JUDCon

JBoss Users & Developers Conference

Boston:2011

Transacting the Cloud

Jonathan Halliday

JBossTS dev team lead

jonathan.halliday@redhat.com

www.jboss.org/jbosstm

jbossts.blogspot.com

Agenda

- Concepts & terminology
- Transactions in the Cloud
  - ACID
  - NoACID
- Questions

Agenda

- **Concepts & terminology**
- Transactions in the Cloud
  - ACID
  - NoACID
- Questions

What is a Cloud?

- public, private and hybrid clouds
- IaaS, PaaS, SaaS

- How is it payed for?
  - Pay as you go, pay for what you use
- How is it configured?
  - Abstraction layers are key

What is a Transaction?

- A structuring mechanism to provide a group of operations with common, shared properties.
- ACID transactions
  - Atomic, Consistent, Isolated, Durable
- NoACID transactions
  - Different properties for different use cases

ACID Transactions

- Traditional transaction model
  - XA protocol: TM, RMs, 2PC
- Widely supported
  - SQL databases and messaging systems
- Widely understood (more or less)
  - User API is simple
  - Characteristics are fairly intuitive

Agenda

- Concepts & terminology
- **Transactions in the Cloud**
  - **ACID**
  - NoACID
- Questions

ACID in the Cloud

- Used where traditional apps are run in the Cloud for economic efficiency
- Cloud = pay as you go, virtualized datacenter resources.
- Configuration must be carefully handled
  - Do you have sufficient control?
  - What are the cost implications?

Configuration & Control

- Shared resource managers
  - DBA policy and procedures
- Stateful nodes
  - Uniq identity, persistent storage
- Transaction recovery logs
  - Fast, fault tolerant storage

Isolation

- Dedicated resources
  - Or at least guaranteed QoS
- Pay for special consideration
  - e.g. hardware nodes with a SSD
- Pay for what you ~~use~~ stop others using

Technology roadmap

- Pluggable Objectstore
  - Transactions logs don't have to go to disk
- Testing tools
  - Byteman for fault injection
- Best Practice
  - Guides and consultancy

ACID in the Cloud: Summary

- Specific environmental requirements
  - Are they achievable?
  - At what cost?
- Testing is hard
  - Isolation and fault injection
- Plan ahead
  - Even from the design stage

Agenda

- Concepts & terminology
- **Transactions in the Cloud**
  - ACID
  - **NoACID**
- Questions

NoACID Transactions

- Cloud environments have more varied Resource Managers
  - In-memory data grids, NoSQL stores, large scale HA.
- We need more varied transactions
  - Extended transaction models
  - Polyglot transactions
  - Heterogeneous RMs in transactions

Cloud Resource Managers

- In-memory data grids
  - Memory is the new disk
  - Cache, in-process or out of process
    - Varied transactional requirements
  - Distributed, replicated 'store of record' needs XA
  - 2PC involves interposition, potentially large fan-out.

Cloud Resource Managers

- NoSQL datastores
  - Transaction support varies by design
    - CAP, BASE
  - Apps have varied storage, tactically mixing SQL and NoSQL
  - Implies a need for heterogeneous resources in a single transaction
    - Keep your SQL and NoSQL stores in sync

Consistency Rationing

- Consistency requires blocking on node availability
- Reducing consistency is preferred to reducing availability / partition tolerance.
- Programmers must learn to treat immediate consistency as an expensive resource and code accordingly

Extended Transactions

- Business Activities
  - Non-ACID transaction model for coordinating more loosely coupled processes
- WS-BA defines state machine and protocol for distributed communication over WS
- Not natively supported by current RMs

Extended Transactions Roadmap

- *-BA
  - Decouple from Web Services transport
- BA Framework Annotations
  - @CompensatedBy(method= “foo”)
  - Tied to state machine, not transport
  - Transactional workflow for POJOs
    - Individual steps may be ACID transactions

More Roadmap

- Delayed outcome handling
  - commit() is expensive, may take some time
  - Callback methods for outcome handling, state reconciliation
    - @CompensatedBy, @RollbackHander
- Non-ACID environments may require varying degrees of user supplied code
  - Reconciliation process is app specific

NoACID in the Cloud: Summary

- Mix and match varied transaction types, resource managers
- Trade-off immediate consistency for other desirable properties
- Be prepared to write code to assist the transaction system

Agenda

- Concepts & terminology
- Transactions in the Cloud
  - ACID
  - NoACID
- **Questions**

JUDCon

JBoss Users & Developers Conference

Boston:2011