Modèles de coût analytiques pour le traitement parallèle des requêtes

Par:
DEMBELE Simon Pierre
Ensma

Encadré par:
BELLATRECHE Ladjel
Ensma

Plan

- Contexte
- Analyse de l'Etat de l'art.
- Traitement Parallèle.
- Modélisation.
- Perspectives.

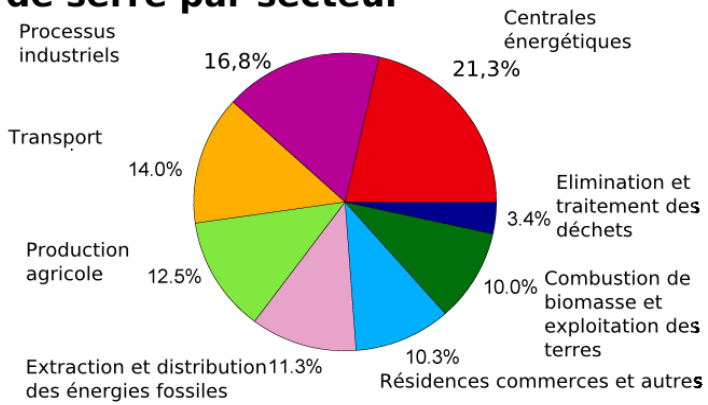
Contexte (I)

Les facteurs de pollution

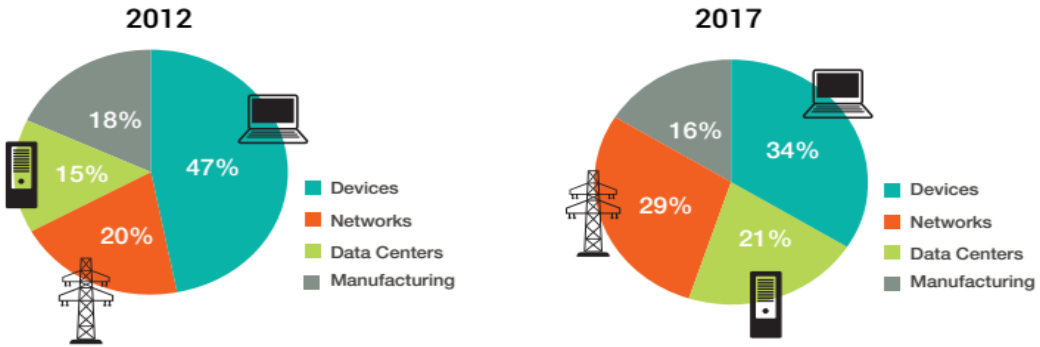
Conséquences directes sur :

- Environnement;
- Santé ;
- Economie
- …

Emissions annuelles de gaz à effet de serre par secteur



Consommation énergétique dans le secteur IT



Contexte (II) : Initiatives

Agences et gouvernements(Etats): COP21 (Paris), COP 23 (Bonn)

Normes: ISO 14001 etc.

Bâtiments : règles de construction, type d' équipement

Véhicules: Hybride, etc.

Energie

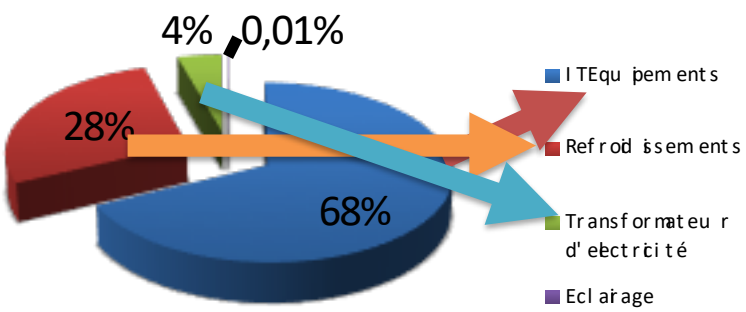
IT: Flash, etc.

Usagers: Tri vert, recyclage, Stop Pub etc.

Energie renouvelables : éolien, solaire, etc.

Contexte (III): Energie dans le Service IT

Répartition de l' énergie



Les SGBD sont les plus gros consommateurs d'énergie électrique

Composants d'un data center

- **Matériel**

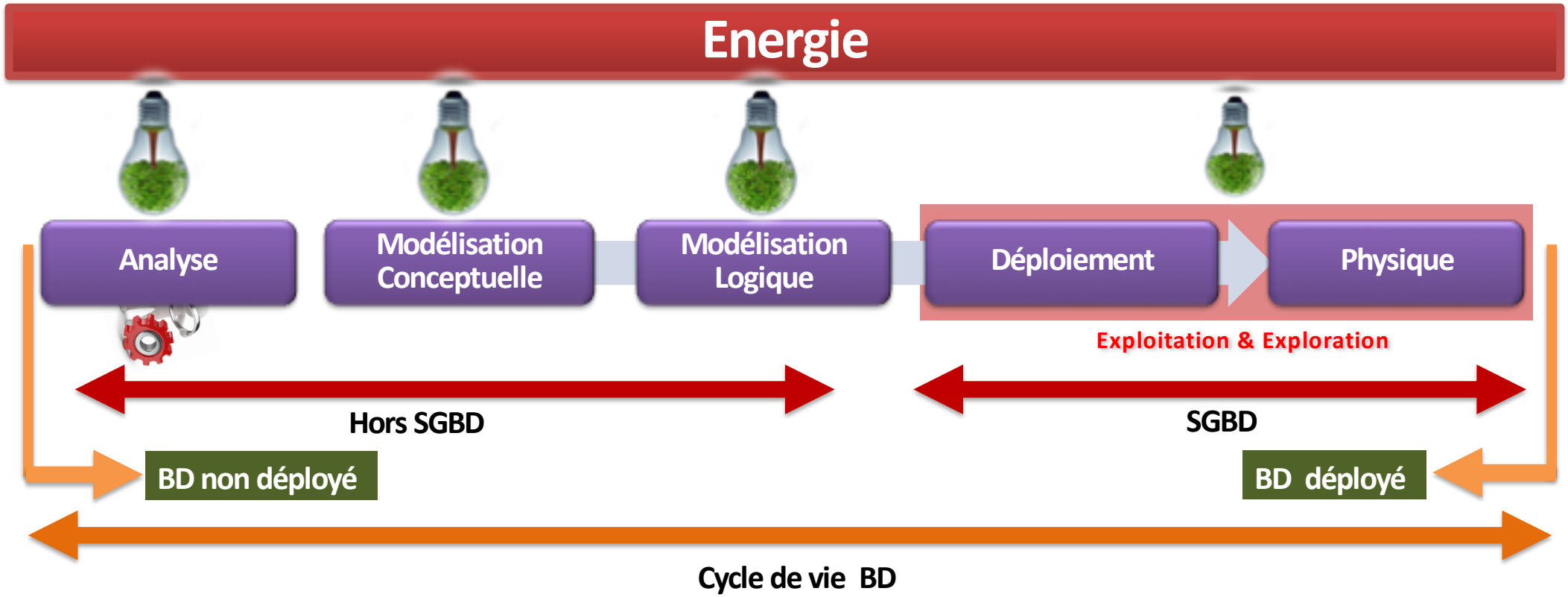
- **Logiciel**

Energie

- **Approches matérielles :**
  - ✓ Efficaces
  - ✓ Robustes
  - ✓ Non-portables
- **Approches logicielles :**
  - ✓ Efficaces
  - ✓ Robustes
  - ✓ Portables
  - ✓ Simples
- **Système de gestion de base de données (SGBD):**
  - → sensibles aux besoins non fonctionnels (QoS: performance, etc.)

- **Avec la dimension Energie**
  - → La question : **à quel niveau du SGBD l'énergie est-elle intégrée ?**

Intégration de l'énergie dans les SGBD(I)



Intégration de l'énergie dans les SGBD(II)

BD Déployés

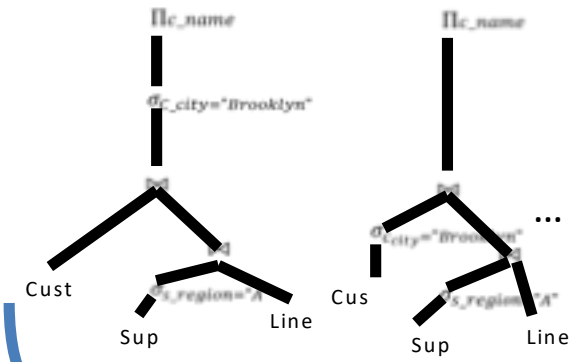
- **Traitement requête**
  - + plans d'exécution (PE) pour la même requête
  - + algorithmes pour la même opération
  - …
- **Conception Physique**
  - + structures d'optimisations (SO)
  - + Sélection d'une SO (exple. index) est un problème NP-difficile
  - …

Traitement d'une requête

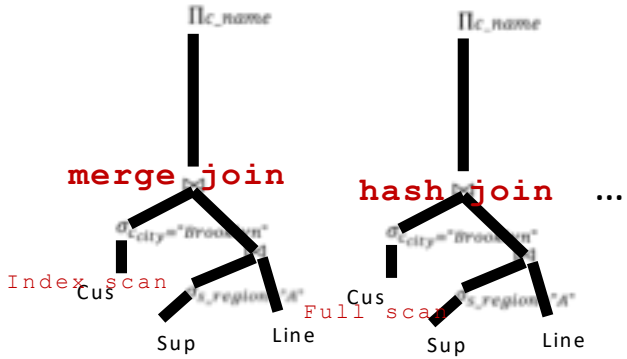
Requête SQL

- Analyse Syntaxique
- Transformation
- Planification/ Optimisation
- Plan d'exécution

Plans physiques



Plans d'exécution (+ algorithmes pour chaque nœuds)

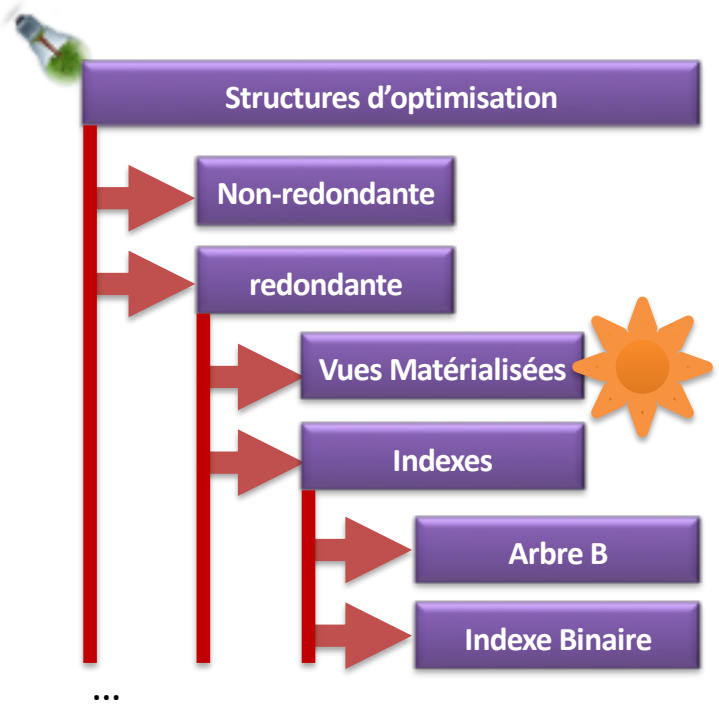


Critère BNF (**modèle de coût**) orienté temps de réponse , énergie…

!!! Bon plan

Intégration de l'énergie dans les SGBD(III)

Techniques d'optimisation (conception physique)



Cas: Vues matérialisées (VM)

Formalisation du problème de sélection des VM

1. Un schéma de BD
2. Une charge requête Q {Q1,⋯, Qn}
3. Un plan d'exécution PE={PE1…}
4. Besoins non-fonctionnels BNF
5. Un ensemble de contraintes C{C1,⋯, Cp}

= BNF + (Energie comme BNF) ou C + (Energie comme contrainte)

Objectif : sélectionner un PE pour Q satisfaisant BNF (modèle de coût) et respectant C

Modèles de coût(I)

Une fonction dédiée à l'estimation des BNF via des formules mathématiques.

Démarche pour sa définition

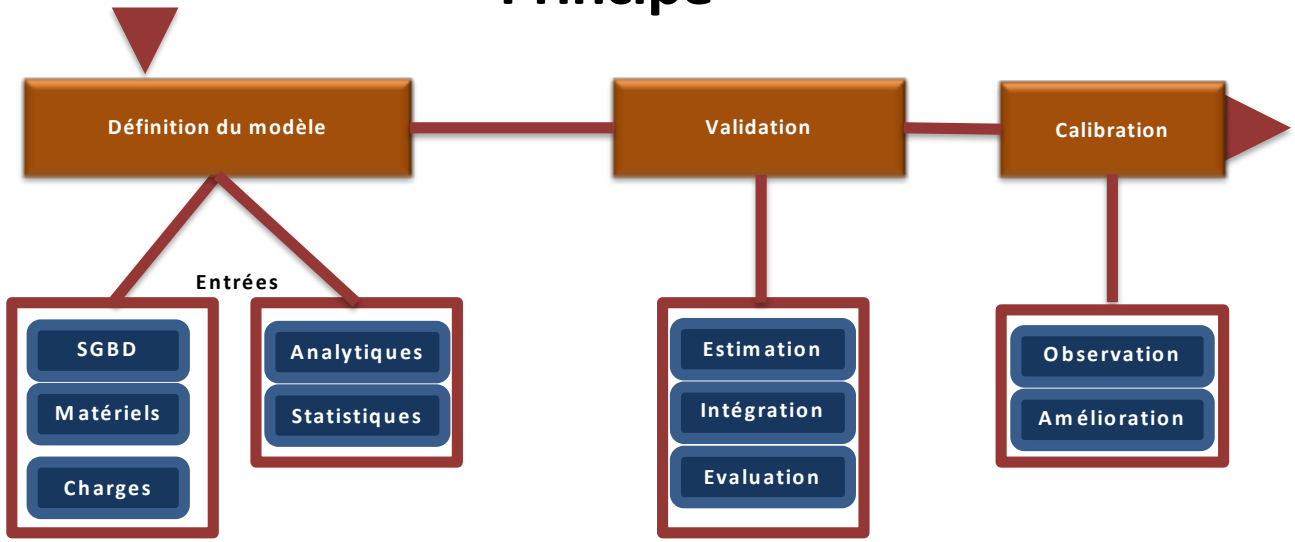
- Modélisation l'environnement (optimiseur de requête, ou couche physique)
- Paramètres du modèles
- Sa signature
  - MC(paramètres) → valeur

Démarche pour sa validation

- Simulation
- Par injection dans un SGBD opérationnel

Objectif : simple, précis, rapide, faible surcharge…

Principe



- *Niveau de modélisation ?*
- *Mode d'exécution ?*
- *Paramètres de modèle ?*

Modèle de coût(II)

PostgreSQL 10.1

- **Paramètres du modèle**

→ seq_page_cost	Cost of a sequential page fetch
→ random_page_cost	Cost of a non-sequential page fetch
→ cpu_tuple_cost	Cost of typical CPU time to process a tuple
→ cpu_index_tuple_cost	Cost of typical CPU time to process an index tuple
→ cpu_operator_cost	Cost of CPU time to execute an operator or function
→ parallel_tuple_cost	Cost of CPU time to pass a tuple from worker to master backend
→ parallel_setup_cost	Cost of setting up shared memory for parallelism

- **Signature**

$$cost = \alpha * cost_{cpu} + \beta * cost_{disk} + \delta * cost_{com}$$

```
        /*      * disk costs        */
→disk_run_cost = spc_seq_page_cost * baserel->pages
        /* CPU costs */
→cpu_run_cost = cpu_per_tuple * baserel->tuples
→cpu_run_cost += path->pathtarget->cost.per_tuple * path->rows
→total_cost = startup_cost + cpu_run_cost + disk_run_cost
```

Etat de l'art(I)

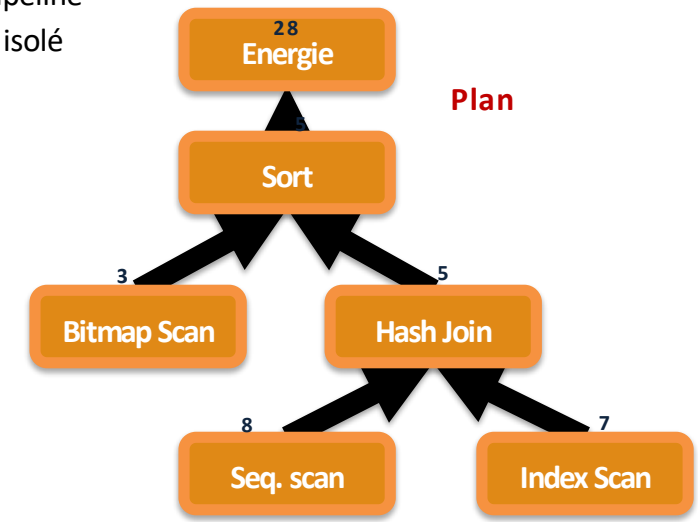
➢ **Dynamic Energy Estimation of Query Plans in Database Systems [Zichen Xu, Yi-Cheng Tu, Xiaorui Wang] en 2013.**

→ **Chaque nœud de l'arbre algébrique est associé à un facteur énergétique.**

→ **Le coût global est la somme des coûts de chaque opérations**

→ Pas de pipeline

→ Requête isolé



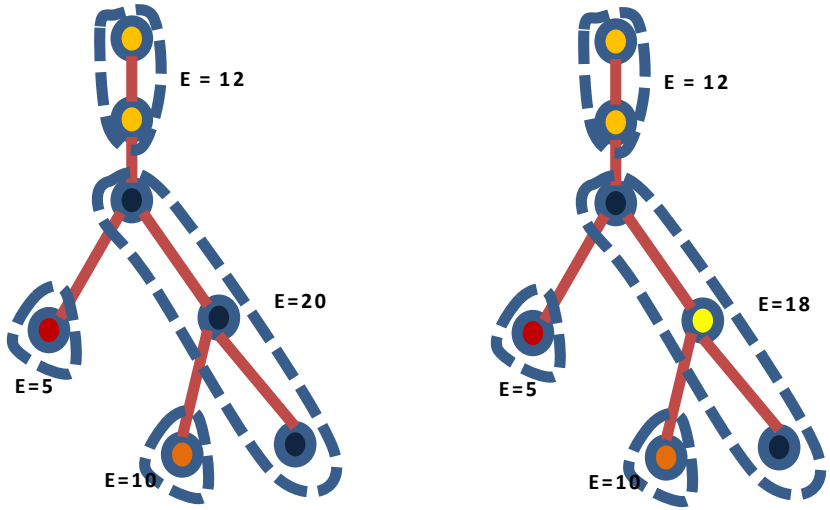
Plan

Energie de chaque opération est estimée

➢ **Peak Power Plays in Database Engines [Mayuresh Kunjir Puneet K. Birwa Jayant R. Haritsa]** en 2012.

→ **Exemple d'intégration du pipeline**

Plan



Energie de chaque pipeline est estimée

Etat de l'art (II)

- **Eco-Processing of OLAP Complex Queries [Amine ROUKH, Ladjel BELLATRECHE]** en 2015.

- **Estimating Power Consumption of Batch Query Workloads Queries [Amine ROUKH,]** en 2015

Niveau de modélisation: pipelines
Mode d'exécution: isolée et concurrentes (batch)

Pipeline : un sous arbre d'opérateurs exécutant simultanément délimités par des opérateurs bloquant

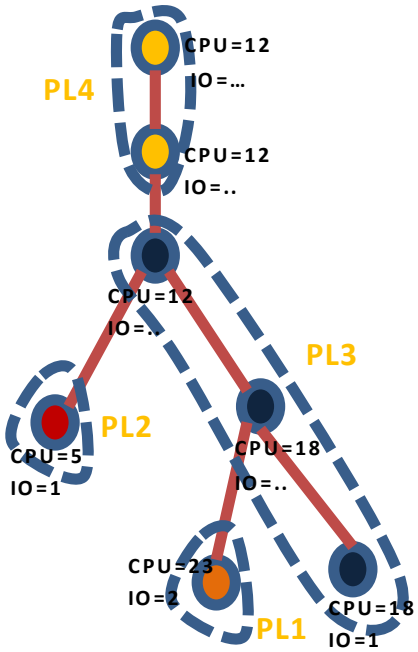
Coût Plan: c'est la somme des coûts du *CPU* de chaque opération et les coûts des entrées sorties *I/O*

Coût Energie : c'est la somme des coûts énergétique de chaque pipeline.

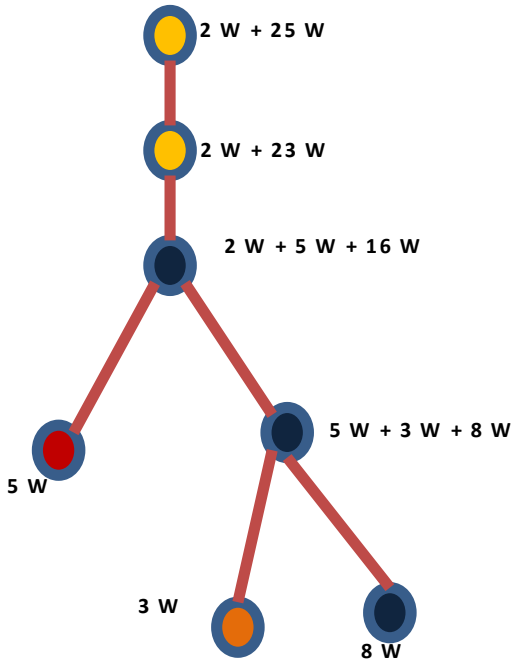
Puissance Pipeline

$$Power(PL_j) = \beta_{cpu} \times \sum_{k=1}^{n_j} CPU_COST_k + \beta_{io} \times \sum_{k=1}^{n_j} IO_COST_k$$

Plan en pipeline

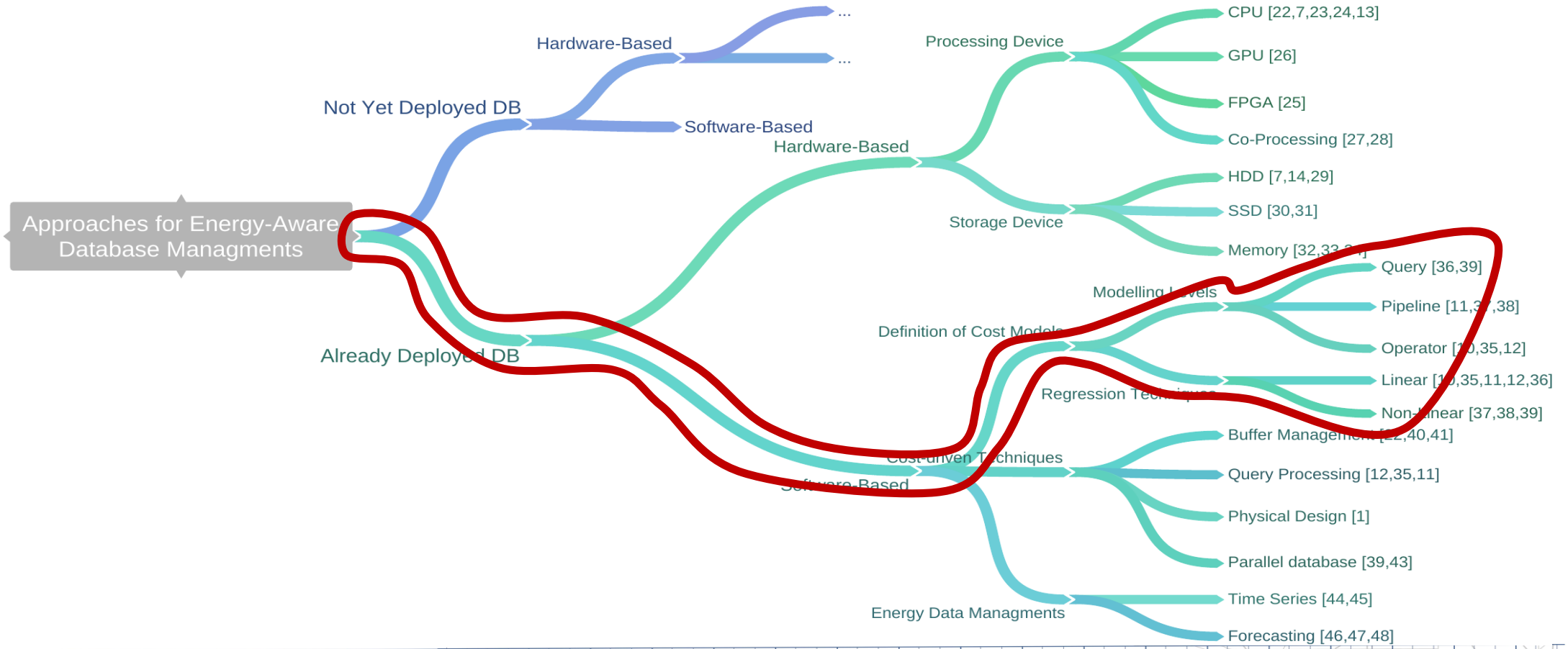


Coût Energie



Etat de l'art (III)

Classification des méthodes

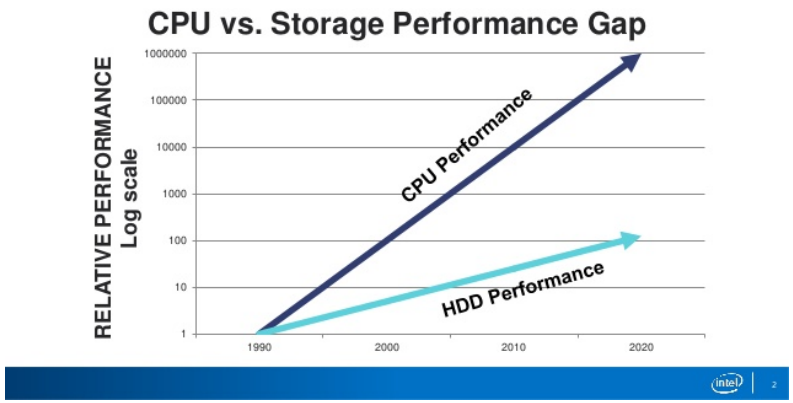


Limites

- Modèle basé sur le plan séquentiel
- Mode d'exécution isolée séquentiel

Motivation (I)

→L'évolution fulgurante des composants hardwares des machines a favorisé et motivé les traitements parallèles des applications.



Challenge:
Comment revisiter les modèles de coût séquentiels pour intégrer le parallélisme pour estimer l' énergie.

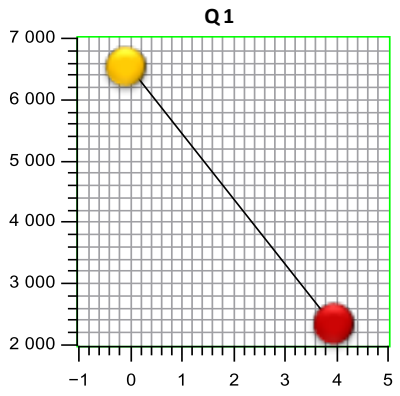
PostgreSQL (9.6; …) traitement parallèle (TP) des requêtes

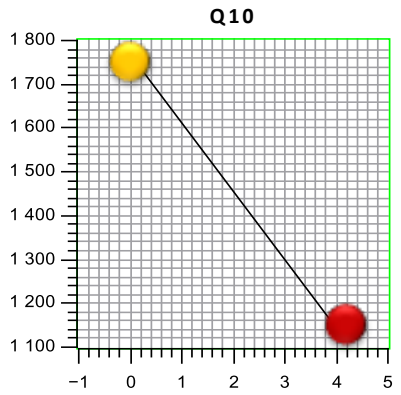
Paramètres

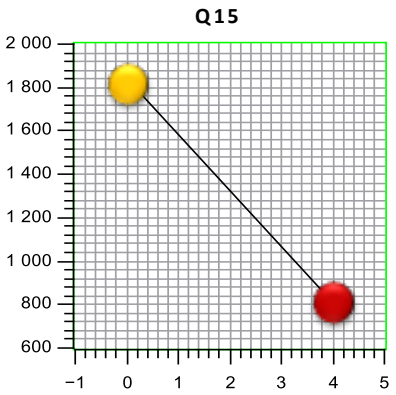
→ parallel_tuple_cost Cost of CPU time to pass
→ parallel_setup_cost Cost of setting memory for parallelism

Motivation(II)

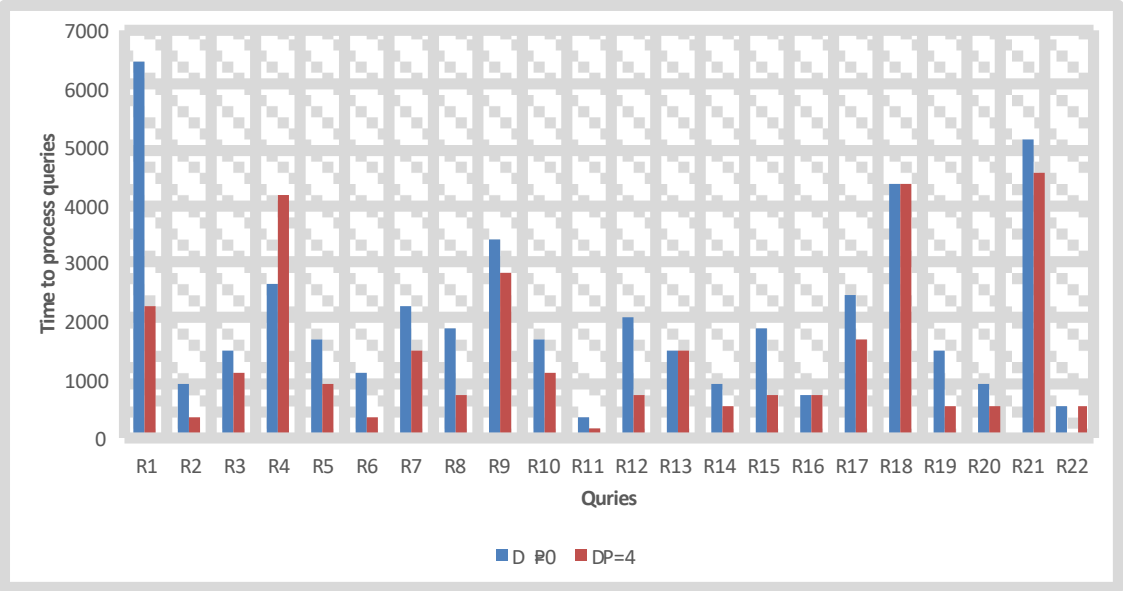
❑ Temps d'exécution des requêtes Q1, Q10, Q15 du TPC-H en séquentiel puis en parallèle degré de parallélisme fixé à 4.







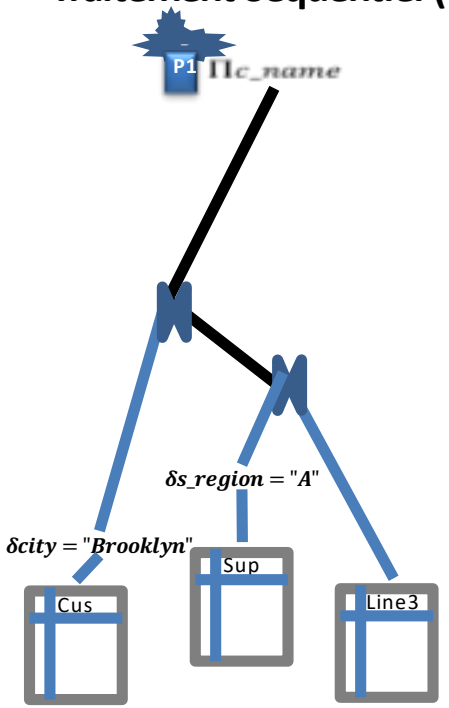
❑ Les 22 requêtes du TPC-H en séquentiel puis en parallèle degré de parallélisme fixé à 4.



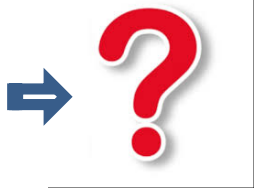
✓ Les modèles de coût séquentiels ne sont plus adaptés
→ le temps de réponse du traitement est meilleur
→ et la consommation énergétique

Traitement parallèle (I)

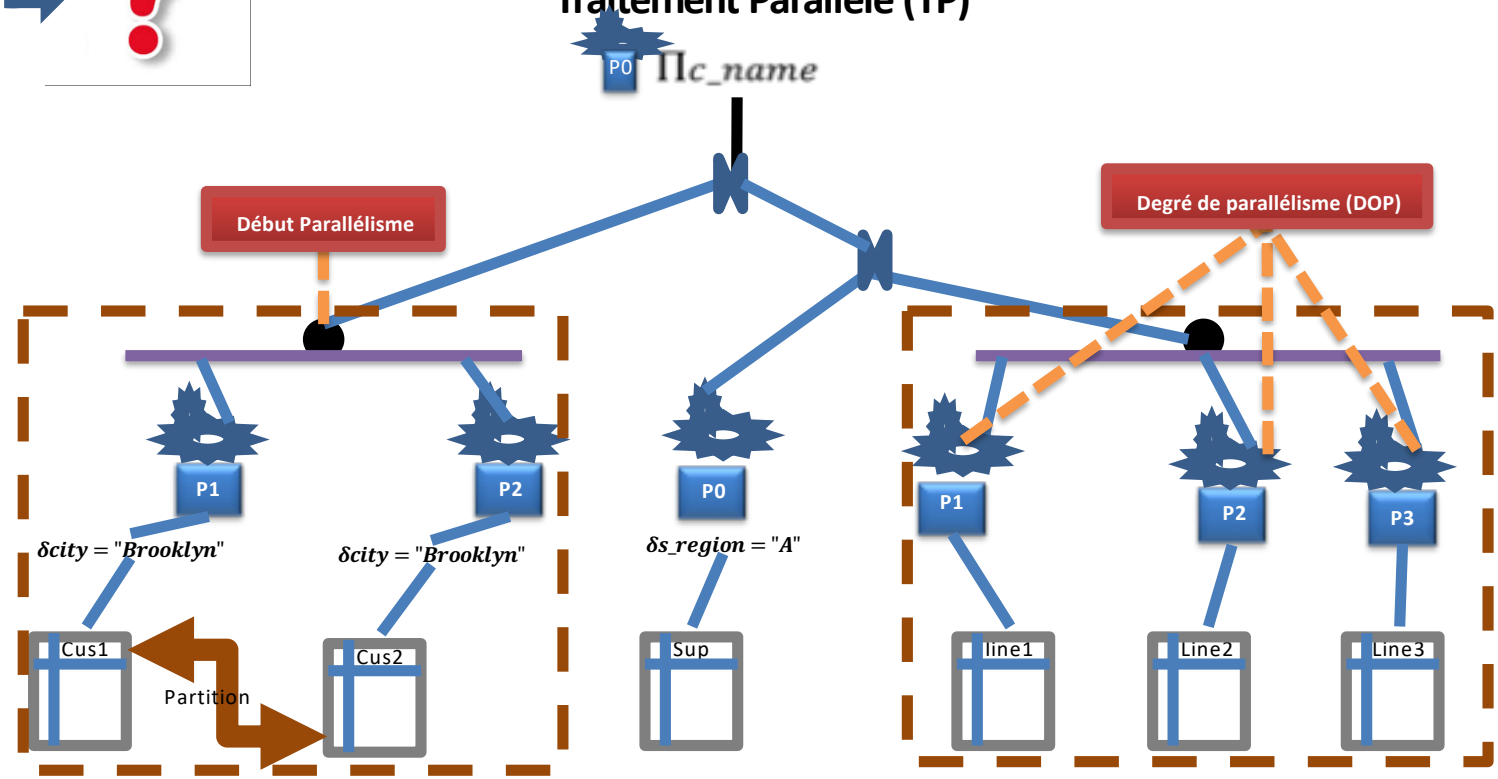
Traitement Séquentiel (TS)



✓ Un seul point de traitement



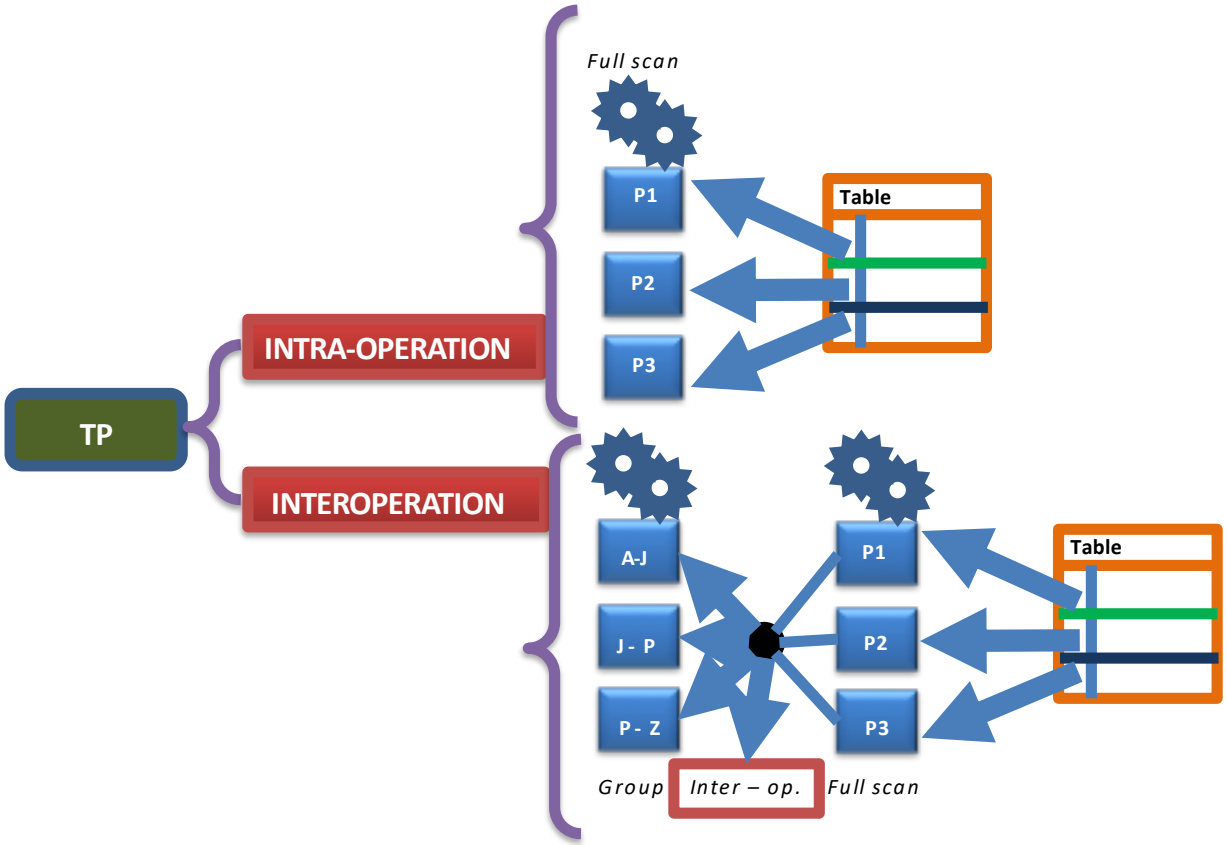
Traitement Parallèle (TP)



✓ Plusieurs points de traitement DOP=2

Traitement parallèle (II)

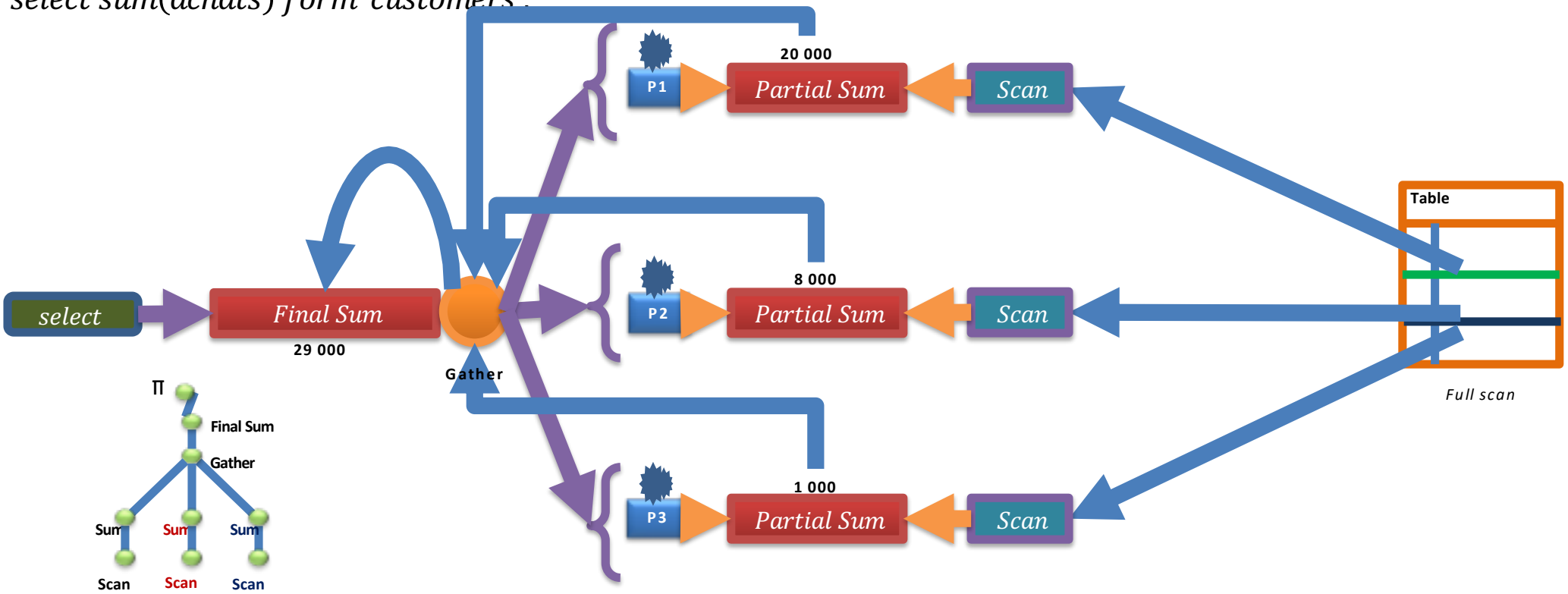
Niveau de parallélisme



Traitement parallèle (V)

Requête SQL

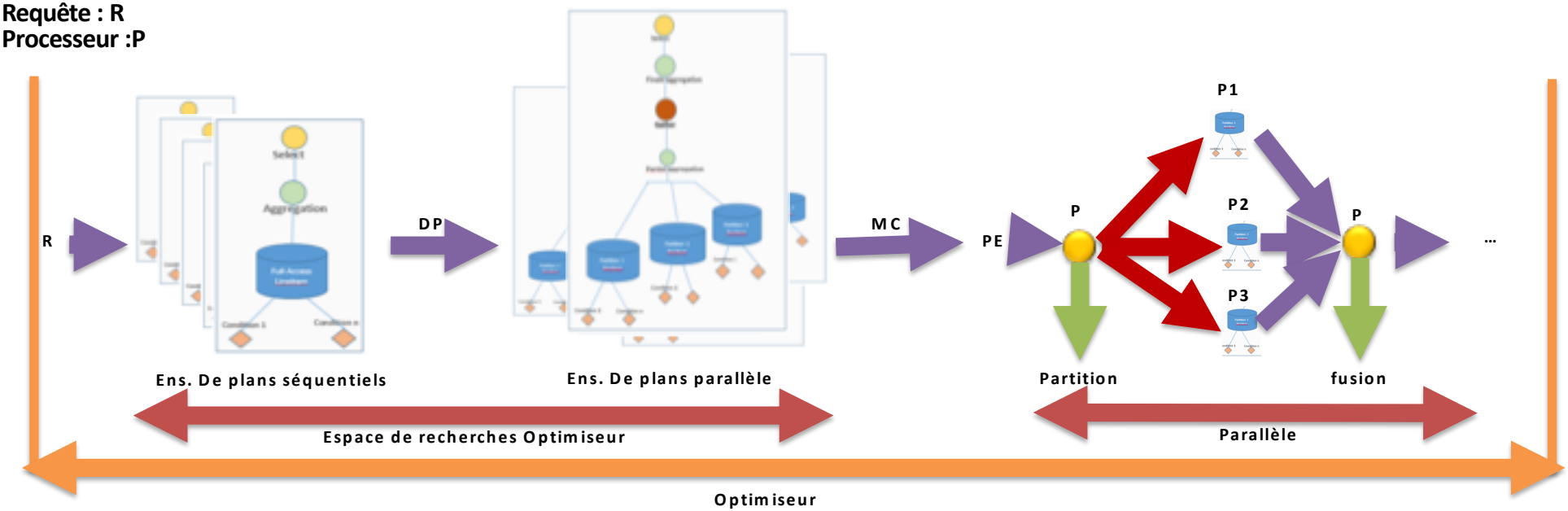
select sum(achats) form customers ;



Traitement parallèle (III)

Degré de parallélisme : DP
Modèle de coût: MC

Requête : R
Processeur :P



- ✓ Très complexe …
- ✓ Plusieurs techniques en même temps (partitionnement, repartitionnement, regroupement….)

Traitement parallèle (IV)

Commande explain (PostgreSQL)

Finalize Aggregate (cost=**1517131.87**...1517131.88 ....)
-> Gather(cost=1517131.45..1517131.86 ...)
Workers Planned: 3
-> Partial Aggregate (cost=1516131.45...1516131.46 .....)
-> Parallel Seq Scan on lineitem (cost=0.00..1478630.36....)

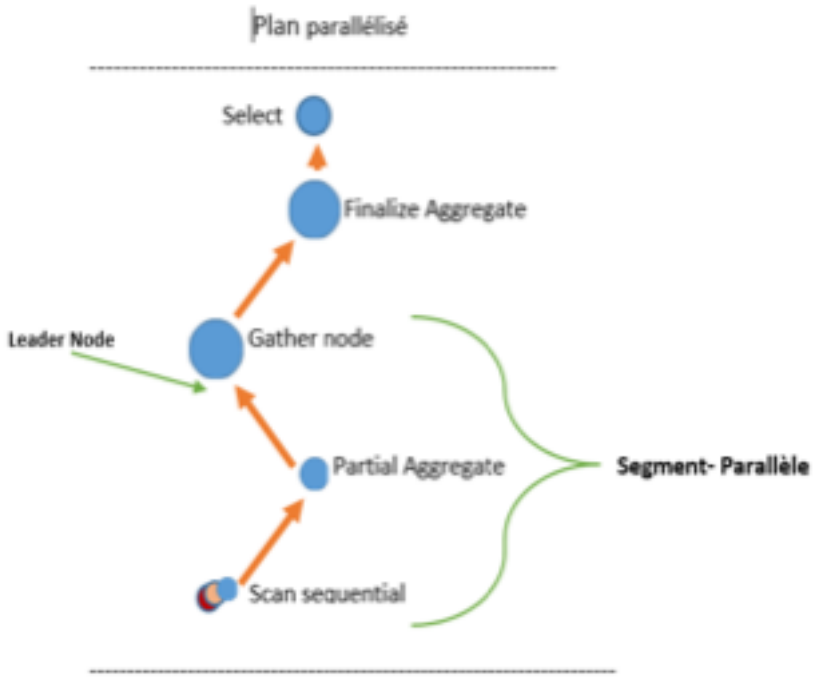
Nœud Parallélisme

DOP

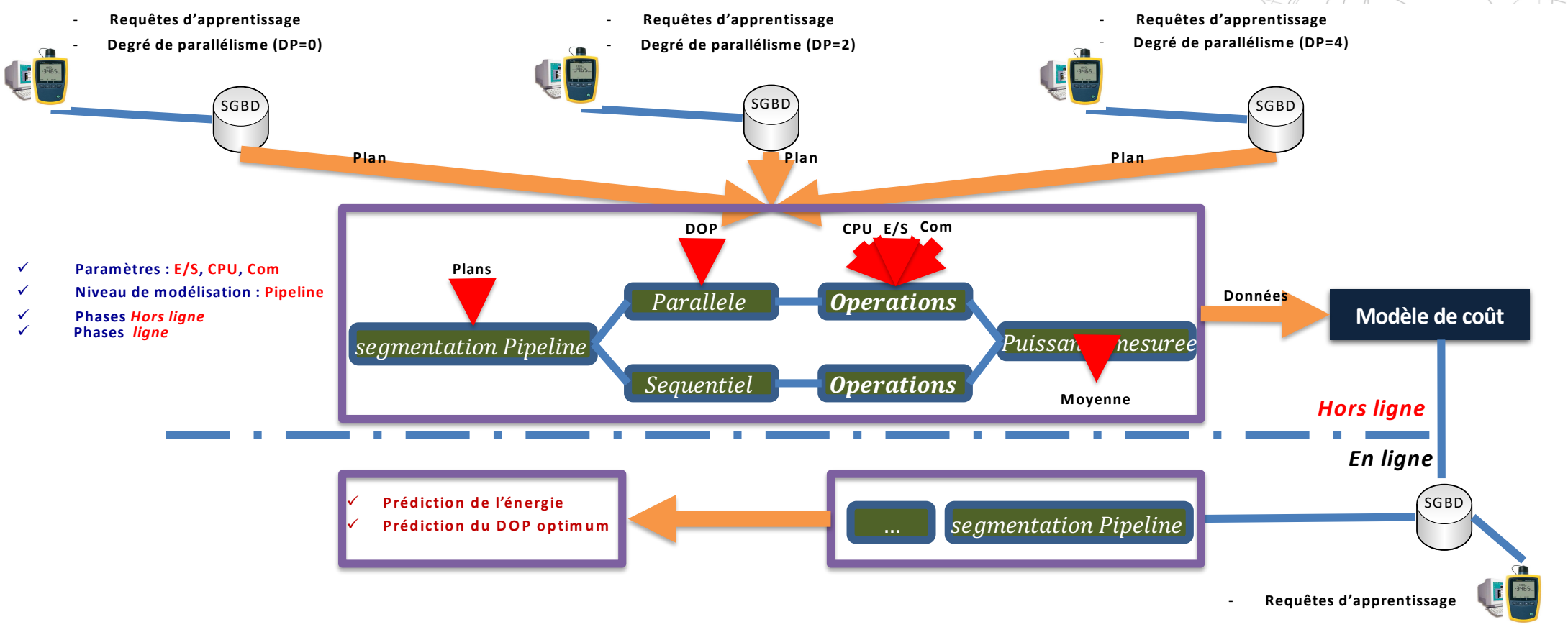
En séquentiel

Aggregate (cost=**2078647.80..2078647.81**...)
-> Seq Scan on lineitem(cost=0.00..1928643.44)

Exemple PostgreSQL



Approches Modélisation



Modélisation

- L'optimiseur segmente le plan en en un ensemble de pipelines, les pipelines non parallélisables Pl_i et les pipelines parallélisables PPl_i.

$$time(plan,1) = \sum_{1}^{i} time(Pl_i,1) + \sum_{1}^{J} time(PPl_j,1)$$

$$E(plan,1) = \sum_{1}^{i} E(Pl_i,1) + \sum_{1}^{j} E(PPl_j,1)$$

- Les pipelines parallélisables contiennent des segments-parallèles qui peuvent s'étendre sur tout le pipeline ou sur une portion débutant par Gather.

$$time(plan,n) = \sum_{1}^{i} time(Pl_i,1) + \frac{1}{n}\sum_{1}^{j} time(PPl_j,1) + time(com,p)$$

$$E(plan,n) = \sum_{1}^{i} E(Pl_i,1) + \sum_{1}^{j} E(PPl_j,n) + E(c,p)$$

- Le coût de puissance moyenne pour un pipeline PL avec n opérateurs est :

$$P(PP_j,1) = \beta_{cpu} * \sum_{K=1}^{n_{cpu}} COUT_{CPU_{cpu}} + \beta_{es} * \sum_{K=1}^{n_{es}} COUT_{ES_{es}}$$

- Le coût de puissance moyenne pour un pipeline parallèle PPL avec n opérateurs est :

$$\boxed{P(PP_j,n) = (f_{c_n}+1) * \beta_{cpu} * \sum_{K=1}^{n_{cpu}} COUT_CPU_{cpu} + \beta_{es} * \sum_{K=1}^{n_{es}} COUT_ES_{es}}$$

- β_{es}, β_{cpu} représentent les unités énergétiques pour l'E/S et pour le coût CPU.

Architecture

Environnement

→ Dell inspiron T1700, 3.40 GHz 4 cœurs 8 processeurs logiques) de 8 Go de Ram and 1TB HDD; kubuntu 17.10

→ PostgreSQL 10.1

→ R langage pour la régression (paramètres)

Base de données

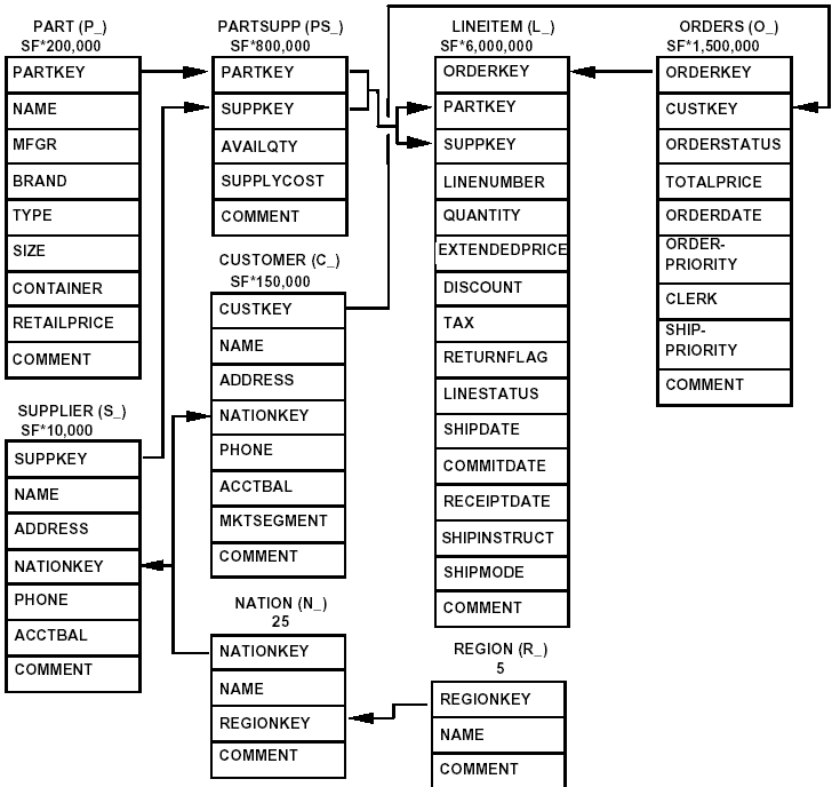
→ Benchmark = TPC-H

→ Taille = 1 Go, 5 Go, 10 Go, 30 Go

Mesure d'énergie :

→ Watts UP? Pro ES multimètre

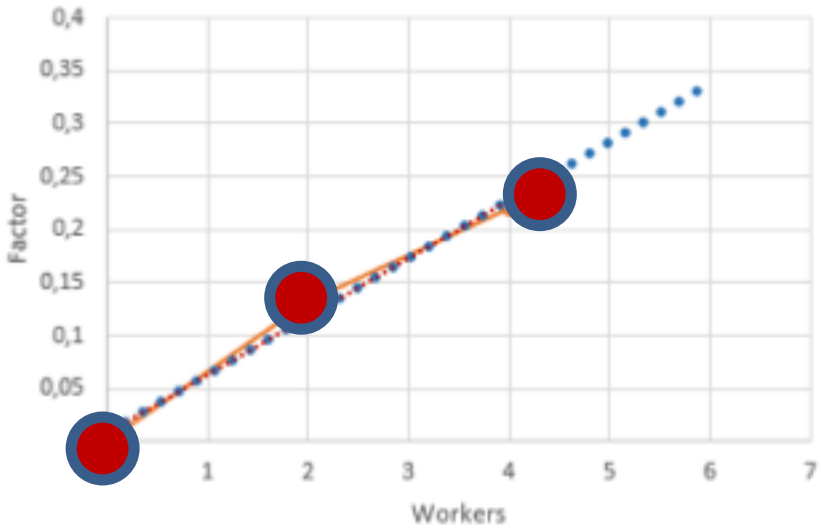
→ Fréquence de 1 hertz



Résultats (I)

Le paramètre f_{c_n}, est définit en fonction du nombre de workers.

$$f_{c_n} = (P_{m_n} - P_0)/P_o$$

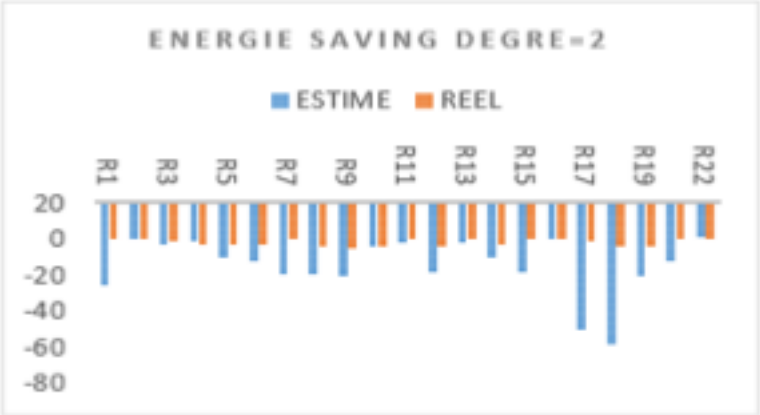


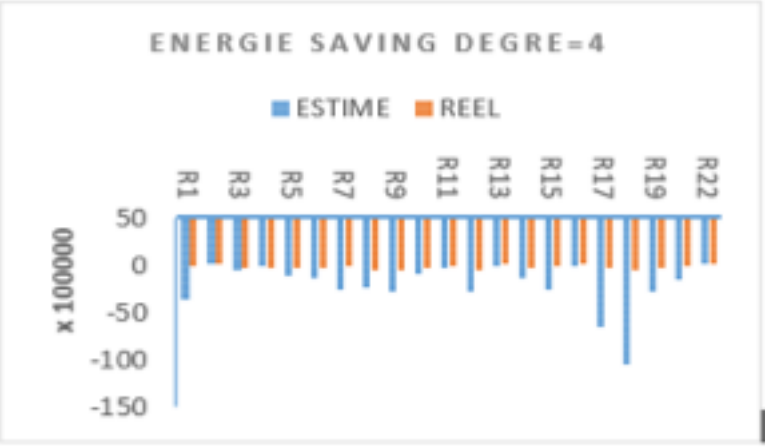
Degre de parallelisme = 2					
Requêtes	Error (%)		Requêtes	Error (%)	
	5 Go	10 Go		5 Go	10 Go
R1	0,416	0,030	R12	0,052	0,012
R2	0,030	0,377	R13	0,036	0,098
R3	0,335	0,004	R14	0,516	0,012
R4	0,130	0,117	R15	0,670	0,052
R5	0,212	0,033	R16	0,398	0,273
R6	0,393	0,018	R17	0,437	0,009
R7	0,401	0,047	R18	0,454	0,038
R8	0,138	0,017	R19	0,009	0,001
R9	0,033	0,029	R20	0,026	0,016
R10	0,600	0,016	R21	_	_
R11	0,093	0,254	R22	0,010	0,036

TPC-H Marge d'erreur de fcn

Résultats (II)

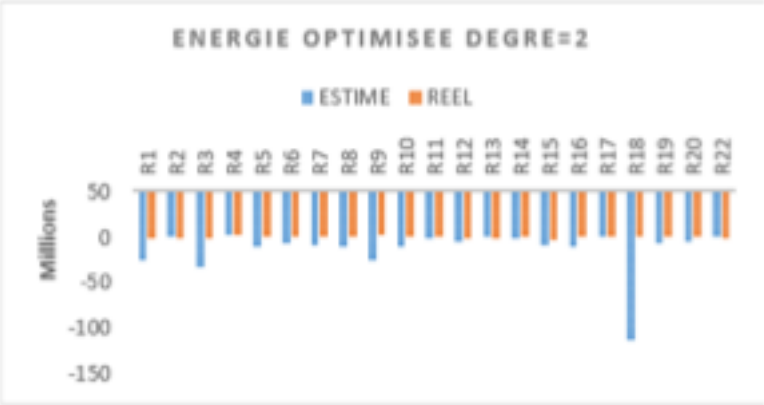
TPC-H 1 Go

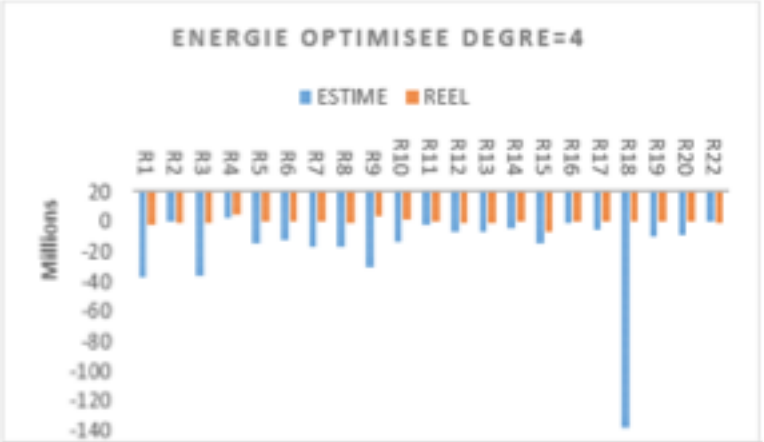




Energie Optimisée

TPC-H 10 Go





Suites

- ❑ Revoir les paramètres de notre modèle (machine Learning).
- ❑ Traiter le parallélisme inter-machines pour le traitement des requêtes.
- ❑ Traiter l'efficacité énergétique sur les techniques de partitionnement.

Questions | Remarques

Merci !!!