

УДК 519.6

## ПРИКЛАДНАЯ СЕРВИСНАЯ СИСТЕМА ДЛЯ ХРАНЕНИЯ И ОБРАБОТКИ РЕЗУЛЬТАТОВ СЧЕТА ЗАДАЧ КОМПЛЕКСОВ ПРОГРАММ МАТЕМАТИЧЕСКОЙ ФИЗИКИ

С. В. Гагарин, Л. М. Зуева, В. В. Комоско, С. А. Маркеленков,  
С. В. Морозов, П. А. Новгородцева, Н. Н. Соколова, А. В. Степанов  
(РФЯЦ-ВНИИТФ)

Рассматривается прикладная сервисная система СОВТ, предназначенная для накопления, обработки и представления результатов расчетов задач математической физики. СОВТ состоит из автономных программных компонентов, которые объединяются через взаимодействие с базой данных стандартных временных таблиц, управляемых СУБД Oracle. Взаимодействие с базой данных стандартных временных таблиц может осуществляться автономно, в составе комплексов программ или в составе специализированных систем обработки и представления информации. СОВТ осуществляет выборку информации из базы данных стандартных временных таблиц сразу о нескольких функциях разных задач, сосчитанных по различным комплексам программ, а также их предварительную обработку и передачу для дальнейшей обработки и анализа в широко используемые системы обработки и представления информации (Matlab, Excel) или в специализированную систему научной визуализации VIZI1D, созданную в РФЯЦ-ВНИИТФ.

Прикладная сервисная система для хранения и обработки результатов расчетов в виде стандартных временных таблиц (СВТ) — СОВТ (система обработки временных таблиц), созданная в РФЯЦ-ВНИИТФ, включает в себя следующие компоненты:

- записи результатов счета в базу данных (БД) СВТ (в автономном режиме и в процессе основного счета);
- выборки результатов счета накопления из БД;
- обработки результатов расчетов;
- графического представления результатов расчетов.

СОВТ ориентируется на СУБД Oracle [1] и в процессе работы использует БД СВТ.

Система СОВТ не является единым программным образованием. Она представляет собой набор автономных программных компонентов, которые объединяются через взаимодействие с БД СВТ. Взаимодействие с БД СВТ может осуществляться автономно, в составе комплексов программ или в составе специализированных систем обработки и представления информации. В

самой СОВТ можно выделить следующие функциональные компоненты:

- конверторы файлов накоплений комплексов программ в БД СВТ;
- библиотеки процедур доступа к БД СВТ из прикладных программ в ОС Linux и Windows;
- сервер доступа к БД СВТ.

**Конверторы.** Комплексы программ в процессе счета с помощью специализированных систем накапливают информацию в файлах специальной структуры. Программа-конвертор выполняет функцию преобразования информации из структуры файла в структуру БД СВТ. Конвертор осуществляют запись в БД СВТ с помощью библиотеки функций, основанных на базовых функциях СУБД Oracle. Загрузка информации в БД СВТ происходит в режиме пост-процессора (автономный режим) после окончания счета задачи. В настоящее время созданы конверторы практически для всех основных двумерных комплексов программ математического моделирования, имеющихся во ВНИИТФ.

**Библиотеки процедур доступа к БД СВТ.** Библиотеки процедур доступа предоставляют прикладным программам необходимый набор операций над БД СВТ под ОС Linux и Windows. Они используются как конверторами в автономном режиме, так и прикладными комплексами программ непосредственно в процессе основного счета.

Автономный режим записи в БД СВТ состоит из нескольких последовательных шагов, требующих обязательного вмешательства человека.

Организация записи в БД СВТ комплексами программ непосредственно в процессе основного счета задачи имеет ряд неоспоримых преимуществ. Во-первых, процесс записи информации в БД СВТ становится полностью автоматическим, благодаря чему существенно повышается оперативность поступления информации в БД и минимизируется количество ошибок. Во-вторых, исчезает множество старых систем накопления результатов счета, одни из которых уже давно никем не сопровождаются, а другие имеют меньшую в сравнении с СВТ функциональность. В-третьих, физик или расчетчик может в случае необходимости видеть результаты не только сразу по завершении расчета, но даже в его процессе.

**Сервер доступа к БД СВТ.** С помощью сервера доступа осуществляется выборка объектов накопления из БД СВТ и их предварительная обработка. Эта обработка заключается в переводе информации из вида, в котором она хранится в БД, в вид математических структур данных (массив, матрица, многомерная матрица). Математические структуры данных передаются для дальнейшей обработки и анализа в стандартные системы (Excel [2], Matlab [3]) или в специализированную систему визуализации VIZI1D [4], созданную во ВНИИТФ. Сервер может выбрать из БД информацию сразу о нескольких функциях разных задач, сосчитанных по различным комплексам программ.

## 1. Архитектура системы СВТ

Функциональная схема СВТ приведена на рис. 1 (внутри затененного квадрата находятся компоненты, непосредственно входящие в состав системы). Система СВТ имеет многоуровневую архитектуру (уровни приводятся в порядке расположения снизу вверх):

- операторы языка манипулирования данными СУБД;
- программы доступа к записям таблиц БД СВТ;
- программы доступа к объектам накопления в целом;
- программы обработки и оформления результатов счета (непосредственно в состав системы не входят).

На уровне операторов языка манипулирования данными СУБД Oracle выполняются операции над записями таблиц БД СВТ с помощью языка SQL.

На уровне программ доступа к записям таблиц БД СВТ используется набор функций, осуществляющих операции выборки, включения, удаления и обновления записей. Эти функции реализуются под ОС Windows для каждой таблицы БД СВТ на основе объектно-ориентированного метода доступа к БД Oracle Objects for Ole, а под ОС Linux — средствами препроцессора Pro\*C. В свою очередь, они базируются на операторах языка манипулирования данными SQL. За одну операцию каждая функция обрабатывает одну запись и передает результат через оперативную память прикладной программы на уровень программ объектов доступа к объектам накопления. На уровне программ доступа к объектам накопления (сервера доступа к БД СВТ) осуществляется формирование структур объектов накопления из отдельных записей таблиц БД СВТ. Каждый объект накопления является многомерным массивом, одним измерением которого обязательно является время. Объекты накопления могут иметь структуру:

- трехмерной матрицы, содержащей координаты узлов фрагментов разностной сетки счетных областей и определенных в них значений конкретной счетной величины на конкретный момент времени;
- двумерной матрицы, содержащей координаты узлов границы счетной области и определенных в них значений конкретной счетной величины на конкретный момент времени;
- вектора, содержащего значения интегральных и балансных величин, характеризующих состояние конкретной счетной области или системы в целом на различные моменты времени.

Формирование объектов накопления осуществляется по запросу с уровня обработки и

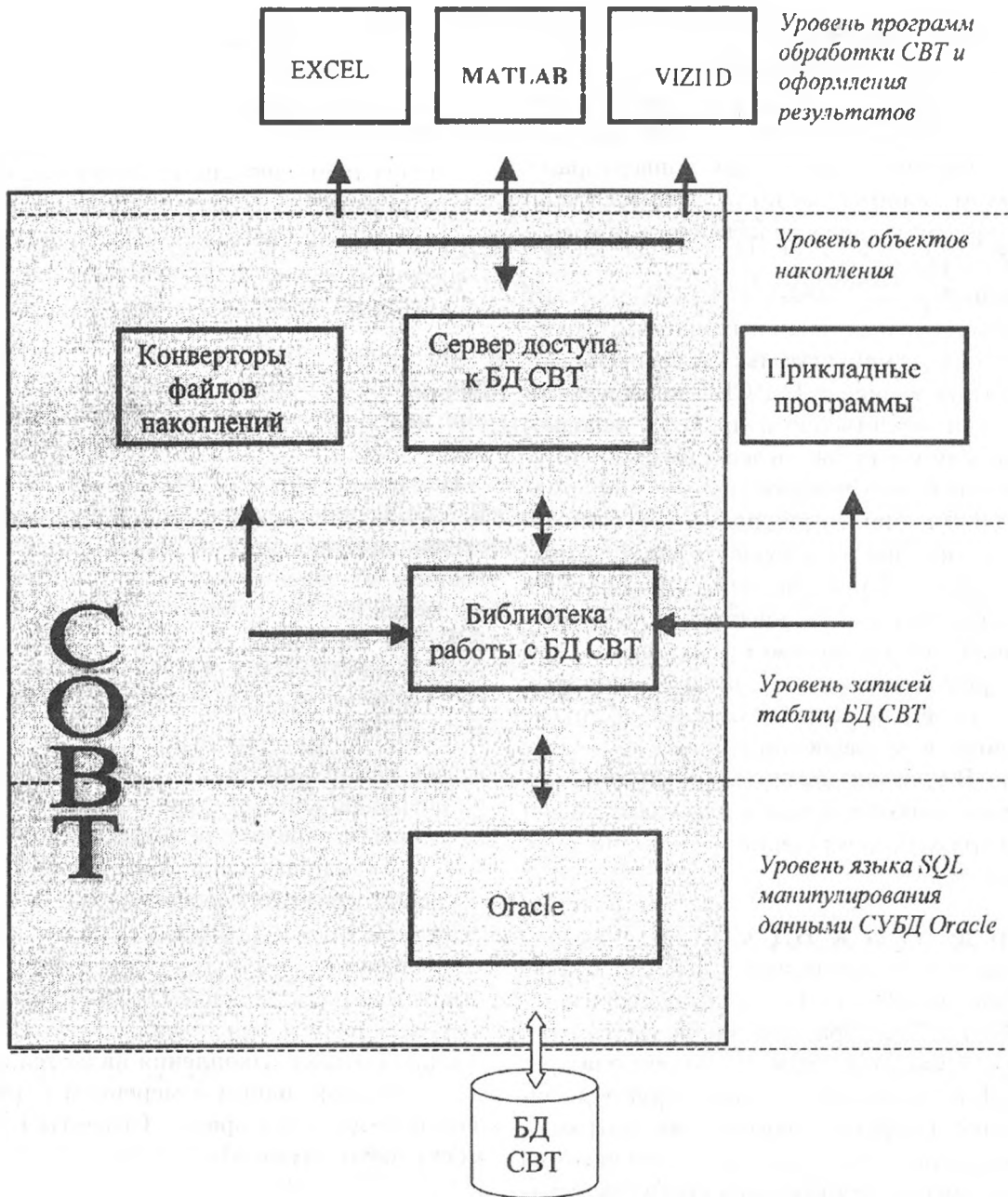


Рис. 1. Функциональная схема системы СОВТ

оформления результатов счета путем определенной последовательности шагов в процессе диалога.

На уровне обработки и оформления результатов счета осуществляется необходимая обработка объектов, выбранных из БД СВТ и подготовленных сервером доступа, их группировка и отображение на экран компьютера или в твердую копию. Это различные программы пересчета на новые сетки, прореживания, выделения фронтов, численного дифференцирования и других

аналогичных числовых обработок. Отличительной особенностью этого класса программ является то, что они берут исходные данные из БД СВТ и помещают результат, как правило, туда же. На этом уровне используются стандартные программные системы Matlab, Origin, Excel или специализированная система VIZI1D. Запрос к БД СВТ для получения объекта накопления может быть сформулирован с помощью специализированного языка (в случае Matlab) или введен в форме диалога (в случае Excel, VIZI1D). Полу-

чив необходимую информацию, можно осуществить ее отображение средствами этих прикладных систем.

## 2. Структура БД СВТ

БД СВТ содержит результаты счета задач математической физики в виде СВТ. Временная таблица — сеточная функция  $f(t)$  или группа однородных функций  $f(t; x_1, x_2, \dots, x_n) \equiv f(t, X)$ , одним из аргументов которой(ых) является время. Под однородными понимаются функции, имеющие одинаковый физический смысл, размерность (нулевую, одно-, двух- или трехмерную), размеры (масштаб величины) и привязку ко времени, но различную пространственную (или иную, но не временную) привязку. На каждый момент времени такая функция может быть представлена матрицей любой размерности (от 0), но ее размеры на другие моменты времени должны быть такими же. Примеры: температура  $T(t)$  в точке — скалярная (нульмерная) величина; давление  $P(t, r)$  на верхней

границе блока — матрица размерности 1 (аргумент  $r$  — координата вдоль границы); энергия в областях блока  $\varepsilon(t, r, \theta)$  — матрица размерности 2 (аргументы — по радиусу и углу);  $N(t, r, \theta, G)$  — числа частиц — матрица размерности 3 (аргументы — по радиусу, углу и группе, можно еще и по направлению) и т. п.

Временные таблицы возникают как результат работы комплексов программ (накопления, обработки) и используются в качестве граничных, источниковых и других условий при проведении расчетов в других или тех же самых комплексах. Преобразование временных таблиц комплексов программ в стандартный вид (см. подразд. 3.2) позволяет хранить их в БД под управление СУБД Oracle в виде СВТ и построить на основе БД СВТ единые для всех комплексов программ системы обработки, визуализации и обмена данными. Общая логическая структура временных таблиц для одной задачи приведена на рис. 2. Каждый комплекс программ хранит (пишет) свои временные таблицы в виде заголовка, каталога и тела.

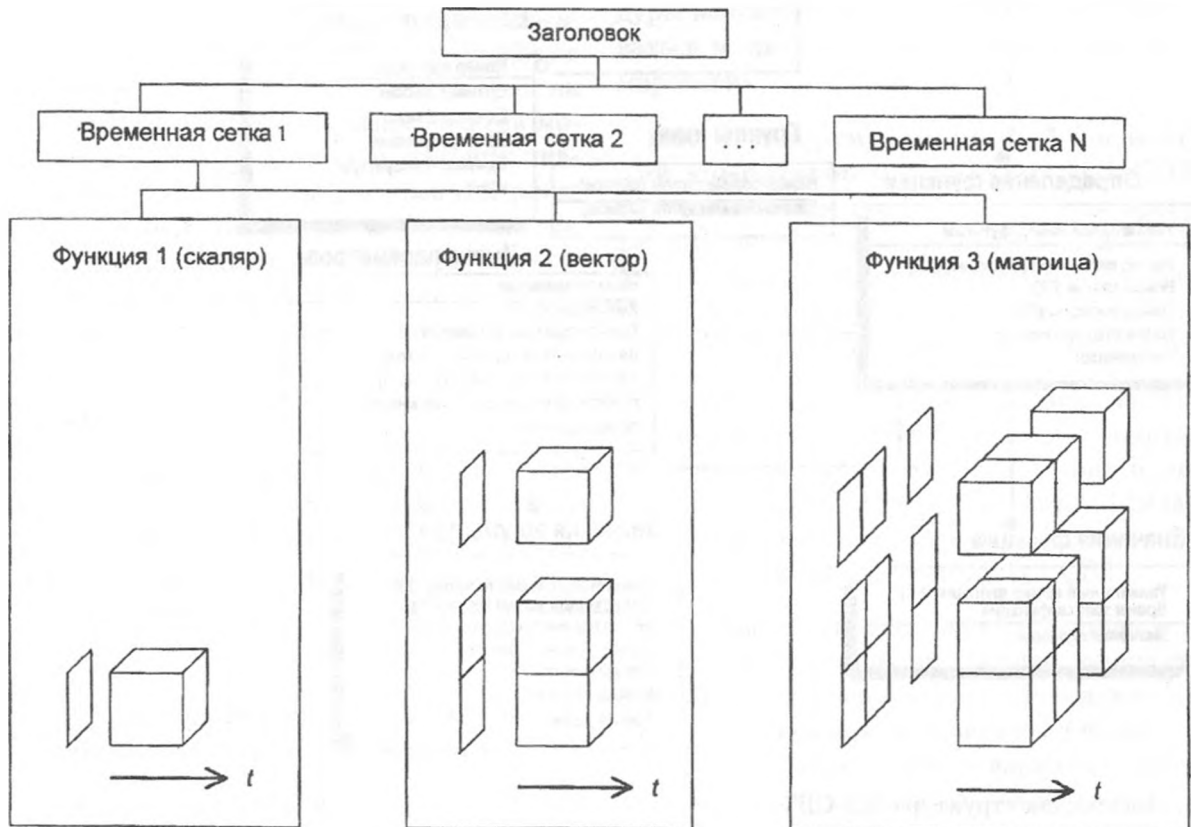


Рис. 2. Общая логическая структура временных таблиц

Представленная на рис. 2 иерархическая схема временных таблиц может быть преобразована в

реляционную логическую структуру БД СВТ и представлена в виде ER-диаграммы [5] (рис. 3)\*.

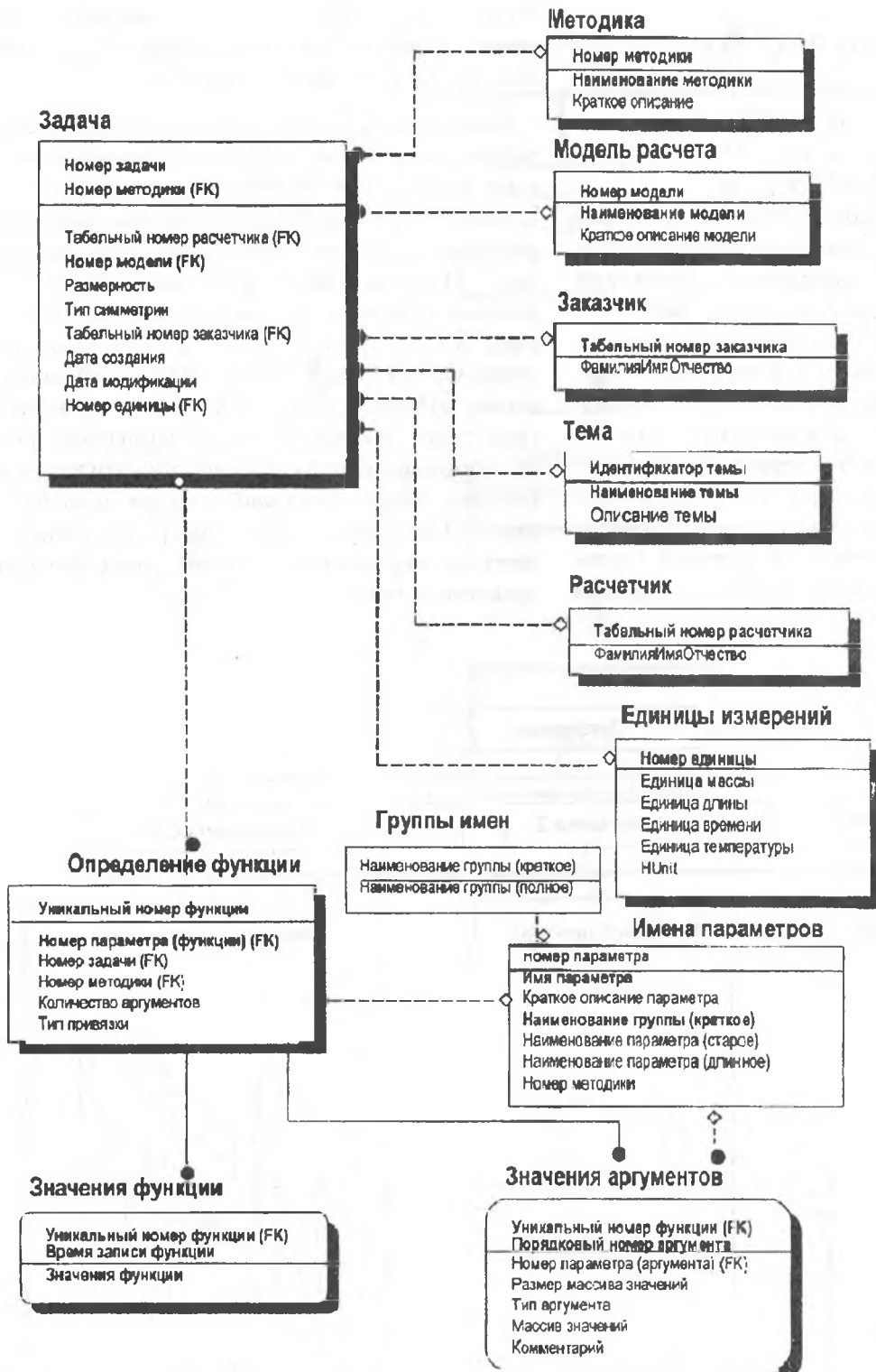


Рис. 3. Логическая структура БД СВТ

\*FK на рисунке обозначает внешний ключ, т. е. поле данной таблицы может принимать значения только из множества, находящегося в соответствующей таблице-справочнике, где это поле является первичным ключом.

Главным объектом логической структуры БД СВТ является объект *Задача*, который содержит определенный набор атрибутов: номер задачи, номер методики, номер модели расчета, индекс темы и т. д. Расшифровка этих номеров также содержится в соответствующих справочниках: *Методики*, *Модели расчета*, *Темы*, *Расчетчики*, *Заказчики*, *Единицы измерения*, *Имена параметров*. В справочнике *Имена параметров* каждому имени функции или аргумента ставится в соответствие номер. Этот номер уникален для каждой методики.

Объекту *Задача* подчиняется объект *Определение функции*, который содержит информацию об имени функции, типе ее привязки к ячейкам разностной сетки. Этот объект однозначно идентифицируется уникальным номером функции, который необходим для минимизации длины ключа в объектах, подчиненных *Определению функции*. Если бы не было уникального номера функции, то вместо него (4 байта) в качестве сцепленного ключа пришлось бы использовать три атрибута: номер задачи (4 байта), номер методики (4 байта) и порядковый номер функции (4 байта).

Объекту *Определение функции* подчиняются объекты *Значения функции* и *Значения аргументов*. Объект *Значения функции* содержит момент времени и массив значений функции на этот момент. Как отмечалось выше, массив может содержать скаляр, вектор и матрицу значений. Объект *Значения аргументов* содержит порядковый номер аргумента, номер имени параметра, размер массива значений аргумента и сам массив значений аргумента. Массив значений аргумента может содержать также скаляр, вектор или матрицу значений.

Для погружения в СУБД Oracle логическая структура БД СВТ преобразуется в физическую. Суть преобразования заключается в выборе СУБД-ориентированных имен и форматов полей реляционных таблиц БД, определении обязательности нахождения в конкретном экземпляре записи информации и т. п.

### 3. Взаимодействие комплексов программ с системой СОВТ

**3.1. Библиотека функций доступа к БД СВТ.** Система СОВТ взаимодействует с другими прикладными программами с помощью библиотеки специальных процедур, используемых

в прикладной программе как автономно, так и в процессе основного счета. Основными являются процедуры записи и чтения информации БД СВТ, а именно:

- 1) занесения данных:
  - в таблицу значений временных функций;
  - в таблицу значений аргументов временных функций;
  - в таблицу определения временных функций;
  - в таблицу-справочник имен функций и аргументов;
- 2) чтения данных:
  - из справочника имен функций и аргументов;
  - из таблицы определения временных функций. Соответствующая процедура извлекает все уникальные номера функций, полученные в процессе счета данного варианта.

Кроме процедур записи и чтения, имеются процедуры открытия и регистрации в БД СВТ и средства синхронизации-отката записей, позволяющие пользователю самостоятельно определять требуемые единицы работы с БД. Процедуры написаны на языке С [6] и могут использоваться в программах на любом языке программирования.

**3.2. Конвертирование файлов накопленных комплексов в структуру БД СВТ после окончания счета.** Комплексы программ, расчеты по которым проводятся под ОС Linux, имеют различные структуры файлов накопленных. Конвертирование файлов накопленных из ОС Linux осуществляется в два этапа (рис. 4):

- конвертирование под ОС Linux из формата файлов накопления комплекса программ в структуру текстового файла фиксированного формата (общего текстового файла);
- конвертирование структуры общего текстового файла в структуру БД СВТ.

В основе структуры общего текстового файла лежит понятие функции, точнее, конкретной ее выборки, имеющей определенную длину и определенные значения аргументов. Общий текстовый файл содержит номер варианта, номер блока (для некоторых комплексов программ), количество моментов времени, в которые производилось накопление, и массив значений времен. Далее для каждой функции следует имя функции,

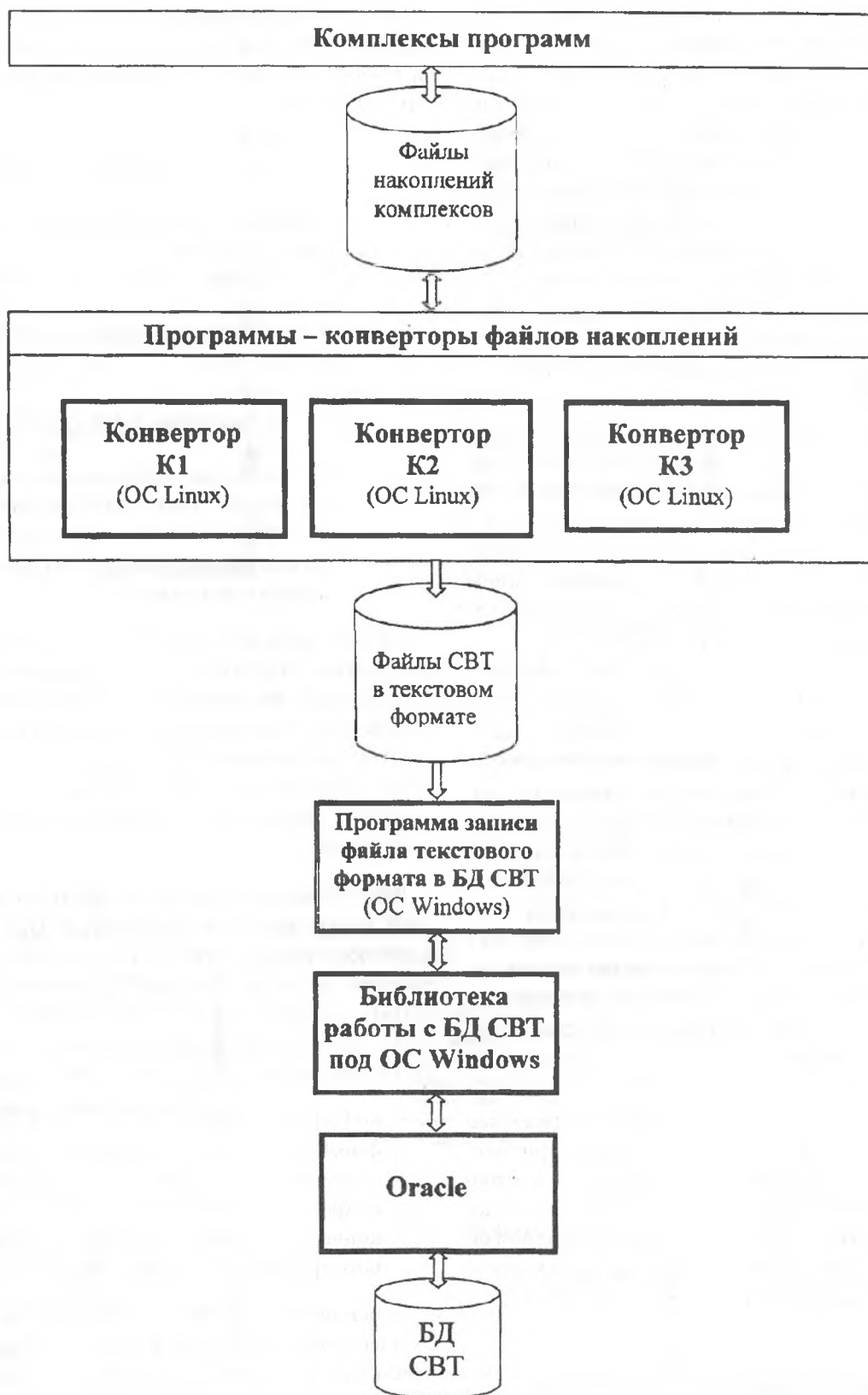


Рис. 4. Схема взаимодействия комплексов программ с системой COBT после окончания основного счета

количество аргументов, описание аргументов и массив значений функции для каждого момента времени.

На первом этапе программы-конверторы под ОС Linux читают информацию с использованием своих специальных средств и разбирают ее в структуру общего текстового файла. Результат записывается в один или несколько (например, по количеству блоков) файлов, которые затем переносятся из ОС Linux на персональный компьютер в ОС Windows.

На втором этапе в ОС Windows осуществляется конвертирование текстового файла в структуру БД СВТ. Конкретная выборка функции заносится в виде записи БД СВТ, получая предварительно уникальный номер в БД (идентификатор функции в БД СВТ). Получение уникального номера функции осуществляется с помощью механизма последовательности СУБД Oracle.

Поскольку текстовые файлы, содержащие данные накопления разных комплексов, имеют сходную структуру, то и программы их конвертирования в формат таблиц БД СВТ имеют много общего. В основе их работы лежит чтение и разбор текстовых файлов приведенной выше структуры.

Экземпляр объекта *Задача* регистрируется в таблице *Задача* БД СВТ с использованием соответствующей формы. В форму вводятся данные о задаче, и в зависимости от методики, указанной в информации о задаче, запускается соответствующий конвертор.

Процедуры занесения в БД СВТ результатов счета задачи выполняются в цикле по количеству функций. Каждая функция с использованием соответствующей процедуры получает уникальный номер в БД. По имени функции одной процедурой запрашивается соответствующий ей номер в справочнике *Имена параметров*, после этого другой процедурой функция регистрируется в таблице *Определение функции*.

Точно так же для каждого аргумента по его имени запрашивается соответствующий ему номер в справочнике *Имена параметров*, после чего значения аргументов заносятся в таблицу *Значения аргументов*.

Значения функций заносятся в таблицу *Значения функции* в цикле по количеству моментов времени.

**3.3. Взаимодействие комплексов программ с БД СВТ в процессе основного счета.** Принципиальная схема взаимодействия ком-

плексов программ с БД СВТ в процессе основного счета приведена на рис. 5.

Как видно из схемы, взаимодействие с БД СВТ в ОС Windows и Linux осуществляется с помощью соответствующих библиотек процедур, созданных на основе интерфейсных средств Oracle. Эти библиотеки являются неотъемлемой частью комплексов и присоединяются к ним при сборке. Функционально библиотеки под ОС Linux и Windows практически эквивалентны.

Среда программирования Oracle [1] включает наряду с основным языком программирования PL/SQL также дополнительные интерфейсы с другими языками программирования. В их состав входят Oracle Call Interface (OCI), Java, Pro\*C/C++ (для Linux и Windows), Oracle Objects for OLE (для Windows). Все эти средства интерфейса предоставляют примерно одинаковые возможности по формированию блоков PL/SQL и передачи их на сервер для по-

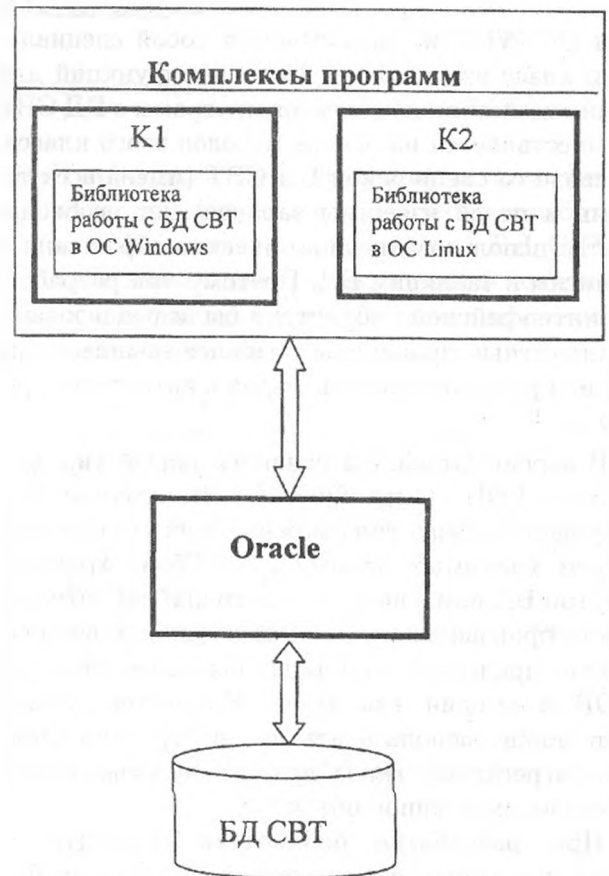


Рис. 5. Схема взаимодействия комплексов программ с БД СВТ в процессе основного счета



следующего выполнения запроса и разбора ответа после его завершения. В результате анализа программного обеспечения под управлением ОС Linux в качестве средства интерфейса был выбран клиент Pro\*C/C++ для языка программирования С. Для программирования под Windows использовалось объектно-ориентированное программное средство Oracle Objects for OLE.

Прекомпилятор Pro\*C/C++ под ОС Linux позволяет встраивать операторы SQL в текст программы на С или С++. После предварительной обработки формируется модифицированный текст программы со встроенными вызовами библиотечных функций прекомпилятора, после чего программа компилируется обычным образом. В операторах SQL можно использовать переменные, массивы и структуры. В программах допускается использование объектных типов или коллекций объектов, а также формирование динамических запросов SQL. Для более сложных вариантов взаимодействия в программах на Pro\*C/C++ можно использовать и вызовы OCI.

Программное средство Oracle Objects for OLE под ОС Windows представляет собой специальный класс языка С++. Создание функций для взаимодействия комплексов программ с БД СВТ осуществляется на основе методов этого класса. В связи со спецификой БД СВТ (имена всех таблиц и полей известны заранее) нет необходимости использовать динамические обращения к записям и таблицам БД. Поэтому при разработке интерфейсной библиотеки были реализованы стандартные процедуры, использовавшиеся при записи результатов счета задач в автономном режиме.

В версии Oracle 9.2 появился новый тип данных — LOB (Large Object). В качестве преимуществ нового типа можно отметить больший объем хранимых данных (до 4 Гбайт хранится внутри БД или в виде внешнего файла), возможность произвольного доступа к данным, возможность определить в таблицах несколько столбцов LOB, а не один, как ранее. Кроме того, новый тип можно использовать при построении сложных агрегатных типов данных пользователей и в коллекциях типов объектов.

При разработке библиотеки процедур из трех доступных возможностей реализации был выбран вариант со встроенными SQL-операторами, что позволило получить эквивалент PL/SQL-интерфейса и избежать сложностей OCI-интерфейса.

#### 4. Взаимодействие системы СВТ с прикладными системами

Общая схема взаимодействия системы СВТ с прикладными системами представлена на рис. 6. Основными компонентами в этой схеме являются:

- сервер доступа к БД СВТ (COM-сервер);
- клиент прикладной системы (COM-клиент);
- библиотека функций доступа к БД СВТ (см. подразд. 3.1).

Сервер доступа к БД СВТ реализован в виде COM-сервера [7] внутри процесса (in-process) и хранится в виде библиотеки DLL. Для доступа к БД Oracle сервер использует библиотеку классов Oracle Objects for OLE. COM-сервер реализует два типа интерфейсов доступа к своим функциям:

- 1) COM-интерфейсы;
- 2) DISP-интерфейсы.

Первый тип интерфейса предназначен для клиентских программ, написанных на языке программирования С++, и является быстрым интерфейсом доступа.

Второй тип интерфейса предназначен для клиентских программ, написанных на скриптовых языках, таких как Visual Basic, JavaScript, VBScript. DISP-интерфейсы работают медленнее COM-интерфейсов, но зато они более универсальны.

Для взаимодействия с системой Matlab используется промежуточная библиотека DLL, которая, с одной стороны, использует библиотеки Matlab для формирования объектов Matlab (скалярные величины, векторы, матрицы), а с другой стороны, является COM-клиентом сервера доступа (COM-сервера).

Для взаимодействия с сервером доступа из среды Excel используется язык программирования Visual Basic for Application (VBA) [8]. Это скриптовый язык, и поэтому взаимодействие с сервером доступа (COM-сервером) осуществляется с использованием DISP-интерфейса.

Система VIZI1D [4] реализована на языке программирования С++, поэтому для взаимодействия с сервером доступа (COM-сервером) используются COM-интерфейсы. Для системы VIZI сервер доступа предоставляет дополнительные возможности по выбору множества функций одновременно из разных методик и разных

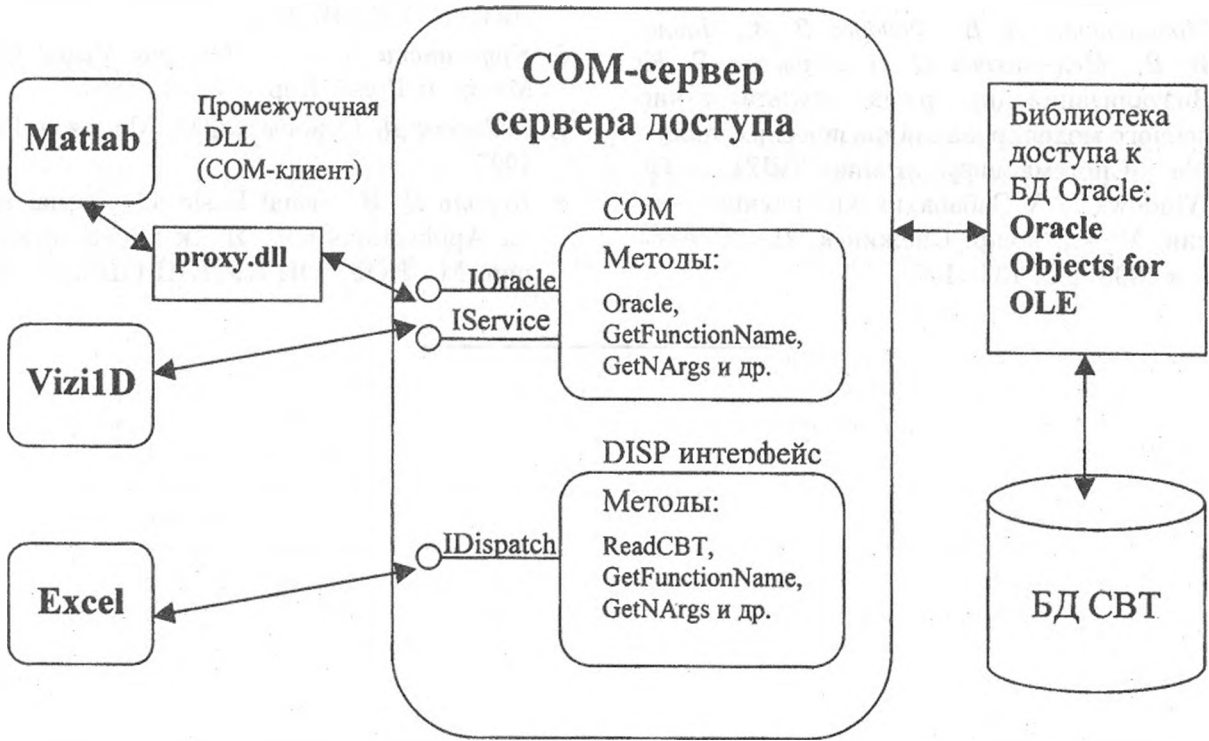


Рис. 6. Схема взаимодействия системы SOBT с прикладными системами

задач, а также по сохранению и восстановлению пользователем значений функций, моментов времени, аргументов в дисковом архиве.

Сервер доступа выводит на экран пользователя ряд диалоговых окон, в которых предлагается выбрать базу данных СВТ, методики, задачи, функции, моменты времени и аргументы.

### Заключение

В настоящее время система SOBT применяется в РФЯЦ-ВНИИТФ в ходе математического моделирования физических процессов. С ее помощью анализ результатов расчетов осуществляется на качественно новом уровне и значительно быстрее по времени, чем ранее, в традиционной файловой технологии. Это достигается за счет:

- унифицированной (единой для всех комплексов программ) структуры хранения результатов счета в единой БД под управлением СУБД Oracle;
- унифицированного (единого для всех пользователей и прикладных систем) интерфейса по доступу к информации в БД СВТ;

- унификации имен функций и аргументов, позволяющей физикам-теоретикам работать с информацией, хранящейся в БД СВТ, в привычных для себя терминах и понятиях;
- эффективности СУБД Oracle для хранения и обработки данных.

Система SOBT постоянно развивается и совершенствуется. В настоящее время основными направлениями развития являются:

- расширение возможностей по аналитической обработке хранящейся в БД СВТ информации;
- оптимизация процесса взаимодействия с БД СВТ.

### Список литературы

1. Мэйкат К. Oracle 7.3. Энциклопедия пользователя. "Диасофт", Киев, 1997.
2. Марк Д., Крис К., Крейг С. Running Microsoft Excel для Windows. М.: Издательский отдел "Русская редакция", 1995.

3. Потемкин В. Г. Matlab 5.0. М.: Диалог—МИФИ, 1998.
  4. Могиленских Д. В., Федоров В. В., Павлов И. В., Мельникова С. Н., Урусова Е. Ю. Визуализация двумерных результатов численного моделирования физических процессов. Система визуализации "VIZI" в ОС Windows // V Забабахинские научные чтения. Межд. конф. Снежинск, 21—25 сентября 1998 г. С. 134—135.
  5. Маклаков С. В. Создание информационных систем с Allfusion Modeling Suite. М.: Диалог—МИФИ, 2003.
  6. Круглински Д. Дж. Основы Visual C++. Microsoft Press: Пер. с англ., 1997.
  7. Роджерс Д. Основы COM. Microsoft Press, 1997.
  8. Король В. И. Visual Basic 6.0, Visual Basic for Applications 6.0. Язык программирования. М.: НОУ "ОЦ КУДИЦ-ОБРАЗ", 2000.
-