# Решение Задачи 11

# Архитектура решения

Проект состоит из следующих компонентов

1) Обработчики- задания jobs, которые разбирают сообщения в пуле и вызывают имитацию стандартного АПИ

2) Загрузчик- задание job, которое периодически (раз в минуту) подгружает данные в пул, по умолчанию 13000 строк- как раз на 1 минуту работы обработчиками плюс небольшой запас.

Загрузчик останавливается, когда нет сообщений в пуле. Можно запустить его на расписании.

3) Монитор – задание job, которое периодически записывает состояние пула в спец. таблицу для дальнейшего вывода в отчет.

В проект входят скрипты инсталяции, деинсталяции и тестирования.

# Инсталляция

1) Для инсталляции необходимо войти в БД Oracle под пользователем APPS или пользователем, который имеет права в создании объектов в APPS и выполнения пакетов.

2) Запустить скрипт install.sql

# Тестирование

1) Параметр job\_queue\_processes должен быть больше или равен 12

2) Запустить скрипт test.sql

3) Система запустит 12 заданий, состояние которых можно мониторить запросами

--текущие jobs в работе

SELECT j.state,j.\* FROM dba\_SCHEDULER\_JOBS j where lower(job\_name) like 'xx%';

-- количество сообщений в пуле

SELECT COUNT(1) FROM XX\_T\_IMPORT;

В скрипте test есть переменная l\_iterations- сколько раз запускать job загрузчика, чтобы пул заполнялся. Для начала рекомендуется небольшое значение- 2. Это значение можно изменить

# Отчет

Когда все задания отработают и пул станет пустым, нужно в sqlplus запустить скрипт create\_report.sql

Он создаст html отчет о работе теста в текущей папке проекта под названием t1\_test\_report.html

Для того, чтобы появился график, необходимо соединение с интернетом, тк используется библиотека построкения графиков graph.js

Результат работы можно также посмотреть с помощью запросов

-- .cnt cnt\_per\_sec: количество обработанных транзакций в секунду,

-- pool\_lines\_count: количество строк в пуле, статистика снимается каждые 5 секунд

*select st.creation\_date, st.cnt cnt\_per\_sec, m.pool\_rows\_count from*

*(select creation\_date,count(1) cnt from xx\_t\_import\_standard*

*group by creation\_date*

*) st,*

*xx\_upload\_test\_monitor m*

*where st.creation\_date=m.creation\_date(+)*

*order by 1 desc*

*– сравнение суммы входящих транзакций и суммы, загруженной в “стандартную” таблицу, должны быть равны*

*select r.\*,(select sum(trx\_amount) from xx\_t\_import\_standard) processed\_amount\_sum from xx\_upload\_test\_results r*

**Важно:** Значение сообщений в секунду должно быть около 200 и чуть ниже. 200 это максимально возможное значение, так как имитация вызова стандартного АПИ останавливает сессию на 0.05 сек (dbms\_lock.sleep), это 20 сообщений в секунду для одной сессии и 200 для 10 сессий. Например 190 значит что 10 сессий потратили на считывание пула менее 0,5% времени от выполнения АПИ

Цифры в количестве сообщений в пуле круглые, так как массив сообщений считывается по 100