

логических задач. В то же время предлагаемая методика поможет программистам, использующим рекурсию, быть более продуктивными при проектировании рекурсивных процедур.

УДК 683.3

Архитектура корпоративных приложений. Построение уровня доступа к данным приложений

Попова Ю.Б., Бураковский А.И.

Белорусский национальный технический университет

Объектом исследования является система управления учебным процессом – LMS (Learning Management System), разработанная студентами и магистрантами кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» и широко используемая на факультете информационных технологий и робототехники. Данная автоматизированная система размещена в локальной сети БНТУ по адресу: <http://Lms.fitr.bntu.by> (или по IP-адресу: <http://172.16.111.26>).

Проект LMS состоит из нескольких модулей: управление проведением лабораторных работ, курсовым проектированием, дипломным проектированием, тестированием знаний студентов, управлением ошибками, сбором статистики, составлением учебно-методических материалов и некоторых других.

Основной целью разработки является создание единой «точки входа» к источнику данных, т.е. создание единого DAL (Data Access Layer). Отделение логики доступа к данным объекта значительно увеличивает гибкость любой системы, использующей хранилища данных. DAL помогает изолировать другие части приложения от подробностей сохранения объектов.

Для создания единого DAL была выбрана технология доступа к данным LinqToSql. Для работы с контекстом данных был выбран шаблон DAO (Data Access Objects – объекты, позволяющие работать с контекстом данным). Все DAO работают с контекстом данных по принципу IoC. Однако, какому провайдеру данных принадлежит этот контекст, DAO не знают. Этим достигается независимость от поставщика данных, будь то Linq2Sql, Entity Framework, XML, WCF, WebAPI или что-то иное. Контекст данных инициализируется в конструкторе DAO, но тут встает вопрос: каким образом создавать тот или иной контекст? Эту проблему решает паттерн AbstractFactory. В LMS используется двойной подход этого паттерна для создания контекстов. Сначала одна из фабрик определяет, для какого источника данных нужен контекст, далее

определяется, какой тип преобразования данных в объекты используется (задается в конфигурационном файле, для него создается требуемая фабрика) и следующим шагом является создание контекста в фабрике контекстов. После этого этапа, контекст создан и DAO готов работать с данными.

УДК 683.3

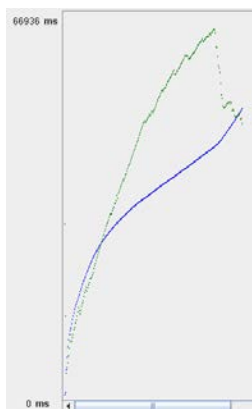
Тестирование производительности LMS

Попова Ю.Б., Романёнок С.В.

Белорусский национальный технический университет

Объектом исследования производительности является система управления учебным процессом – LMS (Learning Management System), разработанная студентами и магистрантами кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» и широко используемая на факультете информационных технологий и робототехники БНТУ, так как при работе с ней в условиях интенсивного использования студентами во время учебных занятий неоднократно были замечены проблемы с низкой производительностью системы: страницы медленно открываются, данные медленно обрабатываются, что приводит к недовольству большинства пользователей.

Основной целью является определение времени отклика для различных страниц приложения. Время откликов измерялось для различных уровней нагрузки: 20 пользователей, когда с системой работает одна группа на занятии. Средняя нагрузка – 100 пользователей – в случае, когда LMS используют в нескольких аудиториях несколько групп. И стрессовая нагрузка, равная 250 пользователям, на случай использования системы в сети Интернет. Основными задачами является тестирование старой и новой версий системы с дальнейшим сравнением результатов.



Инструментом для тестирования был выбран Jmeter. При выборе этого инструмента руководствовались простотой использования и обработки результатов, а также тем, что инструмент бесплатный. Jmeter прослушивает трафик на TCP порту и генерирует из него скрипт. Таким образом, в дальнейшем можно использовать записанный скрипт. После запуска скрипта для выбранного количества пользователей можно наглядно просмотреть полученные результаты: графики усредненного времени отклика и пропускной способности.