Atomic Transaction

UNIT-IV

Introduction

- Transaction is fundamental to many business process requirements
- Corporate data need to wrap a series of changes into a single action
- Atomic transactions implements commit and rollback features to enable cross-service transaction support
- WS-AtomicTransaction specification defines protocols for use with WS-Coordination
- WS-AtomicTransaction coordination type is referenced as part of a SOAP header

ACID transactions

- "ACID" is an acronym representing the following four required characteristics of a traditional transaction:

- Atomic - Either all of the changes within the scope of the transaction succeed, or none of them succeed.
- Consistent - None of the data changes made as a result of the transaction can violate the validity of any associated data models
- Isolated - If multiple transactions occur concurrently, they may not interfere with each other
- Durable - Upon the completion of a successful transaction, changes made as a result of the transaction can survive subsequent failures

- **WS-AtomicTransaction specification enable cross-service transaction functionality**

Transaction Protocols

- WS-AtomicTransaction is an extension created for use with the WS-Coordination context management framework
- So to participate in an atomic transaction, a service first receives a coordination context from the activation service
- It can subsequently register for available atomic transaction protocols

- The following primary transaction protocols are provided:
- A Completion protocol - used to initiate the commit or abort states of the transaction.
- Durable 2PC protocol - for which services representing permanent data repositories should register
- Volatile 2PC protocol - to be used by services managing non-persistent (temporary) data.
- Most of these protocols are used to enable a two-phase commit (2PC)

Atomic transaction coordinator

- When WS-AtomicTransaction protocols are used, the coordinator controller service can be referred to as an *atomic transaction coordinator*
- *The atomic* transaction coordinator manage the participants of the transaction process and decides the transaction's ultimate outcome

Atomic transaction process

- Atomic transaction coordinator decides the outcome of a transaction
- It decides based on feedback it receives from all of the transaction participants
- 2 phase of feedback collection:
- P*repare phase all participants are notified by the* coordinator to issue a vote (either a "commit" or "abort" request)
- if all votes are received are commit, it declares the transaction successful and the changes are committed
- But, if any one vote requests an abort, or any participants fail to respond, then the transaction is aborted, and all changes are rolled back

Atomic transactions and SOA

- Ensures robust execution environment in cross-service interaction
- Promotes interoperability when extended into integrated environments
- Assured a guaranteed all-or-nothing outcome, even span across different solutions built with different vendor platforms