

АРХИТЕКТУРА СЕРВЕРОВ ПРИЛОЖЕНИЙ РАСПРЕДЕЛЕННЫХ СИСТЕМ НА ПЛАТФОРМЕ J2EE

Е.С. Литвина, 3 курс

Научный руководитель – **Я.В. Богатко**, ассистент
Полесский государственный университет

В условиях современного мира индустрия промежуточного программного обеспечения (MW, от англ. Middleware) переживает значительные изменения. Это связано с переходом к трехзвенной архитектуре клиент/сервер, в которой между клиентом и источником данных располагается промежуточный уровень, реализующий бизнес-логику приложения. Тем не менее, ни одна из существующих категорий MW не может полностью удовлетворить все требования, предъявляемые к современным серверам приложений в сложных распределенных корпоративных средах. Хотя брокеры объектных запросов и мониторы транзакций, в сочетании с асинхронными механизмами передачи сообщений, приближаются к решению этой задачи, они скорее служат основой для новой категории прикладных систем – серверов приложений, индустрия которых активно развивается.

Разработка серверов приложений нацелена на создание объектно-ориентированных распределенных систем и построение прикладных программ из готовых компонентов.

В сервере приложения на платформе J2EE (Java to Enterprise Edition – «версия Java для предприятий») для поддержки взаимодействия и презентации предназначены сервлеты, а также язык тегов и его интерпретатор, прикладной интерфейс для работы с XML, служба электронной почты, служба аутентификации и авторизации [1, с. 283].

Под сервлетом понимают web-компонент, представляющий собой класс Java, предназначенный для динамического формирования содержимого ответов на запросы клиентского приложения. Сервлеты размещаются на сервере и выполняются в специальной среде – контейнере сервлетов, – на виртуальной Java-машине сервера. В общем случае сервлеты не зависят от протокола связи, но наиболее часто они применяются для обработки http-запросов [2, с. 27].

Поддержка интеграции приложений обеспечивается интерфейсом именованного и каталогов, службой сообщений и транзакционным интерфейсом. Поддержка доступа к ресурсам осуществляется компонентами обеспечения связи с базами данных и компонентами подключения архитектур.

Целью поддержки прикладного слоя является создание единого окружения для всех видов прикладной логики, работающей в глобальной сети и без нее. В комплексе J2EE компоненты прикладной логики размещаются на серверной стороне и обеспечивают функциональность, специфическую для данного приложения, например, подготовку прайслистов в ответ на запрос покупателя о покупке. В состав некоторых приложений могут входить несколько компонентов прикладной логики, в зависимости от метода, выбранного для управления состоянием и сохранностью.

Сессионный вариант управляет сессией с клиентом. Имеются модификации, отслеживающие состояния и не делающие этого. Если состояния не отслеживаются, один и тот же компонент может использоваться для работы с разными клиентами.

Объектовый вариант продолжает существование за пределами одной сессии с клиентом. Он имеет состояния, хранящиеся в базе данных или в другой сохранной памяти. Разработчик может сам писать SQL-запросы или другие команды, чтобы запомнить состояние в базе данных, а может

управлять сохранностью автоматически. Вариант с управлением сообщениями обеспечивает возможность асинхронного взаимодействия с клиентом.

Компоненты прикладной логики могут помещаться в контейнер, предоставляющий функциональность по поддержке транзакционности, сохранности и безопасности, общую для разных типов компонентов. Контейнер управляет транзакциями, руководствуясь свойствами, приписанными компонентам прикладной логики во время конфигурирования. Привязка к компоненту прикладной логики производится с помощью службы именования и каталогов. Используя эту службу, клиенты могут привязываться к серверу, зная только имя объекта. В состав сервера J2EE входит транзакционный прикладной программный интерфейс. Для совместимости с системами, построенными на основе спецификации CORBA, в сервер J2EE добавлен набор транзакционных интерфейсов.

Для реализации работы со слоем управления ресурсами сервер приложений J2EE использует стандарты, определяющие прикладной интерфейс (что дает разработчику возможность доступа к источнику табличных данных) и правила построения адаптеров ресурсов.

Язык тегов позволяет вставлять в обычные документы HTML (Hyper Text Markup Language – «гипертекстовый язык разметки») дополнительные теги, расширяющие возможности стандартных страниц. С помощью тегов можно задавать параметры страниц HTML, включать в страницы файлы, адреса которых указываются в тегах, выполнять определения объектов Java, вычислять значения выражений Java и выполнять встроенные фрагменты программ Java. Серверы приложений развивают возможности шлюзов и поддерживают сетевые навигаторы, приложения (такие же, как и в традиционных системах), сложные устройства (например, мобильные телефоны), программы электронной почты и программы клиентов сетевых служб, то есть приложения, взаимодействующие с сервером через стандартные протоколы сетевых служб [1, с. 285].

Архитектура J2EE поддерживает разработку многоуровневых приложений J2EE на основе компонентов. Как правило, в состав системы приложений J2EE входят следующие уровни:

1. Уровень клиента. На уровне клиента Web-компоненты, такие как сервлеты и страницы JavaServer (JSP), или автономные приложения Java предоставляют динамический интерфейс с промежуточным уровнем.

2. Промежуточный уровень. На уровне сервера (промежуточном уровне) объекты EJB и Web-службы инкапсулируют многократно используемую распространяемую бизнес-логику приложения. Компоненты уровня сервера выполняются на сервере приложений J2EE, который предоставляет платформу для выполнения действий и хранения данных.

3. Уровень данных предприятия. Уровень данных служит для сохранения данных предприятия. Как правило, для этой цели используется реляционная база данных.

Приложения J2EE состоят из компонентов, контейнеров и служб. Компоненты являются компонентами уровня приложения. Web-компоненты, такие как сервлеты и JSP, в динамическом режиме обрабатывают запросы, поступающие от Web-страниц. В компонентах EJB реализована бизнес-логика приложений J2EE уровня сервера. Контейнеры Web-компонентов и компонентов EJB содержат службы, относящиеся к Web-модулям и модулям EJB [3].

Таким образом, архитектура серверов приложений распределенных систем на платформе J2EE предоставляет мощные инструменты для создания масштабируемых и надежных приложений. Будущее этой архитектуры связано с дальнейшей интеграцией облачных технологий и микросервисной архитектуры, что позволит повысить гибкость и ускорить разработку. В условиях растущих требований к производительности и безопасности, использование J2EE, вместе с современными подходами к разработке, будет оставаться актуальным и востребованным в соответствующих областях.

Список использованных источников

1. Смелов, В. В. Основы web-программирования на java : учеб.-метод. пособие для студентов по направлению специальности «Информационные системы и технологии (издательско-полиграфический комплекс)». / В. В. Смелов, – Минск : БГТУ, 2009. – 140 с.
2. Штепа, В. Н. Распределённые информационные системы : учеб. пособие для студ. специальности 1-40 05 01 «Информационные системы и технологии» / В. Н. Штепа, – Минск : ПолесГУ, 2015. – 359 с.
3. IBM [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/docs/ru/rsm/7.5.0?topic=applications-j2ee-architecture>. – Дата доступа: 02.04.2025.