

Chapitre 2 : Les Systèmes Parallèles et Temps Réel (SPTR)

Ce sont des systèmes numériques programmables intégrant des modules avancés au niveau du matériel et du logiciel.

Ils sont interfacés au monde réel et sont utilisés dans des applications complexes pour lesquelles les spécifications sont nombreuses et exigeantes.

Un de leurs aspects spécifiques est donc un besoin de traitement informatique intense, performant, en partie "intelligent", et soumis à des contraintes dynamiques.

Trois thèmes principaux sont reliés aux aspects fondamentaux et techniques de ces systèmes intelligents en interaction avec un environnement dynamique: les architectures et le traitement parallèles, le traitement en temps réel, et la fiabilité et la tolérance aux fautes.

Dans les SPTR, ces trois thèmes sont intimement interreliés autant au niveau du matériel que du logiciel.

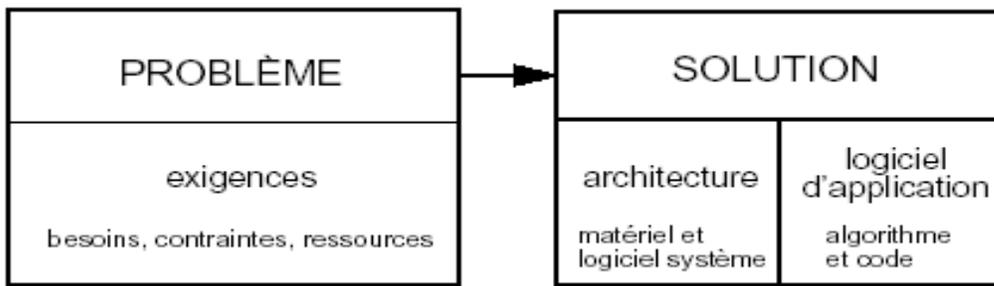
Par exemple, les SPTR ont typiquement de nombreuses contraintes de temps réel sévères et doivent donc s'exécuter sur du matériel performant et fiable.

On utilise alors des architectures de calcul avancées de type multiprocesseurs, en parallèle ou distribués, dans lesquelles les différents éléments de traitement communiquent au moyen d'un réseau d'interconnexion pour échanger des données, se synchroniser et collaborer à la réalisation des applications complexes.

II.1 Développement d'un SPTR

De façon générale, le processus de développement d'un système peut être vu comme la recherche d'une solution à un problème. Ce problème est défini par les **exigences** du système à développer : des **besoins** à satisfaire, à l'aide des **ressources** disponibles, en respectant des **contraintes** imposées.

Dans le domaine du génie informatique en général et des SPTR en particulier, la solution ou application comportera généralement deux parties principales: l'**architecture** (**matériel** et **logiciel système**) et le **logiciel d'application** (**algorithme** et **code**).



Processus de développement d'un système

Une solution à un problème correspond à une alternative, c'est-à-dire à une suite de choix ou décisions qui sont pris pour satisfaire les besoins, en général le mieux possible i.e. en faisant certains compromis, et ce dans le respect des contraintes.

Lorsque le processus de développement se limite à la **conception** du système, la solution est plus ou moins détaillée et elle consiste en un **modèle** plus ou moins général du système à réaliser. Par exemple, on peut définir formellement, dans un langage approprié, l'architecture matérielle du système et l'algorithme de traitement des données qui sera utilisé.

Il y a deux approches principales pour le développement d'un système. La première est une approche qui part des exigences de l'application sans égard aux ressources disponibles. C'est une approche qui définira la solution en allant du général au particulier ("**top-down**"): modèle général de la solution puis division en modules de plus en plus détaillés jusqu'à l'implantation ou réalisation.

La seconde approche part plutôt des ressources disponibles et de leurs limitations précises. C'est une approche qui définira la solution en allant du particulier au général ("**bottom-up**"): modèle détaillé des caractéristiques et possibilités des modules mis en jeu dans l'implantation puis intégration de ces modules en un système complet.

En pratique, le développement d'un SPTR empruntera simultanément aux deux approches dans un processus itératif. Par exemple, le système pourra d'abord être défini par un modèle très général (voir modèle SPTR plus loin) tandis que certaines parties de la solution détaillée pourront être fixées par les exigences du problème, par exemple l'architecture pourra être imposée par les ressources disponibles.

II.2 Méthodologies et objectif

Le développement et la mise en oeuvre d'un SPTR nécessite en pratique une coordination des efforts des différents membres d'une équipe et l'utilisation de méthodologies avancées combinant, entre autres, une approche intégrée de design et des outils informatiques d'aide à la conception.

Ainsi, de nombreuses méthodes ont été proposées et des outils et environnements de développement matériel et logiciel basés sur ces méthodes sont maintenant disponibles commercialement.

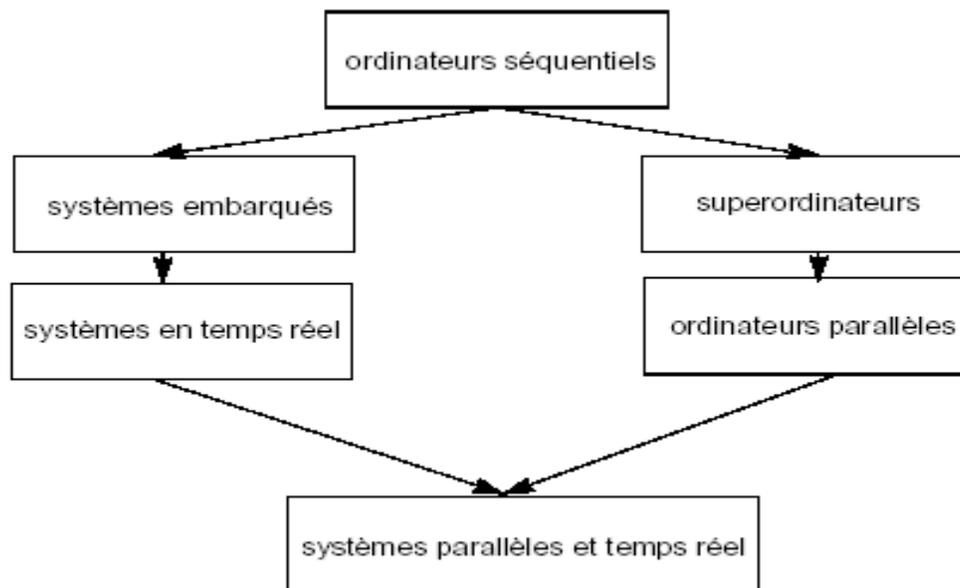
L'étude et l'expérimentation d'une ou plusieurs de ces méthodes se fera plus naturellement en milieu de travail ou dans un cours spécifiquement dédié au design et à la réalisation de projets.

Ici, notre objectif consistera plus simplement à présenter et analyser divers principes fondamentaux et concepts techniques qui sont communs aux différents types de SPTR.

Ces principes et concepts seront présentés dans un contexte de développement intégré et en considérant autant les aspects matériels que les aspects logiciels de ces systèmes.

II.3 Historique des SPTR

Voyons d'abord les grandes étapes ayant conduit à l'étude et au développement des SPTR et qui sont illustrées à la figure suivante, que l'on doit examiner en allant du haut vers le bas.



Historique des SPTR

Les premiers **ordinateurs** utilisés à grande échelle étaient basés sur un modèle de traitement **Séquentiel** avec un processeur unique. Ce modèle, toujours le plus commun, s'est ensuite affiné avec l'introduction de logiciels d'exploitation plus évolués, particulièrement pour permettre le traitement simultané de plusieurs applications en partageant le temps de calcul disponible.

Ces premiers ordinateurs étaient fabriqués à partir de composants discrètes, d'abord des relais électromécaniques puis des tubes et enfin des transistors. La

majeure partie des applications réalisées à l'aide de ces ordinateurs (branche de droite de la figure) utilisaient un modèle de système très simple: réception en entrée d'un ensemble de données structurées, traitement de ces données d'une manière prédéterminée et le plus rapidement possible puis transmission du résultat de ces traitements à la sortie.

Le développement des circuits intégrés et de la micro-électronique a permis d'augmenter la performance des ordinateurs pour ces applications et de créer ainsi les premiers **superordinateurs**. Ceux-ci étaient utilisés dans les calculs scientifiques de plus en plus complexes où la quantité de données manipulées était de plus en plus grande, d'où le besoin d'améliorer la performance.

Étant donné le marché relativement restreint des ordinateurs scientifiques, gouvernementaux ou corporatifs, et dans le but d'assurer leur rentabilité face à des coûts de développement toujours croissants (en 1997, on parlait d'investissements de 1 milliard de dollars US pour une usine de fabrication de circuits VLSI à la fine pointe de la technologie), les compagnies de micro-électronique ont ensuite visé à développer des marchés de masse.

Ceux-ci correspondaient en particulier aux **systèmes embarqués** ou dédiés (branche de gauche de la figure), i.e. aux ordinateurs utilisés pour réaliser des fonctions de contrôle et de calcul dans un système plus vaste. On retrouva rapidement ces systèmes dans les produits d'électronique de consommation, l'automobile, les télécommunications, l'instrumentation, les produits d'automatisation, etc.

Face à tous ces nouveaux besoins, le domaine de l'informatique en général, autant du côté matériel que du côté logiciel, a ensuite évolué rapidement, que ce soit au niveau académique ou de recherche qu'au niveau industriel.

Cette croissance a permis, entre autres, le développement de systèmes embarqués dans des applications de plus grande complexité i.e. traitant sur la base de contraintes diverses et nombreuses, des données d'une plus grande variabilité et donc moins structurées. De plus, ces applications comportaient souvent une composante dynamique beaucoup plus importante et exigeante, nécessitant ainsi des **systèmes en temps réel** pour lesquels la ponctualité des résultats est mise sur un pied d'égalité avec la vitesse et l'exactitude du traitement.

Ce nouveau besoin influa à son tour sur le développement de nouveaux circuits spécialisés pour les systèmes embarqués comme les microcontrôleurs, les processeurs de traitement de signal, les circuits d'interface, etc., de même que sur la définition de normes pour l'intégration de ces systèmes, par exemple les bus d'interconnexion de modules.

Le besoin toujours croissant de performance, en particulier pour les applications structurées, ainsi que la limite technologique entrevue sur le modèle séquentiel,

mena ensuite au regroupement de plusieurs processeurs pour former des **ordinateurs parallèles**, parfois intégrés sur des bus d'interconnexion ou reliés en réseau, donnant lieu au domaine de l'informatique distribuée.

Cette nouvelle évolution des systèmes informatiques a été rendue possible, entre autres, par l'augmentation régulière de la performance des microprocesseurs, par la baisse de leurs ratio prix/puissance, dimensions, poids et consommation d'énergie.

Allié aux besoins plus grands en flexibilité et fiabilité des systèmes en temps réels qui s'attaquaient à des applications de plus en plus exigeantes et complexes et à l'importance de plus en plus grande prises par les réseaux informatiques, ce nouveau domaine s'est aussi développé rapidement.

Ainsi, avec la disponibilité de ressources matérielles et logicielles de plus en plus sophistiquées, il est maintenant possible de considérer de façon intégrée les systèmes ou applications diverses réunissant les deux branches de la figure et qui forment ce que nous appelons les SPTR.

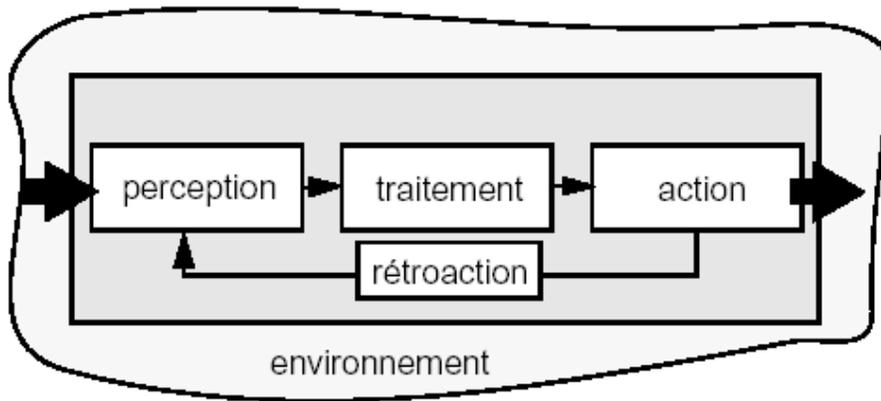
Lorsque le SPTR étudié sera propre à la branche de droite de la figure, nous parlerons d'une application structurée. Inversement, lorsqu'il aura plus d'affinités avec la branche de gauche, nous dirons que l'application visée est non-structurée.

D'une façon générale, les applications d'intérêt seront non-structurées à la base mais elles pourront comporter une partie qui est structurée, par exemple un module spécialisé pour le calcul rapide de fonctions particulières.

II.4 Modèle général des SPTR

D'un point de vue général, les SPTR peuvent être vus comme des systèmes numériques complexes de commande et de contrôle.

La figure ci-dessous donne un vue schématique d'un modèle général des SPTR, que nous appellerons simplement le **modèle SPTR**.



Modèle SPTR

Remarquons d'abord que ces systèmes sont intimement couplés à l'**environnement**. En ce sens, ce sont donc des systèmes embarqués. Cependant, cet environnement n'est pas limité à des mécanismes ou à des procédés. Il peut aussi inclure un environnement ouvert, aux limites physiques non définies, incluant d'autres systèmes et personnes avec lesquels le système peut interagir.

C'est le cas, par exemple pour un véhicule robotisé autonome qui utilise la vision numérique pour se déplacer dans des environnements complexes, à l'intérieur ou à l'extérieur, et qui peut rencontrer d'autres véhicules ou des personnes.

Dans ce cas précis, la **perception** consiste à faire l'acquisition d'images de l'environnement, le **traitement** consiste à les analyser pour en comprendre le contenu et l'**action** peut consister, par exemple, à se déplacer pour éviter certains obstacles, à changer l'orientation des caméras de prise de vue, à reconnaître des personnes et interagir avec elles ou à saisir et déplacer des objets.

On voit donc que ces actions peuvent avoir une influence autant sur le système lui-même que sur l'environnement dans lequel il évolue. C'est pourquoi on remarque dans le diagramme un module de **rétroaction** interne au système et un deuxième couplage de la sortie sur l'entrée par l'intermédiaire de l'environnement.

Dans ces systèmes, le traitement est souvent varié et complexe et peut donc nécessiter des architectures matérielles avancées, par exemple des ordinateurs parallèles. Comme on l'a dit, une partie de ce traitement peut être structurée. En fait, le modèle SPTR peut même s'appliquer directement, dans son ensemble, aux SPTR structurés. Dans ce cas, l'environnement représente la source et la destination des données traitées.

Aussi, même si un seul bloc est utilisé pour représenter chacun des modules de perception, de traitement et d'action et que ces blocs sont dans une boucle unique, la réalisation d'un SPTR implique généralement du matériel distribué, reproduit en plusieurs exemplaires, parfois hétérogène, ainsi que du traitement

parallèle avec des contraintes temporelles variées. Par ailleurs, autant le système que l'environnement ont des composantes dynamiques.

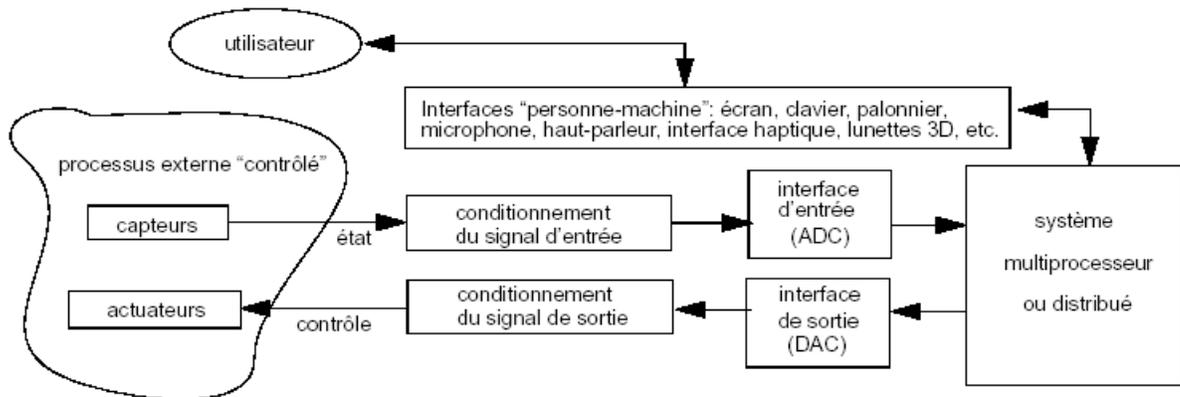
Dans l'exemple donné plus haut, le véhicule autonome se déplace, ses caméras peuvent changer d'orientation et des objets (autres véhicules, personnes) peuvent être mobiles (ou déplacés) dans l'environnement.

Les modules de perception, de traitement et d'action ont donc souvent à faire face à des contraintes temporelles, ce qui fait que ces systèmes doivent fonctionner en **temps réel**.

Finalement, on peut mentionner que ces systèmes fonctionnent souvent en continu, sur de longues périodes et doivent donc être **fiables** et **tolérants aux fautes**.

II.4.1 Architecture matérielle des SPTR

Une représentation schématique de l'architecture matérielle d'un SPTR est donnée à la figure ci-dessous.



Architecture matérielle d'un SPTR

On a dit plus haut qu'un SPTR était un système embarqué, au sens large, avec des propriétés particulières résultant de la complexité des domaines d'application, comme par exemple le traitement en temps réel, le parallélisme et la tolérance aux fautes.

Il est aussi à noter que dans les SPTR, l'utilisateur est considéré partie intégrante de l'environnement.

II.4.2 Exemples de SPTR

Le modèle général présenté ci-dessus s'applique presque sans distinction à tous les domaines d'application avancés en ingénierie.

Pour ce faire, il s'agit de déterminer ce que sont les différents modules de perception, de traitement et d'action, ainsi que l'environnement du système, dans chaque cas.

La liste suivante donne divers domaines d'application où ce modèle s'applique bien:

- commande de procédés industriels.
- production automatisée intégrée et flexible.
- commande de mécanismes divers.
- robotique intelligente et vision artificielle.
- surveillance et reconnaissance automatisée.
- avionique et aérospatiale.
- exploration sous-marine.
- réseaux de télécommunication numériques avec ou sans fil.
- véhicules automobiles et autres véhicules autonomes ou assistés.
- système d'accès à des bases de données distribuées.
- traitement distribué de transactions et commerce électronique.
- systèmes experts et de prise de décision.
- télémanipulation et téléopération.
- télé-conférences et télé-apprentissage.
- multimédia et réalité virtuelle ou augmentée.
- gestion du trafic terrestre et aérien.
- acquisition de données distribuée.
- imagerie médicale et monitoring.
- simulation distribuée et intervention assistée.

À titre d'exemple, le système informatique embarqué d'un véhicule automobile peut être vu comme un SPTR.

En effet, l'automobile, par l'entremise de ce système informatique, est en interaction avec un environnement dynamique et complexe (réseau routier et conducteur) et elle comporte plusieurs sous-systèmes embarqués ayant des fonctions diverses et qui doivent fonctionner de façon partiellement autonome.

Parmi les sous-systèmes informatisés actuels et futurs de l'automobile, nous pouvons citer :

- le contrôle dynamique du véhicule (moteur, transmission, freins, vitesse de croisière, suspension, traction, direction).
- l'aide à la perception (éclairage visible et ultra-violet, télémétrie radar courte-distance, alerte).
- les systèmes d'information du conducteur (circulation, navigation, reconnaissance des panneaux routiers, tableau de bord électronique).

- les fonctions de confort du conducteur (climatisation et chauffage, verrouillage central, contrôle des fenêtres, sièges, miroirs, etc., systèmes anti-vol).

Le concept des autoroutes intelligentes va encore plus loin en considérant les différents systèmes contrôlant l'ensemble du réseau routier, incluant les axes routiers, les véhicules, la signalisation, etc., comme un seul SPTR qui doit être développé pour maximiser le débit des véhicules, réduire le nombre d'accidents et la consommation d'énergie, etc.

On voit donc qu'un SPTR, e.g. celui d'un réseau routier intelligent, peut être une structure hiérarchique où chaque élément, e.g. les systèmes informatisés des véhicules automobiles, est aussi un SPTR. Une partie du contrôle est globale (pour optimiser l'ensemble du réseau routier) et une partie est locale (pour optimiser chaque véhicule).

(Réf. : Robert Bergevin, (2002). *Systèmes parallèles et temps réel, Notes de cours, Université Laval*)