



ni.com

Architecture de programmation et communication dans LabVIEW™ Real-Time

Alexandre Stanurski
National Instruments France

Sommaire

- Architecture de programmation
- Transfert des données
- Benchmarks
- Conclusion

Définitions

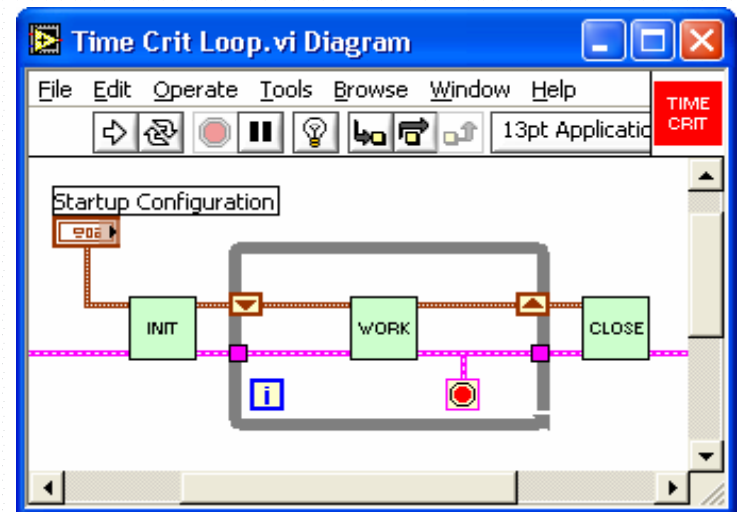
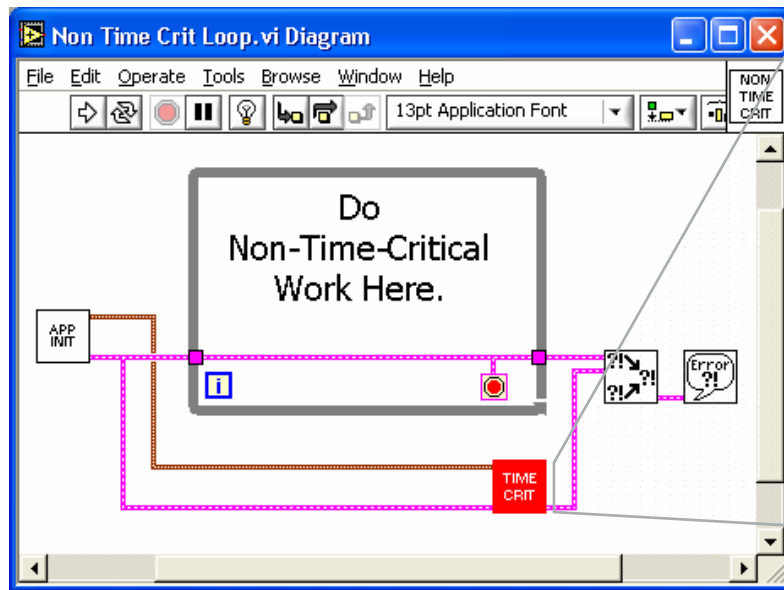
- Déterminisme
- Jitter
- Mécanismes de verrouillage (exclusions mutuelles, sémaphores, sections critiques...)
- Conflits de ressources (bus, mémoire, fichier...)

Objectifs de l'architecture de programmation

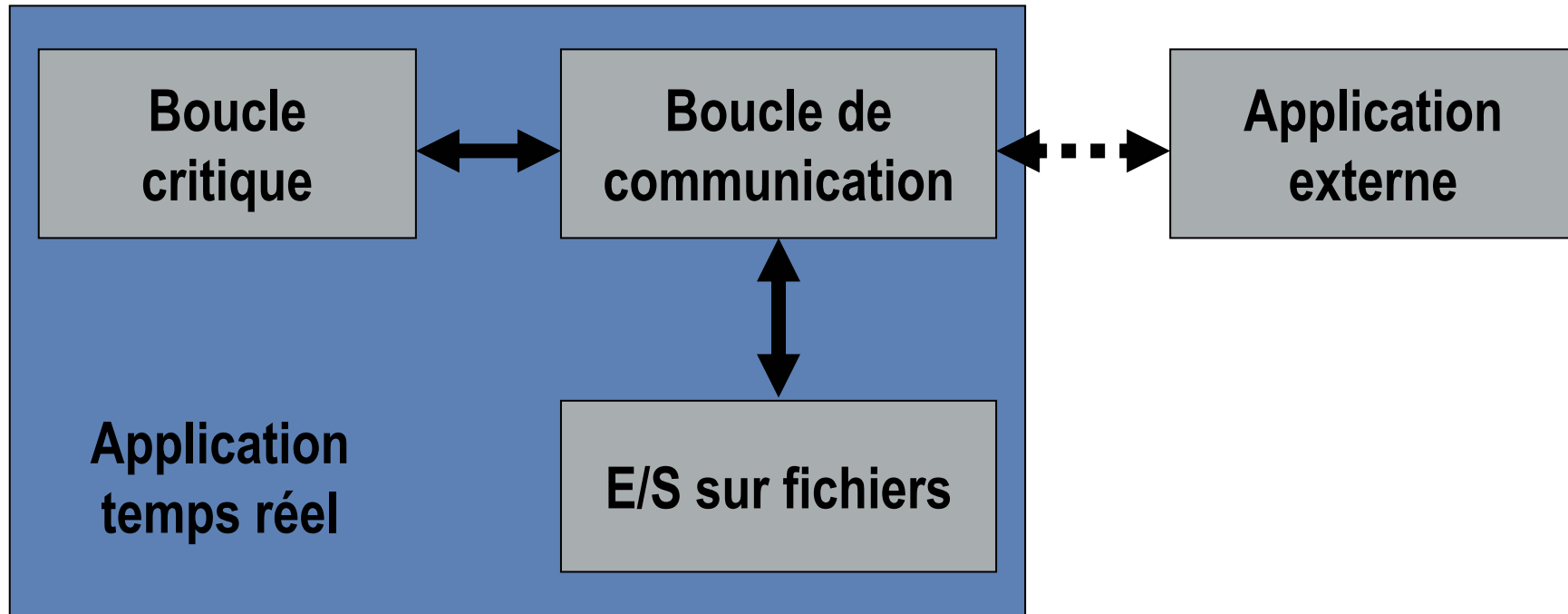
- Optimiser les performances
- Optimiser le déterminisme
- Minimiser le Jitter

Architecture de programmation

- Prévenir les conflits de ressource
- Retirer les opérations non déterministes des boucles critiques

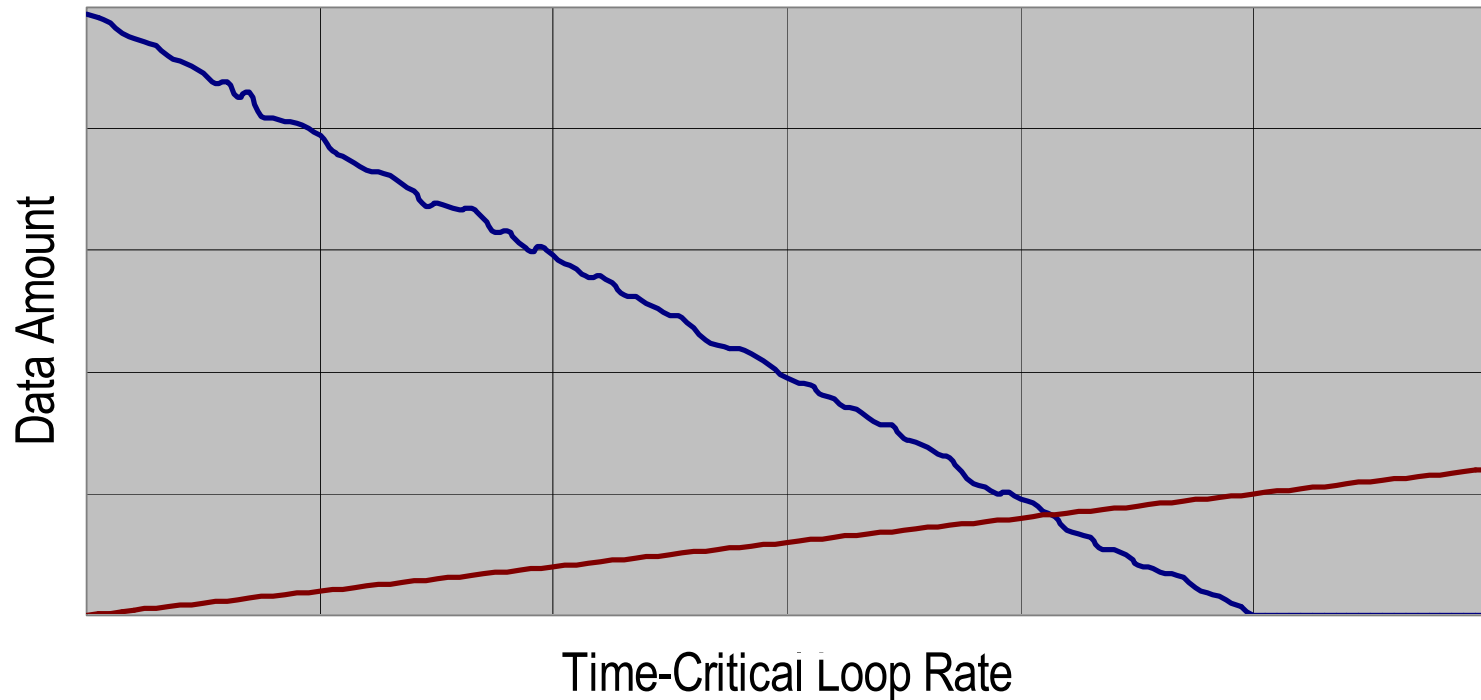


Transfert des données dans une application temps réel

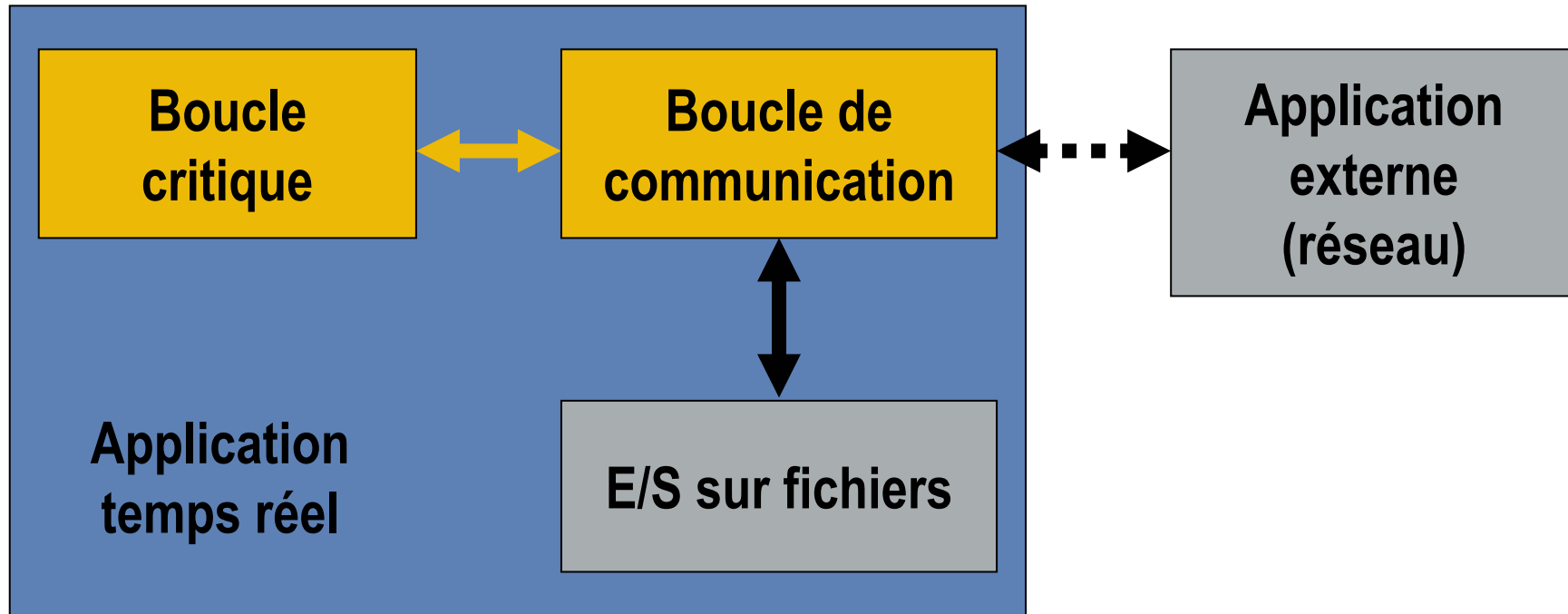


Communication et ressources CPU

— Communication Data
— Time Critical Loop Data



Transfert des données dans une application temps réel



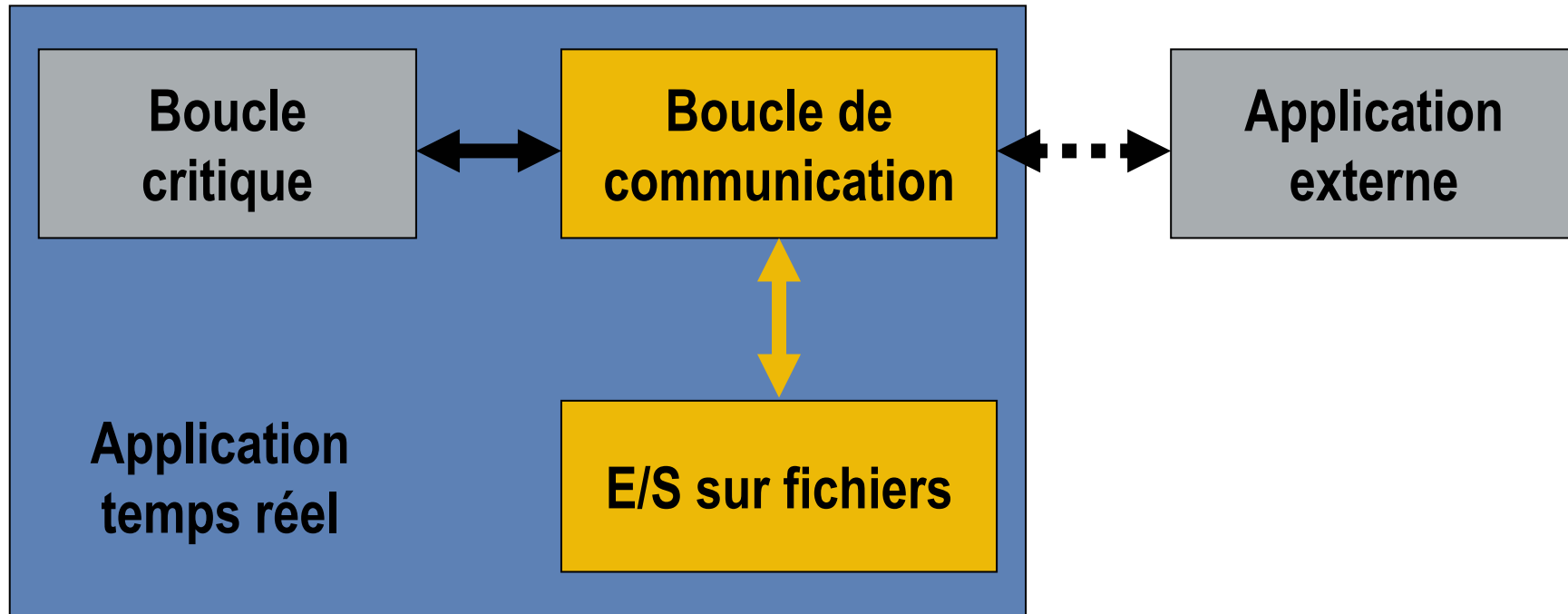
Transfert des données dans une boucle critique

- Non sécurisé
 - Série, GPIB
 - E/S sur fichiers, TCP/IP
 - Files d'attente, notifications, VI Server
- Sécurisé
 - DAQ, VISA (par registres), CAN
 - Variables globales (faibles volumes de données)
 - FIFO temps réel (RT FIFO)

FIFO temps réel

- Liste FIFO avec pertes, de longueur fixe
- Éléments de taille fixe
- *Reader* unique, *writer* unique
- Pas de mécanisme de verrouillage – c'est sécurisé
- Conçu pour l'OS temps réel utilisé
- Variables globales pour des valeurs simples (ex. : paramètres utilisateur)

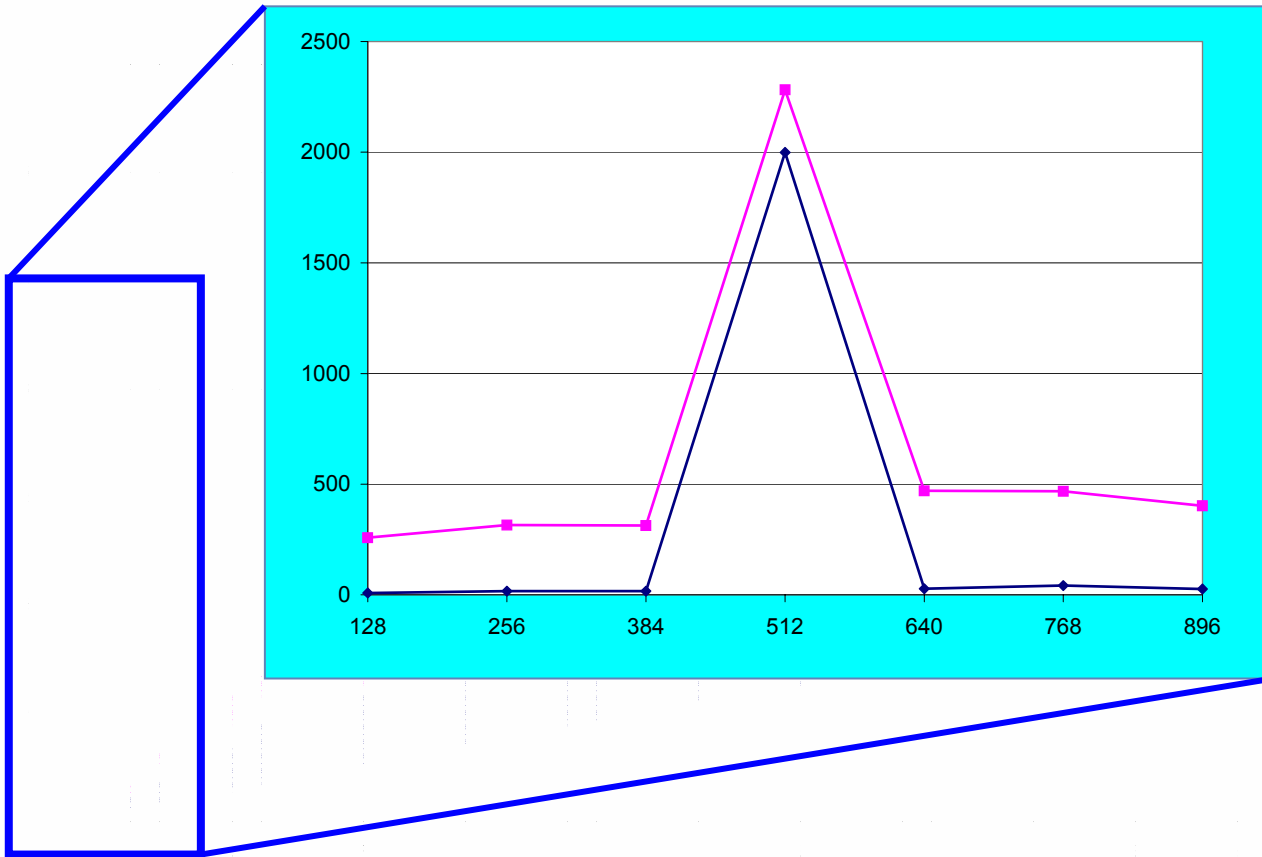
Transfert des données dans une application temps réel



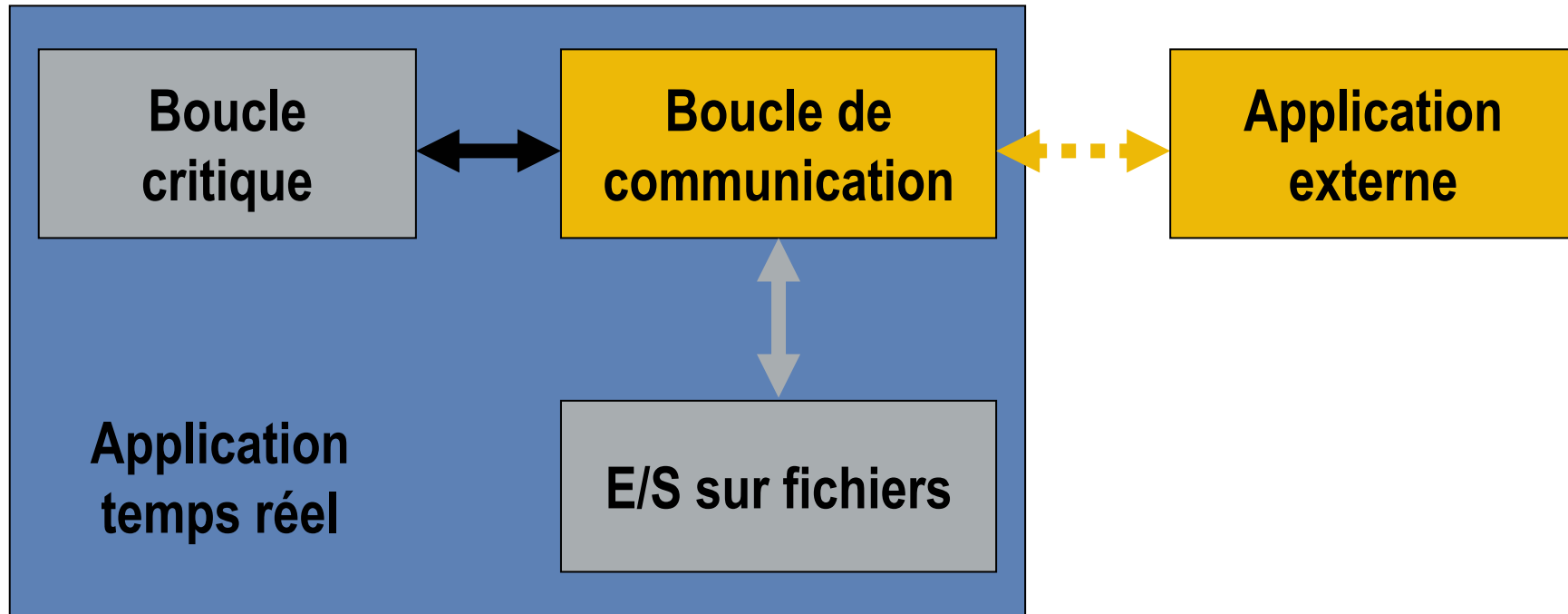
Transfert des données sur fichiers

- Le taux de transfert dépend de la taille des paquets
 - Taille optimale : 512 octets

Taux de transfert des données sur fichiers



Transfert des données dans une application temps réel

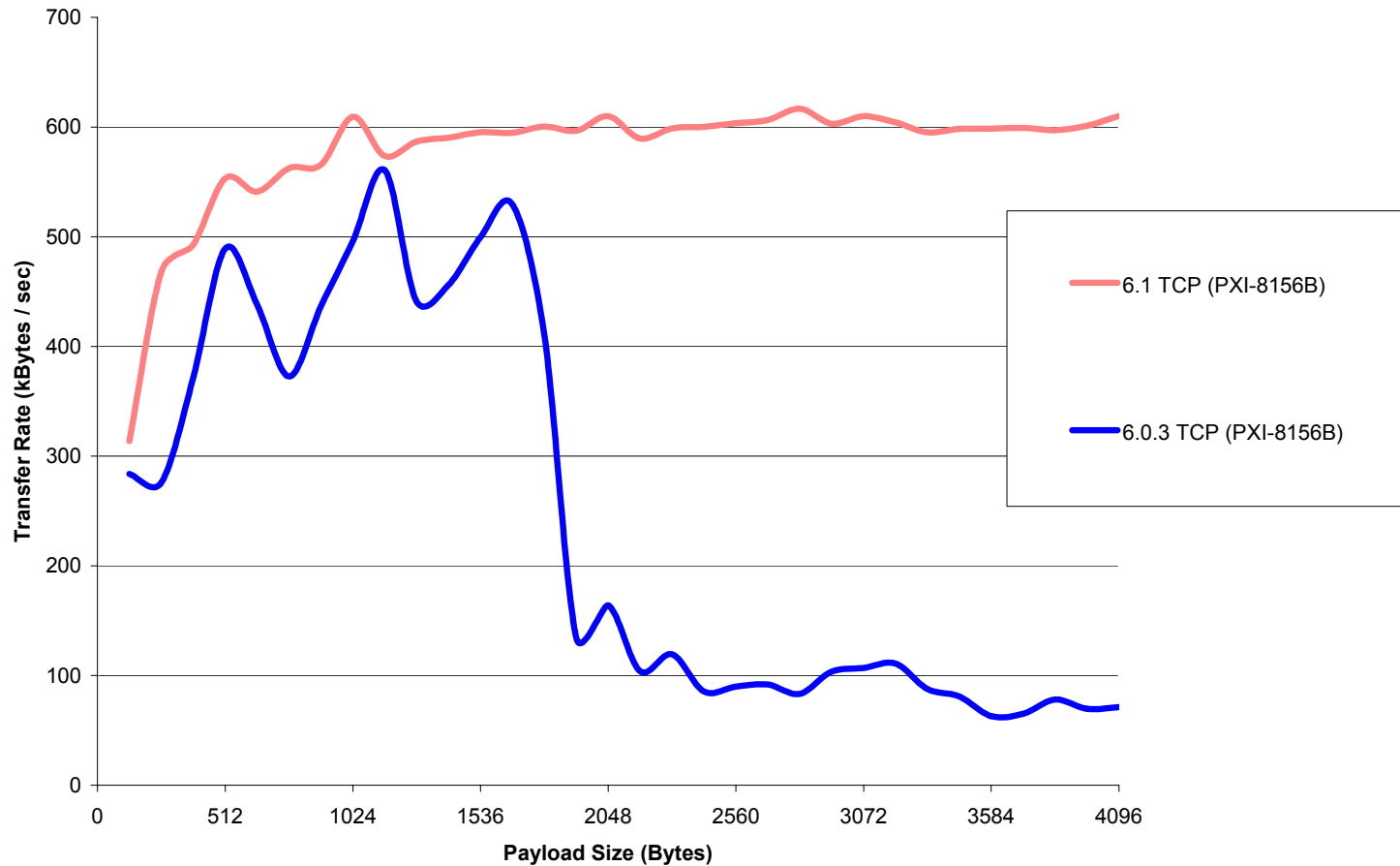


Méthodes de transfert des données à distance

- Série
- CAN : déterministe
- Réseau
 - TCP : sans perte, synchrone, rapide, conversion des données
 - UDP : avec pertes, le plus rapide, conversion des données
 - VI Server : basé sur TCP, plus simple, plus lent
 - DataSocket™ : le plus simple, avec pertes, sans conversion de données

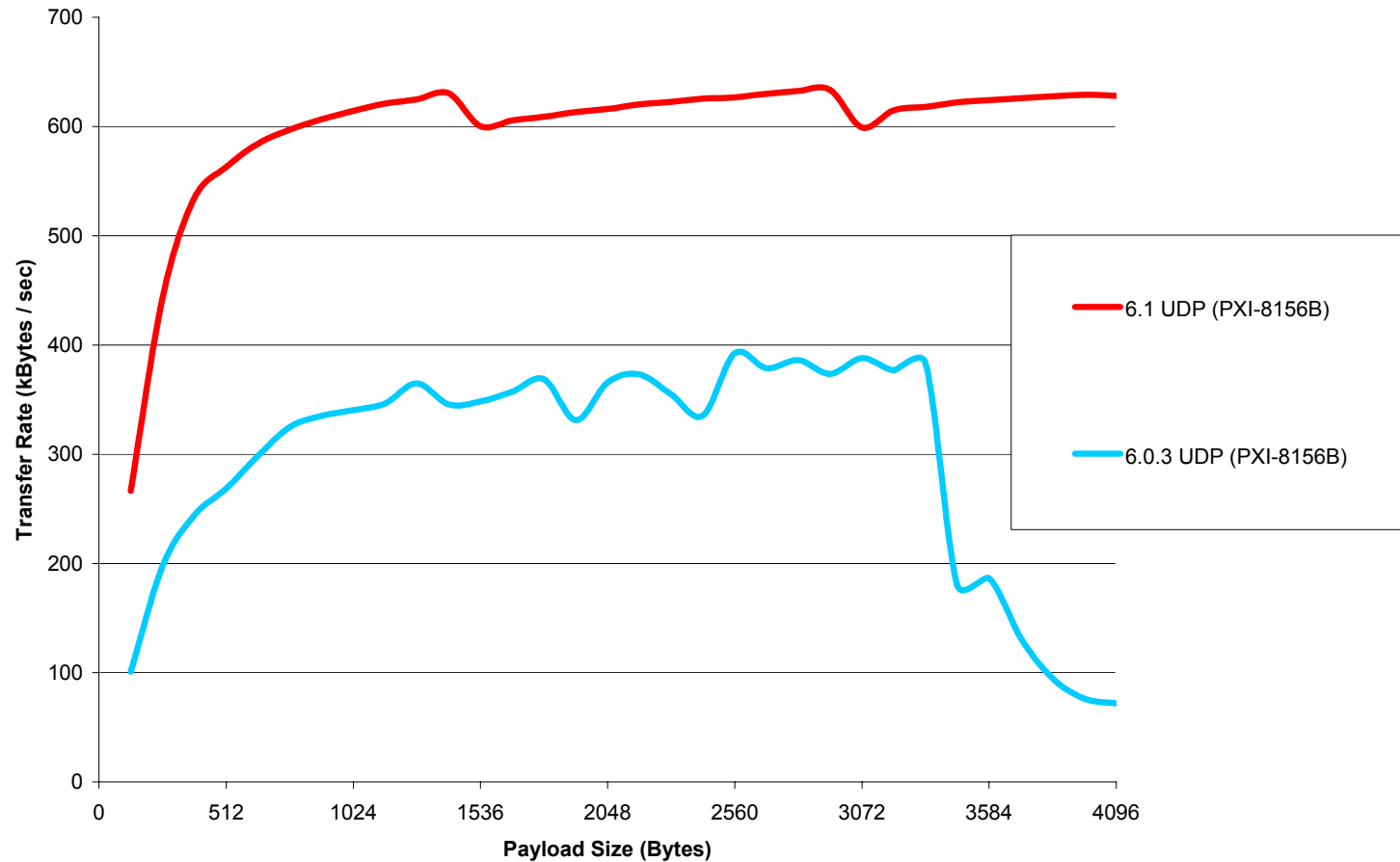
Taux de transfert des données sur le réseau

Maximum Network Transfer Rate



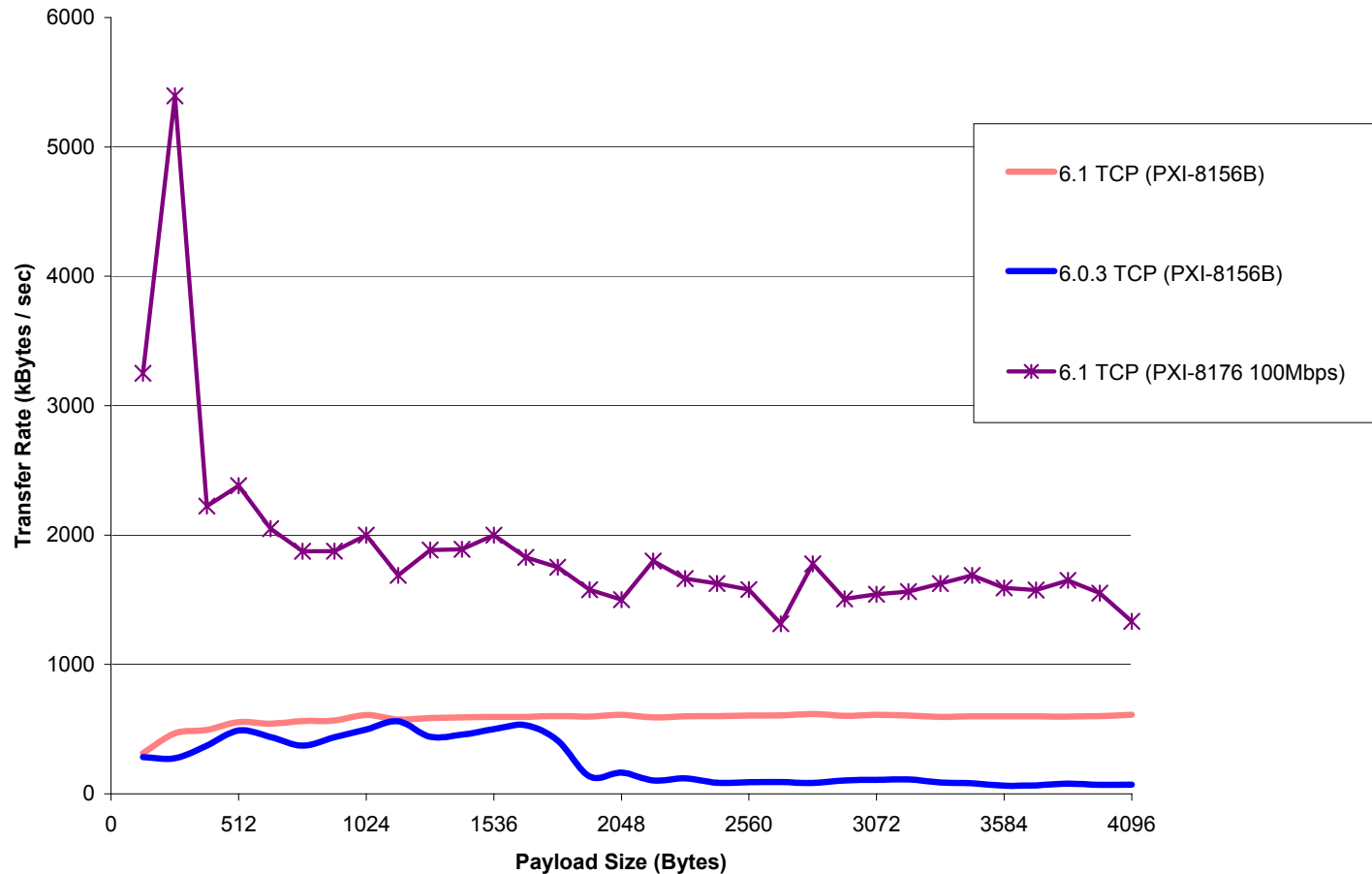
Taux de transfert des données sur le réseau

Maximum Network Transfer Rate

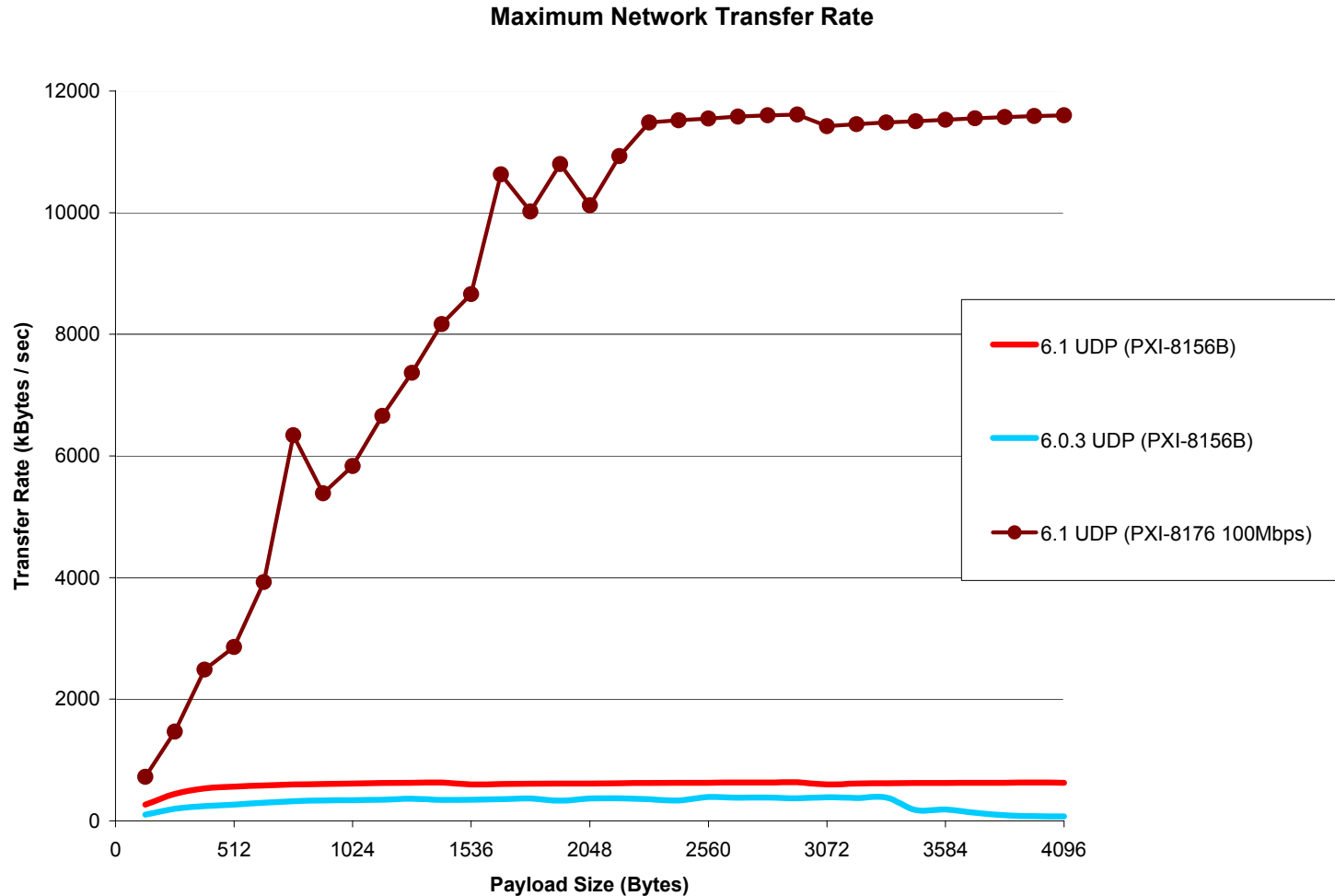


Taux de transfert des données sur le réseau

Maximum Network Transfer Rate

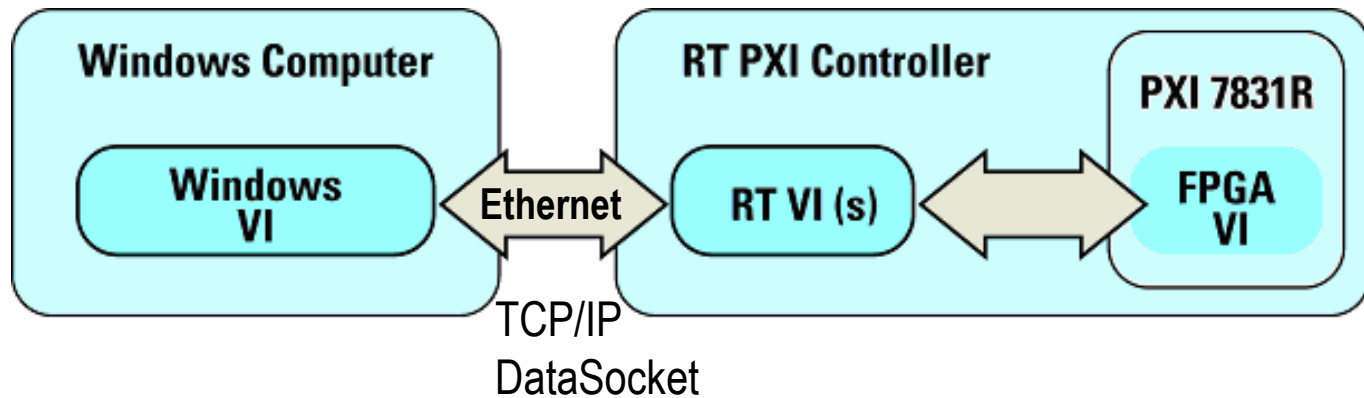


Taux de transfert des données sur le réseau



Nouveau : module LabVIEW 7 FPGA

- Combiné au module LabVIEW 7 Real-Time, il permet d'aller plus loin dans la distribution des tâches critiques et non critiques.
- Communication entre la carte RIO et le contrôleur PXI RT hôte par les *RIO Host Interface VIs*, similaires au VI Server.



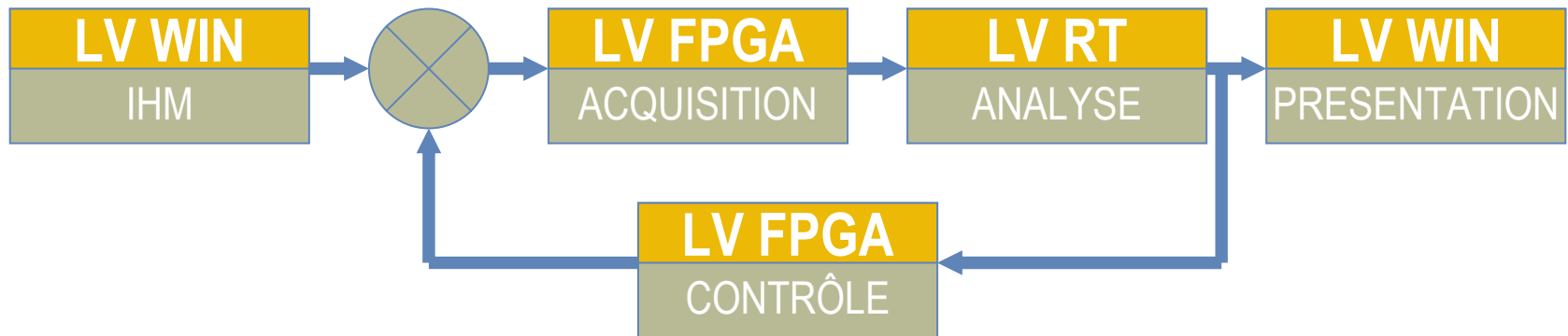
Exemple : applications de régulation

LabVIEW FPGA & PXI-7831R

- E/S synchronisées
- Signaux numériques personnalisés
- Déclenchements personnalisés

LabVIEW Real-Time

- Calculs à virgule flottante
- Taille d'application « illimitée »
- Communication réseaux
- E/S fichiers



Conclusions

- Architecture de programmation recommandée
 - Isoler les tâches critiques
 - Prévenir les conflits de ressources
- Méthodes de transfert des données
 - FIFO temps réel dans l'application temps réel
 - Paquets de 512 octets pour les opérations d'E/S sur fichiers
 - Choisir la méthode de transfert des données sur le réseau qui convient le mieux à votre application
- Notion de distribution des tâches Windows/Real-Time/FPGA

Questions ?