection de conception ou est «auto-dépannable»).

— La clé de voûte des phénomènes liés à «l'intelligence distribuée» où l'on recherche souvent des fonctions répétitives, performantes et souples. Il est très peu d'unités où la microprogrammation n'ait un rôle important (processeurs 115/125, unités de contrôle, terminaux 3740, 3600, 3650, 3767, 3770, 3790, ...3800, 3850,...).

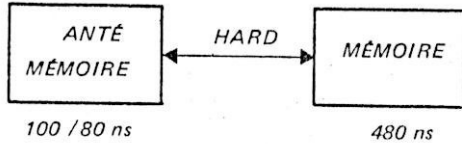
— Pour les logiciels utilisateurs, le microprogramme peut donner des possibilités fonctionnelles accrues (modification de la logique de fonctionnement ...), Donnons en exemple sur le 370 les logiques «Program Event Recording», «Monitor Call», où la mise d'indicateurs dans des registres de contrôle - intermédiaires naturels entre les logiciels et les microprogrammes - permet des fonctions de trace sélective, de dépannage «on-line» des logiciels par surveillance du système ...

— Une contribution fondamentale est enfin de modifier le profil des problèmes de performances. En effet, le code machine n'est pas exécuté par le câblé mais interprété par le micro-code. En spécialisant des routines de micro-code pour interpréter des ordres complexes, on peut changer la performance globale du logiciel. Ceci a été fait récemment pour VM (VM Assist) et pour APL où l'ensemble du langage a été microprogrammé sur 145. On pourrait d'ailleurs imaginer d'autres applications.

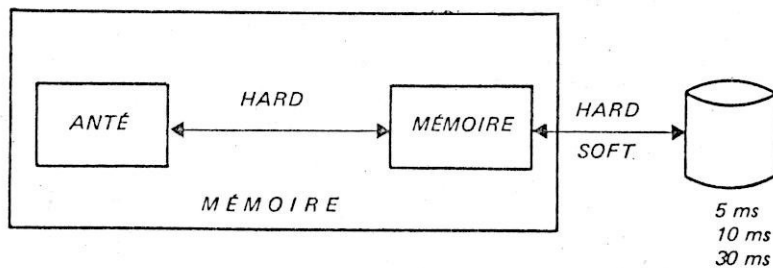
Les spécifications du câblé élémentaire ne sont plus directement ressenties par l'utilisateur qui communique avec le microprogramme, véritable virtualisation du câblé.



Les stockages de l'information ou les mémoires virtuelles

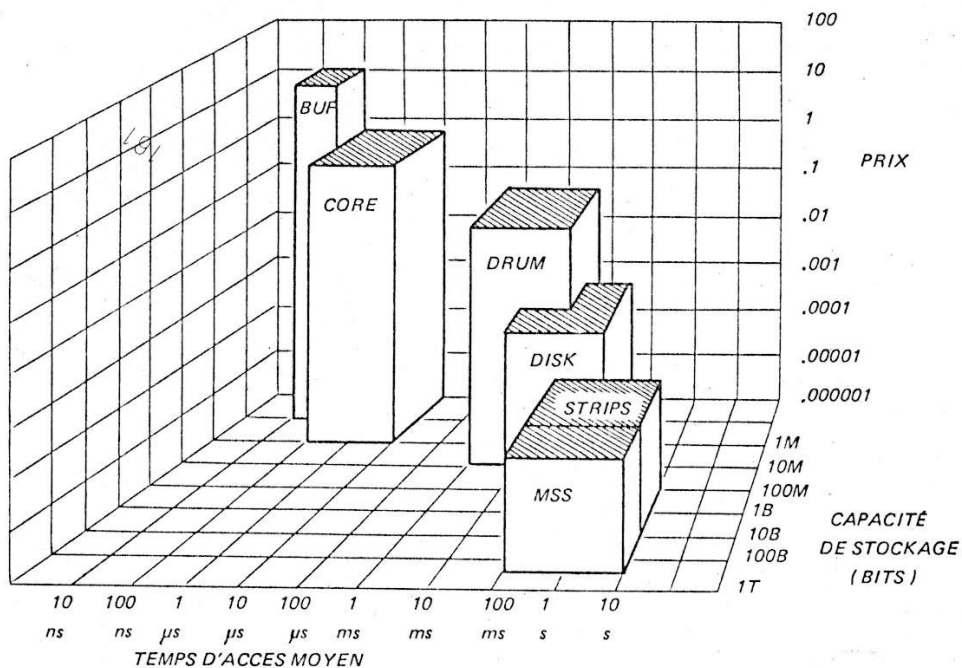


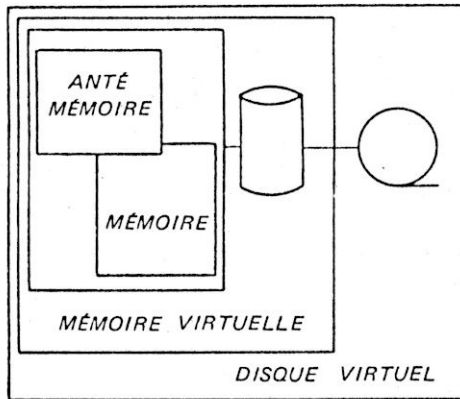
Certains de nos ordinateurs (158, 168) sont bâtis autour d'un concept d'antémémoire dont la liaison avec la mémoire est totalement «hardware», l'utilisateur bénéficiant alors dans 95 % des cas du cycle de base apparent de l'antémémoire. Ceci prouve, s'il en est besoin, le principe de localité selon lequel la probabilité de rester dans les mêmes zones d'un programme est supérieure à celle de se débrancher ailleurs. Ce principe sous-tend le développement de la mémoire virtuelle.



On peut considérer la mémoire virtuelle comme une généralisation de ce concept de hiérarchie à un espace disque de 16 M octets (DOS/VS, VS1), de $n \times 16$ M octets (VS2-2, VM).

Si l'échange de données (mémoire \leftrightarrow disque) n'est pas câblé, et que les logiciels participent à cet échange, c'est qu'il existe un «trou» technologique entre la mémoire la plus lente (1/2 us) et le disque le plus rapide (tambour à 5 ms, 3340 F à 10 ms, 3340 à 30 ms...). Il est bien évident que l'apparition d'une technologie X, intermédiaire, donnerait un sens accru à la mémoire virtuelle et aux concepts de hiérarchie (au-delà de leurs avantages actuels), en câblant les interfaces.

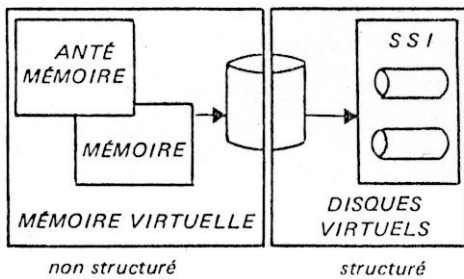




Néanmoins, dès à présent, l'utilisateur peut ne plus considérer la mémoire comme une ressource physique, mais comme un espace conceptuel, ce qui est fondamental (virtualisation d'une ressource).

On pourrait pousser plus loin la conception et imaginer une extension de la hiérarchisation à tous les supports de l'information (bandes...).

C'est ce que fait le 3350 avec la notion de disque virtuel. Le 3850 malgré tout ne virtualise pas tout l'espace (suppression du concept de fichier, espace-adresse quasi-infini) mais garde une notion de disque, de fichiers. Pourquoi ne pas pousser plus loin la virtualisation et supprimer le concept de fichier ?



Un système virtuel est un système disposant de puissance de traitement et de capacité de stockage.

Mais le stockage doit permettre l'accès à des données structurées, relationnelles, condition essentielle à une croissance cohérente des applications. Il y a donc à garder une différence entre la donnée adressable, utilisable par la puissance de traitement (espace adresse) et la donnée structurée pour stockage et recherche.

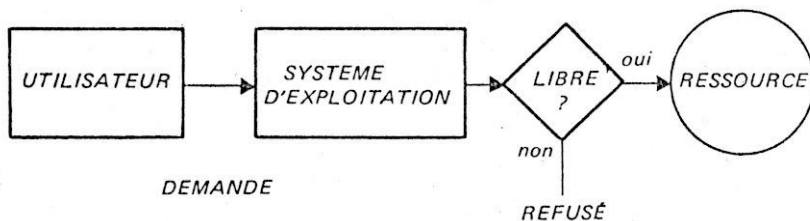
Les notions de mémoire virtuelle, de disque virtuel permettent à l'utilisateur de développer ses applications sans que celles-ci portent trace d'une contrainte quelconque due aux technologies existantes.

LES LOGICIELS DE BASE

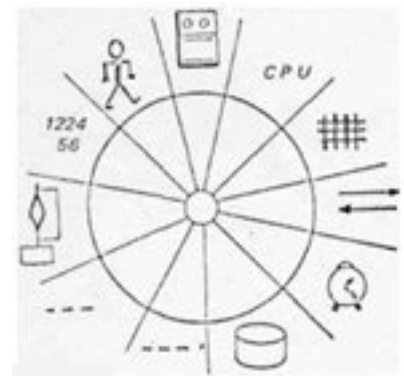
Les systèmes d'exploitation ou les ressources virtuelles

Le partage des ressources d'un système par plusieurs utilisateurs implique une notion de virtualisation. En effet, tous les utilisateurs ne peuvent accéder à un instant à des ressources non partageables (unité centrale, opérateur, canaux, section de code, données ...) du système.

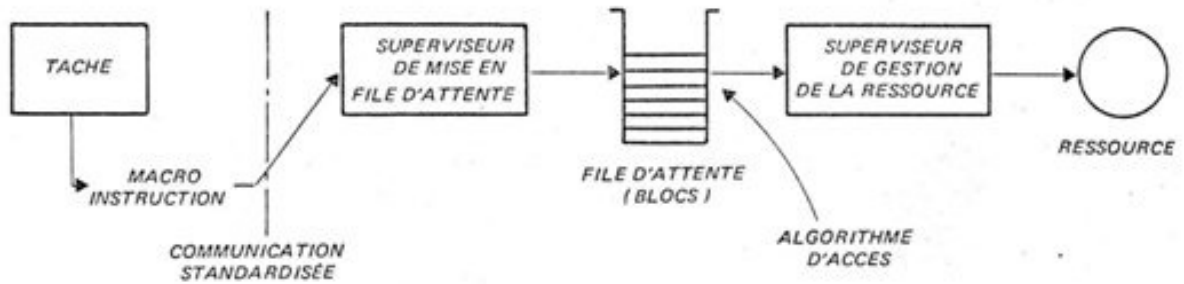
deux solutions sont alors possibles :
1ère solution



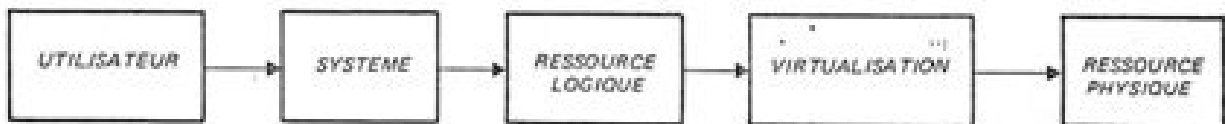
(c'était le cas de la mémoire avant la mémoire virtuelle).



ou 2ème solution

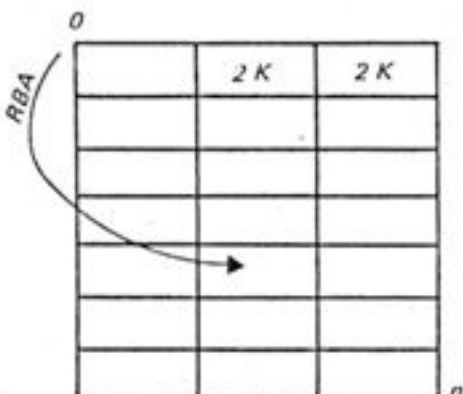


Nos systèmes d'exploitation sont construits autour de la deuxième philosophie qui est une véritable virtualisation de la ressource (l'utilisateur croit y accéder - en réalité, il la partage avec d'autres).



Les intérêts de cette architecture sont évidents : séparation de l'accès et de la gestion de la ressource (protection), possibilité à terme de contrôler l'ensemble des algorithmes pour rendre un certain service (VS 2-2), possibilité de conduite intelligente en cas de défaillance d'une ressource (on utilise une autre ressource sans que l'utilisateur s'en aperçoive, par exemple : reconfiguration dynamique d'unités centrales, de consoles opérateurs ...), possibilité de faire évoluer les algorithmes en fonction de la technologie sans modifier l'interface utilisateur ... etc... On voit que la virtualisation permet de rendre une ressource quelconque «pseudo-partageable».

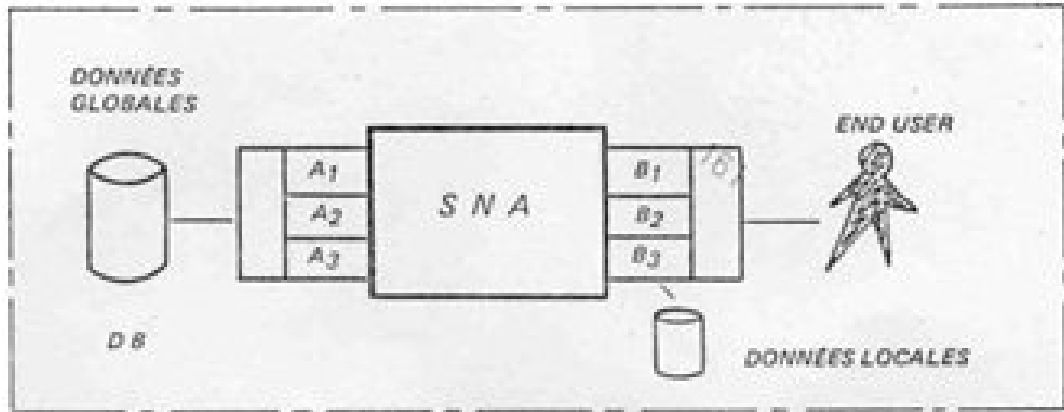
VSAM oule support virtualisé



Indépendamment de ses avantages spécifiques, VSAM est une méthode d'accès pour laquelle le fichier est défini comme un espace conceptuel, découpé en «blocs» abstraits (intervalle de contrôle, zone de contrôle) et où l'adressage d'une donnée se fait par octet relatif par rapport au début du fichier. Ceci rend bien évidemment l'organisation transparente à tout changement technologique ((mémoires de type indifférent) et libère l'utilisateur de toute préoccupation éventuelle liée aux disques (cylindres, pistes ...). Ce niveau de virtualisation est fondamental à terme. Des produits de type VIO (VS2-2) répondent par ailleurs à une généralisation de la mémoire virtuelle pour des fichiers non structurés qui sont traités par la pagination.

VTAM oule réseau virtualisé

La conception unifiée de réseaux (SNA) intègre une des virtualisations les plus importantes du système. Si l'on examine la situation d'un utilisateur accédant à un système à travers un terminal SNA, on a l'architecture suivante.



L'ensemble du réseau de communication est virtualisé.

La symétrie du schéma permet de faire apparaître l'identité des rôles de l'utilisateur et de la «data base» globale (fournisseurs d'informations, demandeurs d'informations pour maintenir leur synchronisme propre).

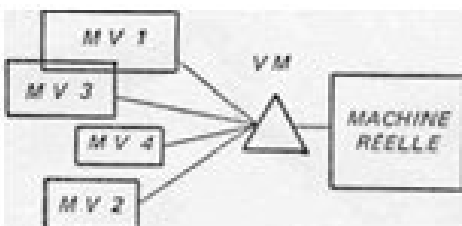
Entre l'application et l'utilisateur il y a simulation d'une Maison quelle que soit la structure du réseau.

A un instant donné l'utilisateur peut donc :

- inter-agir avec les données et les programmes locaux sans dépendre de l'ordinateur central,
- converser en temps réel ou en différé avec une application préprogrammée du système central,
- manipuler des données ou des programmes à travers des outils élaborés (GIS, time sharing, ...).

La technologie des terminaux et mini-ordinateurs se développe de façon exponentielle (intelligence développée, taille mémoire, puissance locale ...). Face à ce développement, la virtualisation est devenue nécessaire pour protéger à terme les investissements de l'utilisateur.

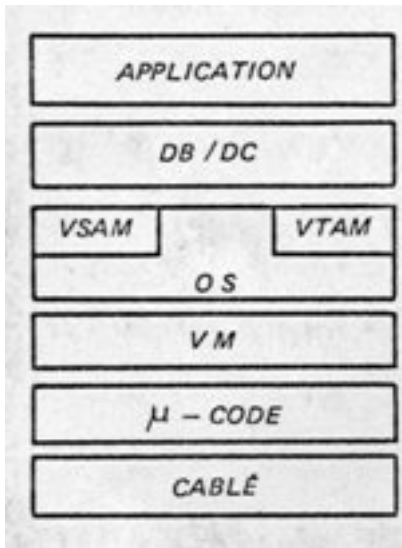
VM ou la machine virtuelle



L'architecture VM donne à chaque utilisateur un environnement optimal en distribuant dynamiquement les ressources physiques d'un système, sans tenir compte des relations logiques entre utilisateurs (ce que ferait un système d'exploitation), et permet de se libérer de toute contrainte liée à la configuration. Il satisfait entre autre le «rêve» du programmeur de développement d'avoir à lui seul une machine interactive de développement (CMS). Ce produit montre assez bien ce que l'on peut obtenir d'un simple niveau supplémentaire de virtualisation.

LA VIRTUALISATION ET SES PROBLEMES

Tout processus de virtualisation implique un coût. Accéder à une ressource virtuelle est moins performant que d'accéder à une ressource réelle.



Une mémoire virtuelle de 16 MO est moins performante qu'une mémoire réelle de 16 MO, un disque virtuel moins performant qu'un disque réel, une machine virtuelle qu'une machine réelle, une unité centrale partagée par n utilisateurs que n machines... Par ailleurs, la simple existence d'une série d'interfaces apporte un "overhead". Pour accéder en DL1 à un simple enregistrement, il faut "traverser" DL1, VSAM, l'OS, VM, le micro-code, le câblé à l'aller et au retour.

Mais ... affecter à un utilisateur l'ensemble des ressources dont il aurait besoin en période de pointe n'est en aucun cas optimal. D'autre part, la virtualisation permet de protéger les applications qui ne seront plus liées à aucune contrainte physique temporaire (espace disque, taille mémoire, machine unique, état des réseaux, télétraitement ...) et de garantir à terme les investissements de l'utilisateur en face du "choc du futur".

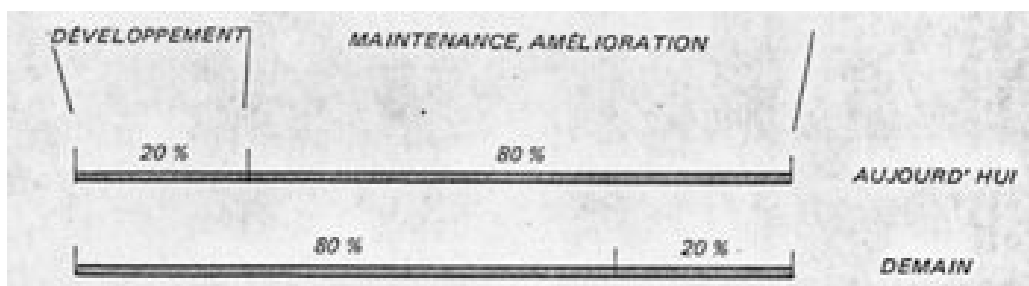
CONCLUSION

A chaque instant, les bénéfices acquis par l'indépendance et la virtualisation : doivent être pesés en face de "l'overhead" que ces fonctions impliquent. Mais, aujourd'hui où rentabiliser les hommes devient encore plus indispensable que rentabiliser le matériel (coûts respectifs du hardware et des hommes dans un service informatique) les techniques de virtualisation deviennent naturellement rentables.



Par ailleurs, dans un système où les ressources ne sont pas virtualisées les applications interfèrent les unes sur les autres.

Supprimant toute rétroaction de ce type, la virtualisation permet de consacrer l'essentiel des ressources à la création de nouvelles applications et de consacrer un minimum aux fonctions d'amélioration et de maintenance.



La technologie permet aujourd'hui d'intégrer la virtualisation dans tous les domaines, fournissant à l'utilisateur un environnement enfin stable et cohérent pour une croissance plus efficace.