УДК 004.652

Методы конвертации параметрических моделей модулей компонентов при проектировании радиотехнических систем

Кузнецов А.С.[1*], Кузнецов С.Н.[2], Постникова В.Н.[2***]**

[1]Государственный музей А.С. Пушкина,

ул. Пречистенка, 12/2, Москва, 101000, Россия

[2]Московский авиационный институт (национальный исследовательский университет), МАИ, Волоколамское шоссе, 4, Москва, А-80, ГСП-3, 125993, Россия

**e-mail: askuznetsov.gmp@gmail.com*

***e-mail: serghei-k@mail.ru*

****e-mail: apostnikov@mtu-net.ru*

Аннотация

Рассматриваются методы повышения производительности баз данных, реализованных на основе СУБД MS SQL Server 2012, при выполнение процедуры конвертации параметрических моделей модулей компонентов в системах проектирования радиотехнических систем (РТС). Приводится алгоритм процедуры конвертации моделей модулей, реализованный по методу с временными таблицами и дается программа данной процедуры, написанная на языке T-SQL. Кроме того, в статье проводится сравнительный анализ различных методов реализации функции выполнения иерархического запроса, которая входит в процедуру конвертации и является наиболее ресурсоемкой. Показано, что при реализации процедуры конвертации, при большом числе модулей, входящих в иерархическое дерево (более

1000), целесообразно использовать структуру БД с временными таблицами, а при выполнение функции иерархического запроса - метод, в котором применяется тип данных HierarchyID.

Ключевые слова: системы автоматизированного проектирования, база данных, конвертация модулей компонентов, иерархические запросы, оптимизация запросов, проектирование радиотехнических систем.

Введение

При разработке радиотехнических систем для авиационной и космической технике все шире используется системное проектирование, основанное на модульном принципе построения аппаратуры. Такой подход позволяет значительно сократить время и стоимость разработки РТС за счет использования модулей различной производственной готовности [1].

Основой информационного обеспечения таких систем проектирования являются базы данных (БД), с помощью которых осуществляется хранение и выборка как параметрических моделей модулей РТС, так и параметрических моделей целых проектов РТС, которые представляют собой совокупность модулей различной производственной готовности и различного иерархического уровня. При этом в процессе проектирования РТС, возникает необходимость осуществлять конвертацию параметрических моделей модулей и их характеристик из формата хранения модуля в формат хранения проекта. Отличие форматов заключается в том, что в таблице параметрических моделей модулей для идентификации каждого

модуля используется уникальный ключ, а в таблице параметрических моделей проектов для идентификации модулей применяется составной ключ, который позволяет привязать параметрическую модель модуля к формату параметрической модели проекта PTC. Процедура конвертации заключается в генерации составных ключей для каждого модуля, входящего в параметрическую модель проекта PTC.

Постановка задачи

Время выполнения процедуры конвертации модулей во многом зависит как от структуры самой БД, так и от алгоритмов выполнения отдельных функций, входящих в данную процедуру. Уменьшить время выполнения процедуры конвертации и тем самым повысить производительность всей САПР можно путем использования возможностей современных СУБД.

В предлагаемой статье рассматриваются различные варианты построения БД, реализованной на основе СУБД MS SQL Server 2012, позволяющие решать указанные выше задачи, а также дается оценка их производительности.

Методы конвертации параметрических моделей модулей компонентов.

Возможности современных СУБД позволяют реализовать структуру БД таким образом, что все необходимые операции, связанные с процедурой конвертации проводятся не со всей базой данных, хранящейся на RAID - массиве сервера, а только с ограниченным объемом данных, которые размещаются в оперативной памяти сервера. Такая структура БД позволяет значительно сократить время

выполнения процедуры конвертации. При этом возможно использовать один из двух методов, поддерживаемых СУБД MS SQL Server 2012 [2,3]:

- Реализация БД с использованием курсора.
- Реализация БД с применением временных таблиц.

В первом случае используется так называемый курсор, который представляет собой область в оперативной памяти сервера БД, в которой сохраняются последовательно выбранные модули компонентов. После этого осуществляется построчная процедура конвертации, которая заключается в присвоение каждому модулю уникального составного ключа.

Во втором случае, при запуске процедуры конвертации формируется временная таблица, которая аналогична таблице, где хранятся параметрические модели модулей, но с дополнительными двумя столбцами, в которые записываются уникальные составные ключи модулей. При выборе требуемого модуля, его параметры сохраняются во временной таблице и одновременно этому модулю присваивается уникальный составной ключ, который соответствует формату текущего проекта. После того, как в таблицу записано полное иерархическое дерево всех выбранных модулей компонентов, происходит их конвертация, которая заключается в генерации составного ключа и модификации иерархических связей. После выбора и конвертации всех требуемых модулей происходит запись данных из временной таблицы в модель проекта.

В принципе и в первом и во втором случаях уменьшение времени выполнения процедуры конвертации происходит за счет того, что все необходимые операции

проводятся только с ограниченным (выбранным) числом модулей, записанных либо во временную таблицу, либо в курсор. Кроме того все операции, связанные с добавлением, удалением и конвертацией модулей осуществляется в оперативной памяти сервера, что также повышает производительность БД.

Однако, в отличие от варианта с временными таблицами, при использование курсора процесс конвертации проводится последовательно (построчно), что увеличивает время конвертации, особенно при большом числе модулей и сложной, многоуровневой иерархической структуре модели проекта. Поэтому, в таких случаях, с точки зрения производительности, целесообразно использовать структуру БД с временными таблицами. Алгоритм процедуры конвертации, реализованный по методу с временными таблицами приведен на рис. 1

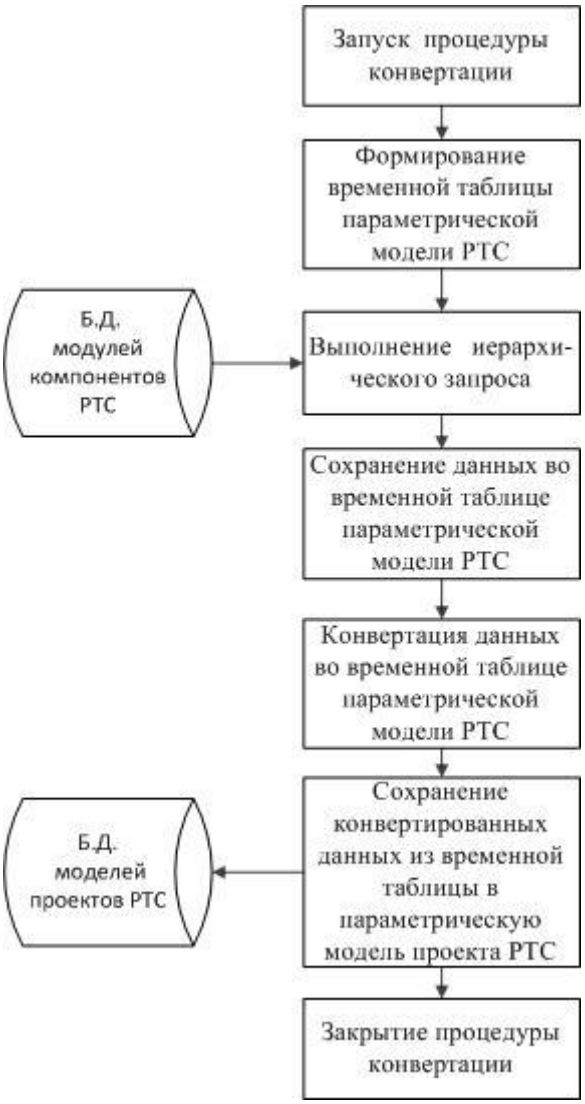

Рис.1. Алгоритм процедуры конвертации, реализованной по методу с временными таблицами.

Данный алгоритм включает в себя последовательность выполнения следующих функций:

- Запуск процедуры конвертации. Осуществляется с помощью команды:

  EXEC dbo.procDIC_COMPO_IMP_TO_MODEL_N;

- Формирование временной таблицы параметрической модели проекта PTC. Временная таблица используется для хранения всех выбранных компонентов в

формате проекта. Структура временной таблицы повторяет структуру таблицы БД, в которой размещаются данные о параметрических моделях модулей. Однако, во временной таблице столбец [CMP_ID], в котором хранится уникальный ключ компонента, заменяется на столбец [DC_ID]. Также добавляется еще один столбец [DC_FN_ID]. Два этих столбца используются для хранения составного ключа, который привязывает модель модуля к формату модели проекта PTC. Формирование ключа осуществляется с помощью функции «Sequence».

- Выполнение иерархического запроса. Данная функция позволяет осуществить выбор требуемых модулей компонентов из всего иерархического дерева и запись их во временную таблицу.
- Функция конвертация компонентов во временной таблице сводится к генерации для каждого компонента составного ключа, который распределяется по двум столбцам [DC_ID] и [DC_FN_ID] , а также обновление массива ссылок на дочерние элементы, т.е. модификации иерархических связей в столбце [CMP_PID_ID].
- Сохранение нового варианта модели проекта в БД При этом происходит перезапись данных из временной таблицы TempDB, находящийся в оперативной памяти сервера СУБД SQL Server в БД моделей проектов, размещенной на RAID - массиве сервера.
- Закрытие процедуры конвертации сводится к удаление временной таблицы и очистке системной базы данных TEMPDB .

Программа процедуры конвертации модулей, написанная на языке T-SQL, приведена ниже.

```
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[procDIC_COMPO_IMP_TO_MODEL_N]
 @DIC_COMPO_ID int ,
 @MDLCOMPO_ID int
AS
BEGIN
 DECLARE @MODEL_ID int
Set @MODEL_ID =(NEXT VALUE FOR seqMODEL_generateID)
IF OBJECT_ID('[#MODEL_COMPO]','U') IS NOT NULL
DROP TABLE [#MODEL_COMPO]
CREATE TABLE [dbo].[#MODEL_COMPO] (
 [DC_ID] int NOT NULL,
 [DC_NAME] nvarchar(max) NULL,
 [DIC_PID_ID] int NOT NULL,
 [[DC_FN_ID] int NOT NULL,
 [DC_FN_PID_ID] int NOT NULL,
 [P_COMPANY] int NULL,
 [DC_TYPE] int NOT NULL,
        DC_TEMPLATE int NOT NULL ,
         [MODEL_MDL_ID] int NULL
         CONSTRAINT [PK_MODEL_COMPO_TMP] PRIMARY KEY CLUSTERED
(
 [DC_ID] ASC
```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

Методы формирования иерархических запросов

Из всех функций, которые входят в процедуру конвертации наиболее ресурсоемкой является функция выполнения иерархического запроса. Это объясняется тем, что при иерархическом запросе происходит прохождение полного дерева иерархии и выборка всех модулей, входящих в данное иерархическое дерево. Очевидно, что чем выше число модулей компонентов входит в иерархическое дерево и чем больше количество уровней иерархии, тем больше времени требуется для выполнения иерархического запроса, т.е. можно записать:

$$\overline{T}\text{из} = f(N, K) \qquad (1)$$

Где: $\overline{T}\text{из}$ – среднее время выполнения иерархического запроса; N- число модулей компонентов, входящих в иерархическое дерево; K – количество уровней иерархии. Среднее время используется для исключения факторов, связанных с загрузкой сервера другими программами в момент выполнения запроса.

В базах данных, построенных на основе современных СУБД MS SQL Server для формирования иерархического запроса нашли применение в основном два метода – метод типа «parent-child» и метод, в котором используется тип данных HierarchyID [4, 5].

В первом случае для выборки данных применяются CTE-выражения (common table expression), которые выполняют рекурсивные вычисления, в том числе и по

древовидным структурам. Для выборки данных по второму методу используют функции расширения, которые были введены в СУБД, начиная с MS SQL Server 2008., например, функция GetLevel, которая позволяет формировать уровень HierarchyID и соответствующий дополнительный столбец в таблице. Тип данных HierarchyID — это системный тип данных, размер которого может меняться в зависимости от структуры дерева (его глубины) и среднего числа потомков.

Основным преимуществом метода HierarchyID, по сравнению с методом Parent/Child, является возможность избавиться от циклов и рекурсивных запросов CTE , выполнение которых требует больших временных затрат.

Как правило, модули, входящие в радиотехническую систему, распределяются по четырем иерархическим уровням - система, комплексы, устройства и функциональные узлы [6]. Для такого варианта структуры РТС иерархический запрос, реализованный по методу Parent/Child и написанный на языке T-SQL, выглядит следующим образом:

```
With ParamCTE(CMP_ID, CMP_NAME,CMP_PID_ID,level) AS
( SELECT CMP_ID, CMP_NAME, CMP_PID_ID,0 As level
FROM COMPO WHERE CMP_ID= @cmp_par
UNION ALL SELECT E.CMP_ID, E.CMP_NAME, E.CMP_PID_ID, level +1
FROM COMPO AS E JOIN paramCTE AS M ON E.CMP_PID_ID=M.CMP_ID)
SELECT CMP_ID, CMP_NAME,CMP_PID_ID,level from paramCTE
```

Таблица с результатами запроса, реализованного по данному методу, представлена на рис.2.

	CMP_ID	CMP_NAME	CMP_PID_ID	CMP_COMPANY_ID	CMP_TYPE	PROTOPRT_ID
1	1	Система	NULL	6170	1	1216
2	1000	Комплекс_К_1	1	6170	2	1216
3	1100	Комплекс_К_2	1	6170	2	1216
4	1101	Устройство_У1	1000	6170	3	1216
5	1102	Устройство_У2	1100	6170	3	1216
6	1103	Функциональный узел_Ф1	1101	6170	4	1216
7	1104	Функциональный узел_Ф3	1102	6170	4	1216
8	1105	Функциональный узел_Ф2	1101	6170	4	1216
9	1106	Функциональный узел_Ф4	1107	6170	4	1216
10	1107	Устройство_У3	1000	6170	3	1216
11	1108	Функциональный узел_Ф5	1107	6170	4	1216

Рис.2 . Результат запроса реализованного по методу Parent/Child

В указанную таблицу входят следующие атрибуты:

CMP_ID - первичный ключ каждого модуля.

CMP_NAME - название модуля.

CMP_PID_ID - внешний ключ, который соответствует первичному ключу предка для данного модуля.

CMP_TYPE – иерархический уровень, к которому принадлежит данный модуль

PROTOPRT_ID- внешний ключ, который соответствует номеру проекта, к которому принадлежат модули, входящие в таблицу.

При такой структуре построения БД потомок определяет своего предка по записи в столбце CMP_PID_ID.

Иерархический запрос для того же самого варианта построения РТС, сформированный по методу типа данных HierarchyID, записывается как:

*select hid.ToString(), hid.GetLevel(), * from COMPO_OS where level <4*

Таблица с результатами запроса, реализованного по методу типа данных HierarchyID, представлена на рис.3.

	EMPL_STRING	CMP_ID	HID	CMP_NAME	CMP_TYPE	PROTOPRT_ID	level
1	/	1	0x	Система	1	1	0
2	/1/	2	0x58	Комплекс_К1	2	1	1
3	/2/	614677	0x68	Комплекс_К2	2	1	1
4	/1/1/	653361	0x5AC0	Устройство_У1	3	1	2
5	/2/1/	653362	0x6AC0	Устройство_У3	3	1	2
6	/1/1/1/	653363	0x5AD6	Функциональный узел_Ф1	4	1	3
7	/2/1/1/	653364	0x6AD6	Функциональный узел_Ф4	4	1	3
8	/1/1/2/	653366	0x5ADA	Функциональный узел_Ф2	4	1	3
9	/2/1/2/	653370	0x6ADA	Функциональный узел_Ф5	4	1	3
10	/1/2/	653371	0x5B40	Устройство_У2	3	1	2
11	/1/2/1/	653372	0x5B56	Функциональный узел_Ф3	4	1	3

Рис.3. Результат запроса, реализованного по методу типа данных HierarchyID.

В таблице использованы следующие атрибуты:

EMPL_STRING – положение модуля в иерархии в текстовом формате.

CMP_ID - первичный ключ модуля.

HID - иерархический тип значений HierarchyID.

CMP_NAME - название модуля.

CMP_TYPE – иерархический уровень, к которому принадлежит данный модуль.

PROTOPRT_ID- внешний ключ, который соответствует номеру проекта, к которому принадлежат модули, входящие в таблицу.

LEVEL - логический столбец, уровень вложения модуля.

Значение типа данных HierarchyID описывает положение модуля в иерархической структуре РТС. Другими словами, значение типа данных HierarchyID содержит список всех модулей, которые нужно пройти от корня иерархии до заданного модуля. В текстовом формате, который использован в столбце «EMPL_STRING», уровень иерархии разделяется символом "/". Номер каждого отдельного модуля представляет собой набор числовых значений, разделенных точкой (в нашем случае это одно число). В цифровом формате значения типа данных HierarchyID хранятся в столбце «HID».

Такой вариант построения БД позволяет значительно сократить время ответа на иерархический запрос. Это объясняется тем, что в столбце HID прописывается путь не только к предку модуля, как в столбце CMP_PID_ID, но и к потомку данного модуля. В связи с этим отсутствует необходимости проходить циклически всю иерархию.

На рис. 4 приведены зависимости среднестатистического времени выполнения иерархического запроса от общего числа модулей (N), входящих в иерархическое дерево, для двух, рассмотренных выше, вариантов построения базы данных: $\overline{T}^1$из – база данных, реализованная по методу Parent/Child, $\overline{T}^2$из – база данных, реализованная по методу, в котором используется тип данных HierarchyID.

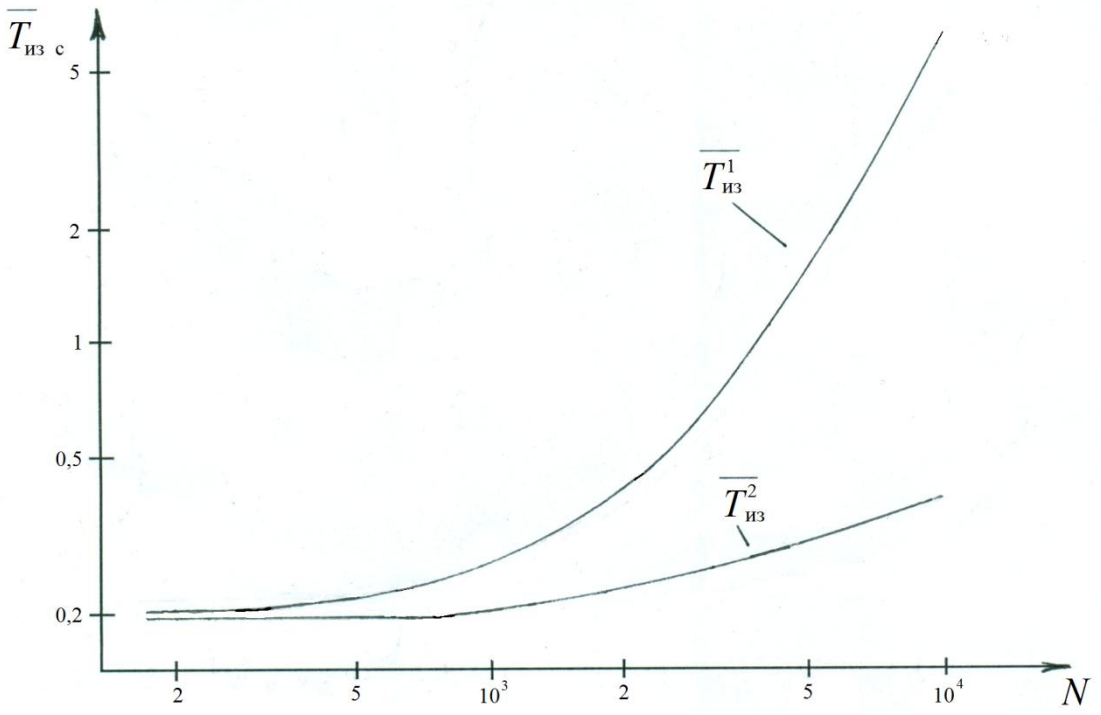

Рис.4. Зависимости среднестатистического времени выполнения иерархического запроса от общего числа модулей, входящих в иерархическое дерево.

Для исключения влияния аппаратных средств на результат сравнительного анализа различных вариантов построения БД все измерения проводились на одном и том же сервере, имеющем следующие параметры: процессор - Intel Xeon E5620 2.4GHz, ОЗУ - 8Gb, ОС windows 2008 R2, 64 разрядная.

Как видно из графиков, при построении БД по методу Parent/Child, $\overline{T}$из изменяется от 0,2 сек. (при N=100) до 6 сек. (при N=10000). В тоже время, при использование метода, в котором применяется тип данных HierarchyID,

среднестатистическое время выполнения запроса ($\overline{Тиз}$) с увеличением N растет незначительно и изменяется от 0,2 сек. (при N=100) до 0,4 сек. (при N=10000)., т.е. при N=10000 время выполнения запроса будет в 15 раз меньше, чем в первом случае. При дальнейшем увеличении N разница во времени выполнения запроса возрастает еще сильнее. Однако, следует заметить, что при небольшом числе компонентов, входящих в иерархическое дерево (в данном случае $N < 1000$), разница во времени выполнения запроса между первым и вторым вариантами построения БД практически равна нулю.

Выводы

Таким образом, на основание проведенных исследований можно сделать вывод, что при реализации процедуры конвертации, при больших значениях N ($N > 1000$), с целью повышения производительности информационного обеспечения САПР целесообразно использовать структуру БД с временными таблицами, а при выполнение функции иерархического запроса - метод, в котором применяется тип данных HierarchyID.

В тоже время, при небольшом числе модулей ($N < 1000$), как при реализации процедуры конвертации, так и при выполнение иерархических запросов, производительность БД не существенно зависит от ее структуры. В этом случае имеет смысл применять БД с более простой структурной реализацией, например, с использованием курсора, а для выполнения иерархических запросов – метод Parent/Child.

Библиографический список

1.Ушкар М.Н Автоматизация системного проектирования информационных радиосистем // Электронный журнал «Труды МАИ», 2014, выпуск № 78: http://www.mai.ru/science/trudy/published.php?ID=53558 (дата публикации 02 декабря 2014).

2.Использование курсоров и циклов в Transact-SQL: http://info-comp.ru/obucheniest/352-cursors-loops-in-transact-sql.html (дата публикации 23.04.2014).

3.Фернандо Герреро. Применение псевдовременных таблиц. Журнал «SQL Magazine OnLine», 2001, № 3: http://www.osp.ru/data/www2/win2000/sql/2001/03/730.htm.

4.Ян Либерман, ORDPATH – новый подход к работе с иерархиями (деревьями) в SQL Server 2008, RSDN Magazine, 2007, №4: http://rsdn.ru/article/db/ordpath.xml (дата обращения: 12.03.2015).

5.Kent Tegels, Model Your Data Hierarchies With SQL Server 2008, MSDN Magazine. Issues and Downloads. 2008, URL: https://msdn.microsoft.com/en-us/magazine/cc794278.aspx (дата обращения: 18.02.2015).

6.Уровни разукрупнения радиоэлектронных средств. ГОСТ Р 52003-2003. – М.: Изд-во стандартов, 2003.