Disaster Recovery in an IBM Business Process Management V7.5 Production Environment

Contents

1. Background

Today's business environment is characterized by rapid, unpredictable change. Some changes bring opportunities for the business, while others bring challenges and even threats. The business has to be responsive and resilient, seamlessly taking advantage of opportunities while mitigating risks. Your information technology (IT) infrastructure must be designed to secure data integrity and to ensure the continuity of business operations in the event of an unexpected disruption.

Business continuity is an overall plan to keep all aspects of a business functioning in the midst of disruptive events. Disaster recovery (DR) is a subset of business continuity, focusing on the technology systems that support business continuity.

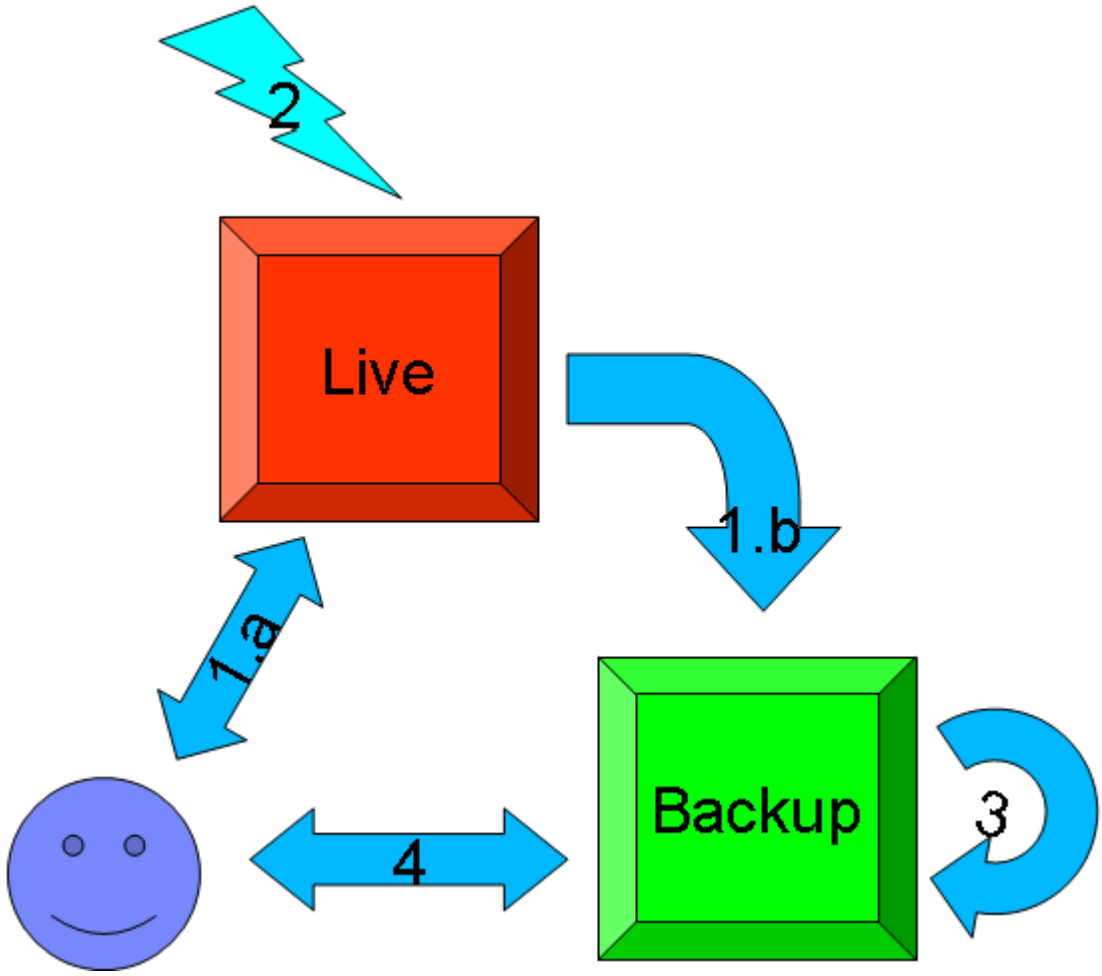

Figure 1: Disaster recovery

Figure 1 illustrates a system with disaster recovery.

1. During normal operation:
    a. Customers use the live system.
    b. The backup programs run in the background to save environmental information and application data.

2. When the live system goes down:
3. The backup system is restored from the backed up data.
4. Customer can use the system again.

Disaster recovery consists of the policies and procedures that describe how to recover or continue the technology infrastructure critical to an organization after a natural or human-induced disaster. Usually, disaster recovery consists of well-defined strategies to back up the primary data center and restore its data to a secondary data center.

Usually, the data center of a customer IT environment consists of various systems and environments, such as Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Human Resource Management (HRM). The disaster recovery strategy must define the general rules from a high level point of view, with detailed plans for each system. Each system might be a complicated combination of software and hardware deployments, each of which provides different functionality. The disaster recovery for the system must take all components into consideration to provide a complete solution.

This document describes the disaster recovery scenario for a business process management (BPM) and business activity monitoring (BAM) production environment, including the installation, configuration, and underlying database. The recovery scope covers only the V7.5 production environment and no other systems and components that interact with it. In a complete scenario, the suggestions in this document would be incorporated into the overall disaster recovery document to provide a complete solution.

2. BPM topology and DR concepts

2.1 Production topology

The production environment described in this document consists of an IBM® Business Process Manager and an IBM Business Monitor golden topology setup, which are located within the same cell of the Network Deployment environment. The underlying database to support Business Process Manager, the messaging engine, Business Space powered by WebSphere®, and IBM Business Monitor are also regarded as part of the production environment and included in the same recovery scope, because the whole production environment must be in a consistent state during the restoration phase.

2.2 Data classification

The production environment contains four types of data, as illustrated in Figure 2.

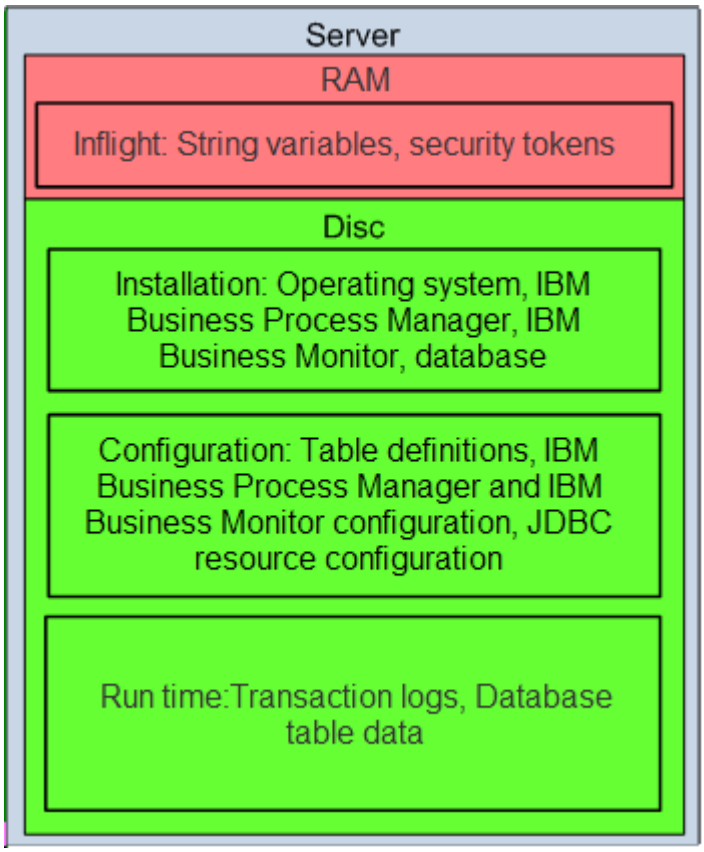

Figure 2: Data classification

- **Installation data**: Installation data is the data associated with the installation of IBM Business Process Manager and IBM Business Monitor, the underlying database installation, and the operating system (OS) data related to IBM Business Process Manager and IBM Business Monitor. The installation data does not change after initial installation.
- **Configuration data**: Configuration data is the data associated with profile configuration, applications, resource configuration of IBM Business Process Manager and IBM Business Monitor, and related database and table definitions. The configuration data changes when you install an application, create a profile, generate a new cluster member, or make other configuration changes.
- **Runtime data**: Runtime data is the data associated with transaction logs, messages saved in the database table, process instance information persisted in the database table, and other persistent business states. Runtime data changes continuously while the production environment is running.
- **Memory data**: Memory data is the intermediate data kept inside memory.

Some kinds of data, such as operating system installation and configuration data, IBM Business Process Manager and IBM Business Monitor installation data, and database installation data, can be rebuilt or reinstalled. Other kinds of data, such as transaction logs, application data, and configuration data for IBM Business Process Manager and IBM Business Monitor, must be recovered.

2.3 Recovery Scope

Recovery scope defines what resources are part of a backup. In this case, the resources include IBM Business Process Manager and IBM Business Monitor configuration, runtime data, and all the customer data, including

customer applications and process templates and instances. You should usually put all the files such as IBM Business Process Manager and IBM Business Monitor underlying database files and all the profiles into the same volume group (for OS Snapshot) or consistency group (IBM San). This will affect your sizing result in the disaster recovery plan.

2.4 Consistency

After a disaster and a successful recovery of the production system from backup, you must ensure that you have consistent data. For IBM Business Process Manager and IBM Business Monitor, this consistency must apply to all cell members. If one node in a cell is inconsistent, the backup image and restore attempt is invalid. You must have crash consistency and application consistency as defined below.

- **Crash consistency**: The bytes in the restoration match those at the time of the backup. In a shared, multi-node environment, the data for the cluster is assured to be in the same time sequence as the writes.
- **Application consistency**: When the OS starts, there are no file system recovery errors. Applications are able to access data from the time of the backup without failure. The applications recover in-flight transactions upon restarting.

2.5 Disaster recovery procedures

2.3.1 Overview

From the perspective of IBM Business Process Manager and IBM Business Monitor, disaster recovery means that the production environment can be restored to the secondary data center through a well-defined backup method. Disaster recovery for IBM Business Process Manager and IBM Business Monitor is supported through disk replication technology, which takes a snapshot of the original production environment, and restores and validates the data in the secondary data center, as illustrated in Figure 3.

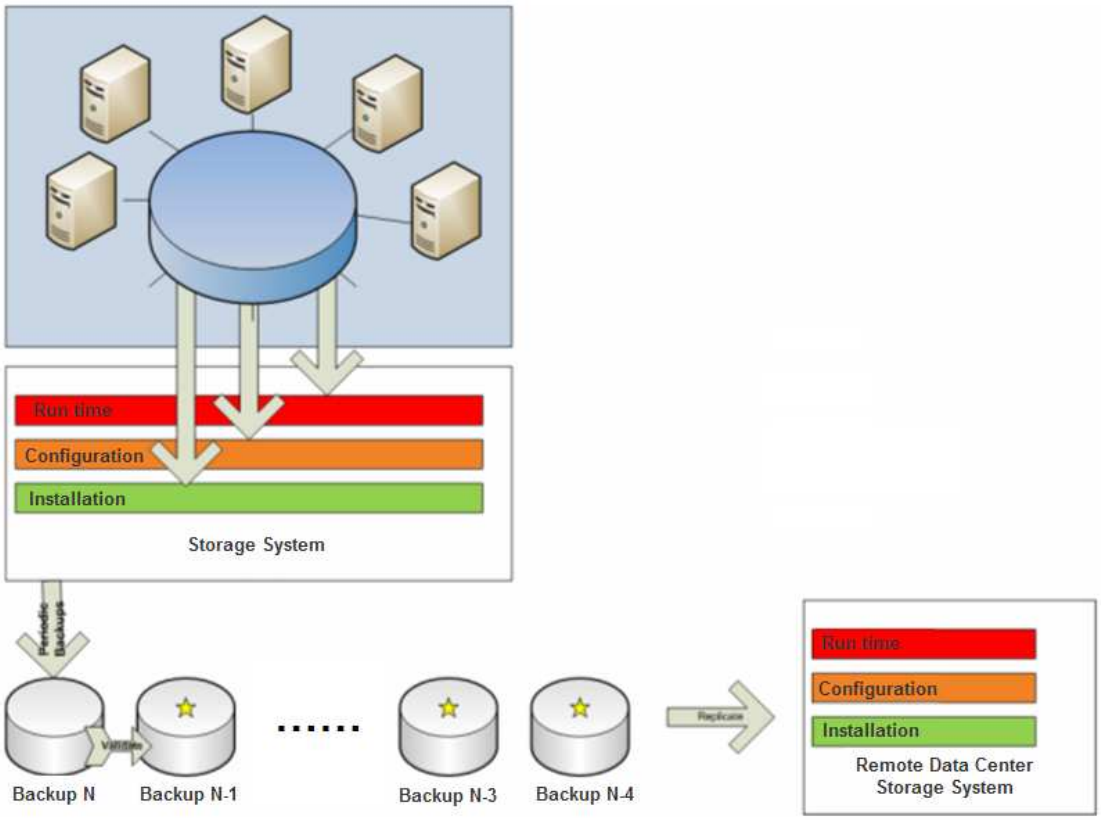

Figure 3: Backup and restoration procedure

The figure shows that some backups might not work properly after restoration. You need to identify and discard those backups, and use only the valid ones.

You must have a comprehensive disaster recovery plan before you implement the disaster recovery plan. Consider the Recovery Point Objective (RPO) and Recovery Time Objective (RTO) based on your real business needs.

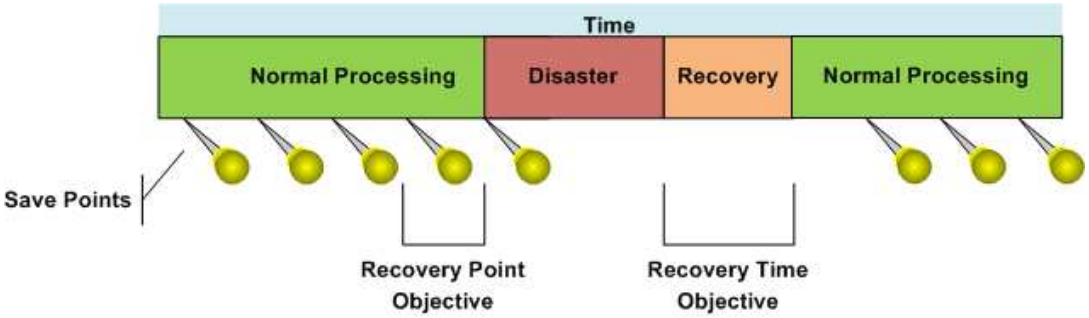

Figure 4: RPO and RTO

As illustrated in Figure 4, the Recovery Point Objective (RPO) defines how much data you can afford to lose between the original environment and the restored environment. From a business perspective, a smaller RPO means that fewer business transactions are lost, which is critical for normal business operations. To achieve a smaller RPO, you must increase the frequency with which you back up the production environment. However, you must also consider the cost and effect of frequent backups on your production environment. The more

times you back up, the more copies you have to maintain.

Recovery Time Objective (RTO) defines how long you can wait until the restored environment can continue with normal processing. From a business perspective, you may want to achieve different RTOs based on your own business needs. To define the appropriate RTO, consider the work that must be done during disaster recovery. Increasing the frequency of your backups does not always lead to a smaller RTO. For example, if the server startup takes 20 minutes, you cannot reduce recovery time below 20 minutes, no matter how often you back up. You would have to re-architect your server to start faster or get a faster machine.

Generally, you must define your RPO and RTO goals according to your business needs, and you can achieve those goals through regular IT operations.

2.3.2 Backup

A backup is a copy of the production environment. There are several ways to make a backup. Each method imposes some constraints on the production environment and each presents some advantages and disadvantages.

2.3.2.1 Simple server backup

The simple server backup approach uses a packaging or compression utility to back up the file system. In the case of a distributed environment, you must back up the data for each server separately.

Constraint

- No file system update operations are allowed during the entire recovery scope. This means that no work can occur throughout the entire production environment during the backup.

Advantages

- Safe and consistent
- No additional requirements for underlying file system configuration

Disadvantage

- Requires significant downtime because no active workload can occur during the backup

The simple server backup approach is illustrated in Figure 5.

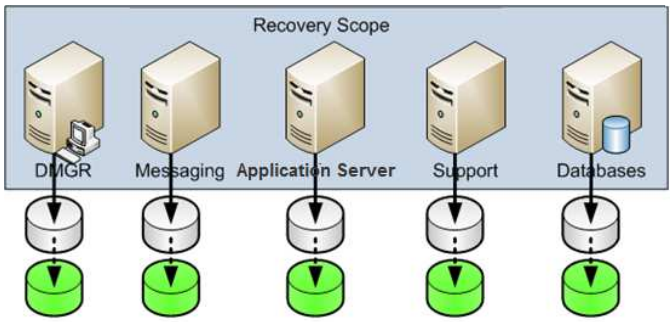

Figure 5: Simple server backup

2.3.2.2 OS snapshot

The OS snapshot approach captures the state of a running production system at a specific point in time.

Constraint

- The production system can write to only one file system.

Advantages

- Supports hot backup without requiring a stoppage of activity
- Inexpensive, because it requires only OS snapshot support

Disadvantages

- Backup might not be consistent because of the load
- Verification process is required to validate the backup

The OS snapshot approach is illustrated in Figure 6.

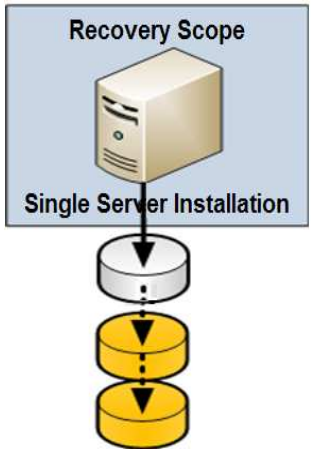

Figure 6: OS snapshot

2.3.2.3 OS snapshot with shared file system

The OS snapshot approach captures the state of a running production system at a specific point in time. With

the support of a shared file system, the OS snapshot can back up a distributed production environment. An example of a shared file system is the Network File System (NFS) on UNIX.

Constraint

- Requires one dedicated server to act as the NFS server

Advantages

- Inexpensive, without requiring a special device
- Supports hot backup without requiring a stoppage of activity

Disadvantages

- NFS might introduce performance bottlenecks because all access goes into the same file system through the network
- NFS is a single point of failure because of centralized management

The OS snapshot with NFS support is illustrated in Figure 7.

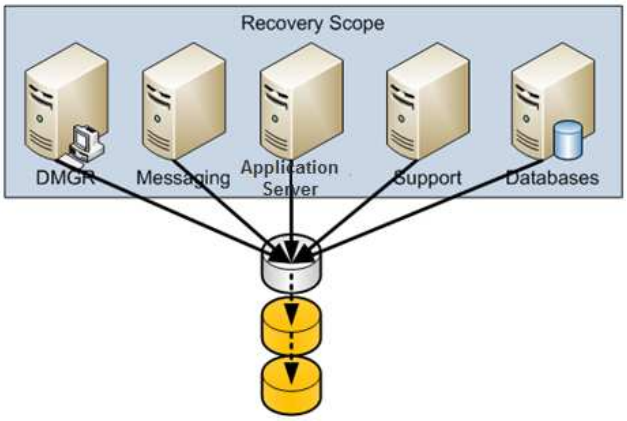

Figure 7: OS snapshot with NFS support

2.3.2.4 OS snapshot with storage system support

A storage system such as SAN can be used to provide a central repository of production environment data. From a functionality point of view, this approach is the same as the OS snapshot with shared file system approach.

Constraint

- Requires special hardware support

Advantages

- Supports hot backup without requiring a stoppage of activity
- Designed for high availability and high performance

- Most SAN systems support both periodic snapshots and synchronous replication of data to a remote site

Disadvantages

- More expensive
- Requires complicated configuration

The OS snapshot with storage system support approach is illustrated in Figure 8.

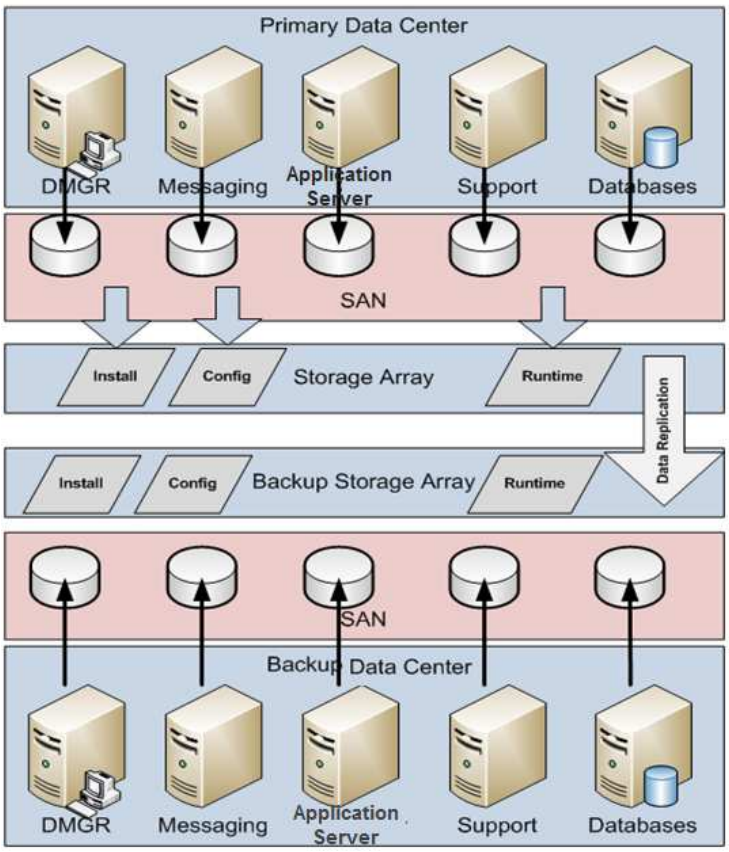

Figure 8: OS snapshot with storage system support

2.3.2.5 Summary

Each backup method has own constraints, advantages, and disadvantages. When you choose your method, the most important consideration is your business requirements. For example, if a period of downtime is acceptable, the simple server backup is the best choice, because it ensures the consistency of your system.

To minimize your RPO, determine your ideal backup frequency by considering the business requirements and resources to take and maintain your backups.

2.3.3 Restoration and verification

The objective of a backup is to restore data from a valid backup to the secondary data center. If a disaster occurs in the primary data center, a valid backup must be available to restore to the secondary data center, so that you can continue to provide business support. Verification of the restored data is very important during the backup and restore procedure.

2.3.3.1 Restoring data

Restoration is the process of rebuilding all or part of a backup to the corresponding secondary environment.

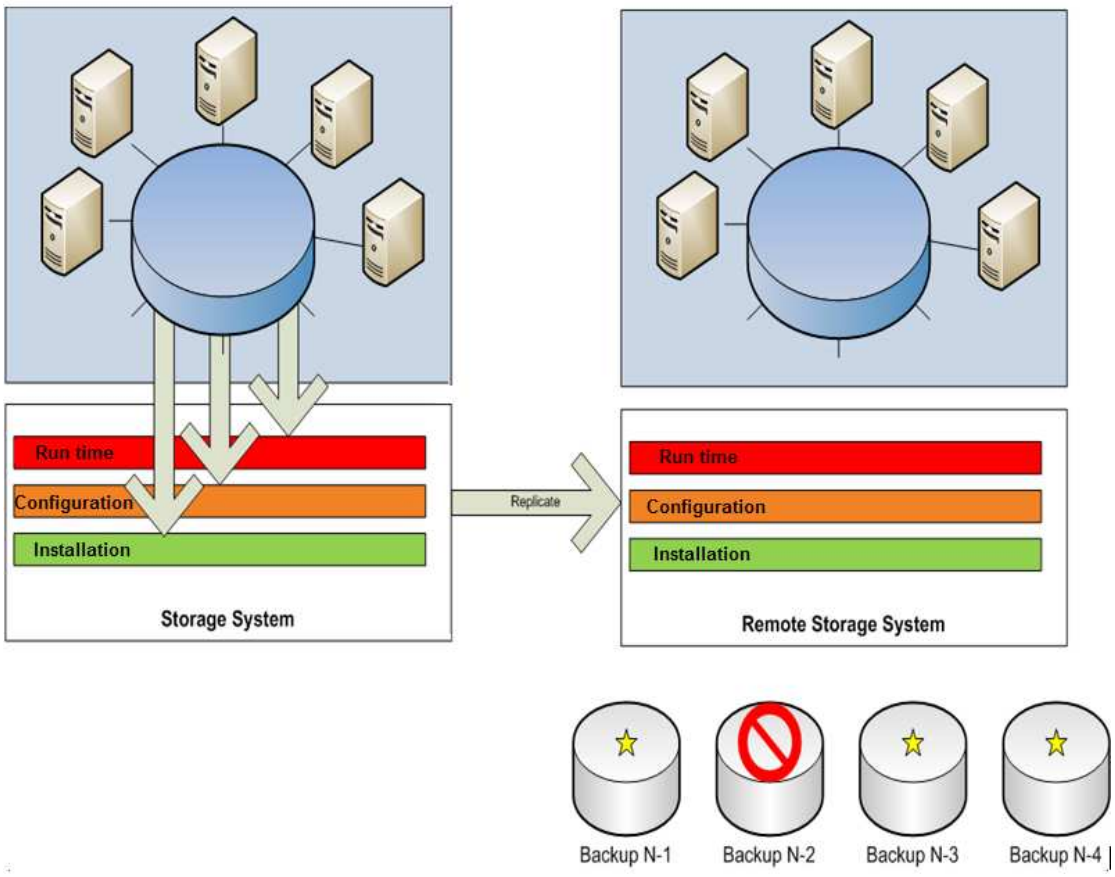

Figure 9: Restoration process

To restore the production environment to the secondary environment, complete the following steps:

1. Reinstall the BPM installation data, including the IBM Business Process Manager installation data and the IBM Business Monitor installation data.
2. Reinstall DB2 and create the DB2 instance.
3. Restore the configuration data to all servers from the backup configuration data.
4. Restore the runtime data to all servers by replicating the backup runtime data.
5. Perform changes that are specific to the environment. For example, update the host name to reflect the

secondary environment, or change the data source configuration to point to the secondary database.

6. Validate the connectivity to the resources outside the recovery scope.
7. To restart the environment:
    a. Start the database server.
    b. Start the deployment manager and node agents.
    c. Start the message servers of IBM Business Process Manager.
    d. Start the support servers of IBM Business Process Manager.
    e. Start the application servers of IBM Business Process Manager.
    f. Start the message servers of IBM Business Monitor.
    g. Start the support servers of IBM Business Monitor.
    h. Start the application servers of IBM Business Monitor.
8. Verify the restored environment and determine whether it is valid.
9. Recover in-flight transactions.
10. Redirect load to the new environment. (Usually you can set the same host name and IP address for the secondary environment as for the primary environment. This step depends on your backup policy.)

Important: To ensure the consistency of all data, the restoration must occur for the whole cell and underlying database.

2.3.3.2 Verifying data

After you restore the backup of your production environment to the secondary data center, verify the data to determine whether the backup is a valid copy.

A failure, loss of data, or inconsistency from the instance level can be tolerated, but an abnormal state from the system level or application level must be fixed. In the latter case, the backup must be deemed invalid.

To verify that the restored data in the secondary environment is valid, complete the following steps:

1. **System level**: Verify that the system-level services such as the Business Process Choreographer container and the Human Task Manager container are working properly. Verify that the messaging engines for various buses can be started successfully. To perform these verifications, you can use the System Health widget in Business Space.
2. **Module and application level**: Verify that the modules and applications can be started successfully. Verify that the process templates can be started normally.
3. **Process instance level**: Verify that the process instances are in a consistent state.
4. **Consistency level**: Verify that the process instance state between IBM Business Process Manager and IBM Business Monitor is consistent.
5. **SCA level**: Verify that synchronous and asynchronous invocation for Service Component Architecture (SCA) can continue for processing.
6. **Monitoring level**: Verify that you see new instances in your monitor dashboards when you run new process instances.

Generally, verification is relatively simple for the system, module, and application levels. However, verification of the instance level might be more difficult, because the number of instances might be very large. Use a real runtime scenario for the disaster recovery test, which takes the backup of the running instances and

verifies that the specific instances are working properly.

3. Installation and configuration considerations

To ensure that your disaster recovery procedures run smoothly, consider the following factors when you install and configure operating systems, databases, and the BPM production environment.

3.1 Operating system considerations

Generally, you should ensure that the basic operating system configurations for the primary and secondary environments are the same. For example, if there are five operating system configurations in the primary environment, there must be five operating system configurations in the secondary environment with the same operating system version and file system hierarchy. In the following examples, all operating systems are deployed with Red Hat Enterprise Linux.

3.1.1 Host name and IP configuration

Because the host name and the IP configuration of the primary and secondary environments will appear in both the database and the BPM configuration data, configure the same IP and host name for each pair of primary and secondary operating systems. If you do so, when the configuration and runtime data is moving from the primary environment to the secondary environment, it will not be necessary to update the IP and host name information inside the configuration file.

3.1.2 Snapshot support

If you want to back up the primary environment without affecting normal functioning, you need the additional support of an OS snapshot. This section uses Linux as an example.

On the Linux platform, you can use Logical Volume Management (LVM). LVM provides a higher-level view of the disk storage on a computer system than the traditional view of disks and partitions. With LVM, the system administrator has more flexibility in allocating storage to applications and users by demand. The physical volumes of the disk are organized as logical volumes, and the file system is mounted on logical volumes. This organization allows the flexible and dynamic management of the disk size of the file system.

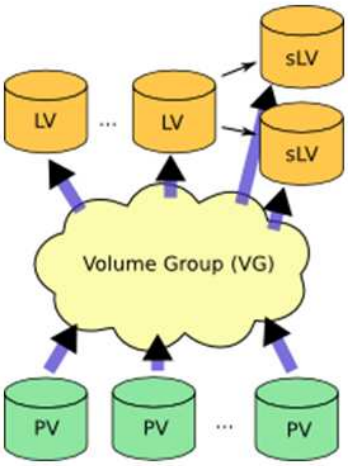

Figure 10: LVM concept

When you use LVM support, the file system supports concurrent backup while the file system is undergoing a write operation by snapshot. Without snapshot support, the native backup of a large number of files consumes a great deal of time. During this period of time, some files might be updated because transactions are continuing in the production environment, which means that the backup contains files saved at different points in time. If any files are in an inconsistent state, the backup is not acceptable.

To support the snapshot functionality through LVM, you can use the Copy on Write (CoW) mechanism. The following figure demonstrates the CoW concept.

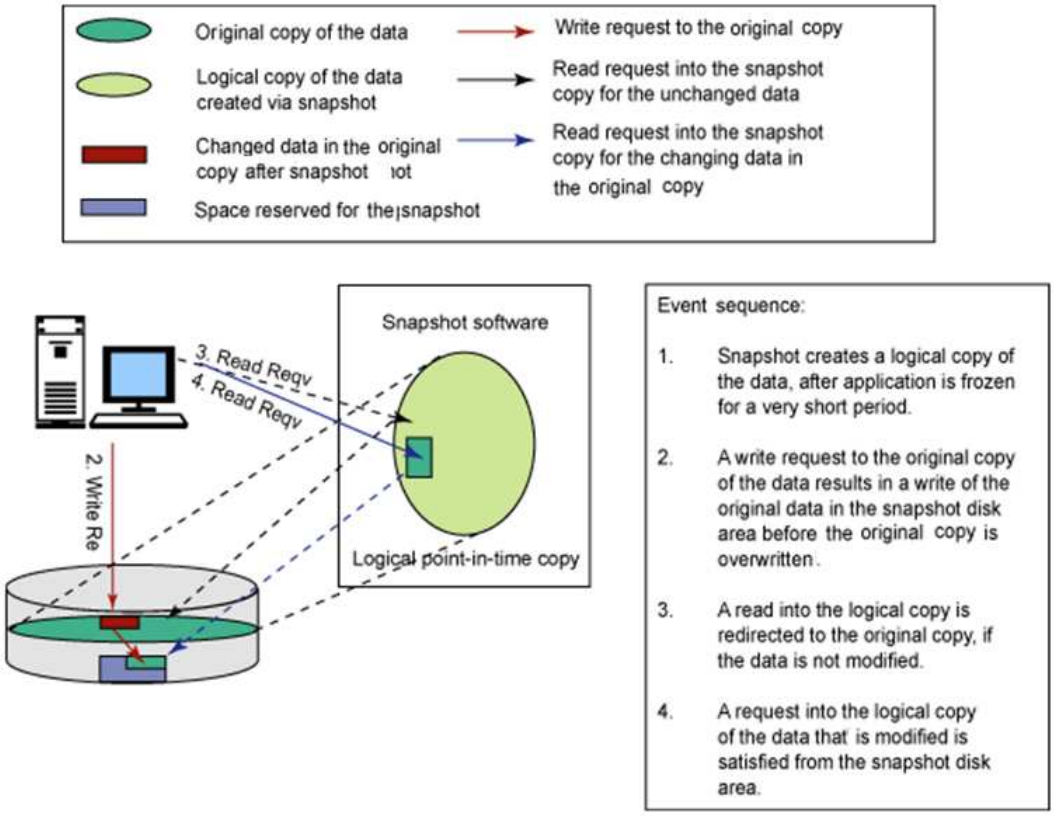

Figure 11: CoW mechanism

3.1.2.1 Preparing the operating system before a snapshot

When you prepare for an OS snapshot, consider the following factors.

The `/opt` and `/home` directories must be vacant, because any data in the directory will be destroyed when you mount a directory. You can mount an extra disk with about 10 GB of vacant space, and then you can create a physical volume on it. After you extend the physical volume to a volume group, you can create the new logical volume in the volume group.

To prepare the operating system before you take a snapshot, complete the following steps:

1. List the general information (PV, VG, LV) of the Linux operating system:
   ```
   # pvdisplay
   # vgdisplay
   # lvdisplay
   ```
2. List the disk information:
   ```
   # fdisk-1
   ```
3. Create a physical volume on the disk partition, for example `/dev/sda2`:

```
# pvcreate /dev/sda2
```

4. Extend the new physical volume to the volume group:
```
# vgextend VolGroup00/dev/sda2
```
5. Create a logical volume on the volume group:
```
#lvcreate –name homebackup –size 10G VolGroup00
```
6. Make the file system format for the new logical volume:
```
#mkfs.ext3 /dev/VolGroup00/homebackup
```
7. Mount the logical volume to the `/home directory`:
```
# mount /dev/VolGroup00/homebackup/home /home
```

3.1.2.2 Taking an OS snapshot

To take an OS snapshot, complete the following steps:

1. Take a snapshot of the `/home` directory. The snapshot is also a new logical volume:
```
# lvcreate -L1G -s -n homesnapshot /dev/VolGroup00/homebackup
```
You can also use the GUI tool in the OS. For Red Hat Linux, it is Logical Volume Management.
2. To use the logical volume, create a directory under `/mnt` to store the snapshot files:
```
# mkdir /mnt/homesnapshot
```
3. Mount the snapshot logical volume to the new directory:
```
# mount /dev/VolGroup00/homesnapshot /mnt/homesnapshot
```
4. If you no longer need a snapshot, unmount it and remove it to save disk space:
```
# lvremove /dev/VolGroup00/homesnapshot
```

You can also use other methods to create a snapshot. After each snapshot, you will need to compress it, and then FTP it to the secondary environment. On the secondary environment, extract the snapshot files and test them, and then wait for disaster recovery. Also, the longer you keep the snapshot, the more disc space will be taken up, so you should create snapshots periodically and sort them based on your RPO.

3.1.3 NFS support

In a distributed environment, the data of the production environment is distributed over several operating systems. Without special configuration, during run time, it is highly possible to get an inconsistent copy of the entire environment even through a snapshot. This problem might occur because the snapshot is performed at the operating system level and the snapshot for different operating systems might correspond to the state at different points in time. A consistent copy of the entire environment is required to ensure the proper behavior of the system. Therefore, you need a method to make the copy consistent throughout the entire production environment.

When you use a Network File System (NFS), users on a client computer can access files over the network as if the files were on their local server. In this architecture, a file server is configured on one operating system, which functions as the central repository for all files. The NFS client operating system can connect with the file server and mount the specific directory to the file server. The NFS client operates transparently on the directory mapped on the file server.

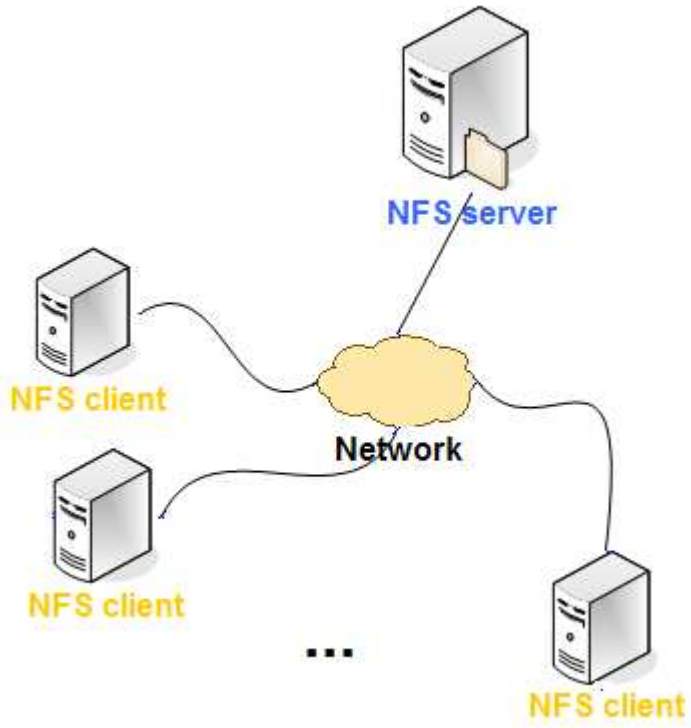

Figure 12: NFS architecture

When NFS is enabled, the configuration and installation data of the production environment can be configured on a centralized NFS file server. In combination with the snapshot support of the file server operating system, you can create a consistent backup of the entire production system.

Before you create a snapshot, you must set up your NFS server and clients.

3.1.3.1 Configuring the NFS server

The following example shows how to configure your NFS server.

1. Create the directories that you want to mount to the NFS client directories, for example `/home/machine1`, `/home/machine2`, and `/home/machine3`. Make sure that these directories have write authority.
2. Configure the `/etc/exports` file:

   ```
   /home/machine1 *(rw, sync)
   /home/machine2 *(rw, sync, no_wdelay, nohide)
   /home/machine3 *(rw, sync, no_root_squash)
   /home/machine4 *(rw, sync, no_root_squash)
   ```

   In this example, the `/home/machine3` and `/home/machine4` directories will be mounted to the remote custom profile directory for IBM Business Monitor. You must have the `no_root_squash` parameter, or you will get an error (cp:failed to preserve ownership) when you create the custom profile for IBM Business Monitor.

3. Before the NFS service starts, the **portmap** service must be running. To check its status, use the following command:
```
# service portmap status
```
4. If the **portmap** service has stopped, use the following command to start it:
```
# service portmap start
```
5. To start or restart the NFS service, use one of the following commands:
```
# service nfs start
# service nfs restart
```
6. To make the NFS service start up automatically with the system, use the following command:
```
# chkconfig --level 35 nfs on
```
7. To check the NFS export directories, use the following command. You can use this command on both the NFS server and the NFS client:
```
# showmount –e %server_ip%
```

3.1.3.2 Configuring the NFS clients

For each NFS client, complete the following steps:

1. To mount the corresponding directory to the remote NFS server, use the following commands:
```
# mount    %server_ip%:/home/machine1   /home/dmgr
# mount    %server_ip%:/home/machine2   /home/db2
# mount    %server_ip%:/home/machine3   /home/custom01
```
2. Make these mounts start automatically with the system so that you will not have to run these commands every time that you start your system.
3. Repeat for all other NFS clients.

3.2 Database considerations

The underlying database must be included in the same recovery scope of the BPM or BAM production environment. In the examples in this document, DB2 is the underlying database type.

3.2.1 Installation

For the database installation in the primary environment, follow the instructions in the DB2 installation manual to install and create the DB2 instance and related database users. For the secondary environment, install DB2 with the same installation path and instance name as in the primary environment and also the same user names and passwords used by DB2 in the system.

3.2.2 Configuration

The database configuration involves the creation of the database and tablespace. For the primary environment, manually create all the necessary databases for the environment and set the database path to the directory that is mounted on the NFS server. For the secondary environment, mount the same directory of the database server

on the NFS server. No configuration is required before restoration.

3.3 Environment considerations

3.3.1 Installation

As the root user, when you install the BPM or BAM environment, there are no special instructions for the primary environment. For the secondary environment, reinstall the environment with the same information, such as installation path, product version, and patch level, as in the primary environment.

3.3.2 Configuration

Configuration includes creating profiles and configuring cluster environments. When you create profiles in the primary environment, the profile path must be located in the directory that is targeted at the NFS server. In the secondary environment, the same directory of the corresponding operating system must be mounted on the NFS server. No configuration is required before the restoration.

To configure the cluster environment, follow the normal process of cluster configuration.

Figure 13 illustrates the entire test scenario example.

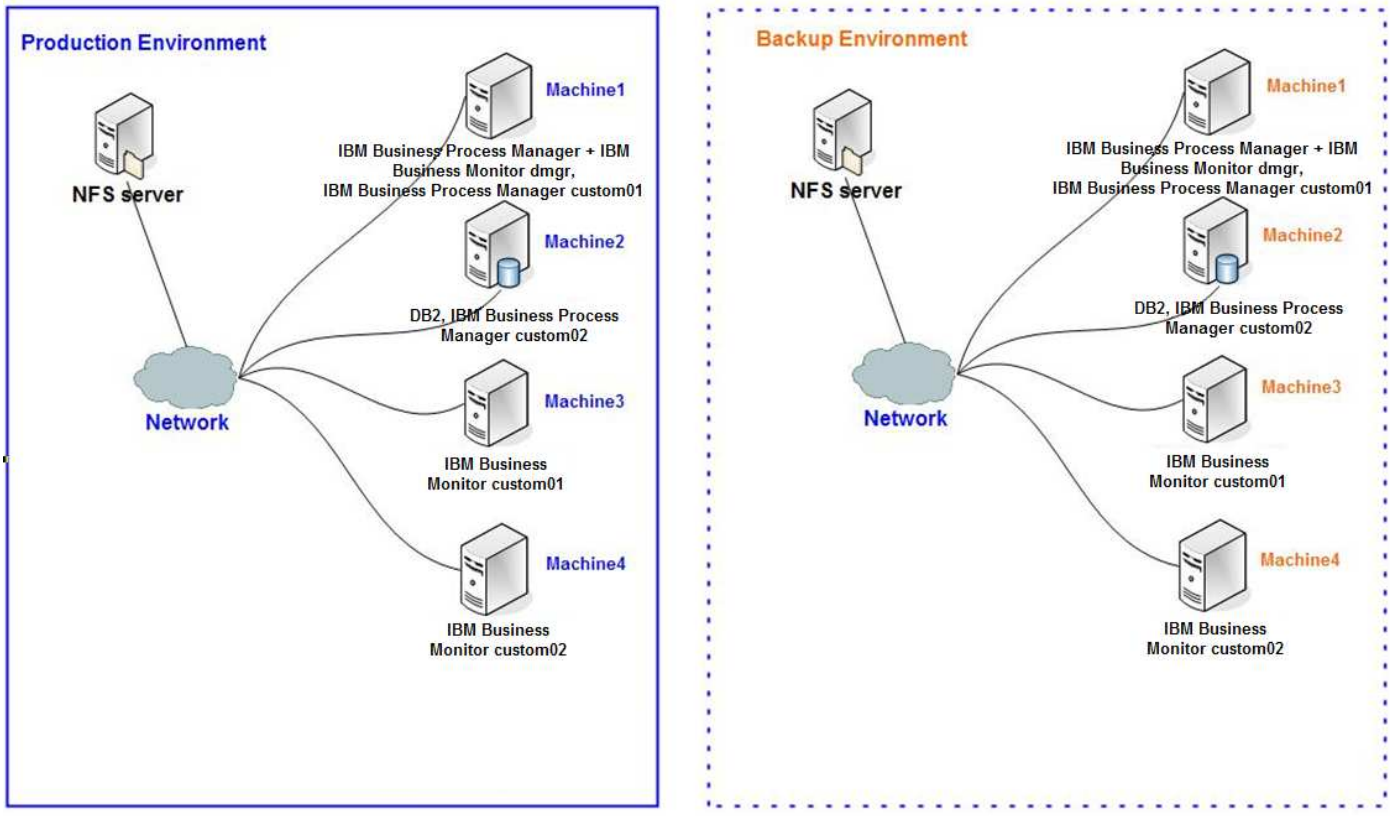

Figure 13: Example of BPM with NFS

This example contains a total of ten servers, five for each environment (NFS server, Machine1, Machine2, Machine3, and Machine4). Each server except the NFS server in the production environment has the same IP address and host name as the one in the primary environment. The NFS servers in the two environments have different IP addresses and host names.

DB2 and IBM Business Process Manager are all installed in the `/opt/ibm` directory under their installation servers. DB2 is installed on Machine2, IBM Business Process Manager is installed on Machine1 and Machine2, and IBM Business Monitor is installed on Machine1, Machine3, and Machine4. For DB2, the databases related to IBM Business Process Manager and IBM Business Monitor are created under `/home/db2`, and the `dmgr` and custom profiles for IBM Business Process Manager and IBM Business Monitor are created under `/home`. The `dmgr` files for IBM Business Process Manager and IBM Business Monitor are created on Machine1, IBM Business Process Manager custom profiles are created on Machine1 and Machine2, and IBM Business Monitor custom profiles are created on Machine3 and Machine4.

Figure 14 provides more information about the test scenario example. The structure in the figure is just an example. You can arrange your directories according to the requirements of your system.

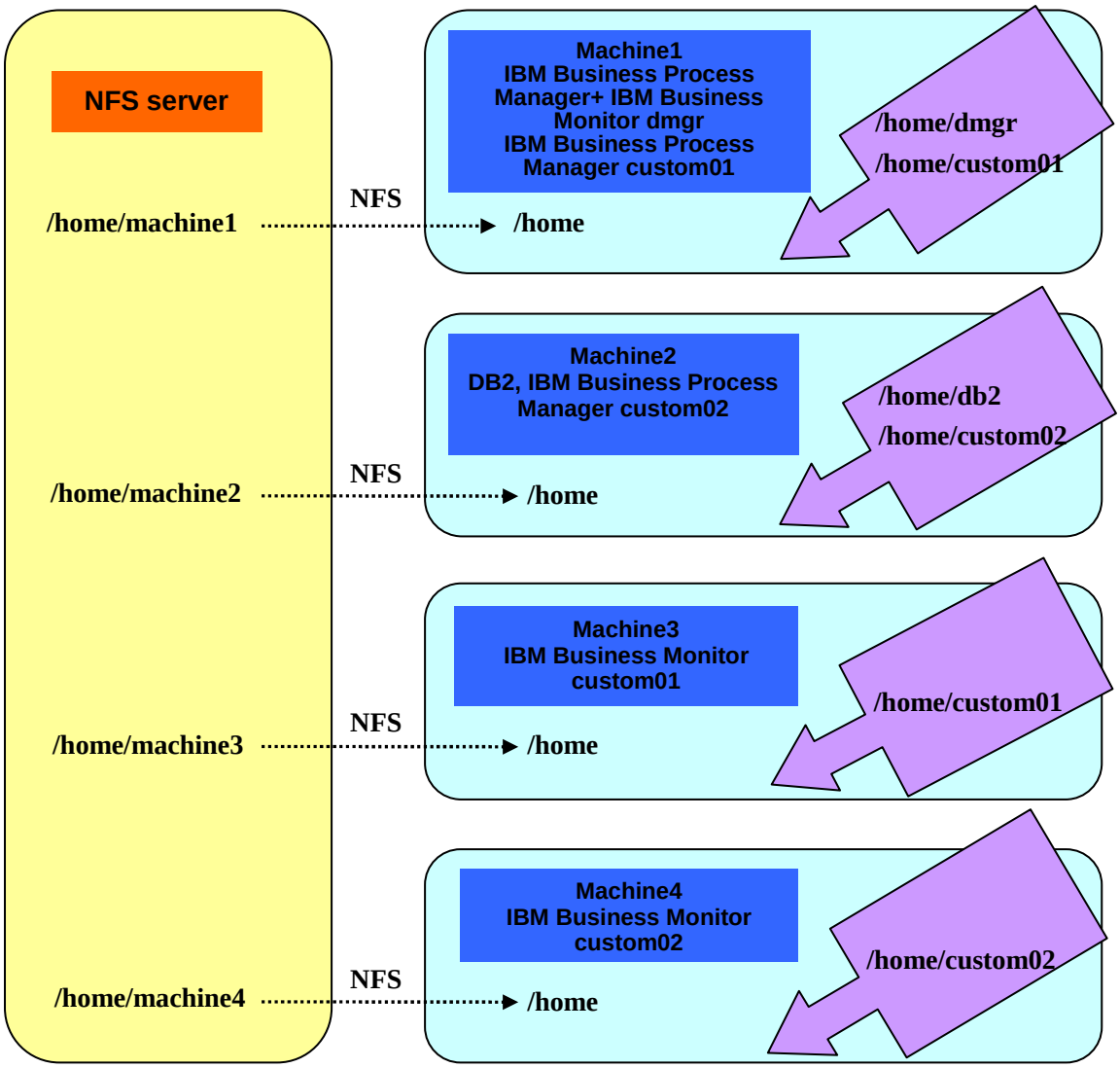

Figure 14: Example of BPM with NFS (Directory level)

With this structure, to back up all profiles and database files, you can conveniently take a snapshot of the `/home` directory under the NFS server. Alternatively, you could separate the runtime and configuration data and make snapshots for them individually. Keep the backups as small as possible because otherwise your processing time could be greater than your RPO.

4. Disaster recovery scenarios

In a real production environment, a backup occurs according to the backup schedule of the production environment. The production environment might go through various states while the backup is taking place. The following sections describe three typical scenarios for backup and restoration.

4.1 Configuration backup and restoration

After a configuration change such as creating a profile, configuring a deployment environment, or installing an application, you need to back up the configuration data of the primary environment. Then you need to verify whether the configuration change can be restored successfully in the secondary environment.

To verify the data for this scenario, complete the following steps:

1. After a configuration change, create a snapshot of the environment.
2. Restore the snapshot to the secondary environment.
3. To verify the secondary environment, start the whole environment in an isolated environment that does not share any resources with the primary environment.

After you verify the data, you should discover that the configuration changes are still valid in the secondary environment. It is safe to take a snapshot of configuration changes, because configuration changes are protected through the backup and restoration procedure.

4.2 Runtime backup and restoration

After you back up and restore the configuration and runtime data, you need to verify whether the current instances, such as long-running process instances, short-running process instances, SCA invocation instances, and IBM Business Monitor monitored instances, can be restored to the secondary environment. This is the most challenging scenario, and it requires special design considerations.

- Because RAM data will be lost during the backup and restoration procedure, you must depend on global transactions to keep data integrity.
- To ensure overall consistency, all modified resources inside the scenario design must be included in the same recovery scope.
- For asynchronous invocation, you can get different replay results because you can have different settings on the transaction boundaries. Because the transaction cannot pass through the boundary of caller and partner, a separate transaction context is required for both caller and partner, so that they can be restored through the DR procedure.

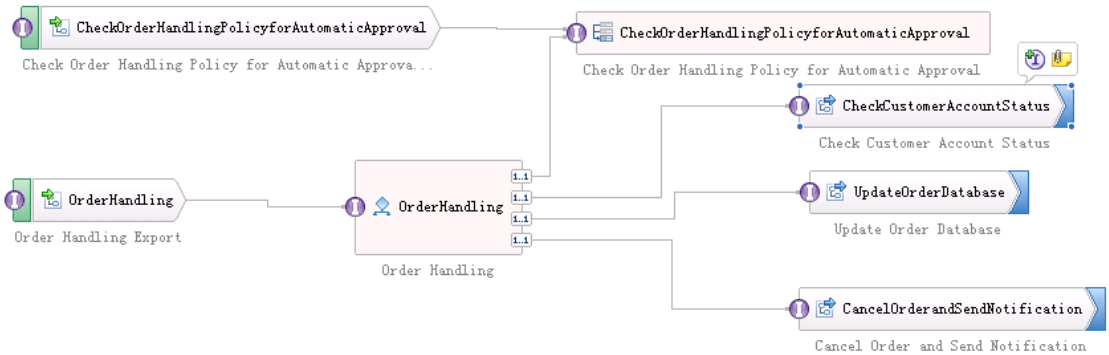

Figure 15: Runtime and backup restoration scenario

- The implementation for the `CheckCustomerAccountStatus` BPEL process from Figure 15 is shown in Figure 16. It is a long-running process. The transactional behavior for `Receive` is `Commit After`. For `Snippet`, it is `Participates`. For `Invoke`, it is `Commit After`. So when you are restoring in the backup environment, the strings in `Snippet` and `Invoke` will be printed.

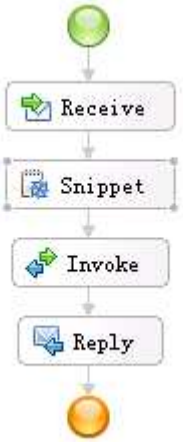

Figure 16: Transaction setting for CheckCustomerAccountStatus

- The implementation for the `CancelOrderandSendNotification` BPEL process from Figure 15 is shown in Figure 17. It is a microflow, so by default the transactional behavior is `Participates`. However, because the invocation style for `InvokeNotification` is synchronized and it is a one-way invocation, only the strings in `InvokeNotification` will be printed in the backup environment.

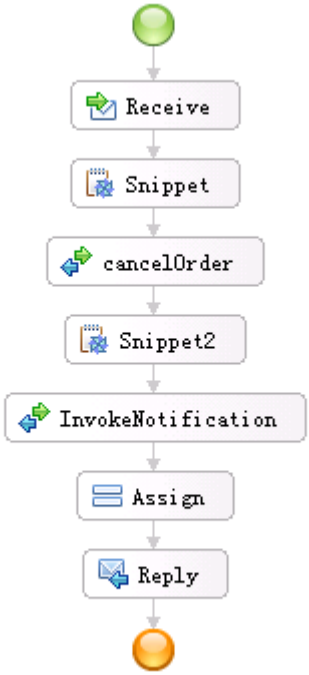

Figure 17: Transaction setting for CancelOrderandSendNotification

The testing scenario consists of the following steps:

1. The `OrderHandling` main process is a long-running process, which itself is contained in a global transaction context. During the navigation, the transaction might be demarcated by invocation or

human task activity; however, for each partition, it is still wrapped by a global transaction.

2. The `CheckCustomerAccountStatus` subprocess is a long-running process as the partner of the main process, which is contained inside a global transaction as well. It will be invoked through asynchronous invocation.
3. The `UpdateOrderDatabase` subprocess is a short-running process as the partner of the main process, which is contained inside a global transaction and invoked through asynchronous invocation.
4. The `CancelOrderandSendNotification` component is an SCA component and invoked as asynchronous one-way.

To verify the data for this scenario, complete the following steps:

1. Generate some load on the environment, and make sure that some instances are still running.
2. Take a snapshot of the environment.
3. Restore the snapshot to the secondary environment.
4. To observe the behavior of the restored environment, start the whole environment in an isolated environment that does not share any resources with the primary environment.

After you verify the data, you should discover that the running instances will continue for navigation in the secondary environment as normal and the instance states from IBM Business Process Manager and IBM Business Monitor are consistent. Through the persistence and transaction support of the underlying implementation, the running instances will continue to run through the backup and restoration procedure.

4.4. Scenario summary

For real production environment and application scenarios, you must test your backup and restoration procedure, so that you can identify any problems that might exist in your procedure.

When your primary environment comes back, perform a clean shutdown of the secondary environment and move all the data back to your primary environment. Start the primary system and switch all the connections back.

5. Summary

You should now understand the concept of disaster recovery (DR), the DR procedure of a typical BPM or BAM environment, and best practices for installation and configuration when you set up an environment that will support DR. You should also be able to recognize common scenarios that can occur through the DR process.