Utilisation du calcul parallèle pour les économies d'energie sur les systèmes embarqués

Jérôme Galtier

INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93
06902 Sophia-Antipolis Cedex

E-mail: Jerome.Galtier@sophia.inria.fr

27 novembre 1998

$\boxed{\text{OUTLINE}}$

Introduction

Cadre de travail:

- Augmentation du nombre de calculs CPU embarqués (norme UMTS, services traités à bord des satellites)
- Ressources en energie limitées (batteries, plus les panneaux pour les satellites) ⇒ réduire la puissance consommée c'est réduire le poids de l'engin
- Diminution du poids très profitable (ergonomie d'utilisation pour les mobiles, coût de lancement réduit pour les satellites)

Plusieurs techniques d'économie d'énergie ont déjà été expérimentées:

- Techniques de *spin-down*: on coupe tout quand le système est inactif, avec un coût de redémarrage lorsqu'il faut repartir (disques durs).
- Techniques de variation du voltage en fonction de la charge: quand la charge est faible, on ralentit le système et on peut aussi baisser le voltage.

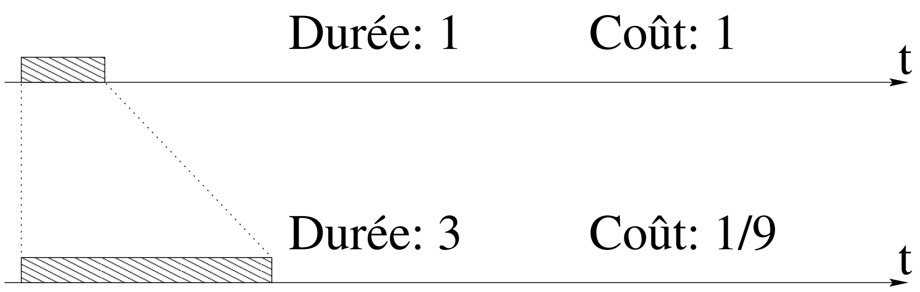
Les techniques de baisse de voltage pour les économies d'énergie

$$\text{energie}/\text{temps} \propto \text{voltage}^2 \cdot \text{vitesse\_horloge}.$$

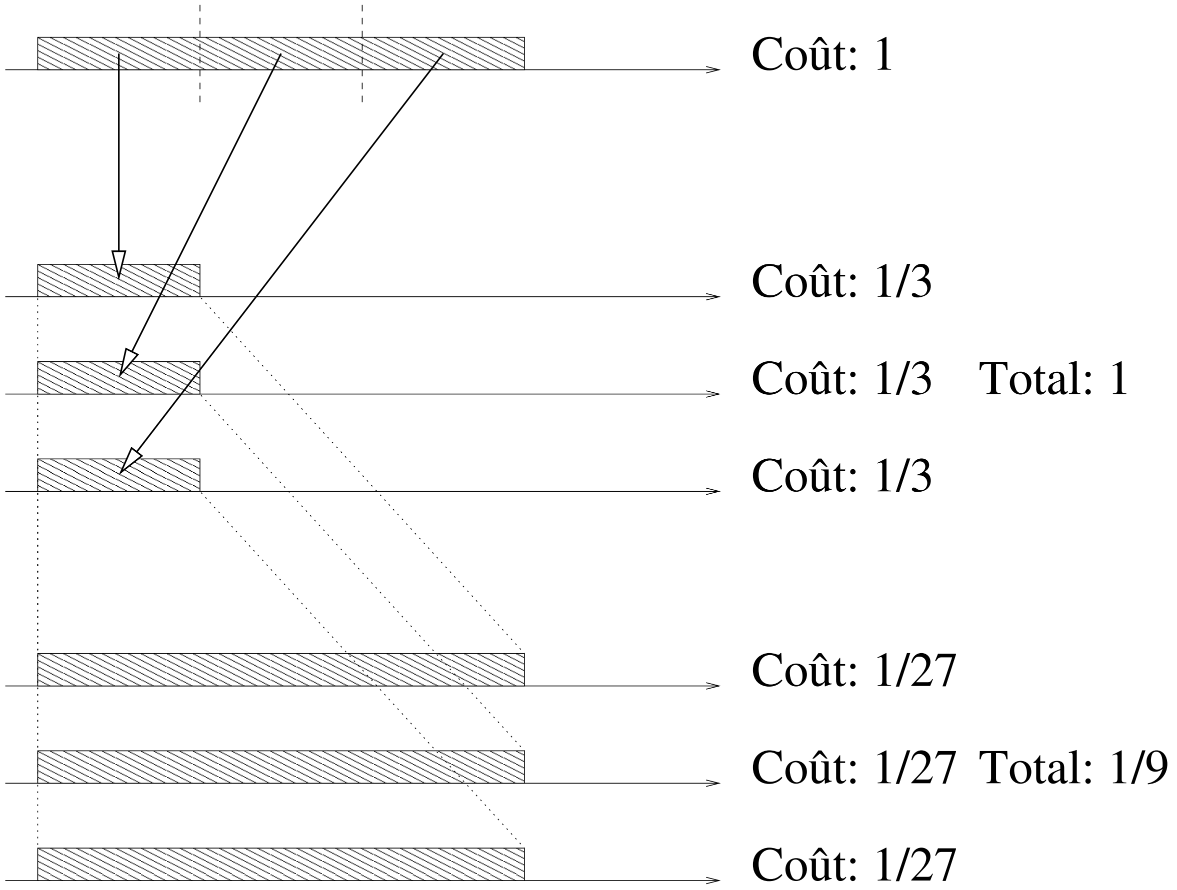
Or une tâche donnée requiert un nombre constant de cycles d'horloge, donné par $c := \text{temps} \times \text{vitesse\_horloge}$. Nous avons alors:

$$\text{energie}/\text{tâche} \propto \text{voltage}^2$$

Or le voltage peut être considéré comme proportionnel à la vitesse d'horloge...



Utilisation du parallélisme

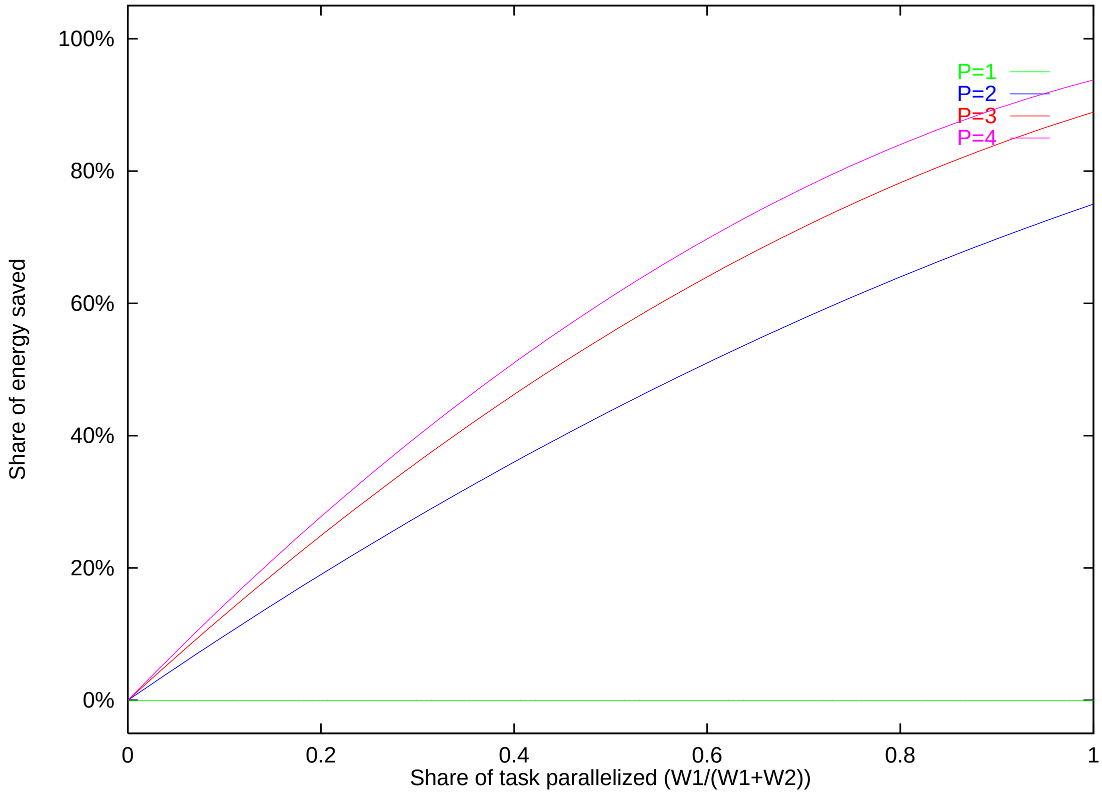


$\rightarrow$ une parallélisation parfaite sur P processeurs permet de consommer seulement $1/P^2$ de l'énergie initiale.

"Loi d'Amdahl" pour les économies d'énergie

- W_1: parallélisable sur P processeurs
- W_2: non parallélisable
- F: facteur de gain

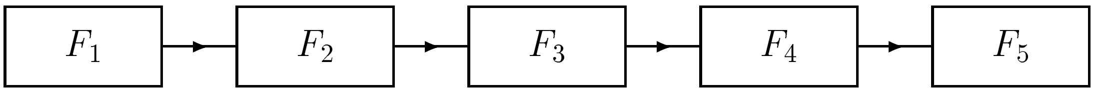
$$F = \left(1 - \frac{W1}{W1 + W2} \frac{P-1}{P}\right)^2$$



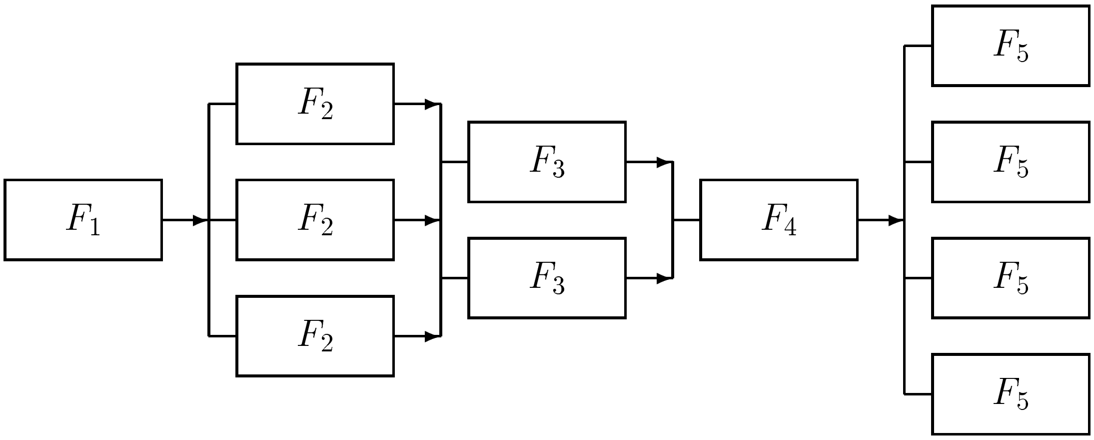
Modélisation d'un filtre

On cherche à modéliser un filtre de traitement CPU de l'information.
Tous les éléments travaillent successivement, chacun prenant T_i clocks d'horloge.

Soit un circuit $\mathcal{F} = (F_1, \ldots, F_p)$ d'éléments en série:



qu'on cherche à paralléliser



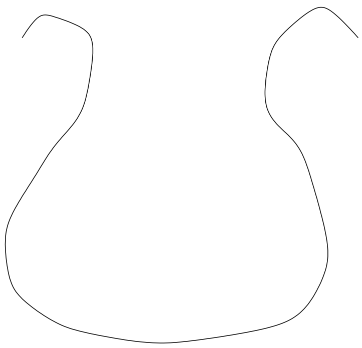
On veut gagner de l'énergie sous la contrainte que le temps **total** $\sum T_i$ reste constant.
Autrement dit, un gain en nombre **total** de clocks de 50% se traduit par un gain en énergie de 75%.
1/3 1/3 1/3 -> 1/3 1/12 1/12

Contraintes de place

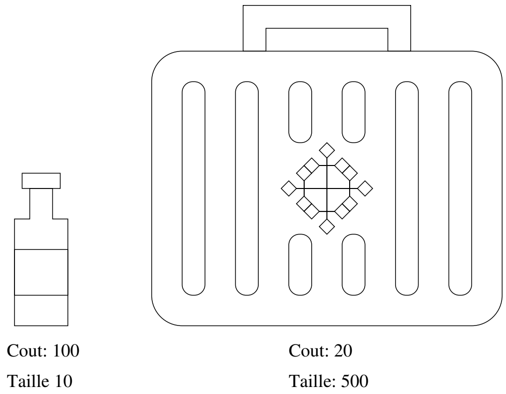
Exemple: puce.
Une copie de F_i prend une place c_i.
L'aire totale disponible est c.

Minimiser $\sum_{i=1}^{i=n} T_i/p_i$ sous la contrainte $\begin{cases} p_i \leq p_i^{max}, & 1 \leq i \leq n \\ \Sigma_{i=1}^{i=n} c_i p_i \leq c \end{cases}$

Le problème du sac à dos



Sac: capacité C



Objets: taille + coût

Maximiser la valeur (la somme des coût) dans le sac sous des contraintes de taille.
Problème pseudo-polynomial.

Réduction au problème du sac à dos en 0/1

On pose:

$$\begin{cases} A_{(i,j)} = \frac{T_i}{j} - \frac{T_i}{j+1} \\ K_{(i,j)} = c_i \end{cases}$$

pour $1 \leq i \leq k$ et $1 \leq j \leq p_i^{max} - 1$.
Et on cherche les $x_{(i,j)} \in \{0, 1\}$ qui résolvent:

$$\begin{cases} \text{Maximiser} \, \Sigma \, A_{(i,j)} x_{(i,j)} \\ \text{ sous la contrainte } \, \Sigma \, K_{(i,j)} x_{(i,j)} \leq c \end{cases}$$

Preuve:
Comme $K_{(i,j_1)} = K_{(i,j_2)}$ et $A_{(i,j_1)} > A_{(i,j_2)}$ dès que $j_1 < j_2$, on a:
$x_{(i,j_2)} = 1 \Rightarrow x_{(i,j_1)} = 1$
Il suffit donc de prendre:
$p_i = \max\{j : x_{(i,j)} = 1\}$

Minimisation de poids

- m_i la masse d'une copie du circuit F_i
- M la masse de référence du système de génération d'énergie, en mode non-optimisé (i.e. la masse en mode séquentiel est $M(\Sigma_{i=1}^{i=n} T_i)^2$).

On suppose que la masse est proportionnelle à la puissance consommée.

Notre problème devient:

$$\text{Minimiser } \sum_{i=1}^{i=n} p_i m_i + M \cdot \left(\sum_{i=1}^{i=n} \frac{T_i}{p_i} \right)^2$$
$$\text{sous la contrainte } p_i \leq p_i^{max}, \quad 1 \leq i \leq n$$

Programmation semi-définie positive (PSP)

On se donne $c_1, \ldots, c_p$ réels, $B_0, \ldots, B_p$ matrices symétriques $n \times n$.

Inconnues $x_1, \ldots, x_p$

On construit $A(x_1, \ldots, x_p) = B_0 + \Sigma_{i=1}^{i=p} x_i \cdot B_i$

On cherche à minimiser $\Sigma_{i=1}^{i=p} c_i \cdot x_i$
sous la contrainte $A(x_1, \ldots, x_p) \geq 0$.

Résultat: Il existe des algorithmes rapides de résolution de PSP.
En théorie, temps polynomial par la méthode de l'ellipsoïde.
En pratique, algos efficaces.

Exemple: (réduction de la programmation linéaire à PSP)

Problème initial:
On cherche à minimiser $\Sigma_{i=1}^{i=p} c_i \cdot x_i$
sous la contrainte $\forall j \qquad b_{j,0} + \Sigma_{i=1}^{i=p} b_{i,j} x_i \geq 0$.

$$B_i = \begin{pmatrix} b_{1,i} & & 0 \\ & \ddots & \\ 0 & & b_{p,i} \end{pmatrix}$$

Approche de réduction de notre problème à PSP

On introduit une série de variables $(y_i)_{1\leq i\leq n}$ et t et on construit X comme suit:

$$X = \begin{bmatrix} \begin{pmatrix} y_1 & \sqrt{T_1} \\ \sqrt{T_1} & p_1 \end{pmatrix} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \\ 0 & \dots & \begin{pmatrix} y_n & \sqrt{T_n} \\ \sqrt{T_n} & p_n \end{pmatrix} & \\ 0 & & & \begin{pmatrix} 1 & \Sigma_{i=n}^{i=1} y_i \\ \Sigma_{i=n}^{i=1} y_i & t \end{pmatrix} \end{bmatrix}$$

Notre problème revient alors à minimiser $\Sigma_{i=n}^{i=1} p_i m_i + Mt$ sous les conditions

- $X \geq 0$ (i.e. X semi-definie positive)
- Les $(p_i)_{1\leq i\leq n}$ sont des entiers

$\rightarrow$ Par programmation semi-définie positive, on obtient en temps polynomial des valeurs flottantes pour les p_i.

Preuve de correction

$X \geq 0$ est équivalent aux inégalités:

$$\left|\begin{array}{ll} y_i \geq 0 & 1 \leq i \leq n \\ y_i \geq \frac{T_i}{p_i} & 1 \leq i \leq n \\ t \geq \left(\Sigma_{i=n}^{i=1} y_i\right)^2 & \end{array}\right.$$

Ce qui implique

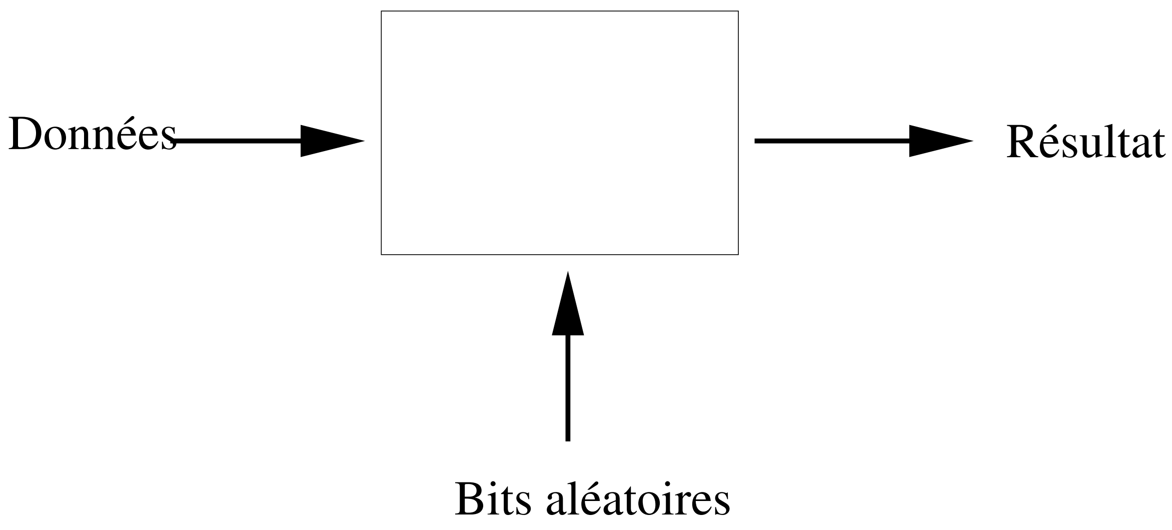
$$t \geq \left(\sum_{i=n}^{i=1} \frac{T_i}{p_i}\right)^2$$

et donc

$$\sum_{i=n}^{i=1} p_i m_i + Mt \geq \sum_{i=n}^{i=1} p_i m_i + M \left(\sum_{i=n}^{i=1} \frac{T_i}{p_i}\right)^2$$

qui est bien la quantité que l'on minimise.

Notions de programmation randommizée



- Le résultat est aléatoire
- L'aléa ne dépend pas du problème posé: sur les mêmes données d'entrée on peut faire tourner le même programme et obtenir un résultat différent.

Passage aux valeurs entières pour les p_i

Résultat: Il existe un programme randomizé de complexité linéaire qui peut donner des valeurs entières $\hat{p}_i$ vérifiant:

$$E\left[\sum_{i=n}^{i=1} \hat{p}_i m_i + M \cdot \left(\sum_{i=1}^{i=n} \frac{T_i}{\hat{p}_i}\right)^2\right] \leq \frac{9}{8}\left[\sum_{i=n}^{i=1} p_i m_i + M \cdot \left(\sum_{i=1}^{i=n} \frac{T_i}{p_i}\right)^2\right]$$

Preuve:

Une fois les p_i obtenus, on choisit indépendamment $\hat{p}_i$ suivant la valeur (flottante) de p_i.

Dans ce cas, nous avons:

$$E\left[\sum_{i=n}^{i=1} \hat{p}_i m_i + M \cdot \left(\sum_{i=1}^{i=n} \frac{T_i}{\hat{p}_i}\right)^2\right] = \left| \begin{array}{l} \Sigma_{i=n}^{i=1} m_i E[\hat{p}_i] \\ +M\, \Sigma_{i=1}^{i=n} T_i E\left[\frac{1}{\hat{p_i}^2}\right] \\ +M\, \Sigma_{i=1}^{i=n} \Sigma_{j=i+1}^{i=n} T_i T_j E\left[\frac{1}{\hat{p}_i}\right] E\left[\frac{1}{\hat{p}_j}\right] \end{array} \right.$$

et on choisit les $\hat{p}_i$ comme suit:

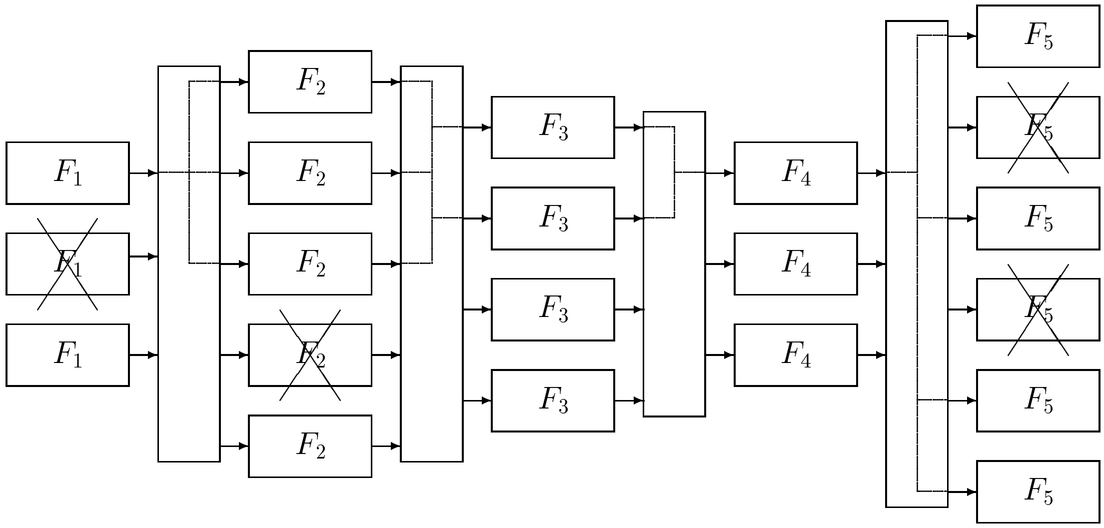
$$\begin{array}{rcl} P[\hat{pi} = \lfloor p_i \rfloor] & = & \alpha \\ P[\hat{pi} = \lfloor p_i \rfloor + 1] & = & 1 - \alpha \end{array} \quad \text{où } \alpha = \frac{\lfloor p_i \rfloor (\lfloor p_i \rfloor + 1)}{p}_i - \lfloor p_i \rfloor$$

On peut vérifier que:

$$\begin{array}{rcl} E[\hat{p}_i] - p_i & = & \alpha \lfloor p_i \rfloor + (1-\alpha)(\lfloor p_i \rfloor + 1) = p_i \frac{\alpha(1-\alpha)}{\lfloor p_i \rfloor (\lfloor p_1 \rfloor + 1)} \leq \frac{1}{8} p_i \\ E[\frac{1}{\hat{p}_i}] - \frac{1}{p_i} & = & \frac{\alpha}{\lfloor p_i \rfloor} + \frac{1-\alpha}{\lfloor p_i \rfloor + 1} = 0 \\ E[\frac{1}{\hat{p_i}^2}] - \frac{1}{p_i^2} & = & \frac{\alpha}{\lfloor p_i \rfloor^2} + \frac{1-\alpha}{(\lfloor p_i \rfloor + 1)^2} = \frac{1}{p_i^2} \frac{\alpha(1-\alpha)}{(\lceil p_i \rceil + \alpha)^2} \leq \frac{1}{8} \frac{1}{p_i^2} \end{array}$$

Directions futures de recherche

- La **tolérance aux pannes** est un des défauts de cette théorie.



- L'utilisation simultanée de plusieurs systèmes d'économie d'énergie:

parallélisme, spin-down, voltage
en fonction des contraintes du réseau...

- Nouvelles applications pour le parallélisme