

Функциональные характеристики ПО Датафлот Репликация

Оглавление

1. Основные характеристики	3
1.1 Гетерогенная репликация	3
1.2 Режимы извлечения данных	3
1.3 Режимы загрузки данных	3
1.4 Клонирование схемы	4
1.5 Начальная синхронизация	4
1.6 Управление проектами	4
1.7 Управление задачами	4
1.8 Топологии развертывания	4
2. Дополнительные характеристики	5
2.1 Дополнительная обработка данных	5
2.2 Мониторинг	5
2.3 CLI	5
2.4 Обработка ошибок загрузки	5



1. Основные характеристики

1.1 Гетерогенная репликация

- **DML Репликация**

Система позволяет реплицировать изменения данных в режиме реального времени. Поддерживается подавляющее большинство типов данных. Имеется функционал фильтрации данных.

- **DDL Репликация**

Репликация изменений схемы доступна для колонок, таблиц и индексов. Ограничением, как правило, является только ограничения производителя. Например, изменение длины колонки возможно не во всех базах данных, поэтому репликация таких изменений невозможна (даже если источник поддерживает такой тип операции).

1.2 Режимы извлечения данных

- **Дискретный режим**

В дискретном режиме процесс извлечения данных длится пока все данные, доступные для чтения, не извлечены. После этого процесс завершается. Для некоторых производителей БД этот режим позволяет исключить подключение к БД.

- **Непрерывный режим**

В непрерывном режиме процесс извлечения всегда работает. Если данных, доступных для чтения нет, то процесс зависает на непродолжительный промежуток времени. После чего просыпается, проверяет наличие данных. Далее обрабатывает все доступные данные и засыпает. Данный режим позволяет организовать репликацию максимально приближенную к режиму реального времени.

1.3 Режимы загрузки данных

Режимы загрузки данных делятся по формату представления данных в БД-приемнике и по дискретности загрузки. Дискретный и непрерывный режимы во многом схожи с соответствующими режимами процесса извлечения данных.

- **Стандартный**

DML изменения переносятся как есть, структура таблицы-приемника соответствует структуре источника. Данный режим оптимален при использовании OLTP баз в качестве приемника.

- **Исторический**

В данном режиме в структуре таблицы-приемника одной колонке источника соответствуют две колонки приемника. Каждая пара колонок хранит значение до изменения и после изменения. Так же таблица-приемник имеет несколько служебных колонок, которые хранят мета информацию об изменении, например время, тип изменения, данные транзакции. Строки в таблице-приемнике только добавляются, это позволяет отслеживать историю изменений каждой строки источника.

- **Хранилище**

Данный режим оптимизирован для репликации в хранилища данных. На приемнике создается пара таблиц, которая соответствует одной исходной таблице. В этой паре одна таблица временная и имеет структуру, схожую со структурой режима Исторический. Вторая таблица постоянная и имеет структуру, соответствующую

структуре источника. Механизм загрузки представляет собой своего рода комбинацию стандартного и исторического режимов.

1.4 Клонирование схемы

При отсутствии готовой схемы приемника система предоставляет возможность создать ее, опираясь на структуру источника. Данный механизм позволяет создать схему как для стандартного режима, так и для исторического. Имеется возможность сразу создать схему в приемнике либо, сгенерировать SQL скрипт для самостоятельного выполнения.

1.5 Начальная синхронизация

Этот процесс предназначен для начального переноса данных из источника в приемник и является обязательным для режимов стандартный и хранилище.

Для исторического режима процедура начальной синхронизации не предусмотрена.

Имеется возможность частичной синхронизации, когда переносятся данные только части связанных таблиц. Выполнить это действие можно в любой момент жизненного цикла процесса репликации.

1.6 Управление проектами

Система предоставляет графический инструмент для управления проектами, основными возможностями которого являются:

- Создание и редактирование проектов репликации
- Импорт и экспорт проектов репликации
- Развертывание (deploy) проектов репликации

Это механизм упрощает процесс переноса изменений проекта между разными средами. Есть возможность как полного развертывания, так и частичного. В первом случае проект переносится полностью в другую среду, во втором, проект должен существовать в обеих средах и переносятся только изменения.

1.7 Управление задачами

- Задачи репликации

Эти задачи являются встроенными и позволяют создать производные для более тонкой настройки процессов извлечения и загрузки данных.

- Второстепенные задачи

Этот класс задач позволяет решать инфраструктурные вопросы, например создание бэкапа журнала транзакций, выполнение пользовательских скриптов.

- Расписания

Расписания позволяют автоматизировать управление задачами. Мастер создания расписания позволяет выбрать один из трех шаблонов и быстро создать и настроить расписание.

1.8 Топологии развертывания

- Одиночный сервер

В данной топологии все процессы репликации физически выполняются на одном сервере.

- Многосерверная архитектура

В этом варианте один главный сервер управляет остальными. Задачи могут быть распределены по нескольким серверам. Распределенная топология способна решать задачи репликации, когда одна база находится в защищенной зоне и установить прямое подключение к ней нет возможности.

2. Дополнительные характеристики

2.1 Дополнительная обработка данных

Часто в рамках задач репликации возникает необходимость дополнительной обработки данных, например реплицировать дополнительную информацию, либо реализовать специальную логику обработки данных. Для этих целей используются виртуальные колонки и пост-процессинг.

- SQL скрипт
Позволяет заполнять виртуальные колонки данными с использованием конструкций SQL
- TCL скрипт
Позволяет реализовывать сложную логику обработки данных. Например, этот механизм имеет доступ к метаданным транзакций, которые можно передавать в виртуальные колонки.
- Пост-процессинг
Этот механизм позволяет выполнять хранимую процедуру в БД-приемнике после каждого цикла заливки данных.

2.2 Мониторинг

- Журнал событий
Журнал содержит консолидированную информацию по всем проектам репликации, отображаются наиболее важные события. Так же имеется возможность просмотра логов выполнения задач.
- Уведомления
В системе есть два механизма рассылки уведомлений о возникновении различных событий – SMTP и SNMP уведомления.
- Мониторинг производительности
В системе имеется функция мониторинга производительности отдельных процессов репликации.

2.3 CLI

Командный интерфейс позволяет управлять задачами репликации без графического пользовательского интерфейса. Данный функционал особенно полезен при автоматизации процессов репликации.

2.4 Обработка ошибок загрузки

В системе есть функционал, который позволяет игнорировать ошибки загрузки данных. Для этого код ошибки нужно добавить в список исключений.