



ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование



Планы запросов, а теперь подробнее




Аристов Евгений

telegram @AEugene

<https://aristov.tech>



Меня хорошо видно && слышно?

Ставьте  , если все хорошо
Напишите в чат, если есть проблемы
заодно проверяем, включена ли запись занятия

Включил Юджин запись ли ты



- более 20 лет в IT
- начинал с Delphi и Oracle (PLSQL), web PHP, perl
- окунулся в системное администрирование - развернул и поддерживал виртуальные инфраструктуры Hyper-V и VMware ESXi с сотнями серверов
- постепенно перешел на C#,php/js высоконагруженный MS SQL (T-SQL)
- последние 4 года Java/Spring
- сейчас основные деплои в kubernetes/docker на GCP
- десятки реализованных проектов, в т.ч. для сетей магазинов, фитнесов, отелей



Аристов Евгений

telegram @AEugene

Правила вебинара



Активно участвуем



Задаем вопрос в чат



Вопросы вижу в чате, могу ответить не сразу

Маршрут вебинара



Цели вебинара | После занятия вы сможете

1. Читать статистику

2. Объяснять, что влияет на построение плана

3. Рассказать про parameter sniffing и когда он возникает

4. Оптимизировать запрос по плану

Смысл | Зачем вам это уметь, в результате:

1. Увеличите производительность запросов

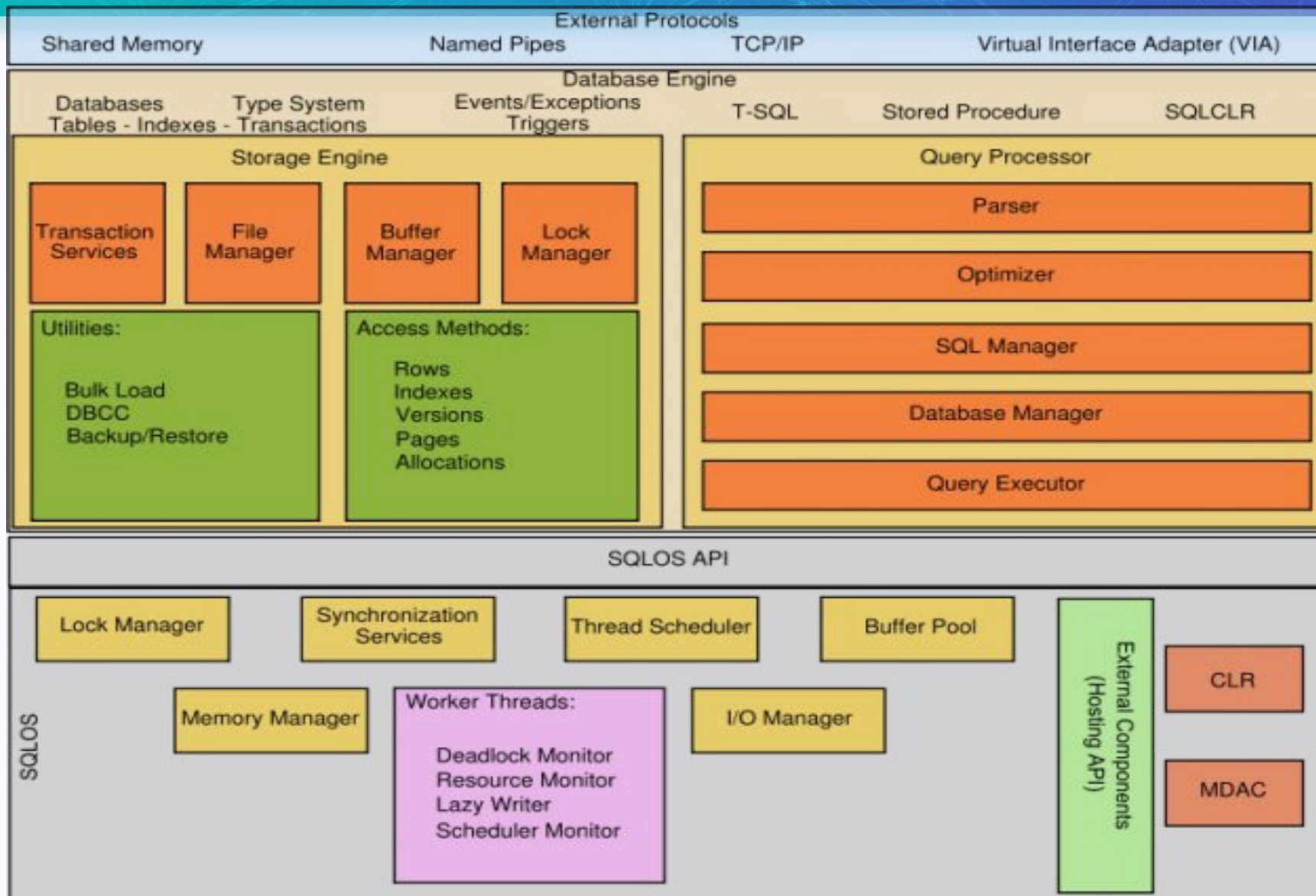
2. Оптимизируйте плохие планы

3. Поймете внутренности выполнения запросов

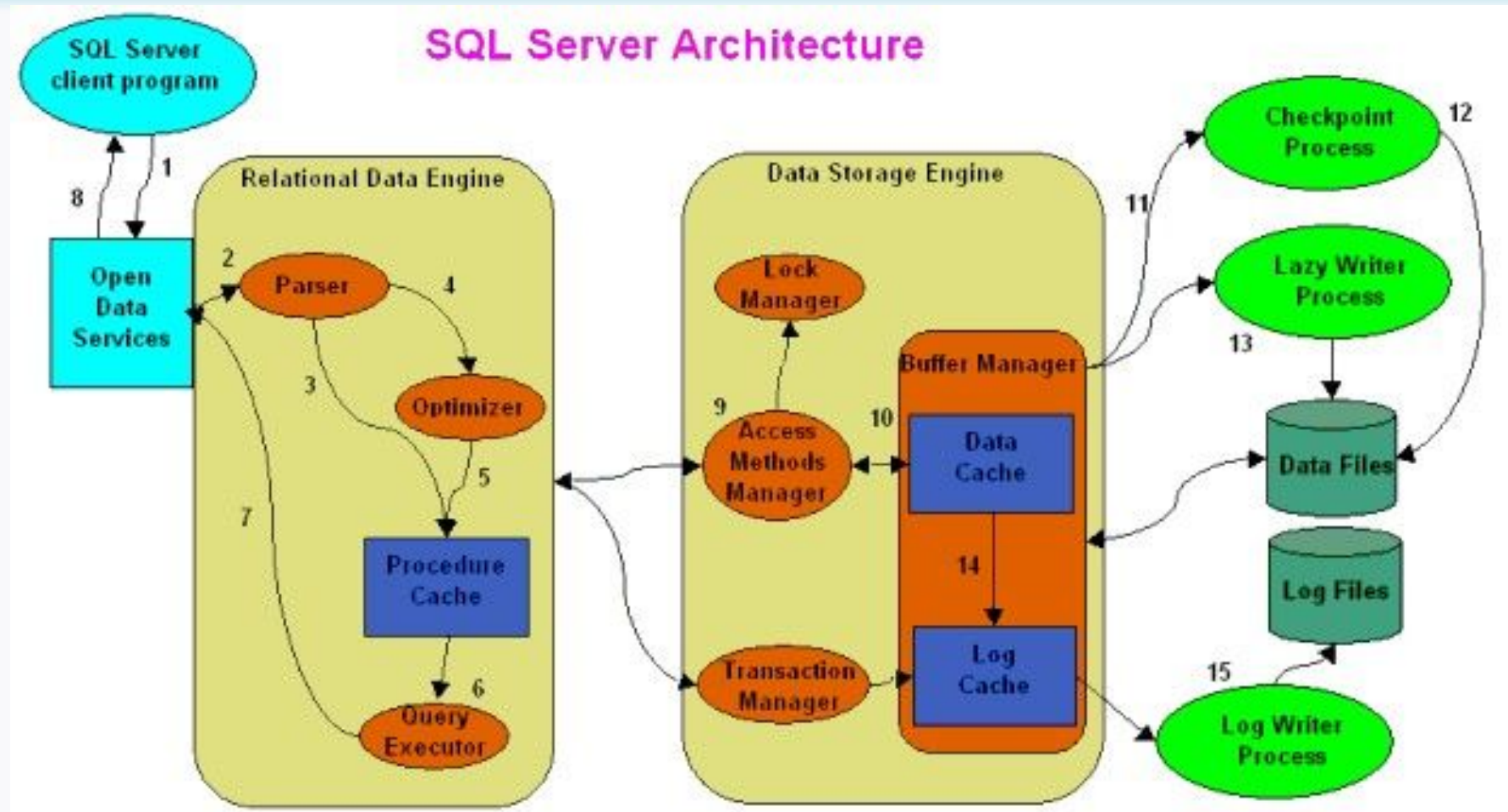
The background of the slide is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this image is a semi-transparent blue band across the middle, which contains a white network pattern of dots and lines. The text "Как строится план" is centered within this band.

Как строится план

Что внутри SQL сервера



Что внутри SQL сервера



Что внутри SQL сервера

1 SQL Statements

The client application sends “batches” of SQL statements to SQL Server for execution.

2 Syntax Check

The Parser checks the syntax of the SQL statements.

3 Cache Lookup

The Parser checks the syntax of the SQL statements, and calls the SQL Manager to see if matching SQL is already in the Procedure Cache. If a match is found, the Optimizer is bypassed.

4 Parsing

If no matching SQL is found in the Procedure Cache, the Parser translates the batch into a Query Tree and passes it to the Optimizer.

5 Optimization

The Optimizer takes the Query Tree and produces an optimal Execution Plan based on data sizes, indexes, join techniques and CPU, I/O & memory estimates.

6 Execution

The Query Executor runs the Execution Plan, often interacting heavily with the Access Methods Manager and Transaction Manager.

7 Results

The Query Executor passes results back to ODS.

Что внутри SQL сервера

8 Results

ODS buffers the results and returns them to the client only when the buffer fills or the batch of SQL statements ends.

9 Row Operations

The Access Methods Manager locates and modifies data at the row and index level. It calls the Buffer Manager to retrieve and modify individual pages.

10 Page Retrieval

When a data or index page is required, the Buffer Manager checks the data cache. If the page is already there, disk reads are avoided. If not, it is read from disk.

11 Page Modification

Pages in the cache that are modified will eventually be written back to disk (usually by the Checkpoint process). They may be modified many times in the cache before this happens.

12 Checkpoints

The Checkpoint process minimizes the amount of work SQL Server has to do on startup by periodically flushing dirty pages to disk.

13 Free Pages

The Lazy Writer process maintains the list of buffer pages that are available for immediate re-use. If necessary, it will write dirty pages to disk to make them available.

Что внутри SQL сервера

14 Log Caching

When the Buffer Manager modifies a page, it gets the Log Manager to add the log record(s) to the Log Cache. Once the user commits their changes, the Log Manager places these pages into a queue to be written to disk by the Log Writer. The user process waits for the log writer to flush the log pages to disk.

15 Log Flushes

The Log Writer process writes pages from the log flush queue out to disk. Once the writes are complete, any user processes waiting on that log cache can continue.

план

- Ищет достаточно хороший план
- Оценка количества строк делается с помощью статистики
- Cost в плане это абстракция, она считается на estimated и не пересчитывается в актуальном плане



Вопросы?

The background of the slide is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this image is a semi-transparent blue band that contains a white network pattern of interconnected dots and lines. The word "Статистика" is centered within this band in a white, sans-serif font.

Статистика

Статистика

- Статистика по полю\индексу – содержит информацию о распределении данных в таблице.
- План строится с учетом информации в статистике

Статистика

Опции Auto Update Statistics

- Auto Update Statistics Asynchronously
- DBCC SHOW_STATISTICS(table_or_indexed_view_name, target)
- UPDATE STATISTICS Application.Cities;
- UPDATE STATISTICS Application.Cities WITH FULLSCAN;

<https://docs.microsoft.com/ru-ru/sql/t-sql/statements/update-statistics-transact-sql?view=sql-server-ver15>

Статистика

- Если используется версия до SQL Server 2014 (12.x), SQL Server применяет пороговое значение в зависимости от процента измененных строк. Это значение не зависит от числа строк в таблице. Пороговое значение:
 - Если на момент оценки статистических данных кратность в таблице не превышала 500, обновление выполняется для каждых 500 модификаций.
 - Если на момент оценки статистических данных кратность в таблице превышала 500, обновление выполняется для каждых $500 + 20\%$ модификаций.
- Начиная с версии SQL Server 2016 (13.x); и при [уровне совместимости базы данных 130](#) SQL Server используется пороговое значение для динамического обновления статистических данных по убыванию. Значение изменяется в зависимости от числа строк в таблице. Оно вычисляется как квадратный корень из произведения текущего значения кратности в таблице и 1000. Например, если таблица содержит 2 миллиона строк, значение вычисляется как квадратный корень из $(1000 * 2000000) = 44721,359$. Благодаря этому изменению статистика для больших таблиц будет обновляться чаще. Но если уровень совместимости для базы данных ниже 130, применяется пороговое значение SQL Server 2014 (12.x).

Статистика

- **<automatic_tuning_option> ::=**
- **Область применения:** SQL Server (начиная с SQL Server 2017 (14.x);)
 - Включает или отключает параметр FORCE_LAST_GOOD_PLAN [автоматической настройки](#).
 - FORCE_LAST_GOOD_PLAN = { ON | **OFF** }
 - Компонент Database Engine автоматически включает последний известный удачный план для Transact-SQL запросов, когда новый план запроса приводит к снижению производительности. Компонент Database Engine непрерывно отслеживает производительность запроса Transact-SQL в форсированном плане.
 - В случае повышения производительности Компонент Database Engine будет продолжать использовать последний известный удачный план. Если повышений производительности нет, Компонент Database Engine создаст новый план запроса. Инструкция завершится ошибкой, если хранилище запросов не включено или не находится в режиме *Чтение и запись*.
 - OFF
- Компонент Database Engine сообщает о потенциальном снижении производительности запросов, вызванном изменениями планов запросов в представлении [sys.dm db tuning recommendations](#).

Статистика

Параметр конфигурации сервера "optimize for ad hoc workloads" - SQL Server

- Демо

The background of the slide is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this image is a semi-transparent blue band across the middle, which contains a white network pattern of dots and lines. The word "Вопросы?" is centered in this band in a large, white, sans-serif font.

Вопросы?



МИНИТЕСТ

<https://forms.gle/EoZMpSJtuLBo2yEN6>

3-5 минут

The background of the slide is a blue-tinted aerial photograph of a dense city skyline, likely New York City. Overlaid on this image is a network diagram consisting of numerous small dots connected by thin, light blue lines, creating a web-like pattern across the center of the slide. The text "Parameter sniffing" is centered in a large, white, sans-serif font.

Parameter sniffing

Parameter sniffing

Parameter sniffing – использование плана из кэша для разных параметров, что приводит к неэффективному выполнению запроса и нестабильной работе сервера

Parameter sniffing

Примеры:

Разное количество данных в зависимости от параметров

Процедура с большим количеством параметров, которые могут быть не заданы

Parameter sniffing

Лечение:

- Локальные переменные // файл 61 parameter sniffing
- OPTION (OPTIMIZE FOR UNKNOWN)
- OPTION (RECOMPILE)
- Sp_recompile
- Флаг 4136 // перекомпилируем все процедуры по умолчанию

очистка кеша:

- DBCC FREEPROCCACHE (plan_handle | sql_handle)
- Sp_recompile
- DBCC FREESYSTEMCACHE('SQL PLan')
- DBCC FREESYSTEMCACHE('ALL')

The background of the slide is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this image is a semi-transparent blue band across the middle, which contains a white network diagram of interconnected nodes and lines. The title text is centered within this band.

Оптимизация запроса

оптимизация запроса

Кэш планов

1. Хранит планы скомпилированных запросов
2. Выкидывает информацию о плане в зависимости от частоты использования и информации о стоимости
3. План также удаляется из кэша при перестройке статистики по объекту

оптимизация запроса

Когда кэш это плохо

- Много Ad Hoc запросов и каждый план используется только 1 раз
- Запрос с разным количеством информации и стратегии выборки (parameter sniffing)

The background of the slide is a high-angle, blue-tinted aerial photograph of a dense urban skyline, likely New York City. Overlaid on this image is a semi-transparent blue band that contains a white network pattern of interconnected dots and lines. Centered within this band is the word "Вопросы?" in a large, white, sans-serif font.


Вопросы?

The image features a blue-toned aerial photograph of a dense city skyline, likely New York City, with numerous skyscrapers. A semi-transparent blue band with a white geometric network pattern of dots and lines runs horizontally across the middle of the image. The Russian word "Порефлексируем" is centered within this band in a large, white, sans-serif font.


Порефлексируем

Вопросы?

- **Зачем нужна статистика?**
- **Что такое Ad Hoc?**
- **Как лечить parameter sniffing?**

The background of the image is an aerial view of a city skyline, likely New York City, with numerous skyscrapers. The image is overlaid with a semi-transparent blue layer that features a white network pattern of dots and lines. The text is centered within this blue layer.

Заполните, пожалуйста,
опрос о занятии по ссылке в чате
<https://otus.ru/polls/24108/>



Спасибо за внимание!
Приходите на следующие вебинары

Аристов Евгений