

УДК 004.738.5

ПРОЕКТИРОВАНИЕ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ

Серикжанулы Н., студент гр.МИ-121, I курс,
Научный руководитель: к.ф.-м.н. Гайнова Л.Е.
Казахский гуманитарно-юридический инновационный университет,
г.Семей

Большинство Internet и intranet приложений определяются как "статические" или "динамические" в зависимости от содержания и доступа к данным (смотри Рис. 1). Сегодня большинство приложений в Internet - это публикации, предлагающие статическое содержание и обеспечивающие предсказуемый, универсальный доступ. Компании, разрабатывающие свои первые Web-приложения, обычно публикуют информацию о продукте в Internet или данные о служащих в корпоративных сетях intranet.

Последняя волна Internet-приложений - публикация баз данных. Эти системы предоставляют простой статический доступ к динамическим данным. Примерами являются контроль складских остатков через Internet или пересмотр статуса заказа через intranet. Оба этих класса приложений (наряду с простыми приложениями накопления данных) могут развиваться в транзакционные бизнес-приложения. Компаниям – поставщикам информационных систем придется создавать и управлять приложениями в Сети, которые имеют динамический доступ к корпоративным БД и обрабатывают разнородную информацию.

Имея целью Управляемую Диалоговую Обработку Запросов (Online Transaction Processing - OLTP) , а Web как способ доступа, Sybase предлагает название этого нового типа приложений – “WebOLTP”. Такие приложения становятся не простыми программами для просмотра данных, а приложениями для обработки в реальном режиме времени важной деловой информации, например, операций в банке, прием заказов, работа с клиентами.

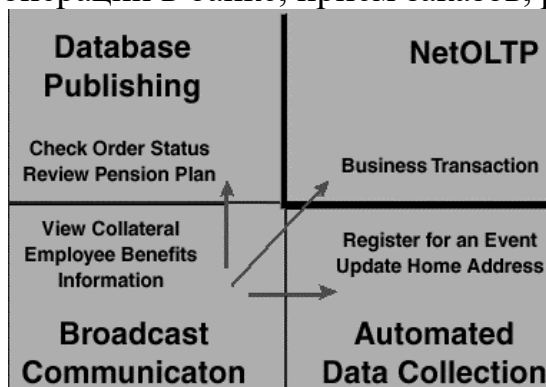


Рисунок 1. Большинство Internet и Intranet приложений классифицируются как “статические” или “динамические” в зависимости от содержания и вида доступа

Современная инфраструктура Internet включает в себя:

- Web-браузер, служащий для отображения страниц в формате HTML (Hypertext Mark-up Language)
- Web-сервер, который занимается хранением и управлением HTML-страниц

Базовая инфраструктура была разработана и до сих пор вполне подходит для публикации статической информации, например, данных маркетинговых исследований.

Базовая инфраструктура Internet за последнее время была расширена в смысле большей динамичности приложений (интерактивных возможностей пользователей) за счет:

- Простых форм запросов и форматирования данных на основе JavaScript для браузера
- API для Web сервера, таких как, например, NSAPI и ISAPI, позволяющих браузерам вызывать приложения на стороне сервера.
- Серверов динамической обработки, которые преобразуют данные из БД в страницы в формате HTML (например, "Dynamo").

Расширенная инфраструктура Web за счет динамической обработки данных, т.е. способности сервера возвращать данные браузеру в соответствии с запросом пользователя или в другой интерактивной форме, предоставляет возможность создания совершенно нового и важного класса приложений от систем поддержки принятия решения через intranet до персональных новостей в Internet.

Однако, даже с учетом этих расширений большинство реализаций Internet-инфраструктур неспособны обрабатывать крупные транзакции. До настоящего времени, различные компании пытались соединять базы данных и Web-серверы вместе. Но без инструментальных средств разработки и администрирования, результаты оказываются в лучшем случае неудобными и сложными в сопровождении.

WebOLTP – имя, которое было предложено компанией Sybase, для описания приложений, выполняющих транзакции в Internet, intranet, extranet или традиционных корпоративных сетях. Отличительные черты WebOLTP при сравнении с OLTP-технологией на мэйнфреймах или в системах клиент/сервер :

“Тонкие клиенты”

В традиционных системах клиент/сервер для запуска приложений необходимо, чтобы клиентское ПО было заранее установлено. В архитектуре с “тонким” клиентом специализированное программное обеспечение не обязательно устанавливается на клиенте, поскольку исполняемые компоненты могут загружаться с Web site для последующего взаимодействия с клиентом. Таким образом, “тонкий” клиент или клиент с “нулевой инсталляцией” получает два ключевых преимущества при работе в сети: универсальный доступ и уменьшение затрат на инсталляцию и управление. Однако, из-за наличия браузеров и HTML, тонким клиентам для динамического управления

бизнес-приложениями необходимо использование дополнительных средств, таких как Java-апплеты.

Большие объемы при большом количестве соединений

В отличие от приложений клиент/сервер и их предшественников, пользователи WebOLTP могут существенно расширить границы отдела или компании, используя extranets или Internet. С этими новыми приложениями "самообслуживания", доступ больше не ограничивается небольшим количеством клерков, регистрирующих заказы, но вместо этого становится открытым для тысяч пользователей, одновременно выполняющих транзакции. Это потребует хорошо масштабируемую архитектуру сервера для построения WebOLTP приложений.

Непредсказуемые нагрузки

Приложения с отчетливо выраженным кругом потребителей работают с достаточно ясно определенными наборами действий и нагрузок. Использование WebOLTP-приложений, благодаря открытому кругу пользователей в них, должно привести к непредсказуемым нагрузкам. По мере того, как любое Web-приложение становится доступно, менеджерами Web-узлов обычно затрудняются предсказать, когда и сколько пользователей попытаются загрузиться. Чтобы справляться с неожиданными скачками загрузки при сохранении приемлемого времени отклика, требуется наличие хорошо регулируемых и адаптирующихся систем.

Короткий жизненный цикл приложения

Каждая следующая генерация приложений кажется обреченной на более короткий жизненный цикл, чем предшествующая. Хотя жизненные циклы, по-видимому, немного удлинятся по мере развития технологии Internet, современные приложения для Сети требуют всего лишь несколько недель или месяцев на разработку, и разрабатываться только в последние 12-18 месяцев. Web-менеджеры стараются корректировать содержимое узла ежедневно и обновлять их графику по крайней мере каждые 9-12 месяцев. Поскольку текущие ожидания потребителя заставят WebOLTP-приложения последовать этому примеру, технология для построения этих систем должна быть очень легкой для использования и развертывания.

Появление WebOLTP-архитектуры

Новая архитектура и модель использования призваны удовлетворить строгие требования к WebOLTP-приложениям. Возникновение 3-уровневой или многоуровневой архитектуры (см. Рис. 4) отвечает потребностям WebOLTP с точки зрения масштабируемости (scalability) и динамического доступа при сохранении всех преимуществ базовой архитектуры.

Инфраструктура Web

В этой новой модели, пользователи находят и запускают приложения, использующие традиционные HTML страницы и Web-серверы. Но вместо просто загрузки статической страницы, динамическая "апплета" загружается в индивидуальный браузер. К тому же апплета поддерживает высокоскоростные протоколы, которые позволяют ей соединяться

непосредственно с сервлетами ("servlets") - ПО, работающим на промежуточном уровне. Обычно сервлеты обеспечивают доступ к одной или более баз данных, реализуют бизнес-логику и возвращают результаты апплете для отображения на клиенте.

Список литературы:

1. Я. Нильсен, Х. Лоранжер. Web-дизайн: удобство использования Web-сайтов: Перевод с английского. М., «Вильямс», 2007. – 255 с.
2. Разработка Internet-ресурса. Часть I: Методические указания по выполнению лабораторных работ по курсу «Проектирование Internet-приложений» / Моск. гос. ин-т электроники и математики. Сост.: С. Л. Макаров. М., 2011. – 32 с.
3. Лазаро Исси Коэн, Джозеф Исси Коэн. Полный справочник по HTML, CSS и
4. JavaScript. М.: «ЭКОМ Паблишерз», 2007. – 1168 с.
5. Основы web-технологий // Храмцов П.Б. , Брик С.А. , Русак А.М. , Сурин А.И. 2-е издание. Бином, 2007. - 376 с.