

Bachelor of Information Technology

Middleware Architecture
IT6505



Teacher's Notes

Lesson 09



Viraj Brian Wijesuriya

University of Colombo School of Computing

No. 35, Reid Avenue , Colombo 7, Sri Lanka.

Message Oriented Middleware (1)

- Message-oriented middleware (MOM) is software or hardware infrastructure supporting sending and receiving messages between distributed systems.
- MOM allows application modules to be distributed over heterogeneous platforms and reduces the complexity of developing applications that span multiple operating systems and network protocols.
- The middleware creates a distributed communications layer that insulates the application developer from the details of the various operating systems and network interfaces.

Message Oriented Middleware (2)

- APIs that extend across diverse platforms and networks are typically provided by MOM.
- MOM comprises a category of inter-application communication software that generally relies on asynchronous message-passing, as opposed to a request-response architecture.
- In asynchronous systems, message queues provide temporary storage when the destination program is busy or not connected.

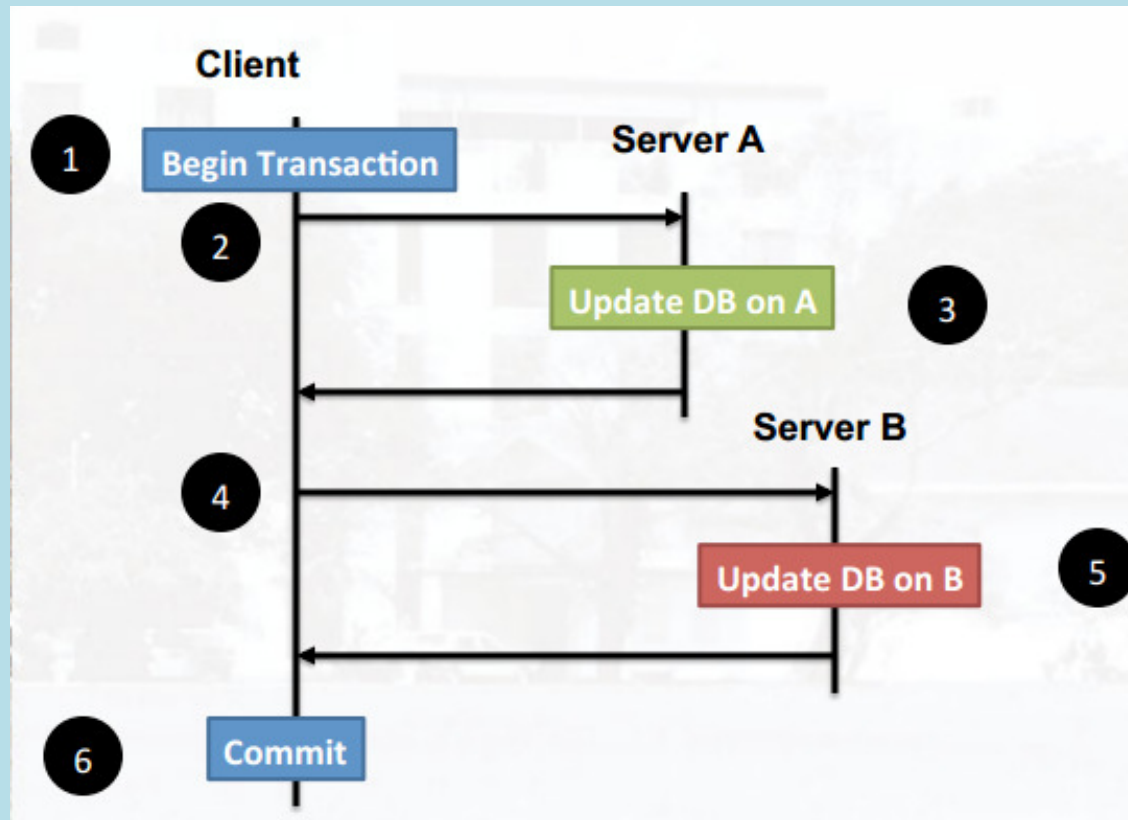
Transaction Processing

- A transaction is a unit of work that updates a database (and/or maybe other resources) .
- Transaction are either completed (committed) or completely undone.
- Distributed transaction processing is about having more than one database participate in one transaction.

Distributed Transaction Processing (1)

- Need protocols
- Define an SQL table on one system, and have it implemented by remote access to a table on a different database
- Horrendous performance & resiliency issues
- Some technologies developed are excellent for report generating and decision support, but not accurate for transaction support

Distributed Transaction Processing (2)



Question 1 (1)

```
UPDATE scott.dept@sales.us.americas.acme_auto.com
  SET loc = 'NEW YORK'
  WHERE deptno = 10;
UPDATE scott.emp
  SET deptno = 11
  WHERE deptno = 10;
COMMIT;
```

- What does this code segment do? Is this remote transactions processing?

Question 1 (2)

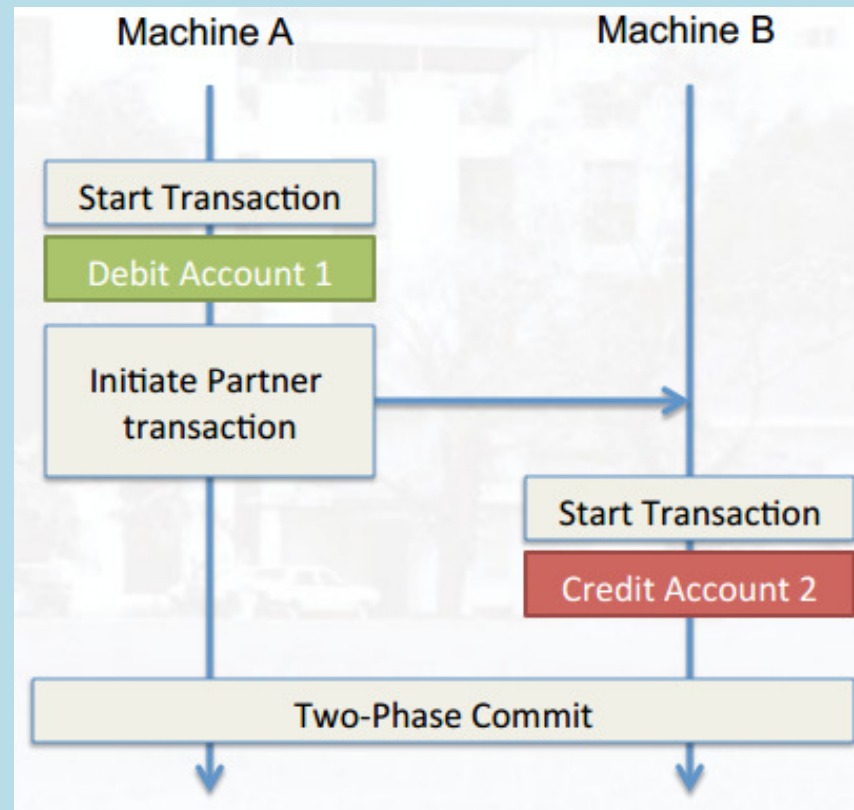
```
UPDATE scott.dept@sales.us.americas.acme_auto.com
  SET loc = 'NEW YORK'
  WHERE deptno = 10;
UPDATE scott.emp
  SET deptno = 11
  WHERE deptno = 10;
COMMIT;
```

- this transaction updates the local database and the remote sales database
- If all statements of a transaction reference only a single remote node, the transaction is remote, not distributed.
- This is a distributed transaction processing.

Message Queueing vs Distributed Transaction

- A complete distributed transaction processing environment can be built using message queueing.
- Technology is Different.
- How is this possible?
- Example Suppose a Person is moving money from account A to account B.

Done with Distributed Transaction

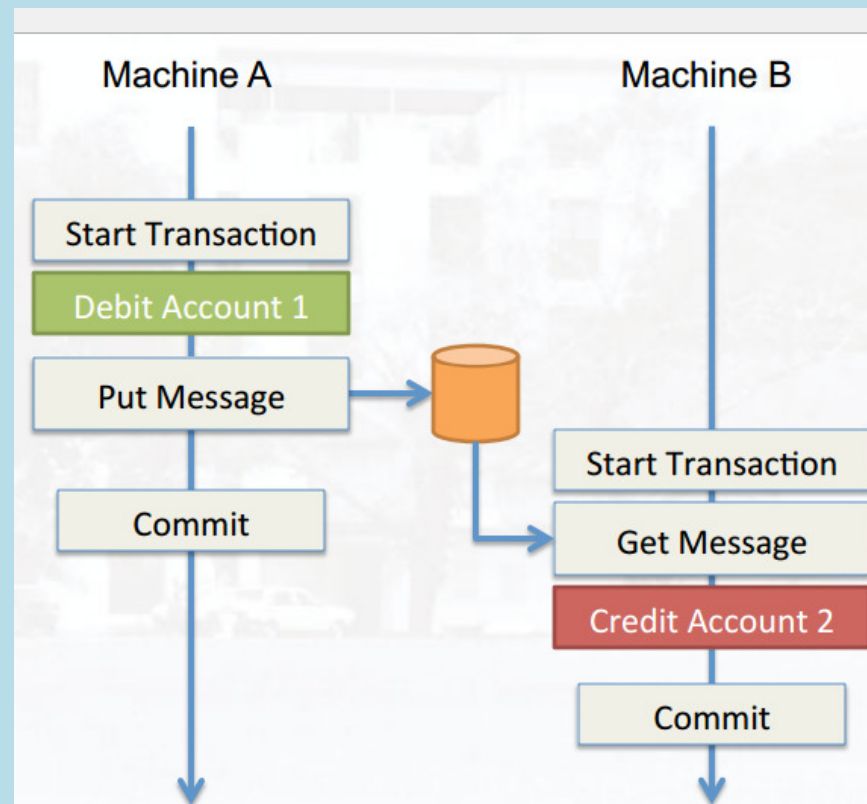


What is the problem with this setup?

Problems with the Previous Implementation

- Performance is degraded because of the overhead of sending additional messages for two-phase commit.
- If either system is down or the network is down, the transaction cannot take place.

Done with Message Queueing

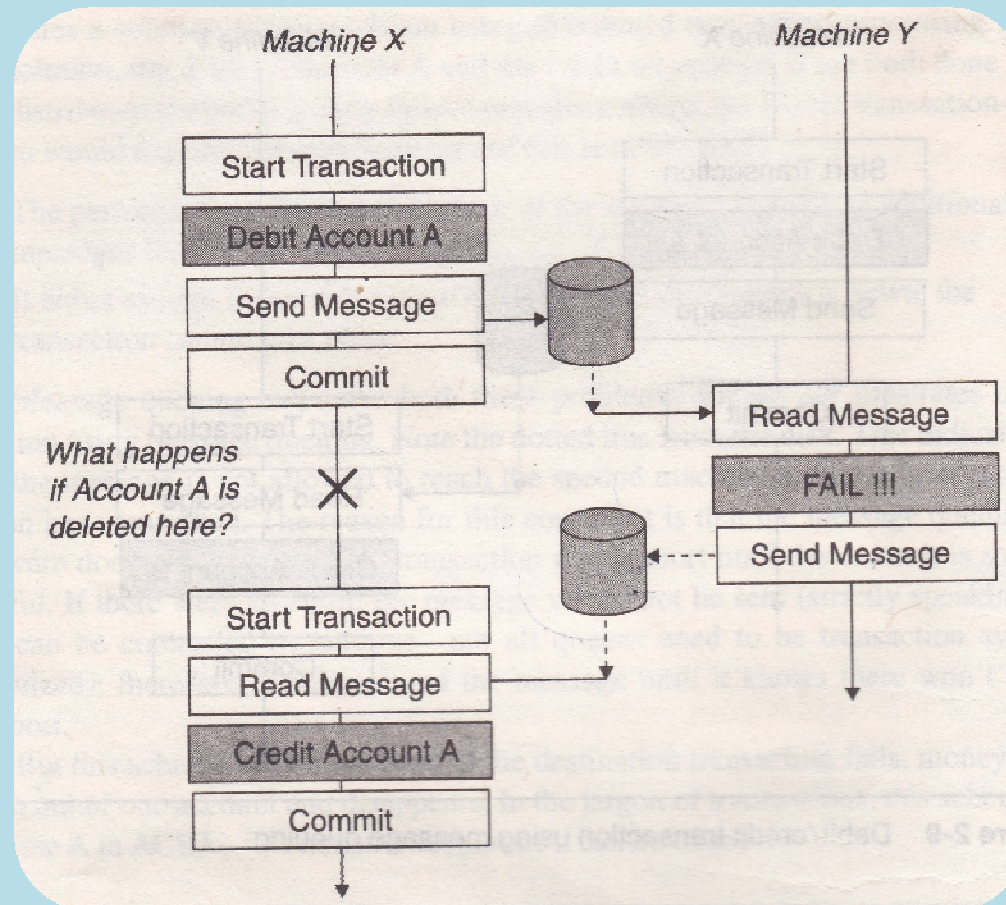


What is the problem with this setup?

Problems with the Previous Implementation

- If the destination transaction fails, money is taken out of one account and disappears.
- The solution is to have reversal transaction.
- The bank can reverse the failed transaction by having a credit transaction for the same amount.

Solution with Reversal



Question 2

- What is the problem with the debit/credit transaction with reversal solution?
- It fails if the account A is deleted before the reversal takes effect.
- Solution: ensure that the accounts are not deleted until all monetary flows have been completed.
- Having a account status “in the process of being deleted”.
- This is rather a business choice.

Amazon Cloud

- Amazon Web Services (AWS), is a subsidiary of Amazon.com, which offers a suite of cloud computing services that make up an on-demand computing platform.
- These services operate from 13 geographical regions across the world.
- The most central and best-known of these services arguably include Amazon Elastic Compute Cloud, also known as "EC2", and Amazon Simple Storage Service, also known as "S3".
- AWS now has more than 70 services that span a wide range including compute, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools and tools for the Internet of things.

Amazon SQS (1)

- Part of Amazon AWS.
- Amazon Simple Queue Service (Amazon SQS) is a messaging queue service: it's a service that handles message or work flows between other components in a system.
- Massive message processing.
- Fast, reliable, scalable, fully managed message queuing service. SQS makes it simple and cost-effective to decouple the components of a cloud application.

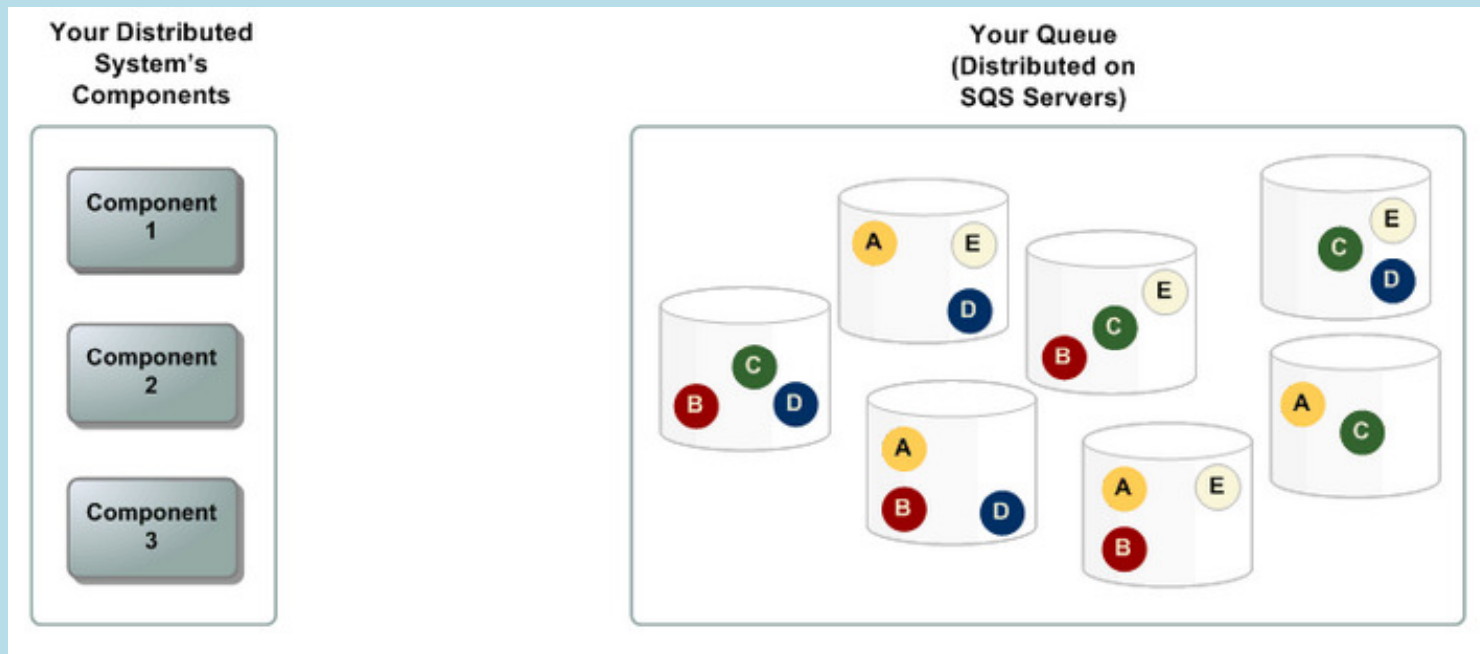
Amazon SQS (2)

- With SQS, you can offload the administrative burden of operating and scaling a highly available messaging cluster, while paying a low price for only what you use.
- **Reliable:** Amazon SQS runs within Amazon's high-availability data centers, so queues will be available whenever applications need them. To prevent messages from being lost or becoming unavailable, all messages are stored redundantly across multiple servers and data centers.

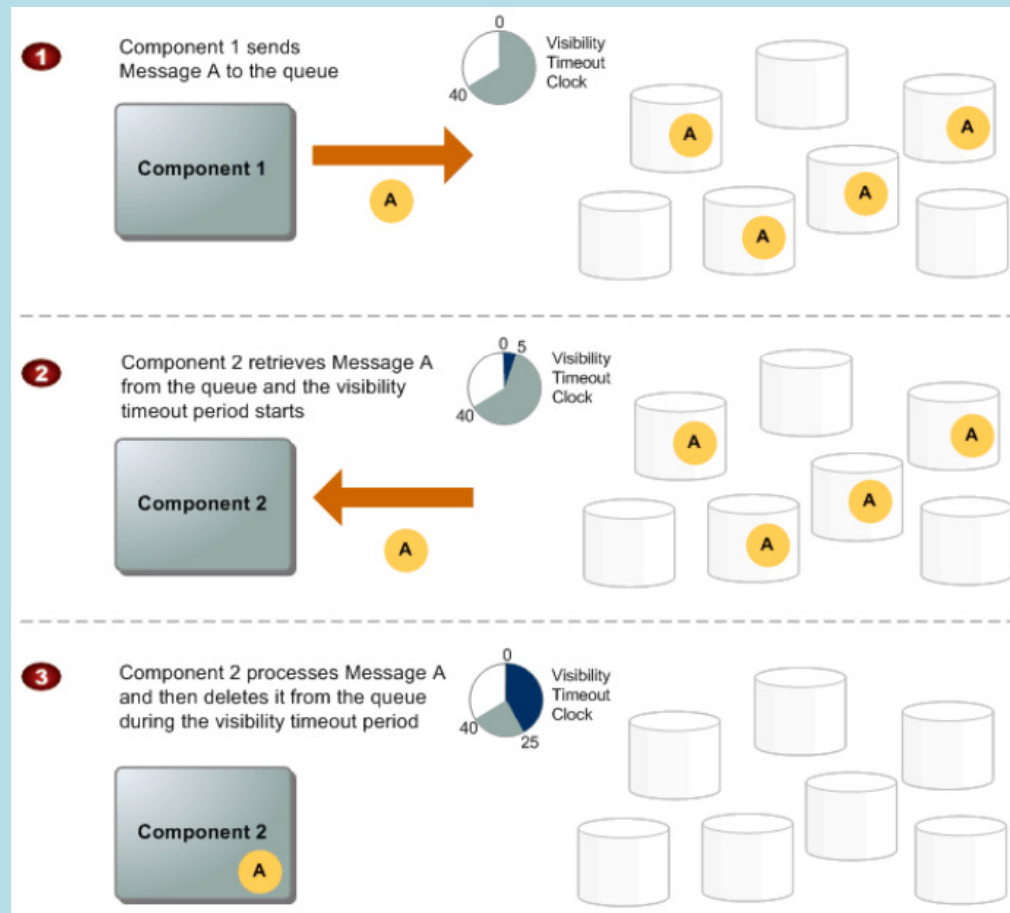
Amazon SQS (3)

- **Simple:** Developers can get started with Amazon SQS by using only three APIs: SendMessage, ReceiveMessage, and DeleteMessage. Additional APIs are available to provide advanced functionality.
- **Scalable:** Amazon SQS was designed to enable an unlimited number of messaging services to read and write an unlimited number of messages at any time.
- **Secure:** Authentication mechanisms are provided to ensure that messages stored in Amazon SQS queues are secured against unauthorized access.

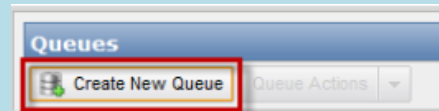
Amazon SQS Architecture



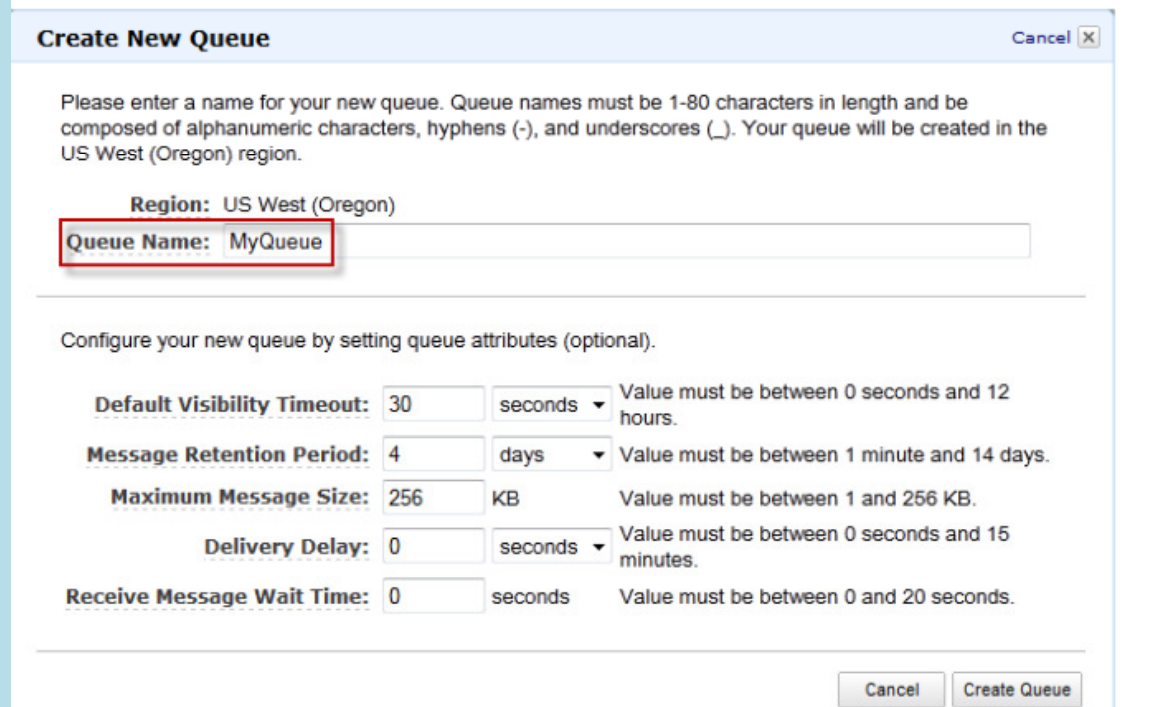
Amazon SQS Message Life Cycle



Amazon SQS - Create a Queue



In the **Create New Queue** dialog box, enter `MyQueue` in the **Queue Name** field, and leave the default value settings for the remaining fields.

A screenshot of the 'Create New Queue' dialog box. The dialog has a title bar with 'Create New Queue' and a 'Cancel' button. The main text says: 'Please enter a name for your new queue. Queue names must be 1-80 characters in length and be composed of alphanumeric characters, hyphens (-), and underscores (_). Your queue will be created in the US West (Oregon) region.' Below this, the 'Region' is set to 'US West (Oregon)'. The 'Queue Name' field is highlighted with a red rectangle and contains the text 'MyQueue'. Below the queue name field, there is a section titled 'Configure your new queue by setting queue attributes (optional)'. This section contains five attributes, each with a text input, a unit dropdown, and a validation message: 'Default Visibility Timeout' (30 seconds, Value must be between 0 seconds and 12 hours), 'Message Retention Period' (4 days, Value must be between 1 minute and 14 days), 'Maximum Message Size' (256 KB, Value must be between 1 and 256 KB), 'Delivery Delay' (0 seconds, Value must be between 0 seconds and 15 minutes), and 'Receive Message Wait Time' (0 seconds, Value must be between 0 and 20 seconds). At the bottom right, there are 'Cancel' and 'Create Queue' buttons.

Question 3

- In Amazon SQS, what is the difference between the 'message retention period' and the 'Visibility Timeout'?
- SQS automatically deletes messages that have been in a queue for more than maximum message retention period. The default message retention period is 4 days . However , you can set the message retention period to a value from 60 seconds to 1209600 seconds (14 days) with `SetQueueAttributes` .
- While Message A is being processed, it remains in the queue and is not returned to subsequent receive requests for the duration of the visibility timeout.

Lesson Summary

- Message Oriented Middleware
- Transaction Processing
- Distributed Transaction Processing
- Message Queueing vs Distributed Transaction Processing
- Massive Message Processing with Amazon SQS