

Ashkan Farivarmoheb

Transaction Processing Assignment

Advanced Software Architecture

Part I) Nested transactions

(App.java)

I created a resource that is prepared with `@Transactional`. This resource has implemented an interface that it also is prepared with `@Transactional` too. I use this resource and create it with Container.

```
Container<Counter> container = new Container<Counter>();

Counter counterImpl1 = container.create(new CounterImpl());
Counter counterImpl2 = container.create(new CounterImpl());
Counter counterImpl3 = container.create(new CounterImpl());
Counter counterImpl4 = container.create(new CounterImpl());
Counter counterImpl5 = container.create(new CounterImpl());
Counter counterImpl6 = container.create(new CounterImpl());
Counter counterImpl7 = container.create(new CounterImpl());
```

These counters will be used in t1 – t7 transactions. Transactions are AtomicAction that are Narayana based. I created transactions as the assignment. And if the parent aborts, all its children will be aborted too. If parent commits, all its children will be committed too. I throw a RuntimeException in t4 transaction. Because the t4 is parent of t7, as a result the t4 and t7 will be aborted. Also values for counterImpl4 and counterImpl7 won't be changed.

```
t4 = new AtomicAction();
t4.begin();
try {
    counterImpl4.increment();

    t7 = new AtomicAction();
    t7.begin();
    try {
        counterImpl7.increment();
    } catch (Exception e) {
        abortRecursive(t7);
    }

    //int b = 1 / 0;
```

```

        if (true)
            throw new RuntimeException();
    } catch (Exception e) {
        abortRecursive(t4);
    }
}

```

```

Run App
"C:\Program ...
Jul 20, 2015 9:45:37 PM com.arjuna.ats.arjuna.recovery.TransactionStatusManager addService
INFO: ARJUNA012163: Starting service com.arjuna.ats.arjuna.recovery.ActionStatusService on port 52110
Jul 20, 2015 9:45:37 PM com.arjuna.ats.internal.arjuna.recovery.TransactionStatusManagerItem <init>
INFO: ARJUNA012337: TransactionStatusManagerItem host: 127.0.0.1 port: 52110
Jul 20, 2015 9:45:37 PM com.arjuna.ats.arjuna.recovery.TransactionStatusManager start
INFO: ARJUNA012170: TransactionStatusManager started on port 52110 and host 127.0.0.1 with service com.arjuna.ats.arjuna.recovery.ActionStatusService
Jul 20, 2015 9:45:37 PM com.arjuna.ats.arjuna.coordinator.BasicAction checkChildren
WARN: ARJUNA012098: Abort of action id 0xffff0a83801:cb8d:55ad2cb8:23 invoked while child actions active
Jul 20, 2015 9:45:37 PM com.arjuna.ats.arjuna.coordinator.BasicAction checkChildren
WARN: ARJUNA012099: Aborting child: 0xffff0a83801:cb8d:55ad2cb8:28
Jul 20, 2015 9:45:37 PM com.arjuna.ats.arjuna.coordinator.BasicAction checkIsCurrent
WARN: ARJUNA012099: BasicAction.checkIsCurrent 0xffff0a83801:cb8d:55ad2cb8:23 - terminating non-current transaction: 0xffff0a83801:cb8d:55ad2cb8:28
Jul 20, 2015 9:45:37 PM com.arjuna.ats.arjuna.coordinator.BasicAction Abort
WARN: ARJUNA012077: Abort called on already aborted atomic action 0xffff0a83801:cb8d:55ad2cb8:28
Transaction T1 : BasicAction: 0xffff0a83801:cb8d:55ad2cb8:8 status: ActionStatus.COMMITTED Counter : 1
Transaction T2 : BasicAction: 0xffff0a83801:cb8d:55ad2cb8:f status: ActionStatus.COMMITTED Counter : 1
Transaction T3 : BasicAction: 0xffff0a83801:cb8d:55ad2cb8:1e status: ActionStatus.COMMITTED Counter : 1
Transaction T4 : BasicAction: 0xffff0a83801:cb8d:55ad2cb8:23 status: ActionStatus.ABORTED Counter : 0
Transaction T5 : BasicAction: 0xffff0a83801:cb8d:55ad2cb8:14 status: ActionStatus.COMMITTED Counter : 1
Transaction T6 : BasicAction: 0xffff0a83801:cb8d:55ad2cb8:19 status: ActionStatus.COMMITTED Counter : 1
Transaction T7 : BasicAction: 0xffff0a83801:cb8d:55ad2cb8:28 status: ActionStatus.ABORTED Counter : 0

Process finished with exit code 0

```

Part II) Compensation for Chained Transactions

(ChainWithCompensate.java)

Like the above I use a Counter resource that is prepared with @Transactional.

First I begin the t1 transaction, increment the counter by one and then commit it. For beginning the t2 transaction, I check the status of t1 is committed or not. If it equals with committed then I will begin the t2 transaction, increment by one. After that I throw a RuntimeException and the catch section will execute a compensate and decrement the counter by one. Because the t2 can't commit the counter will be one and in compensate section I decrement by one to reach initial state of t1 or compensate t1.

```

Run ChainWithCompensate
"C:\Program ...
Jul 20, 2015 11:18:03 PM com.arjuna.ats.arjuna.recovery.TransactionStatusManager addService
INFO: ARJUNA012163: Starting service com.arjuna.ats.arjuna.recovery.ActionStatusService on port 52955
Jul 20, 2015 11:18:03 PM com.arjuna.ats.internal.arjuna.recovery.TransactionStatusManagerItem <init>
INFO: ARJUNA012337: TransactionStatusManagerItem host: 127.0.0.1 port: 52955
Jul 20, 2015 11:18:03 PM com.arjuna.ats.arjuna.recovery.TransactionStatusManager start
INFO: ARJUNA012170: TransactionStatusManager started on port 52955 and host 127.0.0.1 with service com.arjuna.ats.arjuna.recovery.ActionStatusService
t1 begun
t1 committed
t2 begun
t2 Not committing
Compensate t1
Counter : 0

Process finished with exit code 0

```