

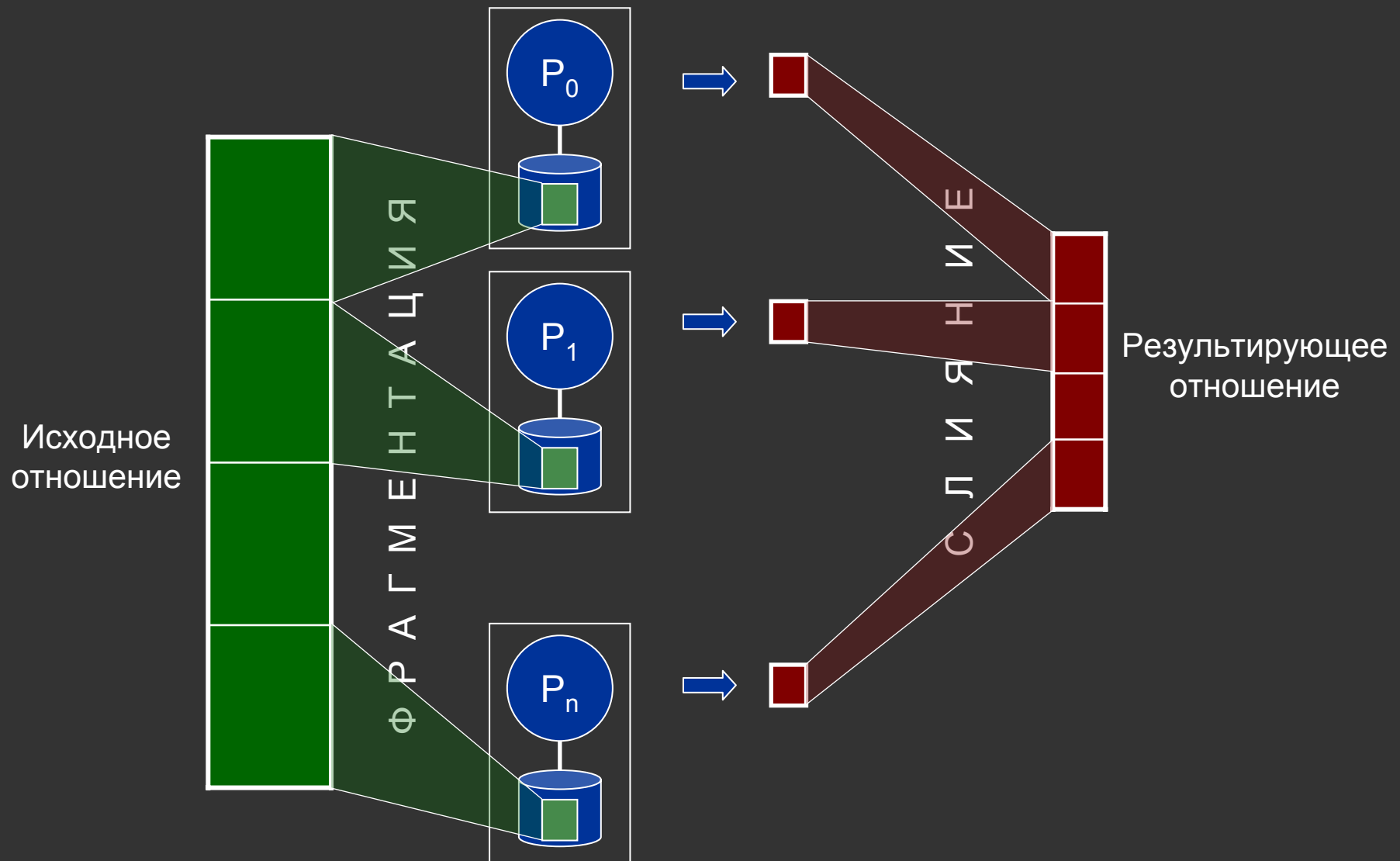
Методы организации систем баз данных для вычислительных кластеров и GRID

*П.С. Костенецкий, А.В. Лепихов,
Л.Б. Соколинский, М.П. Цымблер*

Южно-Уральский государственный университет

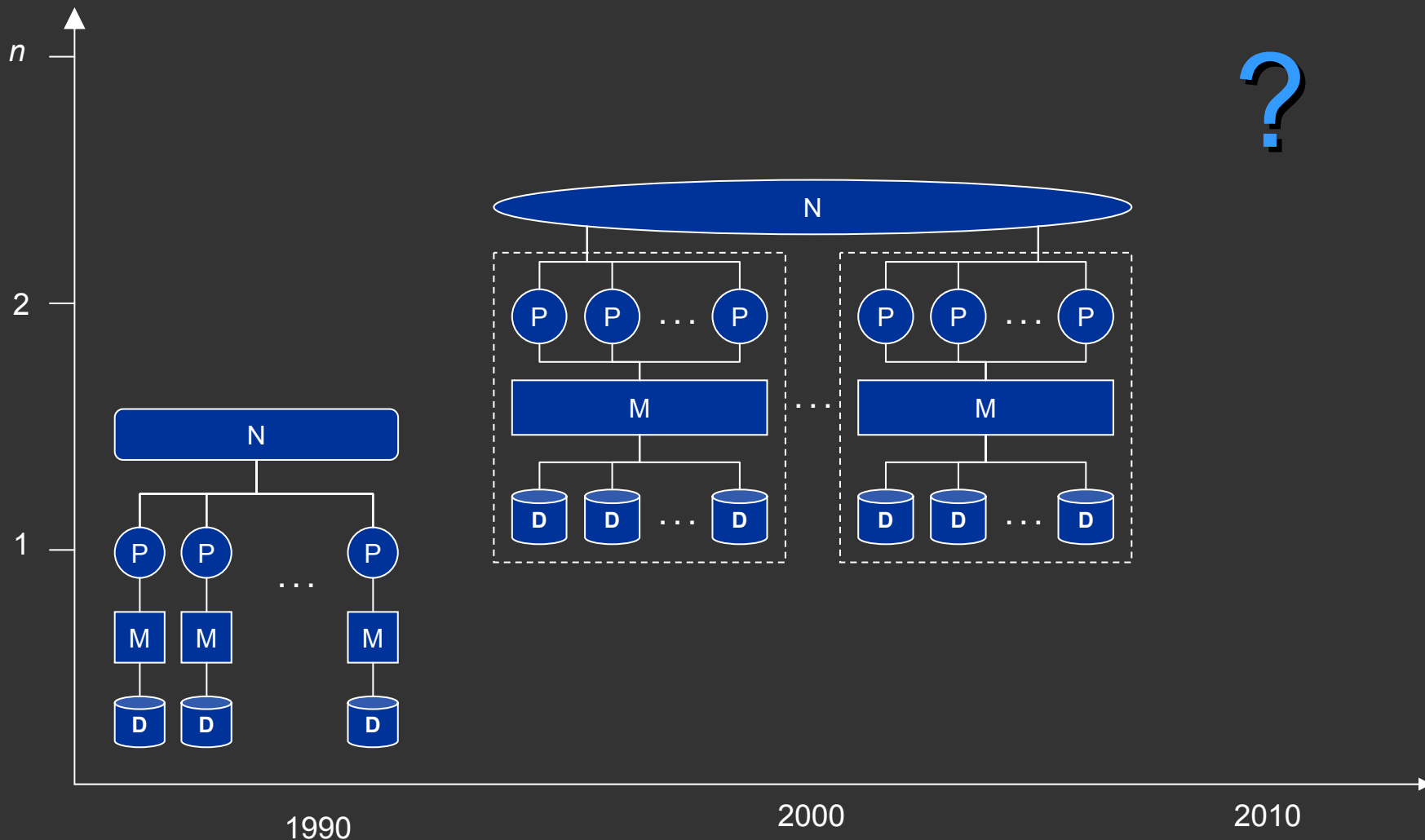
Челябинск

Параллельная система баз данных



Эволюция мультимикропроцессоров баз данных

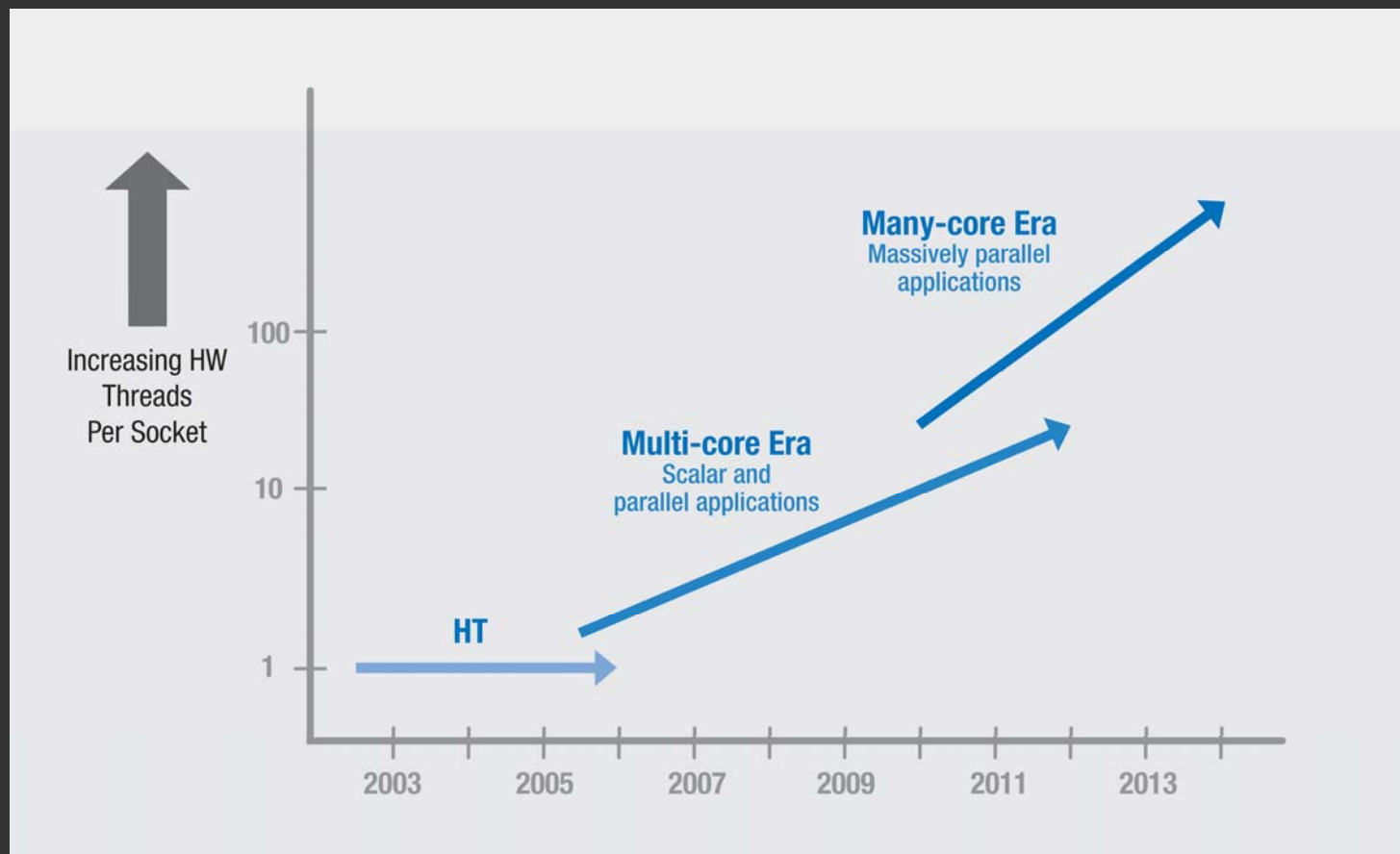
Уровни
иерархии



Предпосылки

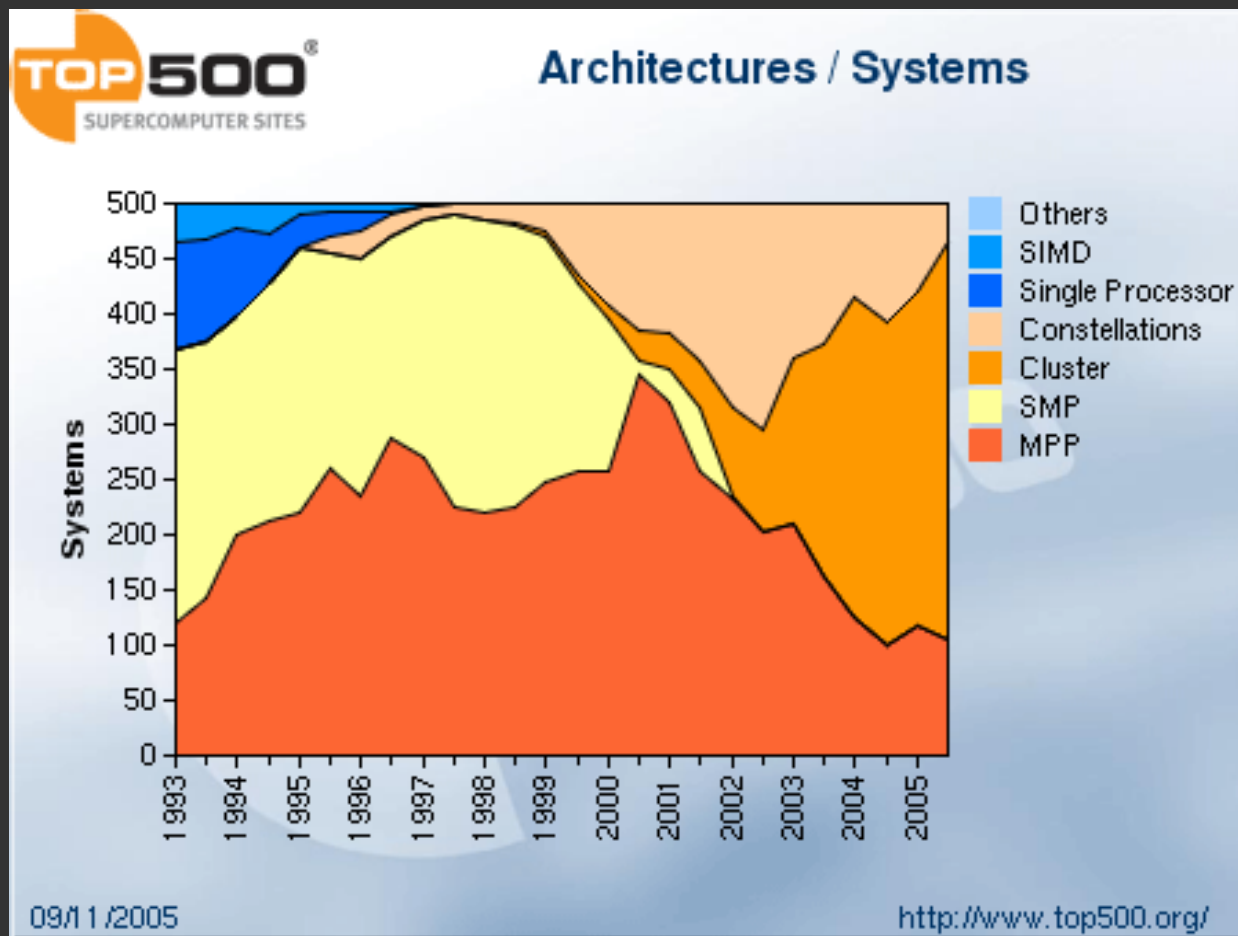
- Многоядерные процессоры
- Кластеры
- Grid

Многоядерные процессоры

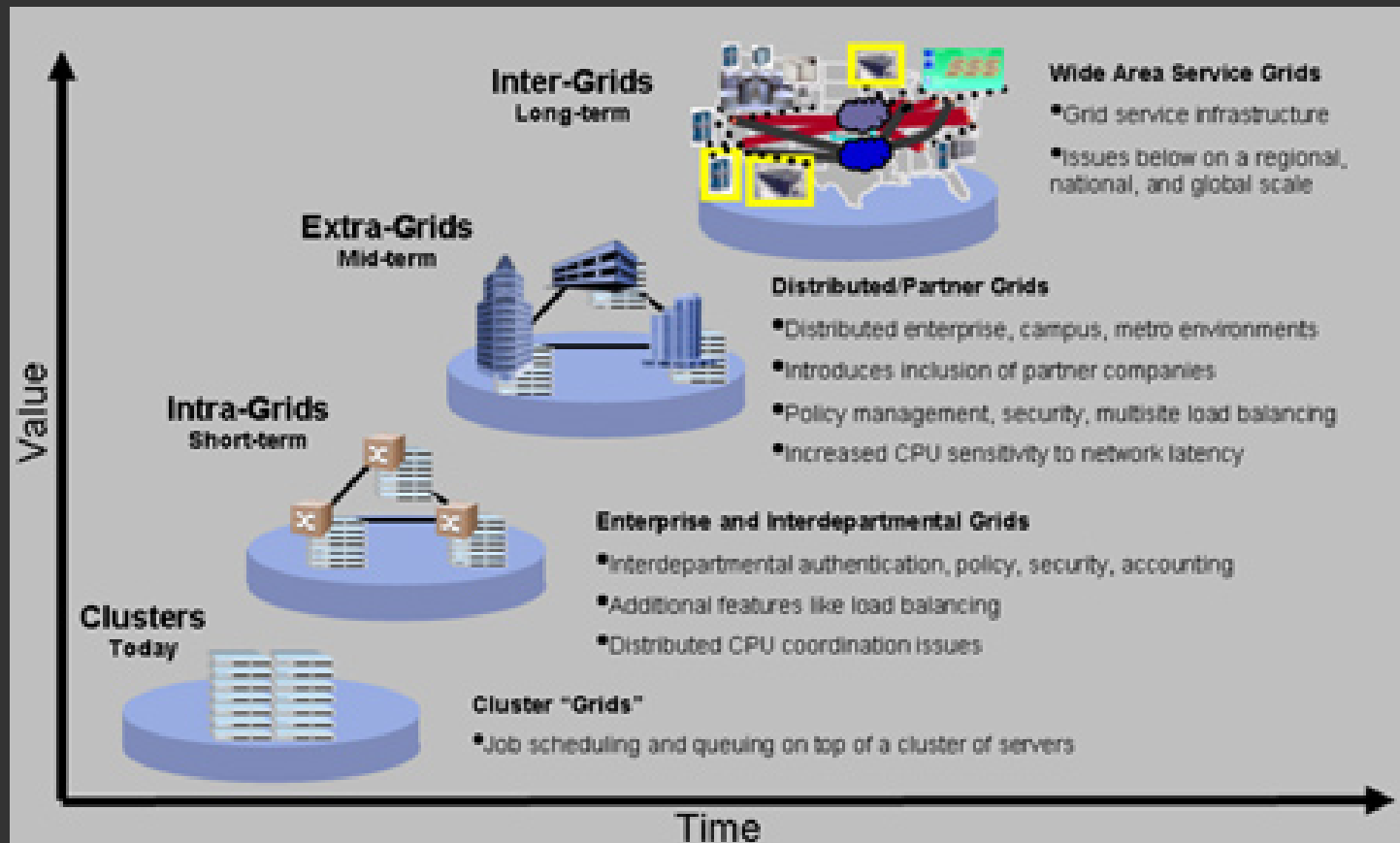


Platform 2015: Intel Processor and Platform Evolution for the Next Decade. White Paper. -Intel Corporation, 2005.

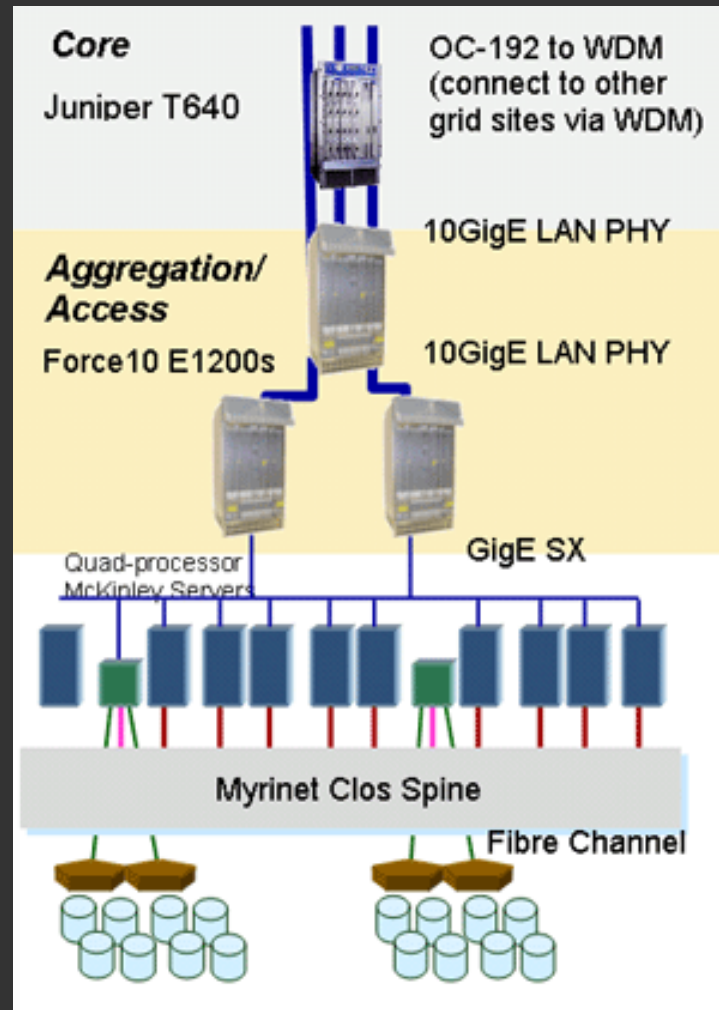
Кластеры



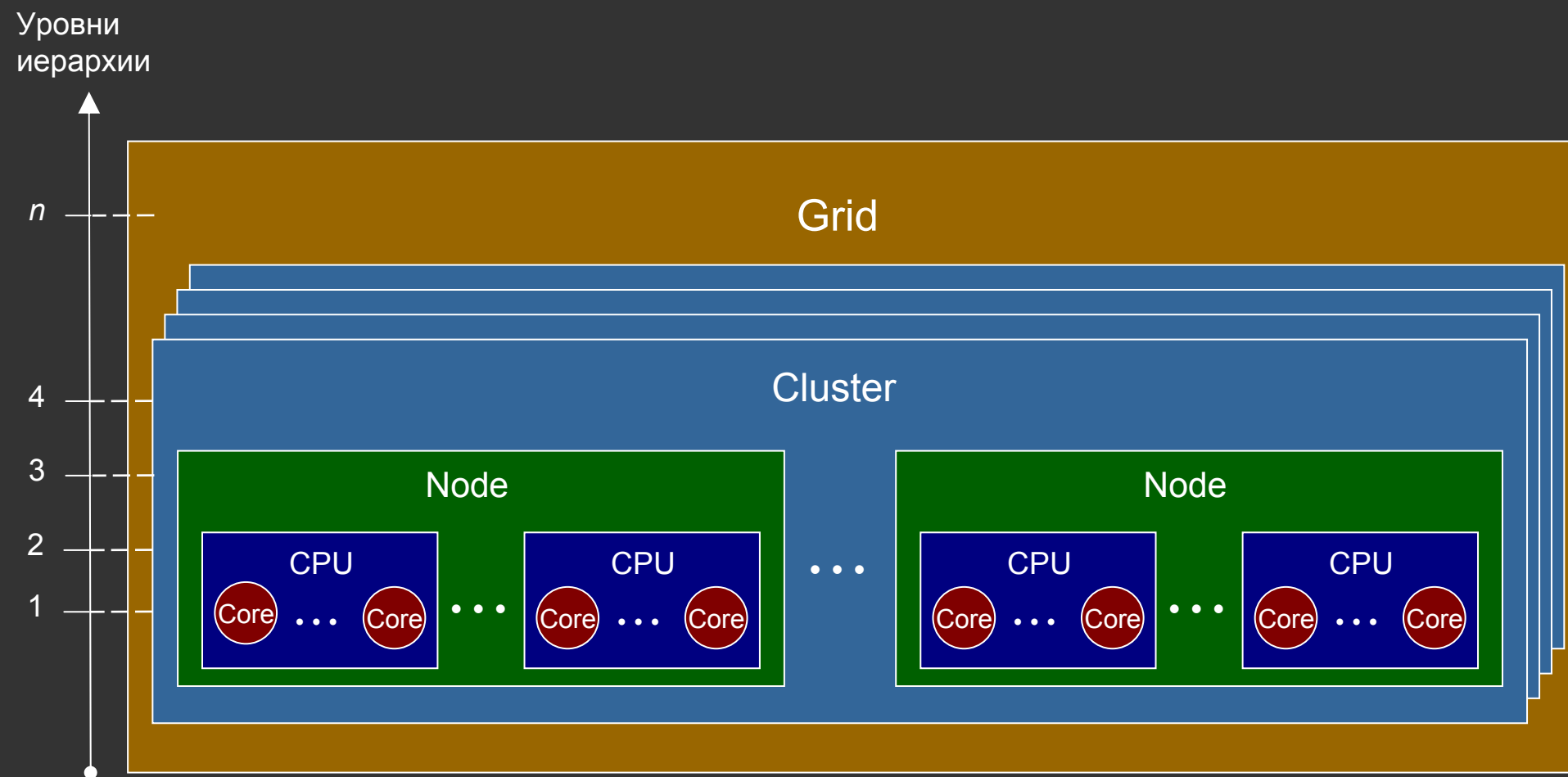
Grid



Сетевая структура TeraGrid



Естественная иерархия



Grid СУБД

Grid СУБД = Параллельная СУБД?

Grid СУБД = Распределенная СУБД?

Сравнительный анализ

	Параллельная СУБД	Распределенная СУБД	Grid СУБД
Контекстная независимость узла	+	-	+
Гомогенность соединительной сети	+	-	-
Фрагментация данных	+	-	+
Репликация данных	-	+	+
Балансировка загрузки	+	-	+

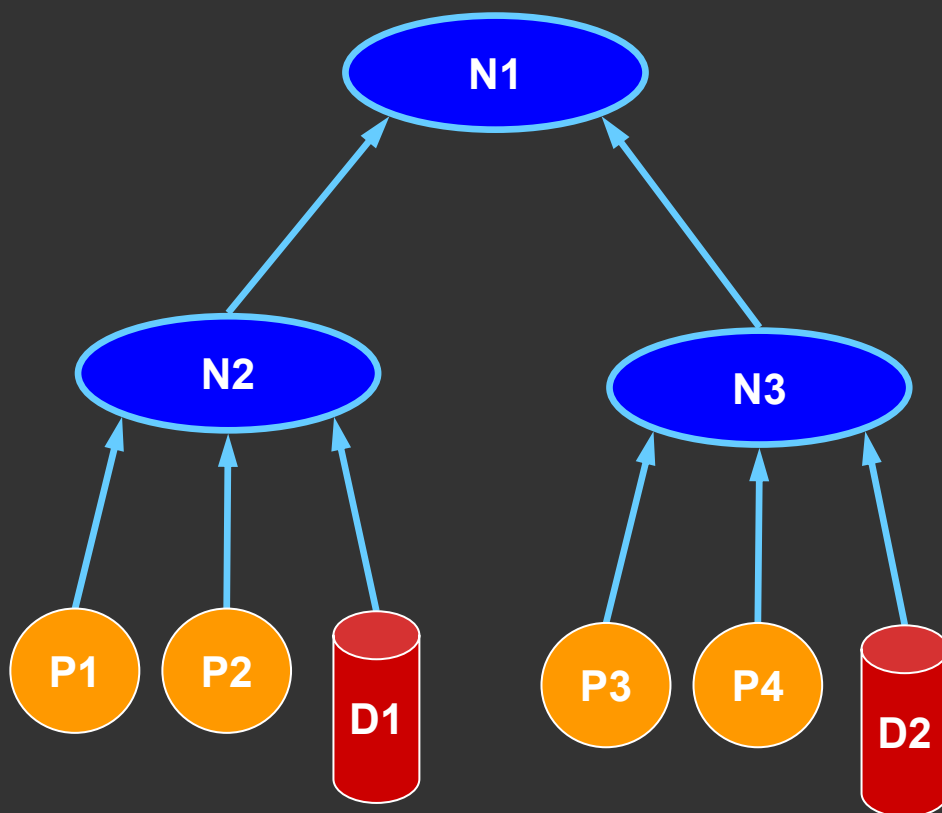
Проблематика

- **Моделирование иерархических систем**
- **Инкапсуляция параллелизма**
- **Балансировка загрузки**

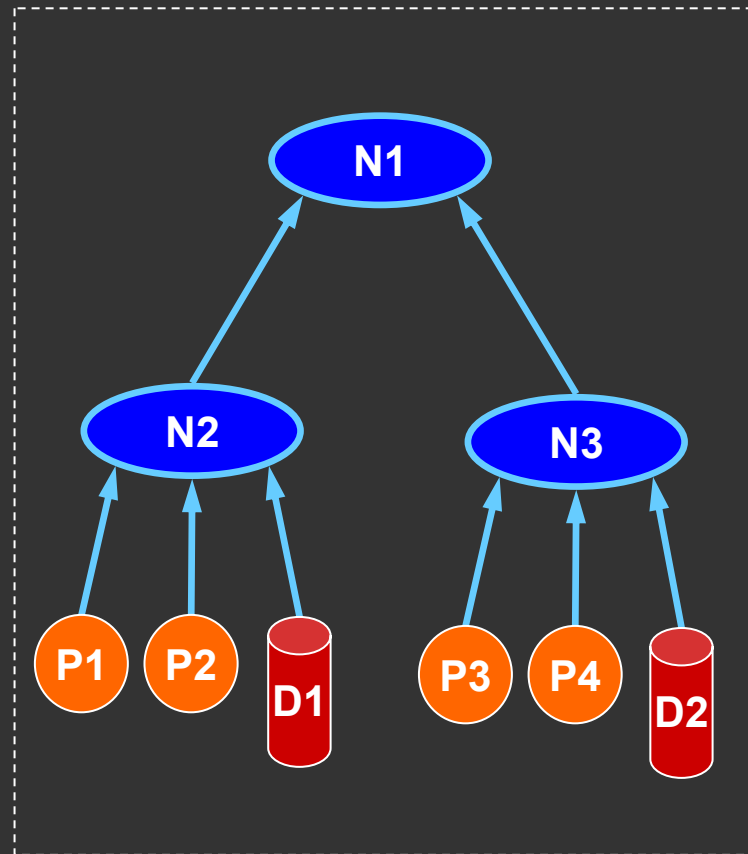
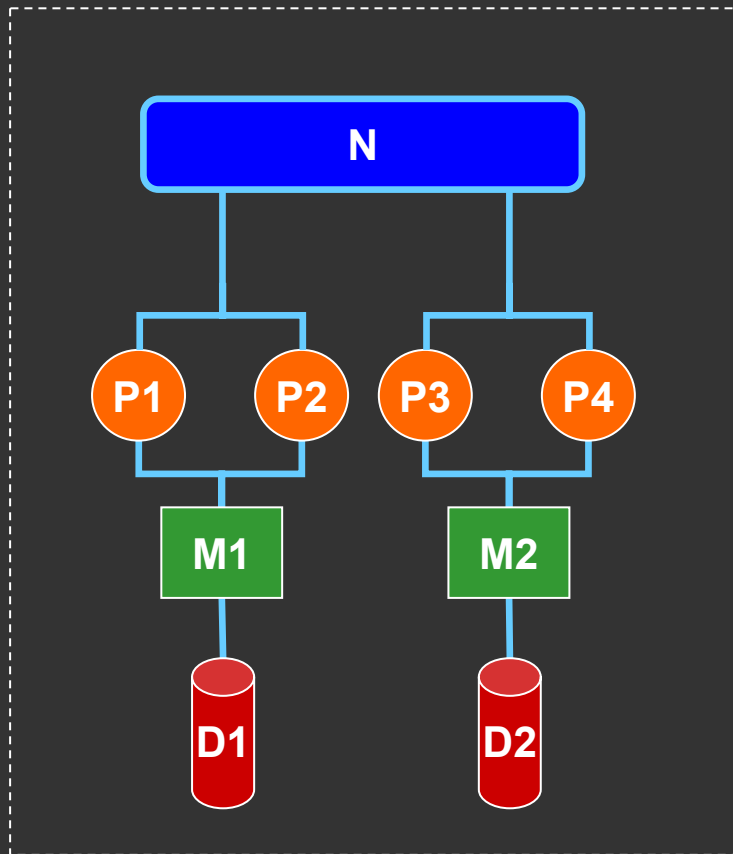
Модель мультипроцессоров баз данных (DMM-модель)

- **Модель аппаратного обеспечения**
- **Модель программного обеспечения**
- **Стоимостная модель**
- **Модель транзакций**

DM-дерево



Представление 2-х уровневой архитектуры



Модель программного обеспечения

- Наименьшей неделимой единицей обработки данных является пакет.
- В качестве пакета может фигурировать один или несколько кортежей.
- Все пакеты имеют одинаковый размер.
- Пакет содержит адрес отправителя, адрес получателя и другую вспомогательную информацию.
- С каждым дисковым модулем и модулем сетевого концентратора ассоциируется *очередь*, в которую помещаются пересылаемые пакеты.
- Любой процессорный модуль может обмениваться данными с любым дисковым модулем.

Такт

Такт определяется как фиксированная последовательность шагов:

- 1) каждый модуль сетевого концентратора обрабатывает все пакеты, ожидающие передачи;
- 2) каждый *активный* процессорный модуль выполняет одну операцию чтения или записи;
- 3) каждый дисковый модуль обрабатывает один пакет из своей очереди.

Процессорный модуль P : операция чтения пакета E с диска D

```
if  $r(P) < s_r$  then  
    поместить  $E$   
    с адресом  $P$   
    в очередь  $D$ ;  
     $r(P)++$ ;  
else  
    wait;  
end if
```

$r(P)$ – количество
незавершенных операций
чтения процессора P

s_r – максимальное
допустимое число
незавершенных операций
чтения

Процессорный модуль Р: операция записи пакета Е на диск D

```
if  $w(P) < s_w$  then
```

```
    поместить пакет  $E$   
    с адресом  $D$  в очередь  
    родительского сетевого  
    концентратора;
```

```
     $w(P) ++;$ 
```

```
else
```

```
    wait;
```

```
end if
```

$w(P)$ – количество
незавершенных
операций записи
процессора Р

s_w – максимальное
допустимое число
незавершенных
операций записи

Модуль сетевого концентратора N :

пересылка сообщений

```

Извлечь пакет  $E$  из очереди  $N$ ;
if  $\alpha(E) \notin T(N)$  then
    Поместить  $E$  в очередь  $F(N)$ ;
else
    Найти максимальное поддереву  $U$ 
    дерева  $T(N)$ , содержащее  $\alpha(E)$ ;
    if  $T(\alpha(E)) = U$  then
        if  $\alpha(E) \in \mathfrak{P}$  then
             $r(\alpha(E)) --$  ;
        else
            Поместить  $E$  в очередь  $\alpha(E)$ ;
        end if
    else
        Поместить  $E$  в очередь  $R(U)$ ;
    end if
end if

```

E – пакет

$\alpha(E)$ – адресат пакета E

$T(N)$ – поддереву с корнем N

$F(N)$ – родительский модуль
узла N

\mathfrak{P} – множество процессорных
модулей

$r(P)$ – Количество
незавершенных
операций чтения
процессора P

$R(U)$ – корень поддереву U

Дисковый модуль D : операция чтения-записи

Извлечь пакет E из очереди D ;

if $\alpha(E) \in \mathfrak{D}$ **then**

$w(\beta(E)) --$;

else

 Поместить E в очередь родительского узла;

end if

$\alpha(E)$ – адресат пакета

$\beta(E)$ – отправитель пакета

\mathfrak{D} – множество дисковых модулей

Стоимостная модель

С каждым модулем M связывается коэффициент *трудоемкости*:

$$h_M \in \mathbb{R}, \quad 1 \leq h_M < +\infty$$

Для процессорных модулей:

$$h_P = 1$$

Функция помех

Модуль сетевого концентратора за один такт может передавать несколько пакетов, поэтому для каждого модуля сетевого концентратора $N \in \mathbf{N}$ мы вводим функцию помех:

$$f_N(m_i^N) = e^{\frac{m_i^N}{\delta_N}}$$

m_i^N - число пакетов, проходящих через N на i -том такте;

$\delta_N > 1$ – масштабирующий коэффициент.

Таким образом, время, требуемое модулю сетевого концентратора N для выполнения i -того такта, вычисляется по формуле

$$t_i^N = h_N f_N(m_i^N), \quad \forall N \in \mathfrak{N}$$

Общее время

Общее время работы системы, затраченное на обработку смеси транзакций в течении k тактов, вычисляется по формуле:

$$t = \sum_{i=1}^k \max(\max_{N \in \mathfrak{N}} (t_i^N), H_{\mathfrak{D}})$$

Δ – множество дисковых модулей

\mathfrak{N} – множество модулей сетевых концентраторов

$$H_{\mathfrak{D}} = \max_{D \in \mathfrak{D}} (h_D)$$

Модель транзакций

Транзакция T_i моделируется путем задания двух групп процессов:

- группа ρ_i - *читающие* процессы,
- группа ω_i - *пишущие* процессы.

Множество процессов моделирующих выполнение смеси из m транзакций процессорного модуля:

$$\Phi = \bigcup_{i=1}^m (\rho_i \cup \omega_i)$$

Вероятность срабатывания процесса

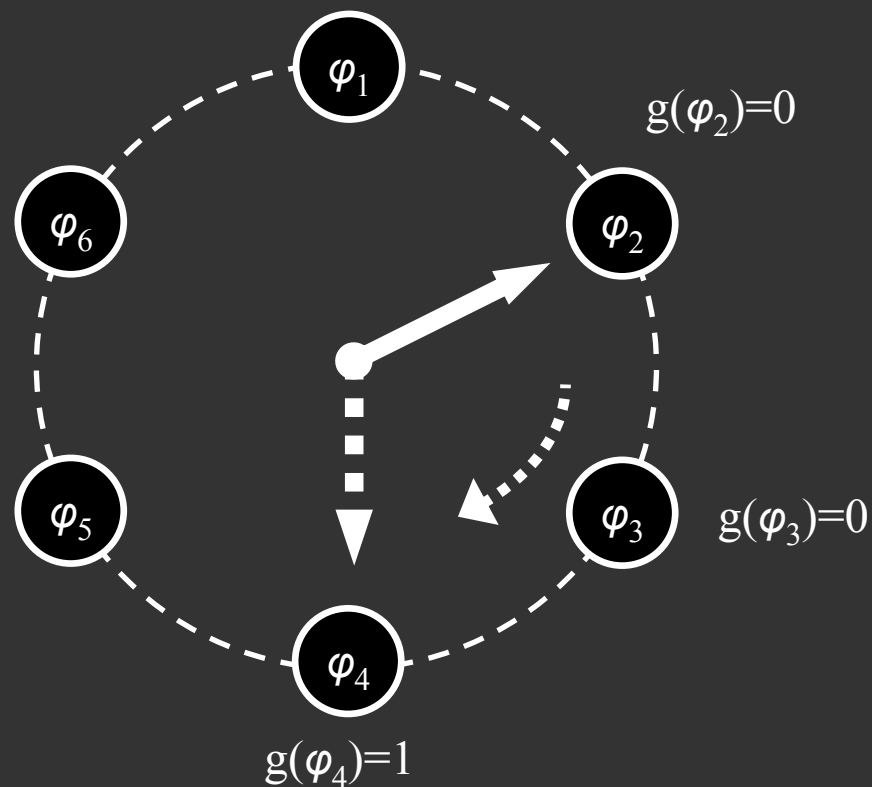
Для каждого процесса $\varphi \in \Phi$ задается вероятность срабатывания p_φ и определяется *функция активности*:

$$g(p_\varphi) = G$$

Функция активности - функция дискретной случайной величины G , закон распределения которой задаётся приведенным рядом распределения:

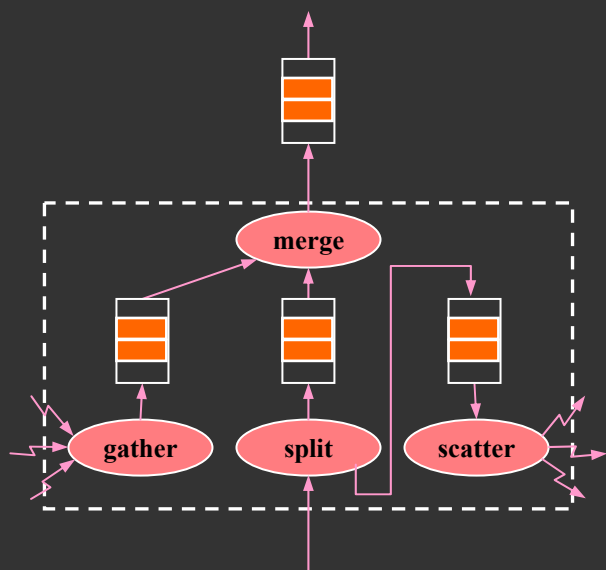
G	1	0
Вероятность	p_φ	$1-p_\varphi$

Выбор активного процесса



Инкапсуляция параллелизма

Структура оператора
Exchange



Дерево запроса
 $Q = R \triangleright \triangleleft S$

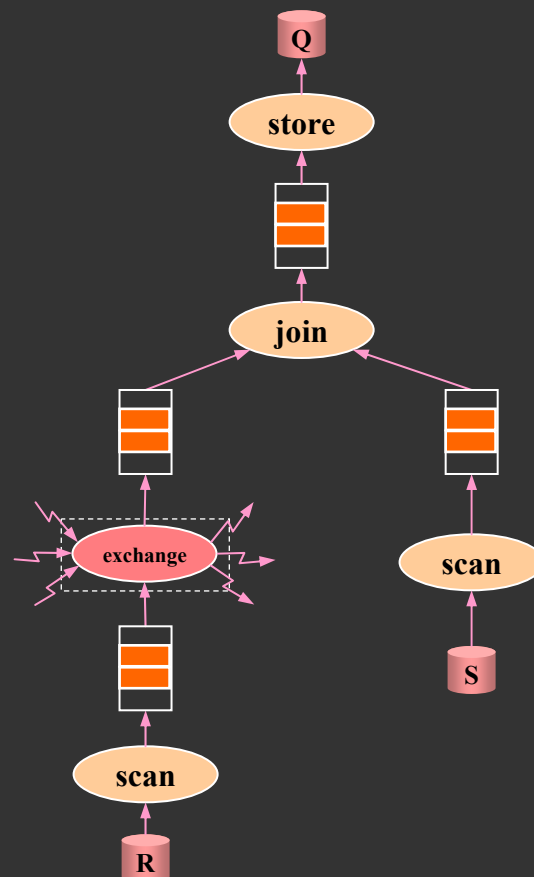
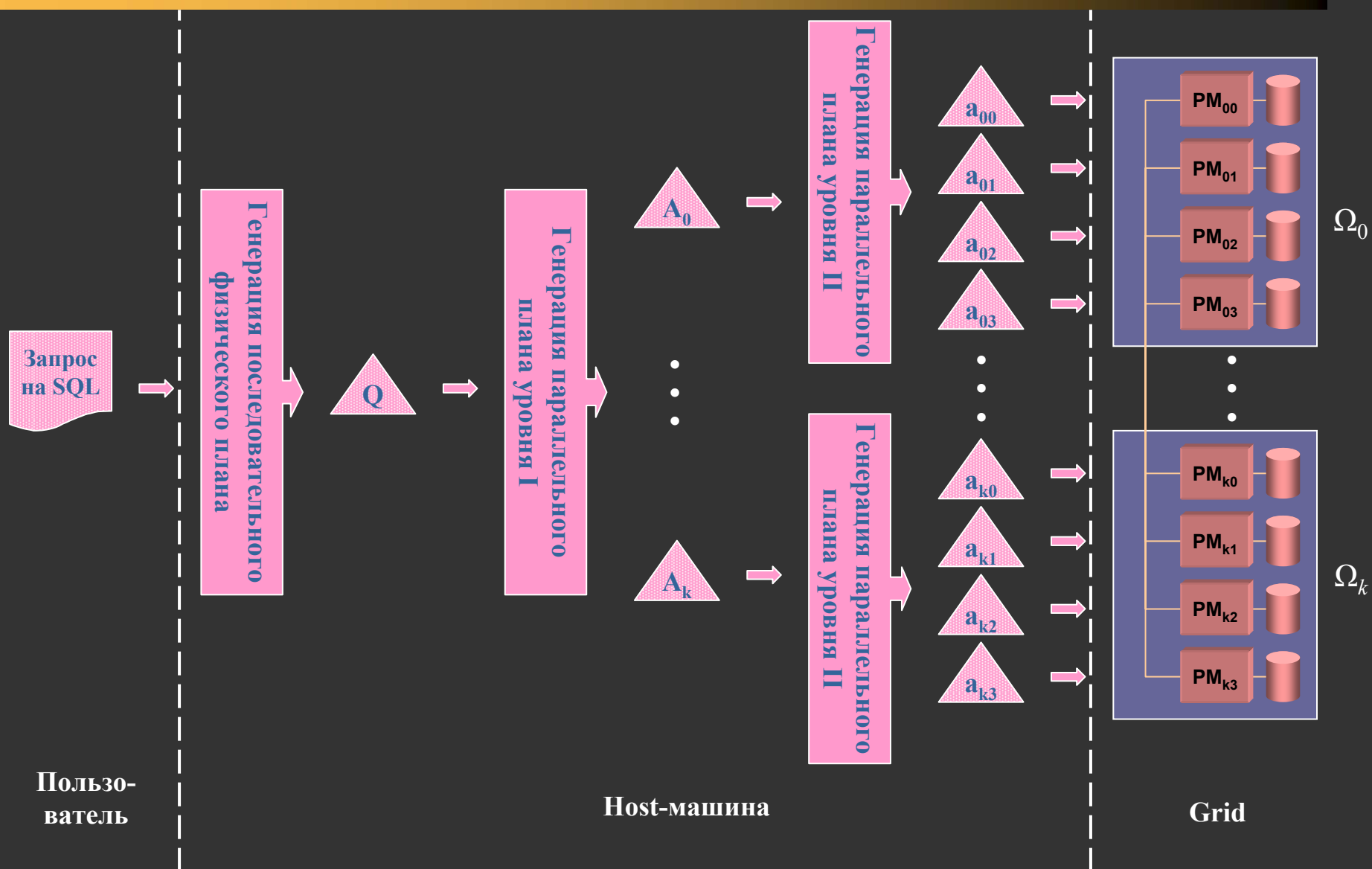
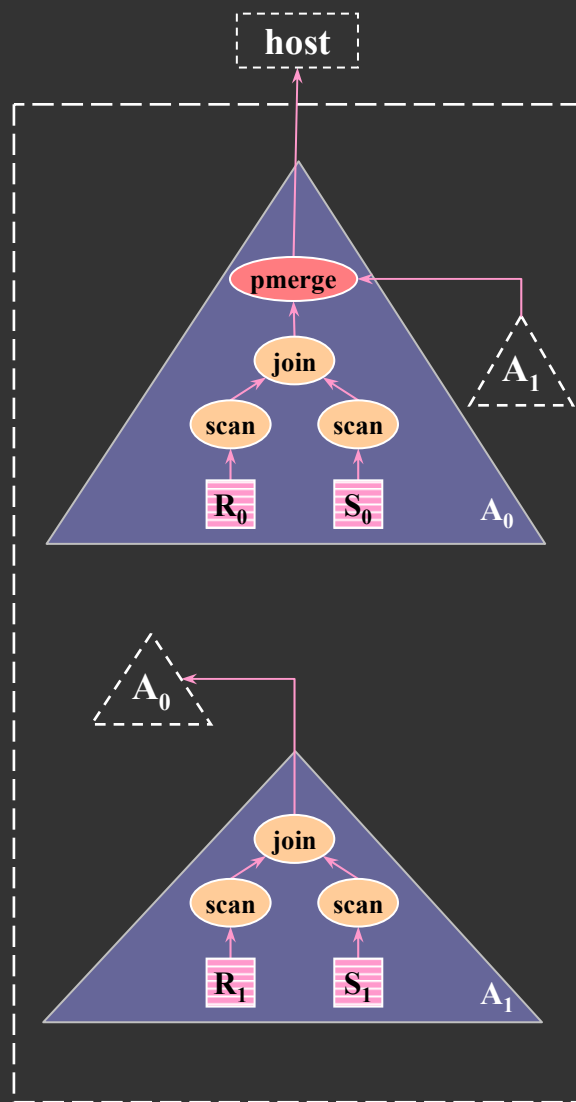
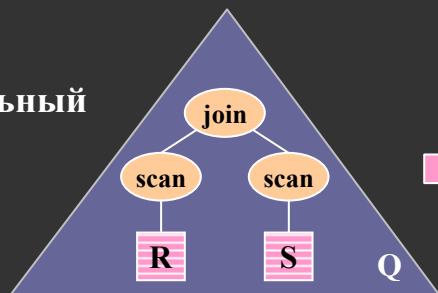


Схема обработки запроса в 2-х уровневой иерархии



Последовательный физический план и параллельный план 1-го уровня для запроса $Q = R \bowtie S$.

Последовательный
план

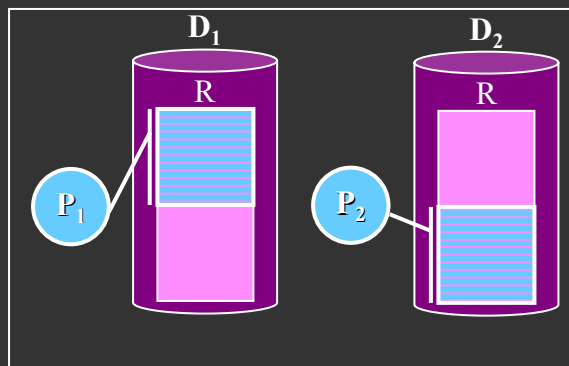
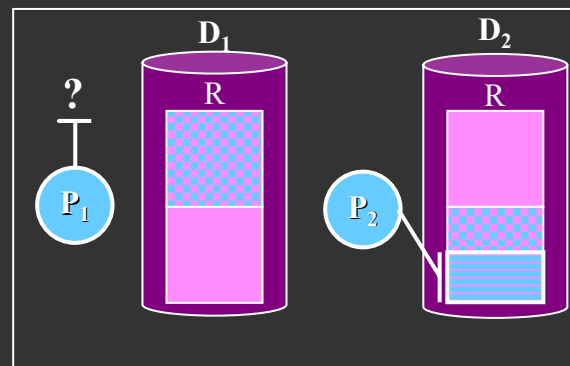
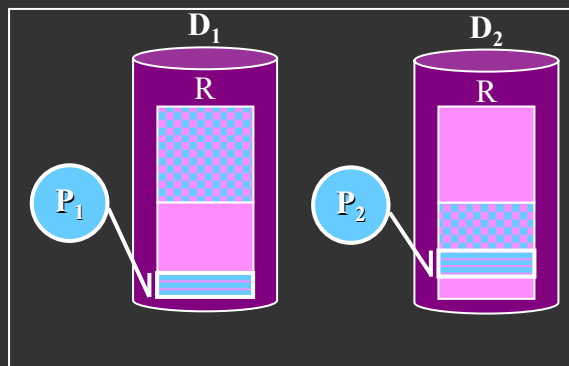
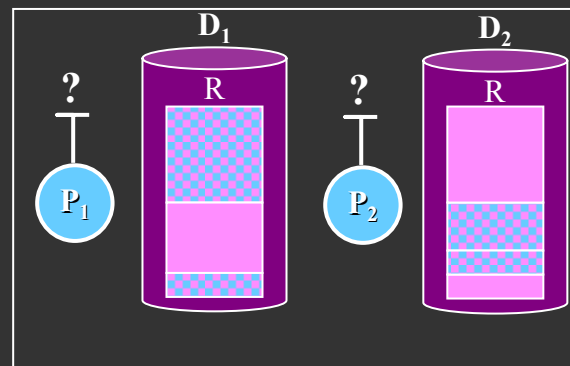


Параллельный план
1-го уровня
(для двух-узлового
кластера)



\sim – балансировка загрузки.

Балансировка загрузки


 t_0

 t_1

 t_2

 t_3


- назначено для обработки



- обработано

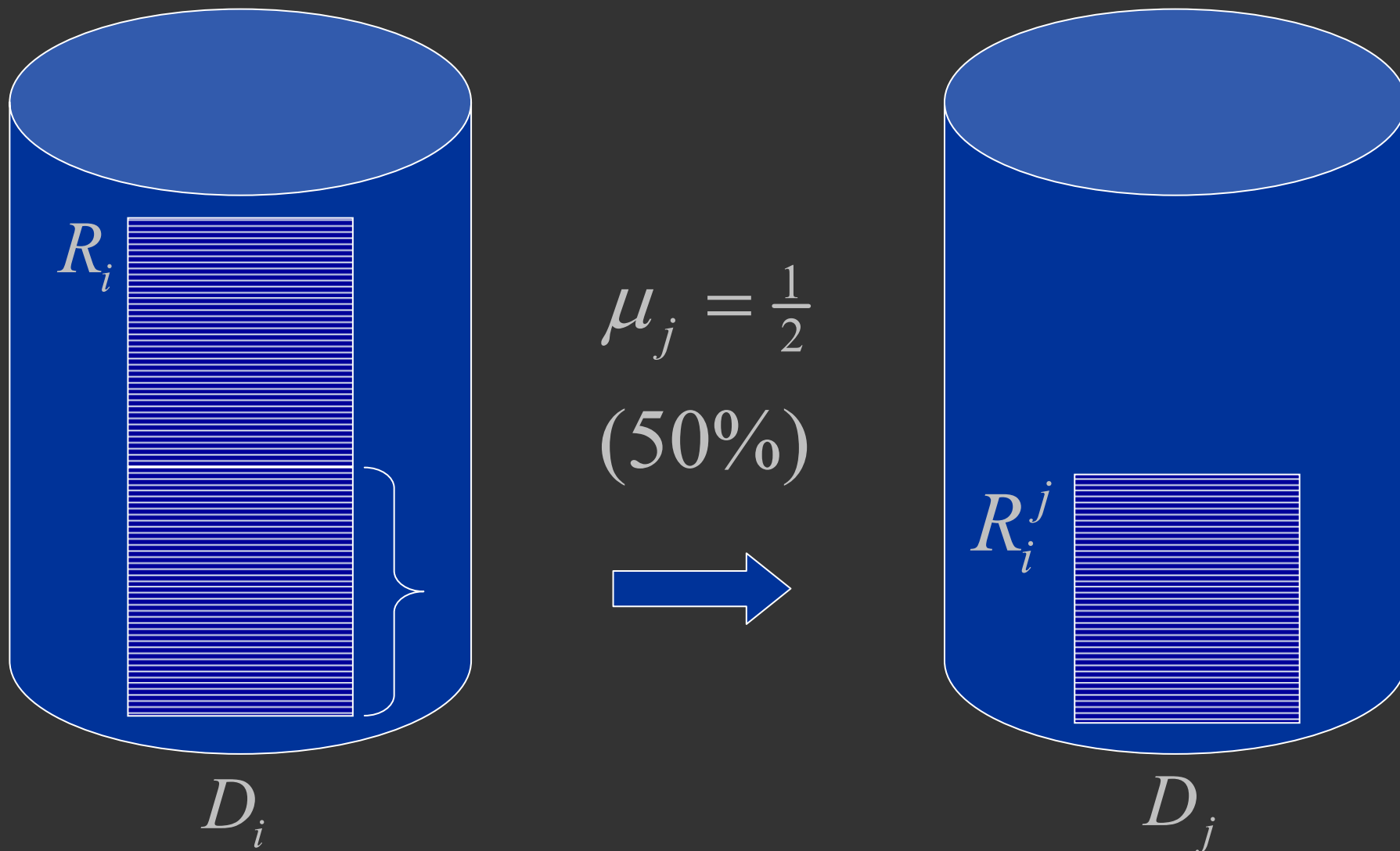
Распределение данных

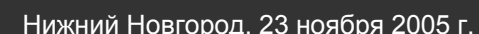
- Отношение (таблица) разбивается на фрагменты, располагающиеся на различных дисках
- Фрагмент делится на сегменты, между которыми определено отношение порядка
- С каждым фрагментом связывается константа s , определяющая длину его сегментов в кортежах (записях):
 - все сегменты, кроме последнего, имеют длину s ;
 - последний сегмент имеет длину $\leq s$
- Сегмент является наименьшей единицей репликации

Зеркалирование

- Фрагмент может иметь несколько (возможно неполных) зеркальных копий, называемых репликами, которые располагаются на других дисках
- На каждом диске может находиться не более одной реплики данного фрагмента
- Содержимое реплики однозначно определяется *коэффициентом зеркалирования μ* , назначенным диску, на котором хранится реплика

Построение реплики





Реализация

- Прототип «Омега»
<http://omega.susu.ru/>
- Интеграция параллелизма в MySQL

Электронный учебный курс <http://pddb.susu.ru/>

Курс "Параллельные системы баз данных" - Microsoft Internet Explorer

Файл Правка Вид Избранное Сервис Справка

Адрес: <http://pddb.susu.ru/>

Параллельные системы баз данных

Л.Б. Соколинский, М.Л. Цымблер



Электронный учебный курс
(учебное пособие)



Челябинск: ЮУрГУ, 2004



Sponsored by
intel



РАБОТА ВЫПОЛНЕНА
ПОДДЕРЖКА



Аннотация



Программа курса  


Тексты лекций  


Слайды презентации  


Задания к лабораторным работам  


Внутренние спецификации  


Описание комплексных тестов  

Автономные тесты 

Поддерживающая библиотека 

Текст головного модуля 

Файлы данных для комплексного тестирования 

Исходные тексты поддерживающей библиотеки (ограниченный доступ) 

Готово

Internet

5 мая 2005 г.