

**Master d'Informatique**  
**spécialité DAC**  
BDLE (Bases de Données Large Echelle)  
-Seconde Partie-

**Cours 3 : Requêtes relationnelles en**  
**M/R (2/2)**

Mohamed-Amine Baazizi – email: prénom.nom@lip6.fr  
<http://dac.lip6.fr/master/ues-2014-2015/bdle-2014-2015/>

## Objectifs

### **Cours**

Traduction des requêtes relationnelles SQL

### **TME**

1. Jointures parallèles sur le cluster
2. Traduction des requêtes SQL

## Bilan cours précédent

### Traduction d'opérateurs algébriques

- Algorithmes facilement généralisables
- Jointures n-aires réalisables en une seule passe, coût polynomial.

Exemple :

$$Q = R(A,B) \bowtie S(B,C) \bowtie T(C,D)$$

$$\text{coût}(Q) = |S| + |D_C| \times |R| + |D_B| \times |T|$$

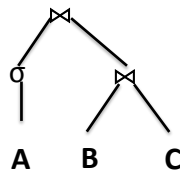
où  $D_B$  et  $D_C$  domaines fonctions hachage sur B et C

3

## SQL (version simplifiée)

### Opérateurs algébriques + fonctions d'agrégats

- Sélection, projection, jointure
- COUNT, SUM, AVG, MIN, MAX



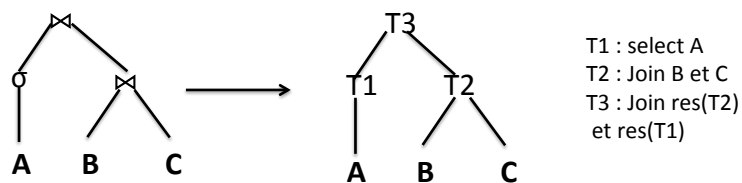
Exemple d'un arbre algébrique

4

## Traduction des requêtes SQL

### Naïve (one-to-one)

- Chaque nœud = une tâche MR
- Une requête = enchaînement de tâches MR



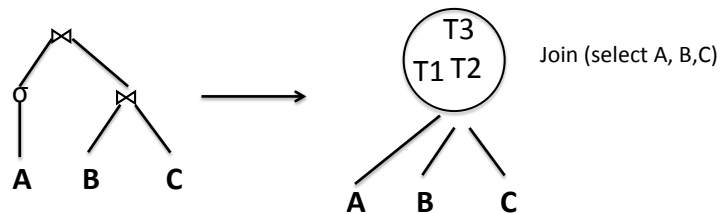
matérialisation + chargement résultats intermédiaires  
→ Coût élevé

5

## Traduction des requêtes SQL

### Optimisée

- Chaque groupe de nœuds = une tâche MR
- Une requête = enchaînement de tâches MR



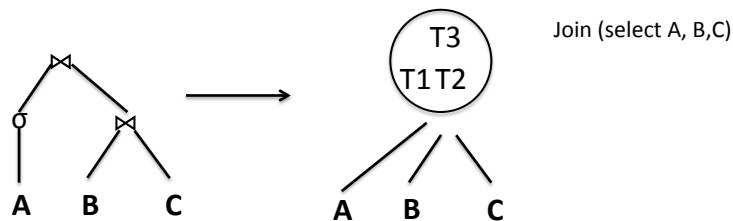
Pas de matérialisation/chargement inutile  
→ Coût réduit

6

## Traduction optimisée

**Principe** : combiner les opérateurs bottom-up

- Jointures seules : cours précédent
- Jointures avec opérateurs : en général toujours possible excepté agrégation ou tri



7

## Traduction optimisée : illustration

```
SELECT sum(l_extendedprice) / 7.0 AS
avg_yearly
```

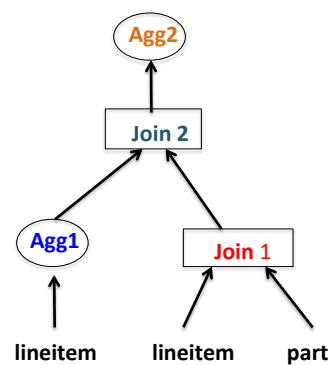
```
FROM (SELECT l_partkey, 0.2*
avg(l_quantity) AS t1
```

```
FROM lineitem GROUP BY
l_partkey) AS inner,
```

```
(SELECT
l_partkey,l_quantity,l_extendedprice
FROM lineitem, part
WHERE p_partkey = l_partkey) AS
outer
```

```
WHERE outer.l_partkey =
inner.l_partkey; AND outer.l_quantity <
inner.t1;
```

Q17 de TPC-H



Workflow MR

Adapté de [Lee]

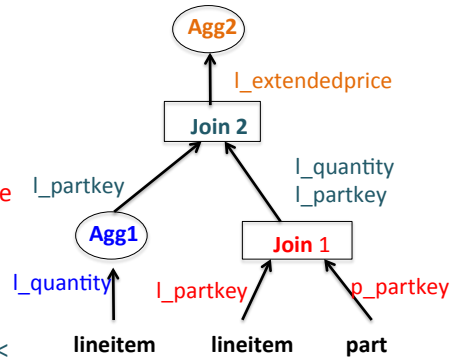
8

## Traduction optimisée : illustration

```

SELECT sum(l_extendedprice) / 7.0 AS
avg_yearly
FROM (SELECT l_partkey, 0.2*
avg(l_quantity) AS t1
      FROM lineitem GROUP BY
           l_partkey) AS inner,
(SELECT
l_partkey,l_quantity,l_extendedprice
FROM lineitem, part
WHERE p_partkey = l_partkey) AS
outer
WHERE outer.l_partkey =
inner.l_partkey; AND outer.l_quantity <
inner.t1;
    
```

Q17 de TPC-H



Workflow MR

Adapté de [Lee]

9

## Traduction optimisée : illustration

```

Agg1
Map : lineitem → (l_partkey, l_quantity)
Reduce : 0.2 * avg = v
    
```

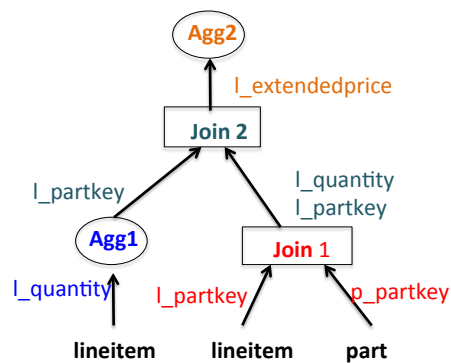
```

Join1
Map : lineitem → (l_partkey, (l_quantity,
l_extendedprice))
      part → (l_partkey, null)
Reduce : jointure
    
```

```

Join2
Map : inner → (l_partkey, v)
      outer → (l_partkey, (l_quantity,
l_extendedprice))
Reduce : jointure
    
```

**Agg2 ...**  
Programmes MR



Workflow MR

Adapté de [Lee]

10

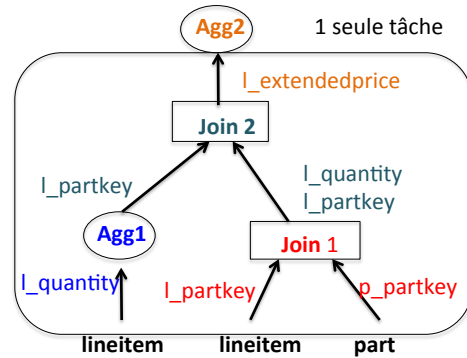
## Traduction optimisée : illustration

### Join

Map :  
 lineitem → (p\_partkey, (l\_quantity, l\_extendedprice))  
 part → (l\_partkey, null)

Reduce :  
 agg1 sur l\_quantity  
 join1 sur l\_partkey = p\_partkey  
 join2 sur l\_partkey = l\_partkey

### Programme MR



Coût communication --

Workflow MR

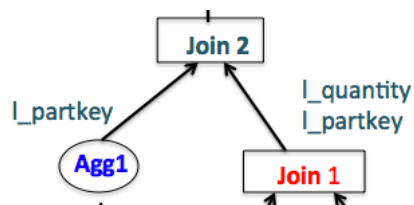
Adapté de [Lee]

11

## Traduction optimisée : intuition

Résultat d'une tâche = entrée de la suivante

Même(s) attribut(s) de partitionnement → *transit correlation (TC)*

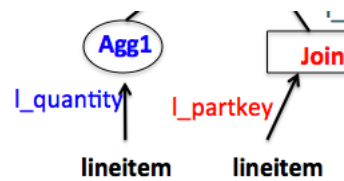


12

## Traduction optimisée : intuition

### Auto-jointures

Mêmes données initiales → *Input correlation (IC)*

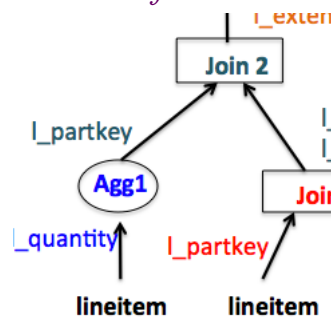


13

## Traduction optimisée : intuition

Résultat d'une tâche = entrée de la suivante

Mêmes données initiales + même(s) attribut(s) de partitionnement → *Jobflow correlation (JFC)*



14

## Traduction optimisée : gain

<i>Input correlation</i>	Partager les <u>Map</u>	<ul style="list-style-type: none"> <li>Gain local (accès disque)</li> <li>communication <u>si map distant</u></li> </ul>
<i>Transit correlation</i>	Partager Map, mutualiser reduces	<ul style="list-style-type: none"> <li>Gain local (accès disque)</li> <li>communication</li> </ul>
<i>Jobflow correlation</i>	Mutualiser reduce	<ul style="list-style-type: none"> <li>Gain local (accès disque)</li> <li>communication</li> </ul>

15

## Traduction optimisée : gain

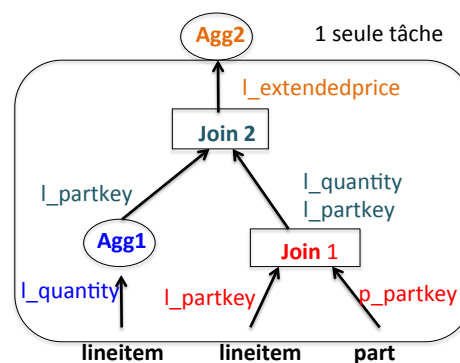
### Join

Map :  
 lineitem → (p\_partkey, (l\_quantity, l\_extendedprice))  
 part → (l\_partkey, null)

Reduce :  
 agg1 sur l\_quantity  
 join1 sur l\_partkey = p\_partkey  
 join2 sur l\_partkey = l\_partkey

### Programme MR

**Comparer coût communication des approches naïve et optim**



### Workflow MR

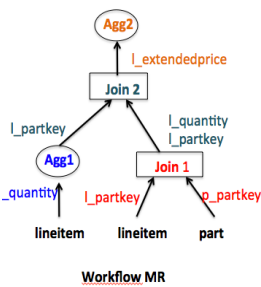
Adapté de [Lee]

16



## Traduction optimisée : aperçu

SQL → MR<sup>std</sup> → MR<sup>optim</sup>



Workflow MR

**Agg1**  
 Map : lineitem → (l\_partkey, l\_quantity)  
 Reduce : 0.2 \* avg = v

**Join1**  
 Map : lineitem → (l\_partkey, (l\_quantity, l\_extendedprice))  
 part → (l\_partkey, null)  
 Reduce : jointure

**Join2**  
 Map : inner → (l\_partkey, v)  
 outer → (l\_partkey, (l\_quantity, l\_extendedprice))  
 Reduce : jointure

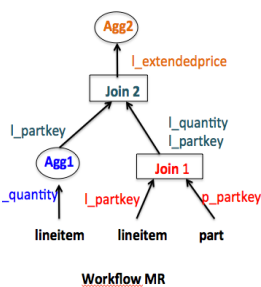
**Agg2 ...**

Map :  
 lineitem → (p\_partkey, (l\_quantity, l\_extendedprice))  
 part → (l\_partkey, null)  
 Reduce :  
 agg1 sur l\_quantity  
 join1 sur l\_partkey = p\_partkey  
 join2 sur l\_partkey = l\_partkey

17

## Traduction optimisée : aperçu

SQL → MR<sup>std</sup> → MR<sup>optim</sup>



Workflow MR

Proj-sel-Job \*  
 Agg-Job  
 Join-job\*\*  
 Sort-job

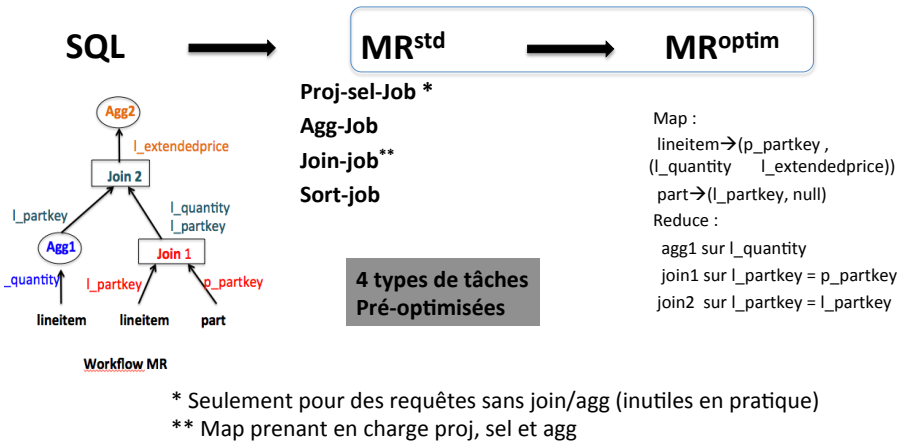
4 types de tâches  
 Pré-optimisées

Map :  
 lineitem → (p\_partkey, (l\_quantity, l\_extendedprice))  
 part → (l\_partkey, null)  
 Reduce :  
 agg1 sur l\_quantity  
 join1 sur l\_partkey = p\_partkey  
 join2 sur l\_partkey = l\_partkey

\* Seulement pour des requêtes sans join/agg (inutiles en pratique)  
 \*\* Map prenant en charge proj, sel et agg

18

## Traduction optimisée : aperçu



19

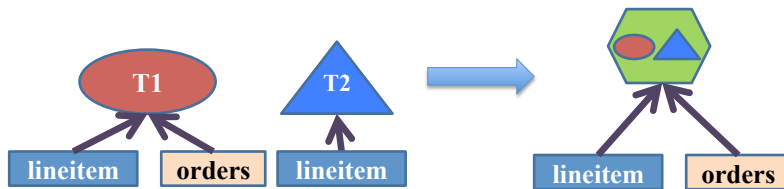
## Traduction optimisée : étapes

- A. Aplatir l'arbre MR<sup>std</sup> : parcours post-order  
 → séquence de tâches (PS-A-J-S)
- B. Réduire la séquence de tâches
  1. Fusionner des opérateurs
    - a. (SP, Agg, Sort, Join) x (SP, Agg, Sort)
    - b. Jointures successives
  2. Agrégation au vol

20

## Règles de transformation

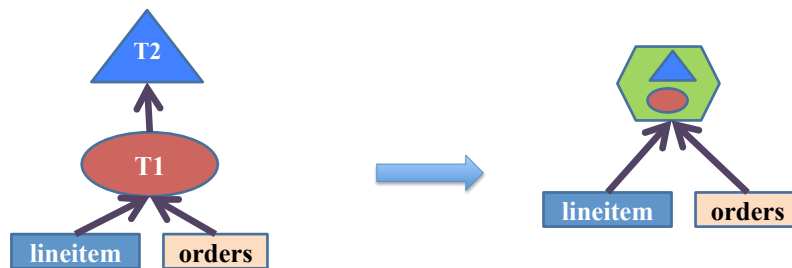
$$\text{Règle 1} \frac{\text{IC}(T1, T2) \quad \text{TC}(T1, T2) \quad T12(s) = T1(s) \circ T2(s) \quad s \in \{\text{Map}, \text{Reduce}\}}{T12}$$



21

## Règles de transformation

$$\text{Règle 2} \frac{\text{JFC}(T1, T2) \quad \text{parent}(T2, T1) \quad T1 : \text{agg} \quad T12(\text{Reduce}) = \text{agg}}{T12}$$



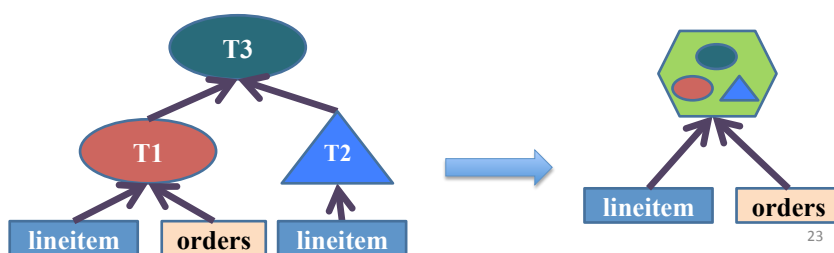
22

## Règles de transformation

JFC(T3,T1) JFC(T3,T2) TC(T1,T2)  $T1 < T2 < T3$   
 $T123(s) = T1(s) \circ T2(s) \circ T3(s)$   $s \in \{\text{Map, Reduce}\}$

Règle 3

T123



23

## Déclenchement des règles

**Phase 1** : appliquer règle 1 (fusion) jusqu'à plus de tâches IC et TC

**Phase 2** : appliquer les règles 2 à 3 (pour JFC)

Règle 2 : Agrégation

Règle 3 : JFC avec les deux précédentes tâches

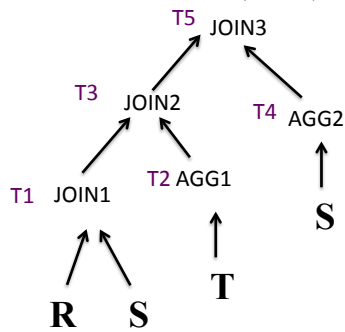
24

## Exemples

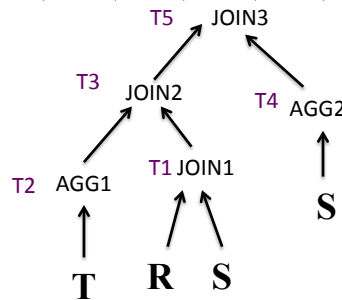
R(A,B) S(B,C) et T(C,D) trois schémas de relations

$Q = R \bowtie S \bowtie \text{agg}(T) \bowtie \text{agg}(S)$

Les corrélations : TC(T1,T4), JFC(T3,T1), JFC(T5,T3), JFC(T5,T4)



Résultat : T14, T2, T35  
(3 tâches)



Résultat : T2, T1435  
(2 tâches)

25

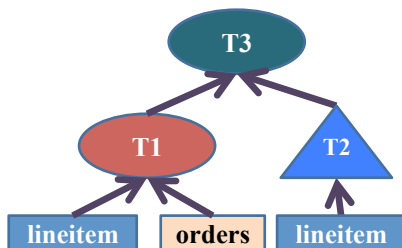
## Règles de transformation

JFC(T3,T1)  $T1 < T2 < T3$

$T13(\text{Reduce}) = T1(\text{reduce}) \circ T3(\text{reduce})$

Règle 4

T2, T13



Attention : aucune corrélation entre T3 et T2!

1. T2 s'exécute
2. T3 exécutée dans le reduce de T1

26

## Déclenchement des règles

**Phase 1** : appliquer règle 1 (fusion) jusqu'à plus de tâches IC et TC

**Phase 2** : appliquer les règles 2 à 3 (pour JFC)

Règle 2 : Agrégation

Règle 3 : JFC avec les deux précédentes tâches

Règle 4 : JFC avec une seule tâche précédente

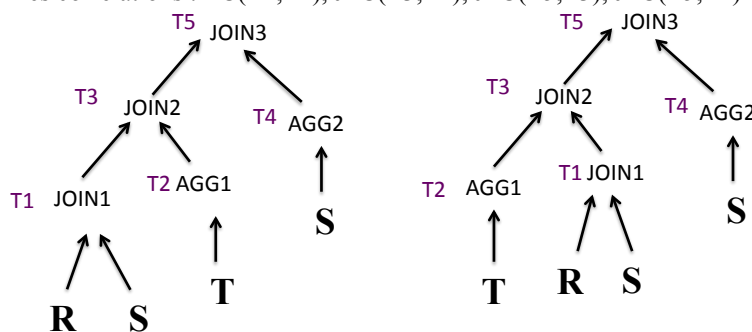
27

## Exemples

R(A,B) S(B,C) et T(C,D) trois schémas de relations

$Q = R \bowtie S \bowtie \text{agg}(T) \bowtie \text{agg}(S)$

Les corrélations : TC(T1,T4), JFC(T3,T1), JFC(T5,T3), JFC(T5,T4)



**Même résultat** : T2, T1435  
(2 tâches)

28

## Références

**[Lee et al.]** *YSmart: Yet Another SQL-to-MapReduce Translator*, in ICDCS'2011

**[YSmart]** <https://github.com/YSmart/YSmart>