

Алексей Тюренков. Параллельная SQL-загрузка в Essbase 11.1.1.2.0 для ASO (Multiple SQL Data Loads in Parallel to Aggregate Storage Databases):

Ключевые моменты:

1. Включить SQL-параллельность для Merant DataDirect ODBC drivers в файле [ESSBASEPATH/bin/esssql.cfg](#)
2. В Essbase 11.1 поддерживается до 8 rule-файлов на загрузку в ASO (для каждого rule-файла - один буфер загрузки Essbase, и соответственно одна сессия к СУБД-источнику).
3. В каждом rule-файле должна быть одинаковая идентификационная информация (user name / password). rule-файлы должны находиться на сервере Essbase.
4. Инициализация буферов загрузки (load buffers) происходит автоматически, ресурсы памяти для буфера выделяются в равных долях по количеству rule-файлов (один буфер для каждого rule-файла). Не нужно явно выделять ресурсы памяти для буферов (не требуется `alter database ...initialize load_buffer with buffer_id X resource_usage XXX`).
5. Essbase создает сессии к СУБД (одна сессия для каждого rule-файла), параллельно загружает данные в буферы. После заполнения буферов происходит автоматическое объединение буферов и сброс в ASO. Преимущество в том, что эти этапы проводятся параллельно и в одной операции (`import database ... using multiple rules_file`), которая быстрее чем индивидуальная работа с каждым буфером.
6. Essbase при доступе к Oracle не использует OCI, не знает о tnsnames.ora и Oracle-клиенте. Доступ Essbase к Oracle осуществляется через Merant DataDirect ODBC-драйвер (входит в дистрибутив Essbase). В поставку входят драйвера для наиболее распространённых СУБД.
7. Получить прирост производительности за счет использования OCI + параллельность в «родном» ODBC для Windows не удалось.
При установке Essbase под Windows возможно использование OCI с «толстым» ODBC-драйвером Oracle. Для Essbase это non-DataDirect ODBC и распараллеливание в SQL-методе не возможно.
Работает только одна Oracle-сессия.

«толстый» ODBC-драйвер можно скачать с oracle.technet:

<http://www.oracle.com/technology/software/tech/windows/odbc/index.html>

8. Системные установки параллельности в essbase.cfg не связаны с новым методом параллельной SQL-загрузки, что не снижает важности их настройки:

http://download.oracle.com/docs/cd/E12825_01/epm.111/esb_dbag/dldebug.htm

Setting	Description
DLTHREADSPREPARE	Specifies how many threads Essbase may use during the data load stage that codifies and organizes the data in preparation to being written to blocks in memory.
DLTHREADSWRITE	Specifies how many threads Essbase may use during the data load stage that writes data to the disk. High values may require allocation of additional cache. See Implications in Sizing the Data Cache . Note: For aggregate storage databases, Essbase Server uses one thread with aggregate storage cache. The DLTHREADSWRITE setting is ignored.
DLSINGLETHREADPE RSTAGE	Specifies that Essbase use a single thread per stage, ignoring the values in the DLTHREADSPREPARE and DLTHREADSWRITE settings.

Для наглядности, сравним два метода загрузки:

- загрузка из txt на rule-файлах.
- загрузка через SQL на rule-файлах.

Пример использует два буфера (два rule-файла).

1. Вводится новый синтаксис MaxL:

```
-- TXT
  "import database ... data to load_buffer with buffer_id"
-- SQL
  "import database ... using multiple rules_file"
```

1.1. Загрузка из txt на rule-файлах:

В первой MaxL Shell-сессии:

```
-- инициализировать буфер загрузки №1, выделив ему 50% памяти
alter database AsoSamp.Sample
  initialize load_buffer with buffer_id 1 resource_usage 0.5;

-- загрузить данные в буфер №1 из dataload1.txt (данные предварительно разрезаются на файлы)
import database AsoSamp.Sample data
  from data_file "dataload1.txt" using server rules_file 'rule1'
  to load_buffer with buffer_id 1
  on error abort;
```

Одновременно в другой MaxL Shell-сессии:

```
-- инициализировать буфер загрузки №2, выделив ему 50% памяти
alter database AsoSamp.Sample
  initialize load_buffer with buffer_id 2 resource_usage 0.5;

-- загрузить данные в буфер №2 из dataload2.txt
import database AsoSamp.Sample data
  from data_file "dataload2.txt" using server rules_file 'rule2'
  to load_buffer with buffer_id 2
  on error abort;
```

После заполнения буферов, объединить их и сбросить в базу.

```
import database AsoSamp.Sample data
  from load_buffer with buffer_id 1, 2;
```

1.2. Загрузка через SQL на rule-файлах:

```
-- Включить SQL-параллельность для DataDirect ODBC drivers в файле ESSBASEPATH/bin/esssql.cfg
http://download.oracle.com/docs/cd/E12825\_01/epm.111/esb\_sqlint/frameset.htm?non\_dd\_drivers.html
```

Для Oracle (параметры вступят в силу после рестарта Essbase Server):

```
[
Description "Merant Oracle 5.2 32-bit Wire Protocol"
DriverName  ORACLE_DWH_ODBC
UpperCaseConnection 0
UserId 1
Password 1
Database 1
SingleConnection 0
IsQEDriver 1
]
```

```
-- Все операции параллельно в одной команде.
```

```
Стартовать начиная с buffer_id = 1 (это rule1.rul) и далее по списку.
```

```
import database AsoSamp.Sample data
  connect as TBC identified by 'password'
  using multiple rules_file 'rule1' , 'rule2'
  to load_buffer_block starting with buffer_id 1
  on error write to "error.txt";
```

Эксперимент:

Сравнивались два вида загрузки:

- txt-загрузка на rule-файлах.
- параллельная SQL-загрузка на rule-файлах.

Уловия эксперимента:

2 буфера в обоих экспериментах (буферы 50% + 50%)
2 rule-файла.

Essbase поднят на виртуалке (рабочая машина, P4 3GHz, 2GB RAM, SATA HDD).

В качестве источника выступал DataWarehouse (Oracle 10gR2).

txt-файлы копируются на Essbase.

Сеть fast ethernet.

Результат:

Прирост производительности +15%

Учитывается чистое время загрузки.

Копирование txt-файла на сервер Essbase не учитывается.

Комментарии результатов:

1. Мониторинг Oracle показывает что Essbase создаёт 2 Oracle-сессии загрузки (по количеству rule-файлов).

Рекомендую использовать параллельность на таблицах Oracle.

В этом примере DOP = 4: `ALTER TABLE ... PARALLEL (DEGREE 4)`

В итоге, имеем дополнительную параллельность в сессиях загрузки за счет использования PX-координатора Oracle.

2. Прирост производительности можно объяснить несколькими причинами:

- Эффективная реализация параллельной загрузки в Essbase по SQL-методу, а именно реализация в одной команде параллельной загрузки буферов / объединение буферов / сброс в ASO.
- Использование параллельности в Oracle.
- Медленные диски SATA на виртуалке, что приводит к снижению скорости чтения txt-файла.

3. Увеличение кол-ва сессий загрузки (до 8-ми), переход на производительное серверное железо для Essbase, а также правильная установка параллельности в Oracle, должна изменить разницу в производительности.

4. С учетом того, что формирование txt-файла на источнике данных и его копирование на Essbase это весьма затратный процесс зависящий от реализации (т.е. вообще-то время копирования нужно включать), новый метод SQL-загрузки может оказаться конкурентоспособным.

Следующий этап:

Сравнение SQL-метода с free form data loading (без rule-файлов).

free form data loading на сегодня является самым быстрым методом загрузки данных в Essbase.

При правильной реализации, результаты будут явно не в пользу первого метода, особенно если выгрузку данных осуществлять на сервере Oracle через ProC.

Выгрузка последним методом быстрее чем через UTL_FILE в среднем в 10-15 раз.

Инфа и связанные темы:

-- Подготовка и процесс в целом.

http://download.oracle.com/docs/cd/E12825_01/epm.111/esb_sqlint/frameset.htm?load_sql_data.html

-- Загрузка данных - суть методики "Performing Multiple SQL Data Loads in Parallel..."

http://download.oracle.com/docs/cd/E12825_01/epm.111/esb_sqlint/load_sql_data.html

-- Конфигурирование ODBC для Essbase

http://download.oracle.com/docs/cd/E12825_01/epm.111/esb_sqlint/frameset.htm?config_ds.html

-- Loading, Calculating, and Retrieving Aggregate Storage Data (основные методы загрузки и раздел "Performing Multiple Data Loads in Parallel")

http://download.oracle.com/docs/cd/E12825_01/epm.111/esb_dbag/allocare.htm

-- Родные ODBC-драйвера от Oracle

<http://www.oracle.com/technology/software/tech/windows/odbc/index.html>