Distributed System Using Java 2 Enterprise Edition (J2EE)

B.Ramamurthy

Introduction

- Sun Microsystems provides specifications for a comprehensive suite of technologies to solve large scale distributed system problems.
- This suite is the Java 2 Enterprise Edition, commonly known as J2EE.
- In this discussion we will discuss the architecture of J2EE and how it can be used to develop distributed multi-tiered applications.
- This discussion is based on the tutorial by Sun Microsystems Inc.

J2EE Suite

- Core technology: Container infrastructure, language and environment support
- XML technology
  - The Java API for XML Processing (JAXP)
  - The Java API for XML-based RPC (JAX-RPC)
  - SOAP with Attachments API for Java (SAAJ)
  - The Java API for XML Registries (JAXR)
- Web Technology
  - Java Servlets
  - JavaServer Pages
  - JavaServer Pages Standard Tag Library
- Enterprise Java Bean (EJB) technology
  - Session beans
  - Entity beans
    - Enterprise JavaBeans Query Language
  - Message-driven beans
- Platform services
  - Security
  - Transactions
  - Resources
  - Connectors
  - Java Message Service

Distributed Multi-tiered Applications

- Services, clients (people and application) and data are distributed geographically across many platforms and many machines.
- Multiple tiers:
  - Client-tier (browser or client-application)
  - Web-tier (web-server: Java Server Pages)
  - Business-tier (logic; Examples: Enterprise Java Beans)
  - Enterprise-Information-System (EIS) tier (database)

J2EE clients

- Web clients
  - Dynamic web pages with HTML, rendered by web browsers.
  - Can include applets.
  - Communicates with server typically using HTTP.
- Application clients
  - User interface using GUI components such as Swing and AWT.
  - Directly accesses the business logic tier.

Web-tier Components

- Client can communicate with the business tier either directly or through servlets ot JSP that are located in the web-tier.
- Web-tier can help in pre-processing and allows distribution of the functionality.
- See Figure 2-1
- Servlets are special classes to realize the request-response model (get, post of HTTP).
- JSP is a developer-friendly wrapper over the servlet classes.

Business-tier Components

- This is defined by the logic that pertains to the (business) application that is being developed.
- Enterprise Java Beans (EJB) can be used to implement this tier.
- This tier receives the data from the web-tier and processes the data and sends it to the EIS-tier and takes the data from the EIS and sends it to the web-tier.
- See Figure 1-3, and Figure 1-4

Enterprise Java Beans

- Session beans
- Entity Beans
  - Bean-managed Persistence (BMP)
  - Container-managed Persistence (CMP)
  - Enterprise Javabeans Query Lanaguage
- Messaging Bean
  - Session bean with Java Messaging features

Session Beans

- For transient functions
- Represents “conversational” state
- Typically one per request
- Data is non-persistent
- Lifetime is limited by the client’s: once the client exits, the session bean and data are gone.
- Simple and easy to program.
- Light-weight.

Entity Bean

- “Transactional” in behavior
- Can be shared among clients
- Persistent: data exists permanently after client quits.
- Corresponds to a row a relational database.
- The persistence (storing into the database) can be automatically done by the “container” (CMP) or explicitly by the bean (BMP)

Enterprise Information System (EIS) Tier

- In general this corresponds to the database (relational database) and other information management system.
- The other information management systems may include Enterprise Resource Planning (ERP) and legacy system connected through open database connectivity.

Container Services

- A container interfaces the programmatic components such as EJBs to the declarative components.
- Container services include security, transaction management, naming services, and remote connectivity.
- The fact that the J2EE architecture provides configurable services means that application components can behave differently based on where they are deployed.
- The concept of “deployable units” and “containers” where they can be deployed is central to J2EE.