Java Object Persistence

Rakesh Vidyadharan
rakesh@sptci.com

2008-05-20

Object Databases

Object Databases are systems that provide:

- Persistent store for objects and object graphs.
- Transactional semantics for object storage and retrieval.
- Query mechanism to retrieve objects.
- Transparent handling of references.

What is Prevayler?

- Not an object database.

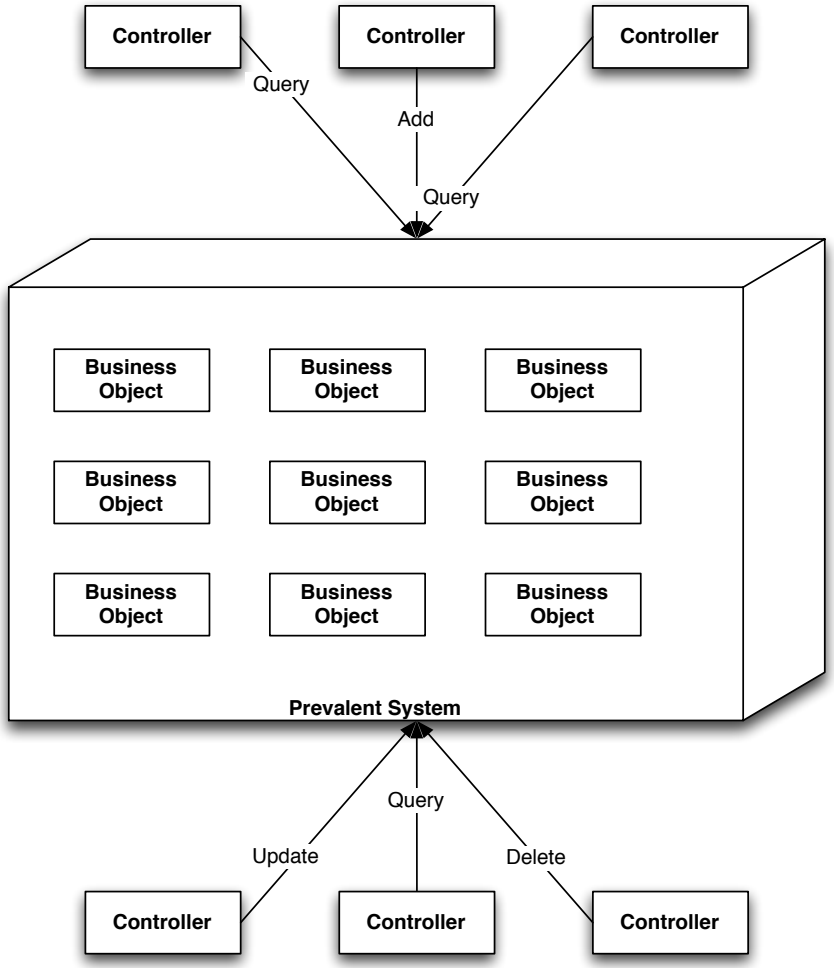
What is Prevayler?

- Not an object database.
- Transactional object serialization framework.
- Prevalent system is held entirely in memory.
- Prevalent system is updated through transactions.
- Transactions are serialized as journals.
- No rollback. Pre-check prior to updating system.
- Snapshot of entire object graph may be taken.
- System is restored by replaying journal files and snapshots in sequence.
- Prevalent system responsible for storage, indexing, ...

What is Prevayler?

- Not an object database.
- Transactional object serialization framework.
- Prevalent system is held entirely in memory.
- Prevalent system is updated through transactions.
- Transactions are serialized as journals.
- No rollback. Pre-check prior to updating system.
- Snapshot of entire object graph may be taken.
- System is restored by replaying journal files and snapshots in sequence.
- Prevalent system responsible for storage, indexing, ...
- You implement an object database using Prevayler.
- Available for non-Java platforms.

Prevalent System



PrevalentSystem.java

```
try
{
  final String directory = System.getProperty(
      DATA_DIRECTORY, DEFAULT_DIRECTORY );
  prevayler = PrevaylerFactory.createPrevayler(
      new TimeSystem(), directory );
}
catch ( Throwable t )
{
  throw new RuntimeException( t );
}
```

TimeSystem.java

```
class TimeSystem implements Serializable
{
   private long objectId;
   private Map<Long,Time> map =
     new LinkedHashMap<Long,Time>();
   public Time add( final Time time )
     throws DuplicateException
   {
     check();
     time.setObjectId( objectId++ );
     map.put( objectId, time );
     return time;
   }
}
```

AddTime.java

```
public class AddTime implements
  org.prevayler.TransactionWithQuery
{
 private final Time time;
 public AddTime( final Time time )
 {
  this.time = time;
 }
 public Object executeAndQuery(final Object system,
   final Date executionTime ) throws Exception
 {
  return ((TimeSystem)system).addTime( time );
 }
}
```

Client.java

```
import ...AddTime;
import ...PrevalentSystem;
public class Client
{
  public void add()
  {
    Time time = new Time();
    time.setXXX( ... );
    PrevalentSystem.getPrevayler().
      execute( new AddTime( time ) );
  }
}
```

What is a prevalent system?

- Provide storage mechanism for business objects
- Must be serializable.

What is a prevalent system?

- Provide storage mechanism for business objects
  - Business objects are stored in collections.
  - Indexing of query fields through maps.
  - Implement methods required to maintain (create, update, delete) and retrieve business objects. Maintenance methods are executed within the bounds of a transaction.
- Must be serializable.

What is a prevalent system?

- Provide storage mechanism for business objects
  - Business objects are stored in collections.
  - Indexing of query fields through maps.
  - Implement methods required to maintain (create, update, delete) and retrieve business objects. Maintenance methods are executed within the bounds of a transaction.
- Must be serializable.

Rules for Business Objects

- Must be serializable.
- Implement `hashCode` method.
- Object graphs only through key fields. Direct object references become local copies due to nature of serialization.

Transactions

- Must be serializable.
- Must be deterministic.

- No direct object references to business objects.

Transactions

- Must be serializable.
- Must be deterministic. Transactions must have the same effect on business objects when they are de-serialized as when they are serialized.
- No direct object references to business objects.

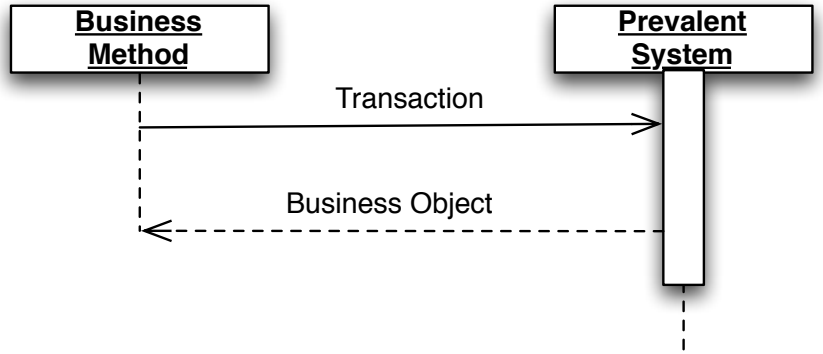
Transactions

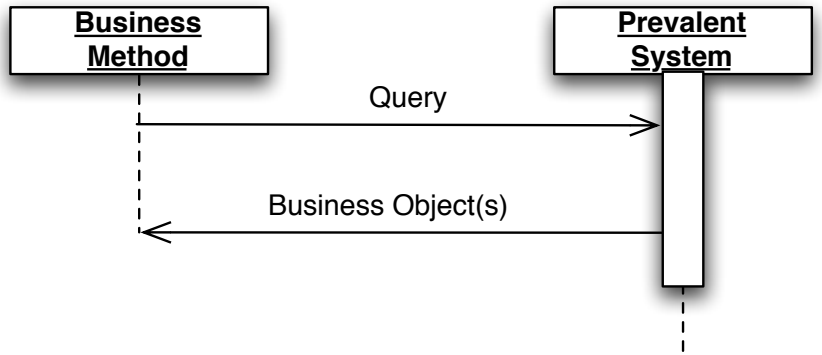
- Must be serializable.
- Must be deterministic. Transactions must have the same effect on business objects when they are de-serialized as when they are serialized.
- No direct object references to business objects.

Things to avoid

- `System.currentTimeMillis()`
- `Object.hashCode()` default implementation.
- Iterators of Hashtable, HashMap, or HashSet.
- No Random without a deterministic seed.
- No hardware or network access directly.

Interactions





When to use Prevayler

- Small data set. All data fits into memory.
- Business objects are simple without too many interdepencies.
- Simple querying requirements.
- Application start-up time is not critical.
- You require high degree of fault tolerance over and above direct object serialization.

Disadvantages

- All data must fit into memory.
- Business objects cannot hold direct business object references.
- Requires creation of transaction and query objects.

- Long start-up time while prevalent system is reinstated from snapshot and journals.
- Storage and indexing must be handled in implemented prevalent system.
- No schema evolution.

Disadvantages

- All data must fit into memory.
- Business objects cannot hold direct business object references.
- Requires creation of transaction and query objects.
  - Dynamic proxies may be used to represent transaction and query objects.
  - Related project Finevayler available that removes some of these inconveniences.
- Long start-up time while prevalent system is reinstated from snapshot and journals.
- Storage and indexing must be handled in implemented prevalent system.
- No schema evolution.

What is ObjectDB?

- A pure Java Object Database.
- Embedded or client-server modes.
- JDO 2.1 and JPA 1.0 (partial) compliant.

- Seamlessly persist and retrieve complex object graphs.
- No huge memory requirements.
- Performance many times that of MySQL/Oracle/...
- ObjectDB Explorer tool provides database management features.
- Very responsive support.

What is ObjectDB?

- A pure Java Object Database.
- Embedded or client-server modes.
- JDO 2.1 and JPA 1.0 (partial) compliant. Production version 1.04 supports only JDO 1.0, while version 2.0 is still in beta. Backup and recovery, index rebuilding, managed relationships etc are still not supported.
- Seamlessly persist and retrieve complex object graphs.
- No huge memory requirements.
- Performance many times that of MySQL/Oracle/...
- ObjectDB Explorer tool provides database management features.
- Very responsive support.

Advantages of Object Databases

- Seamlessly persist complex object graphs.
- Performance orders of magnitude higher than relational databases.
- Extremely efficient handling of object references.

- Domain specific object modelling.
- No OR mapping - quick development cycle.
- Programmer's system, no DBA's required.

Advantages of Object Databases

- Seamlessly persist complex object graphs.
- Performance orders of magnitude higher than relational databases.
- Extremely efficient handling of object references.
  - Composite objects may be used to achieve extreme performance.
  - No join queries required.
- Domain specific object modelling.
- No OR mapping - quick development cycle.
- Programmer's system, no DBA's required.

Disadvantages of Object Databases

- Restricted query language.

- No bulk data loaders and transformers.
- Cumbersome data management processes.
- No stored procedures.
- No job security.

Disadvantages of Object Databases

- Restricted query language.
  - Poor join performance.
  - Join queries are used to fetch loosely related objects.
  - Poor performance of ad-hoc queries.
- No bulk data loaders and transformers.
- Cumbersome data management processes.
- No stored procedures.
- No job security.

Links

- Prevayler
- Finevayler
- ObjectDB
- db4o - JPOX provides an adapter for db4o.
- MyOODB
- JODB