App Servers & J2EE Platform

Contents:
- TP-Monitors, OTS
- CORBA
  - service
  - evolution
- Application Servers
- Component-based development
- J2EE Platform
  - EJ Bs

Mariano Cilia / cilia@informatik.tu-darmstadt.de

Transaction Processing Monitors

TP-Monitors
- TPMs are SW systems intended to provide transactional processing for large applications with many clients
- No "standard" TPM (many ad-hoc solutions)
- Main function of TPMs is to integrate systems and manage resources
- Repetitive workloads, i.e. users execute predefined simple functions (canned transactions)

TP Monitors (cont)
- OLTP functions typically simple, 10 disk I/Os, $10^5$ to $10^7$ instructions
- High availability
- Many terminals (clients): 1000 to 100000
- Automatic load balancing
- TPMs are typically built according to a three tier architecture

TP Monitors (cont)
- 30 years old technology (robust)
- maximize the reuse of scarce system resources
- high-volume of transactions
- automatically manage the entire environment that the business system runs in, including:
  - transactions
  - resource management
  - fault tolerance
  - load balancing
  - communications
  - procedure oriented

TP Standards
- Standardization of APIs ==> portability of applications and DBMSs across TPMs
- Standardization of protocols ==> interoperability of TPMs
- Standardization began mid-80s and standards are still evolving

Transactions
Commercial TPM Products
- CICS (IBM)
- IMS (IBM)
- TUXEDO (BEA)
- ACMS (Digital)
- Encina (Transarc)
- TOP END (AT&T/NCR)
- Pathway/TS (Tandem)
- Adabas TPF (Software AG)
- Microsoft Transaction Server (Microsoft)
- X/Open DTP standard
- OTS

TP-Monitors - Trends
- TP monitors become portable application server environments
- TP monitors become universal traffic managers
- TP monitors become resource brokers
- TP monitors discover client/server tools
- TP monitors meet objects ==> Object Monitors

CORBA
- CORBA - Common Object Request Broker Architecture
- Object-based middleware layer
- Main goal is interoperability of objects
- Defined by Object Management Group (OMG)
- OMG has more than 800 members (all major SW vendors except Microsoft, many major application developers)

OMG Object Management Arch.

IDL
- Declarative language used to define objects and their contractual interfaces
- IDL provides OS- and PL-independent interfaces to all the services on the CORBA bus
- IDL only for specification of attributes, parent classes, methods and events supported, and exceptions raised, no implementation
- Implementations in any language for which bindings exist (C, C++, Ada, Smalltalk, COBOL, Java)
IDL (cont.)
- To request a service from another object, an object must know the target object’s supported interface
- Interface Repository contains the metadata needed to discover other components at run time

Object Request Broker (ORB)
- ORB functions as an object bus
- Static (compile-time) and dynamic (run-time) method invocation
- High-level language bindings: standard interfaces and language-independent types allow invocation of a service independent of the language it is written in
- Self-describing system through metadata for interface definitions, either IDL precompiler or directly from OO-PL compiler (Java bytecode)

ORB (cont.)
- Local/remote transparency: ORB can function on stand-alone machine or interoperate with other ORBs (via IIOP - Internet Inter-ORB Protocol)
  - while user doesn't need to be concerned with location of an object, price is in the performance
- Support for security and transactions
- Polymorphic messaging (target specific)
- Coexistence with legacy systems through encapsulation of legacy code with IDL interface

CORBA Services
- Life Cycle Service
  - defines operations for creating, copying, moving and deleting components on the bus
- Persistent State Service
  - provides single interface for storing objects persistently on various storage servers (OODBMSs, RDBMSs, files)
- Event Service
  - allows components to register/unregister interest in specific events. Event channel collects and distributes events among objects

CORBA Services (cont.)
- Naming Service
  - allows components on the bus to locate other components by name. Also allows objects to be bound to existing network directories or naming contexts (ISO X.500, OSF DCE, Sun NIS+, Internet LDAP)
- Concurrency Control Service
  - provides a lock manager that can obtain locks on behalf of transactions or threads

CORBA Services (cont.)
- Transaction Service
  - provides two-phase commit coordination among recoverable components using either flat or nested transactions
- Relationship Service
  - provides a way for creating dynamic links among components and to traverse them. May be used for enforcing referential integrity, containment, etc.
CORBA Services (cont.)
- Externalization Service
  - provides a standard way for getting data into and out of a component using a stream-like mechanism
- Query Service
  - provides query operations on objects. Based on SQL3 and ODMG’s OQL (not implemented)
- Licensing Service
  - provides metering of a component’s use for fair charging (per session, per node, per site, …)

CORBA 1.0
- CORBA 1.0 specified minimal ORB functionality:
  - Basic ORB
  - IR (Interface Repository, BOA (Basic Object Adaptor))
  - C Bindings
  - Naming
  - Events
  - Life Cycle
  - Persistence

CORBA 2.0
- CORBA 2.0 provides for interoperability among ORBs
  - IIOP
  - Federated IR
  - Documents
  - C++ bindings
  - Trader
  - Transactions
  - Concurrency
  - Security
  - Externalization
  - Collections
  - Relationships
  - Time

CORBA Services (cont.)
- Properties Service
  - associates named values (properties) with any component.
- Time Service
  - provides interfaces for synchronizing time in a distributed object environment and for defining and managing time-triggered events

CORBA Services (cont.)
- Security Service
  - provides framework for distributed object security (authentication, access control, confidentiality and non-repudiation, delegation of credentials)
- Trader Service
  - provides “yellow pages” allowing objects to publicize their services and bid for jobs
- Collection Service
  - provides interfaces to create and manipulate collections

CORBA 3.0
- CORBA 3.0 addresses the issues of portability, vertical integration, life in the object-Web
  - Messaging (MOM) Server Portability (POA)
  - Multiple Interfaces Java Bindings
  - Business Objects/Java Beans
  - Objects-by-Value Mobile Agents
  - CORBA/DCOM Automatic Persistence
  - IIOP Firewall Support Workflow
  - Domain-Level Frameworks
Object Transaction Service (OTS)
- Object Transaction Service specification defines transactional service based on CORBA for OO programming environment
- Specified by OMG (Object Management Group)

ORB vs. TP monitor
- ORB performs many functions of a TP monitor
  - if object is defined as transactional (via IDL attribute) ORB directs transactional client calls to transactional server
  - when client issues method invocations, ORB manages the transaction context using the transaction service
- OTS does not specify a required 2PC protocol ==> interoperability depends on implementation
- ORB doesn’t provide performance of TPM

from TP-Monitors to Component Transaction Monitors
(also known as Application Servers)

N-tiered Systems
- n-tiered systems developed with applications deployed over the Internet
  - client
  - [the network: ISP - Internet Service Provider]
  - web server
  - application server(s)
  - resource (e.g. database) server(s)

N-tiered Sys: Reference Arch.

ORBs
- facilitate connectivity between the client application and the distributed objects
  - locate and use distributed objects
  - communication backbone
- let distributed objects interoperate across address spaces, languages, OS, and networks
- not always adequate in high-volume transactional environments
Object Transaction Monitor

- (OTM)
- hybrid of:
  - TP Monitors
  - ORB Technologies
- make easier for developers to create, use and deploy business systems
- capable of handling huge user population and mission-critical work
- provide an infrastructure to manage:
  - transactions, object distribution, concurrency, persistence and resource management

Component Transaction Monitor

- (Application Servers)
- Component Model + (TP-Monitor + ORB)
  - robust server-side component model
  - deployers define and administer declaratively the properties of the components by setting their attributes
  - resource and service management (monitor)
- CTMs are to business objects what ...
  - RDBMS are to data
  - the railway system is to the trains

CTM Approach

At runtime,
- it "intercepts" all incoming calls
- invokes the appropriate callback objects within a container
- and then passes the request to your object
- Also,
  - pre-starts pools of objects
  - distributes loads
  - provides fault-tolerance
  - coordinates multi-component transactions
- if you play by the CTM’s rules, your objects become managed, transactional, robust, persistent, secure and high-performing

Application Servers

- Modern equivalents to traditional TP monitors
- All based on Java 2 Enterprise Ed (except M$)
- Commercial:
  - BEA WebLogic, IBM WebSphere, Oracle 9i AS, Borland AS, CA, IONA, Fujitsu, ...
- Open Source
  - JBoss, JOnAS
- Vendors differentiation based on:
  - Scalability, high availability, reliability, ease of use, legacy data and app integration, complementary prods (e.g. personalization, workflow,...), etc.

Application Servers (cont.)

- Trend to migrate business logic back to the server and away from the client
  - intellectual property protection/sensitive business data better kept on server, rights can be revoked
  - manageability through thin clients (upgrades, versioning, bug fixes managed on servers)
  - performance (especially of DB intensive apps.)
  - secure network communications
  - minimize downtime
  - reuse of components
Application Servers (cont.)

- increased reliability through server redundancy
- increased flexibility through multiple tiers
- multi-client support
  - conventional desktops
  - web-clients
  - esoteric devices (smartcards, PDAs, information appliances, cell phones)
- support of variety of middleware services and resource management (multithreading, resource sharing, replication, load balancing)

Open Standards

- Open Source: open reference impl.
- HTML: Web user interface
- XML: Web data & vocabularies
- EJBs: Business logic
- JCA: connectors
- Web Services (SOAP, WSDL, UDDI)
  - Dynamic app navigation, discovery and interaction
- BPEL4WS: flow composition

App. Server Functionality

- Component container
- Transaction integration
- Thread/connection pooling
- Memory management
- Load balancing/failover

- Session/state mgmt
- Connections
  - RDBMS
  - Transaction Systems
  - ERP Systems
- Dynamic Page generation
- HTTP

App. Server Functionality (2)

- Quality of service: clustering, cloning, routing, failover, session persistence, session migration, load balancing
- Caching: data and objects
- Authentication: single sign-on
- Messaging

The Application Server Market

<table>
<thead>
<tr>
<th></th>
<th>2000</th>
<th>2001</th>
<th>2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>(million dollars)</td>
<td>990</td>
<td>1180</td>
<td>2100</td>
</tr>
<tr>
<td>Market share (%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BEA</td>
<td>33</td>
<td>34</td>
<td>27</td>
</tr>
<tr>
<td>IBM</td>
<td>22</td>
<td>31</td>
<td>30</td>
</tr>
<tr>
<td>Sun</td>
<td>10</td>
<td>9</td>
<td>4</td>
</tr>
<tr>
<td>Iona</td>
<td>0</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Oracle</td>
<td>1</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>Others</td>
<td>34</td>
<td>22</td>
<td></td>
</tr>
</tbody>
</table>

Java Application Servers

- Java has grown into one of the premier PLs
  - ubiquitous on the I*Net
  - as 3GL it competes with C++ (less powerful but easier to use, interpreted, no memory leaks)
  - as 4GL it provides classes for business abstractions
  - loading of new functionality via standard bytecodes
  - platform for component-oriented computing
  - reasonably fast
Java – Open Community

- Java Community Process
  - JCP.org (starting on 1999)
- members: companies + individuals
- Java Specification Request
  - 60% are led by Java partner companies
  - 40% are led by Sun

The J2EE Platform for E-Business

- Technology of Choice for Today’s E-Business Systems
  - 80% of all enterprises use the Java language (Gartner)
  - 92% of companies that choose J2EE for enterprise computing are happy with their choice (Forrester)
  - 78% of executives view J2EE as the most effective platform for building and deploying Web Services (Giga poll)
  - The J2EE Platform market is more than $9 billion strong and growing

Some Java Numbers

- More than 2.5 Million of registrations at Sun's Java Developer Connection
- 60% of developers expect to use Java when building web services
  - (Evans, Spring 2002)
- J2EE is was $2.19 billion market in 2001
  - $9 billion in 2003
- 53.8% of developers are either currently use J2EE or plan to use it
  - (Evans, Spring 2002)
- 31 v1.3 J2EE Commercial Licensees (Spring 2003)
- 6 v1.4 J2EE Commercial Licensees (Winter 2004)