

# L'ÉNERGIE DANS LES BASES DE DONNÉES

Amine Roukh

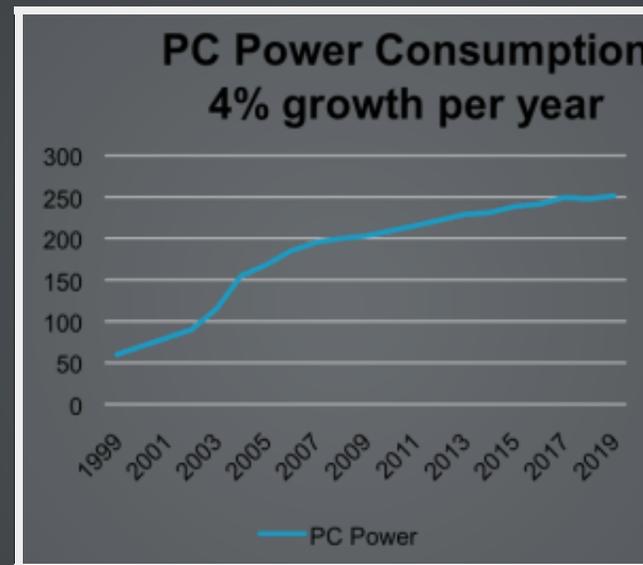
Faculté des Sciences Exactes & Informatique  
Université de Mostaganem  
aminroukh@gmail.com

# CONTEXTE

- L'amélioration de performance des systèmes informatiques reste l'objectif principal des chercheurs.
- L'énergie reste dans la 2ème position en terme d'importance, jusqu'à maintenant...

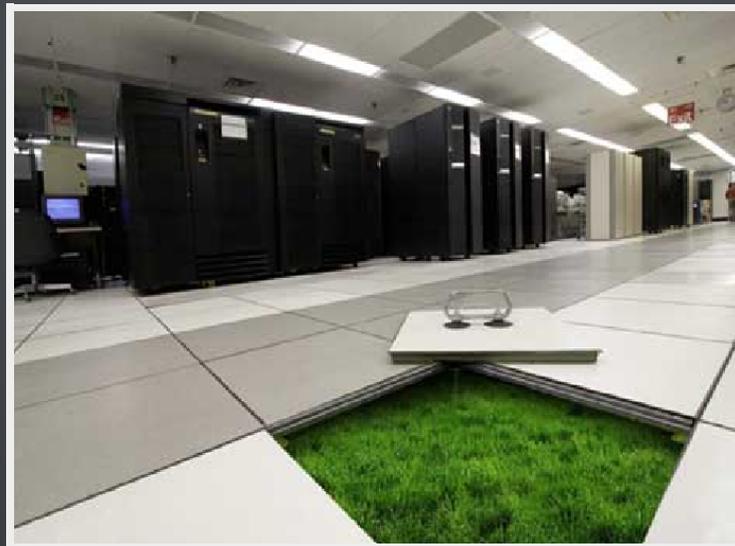


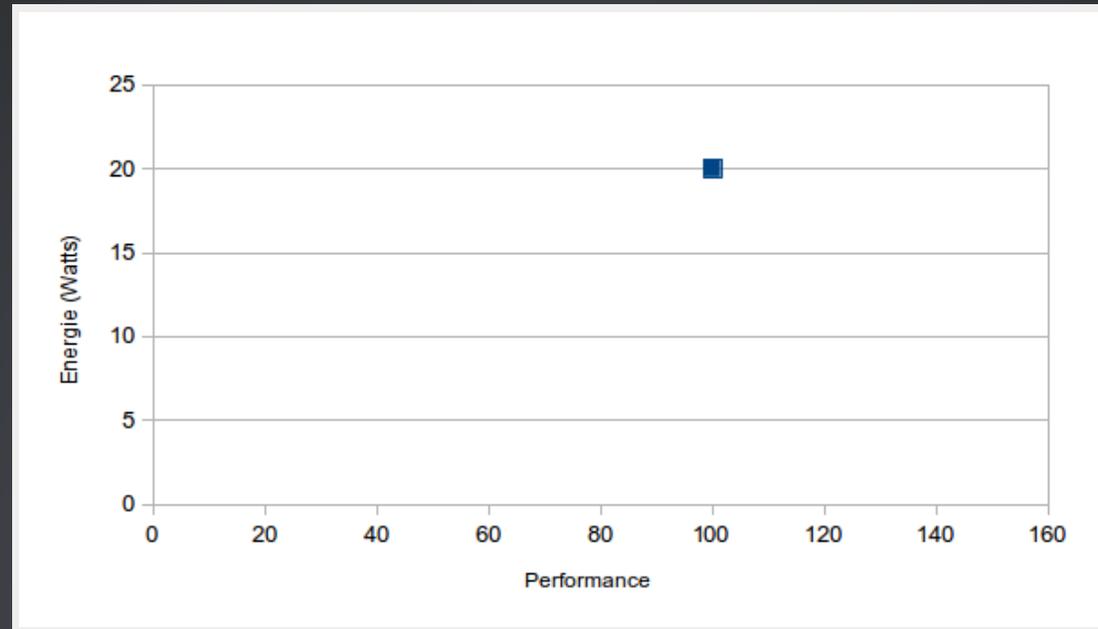
- L'électricité utilisée pour alimenter les ordinateurs dans le monde:  $\sim 225$  TWh en 2013.
- Ce nombre ne cesse de croître chaque année.
- Problème de coût, production de chaleur ( $CO_2$ ).



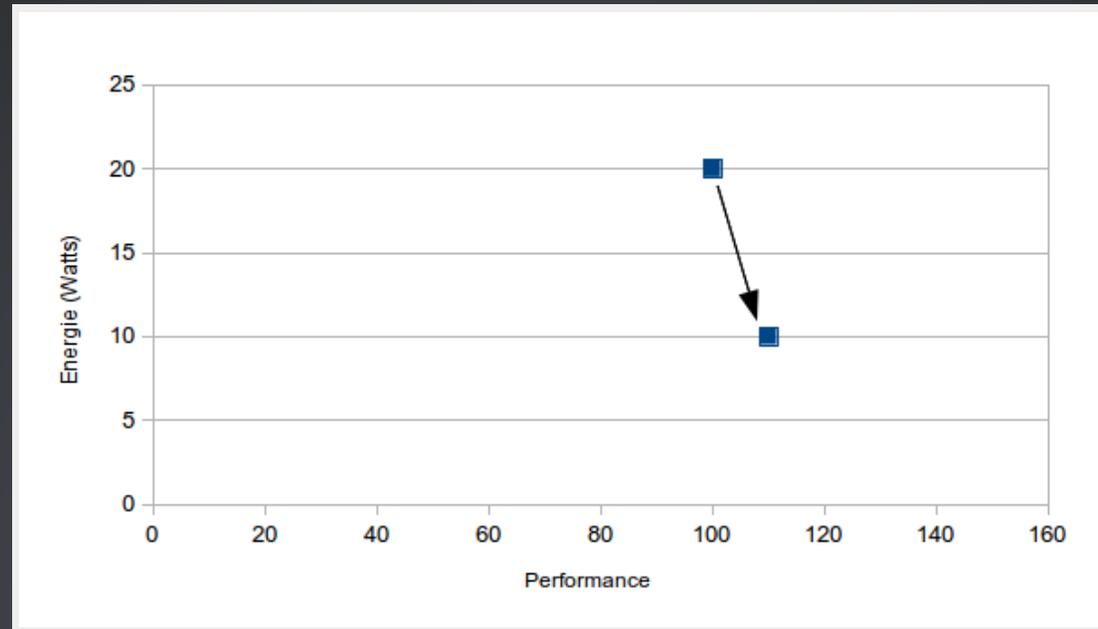
# L'ENERGIE DANS LES BASES DONNEES

- La gestion de l'énergie commence à attirer l'attention de la communauté des bases de données (depuis 2009).
- S'adapter dynamiquement et affiner les stratégies de gestion des données pour respecter le temps de réponse et permettre une meilleur gestion de l'énergie.





- La solution la plus performante consomme plus d'énergie



- Perdre quelque millisecondes et gagne plus d'énergie !

# DEUX APPROCHES

## APPROCHE MATÉRIEL:

- Un matériel avec faible profils d'énergie:
  - Solid State Drive (SSD) et la mémoire Flash.
  - Graphics Processing Units (GPU).
  - CPUs avec fréquence dynamique de tension (DVFS).

# APPROCHE LOGICIELS:

- Introduit des nouveaux algorithmes ou refonder les SGBDs pour réduire la consommation d'énergie.
  - QED : Improved Query Energy efficiency by Introducing Explicit Delays.
  - PET : (Power Energy Time).
  - E<sup>2</sup> DBMS, P-DBMS.

# MOSELISATION D'ENERGIE

- Pour estimer l'énergie d'une requête *SQL* il faut modéliser chaque opérateur algébrique.
- Sélection, Projection, Jointure.
- Avec différents algorithmes (accès séquentiel, index, boucle imbriquée...).
- Utiliser la régression linéaire.

# LA RÉGRESSION LINÉAIRE

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_n x_{i,n} + \epsilon_i$$

- $y = [y_1, y_2, \dots, y_k]$  : vecteur des puissances mesurées.
- $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$  : vecteur des  $n$  variables.
- $\epsilon_i$  : erreur des mesures.
  
- $x_i$  est lu à partir des statistiques de la base de données.
- L'estimation des paramètres de modèle  $\beta$  résoudre l'équation.

# MOSELISATION DES OPERATEURS ALGEBRIQUES

- Implémentation du modèle de Lang *et al* pour les opérations *SQL* de type accès séquentiel et jointure.
- Pas de cache, pas de requêtes parallèle.
- Supposons les deux tables  $R$  et  $S$ , où  $S$  est table la plus large.
- $||R||$  et  $||S||$  le cardinal des relations.
- $sel$  est la sélectivité des prédicats (connue).

Pour exécuter une requête de type accès séquentiel sur la table  $R$ :

- Le CPU doit effectuer une comparaison pour chaque tuple.
- Une opération de sauvegarde pour les tuples qui satisfont le prédicat.

L'équation de prédiction peut être écrit comme suit :

$$y = \beta_0 + \beta_1 ||R|| + \beta_2 sel * ||R||$$

On suppose que le SGBD utilise la jointure par hachage avec l'algorithme *Grace*:

- Partitionner  $R$  et  $S$  avec une fonction de hachage.
- Chaque partition de  $R$  est lue dans la mémoire.
- Chaque tuple dans la partition correspondante dans  $S$  est lu et comparé.
- S'il y a une correspondance, le tuple est ajouté au résultat.

L'équation devient :

$$y = \beta_0 + \beta_1(2 * ||R|| + ||S||) + \beta_2||S|| \\ + \beta_3(||R|| + ||S|| + sel * ||R|| * ||S||)$$

# APPRENTISSAGE

1. Dans une deuxième étape il faut trouver les  $\beta_i$ , en effectuant une série d'observations.
2. Lors d'une observation, la puissance est capturée par un équipement de mesure.
3. De même, les paramètres de requête sont lues à partir de la base de données.

⇒ Cela conduit à un système d'équations linéaires de forme:

$$y_1 = \beta_1 x_{1,1} + \beta_2 x_{1,2} + \dots + \beta_n x_{1,n}$$

$$y_2 = \beta_1 x_{2,1} + \beta_2 x_{2,2} + \dots + \beta_n x_{2,n}$$

$$\dots$$
$$y_i = \beta_1 x_{i,1} + \beta_2 x_{i,2} + \dots + \beta_n x_{i,n}$$

⇒ On obtient le meilleur ajustement des coefficients  $\beta$  par une régression linéaire.

Nous avons créé une charge de requêtes de type accès séquentiel, de forme :

```
SELECT * FROM LINEORDER
WHERE L_ORDERKEY > k;
```

Pour la jointure par hachage :

```
SELECT * FROM LINEORDER, PART
WHERE LO_PARTKEY = P_PARTKEY
AND LO_ORDERKEY > k;
```

Le paramètre  $k$  permet de contrôler la sélectivité des prédicats.

# TESTE

Nous avons crée une autre charge de requêtes avec des sélectivités différentes pour la phase de teste.

| Req | Tuple | Train | Test |
|-----|-------|-------|------|
| 1   | 200   | 0,1   | 0,3  |
| 2   | 200   | 0,5   | 0,6  |
| 3   | 200   | 0,7   | 0,8  |
| 4   | 200   | 0,9   | 1,0  |
| 5   | 6001  | 0,1   | 0,2  |
| 6   | 6001  | 0,3   | 0,4  |
| 7   | 6001  | 0,5   | 0,6  |
| 8   | 6001  | 0,7   | 0,8  |
| 9   | 6001  | 0,9   | 1,0  |

(a) Accès séquentiel

| Req | Tuple      | Train | Test |
|-----|------------|-------|------|
| 1   | 200 ⋈ 6001 | 0,1   | 0,2  |
| 2   | 200 ⋈ 6001 | 0,3   | 0,4  |
| 3   | 200 ⋈ 6001 | 0,5   | 0,6  |
| 4   | 200 ⋈ 6001 | 0,7   | 0,8  |
| 5   | 200 ⋈ 6001 | 0,9   | 1,0  |

(b) Jointure par hachage

Les requêtes et leur sélectivités (Tuple X1000).

# EXPERIMENTATIONS ET RESULTATS

- HP Compaq 600 Pro, Intel Core2 Duo E7500 2.93GHz.
- 2x1Go RAM et un HDD Hitachi de 300Go.
- Multimètre numérique Uni-Trend UT805A.
- Le SE est Linux Kubuntu 13.10.
- Le SGBD est Oracle 11gR2.
- Entrepôt de données SSB (Star Schema Benchmark) de 1Go.
- Le langage R pour la régression.

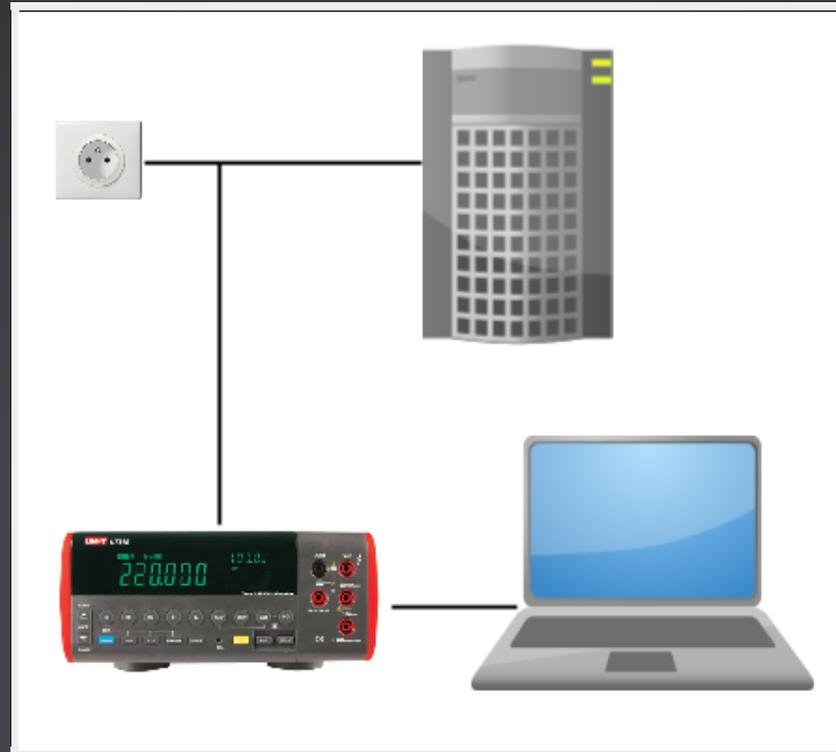


Schéma de simulation.



Architecture de simulation.

Erreur:  $\pm 3.2$  Watts.

Erreur:  $\pm 4.3$  Watts.

# PROBLEME

Validation du modèle avec un autre entrepôt de données  
(BENCHW).

Erreur:  $\pm 23.8$  Watts.

# CONCLUSION & PERSPECTIVES

- Le problème de la modélisation d'énergie dans les base de données.
- Expérimentations avec une méthodologie d'estimation d'énergie.
- La méthodologie est basée sur l'apprentissage et la régression linéaire.
- La cardinalité de la table ainsi que le nombre de colonnes et la sélectivité influencent sur la consommation d'énergie.

- Modéliser la consommation de la mémoire, le disque dur.
- Développer un modèle de coût énergétique.
- Intégrer le modèle dans une structure d'optimisation.

# RÉFÉRENCES

- McCullough *et al.* Evaluating the effectiveness of model-based power characterization. 2011
- Xu *et al.* Exploring power-performance tradeoffs in database systems. 2010.
- Lang *et al.* Rethinking query processing for energy efficiency: Slowing down to win the race. 2011.
- Xu *et al.* Dynamic Energy Estimation of Query Plans in Database Systems. 2013.