Disaster recovery guidance for IBM Business Process Manager

An updated approach for IBM BPM V8.x

Chris Richardson
Eric Herness

April 30, 2015

IBM® Business Process Manager (BPM) is a powerful tool for modeling and running an organization's most critical business processes. Over the past several years, features have been developed and verified to enable cross-site replication and recovery, achieving very sophisticated disaster-recovery objectives. Learn about the core principles that guide an infrastructure architect to design a successful disaster-recovery strategy.

Introduction

Business process management is a concept and an approach for running business transactions that continues to gain momentum. A full-powered business process management system delivers value to stakeholders in an agile, flexible, and consist way. Your organization might be just starting to apply IBM BPM to your business, so you might not have encountered the need for robust disaster recovery solutions. However, many organizations are already hosting mission-critical applications on their IBM BPM systems. If your organization is one of these, then you already understand the costs associated with the loss of your IBM BPM system in the event of a disaster and the procedures for recovering business operations after a disaster occurs.

Common guidance for using a business process management system is to focus on high-value processes that provide a business competitive advantage or unique value in the marketplace. Whether the advantage is cost effectiveness, customer retention, or another measure of business value, business processes often need to be highly available and protected by increasingly aggressive disaster recovery requirements.

It's understood that disaster recovery requirements are measured by recovery time objectives (RTO) and recovery point objectives (RPO). In recent years, disaster recovery requirements for IBM BPM systems have moved beyond the traditional RTO and RPO numbers to much more aggressive goals. The aspiration to recover and be back in business in 30 minutes, for example, has led large organizations to pursue new and additional ways of approaching disaster recovery.
Charlie Redlin's seminal work *Asynchronous replication of WebSphere Process Server and WebSphere Enterprise Service Bus for disaster recovery environments*, published on developerWorks in 2008, established the baseline for disaster recovery strategies, defining an approach that as come to be known as "classic disaster recovery" because of its widespread appeal and success. The strategy Charlie describes remains popular among IBM BPM customers and relies on techniques referred to as cloned cell and storage managed replication in the sections that follow. Hong Yan Wang and others describe an alternative approach in *Storing transaction and compensation logs in a relational database for high availability and disaster recovery in IBM Business Process Manager*, published on developerWorks in 2014. Her approach uses new WebSphere Application Server capabilities and new topological constructs to approach the same technical problems that Charlie Redlin did six years earlier. Which approach is better? Which should you use as the basis for you own disaster recovery design? The answer, of course, is that it depends on a number of factors. The first step is to review key concepts in disaster recovery that matter to all IBM BPM workloads and the key challenges brought about by the stateful nature of workloads commonly managed by IBM BPM systems. The second step is to notice common patterns in successful disaster recovery solutions for IBM BPM in order to gain an understanding of some of the advantages and disadvantages of each. Armed with these two pieces of information, an understanding of the challenges to be addressed and common techniques for addressing them, you as the IBM BPM administrator can begin the process of designing a disaster recovery strategy that meets your business needs.

You should continue reading if you want to explore IBM BPM as a platform on which to run a comprehensive and exhaustive set of business processes, including the most mission-critical and business-critical processes in an organization. You might be an application architect, enterprise architect, or senior business professional and familiar with the business-driven requirements for maintaining availability in the presence of disasters. The techniques discussed apply to IBM BPM versions 8.0.1.3, 8.5.0.1, 8.5.5.0, 8.5.6, and 8.5.7.

**Key concepts in IBM BPM disaster recovery**

Business process management platforms consist of hardware and software components working together, any of which might fail. One goal of the system architect and system administrator is to ensure that there is sufficient redundancy so that business activities can continue, even when information technology resources fail and while they are repaired or replaced.

For example, physical disk drives are grouped into arrays that are governed by a controller that distributes data across the group and automatically redirects traffic if one of the disks fails. Similarly, IBM WebSphere Network Deployment groups application servers into clusters and coordinates communication among them so that the group can tolerate losing any member. Such building blocks comprise the tool set used to make a business process management system highly available and to successfully use this system to automatically accommodate losing any single element. These two examples address high availability events, that is, events comprising the failure of a single component of a larger system. In contrast, a disaster is a catastrophic event comprising simultaneous loss of multiple components of an entire data center. Nonetheless, the core concept of redundancy can be extended to include replication of key product components and
runtime data across data center boundaries so that business operations can continue, even if the entire primary data center is lost.

One important consideration in designing a disaster recovery strategy is that, when doing so, the infrastructure architect must assume that the primary storage used to persist the business state is lost. Therefore, in addition to replacing all relevant hardware and software components of the business process management system, a disaster recovery solution must replace the data that represents the business state. This is handled by using replication: a copy of the data is kept (and kept up to date) at a location sufficiently separated from the primary data center to ensure that both sets are not affected by the same disaster. Replication is particularly important for business process management systems, because they manage the data that is critical to maintaining the state of the activities upon which the business relies, often coordinating activities across multiple data storage systems. To better understand replication requirements, divide the data to be replicated into two categories: configuration data and runtime data.

**Configuration data**

As its name suggests, configuration data consists of all the metadata required to define the state of a business process management system, including the definition of the IBM WebSphere Network Deployment topology on which IBM BPM is installed and all of the information needed to connect to external resources and data sources. If you are familiar with IBM WebSphere Application Server, you can think of configuration data as the contents of the IBM WebSphere Application Server profile. In addition, IBM BPM configuration data also includes the metadata that defines the business processes that are built in the authoring environment and deployed to runtime servers. This process definition metadata is in both the WebSphere Application Server profile and the IBM BPM database.

For the purposes of replication, an important feature of configuration data is that discrete, isolated events change it. For example, changes to the profile information made in the administrative console, applying code changes to process applications, or upgrading IBM BPM, change the configuration data. Usually, operational procedures gather these types of changes to configuration data into groups that are applied to the system together, perhaps during a period of maintenance. Therefore, use a discrete replication technique for configuration data, collecting a snapshot of the configuration state after each group of changes is applied and confirmed to produce the effect you want, which results in a stable system.

**Runtime data**

In contrast to configuration data, runtime data consists of all the information that describes the process instances that the business deals with. Metadata describing the current state of process instances, including which tasks are currently active and the values of the associated business data variables, is stored in the IBM BPM database.

Because IBM BPM uses messaging engines for both internal and external communication, runtime data also includes the information stored in the messages on the queues. IBM BPM allows application developers to integrate their business processes with application databases, maintaining consistency by using Java Database Connectivity (JDBC) communication built into the
business process implementation. The content of these application databases, if they exist, is also included in the scope of runtime data.

Finally, IBM BPM relies on the WebSphere Application Server transaction service to coordinate data changes and group them into atomic units, ensuring that either all the changes happen or that none of them do. So, the log files associated with the WebSphere Application Server transaction service are a critical piece of runtime data, linking IBM BPM product resources and application-specific resources.

For the purposes of replication, an important characteristic of runtime data is that it changes continuously. Therefore, it is natural to replicate runtime data by using a continuous strategy, whether that strategy is synchronous (each change committed to the source is guaranteed to also be committed to the replica) or asynchronous (changes to the source can be sent in batches and copied to the replica periodically). Either way, the replication does not resemble a series of discrete snapshots but a steady stream of information passing from the source to the replica. Typically, synchronous replication techniques are not ideal for disaster recovery, because they impose unacceptable performance costs when applied across sites that are separated geographically. Therefore, asynchronous replication is usually preferred for runtime data.

Although preferable (and many times mandatory) from a performance perspective, replicating runtime data asynchronously introduces a challenge for consistency. Because runtime data is distributed across multiple resources, at any point in time, an asynchronously updated replica of one of those resources might be inconsistent with the independent asynchronous replica of another resource. At recovery time, this inconsistency causes corruption that might result in the loss of business data or prevent locked resources (such as a database row) from being released until transactions time out.

Many storage-based replication management technologies (for instance, those provided by a storage area network) provide a feature called consistency groups to address this issue. With a consistency group, you can group a collection of physically and logically distinct storage volumes for replication. In this way, the replication manager can ensure write-order consistency is preserved across all the volumes, even when using asynchronous replication. So, even though the image at the replica might never match the original at the source (because of the asynchronous batching), it faithfully reflects the state of the source at some point in the past.

**Recovery metrics**

When you consider the mechanics of recovery that is based on an asynchronously replicated image of runtime data, you understand the two key metrics of disaster recovery: recovery point and recovery time.

The recovery point is a measure of the lag, usually measured in the time elapsed, between the state of the runtime data at the source and at the replica. Synchronous replication could deliver a replica with zero lag and, therefore, a recovery point could match the source exactly. Because asynchronous replication does not guarantee that all updates to the source are simultaneously applied at the replica, the recovery point metric is important. The amount of lag will depend upon
the details of the replication management software, but generally a smaller recovery point measure (less lag) is more expensive. Most businesses establish a tolerance, called the recovery point objective (RPO), describing the maximum amount of data allowed to be lost in the event of a disaster and subsequent recovery.

The recovery time metric describes the elapsed time from the occurrence of the disaster until business operations are restored. Again, different businesses (and, in fact, different applications in the same business) can tolerate different amounts of down time in the event of a disaster. This tolerance for downtime is usually expressed as a recovery time objective (RTO).

**Disaster recovery options for IBM BPM**

The IBM BPM team has tested and certified disaster recovery strategies built upon two fundamentally different IBM WebSphere Network Deployment topological constructs and upon two distinct approaches to replicating runtime data. You could combine these techniques to produce four distinct options for accomplishing disaster recovery of an IBM BPM solution. Adding another, simpler approach and extending these basic concepts to a more-advanced hybrid further extends the various approaches available for disaster recovery with IBM BPM.

In each of these cases, it is important to ensure that your IBM BPM installation in the primary data center follows a good high-availability practice and the recommendations in the IBM Redbooks publication *Business Process Management Deployment Guide: Using IBM Business Process Manager V8.5*.

The following section explores some of the key features of the disaster-recovery options that can be applied.

**Replicating configuration data: cloned cell versus stray nodes**

The previous section introduced the need to replicate IBM WebSphere Application Server and IBM BPM configuration data from the primary data center to a remote, disaster recovery data center. Strategies that accomplish this by using file-by-file copies of the configuration data from the source to the replica are called cloned cell topologies. The cell in the remote data center is an exact copy of the one in the primary data center. The cell name, node name, and server names are the same. In fact, the universally unique identifiers (UUIDs) that the WebSphere Application Server and IBM BPM code use to describe all the components of the cell are identical. Additionally, the transaction recovery processing managed by WebSphere Network Deployment requires that the host names associated with the operating systems to which the servers are deployed must match across both data centers. Therefore, the WebSphere Network Deployment topology in the secondary data center is constructed by backing up the files that constitute the topology definition from the primary data center and restoring those to the secondary data center. Attempts to construct the secondary cell by using a parallel installation and running profile creation scripts do not work (because, among other things, the UUIDs do not match).

Stray nodes topologies use an entirely different approach. Rather than making copies of the servers from the primary data centers in the alternate data center, the stray nodes approach extends the IBM BPM cell across both data centers. In this way, the cell contains nodes in both
data centers and cluster members in both data centers that are federated into the same cell. Generally, spanning a cell across data centers is discouraged because it risks data corruption (resulting from a partitioned network) and performance problems (resulting from high latency in the network when connecting the two data centers), so, additional steps must be taken to mitigate these risks, as described in the following paragraphs.

Both IBM BPM and IBM WebSphere Network Deployment were engineered based on the expectation that the servers composing an entire cell are connected by a fast, reliable local area network (LAN) rather than a slower, less reliable wide area network (WAN). Stray nodes topologies introduce an important, additional requirement: operational procedures must exist to ensure that application servers may be started in only one data center at a time. During normal operations, these are the servers in the primary data center. After a disaster occurs, the procedures ensure that all servers in the primary data center are, in fact, stopped, and then the servers in the recovery data center can be started. In this way, cross-data center communication of runtime data does not occur. Many teams find that using scripts to automate the running of operational procedures is a valuable technique to improve efficiency, correctness, and repeatability and helps reduce the total recovery time and avoid violations, such as accidentally starting up the nodes in both data centers at the same time.

In contrast, communicating configuration data across data centers is acceptable in a stray nodes topology. This means that node agents can be running in both data centers to propagate configuration changes by communicating between the deployment manager and the node agents. As a result, maintaining configuration changes across the data center boundary is automated, compared with the backup and restore techniques used with a cloned cell approach. Therefore, stray nodes topologies might be quicker, simpler and less costly to implement and maintain than cloned cell topologies, because all the configuration and replication capability is in the WebSphere Network Deployment infrastructure.

Replicating runtime data: storage subsystem versus database

As described in the previous section, many enterprise class storage solutions provide features that you can use to coordinate asynchronous replication across multiple volumes. For example, an infrastructure administrator can use a storage area network to provide two volumes to the IBM BPM infrastructure and use tools provided by the storage area network to construct a consistency group that contains both volumes. The storage area network ensures that write-order consistency is preserved across both volumes, even when asynchronous replication is used. One of these volumes might be provided to the database manager for use in constructing the database containers and logs that IBM BPM will use. The other volume in the consistency group might be used as backing storage for a distributed file system (network file system, general parallel file system, or similar) hosting the recovery logs associated with the WebSphere Application Server transaction service and the compensation service and used by IBM BPM.

Many database management systems, such as Oracle Data Guard Replication and the high availability disaster recovery (HADR) feature in IBM DB2, provide replication features, just as storage subsystems do. Using these database replication features with IBM BPM was previously impossible because they did not provide a way to maintain consistency across the database
contents and the file system that hosts the WebSphere Application Server recovery logs. However, recent versions of WebSphere Application Server include an option that an IBM BPM server can use to store the recovery logs in a database instead of directly on a file system, as described in [Storing transaction and compensation logs in a relational database for high availability](https://www.ibm.com/support/knowledgecenter/SSBS65_8.5.5/com.ibm.websphere.admin.doc/topics/aeas_storage_tcm044114.htm) in the WebSphere Application Server documentation on IBM Knowledge Center. If this database is the same one that houses the IBM BPM database, database management techniques become viable.

If you want to store the transaction and compensation logs in a database, be aware that doing so might introduce additional performance overhead for some workloads, as compared with storing this information directly on a well provisioned file system. The impact of this additional overhead varies from one IBM BPM application to another. IBM BPM Standard implementations and processes implemented by using Business Process Modeling Notation (BPMN) experience the least impact, whereas Business Process Execution Language (BPEL) processes, especially transactionally complex macroflow applications, experience greater impact. As always, when you evaluate new technology, be sure to measure performance variables that are specific to your own processes and infrastructure. Despite performance considerations, the simplicity and flexibility provided by storing transaction and compensation logs in a database has made it a popular option for many applications.

### Recovering the WebSphere Application Server deployment manager

After a disaster occurs, the first priority is restoring business operations, which requires starting up the alternative IBM BPM servers that allow you to view and work on business processes. Other elements of the WebSphere Network Deployment infrastructure, most notably the deployment manager, are required to make changes to the configuration of the system and must be replaced eventually.

[The WebSphere Contrarian](https://www.ibm.com/support/knowledgecenter/SSBS65_8.5.5/com.ibm.websphere.admin.doc/xref/aeas_tcm014169.htm) column in the January 2010 edition of the IBM WebSphere Developer Technical Journal, *Runtime management high availability options, redux*, describes techniques that you can use to replicate the deployment manager configuration to establish a replacement for the administrative capability that the deployment manager provides if a disaster occurs.

One popular technique involves file-based replication of the deployment manager configuration, exactly as is done in the cloned cell strategies already discussed. So, it is natural to extend this approach to the deployment manager as well. This style of deployment manager replication and replacement works equally well for stray nodes approaches, even though the rest of the cell configuration does not require replication. The WebSphere Network Deployment infrastructure maintains the stray nodes configuration directly during normal operations.

### Capacity in the secondary data center

Regardless of whether a stray nodes or cloned cell topology is used to replicate IBM BPM configuration information, it is important to plan for adequate capacity in the secondary data center after recovering from a disaster. Generally, organizations require the same capacity (number of nodes and number of cores per node) after recovering from a disaster that they do during normal operations. After all, the objective of the disaster recovery plan is to restore the system to normal behavior.
However, one common shortcoming in disaster recovery plans is the failure to ensure sufficient available capacity in the alternate data center to support full-production load. In these cases, the disaster recovery procedure might complete successfully, but you might find that the replacement IBM BPM and database servers are immediately overloaded. For this reason, provision standby capacity in the secondary data center equal to that in the primary data center. Of course, the servers in the secondary data center do not have to remain idle during normal operations. They could be used for other, low-priority work, provided that this work can be removed from the system immediately after the disaster recovery procedure is enacted.

**Disaster recovery strategies**

There are seven approaches that cover the broad spectrum of techniques that you can use with IBM BPM to achieve availability and recovery of business requirements.

**Simple disaster recovery**

The easiest way to ensure consistency during replication is to quiesce the entire system. When the IBM BPM servers are stopped by using the administrative tools, all resources that participate in business processes are consistent, regardless of their storage subsystem, and can be replicated individually. As a result, you have a simple disaster recovery approach that is useful in situations that can tolerate periodic planned system outages or that lack the capability for unified replication across resource managers. This strategy has proven useful when replicating test and performance verification systems is required. Either a stray nodes or a cloned cell topology could be used with a replication scheme based on offline backups. A cloned cell topology is typical because all configuration data and runtime data are copied together during the maintenance window, as shown in Figure 1.

<table>
<thead>
<tr>
<th>Simple disaster recovery</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical RTO</td>
<td>4-8 hours</td>
</tr>
<tr>
<td>Typical RPO</td>
<td>Depends on the maintenance interval, usually 24 hours</td>
</tr>
<tr>
<td>Advantages</td>
<td>Simplicity</td>
</tr>
<tr>
<td>Cautions</td>
<td>Periodic, planned outages on the primary data center</td>
</tr>
<tr>
<td>Recommended for</td>
<td>Test systems</td>
</tr>
<tr>
<td></td>
<td>Processes that can tolerate regular downtime and high RPO</td>
</tr>
</tbody>
</table>
Figure 1. Simple disaster recovery

Classic disaster recovery (storage area network replication)

Combining a cloned cell topology with storage-managed replication produces one of the most popular disaster recovery approaches for IBM BPM. It remains the most widely tested approach both in the IBM development labs and at client installations. See Figure 2 for an illustration. Using a cloned cell topology requires that the recovery procedures ensure host names from the primary data center are available for use in the secondary data center.

<table>
<thead>
<tr>
<th>Classic disaster recovery (storage area network replication)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical RTO</td>
<td>4-8 hours</td>
</tr>
<tr>
<td>Typical RPO</td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td>Advantages</td>
<td>storage area network replication provides top notch enterprise replication features</td>
</tr>
<tr>
<td>Cautions</td>
<td>Host names from primary data center must be available in the secondary data center.</td>
</tr>
<tr>
<td>Recommended for</td>
<td>Large enterprises with existing storage area network replication practices</td>
</tr>
</tbody>
</table>
Classic disaster recovery (database replication)

Organizations that have an established replication practice based on Oracle Data Guard Replication or the DB2 HADR feature rather than storage subsystem components might place the WebSphere Application Server transaction and compensation logs directly in the IBM BPM database to use their existing operational procedures for IBM BPM also (see Figure 3). Extensive testing has proven this approach to be reliable. Also, because the storage backing up the database server in the secondary data center remains mounted during normal operations, this approach can provide somewhat faster total recovery time than replication managed by the storage subsystem does.

<table>
<thead>
<tr>
<th>Classic disaster recovery (database replication)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Typical RTO</strong></td>
<td>2-4 hours</td>
</tr>
<tr>
<td><strong>Typical RPO</strong></td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td><strong>Advantages</strong></td>
<td>Uses database-replication technologies</td>
</tr>
<tr>
<td><strong>Cautions</strong></td>
<td>Host names from primary data center must be available in the secondary data center. Transaction log storage and replication performance might be a factor for some applications.</td>
</tr>
<tr>
<td><strong>Recommended for</strong></td>
<td>Organizations with existing replication practice based on database-managed replication</td>
</tr>
</tbody>
</table>
Stray nodes (database replication)

A stray nodes topology combined with database managed replication has demonstrated the fastest total recovery time in internal testing. In addition to the fact that database storage in the secondary data center remains mounted during normal operations, the stray nodes approach allows node agents in the secondary data center to be up and running, reducing the number of steps to run during the recovery procedures. Since its introduction with the release of IBM BPM versions 8.5.0.1 and 8.0.1.1, this approach has proven very popular, especially among IBM BPM Standard Edition installations relying on Oracle Data Guard Replication. See Figure 4.


<table>
<thead>
<tr>
<th>Stray nodes (database replication)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical RTO</td>
<td>~1 hour</td>
</tr>
<tr>
<td>Typical RPO</td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td>Advantages</td>
<td>Fast, flexible recovery using database-managed replication technologies</td>
</tr>
<tr>
<td>Cautions</td>
<td>Operational procedures must ensure stray nodes are not accidentally started during normal operation. Transaction log storage and replication performance might be a factor for some applications.</td>
</tr>
</tbody>
</table>
Figure 4. Stray nodes (database replication)

Stray nodes (storage area network replication)

You can also pair a stray nodes configuration with a replication strategy for runtime data that relies on the storage subsystem, as shown in Figure 5. This combination is particularly popular among organizations that already use storage area network replication for other applications and want to apply this practice to their IBM BPM applications as well. This combination offers the power and flexibility of a storage area network for replication with the potentially simplified recovery of the IBM BPM cell that is provided by the stray nodes.
Stray nodes (storage area network replication)

<table>
<thead>
<tr>
<th>Typical RTO</th>
<th>2 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical RPO</td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td>Advantages</td>
<td>Flexible WebSphere Application Server Network Deployment configuration with top-notch enterprise replication features</td>
</tr>
<tr>
<td>Cautions</td>
<td>Operational procedures must ensure stray nodes are not accidentally started during normal operation</td>
</tr>
<tr>
<td>Recommended for</td>
<td>Organizations with strong WebSphere Application Server Network Deployment skill and replication practice based on storage area network techniques</td>
</tr>
</tbody>
</table>

**Figure 5. Stray nodes (storage area network replication)**

**Metro pair**

Using a single IBM BPM cell with active members in multiple data centers remains a practice that is not generally recommended for disaster recovery purposes. The IBM BPM server makes extensive use of its database, and the network separation that these approaches imply raises the risk of performance problems and data corruption issues through network partition (split brain scenarios). To reduce the risk of network problems, these installations typically rely on a pair of data centers located close to each other (which is why the approach is called "metro") and connected by a redundant, high-capacity network infrastructure, as shown in Figure 6. This proximity reduces protection against natural disasters and other events that affect a broad geographic region. For this reason, this type of approach is considered an advanced high-availability technique and not a solution for disaster recovery.
Before implementing an approach based on a metro pair with no geographic dispersion, carefully consider all availability and recovery requirements to ensure that the benefits justify the risks. In particular, consider the behavior of the database server when the primary data center is lost. If switching the database from one of the metro pair data centers requires the IBM BPM servers to be restarted, you have lost the recovery time advantage that the topology tried to achieve.

Many organizations find that a single cell split among data centers forms an important element of their overall availability and recovery strategy. When backed by a fast and reliable network and supported by exhaustive testing to characterize actual failover times and the risk of data corruption, topologies like the metro pair can serve a purpose. When augmented to include a true, remote disaster-recovery site, as described in the Metro pair and disaster recovery section, these strategies can become a valuable enterprise class solution.

<table>
<thead>
<tr>
<th>Metro pair</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical RTO</td>
<td>Minutes</td>
</tr>
<tr>
<td>Typical RPO</td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td>Advantages</td>
<td>Gives the appearance of Active/Active</td>
</tr>
<tr>
<td>Cautions</td>
<td>No geographic dispersion</td>
</tr>
<tr>
<td></td>
<td>Network latency and partition exposures</td>
</tr>
<tr>
<td>Recommended for</td>
<td>Not generally recommended for disaster recovery</td>
</tr>
</tbody>
</table>
Figure 6. Metro pair

You can augment the metro pair topology approach with a disaster recovery strategy that uses replication across geographically dispersed data centers to a passive standby system. This solution resolves the exposure to total system loss in the event of a natural disaster that is present when only a metro pair topology is used. Because the metro pair topology is nothing more than a single, extended IBM BPM cell, you could implement this augmentation by using any of the four core disaster-recovery strategies described previously.

For simplicity, Figure 7 depicts a cloned cell with database managed replication, although stray nodes and storage-area-network-managed replication techniques are equally well suited to this type of approach. Because a third data center is introduced, the operational procedures required to manage this type of infrastructure are more complex than the other approaches described.
Disaster recovery guidance for IBM Business Process Manager

Metro pair and disaster recovery

<table>
<thead>
<tr>
<th>Typical RTO</th>
<th>Local: Minutes / Geo: up to 4 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical RPO</td>
<td>Seconds to minutes</td>
</tr>
<tr>
<td>Advantages</td>
<td>Hybrid approach provides the best of both worlds</td>
</tr>
<tr>
<td>Cautions</td>
<td>Complexity</td>
</tr>
<tr>
<td>Recommended for</td>
<td>Sophisticated organizations with a well-defined recovery requirements and strong test practice</td>
</tr>
</tbody>
</table>

Figure 7. Metro pair and disaster recovery

Summary of disaster recovery strategies

The previous sections explored core replication technologies for IBM BPM configuration data (stray nodes and cloned cells) and runtime data (storage-managed and database-managed), leading to four primary disaster recovery strategies. Extending this set of four to include the simple disaster recovery strategy and the two strategies based on a metro pair of IBM BPM data centers, gives seven approaches that cover the broad spectrum of techniques that you can use with IBM BPM to achieve availability and recovery business requirements. Figure 8 provides a quick reference that compares the features that distinguish these approaches from each other.
Validation of disaster recovery

As with any set of nonfunctional requirements, verifying that business requirements for recovery behavior are met is an important element of the disaster recovery plan. Because simulations including cross-site failover are disruptive, it is extremely useful to provision a dedicated test environment for this type of testing. Primary objectives include validating cross-site configuration to ensure that all systems connect and communicate properly after recovery and measurement of total recovery time to set proper expectations in the event of an actual disaster. In addition, many organizations improve execution efficiency through practice, as administrators become more comfortable and confident in the steps to run. Beyond this, new opportunities for scripting and other sorts of automation become apparent only after repeated execution.

The future of disaster recovery

While techniques such as cloned cells, stray nodes, database-managed replication and storage-managed replication originated in traditional on-premise environments, it is natural to consider to what extent the same techniques can be applied or extended to virtual environments. Beyond this, virtualization of infrastructure introduces new opportunities for automating the operational procedures upon which a successful disaster recovery procedure relies. Look for future IBM content that more deeply covers how disaster recovery can be achieved in the presence of the various virtualization approaches that dominate the world of infrastructure and topology for IBM BPM systems.

In addition to virtualization influences, more and more IBM BPM workloads are running in various cloud configurations, including public clouds, hosted configuration, and private cloud environments.
such as those running on IBM PureApplication System. Look for updates on how to best achieve disaster recovery in these environments to complement this update.

As a preview specifically on the subject of running IBM BPM systems on IBM PureApplication System, the recently released IBM BPM 8.5.5 pattern supports the same approaches for disaster recovery that are popular when replicating configuration data and runtime data in physical systems.

**Conclusion**

Ensuring that a business process management system running mission-critical business processes can be recovered in the event of a disaster remains a key requirement of large businesses. Stateful workloads bring challenges to the table that are beyond those of traditional stateless workloads.

You can configure IBM BPM V8.x to recover from a disaster by using various techniques. Recent evolutions in how IBM BPM can be configured and run now provide new and valuable configuration options that significantly reduce the possible RTO values down to less than an hour. You should now be familiar with the current set of options, the current set of implications for using those options in terms of topology, and the underlying infrastructure capabilities necessary to get to these aggressive RTO numbers considered unachievable before.

Each of the possible configurations impacts the underlying infrastructure and has associated costs. Charlie Redlin engaged clients with a simple message when they requested aggressive RTO and RPO numbers: "How much money do you have to achieve required numbers?" Although aggressive disaster recovery still costs money and takes a deep infrastructure commitment, the good news is that, with the current infrastructure and database capabilities, the costs are coming down to more affordable levels. You can use these capabilities to achieve aggressive RTO and RPO disaster recovery.

**Acknowledgements**

The authors would like to thank Hong Yan BJ Wang, Yu Zhang, Uday Pillai and Mahesh Sharma for their review and valuable input.
Related topics

- Asynchronous replication of WebSphere Process Server
- IBM Business Process Manager Version 8.0 Production Topologies IBM Redbooks Publication
- Transaction support in WebSphere Application Server support document
- Managing transaction logging for optimum server availability (including database storage option) support document
- Storing transaction and compensation logs in a relational database for high availability and disaster recovery in IBM Business Process Manager
- IBM BPM Disaster Recovery documentation on IBM Knowledge Center
- Product support website for IBM Business Process Manager Advanced
- The WebSphere Contrarian: Runtime management high availability options, redux
- IBM Business Process Manager Pattern V8.5.5 announcement
- IBM Business Process Manager documentation on IBM Knowledge Center
- IBM Business Process Manager Developer Center

© Copyright IBM Corporation 2015

Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)