Modélisation matricielle de l'auto-stabilisation

Bertrand Ducourthial
UMR CNRS 6599 HEUDIASYC
Université de Technologie de Compiègne
Bertrand.Ducourthial@hds.utc.fr

Juin 2002

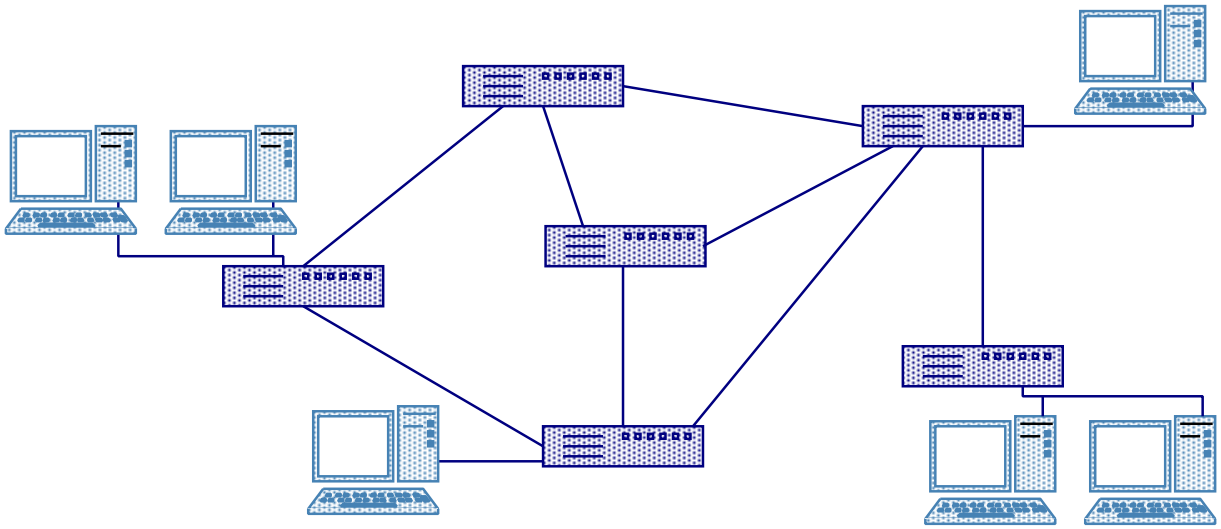
Résumé

Dans cet exposé, nous montrons comment certains algorithmes répartis auto-stabilisants peuvent être étudiés *via* une équation de point fixe matricielle.

Sommaire

1. Auto-stabilisation et convergence

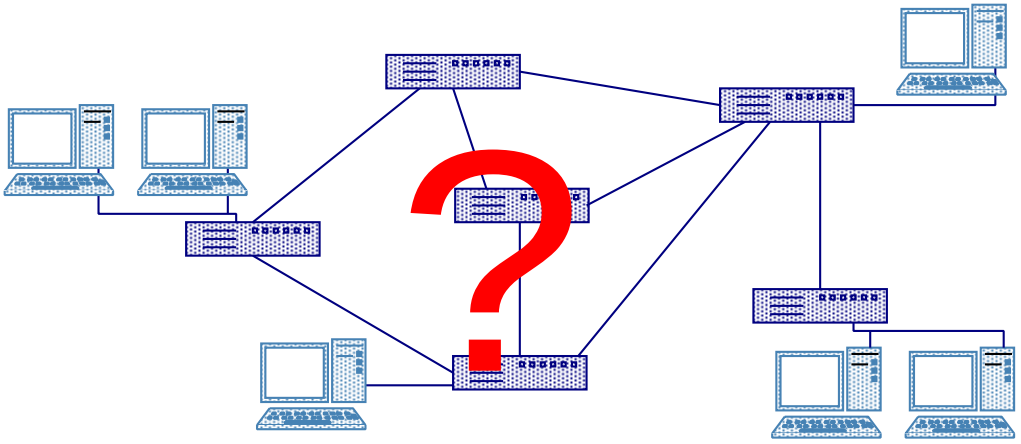
1.1 Etude d'un algorithme réparti



- Réseau
  topologie, liens de communication (registres), *etc.*
- Processeurs
  traitements + communications
- Algorithme local
  liste des traitements locaux
- Algorithme réparti
  ensemble des traitements locaux interagissant
- Système réparti
  réseau + algorithme réparti + hyp. de synchronisation

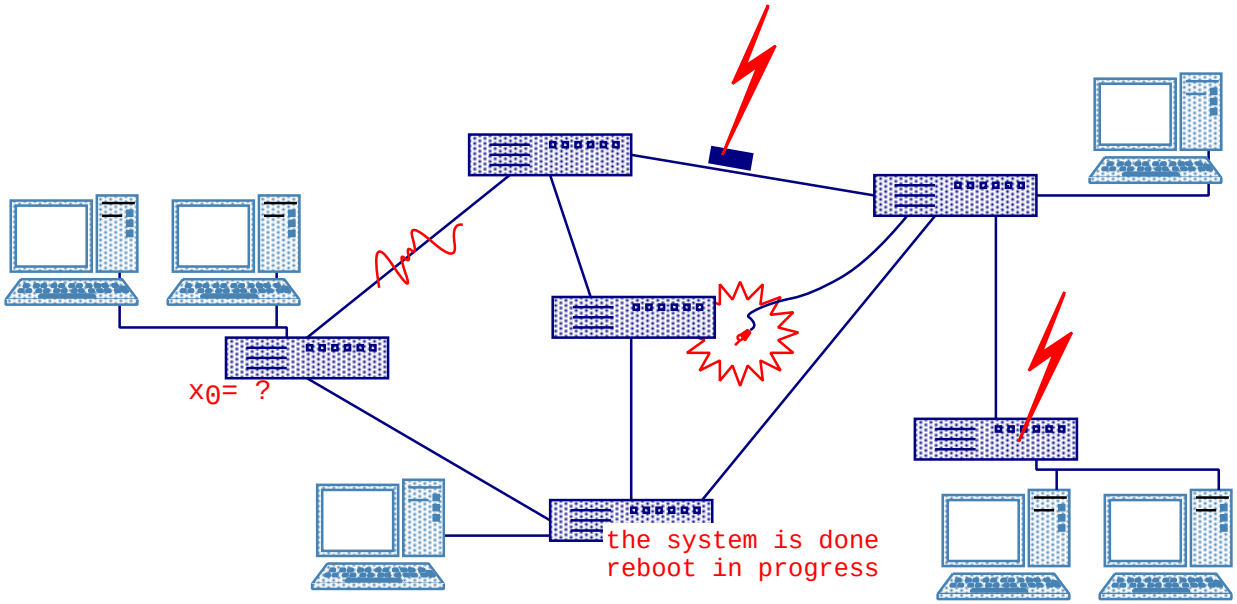
  un algorithme peut être correct avec certaines hypothèses de synchronisation, et faux avec d'autres.

1.2 Auto-stabilisation



- Fautes temporaires
  - faute $\rightsquigarrow$ modification d'une donnée
  - registres, RAM susceptibles d'être corrompues
  - ROM et code non altérables
    - algorithme cablé
    - code réinitialisé depuis une ROM
- Auto-stabilisation
  - une fois les fautes terminées, le système retrouve un état correct en temps fini [Dijkstra 74, Schneider 93]

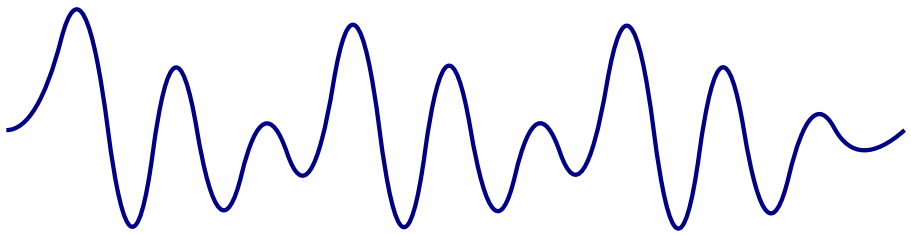
1.3 Applications



- Applications de l'auto-stabilisation
  - corruption de données
  - débranchement/rebranchement
  - redémarrage d'une machine
  - suppression des initialisations
  - adaptation aux changements dans le réseau...

1.4 Convergence

- Temps de convergence
  - valeur erronée sur un nœud
    $\rightsquigarrow$ tous ses descendants peuvent être affectés
  - délai de correction
  - exemple calcul de forêt :
    - si absence de circuits, $O(D)$
    - topologie quelconque $O(D + |\mathbb{S}|)$
- Problème



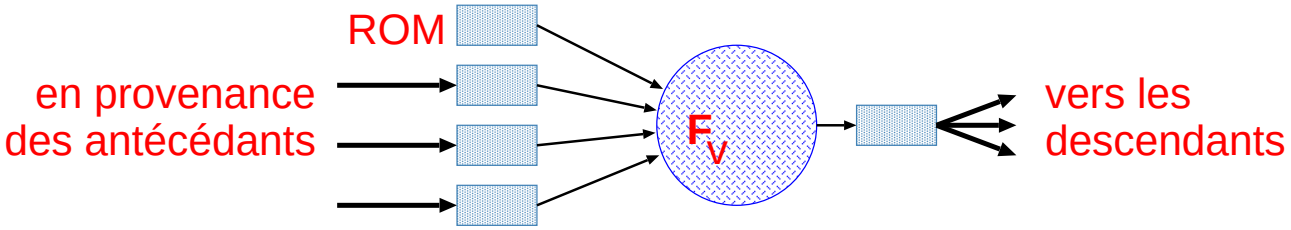
- Solutions
  - tolérance aux défaillances :
    fréquence des fautes
  - algorithmes adaptatifs :
    $\rightsquigarrow$ moyennes mobiles, seuil d'alerte, *etc.*

Etude de la convergence globale ?

2. Modélisation des calculs locaux

2.1 Algorithme local

- Type d'algorithme
  - tâches statiques :
    le résultat global est une configuration souhaitée
  - calcul de distance, de tâbles de routage, de forêts de diffusion...
  - sur chaque processeur, stabilisation vers un résultat local
- Algorithme local

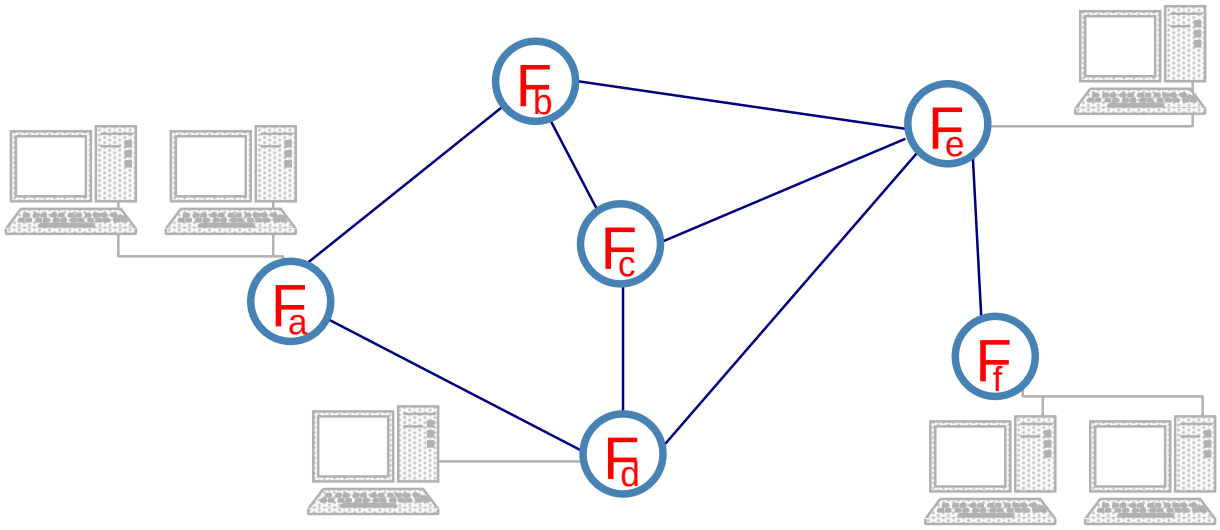


  - sortie $\leftarrow \mathbf{F}\,(\text{ROM}, \text{entrées})$
  - opérateur binaire : $\text{reg}_{\text{out}} \leftarrow \text{reg}_{\text{in}}^{0} \diamond \cdots \diamond \text{reg}_{\text{in}}^{\delta_v^-}$

2.2 Hypothèses de synchronisation

- Démon synchrone
  tous les processeurs actifs au même moment et
  - lisent les valeurs de tous leurs antécédants
  - calculent un nouveau résultat
  - écrivent le résultat dans le registre de sortie
- Démon réparti
  certains processeurs actifs et
  - lisent les valeurs de tous leurs antécédants
  - calculent un nouveau résultat
  - écrivent le résultat dans le registre de sortie
- Démon totalement réparti
  certains processeurs sont actifs et
  - lisent les valeurs de tous leurs antécédants
  - calculent un nouveau résultat
  - écrivent plus tard le résultat
- Démon lecture-écriture
  un processeur est actif et lis ou écrit

2.2 Quel opérateur ?



- Paramètres de l'algorithme (*i.e.* opérateurs F) ?
  - tâche silencieuse
    i.e., convergence vers une configuration
  - auto-stabilisante
    i.e., convergence en dépis des pannes
  - conforme aux spécifications
    - distance, forêt des plus court chemins
    - forêt des chemins les plus fiables
    - forêt des meilleurs débits
    - arborescence en profondeur
    - liste représentative des antécédants
    - . . .

2.3 r-opérateurs

- s-opérateur $\oplus$ (infimum)

associatif	$(x$	$\oplus$	$y)$	$\oplus$	z	$= x \oplus (y \oplus z)$
commutatif	x	$\oplus$	y	$=$	$y \oplus x$	
idempotent	x	$\oplus$	x	$=$	x	
élément neutre	x	$\oplus$	$e_\oplus$	$=$	x	

- r-opérateur [SIROCCO98]

  $\triangleleft$ est un r-opérateur sur $\mathbb{S}$ s'il existe une bijection $\mathsf{r} : \mathbb{S} \to \mathbb{S}$ telle que $\triangleleft$ is :

r-associatif	$(x$	$\triangleleft$	$y)$	$\triangleleft$	$\mathsf{r}(z)$	$= x \triangleleft (y \triangleleft z)$
r-commutatif	$\mathsf{r}(x)$	$\triangleleft$	y	$=$	$\mathsf{r}(y) \triangleleft x$	
r-idempotent	$\mathsf{r}(x)$	$\triangleleft$	x	$=$	$\mathsf{r}(x)$	
neutre à droite	x	$\triangleleft$	$e_\triangleleft$	$=$	x	

2.4 Utilisation

- s-opérateurs
  - min, max, and, or, $\cup$, $\cap$, $\mathbf{lcm}$, $\mathbf{gcd}$, $\ldots$
  - Résultat du nœud v

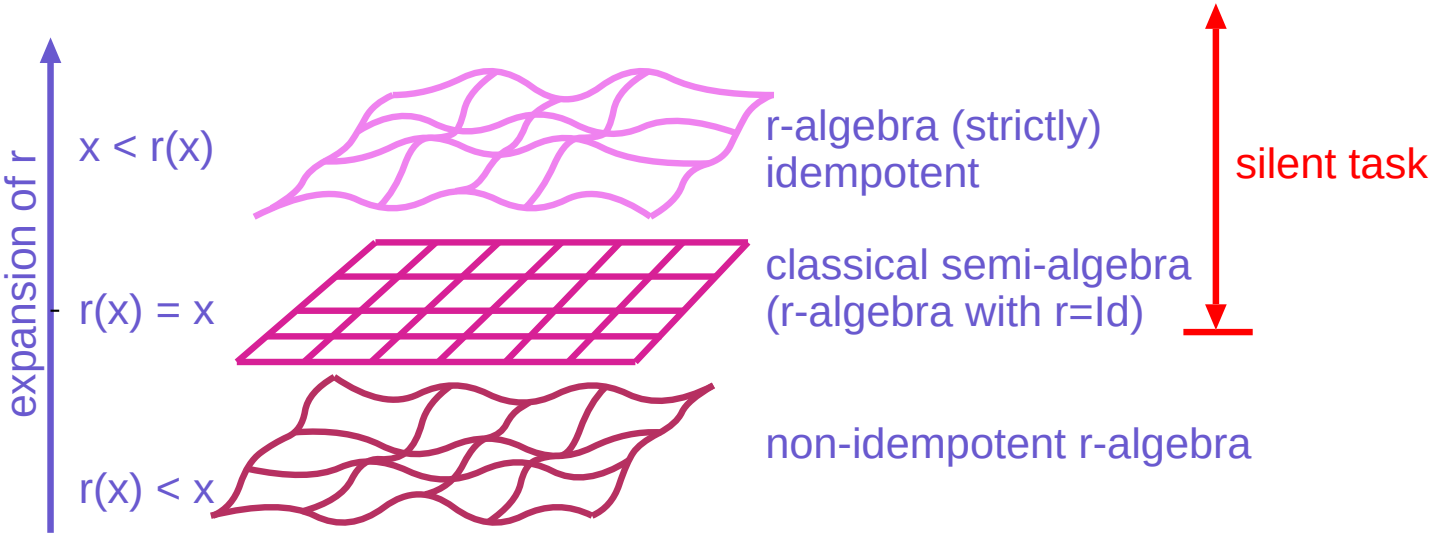
$$\bigoplus \{\mathsf{ROM}_u, u \text{ antécédant de } v \text{ dans le réseau } \}$$

- r-opérateurs
  - $\mathsf{minc}(x, y) = \mathsf{min}(x, y + 1)$
  - $\mathsf{minc}_w(x_0, \ldots, x_n) = \mathsf{min}(x_0, x_1 + w_1, \ldots, x_n + w_n)$
  - $\mathsf{maxmul}_\pi(x_0, \ldots, x_n) = \mathsf{max}(x_0, x_1 \times \pi_1, \ldots, x_n \times \pi_n)$
  - lexicat $((a, b, c), (a', b', c')) = (a, b, c) \oplus (a', b', c', v)$
  - $\ldots$
  - Résultat sur le nœud v

$$\bigoplus \{r_{P_{u \to v}}(\mathsf{ROM}_u), u \text{ antécédant de } v\}$$

3. Intérêt des r-opérateurs

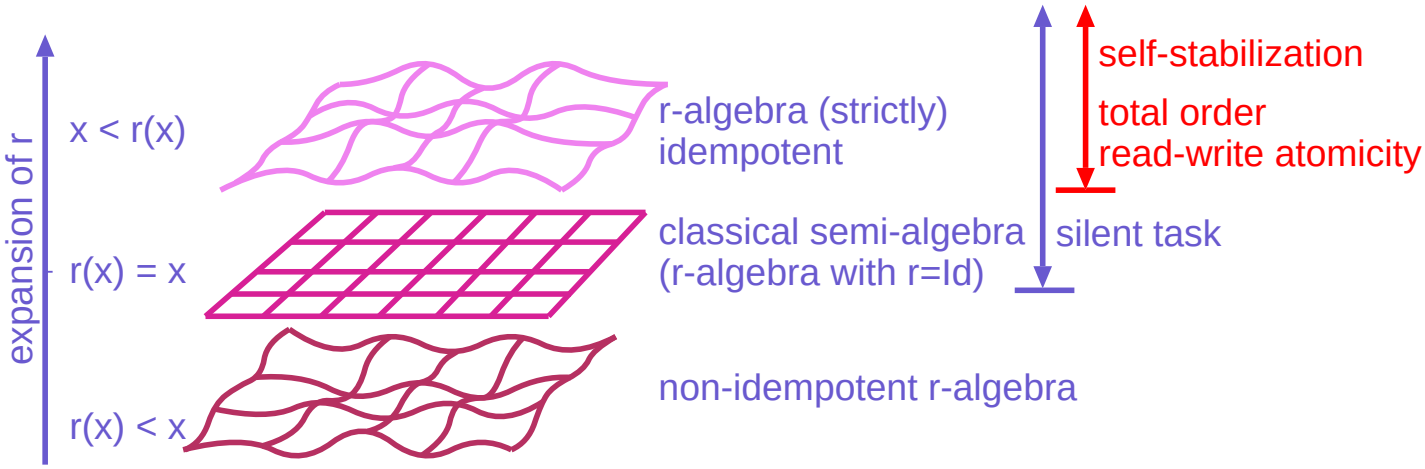
3.1 Tâches statiques.



$$x \triangleleft y = x \oplus \mathsf{r}(y)$$

- s-opérateur
  - stabilisation sur toute topologie, en l'absence de pannes [Tel91]
- r-opérateurs
  - stabilisation sur toute topologie, en l'absence de panne, avec un r-opérateur idempotent [SIROCCO98]
  - hypothèse de synchronisation la plus forte
- Topologie particulières

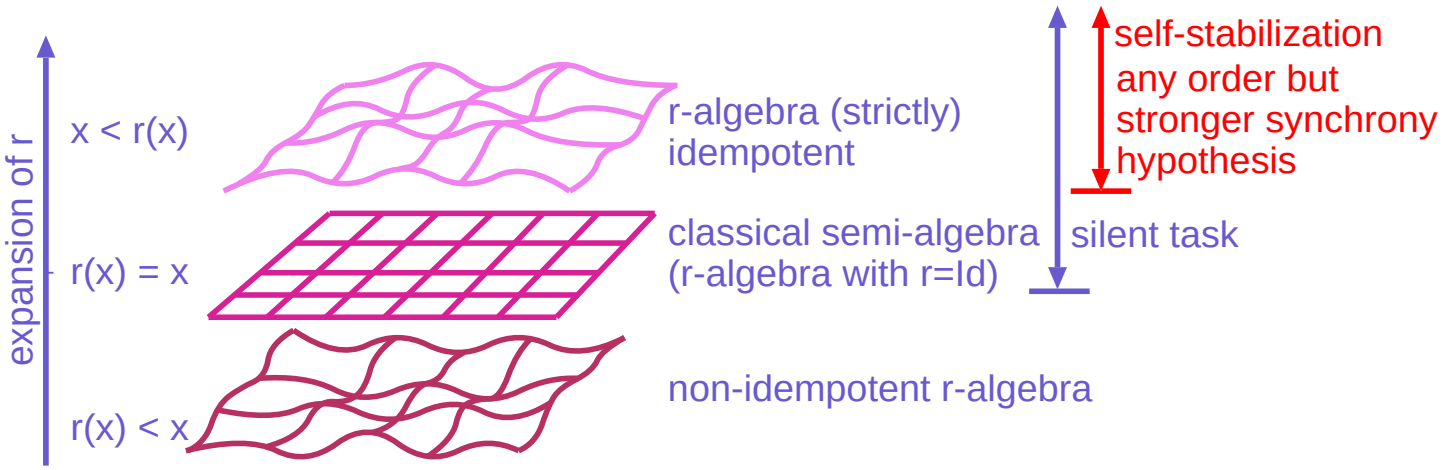
3.2 Tâches statiques auto-stabilisantes.



$$x \triangleleft y = x \oplus \mathsf{r}(y)$$

- auto-stabilisation sur toute topologie malgré les défaillances avec un r-opérateur strictement idempotent basé sur un ordre total $\preceq_{\oplus}$ [DistComp00]
- Stricte idempotence nécessaire
- Hypothèse de synchronisation la plus forte

3.3 Tâches statiques auto-stabilisantes (suite).



$$x \triangleleft y = x \oplus \mathsf{r}(y)$$

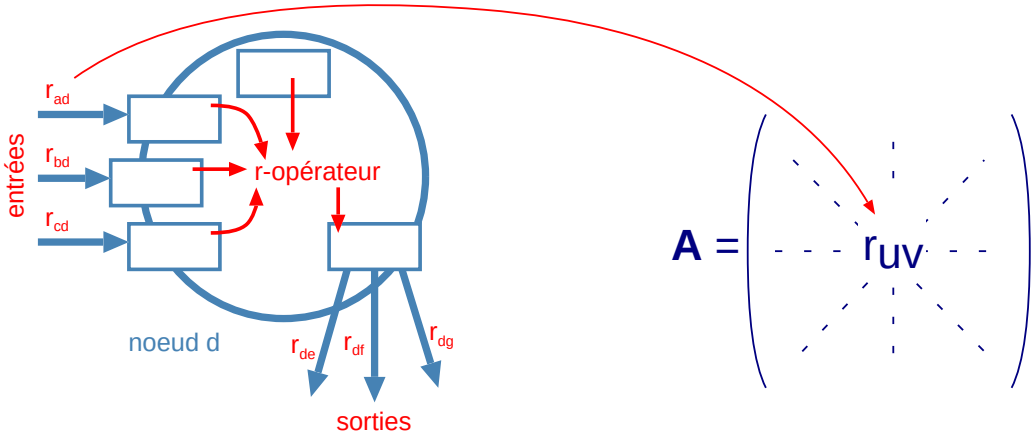
- auto-stabilisation sur toute topologie malgré les défaillances avec un r-opérateur strictement idempotent [SIROCCO00,TCS02]
- Plus d'ordre total, mais hypothèse de synchronisation moins générale
-

ordre total $\preceq_\oplus$	hyp. de sync. la plus générale
ordre partiel $\preceq_\oplus$	hyp. de sync. plus forte

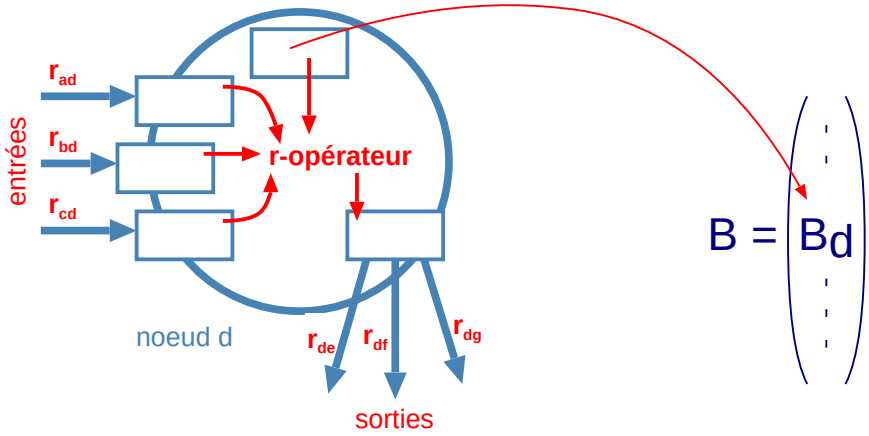
4. Calcul matriciel

4.1 Modélisation matricielle.

- Hypothèses
  - tâches statiques, un résultat par nœud
  - atomicité composite
  - communication par registres
- Matrice $n \times n$ de r-fonctions

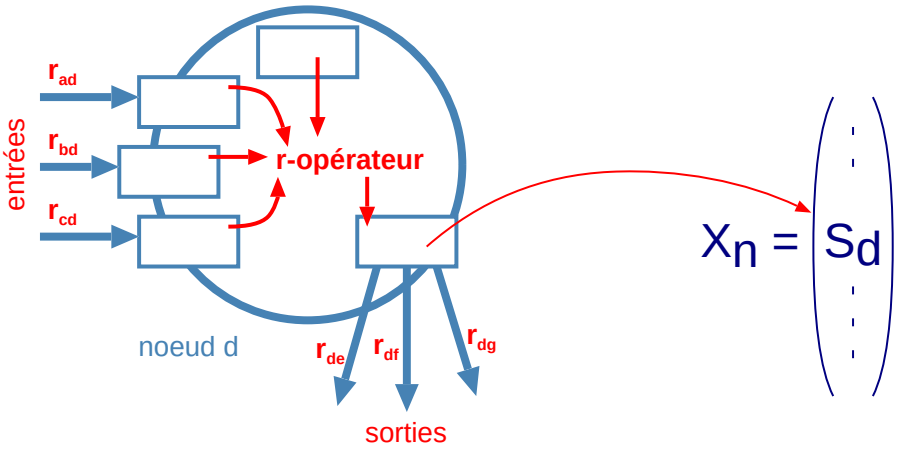


- Vecteur $n \times 1$ de valeurs initiales

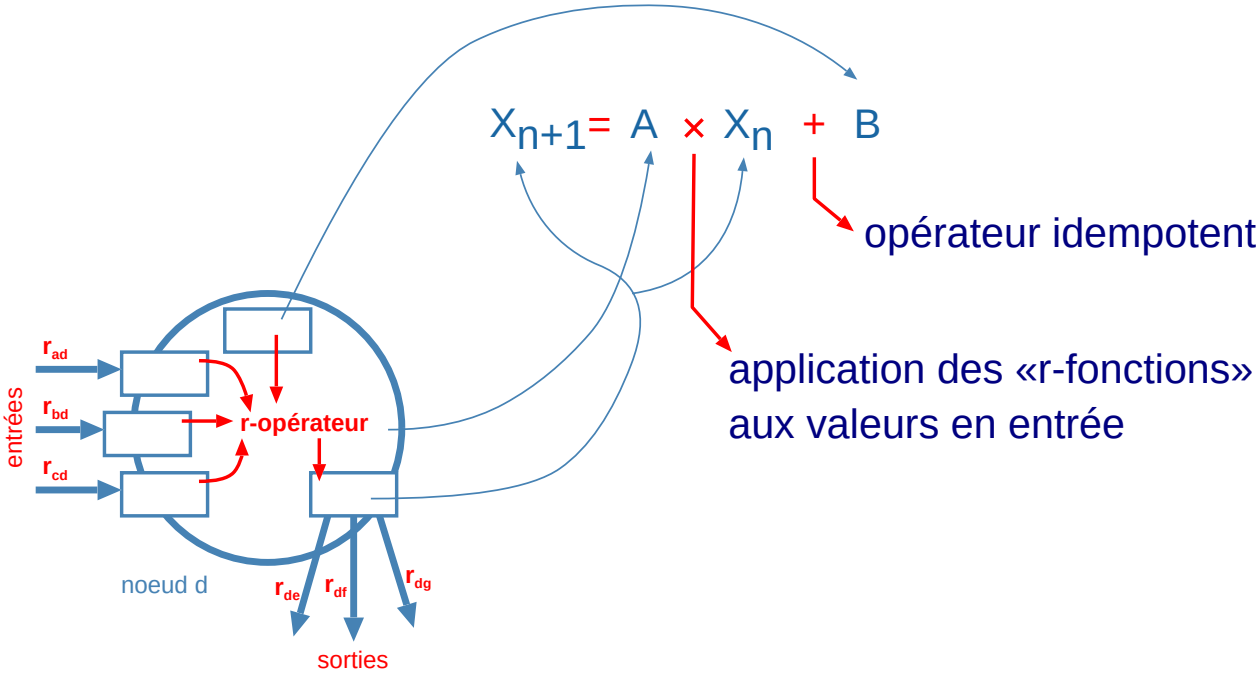


4.2 Calcul matriciel.

- Vecteur $n \times 1$ résultat ($\approx$ configuration)



- Algorithme réparti $\equiv$ multiplication matricielle



$\rightsquigarrow$ démon synchrone

4.3 Démon réparti.

- Démon réparti
  certains processeurs $P_i, i \in \mathcal{J}$ sont activés et
  - lisent les valeurs de tous leurs antécédants
  - calculent un nouveau résultat
  - écrivent le résultat dans le registre de sortie
- Itérations asynchrones

$$\mathtt{X}_{n+1}[i] = \begin{cases} \mathtt{X}_n[i] & \text{si } i \notin \mathcal{J}_n \\ (\mathsf{F}(\mathtt{X}_n))[i] = & \\ (\mathbf{A} \otimes \mathtt{X}_n \ \oplus \ \mathtt{B})[i] & \text{si } i \in \mathcal{J}_n \end{cases}$$

4.4 Démon totalement réparti.

- Démon totalement réparti
  certains processeurs $P_i, i \in \mathcal{J}$ sont actifs et
  - lisent les valeurs de tous leurs antécédants
  - calculent un nouveau résultat
  - écrivent plus tard le résultat (délai $D[i]$)
- Itération asynchrone à retard

$$\mathtt{X}_{n+1}[i] = \begin{cases} \mathtt{X}_n[i] & \text{si } i \notin \mathcal{J}_n \\ \left(\mathsf{F}(\ (\mathtt{X}_{D_n[1]}[1], \ldots, \mathtt{X}_{D_n[N]}[N])^{\mathsf{t}}\ ) \right)[i] = & \\ \left(\mathbf{A} \otimes (\mathtt{X}_{D_n[1]}[1], \ldots, \mathtt{X}_{D_n[N]}[N])^{\mathsf{t}} \oplus \mathtt{B} \right)[i] & \text{si } i \in \mathcal{J}_n \end{cases}$$

4.6 Utilisation.

- Prouver la convergence en l'absence de panne

  [Üresin et Dubois, 1990]

  - $\mathtt{F}(\mathtt{X}) = \mathbf{A} \times \mathtt{X} \oplus \mathtt{B}$ est clos sur $\mathbb{S}^N$;
  - les itérations synchrones convergent et $\mathtt{X}_{n+1} \preceq \mathtt{X}_n \ \forall n \in \mathbb{N}$;
  - F est monotone sur $\mathbb{S}^N$
- Prouver la stabilisation en présence de panne
  - les itérations synchrones convergent $\forall \mathtt{X}_0$
- Idée de preuve
  - $\mathbf{A}^{(k)} = \mathbf{E}_{\otimes} \oplus \mathbf{A} \oplus \cdots \oplus \mathbf{A}^k$
  - $\mathbf{A}^* = \lim_{k \to +\infty} \mathbf{A}^{(k)}$ si pas de circuit absorbant
  - $\mathtt{X}_k = \mathbf{A}^k \otimes \mathtt{X}_0 \ \oplus \ \mathbf{A}^{(k-1)} \otimes \mathtt{B}$
  - idempotence des r-opérateurs $\rightsquigarrow$ $\mathbf{A}^*$ existe
  - stricte-idempotence $\rightsquigarrow$ $\mathbf{A}^k \to \mathbf{E}_{\oplus}$
  - d'où $\mathtt{X}_k \to \mathbf{A}^* \times \mathtt{B}$

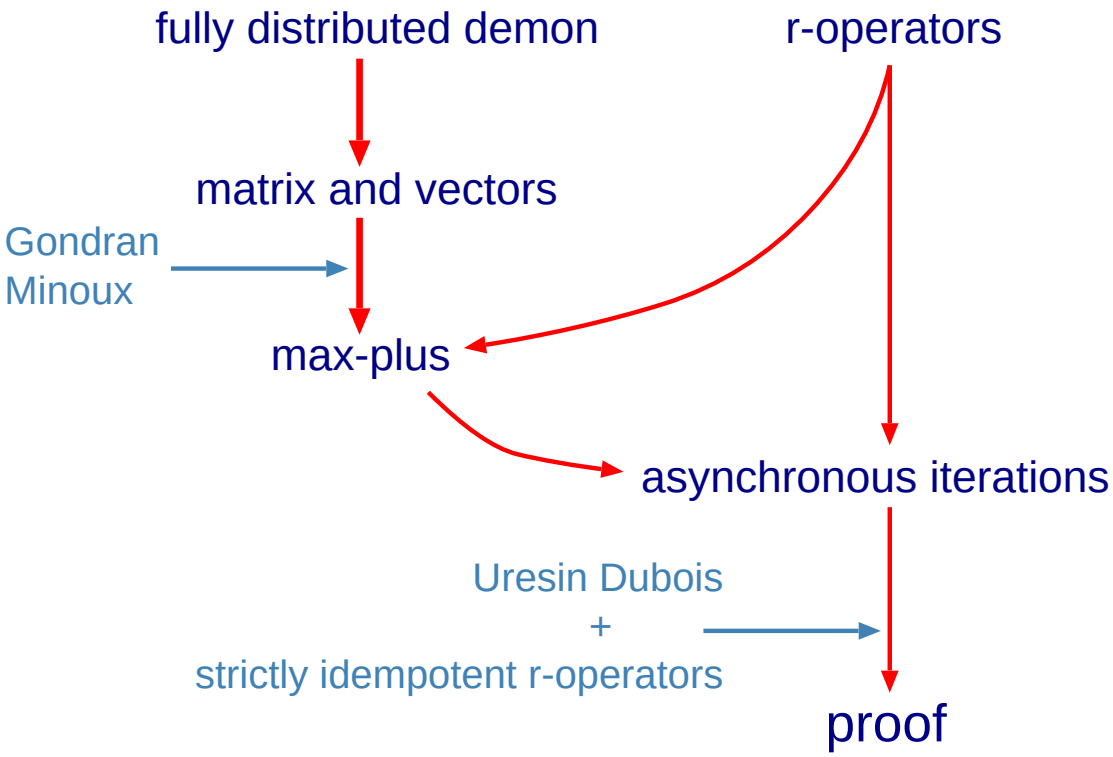
5. Conclusion

5.1 Résumé.

- Auto-stabilisation
  - intérêt et utilisation
  - convergence ?
- Calculs locaux
  - sous la forme d'opérateurs
  - infimum
  - r-opérateurs
- Intérêt des r-opérateurs
  - propriétés auto-stabilisantes
  - modélise les tâches statiques
- Calcul matriciel
  - atomicité composite
  - tâches statiques
  - modélise les tâches statiques

5.2 Contribution et perspective.

- Preuve générique courte *via* l'algèbre max-plus :
  auto-stabilisation $\rightsquigarrow$ point fixe matriciel



- Question ouverte
  $\preceq_{\oplus}$ ordre partiel/total versus type d'atomicité