БАЗЫ ДАННЫХ

Лекция 13 Оптимизация запросов

FILLFACTOR

- FILLFACTOR = n задает заполнение в процентах каждой страницы индекса
- При n = 100 нет свободного места для вставки новых строк - только для статических таблиц.
- При n = 0 страницы листьев индекса заполняются полностью, а каждая из промежуточных страниц содержит свободное место для одной записи

PAD_INDEX

- PAD_INDEX указывает, что значение параметра FILLFACTOR применяется как к страницам индекса, так и к страницам данных в индексе

Фрагментация индекса

- Внутренняя фрагментация - объем данных, хранящихся в каждой странице
- Внешняя фрагментация - нарушение логического порядка страниц

sys.dm_db_index_physical_stat

```
DECLARE @db_id INT;
DECLARE @tab_id INT;
DECLARE @ind_id INT;

SET @db_id = DB_ID('TMP1_BSTU');
SET @tab_id = OBJECT_ID('AUDITORIUM');

SELECT avg_fragmentation_in_percent, avg_page_space_used_in_percent
FROM sys.dm_db_index_physical_stats
(@db_id, @tab_id, NULL, NULL, NULL)
```

Results | Messages

avg_fragmentation_in_percent	avg_page_space_used_in_percent
0	NULL

Представления индексов

- sys.indexes
- sys.index_columns
- sp_helpindex
- sys.dm_db_index_usage_stats
- sys.dm_db_missing_index_details

Изменение индекса

- ALTER INDEX
- ALLOW_ROW_LOCKS, ALLOW_PAGE_LOCKS, IGNORE_DUP_KEY
- REBUILD - пересоздание индекса
- REORGANIZE - реорганизация страниц листьев индекса
- DISABLE - отключение индекса

REBUILD

- Параметр REBUILD применяется для пересоздания индексов
- ALL - все индексы таблицы

REBUILD

```
alter index #EXPLORE_TKEY on #EXPLORE rebuild with (online = off);
go

----  данные о фрагментации ----------------------------------
select name [Индекс], avg_fragmentation_in_percent [Фрагментация (%)]
from sys.dm_db_index_physical_stats(DB_ID(N'TEMPDB'),OBJECT_ID(N'#EXPLORE'), NULL, NULL, NULL) ss
     join sys.indexes ii on ss.object_id = ii.object_id and ss.index_id = ii.index_id
where name is not null;
go
```

Индекс	Фрагментация (%)
#EXPLORE_TKEY	0

REORGANIZE

- REORGANIZE задает реорганизацию страниц листьев индекса, чтобы физический порядок страниц совпадал с их логическим порядком — слева направо

REORGINEZE

```
alter index #EXPLORE_TKEY on #EXPLORE reorganize;
go

----  данные о фрагментации ----------------------------------
select name [Индекс], avg_fragmentation_in_percent [Фрагментация (%)]
from sys.dm_db_index_physical_stats(DB_ID(N'TEMPDB'),OBJECT_ID(N'#EXPLORE'), NULL, NULL, NULL) ss
     join sys.indexes ii on ss.object_id = ii.object_id and ss.index_id = ii.index_id
where name is not null;
```

Индекс	Фрагментация (%)
#EXPLORE_TKEY	3,94736842105263

DROP INDEX

- DROP INDEX
- Для кластеризованного индекса - потребуется пересоздать все некластеризованные индексы
- MOVE TO - куда переместить строки данных, находящиеся в страницах листьев кластеризованного индекса - файловая группа по умолчанию или именованная файловая группа
- Нельзя для PRIMARY KEY и UNIQUE

Вычисляемые столбцы

- Вычисляемым называется столбец таблицы, в котором сохраняются результаты вычислений данных таблицы.
  - Виртуальный
  - Постоянный

Вычисляемые столбцы

```
CREATE TABLE Orders
(orderid INT NOT NULL,
price MONEY NOT NULL,
quantity INT NOT NULL,
orderdate DATETIME NOT NULL,
total AS price * quantity,
shippeddate AS DATEADD (DAY, 7, orderdate));

CREATE CLUSTERED INDEX i1 ON orders (total);
```

Индексированные представления

- Создается представление CREATE VIEW с предложением SCHEMABINDING
- Создается кластеризованный индекс для этого представления – в инструкции CREATE INDEX вместо имени таблицы указывается имя представления

Индексированные представления

- все используемые в представлении функции должны быть детерминированными
- представление должно ссылаться только на базовые таблицы
- представление и базовые таблицы должны иметь одного владельца и принадлежать к одной и той же базе данных
- инструкция SELECT в представлении не должна содержать DISTINCT, UNION, TOP, ORDER BY, MIN, MAX, COUNT, OUTER, SUM (для выражений, допускающих значения NULL), подзапросы или производные таблицы

Индексированные представления

```
create view Aud(AUDITORIUM_NAME, AUDITORIUM_CAPACITY)
WITH SCHEMABINDING
as
select AUDITORIUM_NAME, AUDITORIUM_CAPACITY from dbo.AUDITORIUM;
go

SELECT objectproperty(object_id('dbo.AUDITORIUMS'), 'IsIndexable');
SELECT objectproperty(object_id('dbo.Aud'), 'IsIndexable');
```

%

Results | Messages

(No column name)
0

(No column name)
1

Индексированные представления

- запросы, которые обрабатывают большое количество строк и содержат операции соединения или агрегатные функции
- операции соединения и агрегатные функции, которые часто выполняются в одном или нескольких запросах

Колоночные индексы

- Строки таблиц сохраняются в страницах - построчное хранение - row store
- Данные группируются и сохраняются по одному столбцу - постолбцовое хранение - column store
- реализуется посредством использования колоночного индекса - columnstore index

Колоночные индексы

- Система извлекает только требуемые столбцы
- Оптимальное сжатие значений.
- Значительное ускорение времени выполнения запросов с особыми характеристиками

```
CREATE NONCLUSTERED COLUMNSTORE INDEX cs_index1
ON AUDITORIUM (AUDITORIUM_NAME, AUDITORIUM_CAPACITY, AUDITORIUM_TYPE);
```

Колоночные индексы

- Таблица с колоночным индексом доступна только для чтения
- Колоночные индексы поддерживают только типы данных CHAR, VARCHAR, INT, DECIMAL и FLOAT
- Ограничений на кластеризованные и некластеризованныу колоночные индексы – не больше одного некластеризованного колоночного индекса, а кластеризованные колоночные индексы не поддерживаются вообще

Оптимизация запроса

- анализ запроса
- выбор индекса
- выбор порядка выполнения операций соединения
- выбор метода выполнения операций соединения

Селективность и плотность

- Селективность запроса – соотношение количества строк, удовлетворяющих условию, к общему количеству строк в таблице
- Плотность запроса – количество возвращаемых строк запроса

Селективность

- Индекс успешно работает при <= 5%.
- Не нужен индекс при 80% или более

Анализ запроса

- Наличие аргументов поиска
- Использование оператора OR
- Существование критериев соединения

Анализ запроса

- Аргумент поиска — это часть запроса, которая ограничивает промежуточный результирующий набор запроса:
- name = ‘Иванов С.П.'
- capacity >= 100
- name = ‘Иванов С.П.‘ AND idgroup = 512

Анализ запроса

- Нельзя использовать в качестве аргументов поиска:
  - Выражение с оператором отрицания NOT
  - <>
  - Выражение

- NOT IN('d1', 'd2')
- aud_no <> 9031
- capacity * 0.6 > 100

Создание индексов

- SELECT … WHERE column_name – для этого столбца следует создать индекс

- SELECT … WHERE column_name1 AND column_name2 – создать составной индекс по всем столбцам

Индексы при соединении

- Для каждого соединяемого столбца
- Некластеризованный индекс для столбца внешнего ключа

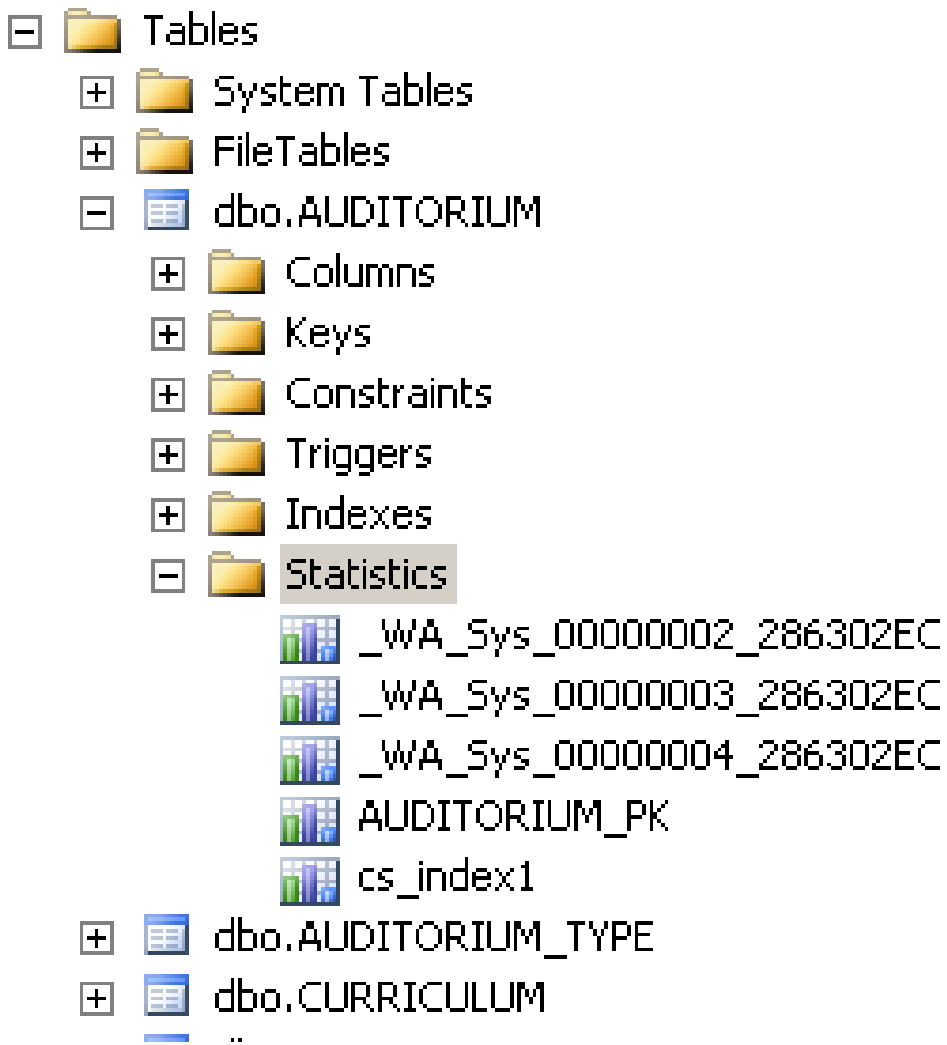
Выбор индексов

- Оптимизатор проверяет селективность выражения с индексированным столбцом, используя статистические данные
- AUTO_CREATE_STATISTICS ON(OFF)

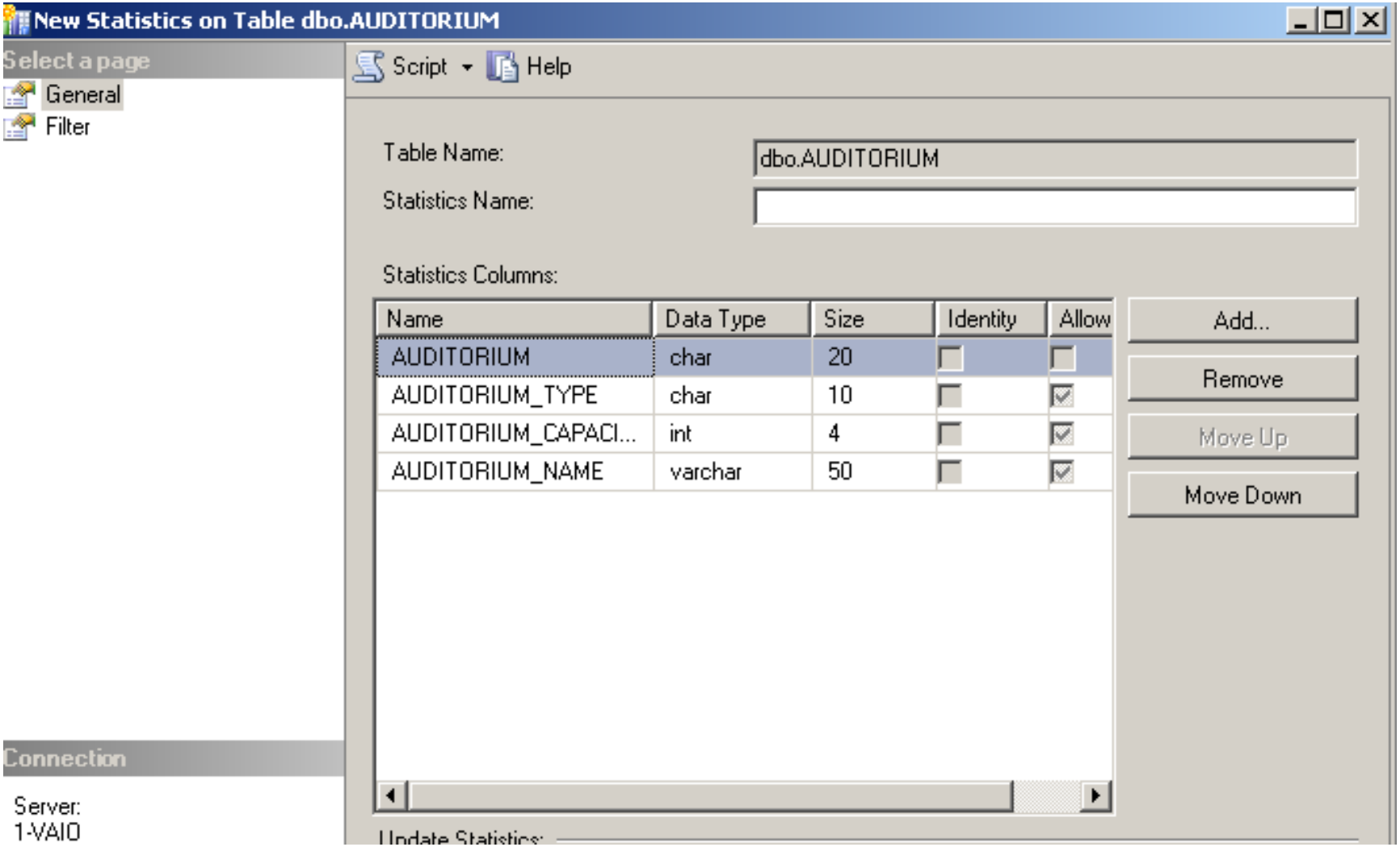
Обход индекса

- Для кучи сначала выполняется обход некластеризованного индекса, а затем извлекается строка, используя идентификатор строки
- Для кластеризованной таблицы, после обхода структуры некластеризованного индекса следует обход структуры кластеризованного индекса таблицы

Сбор статистики



Сбор статистики



Выбор порядка соединения

- Вложенный цикл
- Соединение слиянием
- Соединение хешированием

Вложенный цикл

- Для каждой строки внешней таблицы извлекается и сравнивается каждая строка внутренней таблицы
- Сканирование внешней таблицы и n раз – сканирование внутренней

Соединение слиянием

- Строки соединяемых таблиц должны быть физически упорядочены с использованием значений столбца соединения
- Выполняется сканирование обеих таблиц в порядке столбцов соединения, сопоставляя строки с одинаковыми значениями для столбцов соединения

Соединение хешированием

- Вычисляется хеш для значения соединяемого столбца из меньшей таблицы и сохраняется в определенном сегменте
- Вычисляется хеш для значения соединяемого столбца из бальшей таблицы и сравнивается с хешами на предыдущем этапе
- Совпадающие строки попадают в результирующий набор

Кэширование планов

- При первом выполнении запроса его скомпилированная версия сохраняется в кэше планов — plan cache
- При повторном выполнении запроса проверяется, нет ли для него плана в кэше планов

Просмотр планов кэша

- sys.dm_exec_cached_plans
- sys.dm_exec_query_stats
- sys.dm_exec_sql_text

Просмотр планов

- SET
- среда Management Studio

Инструкция SET

```
---------------------------------
SET SHOWPLAN_TEXT ON;
GO

SELECT * FROM FACULTY  JOIN PULPIT
ON FACULTY.FACULTY = PULPIT.FACULTY
AND FACULTY.FACULTY = 'ИДиП';
GO

SET SHOWPLAN_TEXT OFF;
GO
```

Results | Messages

StmtText
SELECT * FROM FACULTY JOIN PULPIT ON FACULTY...

StmtText
|--Nested Loops(Inner Join)
|--Clustered Index Seek(OBJECT:([TMP1_BSTU].[dbo].[FACULTY].[FACULTY_PK]), SEEK:([TMP1_BSTU].[dbo].[FACULTY].[FACULTY]='ИДиП') ORDERED FORWARD)
|--Clustered Index Scan(OBJECT:([TMP1_BSTU].[dbo].[PULPIT].[PULPIT_PK]), WHERE:([TMP1_BSTU].[dbo].[PULPIT].[FACULTY]='ИДиП'))

Инструкция SET

- |--Nested Loops(Inner Join)
- |--Clustered Index Seek (OBJECT: ([TMP1_BSTU].[dbo].[FACULTY].[FACULTY_PK]), SEEK:([TMP1_BSTU].[dbo].[FACULTY].[FACULTY]='ИДиП') ORDERED FORWARD)
- |--Clustered Index Scan(OBJECT: ([TMP1_BSTU].[dbo].[PULPIT].[PULPIT_PK]), WHERE:([TMP1_BSTU].[dbo].[PULPIT].[FACULTY]='ИДиП'))

Просмотр плана выполнения

- Оператор с самым большим отступом выполняется первым
- Если два или более операторов имеют одинаковый отступ, они выполняются в порядке сверху вниз
- Compute Scalar вычисляет выражение, выдавая в результате скалярное значение
- Clustered Index Seek выполняет поиск строк по кластеризованным индексам
- Index Scan - обрабатываются все листья страницы дерева индексов

Подсказки оптимизации

- Подсказки оптимизации являются частью инструкции SELECT, которые указывают оптимизатору запросов, что нужно выполнять данную инструкцию определенным образом

Подсказки оптимизации hints

- табличные подсказки
- подсказки соединения
- подсказки запросов
- структуры планов

Табличные подсказки

- INDEX
- NOEXPAND
- FORCESEEK

```
SELECT * FROM AUDITORIUM WITH (INDEX(cs_index1))
WHERE AUDITORIUM = '461-2';
```

Подсказки соединения

- FORCE ORDER
- LOOP
- HASH
- MERGE

```
SELECT * FROM FACULTY  JOIN PULPIT
ON FACULTY.FACULTY = PULPIT.FACULTY
AND FACULTY.FACULTY = 'ИДиП'
OPTION (FORCE ORDER);
```

Подсказки запросов

- FAST n
- OPTIMIZE FOR
- OPTIMIZE FOR UNKNOWN
- USE PLAN

```
SELECT * FROM TEACHER OPTION (FAST 10);
```

Структуры планов

- sp_create_plan_guide
- sys.plan_guides

```
sp_create_plan_guide @name = N'Example_hint_15',
@stmt = N'SELECT * FROM FACULTY  JOIN PULPIT
ON FACULTY.FACULTY = PULPIT.FACULTY',
@type = N'SQL',
@module_or_batch = NULL,
@params = NULL,
@hints = N'OPTION (HASH JOIN)';

------------------------------
select * from  sys.plan_guides;
```

Results | Messages

plan_guide_id	name	is_disabled	query_text
65537	Example_hint_15	0	SELECT * FROM FACULTY JOIN PULPIT ON FACULTY.FACULTY = PULPIT.FACULTY

Настройка производительности

- Факторы, влияющие на производительность
- Мониторинг производительности
- Средства для настройки производительности

Факторы производительности

- прикладные программы баз данных
- система баз данных
- системные ресурсы

Приложения базы данных и производительность

- эффективность кода приложения
- проектирование на физическом уровне

Приложения базы данных и производительность

- использовать кластеризованные индексы
- исключить использование предиката NOT IN

Проектирование на физическом уровне

- Денормализация таблиц означает соединение вместе двух или более нормализованных таблиц в одну с некоторой избыточностью данных

Денормализация

- позволяет избежать использования операции соединения
- для денормализованных данных требуется меньшее число таблиц, чем для нормализованных
- для хранения денормализованной таблицы требуется больший объем дискового пространства
- модификация данных усложняется вследствие избыточности данных

СУБД и производительность

- оптимизатор запросов
- блокировки

Оптимизатор запросов

- Оптимизатор формулирует несколько планов выполнения запроса для выборки

Блокировки

- Используются для управления одновременным доступом к данным и для предотвращения потенциальных ошибок в случае одновременного доступа к одним и тем же данным
- Оказывают влияние на производительность системы вследствие гранулярности
- Блокировка на уровне строк обеспечивает наилучшую производительность
- Уровни изоляции влияют на длительность блокировки для инструкций SELECT

Системные ресурсы и производительность

- Центральный процессор
- Оперативная память
- Дисковые операции ввода/вывода
- Сетевое окружение

Процессор

- выполняет пользовательские процессы и взаимодействует с другими ресурсами системы
- Проблема:
  - операционная система и пользовательские программы обращаются со слишком большим количеством запросов

Оперативная память

- Динамическое освобождение памяти
- Проблемы:
  - Недостаточно памяти для выполнения требуемой работы.

Дисковые операции ввода/вывода

- Скорость передачи данных определяется объемом данных, который можно записать на диск в единицу времени
- При одновременном использовании системы базы данных большим количеством пользователей, несколько дисков лучше, чем один диск

Сетевая инфраструктура

- если сервер баз данных отправляет приложению какие-либо строки, отправлять следует только лишь те строки, которые действительно требуются приложению
- если продолжительное пользовательское приложение выполняется только на стороне клиента, то его следует переместить на сторону сервера

Зависимость ресурсов

- При увеличении количества процессоров нагрузка на них распределяется равномерно, что может решить проблему узкого места с дисковыми операциями
- Больший объем оперативной памяти повышает шансы, что страница будет найдена в памяти
- Чтение данных с диска вместо получения их из кэша тормозит систему при большом количестве конкурентных процессов

Дисковые операции ввода/вывода

- Выполняется большой объем дисковых операций
- Операции чтения с диска и записи на него являются двумя наиболее затратными операциями
- системы баз данных
- Данные хранятся в страницах размером в 8 Кбайт
- Кэш оперативной памяти разделен на страницы размером по 8 Кбайт
- Система читает данные по страницам
- Операции чтения выполняются для операций выборки и модификации

Дисковые операции ввода/вывода

- Если требуемая страница отсутствует в кэше, то она считывается с диска и помещается в буферный кэш - физический ввод/вывод или физическое чтение
- Буферный кэш является памятью совместного использования - к одной и той же странице могут обращаться несколько пользователей
- При модифицировании данных в буферном кэше выполняется операция логической записи
- Операция физической записи происходит только при сохранении данных из кэша на диск
  - упреждающее чтение
  - контрольные точки

Упреждающее чтение

- Выполнение чтения данных только из памяти и никогда не ожидать завершения операции чтения с диска
- Знать, какие следующие несколько страниц потребуются пользователю, и считывать эти страницы до того, как пользовательский процесс запросит их

Упреждающее чтение

- Read Ahead Manager
- Единица памяти для упреждающих операций чтения 64 Кбайт
- Выполнение объемных сканирований таблиц и сканирований диапазонов индексов
- Read Ahead Manager считывает до 2 Мбайт данных за один раз
- Каждый экстент считывается с помощью одной операции

Упреждающее чтение

- Множественные одновременные операции упреждающего чтения для каждого файла
- Используется информация из промежуточного уровня индексных страниц, находящегося сразу же над уровнем листьев
- В процессе распознаются смежные страницы и считываются один раз
- Может иметь отрицательное воздействие на производительность, если приходится читать слишком много страниц, излишне заполняя кэш
- Создавать индексы, которые в действительности необходимы

Вопросы?