ВЫЧИСЛЕНИЕ ВЕРХНЕЙ ОЦЕНКИ ИЗБЫТОЧНОСТИ ДАННЫХ И ЕЕ ИСПОЛЬЗОВАНИЕ ПРИ ОПРЕДЕЛЕНИИ ВРЕМЕНИ ДОСТУПА МОДУЛЕЙ К БД В РЕАЛЬНОМ ВРЕМЕНИ

Мирошник С. Н.[1]**,**
(*Вычислительный центр им. А.А. Дородницына РАН ФИЦ «Информатика и управление» РАН, Москва*)
Гончар Д. Р.[2]
(*Вычислительный центр им. А.А. Дородницына РАН ФИЦ «Информатика и управление» РАН, Москва, Московский физико-технический институт* (*государственный университет*)*, Москва*)

Исследуется задача минимизации избыточности информации в БД реального времени, что влияет на время доступа к БД и реализацию модулей. Задан набор программных модулей, которые используют информацию из набора полей, с известной частотой заполняющихся информацией в реальном времени. Задача решена, если работа всех модулей завершена к заданному сроку. Сложность данной постановки задачи и в том, что модули могут быть зависимыми, то есть работают в определенной последовательности, и в том, что на оптимизацию в режиме реального времени выделить достаточные вычислительные ресурсы и время затруднительно. Поэтому решение задачи в разрабатываемой авторами инструментальной САПР систем реальном времени разделяется на два этапа: (а) предварительный этап (не в реальном времени), на котором осуществляется формирование групп близких модулей и (б) этап решения задачи в реальном времени, используя выполненную на предварительном этапе оптимизацию расположения полей в файлах. Определяется избыточность информации трёх типов: внутрифайловая, межфайловая, внутримодульная. Предлагается специальная модель спроектированной БД и построены аналитические формулы для вычисления количества неиспользуемых модулями полей.

Ключевые слова: системы управления базами данных, системы реального времени, эвристические алгоритмы, оптимизация.

---

[1] *Сергей Николаевич Мирошник, к.ф.-м.н., н.с.* (*rtsccas@ya.ru*).
[2] *Дмитрий Русланович Гончар, к.т.н., с.н.с.* (*trpl@ya.ru*).

1. Введение

Рассматривается база данных (БД) реального времени, входная информация в которую поступает от датчиков некоторого исследуемого объекта после соответствующей обработки (оцифровки, калибровки и т.д.) N программными модулями. Трудность решения задачи в реальном времени может состоять в том числе в том, что модули из $\{M\}_N$ могут быть зависимыми, т.е. должны работать в определенной последовательности. Это и многое другое накладывает особые требования при проектировании соответствующей структуры БД реального времени, в частности, к такой ее характеристике как избыточность [1, 2, 7, 8]. В данной постановке под избыточностью понимается неиспользуемые модулями поля. Эти поля участвуют в процедуре поиска модулями информации, тем самым увеличивают время доступа модулей к своим полям. Отсюда требования минимизировать такую избыточность.

Отметим, что решение задачи в разрабатываемой авторами инструментальной системе автоматизации проектирования систем реальном времени разделяется на два больших этапа.

A. Предварительный этап (не в реальном времени), на котором в том числе осуществляется формирование групп близких модулей.

B. После этого решается задача в реальном времени, используя выполненную на предварительном этапе оптимизацию расположения полей в файлах.

2. Постановка задачи

Задан набор программных модулей $\{M\}_N$ которые используют информацию из набора полей $\{\phi\}_r$. Здесь N – число модулей, r – число полей. Все поля пронумерованы натуральным рядом чисел. Предполагаем, что поля модулей расположены подряд, т.е. занимают связный сегмент в наборе полей. Каждый модуль задан своим первым полем и длиной записи, т.е. числом полей. Здесь не рассматривается случай, когда запись содержит неиспользуемые поля. Поля $\{\phi\}_r$ с известной частотой заполня-

ются информацией в реальном времени. Задача решена, если работа всех модулей завершена и часто к заданному сроку.

Введем некоторые ключевые понятия. Определим избыточность информации трёх типов.

1. *Внутрифайловая избыточность* I_1 образуется из разности длин l_i модуля M_i, входящего в файл F, и длиной L этого файла. Она вычисляется по формуле

$$I_1(F) = Ln - \sum_{i=1}^{n} l_i,$$

где n – число модулей в файле F.

2. *Межфайловая избыточность* I_2 есть число повторяющихся полей записей $L_1, \ldots, L_k$ файлов $F_1, \ldots, F_k$. Она вычисляется по формуле

$$I_2(F_1,\ldots, F_k) = \sum_{i=1}^{k} L_i - r,$$

где r – суммарное число полей всех файлов. Приведённая формула может быть использована для вычисления количества повторяемости используемых полей модулей, входящих в файл F:

$$I_2(F) = \sum_{i=1}^{n} l_i - L,$$

где n – число модулей в файле F, L – длина файла. Слагаемое L служит для вычисления повторяемости полей. Без L эта формула означает просто число используемых полей файла F. В частности, $I_1(F) + I_2(F)$ есть число полей (используемых и неиспользуемых) всех модулей файла F. Эта сумма не корректна для отдельного файла F, но имеет смысл для набора файлов.

3. *Внутримодульная избыточность* есть число неиспользуемых модулем M полей в записи L модуля.

Для минимизации избыточности предлагается распределять все модули между файлами $F_1, \ldots, F_k$, причем каждый модуль принадлежит только одному файлу.

Вычислим $I_1 = \sum_{i=1}^{k} I_1(F_i)$ и $I_2(F_1,...,F_k)$. Число $I = I_1 + I_2$ является *качеством спроектированной БД*.

Объединение модулей в группы основано на определении *близости модулей* [2–5].

Пусть имеется набор модулей $\{M\}_s$, где s – число модулей в наборе. Включение модуля M_{s+1} в набор $\{M\}_s$ изменяет внутрифайловую избыточность на величину $\Delta I_1 = I_1^{s+1} - I_1^s$, где I_1^s – внутрифайловая избыточность набора $\{M\}_s$, I_1^{s+1} – внутрифайловая избыточность набора $\{M\}_{s+1}$ после включения M_{s+1} в $\{M\}_s$. С другой стороны, важной информацией для близости M_{s+1} к $\{M\}_s$ является количество совпадающих полей модуля M_{s+1} и $\{M\}_s$. Для вычисления совпадающих полей можно воспользоваться формулой, приведённой выше (для $I_2(F)$).

Определение. Модуль M_{s+1} и набор $\{M\}_s$ являются близкими и M_{s+1} может быть включен в состав набора $\{M\}_s$, если $\Delta I_1 \le I_2(M_{s+1},\{M\}_s)$.

3. Оценка внутрифайловой избыточности

На примере одной группы F вычислим внутрифайловую избыточность I_1, а именно той ее части, которая влияет на время доступа модулей к своим полям группы F.

Пусть сформирована группа F близких модулей. В этой группе число модулей есть n, и L – число полей, используемых всеми этими модулями. В группе F определим опорный модуль M, для которого l есть наибольшая длина среди длин всех модулей группы, и $\{\phi\}_l$ есть поля, используемых опорным модулем. Разделим набор модулей $\{M\}_n$ группы F на внутренние $\{M\}_v^{in}$ и внешние $\{M\}_w^{out}$. Поля группы F есть $\{\phi\}_L$. Здесь $\{M\}_n = \{M\}_v^{in} \cup \{M\}_w^{out}$. Числа v и w будут вычислены в п. 3.1 и в п. 3.2 соответственно. Далее, для $\{M\}_v^{in} : \{\phi\}_l^{in} = \{\phi\}_l$; $\{\phi\}_L^{out} = \{\phi\}_l^{in} \cup \{\phi\}_d$, $n = v + w$.

Здесь $L = l + d$, где d – допустимое увеличение количества полей опорного модуля M за счет тех полей внешних модулей, которые являются близкими к M.

Воспользуемся приведенными выше формулами для вычисления I_1 и I_2:

$$I_1 = L \cdot n - \sum_{i=1}^{n} l_i , \qquad I_2 = \sum_{i=1}^{n} l_i - L .$$

Здесь L – длина набора полей в $\{\phi\}_L$, l_i – длина записи модуля M_i из $\{M\}_n$, $i = 1, \ldots, n$, n – число модулей.

Построим оценку той части $\{\phi\}_L$ неиспользуемых модулями из $\{M\}_n$ полей, которые влияют на поиск своих полей в реальном времени. Предполагается, что этот поиск осуществляется с первого поля набора $\{\phi\}_L$. Заметим, что если в БД много групп близких модулей и просуммировать эти части полей, то получим большую задержку времени. Упомянутая оценка неиспользуемых полей состоит из двух частей $\tilde{I}_1^{in}$ и $\tilde{I}_1^{out}$.

Здесь $\tilde{I}_1^{in}$ – не используемые поля внутренних относительно опорного близких модулей $\{M\}_v^{in}$, соответственно $\tilde{I}_1^{out}$ – для внешних близких модулей $\{M\}_w^{out}$.

Построим аналитические формулы для вычисления обеих частей. Для этого воспользуемся гипотетической моделью, в которой предполагается, что группа модулей $\{M\}_n$ состоит из достаточно большого числа модулей. Это необходимо, чтобы построить верхнюю оценку избыточности I_1. В этом случае можно найти все модули, близкие к опорному. Выберем из набора $\{M\}_n$ опорный модуль M.

3.1. ОПРЕДЕЛЕНИЕ ВНУТРИФАЙЛОВОЙ ИЗБЫТОЧНОСТИ $\tilde{I}_1^{in}$

Вычислим оценку $\tilde{I}_1^{in}$, обозначающую ту часть полей I_1^{in}, которые «мешают» модулям находить свои поля в группе $\{M\}_n$. Найдем все внутренние близкие модули. Формула для вычисления I_1^{in} всех неиспользуемых полей модулями $\{M\}^{in}$ есть

$$I_1^{in} = \tilde{\nu} \cdot l - \sum_{i=1}^{\tilde{\nu}} l_i,$$

где $\tilde{\nu}$ – число внутренних близких модулей, включая опорный, $\tilde{\nu} = \nu + 1$, $\sum_{i=1}^{\tilde{\nu}} l_i$ – сумма длин записей всех близких внутренних модулей включая опорный.

Построим конечные формулы для вычисления $\tilde{\nu}$ и $\sum_{i=1}^{\tilde{\nu}} l_i$.

Упорядочим близкие внутренние модули $\{M\}_{\nu}^{in}$ по числу полей:

$l > l_1 \geq l_2 \geq \ldots \geq l_{\nu}$.

Здесь $l_1 = l - 1$, $l_2 = l_1 - 1, \ldots, l_{\nu} = l_{\nu} - 1$.

Разделим эти модули на подгруппы.

Первая подгруппа: $\nu_1 = 1$ – есть опорный модуль M с числом полей l; вторая: $\nu_2 = 2$– модули длиной $l - 1$ в количестве 2 и т.д. Всего подгрупп: $t = \left[\frac{1}{2} l\right] + 1$. Число модулей $\tilde{\nu} = \sum_{i=1}^{t} \nu_i$, $\nu_i = i$, $\nu = \tilde{\nu} - 1$ (ν – число внутренних близких модулей без опорного). Получаем: $\tilde{\nu} = \frac{1}{2} t(t+1)$.

Далее, число используемых модулями полей в подгруппе ν_i есть

$$\tilde{l}_i = i(l - (i-1)), \; i = 1, \ldots, t.$$

Тогда $\sum_{i=1}^{\tilde{\nu}} l_i = \sum_{j=1}^{t} \tilde{l}_j$, или:

$$\sum_{i=1}^{\tilde{\nu}} l_i = \sum_{j=1}^{t} (l+1) \sum_{j=1}^{t} j - \sum_{j=1}^{t} j^2.$$

Воспользуемся известной формулой:

$$\sum_{i=1}^{t} i^2 = \frac{1}{6} t(t+1)(2t+1).$$

После несложных вычислений получаем:

$$\sum_{i=1}^{t} \tilde{l}_i = \frac{1}{6} t(t+1)(3l - 2t + 2).$$

Подставляем в I_1^{in} выражения для $\tilde{v}$ и $\sum_{i=1}^{t} \tilde{l}_i$, получаем:

$$I_1^{in} = \frac{1}{3} t(t^2 - 1).$$

Рассмотрим подробнее структуру полей I_1^{in}. Все поля близких внутренних модулей можно разделить на 3 части:

$$\tilde{v} \cdot l = \tilde{I}_1^{in} + \sum_{i=1}^{\tilde{v}} l_i + (I_1^{in} - \tilde{I}_1^{in}).$$

Как следует из способа упорядочивания внутренних близких модулей все неиспользуемые ими поля I_1^{in} состоят из двух равных частей: $I_1^{in} = 2 \cdot \tilde{I}_1^{in}$. Отсюда:

$$\tilde{I}_1^{in} = \frac{1}{6} t(t^2 - 1), \text{ где } t = \left[\frac{1}{2} l \right] + 1.$$

3.2. ВЫЧИСЛЕНИЕ ВНУТРИФАЙЛОВОЙ ИЗБЫТОЧНОСТИ $\tilde{I}_1^{out}$

Вычислим аналогичную оценку для внешних модулей $\tilde{I}_1^{out}$.

Пусть число d – допустимое расширение полей базового модуля M с записью l. Предполагается, что поля этого расширения будут использоваться внешними близкими модулями к опорному модулю M. Вычислим max d.

Рассмотрим модуль $\tilde{M}$ с числом полей $\tilde{l}$ ($\tilde{l} \le l$). Поля модуля $\tilde{M}$ имеют общие поля с полями модуля M вместе с расширением d. Пусть среди полей модуля $\tilde{M}$ есть поле, не входящее в состав полей модуля M вместе с расширением d. Для того чтобы вычислить max d, полагаем $\tilde{l} = l$. Покажем, что $\tilde{M}$ не является близким к M. Тогда с помощью формул для I_1, I_2 и определения близости модулей получаем:

$$I_1 = 2(l + d + 1) - (l + d + \tilde{l}), \quad I_2 = (l + d + \tilde{l}) - (l + d + 1).$$

Пояснение к формулам.

Здесь $(l + d + 1)$ – общее число полей, используемых двумя модулями – M и $\tilde{M}$, $(l + d)$ – поля модуля М вместе с расширением, l – длина модуля $\tilde{M}$. Число избыточных полей для двух модулей – I_1. Число общих полей для M и $\tilde{M}$ есть I_2. Согласно определению близости, M и $\tilde{M}$ не являются близкими, если $I_1 > I_2$. Отсюда: $l - 1 < d + 2$, или $\max d = l - 2$.

Количество всех избыточных полей внешних модулей есть

$$I_1^{out} = (l+d)\tilde{w} - \sum_{i=1}^{\tilde{w}} l_i \, .$$

Здесь $\tilde{w}$ – общее число внешних модулей (включая опорный).

Построим формулы для $\tilde{w}$ и $\sum_{i=1}^{\tilde{w}} l_i$.

Поля I_1^{out}, так же как и $\tilde{I}_1^{in}$, увеличивают время поиска модулями своих полей. Построим все внешние модули, близкие к опорному модулю M. Первое поле опорного модуля есть ϕ_s.

Упорядочим внешние близкие модули. Способ упорядочивания другой, чем в п. (3.1). Выберем модуль M_1 тоже длиной записи l, но с первым полем ϕ_{s-1}, и найдем все близкие внешние модули с длиной записи меньше l, но первое поле которых осталось прежним: ϕ_{s-1}.

Количество таких модулей есть: $w_1 = \left[\frac{1}{2}\tilde{l}_1\right] - 0$, где $\tilde{l}_1 = l + 1$.

Следующий модуль M_2 в процедуре упорядочивания модулей имеет также длину l, но первое поле для M_2 есть ϕ_{s-2}. Напомним, что все поля пронумерованы натуральным рядом чисел. Находим все близкие внешние модули к набору из M и M_1, для которых длины записей меньше l, но первое поле попрежнему ϕ_{s-2}. Количество таких модулей есть:

$$w_2 = \left[\frac{1}{2}\tilde{l}_2\right] - 1, \text{где } \tilde{l}_2 = l + 2.$$

В общем виде $w_i = \left[\frac{1}{2}\tilde{l}_i\right] - (i-1)$, где $\tilde{l}_i = l + i,\ i = 1,\ldots,d$. Допустимое расширение полей опорного модуля, как показано выше, есть d. Теперь $w = \sum_{i=1}^{d} w_i$ или $w = \sum_{i=1}^{d} w_i \left[\frac{1}{2}(l+i)\right] - \sum_{i=1}^{d}(i-1)$.

После несложных преобразований:

$$w = \frac{1}{2}\left(l \cdot d + \left]\frac{1}{2}d^2\right[\right) - \frac{1}{2}d(d-1) \text{ (без опорного модуля).}$$

Построим формулу для $\sum_{i=1}^{\tilde{w}} l_i$ – число полей используемых внешними близкими модулями. Выражение для $\sum_{i=1}^{\tilde{w}} l_i$ можно записать как

$$\sum_{i=1}^{\tilde{w}} l_i = \sum_{i=1}^{d}\sum_{t=1}^{w_i}(l-(t-1)) + l\,.$$

Здесь $\tilde{w} = w + 1$. Слагаемое l в формуле означает, что учитывается опорный модуль.

Выражение для $\sum_{i=1}^{\tilde{w}} l_i$ запишем в виде:

$$\sum_{i=1}^{\tilde{w}} l_i = \sum_{i=1}^{d}\sum_{t=1}^{w_i} l - \sum_{i=1}^{d}\sum_{t=1}^{w_i} t + \sum_{i=1}^{d}\sum_{t=1}^{w_i} 1 + l$$

или

$$\sum_{i=1}^{\tilde{w}} l_i = l\sum_{i=1}^{d} w_i - \sum_{i=1}^{d}\frac{1}{2}w_i(w_i+1) + \sum_{i=1}^{d} w_i + l$$

окончательно

$$\sum_{i=1}^{\tilde{w}} l_i = l \cdot \tilde{w} - \left]\sum_{i=1}^{d}\frac{1}{2}{w_i}^2\right[+ \left[\frac{1}{2}w\right]$$

Таким образом построены формулы для $\tilde{w}$ и $\sum_{i=1}^{\tilde{w}} l_i$. Теперь легко вычислить I_1^{out}.

Рассмотрим подробнее структуру I_1^{out}. Так же, как в п. 3.1, величина I_1^{out} состоит из двух частей:

$\tilde{I}_1^{out}$ и $(I_1^{out} - \tilde{I}_1^{out})$, причём $\tilde{I}_1^{out} < (I_1^{out} - \tilde{I}_1^{out})$.

Здесь $\tilde{I}_1^{out}$ есть та часть избыточных полей I_1^{out}, которая увеличивает время доступа модулями к своим полям в наборе полей группы *F*. Заметим, как следует из способа упорядочивания общих внешних модулей, последнее поле опорного модуля не используется всеми внешними модулями.

Как указано выше, первые поля внешних модулей находятся среди полей расширения *d*, но их длины не превосходят длины *l* опорного модуля. Число этих неиспользуемых полей есть *w* (число внешних модулей)

После удаления последнего неиспользуемого модулями поля опорного модуля неравенство $\tilde{I}_1^{out} < (I_1^{out} - \tilde{I}_1^{out})$ переходит в равенство $\tilde{I}_1^{out} = (I_1^{out} - w - \tilde{I}_1^{out})$.

Отсюда окончательная верхняя оценка избыточных близких внешних полей есть

$$\tilde{I}_1^{out1} = \frac{1}{2}(\tilde{I}_1^{out} - w).$$

4. Заключение

Таким образом, построены формулы для $\tilde{I}_1^{in}$ и $\tilde{I}_1^{out}$, позволяющие легко вычислить верхние оценки внутрифайловой избыточности одной группы близких модулей. Эти же формулы могут быть использованы для вычисления величины избыточности и других групп близких модулей, составляющих базу данных.

Вычисленные величины позволяют более точно оценить затраты времени при запросе модулями информации из базы данных в реальном времени и оптимизировать структуру базы данных.

Литература

1. ДЕЙТ К.ДЖ. *Введение в системы баз данных*.– 8-е изд. – М.: Вильямс, 2006. – С. 1328.
2. ДЯТЧИНА Д.В. *Применение алгоритма оптимизации запросов на основе внесения контролируемой избыточности в базах данных* // Вести высших учебных заведений Черноземья. – 2012. – №4. – С. 48–51.
3. МИРОШНИК С.Н. *Алгоритм оптимизации структуры базы данных реального времени с минимальной избыточностью информации* // Некоторые алгоритмы составления расписаний в многопроцессорных системах. – М.: ВЦ РАН, 2015. – С. 25–34.
4. МИРОШНИК С.Н. *Алгоритмы построения базы данных с минимальной избыточностью информации для систем реального времени* // Труды межд. конф. по исследованию операций ORM-2016, 17-22 октября 2016, Москва. – М: ФИЦ ИУ РАН, 2016. – С. 51–52.
5. МИРОШНИК С.Н., ГОНЧАР Д.Р., ФУРУГЯН М.Г. *Оптимизация структуры базы данных реального времени* // Управление большими системами. – 2017. – Вып. 66. – С. 158–170.
6. МИРОШНИК С.Н. *Алгоритм оптимизации структуры базы данных с неограниченным числом файлов и минимальной избыточностью информации для систем реального времени*. // Некоторые алгоритмы планирования вычислений в многопроцессорных системах. – М.: ФИЦ ИУ РАН, 2017. – С. 29–38.
7. ТОНОЯН С.А., ЕЛИСЕЕВ Д.В., БАЛДИН А.В. *Избыточность темпоральных данных хранимых в реляционных СУБД* // Территория инноваций. – 2017. – №8(12). – С. 15–23.
8. ULMAN L. *PHP6 and MYSQL for dynamic web sites*. – Peachpit Press, 2007.

REAL TIME DATABASE STRUCTURE OPTIMIZATION

Sergey Miroshnik, Federal Research Centre «Informatics and Control» of RAS, Moscow, Cand. Sc., (rtsccas@ya.ru).
Dmitry Gonchar, Federal Research Centre «Informatics and Control» of RAS, Moscow, Moscow Institute of Physics and Technology, Moscow, Cand. Sc., (trpl@ya.ru).

Abstract: The problem of minimizing the redundancy of information in the real-time database, which affects the time of access to the database and the implementation of modules is studied. A set of software modules that use information from a set of fields with a known frequency of filling in real-time information is specified. The problem is solved if all modules are completed by the specified time. The complexity of this problem is caused by possible modules dependency (may work in a certain sequence) and the difficulty to allocate sufficient computing resources and time for real-time optimization. Therefore, the solution developed by the authors in the instrumental CAD real time systems is divided into two stages: (a) the preliminary stage (not in real time), which is the formation of groups of close modules and (b) the stage of solving the problem in real time, using the performed at the preliminary stage of optimization of the location of fields in files. The information redundancy of three types is defined: intra-file, cross-file, intra-module. A special model of the designed database is proposed and analytical formulas for calculating the number of fields not used by the modules are constructed.

Keywords: the database management system, real-time systems, heuristic algorithms, optimization.

УДК 519.86
ББК 22.18
DOI: 10.25728/ubs.2018.76.9

Статья представлена к публикации членом редакционной коллегии Э.Ю. Калимулиной.

Поступила в редакцию 29.01.2018.
Опубликована 30.11.2018.