



HAL
open science

**Une approche pour optimiser les traitements des
requêtes dans un environnement de bases de données
réparties**
In-Sup Paik

► **To cite this version:**

In-Sup Paik. Une approche pour optimiser les traitements des requêtes dans un environnement de bases de données réparties. Modélisation et simulation. Institut National Polytechnique de Grenoble - INPG; Université Joseph-Fourier - Grenoble I, 1981. Français. NNT: . tel-00294245

HAL Id: tel-00294245

<https://theses.hal.science/tel-00294245>

Submitted on 9 Jul 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

l'Université Scientifique et Médicale de Grenoble

et à

l'Institut National Polytechnique de Grenoble

pour obtenir le grade de

DOCTEUR INGENIEUR

par

In-Sup PAIK



**UNE APPROCHE POUR OPTIMISER LES TRAITEMENTS
DES REQUETES DANS UN ENVIRONNEMENT
DE BASES DE DONNEES REPARTIES.**



Thèse soutenue le 12 février 1981 devant la commission d'examen

L. BOLLIET Président

C. DELOBEL
E. ANDRE
M. ADIBA } Examineurs

UNIVERSITÉ SCIENTIFIQUE ET MÉDICALE DE GRENOBLE

Monsieur Gabriel CAU : Président

Monsieur Joseph KLEIN : Vice-Président

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

PROFESSEURS TITULAIRES

MM.	AMBLARD Pierre	Clinique de dermatologie
	ARNAUD Paul	Chimie
	ARVIEU Robert	I.S.N.
	AUBERT Guy	Physique
	AYANT Yves	Physique approfondie
Mme	BARBIER Marie-Jeanne	Electrochimie
MM.	BARBIER Jean-Claude	Physique expérimentale
	BARBIER Reynold	Géologie appliquée
	BARJON Robert	Physique nucléaire
	BARNOUD Fernand	Biosynthèse de la cellulose
	BARRA Jean-René	Statistiques
	BARRIE Joseph	Clinique chirurgicale A
	BEAUDOING André	Clinique de pédiatrie et puériculture
	BELORIZKY Elie	Physique
	BARNARD Alain	Mathématiques pures
Mme	BERTRANDIAS Françoise	Mathématiques pures
MM.	BERTRANDIAS Jean-Paul	Mathématiques pures
	BEZES Henri	Clinique chirurgicale et traumatologie
	BLAMBERT Maurice	Mathématiques pures
	BOLLIET Louis	Informatique (I.U.T. B)
	BONNET Jean-Louis	Clinique ophtalmologie
	BONNET-EYMARD Joseph	Clinique hépato-gastro-entérologie
Mme	BONNIER Marie-Jeanne	Chimie générale
MM.	BOUCHERLE André	Chimie et toxicologie
	BOUCHEZ Robert	Physique nucléaire
	BOUSSARD Jean-Claude	Mathématiques appliquées
	BOUTET DE MONVEL Louis	Mathématiques pures
	BRAVARD Yves	Géographie
	CABANEL Guy	Clinique rhumatologique et hydrologique
	CALAS François	Anatomie
	CARLIER Georges	Biologie végétale
	CARRAZ Gilbert	Biologie animale et pharmacodynamie

MM.	CAU Gabriel	Médecine légale et toxicologie
	CAUQUIS Georges	Chimie organique
	CHABAUTY Claude	Mathématiques pures
	CHARACHON Robert	Clinique ot-rhino-laryngologique
	CHATEAU Robert	Clinique de neurologie
	CHIBON Pierre	Biologie animale
	COEUR André	Pharmacie chimique et chimie analytique
	COUDERC Pierre	Anatomie pathologique
	DEBELMAS Jacques	Géologie générale
	DEGRANGE Charles	Zoologie
	DELORMAS Pierre	Pneumophtisiologie
	DEPORTES Charles	Chimie minérale
	DESRE Pierre	Métallurgie
	DODU Jacques	Mécanique appliquée (I.U.T. I)
	DOLIQUE Jean-Michel	Physique des plasmas
	DREYFUS Bernard	Thermodynamique
	DUCROS Pierre	Cristallographie
	FONTAINE Jean-Marc	Mathématiques pures
	GAGNAIRE Didier	Chimie physique
	GALVANI Octave	Mathématiques pures
	GASTINEL Noël	Analyse numérique
	GAVEND Michel	Pharmacologie
	GEINDRE Michel	Electroradiologie
	GERBER Robert	Mathématiques pures
	GERMAIN Jean-Pierre	Mécanique
	GIRAUD Pierre	Géologie
	JANIN Bernard	Géographie
	KAHANE André	Physique générale
	KLEIN Joseph	Mathématiques pures
	KOSZUL Jean-Louis	Mathématiques pures
	KRAVTCHENKO Julien	Mécanique
	LACAZE Albert	Thermodynamique
	LACHARME Jean	Biologie végétale
Mme	LAJZEROWICZ Janine	Physique
MM.	LAJZEROWICZ Joseph	Physique
	LATREILLE René	Chirurgie générale
	LATURAZE Jean	Biochimie pharmaceutique
	LAURENT Pierre	Mathématiques appliquées
	LEDRU Jean	Clinique médicale B
	LE ROY Philippe	Mécanique (I.U.T. I)

MM.	LLIBOUTRY Louis	Géophysique
	LOISEAUX Jean-Marie	Sciences nucléaires
	LONGEQUEUE Jean-Pierre	Physique nucléaire
	LOUP Jean	Géographie
Mlle	LUTZ Elisabeth	Mathématiques pures
MM.	MALINAS Yves	Clinique obstétricale
	MARTIN-NOEL Pierre	Clinique cardiologique
	MAYNARD Roger	Physique du solide
	MAZARE Yves	Clinique Médicale A
	MICHEL Robert	Minéralogie et pétrographie
	MICOUD Max	Clinique maladies infectieuses
	MOURIQUAND Claude	Histologie
	MOUSSA André	Chimie nucléaire
	NEGRE Robert	Mécanique
	NOZIERES Philippe	Spectrométrie physique
	OZENDA Paul	Botanique
	PAYAN Jean-Jacques	Mathématiques pures
	PEBAY-PEYROULA Jean-Claude	Physique
	PERRET Jean	Séméiologie médicale (neurologie)
	RASSAT André	Chimie systématique
	RENARD Michel	Thermodynamique
	REVOL Michel	Urologie
	RINALDI Renaud	Physique
	DE ROUGEMONT Jacques	Neuro-Chirurgie
	SARRAZIN Roger	Clinique chirurgicale B
	SEIGNEURIN Raymond	Microbiologie et hygiène
	SENGEL Philippe	Zoologie
	SIBILLE Robert	Construction mécanique (I.U.T. I)
	SOUTIF Michel	Physique générale
	TANCHE Maurice	Physiologie
	VAILLANT François	Zoologie
	VALENTIN Jacques	Physique nucléaire
Mme	VERAIN Alice	Pharmacie galénique
MM.	VERAIN André	Physique biophysique
	VEYRET Paul	Géographie
	VIGNAIS Pierre	Biochimie médicale

PROFESSEURS ASSOCIES

MM. CRABBE Pierre
SUNIER Jules

CERMO
Physique

PROFESSEURS SANS CHAIRE

Mlle	AGNIUS-DELORS Claudine	Physique pharmaceutique
	ALARY Josette	Chimie analytique
MM.	AMBROISE-THOMAS Pierre	Parasitologie
	ARMAND Gilbert	Géographie
	BENZAKEN Claude	Mathématiques appliquées
	BIAREZ Jean-Pierre	Mécanique
	BILLET Jean	Géographie
	BOUCHET Yves	Anatomie
	BRUGEL Lucien	Energétique (I.U.T. I)
	BUISSON René	Physique (I.U.T. I)
	BUTEL Jean	Orthopédie
	COHEN-ADDAD Jean-Pierre	Spectrométrie physique
	COLOMB Maurice	Biochimie médicale
	CONTE René	Physique (I.U.T. I)
	DELOBEL Claude	M.I.A.G.
	DEPASSEL Roger	Mécanique des fluides
	GAUTRON René	Chimie
	GIDON Paul	Géologie et minéralogie
	GLENAT René	Chimie organique
	GROULADE Joseph	Biochimie médicale
	HACQUES Gérard	Calcul numérique
	HOLLARD Daniel	Hématologie
	HUGONOT Robert	Hygiène et médecine préventive
	IDELMAN Simon	Physiologie animale
	JOLY Jean-René	Mathématiques pures
	JULLIEN Pierre	Mathématiques appliquées
Mme	KAHANE Josette	Physique
MM.	KRAKOWIACK Sacha	Mathématiques appliquées
	KUHN Gérard	Physique (I.U.T. I)
	LUU DUC Cuong	Chimie organique - pharmacie
	MICHOULIER Jean	Physique (I.U.T. I)
Mme	MINIER Colette	Physique (I.U.T. I)

MM.	PELMONT Jean	Biochimie
	PERRIAUX Jean-Jacques	Géologie et minéralogie
	PFISTER Jean-Claude	Physique du solide
Mlle	PIERY Yvette	Physiologie animale
MM.	RAYNAUD Hervé	M.I.A.G.
	REBECQ Jacques	Biologie (CUS)
	REYMOND Jean-Charles	Chirurgie générale
	RICHARD Lucien	Biologie végétale
Mme	RINAUDO Marguerite	Chimie macromoléculaire
MM.	SARROT-REYNAULD Jean	Géologie
	SIROT Louis	Chirurgie générale
Mme	SOUTIF Jeanne	Physique générale
MM.	STIEGLITZ Paul	Anesthésiologie
	VIALON Pierre	Géologie
	VAN CUTSEM Bernard	Mathématiques appliquées

MAITRES DE CONFERENCES ET MAITRES DE CONFERENCES AGREGES

MM.	ARMAND Yves	Chimie (I.U.T. 1)
	BACHELOT Yvan	Endocrinologie
	BARGE Michel	Neuro-chirurgie
	BEGUIN Claude	Chimie organique
Mme	BERIEL Hélène	Pharmacodynamie
MM.	BOST Michel	Pédiatrie
	BOUCHARLAT Jacques	Psychiatrie adultes
Mme	BOUCHE Liane	Mathématiques (CUS)
MM.	BRODEAU François	Mathématiques (I.U.T. B) (Personne étrangère habilitée à être directeur de thèse)
	BERNARD Pierre	Gynécologie
	CHAMBAZ Edmond	Biochimie médicale
	CHAMPETIER Jean	Anatomie et organogénèse
	CHARDON Michel	Géographie
	CHERADAME Hervé	Chimie papetière
	CHIAVERINA Jean	Biologie appliquée (EFP)
	COLIN DE VERDIERE Yves	Mathématiques pures
	CONTAMIN Charles	Chirurgie thoracique et cardio-vasculaire
	CORDONNER Daniel	Néphrologie
	COULOMB Max	Radiologie
	CROUZET Guy	Radiologie

MM.	CYROT Michel	Physique du solide
	DENIS Bernard	Cardiologie
	DOUCE Roland	Physiologie végétale
	DUSSAUD René	Mathématiques (CUS)
Mme	ETERRADOSSI Jacqueline	Physiologie
MM.	FAURE Jacques	Médecine légale
	FAURE Gilbert	Urologie
	GAUTIER Robert	Chirurgie générale
	GIDON Maurice	Géologie
	GROS Yves	Physique (I.U.T. I)
	GUIGNIER Michel	Thérapeutique
	GUITTON Jacques	Chimie
	HICTER Pierre	Chimie
	JALBERT Pierre	Histologie
	JUNIEN-LAVILLAVROY Claude	O.R.L.
	KOLODIE Lucien	Hématologie
	LE NOC Pierre	Bactériologie-virologie
	MACHE Régis	Physiologie végétale
	MAGNIN Robert	Hygiène et médecine préventive
	MALLION Jean-Michel	Médecine du travail
	MARECHAL Jean	Mécanique (I.U.T. I)
	MARTIN-BOUYER Michel	Chimie (CUS)
	MASSOT Christian	Médecine interne
	NEMOZ Alain	Thermodynamique
	NOUGARET Marcel	Automatique (I.U.T. I)
	PARAMELLE Bernard	Pneumologie
	PECCOUD François	Analyse (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse)
	PEFFEN René	Métallurgie (I.U.T. I)
	PERRIER Guy	Géophysique-glaciologie
	PHELIP Xavier	Rhumatologie
	RACHALL Michel	Médecine interne
	RACINET Claude	Gynécologie et obstétrique
	RAMBAUD Pierre	Pédiatrie
	RAPHAEL Bernard	Stomatologie
Mme	RENAUDET Jacqueline	Bactériologie (pharmacie)
MM.	ROBERT Jean-Bernard	Chimie-physique
	ROMIER Guy	Mathématiques (I.U.T. B) (Personnalité étrangère habilitée à être directeur de thèse)
	SAKAROVITCH Michel	Mathématiques appliquées

MM. SCHAERER René	Cancérologie
Mme SEIGLE-MURANDI Françoise	Crytogamie
MM. STOEBNER Pierre	Anatomie pathologie
STUTZ Pierre	Mécanique
VROUSOS Constantin	Radiologie

MAITRES DE CONFERENCES ASSOCIES

MM. DEVINE Roderick	Spectro Physique
KANEKO Akira	Mathématiques pures
JOHNSON Thomas	Mathématiques appliquées
RAY Tuhina	Physique

MAITRE DE CONFERENCES DELEGUE

M. ROCHAT Jacques	Hygiène et hydrologie (pharmacie)
-------------------	-----------------------------------

Fait à Saint Martin d'Hères, novembre 1977

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Année universitaire 1979-1980

Président : M. Philippe TRAYNARD
Vice-Présidents : M. Georges LESPINARD
M. René PAUTHENET

PROFESSEURS DES UNIVERSITES

MM.	ANCEAU François	Informatique fondamentale et appliquée
	BENOIT Jean	Radioélectricité
	BESSON Jean	Chimie Minérale
	BLIMAN Samuel	Electronique
	BLOCH Daniel	Physique du Solide - Cristallographie
	BOIS Philippe	Mécanique
	BONNETAIN Lucien	Génie Chimique
	BONNIER Etienne	Métallurgie
	BOUVARD Maurice	Génie Mécanique
	BRISSONNEAU Pierre	Physique des Matériaux
	BUYLE-BODIN Maurice	Electronique
	CHARTIER Germain	Electronique
	CHERADAME Hervé	Chimie Physique Macromoléculaires
Mme	CHERUY Arlette	Automatique
MM.	CHIAVERINA Jean	Biologie, Biochimie, Agronomie
	COHEN Joseph	Electronique
	COUMES André	Electronique
	DURAND Francis	Métallurgie
	DURAND Jean-Louis	Physique Nucléaire et Corpusculaire
	FELICI Noël	Electrotechnique
	FOULARD Claude	Automatique
	GUYOT Pierre	Métallurgie Physique
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du Solide - Cristallographie
	LACOUME Jean-Louis	Géographie - Traitement du Signal
	LANCIA Roland	Electronique - Automatique
	LESIEUR Marcel	Mécanique
	LESPINARD Georges	Mécanique
	LONGEQUEUE Jean-Pierre	Physique Nucléaire Corpusculaire
	MOREAU René	Mécanique
	MORET Roger	Physique Nucléaire Corpusculaire
	PARIAUD Jean-Charles	Chimie - Physique
	PAUTHENET René	Physique du Solide - Cristallographie
	PERRET René	Automatique

.../...

MM.	PERRET Robert	Electrotechnique
	PIAU Jean-Michel	Mécanique
	PIERRARD Jean-Marie	Mécanique
	POLOUJADOFF Michel	Electrotechnique
	POUPOT Christian	Electronique - Automatique
	RAMEAU Jean-Jacques	Chimie
	ROBERT André	Chimie Appliquée et des matériaux
	ROBERT François	Analyse numérique
	SABONNADIÈRE Jean-Claude	Electrotechnique
Mme	SAUCIER Gabrielle	Informatique fondamentale et appliquée
M.	SOHM Jean-Claude	Chimie - Physique
Mme	SCHLENKER Claire	Physique du Solide - Cristallographie
MM.	TRAYNARD Philippe	Chimie - Physique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNÝ François	Electronique

CHERCHEURS DU C.N.R.S. (Directeur et Maître de Recherche)

M.	FRUCHART Robert	Directeur de Recherche
MM.	ANSARA Ibrahim	Maître de Recherche
	BRONOËL Guy	Maître de Recherche
	CARRE René	Maître de Recherche
	DAVID René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	KAMARINOS Georges	Maître de Recherche
	KLEITZ Michel	Maître de Recherche
	LANDAU Ioan-Doré	Maître de Recherche
	MERMET Jean	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Personnalités habilitées à diriger des travaux de recherche (décision du Conseil Scientifique)

E.N.S.E.E.G.

MM.	ALLIBERT Michel
	BERNARD Claude
	CAILLET Marcel
Mme	CHATILLON Catherine
MM.	COULON Michel
	HAMMOU Abdelkader
	JOUD Jean-Charles
	RAVAINE Denis
	SAINFORT

C.E.N.G.

MM. SARRAZIN Pierre
 SOUQUET Jean-Louis
 TOUZAIN Philippe
 URBAIN Georges

Laboratoire des Ultra-Réfractaires ODEILLO

E.N.S.M.E.E.

MM. BISCONDI Michel
 BOOS Jean-Yves
 GUILHOT Bernard
 KOBILANSKI André
 LALAUZE René
 LANCELOT François
 LE COZE Jean
 LESBATS Pierre
 SOUSTELLE Michel
 THEVENOT François
 THOMAS Gérard
 TRAN MINH Canh
 DRIVER Julian
 RIEU Jean

E.N.S.E.R.G.

MM. BOREL Joseph
 CHEHIKIAN Alain
 VIKTOROVITCH Pierre

E.N.S.I.E.G.

MM. BORNARD Guy
 DESCHIZEAUX Pierre
 GLANGEAUD François
 JAUSSAUD Pierre
 Mme JOURDAIN Geneviève
 MM. LEJEUNE Gérard
 PERARD Jacques

E.N.S.H.G.

M. DELHAYE Jean-Marc

E.N.S.I.M.A.G.

MM. COURTIN Jacques
 LATOMBE Jean-Claude
 LUCAS Michel
 VERDILLON André

Je tiens particulièrement à remercier ici :

Monsieur Louis BOLLIET, Professeur à l'Université Scientifique et Médicale de Grenoble, de m'avoir fait l'honneur de présider le jury de cette thèse ;

Monsieur Claude DELOBEL, Professeur à l'Université Scientifique et Médicale de Grenoble, pour m'avoir donné les moyens de mener à bien cette recherche. Je veux aussi lui exprimer ma plus sincère gratitude pour la confiance qu'il m'a toujours témoignée, pour sa totale disponibilité à mon égard, pour ses nombreux conseils et encouragements sans lesquels ce travail n'aurait pas été possible ;

Messieurs Michel ADIBA, animateur et responsable du projet POLYPHEME, et Edouard ANDRE, Ingénieur de recherche au Centre National d'Etude des Télécommunications, pour leurs critiques et conseils concernant le manuscrit de ce travail et leur participation au jury ;

Tous ceux qui ont contribué aux travaux rapportés ici ;

Je suis également reconnaissant :

à Madame Andrée CHAPEL-STIERS et Monsieur Gilles BOGO qui ont lu et relu mon manuscrit pour permettre d'en améliorer la rédaction. Bien entendu il est très difficile de tout corriger et le lecteur voudra bien m'excuser pour les erreurs qu'il pourrait trouver ;

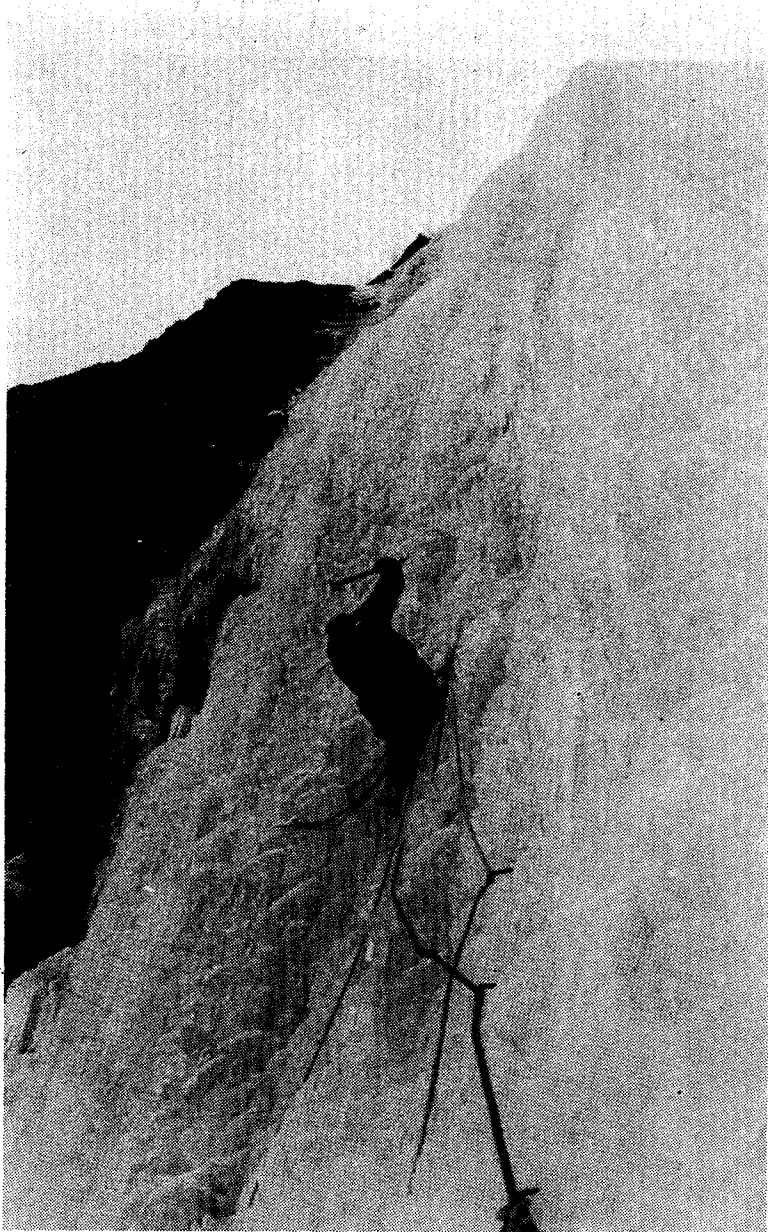
à tous les membres de l'équipe POLYPHEME avec lesquels j'ai eu d'excellentes relations de travail ;

à la nature merveilleuse autour de Grenoble où je reposais mon esprit ;

aux gouvernements français et coréen qui m'ont offert cette occasion d'étude en France ;

à ma femme et mes enfants qui ont bien adopté une vie étrangère qui leur a souvent donné des difficultés et des souffrances ;

au service de reprographie qui a terminé la réalisation matérielle de ce travail.



이제 하늘로 끝없는 산들 밭길에서
거기 모퉁이를 품는 것이
이미 한 영광이라면
그것을 나의 부모와 아내에게 바친다.
1981년 정월
à Grenoble, 白石.

(Maintenant qu'il est trop tard,
je suis arrivé aux pieds des
montagnes qui lèvent leurs têtes
vers le ciel.
Je prends la décision d'y monter,
et si cela est déjà un honneur,
je le donnerai à mes parents et
à ma femme.)

1968 .1. A la Cascade "To-Wang-Sung" du "Sul-Ac" de
la Corée (moi-même)

La "Montagne Blanche" (Sul-Ac) coréenne atténuera mes regrets
de Grenoble et de ses montagnes. Celle-là attirera certainement
quelques informaticiens grenoblois dans mon pays.

RESUME :

Cette thèse a pour objectif de proposer une démarche pour effectuer l'optimisation des requêtes dans un système de bases de données réparties. Dans une première partie on énumère l'ensemble de tous les critères d'optimisation qui peuvent être pris en compte et on décide les objectifs d'optimisation. Dans une deuxième partie on propose une démarche progressive divisée en plusieurs étapes qui visent aux objectifs d'optimisation : la localisation, les permutations des opérations grâce aux propriétés algébriques, l'allocation des données réparties par le regroupement et la distribution des opérations aux processus répartis par le partitionnement. Dans une troisième partie on montre en quoi la démarche, d'une part, se distingue d'autres approches qui ont été prises en compte dans des réalisations ou des propositions, et d'autre part, quels avantages elle possède. On termine en donnant une caractérisation des applications qui sont susceptibles d'être traitées par cette méthode.

Mots clés :

optimisation de traitement réparti, permutation des opérations, décomposition, allocation des données réparties, distribution des opérations, partitionnement des opérations, matérialisation, processus répartis, utilisation des données redondantes.

PLAN DE THESE

I. INTRODUCTION

- 1.1. Les Systèmes Généraux de Base de Données Réparties
- 1.2. Les Problèmes d'Efficacité des Traitements Réparties

II. UNE APPROCHE POUR OPTIMISER LE TRAITEMENT DES REQUETES DANS UN ENVIRONNEMENT REPARTI

- 2.1. Introduction
- 2.2. Les Différents Niveaux de Représentation

III. DECOMPOSITION ET OPTIMISATIONS

- 3.1. Introduction
- 3.2. Localisation des Relations Globales
- 3.3. Les Optimisations par Permutations des Opérateurs
- 3.4. Allocation des Données Réparties
- 3.5. Distribution des Opérations
- 3.6. Sélection d'une Arborescence de Transactions
- 3.7. Génération d'un Ensemble de Processus Répartis

IV. EVALUATION

4.1. Introduction

4.2. Evaluation par une Classification des Stratégies des Traitements
Répartis

4.3. Evaluation par le Mesure de Satisfaction aux Objectifs

V. LES APPLICATIONS DE NOS STRATEGIES

5.1. Introduction

5.2. L'Applicabilité de Nos Stratégies

VI. CONCLUSION

BIBLIOGRAPHIES

CHAPITRE I

I N T R O D U C T I O N

Sommaire

1.1. LES SYSTEMES GENERAUX DE BASE DE DONNEES REPARTIES	1.1
1.1.1. Pourquoi les Systèmes de Bases de Données Réparties	1.1
1.1.2. Les Hypothèses des Approches des SGBDR	1.3
1.1.3. Les Problèmes Techniques pour le Développement	1.6
1.2. LES PROBLEMES D'EFFICACITE DES TRAITEMENTS REPARTIES	1.7

1.1. LES SYSTEMES GENERAUX DE BASE DE DONNEES REPARTIES

1.1.1. Pourquoi les systèmes de bases de données réparties

L'approche base de données, introduite depuis une bonne quinzaine d'années, avait pour tâche essentielle de supprimer les inconvénients multiples d'une informatique de gestion basée sur la création et l'exploitation de fichiers spécifiques, ceci introduisant indubitablement un cloisonnement préjudiciable entre les traitements. L'intérêt d'une telle approche était l'intégration de données structurées évitant des duplications de données. Avec l'utilisation de base de données, les grosses entreprises et administrations ont été aussi confrontées au problème délicat du choix de type d'organisation de leur systèmes d'information parce qu'il existe deux types d'applications : (1) les applications locales distribuées sur un large espace géographique chacune ne faisant que la gestion d'une parties des données, (2) les applications coopératives qui gèrent la totalité des données. Pour les applications locales, il parait intéressant de distribuer des données pour que chaque application locale ait seulement ses propres données, mais pour les applications coopératives, il parait bénéfique de centraliser les données en un seul endroit.

Face à cette alternative contraignante, les progrès technologiques réalisés durant ces dernières années, principalement ceux menés dans les réseaux généraux de télécommunication et les développement d'équipement informatique relativement peu coûteux, apportent une autre solution en permettant une "intégration décentralisée". Celle-ci rend possible la mise en place d'applications réparties qui utilisent des bases de données elles-même réparties. Ces dernières bénéficient des avantages de l'approche base de données avec une vision "globale" de données intégrées tandis que dans le

même temps elles n'imposent en aucune façon une organisation centralisée. En fait, dans la communauté des chercheurs et des équipes de développement, tout le monde s'accorde à définir l'architecture générale d'un système de gestion de base de données réparties (SGBDR) comme un ensemble d'installations informatiques, chacune supportant une portion de la base de données réparties gérée par un SGBD local, interconnectées de façon à pouvoir communiquer et échanger de l'information. Les SGBDR se situent donc au croisement de plusieurs disciplines de l'informatique (base de données, système de gestion, réseau d'ordinateurs, organisation, etc).

L'approche SGBDR a trois avantages principaux par rapport à l'approche SGBD centralisé :

(1) Amélioration de la disponibilité et sécurité des données :

Un SGBDR se compose d'un ensemble d'installations informatiques, les SGBDR ne sont donc pas susceptibles d'une panne totale dans le cas où une ou plusieurs installations sont en panne ou inaccessibles.

(2) Amélioration des performances :

Puisqu'il existe plusieurs installations de données, il est possible d'allouer les données aux installations où elles sont utilisées le plus souvent. Cela a comme effets l'accélération des accès et la réduction des coûts de transmission.

(3) Amélioration de la flexibilité de capacité du système :

Puisqu'un SGBDR se compose d'un ensemble de bases de données locales, l'expansion de la capacité du système peut être résolue par l'addition des nouvelles bases de données locales.

1.1.2. Les hypothèses des approches des SGBDR

L'approche SGBDR, approche nouvelle, n'est pas encore fondée sur les concepts bien développés et sur des hypothèses bien stables. Il existe donc différents concepts et hypothèses dans les approches SGBDR. Nous voulons citer ceux qui nous paraissent les plus importants [14].

(A) La visibilité de la distribution des données :

Dans le plus part des approches SGBDR, la distribution des données est supposée être invisibles des utilisateurs [1, 3, 6, 10]. Les utilisateurs peuvent interroger le SGBDR comme un SGBD centralisé. La distribution des données peut être rendre visible aux utilisateurs pour certain raisons, par exemple, l'amélioration des performances [2].

(B) La distribution de données :

On peut distinguer trois types différents pour la distribution des données.

a. Répartition à redondance totale (duplication totale)

C'est le cas où toutes les données sont identiquement dupliquées sur toutes les installations. Dans cette approche, chaque installation a son propre SGBDR comme SGBD centralisé, cela permet de maximiser la disponibilité, la sécurité et la performance des applications locales. Mais il est clair que cela crée des problèmes de stockage et de mise à jour à cause de nombreuses duplications redondantes.

b. Partitionnement (sans duplication)

C'est le cas où chaque installation contient un sous-ensemble unique de la base de données répartie. Cela peut être réalisé par l'allocation de chaque fragment à une seule installation. Cette

approche permet de résoudre les problèmes de l'approche précédente (répartition à redondance totale) mais elle perdra presque tous ses avantages puisqu'il n'existe pas de duplications redondantes.

c. Répartition à redondance partielle (duplication partielle)

C'est le cas où chaque installation contient un sous-ensemble non unique de la base de données répartie. Les fragments des installations peuvent se chevaucher partiellement. C'est à dire que certaines données peuvent être dupliquées sur plusieurs installations. Cette approche permet de résoudre les problèmes des approches précédentes, a et b, en tentant d'obtenir tous les avantages de SGBDR qui sont déjà présentés à la section 1.1.1. Tout le problème est de trouver le point d'équilibre.

(C) Les Caractéristiques du réseau de communication :

Il existe de nombreux paramètres qui caractérisent un réseau de communication, nous voulons citer ceux qui nous paraissent les plus importants [13] :

- Les types de topologie du réseau :
type anneau, totalement maillé et partiellement maillé.
- Les types de communication :
communication point à point et à diffusion.
- Les types de transmission :
commutation de circuits, de paquets et de messages.

(D) Les types d'applications :

Un SGBDR est en fait un modèle d'un ou plusieurs types des systèmes réels d'application. Il est clair que une approche de SGBDR ne peut

pas convenir à tous les types d'applications réelles. Il faut donc définir les types d'application pour chaque approche de SGBDR. Nous considérons les aspects suivants qui nous paraissent les plus importants pour définir les types d'applications :

- taux d'applications locales sur l'applications coopératives,
- taux de mise à jour et leur complexité,
- volume de données,
- espace de répartition des données,
- sémantique des données,
- exigences prioritaires,
- caractéristiques des requêtes, etc.

Nous discuterons précisément ce sujet dans le chapitre 5 pour montrer que notre approche peut être mise en oeuvre.

(E) Les modèles de base de données répartie :

Les modèles hiérarchique, réseau et relationnel peuvent être utilisés comme le modèle de données. Le modèle relationnel était utilisé dans la plupart des approches de SGBDR.

Nous venons de présenter les principales hypothèses dans les approches SGBDR. Nous voulons fixer les hypothèses que nous prendrons dans la suite de notre travail.

- [Hypothèses Prises en Compte dans Notre Approche] -

-
- La distribution des données n'est pas visible des utilisateurs.
 - La répartition se fait à redondance partielle.
 - Le réseau est partiellement maillé et la communication se fait point à point et la transmission se fait par paquets.
 - Le type d'application consiste principalement en des grosses applications typiques pour SGBDR (voir chapitre 5).
 - Le modèle de base de données répartie est le modèle relationnel.
-

1.1.3. Les problèmes techniques pour le développement

Tous les problèmes techniques qui existaient dans les développements des SGBD centralisés prennent ici une dimension supplémentaire par la nature des données et des processeurs répartis. Nous voulons citer les principaux problèmes techniques qui nous paraissent les plus importants [1, 10, 14].

- Le problème de synchronisation pour maintenir la cohérence des données,
- Les problèmes d'efficacité des traitements répartis :
 - Le problème de la transformation de la requête externe en plusieurs requêtes locales,
 - Le problème d'allocation des ressources distribués (données et processeurs),
 - Le problème d'interprétation,

- Le problème d'intégrité causé par les pannes des installations,
- Le problème de gestion des dictionnaires du système,
- Le problème d'allocation des données (y compris des duplications),
- Le problème d'homogénéisation dans le cas où les SGBD locaux sont hétérogènes, etc.

Dans cette thèse, nous nous sommes intéressés spécialement aux problèmes d'efficacité des traitements répartis, et nous donnons plusieurs stratégies pour résoudre ce type de problème.

1.2. LES PROBLEMES D'EFFICACITE DES TRAITEMENTS REPARTIES

Puisque les SGBDR utilisent des processeurs et des données qui se distribuent largement dans un réseau de communication, ils posent des nouveaux problèmes technologiques à cause de leurs exigences inhérentes à la communication des informations et à cause de la potentialité de traitement parallèle. Cela signifie que, en ce qui concerne les performances du système, les problèmes d'optimisation seront plus importants dans l'approche SGBDR que dans l'approche SGBD classique. Afin d'optimiser les performances du système, il est nécessaire de formuler les exigences de l'utilisateur et celles du système [4].

- [Exigences] -

Au niveau de l'utilisateur :

- minimisation du temps de réponse
- minimisation du coût de traitements
- maximisation de la disponibilité des données

Au niveau du système :

- maximisation de la productivité du système total
-

Puisque le temps de réponse se compose des durées d'exécution et de transmission, la minimisation du temps de réponse peut être obtenue plus précisément par :

- minimisation du temps d'exécution,
- maximisation d'exécutions parallèles,
- minimisation du temps de transmission,
- maximisation de transmissions parallèles.

Puisque le coût du traitement se compse des coûts d'exécution, de transmission et de stockage, la minimisation du coût de traitements peut être obtenue plus précisément par :

- minimisation du temps d'exécution,
- minimisation du temps de transmission,
- minimisation des stockages nécessaires.

Ceci suppose que les coûts d'exécution et de transmission par unité du temps sont les mêmes en tout point du SGBDR.

La disponibilité des données peut s'améliorer par des duplications redondantes, alors la maximisation de la disponibilité des données peut être obtenue par :

- maximisation des duplications redondantes.

La productivité du système SGBDR peut principalement dépendre de la distribution du trafic dans le réseau et de la distribution de la charge des processeurs, alors la maximisation de la productivité du système peut être obtenue par :

- maximisation de la distribution de la charge des processeurs et de la distribution du trafic dans le réseau.

Il est certain que toutes ces exigences ne peuvent pas être satisfaites à la fois car certaines sont conflictuelles, par exemple, les minimisations du temps d'exécution et de transmission peuvent être en conflit avec les maximisations du parallélisme d'exécution et de transmission (voir section 3.5.3.). Il est aussi évident que la minimisation du stockage est en conflit avec la maximisation des duplications redondantes.

Pourtant tout le monde s'accorde à penser que la transmission des données est le principal goulot d'étranglement des performances des SGBDR car tous les moyens de communication à longue distance ont un débit relativement faible [13] par rapport à des communications sur courte distance.

Nous avons donc envisagé les objectifs principaux suivants :

- Objectifs -

-
1. minimisation du temps de transmission
 2. minimisation du temps d'exécution
 3. amélioration du parallélisme dans les exécutions et transmissions
 4. amélioration de la distribution du trafic et des charges
 5. réduction des duplications redondantes
-

Cet ensemble d'objectifs doit être pris en considération dans les développements de SGBDR afin d'en améliorer leurs performances.

Malheureusement, la plupart des SGBDR actuels et en cours de développement ne donnent pas entière satisfaction par rapport à ces critères d'optimisation [14, 20]. Dans cette thèse nous allons proposer une approche nouvelle qui

optimise le traitement des requêtes dans un environnement réparti en tentant de satisfaire tous ces critères d'optimisation.

Au chapitre 2, nous présenterons la démarche générale de notre approche : les différents niveaux de représentation des données et les différents niveaux d'exécution.

Au chapitre 3, nous présenterons de manière détaillée les stratégies d'optimisation pour la décomposition de la requête externe : les optimisations de localisation des données par la permutation des opérateurs, l'optimisation d'allocation des données par une sorte de matérialisation et l'optimisation de la distribution des opérations vers des processeurs distribués.

Au chapitre 4, nous évaluerons notre approche en la comparant aux autres approches et nous montrerons l'applicabilité de notre approche pour les différents types d'applications réelles au chapitre 5.

En conclusion nous indiquerons les problèmes non résolus dans cette thèse et nous ferons des propositions pour des études futures.

CHAPITRE II

UNE APPROCHE POUR OPTIMISER LE TRAITEMENT
DES REQUETES DANS UN ENVIRONNEMENT REPARTI

Sommaire

2.1. INTRODUCTION	2.1
2.2. LES DIFFERENTS NIVEAUX DE REPRESENTATION	2.2
2.2.1. Les Niveaux des Données	2.2
2.2.2. Les Niveaux d'Exécution d'une Requête	2.7

2.1. INTRODUCTION

Les premiers prototypes de bases de données réparties sont apparus [1, 2] récemment [3, 4, 6, 7, 8, 10] et nombreux autres sont en cours de développement. Parmi les problèmes techniques qui ont du être résolue, celui de l'exécution des requêtes est un des plus importants dans la mesure où la stratégie d'exécution des requêtes conditionne en partie les performances. Les systèmes actuels donnent des réponses partielles à ce problème [14].

Le but de cette thèse est de présenter un modèle intégrant les principaux paramètres qui interviennent dans l'optimisation des requêtes.

Nous supposerons que le modèle de données utilisées est le modèle relationnel et que l'utilisateur interroge la base de données à l'aide de requêtes exprimées dans un langage du type algébrique. La première hypothèse concorde bien avec le fait que la majorité des systèmes développés ou en cours de développement utilise le modèle relationnel. Quant à la seconde, elle n'est nullement restrictive puisqu'il est prouvé que le langage algébrique est équivalent aux langages du type prédicatif [15] et que d'autres types comme par exemple le langage SEQUEL, sont complets car ils peuvent simuler les opérations des langages algébriques ou prédicatifs [16].

Le choix du langage algébrique est du à la possibilité d'exprimer toute requête sous la forme d'un arbre syntaxique dans lequel les opérandes sont les relations et les opérateurs des opérations algébriques. Nous verrons que cette structure est particulièrement bien adaptée à nos objectifs d'optimisation.

Dans la suite nous supposerons que le lecteur est familier avec les notions de base du modèle relationnel et des opérateurs algébriques qui sont utilisés

Parmi les opérateurs que nous envisagerons, nous citerons : l'union, le produit cartésien, la projection, la sélection et la différence qui forment un noyau minimal, et d'autres qui sont utilisés pour des raisons de commodité, comme le produit, l'antiprojection, l'intersection, etc. Le lecteur pourra se reporter aux études [11] et [15].

2.2. LES DIFFERENTS NIVEAUX DE REPRESENTATION ET D'EXECUTION

2.2.1. Les niveaux des données

Dans un système de bases de données réparties, nous considérons trois niveaux principaux :

- le niveau global
- le niveau local
- le niveau des copies

Le niveau global correspond à la vue de l'utilisateur qui connaît l'ensemble des noms des relations, et leurs attributs, accessibles dans la base de données répartie. Les relations de ce niveau, appelées relations globales, n'ont pas d'existence matérielle dans la base de données mais seulement une existence logique ou virtuelle dans la mesure où l'on peut reconstituer l'ensemble des n-uplets définissant une relation en utilisant les propriétés des niveaux inférieurs.

Le niveau local correspond aussi à un espace de relations appelées relations locales. Le passage du niveau global au niveau local se fait en indiquant pour chaque relation globale comment elle peut être construite à partir des relations locales permettant ainsi d'exprimer une partie de la répartition des données. Nous envisageons ici trois types de construction possible :

- l'identité : dans ce cas la relation globale est identique à la relation locale, on parle parfois de relation collectée ;
- répartition horizontale : la relation globale peut s'exprimer comme l'union de plusieurs relations locales ;
- répartition verticale, la relation globale peut s'obtenir en faisant le produit de plusieurs relations locales.

Si en fait, on considère trois principes de répartitions, rien n'interdit de considérer que toute relation globale peut s'exprimer à l'aide d'une expression algébrique ne faisant intervenir que des relations locales. Cette répartition n'est pas connue de l'utilisateur mais seulement du système.

Enfin le niveau des copies correspond à la matérialisation des relations locales sur des sites du réseau d'ordinateurs. C'est ainsi que pour chaque relation locale on indiquera la localisation des copies. Seules les copies ont une existence réelle dans la base de données répartie et correspondent à l'expression finale de la répartition des données.

Les trois niveaux différents de la représentation des données et l'expression de la répartition sont présentés dans la figure 1.

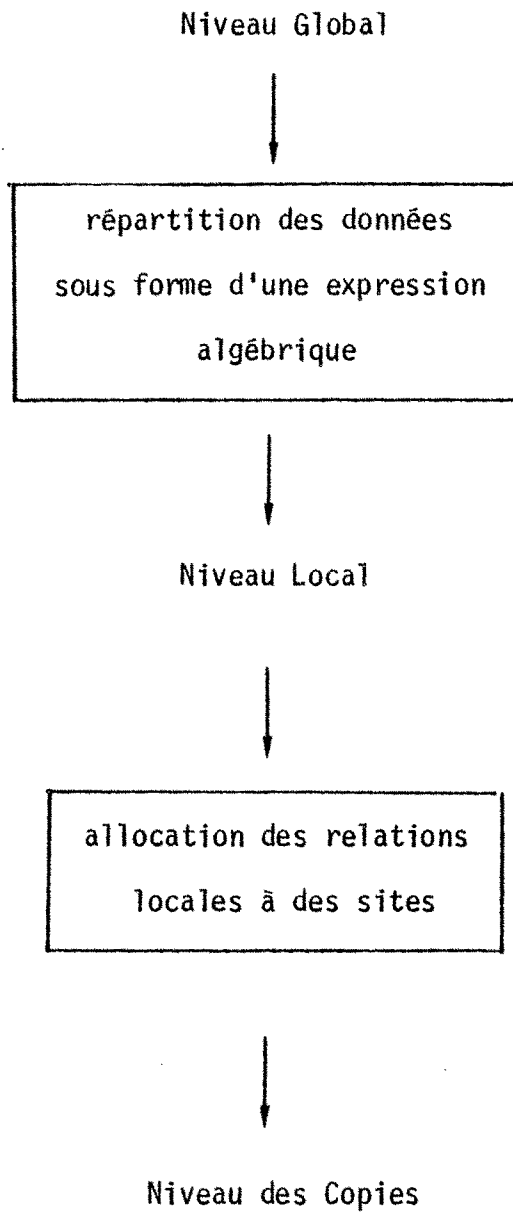


Figure 1 : les Différents Niveaux de la Représentation des Données et l'Expression de la Répartition

Pour concrétiser ces notions nous considérerons un exemple spécifique qui servira d'illustration tout au long de cette thèse.

- [Exemple] -

Les trois relations globales qui sont connues des utilisateurs sont données par la figure 2.

relations globales	attributs	variable	type	relations locales-
fournisseur	SNO, SNAME, CITY	S	partitionné horizontalement	s1, s2, s3
approvisionnement	SNO, PNO, AMT	Y	collecté	y
pièce	PNO, PNAME	P	collecté	p

SNO : numéro de fournisseur
SNAME : nom du fournisseur
CITY : ville du fournisseur
PNO : numéro de pièce
AMT : quantité approvisionnée
PNAME : nom de la pièce

Figure 2 : les Relations Globales et Leur Répartition

La relation S est partitionnée en trois relations locales dépendant du numéro de fournisseur SNO, le critère de répartition est le suivant :

- la relation locale s1 est composée des n-uplets de S tels que le numéro du fournisseur SNO est plus petit que 100.

- la relation locale s2 est composée des n-uplets de S tels que le numéro du fournisseur SNO est compris entre 100 et 1 000.
- la relation locale s3 est composée des n-uplets de S tels que le numéro du fournisseur SNO est supérieur à 1 000.

Nous supposons que les copies multiples associées à ces relations locales sont réparties comme le montre l'exemple du réseau d'ordinateur dont la topologie est donnée par la figure 3.

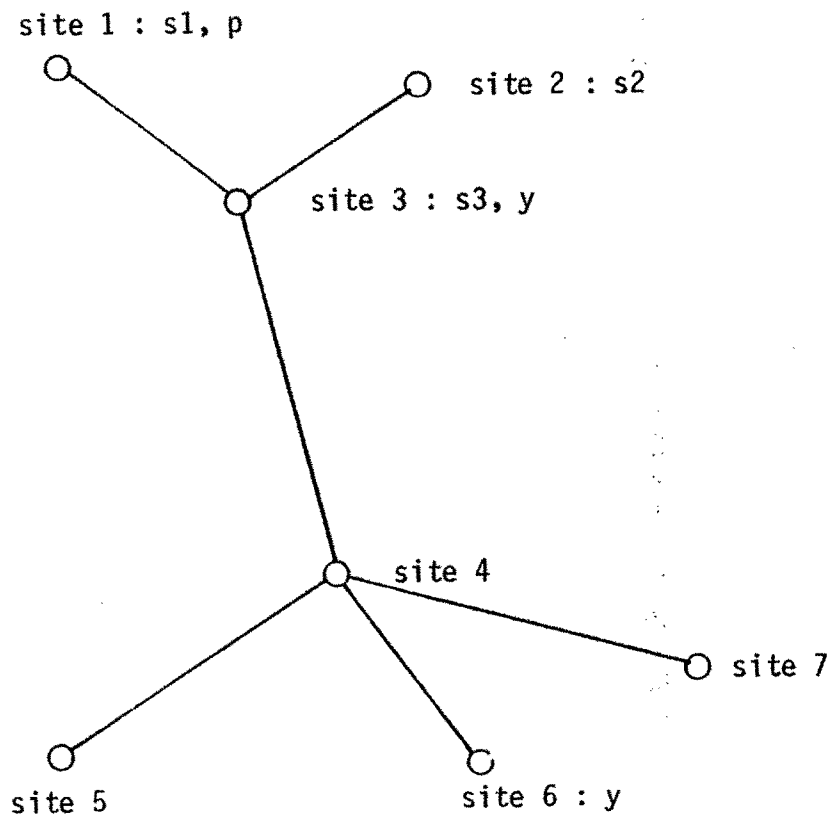


Figure 3 : la Structure du Réseau et la Distribution des Copies

Dans cet exemple, seule la relation locale "y" a des copies multiples en des sites différents.

Une requête externe, demandant la liste des numéros de pièces PNO, des noms de fournisseurs SNAME, et de la qualité approvisionnée AMT pour toutes les roues "wheels" produites à Paris et dont la quantité AMT est supérieure à 1 000 pour un certain fournisseur, est émise à partir du site 7. Cette requête peut être représentée par une expression algébrique de la forme :

$$\begin{aligned} &(((P[pname = "wheels"] \overset{PNO}{*} Y) [AMT > 1\ 000]) \overset{SNO}{*} \\ & (S[city = "Paris"])) [PNO, SNAME, AMT] \end{aligned}$$

où l'opération produit est notée par "nom d'attribut", l'opération de projection est notée par "[noms d'attributs]", et l'opération de sélection est notée par "[expression conditionnelle]".

Cette requête peut être exprimée d'une façon équivalente sous la forme d'un arbre syntaxique (figure 4) où les symboles " \triangleleft ", " \triangleright ", et " \bowtie " dénotent respectivement les opérations de projection, sélection et produit de l'algèbre relationnelle et " \square " désigne le nom d'une relation.

2.2.2. Les niveaux d'exécution d'une requête

La démarche générale du traitement d'une requête peut être représentée par la figure 5. Elle est divisée en deux niveaux principaux : le niveau d'optimisation et le niveau d'exécution répartie. Le premier niveau correspond à la phase de génération d'une stratégie "optimale" d'exécution de la requête. Cette phase est réalisée en plusieurs étapes au cours desquelles on effectue des transformations successives sur la requête initiale ex-

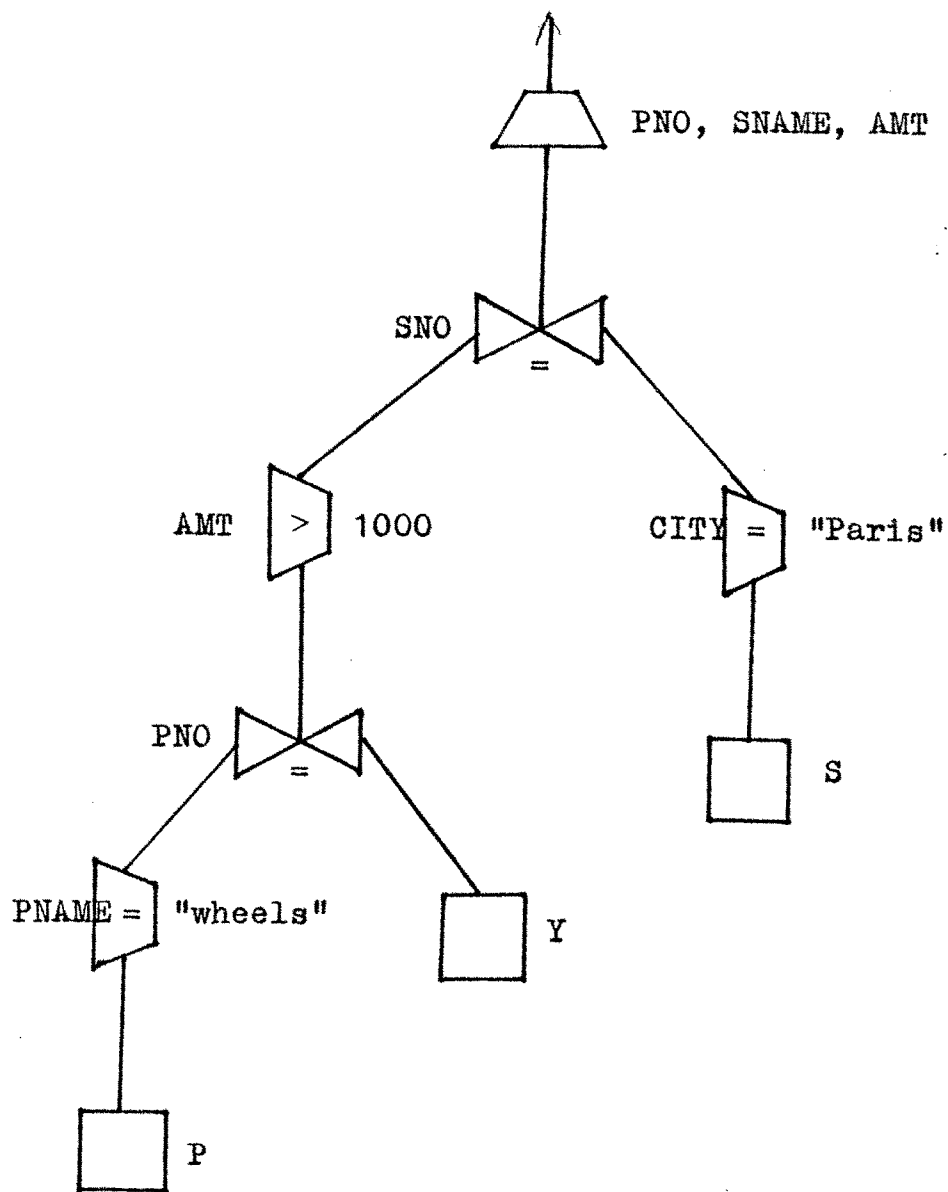


Figure 4 : l'arbre syntaxique de la requête externe

primée au niveau global, tout en explorant différentes stratégies et en les évaluant pour retenir la meilleure. Le deuxième niveau est un niveau d'exécution répartie au cours duquel la requête est effectivement traitée suivant la stratégie élaborée dans le niveau d'optimisation.

Parmi les optimisations que l'on peut effectuer, on en envisagera de deux types : les optimisations locales à une étape, les optimisations globales pour l'ensemble des étapes. Une optimisation locale consiste à fixer au cours d'une étape un paramètre à une valeur donnée. Pour toutes les étapes suivantes le paramètre prendra la valeur déterminée. Par contre dans le cas d'une optimisation globale tous les paramètres sont laissés libres, on attend d'arriver à la dernière étape pour en faire une évaluation suivant des critères qui seront examinés.

Dans l'approche qui est ici proposée nous verrons que partant d'une requête initiale il y a à chaque étape un nombre croissant de solutions possibles. Parmi toutes les solutions possible il faudra en retenir une, celle qui sera transmise au niveau d'exécution répartie.

La démarche que nous proposons est une optimisation statique puisqu'elle consiste à fixer une stratégie d'exécution, stratégie qui sera exécutée conformément au plan prévu. Il existe une autre classe d'optimisation, l'optimisation dynamique [2, 8, 9] dans laquelle au cours de l'exécution la stratégie est modifiée en fonction d'un certain nombre de paramètres.

Avant d'examiner dans le détail les différentes étapes du processus d'optimisation nous allons présenter le rôle de chacune de ces étapes en fonction du problème à résoudre. Nous avons déjà dit ci-dessus que chaque étape peut être vue comme des transformations successives sur l'arbre syntaxique de la requête initiale. Examinons chaque transformation

indépendamment de l'ordre dans lequel elle est effectuée.

La première transformation correspond à l'étape de localisation. Elle consiste à remplacer les relations globales par leur expression en termes de relations locales en fonction du type de partitionnement des données. On obtient un nouvel arbre syntaxique de même nature dans lequel les feuilles de l'arborescence sont cette fois des relations locales.

La deuxième transformation correspond en fait à un groupe de transformations possibles, toutes liées à des propriétés des opérateurs de l'algèbre relationnelle. Elle consiste en :

- la permutation des opérateurs de projection et de sélection afin de les placer le plus près possible des feuilles de l'arborescence,
- la permutation des opérateurs de produit et d'union grâce à la propriété de double distributivité,
- la permutation des opérateurs de produit grâce à la propriété de commutativité.

La première transformation engendre un nouvel arbre syntaxique unique équivalent au premier tandis que les deux autres transformations peuvent en engendrer plusieurs qui sont aussi équivalents au premier.

Un autre problème est celui de l'allocation des relations locales à des sites. En effet chaque relation locale est matérialisée par une copie située sur un site du réseau. Parmi les différents choix possibles il faudra en retenir un.

Un autre problème est celui de distribution des opérations de chaque arbre syntaxique engendré par les transformations précédentes. Celui-là sera résolu par un groupe de traitement qui consiste en :

- la définition des transactions initiales : une fois que la décision d'allocation des relations locales a été prise, on peut examiner quels sont les regroupements d'opérations qui peuvent être effectués sur un site en vue d'engendrer ce que nous appellerons une transaction initiale pour chaque arbre syntaxique engendré par les transformations précédentes,
- la définition des transactions intermédiaires : cette phase peut être résolue par une transformation sur chaque arbre syntaxique qui consiste à regrouper plusieurs opérateurs à condition que les données utilisées par ces opérateurs puissent être localisées sur le site qui exécutera cette tâche, afin de distinguer ce type de regroupement de ceux qui ont conduit à la notion de transaction initiale on les appellera des transactions intermédiaires,
- la distribution optimale des transactions intermédiaires de chaque arbre syntaxique engendré par les transformations précédentes sur les sites du réseau en prenant en compte les différents choix possible et selon un critère donné, par exemple : le volume des données transférées pour exécuter l'ensemble des transactions doit être minimal,
- la sélection de l'arbre syntaxique optimal parmi tous les arbres syntaxiques engendrés par les transformations précédentes en estimant le coût d'exécution de chaque arbre.

Enfin la dernière étape consiste à exprimer la synchronisation d'un ensemble d'actions pour exécuter la requête à l'aide de la notion de processus réparti.

L'ordre dans lequel nous avons décidé d'effectuer les différentes transformations et décisions est celui présenté par la figure 5. Pour des

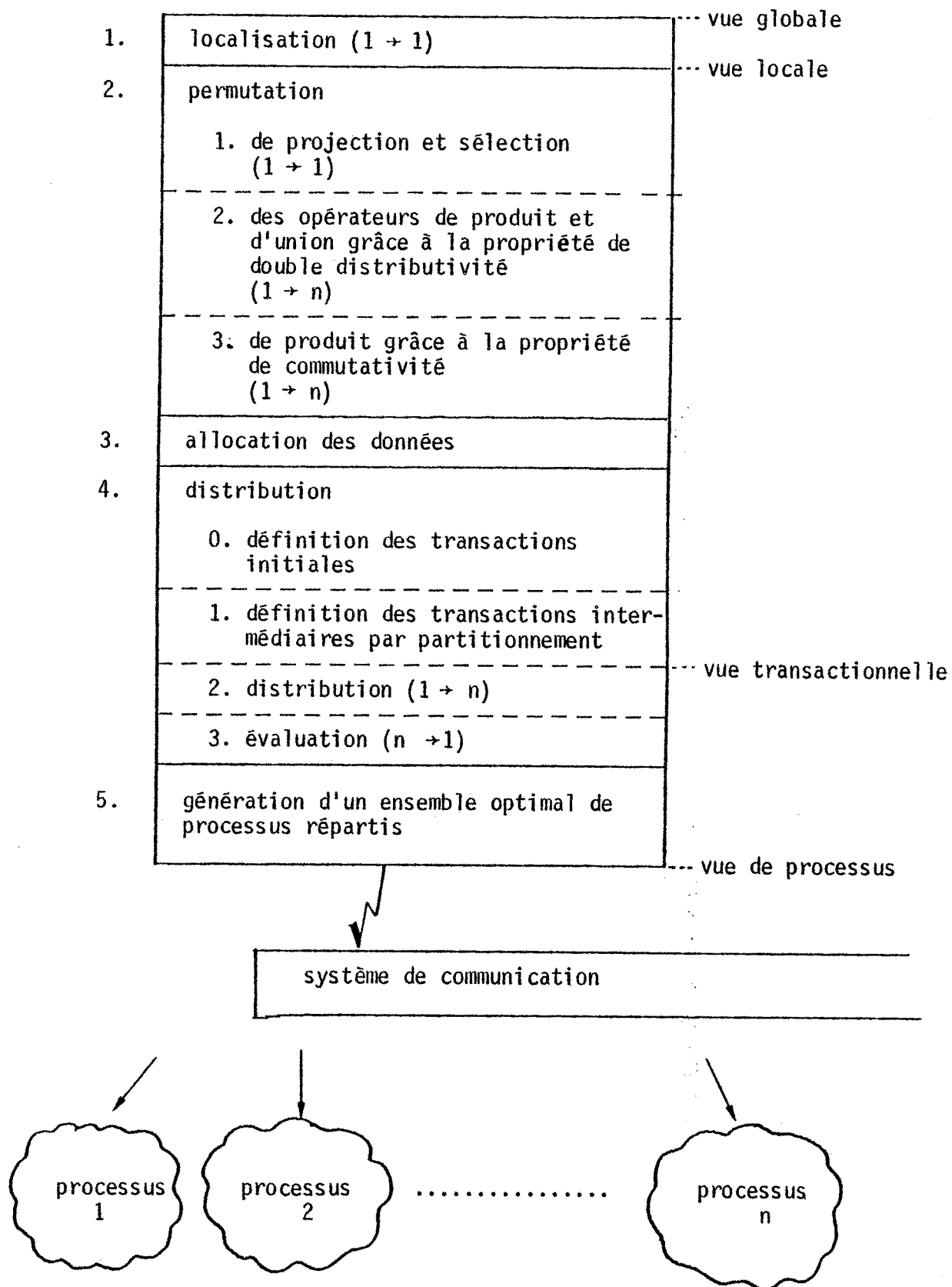


Figure 5 : la Démarche de Stratégie

raisons de présentation il nous a semblé nécessaire de désigner par un nom différent le résultat de nos transformations successives. C'est ainsi que si la requête initiale est exprimée en termes de la vue globale, après l'étape 1 elle correspond à une vue locale, puis au milieu de l'étape 4 à une vue transactionnelle, et enfin après l'étape 5 à une vue de processus. Ces notions de vues différentes correspondent aussi à des unités différentes de représentation des noeuds de l'arbre syntaxique :

- Les opérations : ce sont les opérateurs de l'algèbre relationnelle du langage, ils seront utilisés dans les étapes 1, 2, 3 et 4.
- Les transactions : ce sont les ensembles d'opérations, ils sont utilisés dans les étapes 4 et 5.
- Les processus : ce sont des tâches exécutables sur les sites.

CHAPITRE III

DECOMPOSITION ET OPTIMISATIONS

Sommaire

3.1. INTRODUCTION	3.1
3.2. LOCALISATION DES RELATIONS GLOBALES	3.1
3.3. LES OPTIMISATIONS PAR PERMUTATIONS DES OPERATEURS	3.4
3.3.1. La Permutation des Opérateurs de PROJECTION et SELECTION .	3.4
3.3.2. La Permutation des Opérateurs de PRODUIT et UNION	3.6
3.3.3. La Permutation des Opérateurs de PRODUIT	3.7
3.4. ALLOCATION DES DONNEES REPARTIES	3.11
3.5. DISTRIBUTION DES OPERATIONS	3.16
3.5.1. Les Transactions Initiales	3.16
3.5.2. Les Transactions Intermédiaires	3.18
3.5.3. Distribution des Transactions	3.24
3.6. SELECTION D'UNE ARBORESCENCE DE TRANSACTIONS	3.27
3.7. GENERATION D'UN ENSEMBLE DE PROCESSUS REPARTIS	3.28

3.1. INTRODUCTION

Nous appelons décomposition la fonction qui transforme une requête globale en un ensemble de transactions, chacune exécutable sur un site, en produisant un ordre d'exécution sur ces transactions pour décrire la synchronisation logique de leur activation et en faisant les optimisations pour satisfaire nos objectifs de SGBDR. La décomposition de notre approche contient donc les fonctions suivantes : (voir la figure 5)

- Décomposition -

- (1) localisation des données globales en fonction des données locales et localisation des opérateurs monadiques et génération de tous les arbres syntaxiques possibles par les permutations des opérateurs,
- (2) allocation des copies réelles aux données locales de chaque arbre syntaxique en regroupant les sites concernés,
- (3) distribution des opérations de chaque arbre syntaxique aux processeurs répartis en vue de minimisation du temps de transmission des données,
- (4) sélection d'une arborescence optimale qui minimise le temps de transmission des données et génération d'un ensemble de processus répartis.

Nous allons présenter précisément toutes ces fonctions de décomposition dans les sections suivantes.

3.2. LOCALISATION DES RELATIONS GLOBALES

La première étape a pour but de remplacer toutes les relations globales par leur expression sous forme de relations locales. Cette étape peut être réalisée grâce à un tableau connue du système qui mémorise les cri-

tères de répartition de données. Dans le cas de notre exemple le tableau correspondrait à celui de la figure 6.

relations globales	type	relations locales correspondantes	critère de répartition
S	répartition horizontale	s1 s2 s3	SNO < 100 100 ≤ SNO < 1 000 1 000 ≤ SNO
Y	identité	y	
P	identité	p	

Figure 6 : le Tableau de la Répartition des Données

D'une façon générale si une relation globale S est répartie horizontalement on pourra écrire $S = \bigcup_{i=1}^n s_i$, où s1, s2, ..., sn désignant n relations locales et "+" l'opérateur d'union. A l'opposé, si S est répartie verticalement on pourra écrire $S = \prod_{i=1}^n s_i$, où les s_i sont des relations locales et "*" l'opérateur de produit.

Dans le cas de notre exemple on écrira $S = s1 + s2 + s3$ et on peut remarquer que chaque relation locale peut aussi s'exprimer à partir de S. En effet $s1 = S[SNO < 100]$, $s2 = S[100 \leq SNO < 1\ 000]$ et $s3 = S[SNO \geq 1\ 000]$ où "[expression]" dénote l'opération de sélection et "expression" une expression conditionnelle.

L'arbre syntaxique obtenu après la localisation est celui de la figure 7.

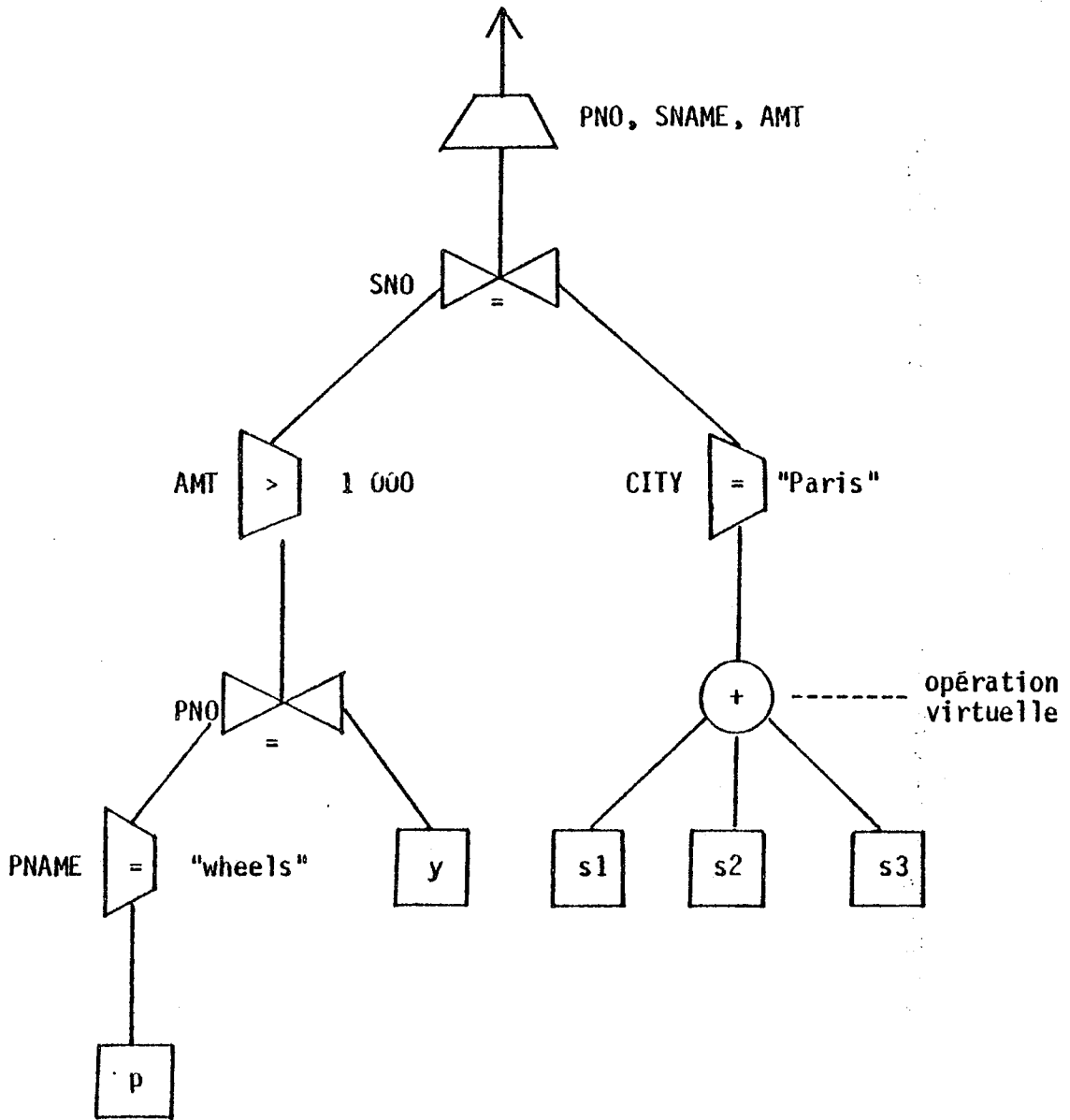


Figure 7 : Arbre Syntaxique après Localisation

Il consiste à remplacer chaque nom de relation globale par son expression syntaxique. Les opérateurs qui se substituent aux relations globales sont appelées pour l'étape suivante une opération virtuelle. Cette opération virtuelle correspond à un marquage qui sera utilisé dans l'étape suivante. Sur la figure 7 nous avons une seule opération virtuelle qui est une union.

3.3. LES OPTIMISATIONS PAR PERMUTATIONS DES OPERATEURS

Au cours de cette étape on va effectuer plusieurs transformations.

3.3.1. La permutation des opérateurs de projection et de sélection

La première transformation conduit à une arborescence unique et correspond à une tâche d'optimisation locale. Elle peut être obtenue de plusieurs façon par permutation dans l'arbre syntaxique, des opérateurs de projection et sélection. Chaque fois que cela est possible de telle sorte que ces opérateurs soient le plus près possible des feuilles de l'arbre.

Les règles de ce type de permutation ont été particulièrement étudiées par [7]. L'objectif de cette transformation correspond bien à une phase d'optimisation car elle vise à limiter le volume des données transférées au minimum. On peut aussi effectuer des transformation visant soit à simplifier des expressions logiques par dérivation dans les expressions de sélection [1, 2, 6, 7], par exemple $(A > B)$ et $(B=100) \rightarrow A > 100$, soit à éliminer des feuilles terminales dans la mesure où elles n'interviennent pas dans l'expression de la requête [2, 6, 7].

Dans notre exemple seule la propagation de l'opérateur de sélection peut être appliqué et conduit au schéma de la figure 8. C'est ainsi que les

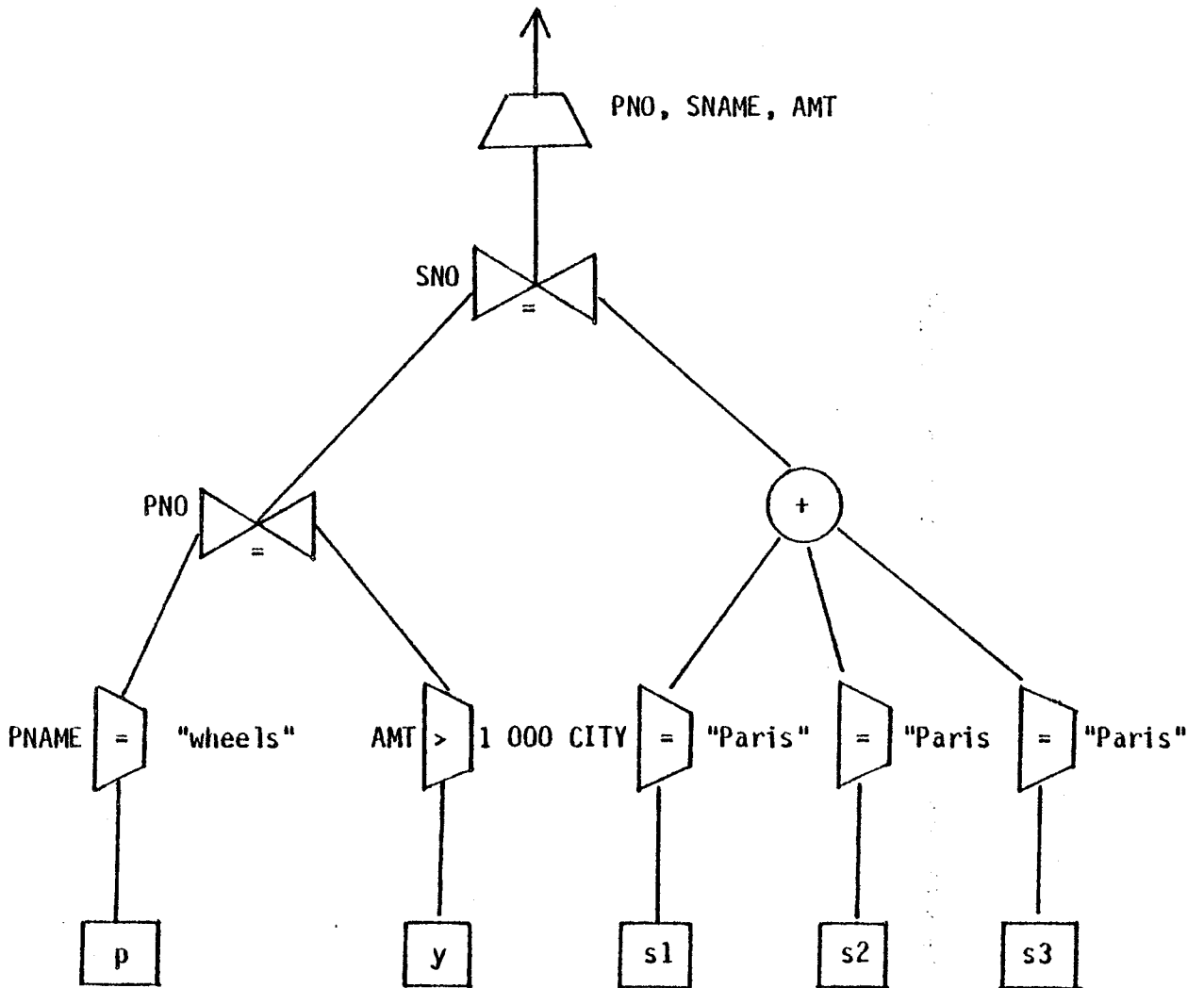


Figure 8 : Transformation de l'Arbre Syntaxique après Propagation des Opérations de Sélection

expressions de sélection $AMT > 1\ 000$ et $CITY = "Paris"$ sont descendues le plus bas possible dans l'arborescence.

3.3.2. La permutation des opérateurs de produit et union

La deuxième transformation que l'on effectuera au cours de cette étape va porter sur les opérations virtuelles relevées au cours de la première étape en appliquant deux stratégies : la stratégie de centralisation et la stratégie de distribution.

La stratégie de centralisation ne change rien à l'arbre syntaxique. Elle consiste simplement à dire que si une opération virtuelle était une union (respectivement un produit) l'opération réelle effectuée en ce noeud sera bien une union (respectivement un produit). La signification de cette stratégie est en fait de construire la relation résultante de cette opération en un site du réseau le plus tôt possible dans l'évaluation de la requête. On dit que c'est une stratégie qui transporte les données au traitement.

La stratégie de distribution consiste à appliquer la propriété de double distributivité des opérateurs de produit et d'union. En effet on peut montrer que :

$$R * \left(\bigcup_{i=1}^n S_i \right) = \bigcup_{i=1}^n (R * S_i)$$

$$R + \left(\bigcap_{i=1}^n S_i \right) = \bigcap_{i=1}^n (R + S_i) .$$

Cette propriété a comme effet de faire remonter l'opérateur d'union (respectivement de produit) le plus haut possible dans l'arbre syntaxique. Elle a comme effet secondaire de multiplier le nombre de branches de l'arbre

syntaxique, mais cet effet est bénéfique car cela va permettre d'une part d'accroître les possibilités d'exécution parallèle et de diminuer le flux de données sur les branches, qui ira en s'accroissant du bas vers le haut de l'arbre. On dit que c'est une stratégie qui distribue les traitements aux données.

La stratégie de centralisation a été mise en oeuvre dans les prototypes de SDD-1 [1] et POLYPHEME [7, 10], et est préconisée dans les modèles de CHU [7] et ADD [6]. L'application de la stratégie de centralisation ne modifie en rien l'arbre syntaxique de la figure 8, par contre l'application de la stratégie de distribution donne le schéma de la figure 9.

On peut illustrer cette transformation sur notre exemple par l'application de la propriété de double distributivité de la façon suivante :

$$p' \overset{PNO}{*} y' \overset{SNO}{*} (s1' + s2' + s3') = p' \overset{PNO}{*} y' \overset{SNO}{*} s1' + p' \overset{PNO}{*} y' \overset{SNO}{*} s2' + p' \overset{PNO}{*} y' \overset{SNO}{*} s3'$$

$$\begin{aligned} \text{où } p' &= p[\text{PNAME} = \text{"wheels"}], \\ y' &= y[\text{AMT} > 1000], \\ s1' &= s1[\text{CITY} = \text{"Paris"}], \\ s2' &= s2[\text{CITY} = \text{"Paris"}], \\ s3' &= s3[\text{CITY} = \text{"Paris"}]. \end{aligned}$$

A la fin de cette étape nous avons obtenu, à partir de la requête initiale, deux arbres différents. Tout ce qui sera fait au cours des étapes suivantes devra bien entendu s'appliquer à ces deux cas.

3.3.3. La permutation des opérateurs de produit

Dans les arbres syntaxiques, l'opération algébrique de produit joue un

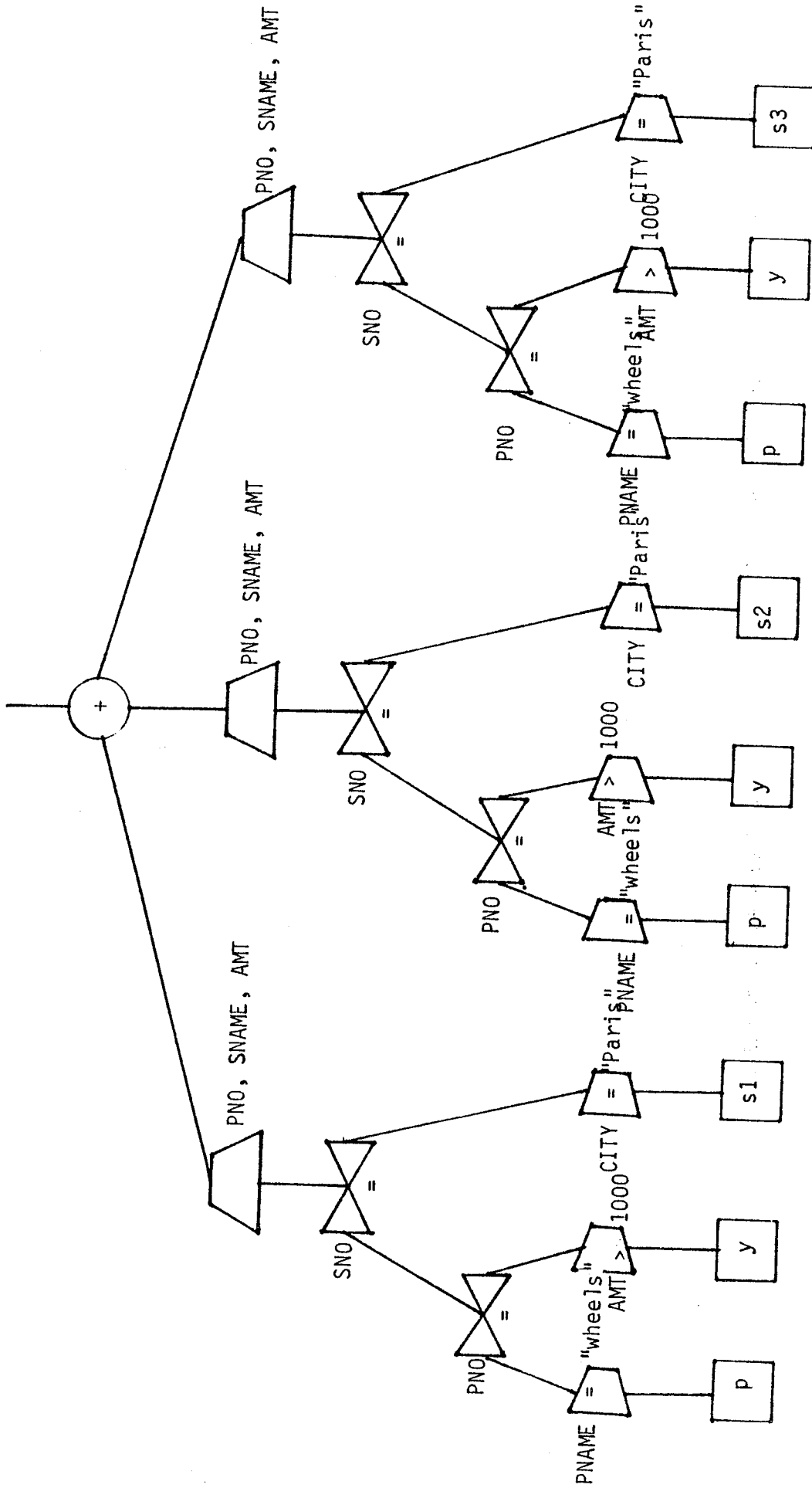


Figure 9 : Transformation de l'Arbre après l'Application de la Stratégie de Distribution

très grand rôle car c'est une opération qui par elle-même est coûteuse en temps et quand elle intervient plusieurs fois, l'ordre dans lequel on effectue ces produits peut avoir des effets néfastes sur les performances. Un exemple très simple suffit pour illustrer cela, considérons trois relations $R1(A, B)$, $R2(B, C)$ et $R3(C, D)$ et supposons que l'on fasse le produit de ces trois relations dans l'ordre :

$$(R1(A, B) * R2(B, C)) * R3(C, D).$$

Comme le produit possède la propriété de commutativité et d'associativité on peut écrire

$$(R1(A, B) * R3(C, D)) * R2(B, C).$$

L'exécution du produit de $R1$ par $R3$ est équivalent en fait au produit cartésien des relations $R1$ et $R3$, ce qui est le plus mauvais choix possible du point de vue du temps d'exécution.

L'objectif de cette phase va donc être de voir quelles permutations on peut effectuer parmi les opérations.

Par exemple à partir de l'expression

$$p' \overset{PNO}{*} y' \overset{SNO}{*} (s1' + s2' + s3'),$$

on peut écrire

$$((s1' + s2' + s3') \overset{SNO}{*} y') \overset{PNO}{*} p'.$$

Cette dernière expression correspond à un autre arbre syntaxique comme le montre de la figure 10 obtenue à partir de l'arbre de la figure 8. Ce type de transformation devra bien entendu s'appliquer aussi à l'arbre syntaxique de la figure 9.

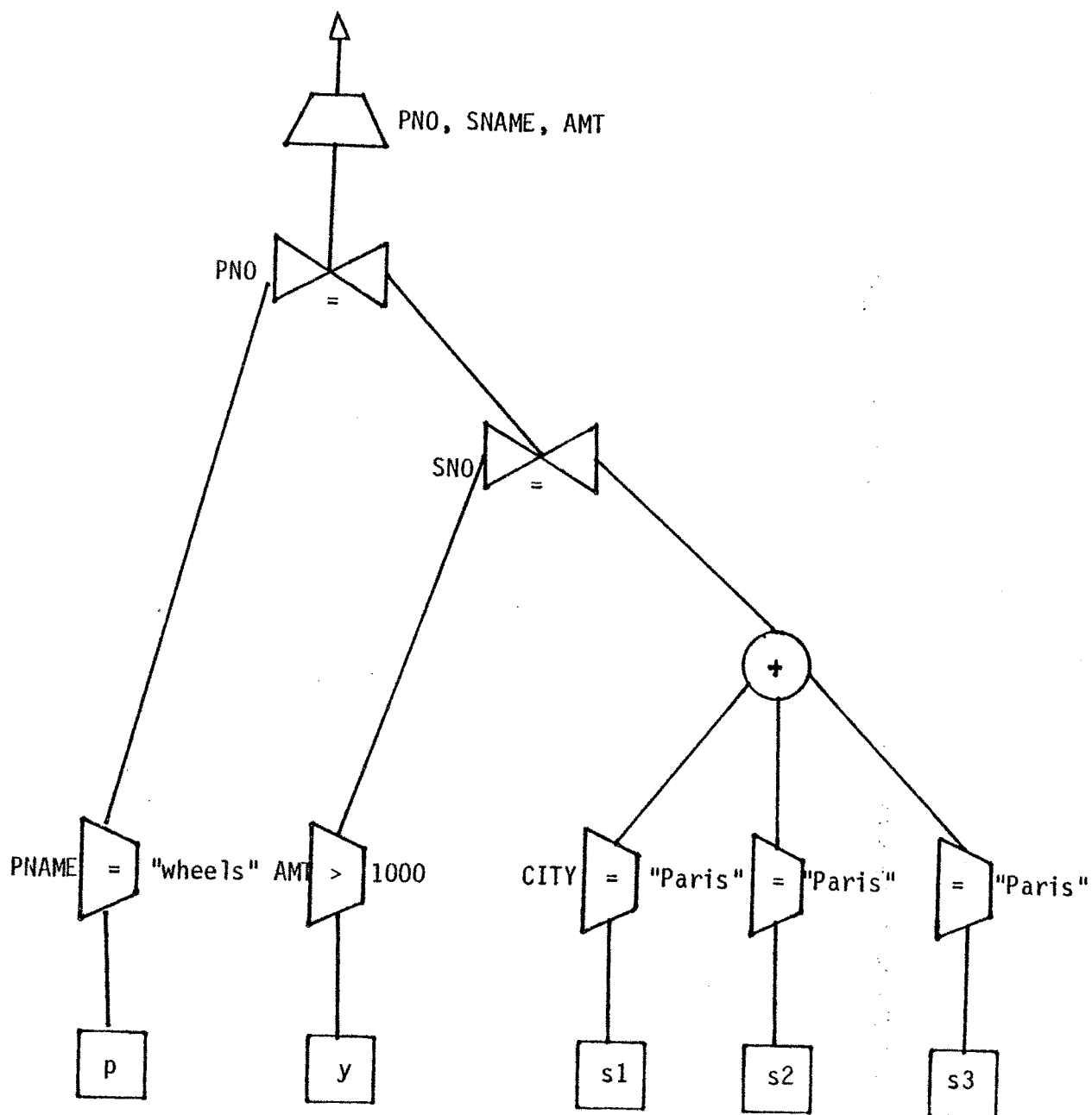


Figure 10 : Transformation de l'Arbre après la Permutation de Produit à partir de la Figure 8

Ce type d'optimisation a été mis en oeuvre dans le système R [5] où les permutations possibles ne peuvent pas impliquer toutes les possibilités, et il est préconisé dans le modèle de CHU [7].

A la fin de cette étape nous pouvons obtenir plusieurs autres arbres syntaxiques pour notre exemple à partir des deux arbres syntaxiques de la figure 8 et de la figure 9. Tout ce qui sera fait au cours des étapes suivantes devra bien entendu s'appliquer à tous ces cas.

3.4. ALLOCATION DES DONNEES REPARTIES

L'objectif de cette étape est de décider pour chaque relation locale la copie actuelle qui sera utilisée dans l'exécution de la requête. L'existence de copies multiples sur des sites différents pour la même relation locale nous amène à considérer un ensemble de combinaisons possibles. On appellera domaine d'exécution d'une requête un ensemble de sites qui permet l'exécution de la requête. Une stratégie de matérialisation arbitraire a été mise en oeuvre dans le prototype de SDD-1 [1] où le domaine peut être l'une parmi plusieurs matérialisations possibles qui ne se chevauchent pas pour que la disponibilité des données soit améliorée. On va s'intéresser à rechercher un domaine optimal en prenant le temps global de transmission entre les sites du domaine comme critère. Pour cela on va définir la distance logique entre deux sites i et j , notée $DL(i, j)$.

La distance logique (DL) est définie comme le temps de transmission d'un paquet de données entre deux sites en supposant qu'il y ait un seul trajet entre deux sites (on peut le considérer comme le trajet le plus court). On appellera parcours les couples de deux noeuds adjacents, par exemple, les IMP ("Inter Message Processor") adjacents dans le réseau ALPHA, qui

se trouvent sur le même trajet.

Le temps de transmission d'un paquet de données pour un parcours, noté par P_t , peut être défini comme la suivante dans un réseau partiellement maillé, par exemple, notre environnement du réseau ou réseau "ALPHA", etc.

[13]

$$P_t = P_e + m \times L_e + K_t + Q_t$$

où P_e signifie le temps d'encodage d'un paquet représenté par P/c où P signifie le volume d'un paquet et c signifie le taux de transmission,

m signifie la vitesse de propagation de la ligne,

L_e signifie la longueur de la ligne du parcours,

K_t signifie le temps de traitement au noeud,

Q_t signifie le temps d'attente au noeud.

Puisqu'un trajet entre deux sites peut se composer de plusieurs parcours, on peut donc représenter la distance logique (DL) entre deux sites comme la suivante [4] :

$$DL = \sum_{\text{parcours}} P_t .$$

Mais le temps de propagation et de traitement au noeud sont négligeables par rapport aux autres [13]. Le temps d'attente sera aussi négligeable si le trafic de transmission est bien réparti dans le réseau [4]. Dans ce cas la distance logique peut être considérée comme le temps d'encodage multiplié par le nombre $n-1$ si n est le nombre de noeuds sur le trajet entre deux sites.

Pour un domaine D de sites, on appellera la surface logique du domaine notée $SL(D)$, la somme de toutes les distances logiques entre des sites appartenant à D , c'est à dire :

$$SL(D) = \sum_{i,j \in D} DL(i, j)$$

Le domaine optimal D^* sera celui dont la surface logique sera la plus petite. Pour rechercher le domaine optimal on doit disposer du catalogue des copies et de la matrice des distances logiques inter-sites. Dans le cas de notre exemple ces informations nous sont fournies par la figure 11. Ici nous avons pris comme distance logique une formule simplifiée comme étant égale à $n-1$, si n est le nombre de noeuds dans le trajet optimal entre deux sites. (voir figure 3 pour le schéma du réseau)

Les domaines $D1 = \{1, 2, 3\}$ et $D2 = \{1, 2, 3, 6\}$ sont entre autres des domaines d'exécution de la requête, comme $D1 \subset D2$

$$SL(D1) < SL(D2)$$

Comme dans notre exemple $D1$ est le plus petit domaine parmi tous ceux possibles, on peut conclure que $D1$ est le domaine optimal et sa surface logique $SL(D1) = 8$.

Cet exemple nous suggère un algorithme pour chercher le domaine optimal sans explorer toutes les combinaisons, il suffit d'énumérer les domaines minimaux ("Un domaine est minimal si la suppression d'un seul de ces éléments ne permet plus d'exécuter la requête"), puis d'évaluer pour chaque domaine minimal la surface logique. Dans notre exemple $D1$ est le seul domaine minimal.

En effet tous les arbres syntaxiques engendrés par les transformations

relations locales	localisation des copies
s1	site 1
s2	site 2
s3	site 3
y	site 3, site 6
p	site 1

Catalogue des Copies

site site	1	2	3	4	5	6	7
1	0	2	1	2	3	3	3
2	2	0	1	2	3	3	3
3	1	1	0	1	2	2	2
4	2	2	1	0	1	1	1
5	3	3	2	1	0	2	2
6	3	3	2	1	2	0	2
7	3	3	2	1	2	2	0

les Distances Logiques entres des Sites

Figure 11 : Catalogue des Copies et Distance Logique
entre les Sites

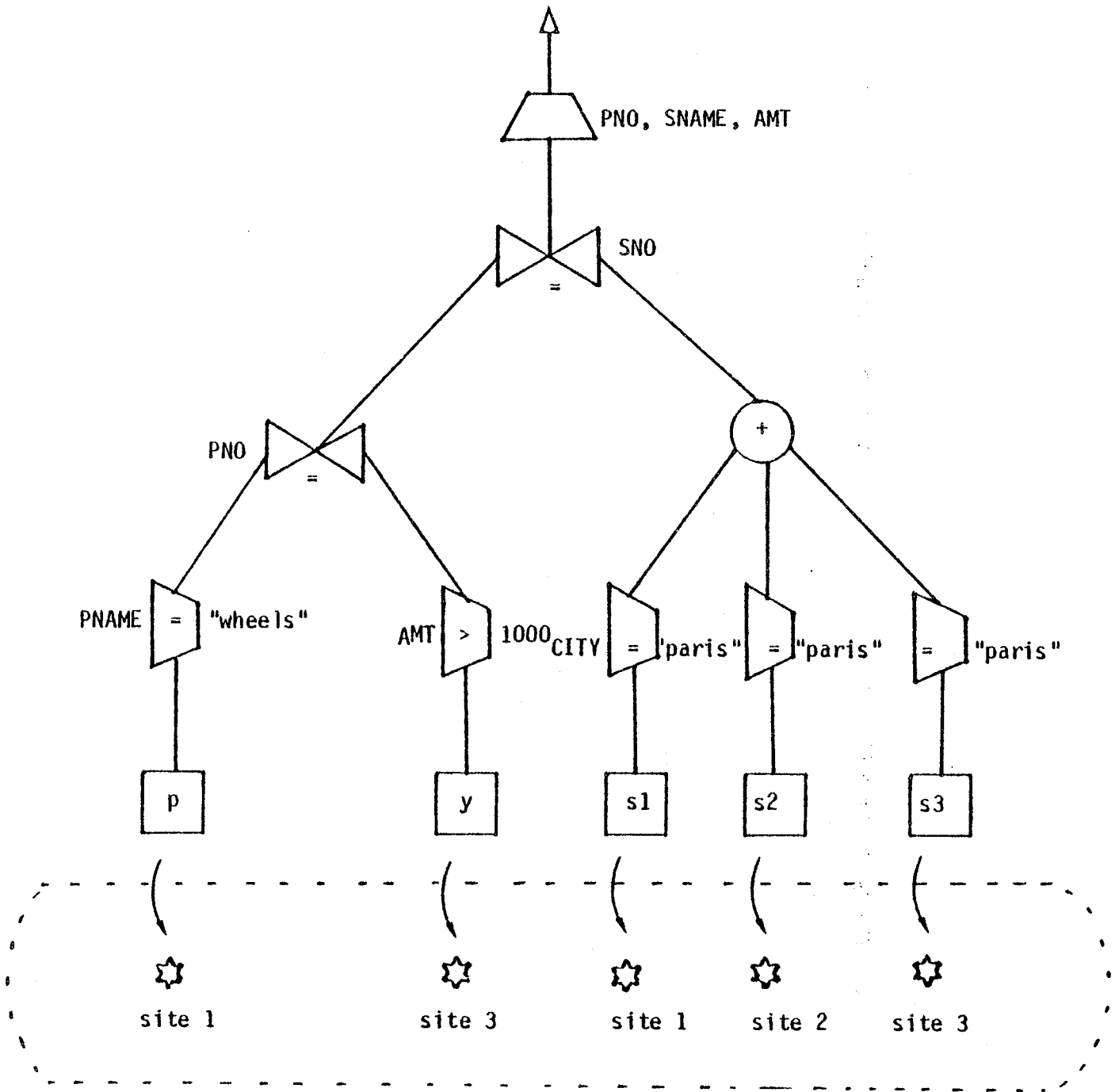


Figure 12 : Allocation des Copies

précédentes auront tous le même domaine optimal D1. Par exemple, l'arbre syntaxique de la figure 8 aura le domaine optimal représenté sur la figure 12.

3.5. DISTRIBUTION DES OPERATIONS

Le point de départ de cette étape correspond à un arbre syntaxique dont les relations locales ont été allouées à un site. Si l'étape précédente a produit plusieurs arbres syntaxiques localisés, l'étape que nous allons appliquer maintenant devra être répétée autant de fois que nécessaire. Cette étape a pour but de distribuer de manière optimale des opérations en des sites du réseau. Les problèmes liés à ce type d'optimisation dépendront de plusieurs paramètres, par exemple, du volume des données transférées, de la nature des opérations et des paramètres du réseau. Notre solution consiste à effectuer plusieurs traitements :

- recherche des transactions initiales,
- énumération des transactions intermédiaires par partitionnement,
- distribution optimale des transactions,
- sélection de l'arbre syntaxique optimal.

3.5.1. Les transactions initiales

Les feuilles de l'arborescence correspondent aux relations locales et chaque relation locale a été allouée à un site. Nous dirons qu'une opération est localisée lorsqu'on a pu affecter un site pour exécuter cette opération. Une transaction initiale est un sous arbre de l'arbre syntaxique tel que toutes les opérations peuvent s'exécuter sur le même site où se trouvent localisées les relations locales du sous arbre. La recherche

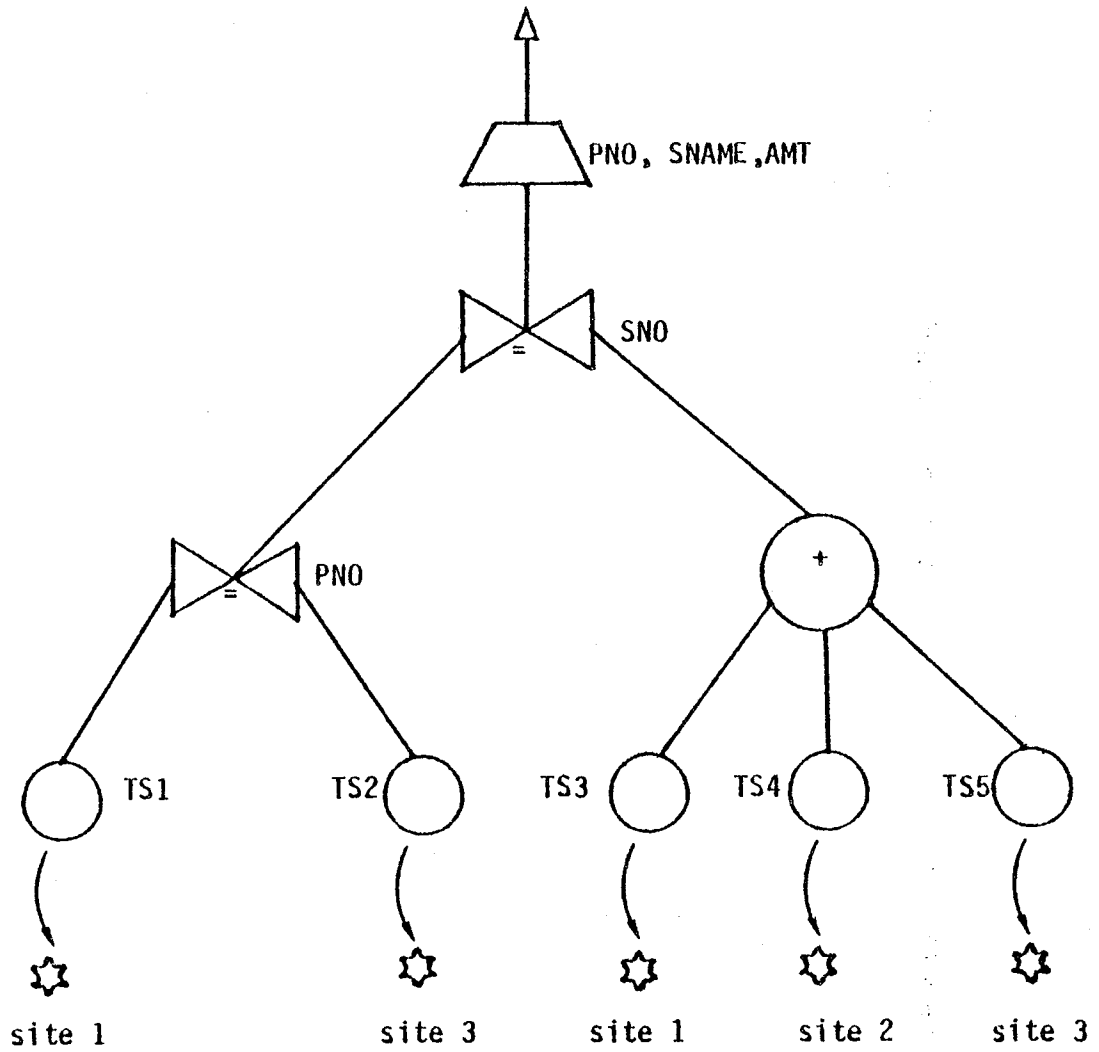


Figure 13 : Génération d'un Ensemble de Transactions
Initiales et l'Arbre Réduit

des transactions initiales se fait par exploration de l'arbre syntaxique localisé à partir des feuilles et en remontant vers la racine. Une opération pourra être localisée si tous les descendants de cette opération sont localisée sur le même site, auquel cas cette opération aura lieu aussi sur ce site. Dans le cas où l'opération ne peut être localisée parce que les descendants sont localisés sur des sites différents, on l'appellera une opération intermédiaire. Les opérations intermédiaires donneront naissance à l'ensemble des transactions intermédiaires.

Le schéma de la figure 13 donne les transactions initiales et les opérations intermédiaires de l'arbre syntaxique de la figure 12. Les feuilles terminales (noeud rond) représentent les transactions initiales qui correspondent aux expressions suivantes :

TS1 : p[PNAME = "wheels"] au site 1

TS2 : y[AMT > 1 000] au site 3

TS3 : s1[CITY = "Paris"] au site 1

TS4 : s2[CITY = "Paris"] au site 2

TS5 : s3[CITY = "Paris"] au site 3

Ce graphe réduit va servir à l'étude de partitionnement des opérations intermédiaires pour générer les transactions intermédiaires.

Cet ensemble de transactions initiales sera identique dans tous les arbres syntaxiques engendrés par les transformations précédentes.

3.5.2. Les transactions intermédiaires

Nous avons vu que seules les transactions initiales ont un site d'exécution par localisation des opérations initiales, les opérations interméd-

diaires ne sont pas encore localisées. Cette latitude va nous permettre d'étudier les effets du regroupement de plusieurs opérations intermédiaires en une seule transaction intermédiaire. Une transaction intermédiaire sera non seulement vue comme un ensemble d'opérations intermédiaires mais comme une séquence ordonnée d'opérations. La signification pratique de cette notion de regroupement est simple, si par exemple, on décide de regrouper les opérations intermédiaires de la figure 13, cela signifie que l'on construit premièrement une seule transaction intermédiaire avec toutes les opérations intermédiaires et après avoir choisi pour cette transaction intermédiaire un site d'exécution on aura une exécution centralisée où les résultats des transactions initiales seront envoyés. L'objectif de cette phase est donc de déterminer, à partir d'une seule transaction intermédiaire, toutes les transactions intermédiaires possibles par regroupement des opérations intermédiaires et de construire tous les arbres possibles regroupant un ensemble de transactions initiales et intermédiaires que nous appellerons l'arborescence de transactions.

Ce problème nous suggère un algorithme pour énumérer toutes les arborescences de transactions possibles sans énumérer les duplications. Il suffit de diviser successivement les expressions algébriques des transactions intermédiaires en deux sous expressions en gardant l'ordre d'exécution des opérateurs jusqu'à ce que toutes les expressions ou sous-expressions ne soient pas divisibles (une expression ou sous-expression est divisible si et seulement si elle consiste en au moins deux opérations bi-relationnelles). En effet chaque division constitue un séquence de transactions intermédiaires qui est actuellement un arborescence de transaction.

Par exemple, pour l'arbre de la figure 13 on peut construire un séquence

d'une seule transaction intermédiaire notée T_{i11} dont l'expression algébrique est ;

$$T_{i11} : \overline{TS1} \overset{PNO}{*} \overline{TS2} \overset{SNO}{*} (\overline{TS3} + \overline{TS4} + \overline{TS5}) [PNO, SNAME, AMT]$$

où $\overline{TS1}$, $\overline{TS2}$, ..., $\overline{TS5}$ signifient les relations obtenues par les transactions initiales $TS1$, $TS2$, ..., $TS5$.

L'expression de T_{i11} peut être divisée par une seule manière donc on ne peut construire qu'une séquence de deux transactions intermédiaires :

$$T_{i21} : \overline{T_{i21}} = \overline{TS1} \overset{PNO}{*} \overline{TS2}$$

$$T_{i22} : \overline{T_{i22}} = \overline{T_{i21}} \overset{SNO}{*} (\overline{TS3} + \overline{TS4} + \overline{TS5}) [PNO, SNAME, AMT].$$

L'expression de T_{i22} est encore divisible donc on peut construire un autre séquence de trois transactions intermédiaires :

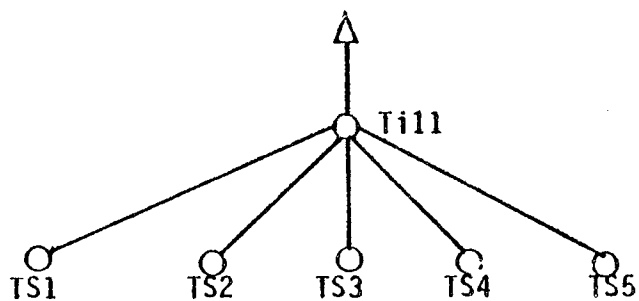
$$T_{i31} : \overline{T_{i31}} = \overline{TS1} \overset{PNO}{*} \overline{TS2}$$

$$T_{i32} : \overline{T_{i32}} = (\overline{TS3} + \overline{TS4} + \overline{TS5})$$

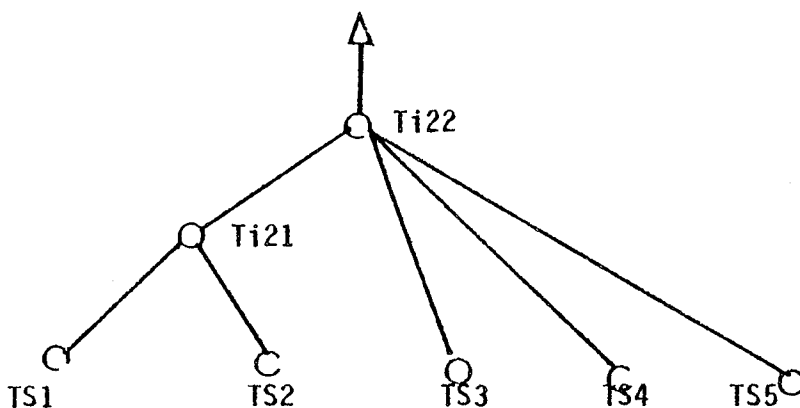
$$T_{i33} : \overline{T_{i33}} = \overline{T_{i31}} \overset{SNO}{*} \overline{T_{i32}} [PNO, SNAME, AMT].$$

Dans ce dernier ensemble de transactions intermédiaires, chaque sous-expression n'est plus divisible. La transaction T_{i33} se compose actuellement encore de deux opérateurs mais nous pensons qu'elle n'est plus divisible puisqu'il n'y a qu'un opérateur bi-relationnelle, " $\overset{SNO}{*}$ ". Chaque séquence de transactions peut être représenté aussi par un arbre, comme le montre de la figure 14, que nous appelons l'arborescence de transactions.

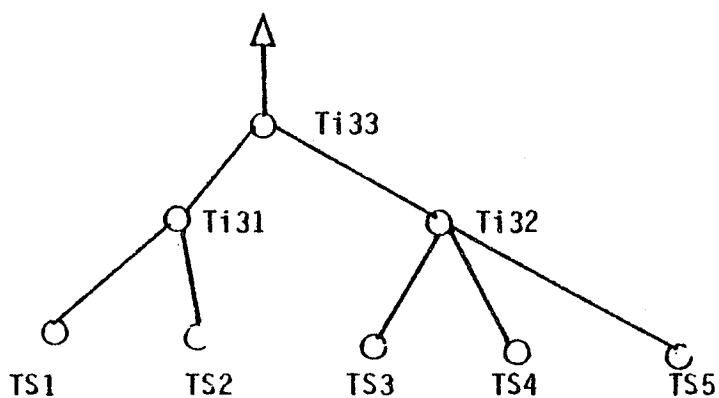
De la même façon nous pouvons construire quatre autres arborescences de transactions à partir de la figure 10 qui était aussi dérivée de la figure 8 par la transformation, la permutation de produit (voir section 3.3.3.)



(a) avec une seule transaction intermédiaire



(b) avec deux transactions intermédiaires



(c) avec trois transaction intermédiaires

Figure 14 : Génération d'un Ensemble d'Arborescences de Transactions
à partir de l'Arbre Syntaxique de la Figure 8

Ceci est représenté à la figure 15. Dans la figure 15, l'arborescence (a) est en fait la même structure que l'arborescence (a) de la figure 14, mais l'ordre d'exécution des opérateurs était changé comme :

$$T_{i11} : \overline{T_{i11}} = (\overline{TS3} + \overline{TS4} + \overline{TS5})^{SNO} \overline{TS2}^{PNO} \overline{TS1} [PNO, SNAME, AMT]$$

où $\overline{TS1}$, $\overline{TS2}$, ..., $\overline{TS5}$ sont identiques comme la figure 14,

L'expression T_{i11} de ci-dessus peut être divisée en deux sous-expressions de deux manières différentes en gardant l'ordre d'exécution des opérateurs de T_{i11} :

- premièrement on peut construire l'arborescence (b) de la figure 15

où

$$T_{i21} : \overline{T_{i21}} = (\overline{TS3} + \overline{TS4} + \overline{TS5})$$

$$T_{i22} : \overline{T_{i22}} = \overline{T_{i21}}^{SNO} \overline{TS2}^{PNO} \overline{TS1} [PNO, SNAME, AMT],$$

- deuxièmement on peut construire l'arborescence (c) de la figure 15

où

$$T_{i31} : \overline{T_{i31}} = (\overline{TS3} + \overline{TS4} + \overline{TS5})^{SNO} \overline{TS2},$$

$$T_{i32} : \overline{T_{i32}} = \overline{T_{i31}}^{PNO} \overline{TS1} [PNO, SNAME, AMT].$$

Les expressions T_{i22} et T_{i31} sont encore divisibles et l'expression T_{i11} peut être divisée aux trois transactions intermédiaires comme :

$$T_{i41} : \overline{T_{i41}} = (\overline{TS3} + \overline{TS4} + \overline{TS5}),$$

$$T_{i42} : \overline{T_{i42}} = \overline{T_{i41}}^{SNO} \overline{TS2},$$

$$T_{i43} : \overline{T_{i43}} = \overline{T_{i42}}^{PNO} \overline{TS1} [PNO, SNAME, AMT].$$

Ceci est représenté par l'arborescence (d) de la figure 15.

En conclusion on peut dire qu'à partir d'une requête externe, analogue à celle de notre exemple, on peut engendrer six arborescences différentes

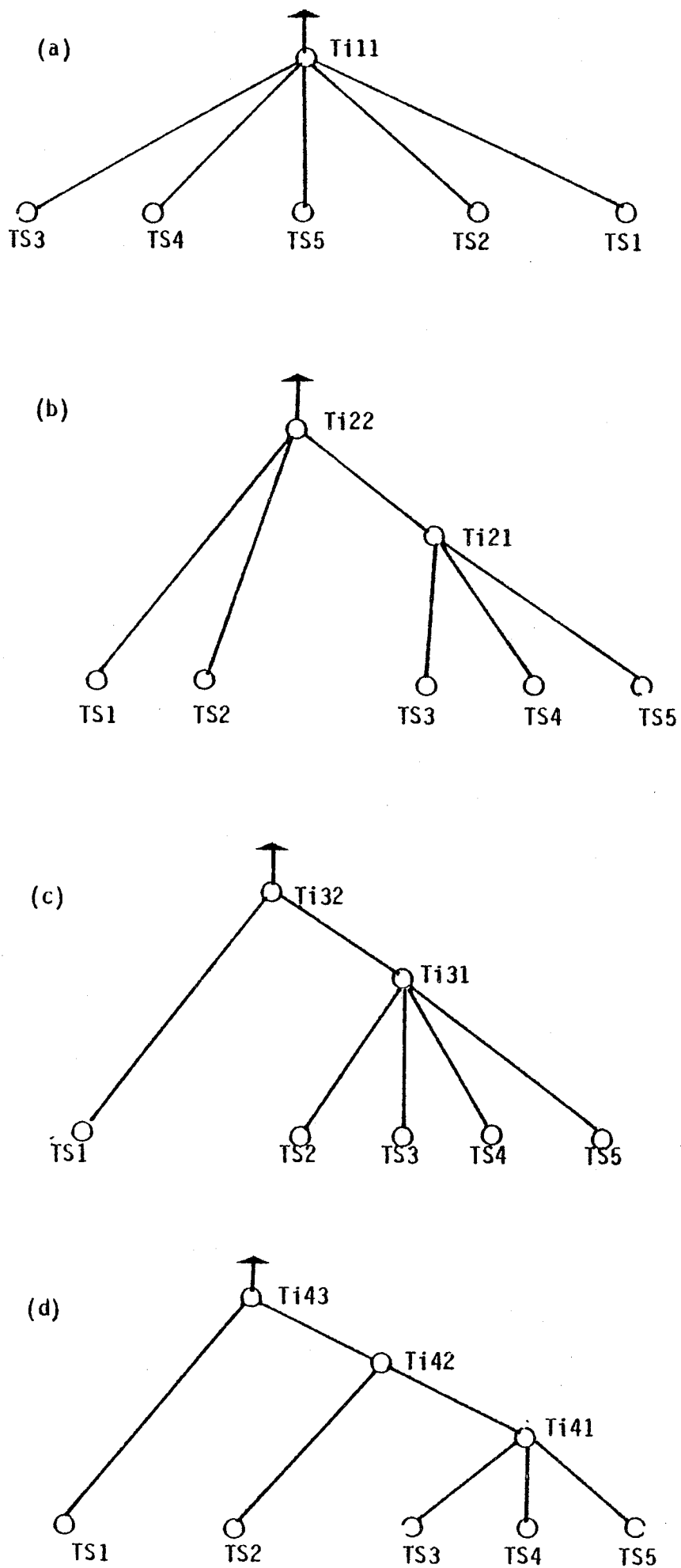


Figure 15 : Génération d'un Ensemble des Arborescences de Transactions à partir de l'Arbre Syntaxique de la figure 10

de transactions (figure 14 et 15) après avoir permuté les opérateurs de produit et d'union à l'aide de la stratégie de centralisation (voir section 3.3.2). Elle peut également engendrer les autres après permutation à l'aide de la stratégie de distribution.

On peut remarquer que grâce à notre procédé de génération des transactions, la permutation des opérateurs de produit et d'union et le partitionnement des opérateurs sur un arbre syntaxique ne produisent pas de duplications au niveau de l'arborescence de transactions. Ce type de considération était préconisé dans le modèle de CHU [3] pour l'optimisation, mais la génération d'un ensemble d'arborescences de transactions, appelée "query opération graphs", était résolu par une manière préjudiciable qui génère des duplications.

3.5.3. Distribution des transactions et évaluation

Cette étape consiste à choisir pour chaque transaction d'une arborescence de transactions un site d'exécution, puis à évaluer globalement le coût d'exécution de l'arborescence afin de sélectionner parmi tous les arborescences de transactions énumérées à partir d'une requête externe, l'arborescence optimale de transactions qui conduira à un minimal coût de transmission. Cette phase d'évaluation ne peut être faite que si on connaît les volumes des données pour chaque relation locale avant et après certaines opérations. Il existe différentes solutions pour faire cette estimation des volumes de données [1, 2, 3] qui dépendent de la nature de l'opération algébrique. Nous supposons ici que cette estimation a été effectuée.

La stratégie de distribution consiste à choisir comme site d'exécution pour chaque transaction le site de l'un de ses fils dans l'arborescence.

Il est évident que les transactions initiales se distribuent aux sites de leur données. C'est donc seulement pour les transactions intermédiaires que le problème se pose. C'est le coût de transfert de l'information qui va être le critère d'évaluation pour le choix. Dans un réseau complètement inter-connecté, le choix peut être fait indépendamment du choix des autres transactions intermédiaires car dans ce cas les coûts de transmissions d'un site sélectionné à tous les autres sites peuvent être supposés identiques. Nous appellerons ce type de stratégie distribution absolue qui était utilisé dans la plupart des approches [1, 2, 3]. Par contre dans le réseau distribué analogue à celui de notre exemple les choix doivent être fait en tenant compte des choix précédents et prochains car les coûts de transmissions d'un site sélectionné aux autres sites sont différents, dans ce cas le choix des sites pour toutes les transactions intermédiaires d'une arborescence sera fait d'un seul coup après avoir estimé le coût total des transmissions pour toutes les distributions possibles d'une arborescence. Nous appellerons ce type de stratégie distribution relative.

Par exemple la figure 16 est la distribution optimale de l'arborescence de la figure 15-b, si on suppose que les volumes de données de chaque transactions sont estimées à :

$$\overline{TS1} = 300, \overline{TS2} = 500, \overline{TS3} = 60, \overline{TS4} = 100, \overline{TS5} = 50, \overline{Ti21} = 210, \overline{Ti22} = 10.$$

Chaque transaction intermédiaire peut se distribuer sur l'un des sites candidats de la façon suivante :

- candidats pour $Ti21 = \{\text{site 1, site 2, site 3}\}$
- candidats pour $Ti22 = \{\text{site 1, site 3, site sélectionné pour } Ti21\}$.

On peut donc considérer neuf différentes distributions pour cette arborescence dont sept sont différentes. Après avoir évalué les sept différentes

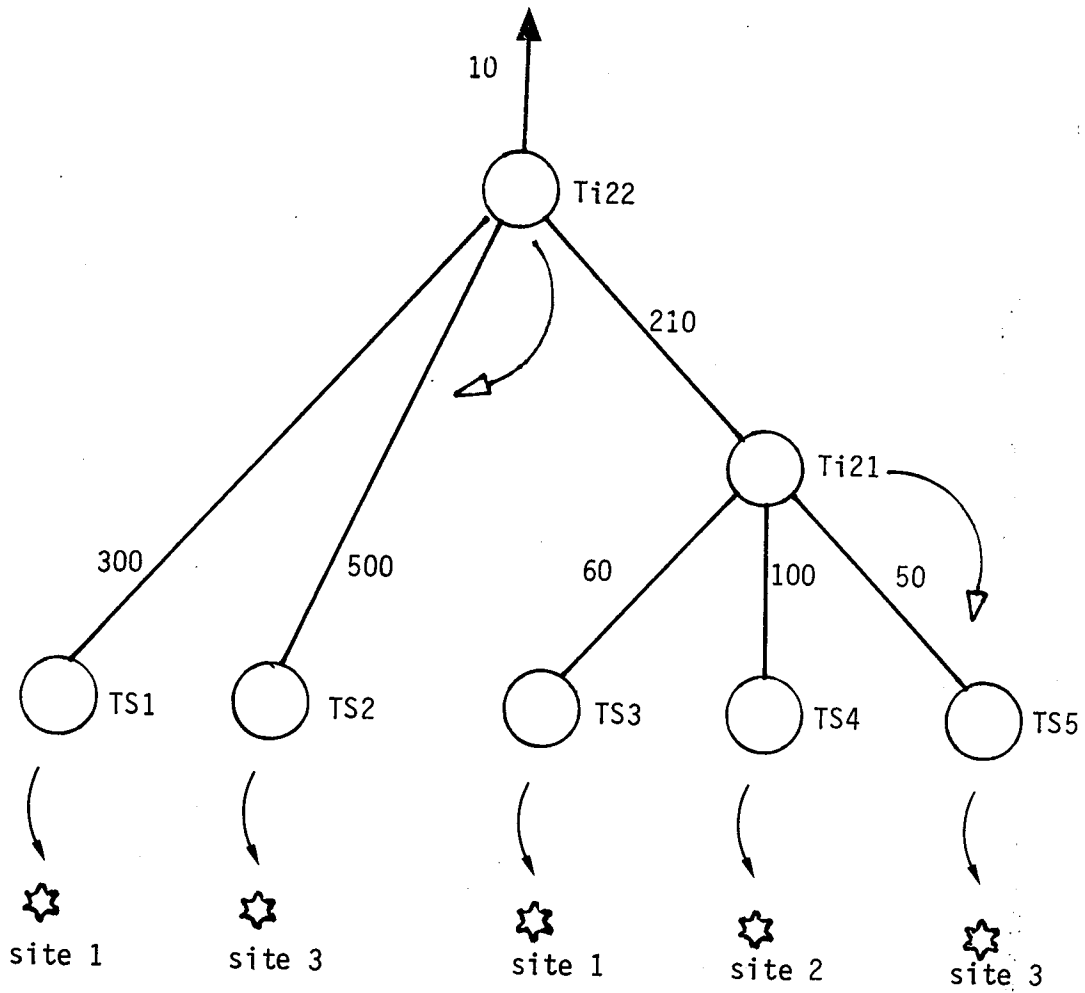


Figure 16 : la Distribution Optimale d'une Arborescence
de Transactions de la Figure 15-b

distributions possibles on peut remarquer que la solution optimale est atteinte, dans ce cas, avec une stratégie centralisée où toutes les transactions intermédiaires sont regroupées sur le site 3. Le coût de transmission de cette solution en utilisant les distances logiques de la figure 11 sera :

$$(60 \times 1 + 100 \times 1) + (300 \times 1 + 210 \times 0) = 460.$$

On peut remarquer que si la stratégie de distribution de cet exemple était la distribution absolue, la transaction T_{i21} de la figure 16 serait allouée au site 2. Celle-là n'est pas indubitablement la distribution optimale dans un réseau du type partiellement maillé comme nous l'avons vu dans cet exemple.

On peut aussi remarquer que la distribution optimale minimisant le coût de transmission peut porter préjudice à l'exécution parallèle, par exemple, dans la figure 16, les transactions TS_2 et TS_5 ne peuvent pas s'exécuter parallèlement si chaque site se compose d'un seul processeur, car elles étaient allouées au même site bien qu'il y ait la possibilité d'exécution parallèle de ces deux transactions.

3.6. SELECTION D'UNE ARBORESCENCE DE TRANSACTIONS

La distribution optimale était faite pour une arborescence de transactions. Nous avons vu qu'un ensemble d'arborescences de transactions peut être engendré par l'optimisation globale, permutation et partitionnement des opérations. L'objectif de ce traitement sera donc de sélectionner l'arborescence optimale de transactions. C'est le coût total de transmission d'une arborescence qui va être le critère de sélection. Le coût total de chaque arborescence était estimé pour la distribution optimale,

il suffit donc de comparer le coût présent et le coût minimal des arborescences déjà évaluées. En fin une arborescence de transactions correspondant au coût minimal sera sélectionnée.

3.7. GENERATION D'UN ENSEMBLE DE PROCESSUS REPARTIS

L'objectif de cette étape est de générer un ensemble de processus répartis à partir de l'arborescence de transactions optimale sélectionnées à l'étape précédente.

Pour décrire les processus répartis, nous allons définir un langage de type procédural qui se compose d'un ensemble de primitives. Le fonctionnement de ces primitives est récapitulé brièvement dans la figure 17.

Supposons que c'est l'arborescence de la figure 15-b qui est celle que nous avons retenue comme optimale pour la requête de notre exemple. Après la distribution optimale de cette arborescence (figure 16), nous pouvons générer un ensemble de processus répartis qui sont indiqués dans la figure 18. On peut voir que chaque processus est complet, donc tous les processus peuvent être distribués à la fois aux sites concernés. Premièrement les processus initiaux, notés PS1, PS2, ..., PS5 peuvent être exécutés car leurs conditions permettent l'exécution directe. Après avoir terminé l'exécution de ces transactions chaque processus transporte ses résultats aux sites distingués par la primitive "TRANSFERT", et puis signale la fin aux sites distingués par les variables de synchronisation pour l'activation automatique des processus intermédiaires. Chaque processus sera activé par le processeur de site lorsque les variables de la "CONDITION" primitive sont toutes satisfaites. Le processus final transportera les résultats au site original de la requête externe.

Primitives	Fonctionnement
DEBUT-FIN	Primitives qui signalent le début et la fin d'un processus ou d'une procédure.
VARIABLE GLOBALE	déclaration de variables globales de synchronisation entre processus, ce sont en fait des variables booléennes.
I-RELATION O-RELATION	déclaration de nom de variable locale de type tableau permettant le stockage temporaire des résultats reçus (I-RELATION) et des résultats à transmettre (O-RELATION).
CONDITION	condition de synchronisation exprimée à l'aide des variables globales pour qu'un processus puisse continué à s'exécuter. C'est en fait l'expression booléenne.
EXECUTE	demande de l'exécution des transactions au processeur d'un site sous forme d'un appel procédural.
TRANSFERT	demande d'une transmission des données intermédiaires (O-RELATION) à un site désigné (I-RELATION).
RESET	signalisation de fin des transmissions au site désigné par les variables globales.
INITIAL	les variables globales peuvent être initialisées à la valeur "NON" et les tableaux peuvent être initialisés à la valeur zéro.

Figure 17 : les Primitives

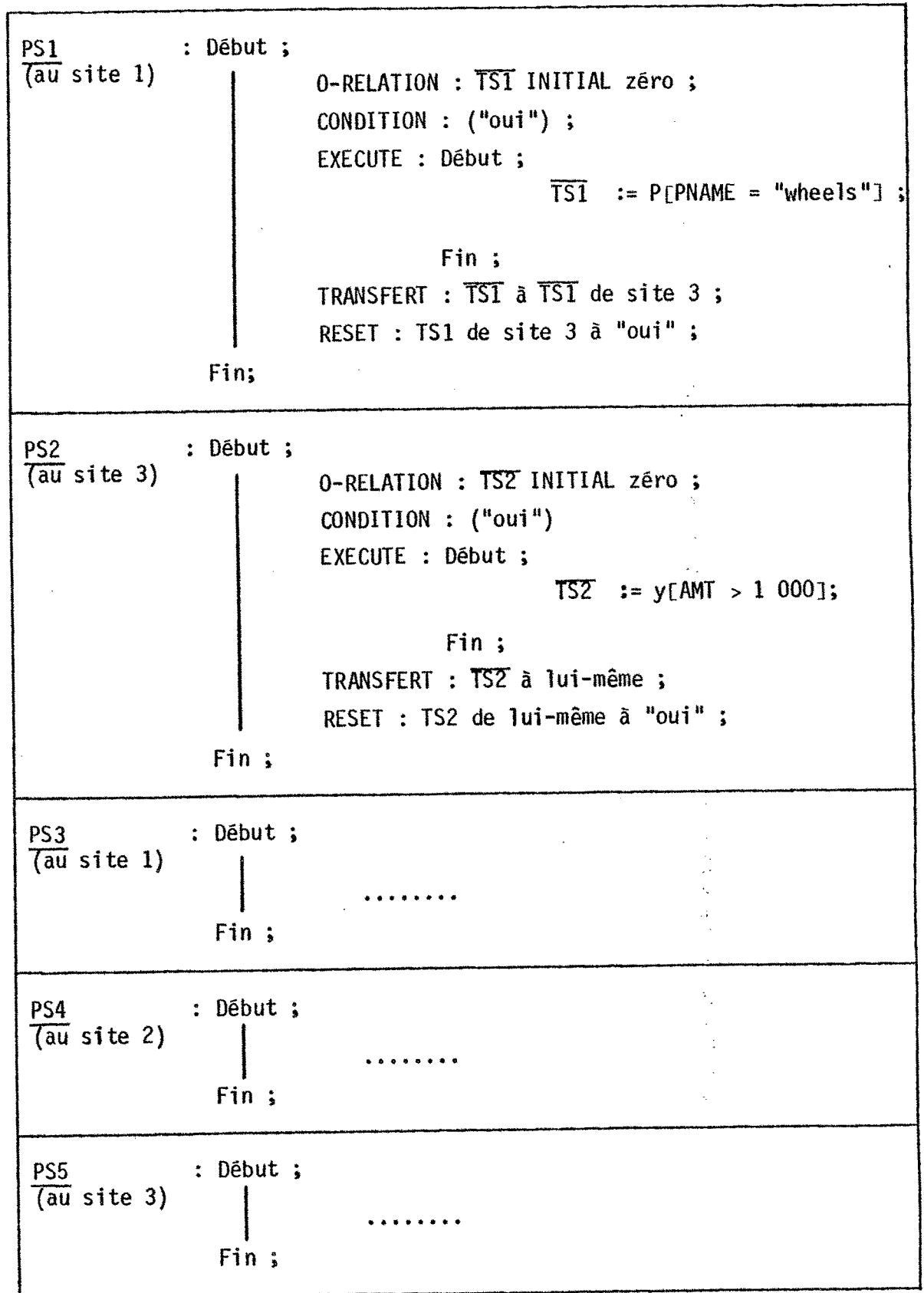


Figure 18-a : les Processus Initiaux

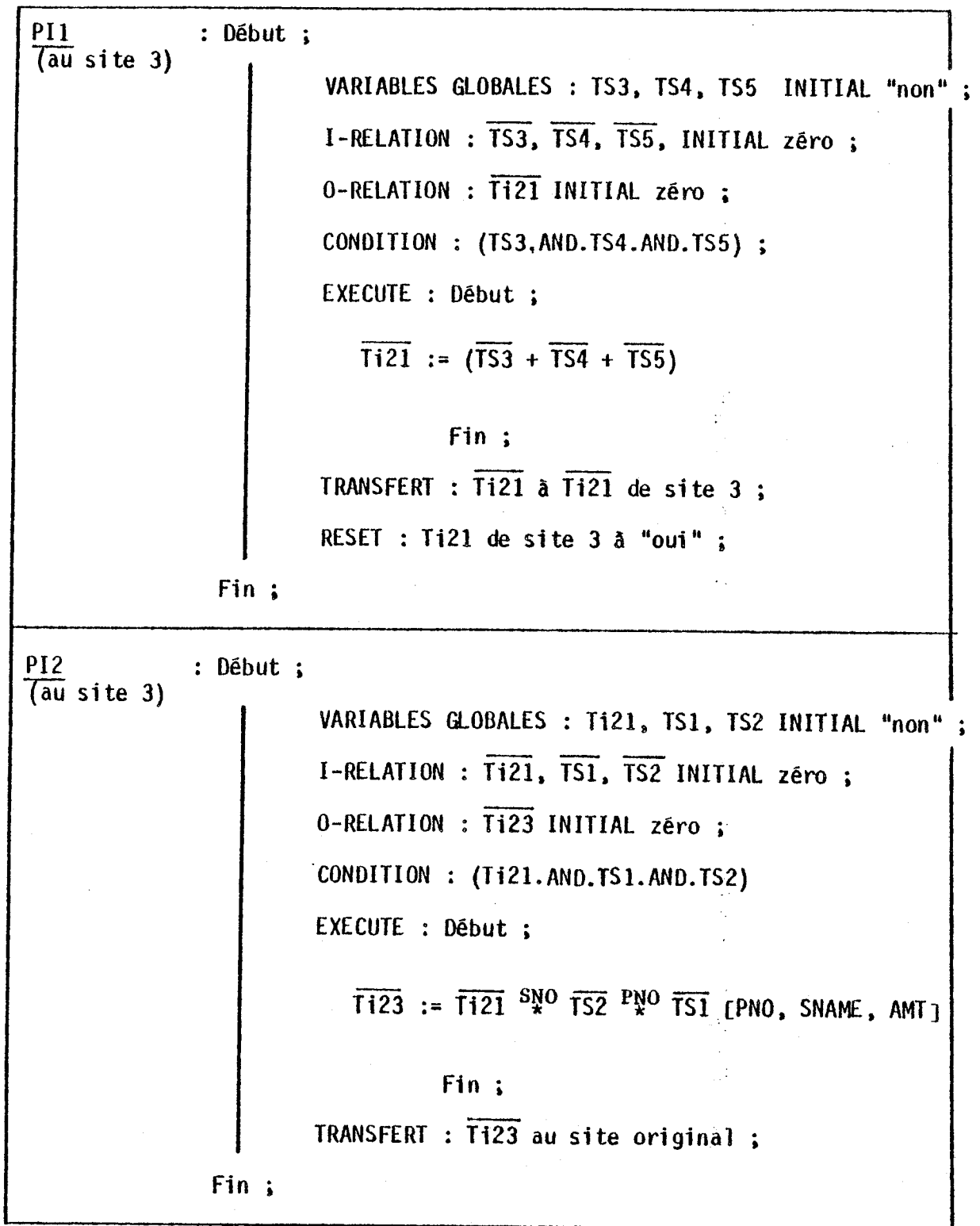


Figure 18-b : les Processus Intermédiaires

CHAPITRE IV

EVALUATION

Sommaire

4.1. INTRODUCTION	4.1
4.2. EVALUATION PAR UNE CLASSIFICATION DES STRATEGIES DES TRAITEMENTS REPARTIS	4.1
4.3. EVALUATION PAR LE MESURE DE SATISFACTION AUX OBJECTIFS	4.6

4.1. INTRODUCTION

Après avoir présenté dans les chapitres 2 et 3 les stratégies d'optimisation, nous aborderons maintenant leur évaluation. Premièrement nous allons classer les stratégies des traitements répartis en quatre types différents et nous évaluerons ces stratégies en examinant leurs performances globales. Deuxièmement nous montrerons comment celles-ci satisfont les objectifs de SGBDR qui ont été introduits dans le chapitre 1, et comment elles se situent par rapport aux autres stratégies d'optimisation utilisées dans les autres SGBDR, par exemple, SDD-1 [1], INGRES [2], ADD [6], POLYPHEME [7, 10], modèle de CHU [3] et modèle de NGUYEN [8].

4.2. EVALUATION PAR UNE CLASSIFICATION DES STRATEGIES DES TRAITEMENTS REPARTIS

Nous pensons que la décomposition de la requête externe doit se faire en deux étapes dans tous les SGBDR.

Premièrement, l'étape de génération d'un plan d'exécution qui correspond dans notre terminologie à l'arbre syntaxique (voir chapitre 3). Un plan d'exécution peut être obtenu de deux manières différentes :

- soit d'une manière limitée basée sur une possibilité de génération d'un seul plan d'exécution,
- soit d'une manière exhaustive fondée sur la recherche de toutes les possibilités de génération des plans d'exécution afin de sélectionner le plan optimal.

Deuxièmement, l'étape d'allocation des ressources réparties (les données et les processeurs). Cette étape peut se faire de deux manières différentes :

- soit d'une manière statique qui alloue toutes les ressources avant l'exécution en utilisant les dictionnaires du système pour choisir la meilleure,

- soit d'une manière dynamique qui alloue les ressources au fur et à mesure des exécutions en utilisant les informations actuelles de l'état du système. De cette manière il n'est pas possible de contrôler à un instant donné pendant l'exécution les allocations déjà faites et les allocations qui se feront après. Cette manière correspond donc dans notre terminologie à la distribution absolue. Il peut apparaître un problème inhérent à la nature de cette méthode dans certains environnements, par exemple, dans un réseau partiellement maillé. (voir section 3.5.3.)

La figure 19 représente ces deux étapes de la décomposition. Les solutions multiples de ces deux étapes nous suggèrent quatre stratégies différentes de décomposition montrées dans la figure 20, dont on discutera d'une manière globale les avantages et les inconvénients.

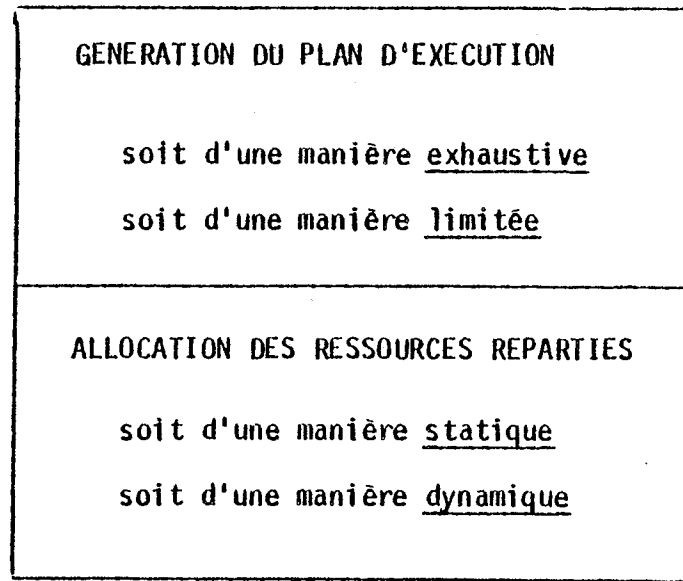
Le Type 1 : génération exhaustive des plans d'exécution et allocation statique des ressources réparties :

(Avantages) Le plan d'exécution sélectionné peut être le plan optimal dans tous les environnements,

(Inconvénients) Puisque l'allocation statique doit allouer toutes les ressources à chaque plan d'exécution avant de l'exécuter, elle ne peut utiliser que les estimations approximatives pour savoir les volumes des résultats intermédiaires. De cette nature on a des risques d'avoir les allocations non optimales causées par les estimations erronées. Il y a aussi des risques de donner les charges supplémentaires au système de garder et maintenir les informations de l'état du système utilisables pour les estimations.

Le Type 2 : génération exhaustive des plans d'exécution et allocation dynamique des ressources réparties :

Requêtes Externes



Traitements répartis

Figure 19 : Deux Etapes de la Décomposition des Traitements Répartis

les étapes de les types de décomposition des stratégies	méthode de génération des plan d'exécution	méthode d'allocation des ressources réparties	exemples
type 1	exhaustive	statique	<ul style="list-style-type: none"> - nos stratégies - SDD-1 - modèle de CHU
type 2	exhaustive	dynamique	<ul style="list-style-type: none"> - INGRES
type 3	limitée	statique	<ul style="list-style-type: none"> - ADD - POLYPHEME
type 4	limitée	dynamique	<ul style="list-style-type: none"> - modèle de NGUYEN

Figure 20 : Classification des Quatre Types de Stratégie de
Décomposition pour les Traitements Répartis

(Avantages) Le plan d'exécution sélectionné peut être meilleur que le plan optimal du type 1 dans tous les environnements puisque l'allocation se fait actuellement par les deux méthodes (premièrement l'allocation statique et deuxièmement l'allocation dynamique afin de diminuer les nombres d'estimations erronées qui ont été faites lors de l'allocation statique,

(Inconvénients) L'allocation dynamique peut exiger beaucoup de temps d'exécution supplémentaire par rapport au type 1. Pourtant sa contribution à l'amélioration des estimations erronées est relativement moins importante [20].

Le Type 3 : génération limitée à un seul plan d'exécution et allocation statique des ressources réparties :

(Avantages) Le processus de décomposition est très simple. Cette simplicité facilite les développements des SGBDR sans risques de surcharges supplémentaires,

(Inconvénients) Le problème principal est que le plan d'exécution généré n'est pas toujours le plan optimal bien qu'il soit optimisé dans l'étape d'allocation. Il y a aussi des risques d'avoir des estimations erronées et d'avoir des charges supplémentaires pour maintenir les informations nécessaires à l'allocation optimale.

Le Type 4 : génération limitée à un seul plan d'exécution et allocation dynamique des ressources réparties :

(Avantages) Par rapport au type 3, il n'y a pas de risques d'avoir des estimations erronées et d'avoir des charges supplémentaires, parce que l'allocation dynamique utilise les informations actuelles de l'état du système pour estimer les résultats intermédiaires,

(Inconvénients) Premièrement, le plan d'exécution généré ne représente

pas toujours l'optimal bien qu'il soit optimisé de manière dynamique au fur et à mesure de l'exécution. Deuxièmement, puisque l'allocation dynamique est de même nature que l'allocation absolue, dans certains environnements (par exemple dans un réseau partiellement maillé) l'allocation dynamique ne représente pas la solution optimale même pour le problème d'allocation (voir section 3.5.3). Pourtant cette méthode exige du temps supplémentaire d'exécution.

A partir de ces analyses des quatre types de stratégies de décomposition, on peut remarquer que ce sont les types 1 et 2 qui sont les meilleures stratégies en toutes circonstances. En particulier le type 1 peut être considéré comme la meilleure stratégie puisque le type 2 ne peut pas contribuer suffisamment à l'amélioration des performances par rapport aux inconvénients qu'il apporte. Ces conclusions globales ont été aussi obtenues par un travail de [20] en utilisant la simulation.

Il est certain que notre stratégie se situe dans le type 1. Nous avons montré, dans la figure 20, quelles étaient les stratégies de décomposition dans les SGBDR déjà développés ou en cours de développement. Les stratégies de SDD-1 [1] et de modèle de CHU [3] se situent dans le type 1 comme notre stratégie. La stratégie de INGRES [2] peut être considérée comme une stratégie de type 2, mais les performances des méthodes exhaustives de génération des plans d'exécution et les méthodes d'allocation optimale des différentes approches ne sont pas identiques. Nous allons montrer à la section suivante que les autres approches ne donnent que des solutions partielles par rapport à notre approche.

4.3. EVALUATION PAR LE MESURE DE SATISFACTION AUX OBJECTIFS

Après avoir effectué la première évaluation de nos stratégies dans la

section précédente par une classification générale des stratégies de décomposition, nous effectuerons maintenant la deuxième évaluation par rapport aux objectifs de satisfaction des SGBDR. Le tableau de la figure 21 résume comment nos stratégies permettent de satisfaire les cinq objectifs des SGBDR décrits dans le chapitre 1.

La première stratégie, la permutation des opérateurs uni-relationnelles (PROJECT, SELECT, etc.) grâce aux propriétés de commutativité, permet de réduire la taille des résultats intermédiaires par l'exécution de ces opérateurs dans les phases initiales. Cela peut donc contribuer directement à satisfaire les objectifs (1), (2), (3) et (4). Ce type d'optimisation est utilisé dans l'étape de localisation dans la plupart des approches, par exemple, SDD-1, INGRES, ADD et POLYPHEME, etc.

La deuxième stratégie, l'allocation du domaine optimal grâce aux duplications redondantes (voir section 3.4), consiste en fait à diminuer la superficie logique de chaque traitement par le regroupement des sites concernés dans le traitement. On peut donc réduire les distances logiques de chaque transmission. Cela peut donc satisfaire directement l'objectif (1) et indirectement les objectifs (4) et (5). Ce type d'optimisation n'a pas été utilisé dans les autres approches, pourtant le système SDD-1 utilise le concept de matérialisation "non regroupé" ("not clustered") en utilisant des duplications redondantes afin d'améliorer la disponibilité des données.

La troisième stratégie, la distribution totale et relative des opérations (transactions) grâce aux processeurs distribués (voir section 3.5), permet en effet de réduire le temps de transmission. Cela peut donc directement contribuer à atteindre l'objectif (1). Le concept de distribution totale peut aussi indirectement contribuer aux objectifs (3) et (4).

Objectif	(1) Minimisation du temps de transmission	(2) Minimisation du temps d'exécution	(3) Amélioration de parallélisme	(4) Distribution du trafic et de la charge	(5) Réduction des duplications
Stratégie d'optimisation					
1. Permutation de PROJECT et SELECT	x	x	x	x	
2. Allocation du domaine optimal	x			x	x
3. Distribution totale et relative des transactions	x		x	x	
4. Permutation de : PRODUCT et UNION	x	x	x	x	
5. Permutation de PRODUCT	x	x			

Figure 21 : Influence des Stratégies sur les Objectifs Fixés

dans un Système SGBDR

Le concept de distribution totale était utilisé dans les systèmes SDD-1 et INGRES et préconisé dans les modèles de CHU et NGUYEN. Cependant, ADD et POLYPHEME n'ont utilisé que la distribution partielle. Le concept de distribution relative, conséquence de la notion de distance logique n'a pas été utilisé dans les autres approches.

Le concept de distribution absolue qui ne tient pas compte de la distance logique de chaque transmission a été utilisé dans la plupart des approches, SDD-1, POLYPHEME et modèle de CHU. Le concept de distribution, préconisé dans les approches INGRES et modèle de NGUYEN, est aussi voisin de la distribution absolue en raison de leur conception de la distribution dynamique. Car dans ce concept, la distribution des opérations est déterminée au fur et à mesure de l'exécution, on ne peut pas contrôler les distributions déjà passées et futures pour optimiser la distribution à un instant donné.

La quatrième stratégie, la permutation des opérateurs de PRODUIT et UNION grâce à la propriété de double distributivité (voir section 3.3.2), signifie en fait l'utilisation de deux philosophies de distribution (transfert des opérations vers les données et transfert des données vers les opérations) afin de réduire le temps de transmission. Cela peut donc satisfaire directement les objectifs (1) et (2). Les objectifs (3) et (4) peuvent être aussi améliorés par la philosophie de transfert des opérations vers les données. Ces deux types de philosophies n'ont pas été envisagés dans les autres approches pour réduire le temps de transmission. Une seule était utilisée. Les approches, SDD-1, ADD, POLYPHEME et modèles de CHU et NGUYEN ont utilisé la philosophie du transfert des données vers les opérations, alors que l'approche, INGRES, a utilisé la philosophie de transfert des opérations vers les données.

La cinquième stratégie qui permet de permuter les opérands de l'opérateur de PRODUIT grâce à la propriété de commutativité (voir section 3.3.3), a pour conséquence de réduire le volume des résultats intermédiaires des opérations PRODUIT. Elle peut donc satisfaire directement les objectifs (1) et (2). Ce type d'optimisation est fait de différentes manières dans les approches précédentes. Par exemple, dans le système INGRES, les sélections d'une partie de l'expression ("piece") grâce à cette propriété de PRODUIT peut réaliser ce type d'optimisation. [2] Le séquençement ("sequencing") du modèle de CHU [3] est pour réaliser aussi ce type d'optimisation. Le système-R [5], un SGBD centralisé, a aussi utilisé ce type d'optimisation en considérant l'ordre des opérations de PRODUIT, mais dans lequel il n'est pas possible d'énumérer toutes les permutations possibles, puisque certaines possibilités de réaliser le PRODUIT non cartésien entre deux relations peuvent être cachées dans les expressions du langage SEQUEL. [5]

La figure 21 montre bien que tous les objectifs peuvent être satisfaits par nos stratégies d'optimisation. Plus particulièrement l'objectif (1), qui permet de résoudre le principal goulot d'étranglement des SGBDR, est satisfait par toutes nos stratégies.

Le tableau de la figure 22 dresse une synthèse des nos stratégies par rapport aux principaux systèmes SGBDR actuellement développés. On peut remarquer que toutes les autres approches ne donnent qu'une réponse partielle en ce qui concerne la méthode exhaustive de décomposition dans un environnement réparti.

Nos stratégies d'optimisation \ les autres approches	SDD-1	INGRES	ADD POLYPHEME	Modèle de CHU	Modèle de NGUYEN
Optimisation dans l'étape de localisation					
1. Permutation de PROJECT et SELECT	x	x	x		x
2. Permutation de PRODUIT et UNION					
3. Permutation de PRODUIT		x		x	
Optimisation dans l'étape d'allocation des données					
4. Allocation du domaine optimal ("clustered materialization")	x ("not clustered materialization")				
Optimisation dans l'étape de distribution des opérateurs					
5. Distribution totale	x	x		x	x
6. Distribution relative					

Figure 22 : l'Existence des Stratégies d'Optimisation dans les Approches SGBDR

CHAPITRE V

LES APPLICATIONS DE NOS STRATEGIES

Sommaire

5.1. INTRODUCTION	5.1
5.2. L'APPLICABILITE DE NOS STRATEGIES	5.1
5.2.1. Définition des Variables	5.1
5.2.2. Applicabilité de Nos Strategies	5.5

5.1. INTRODUCTION

L'approche du SGBDR convient aux applications qui présentent à la fois les deux caractéristiques : la distribution géographique des ressources (les données et les processeurs) et des utilisateurs, et l'exigence de deux types d'activité (les activités coopératives qui nécessitent des informations intégrées et les activités locales qui nécessitent seulement une partie des informations). Nous pensons qu'il y a beaucoup d'applications réelles qui possèdent ces deux caractéristiques. Par exemple, six types différents d'applications réparties ont été répertoriées dans le travail [18], comme le montre de la figure 23, et dans lequel la classification est basée sur la caractéristique de transmission des données. La caractéristique de transmission est une des caractéristiques principales, mais cette caractéristique n'est pas suffisante pour distinguer les modèles pour lesquelles nos stratégies sont adoptables. Nous voulons donc aussi considérer d'autres caractéristiques des applications réelles avec lesquelles nous décrirons dans la section suivante le type d'application de nos stratégies.

5.2. L'APPLICABILITE DE NOS STRATEGIES

5.2.1. Définition des variables

Afin de mieux caractériser les types d'application pour nos stratégies, nous considérerons quatre paramètres principaux qui influent forcément sur l'architecture de la base de données ou sur les critères d'optimisation :

- types de traitements,
- types de répartition des ressources et des utilisateurs,

Types d'Application	Caractéristiques	Exemples
Applications conventionnelles	émission de séries des messages courtes et transmission équilibré	services de réservations pour transports aériens
Question/Réponse	réponses longues à des questions courtes	service de renseignement
Acquisition des données	entrées longues et réponses courtes	contrôle d'inventaire
Application de traitement par lots	sorties très longues	soumission de travaux à distance
Application du temps réel	temps d'exécution rapide	système de temps réel
Application à application	communication déséquilibrée	système bancaire

Figure 23 : Six Types Différents d'Application Répartie

selon les Caractéristiques de Transmission des Données

- types des exigences principaux pour les performances du SGBDR
- types de requêtes

Pour décrire ces caractéristiques, nous définirons un ensemble de variables.

- Pour les types de traitements, nous définirons les deux variables suivantes :

$$\underline{\text{le taux d'exception}} := \frac{\text{nombre d'activités coopératives}}{\text{nombre d'activités}}$$

$$\underline{\text{le taux de mise à jour}} := \frac{\text{nombre de mise à jour}}{\text{nombre de requêtes}}$$

par exemple, si le taux d'exception d'une application est "1" ou voisin de "1", c'est une application où il n'y a que des activités coopératives. Dans ce cas l'architecture de la base de données de cette application sera sans doute une base de données centralisée. Par contre si le taux d'exception d'une application est "0" ou voisin de "0", c'est une application où il n'y a que des activités locales, dans ce type d'application il est suffisant que chaque activité locale gère ses données soit par des fichiers indépendants soit par une base de données centralisée et indépendante. Le choix d'une base de données réparties sera valable pour les applications qui satisfont la condition : $0 \ll \text{taux d'exception} \ll 1$. Quant au taux de mise à jour, il est évident que la répartition à redondance (totale ou partielle) sera très fortement influencée par la valeur de cette variable. Par exemple, si le taux de mise à jour est voisin de "0", on peut bénéficier presque sans limite des duplications redondantes pour améliorer les performances du SGBDR, par contre si le taux de mise à jour est très grand, il vaut mieux éviter l'utilisation des duplications redondantes.

- Pour les types de répartition des ressources et des utilisateurs, nous considérerons les deux variables suivantes :

le volume des données et l'espace de répartition des ressources et des utilisateurs qui peut être représenté en terme géographique ou en terme de surface logique dans un environnement donné (voir chapitre 3). Avec ces deux variables qui représentent globalement la cause principale des délais de transmission, on peut dire si un SGBDR est très grand ("Very Large Data Base") ou non.

- Pour les types d'exigence, nous considérerons les trois variables suivantes :

le temps de réponse, le coût d'exécution et la productivité du SGBDR et du réseau de communication. Il est évident que une pondération de ces trois variables conduit à des critères d'optimisation dans les environnements répartis.

- Pour les types de requêtes, nous définirons les deux variables suivantes :

la condition d'interrogation (soit complexe soit simple) qui montre la complexité des conditions d'interrogation des requêtes et le type de sélection (soit recherche par clé soit recherche par qualification) qui montre la caractéristique de la sélection : par élément ou par groupe. Par exemple, si la condition d'interrogation est très complexe, les optimisations des transformations de requêtes seront très efficaces pour l'amélioration des performances. Si le type de sélection est la recherche par qualification, les traitements de requêtes auront beaucoup de données intermédiaires à transférer, c'est à dire que les optimisations des transformations de requêtes et d'allocation des ressources seront très efficaces.

5.2.2. Applicabilité de nos stratégies

Nous avons défini à la section précédente un ensemble de variables. En utilisant cet ensemble de variables, nous caractérisons les modèles d'applications réelles, pour lesquelles nos stratégies sont applicables, comme le montre de la figure 24. C'est à dire :

- Puisque nos stratégies ne servent qu'aux requêtes globales, plus le taux d'exception est grand en satisfaisant la condition ($0 \ll \text{le taux d'exception} \ll 1$), plus nos stratégies seront efficaces ;
- Puisque notre stratégie d'allocation des ressources est basée sur les duplications redondantes, plus le taux de mise à jour est petit en satisfaisant la condition ($0 < \text{le taux de mise à jour} < 1$), plus nos stratégies seront efficaces ;
- Grâce aux critères d'optimisation de nos stratégies (voir chapitre 1), plus les SGBDR sont grand, plus nos stratégies seront efficaces ;
- Grâce aux objectifs de nos stratégies (voir chapitre 1), plus l'accélération du temps de réponse, la réduction du coût d'exécution et l'amélioration de la productivité sont exigés, plus nos stratégies seront adoptables ;
- Puisque nos stratégies de permutation des opérateurs sont basées sur la multiplicité des opérateurs, plus la condition de répercution est complexe et plus nos stratégies seront efficaces ;
- Puisque le principal objectif de nos stratégies est de réduire les transmissions intermédiaires, nos stratégies sont plus bénéfiques pour les sélections par qualification qui donnent large volume de données intermédiaires à transférer.

Paramètres	Variables	Caractéristiques
1. type de traitements	1. taux d'exception 2. taux de mise à jour	0 << taux < 1 0 < taux << 1
2. type de ressources et utilisateurs	1. volume de données 2. l'espace géographique de répartition des ressources et utilisateurs	large large
3. type des exigences principaux	1. temps de réponse 2. coût d'exécution 3. productivité	haute priorité haute priorité haute priorité
4. type de requête	1. condition d'interrogation 2. sélection	complexe par qualification

Figure 24 : les Caractéristiques des Modèles d'Application de
Nos Stratégies

Nous pensons qu'il existe des applications réelles dont les caractéristiques satisfont les conditions d'applicabilité de nos stratégies (la figure 24). On peut citer en exemples les applications suivantes :

- Les services d'information basés sur un large volume de données qui sont géographiquement réparties sur plusieurs installations informatiques comme un ensemble d'applications locales dont la sémantique est relativement statique, par exemple, les services de renseignements, "Conventional Information Retrieval Systems" et les services statistiques des administrations de niveau national ou international, etc. ;
- Les applications temps réels basées sur les données géographiquement réparties dans un gros réseau de communication, par exemple, "missile control systems", etc. ;
- etc.

CHAPITRE VI

CONCLUSION

La performance du SGBDR peut être améliorée par optimisations de plusieurs niveaux. Nous voulons citer les principaux paramètres de la performance du SGBDR :

- Les paramètres du réseau de communication,
 - 1) Le type de topologie,
 - 2) Le type de communication,
 - 3) Le type de transmission,
 - 4) La capacité des lignes, etc.
- Les paramètres du site,
 - 1) La vitesse d'exécution de la machine,
 - 2) La capacité du stockage,
 - 3) Les méthodes d'accès,
 - 4) Les "hardware" spéciaux (DB-machine, "sorter", etc.),
 - 5) Les "software" spéciaux (indexation secondaire, "link and clustering", etc.), etc.
- Les paramètres de l'utilisateur,
 - 1) Les taux d'exception et de mise à jour,
 - 2) Les exigences (temps de réponse, coût d'exécution, productivité),
 - 3) Les types de requêtes, etc.
- Les paramètres de la distribution des ressources,
 - 1) Les volumes des données et la distribution des duplications redondantes,
 - 2) La distribution des processeurs, etc.
- Les paramètres des traitements de la requête,
 - 1) La localisation des données,
 - 2) Les transformations de requête,

- 3) L'utilisation des données réparties,
- 4) L'utilisation des processeurs répartis, etc.

La meilleure performance d'un SGBDR peut être obtenu par les optimisations de tous ces paramètres. Mais les paramètres du réseau et du site ne peuvent pas être optimisés pour un seul SGBDR car le réseau et les sites sont utilisés aussi par les autres applications réparties ou non réparties. De même on ne peut pas contrôler les paramètres de l'utilisateur. Nous considérerons donc ces trois groupes des paramètres comme les paramètres de l'environnement et nous ne les avons pas compté dans cette étude. Nos études portaient plus précisément sur les optimisations des paramètres de traitements de la requête (le dernier groupe).

L'idée principale de nos stratégies est d'effectuer sur la requête externe toutes les transformations possibles. Ces transformations résultent des possibilités :

- de permuter les opérateurs grâce à leur propriétés algébriques,
- d'allouer à chaque transformation un domaine minimal par le groupement des sites concernés,
- de distribuer les opérations sur les sites en vue de minimiser le temps de transmission.

Finalement on prend une transformation qui minimise le temps de transmission entre toutes les transformations. Nos stratégies ont permis néanmoins d'explorer certaines solutions, il reste donc des problèmes à résoudre. Par exemple,

- l'estimation du volumes des résultats des transactions initiales et intermédiaires,
- la manipulation des dictionnaires du système qui sert à l'optimisation.

En effet, ce sont des problèmes inhérents à l'optimisation statique dans un environnement réparti. Nos prochaines études seront consacrées à résoudre ces problèmes de façon convenable, à adapter nos stratégies en conséquence et aussi à résoudre les optimisations des paramètres des données réparties.

BIBLIOGRAPHIES

- [1] E. WONG,
"Retrieving Dispersed Data from SDD-1 : A System for
Distributed Data Bases",
Proceedings of the Second Berkeley Workshop on Distributed
Data Management and Computer Networks, Mai 1977.
- [2] R. EPSTEIN, M. STONEBRAKER and E. WONG,
"Distributed Query Processing in a Relational Data Base
System", SIGMOD Proceedings, Mai 1978.
- [3] W. Chu. WESLEY, Paul HURLEY,
"A Model for Optimal Query Processing for Distributed
Data Bases",
Digest of Papers, COMPCOM 79 printemps.
- [4] In-Sup PAIK, C. DELOBEL,
"A Strategy for Optimizing the Distributed Query Processing",
Proceedings of the First International Conference on
Distributed Computing System, Huntsville Alabama, Octobre
1979.
- [5] P. GRIFFITHS SELINGER et AL.,
"Access Path Selection in a Relational Data Base Management
System",
Research Report for Computer Science, IBM Research Lab.,
San José, California, Août 1979.
- [6] K.C. TOTH, S.A. MAHMOUD, J.S. RIOUDON, O. SHERIF,
"The ADD Systems",
Proceedings of the Fourth International Conference on
V.L.D.B., Berlin, Septembre 1978.

- [13] Stephen R. KIMBLETON,
"Computer Communication Network : Approaches, Objectives
and Performance Considerations",
Computing Survey, Vol.7 N°3, Septembre 1975
- [14] James B. ROTHNIE, Nathan GOODMAN,
"A Survey of Research and Development in Distributed
Data Base Management",
Proceedings of VLDB, Tokyo, Japan, 1977.
- [15] E.F. CODD,
"Relational Completeness of Data Base Sublanguage",
Courant Computer Science Symposia 6, Data Base System,
Prentice Hall, 1972.
- [16] D. CHAMBERLIN et Al,
"SQEL 2 - A Unified Approach to Data Definition, Mani-
pulation and Control",
IBM Journal of Research and Development, Vol.20 N°6,
Novembre 1976.
- [17] In-Sup PAIK, C. DELOBEL,
"L'optimisation des Requêtes dans un Environnement de
Base de Données Réparties",
soumis à la revue de l'AFCEC, Juillet 1980.
- [18] G.A. CHAMPINE,
"Six Approaches to Distributed Data Base",
Datamation, Vol.23 N°5, Mai, 1977.

- [7] J.Y. CALECA,
"L'expression et la Décomposition de Transactions dans
un Système de Base de Données Réparties",
Thèse de 3ème Cycle, USMG/INGP, France, Septembre 1978.
- [8] NGUYEN GIA TOAN,
"A Unified Method for Query Decomposition and Shared
Information Updating in Distributed Systems",
Proceedings of the first International Conference on
Distributed Computing Systems,
Huntsville, Alabama, Octobre 1979.
- [9] R.Y. CHANG, J.M. SMITH,
"Optimizing the Performance of a Relation Algebra Data
Base Interface",
CACM, Vol. 18 N° 10, Octobre 1975.
- [10] M. ADIBA,
"Un Modèle Relationnel et une Architecture pour les
Systèmes de Base de Données Réparties",
Thèse de Docteur es-science, USMG France, Septembre 1978.
- [11] In-Sup PAIK,
"Définition et Réalisation d'un Ensemble d'Opérateurs
Relationnels pour une Machine POLYPHEME",
Rapport D.E.A., USMG/INPG, France, 1978.
- [12] E.F. CODD,
"A Relational Model of Data for Large Shared Data Banks",
CACM, Vol.13 N°6, Juin 1970.

- [19] Eugene WONG, Karel YOUSSEFI,
"Decomposition - A Strategy for Query Processing",
Transactions on Data Base System, ACM, Vol.1 N°3, Septembre
1976.
- [20] Robert EPSTEIN, Michael STONEBRAKER,
"Analysis of Distributed Data Base Processing Strategies",
Proceedings of VLDB, Montreal, Canada, 1980.

Dernière page d'une thèse

VU

Grenoble, le 7 Janvier 1980

Le Président de la thèse

M. B. Lhot

Vu, et permis d'imprimer,

Grenoble, le 12.1.81

Le Président de l'Université
Scientifique et Médicale

J. G. Cau

Dr G. Cau

