

РАСШИРЕННАЯ ОПТИМИЗАЦИЯ СКРИПТОВ В ORACLE

Бухвалова И. А.

*Белорусский национальный технический университет, Минск,
Беларусь, irena_buchval@mail.ru*

Оптимизация скриптов – важный момент в работе каждого администратора базы данных особенно когда необходимо хранить довольно много информации и в любой момент легко и быстро ее извлечь. Этот процесс является критическим для любой организации, когда от быстрейшего действия обработки данных напрямую зависит полученная компанией прибыль.

Для облегчения работы администратора при проведении мониторинга запросов в базе данных предлагается программное средство «Расширенная оптимизация скриптов в Oracle».

Для корректной работы программного средства необходимо было организовать соединение между приложением и базой данных Oracle, настроить работу на уровне проверки подключения к базе данных, создать проверку на корректность вводимой информации при идентификации администратора, а также организовать выборку данных о запущенном запросе по следующим метрикам:

- 1) SID – уникальное имя инстанса БД;
- 2) под каким пользователем запущен запрос со стороны БД;
- 3) под каким пользователем запущен запрос со стороны операционной системы (далее ОС);
- 4) SQLID – уникальный идентификатор запроса в БД;
- 5) SPID – уникальный номер процесса в ОС;
- 6) статус запроса в БД;
- 7) прогресс запроса в БД;
- 8) текст запроса в БД.

Когда пользователь начинает операцию по извлечению данных, SQL-оператор этого пользователя проходит несколько последовательных этапов, которые все вместе называются обработкой запроса.

Под обработкой запроса подразумевается преобразование SQL-оператора в эффективный план выполнения для возврата запрашиваемых данных из базы. Под оптимизацией запроса

понимается процесс выбора наиболее эффективного плана выполнения для достижения результата с наименьшими затратами в плане потребления ресурсов, наподобие ресурсов подсистемы ввода-вывода и ЦП, на том сервере, где работает база данных, а также сокращения общего времени выполнения запроса, представляющего собой просто сумму показателей времени выполнения всех входящих в состав данного запроса операций [1].

Тюнинг статистики оптимизатора предполагает настроенный сбор и время хранения статистики, необходимой для работы оптимизатора. Статистика необходима для того, чтобы оптимизатор мог больше знать о распределении данных в схемах и соответственно об используемых таблицах.

Время последнего сбора статистики определяется значением поля `Last_Analyzed` для таблиц и индексов, которое находится из таблиц `all_tables (dba_tables)` и `all_indexes (dba_indexes)` соответственно. Обычно сбор статистики не отключается, но если в базе данных производится очень большое количество изменений числа строк в течение дня, статистика собирается вручную.

Целями оптимизации всегда являются или уменьшение времени отдачи результатов пользователю, или уменьшение потребления ресурсов за это же время. Добиться результатов можно уменьшением нагрузки (оптимизация плана выполнения), балансированием нагрузки (по времени) или распараллеливанием нагрузки. Для начала необходимо определить наиболее ресурсоемкие запросы, это можно сделать с помощью `Oracle Enterprise Manager Cloud Control 13c`.

Далее, после определения проблемных запросов, необходимо собрать предварительные данные для тюнинга, а именно: полный текст SQL запроса (из `v$sqltext`); структуру таблиц, задействованных в запросе; определения индексов (включая перечень и порядок полей, уникальность и т. п.); статистику оптимизатора (включая количество строк каждой таблицы, дату, когда проводился анализ); информацию о связанных представлениях (`view`); текущий и, возможно, предыдущий планы выполнения.

Есть несколько основных вариантов для оптимизации SQL запроса:

- 1) пересобрать статистику;
- 2) изменить план выполнения запроса, при этом обращать внимание надо на то, чтобы:
 - фильтр был правильным;

- порядок объединения должен предполагать, что на следующий шаг передается минимальное количество значений;
 - метод объединения должен соответствовать отдаваемым значениям;
 - эффективное использование представлений;
 - доступ к таблицам должен быть эффективным (например, full scan не означает неэффективность – он может быть предпочтителен на малых таблицах или при использовании hash join).
- 3) перестроить запрос, в случае если он устарел;
 - 4) перестроить, может быть даже изменить индексы;
 - 5) изменить или выключить триггеры;
 - 6) изменить или выключить правила ссылочной целостности (constraints);
 - 7) следить за ростом данных и изменять параметры администрирования таблиц в зависимости от частоты использования.

Цели оптимизации для разных запросов могут отличаться, например, где-то необходимо получить максимально быстрый полный ответ или же максимально быстрый возврат первых строк [2].

Соответственно и методы работы оптимизатора будут отличаться и зависеть от поставленной цели, которая может выставляться опцией OPTIMIZER_MODE (выставляется для экземпляра или сессии), подсказками SQL (FIRST_ROWS(n),FIRST_ROWS,ALL_ROWS,CHOOSE,RULE) и зависеть от собираемой статистики (пакет DBMS_STATS).

СПИСОК ЛИТЕРАТУРЫ

1. Милсап, К. Oracle. Оптимизация производительности / В. Дунаев – СПб.: БХВ-Петербург, 2006.
2. Карвин, Б. Программирование баз данных SQL: типичные ошибки и их устранение / Б. Карвин, М. Райтман – Рид Групп, 2012.