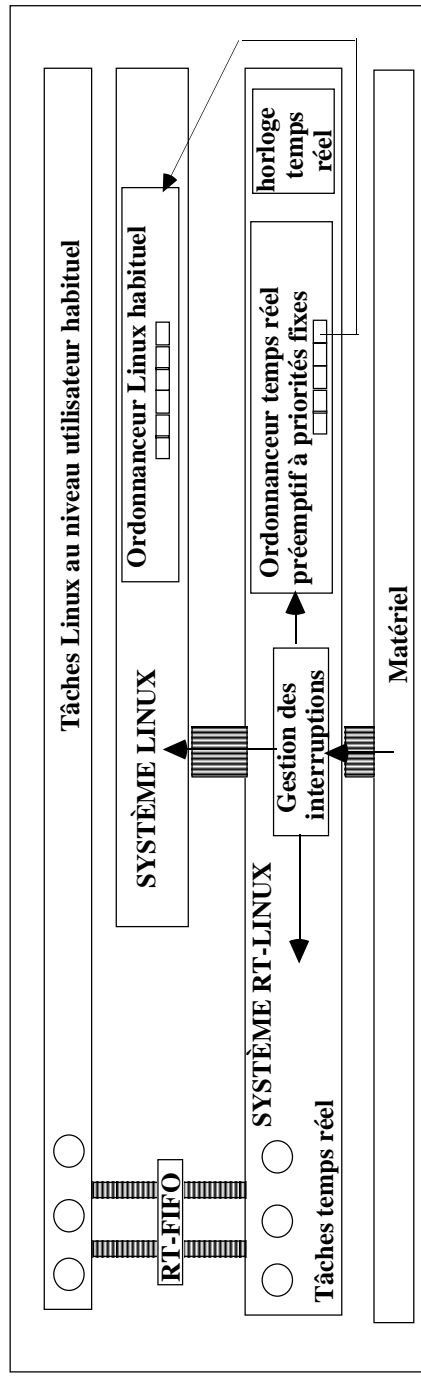


## 4. QUELQUES EXEMPLES DES ÉVOLUTIONS ACTUELLES DU TEMPS RÉEL

### RAPPEL DES QUALITÉS REQUISES DES STRUCTURES D'ACCUEIL MATÉRIELLES ET LOGICIELLES

- prévisibilité ("predictability") des temps d'exécution : dans le pire des cas, quelle que soit la charge et le taux de partage  
=> calcul de la capacité à respecter les contraintes temporelles
- déterminisme : faible dispersion autour de la valeur moyenne des temps de réponse  
=> robustesse des protocoles et des allocations de ressource, en présence de variation de trafic, de pannes matérielles, de congestion et de surcharge
- contrôle et respect des échéances de bout en bout
- précision suffisante de la mesure du temps et des datation des événements,
- fiabilité et disponibilité pour assurer la permanence du service

### 4.1. LINUX-RT : ADAPTER LINUX POUR LE TEMPS RÉEL



Sous le noyau normal de Linux un module temps réel qui :

- récupère les interruptions matérielles et transmet au système standard celles qui lui sont destinées.
- gère aussi une horloge de meilleure définition que l'horloge standard,
- contient des tâches temps réel, sans entrées-sorties tamponnées, mais RT-TFIFO vers les tâches Linux
- gère un ordonnanceur temps réel préemptif à priorités fixes

L'ordonnanceur Linux, et ses processus, est une tâche moins prioritaire que les tâches temps réel

## 4.2. ADA : PROPOSITIONS POUR UN ANNEXE TEMPS RÉEL DISTRIBUÉ

### DISTRIBUTION ET PLACEMENT EN ADA 95

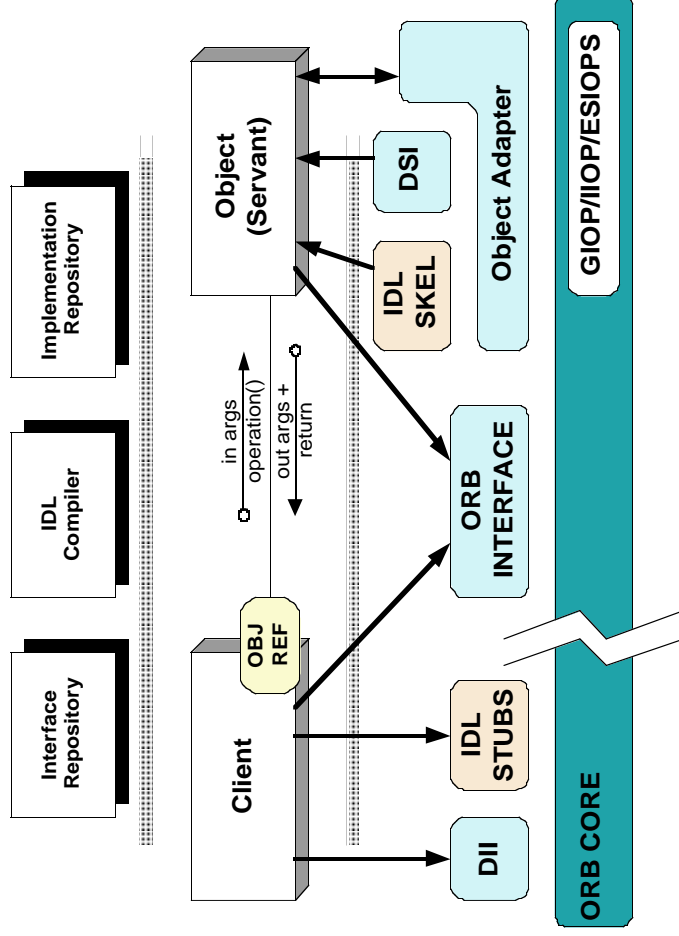
- Un programme ADA est un ensemble de PARTITIONS, actives ou passives, exécutables en parallèle
- Chaque partition active a un espace d'adressage qui lui est propre. On peut placer explicitement des unités de bibliothèque (sous-programmes, paquetages et unités génériques) dans une partition active.
- Si des unités de bibliothèque ont des variables d'état, il y a une copie de ces variables dans chaque partition qui l'utilise et ces copies évoluent indépendamment dans chaque partition.
- On peut placer plusieurs copies d'une partition sur un même site ou sur des sites différents.
- Une partition peut être avortée et rechargée dynamiquement à tout moment, sans arrêter les autres.
- Toute référence à des données ou tout appel de sous-programme d'une partition à une autre est un accès distant ("remote access"). L'interface de l'appel à distance ("RCI" : "remote call interface") est défini par paquetages qui sont placés dans chaque partition communicante et l'appel de procédure distante correspondante ("RPC" : "remote procedure call") est réalisé par des paquetages de bibliothèque. On peut avoir des RPC asynchrone pour des procédures avec paramètres "in" seulement (appel) .

### COMPLÉMENTS PROPOSÉS POUR UN ANNEXE TEMPS RÉEL DISTRIBUÉ

- Interfaces pour une conduite par priorité des entités ordonnançables : tâches, RPC, messages type priorité globale et mappage avec les priorités locales à chaque site
- Mécanismes pour définir la priorité des serveurs
  - priorité des pilotes de RPC, émetteur et récepteur,
- Mécanismes pour éviter ou limiter l'inversion de priorité
- Gestion spécifique de l'allocation de ressources
  - configuration de pools de pilotes de RPC

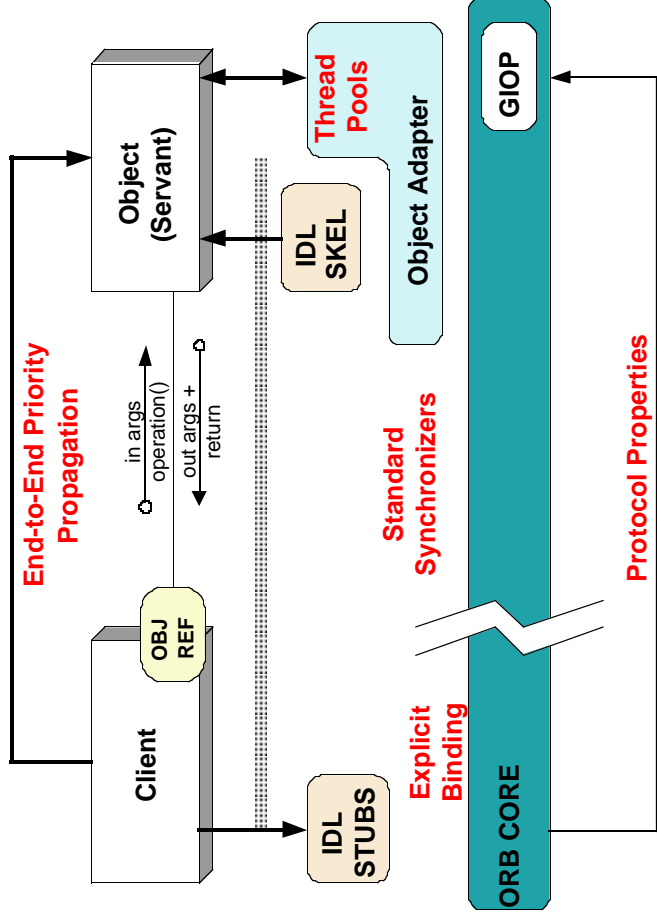
Besoin ressenti d'étendre aussi le profil RAVENSCAR pour les applications distribuées

## 4.3. REAL TIME CORBA : ADAPTER CORBA POUR LE TEMPS RÉEL CORBA NON TEMPS RÉEL



**RPC** : STUB et SKEL : souche et squelette statique de RPC    **DII** et **DSI** : souche et squelette dynamiques  
**ORB** : object request broker    **GIOP** : General Inter-Orb Protocol (IIOP : sur TCP)

### 4.3. REAL TIME CORBA : ADAPTER CORBA POUR LE TEMPS RÉEL



### 4.3. REAL TIME CORBA : ADAPTER CORBA POUR LE TEMPS RÉEL

#### SPÉCIFICATIONS POUR CORBA TEMPS RÉEL

#### OMG SPECIFICATION : ORBOS/99-02-12, AVEC UN ORDONNANCEMENT PAR PRIORITÉS FIXES

Vise les applications dont le fonctionnement nominal est bien connu, avec des charges statiques qui peuvent être étudiées a priori hors ligne, qui peut être géré avec des priorités fixes si les tâches, le système, le réseau et la plate-forme respectent ces priorités.

- *Attribution de priorités fixes globalement*

Priorités fixes globales au niveau de la plate-forme pour les processus ("threads") : permet d'utiliser RM

Tables de mappage : priorité globale <-> priorité dans le système de chaque site.

- *Propagation de priorité de bout en bout:*

Soit le serveur reçoit une priorité à sa création ( $\pm$  dynamique) , il n'y a pas vraiment propagation

Soit la priorité globale du client est propagée dans la requête vers le serveur

Action sur les paramètres des protocoles de communication ou choix du protocole de communication

Choix par le client de la nature des connexions vers l'objet client :

connexions multiplexées, connexions privées réservées, connexion dynamiques avec priorités

- *Limitation de l'inversion de priorité*

Exclusion mutuelle par "mutex" global (verrou, sémaphore binaire) associé à un protocole de priorité plafonnée ou à un héritage de priorité.

- *Limitation du coût temporel de l'allocation dynamique* : définition de lots de ressources préallouées

Pool de processus préparés ("threadpools"), pool de connexions, pool de tampons

- *Autres mécanismes globaux* : délais de garde (temps global)

### 4.3. REAL TIME CORBA : ADAPTER CORBA POUR LE TEMPS RÉEL

#### OMG SPECIFICATION : ORBOS/2001-04-01 AVEC UN ORDONNANCEMENT DYNAMIQUE

- Contexte : La charge nominale n'est pas bien connue (on ne peut prédire un fonctionnement nominal) ou elle est trop variable pour une planification a priori (futur mal connu, changements de modes).
- Abstraction d'une tâche répartie : "distributable thread" : le traitement évolue d'un site à un autre, en emportant un contexte d'ordonnancement. Il a un nom unique global et un seul site à tout instant est actif pour la tâche répartie, même si elle est implémentée comme un groupe de processus, un par site.
- Une tâche répartie ("distributable thread") est un ensemble de segments ordonnancables, éventuellement emboîtés, chaque segment pouvant avoir une politique d'ordonnancement, temps réel ou non, et chaque segment pouvant se dérouler séquentiellement sur plusieurs sites.
- Coopération entre ordonnanceurs des divers sites : propagation des informations d'ordonnancement concernant une tâche répartie (segment, discipline d'ordonnancement). Interfaces pour un ensemble de disciplines connues (priorités fixes, "earliest deadline", ED et importance) : devrait permettre le changement dynamique de politique d'ordonnancement
- Limitation des surcoûts temporels en limitant l'interopérabilité

version novembre 2001

7

00000000

### 4.4. AUTRES ADAPTATIONS POUR LE TEMPS RÉEL

- **JAVA TEMPS RÉEL ET UML TEMPS RÉEL** : même genre d'approche (souvent par les mêmes personnes)
  - Seul travail avancé pour les applications temps réel critiques réparties, celui fait pour ADA
- **TOLÉRANCE AUX FAUTES DANS SYSTÈMES RÉPARTIS TEMPS RÉEL STRICT**
- Parmi les qualités requises figurent la fiabilité et la disponibilité pour assurer la permanence du service
  - => architecture redondante et prise en compte du temps de gestion de la tolérance aux fautes
- Détection des erreurs : émission périodique par chaque site de messages de vie
- Traitement d'erreur par réplication pour redondance active, passive ou semi passive :
  - => problème important du déterminisme des actions des tâches répliquées
- livraison des messages de calcul, décisions d'ordonnancement des différentes copies, choix du bon résultat
- Correction de faute par passivation du matériel défaillant et reconfiguration
- Aspects spécifiques à la répartition
- Les décisions dans une architecture répartie, en présence de défaillances de site ou de messages font appel à un protocole de CONSENSUS réparti ou d'un protocole de DIFFUSION ATOMIQUE (diffusion fiable et totalement ordonnée) qui résout l'indéterminisme au niveau du protocole de communication. Ces protocoles sont coûteux en temps d'exécution et il faut savoir en tenir compte dans les temps de calcul en prenant des bornes supérieures.
  - Voir [Chevochet 1999] ou [Ledinot 1998]
- **CONCLUSION POUR LE TEMPS RÉEL** : très grand dynamisme actuel des professionnels du domaine

version novembre 2001

8

00000000