

Minimisation de la consommation énergétique à l'aide du système d'exploitation

Gilbert Cabillic
INRIA

Projet ACES
IRISA

Ecole ARCHI 2003

Introduction Niveaux

- La gestion de l'énergie peut être faite à plusieurs niveaux



- Plus le niveau est élevé, plus le gain sera important (car la connaissance des informations est plus grande) [SRC84]
- Mais
 - L'utilisateur n'a pas la capacité de prendre des décisions sur des délais très courts
 - Dans un contexte multi-application, les décisions peuvent être contradictoires
- Aussi
 - L'OS est bien placé pour réaliser la gestion de l'énergie

Introduction

Evaluation d'une stratégie (1/2)

- **Outre son évaluation basique, il faut toujours relativiser l'importance de la stratégie d'une manière globale**
 - **Exemple:**
 - Une stratégie permet 50% de réduction énergie pour l'exécution du modem.
 - Si le modem correspond à 4% de l'utilisation globale du processeur, il y aura 2% de gain au final.
 - Il faut évaluer chaque étape afin d'être convaincu de passer à une stratégie qui donnera de meilleurs résultats
- **Il faut également évaluer le temps et l'énergie nécessaire à sa réalisation**
- **Il faut également prendre en compte l'effet de la stratégie sur la batterie elle-même**
 - En fonction de la consommation, la capacité d'énergie peut varier suivant le type de batterie [Bel01] (voir slide 7)

Introduction

Evaluation d'une stratégie (2/2)

- **4 critères d'évaluation**
 - De combien est réduite l'énergie sur le composant
 - Quel est le pourcentage d'énergie affectée à un composant dans le système complet
 - Quelles sont les pénalités sur les autres composants
 - Quel est l'impact sur la batterie

Introduction

Différentes classes de stratégies

1. Transition

- Passage d'un mode de fonctionnement à un autre
 - Exemple: mise en veille du processeur

2. Load/change

- Modification de la charge d'un composant matériel
 - Exemple: mémoire cache de disque (diminution charge du disque)

3. Adaptation

- Adaptation d'un composant en utilisant des techniques nouvelles
 - Exemple: remplacement d'un disque dur par une smartcard
- Le compromis entre ce qu'attend l'utilisateur et l'action de la stratégie n'est pas évident
 - Exemple: à quoi cela sert de diminuer de 50% la consommation si on diminue de 50% la performance temporelle du processeur ?
 - Aussi, le gain se fera en fonction des attentes de l'utilisateur.

Introduction

Utilisation des différents composants [Lor01]

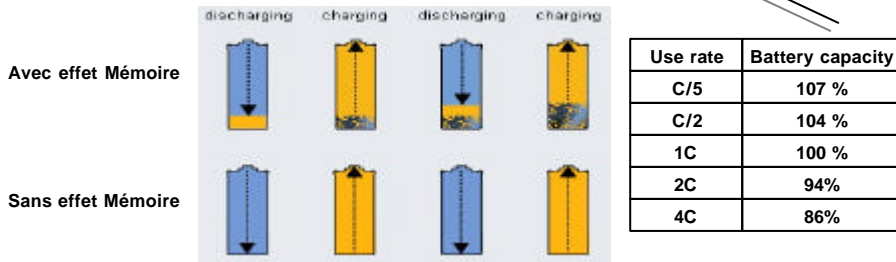
Component	Hypo- thetical 386	Duo 230	Duo 270c	Duo 280c	Average
Processor	4%	17%	9%	25%	14%
Hard disk	12%	9%	4%	8%	8%
Backlight	17%	25%	26%	25%	23%
Display	4%	4%	17%	10%	9%
Modem	n/a	1%	0%	5%	2%
FPU	1%	n/a	3%	n/a	2%
Video	26%	8%	10%	6%	13%
Memory	3%	1%	1%	1%	2%
Other	33%	35%	28%	22%	30%
Total	6 W	5 W	4 W	8 W	6 W

- **Wireless network**
 - Consommation de l'ordre de 100mW à 1W (voir slide 55)

Introduction

Caractéristiques des batteries [Fuj97][Bel01]

- **Lithium ion**
 - 1991: 180 W-h/L
 - 1995: 260 W-h/L 120 W-h/Kg
 - 1997: 380 W-h/L 135 W-h/Kg
- **Nickel Metal Hydride**
 - 1995: 150 W-h/L 50 W-h/Kg
- **Nickel Cadmium**
 - 1995: 125 W-h/L 50 W-h/Kg



Plan

- **Introduction**
- **Les stratégies du système d'exploitation**
 - concernant le support de stockage
 - concernant l'écran
 - concernant le processeur
 - concernant les périphériques Wireless
- **Conclusion**

Support de stockage

Caractéristiques de quelques disques durs

Hard disk	Toshiba MK3017GAP	IBM Travel- star 48GH	Fujitsu MHL2300AT	Hitachi DK22AA-18
Capacity	30 GB	48 GB	30 GB	18 GB
Idle power	0.7 W	0.9 W	0.85 W	0.8 W
Standby power	0.3 W	0.25 W	0.28 W	0.25 W
Sleep power	0.1 W	0.1 W	0.1 W	0.125 W
Spin-up time	4 sec	1.8 sec	5 sec	3 sec
Spin-up energy	10.8 J	9.0 J	22.5 J	13.5 J

Support de stockage

Caractéristiques des Flash Memory

- **Mémoire non volatile (sans consommation d'énergie)**
- **Contrainte**
 - Techniquement, support Read only
 - Pour faire une écriture, il faut détruire et réécrire un segment complet à chaque fois
 - Taille du segment entre 0.5-128 KB
 - 15 usec per byte pour détruire le segment
 - Un segment peut être détruit un nombre limité de fois (100000 à 1.000000)
- **Coût**
 - 1MB= (de 1\$ à 3\$), soit 125-450 fois plus qu'un disque dur et 8-24 fois plus cher que de la mémoire externe DRAM
- **Caractéristiques**
 - Consommation Read/Write: 0.15 W – 0.47 W
 - Read Speed: 85 ns per byte (idem DRAM)
 - Write Speed: 4-10 usec (10-100 fois plus lent qu'un disque dur)
 - Néanmoins, comme il n'y a pas d'opération de positionnement (seek), consommation en écriture globalement équivalente

Support de stockage

Transition – Mise en sommeil/Arrêt du disque (1/2)

- **Plusieurs approches**
 - **Mettre en sommeil le disque**
 - **Mettre en standby le disque**
 - La différence principale est qu'en mode standby on garde le contenu du cache du disque par rapport à une mise en sommeil
 - Fonctionnalité nouvelle des disques, peu abordée dans la littérature
 - On préférera la mise en sommeil car
 - L'énergie gagnée est plus importante
 - Le temps de redémarrage du disque est quasiment identique
 - **Arrêter le fonctionnement du disque**

Support de stockage

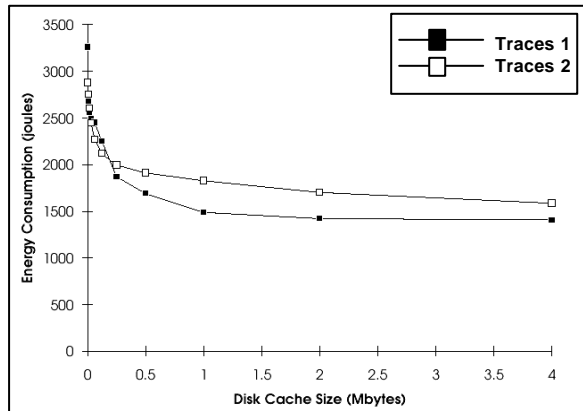
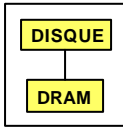
Transition – Mise en sommeil/Arrêt du disque (2/2)

- **Mise en sommeil du disque après une période d'inactivité fixe**
 - Les constructeurs : 3-4 minutes
 - Il apparaît que 1-10 secondes permet de gagner près du double d'énergie [LKHA94]
- **Mise en sommeil du disque après une période d'inactivité variable**
 - [KLV95] définit un ensemble de seuils possibles et une politique de surveillance de l'utilisation du disque afin de choisir le seuil approprié

Support de stockage

Load/Change - Changement de la charge mémoire du disque (1/2)

- **Données**
 - Introduction d'un cache [LKHA94] de 1MB -> 50% de réduction.
 - Au delà d'un certain seuil, pas de gain (idem cache mémoire)



Support de stockage

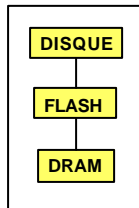
Load/Change - Changement de la charge mémoire du disque (2/2)

- **File name et attributs**
 - Introduction d'un cache [LKHA94] de 50 KB -> 17% de réduction.
- **Problème posé sur le swap (pagination mémoire)**
 - Identifier les bonnes pages qui permettent de minimiser les transferts mémoires.

Support de stockage

Adaptation - Utilisation d'une flash memory comme cache disque

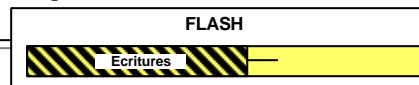
- Dû à la performance des écritures, on place la flash memory comme un cache L2, entre le cache disque DRAM et le disque.
- [MDK93]
 - Taille de Cache de 1-40 MB permet de minimiser de 20-40 % d'énergie et accélérer les accès de 30-70 %
- Problème :
 - Nécessité de maintenir une liste des segments écrits afin de répartir les écritures sur la flash memory afin de ne pas perdre les données



Support de stockage

Adaptation - Utilisation d'une flash memory comme disque

- Gestion différente d'un disque dur
 - [KNM95] Utilisation d'un journal des écritures afin de minimiser le nombre de destruction de segments
 - Réaliser des permutations de segments afin de ne pas réaliser les écritures toujours sur le même segment
- Réduction énergétique de 60-90 % par rapport à un disque dur pour une performance similaire [DKM94]



Support de stockage

Adaptation – Utilisation du réseau Wireless comme un disque distant

- Limitation
 - La Faible bande passante du réseau ne permet pas d'atteindre des débits importants

Plan

- **Introduction**
- **Les stratégies du système d'exploitation**
 - concernant le support de stockage
 - concernant l'Ecran
 - concernant le processeur
 - concernant les périphériques Wireless
- **Conclusion**

Périphérique d'affichage

- **Duo280**
 - Périphérique d'affichage écran: 0.75W
 - Lumière de fond: 3.40W
- **Stratégie de transition**
 - Extinction de la lumière de fond au bout d'un certain temps
 - 32-67% de gain [Lor95a]
 - Utilisation d'un périphérique détectant la luminosité externe afin de déterminer la puissance d'éclairage [Soh95][IPAQ spec]
 - Travail sur les écrans / luminosité

Plan

- Introduction
- Les stratégies du système d'exploitation
 - concernant le support de stockage
 - concernant l'Ecran
 - concernant le processeur
 - concernant les périphériques Wireless
- Conclusion

Processeur

Transition - Stratégies de mise en sommeil du processeur (1/3)

- L'OS décide de la mise en sommeil du processeur
 - Soit en utilisant un mode sommeil (IDLE1,2,3) pour les Processeurs DSP
 - Soit en utilisant une instruction du type WAIT_FOR_INTERRUPT qui met en sommeil le processeur jusque la prochaine interruption. La latence de réveil dépend du processeur
- Quand aucun process n'est prêt à s'exécuter, l'OS réalise un WAIT_FOR_INTERRUPT afin de traiter un événement externe qui a pour action de réaliser l'exécution de process.
- L'os peut aussi décider de changer la fréquence du processeur afin de limiter son énergie et réactiver la fréquence lors la prochaine IT.

Exemple pour la mise en sommeil Pocket Remote control RF transmitter

- **Objectif**
 - Ouvrir et fermer les portes à distance
- **Caractéristiques**
 - 0.1 Mips,
 - Résistant à l'eau,
 - Petite mémoire, logiciel en Eprom de moins de 1KB
 - Autonomie de 5 ans pour une batterie



Processeur

Transition - Stratégies de mise en sommeil du processeur (2/3)

- **1 - MACOs 7.5**
 - Incrémentation d'un compteur d'activité et d'un compteur d'IO
 - Mise en sommeil du processeur lorsque
 - Pas d'activité processeur pendant plus de 2 secondes
 - Pas d'IO depuis 15 secondes
 - Réveil sur prochain événement
- **2 - Windows 3.1**
 - Présence d'une boucle IDLE système qui est exécutée dès lors qu'aucune tâche ne peut s'exécuter et lorsqu'aucune interruption n'est à traiter
 - Mise en sommeil immédiate du processeur
 - Réveil sur prochain événement

Processeur

Transition - Stratégies de mise en sommeil du processeur (3/3)

- **3 – Application specific**
 - **Traitement spécifique des process event-driven qui sont exécutés via des évènements, réalisent une action, puis sont suspendus.**
 - **Dans le cas où un traitement ne se termine pas (du genre un busy-wait) dans son code, le système le suspend volontairement pendant une certaine période de temps**

Strategy	Energy saved	Performance impact
Optimal	82.33 %	0.00 %
1	28.79 %	1.84 %
2	31.89 %	0.00 %
3	51.72 %	1.84 %

Evaluation des différentes stratégies [Lor95]

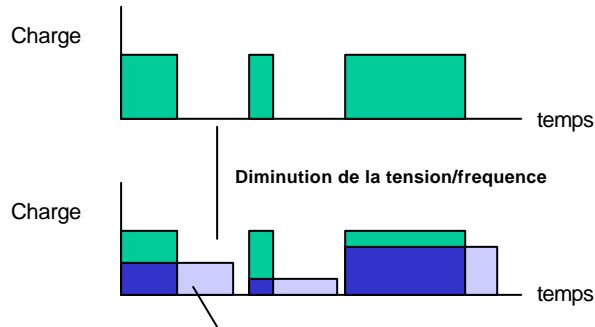
Processeur

Load/change - Réduire la consommation du processeur en travaillant sur les traitements

- **Travail sur le code des applications**
 - **OS:Analyse énergétique d'un OS et optimisation des parties gourmandes**
- **Utilisation d'instructions moins gourmande en énergie**
- **Utilisation de compilateurs energy-aware [TMWL96]**

Processeur

Adaptation - Changement dynamique de la fréquence du processeur (1/9)



Cadencement plus faible,
Consommation beaucoup moins faible,
Temps d'exécution plus long, impact sur le temps-réel,
Comme temps d'exécution plus long,
Le temps d'activation des autres composants est plus long

Processeur

Adaptation - Changement dynamique de la fréquence du processeur (2/9)

[WWDS94]

- Division du temps par période de 10-50 ms
- Au début de chaque période, on caractérise l'utilisation du processeur sur la période précédente
- Si l'utilisation croît, on augmente le cadencement du processeur, si l'utilisation décroît on diminue le cadencement du processeur
- 50% de gain si tension peut être réduite de 5V à 3.3V
- 70% de gain si tension peut être réduite de 5V à 2.2V

Processeur

Adaptation - Changement dynamique de la fréquence du processeur (3/9)

[CGW95]

- Division du temps par période de 10-50 ms
- Prédiction au début de la période de l'activité du processeur et cadencement en fonction de cette activité
- La difficulté est de bien prédire l'avenir...

[PBB98] Prédictions réalistes :

- *Past*, prédiction=activité période précédente
- *Aged*, prédiction=moyenne des activités des périodes précédentes en pondérant plus fortement les périodes les plus récentes
- *LongShort*, moyenne des 12 activités précédentes. Les 3 dernières ont 3 fois plus de poids.

Processeur

Adaptation - Changement dynamique de la fréquence du processeur (4/9)

- [PS01] ordonnancement DVS pour temps-réel strict
- Rappel Ordonnancement EDF
 - Soit un système composé de n Processus P_i de période p_i et de WCET c_i
 - Le taux d'utilisation d'un processeur pour un processus est $U_i = c_i/p_i$
 - Le chargement temporel d'un processeur pour un ensemble de processus est donc donné par
 - $TL = \sum U_i$
 - Le test de faisabilité d'ordonnancement EDF
 - $TL = \sum U_i \leq 1$,
 - Dynamiquement, EDF choisit le processus ayant l'échéance la plus proche comme le plus prioritaire

Processeur

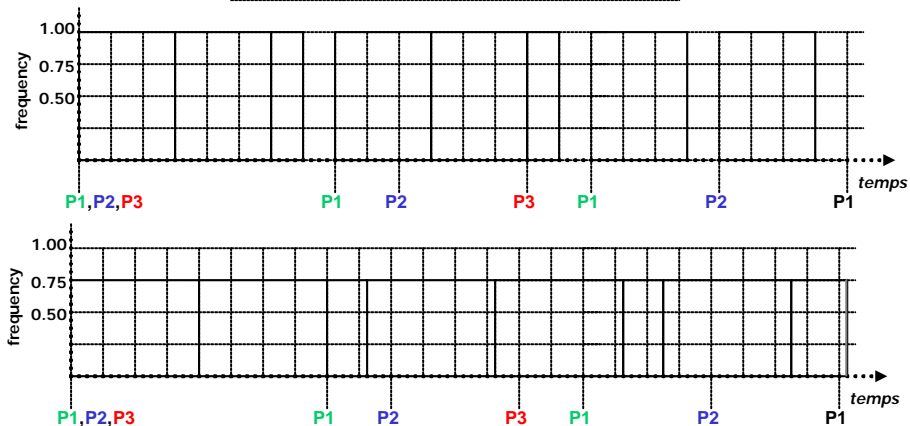
Adaptation - Changement dynamique de la fréquence du processeur (5/9)

- [PS01] ordonnancement DVS pour temps-réel strict
- Lors d'un changement de fréquence processeur d'un facteur α , le temps d'exécution d'un processus P_i est multiplié par $1/\alpha$.
- Le test de faisabilité d'ordonnancement EDF devient pour une fréquence α :
 - $TL = S_i U_i \leq \alpha$,
- Exemple pour un processeur disposant de 3 fréquences de cadencement $\alpha = (1, 0.75, 0.5)$

Processeur

Adaptation - Changement dynamique de la fréquence du processeur (6/9)

Processus	Période	$C_a = 1$	$C_a = 0.75$
1	8 ms	3 ms	4 ms
2	10 ms	3 ms	4 ms
3	14 ms	1 ms	1.33 ms



Processeur

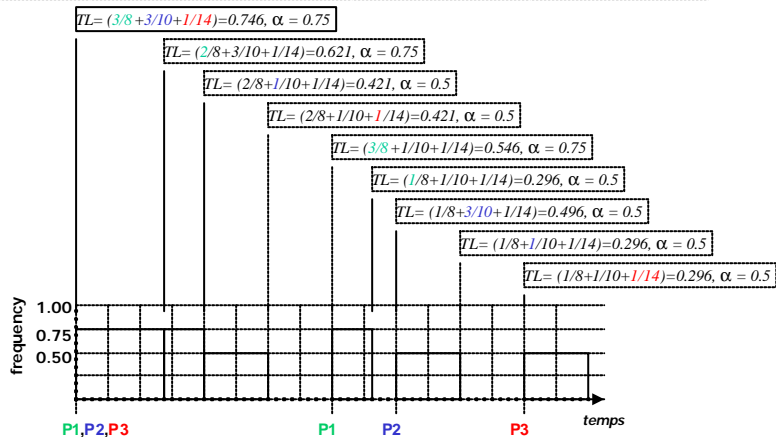
Adaptation - Changement dynamique de la fréquence du processeur (7/9)

- Cycle-conserving RT-DVS (EDF)
 - Prise en compte dynamique du temps d'exécution 'réel' des process
 - Ajustement dynamique de la charge du processeur à partir du temps d'exécution des tâches
 - Quand P_i devient prêt (début de période)
 - Utilisation $U_i = c_i/p_i$
 - Quand P_i se termine cc_i temps d'exécution 'réel'
 - Ajuster l'utilisation à $U_i = cc_i/p_i$ pour $\alpha=1$
 - Comme précédemment, choisir la fréquence pour que $TL = S_i U_i \leq \alpha$,

Processeur

Adaptation - Changement dynamique de la fréquence du processeur (8/9)

Processus	Période	WCET $a=1$	Instance 1 $a=1$	Instance 1 $a=0.75$	Instance 1 $a=0.5$	Instance 2 $a=1$	Instance 2 $a=0.75$	Instance 2 $a=0.5$
1	8 ms	3 ms	2 ms	2.67 ms	4 ms	1 ms	1.33 ms	2 ms
2	10 ms	3 ms	1 ms	1.33 ms	2 ms	1 ms	1.33 ms	2 ms
3	14 ms	1 ms	1 ms	1.33 ms	2 ms	1 ms	1.33 ms	2 ms



Processeur

Adaptation - Changement dynamique de la fréquence du processeur (9/9)

- **Evaluations**

DVS method	Energy saved
EDF	0 %
EDF DVS	36 %
EDF DVS conserving	48 %

Processeur

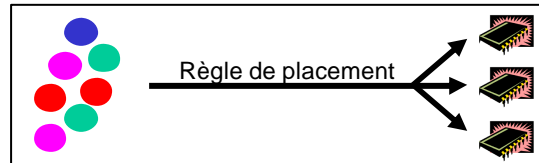
Load/change - Mise en sommeil de certaines parties du processeur

- **Dépend du support matériel**
- **Pour la surveillance,**
 - **Utilisation de compteurs matériel qui indiquent par blocs fonctionnels internes la quantité d'énergie utilisée [TI Patent 2001]**

Processeur

Load/change – Distribution et multiprocesseur hétérogène

- **Contexte multiprocesseur hétérogène :**
 - Plusieurs CORE différents
 - Tension d'alimentation plus basse que sur un uniprocésseur
 - Gain important d'énergie avec potentiel de performance identique
- **Pour l'os :**
 - Distribution des traitements sur l'ensemble des processeurs



- La règle de placement réalise le placement et l'ordonnancement
- **Présentation des principes, détails dans le mémoire de thèse de univ. Rennes I, F. PARAIN “*Ordonnement sous contraintes énergétiques d'applications multimédia sur une plate-forme multiprocesseur hétérogène*”**

Distribution et multiprocesseur hétérogène

Hypothèses

- **Tâches prêtes à s'exécuter**
- **Pas de dépendances entre tâches**
- **Temps d'exécution et consommation énergétique connus**
- **Toutes les échéances doivent être respectées**
- **Problème NP-complet \exists algorithmes « au meilleur effort »**

Distribution et multiprocesseur hétérogène

Algorithme sans optimisation énergétique, RTO

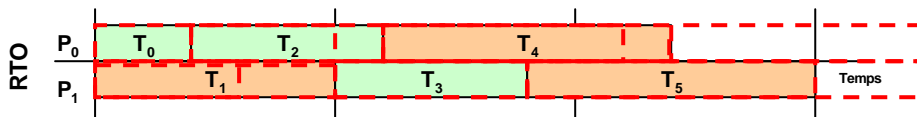
- Principe:
 - Terminer toutes les tâches au plus tôt
 - Pas de prise en compte de la consommation énergétique
- Algorithme :
 - Prendre les tâches par ordre d'échéance croissante
 - pour chaque tâche, choisir le placement qui :
 - permet à la tâche de se terminer au plus tôt

Distribution et multiprocesseur hétérogène

Algorithme sans optimisation énergétique, RTO

- D : échéance
- C_x : temps d'exécution sur P_x
- E_x : énergie consommée sur P_x

	T ₀	T ₁	T ₂	T ₃	T ₄	T ₅
D	4	6	7	10	14	17
C ₀	2	3	4	5	6	7
C ₁	3	5	4	4	9	6
E ₀	10	8	16	23	43	31
E ₁	20	14	18	21	38	37



Distribution et multiprocesseur hétérogène

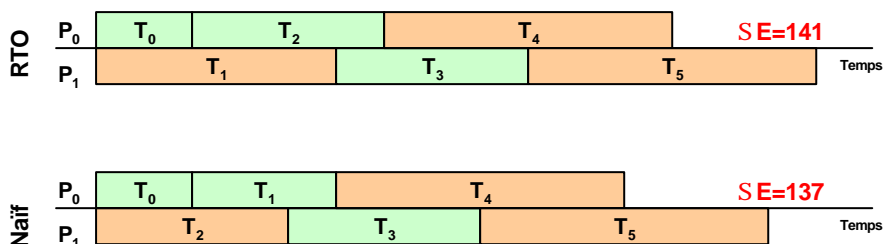
Algorithme à optimisation énergétique naïf

- Principe:
 - Respecter les échéances
 - Optimiser la consommation localement à chaque tâche

- Algorithme :
 - Prendre les tâches par ordre d'échéance croissante
 - Pour chaque tâche, choisir le placement qui :
 - Génère la plus petite consommation
 - Permet le respect de son échéance

Distribution et multiprocesseur hétérogène

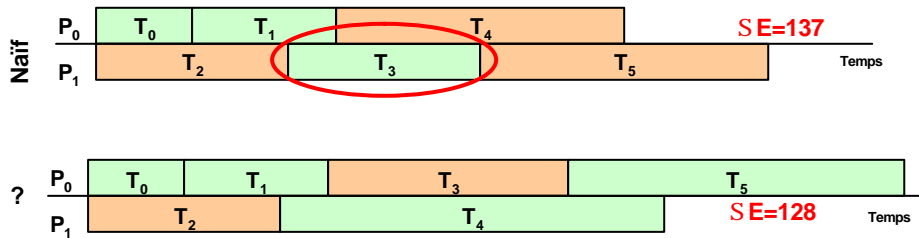
Algorithme à optimisation énergétique naïf



- L'algorithme naïf permet d'obtenir un ordonnancement optimisant la consommation sans dépassement d'échéances

Distribution et multiprocesseur hétérogène

Algorithme à optimisation énergétique naïf



- Le placement de T3 engendre un effet important sur le placement de T4 et T5
 - Comment savoir cela ?

Distribution et multiprocesseur hétérogène

Proposition d'un critère de différenciation énergétique des tâches , $d(T)$

- **Objectif :**
 - Déterminer un critère qui identifie les tâches dont le placement influence le plus fortement la consommation globale
- **Définition :**
 - Le paramètre $d(T)$ est égal à la plus grande des différences entre les consommations engendrées par l'exécution d'une tâche T sur deux processeurs différents
- **Formule :**
 - $d(T) = \text{Max}(|E_x - E_y|)$

Distribution et multiprocesseur hétérogène
Proposition d'un critère de différenciation énergétique des tâches, $d(T)$

	T_0	T_1	T_2	T_3	T_4	T_5
D	4	6	7	10	14	17
C_0	2	3	4	5	6	7
C_1	3	5	4	4	9	6
E_0	10	8	16	23	43	31
E_1	20	14	18	21	38	37
$d(T)$	10	6	2	2	5	6

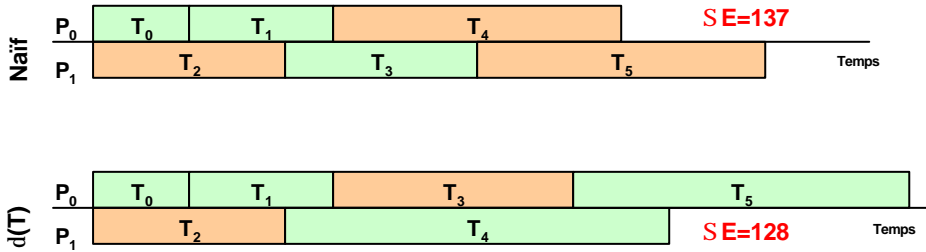
- Ce paramètre $d(T)$ est facilement calculable
- Le paramètre $d(T)$ fait apparaître la plus grande importance des tâches T_4 et T_5 par rapport à T_3

Distribution et multiprocesseur hétérogène
Algorithme à optimisation énergétique basé sur l'exploitation de $d(T)$

- Principe :
 - Respecter les échéances
 - Favoriser le placement des tâches ayant une forte influence sur la consommation en utilisant le paramètre $d(T)$
- Algorithme :
 - Considérer les tâches par ordre de $d(T)$ décroissant
 - Pour chaque tâche, choisir le placement :
 - générant la plus faible consommation
 - ne provoquant pas de dépassement d'échéance
 - Placement d'une tâche sur un processeur
 - Par ordre d'échéances croissantes

Distribution et multiprocesseur hétérogène

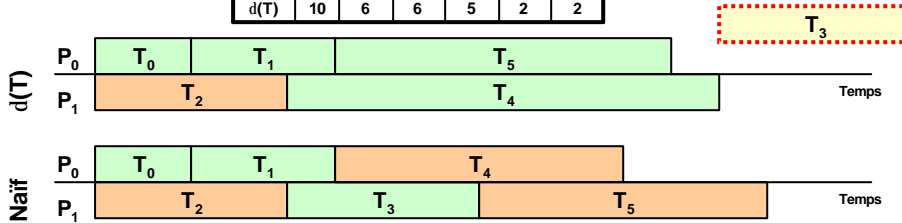
Algorithme à optimisation énergétique basé sur l'exploitation de $d(T)$



Distribution et multiprocesseur hétérogène

Analyse de l'algorithme

	T_0	T_1	T_5	T_4	T_2	T_3
D	4	6	17	14	7	10
C_0	2	3	7	6	4	6
C_1	3	5	6	9	4	4
E_0	10	8	31	43	16	23
E_1	20	14	37	38	18	21
$d(T)$	10	6	6	5	2	2



- L'algorithme $d(T)$ échoue au moment de placer T_3 , pourtant, l'algorithme naïf trouve un ordonnancement

Distribution et multiprocesseur hétérogène

Algorithme EAS (Energy Aware Scheduler)

- **Principe :**
 - Combiner les deux algorithmes précédents
- **Algorithme :**
 - Sélectionner un partitionnement initial des tâches en deux groupes
 - Calculer un ordonnancement :
 - Appliquer l'algorithme $d(T)$ au premier groupe
 - Appliquer l'algorithme naïf au second groupe
 - Si le calcul de l'ordonnancement échoue :
 - sélectionner un nouveau partitionnement des tâches
 - retourner à l'étape 2
 - Répéter la procédure jusqu'à obtenir un ordonnancement

Distribution et multiprocesseur hétérogène

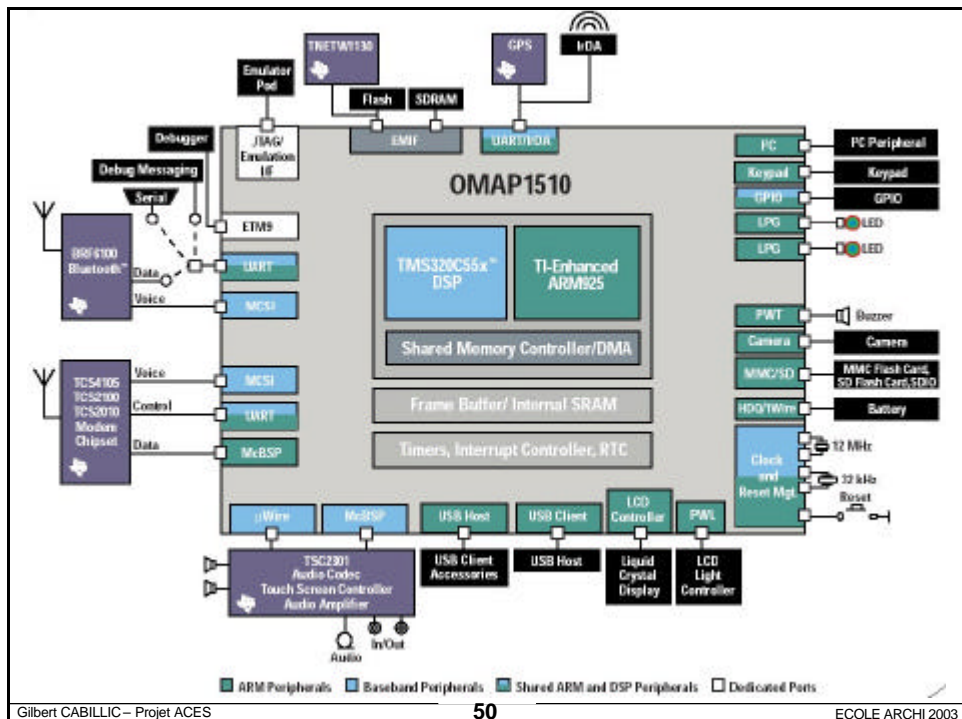
Notre partitionnement

- **Partitionnement initial :**
 - Toutes les tâches sont placées dans le premier groupe ordonné par l'algorithme $d(T)$
- **Calcul d'un nouveau partitionnement:**
 - Transfert de la moitié des tâches du premier groupe vers le second
 - Transfert des tâches avec les paramètres $d(T)$ les plus faibles

Distribution et multiprocesseur hétérogène

Conditions expérimentales

- **Plate-forme OMAP (Texas Instruments)**
- **Profiling des tâches**
 - Traces d'exécutions réelles
 - Modèle mathématique de l'OMAP
- **Simulateur**
 - Exécute le code réel de l'ordonnanceur
 - Permet de nombreuses mesures sans affecter l'ordonnement
- **Applications**
 - Deux décodeurs audio MPEG-1 Layer 3
 - Flux de 8 Kbps et 128 Kbps
 - Un décodeur vidéo MPEG-1

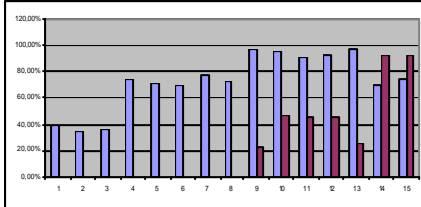


Distribution et multiprocesseur hétérogène Evaluations

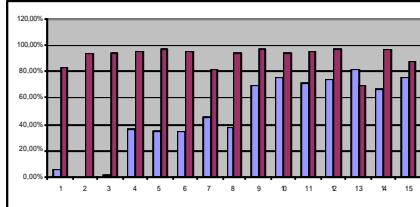
MP3 Audio 8kbps	MP3 Audio 128kbps	MPEG Vidéo	GAIN Energie entre EAS et RTO
1	0	0	29,48%
2	0	0	14,64%
3	0	0	0,10%
0	1	0	30,03%
0	2	0	15,44%
0	3	0	0,94%
0	0	1	30,29%
0	0	2	15,15%
1	1	0	15,46%
2	1	0	5,61%
1	2	0	8,24%
1	0	1	-
0	1	1	16,06%
0	2	1	6,24%
0	1	2	6,52%
1	1	1	5,75%

Distribution et multiprocesseur hétérogène Analyse répartition de charge

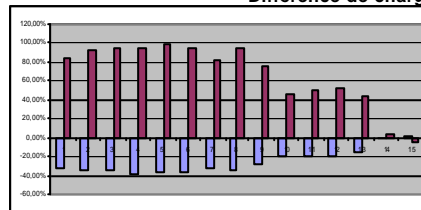
Algorithme RTO



EAS



-
Différence de charge



Le gain sur la consommation diminue lorsque la charge du système augmente

La saturation des processeurs empêche l'ordonnanceur d'effectuer les placements optimisant la consommation

Plan

- **Introduction**
- **Les stratégies du système d'exploitation**
 - concernant le support de stockage
 - concernant l'Ecran
 - concernant le processeur
 - concernant les périphériques Wireless
- **Conclusion**

Périphérique wireless Caractéristiques (1/4)

- **5 modes de fonctionnement**
 - Transmission
 - Réception
 - IDLE (consomme toujours de l'énergie)
 - Sommeil (consomme très peu d'énergie pour la surveillance des connexions)
 - OFF
- **Consommation entre mode réception et IDLE quasiment identique**
- **Temps de mise en sommeil et de réveil**
 - HP HSDL Infrared: 10 usec mise en sommeil, 40 usec réveil
 - AT&T WaveLan infrared: 100 ms réveil
 - Metricom Wireless Modem: 5 secondes réveil

Périphérique wireless

Caractéristiques (2/4)

- La consommation dépend de la distance entre émetteur et récepteur
 - ARDIS, système pour longue distance: 40W
 - Metricom: 1W
 - WaveLan PCMCIA card 1.875W (proximité)
 - HDSL infrared: 55mW (proximité)

Périphérique wireless

Caractéristiques (3/4)

- Bluetooth
 - initiée en 1994 par Ericsson
 - Technologie de moyen débit 720 kbits/s (1 Mbits/s en débit théorique) à basse consommation énergétique.
 - Rayon d'action est limité entre 10 et 30 mètres et ses composants de petite taille lui permettent d'être inséré dans des équipements très mobiles (mais pas seulement).
 - Bluetooth permet de créer un réseau de 8 appareils en communication simultanée.
 - Adaptée à la mobilité (montre, PDA, voiture, espaces publics...),
 - Bluetooth est pour l'instant la seule norme qui a reçu de l'ART l'autorisation d'émettre en France sur la fréquence des 2,4 Ghz.
 - Une norme Bluetooth 2 est en cours de spécification : elle devrait permettre d'atteindre un débit théorique de 2 à 10 Mbits/s.

Périphérique wireless

Caractéristiques (4/4)

- **IEEE 802.11b**
 - Débit théorique de 11 Mbits/s
 - Rayon d'action est de 50 à 100 m.
 - Avec une antenne amplifiée de 100 milliWatt placée en extérieur, on peut couvrir de 400 mètres à 3 kilomètres
 - Utilise la bande de fréquence 2,4 Ghz
 - L'IEEE, qui est à l'origine de la norme, travaille à faire évoluer le standard 802.11b pour en améliorer le niveau de sécurité, le débit, la portée et l'efficacité énergétique.

Périphérique wireless

Transition – mise en sommeil

- **Techniques pour la mise en sommeil identiques que pour les disques**
 - Néanmoins, le seuil doit prendre en compte les temps nécessaire à la mise en sommeil et au réveil du périphérique
 - [SK97] WaveLan Pcmcia pour rattachement de documents Web via http diminue de 67% la consommation sans perception de latence pour l'utilisateur

Périphérique wireless

Adaptation – Extensions du protocole

- **Extension des fonctionnalités du protocole afin de diminuer la puissance d'émission du périphérique**
 - Exemple: Réseaux Adhoc / Spontaneous Information Systems
 - Problème: estimation de la distance

Périphérique wireless

Load/Change – Diminution des paquets à émettre

- **Extension des fonctionnalités du protocole afin de diminuer le nombre de paquets à émettre**
 - Compression de données
 - Suppression de certaines émission en fonction de la qualité de la transmission (c'est inutile d'émettre des paquets si ils vont être perdus)
 - Adaptation des données à transmettre en fonction des besoins des utilisateurs et des terminaux
 - Exemple: émission d'une image noir et blanc sur un terminal noir et blanc, et de la version couleur sur le terminal couleur

Intégration Wireless/processeur

Load/Change - Migration de processus

- Permettre de migrer une application Java depuis un PDA vers une station de travail et vice versa dans l'objectif de réduire la consommation énergétique du PDA
- Le lien vers le PDA est à conserver
 - Affichage des résultats, acquisition

