Both WebSphere Application Server clusters and WebSphere MQ clusters are widely used to provide high availability and balance workloads across an enterprise. Combining these technologies helps you scale both your WebSphere Application Server and WebSphere MQ infrastructures, and enables WebSphere Application Server applications to use the faster bindings mode to retrieve messages from WebSphere MQ queues. However, without careful planning, the combination of WebSphere Application Server clusters and WebSphere MQ clusters can actually cause reduced overall availability, message build-up on queues, and stranded messages that must be processed manually. You can avoid these problems by using the options in this article for integrating WebSphere Application Server and WebSphere MQ clusters.

Introduction to WebSphere Application Server clusters

IBM® WebSphere® Application Server clusters are groups of application servers within a cell, which is a collection of application servers managed by a single instance of WebSphere Deployment Manager. WebSphere Application Server clusters are a key component of the product's high availability framework.

When an application is installed to the cluster, the application is automatically installed on each application server (cluster member) within the cluster, and therefore each cluster member contains the same applications. Figure 1 below shows an example of a WebSphere Application Server cluster. Application ABC runs on JVM_1, JVM_2, and JVM_3, and processes messages from the same WebSphere MQ queue, which means that Queue_Manager_1 is a potential single point of failure.
Cluster members can be hosted on the same node (usually a physical or virtual computer system or logical partition with a distinct host IP address) or distributed across multiple nodes. In Figure 1, the cluster members are split between Nodes 1 and 2. Clusters are responsible for balancing workload among cluster members. On distributed platforms, client tasks can be distributed according to the capacities of the different machines by assigning weights to each server, which can be used to improve performance and fail-over.

**Introduction to WebSphere MQ clusters**

WebSphere MQ enables you to distribute messages destined for the same application queue across a group of queue managers referred to as a WebSphere MQ cluster. This technique enables you to scale an application's workload across queue managers, and it increases the availability of the environment by ensuring that messages are sent only to available queue managers and queues.

Figure 2 shows an example of a small WebSphere MQ cluster consisting of three queue managers. A queue called Clustered_Queue has been defined on Queue_Manager_1 and Queue_Manager_2. Messages coming into the cluster via Queue_Manager_3 are distributed between these two instances of the clusters queue by the clusters workload balancing algorithm. You can configure this workload balancing algorithm to suit your needs -- for example, you can send the majority of messages to the queue manager with the largest processing power:
Combining WebSphere MQ and WebSphere Application Server clusters.

It is possible to combine a WebSphere MQ cluster with a WebSphere Application Server cluster, as shown in Figure 3 below. Queue_Manager_1 and Queue_Manager_2 are now defined locally on the same nodes as the WebSphere Application Servers JVMs. Each application server JVM can access a local instance of the queue Clustered_Queue. This technique provides a more scalable architecture, because extra queues and JVMs can be added as workload increases, single points of failure are reduced, and the JVMs can connect using the higher-performing bindings mode to the local queue manager.
However, there are some drawbacks to this approach. In Figure 3, only one instance of application ABC is taking messages from the instance of Clustered_Queue on Queue_Manager_1. If JVM_1 fails, as shown in Figure 4, then the messages on that instance of the cluster queue will not be available for processing by another application server cluster member, and those messages will be stranded on the queue until the application server JVM is configured to read from this queue is restarted. Manual intervention is also required to prevent new messages being distributed to this queue, because the WebSphere MQ clustering workload management routine is by default based on queue manager availability, and not on the application server’s availability to consume the messages.
Figure 4. Stranded messages on a WebSphere MQ cluster queue

For a more detailed description of this scenario and the manual steps you need to take if an application server fails, search for `cmm_mq_top02_clustered` in the relevant WebSphere Application Server information center in the Related topics section at the bottom of the article.

**Options for combining WebSphere MQ and WebSphere Application Server clusters to maintain a highly available environment**

Two methods described in this article will enable you to effectively combine these technologies to make use of the high availability features of WebSphere Application Server clusters, and the workload balancing features of WebSphere MQ clusters:

1. Monitoring the availability of the WebSphere Application Server JVM using WebSphere MQ tools
2. Vertical scaling of WebSphere Application Server

**1. Monitoring the availability of the WebSphere Application Server JVM using WebSphere MQ tools**

If you have a planned or unplanned outage of your application server JVM, you need a strategy to both deal with the messages that have already arrived on the WebSphere MQ queue it is consuming from, and to prevent further messages from being sent to that queue as part of the WebSphere MQ cluster workload balancing feature.

In the combined cluster architecture shown in Figure 3 above, the instance of application ABC running on JVM_1 is the only process reading messages from the instance of the Clustered_Queue on Queue_Manager_1. If JVM_1 fails then you need to detect this failure to prevent further messages from being sent to this instance of the clustered queue, and also to transfer any messages that have already been sent to the queue but not yet processed.
For approaches you can take to address both of these requirements, see the developerWorks article Enhanced WebSphere MQ Cluster workload balancing with Omegamon XE for Messaging and IBM Tivoli Monitoring.

### 1.1. Implementing the sample program AMQSCLM

WebSphere MQ V7.0.1.8 or later provides a cluster queue monitoring sample program called AMQSCLM. This tool uses the built-in WebSphere MQ cluster workload balancing features to direct messages to instances of queues that have a consuming application attached. You can configure this tool to monitor each instance of the cluster queue that holds messages for retrieval by WebSphere Application Server. If it detects that the application server JVM is no longer attached to the queue to process messages, the tool can automatically redirect messages to other instances of the queue with active consumers. It can also automatically transfer messages that have already arrived on the queue to other instances:

**Figure 5. Implementing AMQSCLM on a WebSphere MQ cluster**

In the combined cluster architecture shown in Figure 5, AMQSCLM needs to run on both Queue_Manager_1 and Queue_Manager_2 to monitor both instances of Clustered_Queue. If JVM_1 fails, then AMQSCLM detects that it no longer has an open connection to the instance of Clustered_Queue on Queue_Manager_1 and it then reconfigures the workload balancing algorithm to direct all messages to Queue_Manager_2 until JVM_1 is running again. It also redistributes the messages that have already arrived on Clustered_Queue on Queue_Manager_1 to the instance of the queue on Queue_Manager_2.

For more information on how to design and implement AMQSCLM, see The Cluster Queue Monitoring sample program AMQSCLM in the WebSphere MQ information center. When starting the sample, use the –t flag to transfer messages that have already been queued to the instance of the cluster queue to instances of the clustered queue with an application server JVM attached.
2. Vertical scaling of WebSphere Application Server

Probably the simplest change to the architecture shown in Figure 3 earlier in the article is to use WebSphere Application Server vertical scaling. Since the MQ resource Queue_Manager_1 is tied to Node 1, simply replicating the application server architecture of Node 2 onto Node 1 by adding a fourth application server in the cluster (JVM_4 in Figure 6) enables that JVM to process the clustered queue on Queue_Manager_1 when it is started:

Figure 6. Vertical scaling of WebSphere Application Server

Should one of the application servers on Node 1 fail, then the other application server on that node takes over reading from the Clustered_Queue on Queue_Manager_1:
Deciding which approach to use

- If you decide to use a WebSphere MQ monitoring solution, the primary responsibility for monitoring the application server’s availability to consume messages shifts from the WebSphere Application Server administrators to the WebSphere MQ administrators. You should decide whether this change is desirable in your enterprise.
- With any approach, if the application server stops consuming message because bad data has arrived on the queue, you need to ensure that the method you have chosen can respond appropriately. For more information, see the developerWorks article *How WebSphere Application Server handles poison messages*.
- With any approach, if the queue manager hosting the clustered queue fails, any messages already on the queue will not be available for consumption until the queue manager is restarted. Implementing a WebSphere MQ high availability solution such as multi-instance queue managers avoids this problem.
- With any approach, when designing your solution, ensure that the JVMs that you are directing the workload to in the case of a failure can handle the additional workload.

Acknowledgements

- The authors would like to thank Matthew White, WebSphere MQ JMS Technical Lead, for his help in contributing content to and reviewing this article.
- The authors would also like to thank Graham Hopkins, WebSphere Messaging Test team, for contributing content to this article.
Related topics

- **WebSphere Application Server resources**
  - WebSphere Application Server V7 information center
  - WebSphere Application Server V8 information center
  - WebSphere Application Server V8.5 information center
  - WebSphere Application Server developer resources page
    Technical resources to help you use WebSphere Application Server.
  - WebSphere Application Server product page
    Product descriptions, product news, training information, support information, and more.
  - WebSphere Application Server information roadmaps
    Roadmap of articles and resources to help you with installation, migration, administration, development, troubleshooting, and understanding the underlying technology.
  - WebSphere Application Server documentation library
    WebSphere Application Server product manuals.
  - WebSphere Application Server support
    A searchable database of support problems and their solutions, plus downloads, fixes, and problem tracking.
  - Download a free trial version of WebSphere Application Server V7
    WebSphere Application Server V7 is a Java EE 5 certified, EJB 3.0 supported application platform that drives business agility with an innovative, performance based foundation for your SOA environment on the broadest range of platforms in the industry.

- **WebSphere MQ resources**
  - WebSphere MQ V7.0.1 information center
  - WebSphere MQ V7.1 information center
  - WebSphere MQ V7.5 information center
  - WebSphere MQ developer resources page
    Technical resources to help you design, develop, and deploy messaging middleware with WebSphere MQ to integrate applications, Web services, and transactions on almost any platform.
  - WebSphere MQ product page
    Product descriptions, product news, training information, support information, trial download, and more.
  - WebSphere MQ product family
    A description of the ten or so different editions of WebSphere MQ.
  - WebSphere MQ documentation library
    WebSphere MQ information centers and product manuals.
  - IBM Redbook: WebSphere MQ V7 features and enhancements
    Describes the fundamental concepts and benefits of message queuing technology, describes the new features in V7, and provides a business scenario that shows those features in action.
  - Download a free trial version of WebSphere MQ V7
    A 90-day, full featured, no-charge trial of WebSphere MQ V7
  - WebSphere MQ support page
Maintaining high availability when implementing WebSphere MQ clusters in a clustered WebSphere Application Server environment.

A searchable database of support problems and their solutions, plus downloads, fixes, and problem tracking.

- **WebSphere MQ forum**
  Get answers to your WebSphere MQ technical questions and share your knowledge with other users.

- **WebSphere MQ SupportPacs**
  Downloadable code, documentation, and performance reports for the WebSphere MQ family of products.

- **WebSphere resources**
  - **developerWorks WebSphere developer resources**
    Technical information and resources for developers who use WebSphere products. developerWorks WebSphere provides product downloads, how-to information, support resources, and a free technical library of more than 2000 technical articles, tutorials, best practices, IBM Redbooks, and online product manuals. Whether you’re a beginner, an expert, or somewhere in between, you'll find what you need to build enterprise-scale solutions using the open-standards-based WebSphere software platform.
  - **developerWorks WebSphere application integration developer resources**
    How-to articles, downloads, tutorials, education, product info, and other resources to help you build WebSphere application integration and business integration solutions.
  - **Most popular WebSphere trial downloads**
    No-charge trial downloads for key WebSphere products.
  - **WebSphere forums**
    Product-specific forums where you can get answers to your technical questions and share your expertise with other WebSphere users.
  - **WebSphere on-demand demos**
    Download and watch these self-running demos, and learn how WebSphere products and technologies can help your company respond to the rapidly changing and increasingly complex business environment.
  - **WebSphere-related articles on developerWorks**
    Over 3000 edited and categorized articles on WebSphere and related technologies by top practitioners and consultants inside and outside IBM. Search for what you need.
  - **developerWorks WebSphere weekly newsletter**
    The developerWorks newsletter gives you the latest articles and information only on those topics that interest you. In addition to WebSphere, you can select from Java, Linux, Open source, Rational, SOA, Web services, and other topics. Subscribe now and design your custom mailing.
  - **WebSphere-related books from IBM Press**
    Convenient online ordering through Barnes & Noble.
  - **WebSphere-related events**
    Conferences, trade shows, Webcasts, and other events around the world of interest to WebSphere developers.

- **developerWorks resources**
  - **Trial downloads for IBM software products**
    No-charge trial downloads for selected IBM® DB2®, Lotus®, Rational®, Tivoli®, and WebSphere® products.