Ordonnancement

Yves Denneulin Yves.Denneulin@grenoble-inp.fr
Grégory Mounié

Problématique

- Occuper efficacement les ressources de calcul
  - sans consommer trop d'énergie
  - avec d'autres critères (réactivité, prise en compte des caches, etc.)

- Ordonnanceur(s)
  - fonctions de l'OS qui décident où et quand exécuter les tâches (≠processus) de la machine

Éléments de base

- Liste des tâches prêtes
  - ne connait que l'état actuel et le passé
- Quand processeur disponible choix dans cette liste

Un peu de théorie…

- ordonnancement de n processus sur m processeurs
- Deux valeurs clés
  - le nombre d'instructions de toutes les tâches / le nombre de processeurs
  - le chemin critique = le nombre d'instructions de la plus grande tâche

- Graham[66]
  - optimal : somme(C_i)/m + C_max
  - les algorithmes de liste seront au pire à un facteur 2
  - tâches séquentielles et processeurs homogènes
  - ici le but est la maximisation de l'occupation processeur
    - pas toujours le plus pertinent (jobs hétérogènes)

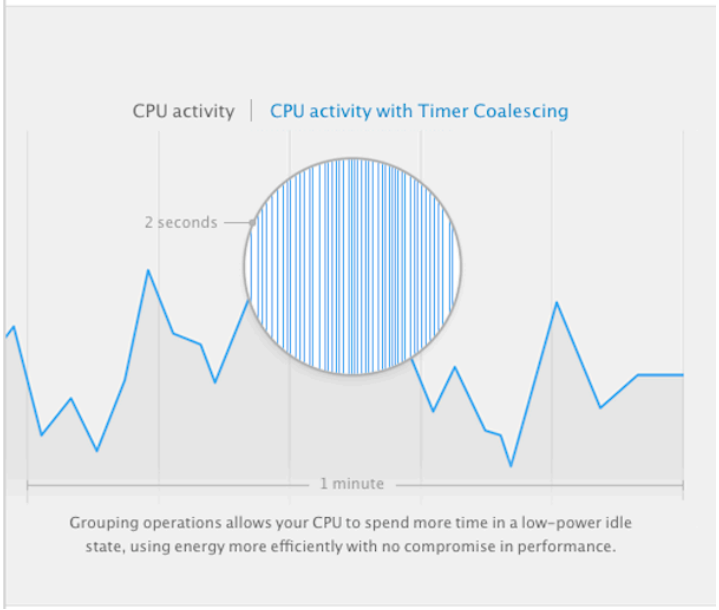
Choix du quantum

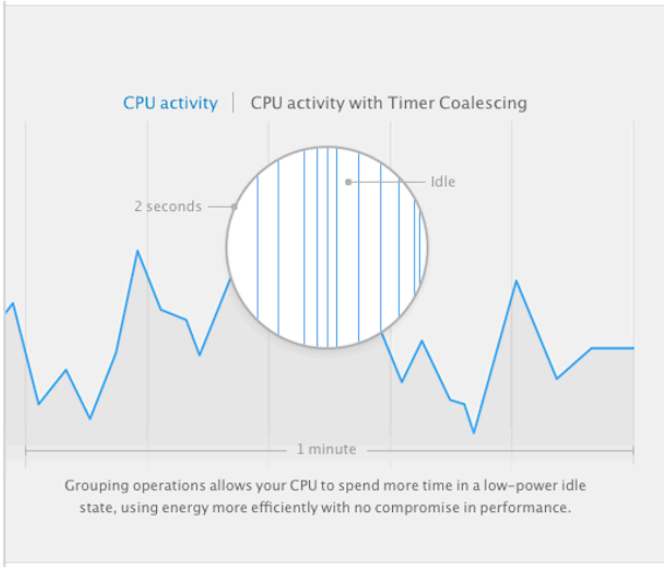
- délai entre deux opérations d'ordonnancement forcées
  - fréquence de l'horloge

- faible (50Hz) -> baisse de la réactivité
- haute (1000Hz) -> surcoûts
  - algorithme d'ordonnancement
  - vidage et mise à jour des caches + TLB

Interrupt coalescing

principe : retarder la prise en compte des interruptions pour laisser le processeur dormir plus longtemps





https://www.macgasm.net/news/reviews/os-x-10-9-mavericks-server-review/

Structure de données

- Liste peut ne pas être suffisante
- Critère : efficacité des opérations

- Tableau de listes (une par priorité)
- Arbres

Dans le noyau Linux, historiquement

- Une liste triée par goodness
  - f(priorité, quantums consommés)
  - on prend en tête O(1), on insère suivant la valeur O(n)
  - optimisation : liste des crédités, liste des expirés
- Multiprocesseurs :
  - un gros verrou pour l'accès à la liste
  - une liste par coeur (équilibrage de charge ?)

Multi-level feedback

- Multi-level : différentes classes de processus
  - RT, Système, interactive, Batch
  - avec des priorités décroissantes (table dans Windows)
  - et éventuellement des quantums différents aussi
- possibilité de passer de l'une à l'autre
  - le système décide
    - ex : I/O prioritaire
    - prend en compte le vieillissement du processus
  - avec risque de triche!

Autres ordonnancements

- Disque
  - ordre de traitement des requêtes d'accès (fusion, ré-ordonnancement)
- Réseau
  - avantager certains trafics sur d'autres