

Научный семинар "Параллельные вычислительные технологии"
3 февраля 2005 г., Челябинск

Прототип параллельной СУБД для мультипроцессорных систем с кластерной архитектурой*

Л.Б. Соколинский,
д.ф.-м.н., проф.
sokolinsky@acm.org

М.Л. Цымблер,
к.ф.-м.н., доц.
mzym@susu.ru

Кафедра системного программирования
Южно-Уральский государственный университет

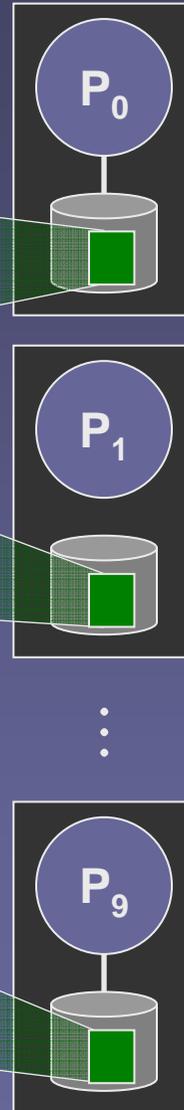
* Работа выполнена при финансовой поддержке РФФИ, проект 03-07-90031

Параллельная СУБД

Исходная таблица

Атр1	...	Атрибут фрагментации
...	...	00 ...
...	...	09 ...
...	...	10 ...
...	...	19 ...
...
...	...	90 ...
...	...	99

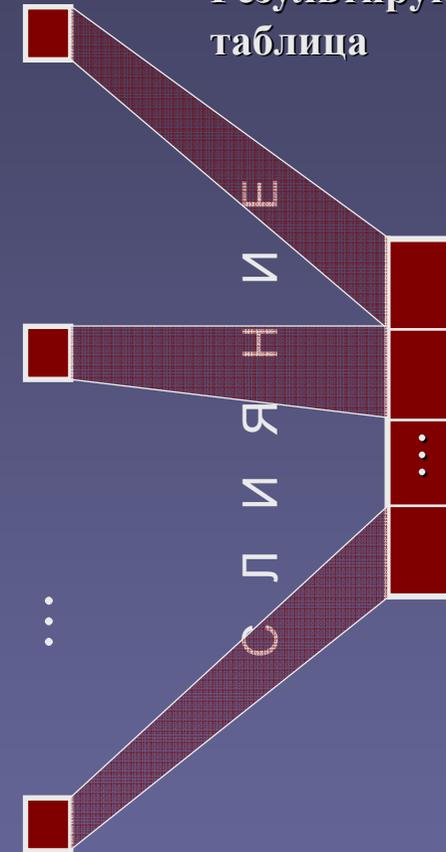
ФРАГМЕНТАЦИЯ



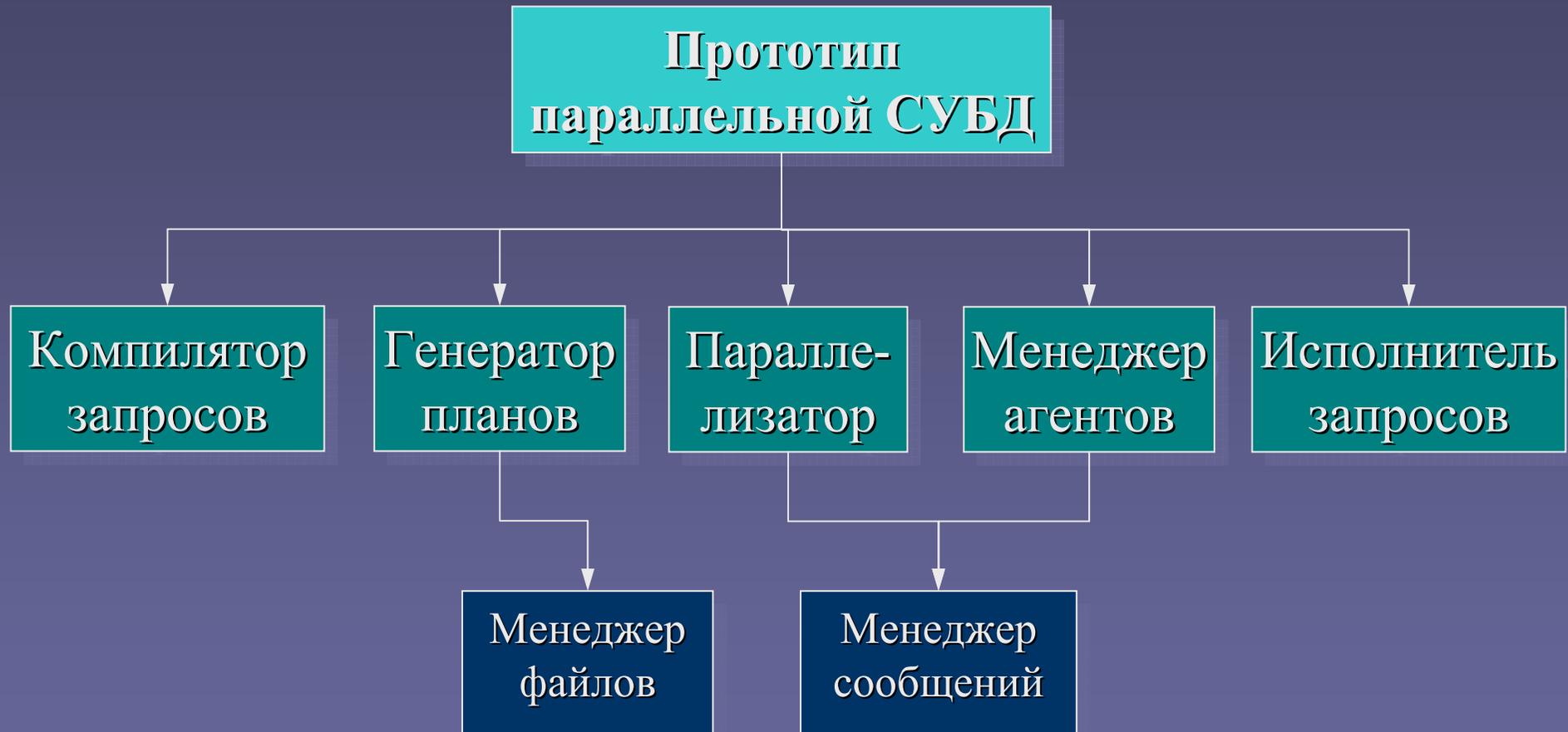
Функция фрагментации ↷

ПАРАЛЛЕЛЬНЫЕ АГЕНТЫ

Результирующая таблица



Омега: прототип параллельной СУБД



Модельная база данных "Поставки"

П (Поставщики)

Код_П*	Имя_П	Город
23	Иванов И.И.	Москва
14	Петров П.П.	Самара

П – фрагментировано по Код_П:
 $\psi_P(p) = p.\text{Код_П} \text{ div } 10$

Д (Детали)

Код_Д*	Имя_Д	Цвет
3	Гайка	Красный
7	Болт	Синий

Д – фрагментировано по Код_Д:
 $\psi_D(d) = d.\text{Код_Д} \text{ div } 10$

ПД (Поставки)

Код_ПД*	Код_П#	Код_Д#
1	14	7
2	23	3

ПД – фрагментировано по Код_Д:
 $\psi_{PD}(pd) = pd.\text{Код_Д} \text{ div } 10$

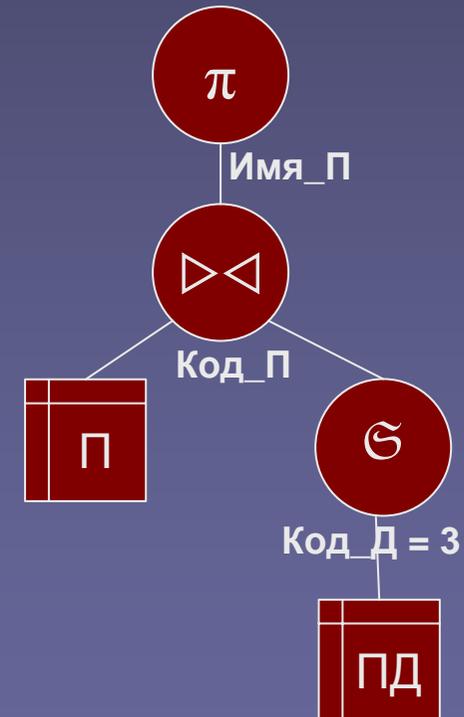
Компилятор запросов

```
-- Имена поставщиков  
-- детали с кодом 3  
SELECT Имя_П  
FROM П, ПД  
WHERE П.Код_П = ПД.Код_П  
AND ПД.Код_Д = 3;
```

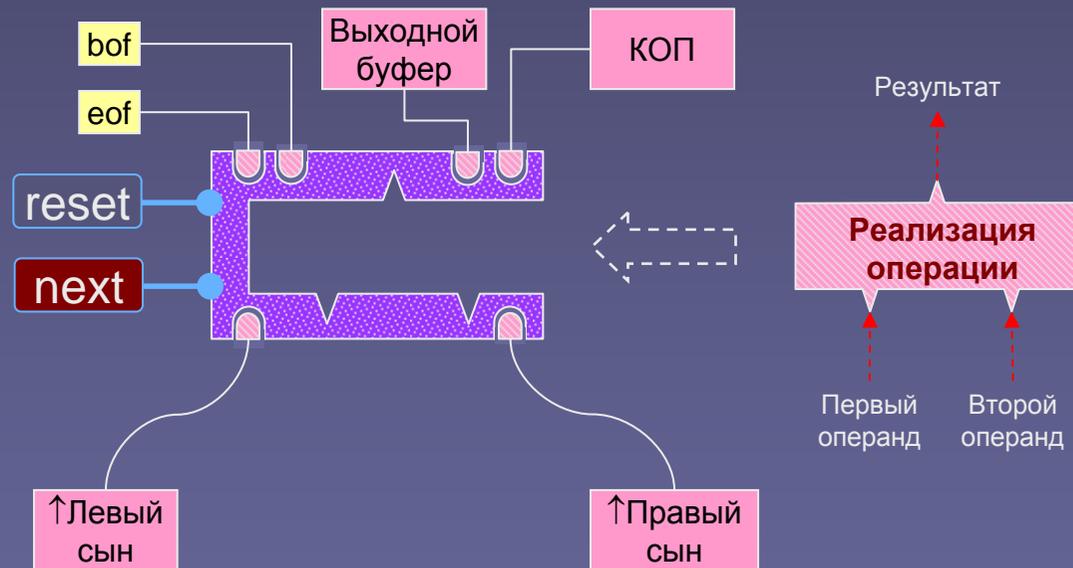


Компилятор запросов

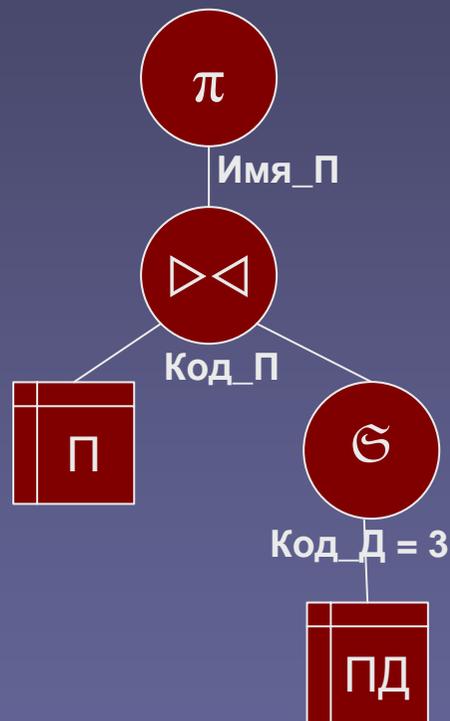
$\pi_{\text{Имя_П}}(\text{П} \bowtie (\sigma_{\text{Код_Д}=3}(\text{ПД})))$



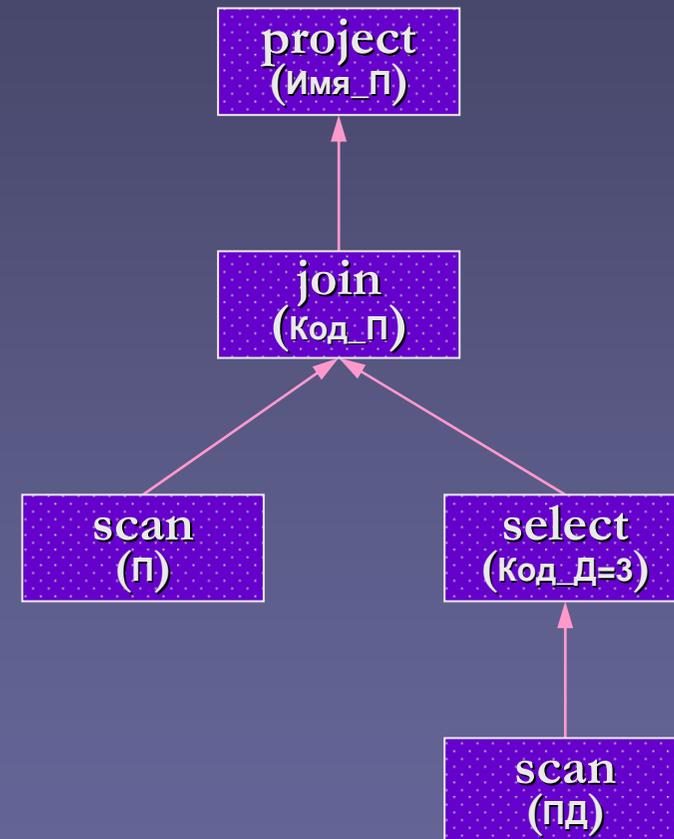
Скобочный шаблон



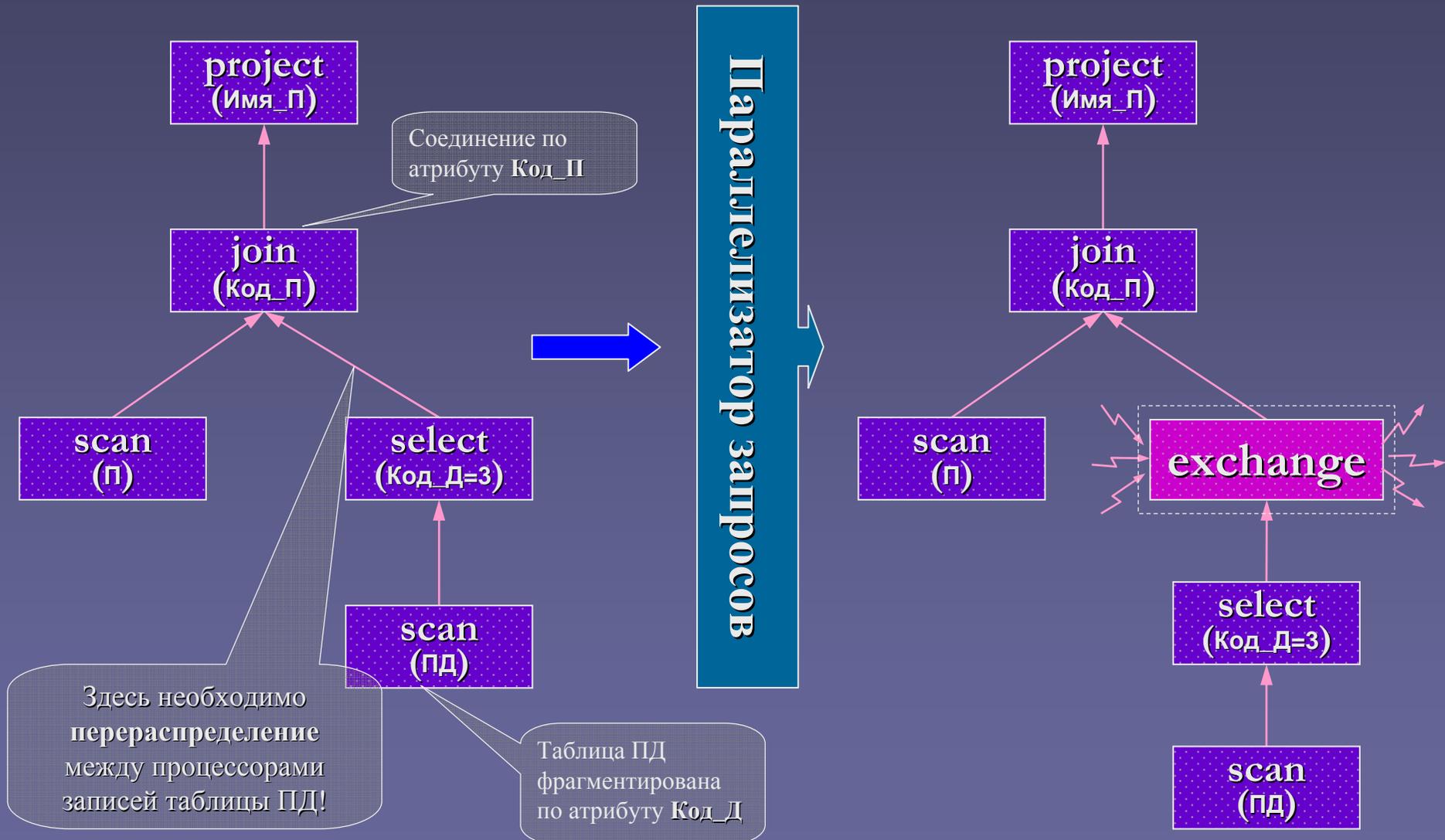
Генератор последовательных планов



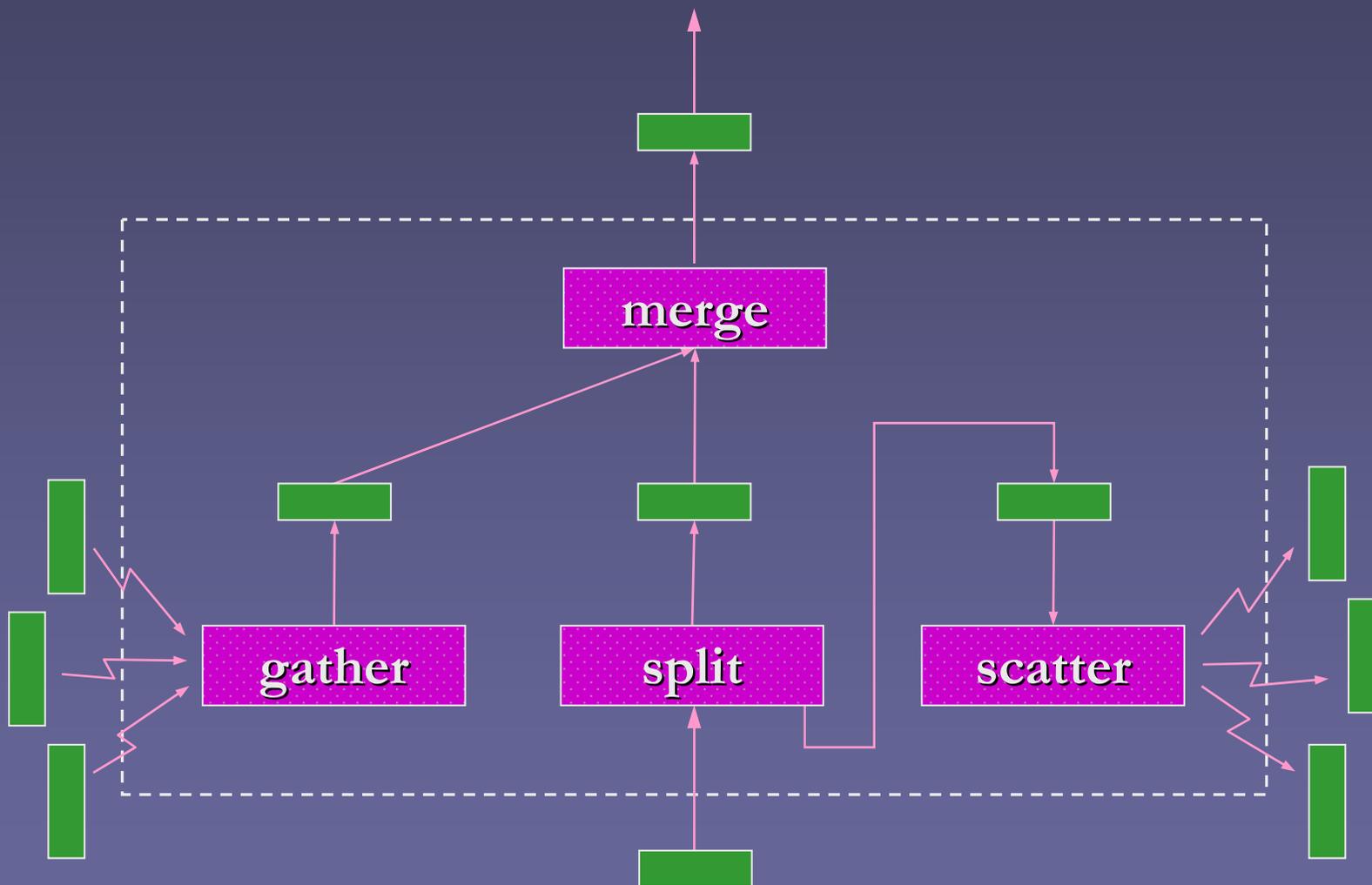
Генератор планов



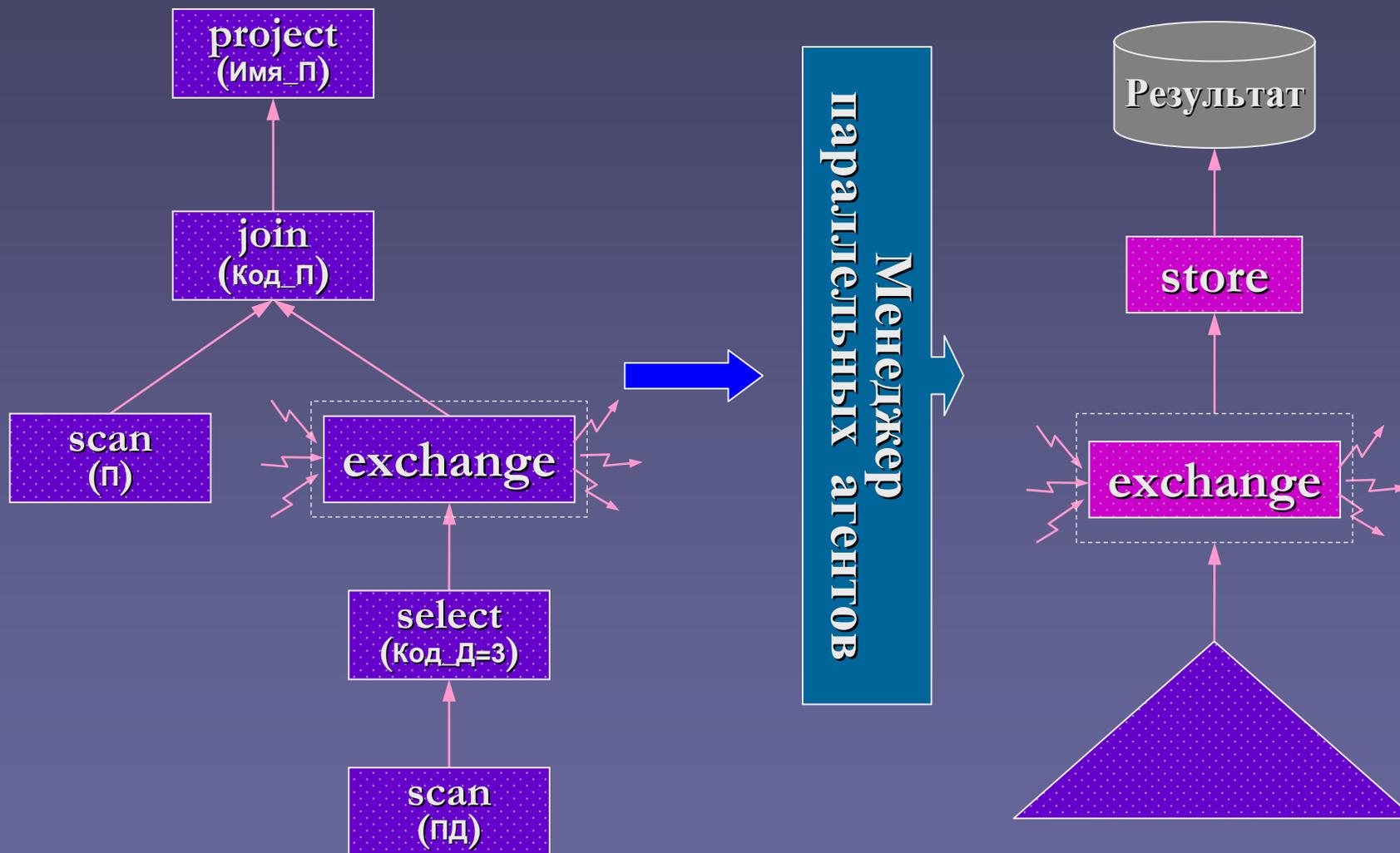
Параллелизатор запросов



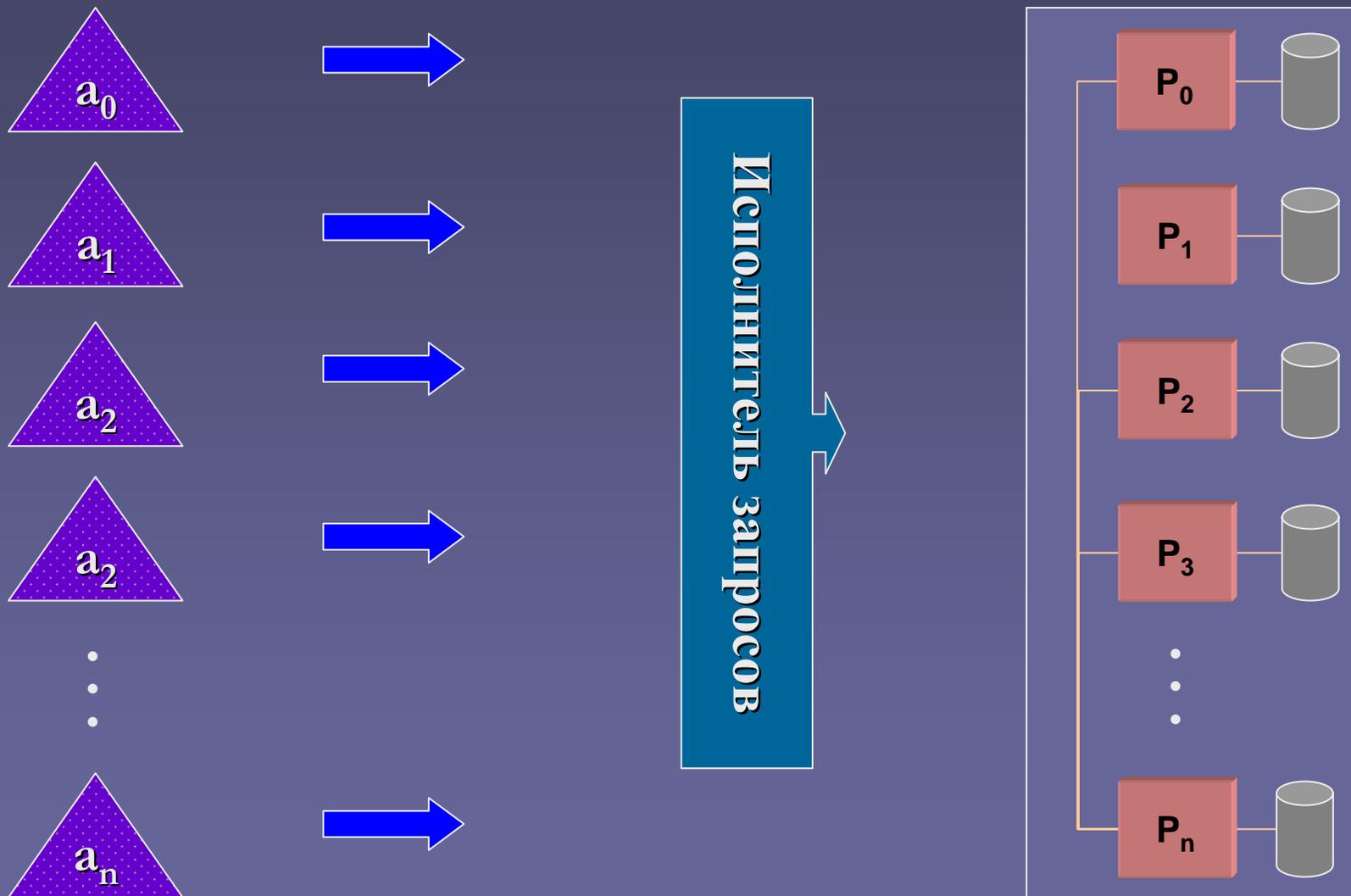
Оператор обмена exchange



Менеджер параллельных агентов



Исполнитель запросов



Менеджер файлов, Менеджер сообщений

/* Открыть отношение */

`void OpenTable (int table);`

/* Закрыть отношение */

`void CloseTable (int table);`

/* Подготовить отношение к сканированию */

`void ResetTable (int table);`

/* Выдать следующий кортеж отношения */

`void * NextRecord (int table);`

/* Запустить отправку сообщения */

`int RunSend (int port, int dest, void * buf, int len);`

/* Запустить прием сообщения */

`int RunRecv (int port, int src, void * buf, int len);`

/* Проверить завершение операции */

`int IsDone (int operation);`

Заключение

- Разработан прототип параллельной СУБД для кластерных вычислительных систем. Спецификации и исходные тексты доступны на сайте <http://omega.susu.ru>.
- Прототип апробирован на следующих аппаратно-программных платформах:
 - ВВК ЮУрГУ, SUSE Linux, MVARICH;
 - ПК Pentium IV, MS Windows, MPICH.
- Начаты вычислительные эксперименты по исследованию масштабируемости прототипа.