Java Transaction API (JTA)

只看重要的2页就好了

都看一下　但是记住重要的两页

Database Transaction

- Atomicity
  - Each transaction be "all or nothing"
- Consistency
  - Bringing the database from one valid state to another
- Isolation
  - The concurrent execution of transactions results in a system state that would be obtained if transactions were executed serially
- Durability
  - Once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors

不考

Distributed Transaction

大概了解，不是在一台主机的，分布式的事务

- A transaction in which two or more network hosts are involved
  - Hosts provide transactional resources
  - Transaction manager responsible for creating and managing a global transaction
- Two-phase commit (2PC): usually applied for updates able to commit in a short period of time (from milliseconds to minutes)
- Long-lived distributed transactions: more sophisticated techniques that involve multiple undo levels are used, practically implemented in systems based on Web Services

Two-Phase Commit Protocol (2PC)

- A distributed algorithm that coordinates all the processes that participate in a distributed atomic transaction on whether to commit or abort (roll back) the transaction

两阶段提交

- Assumptions
  - One node is designated the coordinator, while the rest of nodes are designated the cohorts
  - There is stable storage at each node with a write-ahead log
  - No node crashes forever
  - The data in the write-ahead log is never lost or corrupted in a crash
  - Any two nodes can communicate with each other

2PC's Basic Algorithm

5、6页重点看

- Commit request phase (voting phase)
  1. The coordinator sends a query to commit message to all cohorts and waits until it has received a reply from all cohorts
  2. Cohorts execute the transaction up to the point where they will be asked to commit; they each write an entry to their undo log and an entry to their redo log
  3. Each cohort replies with an AGREEMENT message (voting YES, to commit) if the cohort's actions succeeded, or an ABORT message (voting NO, not to commit) if the cohort experiences a failure that will make it impossible to commit

2PC's Basic Algorithm (cont.)

- Commit phase (completion phase)
  - Success: if the coordinator received an AGREEMENT message from all cohorts during the commit-request phase
    1. The coordinator sends a COMMIT message to all the cohorts
    2. Each cohort completes the operation, and releases all the locks and resources held during the transaction
    3. Each cohort sends an ACKNOWLEDGMENT to the coordinator
    4. The coordinator completes the transaction when all acknowledgments have been received
  - Failure: if any cohort votes NO during the commit-request phase (or the coordinator's timeout expires):
    1. The coordinator sends a ROLLBACK message to all the cohorts
    2. Each cohort undoes the transaction using the undo log, and releases the resources and locks held during the transaction
    3. Each cohort sends an ACKNOWLEDGMENT to the coordinator
    4. The coordinator undoes the transaction when all acknowledgements have been received

X/Open XA

不考XA

- A specification by the Open Group for distributed transaction processing
- Specifying how a transaction manager will roll up the transactions against the different data-stores into an "atomic" transaction and execute this with the 2PC protocol for the transaction
- A type of transaction coordination, allowing many resources to participate in a single, coordinated, atomic update operation
- The XA specification describes what a resource manager must do to support transactional access, and resource managers that follow this specification are said to be XA-compliant

XA Transactions

- Applications that use global transactions involve one or more Resource Managers and a Transaction Manager
  - A Resource Manager (RM) provides access to transactional resources.
  - A Transaction Manager (TM) coordinates the transactions that are part of a global transaction
- The individual transactions within a global transaction are “branches” of the global transaction

要实现XAResource接口

XA Transactions (cont.)

- To carry out a global transaction, it is necessary to bring each component to a point when it can be committed or rolled back
- Depending on what each component reports about its ability to succeed, they must all commit or roll back as an atomic group
- The process for executing a global transaction
  - In the first phase, all branches are prepared
  - In the second phase, the TM tells the RMs whether to commit or roll back
- A global transaction might use one-phase commit (1PC), when a TM finds that a global transaction consists of only one transactional resource

Java Transaction API (JTA)

- Enabling distributed transactions to be done across multiple X/Open XA resources in a Java environment
- Architecture
  - A transaction manager or transaction processing monitor (TP monitor) coordinates the transactions across multiple resources such as databases and message queues
  - Each resource has its own resource manager, which typically has its own API for manipulating the resource
  - The resource manager allows a TP monitor to coordinate a distributed transaction between its own and other resource managers
  - The application communicates with the TP monitor to begin, commit or rollback the transactions
- The JTA architecture requires that each resource manager must implement the *javax.transaction.xa.XAResource* interface in order to be managed by the TP monitor

Lab

- Using JTA, MySQL and ActiveMQ to implement a distributed transaction processing application

Programming Practice

- Design and implement a Java application with distributed transaction
  - The system maintains a student list (matriculation numbers and names), where the matriculation number must be unique
  - The user inputs the matriculation number and name of a new student, to insert a record
  - Whenever a new record is inserted, a message is sent to another application via the broker