

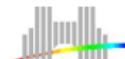
# Soutenance de Thèse

## Analyses statistiques des communications sur puce

**Antoine Scherrer**

LIP - ENS Lyon  
Equipe Compsys

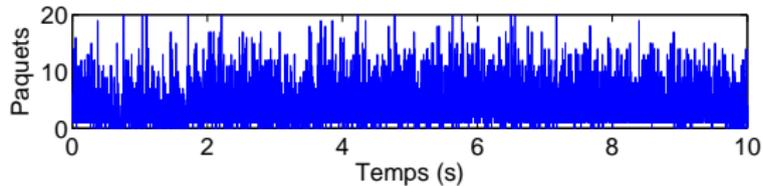
11 décembre 2006



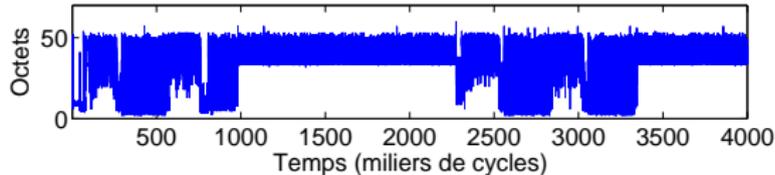
# Problématiques

## ► Modélisation et synthèse de trafic

- Trafic Internet (Partie 1)



- Trafic sur puce (Partie 2)



## ► Outil commun

- Processus stochastiques

# Partie 1

## Analyse et synthèse de trafic Internet

# Plan de la première partie

---

- **Motivations**
- **Modèle de trafic Internet**
- **Synthèse de processus LRD non-gaussiens**
- **Applications**
- **Conclusion**

# Motivations

## ► Modélisation de trafic

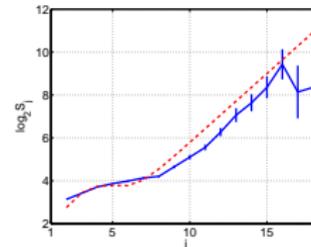
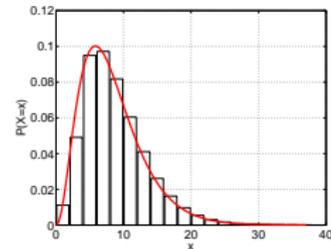
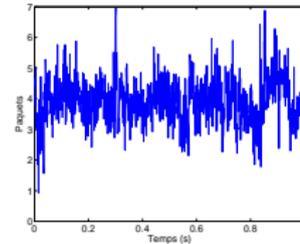
- Caractériser la **variabilité** du trafic
- Évaluation de performance

## ► Synthèse de trafic

- Simulations de trafic **réalistes**

## ► Outil : Processus stochastiques

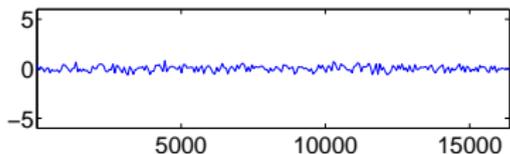
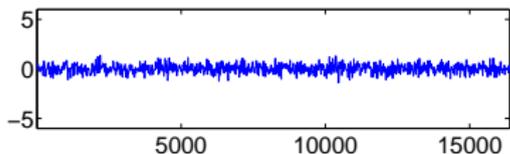
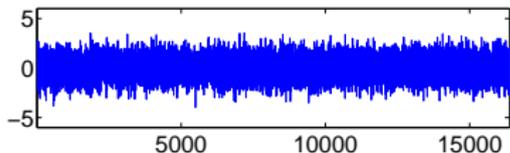
- Distribution marginale
- Covariance (**longue-mémoire**)



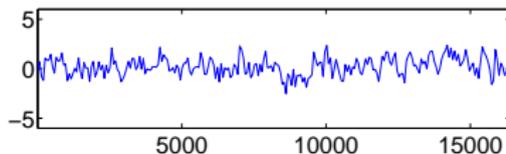
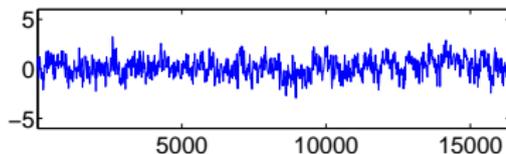
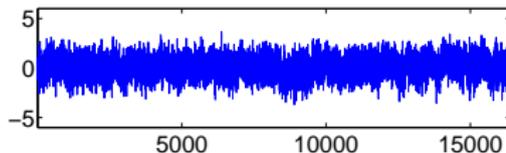
# Longue mémoire (LRD)

- ▶ Décroissance lente de la fonction de covariance
- ▶ Rafale à toute les échelles
- ▶ **Impact important sur les performances des réseaux**

Pas de LRD ( $H = 0.5$ )



LRD ( $H = 0.8$ )



$\Delta = 1$

$\Delta = 16$

$\Delta = 64$

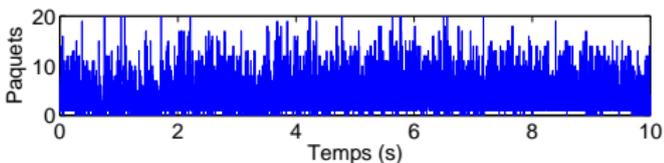
# Plan

---

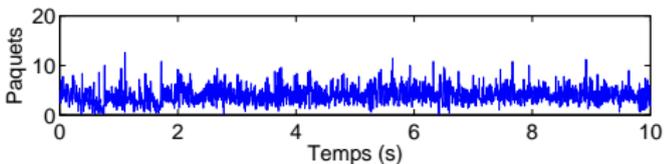
- Motivations
- **Modèle de trafic Internet**
- Synthèse de processus LRD non-gaussiens
- Applications
- Conclusion

# Trafic Internet

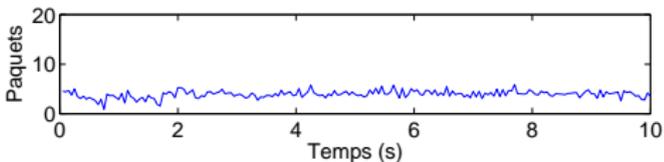
- ▶ Séries temporelles de débit agrégé
- ▶ Exemple : Trace Auckland IV



$$\Delta = 1\text{ms}$$



$$\Delta = 5\text{ms}$$



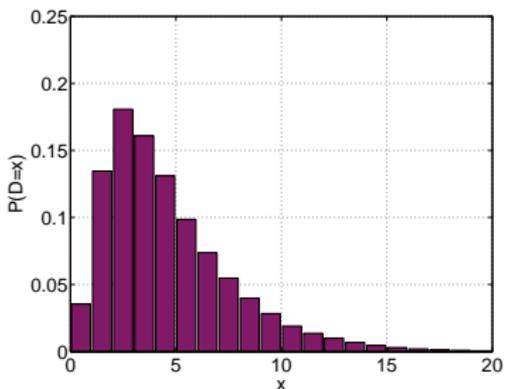
$$\Delta = 50\text{ms}$$

# Propriétés statistiques de ces traces

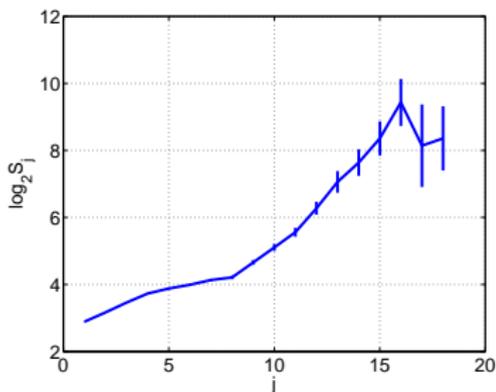
## ► Stationnarité

- Pas de changement abrupte dans la trace

## ► Marginales non-gaussiennes



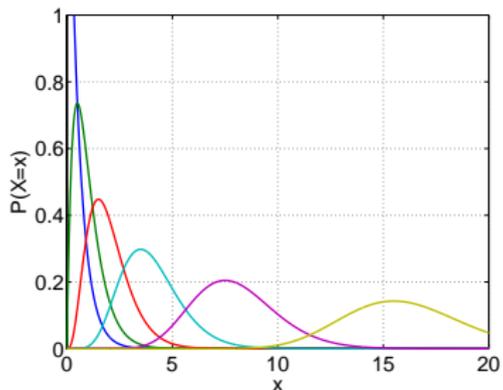
## ► Longue mémoire



# Modèle proposé

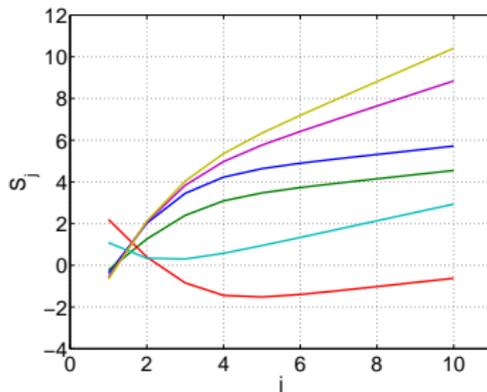
## ► Marginales : loi Gamma

- Variables aléatoires positives
- $\Gamma_{\alpha,\beta} + \Gamma_{\alpha',\beta} = \Gamma_{\alpha+\alpha',\beta}$
- $c\Gamma_{\alpha,\beta} = \Gamma_{\alpha',\beta'}$
- **Adapté à l'agrégation**



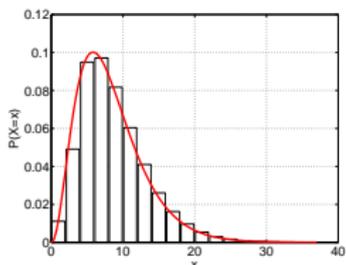
## ► Covariance : FARIMA

- Dépendances courtes :  $\Phi$  et  $\Theta$
- Longue mémoire :  
 $d = H - 1/2$
- **Adapté aux données**

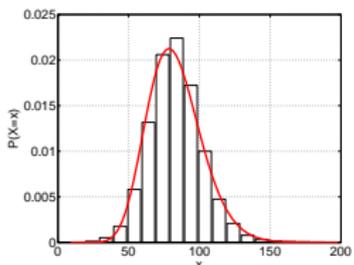


# Trace Auckland IV-d1 (WAN)

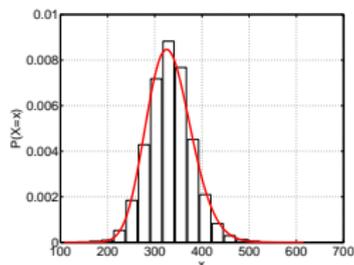
$\Delta = 10\text{ms}$



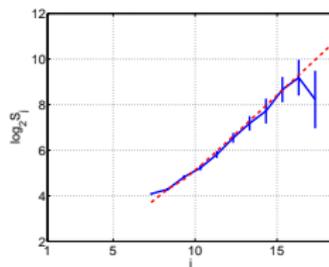
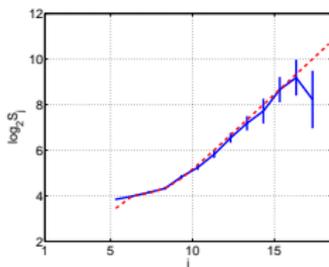
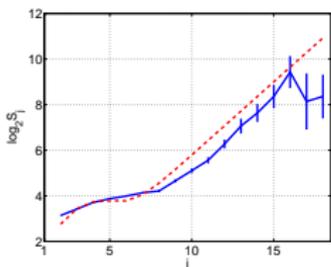
$\Delta = 100\text{ms}$



$\Delta = 400\text{ms}$



Distribution de probabilités



Covariance

# Plan

---

- Motivations
- Modèle de trafic Internet
- **Synthèse de processus LRD non-gaussiens**
- Applications
- Conclusion

# Définition du problème

---

## ► Objectif

- Obtenir des traces de trafic synthétiques

## ► Que sait-on faire ?

- Synthétiser un processus **Gaussien** ayant une **covariance donnée**
- Obtenir un processus  $Y$  suivant différentes lois à partir de processus gaussiens  $X_i : Y = F(X_i)$

## ► Que veut-on obtenir ?

- Un processus  $Y$  suivant une distribution marginale **et** une covariance données

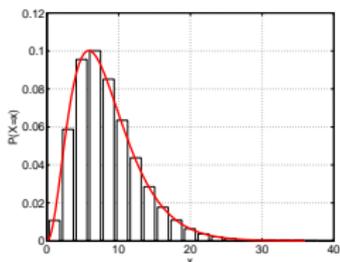
# Méthode proposée

---

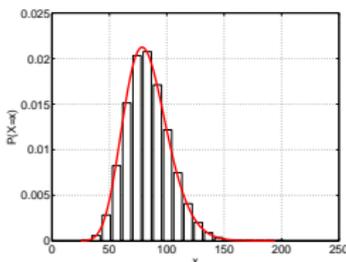
- 1 **La relation  $Y = F(X_i)$  nous permet d'établir une relation entre  $\gamma_X$  et  $\gamma_Y$** 
  - $\gamma_X(k) = g(\gamma_Y(k))$
  - Exemple pour la loi gamma ( $\Gamma$ ) :  $\gamma_X(k) = \sqrt{\frac{\gamma_\Gamma(k)}{\alpha\beta^2}}$
- 2 **On peut alors synthétiser des processus indépendants gaussiens  $X_i$  de covariance  $\gamma_X(k)$**
- 3 **L'application de la fonction  $F$  donnera le processus  $Y$  recherché**
  - ▶ **Distributions implémentées**
    - Gamma, exponentielle,  $\chi^2$ , pareto, lognormale et uniforme

# Résultats

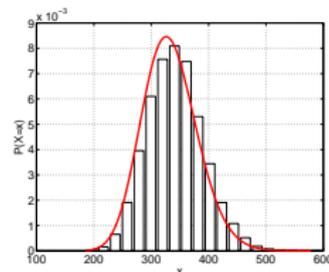
$\Delta = 10\text{ms}$



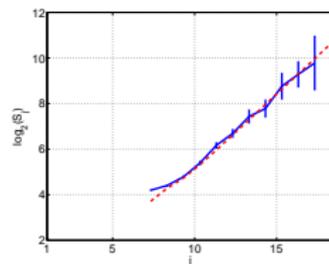
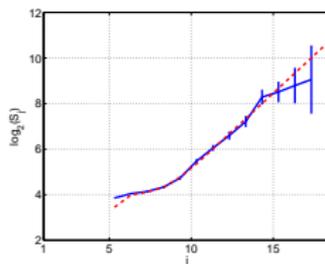
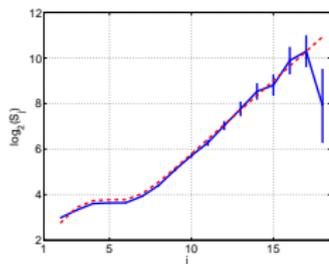
$\Delta = 100\text{ms}$



$\Delta = 400\text{ms}$



Distribution de probabilités



Covariance

# Plan

---

- Motivations
- Modèle de trafic Internet
- Synthèse de processus LRD non-gaussiens
- **Applications**
- Conclusion

# Détection d'anomalie

## ► Objectif

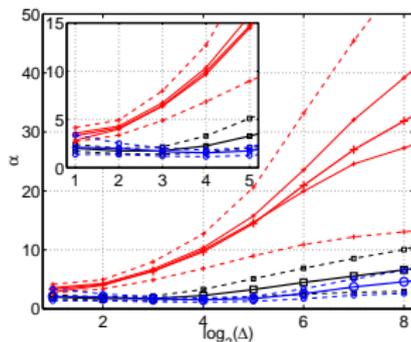
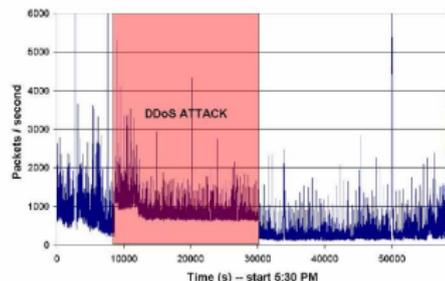
- Détecter une anomalie dans le trafic

## ► Proposition

- **Modèle multi-résolution**
- Estimation par blocs
- Détection des **changements**

## ► Résultats

- + Détection des attaques mais pas des variations légitimes
- Niveau d'agrégation initial faible



# Plan

---

- Motivations
- Modèle de trafic Internet
- Synthèse de processus LRD non-gaussiens
- Applications
- **Conclusion**

# Conclusion de la première partie

---

## ► Contributions

- **Modèle** de trafic Internet (Gamma-Farima)
- **Procédure de synthèse** de processus à longue mémoire non-gaussiens
- Technique de **détection d'anomalie**
- Papier de synthèse à paraître dans *IEEE TDSC*

## ► Collaborations

- Laboratoire de physique de l'ENS de Lyon (équipe SISYPHE)
- Laboratoire LAAS (Toulouse)
- Projet MetroSec

## ► Perspectives

- Extension de la méthode de synthèse
- Utilisation du générateur de trafic

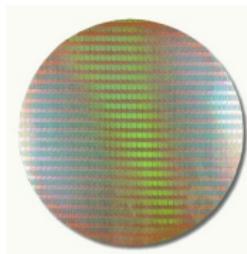
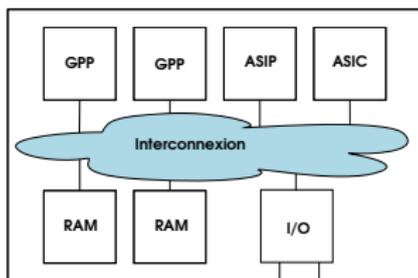
## Partie 2

# Analyse et synthèse des communications sur puce

# Contexte

## ► Système sur puce (SoC)

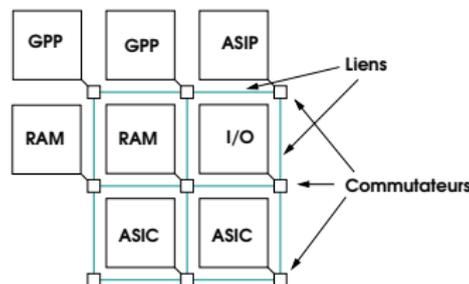
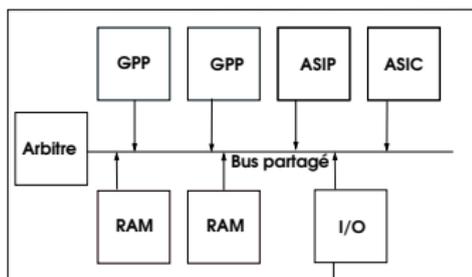
- Assemblage de composants **sur une même puce de silicium**
- Complexité croissante
- Temps de simulation très long
- Environnement **SocLib**



# Réseaux sur puce (NoC)

## ► Pourquoi des réseaux sur puce ?

- Pour satisfaire les demandes des IPs
- Pour diminuer la consommation électrique
- Passage en **commutation de paquets**



## ► Notre objectif : Évaluation et prototypage du NoC

# Plan

---

- **Introduction**
- **Génération de trafic sur puce**
- **Génération de TG**
- **Segmentation**
- **Autres résultats**
- **Conclusion**

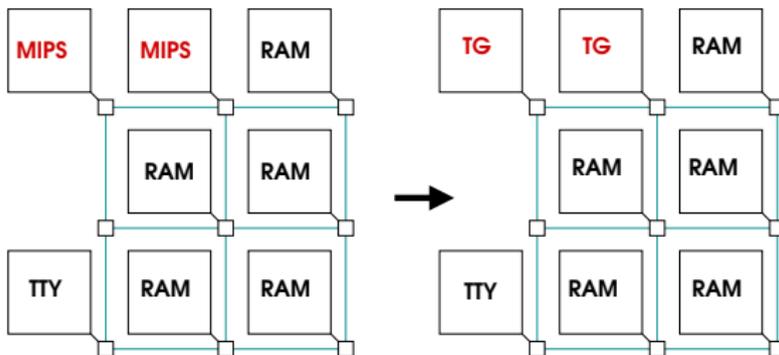
# Plan

---

- Introduction
- **Génération de trafic sur puce**
- Génération de TG
- Segmentation
- Autres résultats
- Conclusion

# Génération de trafic sur puce

- Notre approche : remplacer les composants par des générateurs de trafic (TG)



- Pourquoi ?

- **Évaluation flexible des performances**
- Composant non-disponible
- Gain de temps de simulation

# Différents types de TG

---

## ▶ Ad-hoc - Écrit par un concepteur

- + Très précis
- Pas adapté aux processeurs

## ▶ TG déterministe - Rejeu (*Loghi 2004, Mahadevan 2005*)

- + Précis
- Limité par la trace de référence

## ▶ TG stochastique (*Wiklund 2004, Lahiri 2001*)

- + Modélisation
- Précis en moyenne

## ▶ Notre approche

- + **Multi-phase**
- + **Stochastique avec LRD**
- + **CABA**

# Plan

---

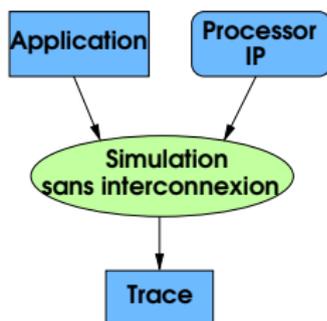
- Introduction
- Génération de trafic sur puce
- **Génération de TG**
- Segmentation
- Autres résultats
- Conclusion

# Flot de génération (1)

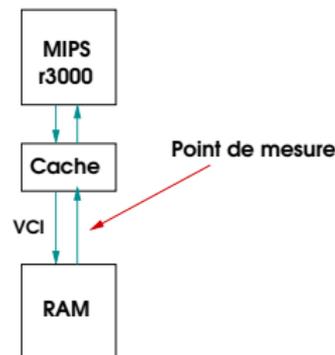
## ► Obtention de la trace de référence

- Simulation rapide
- Extraction au **trafic intrinsèque** d'un composant

### Méthodologie

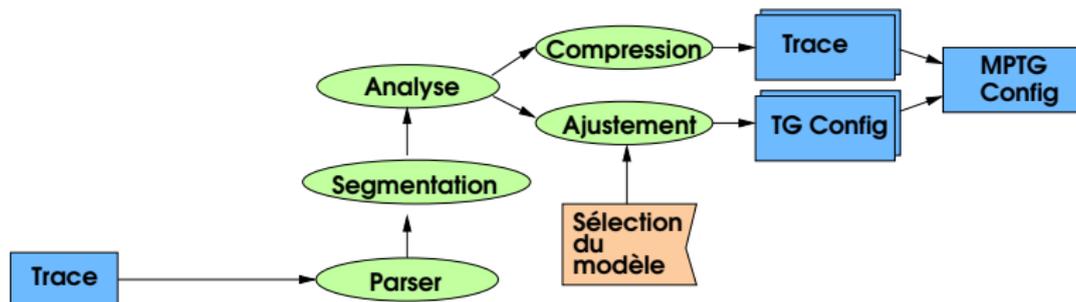


### Plateforme SocLib



## Flot de génération (2)

- Configuration semi-automatique du générateur de trafic
  - Nombreux développements *open-source*

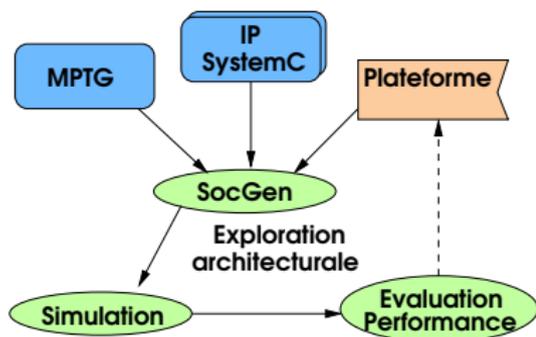


## Flot de génération (3)

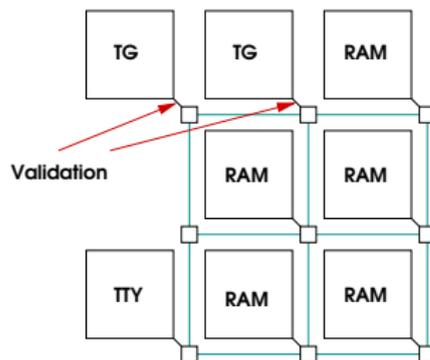
### ► Évaluation de performances

- Validation du flot
- Dimensionnement des FIFOs

#### Méthodologie

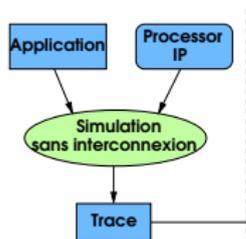


#### Plateforme SocLib

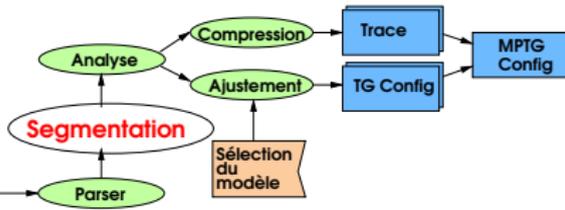


# Flot complet

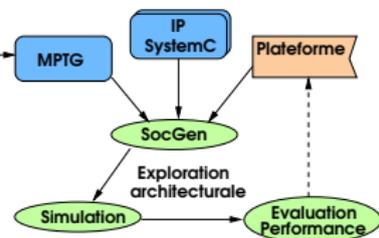
## Simulation de référence



## Configuration MPTG



## Evaluation de performance



## ► Environnement complet pour le prototypage de NoC

- Intègre l'état de l'art
- Ajoute de nouvelles approches

# Plan

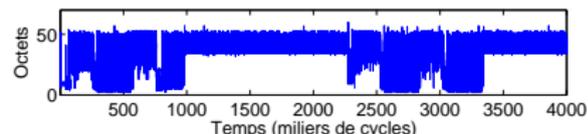
---

- Introduction
- Génération de trafic sur puce
- Génération de TG
- **Segmentation**
- Autres résultats
- Conclusion

# Principe

## ► Objectif

- Identifier des **phases** de trafic **stationnaires**



## ► Approche inspirée par les travaux de Calder et al.

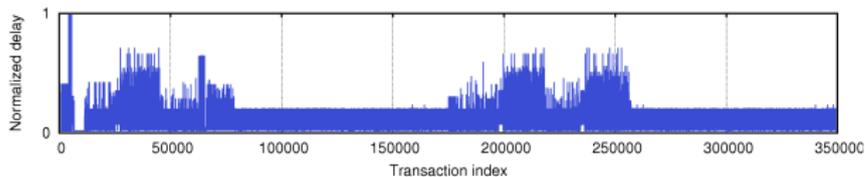
- Évaluation de performance des architectures de processeur

## ► Notre algorithme [CODES 2006]

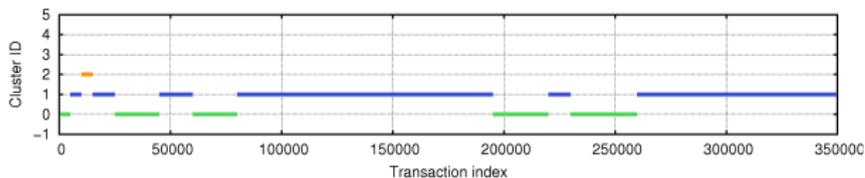
- 1 **Découper** la trace en blocs
- 2 **Calculer** des statistiques sur chaque bloc
- 3 **Partitionner** ces points en  $k$  ensembles ( $k$ -means)

- ⇒ Les blocs ayant un comportement **statistique proche** forment une phase

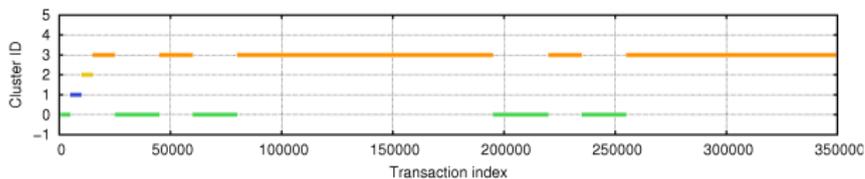
# Exemple : MP3



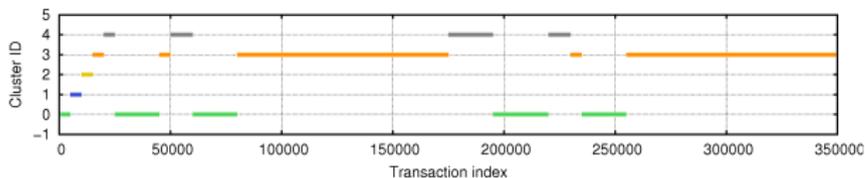
Trace



3 phases



4 phases



5 phases

# Plan

---

- Introduction
- Génération de trafic sur puce
- Génération de TG
- Segmentation
- **Autres résultats**
- Conclusion

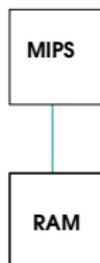
# Simulations effectuées

## ► Applications multimédia

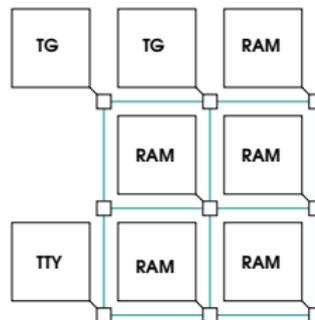
- **Images** : M-JPEG (Multi-thread), JPEG, JPEG 2000
- **Audio** : MP3
- **Vidéo** : MPEG-2

## ► Plateformes

### Collecte



### Validation



# Résultats

---

- ▶ **Temps de simulation** [ASAP 2006]
  - Le gain n'est **pas important** (x2)
  - La valeur ajoutée est dans la **souplesse de l'outil**
  
- ▶ **Validation** [ASAP 2006]
  - **Génération multi-phase stochastique**
  
- ▶ **Longue-mémoire** [NOC 2007 (en cours de relecture)]
  - **Pas de présence systématique**
  - Impact sur le NoC fonction du protocole de bas niveau

# Plan

---

- Introduction
- Génération de trafic sur puce
- Génération de TG
- Segmentation
- Autres résultats
- **Conclusion**

# Conclusion

---

## ► Contributions

- **Mise au point** et **développement** d'un environnement intégré de génération de trafic sur puce
- Algorithme de **découpage en phase**
- **Validation de l'approche** par des simulations dans **SocLib**
- Évaluation de l'impact de la **longue mémoire** sur les NoC

## ► Perspectives

- Étudier le trafic issue d'une **IP dédiée**
- Intégrer le flot dans le projet SocLib
- Étude de cas de dimensionnement de NoC

# Conclusion générale

---

## ▶ Idée principale défendue

- Modélisation **aléatoire** de **traces** de **trafic**

## ▶ Méthodologie

- Allers-retour constant entre **théorie** et **pratique**

## ▶ Perspectives en cours

- Utilisation du **générateur de trafic Internet** au LAAS
- Intégration du flot **MPTG dans SocLib**
- Modélisation de la **mobilité des individus** (ARES)

