



Параллельные системы баз данных

Лекция 2. Формы параллельной обработки транзакций

Разработчик:

Л.Б. Соколинский, д.ф.-м.н., профессор

E-mail: sokolinsky@acm.org

Южно-Уральский государственный университет

Направление 010300.68

«Фундаментальная информатика и информационные технологии»

Проект комиссии Президента по модернизации и техническому развитию экономики России
«Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения»



Проект «Создание системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения»

Применение потенциала суперкомпьютерных технологий (СКТ) как значимой составляющей инновационного развития страны является задачей государственной важности, относится к приоритетному направлению и находится под постоянным контролем Президента и Правительства России. Одним из сдерживающих факторов развития страны в этом направлении является острая нехватка высококвалифицированных кадров в области СКТ, поскольку подготовка таких специалистов сейчас отсутствует как элемент системы высшего профессионального образования.

Стратегической целью проекта является создание национальной системы подготовки высококвалифицированных кадров в области суперкомпьютерных технологий и специализированного программного обеспечения.

Сайт проекта <http://hpc-education.ru>.



Содержание курса

1. Введение
2. Формы параллельной обработки транзакций
3. Определение параллельной системы баз данных
4. Классификация многопроцессорных систем
5. Архитектура параллельных систем баз данных
6. Организация межпроцессорных обменов
7. Балансировка загрузки в многопроцессорных иерархиях



Основная литература

1. *Taniar D., Leung C.H.C., Rahayu W., Goel S.* High Performance Parallel Database Processing and Grid Databases. John Wiley & Sons, 2008.
2. *Гарсиа-Молина Г., Ульман Дж., Уидом Дж.* Системы баз данных. Полный курс. М.: Издательский дом "Вильямс", 2004. 1088 с. (Раздел 15.9)
3. *Соколинский Л.Б.* Параллельные машины баз данных // Природа. Естественно-научный журнал Российской академии наук. - 2001.№8. -С. 10-17.
4. *Соколинский Л.Б.* Обзор архитектур параллельных систем баз данных // Программирование. -2004.№6. -С. 49-63.
5. *Соколинский Л.Б.* Организация параллельного выполнения запросов в многопроцессорной машине баз данных с иерархической архитектурой // Программирование. -2001.№6. -С. 13-29.



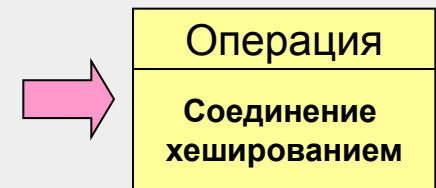
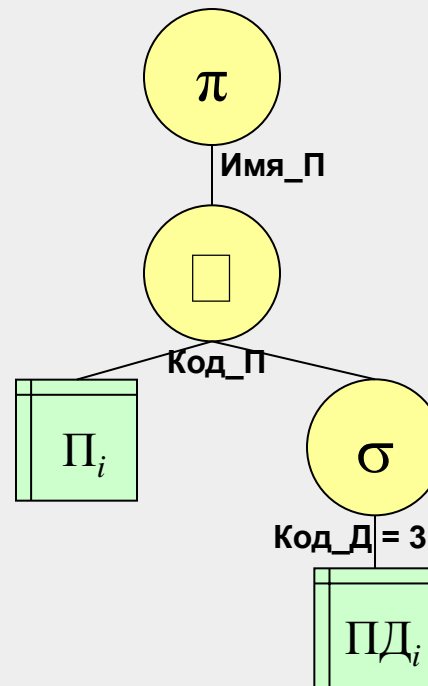
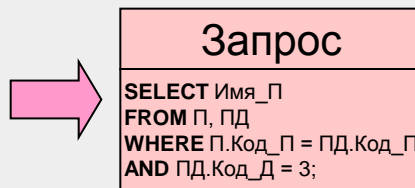
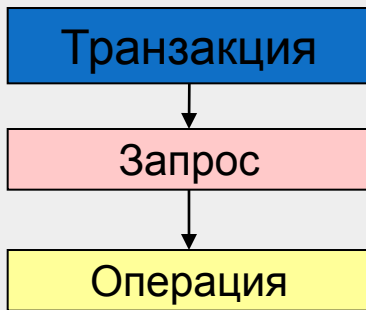
Дополнительная литература

1. *Костенецкий П.С., Лепихов А.В., Соколинский Л.Б.* Технологии параллельных систем баз данных для иерархических многопроцессорных сред // Автоматика и телемеханика. -2007. - Том 68, №5. -С. 847-859
2. *Девитт Д., Грэй Д.* Параллельные системы баз данных: будущее высоко эффективных систем баз данных // СУБД. -1995.№2. -С. 8-31.
3. *Stonebraker M.* The case for shared nothing // Database Engineering Bulletin. -1986. -Vol. 9, No. 1. -P. 4-9.
4. *Graefe G.* Query evaluation techniques for large databases // ACM Computing Surveys. -1993. -Vol. 25, No. 2. -P. 73-169.



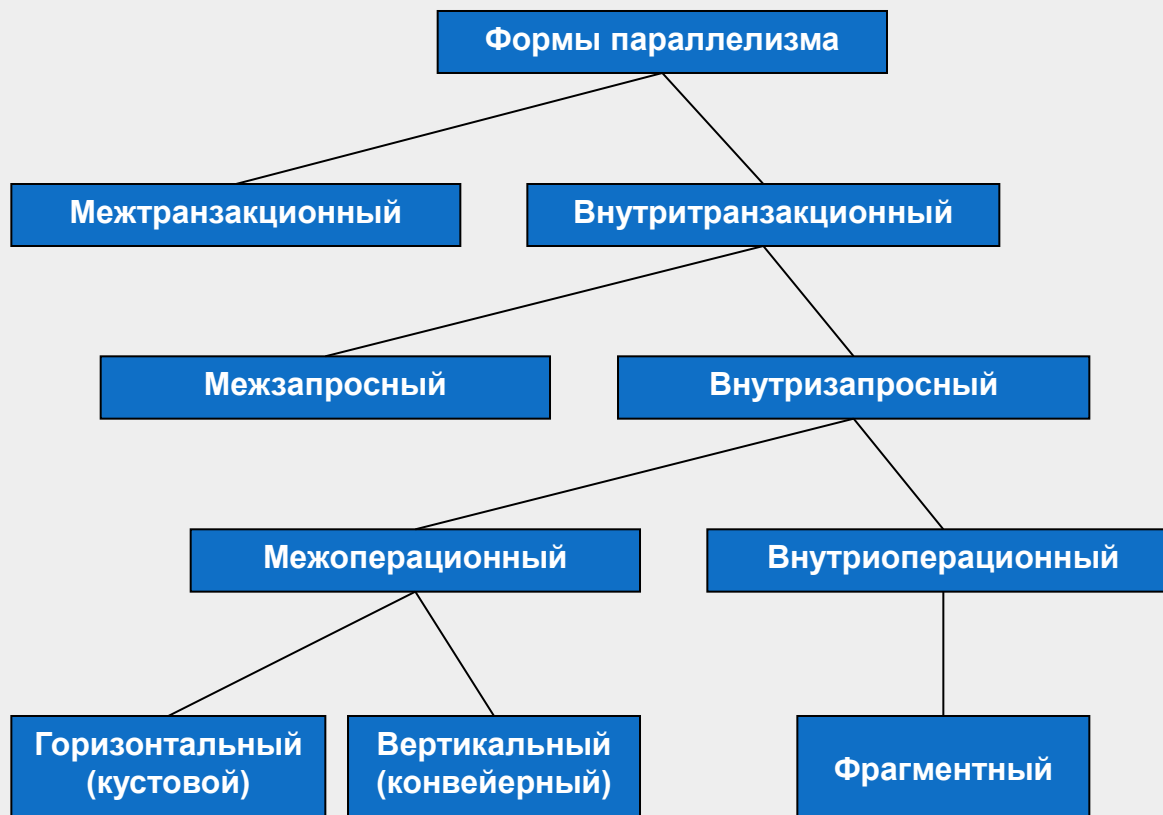
Транзакция – запрос – операция

Иерархическая структура транзакции



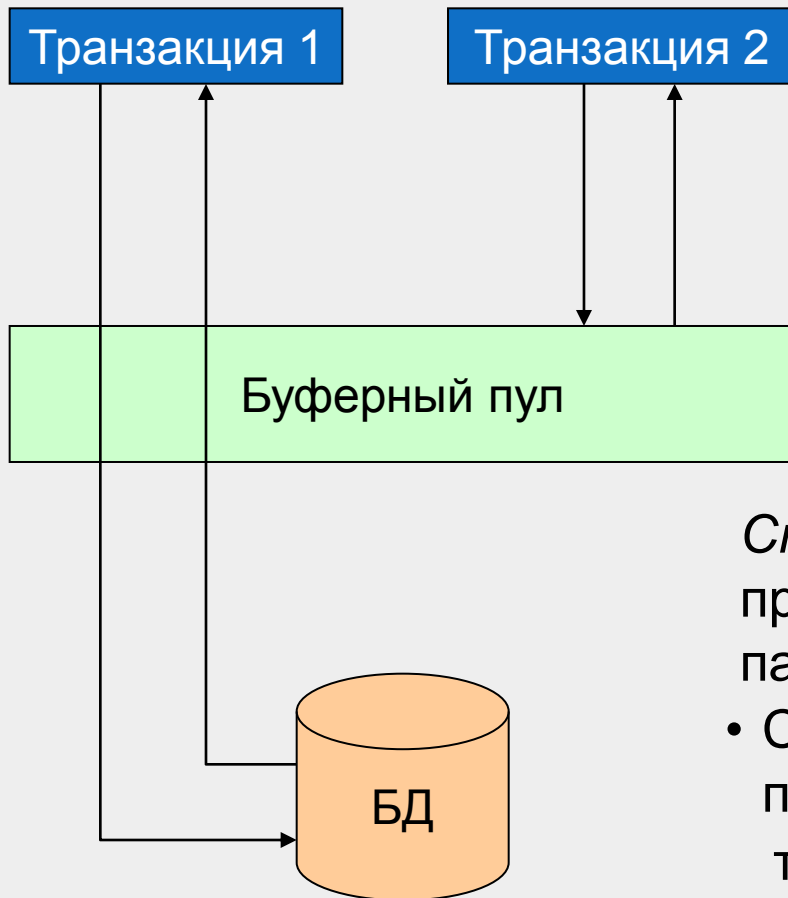


Формы параллелизма





Межтранзакционный параллелизм

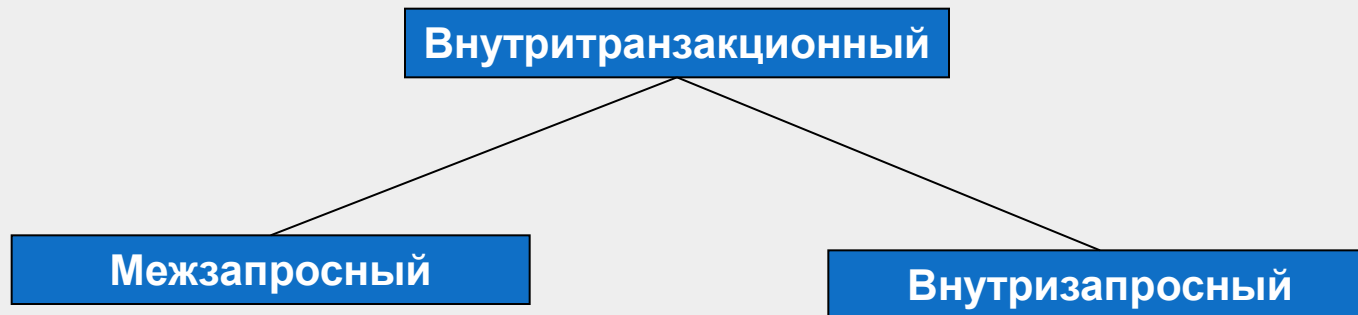


Степень параллелизма – количество процессов, которые могут выполняться параллельно.

- Степень межтранзакционного параллелизма: теоретически неограниченна!

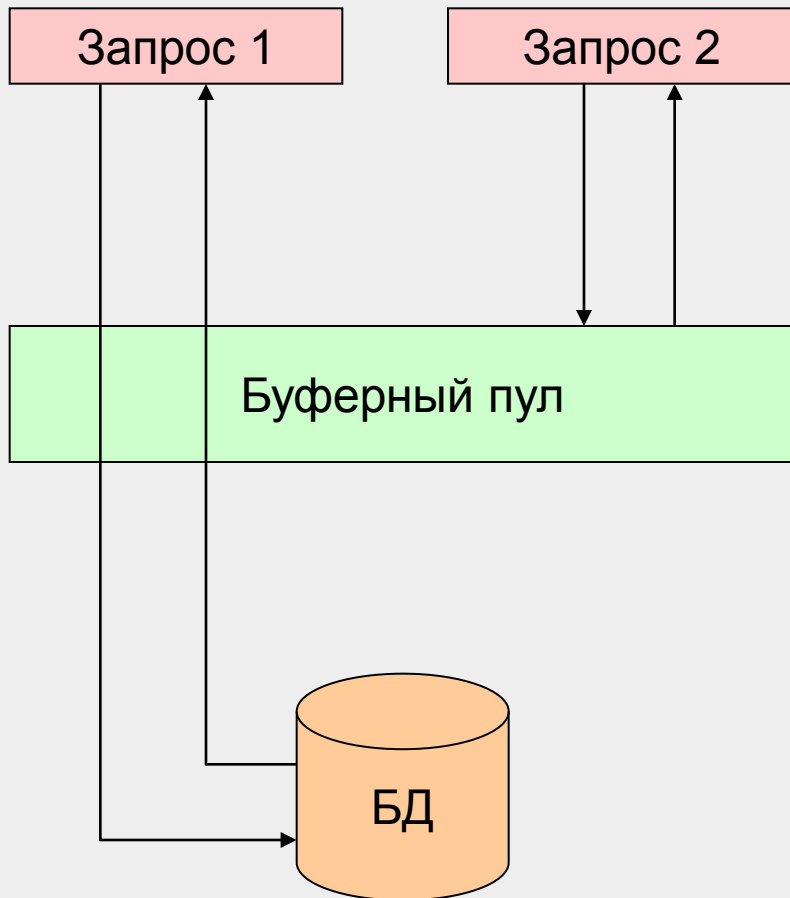


Внутритранзакционный параллелизм





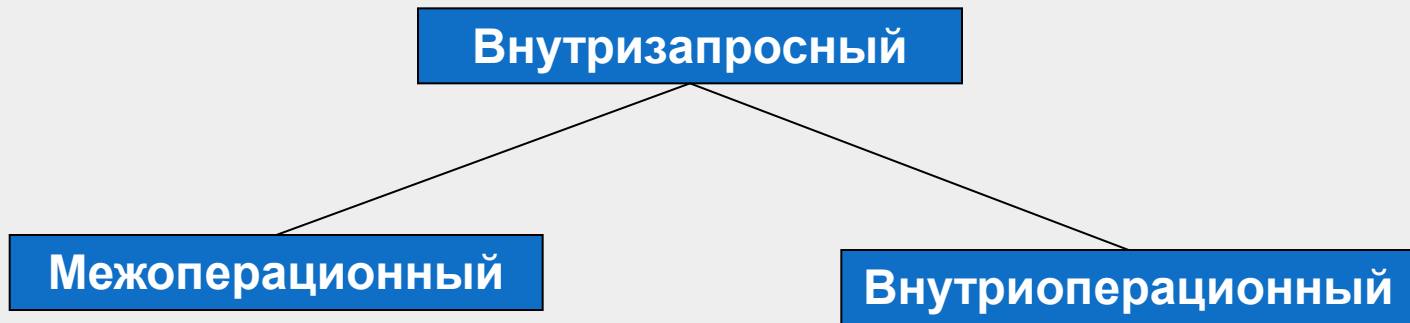
Межзапросный параллелизм



- Степень межзапросного параллелизма ограничена:
 1. Количеством SQL-операторов в транзакции
 2. Ограничениями предшествования между отдельными SQL-операторами
- Межзапросный параллелизм не используется в СУБД

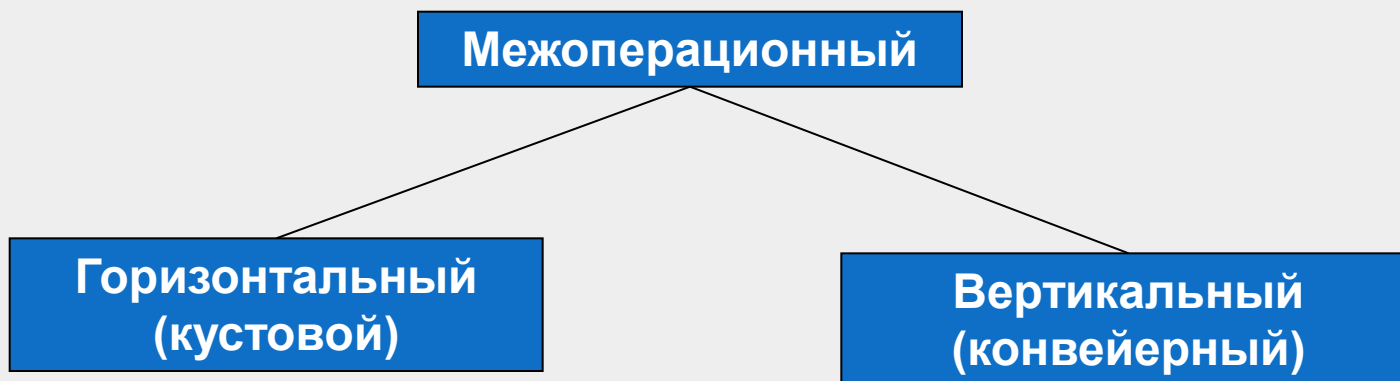


Внутризапросный параллелизм





Межоперационный параллелизм





Кустовой параллелизм

Кустовой (горизонтальный) параллелизм предполагает параллельное выполнение независимых поддеревьев дерева, представляющего план выполнения запроса.

Степень кустового параллелизма ограничена количеством поддеревьев в плане запроса и редко превышает 10.

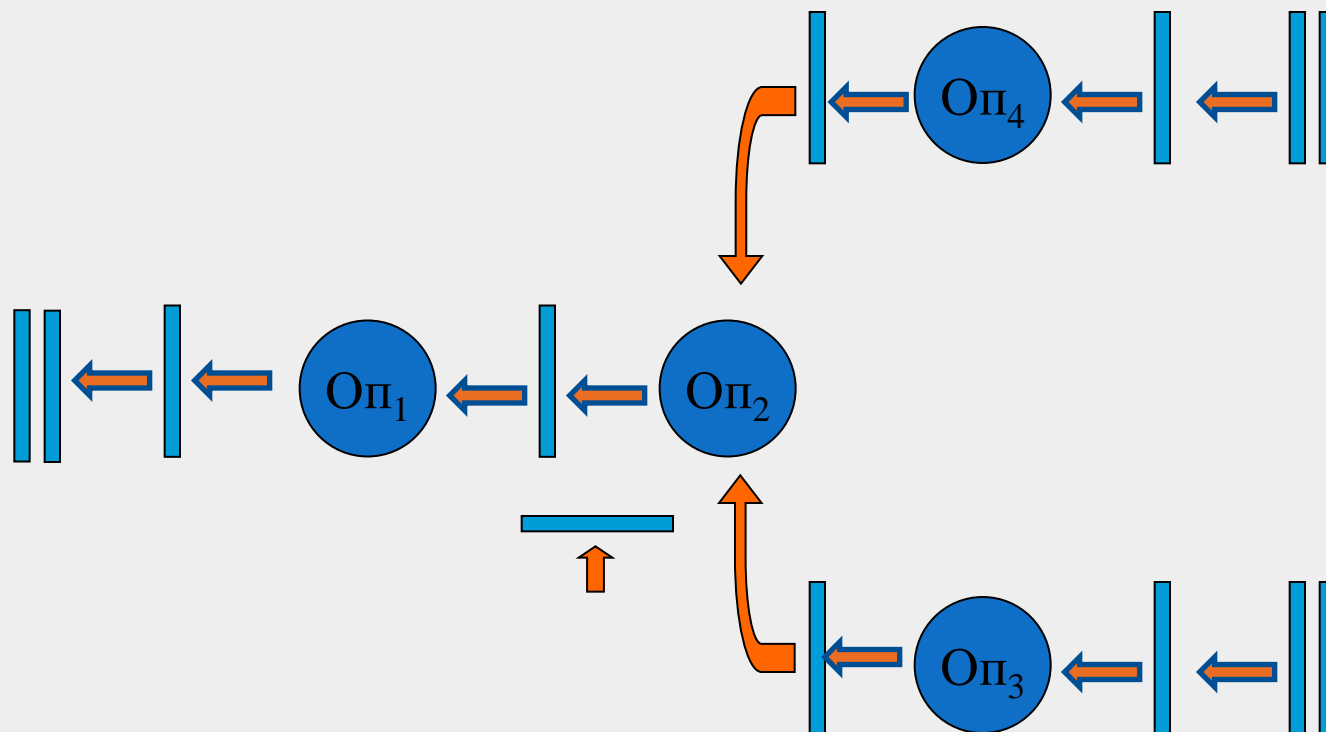


Конвейерный параллелизм

- Синхронный конвейер
- Асинхронный конвейер



Синхронный конвейер





Степень конвейерного параллелизма

- Степень конвейерного параллелизма ограничена: количеством операций, вовлекаемых в конвейер.
- Для реляционных систем баз данных длина конвейера равна высоте дерева запроса и редко превышает 10 операций.



Внутриоперационный параллелизм

Внутриоперационный параллелизм реализуется в основном в форме *фрагментного параллелизма*, предполагающего разбиение входных отношений на фрагменты, которые могут обрабатываться независимо на разных узлах многопроцессорной системы.

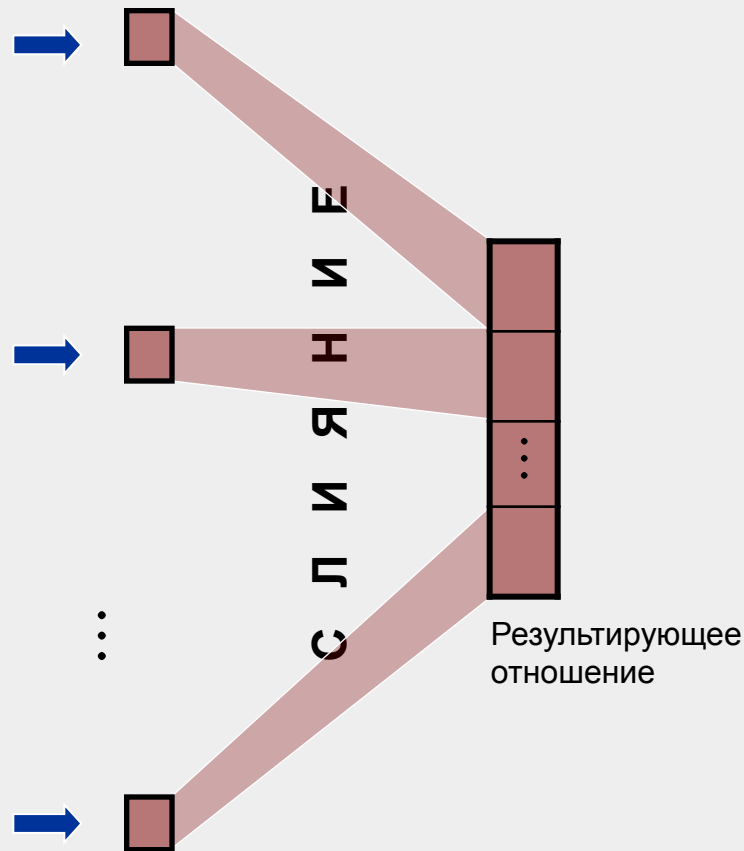
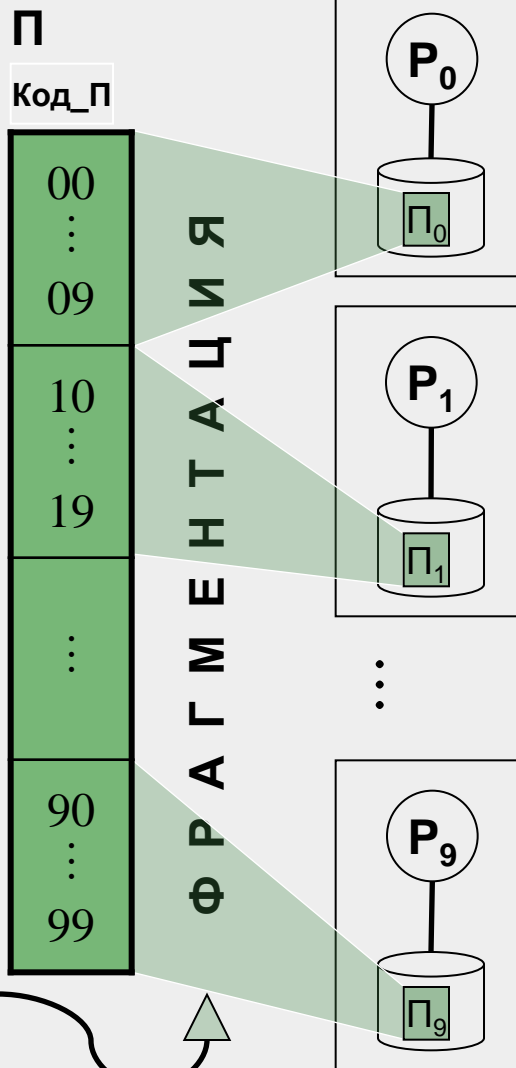
Мы можем рассматривать и другие формы внутриоперационного параллелизма, базирующиеся на делении реляционной операции на подоперации. Однако данные формы параллелизма концептуально ничем не отличаются от рассмотренных выше и на практике большого значения не имеют.



Общая схема параллельной обработки запроса

$$\Pi_i = \{t \mid t \in \Pi, \varphi(t) = i\}$$

$$i = 0, \dots, 9$$



Функция фрагментации:
 $\varphi(t) = (t.\text{Код_П} \text{ div } 10) \text{ mod } 10$



Фрагментация

В реляционных системах баз данных фрагментация подразделяется на *вертикальную* и *горизонтальную*. *Вертикальная фрагментация* подразумевает разбиение отношения на фрагменты по столбцам (атрибутам). *Горизонтальная фрагментация* подразумевает разбиение отношения на фрагменты по строкам (кортежам). Практически все параллельные СУБД, поддерживающие фрагментный параллелизм, используют только горизонтальную фрагментацию. Поэтому везде ниже мы будем рассматривать только горизонтальную фрагментацию.



Степень фрагментного параллелизма

- Теоретически фрагментный параллелизм способен обеспечить сколь угодно высокую степень распараллеливания реляционных операций. Однако на практике степень фрагментного параллелизма может быть существенно ограничена следующими двумя факторами.
 - Во-первых, фрагментация отношения может зависеть от семантики операции. Например, операция соединения одних и тех же отношений по разным атрибутам требует различной фрагментации. Однако повторное разбиение фрагментированного отношения на новые фрагменты и распределение полученных фрагментов по процессорным узлам может быть связано с очень большими накладными расходами.
 - Во-вторых, перекосы в распределении значений атрибутов фрагментации могут привести к значительным перекосам в размерах фрагментов и, как следствие, к существенному дисбалансу в загрузке процессоров.