Как ускорить 1С – Многопоточная обработка данных

Периодически возникает необходимость в ускорении некоторых участков кода, в которых при беглом ознакомлении с ними нет никакого потенциала для ускорения.

Рассмотрим в качестве примера выгрузку и/или загрузку большого количества данных. И в данном случае существенную роль в установлении «планки производительности» начинает играть непосредственно производительность аппаратных средств, причем сами процедуры загрузки-выгрузки выполняются достаточно длительное время, часто измеряемое в часах.

Не все знают, что в платформе «1С: Предприятие 8» часть операций можно выполнять параллельно, путем одновременного выполнения в несколько потоков.

Такая многопоточная обработка вполне применима в случае, когда нужно обработать независимые блоки данных. В частности, создать большой объем элементов данных, осуществить проведение документов, не пересекающиеся по значениям набора измерений, сделать массовое обновление элементов справочников, выполнить загрузку данных в регистры и т.д.

Сформулируем задачу следующим образом: провести обновление реквизита «Цена» для всего справочника «Товары». Количество элементов справочника «Товары» равно 100 000.

Самым простым, естественным решением, выступает нижеприведенный код обработки, которая очень быстро пишется, но при этом очень долго работает:

```
&НаСервере
Процедура ОбновитьЦену()
    ВремяНачала = ТекущаяДата();

    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |   Товары.Ссылка
    |ИЗ
    |   Справочник.Товары КАК Товары";

    ТаблицаТоваров = Запрос.Выполнить().Выгрузить();

    Наценка = 1.10;

    Для каждого ТекСторока Из ТаблицаТоваров Цикл
        ТоварОбъект = ТекСторока.Ссылка.ПолучитьОбъект();
        ТоварОбъект.Цена = ТоварОбъект.Цена*Наценка;
        ТоварОбъект.Записать();
    КонецЦикла;

    Длительность = ТекущаяДата()-ВремяНачала;
```


Бесплатные материалы проекта:
http://курсы-по-1с.рф/free/

Электронная почта: support@Kursy-po-1C.ru
Телефоны: **8-800 1000-613**, +7(495) 7777-096

```
    Сообщить ("Длительность: " + Длительность + " сек.");

КонецПроцедуры
```

В нашем случае обработка выполнялась 1 187 секунд или 19,7 минуты.

Но, к счастью, есть другой путь, при котором мы воспользуемся многопоточным выполнением. Тогда код обработки будет примерно следующим:

```
&НаСервере
Процедура ОбновитьЦену()

    ВремяНачала = ТекущаяДата();

    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |   Товары.Ссылка
    |ИЗ
    |   Справочник.Товары КАК Товары";

    ТаблицаТоваров = Запрос.Выполнить().Выгрузить();

    // определяем максимальное количество потоков
    ЧислоПотоков = 8;

    ЧислоСтрокВТаблице = ТаблицаТоваров.Количество();

    // объем порции данных для обработки каждым потоком
    РазмерПорции = Цел(ЧислоСтрокаВТаблице/ЧислоПотоков);

    // массив, где будут храниться фоновые задания
    МассивЗаданий = Новый Массив;

    Для НомерПотока = 1 По ЧислоПотоков Цикл

        // определяем индекс для начала обработки данных данным потоком
        // разные потоки обрабатывают разные части таблицы
        ИндексНачала = (НомерПотока - 1)*РазмерПорции;

        Если (НомерПотока = ЧислоПотоков) Тогда
            // если это последний поток, то он обрабатывает все оставшиеся данные
            // т.к. число потоков может не быть кратно количеству строк в таблице
            РазмерПорции = ЧислоСтрокВТаблице-
(ЧислоПотоков*РазмерПорции)+РазмерПорции;
        КонецЕсли;

        // определяем массив параметров для процедуры
        НаборПараметров = Новый Массив;
        НаборПараметров.Добавить(ТаблицаТоваров);
        НаборПараметров.Добавить(ИндексНачала);
        НаборПараметров.Добавить(РазмерПорции);

        // запуск фонового задания
```

```
        Задание = ФоновыеЗадания.Выполнить("ОбщийМодуль1.ОбновитьЦенуТовара",
НаборПараметров);

        // добавляем задание в массив, чтобы потом отследить выполнение
        МассивЗаданий.Добавить(Задание);

    КонецЦикла;

    // проверим результат выполнения фоновых заданий
    Если МассивЗаданий.Количество() > 0 Тогда
        Попытка
            ФоновыеЗадания.ОжидатьЗавершения(МассивЗаданий);
        Исключение
            // действия в случае ошибки
        КонецПопытки;
    КонецЕсли;

    Длительность = ТекущаяДата()-ВремяНачала;

    Сообщить("Длительность: " + Длительность + "сек.");

КонецПроцедуры
```

Код общего модуля:

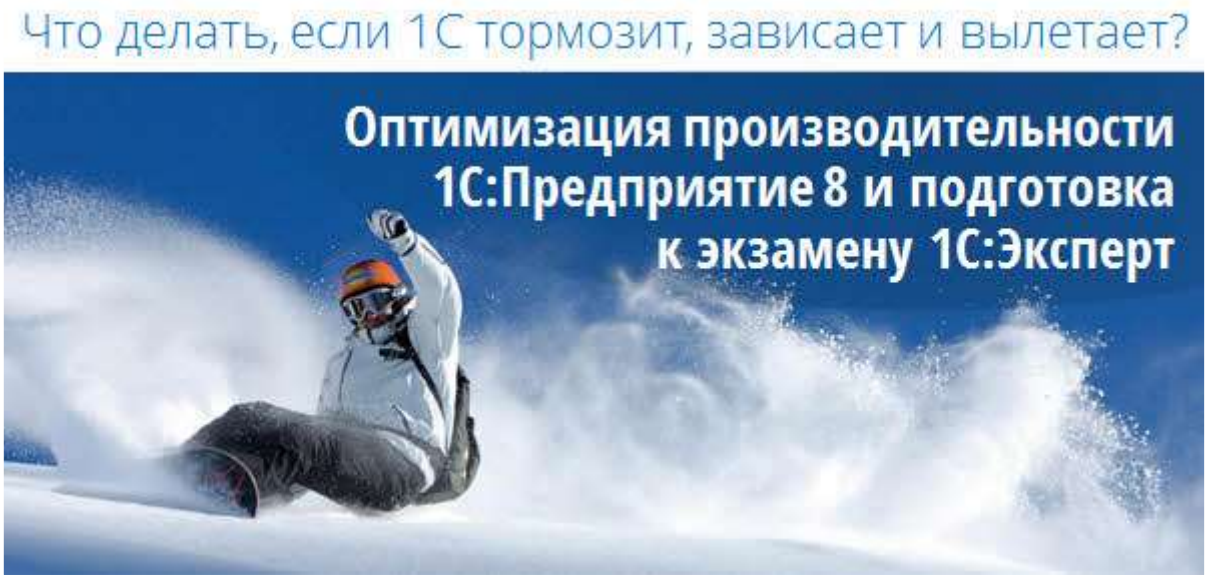
```
Процедура ОбновитьЦенуТовара (ТаблицаТоваров, ИндексНачала, РазмерПорции) Экспорт

    Наценка = 1.10; // наценка 10%

    // обновляем цену только для определенной части таблицы
    Для Сч = 1 По РазмерПорции Цикл
        Индекс = ?(Сч=1, ИндексНачала, Индекс+1);

        ТоварОбъект = ТаблицаТоваров[Индекс].Ссылка.ПолучитьОбъект();
        ТоварОбъект.Цена = ТоварОбъект.Цена * Наценка;
        ТоварОбъект.Записать();
    КонецЦикла;

КонецПроцедуры
```

В нашем случае использовалось восемь потоков, обновление цен было выполнено за 859 секунд или 14,3 минуты. Код выполнялся на виртуальной машине с одним процессором без RAID массивов. На реальном и хорошем «железе» выигрыш в скорости будет существенно больше.

Заметим, что данную задачу можно решить по-разному, мы привели лишь один пример реализации. Мы ведем обработку одну и ту же Таблицу Значений параллельно разными потоками, что приводит к выигрышу в скорости. Считаем нужным предупредить: не нужно устанавливать слишком большое количество потоков, так как большого прироста скорости вы от этого все равно не получите, а стабильность работы может нарушиться.

Наилучшим будет использование 8-10 потоков, а их оптимальное количество можно определить экспериментально.

Попробуйте самостоятельно провести эксперимент, базу и обработки можно скачать здесь.



Если Вам понравились эти материалы, мы приглашаем Вас пройти **предварительную регистрацию** на тренинг «**Оптимизация производительности 1С:Предприятие 8 и подготовка к 1С:Эксперт**» по адресу: http://kursy-po-1c.ru/optimize1C

Чему Вы научитесь после прохождения курса:

- самостоятельно решать проблемы производительности
- проводить анализ системы и выявлять «узкие места» в плане производительности
- находить медленные запросы, наиболее сильно влияющие на систему, и оптимизировать их
- читать и понимать план запроса
- выявлять почему данная конкретная операция выполняется медленно
- оценивать загруженность оборудования
- выявлять и решать проблемы избыточных блокировок
- выявлять и решать проблемы взаимных блокировок
- работать с ЦУП и Тест-Центр
- работать с облачными сервисами контроля производительности
- распараллеливанию кода на 1С
- расследовать и решать проблемы стабильности
- настраивать кластер серверов наиболее оптимальным для производительности образом
- настраивать отказоустойчивый кластер серверов 1С
- настраивать и использовать технологический журнал для решения проблем производительности и стабильности

Бесплатные материалы проекта:
http://курсы-по-1с.рф/free/

Электронная почта: support@Kursy-po-1C.ru
Телефоны: **8-800 1000-613**, +7(495) 7777-096