Transactions and J2EE

Gie Indesteege
Instructor & Consultant
gindeesteeg@abis.be
Answer to Your Questions

- What is a transaction?
- Different transaction types?
- How can J2EE manage transactions?
Abis Training and Consulting

- High-level technological ICT services
  - Training
  - Consulting
- Host technology
- Database consolidation
- Reuse & integration

www.abis.be
Agenda

- Transaction: definitions and glossary
- Transaction participants
- Transaction types
- J2EE transactions
- Q & A
“Set of related operations that must be completed together”

“Atomic Logical Unit of Work, that must be treated in a coherent and reliable way.”
Transaction: Glossary

Transaction properties

• Atomicity
• Consistency
• Isolation
• Durability

Transaction demarcation/boundaries

• Commit
• Rollback
- Transaction: definition and glossary
- **Transaction participants**
- Transaction types
- J2EE transactions
- Q & A
Transaction Participants

- Application
- Resource manager
  - Relational database
  - TP monitor
  - JMS provider
- Transaction resource object (E.g. Connection)
- Resource adapter – connector
- Transaction manager
  - Coordination of distributed transactions
  - Maintains transaction context
  - XA protocol
Transaction Participants

Client

J2EE Application Server

J2EE application component

Transaction manager

Resource manager JDBC

Resource manager JCA/JMS

DB

EIS
Agenda

- Transaction: definition and glossary
- Transaction participants
- **Transaction types**
- J2EE transactions
- Q & A
Transaction Types

- **Local transaction**
  - 1 resource manager
  - 1 phase commit

- **Distributed (global) transaction**
  - Access multiple transactional resources
  - 2 phase commit

- Flat transaction
- Nested transaction
- Compensating transaction
- Extended transaction
Local Transaction

1 resource manager

Application

DBMS
Traditional Transaction Manager

TeleProcessing monitor + multiple resource managers
Application Server As Trx Manager

Web Server system

Enterprise system

Application Server

Program

1

J2EE application server

DBMS-1

DBMS-2
Distributed Transaction

Multiple resource managers

J2EE Application Server

Client

EJB A

EJB B

EJB C

JDBC

JCA

JMS

DB

EIS

Distributed Transaction

Multiple resource managers

J2EE Application Server

Client

EJB A

EJB B

EJB C

JDBC

JCA

JMS

DB

EIS
Distributed Transaction

Multiple transaction managers
Distributed Transaction

Transaction manager

Prepare to commit

Resource manager 1

Ready to commit

Resource manager 2

Prepare to commit

Ready to commit

Commit

Committed

2 Phase Commit
Distributed Transaction

2 Phase abort

Transaction manager

Resource manager 1

Resource manager 2

Prepare to commit

Ready to commit

Prepare to commit

Don't commit

Abort

Aborted

Abort

Aborted
Flat Transaction

Begin trx

commit executed

App1

App2

...

Appn

Transaction context propagation

End trx

commit or rollback executed
Nested Transaction

Begin T1

Begin T2

Begin T3

Commit or Rollback T3

Commit or Rollback T2

Commit or Rollback T1

commit executed

rollback executed

rollback executed

rollback executed

neglected

neglected
Compensating Transaction

- Undo effect of previously committed transaction
  - for local transactions (resource adaptors)
  - programmatic application logic

Example

```java
updateEIS();
try {
    usertrx.begin();
    updateRDBMS();
    usertrx.commit();
} catch (RollbackException ex) {
    undoUpdateEIS();
}
```
Extended Transaction

- Long lived
- Message oriented
- Web services
  - Business Transaction Protocol (OASIS) – JSR-156
  - WS-Transaction, WS-Coordination
  - Activity Service (OMG) – JSR-95
Agenda

- Transaction: definition and glossary
- Transaction participants
- Transaction types
- J2EE transactions
- Q & A
J2EE Transactions

- J2EE Technology
  - JTA – Java Transaction API
  - JTS – Java Transaction Service

- J2EE tiers
  - Client tier
  - Web tier
  - EJB tier
  - EIS tier
  - Web services

- J2EE Resource managers
  - JDBC
  - JCA
  - JMS
J2EE Technology
JTA and JTS

JTA

- `javax.transaction.UserTransaction`
  - explicit in code
  - implicit in EJB container

JTS

- `javax.transaction.TransactionManager`
- `javax.transaction.xa.XAResource`
  handled by J2EE server and EIS resource managers
J2EE Tiers

- Client tier
- Web tier
- EJB tier
- EIS tier
- Web services
No J2EE requirements for applets or application clients

Advise:
delegate transactional responsibility to server tiers
J2EE supports programmatic transaction demarcation in servlets/JSPs

- implement in service() method (begin + commit)
- 2 phase commit implied

Advise:

Use JNDI to lookup for object
java:comp/UserTransaction
Start new transaction context

public void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

    Context ctx = new InitialContext();
    UserTransaction userTrx = (UserTransaction) ctx.lookup("java:comp/UserTransaction");

    userTrx.begin();
    // user code for accessing resources
    userTrx.commit();
}
J2EE EJB Tier

J2EE supports transaction demarcation

- programmatic (bean managed)
  - Session beans
    - `afterBegin()`, `beforeCompletion()`, `afterCompletion()`
  - Message driven beans
    - `onMessage()`

- UserTransaction

- declarative (container managed)
  - Session beans
  - Entity beans
  
  transaction attributes in deployment descriptor
public void myMethod(...) throws RemoteException {
    UserTransaction userTrx = ejbContext.getUserTransaction();
    try {
        userTrx.begin();
        // user code for accessing resources
        userTrx.commit();
    } catch (Exception e) {
        try {
            userTrx.rollback();
        } catch (SystemException se) { ... }
    }
}

Bean managed transaction
## J2EE EJB Tier

### Container managed transaction attributes

<table>
<thead>
<tr>
<th>Required</th>
<th>Client — EJB</th>
<th>Client — EJB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RequiresNew</td>
<td>Client — EJB</td>
<td>Client — EJB</td>
</tr>
<tr>
<td>NotSupported</td>
<td>Client — EJB</td>
<td>Client — EJB</td>
</tr>
<tr>
<td>Supports</td>
<td>Client — EJB</td>
<td>Client — EJB</td>
</tr>
<tr>
<td>Mandatory</td>
<td>Client — EJB</td>
<td>Client — EJB</td>
</tr>
<tr>
<td>Never</td>
<td>Client — EJB</td>
<td>Client — EJB</td>
</tr>
</tbody>
</table>

*Exception:*
... Container managed transaction descriptor

```xml
<container-transaction>
  <method>
    <ejb-name>PersonBean</ejb-name>
    <method-name>*</method-name>
  </method>
  <trans-attribute>Required</trans-attribute>
</container-transaction>

<container-transaction>
  <method>
    <ejb-name>CompanyBean</ejb-name>
    <method-name>updateInfo</method-name>
  </method>
  <trans-attribute>Mandatory</trans-attribute>
</container-transaction>
...
```
## J2EE EJB Tier

### Best practices XXX

<table>
<thead>
<tr>
<th></th>
<th>Required</th>
<th>Requires New</th>
<th>Not Support’d</th>
<th>Supports</th>
<th>Mandatory</th>
<th>Never</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session</td>
<td>XXX</td>
<td></td>
<td>EIS trx (no J2EE support)</td>
<td>NOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity BMP</td>
<td>XXX</td>
<td></td>
<td></td>
<td>NOT re-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>com-</td>
<td>com-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>men-</td>
<td>men-</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ded-</td>
<td>ded-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Entity CMP</td>
<td></td>
<td>logging</td>
<td></td>
<td>get/set</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>of</td>
<td>of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CMP/CMR</td>
<td>CMP/CMR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDB</td>
<td>XXX</td>
<td></td>
<td>idem</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Depends on client*
Notes:

● Use Container Managed Transactions preferably

● trigger rollback by container via method setRollbackOnly() on
  • SessionContext
  • EntityContext
  • MessageDrivenContext
Access via

- **JTA transaction**
  - transaction context propagated via J2EE server

- **resource manager local transaction**
  - only if no JCA connector is available
  - requires explicit commit/rollback
  - provide compensating transactions
Advise:

- access EIS system in transaction scope
- use appropriate isolation level for EIS
  - ReadUncommitted
  - ReadCommitted
  - RepeatableRead
  - Serializable
J2EE Resource Managers

- **JDBC** – Java Data Base Connectivity
  - J2EE defines access to 1 JDBC resource per trx

- **JCA** – Java Connector Architecture
  - integration with EIS via standard resource adapters
    - NoTransaction
    - LocalTransaction
    - XATransaction

- **JMS** – Java Messaging Service
  - J2EE supports at least 1 JMS provider per trx
  - messages are delivered/consumed in UoW
  - transactions are NEVER propagated between sender and receiver of message!
Summary

● Transaction = logical unit of work, sharing ACID properties
● Transaction participants application, trx manager, resource managers, resource adapters
● Transaction types
  • local or distributed
  • flat or nested
  • extended
● J2EE transaction management
Can you live any longer without transactions?
If not, think of J2EE support!
J2EE Transactions - References

● Books
  • Designing Enterprise Applications with the J2EE platform (2nd edition) by Inderjeet Singh, Beth Stearns, Mark Johnson et al. (Addison Wesley 2002) ISBN 0-201-78790-3
  • IBM Redpaper Transactions in J2EE by Jan Smolenski and Peter Kovari (IBM 2003) REDP-3659-00

● URLs
  • http://java.sun.com/products/jta
  • http://www-106.ibm.com/developerworks/java/
Q&A
http://www.abis.be

thanks you