Кафедра информатики и вычислительной техники

Курсовая работа

по информатике на тему:

Методы организации сложных экспертиз компьютерных систем

***Содержание***

Введение

Глава 1.Методы сложных экспертиз компьютерных систем

1.1 Тесты DHRYSTONE, LINPACK и "ЛИВЕРМОРСКИЕ ЦИКЛЫ"

1.2 Методика SPEC

1.3 Тест ICOMP 2.0 для оценки эффективности микропроцессоров INTEL

1.4 МЕТОДИКА AIM

1.5 МЕТОДИКА ОЦЕНКИ СКОРОСТИ ОБРАБОТКИ ТРАНЗАКЦИЙ

1.6 МЕТОДИКА ОЦЕНКИ ГРАФИЧЕСКИХ ВОЗМОЖНОСТЕЙ

1.7 МЕТОДИКА ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ СУПЕРКОМПЬЮТЕРОВ

1.8 МЕТОДИКА ОЦЕНКИ КОНФИГУРАЦИЙ WEB

Глава 2. Описание модуля электронного учебника по теме "Методы организации сложных экспертиз компьютерных систем"

2.1 Понятие и требования к электронному учебнику

2.2 Структура и содержание учебника

Заключение

Библиографический список

Приложение

***Введение***

В современном мире компьютерные технологии приобретают всё большее распространение. Вычислительные системы применяются для решения самых разнообразных задач, что требует их постоянного усовершенствования. Для этой цели были разработаны, а в наше время активно применяются различные методики оценивания компьютерных систем.

Цель работы – провести анализ современных методов организации сложных экспертиз компьютерных систем и разработать модуль электронного учебника по теме "Методы организации сложных экспертиз компьютерных систем".

Объект – процесс создания и экспертизы компьютерных систем.

Предмет – теоретические основы проведения сложных экспертиз компьютерных систем.

Задачи:

1. Провести анализ современных методов сложных экспертиз компьютерных систем.
2. Разработать модуль электронного учебника по теме "Методы организации сложных экспертиз компьютерных систем".

Практическая значимость исследования заключается в том, что разработанный модуль электронного учебника по теме "Методы организации сложных экспертиз компьютерных систем" может быть использован для создания электронных учебников по дисциплинам информатики.

***Глава 1. Методы сложных экспертиз компьютерных систем***

Теоретические положения системного анализа определенное время рассматривались только как некая философия инженера и поэтому при решении задач создания искусственных систем иногда не учитывались. Однако развитие техники привело к тому, что без системного анализа, одним из результатов которого являются концептуальные модели, исследование функционирования систем становится невозможным.

Первоначально компьютер отождествлялся с центральным процессором, основной и понятной характеристикой которого было быстродействие, измеряемое числом команд в единицу времени. Поэтому традиционные методики оценки (benchmarks) отражают только возможности центрального процессора. В основе такой оценки лежит понятие производительности. При этом выделяют так называемое "чистое" процессорное время - период работы собственно процессора при выполнении внутренних операций и время ответа, включающее выполнение операций ввода-вывода, работу ОС и т.д.

Есть *два показателя производительности процессоров* по "чистому" времени:

1) показатель производительности процессоров на операциях с данными целочисленного типа (integer) MIPS (Million Instruction Per Second - миллион машинных команд в секунду) - отношение числа команд в программе к времени ее выполнения;

2) показатель производительности процессоров на операциях с данными вещественного типа (float point) MFLOPS (миллион арифметических операций над числами с плавающей точкой в секунду).

С понятием MIPS связывалась ранее и другая метрика, основанная на производительности вычислительной системы DEC VAX 11/780. Еще одно определение MIPS используется пользователями и производителями техники IBM, когда за норму выбирается одна из моделей RS/6000. При этом 1 MIPS IBM = 1.6 MIPS DEC.

При всей кажущейся простоте критерия оценки (чем больше MIPS (MFLOPS), тем быстрее выполняется программа) его использование затруднено вследствие нескольких причин:

1. Процессоры разной архитектуры (особенно RISC) имеют различный набор команд. Так, совмещенная операция умножения и сложения векторов в процессоре POWER 2 существенно сокращает число операций. Кроме того, можно выделить "быстрые" (например, сложение, вычитание) и "медленные" (например, деление) операции, а в результате рейтинг MFLOPS для разных программ окажется разным.
2. Применение математических сопроцессоров и оптимизирующих компиляторов увеличивает производительность системы, однако рейтинг MIPS может уменьшиться, так как время выполнения команд для операций над данными с плавающей точкой значительно больше и за единицу времени может быть выполнено меньшее число команд, нежели при выполнении соответствующих этим командам подпрограмм.
3. Научные приложения в основном связаны с интенсивными вычислениями над вещественными числами с плавающей точкой, коммерческие и офисные – с целочисленной арифметикой и обработкой транзакций баз данных. Графические приложения критичны и к вычислительным мощностям, и к параметрам графической подсистемы.

Ещё более сложные проблемы появляются при необходимости оценок многопроцессорных систем, в частности SMP (Symmetric MultiProcessing – симметричная мультипроцессорная обработка) к МРР(Massively Parallel Processing - обработка с массовым параллелизмом). В целом показатели MFLOPS и MIPS зависят от архитектуры процессора и типа выполняемой программы. Такое положение привело к разработке и использованию ряда тестов, ориентированных на оценку вычислительных систем с учетом специфики их предполагаемого использования. Поэтому оценка процессоров с разной архитектурой основана на создании тестовой смеси из типовых операторов, влияющих на их производительность.

***1.1 Тесты DHRYSTONE, LINPACK и "ЛИВЕРМОРСКИЕ ЦИКЛЫ"***

Для работы с показателями MIPS и MFLOPS чаще всего используются системы тестов Dhrystone, LINPACK и "Ливеморские циклы".

Тестовая смесь Dhrystone состоит из 100 команд: 53 - операторы присвоения, 32 - управления и 15 - вызова функций. Результатом работы этого теста является число Dhrystone в секунду. При этом на системе DEC VAX 11/780 результат составлял 1757 Dhrystone, и поэтому считалось, что 1 DEC MIPS равен 1757 Dhrystone. Сейчас Dhrystone практически не применяется.

Тесты LINPACK и "Ливерморские циклы" появились в середине 60-х гг.

"Ливерморские циклы" состоят из фрагментов программ для решения численных задач на языке Фортран, имеющих реальное хождение в Ливерморской национальной лаборатории им. Лоуренса в США. В этих фрагментах используются различные вычислительные алгоритмы: сеточные, последовательные, волновые, что существенно относительно соответствия вычислительных и аппаратных структур. Соответствие этих структур друг другу должно обеспечить максимальную эффективность вычислений. При проведении тестовых испытаний может использоваться либо набор из 14 циклов (малый набор), либо набор их 24 циклов (большой набор). При использовании векторных и параллельных машин важным фактором, определяющим эффективность работы для конкретного приложения, является коэффициент векторизуемости алгоритма. На "Ливерморских циклах" этот коэффициент составляет от 0 до 100 % , что подтверждает возможность их применения для широкого круга вычислительных систем.

LINPACK включает набор программ на Фортране, предназначенных для решения систем линейных алгебраических уравнений. Важность этого тестового набора, так же как и "Ливерморских циклов", определяется практической значимостью и применимостью этих алгоритмов для решения реальных задач. В основе используемых в LINPACK алгоритмов лежит метод декомпозиции: исходная матрица представляется в виде произведения двух матриц стандартной структуры, к которому собственно и применяется алгоритм нахождения решения. Важная особенность системы LINPACK - ее структурированность. В частности, выделяется базовый уровень системы, обеспечивающий реализацию элементарных операций над векторами, куда входят подпрограммы умножения векторов на скаляр и сложения векторов, а также скалярного произведения векторов. Этот уровень называется BLAS (Basic Linear Algebra Subprograms). Все операции выполняются над вещественными числами двойной точности, а результат измерения выражается в М FLOPS.

В настоящее время используются два уровня теста: LINPACK DP - для исходной матрицы размером 100x100 и LINPACK ТРР - для матрицы размером 1000x1000. Для многих современных вычислительных систем первый уровень этого теста может дать заведомо превосходящие возможности системы результаты за счет того, что исходная матрица размером 100x100 может быть целиком размещена в кэш-памяти. Использование теста LINPACK ТРР пока снимает эту проблему, однако даже и этот тест для систем с массовым параллелизмом не может быть использован. Для таких систем рекомендуется тест LINPACK HPC (Highly Parallel Computing), который позволяет полностью загрузить вычислительные ресурсы МРР-системы, увеличивая размеры матрицы. При этом следует иметь в виду, что для параллельных систем (SMP и МРР) применяются специальные варианты этого теста, обеспечивающие распараллеливание вычислений.

***1.2 Методика SPEC***

Ведущие производители компьютерных систем в 1988 г. создали некоммерческую корпорацию SPEC (Strandard Performance Evaluation Corporation), призванную дать объективную оценку производительности вычислительных систем. Корпорация SPEC является разработчиком тестов, проводит тестирование и публикует результаты в специальном бюллетене "The SPEC Newsletter", который размещается на WWW-сервере www.SPEC.com. Оценки, публикуемые комитетом SPEC, являются официальными, признаваемыми всеми разработчиками тестов.

Основным набором в SPEC был тест SPECint89 для оценки процессора на операциях с данными целочисленного типа и SPECfp89 для оценки при работе с данными вещественного типа. Появление в начале 90-х гг. нового поколения RISC-процессоров (PowerPC, РА-7200, MIPS, Rxxxx) сделало невозможным использование этого набора из-за резкого уменьшения времени выполнения и влияния на производительность оптимизирующих компиляторов. Тестовый набор был преобразован в смеси SPECint92 и SPECfp92, учитывающие эффективность работы с памятью. Производительность тестируемой системы измерялась в условных единицах относительно базовой DEC VAX 11/780.

Комплексный показатель качества по методике SPEC определяется как среднегеометрическое времени выполнения программ, входящих в тестовую смесь. При этом использовалось среднее значение дли всех тестов, образуемых SPECint92 и SPECfp92. С разработкой нового поколения оптимизирующих компиляторов для RISC-процессоров консорциум SPEC в 1994 г. внес новые поправки-требования к используемым компиляторам. Тесты получили название SPECbase\_int92 и SPECbase\_fp92 и применялись для оценки работы в однозадачном режиме. Известно, что некоторые однопроцессорные системы способны выполнить одну задачу быстрее многопроцессорных, однако этот факт не дает полной картины интегрального поведения системы в целом, так как многопроцессорные комплексы могут выполнять больше заданий в единицу времени, поэтому в режиме многозадачности оценка производительности основана не на вычислении времени выполнения тестовой смеси, а на пропускной способности системы, измеряемой количеством заданий выполненных за единицу времени.

Если один процессор за минуту выполняет одну работу, а система из четырех процессоров делает это за две, то многопроцессорная система работает в два раза медленнее, но имеет загрузку в два раза больше, чем однопроцессорная. Загрузка находится в прямой зависимости от размера кэш-памяти, скорости шины емкости оперативной памяти.

Набор тестовых программ для оценки пропускной способности SPECrate полностью аналогичен наборам SPECint92 и SPECftp92 - этоте же программы, но размноженные на несколько одновременно запускаемых копий. Результирующее значение по методике SPECrate вычисляется по формуле:

SPECrate = число\_копий \* ref\_const \* cpu\_const / общее\_время.

Число одновременно выполняемых задач может выбираться произвольным образом. Очевидное решение - число, равное количеству процессоров, однако для каждойконкретной архитектуры возможны свои особенности. Величины ref\_const и cpu\_const для каждого теста являются постоянными коэффициентами. Общее время - время завершения последней из всех запущенных работ.

В методике используется принцип одноразрядной загрузки (тестовая смесь SPECint92 и SPECftp92), а в качестве конечного результата выступает среднее значение по всем тестам. При работе в мультипрограммной системе может варьироваться количество запускаемых копий, а время фиксироваться по завершении выполнения последней копии. Оценки по данной методике называются SPECrate\_int92 и SPECrate\_ftp92.

C октября 1995 г. для оценки производительности процессоров, оперативной памяти и компиляторов был объявлен новый тестовый комплект, включающий SPECint95 для операций с данными целочисленного типа и SPECftp95 - для операций с данными вещественного типа. Эти тестовые наборы предъявляют следующие ограничения и требования: достаточно большой размер кода и данных, чтобы он гарантированно не размещался целиком в кэш-памяти; увеличения времени выполнения тестов с секунд до минут; реалистичность используемых фрагментов программ; применение усовершенствованного способа измерения времени; реализация более удобных инструментальных средств; стандартизация требований к компиляторам и методов вызова. Оценка систем проводится после пересчета результатов измерений по итоговому рейтингу— ранжировке систем относительно производительности базового процессора в соответствии с комплексным показателем. Подчеркивается, что задача комплексной оценки вычислительной системы в целом, включая периферийное оборудование, графическую подсистему, сетевое оборудование, ввод-вывод данных, остается за рамками тестов SPECint95 и SPECftp95.

В табл. 3.1 приведены результаты тестирования некоторых процессоров.

**Таблица 3.1 Результаты тестирование микропроцессоров**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Микропроцессор | Разрядность | Частота, МГц | SPECint95 | SPECftp95 |
| Alpha 21164 | 64 | 433 | 12,4 | 17 |
| PentiumPro | 32 | 233 | 9.3 | 7.4 |
| UltraSPARC | 64 | 200 | 6.5 | 11 |
| Alpha 21164 | 64 | 466 | 13 | 18 |
| MIPS R10000 | 64 | 275 | 12 | 24 |
| UltraSPARC-II | 64 | 250 | 8.5 | 15 |
| POWER2 Super | 32 | 133 | 5.5 | 15 |

Кроме собственно тестового набора комитетом SPEC разработан и инструментарий, использование которого является обязательным;

• средства оценки, *основанные на вычислении среднего* времени из серии запусков, исключающие внесение какого-либо дополнительного пользовательского кода или использование произвольной выборки из серии запусков тестовых программ. В отчете присутствует "базовое" время (References time) — время выполнения теста на эталонной машине, в качестве которой используется SPARCstation 10/40 в конфигурации с кэш-памятью второго уровня. В отчет включается также относительное время выполнения тестов по сравнению со временем отработки тестов на эталонной машине; Эта оценка является основной для данного набора;

• автоматическое формирование отчета, в котором должно присутствовать полное описание конфигурации тестируемой системы, операционной системы и ключей запуска компилятора.

Оценка производительности проводится по двум частным показателям:

• скорости выполнения теста с оптимизированным (SPECint\_95 и SPECftp\_95) и неоптимизированным (SPECint\_base\_95 и SPECftp\_base\_95) режимами компиляции;

• пропускной способности системы для многопроцессорных архитектур и/или для многозадачного режима работы в оптимизированном (SPECint\_rate95 и SPECftp\_rate95) и неоптимизированном (SPECint\_rate\_base95 и SPECftp\_rate\_basefp95) режимах работы.

Смесь SPECint\_95 включает 8, a SPECftp\_95 - 10 программ. Перечень, программ, время их выполнения на эталонной машине, прикладная область и характеризующие ее спецификации показаны в табл. 3.2. Набор целочисленных программ написан на языке Си, а для работы с плавающей арифметикой - на Фортране.

Как видно из табл. 3.2, в тестовый набор включены программы, используемые в различных прикладных областях. Это допускает проведение не только комплексного сравнения по итоговому рейтингу, но и узкоориентированного - до конкретной программе, в случае если предполагаемое использование вычислительной системы соответствует выбранной предметной области.

В комплект официальной поставки тестового набора входят исходные тексты программ тестового набора, инструментальные средства для компиляции, запуска, сравнения результатов и формирования отчета, описание правил запуска тестов и формирования отчета. В отчете указывается время прогона на тестируемой системе, относительное время по каждой программе и их файловой системе, используемые флаги и ключи.

**Табли**ца **3.2** Описание тестовых смесей по методике SPEC

|  |  |  |  |
| --- | --- | --- | --- |
| Программа | Базовое время, с | Область приложения | Спецификация задачи |
| Тест SPECint\_95 | | | |
| 099. go | 4600 | Искусствен-ный  интеллект | Игра Go - игра сама против себя |
| 124.m&&ksim | 1900 | Моделирова-ние | Моделирование чипа Motorola 88100 |
| 126. gcc | 1700 | Программиpование | Компиляция программы на Си и  компиляция в оптимизированный код для процессоров SPARC |
| 129.compress | 1S0O | Сжатие данных | Сжатие текстового файла размером 16 Мбайт |
| 130. li | 1900 | Интерпрета-ция языков | Lisp-интерпретатор |
| 132.ijpeg | 24D0 | Обработка  изображений | Сжатие изображений графических объектов (JPEG) с различными параметрами |
| 134. perl | 1900 | Shell-интер-  претатор | Манипулирование текстовыми строками |
| 147. vortex | 2700 | Базы данных | Построение и манипулирование таблицами |
| Teсm SPECftp\_95 | | | |
| 101. tomcatv | 3700 | Гидродина-  мика, геометричес-кие операции | Генерация двухмерной координатной сетки преобразования вокруг произвольной области |
| 102. swim | 8600 | Предсказа-ние  погоды | Моделирование волной поверхности методом конечных элементов (вещественная  арифметика с одинарной точностью) |
| 103. su2cor | 1400 | Квантовая физика | Вычисление массы элементарных частиц с использованием метода Монте-Карло |
| 104.hydro2d | 22400 | Астрофизика | Расчет межгалактических газов по уравнению Новье-Стокса |
| Программа | Базовое время, с | Область приложения | Спецификация задачи |
| 107. mgrid | 2500 | Электромаг-нетизм | Расчет трехмерного поля потенциалов |
| 110.applu | 2200 | Гидродина-мика | Решение системы уравнений с частными производными |
| 125. turb3d | 100 | Моделиро-вание | Моделирование турбулентностей в кубическом объеме |
| 141. apsi | 2100 | Предсказа-ние погоды | Вычисление статистики температур, воздушных потоков и уровней загрязнения |
| 145. fpppp | 9600 | Квантовая химия | Отработка порождения потока электронов |
| 146. wave | 3000 | Электромаг-нетизм | Решение уравнения Максвелла |

Оценки SPEC важны для анализа систем, основное назначение которых быть вычислителем вообще, без детального уточнения конкретной специфики. Тестовые наборы дают сравнение по работе с целыми и с вещественными числами.

Консорциум SPEC разработал кроме этих еще несколько тестов, среди которых SDM (тест рабочей нагрузки при использовании UNIX) и SFS (тест рабочей нагрузки файлового сервера). В тесте SDM моделируется многопользовательская среда UNIX и оцениваются как работа ОС, так и производительность процессора и операции ввода-вывода. Тест SFS включает в себя один пакет 097.LADDIS, в рамках которого генерируются типовые сетевые запросы к NFS-серверу. Отчет по этому тесту содержит описание полной конфигурации и время ответа на запросы NFS.

***1.3 Тест ICOMP 2.0 для оценки эффективности микропроцессоров INTEL***

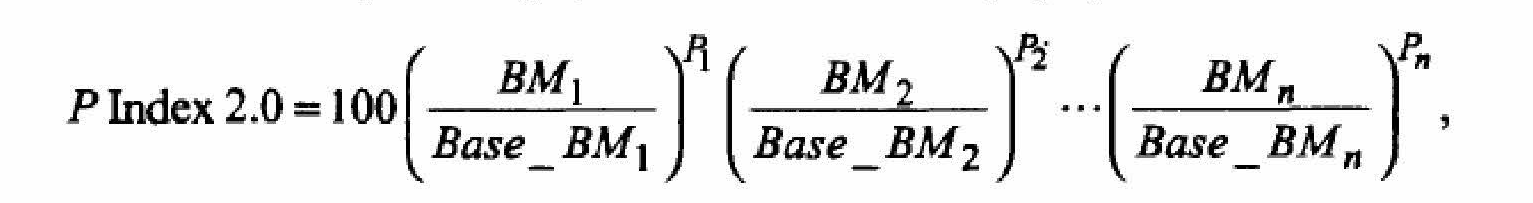
Корпорация Intel разработала тест iCOMP, ранжирующий по эффективности микропроцессоры различных семейств Intel-подобной архитектуры.

Тест iCOMP ориентирован только на выбор микропроцессоров для ПЭВМ. Тест не может служить интегральным показателем качества любых типов микропроцессоров, ПЭВМ или рабочих станций в целом, так как на общую эффективность влияют различия в аппаратных средствах и конфигурации программного обеспечения.

Со временем тест iCOMP был модифицирован и назван iCOMP 2.0. В нем отражены основные тенденции в формировании требований к оценке микропроцессоров: учет современных профилей прикладных программ, определяемых как соотношение времени выполнения регистровых операций ЦПУ, обмена с памятью и ввода-вывода; переход на 32-разрядные операционные системы и прикладные программы, включая Windows 95, NT, OS/2 и UNIX; быстрое увеличение объема мультимедийных, сетевых средств и средств обработки трехмерной графики.

Уникальные для основных прикладных программ смеси операций, определяющие их профили, показаны на рис. 3.1.

Оценка процессоров производится по взвешенному времени выполнения тестовой смеси, нормированному по эффективности базового процессора, в соответствии с формулой

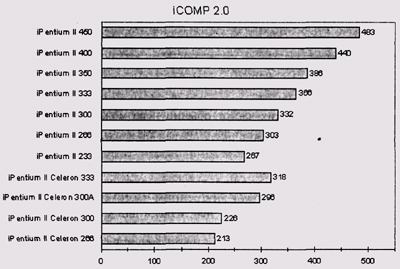


где *BMi -* время выполнения *i-*го теста;

*Рi -* вес *i-*го теста;

*Base\_BMi -* эффективность базового процессора на *i-м* тесте.

Из приведенной формулы следует, что индекс iCOMP 2.0 вычисляется как мультипликативная свертка времени работы процессора на каждом из эталонных тестов смеси.



**Рис. 3.1 Профили типовых прикладных программ**

Состав тестовой смеси выбран так, чтобы охватить различные категории прикладных программ и объемы загрузки процессора. Перечень категорий прикладных программ, состав тестовой смеси *(BMi)* и веса тестов *(Pi),* используемые для расчета индекса iCOMP 2.0 (табл. 3.3), определены исходя из анализа рыночного спроса программ различного типа.

Таблица 3.3 **Категории программ** и **веса тестов** iCOMP 2.0

|  |  |  |
| --- | --- | --- |
| Категория программ | Состав тестовой смеси *(ВМi)* | Вес тестов |
| Инженерные программы типа Autocad | Norton S132 | 15 |
| Программы типа Microsoft Office | CPUmark32 | 40 |
| Программы, оперирующие данными целого типа | SPECint\_base95 | 20 |
| Программы, оперирующие данными с плавающей точкой | SPECfp\_base95 | 5 |
| Программымультимедиа | Intel Media Benchmark | 20 |

За базовый процессор принят Pentium - 120МГц, имеющий оценку, равную 100 ед.

Эффективность базового процессора *(Base\_BMi),* определенная по различным тестам, представлена ниже.



Перечень микропроцессоров, отранжированных по индексу iCOMP 2.0, приведен ниже.

Оценки, основанные на iCOMP 2.0, не могут сравниваться с оценками, основанными на iCOMP, так как они получены из различного набора эталонных тестов с различными весами и нормированы на различный базовый процессор.

В тесте iCOMP за базовый процессор принят процессор Intel486SX, 25МГц.



***1.4 МЕТОДИКА AIM***

Сравнение и оценка производительности вычислительных систем применительно к конкретному приложению и планируемому использованию проводятся по методикам независимой компании AIM Technology, основанной в 1981 г. Предлагаемые AIM Technology методики и тестовые смеси ориентированы на получение интегральных оценок по всем компонентам UNIX-систем в многопользовательском и многозадачном режимах. Разработанные методики позволяют получить более комплексную оценку тестируемой архитектуры, чем тесты SPEC и iCOMP 2.0. Результаты тестовых испытаний систем можно получить на сервере www.ideas.com.au/bench/aim/aim.htm.

В методике AIM при проверке учитываются следующие критерии:

• пиковая производительность (AIM *Performance Rating) -*максимальная производительность в режиме наиболее оптимального использования центрального процессора, процессора работы с вещественными числами и кэш-памяти;

• максимальная нагрузка *(Maximum User Load)* - максимально возможное число заданий при работе наибольшего числа пользователей, которое может выполнить система за минуту. Данный показатель используется при выборе серверов;

• обработка утилит Unix *(Utilities Indexed* или *Milestone) -*оценка возможностей по выполнению 40 утилит ОС Unix. Данный показатель используется при выборе инструментального компьютера, предназначенного для интенсивной работы с утилитами типа *grep* или *make;*

• пропускная способность *(Throughput Graph) -* показатель производительности (число работ в минуту) в зависимости от степени загрузки системы;

• цена *(Price) -* стоимость тестируемой компьютерной системы.

Производительность при выполнении Unix-утилит идентифицирует системы, наиболее эффективно выполняющие утилиты ОС Unix за одну минуту. Основным набором оценки собственно компьютерной системы, без вывода на терминалы, учета производительности при работе с X Window и в составе сети, является AIM *System Benchmark (Suite lit).*

Набор состоит из шести так называемых моделей: обмены с оперативной памятью (20 %), работа с вещественными числами двойной и одинарной точности (10 %), операции работы с целыми числами (20 %), обмены данными между процессорами (10 %), вызовы функций на языке Си с 0, 1, 2 и 15 параметрами (20 %), ввод-вывод на диск (20 *%).* Ниже приведены результаты сравнения компьютерных систем, полученные фирмой AIM.



Полный отчет по компьютерной системе включает данные тестирования по набору тестов AIM *Subsystem Benchmark (Suite III).* Проверка по данному набору производится при работе компьютера в однозадачном режиме и включает следующие оценки производительности:

• *при работе с диском.* Измеряется в килобайтах в секунду для двух вариантов: при использовании кэширования и без него. Оценка применяется при выборе систем для работы с базами данных, файловых серверов и рабочих мест разработчика программного обеспечения;

• *при выполнении операций над вещественными числами.* Измеряется в тысячах операций в секунду отдельно для сложения, умножения и деления, с двойной н одинарной точностью. Оценка используется при выборе систем для работы в научных и физических приложениях;

• *при работе с целыми числами.* Измеряется в тысячах операций в секунду отдельно для сложения, умножения и деления чисел в длинном *(long)* и коротком *(short int)* форматах. Оценка используется при выборе систем для работы в финансовых приложениях;

• *для операций чтения/записи в память.* Измеряется в килобайтах в секунду отдельно при чтении и записи целых чисел в длинном и коротком форматах, а также символов. Оценка используется при выборе компьютеров для работы с издательскими системами и в финансовых приложениях;

• *для операций копирования в памяти.* Измеряется в килобайтах в секунду при пересылке целых чисел в длинном и коротком форматах, а также символов;

• *для операций в памяти над массивами ссылок.* Измеряется в тысячах ссылок в секунду для целых чисел в длинном и коротком форматах;

• *при вызове системных функций.* Измеряется количеством обращений в секунду к таким функциям Unix, как create/close, fork, signal и unmask;

• *при вызове функций в прикладной задаче.* Измеряется количеством вызовов в секунду для функций без аргументов, функций с одним, двумя и пятнадцатью параметрами типа int.

Компания AIM Technology разработала также специальные наборы тестовых смесей, характеризующие использование вычислительной системы в следующих прикладных областях: General Workstation Mix - среда разработки программного обеспечения; Mechanical CAD Mix - среда автоматизации проектирования в машиностроении (с использованием трехмерной графики); GIS Mix - среда геоинформационных приложений; General Business -среда стандартных офисных приложений (электронные таблицы, почта, тестовые процессоры); Shared/Multiuser Mix - многопользовательская среда; Computer Server Mix - среда центрального сервера для большого объема вычислений; File Server Mix - среда файлового сервера; RBMS Mix -среда обработки транзакций реляционной базы данных.

***1.5 МЕТОДИКА ОЦЕНКИ СКОРОСТИ ОБРАБОТКИ ТРАНЗАКЦИЙ***

Коммерческие приложения требуют эффективной работы с внешней памятью в распределенной сети при обработке транзакций.

До недавнего времени все производители рабочих станций и разработчики систем управления базами данных (СУБД) предлагали свои собственные способы оценки. В 1988 г. пять ведущих фирм, среди которых были IBM, Control Data и Hewllett-Packard, организовали Совет по проведению оценки скорости выполнения транзакций ТРС (Transaction Processing Performance Council), положивший конец "войне транзакций" и установивший единые правила измерения и оформления отчетов по их результатам. Методики тестирования ТРС основаны на том, что эффективность систем, предназначенных для решения задач оперативной аналитической обработки данных - OLTP (On-line Transaction Processing), в том числе для работы с базами данных, характеризуется числом транзакций, выполняемых в единицу времени.

Любая компания и фирма может стать членом ТРС, а результаты тестовых испытаний общедоступны на WWW-сервере www.ideas.com.au/bench/spec/spec.httn.

Понятие "транзакция" традиционно связывается с реляционными базами данных, однако применительно к OLTP имеет более общий смысл. Под *транзакцией* понимается последовательность операций ввода-вывода, во время проведения которых база данных остается неизменной. Практически транзакция представляет собой атомарную неделимую операцию, все изменения в результате выполнения которой становятся видны сразу после ее выполнения или отсутствуют до тех пор, пока операция не завершится.

В настоящее время из комплекса ТРС приняты в качестве общепризнанного стандарта три оценки (А, В и С).

Оценка ТРС-А характеризует быстродействие выполнения транзакций в режиме *on-line* для банковского кассира. При выполнении данного теста специально эмулируется операционная обстановка банка (терминалы и линии коммуникации), а в качестве транзакции выбирается обычная операция по обновлению счета клиента. Скорость работы в локальном окружении (без передачи транзакции во внешнюю сеть) измеряется в tsp-A-local. Быстродействие яри работе с внешними межбанковскими сетями оценивается в tsp-A-wide. В отчет о проведенном испытании по данной методике входит стоимость компьютера вместе с необходимым программным обеспечением и дополнительным оборудованием, необходимым для обеспечения работы банка в течение 90 дней. Стоимость вычислительной системы включает также пятилетнее сопровождение. При делении общей стоимости комплекса на полученное значение tsp получают цену одной транзакции (типа wide или local).

Оценка ТРС-В представляет собой усеченный вариант ТРС-А (без эмуляции терминалов и линий связи), ориентированный на проверку возможностей только СУБД в условиях ее интенсивной эксплуатации, Единицами измерения являются tsp-В и стоимость одной транзакции.

Тест ТРС-С появился из проекта корпорации МСТ (Microelectronics and Computer Technology). Программа проверки включает моделирование различных видов деловой деятельности (операции со счетами в банке, инвентаризация и т.п.). Размер транзакций в ТРС-С изменяется от очень простых и коротких до очень сложных и длинных операций, которые, как в реальной практике бизнеса, требуют сложных проводок и много ступенчатых пересылок. Единицами измерения являются tmp -число транзакций в минуту и стоимость одной транзакции.

Показатели по оценке ТРС могут зависеть не только от возможностей аппаратуры, но и от используемой базы данных (БД). Обычно применяются три СУБД: Oracle, Informix и Sybase.

Комитетом ТРС объявлены также тесты TPC-D и ТРС-Е. Тест TPC-D ориентирован на системы принятия решений DSS (Decision Support System). Эти системы характеризуются работой с более сложными запросами, возможностью моделирования хода выполнения транзакций для анализа возникающих ситуаций и т.д. В нем используются 17 аналитических запросов, характерных для расчета цен и скидок, общего анализа и прогнозирования рынка и управления поставками. Тест ТРС-Е также служит для оценки пригодности вычислительных систем для задач DSS.

Тест ТРС-А стал базовым для создания всей серии ТРС, но он не мог охватить всего многообразия требований приложений OLTP. Поэтому в 1995 г. он был изъят из употребления. ТРС-В также утратил актуальность в том же году. В связи с появлением эталонных тестов ТСН-Н и TPC-R тестТРС-Д был изъят из применения в 1999 г.

***1.6 МЕТОДИКА ОЦЕНКИ ГРАФИЧЕСКИХ ВОЗМОЖНОСТЕЙ***

Приведенные выше методики предназначены для тестирования наиболее распространенных типовых вычислительных систем и приложений. Однако массовое внедрение различного рода графических приложений (САПР, геоинформационные системы, мультимедиа и виртуальная реальность, архитектурное проектирование) потребовало разработки своих, специфических методик оценки.

Для оценок графических систем в настоящее время доступны несколько тестов, разработанных комитетом Graphics Performance Characterization (GPC), функционирующим под управлением Национальной графической компьютерной ассоциации (NCGA - National Computer Graphics Association), которая, в свою очередь, взаимодействует со SPEC. Комитет GPC предложил три системы тестов, на основе которых производится тестирование графических систем. Первой тестовой системой является Picture-Level Benchmark (PLВ), фактически измеряющая скорость визуализации. Результаты тестирования, доступные на сервере //sunsite.ync.edu/gpc/gpc.html или www.ideas.com.au/bench/ gpc, приводятся для стандартной (PLBHt) и оптимизированной (PLBopt) конфигурации.

Кроме теста PLB комитет GPC публикует результаты измерений по методике Xmark93, используемой для оценки эффективности работы Х-сервера. Следует отметить, что фирмами-разработчиками чаще всего используется тест Xmark93, позволяющий оценивать не только аппаратуру, но и эффективность реализации Х-сервера и степень его оптимизации под конкретное графическое оборудование. Результаты измерений на основе данного теста обычно доступны на WWW-серверах фирм-производителей.

Далеко не полный список различных систем тестирования состоит из более чем 40 названий и включает такие тесты, как Ханойские пирамидки, EureBen, SYSmark, CPUmark32 (тест, специально разработанный для оценки систем на базе процессора Intel), Приведенные методики и системы тестирования являются наиболее распространенными и, что самое главное, признанными большинством фирм-производителей.

***1.7 МЕТОДИКА ОЦЕНКИ ПРОИЗВОДИТЕЛЬНОСТИ СУПЕРКОМПЬЮТЕРОВ***

На рынке компьютерных технологий сейчас широко представлены и активно продолжают разрабатываться различные суперпроизводительные микропроцессоры: Alpha, MIPS, PowerPC/POWER2, HP72OO/8OOO, Pentium Pro, превосходящие на различных тестах многие вычислительные системы, построенные на процессорах предыдущего поколения. Однако, несмотря на впечатляющие успехи микропроцессорных технологий, разработчики уделяют большое внимание SMP, МРР и кластерным архитектурным решениям. Решение наиболее "емких" вычислительных задач, например численного аэродинамического моделирования, связывается именно с суперкомпьютерными архитектурами, обеспечивающими максимальную степень параллелизма. Так, например, NASA Armes Research Center определяет необходимость тысячекратного увеличения требуемых вычислительных мощностей.

Кроме крупнейших исследовательских центров, таких, как Cornell University, NASA, Air Force High Performance Computing Center, системы массового параллелизма используются для анализа и прогнозирования в бизнесе, что имеет целый ряд особенностей, связанных с вычислительными методами, ОС, мониторами параллельной обработки транзакций, библиотеками параллельных вычислений и т.п. Ведущие производители поставляют на рынок коммерческих приложений вычислительные системы IBM SP2, SNI RM1000, CRAY T916 Intel/Paragon и др.

Широко используемые системы Benchmark SPEC, TPC и LINPACK, применяемые для традиционных архитектур, неприемлемы для МРР- архитектур. Например, тесты SPEC дают возможность определить лишь производительность самих процессоров, тесты ТРС и LINPACK хотя и учитывают текущую конфигурацию вычислительной системы в целом и пригодны для оценки задач OLTP и DSS, все же не достаточны для многопроцессорных архитектур, К тому же объемы используемых в этих тестах данных (даже для теста LINPACK TPP - матрица размером 1000x1000) не позволяют полностью загрузить вычислительные ресурсы для получения реальных оценок. Для решения этой задачи специалистами из исследовательского центра NASA Ames Research Center были сформулированы основополагающие требования, которым должны удовлетворять тестовые методики оценки производительности суперкомпьютерных многопроцессорных систем, особенно МРР;

• системы с массовым параллелизмом часто требуют новых алгоритмических и программных решений, а их конкретные реализации могут существенно зависеть от архитектуры компьютера и, как следствие, отличаться друг от друга;

• тестовые смеси должны носить общий характер и не следовать какой-либо конкретной архитектуре, что исключает использование архитектурно-зависимого кода, например message passing code;

• корректность результатов должна быть легко проверяема, т.е. должны быть точно описаны входные и выходные данные и природа вычислений;

• используемая память и вычислительные ресурсы должны быть масштабируемыми для повышения производительности;

• тесты и спецификации используемых тестов должны быть доступны и подтверждаться повторной реализацией.

Существует подход, удовлетворяющий этим требованиям, при котором выбор конкретных структур данных, алгоритмов распределения процессоров и выделения памяти оставляется на усмотрение разработчика *и* решается в конкретной реализации тестов. Но система тестирования должна соответствовать некоторым правилам:

• все операции с плавающей точкой должны быть выполнены с использованием 64-разрядной арифметики;

• все тесты должны быть запрограммированы на языках Фортран 90 и Си;

• не допускается смешение кодов этих языков;

• допускается использование компилятора High Performance Fortran (HPF) версии от января 1992 г, или более поздней;

• все используемые расширения языка и библиотеки должны официально поставляться фирмой-производителем;

• библиотечные подпрограммы за исключением оговоренного списка должны быть написаны на одном из указанных языков.

Использование языков Си и Фортран обусловлено их распространенностью для подобного класса вычислительных систем. При этом важен запрет на использование ассемблерного кода, для того чтобы уравнять разрабатываемые тесты программ.

**Тест NAS**. Целью программы NAS, в рамках которой был разработан тест, было достижение к 2000 г. возможности проведения за несколько часов полномасштабного численного моделирования полета космического аппарата. Возможно, первой компьютерной системой, способной справиться с этой задачей, будет архитектура МРР.

Комплекс тестов NAS состоит из пяти тестов NAS Benchmarks Kernel и трех тестов, основанных на реальных задачах гидро- и аэродинамического моделирования. Этот круг задач не покрывает всего спектра возможных приложений, однако на сегодняшний день этот комплекс тестов является лучшим и общепризнанным для оценки параллельных многопроцессорных систем.

Как наиболее перспективные для определения производительности систем МРР выделяются именно последние три теста. Все требования к тестам описаны исключительно на уровне общего алгоритма, что позволяет производителям компьютеров выбрать наиболее приемлемые с их точки зрения методы решения задачи, структуры данных, дисциплину распределения заданий между процессорами и т.п.

Тесты NAS призваны в первую очередь оценить вычислительные возможности компьютерной системы и скорость передачи данных между процессорами в параллельных системах, а производительность при выполнении операций ввода-вывода или различных пре- и постпроцессорных функций в данном тесте не оценивается.

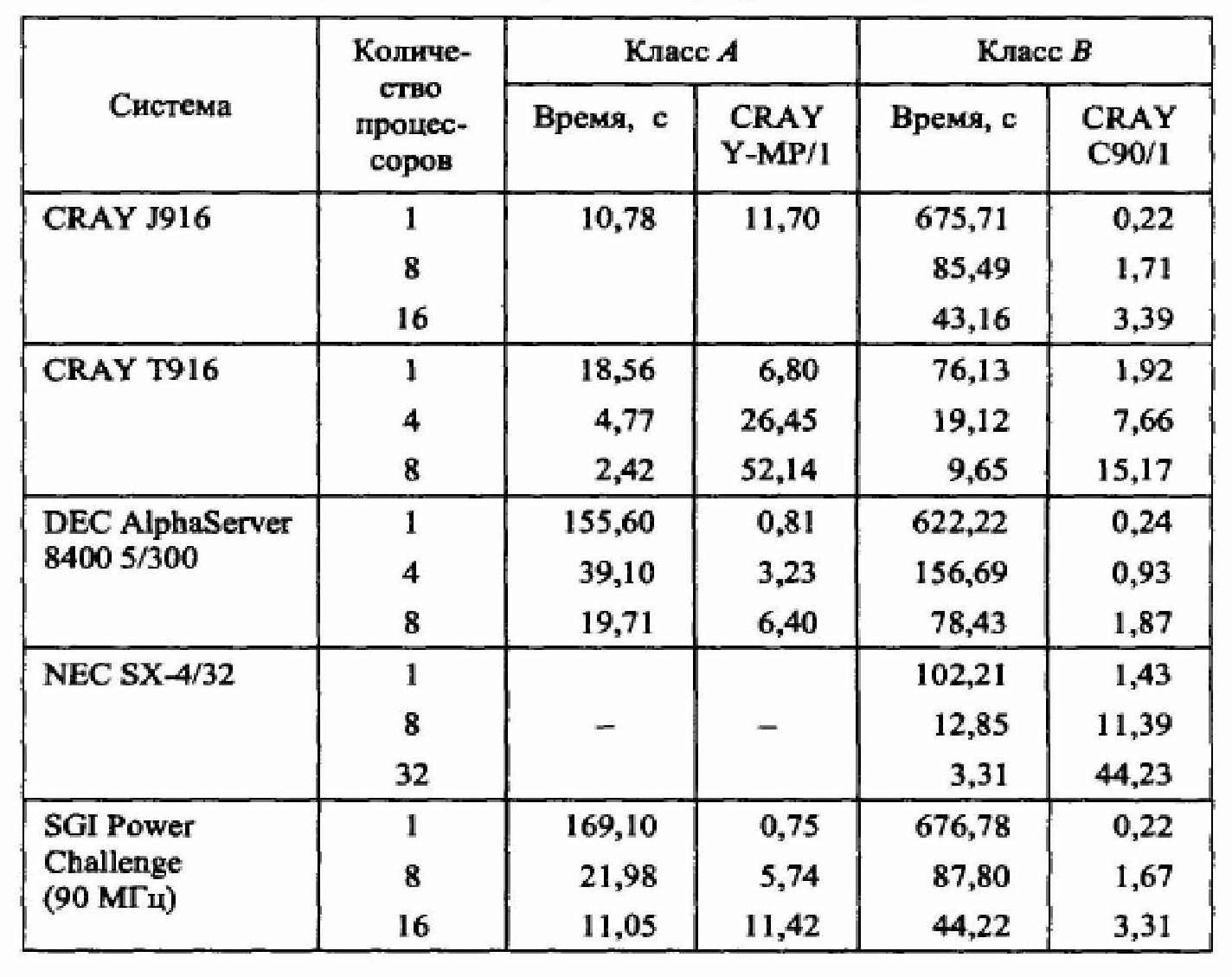
При выполнении каждого теста замеряется время в секундах, необходимое задаче, имеющей конкретный размер.

Для более наглядной оценки потенциальных возможностей тестируемой конфигурации вычисляется относительная производительность по сравнению с показателями традиционного векторного суперкомпьютера, в качестве которого обычно выступает одна из моделей Cray.

Для NAS Benchmarks Kernel определяются два класса тестов: класс *А* и класс *В,* которые фактически отличаются размерностью вычислений. Размер задач из класса *В* превосходит размер задач из класса *А* примерно в четыре раза. Результаты тестирования в классе *А* нормируются на производительность однопроцессорного компьютера Cray Y-MP, а в классе *В* — на производительность однопроцессорного Cray C90. Тесты класса *А* адекватно отражают производительность масштабируемых систем с числом процессорных узлов менее 128. При оценке систем с количеством узлов до 512 следует использовать тесты класса *В.*

Результаты тестирования некоторых известных вычислительных систем приведены в табл. 3.4. Эти данные весьма точно подтверждаются списком используемых во всем мире суперкомпьютеров ТОР500.

**Таблица 3.4Результаты тестирования суперкомпьютеров**



Комплекс тестов NAS Benchmarks kernel включает следующие расчетные задачи:

1. ЕР (Embarrasinghly Parallel). Вычисление интеграла методом Монте-Карло - тест усложненного параллелизма для измерения первичной вычислительной производительности плавающей арифметики. Этот тест минимального межпроцессорного взаимодействия фактически определяет чисто вычислительные характеристики узла при работе с вещественной арифметикой.

2. MG (3D Multigrid). Тест по решению уравнения Пуассона (трехмерная решетка) в частных производных требует высокоструктурированной организации взаимодействия процессоров, тестирует возможности системы выполнять как дальние, так и короткие передачи данных.

3. CG (Conjugate Gradient). Вычисление наименьшего собственного значения больших разреженных матриц методом сопряженных градиентов. Это типичное неструктурированное вычисление на решетке, и поэтому тест применяется для оценки скорости передачи данных на длинные расстояния при отсутствии какой-либо регулярности.

4. FFT (Fast Fourier Transformation). Вычисление методом быстрого преобразования Фурье трехмерного уравнения в частных производных. Эта задача - важный тест для оценки эффективности взаимодействия по передаче данных между удаленными процессорами. При создании программы, реализующей данный тест, могут использоваться библиотечные модули преобразования Фурье различной размерности.

5. IS (Integer Sort). Тест выполняет сортировку целых чисел и используется как для оценки возможностей работы системы с целочисленной арифметикой (главным образом одного узла), так и для выявления потенциала компьютера по выполнению межпроцессорного взаимодействия.

Комплекс тестов NAS Benchmarks Kernel по модельным задачам включает следующие модули:

1. LU (LU Solver). Тест выполняет вычисления, связанные с определенным классом алгоритмов (JNS3D-LU по классификации центра NASA Armes), в которых: решается система уравнений с равномерно разреженной блочной треугольной матрицей 5x5.

2. SP (Scalar Pentadiagonal). Тест выполняет решение нескольких независимых систем скалярных уравнений - с использованием пентаднагональных матриц, в которых преобладают недиагональные члены.

3. ВТ (Block Tridiagonal). Решение серии независимых систем уравнений с использованием блочных трехдиагональных матриц *5x5* с преобладанием недиагональных элементов.

**Тест ЕР**. Чтобы понять принципы построения тестов типа NAS и особенности их реализации на конкретных суперкомпьютерных архитектурах, рассмотрим несколько подробнее тест ЕР. Данный тест формулируется следующим образом: формирование двухмерной статистики из большого числа случайно распределенных по Гауссу чисел, которые генерируются наилучшим (оптимальным) образом для каждой конкретной вычислительной архитектуры. Эта постановка является типичной для большинства приложений, использующих метод Монте-Карло. Как и все остальные тесты ядра NAS Kernel, этот тест имеет два класса, определяемых в данном случае числом сгенерированных и обработанных случайных чисел: первый - 228, второй - в четыре раза больше.

На многопроцессорной архитектуре каждый из процессоров независимо генерирует статистику для множества из *п/р* пар. Так как генерация статистик происходит параллельно на каждом процессоре, то фактически не требуется межпроцессорного взаимодействия. И только лишь десять пар от каждого процессора аккумулируются - пересылаются в один узел, чем, однако, можно пренебречь. Ключевым моментом для данного теста является только оптимизация вычислений на узле, что позволяет оценивать вычислительные возможности системы по работе с вещественными числами.

Для приоритетного определения коммуникационных показателей системы с массовым параллелизмом используются три оставшихся теста, в основе которых - алгоритмы численных методов на решетке. Основная идея реализации этих алгоритмов, в частности, для теста MG заключается в том, что на процессорах строится логическая модель трехмерной решетки. Это предполагает создание подобластей, в узлах которых параллельно проис ходят вычисления. Однако при этом важное значение имеет вычисление граничных условий для каждой подобласти, что требует интенсивного взаимодействия между процессорами. Кроме, того, не менее существенным моментом, влияющим на эффективность реализации теста, является репликация данных при переходе вычислительного процесса на новые слои решетки, что также определяется эффективностью реализации межпроцессорного взаимодействия.

Результаты последних оценок суперкомпьютерных платформ можно найти на WWW-сервере NAS www.nas.nasa.gov/NAS/NPB. Анализ этих данных показывает, что даже самая быстродействующая система VPP500 по соотношению цена/производительность уступает или сравнима с намного более дешевым сервером DEC 8400, суперкомпьютером SGI Power Challenge или RS/6000 SP.

***1.8 МЕТОДИКА ОЦЕНКИ КОНФИГУРАЦИЙ WEB***

Тестовая методика оценки конфигураций Web - WebSTONE - представляет собой одно из первых средств оценки эффективности оборудования и программного обеспечения при работе с протоколом HTTP.

По своему функциональному назначению WWW во многом напоминает NFS, для оценки эффективности которой существует тест LADDIS. Но адаптация этого теста к конкретной архитектуре часто оказывается весьма проблематичной. Тест WebSTONE более точно отражает специфику работы с глобальными сетями с многократными переключениями, исправлением ошибок, переадресациями и т.п. Данный тест способен моделировать разнородную среду, в которой работают одновременно множество клиентов, порождающих разнообразных потомков, способных запрашивать информацию от серверов.

Предусмотрены четыре смеси, моделирующие различные конфигурации подключения Web-сервера. Первая смесь - общий набор, использующийся для моделирования подключения к сети через модем. Файлы данной смеси содержат небольшие, обычно текстовые страницы размером не более 20 Кбайт, для передачи которых по модему со скоростью 14,4 Кбит/с требуется не очень много времени. Вторая смесь применяется для моделирования работы клиентов локальной сети. Размер файлов в данной смеси колеблется от 1 до 100 Кбайт. Третий тестовый набор представляет собой массив информации мультимедиа, использующий для своей обработки все ресурсы тестируемой системы. Файлы этого набора - MPEG и Quicklime, звуковые клипы и большие графические файлы. Размер таких файлов изменяется от 20 Кбайт до нескольких мегабайт. В четвертой смеси объединены первый и третий наборы.

Главными показателями WebSTONE являются *пропускная способность,* измеряемая в байтах в секунду, и *латентность -* время, необходимое для выполнения Запроса. Кроме того, WebSTONE содержит информацию о количестве страниц в минуту, среднем числе соединений и другую информацию, позволяющую провести более точную оценку качества конфигурации и выявить ее узкие места. Пропускная способность измеряется как для всей системы в целом, так и по каждому клиенту в отдельности; в обоих случаях она усредняется за все время проведения тестирования. Различают два типа латентности: время соединения и время запроса. Первое показывает продолжительность установки соединения, а второе - временные затраты на непосредственную передачу данных.

В WebSTONE включена также оценка по закону Литтла *(Little Law),* показывающая, сколько времени затрачивается сервером на выполнение полезной работы по обработке запроса, а не на действия типа коррекции ошибок или вспомогательные операции. В идеальном случае этот показатель прямо пропорционален числу клиентских процессов. Если сервер Web оказывается загружен сверх нормы, происходит массированный выброс ошибочных сообщений, свидетельствующих о том, что запросы клиента не могут быть обработаны в течение отпущенного им временного интервала. Фактически показатель дает представление об уровне максимальной загрузки для конкретного сервера.

WebSTONE достаточно просто настраивается для получения оценок следующих параметров работы сервера: среднее и максимальное время соединения; среднее и максимальное время отклика; пропускная способность; количество обработанных страниц; число открытых файлов.

Основная задача теста - оценить скорость и безошибочность обслуживания установленного множества клиентов. Программы WebSTONE организуют передачу запросов серверу по протоколу HTTP и обрабатывают данные по мере их поступления. Поскольку тест ориентирован на оценку работы программного и аппаратного обеспечения сервера, производительность броузеров или приложений клиента им не учитывается.

Архитектура теста WebSTONE включает две локальные сети. В первой сети работают потомки (Webchildren), управляемые программой WebMASTER, отвечающей также за управление всем ходом тестирования. Сама программа WebMASTER размещается на отдельном сегменте независимо от потомков и может функционировать как на одном компьютере вместе с клиентом, так н на отдельной машине. При определении режима функционирования теста может задаваться произвольная конфигурация сетей. Однако размещение WebMASTER на отдельном компьютере позволяет получить гибкость при моделировании всевозможных конфигураций подключения клиентов.

WebSTONE является распределенным тестом, работающим со многими процессами, когда главный процесс или WebMASTER и считывает файл конфигурации клиента, и выполняет необходимые функции. Затем WebMASTER формирует командные последовательности для каждого потомка и инициирует их. Каждый потомок, в свою очередь, читает команду, устанавливает связь с WebMASTER и выполняет последовательность действий, предусмотренных в тестовом наборе. После завершения работы каждого потомка вызывается WebMASTER с функцией сбора данных от каждого клиента н формирования отчета. В процессе работы каждый потомок является независимым как от других потомков, так и от WebMASTER.

Одна из главных особенностей теста WebSTONE — его гибкость, позволяющая моделировать произвольные конфигурации и имитировать работу с различными узлами. При выполнении стандартной тестовой смеси можно получить показатель общей производительности конфигурации. В качестве параметров настройки используются продолжительность выполнения теста, число повторений, количество тестовых файлов, число страниц, опции программного и аппаратного обеспечения сервера, количество потомков, количество сетей, число клиентов, загрузка страниц, ведение журнала, отладка.

Продолжительность выполнения теста задается в минутах. Максимальное время определяется числом потомков и емкостью памяти, выделяемой для каждого клиента. Многократное повторение теста позволяет устранить элемент случайности, неизбежный при работе в сети, и выявить устойчивые закономерности.

Количество страниц, представляющих собой документ в формате HTML (текст, картинки в форматах GIF или JPEG), задается при формировании тестируемой конфигурации, максимально приближенной к реальной. Обычно страницы, подобранные в тестовом файле, являются типичными страницами Web, используемыми на наиболее распространенных узлах WWW.

При описании операционного окружения для работы теста необходимо задать конфигурацию программ и аппаратуры, число потомков Webchildren, страниц или файлов, загружаемых с сервера. Это позволяет имитировать различные реальные сетевые комплексы. Варьируя параметры и анализируя результаты тестирования, можно выявить оптимальные параметры для решения конкретной задачи. Для имитации нескольких сетей, управляемых одним сервером, достаточно задать параметр "количество сетей", не заботясь об именовании серверных узлов и организации потоков клиентов. Задание числа клиентов н их потомков позволяет имитировать различные режимы использования ресурсов, необходимых для работы каждого клиента и собственно процесса WebMASTER. При варьировании значений данного параметра можно моделировать клинчевые ситуации, когда много потомков одновременно запрашивают один и тот же ресурс. Каждая страница в тестовом наборе WebSTONE обладает весом, задающим активность использования страницы - чем выше вес, тем чаще будет осуществляться обращение к данной странице. Иногда для более подробного анализа работы клиента требуется точный протокол его работы, который не включается в результирующий отчет, формируемый процессом WebMASTER, а используется отдельно.

Тестируемая конфигурация должна состоять не менее чем из двух компьютеров, соединенных сетью. Первый является сервером Web, в качестве которого может выступать любой программный сервер, поддерживающий протокол HTTP 1.0.

Необходимо иметь WebMASTER, а также на этой или другой машине один или несколько клиентов Web - обычно это Unix-узлы. Например, на одной станции Indy с 32 Мбайт памяти могут нормально работать до 120 клиентов вместе с WebMASTER. Далее необходимо определить порядок взаимодействия между WebMASTER и клиентами, каждый из которых должен быть сконфигурирован так, чтобы WebMASTER мог перезапускать на них программу WebSTONE. Также полезно, чтобы суперпользователь имел возможность управлять работой клиентов и Web-сервера.

Основные понятия теста включают следующие определения.

*Клиенты -* один или несколько процессов, работающих с сервером. Увеличивая число одновременно работающих клиентов, можно выявить максимально допустимую для каждого конкретного сервера конфигурацию.

*Число соединений в секунду* - число успешных соединений TCP/IP, выполненных сервером за одну секунду при работе со всеми клиентами. Кроме собственно установления связи в процедуру соединения входят передача подтверждающего сообщения, получение ответа н закрытие ТСР/IР-соединения. Чем больше количество соединений, тем выше производительность конфигурации. Данный показатель определяет, с какой скоростью сервер способен отвечать на новые запросы, одновременно завершая текущие.

*Число ошибок в секунду* - количество сбоев, произошедших в процессе взаимодействия сервера с клиентом, например ошибка типа Connection Refused, возникающая при попытке установить соединение TCP/IP. Чем меньше значение данной характеристики, тем лучше и эффективнее работает конфигурация.

*Латентность -* среднее время, затрачиваемое на соединение клиента и сервера, а также на обработку запроса. Чем меньше латентность, тем лучше.

*Закон Литтла (Little-s Law)* - отношение времени, затрачиваемого на посылку сообщения, к времени ожидания ответа. Чем ближе значение этого показателя к числу клиентов, обслуживаемых данным сервером, тем лучше его производительность. Термин взят из теории очередей.

*Пропускная способность* - суммарное количество мегабит в секунду, проходящих через всех клиентов. Чем выше пропускная способность, тем лучше производительность.

В перспективе возможности WebSTONE будут расширены средствами поддержки proxy-серверов, стоимостными оценками, например затратами на эксплуатацию и модернизацию Web-сервера, оценками организации работы с транзакциями, активно использующими двоичные сценарии CG (Common Gate Interface), оценками эффективности построения защиты и производительности работы с распределенными базами данных.

Одновременно с началом использования WebSTONE рабочая группа SPEC SFS была расширена командой SGI WebSTONE, основная цель которой - доведение данного теста до уровня стандарта и выпуск отчетов SPEC Webperf. Кроме WebSTONE для проведения общего анализа эффективности работы Web можно также использовать программы WebStat - сценарий анализа трафика, WebTap - анализатор работы сервера при взаимодействии с приложениями Java, WebTrac - анализатор Web. Эти программы позволяют получить статистические оценки функционирования выбранной конфигурации.

***Глава 2. Описание модуля электронного учебника по теме "Методы организации сложных экспертиз компьютерных систем"***

***2.1*** ***ПОНЯТИЕ И ТРЕБОВАНИЯ К ЭЛЕКТРОННОМУ УЧЕБНИКУ***

Большие трудности часто возникают при оперативной подготовке, изготовлении и распространении учебных пособий различных видов. Данные факторы негативно сказываются на качестве подготовки обучаемых. В связи с этим большое внимание уделяется применению прогрессивных методик обучения, в том числе предполагающих использование вычислительной техники. Это позволяет существенно повысить качество и эффективность учебного процесса. Одной из форм повышения эффективности обучения являются электронные учебники.

В настоящее время существует множество определений электронного учебника, вот некоторые из них:

- это компьютерное, педагогическое программное средство, предназначенное, в первую очередь, для предъявления новой информации, дополняющей печатные издания, служащее для индивидуального и индивидуализированного обучения и позволяющее в ограниченной мере тестировать полученные знания и умения обучаемого.

- это электронный учебный курс, содержащий систематическое изложение учебной дисциплины или ее раздела, части, соответствующий государственному стандарту и учебной программе и официально утвержденный в качестве данного вида издания.

- это комплекс информационных, методических и программных средств, который предназначен для изучения отдельного предмета и обычно включает вопросы и задачи для самоконтроля и проверки знаний, а также обеспечивает обратную связь.

- основное учебное электронное издание, созданное на высоком научном и методическом уровне, полностью соответствующее федеральной составляющей дисциплины Государственного образовательного стандарта специальностей и направлений, определяемой дидактическими единицами стандарта и программой.

Электронные учебники позволяют решать такие основные педагогические задачи, как:

- начальное ознакомление с предметом, освоение его базовых понятий и конструкций;

- базовая подготовка на разных уровнях глубины и детальности;

- контроль и оценивание знаний и умений;

- развитие способностей к определенным видам деятельности;

- восстановление знаний и умений.

Электронные учебники могут быть использованы на всех уровнях образования: в школах и колледжах, институтах и университетах, для повышения квалификации. Поэтому электронные учебники разрабатываются во многих странах.

**Основные требования** к методике составления электронного учебника следующие:

1. Учебный материал должен быть разбит на блоки.
2. Каждый блок должен быть снабжен иллюстрациями.
3. Иллюстрации должны подбираться таким образом, чтобы более подробно и просто разъяснить трудновоспринимаемый студентами материал.
4. Основной материал блока должен быть объединен в одно целое с помощью гиперссылок. Гиперссылки могут связывать собой и отдельные блоки электронного учебника.
5. Целесообразно, дополнить материал электронного учебника всплывающими подсказками.

При проектировании электронного учебника необходимо заложить в него технологические характеристики, позволяющие впоследствии сделать учебно-воспитательный процесс максимально эффективным. Выступая в качестве автоматизированной обучающей системы, электронный учебник должен выполнять следующие функции:

* эффективно управлять деятельностью обучаемого по изучению учебной дисциплины;
* стимулировать учебно-познавательную деятельность;
* обеспечивать рациональное сочетание различных видов учебно-познавательной деятельности с учетом дидактических особенностей каждой из них и в зависимости от результатов освоения учебного материала;
* рационально сочетать различные технологии представления материала (текст, графику, аудио, видео, анимацию);
* при размещении в сети обеспечивать организацию виртуальных семинаров, дискуссий, деловых игр и других занятий на основе коммуникационных технологий.

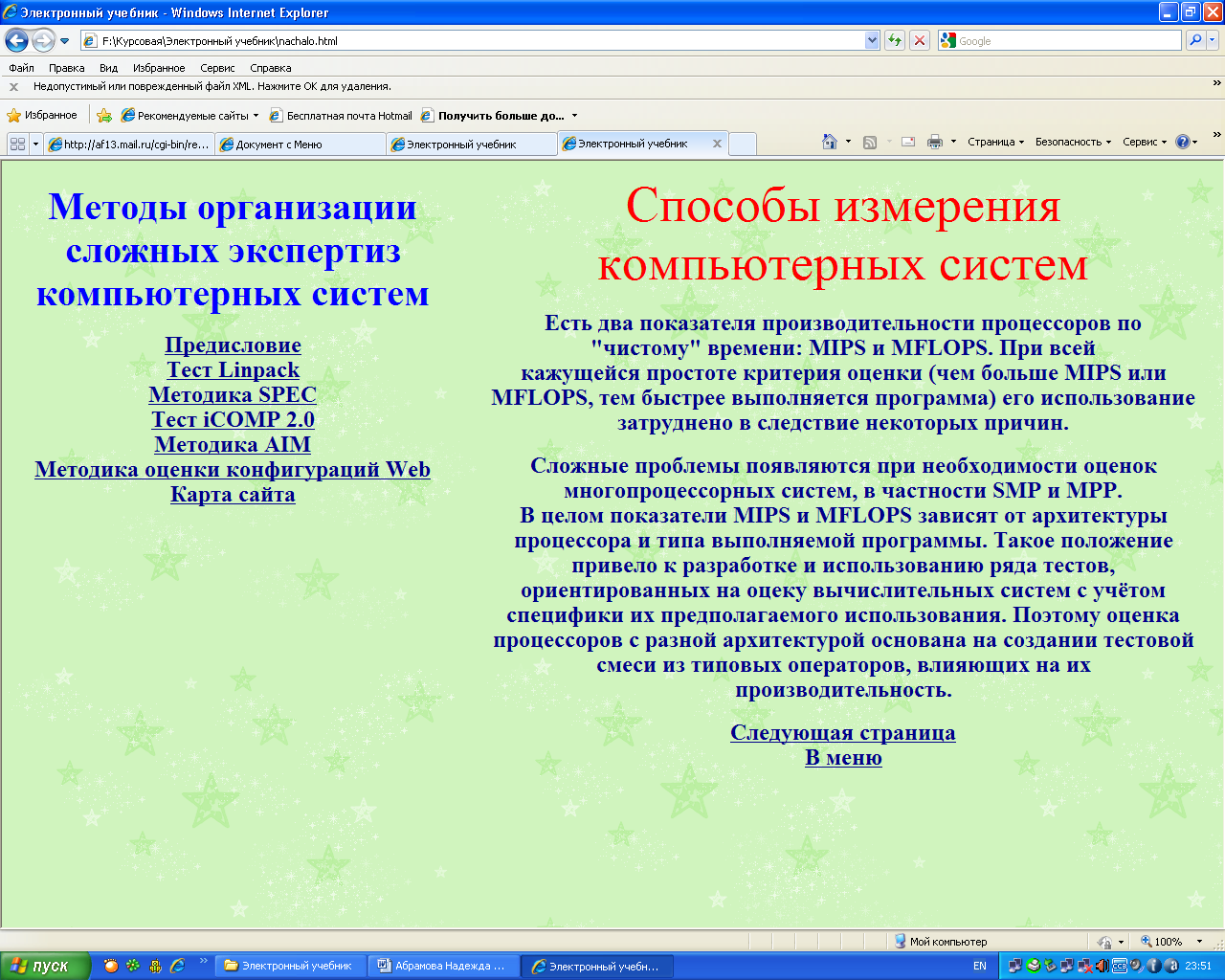
В отличие от классического варианта учебника электронный учебник предназначен для иного стиля обучения, в котором нет ориентации на последовательное, линейное изучение материала.

Одним из достоинств электронных средств обучения и, в частности, электронных учебников, относят индивидуальный темп обучения.

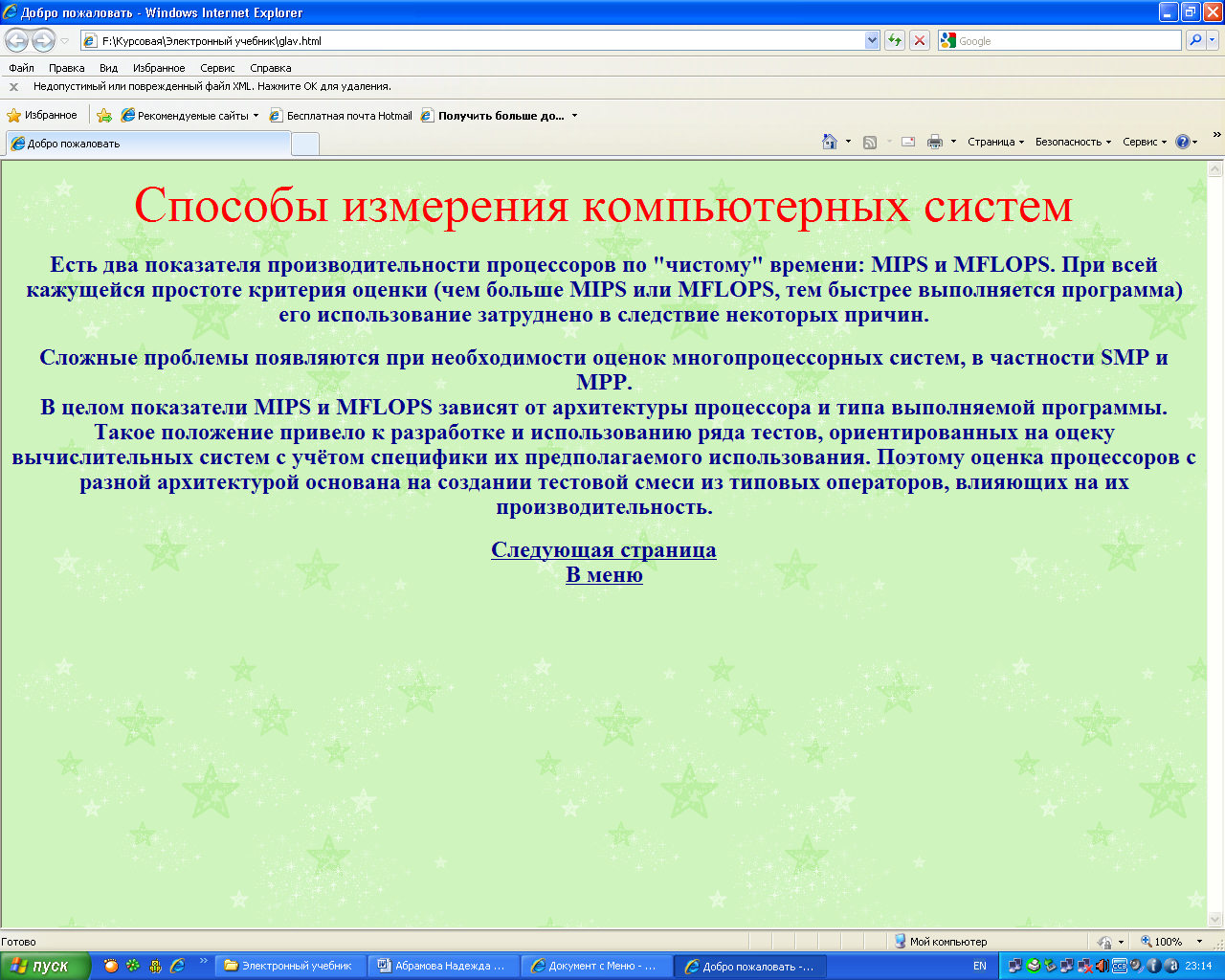
К недостаткам электронного учебника можно отнести не совсем хорошую физиологичность дисплея как средства восприятия информации (восприятие с экрана текстовой информации гораздо менее удобно и эффективно, чем чтение книги) и более высокую стоимость по сравнению с книгой.

***2.2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНИКА***

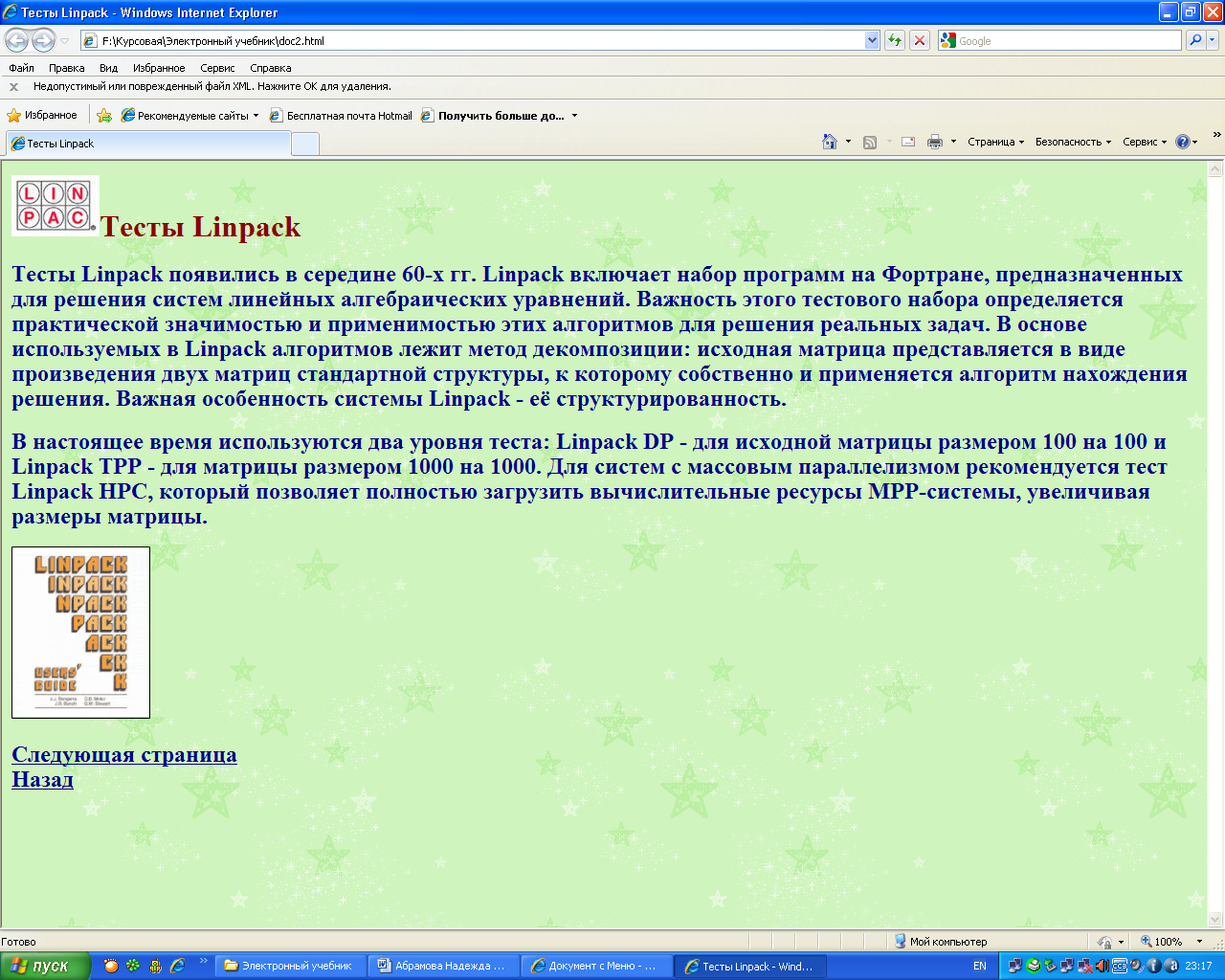
Основное содержание электронного учебника разбито на отдельные блоки или модули. Блок аналогичен главе в обычном учебнике. Переход из одного блока или модуля возможен при помощи гиперссылок. Учебник содержит рисунки в виде логотипов тестов и методик оценки компьютерных систем и результатов работы некоторых из них. В содержание входит краткая характеристика методов оценивания компьютерных систем.



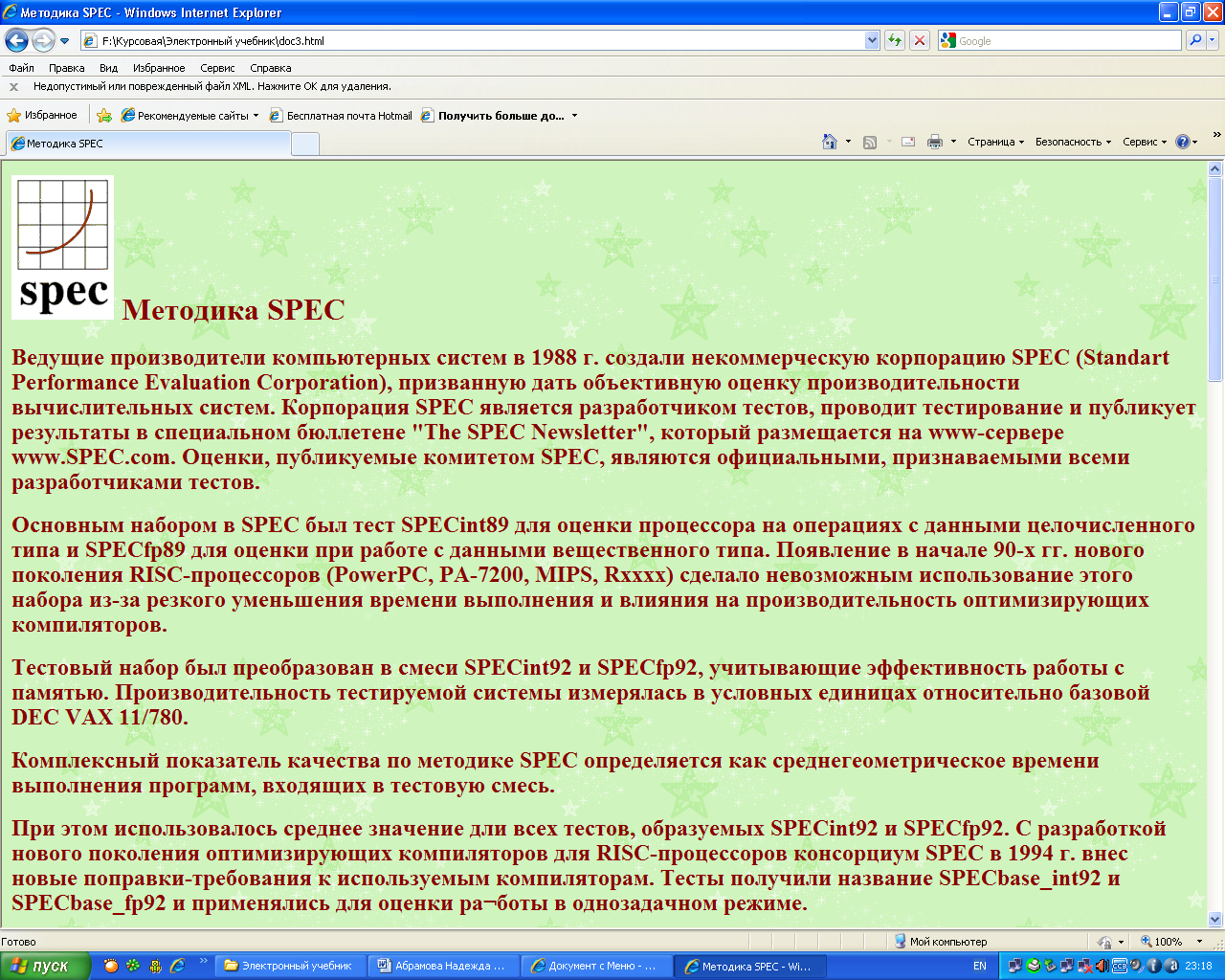
**Рис 1. Начальная страница**



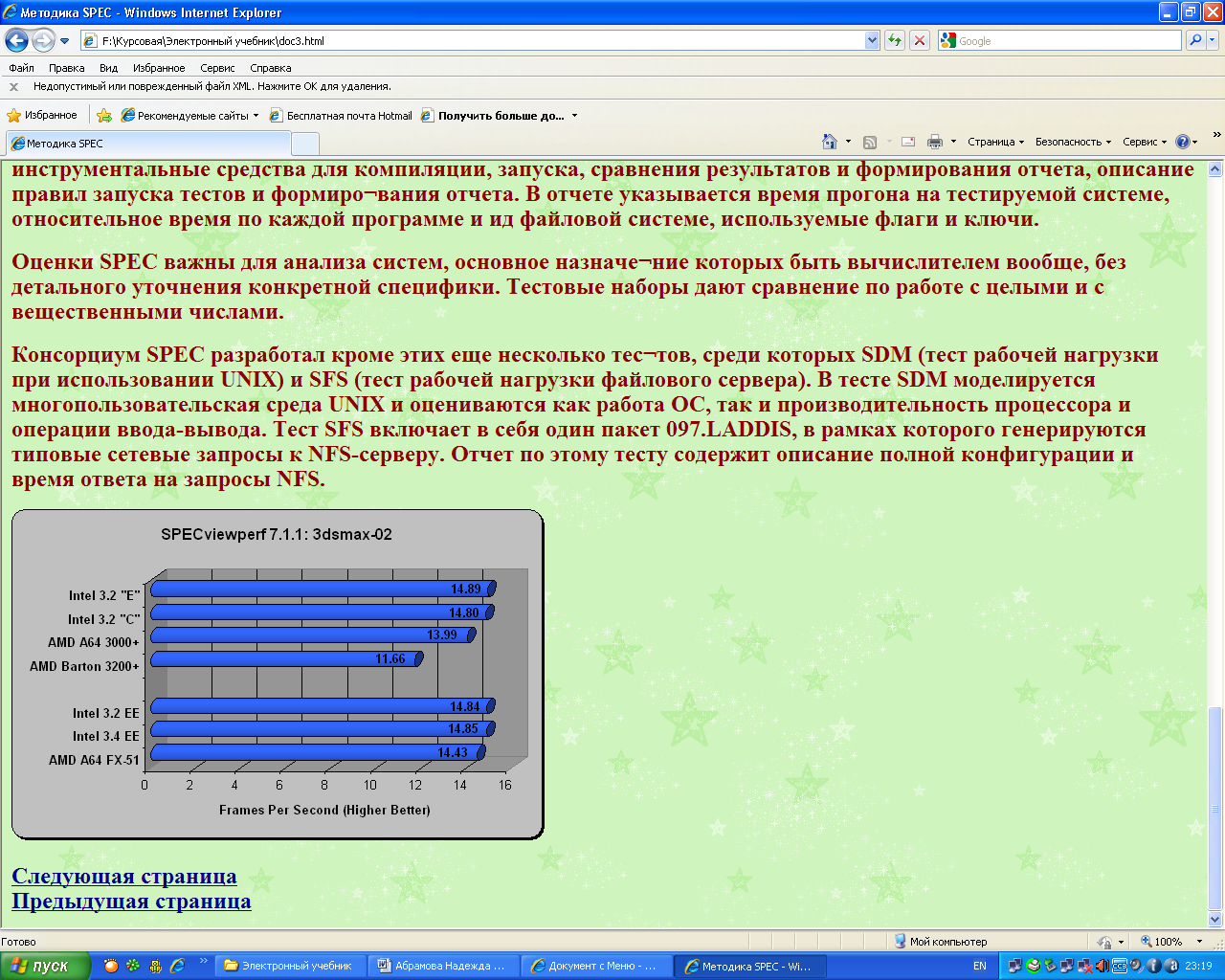
**Рис 2. Предисловие.**



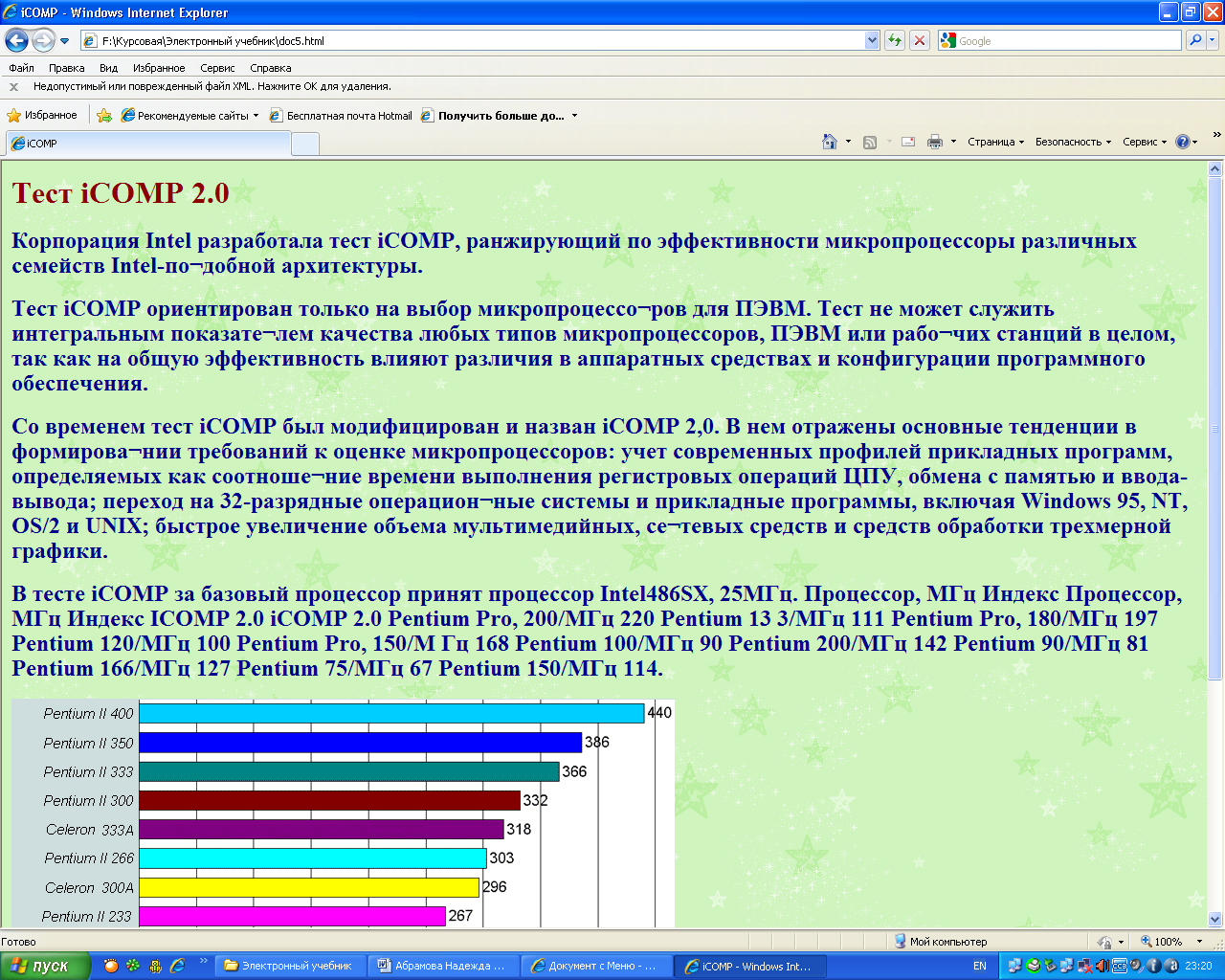
**Рис 3. Тесты LINPACK.**



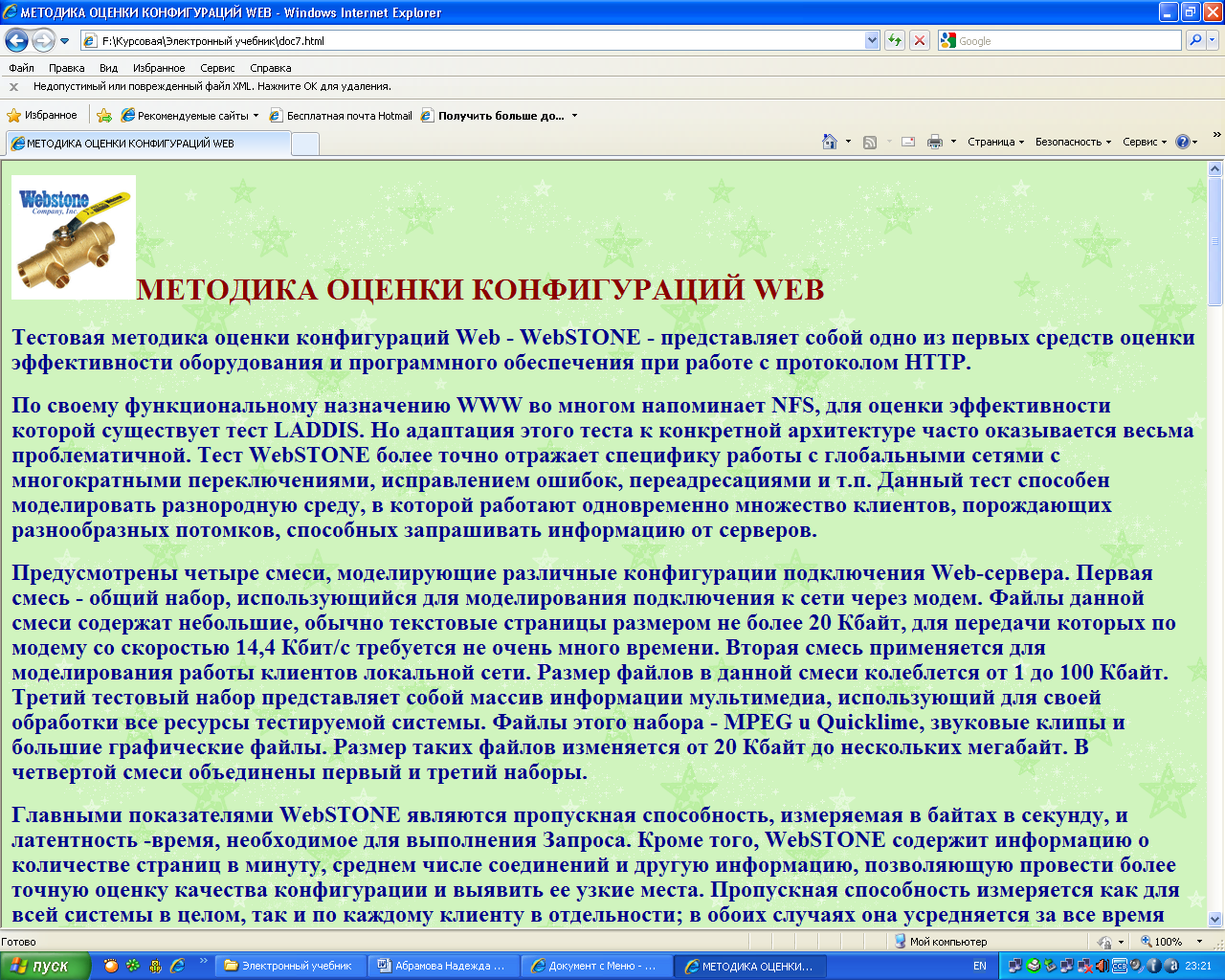
**Рис.4. Методика SPEC**



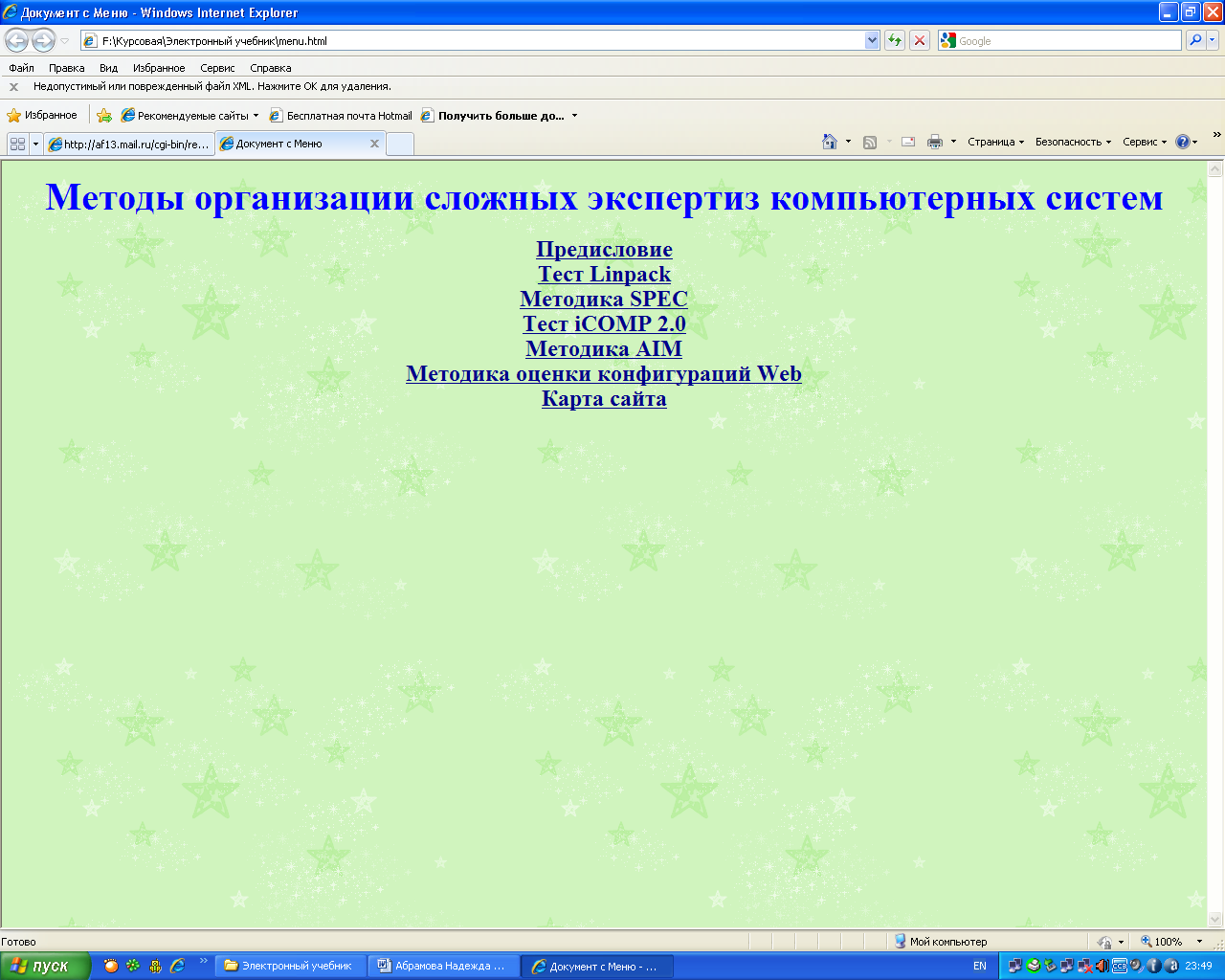
**Рис.5. Результат работы методики SPEC**



**Рис.6. Тест iCOMP**



**Рис.7. Методика оценки конфигураций WEB**



**Рис.8. Меню**

***Заключение***

В курсовой работе были рассмотрены методы организации сложных экспертиз компьютерных систем, в частности тесты DHRYSTONE, LINPACK, "ЛИВЕРМОРСКИЕ ЦИКЛЫ", методика SPEC, тест ICOMP 2.0, методика AIM, методика оценки скоростей обработки транзакций, методика оценки графических возможностей, методика оценки производительности суперкомпьютеров и методика оценки конфигураций WEB.

Результатом курсовой работы является разработка модуля электронного учебника по теме "Методы организации сложных экспертиз компьютерных систем". Практическая значимость исследования заключается в том, что разработанный модуль электронного учебника по теме "Методы организации сложных экспертиз компьютерных систем" может быть использован для создания электронных учебников по дисциплинам информатики.

***Библиографический список***

1. *Анфилатов B.C.* Вычислительные системы. - СПб.: Изд-во ВУС, 1998. - 278 с

2. *Волкова В.Н., Денисов А.А.* Основы теории систем и системного анализа. - СПб: Изд-во СПбГТУ. - 510 с.

3. *Кукушкин А.А.* Теоретические основы автоматизированного управления. Ч. 1: Основы анализа и оценки сложных систем. - Орел: Изд-во ВИПС, 1998. - 254 с.

4. *Кукушкин А.А.* Теоретические основы автоматизированного управления. Ч. 2: Основы управления и построения автоматизированных информационных систем. - Орел: Изд-во ВИПС, 1999.-209 с.

5. *Французов Д., Волков Д.* Новое поколение тестов SPEC // Открытые системы. - 1996. - № 4 (18). - С. 58 - 64

***Приложение***

***Методика создания модуля электронного учебника***

1. **Начальная страница**

HTML-скрипт:

*<html>*

*<head>*

*<title>Электронный учебник</title>*

*<frameset cols="500,\*" border="0">*

*<frameset rows="10,\*">*

*<frame src="logo.html" scrolling="no"><frame src="menu.html">*

*</frameset>*

*<frame src="glav.html" name=" window-1">*

*</frameset>*

*</head>*

*</html>*

**Тег** *<html>* является контейнером, который заключает в себе все содержимое веб-страницы.

**Тег** *<head>* предназначен для хранения других элементов, цель которых — помочь браузеру в работе с данными.

**Тег** *<title>* устанавливает заголовок окна веб-страницы.

**Тег** <*frameset.* определяет структуру фреймов на веб-странице. Фреймы разделяют окно браузера на отдельные области, расположенные вплотную друг к другу. В каждую из таких областей загружается самостоятельная веб-страница определяемая с помощью тега <*frame>.* С помощью фреймов веб-страница делится на два документа, которые содержат навигацию по сайту. Механизм фреймов позволяет открывать документ в одном фрейме, по ссылке, нажатой в совершенно другом фрейме. Тег <*frameset>* заменяет собой элемент <*body>* на веб-странице. Требуется закрывающий тег <*frameset>.*

**Тег** <*cols>*задает размеры колонок фреймов в структуре <*framese>*. Символ звездочки (\*) обозначает всю свободную ширину, оставшуюся после указания размеров в пикселах.

**Тег** <*rows>* **з**адает высоту фреймов в структуре <*frameset>* над первой строкой. Символ звездочки (\*) обозначает всю свободную ширину, оставшуюся после указания размеров в пикселах.

**Тег** <*border>*устанавливает толщину границы между фреймами. В данном случае её нет, т.к. она равна 0.

**Тег** *<frame>* определяет свойства отдельного фрейма, на которые делится окно браузера.

**Параметр** "*src"* указывает адрес файла, который будет загружаться во фрейм.

**Параметр** "*scrolling*".Если содержимое фрейма не помещается в отведенное окно, автоматически появляются полосы прокрутки для просмотра информации. В некоторых случаях, полосы прокрутки нарушают дизайн веб-страницы, поэтому от них можно отказаться. **Аргумент** *"no"*запрещает отображение полос прокрутки.

**Параметр** "*name"*даёт возможность загружать документ в выбранный фрейм из другого. В качестве значения указывается имя фрейма (*"w"*), в который будет загружаться документ, заданный параметром *"name".*

**2. Предисловие**

HTML-скрипт:

*<html>*

*<head>*

*<title>Добро пожаловать</title>*

*</head>*

*<body background="chert.gif" text="#00008B" link="#00008B" alink="#00008B" vlink="#00008B">*

*<center><p style = "color:red; font-size: 40pt">Способы измерения компьютерных систем</p>*

*<p align="center">*

*<H2><center>*

*<p>Есть два показателя производительности процессоров по "чистому" времени … вследствие некоторых причин.</p>*

*<p>Сложные проблемы появляются при необходимости оценок … влияющих на их производительность.</p>*

*</H2>*

*<H2><A href=doc2.html>Следующая страница</A> <br> <A href=menu.html>В меню</A></H2>*

*</body>*

*</html>*

**Параметр *"****background****"*** задает фоновый рисунок на веб-странице.

**Параметр *"****text****"*** устанавливает цвет текста, используемого на веб-странице по умолчанию.

**Параметр *"****link"* устанавливает цвет непосещенных ссылок.

**Параметр "***alink"*устанавливает цвет активной ссылки.

**Параметр *"****vlink"* определяет цвет посещенных ссылок.

В данной работе последние 4 параметра имеют одинаковые значения, поэтому цвет ссылок не меняется.

**Тег** *<р>* определяет текстовый параграф.

**Параметр** *"style"* определяет стиль текста. С помощью него устанавливается цвет текста (*color:red*) и размер шрифта (*font-size: 40pt*).

**Параметр** *"align"* **и аргумент** *"cente"* выравнивают текст по центру.

**Тег** *<H2>…</H2>* определяет уровень заголовка, в данном случае – размер шрифта.

**Аргумент** *"center"* выравнивает текст по центру.

**Тег** *<А>…</A>* предназначен для создания ссылок.

**Параметр** *"href"* задает адрес документа, на который следует перейти.

**3. Главы**

Все главы составлены аналогично. Поэтому рассмотрим одну из них.

HTML-скрипт:

*<html>*

*<head>*

*<title>Тесты Linpack</title>*

*</head>*

*<body background="chert.gif" text="#00008B" link="#00008B" alink="#00008B" vlink="#00008B">*

*<p align="center">*

*<H1><font color="#8B0000"><img src="./pic/i.jpg">Тесты Linpack</font></H1></p> <H2>*

*<p>Тесты Linpack появились в середине 60-х гг …особенность системы Linpack - её структурированность. </p>*

*<p>В настоящее время … вычислительные ресурсы МРР-системы, увеличивая размеры матрицы.</p>*

*<img src="./pic/explorer2.gif">*

*</H2>*

*</p>*

*<H2>*

*<A href=doc3.html>Следующая страница</A> <br> <A href=menu.html>Назад</A></H2>*

*</body>*

*</html>*

**Тег** *"font color"* задает цвет шрифта заголовка.

**Тег***"img"****.*** Он предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Этот тег имеет единственный обязательный параметр *"src"*, который определяет адрес файла с картинкой.

**4. Карта сайта**

HTML-скрипт:

*<html> <head>*

*<title>Карта сайта</title>*

*<body background="chert.gif" text="#00008B" link="#00008B" alink="#00008B" vlink="#00008B">*

*<H2> <a href="glav.html" target="window-1">Главная</a><br>*

*<a href="doc2.html" target="window-1">Тест Linpack</a><br>*

*<a href="doc3.html" target="window-1">Методика SPEC</a><br>*

*<a href="doc5.html" target="window-1">Тест iCOMP 2.0</a><br>*

*<a href="doc6.html" target="window-1">Методика AIM</a><br>*

*<a href="doc7.html" target="window-1">Методика оценки конфигураций Web</a>*

*</H2> </body> </head>*

*</html>*

1. **logo.html**

HTML-скрипт:

*<html>*

*<head>*

*<title>Добро пожаловать</title>*

*</head>*

*<body background="chert.gif" >*

*</body>*

*</html>*

Этот файл содержит рисунок фона, который загружается на начальную страницу, а именно на высоту *"rows"* первого фрейма.

1. **Меню**

HTML-скрипт:

*<html>*

*<head>*

*<title>Документ с Меню</title>*

*<body background="chert.gif" text="#00008B" link="#00008B" alink="#00008B" vlink="#00008B">*

*<H2><center><p style = "color:blue; font-size: 30pt">Методы организации сложных экспертиз компьютерных систем</p>*

*<center><a href="glav.html" target="window-1">Предисловие</a><br>*

*<a href="doc2.html" target="window-1">Тест Linpack</a><br>*

*<a href="doc3.html" target="window-1">Методика SPEC</a><br>*

*<a href="doc5.html" target="window-1">Тест iCOMP 2.0</a><br>*

*<a href="doc6.html" target="window-1">Методика AIM</a><br>*

*<a href="doc7.html" target="window-1">Методика оценки конфигураций Web</a><br>*

*<a href="karta.html" target="window-1">Карта сайта</a></center>*

*</H2>*

*</body>*

*</head>*

*</html>*

**Параметр** *"target"* обозначает имя окна или фрейма, куда браузер будет загружать документ.