

время как переносимые алгоритмы могут иметь переменные возможности распараллеливания, относительно все алгоритмы ЦОС напрямую имеют возможности для оптимизации. Этот параллелизм присутствует там, где большие внутренние циклы могут быть выполнены в параллельном режиме на независимых потоках данных (многоканальная оптимизация) или где нет никакой зависимости данных между последовательными итерациями для единственного набора данных (оптимизация единственного канала).

В качестве базового алгоритма для создания системы тестирования выбран – HINT.

Этот эталонный тест основан на численном моделировании монотонно убывающей функции

$$z = n \frac{(n - x)}{(n + x)},$$

где  $0 \leq x \leq n$ . Как видно,  $z$  также лежит в пределах от 0 (при  $x=n$ ) до  $n$  (при  $x=0$ ). Таким образом, вся функция лежит на плоскости внутри квадратной области размером  $n \times n$ . Что касается  $n$ , то можно считать, что это максимально возможное целое число, которое позволяет представить принятый в тестируемой системе формат представления чисел. В любом случае  $n$  будет равно  $2^k$ , где  $k$  – величина, связанная с форматом представления чисел. Например, при использовании формата с фиксированной точкой  $k$  – это число разрядов регистра.

Численное моделирование указанной функции сводится к ее кусочно-ступенчатой аппроксимации. При этом аппроксимация ведется сразу с двух сторон (строятся две ломаных): сверху и снизу.

#### ЛИТЕРАТУРА

1. Цивилёв А.М., Дорохин С.А., Смехов Д.Г. Плата ADP160QPCI v1.0 Руководство пользователя.
2. ADSP-21160 SHARC Data Sheet.

**В.Ф. Гузик, В.Е. Золотовский, В.А. Переверзев**

#### **ТРЕНАЖЕРНЫЕ СИСТЕМЫ НА БАЗЕ ЛОКАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ**

Умение человека выполнять ту или иную работу зависит от индивидуальных особенностей интеллекта, способности переносить перегрузки. Для определения соответствия способностей отдельного человека требованиям, предъявляемым тем или иным видом деятельности, служит профессиональный отбор. Существует много различных методик отбора, позволяющих довольно точно определить будущих специалистов. Однако еще более важная роль в процессе становления нового специалиста отводится процессу обучения, а в особенности его практической части.

Анализируя современные методы подготовки, можно сделать выводы, что обучение операторов на реальной технике невыгодно из-за следующих недостатков:

- тренировки, как правило, дорогостоящие;
- техникой для проведения тренировок не всегда удастся воспользоваться;
- временные затраты на проведение тренировок ограничены сроком и программами обучения специалистов;
- выпущенный специалист вынужден пополнять свои знания и совершенствовать приобретенные навыки уже в процессе самостоятельной практической работы;
- оценка результатов в процессе тренировок иногда затруднительна.

Избавиться от перечисленных недостатков или, по крайней мере, несколько уменьшить их помогают тренажеры.

Тренажер — это техническое устройство, предназначенное для выработки необходимых навыков и умения у человека или группы людей в результате создания имитированной информационной модели, сложность которой изменяется в зависимости от обучения.

Тренажеры сложных систем обычно стараются сделать как можно более точной копией образца. Стоимость такого тренажера может быть очень высока и приближается к стоимости самого комплекса. И здесь на помощь приходит ЭВМ, на которой можно построить математическую модель системы, способную воспроизводить все ситуации, возникающие при работе реального объекта. Кроме того, на ЭВМ можно изменять темп тренировки. Так, начинающего оператора можно тренировать в медленном темпе, а по мере приобретения им опыта увеличивать темп.

Например, применительно к построению тренажера операторов гидроакустических комплексов можно выделить три основные задачи, решаемые в процессе подготовки операторов с помощью тренажера.

Первая задача заключается в создании имитированной информационной модели подводной обстановки, аналогичной по своим воздействиям на оператора в реальной обстановке. Первые тренажеры использовали упрощенные модели, не позволявшие имитировать все возможные ситуации. Но существующие сегодня высокопроизводительные вычислительные машины позволяют при воссоздании достаточно сложной подводной обстановки в зоне обзора ГАС полностью заменить имитацию математическим моделированием.

Вторая задача состоит в получении информации о реакции оператора на поступающие тестовые ситуации. Это позволяет проводить оценку подготовки обучаемого в процессе обучения.

Третьей задачей является управление учебной информационной моделью подводной обстановки в ходе обучения оператора. Решение этой задачи также целесообразно возложить на ЭВМ, которая с помощью обучающей программы должна модифицировать информационную модель подводной обстановки в зависимости от успешности обучения и выдать необходимые рекомендации по дальнейшим действиям.

Решение вышеописанных задач эффективно решаются с помощью современных компьютерных технологий. Сама структура тренажера предполагает наличие как минимум двух автоматизированных рабочих мест: рабочего места инструктора и рабочего места оператора. Оба эти рабочих места должны иметь возможность взаимодействовать между собой. Поэтому одним из важных вопросов, влияющим на дальнейшую структуру тренажера, является реализация обеспечения сетевого взаимодействия операторов и инструктора.

В результате проведенного анализа ряда сетевых технологий Ethernet, Fast Ethernet, Token Ring, Gigabit Ethernet и FDDI было показано, что технология Ethernet не обеспечивает необходимую скорость передачи данных. Технология Token Ring недостаточно хорошо поддерживается необходимым оборудованием на отечественном рынке. Технологии Gigabit Ethernet и FDDI достаточно хорошо известны, поддерживаются на российском рынке коммуникационного оборудования и обеспечивают высокую пропускную способность каналов до 800 Мбайт/с. Однако для уровня локальных сетей применение таких высокоскоростных технологий экономически нецелесообразно. Таким образом, наиболее оптимальной по показателю скорость передачи данных/стоимость оборудования является технология Fast Ethernet.

Общая структура комплекса подготовки операторов базируется на локальной компьютерной сети и использует технологию «клиент-сервер». Сетевые станции, на которых устанавливается программное обеспечение рабочих мест инструкторов, являются серверами. Соответственно места операторов – клиентами. Возможно объединение станций операторов в рабочие группы. Каждая рабочая группа подключается к одному соответствующему серверу - инструктору. Физическая структура сети комплекса представлена на рис. 1.

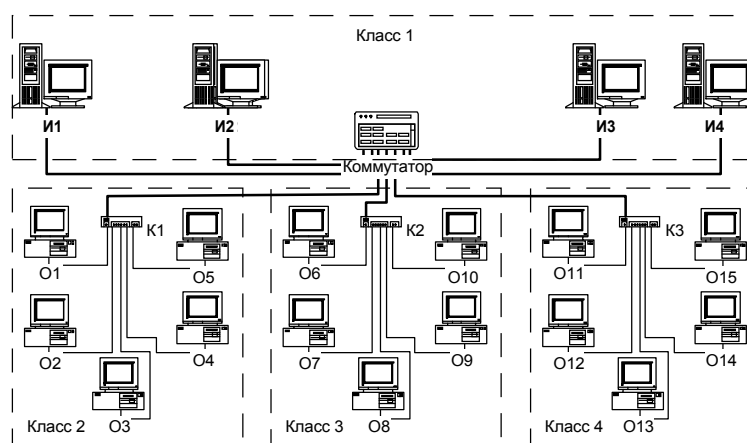


Рис. 1

Рабочие станции операторов объединяются в классы по локальному признаку. Объединение рабочих станций в одном классе осуществляется посредством концентратора. Концентратор физически объединяет станции операторов в узел. Выходной порт каждого концентратора подключается к головному коммутатору, так же к этому коммутатору подключаются рабочие станции инструкторов. Поскольку порты коммутаторов работают параллельно, то их использование на верхнем уровне сети позволяет осуществить одновременное подключение двух классов операторов к соответствующим двум серверам инструкторов.

В качестве основного средства разработки программного обеспечения комплекса был выбран продукт Borland Delphi. Выбор данного продукта обусловлен рядом его достоинств: быстрый оптимизирующий компилятор, большой набор

средств поддержки сетевых приложений и баз данных. Средством управления базами данных комплекса была выбрана система Borland Database Engine (BDE), которая обеспечивает работу с различными типами баз данных. Банки исходных данных предполагается хранить в формате таблиц Paradox. Данный формат обеспечивает необходимые средства для хранения различных типов данных – от изображений до символьных строк.

Выбор архитектуры банка данных комплекса был осуществлен из трех наиболее распространенных архитектур: архитектуры файл-сервер, архитектуры клиент-сервер и трехзвенной архитектуры клиент-сервер.

Наиболее оптимальной для такой структуры тренажера оказалась клиент-серверная архитектура, представленная на рис. 2. Здесь клиент формирует запрос к серверу на языке запросов SQL. SQL-сервер обеспечивает интерпретацию запроса, его выполнение, формирование результата и выдачу этого результата клиенту. При этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса: клиентский компьютер лишь отправляет запрос к серверной БД и получает результат, после чего интерпретирует его необходимым образом и представляет пользователю. Все это повышает быстродействие системы и снижает время ожидания результата запроса.

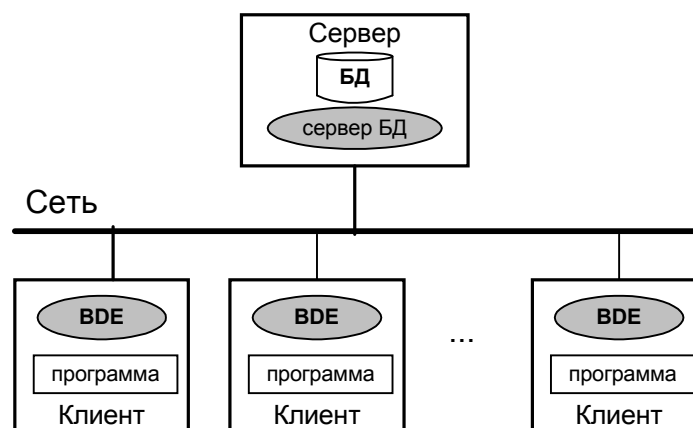


Рис. 2

При разработке программного обеспечения тренажёрного комплекса необходимо решать задачи визуализации подводной обстановки – типовых сцен расположения локализуемых объектов. Данные задачи могут быть успешно решены при помощи средств компьютерной графики – графического ускорителя и графической библиотеки. Одним из первых и наиболее распространенных стандартов графической библиотеки является OpenGL (открытая графическая библиотека). Хотя библиотека OpenGL и считается одной из лучших библиотек, как для профессионального применения, так и для игр, у неё существуют и конкуренты. Среди них можно отметить такие, как GLide, Heidi, а также одного из главных конкурентов – Direct3D из пакета DirectX, разработанного фирмой Microsoft.

С точки зрения программиста, OpenGL – это программный интерфейс для графических устройств, таких как графические ускорители. Он включает в себя

около 150 различных команд, с помощью которых программист может определять различные объекты и производить рендеринг.

Также необходимо отметить, что так как команды библиотеки OpenGL независимы от операционной системы, а среда проектирования Delphi совместима с Kylix, то за счет объединения их возможностей можно создавать мультиплатформенные приложения, работающие на различных операционных системах (Windows, Unix, Linux), что должно обеспечить большую гибкость программного обеспечения комплекса.

Таким образом, применение всех вышеописанных принципов при построении позволит создать мощный тренажерный комплекс, отвечающий всем современным требованиям.

**М.В.Савельев**

### РЕАЛИЗАЦИЯ СОЕДИНЕНИЙ В РАЗВИВАЮЩИХСЯ СИСТЕМАХ

При проектировании развивающихся систем сетевой структуры с большим и средним числом узлов (300–800) характерно появление новых связей, способ соединения которых задаётся функциональной схемой. Создание соединений является одной из трудоёмких задач [1]. При этом необходимо учитывать: минимальную длину связей, пропускную способность каналов связи, соединения не должны пересекаться. Помимо частной задачи соединения заданных узлов [2– 4], наиболее важной (предшествующей задачей) является наилучшее размещение узлов, чтобы затем последующее соединение было также наилучшим. Синтез алгоритмов дискретно-непрерывной оптимизации не допускает декомпозицию задачи без потери качества [2]. Поэтому предлагается использовать метод агрегатирования при сохранении некоторой неопределённости и расплывчатости цели.

В этом случае процедура движения выполняется по множествам [4], которые не могут быть заранее определены, но создаются во время выбора связей в динамическом режиме. Размещение узлов ведётся при последовательном их соединении, включая очерёдность связей и самих узлов. Расположение узла не меняется, а связь с другими узлами меняется.

В реальных условиях оптимальное расположение узлов и их взаимное соединение может уточняться под действием различных факторов, учитывая и кратчайшие связи. Это обычно зависит от принятых традиций проектирования (когда наиболее важными считаются текущие затраты и не рассматриваются перспективные проблемы управления развивающейся системы) или неполной формализации процесса.

Пусть на существующей сети в определённой области  $R$  имеется конечное число  $N$  узлов в виде множеств  $M_i$ ,  $M_i \subset R$ ,  $i \in 1, N$ , взаимно непересекающиеся между собой. В этой области от заданной начальной точки  $x_0$  проводится ломаная линия (связь), которая последовательно соединяет заданные множества узлов инженерной сети  $m_i \in M_i$ ,  $i \in 1, N$ ;

Необходимо создать соединение минимальной длины:

$$\sum_{i=1}^N |m_i - m_{i-1}| \rightarrow \min, \quad i \in 1, N, \quad m_k \in M_k.$$