

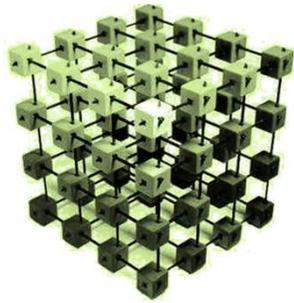
# Circuits électroniques au service de l'optimisation multi-requêtes

**Ahcène BOUKORCA**

**Sous la direction de Ladjel BELLATRECHE et Zoé Faget**

# Contexte: Optimiseur de requêtes

---



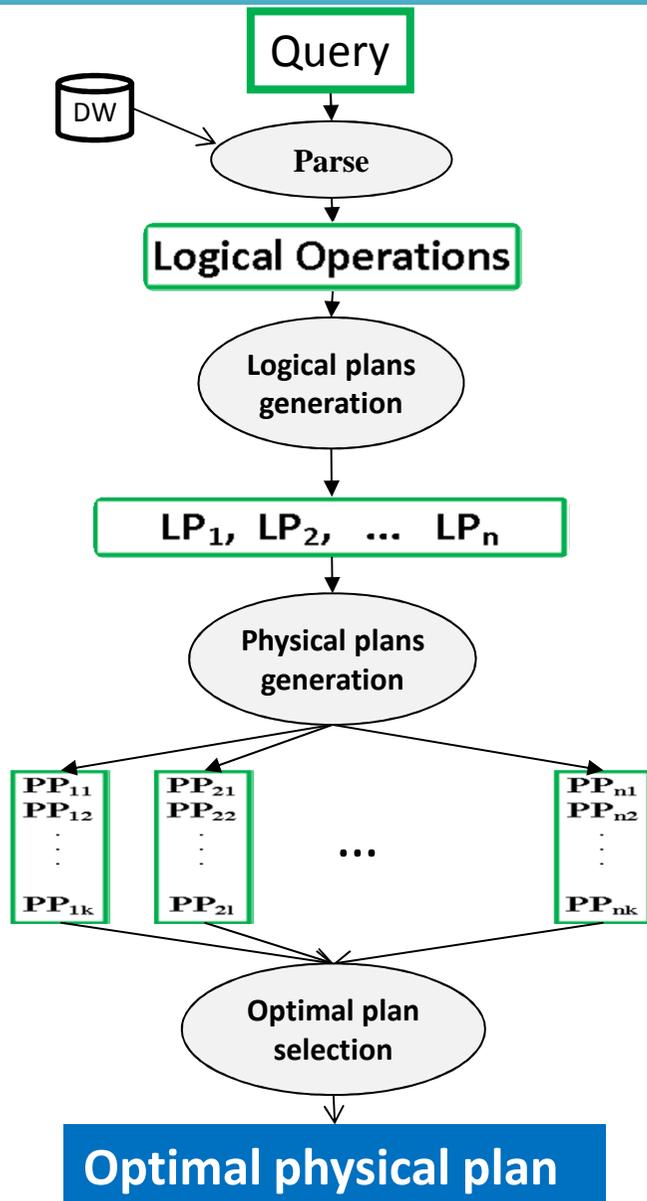
**Entrepôt de Données  
Relationnels**

- ❑ **Big Data à traiter, analyser et sauvegarder.**
- ❑ **Requêtes plus complexes qui nécessitent un temps de réponse immédiat.**
- ❑ **Un nombre de requêtes énorme à répondre.**

Nécessité d'avoir un traitement efficace des requêtes.

**Améliorer l'optimisation des requêtes**

# Contexte: Optimiseur de requêtes



## Formalisation de l'optimiseur de requêtes

### Entrées :

- Une requête Q.
- Entrepôt de données.

### Sorties :

- Un plan d'exécution pour Q.

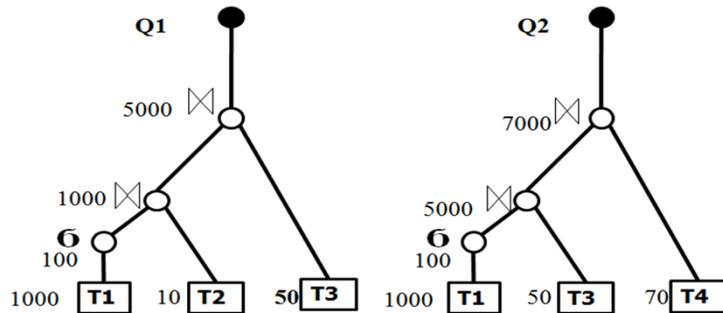
### Objectif :

- Exécution de Q en un temps minimum.

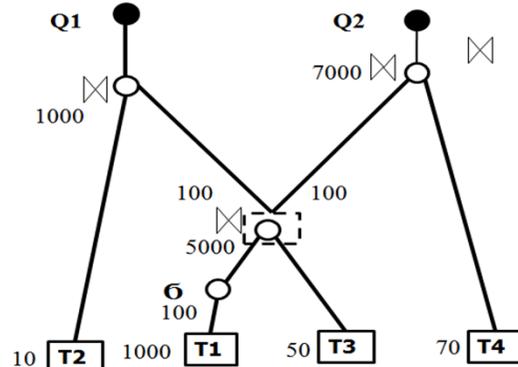
**Exemples** : *Starbust* (Pirahesh et al., 1992), utilisé par l'optimiseur de *IBM DB2* et *Volcano* (Graefe et al., 1993) utilisé par l'optimiseur de *Microsoft SQL-Server*.

**Complexité exponentielle**: un espace de recherche énorme. Exemple: la requête 8 du benchmark *SSB* (4 jointures), a plus de 22 millions possibilités de plan logique).

# Contexte: Optimisation multi-requêtes



(a) Individually query optimization  
Total cost = 20380



(b) Multiple Query optimization  
Total cost = 14430

## Formalisation de l'optimisation multi-requêtes

### Entrées :

- Une charge de  $m$  requêtes  $Q=\{Q1,Q2,...Qm\}$
- Entrepôt de données

### Sorties :

- Choisir un plan pour chaque requête en maximisant la réutilisation des résultats intermédiaires
- Choisir les résultats intermédiaires.

### Objectif :

- Réduire le temps d'exécution de  $Q$

*Sellis (1988)*, a proposé un algorithme  $A^*$  qui permet d'essayer de trouver toutes combinaisons possibles de fusion des plans locaux des requêtes.

**Complexité :** *NP-COMPLET (Sellis et al. 1990).*

Un autre problème est la gestion des résultats intermédiaires dans les fichiers temporaires

# Contexte: Optimisation multi-requêtes (2)

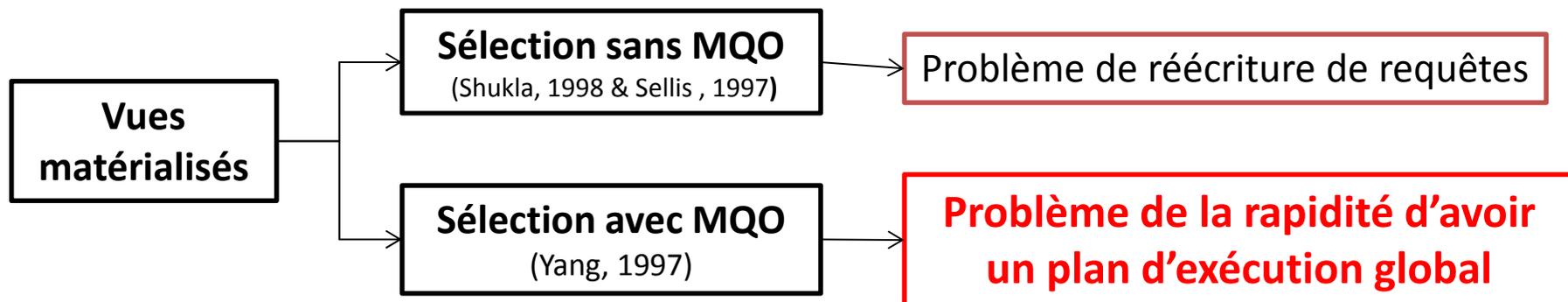
## Combinaison entre deux problèmes:

- Optimisation des requêtes .
- Sélection des vues matérialisées.

Yang (1997) a proposé un Framework pour générer un ensemble de vues matérialisées et un plan d'exécution global (MVPP), exploitant ces vues.

**Complexité:** NP-Complet,

Les méthodes proposées sont basées sur la fusion des plans individuels.



## Formalisation de l'optimisation multi-requêtes (2)

### Entrées :

- Une charge de m requêtes  $Q=\{Q1,Q2,...Qm\}$
- Entrepôt de données

### Sorties :

- Choisir un plan pour chaque requête en maximisant la réutilisation résultats intermédiaires
- **Structure d'optimisation (vues matérialisées).**

### Objectif :

- Réduire le temps d'exécution de Q

# Plan

---

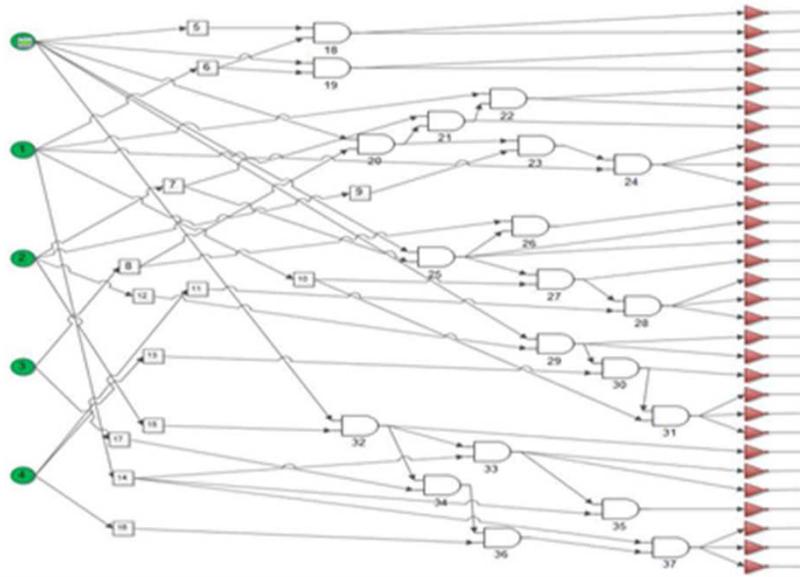
1. Analogie entre MVPP et circuit électroniques
2. Notre approche de génération de MVPP
3. Algorithme de sélection des vues matérialisées
4. Etude expérimentale
5. Travaux actuels et futurs

# Analogie entre MVPP et circuit électronique

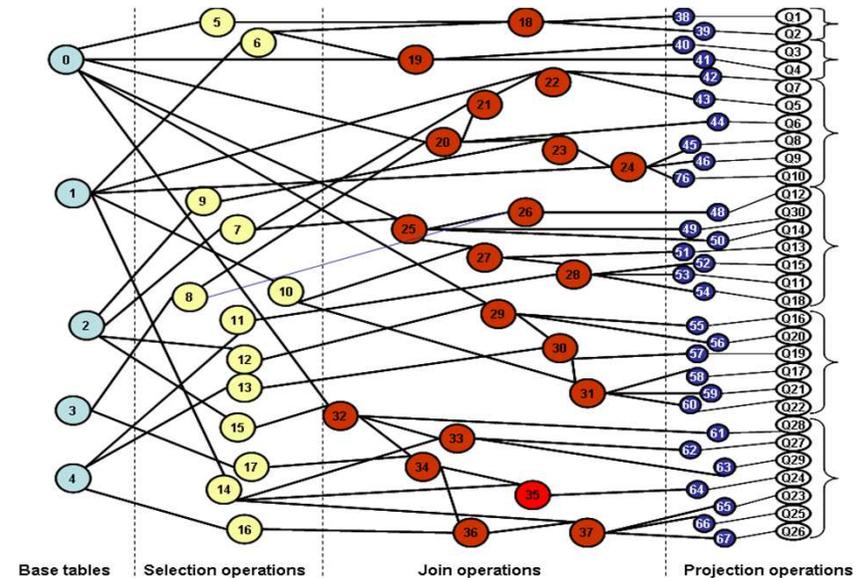
## Analogie entre MVPP & VLSIC

Notre Approche  
Algorithme de sélection des VM  
Etude expérimentale  
Conclusion

Circuit électronique



MVPP



### Objectif (VLSI):

Fractionner le circuit en plusieurs clusters de portes :

- ❑ Forte connexion entre les portes à l'intérieur de cluster,
- ❑ Minimum de connexion entre les clusters.

### Objectif (MQO):

Regrouper les requêtes en plusieurs composante:

- ❑ Forte interaction entre les requêtes à l'intérieur de composante.
- ❑ Minimum de partage entre les composante.

# Analogie entre MVPP et circuit électronique

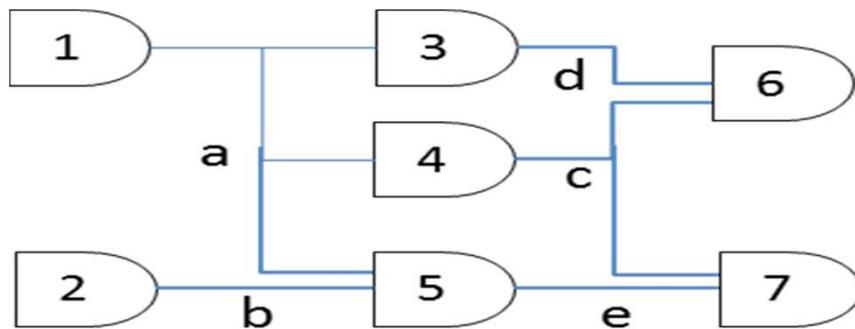
## Analogie entre MVPP & VLSIC

Notre Approche  
Algorithme de sélection des VM  
Etude expérimentale  
Conclusion

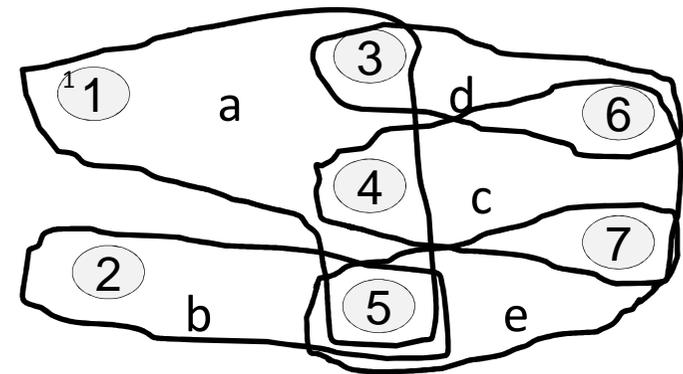
### Motivation de VLSI circuit:

→Manipule des millions de ports.

→Grande panoplie d'outils ayant démontré leur efficacité.



(A) Circuit.

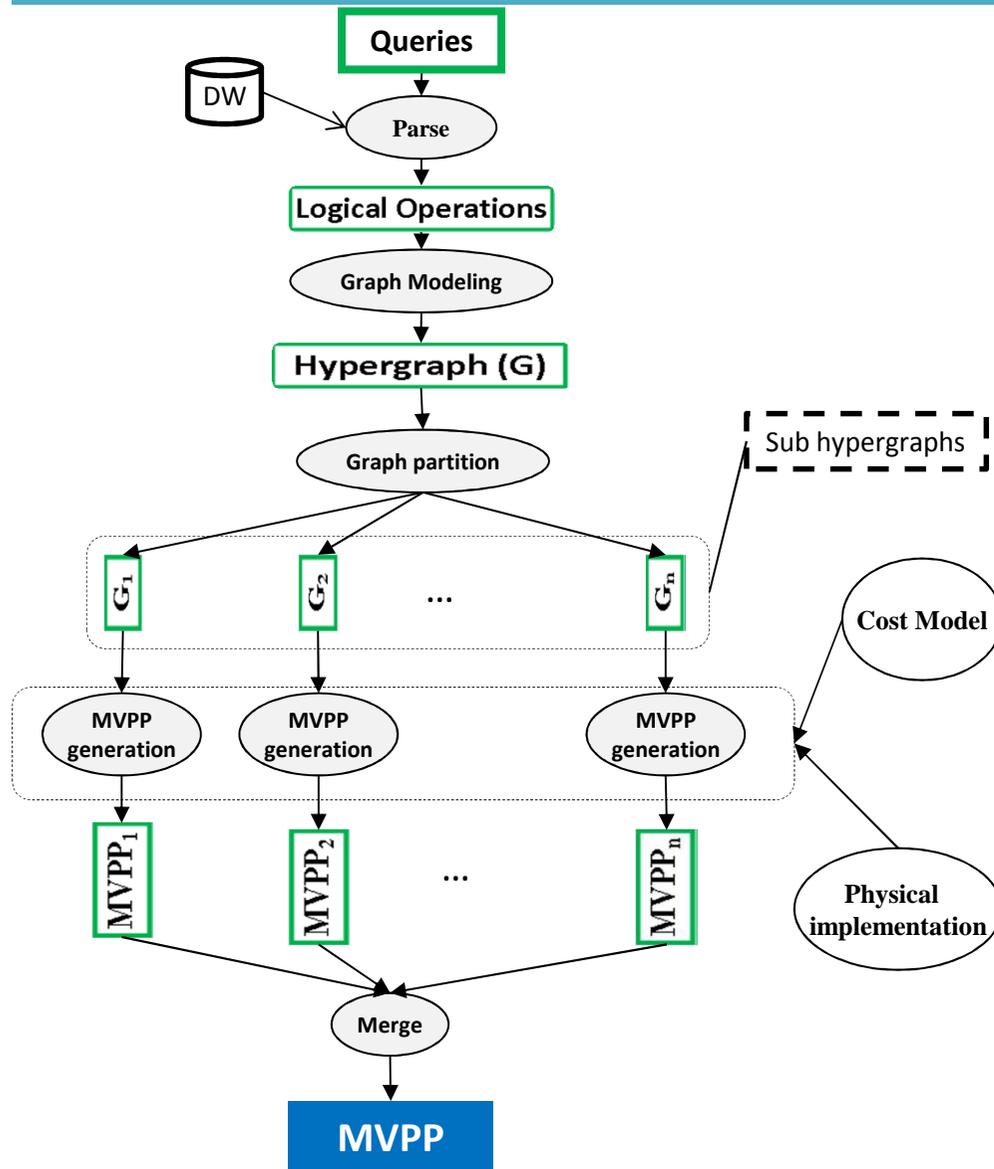


(B) Hypergraphe.

**Comment emprunter ces méthodes pour résoudre le problème d'optimisation multi-requêtes**

# Notre approche

Analogie entre MVPP & VLSIC  
**Notre Approche**  
Algorithme de sélection des VM  
Etude expérimentale  
Conclusion

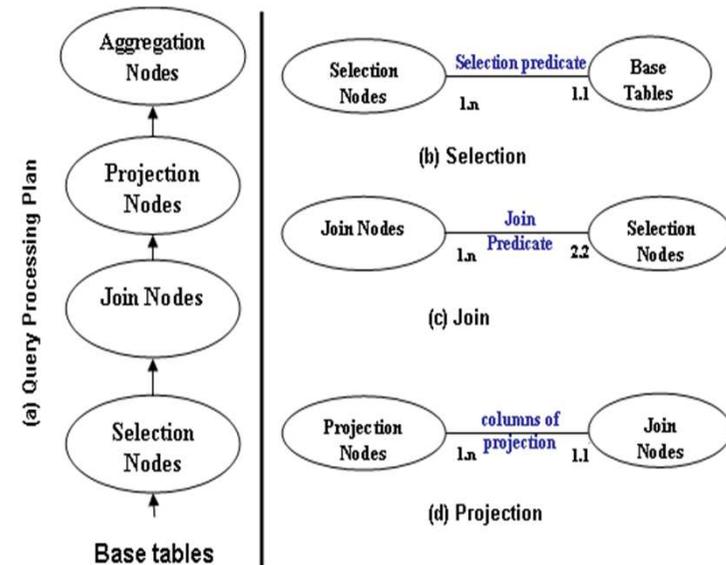
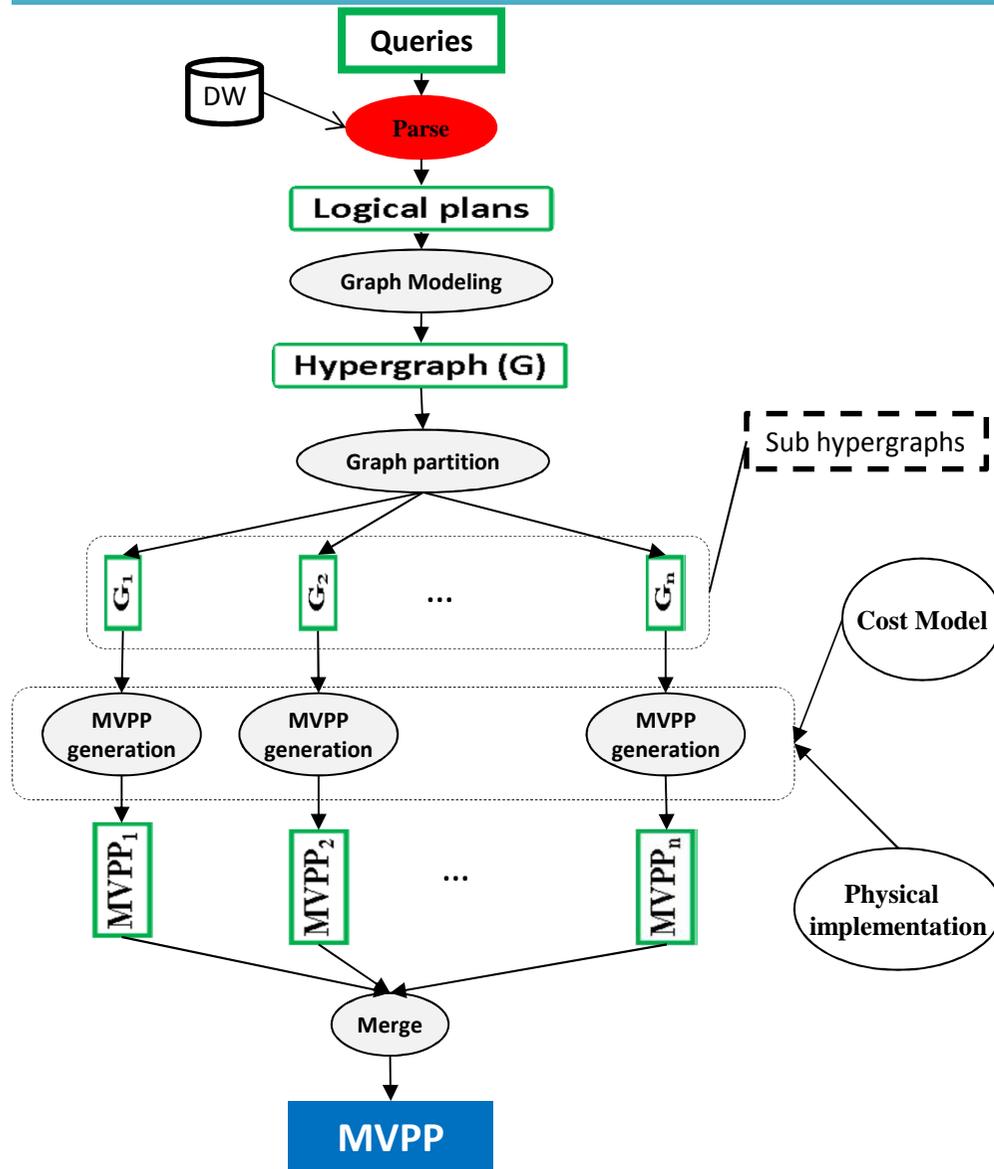


## Notre approche:

- ❑ Une vision de construction et pas de sélection un MVPP.
- ❑ Construire du MVPP sans passer les plans locaux.
- ❑ Division du problème en plusieurs tâches indépendantes.

# Notre approche

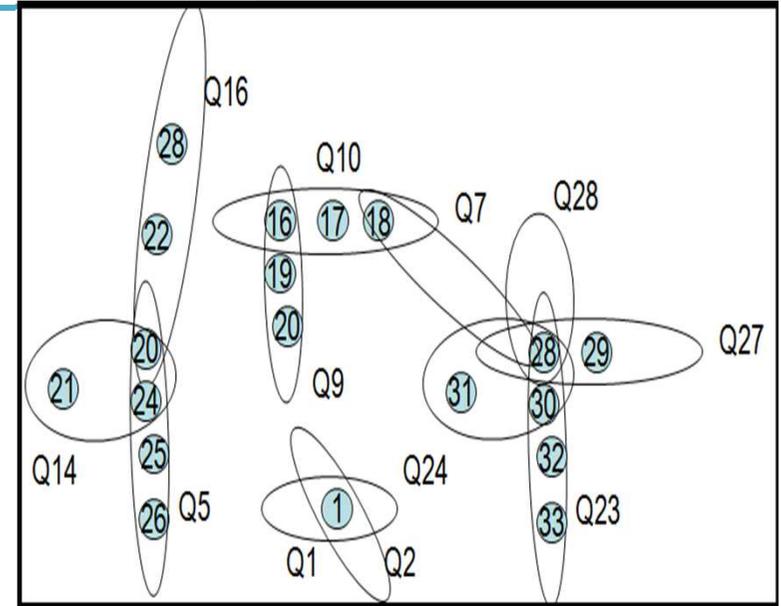
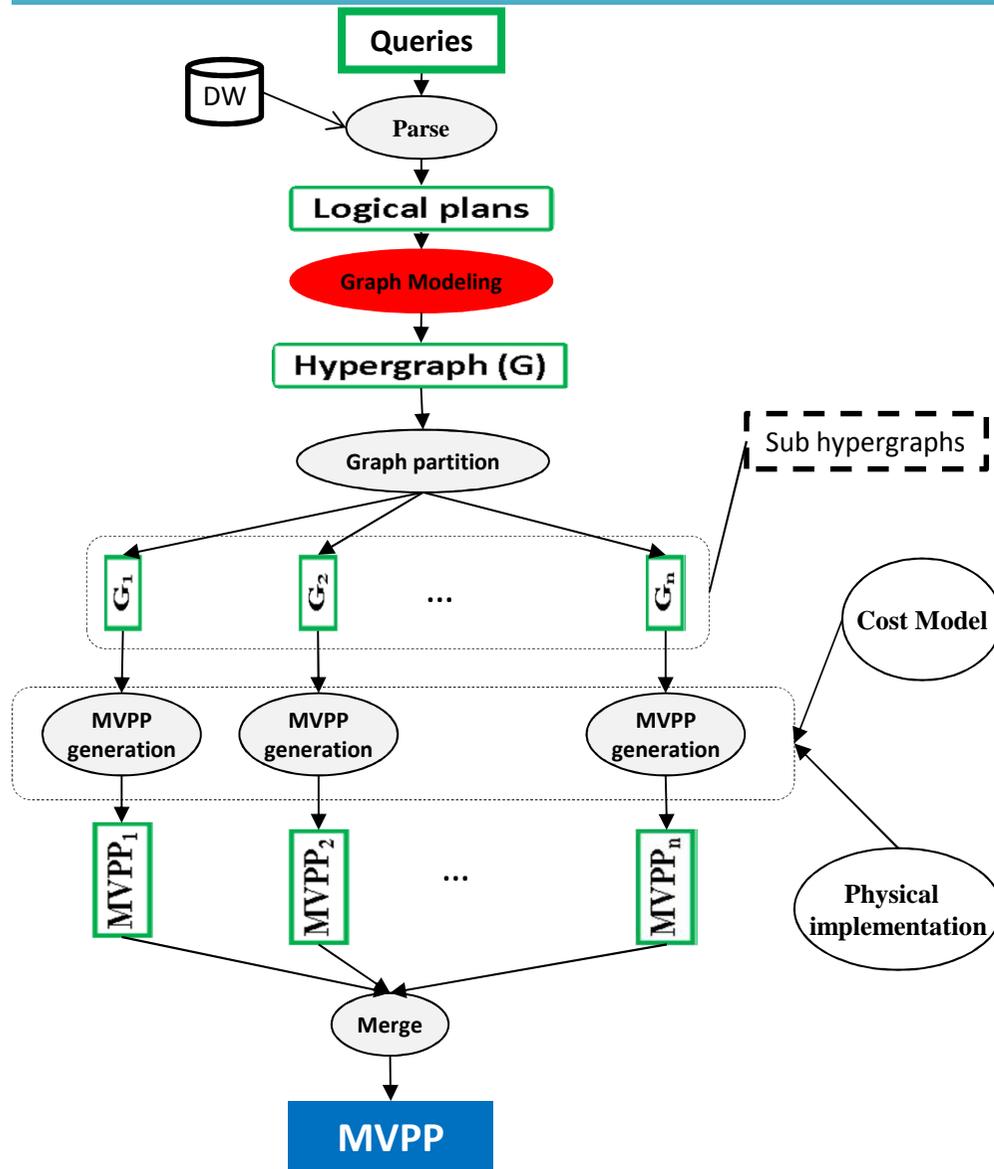
Analogie entre MVPP & VLSIC  
**Notre Approche**  
 Algorithme de sélection des VM  
 Etude expérimentale  
 Conclusion



Analyser la requête SQL et déterminer les différentes opérations (sélection, jointure, projection, et).

# Notre approche

Analogie entre MVPP & VLSIC  
**Notre Approche**  
Algorithme de sélection des VM  
Etude expérimentale  
Conclusion



Modélisation des nœuds de jointures par un hypergraphe,  $G(V,E)$ :

- V: représente les nœuds de jointure;
- E: représente les requêtes, chaque hyperarête connecte les nœuds de jointure qui les utilisent.

# Notre approche

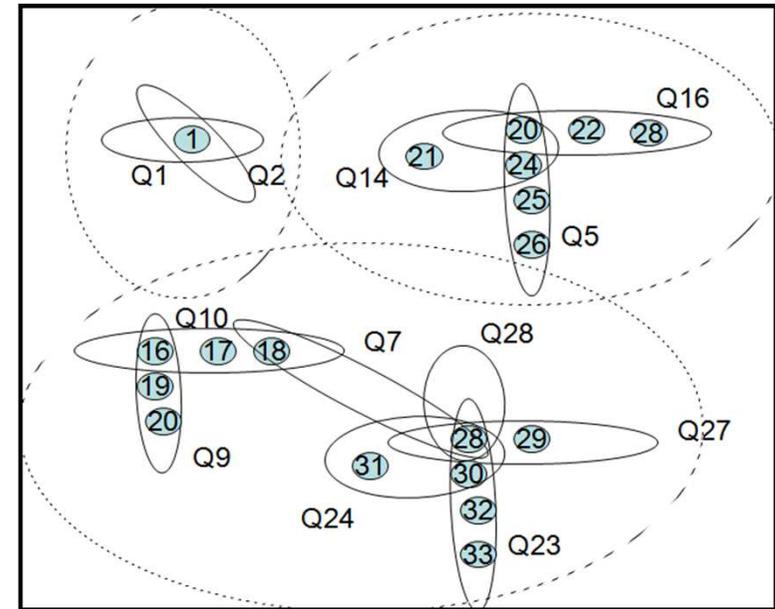
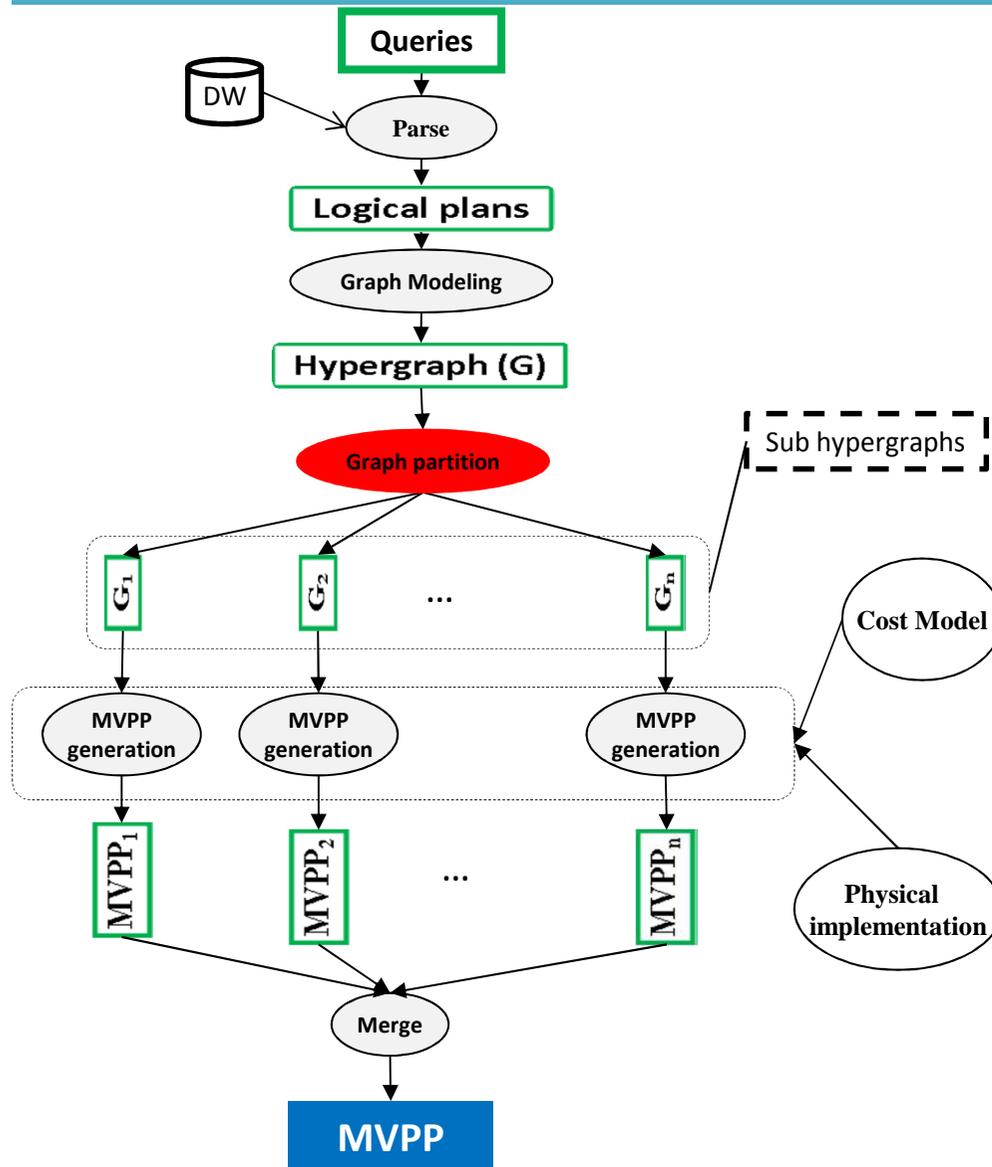
Analogie entre MVPP & VLSIC

## Notre Approche

Algorithme de sélection des VM

Etude expérimentale

Conclusion



Partitionnement en adaptant **HMETIS**

Entrée: Hypergraph ;

Sortie: Plusieurs hypergraphes disjoints, chaque hypergraphe contient les requêtes qui peuvent partager des résultats intermédiaires.

# Notre approche

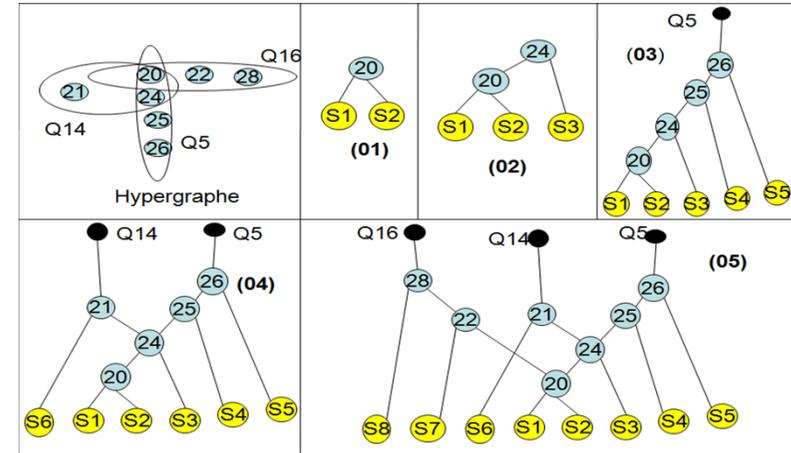
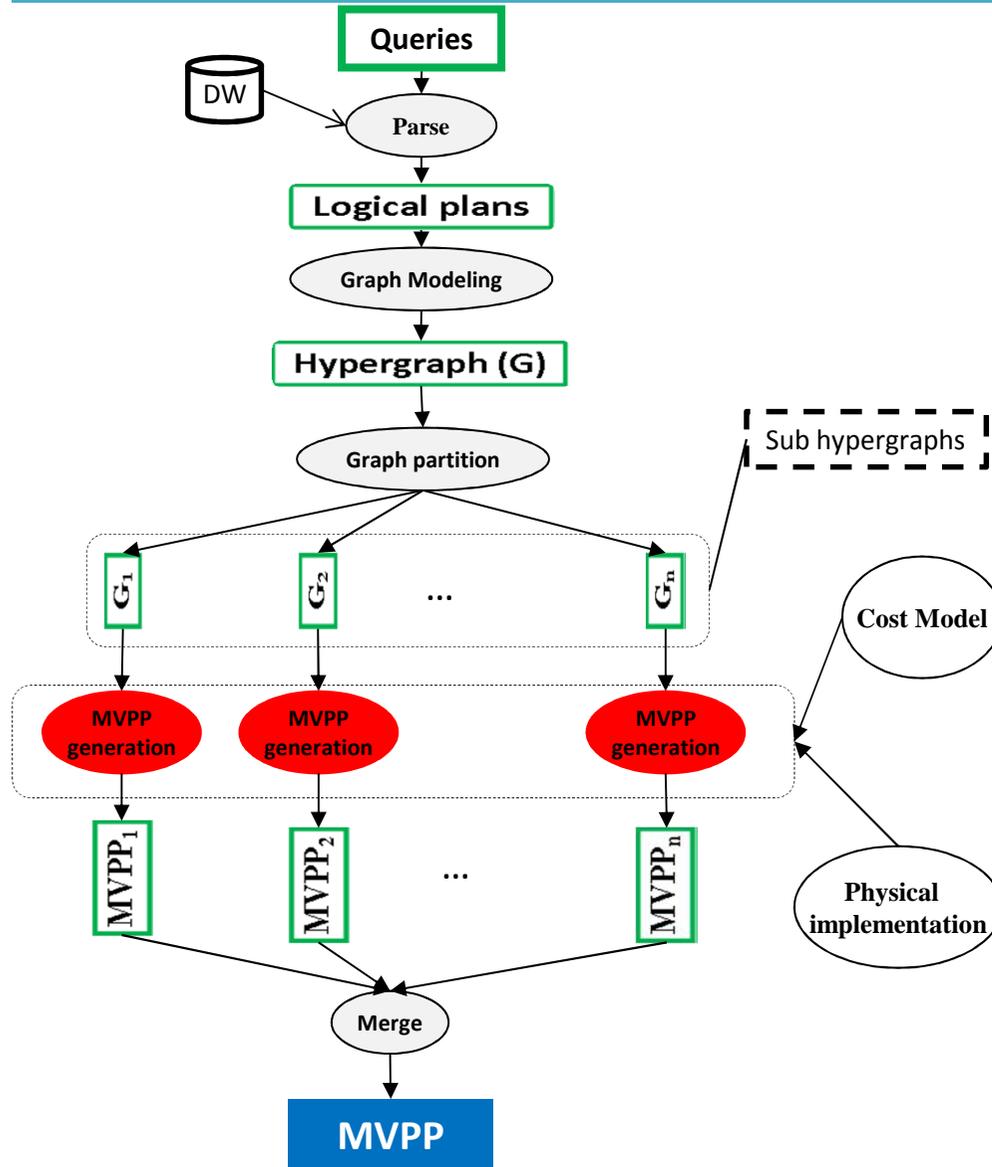
Analogie entre MVPP & VLSIC

## Notre Approche

Algorithme de sélection des VM

Etude expérimentale

Conclusion



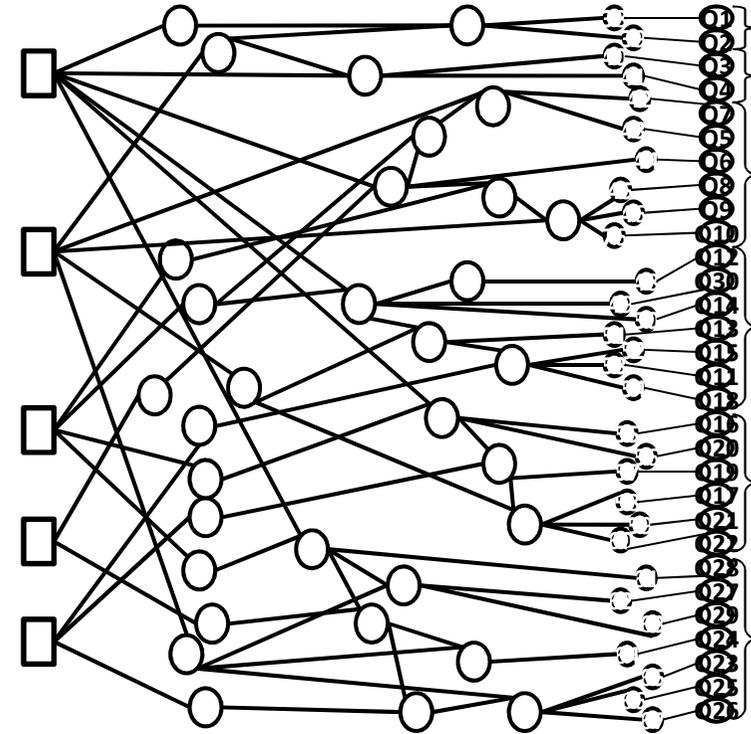
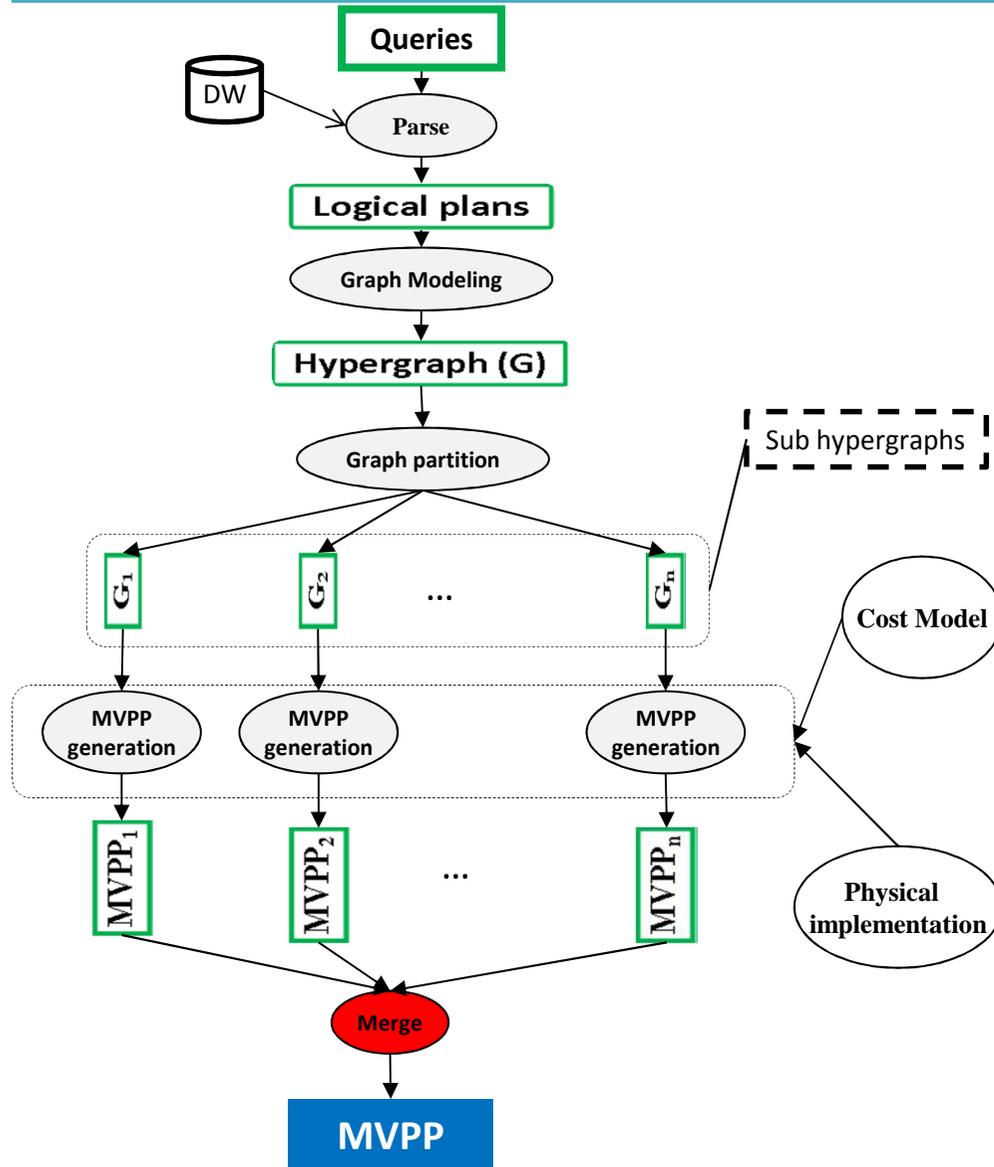
**Objectif :** Transformer chaque hypergraphe en graphe orienté qui représente un MVPP pour une composante.

**Entrée :** un hypergraphe.

**Sortie :** graphe orienté (MVPP).

# Notre approche

Analogie entre MVPP & VLSIC  
**Notre Approche**  
Algorithme de sélection des VM  
Etude expérimentale  
Conclusion



**Objectif :** fusion des différents MVPP des composantes en un seul MVPP global.

**Entrée:** plusieurs MVPP.

**Sortie:** Un MVPP global.

# Algorithme de sélection des vues matérialisées

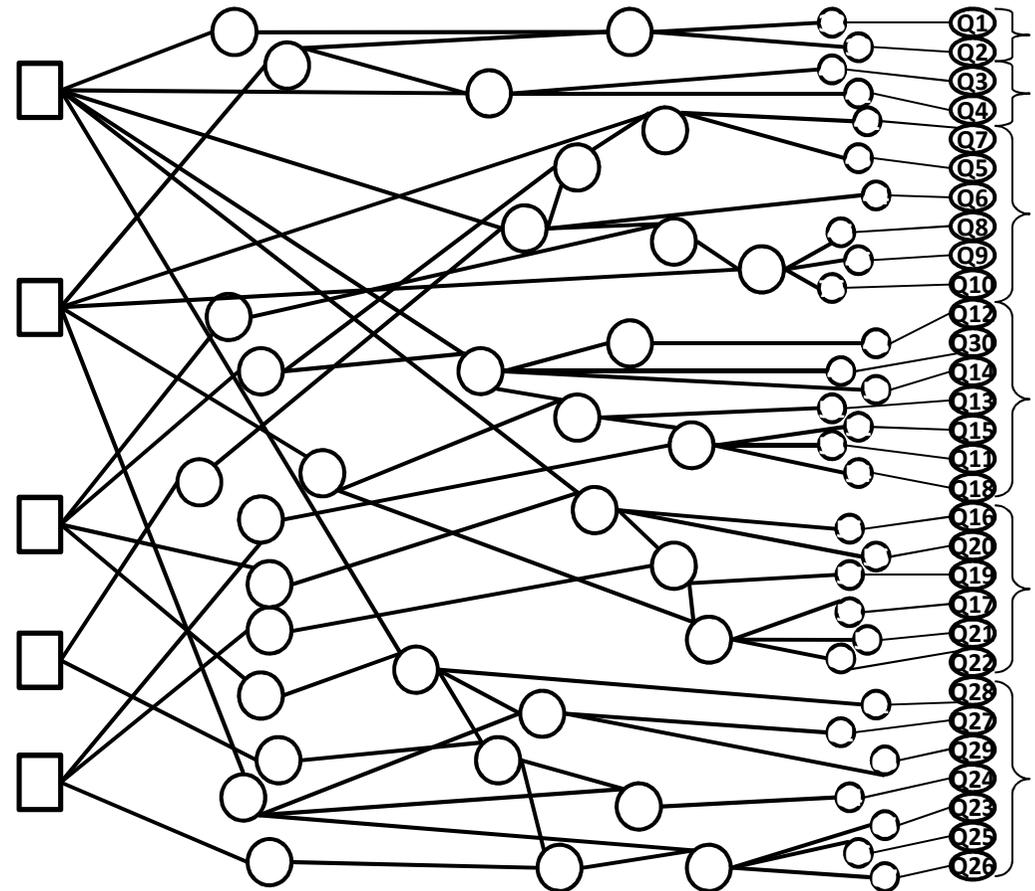
Analogie entre MVPP & VLSIC  
Notre Approche  
**Algorithme de sélection des VM**  
Etude expérimentale  
Conclusion

**Objectif:** Application du MVPP pour la sélection des vues matérialisées

**Entrée :** Un MVPP

**Output :** Un ensemble de vues matérialisées.

Un algorithme qui matérialise toutes les vues qui ont un bénéfice de réutilisation positif.

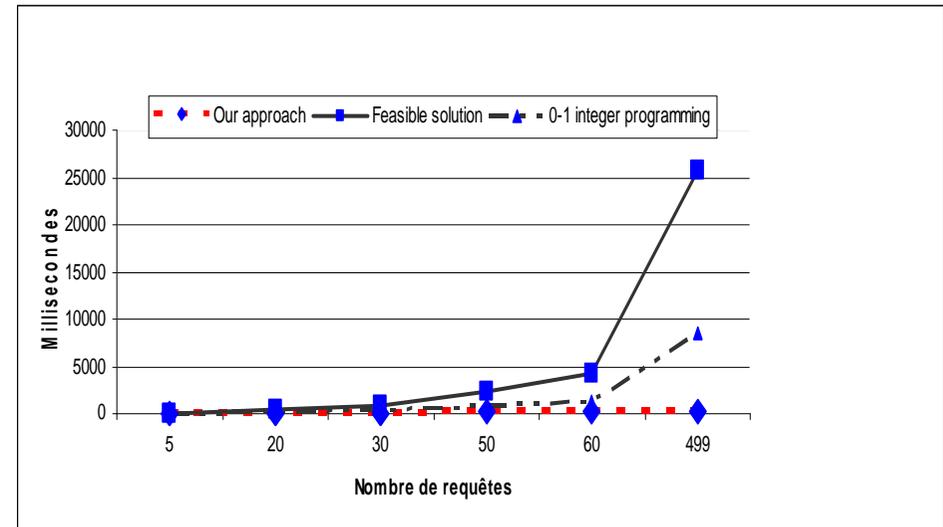


# Etude expérimentale

Analogie entre MVPP & VLSIC  
Notre Approche  
Algorithme de sélection des VM  
**Etude expérimentale**  
Conclusion

Star schema benchmark: SF=1 RAM : 8GB

Query number	Our approach	Feasible solution	0-1 integer programming
5	71	82	69
20	88	514	225
30	104	933	510
50	108	2258	781
60	114	4358	1224
499	283	25654	8604



**Approche de Yang** : le temps d' exécution est une fonction exponentiel par rapport au nombre de requêtes . (**Manque de scalabilité**)

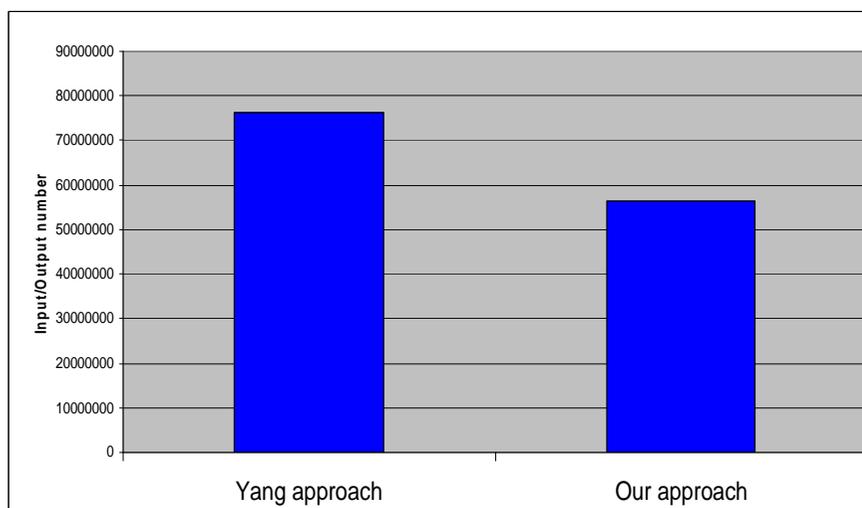
**Notre approche**: le temps d' exécution est une fonction logarithmique par rapport au nombre de requêtes . (**assure la scalabilité**)

# Etude expérimentale

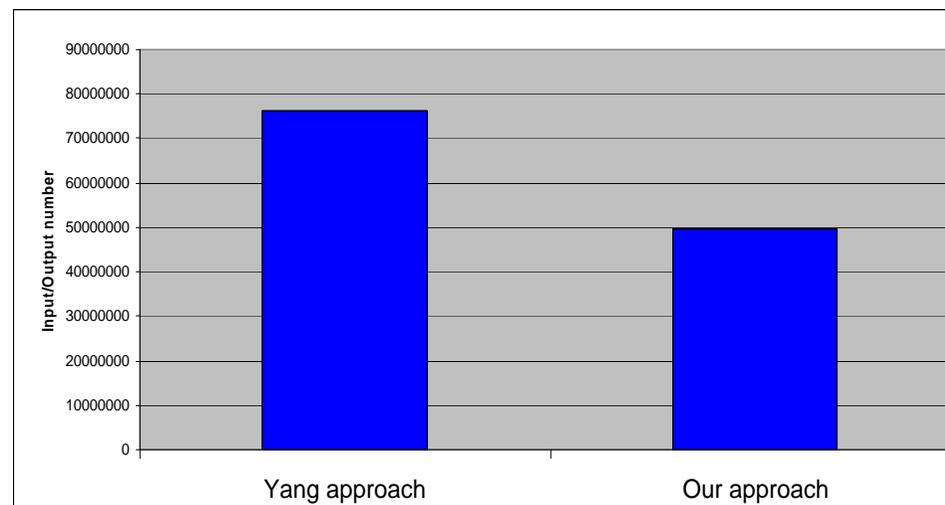
Analogie entre MVPP & VLSIC  
Notre Approche  
Algorithme de sélection des VM  
**Etude expérimentale**  
Conclusion

Star schema benchmark: SF=1 RAM : 8GB

30 Queries



(01) Matérialiser un seul nœud dans chaque composante



(02) Matérialiser tous les nœuds qui ont un benefice positives

# Travaux actuels et futurs

Analogie entre MVPP & VLSIC  
Notre Approche  
Algorithme de sélection des VM  
Etude expérimentale  
**Conclusion**

- ❑ Analyser la structure MVPP
  - Conditionné par les structures d'optimisation;
  - Evolutive ( $Q \rightarrow Q'$ ). Introduit la matérialisation dynamique des vues dans une vision optimisation multiple en continue .
  - Plusieurs tables des faits
- ❑ Preuve de complexité de notre approche ( $n^2$ )
- ❑ Appliquer cette démarche dans le contexte de Map-Reduce.

---

**Merci**

**Des questions... ?**