

BIG DATA ДЛЯ ТРАНСПОРТНО-ЛОГИСТИЧЕСКИХ УЗЛОВ



А.А. Навроцкий

Заведующий кафедрой информационных технологий автоматизированных систем БГУИР, кандидат физико-математических наук, доцент



Р.В. Козарь

Студент кафедры информационных технологий автоматизированных систем БГУИР

*Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: navrotsky@bsuir.by, pozitr0n.kozarroman@gmail.com*

Abstract. Рассматриваются вопросы использования методов Big Data в системе 1С: Предприятие.

В настоящее время большинство предприятий занимаются решением вопросов, связанных с транспортной логистикой. Как правило, для построения своей логистической системы, предприятия пользуются услугами компаний, специализирующихся на решении задач по управлению процессами перевозок. Если предприятие предполагает использовать только собственную транспортную базу, не привлекая сторонних перевозчиков, оно может столкнуться с рядом проблем:

- изношенность подвижного состава;
- плохая информационная поддержка процесса транспортировки;
- сложности построения маршрутов перевозки;
- недогруз подвижного состава;
- необходимость страхования груза и транспортных средств;
- сложности при организации взаимодействия различных видов транспорта;
- недостатки имеющегося ПО для данной области.

В настоящее время на рынке имеется большое число программных продуктов для решения задач учета расходов предприятий. Однако большая часть из них является продуктами для ведения складского учета и плохо подходит для использования в транспортной логистике. Причем, даже использование такой мощной платформы, как 1С: Предприятие не всегда гарантирует необходимую наглядность по расходам и корректность их отображения в управленческом учете с разбиением, например, по рейсам. Имеющиеся на рынке иностранные программы, например Shipnet, не учитывают национальных особенностей, а поэтому их сфера применения ограничена.

Для решения задач транспортной логистики на предприятии был разработан программный модуль, который подключается как внешний модуль в конфигурацию 1С: Предприятие «Управление торговлей». Модуль обладает обширным кругом возможностей, основными среди которых являются:

- распределение заказов по нескольким курьерам/машинам;
- прокладка оптимального маршрута доставки по клиентам/поставщикам;
- ручная корректировка маршрута;
- выгрузка маршрута в навигатор;

- обработка документов по поступлению и реализации товаров и услуг;
- получение полной детализации по процессу доставки товаров.

Программный модуль используется в качестве рабочего стола для менеджера по приему заказов, что позволяет организовать быстрый подбор товара, формирование заказа и определение маршрута доставки.

Так как менеджер ежедневно обрабатывает сотни заказов, для нескольких тысяч товаров, то встает проблема передачи больших объемов данных через разработанный модуль. Это требует разработки специальных технологических инструментов и решений для проведения производительной обработки с учетом динамического роста объема исходных данных.

Установлено, что объемы данных, проходящие через модуль грузоперевозок, являются большими, так как присутствуют три основных признака Big Data:

volume – большое количество данных, независимо от масштаба доступных ресурсов для проведения их обработки (количество обрабатываемых заказов может достигать нескольких сотен в сутки);

variety – разнородность и слабая структурированность данных (заказы покупателей и поставщиков вообще не структурированы и разнородны);

velocity – необходимость быстрой обработки данных с предельно оперативной выдачи результата (клиенты ожидают быстрого и точного выполнения заказа).

При использовании 1С: Предприятие необходимость в обработке больших данных может возникнуть в следующих случаях:

- вычисление производных от больших данных (перезаполнение регистров при изменении первичных данных; проведение большого количества документов; заполнение новых реквизитов в больших таблицах);

- выгрузка, загрузка и конвертация данных (слияние (консолидация) баз, требующая переноса значительной части данных в другую базу; обмен данными вследствие обработки больших данных; восстановление испорченных данных из копии базы).

Для повышения эффективности обработки больших объемов данных в модуле было выполнено следующее:

1. *Отключение регламентных заданий в СУБД.* Выполнение регламентных заданий в СУБД может серьезно снижать производительность обработки из-за ожиданий на блокировках и очередях аппаратных ресурсов. Поэтому их лучше временно отключать, но желательно обеспечить их автоматическое включение после завершения обработки, в том числе аварийного.

2. *Оптимизация записи объекта.*

- 2.1. Минимизация ожидания на блокировках данных. У регистров отключается разделение итогов, а после больших многопоточных обработок пересчитываются итоги в периоды минимальной нагрузки. Конфигурация переводится на управляемые блокировки и используется версионный режим MS SQL (read_committed_snapshot). Для анализа ожиданий на блокировках данных используются центр управления производительностью (для управляемых блокировок и блокировок СУБД) и анализ технологического журнала из подсистемы «Инструменты разработчика» (для управляемых блокировок).

- 2.2 Запись в режиме загрузки. Если допустимо, то используется запись в режиме загрузки (Объект.ОбменДанными.Загрузка = Истина). В этом режиме должен выполняться очень незначительный процент кода обработчиков и подписок событий записи, а платформа отключает ряд своих внутренних обработчиков и потому выполняется меньше вычислений (проверка уникальности кодов и номеров объектов).

- 2.3. Отключение итогов регистров. Оправдано для больших обработок изменения регистров, однако в этом случае перестают работать виртуальные таблицы регистров и многие отчеты, поэтому необходимо обеспечить после успешного и аварийного завершения обратное включение итогов. Так как после включения итогов они могут пересчитываться платформой

некорректно, то желательно делать полный их пересчет.

2.4. Отключение авторегистрации изменений. В распределенных базах допускается в случае, если обработка выполняется одновременно во всех базах и есть уверенность, что изменения объектов общих данных будут одинаковы во всех базах.

2.5. Отключение RLS. По-возможности, следует выполнять запись без использования механизма RLS (ограничения доступа к данным на уровне записей), так как запросы RLS могут служить причиной серьезных потерь производительности. В качестве вариантов отключения данных механизмов можно использовать привилегированный режим, так же можно использовать набор ролей, устанавливающий пустые RLS на нужных таблицах.

3. *Использование многопоточности.* Относительная скорость наращивания однопоточной производительности аппаратных ресурсов постоянно падает по технологическим причинам. Поэтому аппаратные ресурсы все больше растут в сторону многопоточной производительности, например, растет количество ядер в процессорах. Использование многопоточности усложняет программу, но делает ее более масштабируемой по скорости работы.

3.1. Разбиение на порции. Для многопоточной обработки необходимо распределить набор объектов между потоками. Обработка не должна быть чувствительной к порядку обработки порций (обработка любой порции не зависит от успеха обработки других). Нечувствительными к порядку обработки порций могут быть процессы загрузки-выгрузки данных, объединение (замена) дублей, свертка регистра, восстановление последовательности (документов) по разным комбинациям значений измерений, универсальная обработка объектов, проведение документов, не использующих результаты проведения других документов, заполнение реквизитов, пометка удаления и удаление. Примером чувствительных к порядку порций обработок может быть восстановление последовательности (документов) по одной комбинации значений измерений или проведение документов, использующих результаты проведения других документов.

3.2. Многопоточный обмен данными. При переводе в многопоточный режим усложняется логика процессов передачи данных (рис. 1).

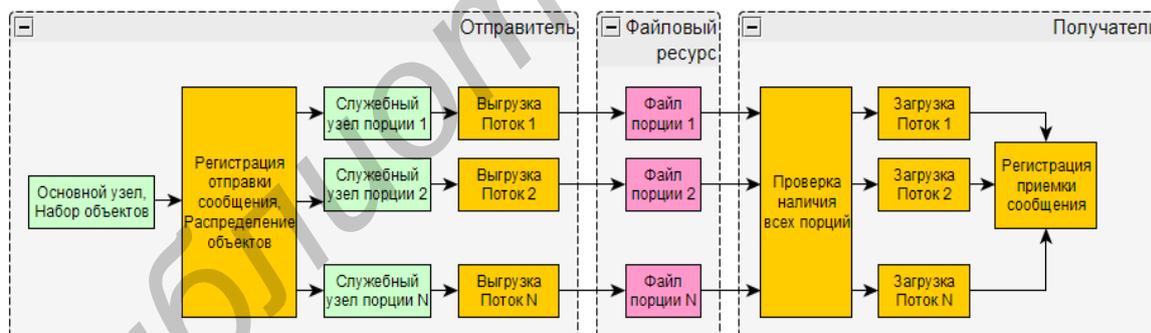


Рис 1. Логика процессов передачи данных

При оценке степени увеличения производительности следует учитывать:

- совокупную многопоточную производительности системы. Для ее оценки можно использовать Многопоточное тестирование производительности сервера 1С – СУБД;
- накладные расходы на многопоточность (построение карты порций, слияние результатов и др.);
- время ожидания при использовании блокировки данных (желательно использовать режим управляемых блокировок и разделения итогов регистров);
- количества потоков.

Многопоточное ускорение обработки набора объектов можно выразить формулой 1:

$$A = \frac{P + M \cdot T}{P + G + E + M \cdot (R + W \cdot T) + (1 - W) \cdot \frac{M \cdot T}{N}} \quad (1)$$

где, A – коэффициент ускорения (отношение длительности выполнения однопоточной обработки к длительности выполнения многопоточной обработки);

P – длительность не распараллеливаемой части вычислений для набора объектов в целом, общей для обоих вариантов обработки (например, выполнение сложного запроса для получения ключей объектов набора);

T – длительность вычислений на один объект в однопоточном режиме;

M – количество объектов;

G – длительность вычислений для набора объектов в целом, необходимых только для многопоточного варианта обработки (например, слияние результатов);

N – количество потоков;

E – длительность вычислений для порции объектов, необходимых только для многопоточного варианта обработки (например, сохранение результата порции);

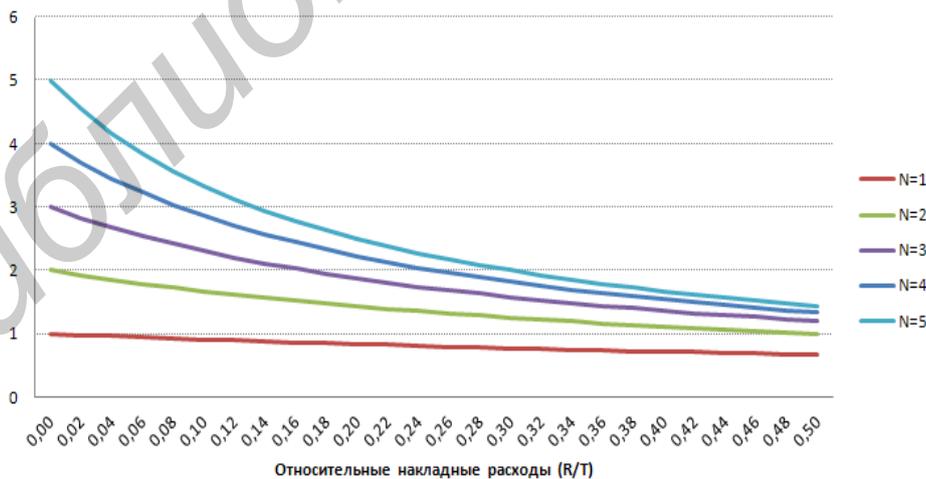
W – степень конкуренции (ожиданий, обусловленных многопоточным режимом), находится в диапазоне $[0;1]$ и зависит от N , (основные типы ожиданий – на блокировках данных между потоками и на очередях аппаратных ресурсов);

R – длительность дополнительных вычислений по объекту в многопоточном режиме (например, привязка и отвязка объекта от порции)/

При достаточно большом количестве объектов и отсутствии конкуренции между потоками формула ускорения выглядит следующим образом:

$$A = \frac{T}{R + \frac{T}{N}} = \frac{1}{\frac{R}{T} + \frac{1}{N}} \quad (2)$$

Таким образом оценив значения R и T , можно достаточно точно оценить ускорение для многих обработок. Чем меньше будут относительные накладные расходы (R по отношению к T), тем больше будет эффект от увеличения количества потоков.



Из рис. 2. видно, что однопоточный режим практически всегда быстрее многопоточного с количеством потоков ($N = 1$), что обусловлено необходимостью выполнения дополнительных вычислений. Поэтому при $N = 1$ многопоточный режим не эффективен.

С увеличением относительных накладных расходов уменьшается эффективность многопоточной обработки.

Литература

[1]. Фирма 1С-Битрикс. Подходы и инструменты работы с Big Data – все только начинается [<https://habrahabr.ru/company/bitrix/blog/256551/>] / Фирма 1С-Битрикс. – Электрон. текстовые дан. – Режим доступа <https://habrahabr.ru/company/bitrix/blog/256551/>, свободный.

[2]. Фирма 1С-Битрикс. Архитектура и технологические подходы к обработке Big Data на примере «1С-Битрикс BigData: Персонализация» [<https://habrahabr.ru/company/bitrix/blog/272041/>] / Фирма 1С-Битрикс. – Электрон. текстовые дан. – Режим доступа <https://habrahabr.ru/company/bitrix/blog/272041/>, свободный.

[3]. Microsoft Inc., 7 Things You Must Know About Big Data Before Adoption [<http://bigdataanalyticsnews.com/7-things-must-know-big-data-adoption/>] / Microsoft Inc. – Электрон. текстовые дан. – Режим доступа <http://bigdataanalyticsnews.com/7-things-must-know-big-data-adoption/>, свободный.

[4]. 1С-Прогресс, Многопоточная обработка данных в системе 1С: Предприятие [<http://1sprogress.ru/mnogopotchnaya-obrabotka-dannyx-v-1s.html>] / 1С-Прогресс. – Электрон. текстовые дан. – Режим доступа <http://1sprogress.ru/mnogopotchnaya-obrabotka-dannyx-v-1s.html>, свободный.

[5]. Фарит Насипов (infostart.ru), Как ускорить 1С – Многопоточная обработка данных [<http://infostart.ru/public/306865/>] / Фарит Насипов (infostart.ru). – Электрон. текстовые дан. – Режим доступа <http://infostart.ru/public/306865/>, свободный.