

студенты, остальные же выпадают из контроля. Студент готовит только свой вопрос, а остальные темы откладываются до подготовки к экзамену. При использовании промежуточного тестирования, входящего в ИОС преподаватель получает в распоряжение дополнительное время, которое могло быть израсходовано на очный промежуточный контроль полученных на каждом этапе обучения знаний. Вместо этого преподаватель может более подробно и обширно представить свой курс.

Не имея постоянного контакта с обучаемыми, преподаватель с помощью ИОС видит в реальном времени, как происходит обучение в целом для группы и для конкретного студента. Процесс обучения в глазах преподавателя становится «прозрачным», наглядным и легко управляемым. ИОС является нитью, связывающей преподавателя и студента. Она обеспечивает повышение эффективности работы преподавателей и, как следствие, качество получаемых студентами знаний повышается.

*Список литературы*

1. В.Аалст, К.Хей. Управление потоками работ: модели, методы и системы / -М.: ФИЗМАТЛИТ, 2007.

**А.В. Лямин, А.А. Скшидлевский**

**СТЕНД ДЛЯ ИССЛЕДОВАНИЯ ВАРИАНТОВ РАЗВЕРТЫВАНИЯ СИСТЕМЫ ДИСТАНЦИОННОГО ОБУЧЕНИЯ ACADEMICNT**

*anton@cde.ifmo.ru*

*Санкт-Петербургский государственный университет информационных технологий, механики и оптики*

*г. Санкт-Петербург*

При планировании развертывания больших многопользовательских программных сред существует известная проблема оценки производительности системы в конкретных условиях без проведения испытаний этой системы на конечном оборудовании в реальных условиях. Для решения этой проблемы была предложена методика расчета на основе имитационного моделирования [1], проведено исследование поведения испытуемых при проведении педагогических измерений и разработана соответствующая модель [2]. В настоящей статье рассматривается подход, дополняющий предыдущие исследования и предоставляющий инструменты для получения необходимых данных для реализации имитационных моделей систем дистанционного обучения.

В качестве основы для исследований взята система дистанционного обучения (СДО) AcademicNT, разработанная в Санкт-Петербургском государственном университете информационных технологий, механики и оптики [3]. Есть несколько типовых схем развертывания этой системы как внутри одной сети, так и на территориально распределенных узлах. В качестве примера рассмотрим схему, изображенную на рисунке 1.

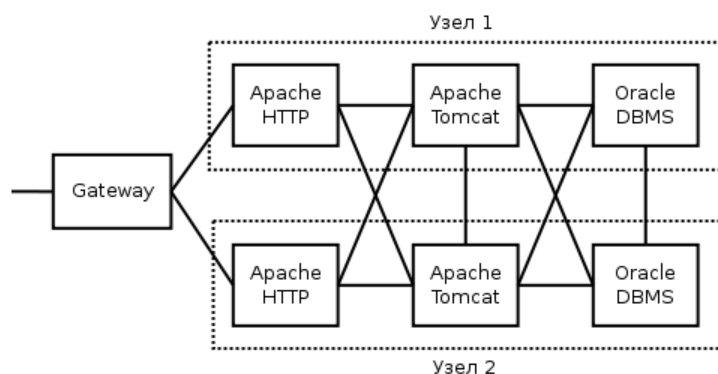


Рис. 1. Типовая схема развертывания СДО AcademicNT

Эта схема представляет собой кластер в локальной сети, где отдельные элементы, такие как веб-серверы Apache HTTP, Apache Tomcat и система управления базой данных (СУБД) Oracle, взаимодействуют друг с другом, создавая единую отказоустойчивый структуру. Каждый элемент этой схемы можно представить в виде элементарного звена обработки запросов. Объединяя такие звенья в соответствии со схемой развертывания системы, получим модель, изображенную на рисунке 2. Основными входными параметрами такой модели являются генератор входного потока запросов и показатели среднего времени обработки запросов каждым элементом модели. На основе результатов апробации, проведенной в рамках исследования поведения испытуемых при проведении педагогических измерений, были получены распределения вероятностей для запросов к системе вовремя аттестаций. Эти исследования легли в основу генератора входного потока запросов. В модели буквами  $H_1, H_2, \dots, H_n$ ;  $T_1, T_2, \dots, T_n$ ;  $D_1, D_2, \dots, D_n$  обозначаются устройства балансировки нагрузки,  $Q_h, Q_t, Q_d$  — очереди, а  $B_1, B_2, B_3$  — устройства обработки.

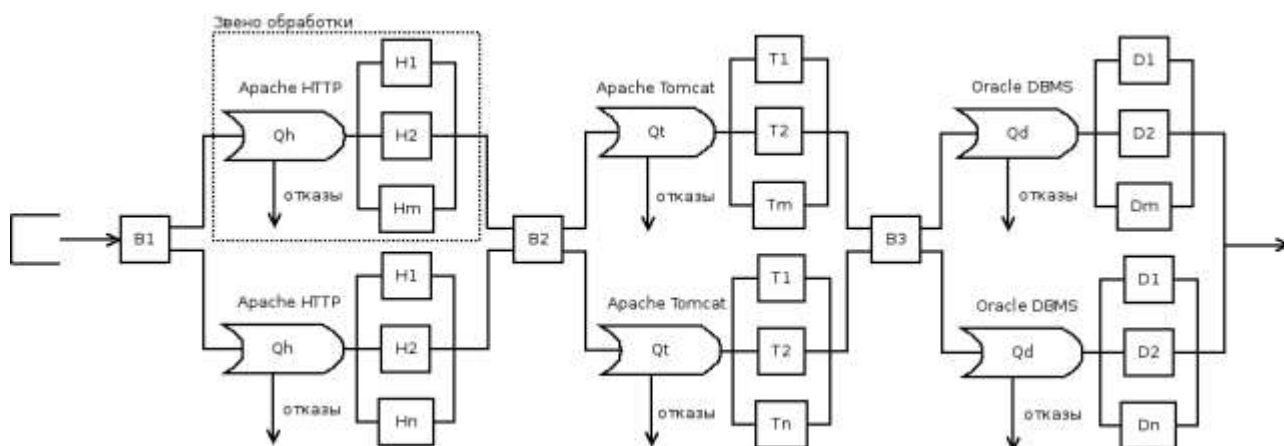


Рис. 2. Модель типовой схемы развертывания СДО AcademicNT

Для определения времени обработки запросов каждым элементом системы при использовании различных схем развертывания было решено разработать экспериментальный стенд. Для повышения эффективности и гибкости разработки, а также минимизации затрат используется технология виртуализации. Эта технология позволяет на одном сервере одновременно запускать множество виртуальных систем, изолировать их друг от друга, распределять ресурсы между ними, создавать виртуальные сети и управлять этими виртуальными системами. В качестве монитора виртуальных машин было решено использовать гипервизор с открытым исходным кодом Xen, поддерживаемый компанией Citrix и распространяемый по лицензии GPL-2. Именно открытость исходных кодов позволила в полной мере использовать технологию виртуализации для сбора и анализа статистической информации. Гипервизор Xen устанавливается поверх операционной системы GNU/Linux.

На виртуальных машинах можно реализовать любую необходимую схему развертывания системы. В схеме приведенной на рисунке 3 используются виртуальные машины с веб-серверами Apache HTTP, Apache Tomcat и СУБД Oracle. Ключевыми компонентами данной схемы являются генератор запросов и агрегатор.

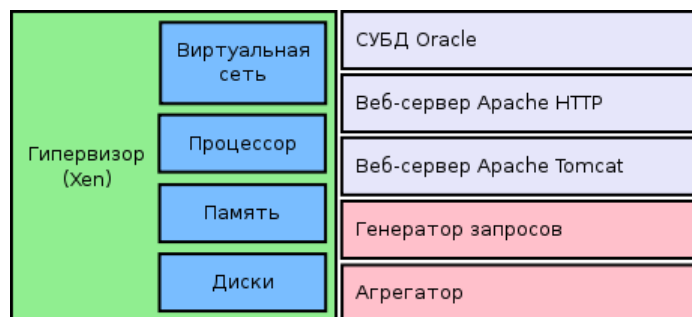


Рис. 3. Виртуализация типовой схемы развертывания СДО AcademicNT

Генератор запросов представляет собой виртуальную систему, генерирующую запросы по заранее составленному сценарию. Сценарий представляет собой набор ссылок и переходов, которые бы выполнял реальный пользователь. При генерации запросов учитываются распределения вероятностей, полученные при апробации. Соответственно, интенсивность запросов можно изменять.

Агрегатор представляет собой систему, собирающую статистику по функционированию реализованной схемы: объем используемой оперативной памяти, использование процессора и объем передачи данных по сети. Также сохраняется дамп сетевого трафика для последующего анализа. На основе полученного дампа рассчитываются входные параметры для модели. Каждый элемент схемы порождает трафик особой структуры, что дает возможность точно сказать, какому элементу принадлежит тот или иной запрос. Каждый промежуточный запрос имеет временную метку. К примеру, если запрос направляется сначала к веб-серверу Apache HTTP, затем на веб-сервер Apache Tomcat и СУБД Oracle, то в рамках одного запроса полное время обработки будет суммой всех временных затрат на каждом элементе данной схемы взаимодействия. Именно это время обработки может определить клиент, отправивший запрос. Но для использования в модели необходимо определить время обработки запроса, затраченное каждым отдельным элементом схемы. И эта задача решается с помощью идентификации фрагментов трафика и временных меток.

В результате проделанной работы был разработан стенд, позволяющий создавать и опробовать различные схемы развертывания СДО AcademicNT. Разработан набор инструментов для исследования работы системы под нагрузкой, а также реализован сбор и анализ статистики проводимых экспериментов для последующего использования в имитационных моделях.

#### *Список литературы*

1. Лямин А.В., Скшидлевский А.А. Методика и алгоритмы расчета вычислительной мощности аппаратной платформы системы для проведения ЕГЭ в компьютерной форме // Труды XV Всероссийской научно-методической конференции "Телематика'2008". - Санкт-Петербург, 2008. - Т. 1. - С. 101-102.
2. Лямин А.В., Скшидлевский А.А. Разработка поведенческой модели испытуемого при проведении педагогических измерений // Труды XVI Всероссийской научно-методической конференции "Телематика'2009". - Санкт-Петербург, 2009. - Т. 2. - С. 349.
3. Лямин А.В., Скшидлевский А.А., Чежин М.С. Организация сетевого межвузовского взаимодействия в информационно-образовательной среде AcademicNT // Информационные технологии в обеспечении нового качества высшего образования. Сборник научных статей. Книга 1 / Труды всероссийской научно-практической конференции с международным участием "Информационные технологии в обеспечении нового качества высшего образования (14 - 15 апреля 2010 г., Москва, НИТУ "МИСиС)". - М.:

**В.А. Максимов**

**ПРОГРАММНЫЙ КОМПЛЕКС ВИРТУАЛИЗАЦИИ ОПЕРАЦИОННОЙ СРЕДЫ ПАРКА  
ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ**

---

*v\_maximov@rsvpi.ru*

*Российский государственный профессионально-педагогический университет*

*г. Екатеринбург*

Главной задачей любой операционной системы является способность унификации способа взаимодействия пользовательских программных продуктов с аппаратной частью компьютера. Общеизвестно, что оптимальным инструментом для её решения является использование технологий виртуализации. Например, популярный текстовый редактор MS Word при сохранении файла на внешние носители скажем на USB-флеш носитель информации, видит накопитель как обобщенный виртуальный диск, не представляя целиком технические детали реализации процесса записи информации на подобный тип носителя. В частности ОС, активно использует возможность виртуализации на уровне аппаратных устройств. Следовательно, возникает некоторая иерархия способов доступа к вычислительным ресурсам ПК. Стандартная иерархия подразумевает наличие трех уровней: нижний уровень – уровень прямого обращения к оборудованию, средний уровень – драйвера устройств с обобщенным интерфейсом управления ими и верхний уровень – набор сервисов операционной системы предоставляемых пользовательским приложениям. В случае MS Word виртуализация применяется на среднем уровне. [1]

Постоянно появляются новые и обновляются старые аппаратные устройства, и что бы пользовательские приложения могли с ним работать создается специализированный драйвер для операционной системы, позволяющий управлять самим устройством. Операционная система осуществляет управление аппаратным устройством через интерфейс использующий API функции и данный интерфейс универсален для разных адаптеров. [2]

Применяются и другие технологии виртуализации на иных уровнях, в частности, на верхнем и нижнем. Если эмулировать программным способом ответы физического устройства на запросы ОС поведение, которого максимально приближено к оригиналу, возникает некая виртуальная среда, включающая в себя соответствующий набор виртуального оборудования. [3]

Сегодня хорошо осознан тот факт, что виртуализация является крайне актуальной и востребованной технологией. Если кратко, то виртуализация — это создание на одном физическом сервере или компьютере нескольких «виртуальных» машин, на каждой из которых может быть установлена своя среда — операционная система, приложения, пользовательские настройки. Такие виртуальные машины функционируют как абсолютно отдельные физические компьютеры. Виртуализация позволяет значительно повысить эффективность имеющегося оборудования, давая возможность производить агрегирование задач и рабочих нагрузок. Будучи открытой системой её можно использовать на любом количестве компьютеров, что дает возможность оптимизировать применение аппаратных ресурсов, и сократить эксплуатационные издержки.

С использованием технологии виртуализации вместо установки всего необходимого программного обеспечения на один компьютер, выделяется отдельная виртуальная машина для каждой новой задачи, например, - одна виртуальная машина для работы с офисным пакетом и другая - для работы с растровой графикой. Вышеописанный подход позволяет обеспечить хорошую изоляцию несвязанных между собой приложений, что значительно влияет на безопасность и стабильность работы. [4] Так как, с увеличением количества необходимого программного обеспечения: