**DaJet MQ — быстрый и надёжный обмен данными !**

**DaJet MQ – это асинхронный обмен сообщениями, основанный на технологиях SQL Server. Простой, эффективный, надёжный. Отлично справляется с сообщениями любого размера. Максимальный размер одного сообщения равен 2 Гб.**

Название DaJet (даджет) образовано от слов *data* (данные) и *jet* (реактивный).

Очереди DaJet "живут" в базах данных. Могут быть созданы в любой из них, в том числе на удалённых серверах SQL Server. Это позволяет настраивать фактически любую схему взаимодействия между базами данных и приложениями, использующими их.

DaJet гарантирует доставку сообщений exactly-once-in-order (ровно один раз строго по порядку). Планируется реализация возможности доступа из 1С к дополнительным стратегиям обработки сообщений.

В случае реализации удалённого взаимодействия между очередями, DaJet берёт на себя все нюансы доставки сообщений из локальной очереди в удалённую. Это включает в себя обработку обрывов связи, доставку сообщений по частям, динамическое изменение маршрутов и прочее.

Безопасность аутентификации и шифрования трафика DaJet может быть обеспечена при помощи сертификатов SQL Server.

**Системные требования.**

1. SQL Server (в том числе Express)

2. OLE DB Provider для SQL Server

**Функциональные возможности.**

1. Простое создание / удаление очередей.

2. Отправка / получение одного или нескольких сообщений в транзакции.

3. Активное ожидание новых сообщений практически без нагрузки на сервер.

4. Автоматический запуск хранимых процедур или внешних приложений по факту поступления сообщений в очередь.

5. Надёжное удалённое взаимодействие "точка-точка" между серверами.

**Преимущества.**

1. Надёжность и гарантии SQL Server.

2. Очень простая и быстрая установка.

3. Программирование на 1С "из коробки".

4. Практически нулевой порог входа для освоения.

5. Возможность использовать для интеграции гетерогенных систем.

5. Если Вы используете SQL Server, значит у Вас уже есть DaJet!

**Планы развития DaJet MQ.**

1. Разработка Web API.

2. Разработка административного web интерфейса.

3. Разработка отчётов мониторинга состояния для 1С.

4. Реализация схемы взаимодействия издатель-подписчик.

5. Реализация управления компенсируемыми распределёнными транзакциями.

6. Упрощение настройки автоматической активации процедур и приложений.

7. Упрощение настройки для аутентификации и шифрования на сертификатах.

**Примеры использования.**

1. Создание очередей на основании плана обмена.

2. Отправка сообщения.

3. Получение сообщений.

4. Активное ожидание новых сообщений.

5. Создание удалённых очередей.

6. Отправка сообщения в удалённую очередь.

7. Удаление очередей.

8. Мониторинг текущего состояния очереди.

**Тест производительности.**

CPU: Intel Core i5-4460 3.20 GHz

RAM: 8 Gb

HDD: 7200 rpm (при копировании больших файлов показывает в среднем 60 Mb/s)

Тестовый файл одного сообщения: 120 Kb (json = СериализаторXDTO.ЗаписатьJSON).

Для получения тестового файла использовался документ "УстановкаЦенНоменклатуры", имеющий 100 позиций, + подчинённые ему записи регистра сведений "ЦеныНоменклатуры". Типовая конфигурация "Управление торговлей" 11.х. Объекты 1С были преобразованы в json при помощи вызова метода "ЗаписатьJSON" сериализатора XDTO.

Количество сообщений: 10 000 шт. x 120 Kb = 1,14 Gb (общий объём данных)

Отправка сообщений из 1С по одному сообщению в транзакции:

50 сек. = 200 msg/s = 23 Mb/s = 20 000 позиций цен / сек.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Получение всех 10 000 сообщений из 1С обработкой на сервере** | | | | |
| Количество сообщений  в 1 транзакции чтения | Общее время  (сек.) | Количество  сообщений  в секунду | Количество  данных  в секунду | Количество  позиций цен  в секунду |
| 1 шт/тран | 50 сек. | 200 msg/s | 23 Mb/s | 20 000 шт/сек |
| 10 шт/тран | 25 сек. | 400 msg/s | 46 Mb/s | 40 000 шт/сек |
| 100 шт/тран | 20 сек. | 500 msg/s | 58 Mb/s | 50 000 шт/сек |

Сообщения получались из очереди с фиксацией транзакции на каждую прочитанную порцию данных и преобразовывались в массив текстовых сообщений в формате json с размещением в оперативной памяти сервера 1С. Запись в базу данных 1С не выполнялась, чтобы это не отражалось на общих показателях тестирования производительности именно DaJet MQ.

Сообщения пересылались / получались как тип данных "nvarchar", то есть использовалось 2 байта на символ. Можно оптимизировать объём трафика, передавая данные как "varchar", например, используя преобразование текста в BASE64 - объём трафика сократится в таком случае в 2 раза. Сообщения ничем не сжимались.

То есть, другими словами, фактически общий объём переданных / полученных данных в рамках одного теста из 10 000 сообщений составлял 2,3 Gb.

Использование типа данных varbinary затруднено из-за особенностей работы с бинарными данными 1С - очень медленно конвертируется тип данных 1С COMSafeArray в ДвоичныеДанные или Строку.

SQL Server, сервер 1С и клиент 1С (обработка) находились на одной машине (сетевые издержки не учитывались).

**Тарифы.**

- DaJet MQ 1.0

(1 год + обновления + 2 час тех. поддержки на время внедрения) = 90 000 руб.

- Техническая поддержка (сроком на 1 месяц, но не более 8 часов) = 24 000 руб.

- Индивидуальные доработки (1 час) = 3 000 руб.