



RealTime JAVA

Thibault Jurado-Leduc RICM5 2011/2012



Sommaire

1. Le temps réel
2. Java Real-Time
 1. RTSJ
 2. RT JVM
3. Démo

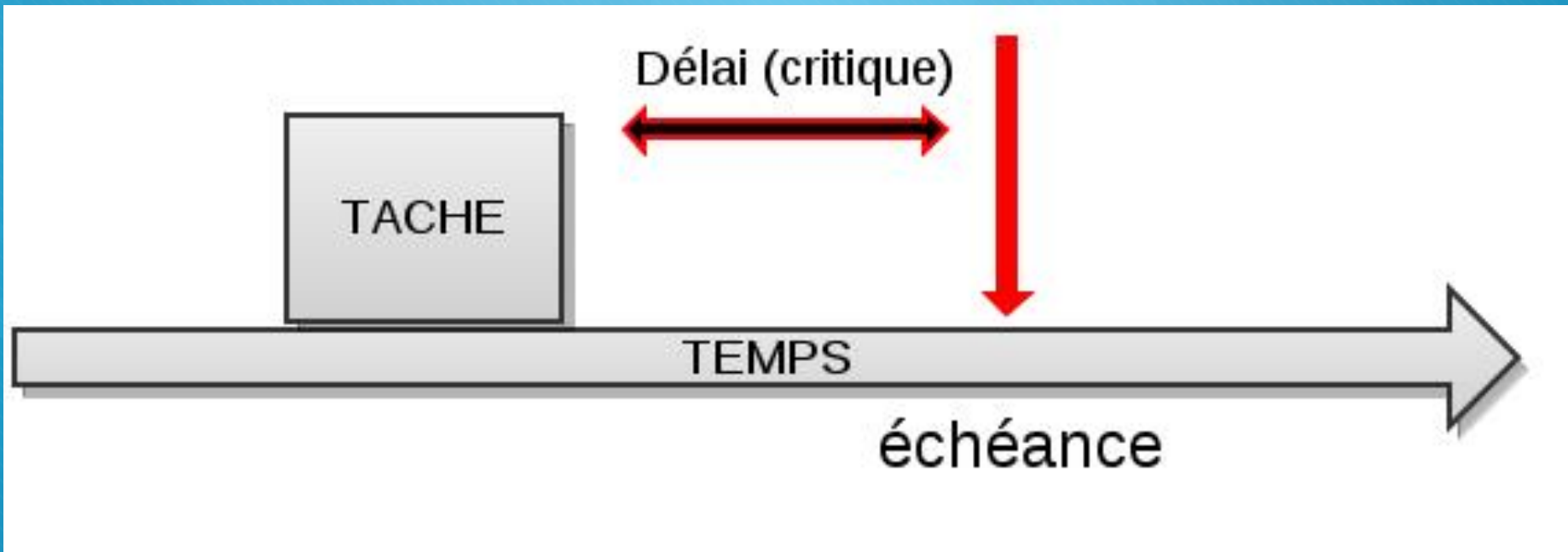


Une définition :

« Un système fonctionne en temps réel s'il est capable d'absorber toutes les informations d'entrée sans qu'elles soient trop vieilles pour l'intérêt qu'elles représentent, et par ailleurs, de réagir à celles-ci suffisamment vite pour que cette réaction ait un sens »

Abrial-Bourgne.

Quèsaco ?

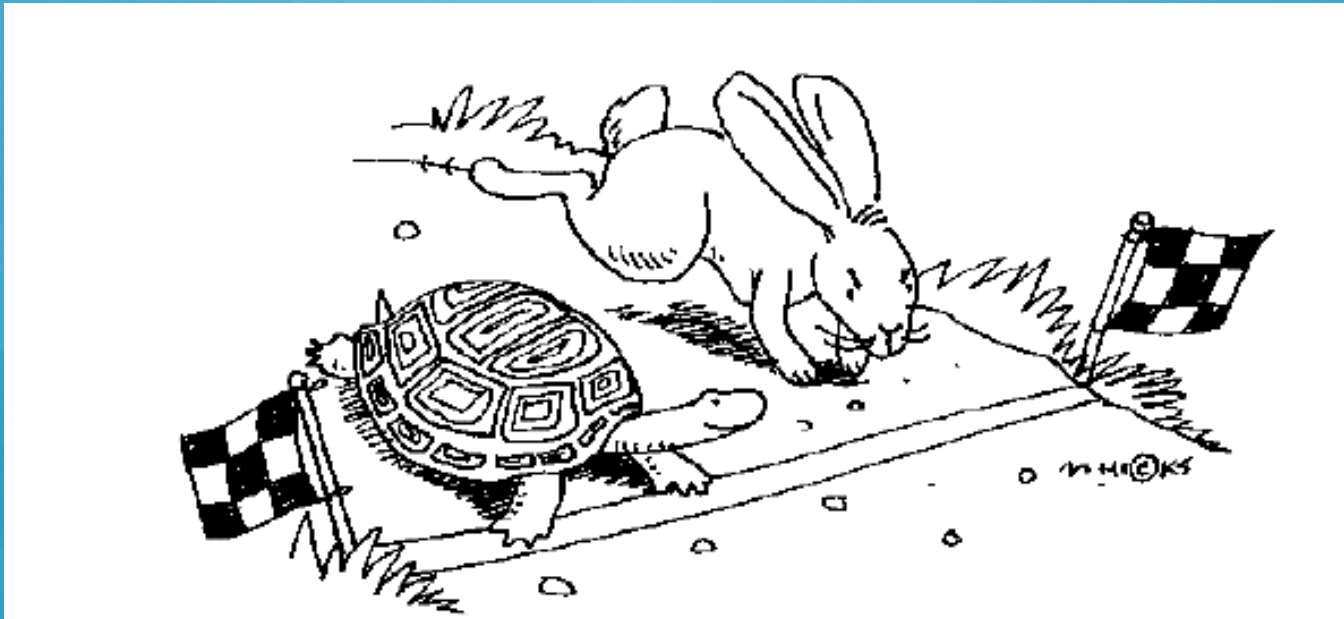


3 types d'échéances

- **Stricte :**
Respect du délai critique : Contrôle aérien,
Gestion du freinage d'une voiture
- **Ferme :**
Respect du délai essentiel : Transaction
Boursière
- **Molle :**
Pertinence du résultat décroissante après
l'échéance : flux vidéo, VoIP

Mauvaise interprétation

« *Real-time is not real-fast* »





Les Fondamentaux

- **Déterminisme logique**
Les mêmes entrées appliqués au système produisent les mêmes résultats
- **Déterminisme temporel**
Respect des contraintes temporelles (échéances)
- **Fiabilité**
Le système répond à des contraintes de disponibilité (matériel/ logiciel)
- **Comportement doit être prédictible**
Maitrise des temps de latence et de leurs variations



Les systèmes temps réel

- Contrainte de temps :
 - Date de livraison d'un résultat
- Décomposés en tâches
 - Séparation des traitements
 - Meilleure utilisation du processeur
 - Meilleure fiabilité en cas de surcharge



Contraintes pour la réalisation d'applications

- Choix de l'ordre d'exécution des fonctions important
- Déterminisme d'exécution (predictability)
- Langage de programmation adapté :
 - Ada95 (Ce n'est pas une blague...)
 - C/C++
 - Java



Domaines d'applications

- Automobile :
 - Systèmes antiblocages de freins, contrôle moteur
- Contrôle de procédés industriels
- Avionique et spatial
- Base de donnée en temps réel :
 - Cotations boursières, enchères sur internet





Java Real-Time

- Langage non prévu pour le temps-réel
 - Mauvaise granularité (millisecondes)
 - Pas assez de maîtrise de l'ordonnancement
 - Problèmes avec Garbage Collector
 - Pas de déterminisme temporel



Java Real-Time : RTS J

- Real-Time Specification for Java :
 - Spécifications pour pouvoir exécuter des applications temps-réel en Java
 - Réalisé par un groupe d'expert de plusieurs sociétés (Sun, IBM, ...)



Java Real-Time : RTS J

- Pas de restriction à une version de Java
- Compatibilité
- Pas de prérequis de puissance ou de matériel
- Granularité améliorée : nanoseconde



Java Real-Time : RTS J

- Threads :

- Une interface :

- `Schedulable` : objet pouvant être ordonnancé

- Deux nouvelles classes :

- `RealtimeThread`

- `NoHeapRealtimeThread` : thread temps réel sans accès au tas



Java Real-Time : RTS J

- Gestion de la mémoire :
 - Mémoire à portée
 - Mémoire permanente
 - Mémoire à accès direct
 - Mémoire physique
 - Tas
- Pas de pile comme Java Classique

RT JVM

- Les principales :
 - Jrts : Développé par Sun
 - Jrockit : Développé par Oracle
 - FijiVM : Développé par Fiji Systems
 - Metronome : Développé par IBM

Démo : Brève Illustration





Démo

- Algorithme de détection de collision
 - Reçoit périodiquement un flux de positions des avions
 - Chaque tour de boucle crée une liste de mouvement représentant les mouvements d'avions
 - Calcul une liste de collisions
 - Une liste de positions des avions est maintenue à jour