DOI: 10.17117/na.2016.01.01.350

Поступила (Received): 31.01.2016

<http://ucom.ru/doc/na.2016.01.01.350.pdf>

Амантай А.Ж., Боярский Э.Ф., Амиров А.Ж.
Исследование способов оптимизации SQL-кода
на примере СУБД Oracle 11g

Amantaj A.Zh., Boyarskij E.F., Amirov A.Zh.
Research ways to optimize SQL-code
an example Oracle 11g database

В статье исследуются время выполнения и способы оптимизации SQL-кода, а также выявляются компоненты окружения и другие характеристики, влияющие на его производительность. Поэтапно показаны задачи, которые необходимо решить для оптимизации SQL-кода. Также были рассмотрены рекомендации по разработке оптимальной структуры БД

Ключевые слова: оптимизация, SQL-код, план выполнения, экстенст, optimization, execution plan, extent, SQL-code

Амантай Ардак Жанатулы

Магистрант

Карагандинский государственный технический университет

г. Караганда, Бульвар-мира, 56

Боярский Эдуард Федорович

Кандидат технических наук, доцент

Карагандинский государственный технический университет

г. Караганда, Бульвар-мира, 56

Амиров Азамат Жанбулатович

Заведующий кафедрой

Карагандинский государственный технический университет

г. Караганда, Бульвар-мира, 56

The article investigates the execution and how to optimize SQL-code and identifies the components of the environment and other characteristics that affect its performance. Incrementally shows the tasks that need to be addressed optimisichatsii SQL-code. Were also considered the recommendations on the development of the optimal structure of the database

Key words: optimization, SQL-CODE, implementation plans extents, optimization, execution plan, extent, SQL-code

Amantaj Ardak Zhanatuly

Master

Karaganda state technical university

Karaganda, Boulevard Mira, 56

Bojarskij Eduard Fedorovich

Candidate of Technical Sciences, Associate Professor

Karaganda state technical university

Karaganda, Boulevard Mira, 56

Amirov Azamat Zhanbulatovich

Head of Department

Karaganda state technical university

Karaganda, Boulevard Mira, 56

При исполнении SQL-команды время тратится на:

- 1) разбор, включая построение проекта выполнения ("hard parse");
- 2) фактически, исполнение в соответствии с подобранным планом.

В СУБД Oracle 11g генерацией планов исполнения занимается стоимостной оптимизатор (Cost Based Optimizer, CBO), работа которого заключается в

том, чтобы обнаружить механизм выполнения, использующий минимальное количество ресурсов для получения достаточных результатов для данной команды [1]. Наилучший из нескольких вероятных планов выбирается СВО на базе оценки стоимости выполнения SQL-команды по этому плану, которую возможно квалифицировать как суммарное прогнозируемое время исполнения команды, выраженное в единицах времени выполнения одноблочного чтения [1].

Ключевые причины погрешностей оптимизатора (кроме погрешностей реализации):

- недоступность, отсутствие или же неправильность статистики о распределении данных и применении системных ресурсов;
- ошибки в коде [1].

Соответственно, минимизировать общее время исполнения SQL-команд (увеличить производительность) возможно:

- снизив количество разборов "hard parse" (к примеру, сменив литералы в команде на связанные переменные);
- изменив свойства системы, оказывающие большое влияние на производительность (к примеру, добавив оперативной памяти или же перераспределив имеющуюся);
- в случае если имеющиеся в достаточном количестве ресурсы системы тратятся не оптимально – заставив оптимизатор подобрать другой план исполнения (к примеру, изменив текста SQL-команды или добавив/удалив индексы).

Компанией Oracle созданы всевозможные программы, утилиты и советчики, которые призваны помочь с анализом, диагностикой и оптимизацией SQL кода: Automatic Database Diagnostic Monitor. SQL Access Advisor. SQL Tuning Advisor. Они диагностируют первопричины проблем с производительностью и дают рекомендации по исправлению обнаруженных проблем [2].

Эти программы имеют большое количество плюсов, хотя не всем доступны по причине стоимости – они считаются частью Oracle Diagnostic Pack и Oracle Tuning Pack, которые доступны лишь в самой дорогой редакции СУБД Oracle (Enterprise Edition). Также, для введения приобретенных рекомендаций нужно будет обладать знаниями в сфере администрирования СУБД Oracle, и еще понимать и учитывать, что изменение одной характеристики сможет позитивно сказаться на времени исполнения далеко не всех команд.

Целью данной работы считается создание методологии, позволяющей на базе имеющихся данных о SQL-коде и свойствах окружения получить рекомендации по повышению производительности применяемых в приложении SQL-команд. При всем при этом ожидается, что внедрение советчиков или недоступно, или необходимо проверить выданные рекомендации, проследив цепочку рассуждений, приведших к определённому выводу.

Для достижения этой цели нужно будет решить следующие задачи:

- обнаружить составляющие окружения и другие свойства, оказывающие большое влияние на производительность SQL кода;
- произвести анализ способов повышения производительности SQL кода исходя из обнаруженных компонентов и их сочетаний:

- создать методологию повышения производительности SQL кода с учетом зависимостей между составляющими окружения;
- спроектировать приложение, позволяющее собрать вполне вероятные алгоритмы повышения производительности исходя из окружения.

Самый обычный и доступный способ понять в первом приближении, что из себя представляет какой-нибудь запрос – это включить Autotrace. При всем этом станет возможно увидеть план выполнения запроса и небольшую статистику по его исполнению. Хотя Autotrace не даст полной картины, им слишком просто воспользоваться и как правило его диагностики достаточно [3].

В качестве примера, рассмотрим такой признак статистики исполнения команды, как «Recursive calls». Данный показатель представляет из себя число рекурсивных вызовов в секунду, на протяжении периода выборки. Рекурсивные вызовы представляют собой вспомогательные SQL-команды, которые сервер Oracle должен выполнять для выполнения SQL-команды пользователя. К примеру, в случае если применяется табличное пространство, управляемое по словарю и для запрошенной пользователем вставки строки в таблицу не хватает места, сервер Oracle делает рекурсивные запросы к словарю данных для динамического выделения дополнительного пространства. Чем больше значение показателя Recursive calls, тем медленнее выполнялся запрос. Уменьшить его значение и время выполнения SQL-команд, можно снизив частоту обращения к словарю данных путем выделения больших экстенгов.

Размер экстенгов, выделяемых при увеличении табличного пространства, определяется параметрами хранения, которые могут быть использованы не только в ходе создания табличного пространства, но и в период создания всевозможных объектов схемы. Для создания экстенгов применяется оператор DEFAULT STORAGE. Характеристики хранения, отмеченные в DEFAULT STORAGE, используются при создании и росте объектов схемы. К объектам схемы, которые создаются с указанием точных характеристик хранения, характеристики хранения, параметры хранения по умолчанию не применяются.

Размер экстенга может быть изменен с помощью программы Enterprise Manager. Для этого необходимо зайти в меню Server и в списке Storage выбрать Tablespaces. Затем, в появившемся окне, создать новое табличное пространство с помощью кнопки Create, управляемое по словарю. В настройках можно изменить размер экстенга. Для повышения производительности, будем его увеличивать, либо изменим тип табличного пространства, на локальное.

В будущем планируется продолжать углубление в тему производительности SQL-кода. вносить конкретные предложения по способам повышения его производительности и проектировать приложение, позволяющее подобрать возможные алгоритмы повышения производительности в зависимости от окружения.

Список используемых источников:

1. Джонатан Льюис – Oracle. Основы стоимостной оптимизации. Питер, 2007.
2. Automatic Performance Diagnostics 11g Release 2 (11.2).
URL: http://docs.oracle.com/cd/E25054_01/server.1111/e16638/diag.htm#
3. Использование Autotrace в SQL *Plus. URL: http://www.oracledba.ru/note_s_autotrace.html

4. Oracle DBA. Обзор табличных пространств.

URL: http://oracle-dba.ru/database/basics/oracle_database_tablespaces/how_to_create_new_tablespace/

© 2016, Амантай А.Ж., Боярский Э.Ф., Амиров А.Ж.
Исследование способов оптимизации SQL-кода на
примере СУБД Oracle 11g

© 2016, Amantaj A.Zh., Boyarskij E.F., Amirov A.Zh.
Research ways to optimize SQL-code an example
Oracle 11g database