GHI Management Guide

GPFS / HPSS Management Guide
Release 2.2.0_patch4

January 2012 ***(Revision 2)***

Printed in the United States of America.

GHI 2.2.0_patch3
January 2012 (Revision 1)

TABLE OF CONTENTS

LIST OF FIGURES

Preface

The GHI Management Guide is intended as a resource for HPSS and site administrators. This document provides chapters contain the details for monitoring and managing a GHI system.

1. GHI BASICS

1.1. Introduction

The GPFS/HPSS Interface feature of HPSS (GHI) is software to connect GPFS and HPSS together under the GPFS Information Lifecycle Management (ILM) policy framework. This integration of GPFS with HPSS creates a hierarchical GPFS file system having virtually unlimited storage capability and provides the option to use the hierarchical capabilities of GPFS and HPSS to provide disaster recovery protection for the GPFS file systems. As an optional feature of HPSS, GHI is offered to HPSS users under the HPSS license agreement. GHI users are expected to acquire or have acquired GPFS under a separate GPFS license agreement.

1.2. GHI Capabilities

Both GPFS and HPSS scalability and performance are designed to meet the needs of data-intensive applications such as engineering design, digital media, data mining, financial analysis, seismic data processing and scientific research.

Typically, users tend to have a large number of files in a file system, and these may be any mixture of sizes from very small to very large. Both GPFS and HPSS are highly scalable, and are capable of ingesting thousands of files per second at rates limited by the hardware – usually the storage hardware and the transfer media. The GHI is a scalable extention of HPSS.

A primary goal of GHI is to offer an integrated Hierarchical Storage Management (HSM) and backup solution for GPFS. GHI uses and extends GPFS ILM capabilities, providing a cost-efficient integrated storage solution that is scalable to 100s of petabytes and billions of files. GHI enables GPFS file data transfers between GPFS high performance storage, usually high-speed disk, and HPSS cost-efficient storage, usually high capacity disk and tape. This movement between GPFS and HPSS occurs automatically under control of GPFS ILM rules, thus providing a complete and scalable HSM and backup solution that exploits HPSS parallel file system capabilities. In order to accomplish these goals, GHI is designed and implemented based on the concepts described in the following subsections.

1.2.1. Network-centered Architecture

The focus of the GHI feature of HPSS is the network. GPFS and HPSS are both network-centered cluster solutions offering horizontal scalability by adding cluster components. The GHI feature extends this architecture. Thus, the archive is not a single processor as in conventional storage systems. GHI provides servers that can be distributed across a high performance network to provide scalability and parallelism.

1.2.2. High Data Transfer Rate

GHI uses the Parallel I/O ((PIO) interface provided as part of the HPSS Client Application Program Interface (Client API) to support parallel access to storage devices for fast access to very large files stored in HPSS. The I/O Manager organizes and manages the data transfer. The I/O Agents collect and transfer the data. Multiple I/O Agents are used based on the HPSS stripe width of the COS configuration.

For small GPFS file data transfers, GHI uses a modified GHI-specific version of the HTAR program, known as "GHI-HTAR"**.** GHI-HTAR is used for aggregating a set of files from GPFS directly into HPSS. It uses a multithreaded buffering scheme to write files directly into HPSS, thereby achieving a high rate of performance.

1.2.3. Standard Components

GHI is written in ANSI C. It uses Remote Procedure Calls (RPC), Kerberos or UNIX for server authentication, and DB2 as the basis for its portable, distributed architecture for maintaining GPFS backups.

The GHI system is supported on IBM AIX and RedHat LINUX platforms.

1.2.4. DB2 Components

GHI uses DB2 to maintain a history of the GHI backups. The DB2 Server configured on HPSS is used to store the backup tables(s) for GHI. There is one table configured in the GHI database on the HPSS Core Server for each GHI enabled GPFS file system. The GHI Session nodes will be be configured as DB2 clients to access the backup tables during GHI backup and restore operations.

1.2.5. Terminology

Some of the following terms are overloaded, meaning GPFS and HPSS have different meanings for the same term. GHI uses the HPSS terminology.

- ***Backup*** - Backup refers to backing up a GPFS file system into HPSS. The information needed to restore a GPFS file system, including the restoration of the cluster will be backed up into HPSS as well.
- ***Cluster*** - A loosely-coupled collection of independent system nodes organized into a network for the purpose of sharing resources and communicating with each other.
- ***Garbage Collection*** - This involves removing GHI files from HPSS that are no longer referenced by GPFS or a valid backup. (See Section *1.2.6.4* Garbage Collection and GPFS File Deletion*)*
- ***Migration*** - Migration refers to the movement of file data from GPFS to HPSS, while maintaining the file data in GPFS. There are two scenarios where migration are performed. The first scenario is when the GPFS

policy engine is run to transfer file copies from GPFS to HPSS. The second scenario is during a backup, which is when the most recent version of all files that have not been copied during a policy triggered HSM migraton are coped to HPSS. The GPFS term is “pre-migration”.

- ***Purge*** – Purge refers to freeing up data segments in the GPFS file to free up GPFS resources. A GPFS policy is used to trigger a threshold request. The data blocks for the selected files are freed, leaving a stub in GPFS. The GPFS term is “migration”. This can also be referred to as “punching a hole” in a file.
- ***Recall*** - Recall refers to the movement of file data from HPSS to GPFS. Candidates are based on files that are not dual-resident and the data only resides in HPSS. This is an asynchronous request. A GPFS ILM policy is used to copy file data from HPSS to GPFS. The GPFS term is “pre-stage”.
- ***Restore*** – Restore refers to the capability to restore either a GPFS file system or a GPFS cluster from a selected backup.
- ***Stage*** - Stage refers to the movement of file data from HPSS to GPFS. This process is invoked by accessing a file that is not dual-resident, and the data only resides in HPSS. This is a synchronous event. This process generates a DMAPI I/O event to stage the file back. A response is sent to the user when then operation is complete. This is sometimes referred to as “stage on-demand”.
- ***Pin*** – Pin or pinning refers to flagging a file so that it can not be purged from the GPFS file system.
- ***GHI Metadata –*** Any data necessary to reconstruct the GPFS file system namespace. Metadata consists of GHI database, GHI backup files and aggregated index files.
- ***GHI User Data –*** User data consists of GPFS data files and GPFS data file attributes. The file attributes consist of UID, GHI, access time, modified time, DMAPI attributes, symbolic/hard link information.

1.2.6. HSM Concept of Operation

GHI uses the GPFS ILM policy driven storage management to efficiently provide migration and staging of GPFS files through tiered storage. To copy file data from GPFS to HPSS storage, the GPFS policy engine is used. GHI supports the following ILM mechanisms to transfer data between GPFS and HPSS, as well as manage available space in the GPFS file system:

- Data Migration.
- Data Recall.

- File system limits.
- Garbage Collection

GHI also uses the GPFS DMAPI interface to stage data back from HPSS on-demand when a user requests access to the GPFS file data (i.e. open, chksum, etc).

1.2.6.1. Transferring data into HPSS

Files are transferred (i.e. migrated) from GPFS into HPSS based on rules sent to the GPFS policy engine. A migration policy is used to provide a set of rules to determine which GPFS files are candidates to be transferred to HPSS. The policy engine can generate two lists of files to be migrated: One list contains the files to be aggregated; the other list contains non-aggregate files. Next, the policy engine invokes the following GHI script:

- ***ghi_migrate*:** One or more instances of the script is invoked by the policy engine to coordinate with the GHI scheduler to migrate the files to HPSS. For aggregation, files are placed in groups of an "aggregate bulk size" so that each ghi_migrate recieves a request for a single aggregate. For non-aggregates, a single ghi_migrate instance will receive a list of several files to be processed based on a "non-aggrregate bulk size".

GHI provides the following migration template in the ***/var/hpss/ghi/policy*** directory as an example of how to generate a list of files to be migrated:

- ***migrate.policy***: The template provides rules to split files to be migrated into two categories: aggregates and non-aggregates, and bulk the aggregates based on a bulk count.
- ***Migrate_size.policy***: The template provides rules to split files to be migrated into two categories: aggregates and non-aggregates, and bulk the aggregates based on the total aggregate size.

1.2.6.2. Transferring data from HPSS

Files are transferred from HPSS to GPFS based on two scenarios:

- ***Recall Operations***: Recall files from HPSS as a background or scheduled task. GPFS policy rules can be defined to retrieve the file data in advance of a user request to access those files. The policy rules create a list of files that are candidates that are eligible for recalling back from HPSS. The list of candidates are sent to GHI to be sorted. The sorting will group files based on location in HPSS. Files that reside on the same tape will be recalled together to minimize tape mounts. Files that reside in the same aggregate will be recalled together, using a single GHI-HTAR request.
- ***Stage Operations***: Stage files back from HPSS synchronously when file contents are accessed.

1.2.6.2.1 Recall Operations

The recall policy provides a set of rules that are used to determine which files are to be copied from HPSS to GPFS. The GPFS policy engine generates one list of files to be recalled and then invokes the following GHI script:

- ***ghi_recall***: One instance of the script is invoked by the policy engine to coordinate with the GHI scheduler to recall the files from HPSS. The scripts parses through the list and generates buckets of requests based on files belonging to the same aggregate and files residing on the same tape. This will optimize the retrieval of the data.

GHI provides a recall template in the ***/var/hpss/ghi/policy*** directory as an example of how to generate a list of files to be recalled:

- ***recall.policy***: The template provides rules to split files to be migrated into two categories: aggregates and non-aggregates.

1.2.6.2.2 Stage Operations

When files reside in HPSS, regions are placed on the files. When a user accesses the file data DMAPI events are generated. The stage operation for a GPFS file is performed differently depending on where the data resides.

If a file resides in both GPFS and HPSS, a WRITE or TRUNCATE event is generated when the user updates the file. This does not cause the file to be staged, since it still reside in both places. It does, however cause GHI to clear out the DMAPI regions since the file contains new data and needs to be migrated again.

If a file only exists in HPSS, which means that no data resides in GPFS, a READ event is generated when the user opens the file in GPFS. This causes the file to be staged from HPSS and become dual resident.

1.2.6.3. **Managing Available Space**

To monitor the high and low water marks for a GPFS file system, the file system is enabled by attaching a set of policies rules to the file system. When the system triggers a high (NO_SPACE) or low (LOW_SPACE) event, the GPFS policy engine generates a list of files to be purged from the file system.

The following GHI provided script is invoked when the event is triggered:

- ***ghi_migrate***: The script sends the list of files to be purged to the GHI Scheduler Daemon.

GHI provides a threshold template in the ***/var/hpss/ghi/policy*** directory as an example of rules to generate a list of files to be purged:

- ***threshold.policy***: The template provides rules to generate purge candidates.

1.2.6.4. Garbage Collection and GPFS File Deletion

GHI Garbage collection is the removal of unreferenced GHI files from HPSS. Unmanaged files are GHI files in HPSS that no longer exist in GPFS and are not referenced by a backup of the GPFS file system. Files are not referenced when:

- They have not been migrated into HPSS.
- The file is not part of a valid backup

GPFS will notify GHI when a file is deleted or updated. GHI will process each notification to see if the file is a candidate for garbage collection.

- If the file is not GHI managed it is deleted from GPFS and GHI has no action to take.
- If the file is GHI managed but not part of a backup GHI can immediatly delete the file from HPSS.
- If the file is part of a backup GHI must store the file information in the Garbage Collection table until the referencing backups are deleted.
- If the file is part of a backup that is invalidated due to a restore the file will be marked as an orphan.

Backups and orphaned files are deleted using the backup manager tool. This will removed the backup metadata files from HPSS. The GC table entries are scanned to see if any of the files are now unreferenced. If so the associated HPSS files are unlinked.

1.2.7. Concept of Operation for Backup

There are two separate types of backup: GPFS Cluster and File System information. The following subsections describe each of these types of backup.

1.2.7.1. GPFS Cluster Information

The GPFS cluster information is initially stored in HPSS when the system is newly GHI enabled. Then, each time the cluster is modified, GPFS invokes the GHI script, ***ghi_backup***, to backup the GPFS cluster configuration file into HPSS again.

1.2.7.2. GPFS File Information

The ability to backup GPFS is accomplished using the GHI ***ghi_backup*** interface. The backup interface uses the ILM policy management interface which uses the following backup policy templates that resides in the ***/var/hpss/ghi/policy*** directory. Each file system has it's own copy of these policies in the

/var/hpss/ghi/policy/<filesystem> directory:

- ***Backup_migration.policy***: The backup migration policy contains the migration rules for the GPFS file system being backed up. The rules can migrate files as aggregate or non aggregates. A majority of the rules in the backup_migration policy should match the normal migration policy. However the last rule in the policy should select any file that has not already been migrated into HPSS. This ensures that all the files will be migrated so they can be backed up.
- ***Backup_metadata.policy***: The metadata policy should only be changed after very careful consideration. It contains 2 rules:
  - ***Name space*** – generates files containing lists of all the files/directories in the file system. These output files are copied directly into HPSS and are used to rebuild the GPFS namespace during a restore.
  - ***changed files*** – generates one or more lists of files used to populate output files that contain the file attributes and DMAPI attributes for the files that have changed. If a full backup is specified, this list will contain all the files in the file system. Otherwise, for an incremental backup, list list contains only the files that have changed since the previous backup.

GHI backups use the GPFS snapshot feature to take a point-in-time image of the file system. When running the backup process, a snapshot of the GPFS namespace is saved, and the state of each of the files is saved. When migrating files to provide a complete backup, GHI uses the snapshot, instead of the active file system. When a snapshot is in place, and files are modified, a copy of the file contents are saved in the "copy on write" file system.

1.2.7.3. Restore Concept of Operation

There are two separate types of restore: GPFS Cluster and File System information. The following subsections describe each of these types of restore.

1.2.7.3.1 GPFS Cluster Information

There is only one copy of the cluster information file stored in HPSS. Each time the cluster configuration is modified, the cluster information file is re-archived into HPSS. To restore the cluster, the file is retrieved from HPSS and written to the specific GPFS directory. The file can then be distributed to all the nodes in the cluster, and the cluster can then be put online. If disk resources have changed during recovery, the administrator can modify the cluster configuration file with the appropriate changes once the file is restored and before putting the cluster online.

1.2.7.3.2 GPFS File Information

GHI provides a restore utility, ***ghi_restore***, to rebuild a GPFS file system after a catastrophic failure. The GHI restore utility allows the administrator to display the full backups stored in HPSS, and if selected, displays each of the incremental backups associated with a full backup. For restoring a file system, the restore process is broken into four phases:

- Restore of the namespace (directories, filenames, hard links, and symbolic links).
- Restore of the the associated attributes (owner, permissions, etc).
- Recall of file data resident on the file system when the backup was taken. (this step is a separate procedure invoked by the administrator).
- Mark future backups as invalid and mark files associated with those backups as orphans.

When complete, administrations should define recall rules to stage file data back from HPSS.

1.2.8. Storage in HPSS – Where Do the Files Go?

The following subsections provide information on GHI object mappings and describe how GHI uses the mapping information.

1.2.8.1. Extended Attributes

To map GPFS file system objects to an HPSS object, mapping information is stored in the GPFS extended attributes. The information contains:

- ***HPSS Identifer*** – An unique identifier to locate where the GPFS file contents are archived in HPSS.
- ***Aggregate Flag*** – A flag to indicate whether the file is in an aggregate or not.
- ***Ordinal*** – The index into the aggregate index file for the member. (applies to aggregates only).
- ***Snaphot Identifier*** – This identifier associates the GPFS object with the backup in which the object was last backed up into HPSS.
- ***Version Number*** – This is used to determine the format of the contents.

A seperate DMAPI attribute contains the flag that indicates a file has been pinned.

1.2.8.2. Location of files in HPSS

1.2.8.2.1 Migrated files

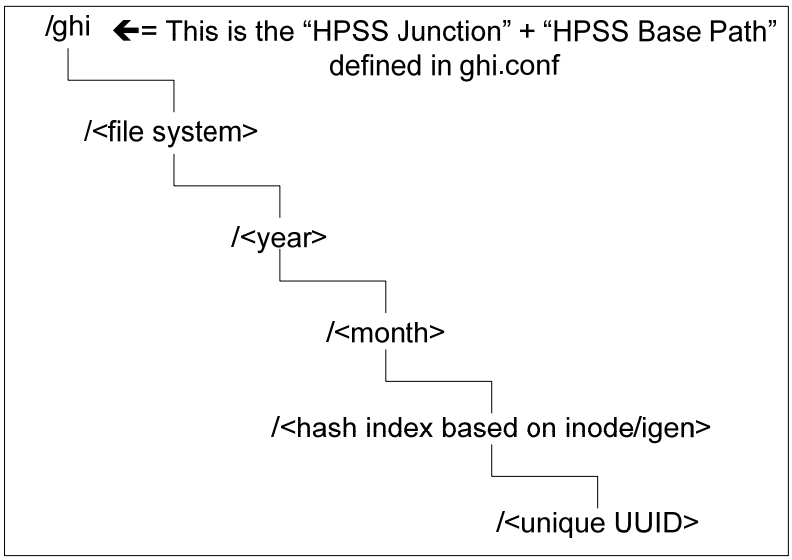

Figure 1 - Location of HSM Files in HPSS

Figure 1 - Location of HSM Files in HPSS shows the location of HSM files in HPSS. Hashing directories are created to store the GPFS files in HPSS. The directories are generated based on the following information:

- ***/ghi*** – This is the default "root" directory in HPSS for storing the GPFS files. It is composed of the HPSS Junction and HPSS base path that is defined in the ghi.conf. This value must not be changed.
- ***file system*** – GPFS file system name.
- ***timestamping criteria*** - "*year/month*".
- ***hash directory*** – For non-aggregate files, the inode and igen are used to determine the directory. For aggregate files, the file's UUID is used to determine the directory. The index and data file for the aggregate are placed in the same directory.

1.2.8.2.2 Backup files

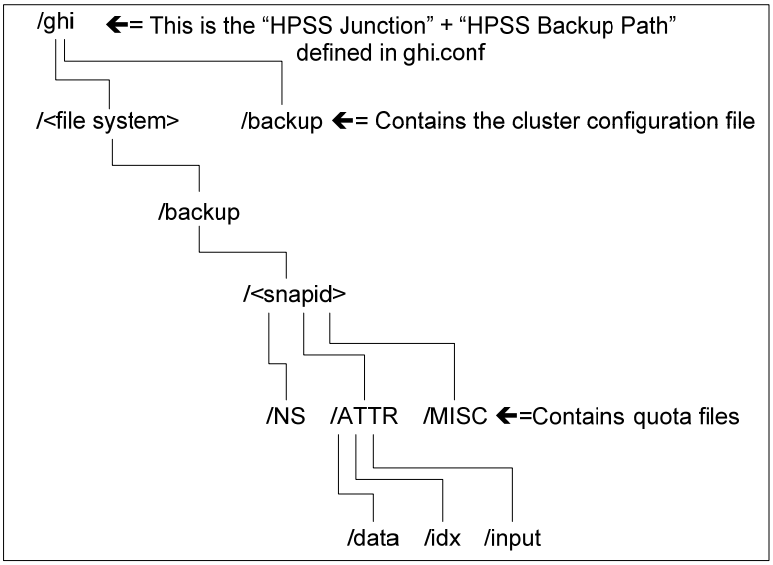

Figure 2 - Location of Backup Files in HPSS

Figure 2 - Location of Backup Files in HPSS shows the location of the backup files in HPSS. Backup files are stored in HPSS based on the snapshot ID at the time of the backup.

- ***/ghi*** – This is the default "root" directory in HPSS for storing the GPFS files. It is composed of the HPSS Junction and HPSS backup path that is defined in the ghi.conf. This value must not be changed.
- ***File system*** – GPFS file system name.
- ***Snapid*** – Snapshot identifier.
- ***NS, ATTR and MISC directories*** – Based on type of files being backed up.
    - ***NS*** – contains the GPFS namespace information.
    - ***ATTR*** – contains the GPFS file attribute information.
    - ***MISC*** – contains the GPFS quota files and the version file.Add reference to teh version file

1.2.8.3. HPSS Class of Service (COS)

Each file in HPSS has an attribute called Class Of Service. The COS defines a set of parameters associated with operations and performance characteristics of a file. The COS results in the file being stored in a storage hierarchy suitable for its anticipated and actual size and usage characteristics.

The following rules are defined for COS selection:

- ***Data files (aggregate and non-aggregate)*** – Selected based on *Maximum File Size Hints*.
- ***Aggregate index files*** – Selected based on the GHI configuration.
- ***Backup Files*** – Selected based on the GHI configuration.

The administrator can also specifiy a COS for individual rules in a policy run. This allows a site administrator to further configure policies to direct file candidates to specific Classes of Service.

1.2.8.4. HPSS File Families

HPSS files can be grouped into families. HPSS supports grouping files on tape volumes only. All files in a given family are stored on a set of tapes assigned to the family. When one of these files is migrated from disk to tape, it is stored on a tape with other files in the same family. If no tape volume associated with the family is available, a blank tape is reassigned from the default family. The family affiliation is preserved when tapes are repacked. File Families can be specified for each rule in a policy run.

1.2.8.5. HPSS Storage Subsystems and Scalability

Storage Subsystems can be used to separate HPSS resources, so that GPFS HSM files can be placed on their own resources. GHI currenly supports one subsystem per GPFS file system.

1.3. GHI Components

The GHI components (see *Figure 3 - GHI Components*) consist of the GHI servers that provide management for the DMAPI enabled, GPFS file systems. The servers process DMAPI events for mounting and unmounting file systems, as well as process I/O (READ, WRITE and TRUNCATE) events.

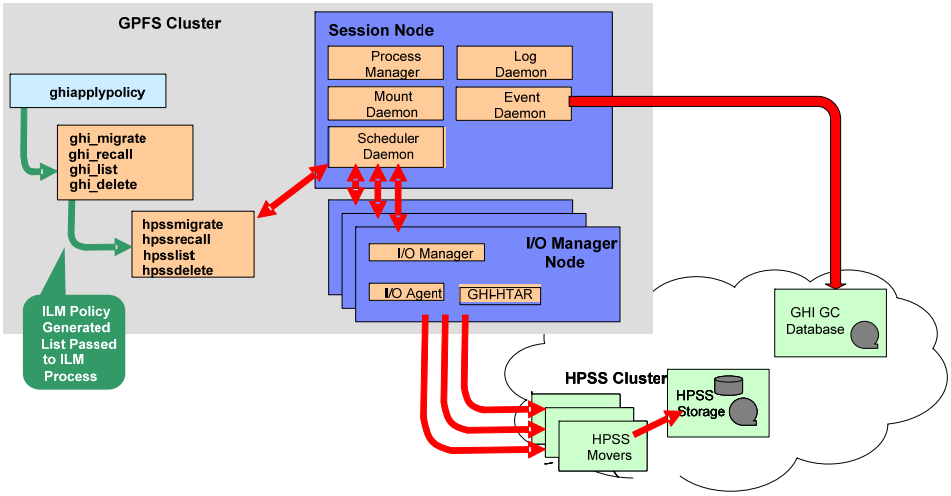

Figure 3 - GHI Components

1.3.1. GHI Servers

GHI consists of the following processes:

- ***Process Manager (PM):*** The Process Manager runs on the GHI Session Node. It is responsible for the following activities:
    1. Starts and stops the other GHI processes.
        a. On startup, the Process Manager starts the Mount Daemon and Log Daemon as a child process.
        b. The Process Manager starts an Event Daemon and a Scheduler when the file system is mounted on the Session node.
    2. Ensures the Mount Daemon, Log Daemon, Event Daemon(s), and Scheduler Daemon(s) stay functional and that they are able to perform work and minimize system hangs from occurring.
    3. In the event of the death of one of the child processes, the Process Manager receives a SIGCHLD signal. After receiving the signal, the Process Manager restarts that process.

The Process Manager is started by GPFS as part of the GPFS heartbeat mechanism. It is started as part of the GPFS cluster configuration manager node start up process.

- ***Event Daemon (ED):*** The Event Daemon runs on the Session node. It is responsible for the following activities:
    1. Registers for DMAPI I/O (DESTROY, READ, WRITE and TRUNCATE) events.
    2. Recieves read, write, and truncate events for files from the DMAPI session queue and submits the requests to the GHI Scheduler Daemon.
    3. Recieves destory events for files from the DMAPI session queue and performs garbage collection logic on the file.
    4. Receives responses from the GHI Scheduler Daemon and responds to the user request with the result.
- ***Log Daemon (LD):*** The Log Daemon runs on the Session Node. It is responsible for the following activities:
    1. Maintains two rotating central log files (similar to the HPSS log files).
    2. Logs Event, Minor, Major and Critical log messages from all GHI components.

3. Archives full log files to HPSS.

- ***Mount Daemon (MD):*** The Mount Daemon runs on the Session Node. It is responsible for the following activities:
    1. Captures mount and unmount events for DMAPI enabled file systems.
    2. Processes remote mounts for DMAPI enabled file systems.
- ***Scheduler Daemon (SD):*** The Scheduler Daemon runs on the Session Node. It is responsible for the following activities:
    1. Communicates with the I/O Managers to transfer data.
    2. Provides a mechanism to pass back transfer results to the clients.
    3. Provides load balancing to the IOMs.
    4. Processes purge requests for threshold processing.
    5. Filters out duplicate file requests.
- ***I/O Manager (IOM)*****:** The IOM runs on one more more nodes in the GPFS file system. The IOM is responsible for the following activities:
    1. Spawns GHI-HTAR to perform aggregate data transfers.
    2. Spawns IOA to perform non-aggregate data transfers.
    3. Gathers GPFS metadata and namespace information to backup GPFS.
    4. Rebuilds the GPFS namespace for file system restores.
- ***I/O Agent (IOA)*****:** The IOA runs on one more more nodes in the GPFS file system. The IOA is responsible for:
    1. Performing non-aggregate data transfers
- ***GHI-HTAR/HSIGWD*****:** Provides an HPSS interface for aggregating and retrieving small files. GHI-HTAR clients resides on each node that the I/O Manager resides on. The HSIGWD server resides on HPSS nodes that are assigned by the HPSS administrator.

1.3.2. GHI Infrastructure

The GHI infrastructure items are those components and services used by the various GHI servers. The RPC communication between each of the GHI processes are shown in *Figure 4 - Intra-process Communication.* The GHI infrastructure components common among servers are discussed below.

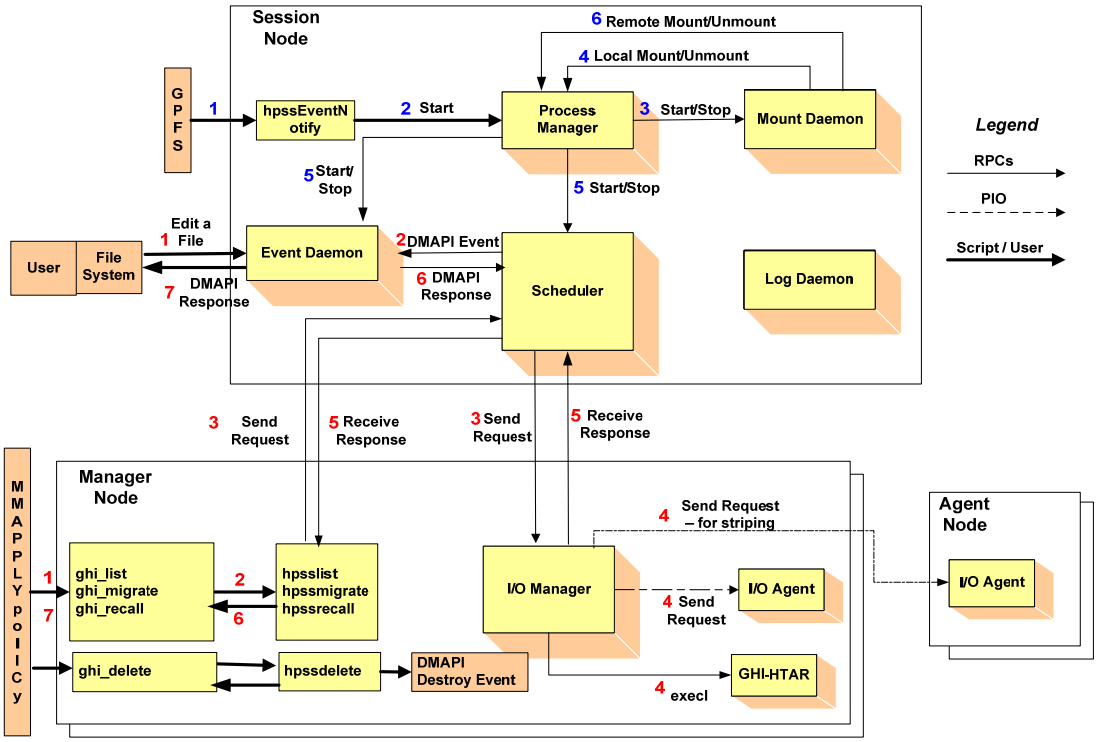

Figure 4 - Intra-process Communication

1.3.2.1. Remote Procedure Calls (RPC)

Most GHI servers, with the exception of the I/O Agent, communicate requests and status (control information) via RPCs. GHI does not use RPCs to transfer user data. RPCs provide a communication interface resembling simple, local procedure calls.

1.3.2.2. Thread Services

GHI uses a threads package for multitasking. The threads package is vital for GHI to serve large numbers of concurrent users and to enable multiprocessing of its servers.

1.3.2.3. Security

GHI uses HPSS security software to allow GHI components to communicate in an authenticated manner, to authorize access to HPSS objects, to enforce access control on HPSS objects, and to issue log records for security-related events. The security components of HPSS provide authentication, authorization, enforcement, and audit capabilities for the HPSS components.

- ***Authentication***: responsible for guaranteeing that the GHI principal, ***hpssdmg***, is the entity that is claimed, and that information received from an entity is from that entity.

- ***Authorization***: responsible for enabling an authenticated entity access to an allowed set of resources and objects. Authorization enables end user access to HPSS directories and files.
- ***Enforcement***: responsible for guaranteeing that operations are restricted to the authorized set of operations.

GHI components that communicate with each other maintain a joint security context. The security context for both sides of the communication contains identity and authorization information for the peer principals as well as an optional encryption key.

1.3.2.4. Logging

A logging infrastructure component in GHI provides an audit trail of server events. Logged data includes alarms, events, requests, migration and backup information. The GHI Log Daemon, maintains a central log. When the central log fills, messages are sent to a secondary log file. When the log file rolls over, the full log file is sent to HPSS to be archived.

GHI-HTAR and the HSIGWD provide separate logs that contain audit trail events and file transfer information. Currently, these logs must be manually maintained. SYSLOG logging is also available as a configuration option, and the standard SYSLOGD mechanisms may be used to control the number and size of the audit trail logs.

1.3.3. GHI User Interfaces

GHI provides the user with a transfer interface, ***ghiapplypolicy***. The interface is a wrapper used to control the location where output files from the policy run are placed. It also controls the number of entries, i.e. bulk rates, of each of the output files.

1.3.4. ILM Policies

There are a number of aspects of storage management that will differ at each GHI site. For instance, sites typically have their own guidelines or policies covering how they want to implement data migration/recalls. In order to accommodate site-specific policies, GHI has implemented a set of policy templates to be used as guidelines to allow a site adminstrator the freedom to tailor management operations to meet their particular needs. Refer to the *GPFS Advanced Administration Guide* for implementation of the ILM policies.

1.3.4.1. ghi_backup

This utility allows a site administrator to backup GPFS metadata into HPSS. The ILM policy management interface is used to generate lists of files that need to be migrated, the file system namespace information, and a list of the GPFS files to be used to gather the file attributes. *Figure 5 GHI Backup Functionality* shows the

steps that are taken to complete a backup.

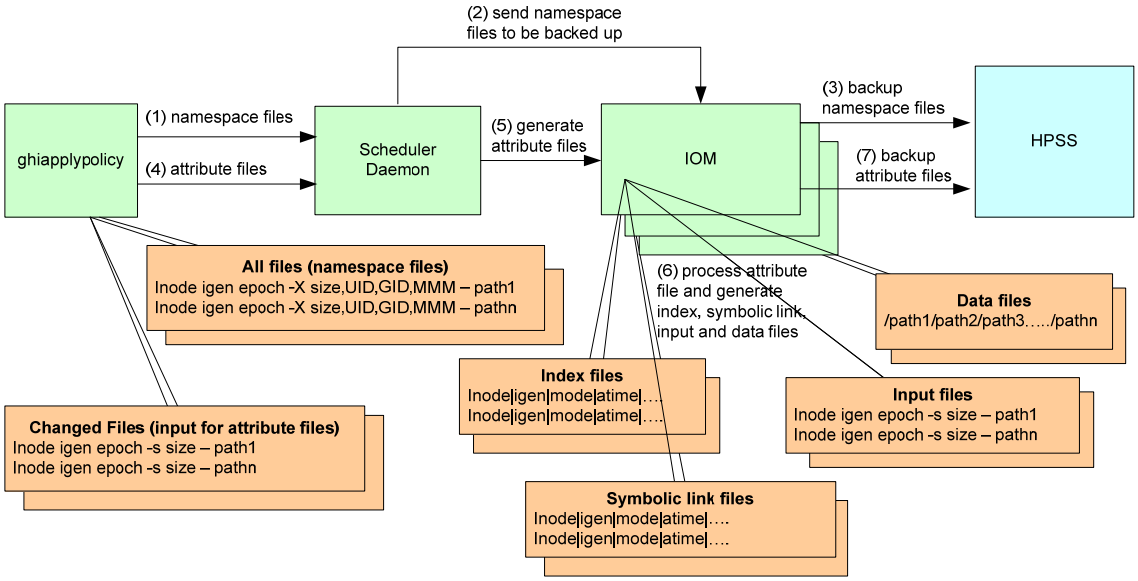

Figure 5 GHI Backup Functionality

Both full and incremental backups are supported. Full backups gather the attributes for all the GPFS files in the file system. Incremental backups only gather the attributes for the files that have changed since the previous snapshot.

Snapshots are used to capture a point in time snapshot for the file system. The file system is quiesced before capturing the namespace and attibutes for the files being backed up.

1.4. GHI Hardware Platforms

A typical GHI system configuration consists of a single Primary Session Node, one or more designated Secondary Session Nodes for fail-over, multiple I/O Manager Nodes, and multiple Client nodes, which do not run any GHI software components. The Secondary Session Nodes can act as I/O Manager Nodes until they are told to take over as the Primary Session Node. The following diagram in *Figure 6 - GHI Hardware Platforms* provides an example of a typical GHI system configuration.

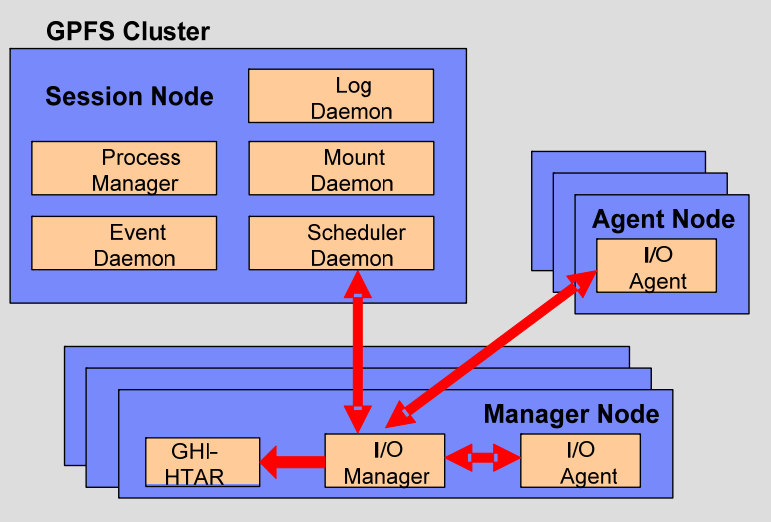

Figure 6 - GHI Hardware Platforms

1.4.1. Session Node Platforms

The Session Node is a machine where a DMAPI session has been instantiated and it has registered to receive DMAPI events. This node also functions as the GPFS Cluster Manager Node. The following GHI servers run on the Session Node:

- Event Daemon (one instance per file system).
- Log Daemon (one instance per cluster).
- Mount Daemon (one instance per cluster).
- Process Manager (one instance per cluster).
- Scheduler Daemon (one per file system).

1.4.2. I/O Manager Platforms

I/O Managers are used to control the logical network attachment of storage devices and are configured to run on one or more nodes. An IOM consists of two parts: The administrative process, I/O Manager, and one or more local and/or remote I/O Agent processes that handle the data transfers. These processes are supported on both AIX and Linux platforms.

This node performs data transfers between GPFS and HPSS. The following GHI servers run on the IOM node:

- I/O Manager.
- One or more I/O Agents.
- GHI-HTAR.

1.4.3. I/O Agent Platforms

This node performs data transfers between GPFS and HPSS as requested by the I/O Manager. The following GHI servers run on this node:

- One or more I/O Agents.

1.5. Process/Node Failover/Recovery

GHI provides a feature to recover from any failed GHI process (either restart the process on the same node, restart the process on a new node, or have another 'like' process take over the work load). The current implementation takes advantage of the GPFS "user exit" mechanism that provides the capability to determine failure of the GHI Session node, and fail-over of the GHI processes to a Secondary Session node.

GHI handles several types of failover scenarios:

- Session node failure or loss of quorum.
- GHI Session node process failure.
- IOM failure.

1.6. Disaster Recovery Plan

GHI metadata disaster recovery requires full consideration by the administrator and the HPSS service team during the "Planning Process". The degree to which the customer wishes to protect GHI metadata and user data, and provision for the protection and recovery of GHI metadata and user data will be documented by the customer and reviewed by IBM. Refer to the *HPSS Disaster Recovery Guide* for more information.

2. GHI PLANNING

2.1. Overview

This chapter provides GHI planning guidelines and considerations to help the administrator effectively plan and make key decisions for utilizing an HPSS GHI system.

Careful planning is required to fully consider how the resulting system will operate in an efficient manner and best meet site requirements.

The following sections describe the preparation steps for the GHI installation, configuration, and operational phases.

2.1.1. GHI System Architecture

Figure 7 - GHI System Architecture, shows the basic architecture of an HPSS GHI system and their relationship to HPSS server nodes, and HPSS Mover nodes.

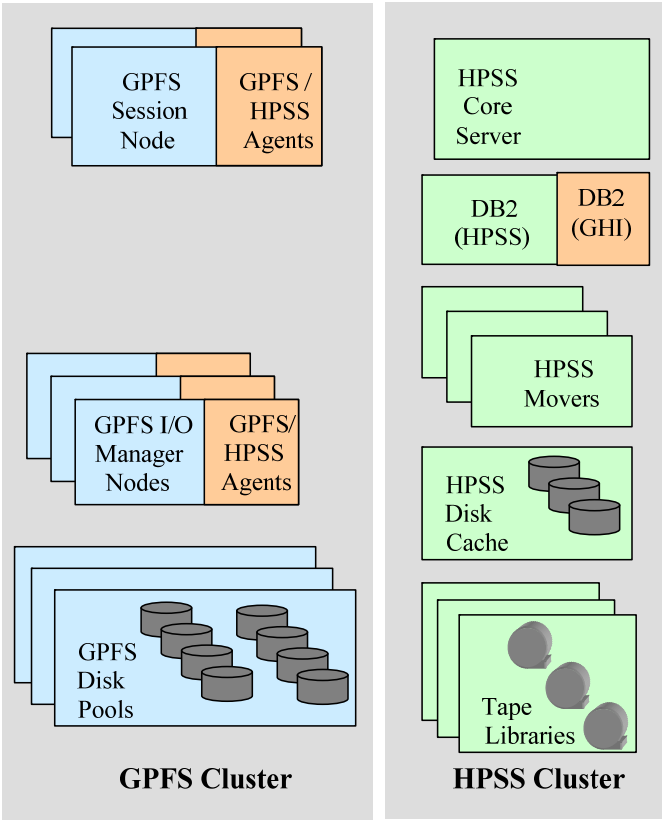

Figure 7 - GHI System Architecture

Specifics of this architecture for a given site are developed during the proposal and initial project planning stages of a deployment. Often the disk and tape data

resources for GPFS and HPSS are dictated by equipment already available and budgetary constraints on what can be purchased. Specific quantities and sizing of these data resources are beyond the scope of this planning document. For this document, it is assumed these parameters were already defined. Contact your HPSS service provider with questions about and for assistance with GHI resource sizing and configuration. See section *2.4 GHI Sizing Considerations* for sizing considerations for the metadata resources as well as the local disk resource requirements.

2.1.2. GHI Configuration Planning

When planning to provide HSM space management and backup and restore services for GPFS using HPSS, it is very important to review the following from the initial proposal and planning of the system:

- The user's or HPC bandwidth and capacity requirement on GPFS.
- The additional capacity requirements being imposed on the GPFS file system to hold the extended attributes used by GHI.
- The additional bandwidth needed to support the HSM activity between GPFS and HPSS.
- The bandwidth and capacity requirements of HPSS for HSM file data and any backup and restore requirements of the solution.

When looking at the GPFS requirements, the following items should be considered important and should be reviewed by the customer:

- Verify the site's storage requirements and policies, such as the initial storage system size, anticipated growth, usage trends, average file size, expected throughput, backup policy, and availability.
- Review the architecture of the entire Storage Subsystem to ensure that it satisfies the above requirements. Confirm with the GPFS architects that the GPFS solution is properly sized to account for HSM activity. The review should include the following for GPFS:
    - Verify the total number of files (both scratch files and HSM space managed files) being stored in GPFS.
    - Verify the types of GPFS files. Obtaining a file size distribution histogram is often helpful.
    - Verify the data pathways (the network) of the GPFS cluster. GPFS clusters are often used with HPC activity. It is important to understand the HPC bandwidth requirements for scratch files, and to understand how the raw data and eventually the product data will be moved and stored.
    - Verify the GPFS file system block size.

- Verify how the GPFS storage is configured into the cluster. Capture the amount of data and metadata resources.
    - Verify the number of inodes allowed is enough for the number of files expected.
    - Verify the amount of inode space that is needed.
- Verify how often to backup the GPFS file system and identify any backup and restore requirements.

- Review the architecture of the entire HPSS system to satisfy the HSM portion and the Backup/Restore portion of the requirements. The review should include the following for HPSS:
    - Verify the amount of disk cache needed for the aggregate index files if aggregation is to be used.
    - Verify the location of the HPSS Mover nodes vs. IOM nodes. IOM nodes can be either GPFS NSD nodes or GPFS client nodes. HPSS Mover nodes may be inside the GPFS cluster or outside of the GPFS cluster.
    - Verify COS selection.
        - ***Non-Aggregate Files***: Consider a disk-to-tape COS with purge, or a tape only COS.
        - ***Aggregate Data Files***: Consider a disk-to-tape COS with purge, or a tape only COS.
        - ***Aggregate Index Files***: Consider a disk-to-tape COS, no purge.
        - ***Backup Files***: Consider a disk-to-tape COS with purge.
    - Verify File Family selection. Consider using file families to guarantee associated files are placed on the same set of tapes.
    - Verify the HPSS subsystems to be configured and how resources will be allocated among them.
    - Verify the amount of network capacity that is required to provide data to the tape drives at rates sufficient to keep them running at their rated speed.
- Confirm the list of prerequisite software that needs to be obtained in order to satisfy the target GHI architecture. Refer to *Section 2.2.5 - Prerequisite Software Overview* for more information on the HPSS prerequisite software requirements.
- Confirm the space requirements needed for temporary policy output files.

- Confirm the resources required to handle the work loads to be imposed on the GHI nodes. Refer to *Section 2.3 - Considerations* for more discussions on the system resource requirements.

2.1.3. Software Planning

Refer to *Section 2.2.5 - Prerequisite Software Overview* for more information on the required software that is needed to run GHI.

2.1.3.1. Compatibility with other HSM applications

GHI can not run on the same GPFS cluster as any other HSM application. This includes TSM. If another HSM managed file system is required it must be run on a separate cluster and remotely mounted on the GHI managed cluster. This incompatibility is due to GHI not always receiving DMAPI mount events when competing HSM applications are running.

2.1.4. Operations Planning

The following section outlines key aspects of the operations planning process which is reviewed by the HPSS system engineering and deployment team with the customer. This information is first obtained during the proposal or opportunity assessment phase. It is then reviewed during the implementation and pre-production phases since requirements and/or assumptions will often need to be updated from the initial understanding of the system. The customer is responsible for providing operations planning information to the HPSS systems engineering and deployment team and for communicating changes and updates. The HPSS systems engineering and deployment team will use the information to review the configuration with the customer to provide an assessment of the system's readiness for operational use.

The following planning steps must be carefully considered and reviewed by the customer for the GHI operational phase:

- Verify the user's operations scenarios, including, but not limited to:
    - ***GPFS filesystems***: Will the customer have multiple GPFS file systems? Are the file systems to be segregated? Are individual directories to be segregated into their own file families?
    - ***Data ingest***: What amount of data are the users going to write into the GPFS file system?
        - Which of the users' files will be archived?
        - Will there be a scratch area which does not need to be archived?
        - Will GPFS files be pinned, where the GPFS file data is required to stay in the GPFS file system?
        - When a file is ingested where does it go in HPSS? Will files

go to different COS's based on size? On GID? On path?

- ***Data archive rates***: How much of the user data will make its way into the archive (HPSS) via GHI?
- ***Data retrieval rates***: How much of the user data will be read from the archive via GHI?
- ***Aggregation or not***: Will GHI aggregation be used? What are the GPFS file sizes to be included in the GHI aggregate? How many files per aggregate?
- ***Filesystem backups***: How often will the GPFS file system be backed up? How many backups will the customer require to be online? How many backups will the customer require to be offline? How long does the customer keep backups?

- Confirm which NSD quorum nodes will be used for potential Session nodes.
- Confirm if the GHI IOMs will be coresident with the GPFS NSD nodes or be resident on GPFS client (Non-NSD) nodes.
- Determine the number of IOMs required to achieve the data transfer rates.

2.1.5. GHI Deployment Planning

The following planning steps must be carefully considered for the GHI deployment phase:

- Once the items in *Section 2.1.2 - GHI Configuration Planning* have been analyzed, it is important to confirm that the GPFS and HPSS architecture and resources allocated to GHI are adequate to meet expected data transfer rates and storage capacity. The HPSS team members are not GPFS solution architects, but will participate in the analysis of the overall GPFS + HPSS coupled solution, along with the customer and the GPFS architects.
- Before the GHI deployment begins, the prerequisite foundations must be installed and operationally verified by the customer and the HPSS team
- The HPSS deployment team will be available to work with the customer and GPFS deployment team to determine a customer test plan that demonstrates that the GPFS file system can manage the aggregate load that the customer is expecting. It is important to understand and plan for the transfer and transaction rates of the system.
- The GPFS and HPSS deployment team will work with the customer to help them determine if the GPFS + HPSS coupled solution is ready for production use.

2.2. Requirements and Intended Uses for GHI

This section provides some guidance for the administrator to identify the site's requirements and expectations of GHI. Issues such as the amount of storage needed, access speed and data transfer speed, typical usage, security, expected growth, data backup, and conversion from an old system must be factored into the planning of a new GHI system.

2.2.1. Storage System Capacity

The amount of GHI user data storage space the administrator must plan for and take into account includes the following considerations:

- The amount of user data storage space required to support a specified number of files to remain in disk cache.
- The amount of disk cache required for HPSS should include the space required for Aggregate Index files as well as for backup files.

2.2.2. Required Throughputs

Determine the required or expected throughput for the various types of data transfers that users will perform. Some users want quick access to small amounts of data. Other users have huge amounts of data they want to transfer quickly, but are willing to wait for tape mounts, etc. In all cases, plan for peak loads that can occur during certain time periods. These findings must be used to determine the type of storage devices and network to be used with HPSS to provide the needed throughput.

Site planners should consider file system functionality which can cause scalability issues.

Policy scan overhead and time to complete increases as the GPFS file system grows. Sites should consider including multiple rule definitions in a single policy run rather than fewer rules, and more policy runs. Also, if sites are generating both aggregate and non-aggregate rules, the rules for non-aggregates should be placed before the aggregate rules. Aggregate lists must be constructed, wherease non-aggregate requests are processed immediately.

POSIX command line interfaces such as "*file **"*, can cause staging of GPFS file data from GPFS. Consider keeping the first block of each file's data on GPFS disk. This is a candidate for a future GHI release to configure the file system such that the first block of the file contents can be kept on the GPFS file system. Currently, releasing the GPFS resources frees up all data blocks from the GPFS file contents.

The time to create and destroy snapshots is relative to the size of the GPFS file system, as well as file modifications to the GPFS file data during the snapshot operations.

2.2.3. Load Characterization

Understand the kind of load users are putting on an existing file storage system provides input that can be used to configure and schedule the GPFS system. What is the distribution of file sizes? How many files and what amount of file contents are transferred in each category? How does the load vary with time (e.g., over a day, week, month)? Are any of the data transfer paths saturated?

Having this file system load information helps to properly size both GPFS and HPSS so that they can meet the peak demands. Also based on this information, maintenance activities such as purge and backups can be scheduled during times when the system is less busy.

2.2.4. Security

Authentication and authorization between GHI servers is done through use of either UNIX or Kerberos security tools for authentication and UNIX for authorization services.

GHI should be configured to use the same authentication/authorization that HPSS is configured with. It is basically a client to HPSS. GHI uses the ***hpssdmg*** principal for all authentication.

If aggregation is used, the same ***hpssdmg*** principal applies.

2.2.5. Prerequisite Software Overview

This section defines the prerequisite requirements for GHI. Some products must be obtained separately from GHI and installed prior to the GHI installation and configuration.

2.2.5.1. DB2

GHI uses the DB2 Universal Database Enterprise Server Edition by IBM Corporation to manage all GHI metadata for backup/restores. DB2 software and a limited-use license is included in the HPSS distribution. Refer to ***db2_install*** to install the DB2 client.

2.2.5.2. GPFS

Please refer to the *GPFS Advanced Administration Guide*, when installing and configuring GPFS. There are many online resources that are available. IBM Redbooks may also be considered as a valuable resource.

2.2.5.2.1 Required GPFS configuration

GHI requires that the GPFS file systems have DMAPI, version 3.4 metadata, and fastea's (fast extended attributes) enabled. When doing a mmlsfs the following fields must be set:

-V **12.03 (3.4.0.0)**

--fastea **yes**

-z **on**

2.2.5.2.2 Required GPFS environment

Changes in GPFS PTF 8 may require linux users to update their /etc/profile to include the following:

export MM_SORT_OPTS='%3'

If you experience problems with sorts when running policies that select files add the MM_SORT_OPTS.

2.2.5.3. HPSS Client API

Please refer to the *Section 5.3.3.1 - Install Client HPSS Source Code* in the *HPSS Installation Guide* to install/build the Client API on the GHI Session nodes.

2.2.5.4. Kerberos

GHI uses Massachusetts Institute of Technology (MIT) Kerberos to implement Kerberos authentication. MIT Kerberos is a network authentication protocol designed to provide authentication for client/server applications by using secret-key cryptography. A free implementation of this protocol can be downloaded from the MIT's website (http://web.mit.edu/kerberos/). Refer to *Section 5.2.2 -*

Install MIT Kerberos in the *HPSS Installation Guide* for more information.

For Linux, Kerberos is installed as part of the operating system.

If UNIX authentication will be used, this product is not required.

2.2.5.5. GHI-HTAR

GHI uses GHI-HTAR to implement file aggregation when migrating files into HPSS. GHI-HTAR bundles small files on a platform into large files in storage. Temporary storage is used to build the index files associated with an aggregate. The temporary files are removed once the file is written to HPSS. IBM will supply a version of GHI-HTAR that is compatible with the version of GHI that is being installed.

GHI aggregated data relies upon GHI-HTAR index files to always be available to the system. If an index is inaccessible (missing, damanged, or delayed for an extended period of time), retrieving user file contents are impacted, including to the point of failure by the end-user to access their files. Storage of the index files must be constructed to protect this data from media failures and/or catastrophic damage. Index files should be considered equilavent to HPSS metadata and require the use of mirrored disk copies as well as multiple tape copies to properly protect the data. This includes using remote or offsite backups of this vital information as one would do for HPSS DB2 metadata.

If aggregation will not be used, this product is not required.

2.2.6. Prerequisite Summary Based on Node Type

The Session nodes and IOM nodes all require the following prerequisite software:

- GPFS 3.3 PTF 7
- HPSS 7.3.3 patch1 (Mover/Client Interface)

This section provides a summary list of prerequisite software required for HPSS. It also lists the software versions which have been verified with HPSS.

2.2.6.1. HPSS Core Server

The HPSS Core Server node requires the following to support aggregation:

- HSI 3.5.7
- Openssl version 0.9.8g or later.

2.2.6.2. GHI Session Nodes

GHI Session nodes contain the following processes: Process Manager, Log

Daemon, Mount Daemon, and multiple Scheduler and Event Daemons. The IOM can also be configured on this node, and can either be active, or remain dormant if the node is the active Session node.

2.2.6.2.1 AIX Requirements

Each AIX Session node must have the following installed:

- IBM RS/6000 (eServer pSeries) with a minimum of 8 cores and 16 GB RAM.
- AIX 6.1 (6100-02_AIX_ML which consists of Technology Level 2 and Service Pack 2).
- DB2 UDB V9.5 Enterprise Server Edition (ESE) for AIX.
- MIT Kerberos 1.6.3 (if planning to use Kerberos authentication).
- C compiler for AIX, version 10.1.0.0 (if planning to recompile the HPSS Client API and GHI code on this node).

2.2.6.2.2 Linux Requirements

Each Linux Session node must have the following installed:

- Linux machine (eServer xSeries) with a minimum of 8 cores and 16 GB RAM.
- Red Hat Enterprise Linux EL release 5.7
- DB2 UDB V9.5 Enterprise Server Edition (ESE) for LINUX.
- MIT Kerberos 1.6.3 (if planning to use Kerberos authentication).
- C compiler for Linux: gcc-4.1.2 (if planning to recompile the HPSS Client API and GHI code on this node).
- Openssl version 0.9.8g or later.

2.2.6.3. **GHI I/O Manager Nodes**

An I/O Manager node consists of the following processes: the I/O Manager administrative process and a combination of either GHI-HTAR and/or the I/O Agent processes to perform data transfers.

2.2.6.3.1 AIX Requirements

Each AIX IOM node must have the following prerequisites:

- IBM RS/6000 (eServer pSeries) with a minimum of 4 cores and 8 GB RAM.
- AIX 6.1 (6100-02_AIX_ML which consists of Technology Level 3 and

Service Pack 2).

- MIT Kerberos 1.6.3 (if planning to use Kerberos authentication).
- C compiler for AIX, version 10.1.0.0 (if planning to recompile the HPSS Client API and GHI code on this node).
- HSI 3.5.4
- Openssl version 0.9.8g or later.

2.2.6.3.2 Linux Requirements

Each Linux IOM node must have the following prerequisites:

- Linux machine (eServer xSeries) with a minimum of 4 cores and 8 GB RAM.
- Red Hat Enterprise Linux EL release 5.7
- MIT Kerberos 1.6.3 (if planning to use Kerberos authentication).
- C compiler for Linux: gcc-4.1.2 (if planning to recompile the HPSS Client API and GHI code on this node.
- HSI 3.5.4
- Openssl version 0.9.8g or later.

2.2.6.4. GHI I/O Agent Nodes

An I/O Agent node is typically the same node as the IOM, since the IOM is a fairly lightweight process that is primarily responsible for spawning off the IOAs.

2.2.6.4.1 AIX Requirements

Each AIX IOA node must have the following prerequisites:

- IBM RS/6000 (eServer pSeries) with a minimum of 4 cores and 8 GB RAM.
- AIX 6.1 (6100-02_AIX_ML which consists of Technology Level 3 and Service Pack 2).
- MIT Kerberos 1.6.3 (if planning to use Kerberos authentication).
- C compiler for AIX, version 10.1.0.0 (if planning to recompile the HPSS Client API and GHI code on this node).

2.2.6.4.2 Linux Requirements

Each Linux IOA node must have the following prerequisites:

- Linux machine (eServer xSeries) with a minimum of 4 cores 8 GB RAM.

- Red Hat Enterprise Linux EL release 5.7
- MIT Kerberos 1.6.3 (if planning to use Kerberos authentication).
- C compiler for Linux: gcc-4.1.2 (if planning to recompile the HPSS Client API and GHI code on this node.

2.3. Considerations

This section describes the infrastructure needed to operate GHI and includes considerations about infrastructure installation and operation that may impact GHI.

2.3.1. Network Considerations

Because of its distributed nature and high-performance requirements, a GHI system is highly dependent on the networks to provide connectivity among the GHI servers, HPSS, and GPFS.

For control communications (i.e., all communications except the actual transfer of data) among the GHI servers, GHI requires TCP/IP services. Since control requests and replies are relatively small in size, a low-latency network usually is well suited to handle the control path.

The data path is logically separate from the control path and may also be physically separate, although this is not required. For the data path, GHI supports the same TCP/IP networks as those supported for the control path. For supporting large data transfers, the latency of the network may not impact overall data throughput.

2.3.2. General ILM Policy Considerations

The GPFS ILM migration policy provides the capability for GHI to copy (migrate) files from GPFS to HPSS. The GPFS ILM migration policy identifies files that are new or recently modified that need to be copied to the HPSS repository. GHI processes the lists of files that have been identified and simply copies them from GPFS to HPSS. Larger files are copied straight to HPSS, while the smaller files may be aggregated into larger HPSS objects.

The GPFS ILM purge policy provides the capability for GHI to space manage the GPFS file system. GPFS files are continuously copied to HPSS. When the GPFS file system reaches a pre-defined space threshhold, the GPFS ILM purge policy is executed to identify file candidates whose data can be removed from the file sytem. The GPFS ILM purge policy will identify the older, larger files as candidates. GHI will “punch a hole” in the files that have been identified to free GPFS disk resources. The inode and metadata for these files are left in the GPFS file system, so from the user’s point of view, nothing about these files has changed.

The GPFS ILM recall policy provides the capability for GHI to stage files in bulk from HPSS back to GPFS. The GPFS administrator will need to author an ILM policy rule to stage a given set of files back from HPSS. These requests will be optimized so that files located on the same tape will be recalled together to

minimize tape mounts.

The site administrator will need to monitor the GPFS ILM policies to ensure that they are completed without errors. Errors in the migration process may lead to undesired file system behavior – file system may fill up, backups may be incomplete, etc. The GHI version 2 software release plan includes a summation of the number of files that were transferred successfully, as well as the number of files that were unsuccessful.

The GPFS ILM backup policy provides the capability for GHI to make a point-in-time backup of the GPFS file system. The lists of files are processed by GHI. Some lists are copied directly to HPSS, while other lists are used to gather file attribute information to generate files containing the metadata. Those files will also be copied to HPSS. The result of the GHI processing is a point-in-time backup of the GPFS file system.

Administrators should experiment to determine the parameter settings that will fit the needs of their site. If a site has a large amount of disk file write activity, the administrator may want to have more free space and more frequent purge runs. However, if a site has a large amount of file read activity, the administrator may want to have smaller disk free space and less frequent purge runs, and allow files to stay on disk for a longer time.

The policy generates multiple ****.exc*** and a ****.ok*** files. The exception files (*.exc) contains all the files that failed during the policy run. The okay (*.ok) files contain all the files that were successfully transferred. The exception files are displayed as a result of the policy run. General options used by the policy files are:

-d: The “-d” option keeps both the exception files and the okay files from being deleted when the policy run is complete.

-D: This is the “dirty flag” option keeps not only the exception and okay files, it also keeps the generated policy files from being deleted.

If files are retained following the policy run, it is up to the administrator to clean those files out.

2.3.2.1. HSM Policy Considerations

Figure 8 - HSM Policy Output shows an example of the output files for a policy run. The reference to “EXT” reflects the operation being performed: “migrate”, “recall” or “purge”.

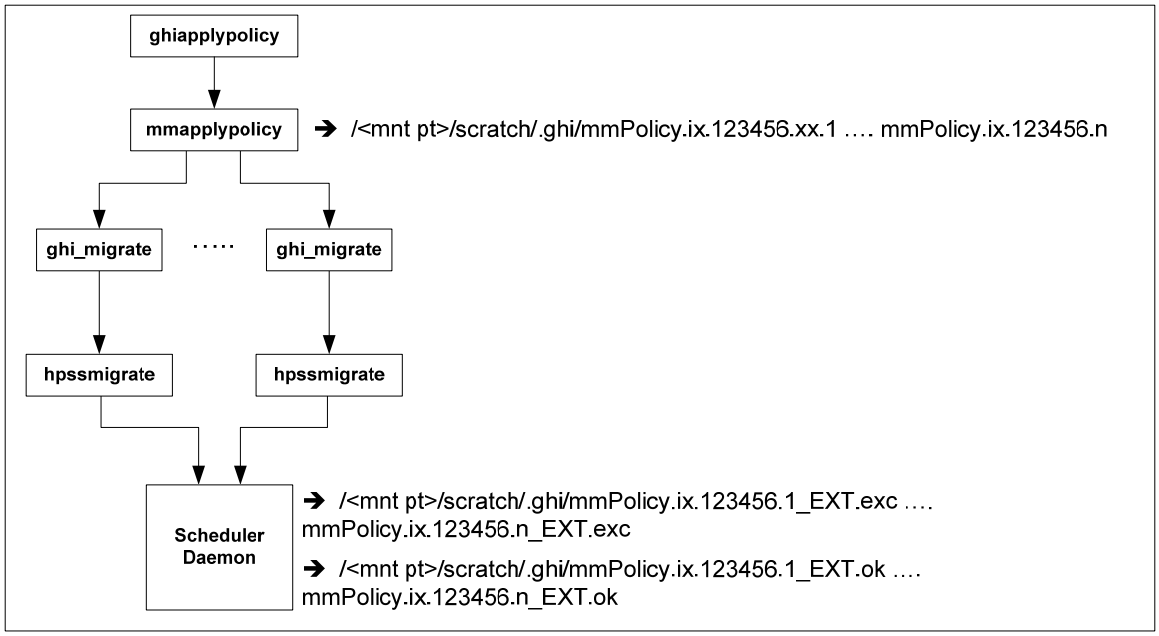

Figure 8 - HSM Policy Output

2.3.2.1.1 migrate.policy

The directory ***/<file system>/scratch/.ghi*** is where GHI places the temporary files for migrations, so that directory tree should be omitted from being migrated to HPSS. For optimization, an additional rule should be defined to exclude files that are already co-managed.

The migration uses a "Max Aggregate Files" value found in the ***ghi.conf*** file. This value is used to generate the policy output files in bulk sizes based on this amount. So, for example, if the bulk size is 100, and 500 files are output candidates from the policy run, then 5 output files will be generated from the policy engine. Migrate policy outputs can also use the GPFS ILM SIZE command in the policy.

2.3.2.1.2 recall.policy

The directory ***/<file system>/scratch/.ghi*** is where GHI places the temporary files, so that directory tree should be omitted from being migrated to HPSS. For optimization, an additional rule should be defined to exclude files that are not co-managed.

The recall policy does not use a bulk size. The policy generates one list. That list is parsed into aggregates and non-aggregates. The non-aggregates are sent directly to the Scheduler. The aggregates are grouped in bulk sizes based on which aggregates they are located in. This allows GHI to make a single request to GHI-HTAR to retrieve all files requested to be retrieved from that aggregate.

2.3.2.1.3 delete.policy (missing)

The directory */<**file system**>/scratch/.ghi* is where GHI places the temporary files, so that directory tree should be omitted from being deleted.

This policy permenently deletes the selected files from GPFS. The selected files are bulked according to the GHI aggregate bulk size and passed to the hpssdelete process. The hpssdelete process unlinks each selected file.

2.3.2.1.4 threshold.policy (missing)

The directory */<**file system**>/scratch/.ghi* is where GHI places the temporary files, so that directory tree should be omitted from being purged. Files that are not comanaged can not be purged.

This policy purges the selected file's data from GPFS. The selected files are bulked according to the GHI aggregate bulk size and passed to the ghi_sd for processing.

2.3.2.2. Backup Policy Considerations

Figure 9 - Backup Policy Output shows an example of the output files for a policy run. The figure only shows the output from the list portion of the policy run. The migration output will be similar to the output shown in *Figure 8 - HSM Policy Output*. There will be two sets of output files generated from the IOMs: "EXT" will be both "backup_ns" and "backup_attr" for both namespace and attribute file generation and backup into HPSS. The "backup" files generated by the SD will contain general backup failures for backup of input files and GPFS quota files into HPSS.

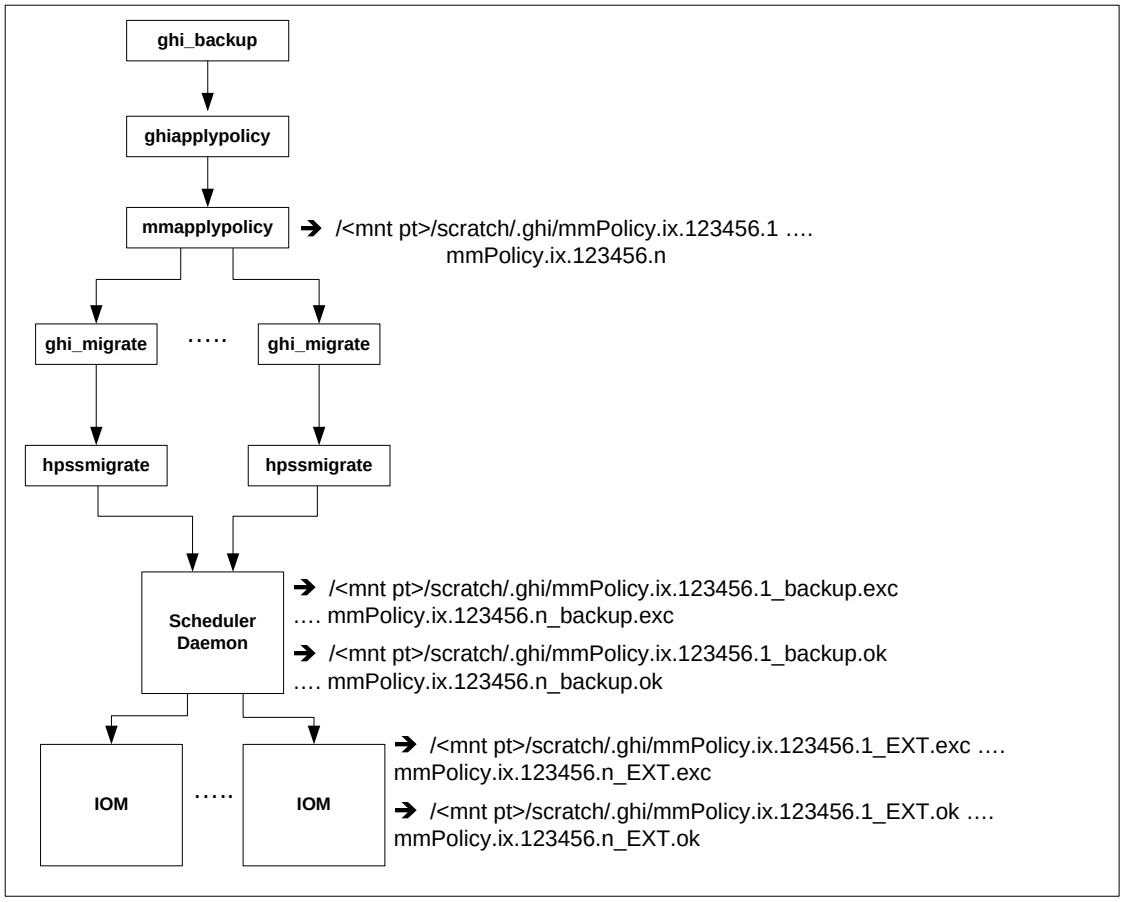

Figure 9 - Backup Policy Output

There are two parts to the backup: the migration of data, and the backup of namespace and attribute files. The migration of data uses the migration policy, and works as described in *Section 2.3.2.1.1 - Migration Policy*. When backing up the namespace and attribute files, the ”Backup Bulk Count” value, which is found in the ***ghi.conf*** file, is used to generate the policy output files.

The workload to gather the namespace and attribute files is distributed across the IOMs based on the “Backup Bulk Count”. The following files are generated during backup:

- ***Namespace***: Contains information for generating the namespace during a restore. Each entry is currently 100-200 bytes in length, based on full path. The format is:

  <inode> <igen> <epoch> -X <size>,<UID>,<GID>,MMM -- fullpath

  Where MMM = Miscelaneous attributes

- ***Index***: Contains metadata for a GPFS file. Each entry is currently 144 bytes in length. The file is a table format containing one entry for each

file in the input file.

- ***Index_sl***: Contains metadata for a GPFS symbolic link. Each entry is currently 72 bytes in length. The file is a table format containing one entry for each file in the input file.
- ***Data***: Contains the full path for a GPFS file. Each entry is variable in length, based on the full path. The file format is a stream of full path names.
- ***Input***: Contains the information that was used to generate the index and data files. Each entry is 100-200 bytes in length, based on full path. The format is:

  ```
  <inode> <igen> <epoch> -s <size> -- full path
  ```

  The file contains up to the bulk size number of entries.

2.3.2.3. Scratch Area

GHI uses a temporary directory in the GPFS file system to store temporary files. The following file types are stored here:

- ***Policy output files***: The output files generated from the GPFS policy.
- ***GHI-HTAR index files*:** GHI-HTAR stores the temporary index files here before writing the file to HPSS.
- ***Session node configuration file*:** This configuration file, ***sn.conf***, is used by each of the nodes to determine which node is the active session node. This file is used to locate the Scheduler Daemon.
- ***Backup configuration file*:** The backup configuration, ***backup.conf***, is generated by the ghi_backup script and is used by the I/O Managers to detemine the location in HPSS for storing the files generated by the backup process.

2.3.2.4. Thresholds

A threshold policy will be defined to manage the available space. It will define the high (NO_SPACE) and low (LOW_SPACE) water marks. It will be associated with one or more file systems. Each file system has its own threshold policy in the /var/hpss/ghi/policy/<filesystem> directory.

When the file system reaches the high-water mark, the policy will be invoked to free up GPFS disk resources. We refer to this as “purging data”. See *Section 2.3.2.4.1- Purging Data* for more information.

Candidates can be considered based upon file age, file size, etc. The GPFS ILM policy allows the candidates to be weighted, so you can specify which files to be considered first. The list of files generated will be enough to free up resources until the low water mark is reached.

To configure the system to react to these events, add the threshold policy and update the GPFS configuration to enable threshold processing. When activated, and one of the events is triggered, GPFS will start a ghiapplypolicy on one of the nodes in the cluster, and the threshold policy will be used to determine what files to "punch holes" into.

2.3.2.4.1 Purging Data

Freeing GPFS disk resources is referred to as "punching a hole". Files to be purged must have already been migrated into HPSS.

Policy rules are used to weigh or exclude files that will not be considered as purge candidates. You can specify a where clause to select files like:

```
WHERE FILE_SIZE >= 262144      # where 262144 = block size of the file system
```

The configuration file allows a site to define how many bytes of data will remain in the GPFS file system after the file is purged.

2.4. GHI Sizing Considerations

There are four types of storage space that must be planned for:

- GHI Infrastructure Storage Space.
- GPFS File Systems.
- GHI Metadata Space.
- System Memory and Disk Space.

2.4.1. GHI File Systems

The following sections describe the various file systems used by GHI. When GHI is deployed into the GPFS cluster, these directories will be needed. Each GPFS node selected for GHI use will require a trivial amount of file system disk space – approximately 15 MB.

2.4.1.1. /opt/hpss

This directory holds HPSS/GHI binaries, source code, include files, libraries, and utilities. The GHI software is installed in the ***/opt/hpss/src/ghi*** directory.

2.4.1.2. /var/hpss

The ***/var/hpss*** directory tree is the default location of a number of HPSS and GHI configuration files and other files needed by the servers. It is required that this file system be at least 1 GB in size.

Within the /***var/hpss*** file system the following sub-directories exist:

- The ***/var/hpss/cred*** is the default directory where some additional UNIX configuration files are placed. These files are typically very small.
- The ***/var/hpss/etc*** is the default directory where some additional UNIX configuration files are placed. These files are typically very small.
- The ***/var/hpss/ghi*** is the default directory where several GHI files are maintained. There are five sub-directories required: config, etc, log, policy, and temp.
- The ***/var/hpss/ndapi*** is the default directory where the log and performance files are found for HSIGWD. This directory resides on the HPSS Core Server node.
- The ***/var/hpss/ghi/tmp*** is the default directory where the Process Manager creates a lock file for each of the GHI process it brought up in the node. GHI may also write diagnostic log files and performance files here as well. The lock files are very small, but the log and performance files may be several tens of kilobytes, or larger.
- The ***/var/hpss/ghi/log*** is the default directory where the Log Daemon creates two central log files. The size of these log files is specified in the Log Daemon specific configuration. By default, these files are 10 MBs each. Each of the GHI servers write log files to this directory as well. They write two rotating log files that are based on the amount of logging that is turned on.

2.4.1.3. /var/hpss/adm/core

/var/hpss/adm/core is the default directory where GHI servers put "core" files if ghi processes terminate abnormally. Core files may be large, so it is required that there be at least 2 GB reserved for this purpose on the Session node and at least 1 GB on IOM nodes.

It is up to the administrator to remove unneeded core files to prevent the /var/hpss/file system from filling up.

2.4.1.4. /var/hpss/hpssdb

The ***/var/hpss/hpssdb*** directory is the location where the database instance is stored, which is used to access the remote DB2 server on the HPSS Core Server. The minimum file system size required is 20-30 MB for the runtime DB2 client.

2.4.1.5. /var/hpss/ndapi

The ***/var/hpss/ndapi*** directory is the default directory containing the audit trail and transfer logs created by GHI-HTAR. These files grow without bound, and must be manually managed by the site. Normally a ***cron*** job is run periodically to copy filled logs into HPSS and then remove or null them out.

2.4.2. GHI Metadata Space

During the GHI planning phase, it is important to properly assess how much disk space will be required by DB2 to store GHI metadata. The first step in this process is to understand the metadata tables managed by DB2. The database table used by GHI is for storing the information in a backup and for storing files for garbage collection.

For the backup table, there is one row generated each time a GHI backup is performed. The row is 48 bytes in length. The table resides on the HPSS Core Server node.

For the garbage collection table, there is one row generated for each deleted file contained in a backup. The row is 64 bytes in length. The table resides on the HPSS Core Server node.

Need to discuss UDA usage here.

2.4.3. System Memory and Disk Space

The following sections discuss requirements for disk space, system memory, and paging space.

2.4.3.1. System Memory and Paging Space Requirements

The memory and disk space requirements for the nodes where the GHI processes will execute depends on the configuration of the servers, the nodes that each server will run on, and the amount of concurrent access they are configured to handle.

At least 8 GB of memory is required for the GPFS cluster nodes running GHI processes. When GPFS is running an ILM policy scan, it consumes a considerable amount of memory. Paging space should be sized with the same amount of space as the memory.

2.5. GHI Interface Considerations

This section describes the user interfaces to GHI and the various considerations that may impact the use and operation of GHI.

2.5.1. GHI Server Considerations

Servers are the internal components of GHI that provide the system's functionality. They must be configured correctly to ensure that GHI operates properly. This section outlines key considerations that should be kept in mind when planning the server configuration for a GHI system.

2.5.1.1. Session Node

The Process Manager, Mount Daemon and Log Daemon get started automatically when GPFS is started. GHI provides a utility, ***hpssEventNotify***, that GPFS calls when it comes online. That utility, in turn, starts the Process Manager. The Process Manager then starts the Mount Daemon and Log Daemon.

When GPFS is stopped, or GPFS loses quorum on the Session node, the ***hpssEventNotify*** utility is called to stop those processes, and in the event of a failover, starts then up on another node.

When a file system is mounted, the Mount Daemon receives the MOUNT event, and notifies the PM to start an Event Daemon and Scheduler Daemon.

When a file system is unmounted, those processes are terminated.

2.5.1.2. I/O Manager / I/O Agent

Both I/O Manager (IOM) and I/O Agent (IOA) processes usually reside on the same node. The IOM is a lightweight process, so it doesn't typically warrant its own node.

The IOMs are typically started via the ***inittab*** processs. They remain in standby mode until they detect that the GPFS file system is mounted. The IOM will spawn a number of IOAs or GHI-HTAR processes to do data transfers. This number is based on the "IO Manager. Thread Pool Size" value in the ***ghi.conf*** file.

The IOAs are started via ***inetd***, and started each time a non-aggregate data transfer is requested.

2.5.1.2.1 Performance

The configuration of the I/O Agents and attached devices can have a large impact on the performance of GHI because of constraints imposed by a number of factors e.g., device channel bandwidth, network bandwidth, processor power.

The IOM/IOA configuration is largely dictated by whether a site runs the processes on an NSD node, or a GPFS Client node. In the case where the the GPFS cluster is using a SAN configuration, and the NSDs can see all the GPFS data blocks, placing the IOA on the NSD nodes, eliminating data transfers to gather the data blocks to send to the HPSS Mover.

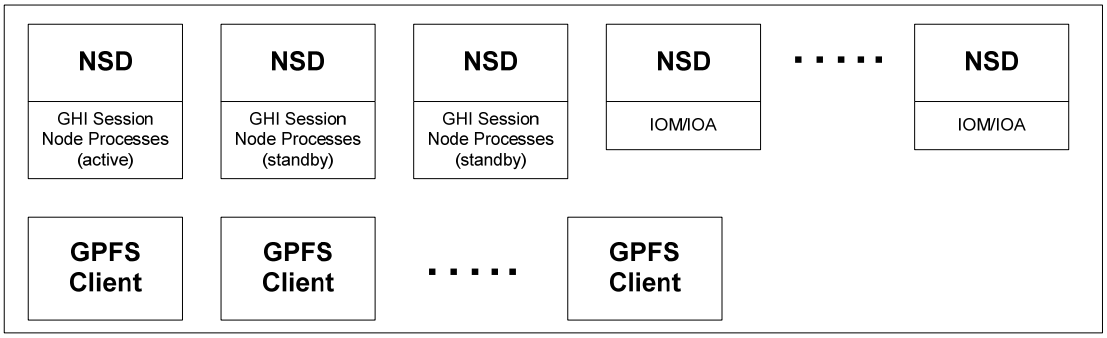

Figure 10 - IOM / IOA Layout – NSD Node Configuration

Figure 10 - *IOM / IOA Layout – NSD Node* Configuration shows an example configuration placing the IOMs on the NSD nodes. An IOM can run on any of the Session nodes, which is not depicted in Figure 10. There is a configuration option such that the IOM is active on the standby nodes, and if the standby Session node becomes the active node, the IOM goes dormant and does not process any data transfers.

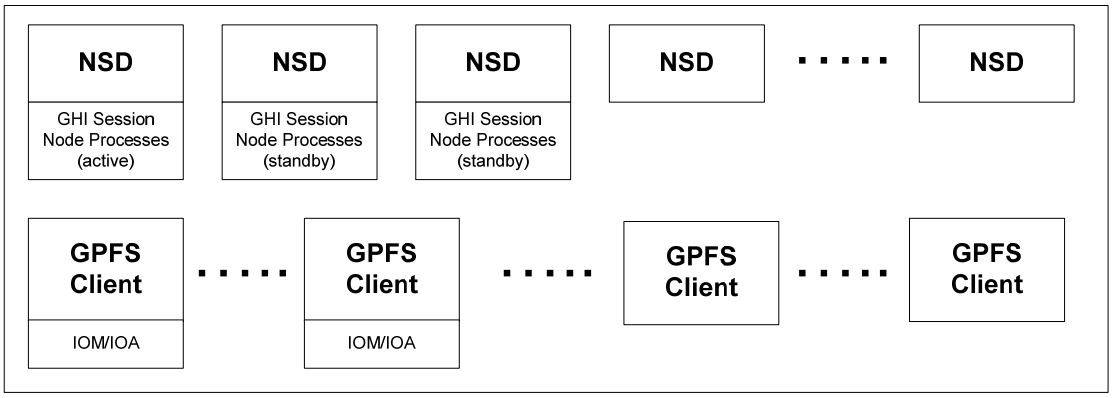

Figure 11 - IOM / IOA Layout – Client Node Configuration

Figure 11 - *IOM / IOA Layout – Client Node Configuration* shows an example of placing the IOMs to run on the client nodes.

For sites configuring their GPFS clusters to use locally attached devices, there will be network traffic generated for the majorify of the data block transfers, since the IOA will have to gather all the data blocks to send to the HPSS Mover.

A balanced configuration, tuned to optimize GPFS, GHI, and HPSS performance involves the coordinated input and consultation of system engineers with expertise from all three of these major components. Because changes in one part of the system can have a significant impact to other components, it is required that GPFS, GHI, and HPSS support personnel are consulted before updating the configuration or a site considers changes to the

HW/architecture of the overall system.

To detemine the sizing for the number of IOMs required to achieve the data throughput required, refere to *Figure 12 IOM Capacity*.

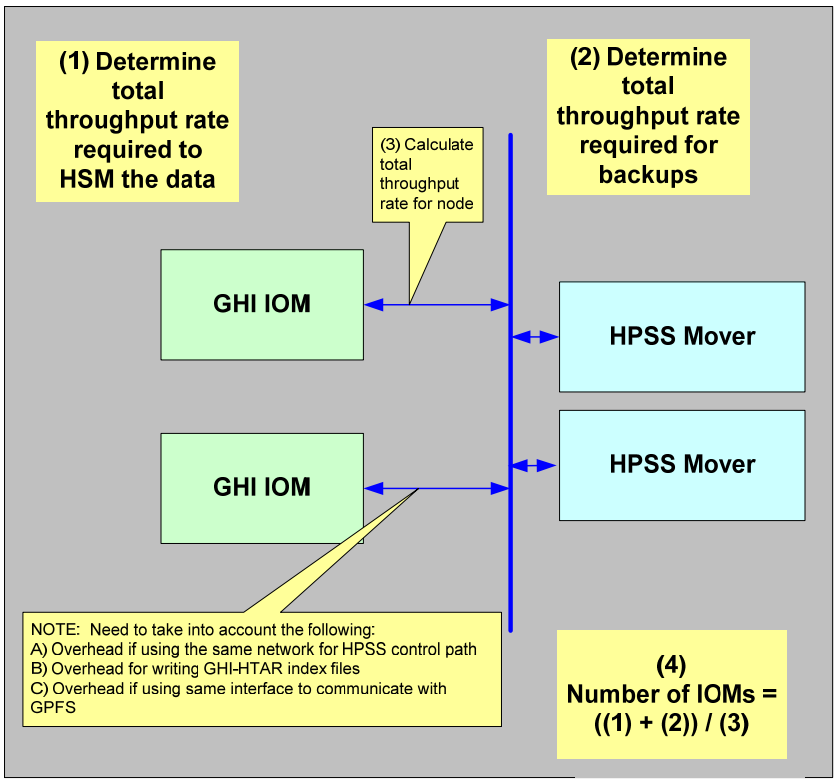

Figure 12 IOM Capacity

2.5.2. GHI-HTAR

Aggregation requires that GHI-HTAR runs on the nodes where the IOMs reside. The GHI-HTAR process is started via a script, ***htar.ksh***, that resides in the ***/opt/hpss/bin*** directory. The script is used to determine authentication of GHI-HTAR, as well as locate the GHI-HTAR executable.

A GHI-HTAR process is started when an aggregate is requested to be processed by an IOM. GHI-HTAR has GHI-specific interfaces to provide the ability to use DMAPI to perform GPFS data transfers.

2.5.3. GHI Policy Engine

The policy engine requires temporary storage space for pre-processing, sorting, and generating the output files to be used by GHI for performing the operations. The GPFS file system is used to store all the generated files. The location is /<Mount Point>/scratch/.ghi

2.5.4. Logging Service

Logging Services are comprised of the GHI Log Daemon, which is spawned from the Process Manager on the Session node.

Log messages from all GHI servers will be written by the Log Daemon to a common log file. There is a single Log Daemon process per GHI cluster system. The Log Daemon toggles between two log files. When a log file fills up, the file is archived into HPSS.

2.6. HPSS Storage Characteristics for GHI

This section defines key concepts of HPSS storage and the impact the HPSS storage on GHI configuration and operation. These concepts, in addition to the policies described above, play a significant role with the usability of GHI.

Before a GHI system can be used, the administrator must create a description of how the system is to be viewed by the HPSS software. This process consists of learning about the intended and desired usage of the system from the GPFS users and then using this information to determine GHI/HPSS hardware requirements and the configuration of the hardware to provide the desired performance. The process of organizing the available hardware into a desired configuration results in the creation of a number of HPSS metadata objects.

2.6.1. Storage Classes

A Storage Class is used by HPSS to define the basic characteristics of storage media. These characteristics include the media type (the make and model), the media block size (the length of each basic block of data on the media), the transfer rate, and the size of media volumes. These are the physical characteristics of the media. Individual media volumes described in a Storage Class are called Physical Volumes (PVs) in HPSS.

2.6.2. Classes of Service

Class of Service (COS) is an abstraction of storage system characteristics that allows HPSS users to select a particular type of service based on performance, space, and functionality requirements. Each COS describes a desired service in terms of characteristics such as minimum and maximum file size, transfer rate, access frequency, latency, and valid read or write operations. A file resides in a particular COS which is selected when the file is created. Underlying a COS is a storage hierarchy that describes how data for HPSS user files in that class are to be stored in the HPSS system. A COS can be associated with a fileset such that all files created in the fileset will use the same COS.

Each GHI file, which appears to HPSS as a user file, belongs to a single Class of Service (COS) which is selected when the file is created. There are three classes of GHI files written to HPSS. They are as follows:

- ***Data files (aggregate and non-aggregate files).*** By default, these files use the Class of Service Maximum File Size Hints information passed to HPSS when the file is created. The policy can be defined to override the default COS by specifying a "OPTS –c <COS>," or "OPTS –c <COS:auto> in the policy for non-aggregates and aggregates respectively.

  NOTE: Each GHI file belongs to a single Class of Service (COS) which is selected when the file is created. The ***Force Selection*** flag can be set in the COS definition on the HPSS GUI to prevent automatic selection. If the flag is set, that COS will not be selected for storing the GHI data files.

- ***Aggregate index files.*** These files are written to HPSS using the "Aggregate Index COS" in the ***ghi.conf*** configuration file as a default. All aggregate index files for a file system will go to the same COS. The site can override the default COS by specifying "OPTS –c COS for data file:COS for index file"

- ***Backup files.*** These files are written to HPSS using the "Backup COS" in the ***ghi.conf*** configuration file. All backup files for a file system will go to the same COS.

The relationship between storage class, storage hierarchy, COS is defined in the *HPSS Installation Guide* as well as the *HPSS Management Guide*.

2.6.3. File Families

File families are an abstraction of storage system characteristics that allows HPSS users to associate "like" files with a set of tapes. A file resides in a particular file family which is selected when the file is created (for a tape-only COS), or when a file is migrated from disk to tape based on the HPSS hierarchy.

There are three classes of GHI files written to HPSS. They are as follows:

- ***Data files (aggregate and non-aggregate files).*** By default, these files are not associated with a file family. However, the policy can be defined to specify the file family by adding a "'OPTS –f <file family>:auto'" or "'OPTS –f .<file family>' in the policy for aggregates and non-aggregates respectively.

  NOTE: Each GHI file belongs to a single file family which is selected when either the file is created (tape-only COS), or when the file is migrated from disk to tape based on the HPSS storage hierarchy.

- ***Aggregate index files.*** By default, these files are not associated with a file family. However, the policy can be defined to specify the file family by adding a "'OPTS –f auto:<file family>'" in the policy. The "auto" tells the system to not use a file file family for the data file. However, the policy can be written as "'OPTS -f <file family>:<file family>'" to associate both the

data and index file file file families.

NOTE: Each GHI file belongs to a single file family which is selected when either the file is created (tape-only COS), or when the file is migrated from disk to tape based on the HPSS storage hierarchy.

- ***Backup files*.** There is currently no way to associate backup files with a file family.

The relationship between storage class, storage hierarchy, COS and file families are defined in the *HPSS Installation Guide* as well as the *HPSS Management Guide*.

2.6.4. Storage Subsystems

Storage subsystems are provided in HPSS for the purpose of increasing the scalability of the system, particularly with respect to HPSS Core Servers. An HPSS system consists of one or more subsystems, and each subsystem contains its own Core Server. If multiple Core Servers are desired, this is accomplished by configuring multiple subsystems.

This can be used to separate HPSS resources for files being migrated via GHI vs. other files being written to HPSS. Only 1 subsystem is supported per GPFS file system.

2.7. DB2

A DB2 database on the HPSS Core Server node is used to store GHI metadata. The metadata describes each of the GHI backups. Since the GHI database resides on HPSS, the backup procedures for the HPSS metadata will need to be modified to backup the GHI database as well.

Integrity of the DB2 backup images is important to HPSS, GHI and GPFS recovery. Mismanagement or corruption in the backup images could impede recovery. Multiple copies can be created, validated, and managed with each set of log files. It is recommended that backup image copies are placed on a physically separate disk and disk controller from the primary copies. That way, the disk and disk controller cannot be a single point of failure.

2.8. GHI Security Considerations

The security requirements between GHI customer environments differ widely. The GHI System Administrators must be aware of the sites security requirements and should be aware of the security configuration required in HPSS. GHI Administrators should contact their site security representative if they have questions regarding security. For more information on security, see *Chapter 2*: *Security and System Access* of the *HPSS Management Guide*.

2.9. Technology Insertion

As new types of digital storage technology are configured into the system, the HPSS Storage Class definition may be updated to the new device and media characteristics. Existing file contents are accessed normally, but all new file data migrations will use the updated definitions and new media.

2.10. Policy Considerations

When GPFS processes the generated policy files they are called in the order the rules appear in the policy. This means that GHI will not initiate proccessing the files generated for rule 2 in the policy file until processing on all the files selected by rule 1 has been initiated. To avoid a delay in processing the generated files, the policy file should be constucted so that the rules that select the least number files are first.

This only applies to the migration of files using GHI. Recalls and Backups are not affected.

3. GHI COMMANDS

3.1. ghiapplypolicy

ghiapplypolicy <file system> -P <policy file> <other mmapplypolicy arguments>

The ghiapplypolicy command is a wrapper for the GPFS mmapplypolicy. The command accepts all the mmapplypolicy arguments. If the arguments are not supplied, ghiapplypolicy will use the configured values instead.

Default values are:

- -g <mount point>/scratch/.ghi
- -f <mount point>/scratch/.ghi
- -B <Max Aggr Size from ghi.conf>
- -N /var/hpss/ghi/etc/ghinode.conf

3.2. ghi_admin

ghi_admin

The ghi_admin command allows the administrator to cancel transfer requests, halt an inprogress backup, and reset the regions/dmapi attributes for a file. This tool should not be used without guidance from an HPSS support representative.

The following features are available:

- Lookup a file in the Scheduler Daemon queue
- Cancel transfer request for a file (currently not supported)
- Move a file to the top of the Scheduler Daemon Queue
- Reset the DMAPI Extended attributes and the Managed Regions for a file
- Stop a backup that is in progress
- Reinitialize the GHI Server processes
- Get the status of transfer requests for files in filelist
- Cancel transfer requests for files in a filelist (currently not supported)
- Retrieve file data that resides in HPSS
- Reset the DMAPI Extended attributes and the Managed Regions for files in filelist

3.3. ghi_backup

ghi_backup <file system> [full|incr]

The ghi_backup command takes a backup of the GPFS file system. The command takes a snapshot of the GPFS file system, migrates any unmigrated files using the /var/hpss/ghi/policy/<file system>/backup_migration.policy, and then gathers the metadata for the GPFS file system with a second policy run.

Arguments are:

1. full – Take a full backup
2. incr – Take an incremental backup from the last backup taken

3.4. ghi_backup_manager

ghi_backup_manager <file system>

The ghi_backup_manager command allows the user to delete individual backups for a GPFS file system. The user is given a list of backups to choose from and then prompted to confirm the deletion.

When a backup is deleted any files that are no longer referenced by a GPFS file system or backup will be deleted from HPSS.

3.5. ghi_df

ghi_df <file system>

The ghi_df command allows the user to gather information about the GPFS file system. The command will run a GPFS ILM policy and shows the user:

- Number of and space used by scratch area files.
- Number of and space used by non managed files.
- Number of and space used by managed files.
- Number of purged files.

3.6. ghi_ls

ghi_ls [-a] [-c] [-e] [-E] [-f] [-h] [-H] [-l] [-n] [-R] [-u] [file…]

The ghi_ls command is similar to the UNIX ls command. It has the output of the ls command but adds some additional information. The utility interacts directly with GPFS via DMAPI without HPSS, so users can determine the residency of their file data without regard to the availability of other HPSS services.

The residency of a file is expressed by a single letter a the start of the output:

- **G**: The file is GPFS resident and has not been migrated to HPSS.

- **B**: The file is dual resident. The data exists in both GPFS and HPSS.
- **H**: The file is HPSS resident. The file data has been purged from GPFS.

Other output is displayed by using the traditional ls command line options. For example the ***"-l"*** option can be used to generate a listing that contains the file residency plus permissions, timestamps, uid, and gid.

Command line options specific to ghi_ls are:

-h Displays where the file resides in HPSS along with if it is an aggregate file.

-e Displays the HPSS file attributes included bytes stored on the different Storage classes.

For more information about the supported command line options you may type ***ghi_ls -?*** at the command line.

3.7. ghi_mon

ghi_mon <file system> [iom|sd] [-a action] [-f frequency] [-p output path]

The ghi_mon command allows the user to start the GHI monitoring of the system or IOM statistics.

3.8. ghi_pin

ghi_pin [-v] [-u] {file | wild card | -f filelist}

The ghi_pin command allows the administrator to flag a GPFS file so that it does not get purged during a threshold policy run. This will ensure that the data remains on the GPFS file system. The administrator may specify a single filename, filenames with wildcards, or a list of files.

3.9. ghi_restore

ghi_restore [-R BUIdx] [-v] <file system>

The ghi_restore command allows the administrator to select a backup to restore to the GPFS file system. This will rebuild the GPFS namespace and DMAPI attributes used by GHI to map files to HPSS. No file contents are restored. It is up to the site administrator to generate a recall ILM policy based on restoring files in priority order.

3.10. ghi_stage

ghi_stage [-t timeout] [–v] {file|directory|-f filelist}

The ghi_stage command allows the user to stage files from HPSS without running a policy or waiting for a DMAPI event to complete. The command allows the user to input a list of files to stage. The files must be in the same GPFS file system.

Optional arguments are:

- ***-t timeout*** This is how long the command will wait for the stages to complete.
    a. ***–t 0*** means don't wait
    b. No argument means wait forever.
- ***-v*** This tells ghi_stage to enable verbose output.

3.11. ghi_state

ghi_state <file system>

The ghi_state command displays the state of the system. The following information is displayed:

- Cluster status.
- Disk status.
- Mount status.
- Manager status.

4. GHI MANAGEMENT

4.1. Start/Stop GHI Servers

The GHI processes are designed to start automatically. The GHI session node processes start when GPFS starts up. This is done by using the GPFS call back functionality. File system specific processes are started by GHI when the file system is mounted.

The IO Managers are managed by each node by the ***init*** process.

4.2. GHI Process Failure/Recovery

GHI needs to provide a fault tolerant system in order to keep the file system online and available. GPFS supports a means for GHI to provide exit scripts to be notified when there are changes in the quorum. This mechanism will allow GHI to either migrate the processes to another node, or do what is needed to stay running on the existing node. There are currently eight events that can be captured for this purpose (init, ready, up, down, node failure, file system recovery, pre-unmount, quorum loss).

The events will invoke either a single "user" defined script only, an HA/NFS define script only, or both. The scripts will be invoked on those nodes that have the exit script installed.

4.2.1. Node Failures

4.2.1.1. Session Node

The node defined as the Session node is selected by GPFS when the system is brought online. It is typically the node that is the GPFS cluster configuration Manager node. GHI will utilize the GPFS heartbeat mechanism to monitor the nodes in the cluster that are potential Session node candidates. During startup, GPFS will execute the script, ***hpssEventNotify***, to start all GHI processes and mount the file systems. Likewise, during failure, GPFS will execute the script, ***hpssEventNotify***, to unmount the file systems and stop all GHI processes. If the node fails, and another node needs to take over, GPFS will select the new Session node.

4.2.1.2. Manager Node

The nodes running the I/O Managers start the processes using ***inittab***. The I/O Managers, once started, will remain idle until the file system is mounted on that node. The I/O Manager will start one or more I/O agents by sending a request to the host/port based on the configuration file. The I/O Agents are configured in ***inetd***.

If an I/O Manager node fails, there are two scenarios:

- The Scheduler will lose the connection to the I/O Manager, and will cancel all requests to the failed I/O Manager and send them to a new I/O Manager that is active.
- If a policy script was being run on the node that failed, the policy manager will be notified that the request failed. In the case of a backup, the backup will have to be rerun. In the case of a migration/recall/purge, no action will be taken.

4.2.1.3. Client Node

There is no special failover logic for these processes. If the node fails, the I/O Manager will detect a completion failure of transferring the data. There is retry logic in the IOM to retry the data transfer, if the request is for a non-aggregate. For GHI-HTAR requests, the IOM does not spawn off a new GHI-HTAR process if it return from GHI-HTAR indicates a failure.

4.2.2. Single Process Failures

4.2.2.1. ILM Client

These processes, ***hpssmigrate***, ***hpssrecall***, ***hpssdelete***, and ***hpsslist***, are started from the corresponding ***ghi_migrate***, ***ghi_recall***, ***ghi_delete***, and ***ghi_list*** policy scripts to perform the requested action. They are used to bridge the communication between the scripts and the GHI Scheduler.

If one of the GHI scripts detect that the HPSS process has terminated abnormally, the process will be restarted. The new process will start processing the policy file from the beginning.

4.2.2.2. Process Daemon

In the case of an error or termination of GPFS on the GHI Session node, the ***hpssEventNotify*** script will be executed. Running this script will shutdown the GHI processes. The script will notify the Process Manager to shut the other processes down, and then terminate itself.

If the Process Manager process terminates abnormally, the Mount Daemon will detect it, and restart it. Once restarted, the Process Manager will have to terminate the Mount Daemon, Event Daemons, and the Schedulers so that the process has control of the SIGCHLD signal for those processes. Those processes will then be restarted.

4.2.2.3. Mount Daemon

Failure of the Mount Daemon will impact the file system by failing to allow file systems to mount. If the Mount Daemon abnormally terminates, the Process Manager will automatically restart it. There is no special recovery logic for this process.

Failure of the Mount Daemon will impact the file system *mount* and *unmount* requests. Those requests that are not handled will simply hang, and the user will need to kill and retry the *mount* or *unmount* requests. *Mount* and *unmount* requests can only be handled when the Mount Daemon is registered to receive those DMAPI events.

If the Mount Daemon abnormally terminates, the Process Manager will automatically restart it. There is no special recovery logic for this process. It will wait for new *mount/unmount* requests.

4.2.2.4. Log Daemon

Failure of the Log Daemon will impact the file system by failing to log critical log messages. If the Log Daemon abnormally terminates, the Process Manager will automatically restart it. There is no special recovery logic for this process.

4.2.2.5. Event Daemon

Failure of the Event Daemon will have a severe effect on the file system since the process is tightly coupled to file system user activity. For example, if the Event Daemon stops responding to synchronous events, the user processes that generated the events will block indefinitely.

If the Event Daemon abnormally terminates, the Process Manager will automatically restart it. Upon restart, it will assume the current Session ID, and check for outstanding DMAPI events. The ED will add the events to the internal queue and then wait for responses from the Scheduler. It will then do normal processing and wait for new DMAPI events.

4.2.2.6. Scheduler Daemon

The Scheduler Daemon will be started and monitored by the Process Manager. If the Scheduler process abnormally terminates, the Process Manager will restart it. There is no special recover logic for this process. The outstanding scheduled tasks will be lost, as well as the tasks being worked by the IOMs.

All client requests that were being processed by the Scheduler at the time it terminated will result in failures to the client.

4.2.2.7. I/O Manager

I/O Managers will be started using the ***inittab*** on the IOM host systems. If the I/O Manager abnormally terminates, it will be automatically restarted by ***inittab***. There is no special recovery logic for this process. It will wait for requests from the Scheduler.

4.2.2.8. I/O Agent

There is no special processing for the I/O Agent for termination and restart. This process is responsible for transferring a piece of data using the HPSS PIO Interface for data transferring to or from HPSS. If this process fails, the I/O Manager will detect the error, and start a new I/O Agent.

4.2.3. Multiple Process Failures

4.2.3.1. ILM Client and Scheduler

If one or more ILM clients abnormally terminate and the Scheduler terminates as well, the original request will have to be resent when the client and Scheduler are restarted.

4.2.3.2. Scheduler and Event Daemon

If the Scheduler and Event Daemon abnormally terminates, the Process Manager will automatically restart both processes. Upon restart, The Event Daemon will send any outstanding DMAPI requests to be processed. Those requests will be sent to one or more I/O Managers, and if the files have already been staged, the managed regions will be updated if needed, and a successful response will be sent back to the application. Otherwise, the I/O Manager will stage the file.

4.2.3.3. Scheduler and I/O Manager

If the Scheduler and one or more I/O Managers abnormally terminates, the Process Manager Scheduler will restart it, and the I/O Manager(s) will be restarted by ***inittab***. All outstanding requests being processed by the I/O Manager, which was abnormally terminated, will have to be re-tried by the application. The Event Daemon will resend all DMAPI requests, which may cause duplicate stages being performed.

4.2.4. HPSS Unavailability

When HPSS is unavailable in the GHI system, most file system operations will continue to work. The operations that require data to be transferred between GPFS/HPSS will fail. The following operations will fail:

- User read events on co-managed files. Files where the data only reside in HPSS cannot be staged back to GPFS.
    - When a user requests the file through a DMAPI event, an abort will be sent to the application.
- All policy manager runs.
    - Files that are recalled using the ILM interface will return an error to ***ghiapplypolicy***. Files that are to be migrated/pre-migrated using the ILM interface will return an error to ***ghiapplypolicy***.

- Backups will fail with an error.

4.3. System Monitoring

GHI provides a monitor utility, ***ghi_mon***, to watch the major activity on the system. GHI currently supports watching the Scheduler Daemon and the IOM progress. Monitoring can be scheduled to start when the SD and/or IOMs are online. This is done by configuring it in ***ghi.conf***. Othewise, the user can call ***ghi_mon*** <***task***> to start the task.

4.3.1. Scheduler

The following figure depicts the internals of the Scheduler.

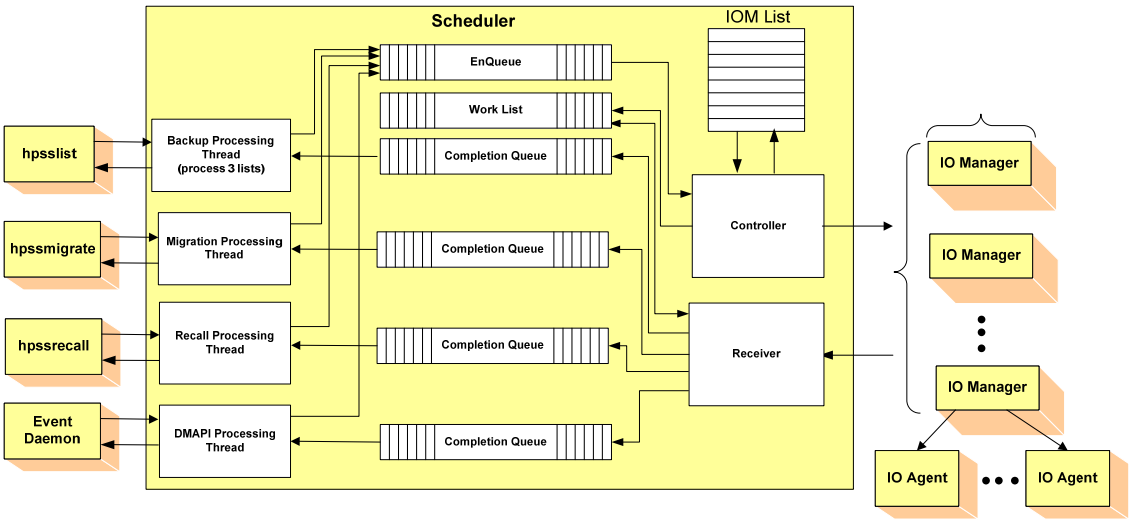

Figure 13 - Scheduler Internals

The Scheduler will contain four different Schedule Queues:

- Backup requests.
- Migration requests.
- Recall requests.
- DMAPI requests from the Event Daemon.
  - Note: Deletion requests do not go directly to the scheduler. Deletes generate a destroy event that goes to the Event Daemon.

As requests come in, the Scheduler will place the items on the appropriate queue. As they are worked off, they are placed on the Work List.

When monitoring the Scheduler, a two lines will be printed out. The first line contains the following information:

- ***Mode***: The mode of the SD:
    - Active: The SD is running in active mode
    - Backup: The SD is running a backup
- ***Queued***: Current number of requests on The Schedule Queue.
- ***Working***: Current number of entries being worked off.
- ***Migrations(A/N)***: Total numbers of aggregates/non-aggregates that have been processed since the Scheduler was started up Aggregates are counted as one per aggregate, and not the total number of requests within an aggregate.
- ***Recalls***: Total number of files that have been recalled since the Scheduler was started. This can be misleading if there were multiple recalls in a single aggregate. The aggregate is only counted as a single increment.
- ***Stages*:** Total number of stage request.
- ***Purged***: Total number of files that have been purged.
- ***Deletes***: Total number of file deletes
- ***Backup index:*** The index for the last good backup.
- ***IOM(A/T)***: Total number of IOMs that are active/Total number of IOMs configured.

The second line contains the following information:

- ***Message***: Critical error message for the system
- ***General errors:*** General error count for issues like ghi_ed communication or DB2 errors.
- ***Migration failures (A/N):*** Total number of migration failures for aggrs and non aggrs.
- ***Recall failures:*** Total number of recall failures
- ***Stage failures:*** Total number of stage failures
- ***Purge failures:*** Total number of purge failures
- ***Delete failures:*** Total number of delete failures
- ***Backup failures*:** Total number of Backup metadata transfer failures

4.3.2. I/O Manager

The following figure depicts the internals of the I/O Manager.

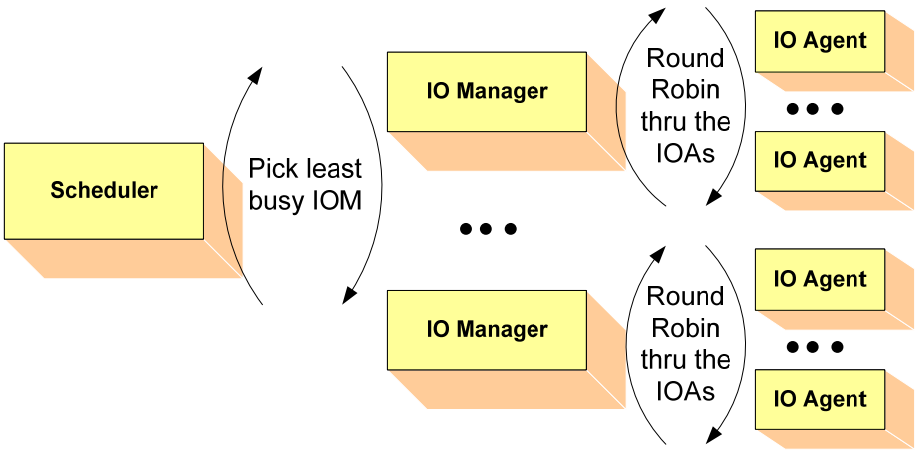

Figure 14 - I/O Manager Internals

The Scheduler sends requests to the configured/active IOMs based on which IOM has the least load until the IOMs have hit the max number of simultanous requests. Once the IOMs are at the max number of requests, jobs are sent to IOMs only after the IOM has completed a request. The IOMs spawn off requests as they are received from the Scheduler.

The IOM can be in four different states:

- ***Active*:** The IOM has the file system mounted, and all the connections are valid.
- ***Inactive*:** The IOM is running on the session node, and it configured to be in an inactive state.
- ***Standby*:** Either the file system is not mounted on the node, or the connection from the IOM to the Scheduler is not valid.
- ***Econn*:** The Scheduler has lost the connection to the IOM.

When monitoring the IOMs, a two lines will be printed out for each IOM configured.

The first line contains the following information:

- ***State***: The current state of the IOM (see node definitions above)
- ***Total Completed:***
- ***Failures***: Total number of errors encounted since the IOM was started.
- ***Requests***: Current number of requets that the IOM is working on.
- ***Workload:*** The current number of bytes that the I/O Manager has remaining to transfer.
- ***Processing rate:*** How fast the IOM is working the information off. It is

not the transfer rate to HPSS as it included other activities like setting DMAPI attributes, creating the HPSS file, etc.

- ***Node***: Hostname/Port where IOM is running.

The second line contains the following information:

- Longest job start time
- Longest job size
- Longest job file information

4.4. GPFS Configuration Modifications / Changes

4.4.1. Cluster Configuration

4.4.1.1. Adding a Node

When a node is added to the cluster, and is going to run an IOM, it needs to be added to the ***ghinode.conf*** file. It must be added to the end of the file, so it will not disturb the indexing which is used to determine the Session Name for that IOM.

If an IOM is to be configured on that node, update the ***ghi_<file system>.conf*** file. It should then be distributed to the other nodes. The Scheduler will have to be restarted to pick up the new node.

The IOM/IOA and GHI-HTAR, if required for aggregates, should be configured on the new node.

4.4.1.2. Deleting a Node

When deleting a node from the cluster, and it was running an IOM, the entry in the ***ghinode.conf*** file must be maintained, so that the Session Ids being used by the other IOMs will be kept intact.

4.4.1.3. Modifying a Node

If the IOM portion of the ***ghi.conf*** file is modified, the IOM needs to be recycled.

If an IOM is to be configured on that node, the ***ghi_<file system>.conf*** file needs to be updated and then be distributed to the other nodes. The Scheduler will have to be restarted to pick up the new node.

The IOM/IOA and GHI-HTAR, if required for aggregates, should be configured on the new node.

If the IOM is being taken off that node, a placeholder must be kept in the ***ghinode.conf*** file.

4.4.1.4. Adding a File System

When adding a file system, a stanza needs to be added to the ***ghi.conf*** file. A new ***ghi_<file_system>*** file will have to be configured. The updated/new files will have to be copied to the other nodes in the cluster running GHI processes. No processes will need to be recycled.

4.4.1.5. Deleting a File System

When deleting a file system, the ***ghi.conf*** file should be cleaned out, but no GHI servers need to be recycled.

4.4.2. Locating a File

The user utility, ***ghi_ls***, will show where a file has resident data. This utility can be invoked by a user but will have to access DM attributes, so it will need to use the sticky bit to run as ***root***. The utilility interacts directly with GPFS via DMAPI without HPSS, so users can determine the residency of their file data without regard to the availability of other HPSS services.

If invoked against a file, it will list the state of the file. If invoked against a directory, it will list the state of all files in that directory. The command will display a variety of options:

- Display residency of a file, expressed as a single letter:
  - **G**: The file is GPFS resident and has not been migrated to HPSS.
  - **B**: The file is dual resident. The data exists in both GPFS and HPSS.
  - **H**: The file is HPSS resident. The file data has been purged from GPFS.
- Display GPFS extended attributes for all files which reside in HPSS.
- Display GPFS extended attributes for all files which reside in HPSS.
- Display HPSS attributes for all files which reside in HPSS.
- Display HPSS attributes for all files which reside in HPSS. UNIX and HPSS data will appear on separate lines to minimize chances of text wrapping around terminal screen.
- Display UNIX details similar to "***ls -l***".
- Display UID and GID.

4.5. Modifying the ghi.conf file

The following is a list of the values in the ***ghi.conf*** file. This section discusses what actions need to be taken when this file is modified. For any changes to the master

copy of this file, requires that this file be distributed to all the nodes in the cluster that are running GHI.

- ***Logging***: This value will require one or more servers to be recycled, depending on which server needs to be affected.
- ***Mount Point***: This value should not change after the initial configuration.
- ***HPSS Junction***: This value may not change after the initial configuration.
- ***Unique Identifier***: This value should not change after the initial configuration.
- ***Max Aggregate Files***: Nothing needs to be recycled. It will take effect during the next policy run.
- ***Min Aggregate Files***: Nothing needs to be recycled. It will take effect during the next policy run.
- ***Aggregate Index COS***: Recycle the I/O Managers.
- ***Aggregate Thread Count*:** Recycle the I/O Managers.
- ***Backup Bulk Count*:** Nothing needs to be recycled. It will take effect during the next policy run.
- ***Backup COS*:** Recycle the I/O Managers.
- ***HPSS Base Path*:** This value may not change after the initial configuration.
- ***HPSS Backup Path*:** This value may not change after the initial configuration.
- ***Purge Blocksize:*** Recycle the Scheduler.
- ***Performance Logging:*** Recycle the associated server based on what logging was turned on/off.
- ***Event Daemon Subsection*:** Recycle the Event Daemon.
- ***Log Daemon Subsection*:** Recycle the Log Daemon.
- ***I/O Manager Subsection*:** Recycle the I/O Managers.
- ***Scheduler Daemon Subsection***: Recycle the Scheduler.

4.6. Upgrade DB2

Use the DB2 upgrade instructions to upgrade to the required DB2 version. Verify the DB2 levels on the GHI nodes are synchronized with the version on HPSS.

4.7. Upgrade GHI

4.7.1. Prepare GHI Code

Install and, if necessary, compile the GHI distribution image.

4.7.1.1. Install GHI Distribution Image

Prior to installing the GHI code, the HPSS Client API software package must be installed. Install the GHI distribution images in ***$BUILD_TOP_ROOT/src/ghi***.

4.7.1.2. Compile GHI Source Code

Compile the GHI code by running ***make*** in the ***ghi*** directory.

4.7.1.3. Perform Remote I/O Manager Configuration

Perform the required additional configuration on each remote I/O Manager node. The following steps need to be performed:

- Copy the ***/opt/hpss/lib/**** files to the remote node.
- Copy the ***/opt/hpss/bin/ghi_ioa*** executable to the remote node.
- Copy the ***/opt/hpss/bin/ghi_iom*** executable to the remote node.

4.8. Upgrade GPFS

Refer to the GPFS Administration and Programming Reference to upgrade GPFS.

Site administrators must coordinate with their HPSS support representative to get concurrence before upgrading GPFS to a newer version or PTF level.

4.9. Upgrade HSIGWD/GHI-HTAR

Refer to the http://www.mgleicher.us/HSI_Admin URL for upgrade instructions.

4.10. Upgrade HPSS

Refer to the *HPSS 7.3 Conversion Guide* for upgrade instructions.

Once the software for the HPSS Client API is upgraded, the GHI sofware needs to be rebuilt. The libraries and executables will then need to be distributed to all the nodes running GHI processes.

4.11. Daily Monitoring of the System

To monitor the system, it is recommended that sites perform the following actions on a daily basis:

- Save output from each of the policy runs to be reviewed. Since policy runs

can take several hours, it is recommended to save the output for each of the runs in a specific location to be reviewed on a daily basis. Policy runs generate **.ok* and **.exc* files; Policies use the "*-b*" option to save these files. It is up to the site administator to delete these files.

- Review the output from a backup on a daily basis in detail. Output will indicate success or failure, but the details need to be reviewed to acknowledge criticality of individual failures.
- Monitor SD and IOM output generated from the *ghi_mon* utility. By default, the output is activated during system startup time. Refer to the *ghi.conf* configuration file.
- Monitor the output in the central log files: */var/hpss/ghi/log/logfile*.log.*

5. PROBLEM DIAGNOSIS AND RESOLUTION

This chapter provides problem determination and resolution advice for GHI infrastructure components, servers, and user interfaces. Note that a problem may have more than one diagnosis and resolution.

GHI logs events, minor, major and critical log messages to the central log file in the /var/hpss/log directory. Each server has additional logging that can be turned on/off. The following message types can be turned on or off:

- DEBUG: Logs information about variable and structure contents.
- TRACE: Logs function entry and exit points.
- INFO: Logs information used by administrators to indicate code paths.

To turn on additional logging for the system, the following steps need to be performed:

- Update the ***/var/hpss/ghi/etc/ghi.conf*** file to add the required logging.
- Send a ***kill SIGHUP <pid>*** for the process that needs more logging turned on.

To turn the logging off:

- Update the ***/var/hpss/ghi/etc/ghi.conf*** file to add the required logging.
- Send a ***kill SIGHUP <pid>*** for the process that needs logging turned off.

5.1. GHI Infrastructure Problems

The sections below describe possible RPC and Security infrastructure errors.

5.1.1. RPC Problems

5.1.1.1. One GHI server cannot communicate with another

***Diagnosis 1*:** The target server may not have registered its RPC endpoint properly.

***Resolution**:* Verify proper registration of the server with RPC. If shutting down the target server and restarting it does not fix the problem, you may have to manually delete the server's RPC entry.

***Diagnosis 2*:** A communications failure may exist or security may be disallowing communication.

***Resolution**:* First verify that the network is up and the server is running. A less obvious cause for the problem may be that the server is not accepting calls from the client because of security reasons. To fix this problem, make sure that the

client and server are using consistent security policies, and that they have authenticated properly.

Diagnosis 3: The ***/var/hpss*** file system may be full.

Resolution*:* If ***/var/hpss*** is full, try to determine what is causing the file system to fill up. Common problems are ***/var/hpss*** being too small, or log files that are not being archived properly.

Diagnosis 4: A server may be too busy to respond.

Resolution*:* If a server is very busy, other servers will not be able to communicate with it. To solve the problem, decrease the load on the server. For example, try increasing the server's thread pool size and/or maximum connection count, moving the server to a different machine, or adjusting one of the server-specific configuration parameters.

Diagnosis 5: A node does not have a network route to an interface being used by a server on a different host.

Resolution: Verify that all nodes (Session node and IOM nodes) have network routes to the network interfaces required.

Diagnosis 6: The server may not have enough RPC connections configured that are necessary for communication.

Resolution*:* Increase the number of **Maximum Connections** in ***ghi.conf*** for the appropriate server configuration.

Diagnosis 7: The server may be trying to communicate with DB duing their initilization.

Resolution*:* Check for DB2 errors on the core server. Verify that the GHI node can connect to the DB2 server on the HPSS core.

5.1.1.2. A server configuration is missing or incorrect

Diagnosis 1: The information in the ***ghi.conf*** file is either missing or incorrect.

Resolution: Review the configuration template and detemine which field is incorrect.

5.1.1.3. A server cannot obtain its credentials

Diagnosis 1: There may be a problem with the keytab table.

Resolution: Make sure the keytab table (usually ***/var/hpss/etc/hpss.keytabs***) is readable by the UNIX username under which the server is running. Make sure that the key contained in the keytab table is the correct one. Look for extra versions of the servers key; they can interfere with the authentication process.

5.1.1.4. A server cannot register its RPC info

Diagnosis 1: Stale RPC information may exist for the server in the RPC table.

Resolution: Issue the ***rpcinfo -p*** command to see if the RPC program number for the server interface is already registered. If the interface is registered it can be removed using the ***rpcinfo -d*** <***program number***> <***version***> command.

5.1.1.5. The connection table may have overflowed

Diagnosis 1: The server may be so heavily loaded that it is unable to free up connections easily.

Resolution: Reduce the load on the server. The problem may also indicate that a server is configured incorrectly, or that there is a software problem in handling connections properly. To solve the problem, increase the **Maximum Connections** parameter in the **ghi.conf** configuration file for the specific server.

5.1.1.6. Servers cannot talk to one another

Diagnosis 1: The Domain Name Service (DNS) is not reachable.

Resolution: Add all necessary entries to the /etc/hosts file. Terminate all GHI servers, DB2, and Kerberos. Restart the system without DNS support then fix the DNS.

5.2. GHI Server Problems

The paragraphs below discuss problems common to all servers.

5.2.1. Process Manager Problems

5.2.1.1. The Process Manager is unable to start

Diagnosis 1: Verify the /var/hpss/ghi/tmp/pm_pid_list does not exist.

Resolution: Remove the /var/hpss/ghi/tmp/pm_pid_list file.

Diagnosis2: Verify the GPFS callbacks are set up correctly.

Resolution: Configure the GPFS callbacks to start GHI.

5.2.1.2. The Process Manager dies after a mount request (PPC only)

Diagnosis 1: The Process Manager core dumps with a stack dump.

Resolution: Set the HPSS_PTHREAD_STACK=262144 in /var/hpss/etc/env.conf

5.2.2. Mount Daemon Problems

5.2.2.1. Failed to get events

Diagnosis 1: Failed to retrieve DMAPI events.

Resolution: Verify the file still exists. Also, verify the Session ID is valid for that node, and is not owned by another node.

5.2.2.2. Failed to respond to an event

Diagnosis 1: Failed to respond to file system mount/unmount request.

Resolution: Verify the file system still exists.

5.2.3. Event Daemon Problems

5.2.3.1. Failed to get events

Diagnosis 1: Failed to retrieve DMAPI events.

Resolution: Verify the file still exists. Also, verify the Session ID is valid for that node, and is not owned by another node.

5.2.3.2. Failed to respond to events

Diagnosis 1: Failed to respond to an event for a request accessing a file.

Resolution: Verify the process had not been restarted.

5.2.3.3. Failed to get attributes on a file

Diagnosis 1: When trying to get the attributes of a file, the call failed.

Resolution: Verify the file still exists. Also, verify the Session ID is valid for that node, and is not owned by another node.

5.2.4. Scheduler Daemon Problems

5.2.4.1. Out of completion queues

Diagnosis 1: There are too many requests for the scheduler.

Resolution: The additional requests will not be lost. They will wait until some existing connections are complete, and then the request will get scheduled.

5.2.4.2. Failed to set regions (punching a hole)

Diagnosis 1: Unable to set the regions for a file, dm_set_region.

Resolution: Verify the file still exists. Also, verify the Session ID is valid for that node, and is not owned by another node.

5.2.4.3. Failed to punch a hole in a file

Diagnosis 1: Unable to punch a hole in a file, dm_punch_hole.

Resolution: Verify the file exists.

5.2.4.4. Recovery started for an IOM

Diagnosis 1: An IOM abnormally terminated. The requests that it was working on are being redirected to another IOM

Resolution: There is no action to be taken.

5.2.4.5. Failed to get a DMAPI handle for a file

Diagnosis 1: When attempting to ready the DMAPI handle for file before performing a data transfer, the call failed.

Resolution: Verify the file has not been deleted. Also, verify the SessionID is still valid for that IOM.

5.2.5. I/O Manager Problems

5.2.5.1. The IOM is in ECONN mode.

Diagnosis 1: The IOM is initializing

Resolution: Wait until it finishes the connection logic. If it has not connected after a short time, contact your GHI Support Representative.

Diagnosis 2: The IOM is configured incorrectly in the ***ghi_<file system>.conf*** file. If the IOM cannot find its host entry or if no valid IOAs are found, it will terminate.

Resolution: Fix the **ghi_<file system>.conf** file, and once it is fixed correctly, distribute to the other nodes running GHI processes.

Diagnosis 3: The IOM is configured incorrectly in **/etc/inittab**.

Resolution: Verify the entry in inittab is correct, and if not, fix the entry and recycle ***inittab*** (***kill –1 <inittab pid>).***

Diagnosis 4: The authentication configuration is incorrect in ***/var/hpss/etc***.

Resolution: Copy the ***/var/hpss/etc/*** directory from the master location.

Diagnosis 5: There is a time difference between the IOM node and the Session node that is greater than 5 minutes.

Resolution: Run an NTP daemon on all the nodes, or change the date with the "date" command.

5.2.5.2. IOM is in STANDBY mode.

Diagnosis 1: IOM lost connection to the SD.

Resolution: Verify the Scheduler is running. Recycle the IOM.

Diagnosis 2: File system is not mounted on that node.

Resolution: Mount the file system on that node.

5.2.5.3. Failed to make a handle to a file

Diagnosis 1: Unable to get a handle for a file.

Resolution: Verify the file still exists. Also, verify the SessionID is still valid for that IOM.

5.2.5.4. The IOM dies unexpectedly

Diagnosis 1: The IOM core dumps with a stack dump.

Resolution: Set the HPSS_PTHREAD_STACK=262144 in /var/hpss/etc/env.conf

5.2.6. I/O Agent Problems

5.2.6.1. The I/O Agent fails to start.

Diagnosis 1: The IOA is not configured correctly in /etc/services.

Resolution: Correct ***/etc/services***. Recycle xinetd. Run the "***netstat –an | grep <port>***" command to verify it is configured correctly.

Diagnosis 2: The IOA is not configured correctly in ***xinetd***. Sometimes the ***xinetd*** has a different name than the name used in **/etc/services**.

Resolution: Correct ***xinetd***. Recycle ***xinetd***. Run the "netstat –an | grep <port>" command to verify it is configured correctly.

5.2.7. GHI-HTAR Problems

5.2.7.1. GHI-HTAR fails to communicate with the HSIGWD

Diagnosis 1: Check the ndapi.log file. There is no output in the ndapi.log

Resolution: Telnet localhost 1217 and verify it appears to hang, and if not, ***/var/log/messages*** is a good place to look for problems. If so, then type ***"<cntl><enter>*** to get back to the telnet prompt and the "quit. When this happens, there should be output in the ***ndapi.log*** file.

Run a sample GHI-HTAR to verify it is correct: *"**/opt/hpss/bin/htar.ksh –cvf /ghi/xxx /etc/motd***".

5.2.7.2. GHI-HTAR fails to run

Diagnosis 1: GHI-HTAR fails with "ndad_keytab_check: failed (code=22) for principal hpssdmg" found in the ndapi.log

Resolution: Issued a new hpss.htar.keytab and distribute the keytab file to all of the GHI nodes. Verify the ***/var/hpss/etc/unix.master.key*** does not contain all zeros.

Diagnosis 2: GHI-HTAR fails with 18431.

Resolution: Unable to open file.

Diagnosis 3: GHI-HTAR fails with 18432.

Resolution: The ***htar.ksh*** file contains the incorrect value for ***HPSS_HOSTNAME***. Fix the ***htar.ksh*** file and run a quick command line test to verify: ***"/opt/hpss/bin/htar.ksh –cvf /ghi/testfile /etc/motd***".

Diagnosis 4: GHI-HTAR fails with -52 htar_GhiClose, Error setting managed regions".

Resolution: Verify the SessionID is correct.

Diagnosis 5: The HSIGWD executable is not found in ***/opt/hpss/bin***.

Resolution: Run "netstat –an | grep 1217" to verify it is listening correctly.

Diagnosis 6: GHI-HTAR fails to open the file in GPFS.

Resolution: Verify the file still exists.

5.2.7.3. GHI-HTAR appears to be hung or locked up.

Diagnosis 1: The location for the GHI-HTAR temporary files is full: "WARNING: OUT OF SPACE writing HPSS archive - delaying/retrying".

Resolution: Verify the file system is not full. Kill the htar.ksh process. Clean up the file system, or allocate more space. Rerun the policy.

5.3. Policy Interface Problems

The paragraphs below discuss interface problems with running the policy for migrations and recalls.

5.3.1. Migration problems

These problems are displayed as output from the ***ghiapplypolicy*** run.

5.3.1.1. A "-1 makeXHandle" error was encountered

The output from a migration policy failed with a call to makeXHandle.

Diagnosis 1: The ***/var/hpss/ghi/etc*** directory is inconsistent with the rest of the

nodes in the cluster.

Resolution: Make sure the directory is consistent with the other nodes in the cluster. Recycle the IOM.

5.3.1.2. A "-5 PIOXferMgr" error was encountered

The output from a migration policy shows a file failed with "-5 PIOXferMgr" error.

Diagnosis 1: The HPSS Movers are having issues (they are not green), or there are errors in the Alarms & Events, or the local.log file.

Resolution: See Chapter 1: HPSS Problem Diagnosis and Resolution in the HPSS Error Manual.

5.3.1.3. A "-28 PIOXferMgr" error was encountered

The output from a migration policy shows a file failed with "-28 PIOXferMgr" error.

Diagnosis 1: The HPSS Movers are having issues (they are not green), or there are errors in the Alarms & Events, or the local.log file.

Resolution: See Chapter 1: HPSS Problem Diagnosis and Resolution in the HPSS Error Manual.

5.3.1.4. A "-78 PIOXfer" error was encountered

The output from a migration policy shows a file failed with a "-78 PIOXfer" error.

Diagnosis 1: The inetd was incorrectly configured.

Resolution: Refer to Section *5.2.5 - I/O Manager Problems* for diagnosis and resolution.

Diagnosis 2: The HPSS Mover(s) are having issues.

Resolution: Verify the HPSS Mover(s) are green. Also, look at the HPSS ***local.log*** to see if there are any error messages.

Diagnosis 3: The HPSS enviroment was not configured correctly.

Resolution: Verify the /var/hpss/etc/env.conf file is configured with the correct HPSS_API_HOSTNAME.

5.3.1.5. GHI-HTAR failed

The output from a migration policy shows that GHI-HTAR failed.

See Section *5.2.7 - GHI-HTAR Problems* for diagnosis and resolution.

5.3.2. Recall problems

These problems are displayed as output from the ghiapplypolicy run.

5.3.2.1. A "-78 PIOXfer" error was encountered

The output from a recall policy shows a file failed with a "-78 PIOXfer" error.

Diagnosis 1: The ***inetd*** Daemon is configured incorrectly.

5.4. File System Problems

The sections below discuss problems encountered when reading or writing to the filesystem. It also discusses problems if the file system is filling up and files are not being purged.

5.4.1. Mounting file system problems

Diagnosis 1: The ***mount*** command returned an error for "Stale NFS Handle"

Resolution: There is potentially a disk problem with the file system, run ***ghi_state***. Also verify the file system is configured in the ***ghi.conf*** file.

Diagnosis 2: The ***mount*** command returned an error saying there was no handle for the mount event.

Resolution: Verify the PM and MD are running.

Diagnosis 3: The Mount Daemon is not running.

Resolution: Verify the Session node did not failover. Verify the Mount Daemon is running on the Session node and was able to take over the Session ID.

5.4.2. Threshold problems

5.4.2.1. Error indicating file is not managed by HPSS.

The output from a policy run shows a file failed with an error indicating that the file is not managed by HPSS.

Diagnosis 1: The purge policy that was configured is not excluding files that are already co-managed.

Resolution: Verify the policy has an entry as follows:

Rule "exclude_rule" EXCLUDE FROM POOL "system" WHERE MISC_ATTRIBUTES NOT LIKE '%M%'

This rule will exclude files that are not HPSS managed.

5.4.2.2. A file fails to purge data blocks from GPFS

Diagnosis 1: Verify the file was not deleted after it was selected as a purge candidate.

Resolution: There is nothing to be done here.

Diagnosis 2: Verify the file meets the purge policy criteria

Resolution: Review the policy as well as the location of the file (via ***ghi_ls***).

Diagnosis 3: Verify the file is larger than 1 data block.

Resolution: Files that are less than or equal to 1 data block will not be selected as purge candidates.

5.4.3. File read/write problems

5.4.3.1. Failed to read/write a file in the file system

Diagnosis 1: The request returns an Input/Output error.

Resolution: Verify neither the Event Daemon nor the Scheduler has been recycled recently. Verify the HPSS Mover(s) are green. Also, look at the HPSS ***local.log*** to see if there are any error messages.

5.4.3.2. Reading/Writing a file appears to hang

Diagnosis 1: The request returns a timeout.

Resolution: Find out where the file resides in HPSS. If it resides on tape, verify there is not an outstanding tape mount request, and there is a free tape drive.

5.4.3.3. A "-2" error was encountered reading a file

The output from an attempt to read the file exits with a -2. This includes things like a cp or mv system command.

Diagnosis 1: GHI was attempting to migrate the file when the user attempted to access the file.

Resolution: Attempt to access the file again.

5.5. GHI Utility Problems

5.5.1. General Utility Problems

Here is a list of items to check when a utility is not running as expected:

- Check command-line syntax. Most utilities will print a usage summary if they are invoked with the **-?** option. Some utilities require several parameters to be specified that may not be obvious.
- Make sure that default arguments are being overridden when necessary. Many utilities use default values for several of their parameters. If the parameter is not overridden with a specific value, unexpected behavior may result.

5.5.2. ghi_mon Problems

5.5.2.1. The ghi_mon SD error count increases

Diagnosis 1: Numerous errors are incountered during migrations, recalls, stages, and/or purges.

Resolution: Review the actions being performed, and determine increase rates to detemine which action is generating the errors. Review the ***ghi_mon*** output for the IOM as well to determine which IOM is encountering the problem.

5.5.2.2. The ghi_mon IOM error count increases

Diagnosis 1: The errors displayed from ***ghi_mon*** inidcate the IOM was failing when perform data transfers.

Resolution: Review the exc files for that IOM (if running with the –d or –D) option. Otherwise, run a migration and review the output from the policy run to determine the issues are. Check the GHI logs to see what issues the system is having. Also, review the local.log file to see what errors are occurring from the HPSS Movers.

5.5.2.3. The ghi_mon shows the SD restarted

Diagnosis 1: The Session node failed over

Resolution: View the central log to determine why the Scheduler was recycled. If the Scheduler Daemon terminates again, turn on additional logging, so that the next time the schedule is recycled, additional error information can be viewed.

5.5.2.4. Failed to connect to the SD

Diagnosis 1: The SD is not running because the file system is unmounted

Resolution: Mount the file system, and verify the SD is started.

5.5.3. Backup Problems

5.5.3.1. GHI backup cannot communicate with DB2

Diagnosis 1: DB2 is not running.

Resolution: Verify whether DB2 is running and restart as appropriate. Authenticate as the DB2 instance owner and start DB2 with the db2start command. There is no harm in executing this if the DB2 instance is already running.

5.5.3.2. Failed to backup a file from a snapshot

The output from a backup shows that GHI failed to backup a file's data from a snapshot.

Check the migration problem section to determine why the file failed to migrate.

5.5.3.3. Failed to backup namespace information

The output from a backup shows that GHI failed to backup a namespace file.

Diagnosis 1: This is caused bya failure transferring the file to HPSS.

Resolution: Refer to the section on IO Manager problems.

Appendix A - GLOSSARY OF TERMS AND ACRONYMS

ACL	Access Control List.
AIX	Advanced Interactive Executive. An operating system provided on many IBM machines.
Alarm	A log record message type used to log high-level error conditions.
ANSI	American National Standards Institute.
API	Application Program Interface.
Archive	One or more interconnected storage systems of the same architecture.
Attribute	When referring to a managed object, an attribute is one discrete piece of information, or set of related information, within that object.
Class of Service	A set of storage system characteristics used to group files with similar logical characteristics and performance requirements together. A Class of Service is supported by an underlying hierarchy of storage classes.
co-managed	File data resides in both GPFS and HPSS.
Configuration	The process of initializing or modifying various parameters affecting the behavior of an GHI server or infrastructure service.
COS	Class of Service.
Core Server	An HPSS server which manages the namespace and storage for an HPSS system. The Core Server manages the Name Space in which files are defined, the attributes of the files, and the storage media on which the files are stored. The Core Server is the central server of an HPSS system. Each storage sub-system uses exactly one Core Server.
Daemon	A UNIX program that runs continuously in the background.

DB2	A relational database system, a product of IBM Corporation, used by HPSS and GHI to store and manage HPSS adn GHI metadata.
Debug	A log record message type used to log lower-level error conditions.
Delog	The process of extraction, formatting, and outputting HPSS central log records.
Directory	An HPSS object that can contain files, symbolic links, hard links, and other directories.
Dismount	An operation in which a cartridge is either physically or logically removed from a device, rendering it unreadable and unwritable. In the case of tape cartridges, a dismount operation is a physical operation. In the case of a fixed disk unit, a dismount is a logical operation.
DMAPI	Data Management Application Programming Interface.
DNS	Domain Name Service.
DOE	Department of Energy.
Drive	A physical piece of hardware capable of reading and/or writing mounted cartridges. The terms device and drive are often used interchangeably.
DRP	Disaster/Recovery Plan.
ED	Event Daemon.
Event	A log record message type used to log informational messages (e.g., subsystem starting, subsystem terminating).
Export	An operation in which a cartridge and its associated storage space are removed from the HPSS system Physical Volume Library. It may or may not include an eject, which is the removal of the cartridge from its Physical Volume Repository.
File	An object than can be written to, read from, or both, with attributes including access permissions and type, as defined by POSIX (P1003.1-1990). HPSS supports only regular files.
file family	An attribute of an HPSS file that is used to group a set of files on a common set of tape virtual volumes.

fileset	A collection of related files that are organized into a single easily managed unit. A fileset is a disjoint directory tree that can be mounted in some other directory tree to make it accessible to users.
fileset ID	A 64-bit number that uniquely identifies a fileset.
fileset name	A name that uniquely identifies a fileset.
file system ID	A 32-bit number that uniquely identifies an aggregate.
FSID	File system unique identifier
GB	Gigabyte (2^{30}).
GPFS	General Parallel File System.
GHI	GPFS/HPSS Interface.
GHI-HTAR	Specially modified GHI-specific version of the HTAR program.
GSS	Generic Security Service.
Hierarchy	See Storage Hierarchy.
HPSS	High Performance Storage System.
HSI	Hierarchical Storage Interface.
HSIGWD	HSI Gateway Daemon.
HTAR	HPSS tar program – a utility to aggregate a set of files directly into HPSS without first writing to local storage, and to randomly retrieve individual member files via creation of a separate index file.
IBM	International Business Machines Corporation.
ID	Identifier.
I/O	Input/Output.
IOA	I/O Agent.
IOM	I/O Manager.

IP	Internet Protocol.
junction	A mount point for an HPSS fileset.
KB	Kilobyte (210).
LAN	Local Area Network.
LANL	Los Alamos National Laboratory.
LLNL	Lawrence Livermore National Laboratory.
MB	Megabyte.
MD	Mount Daemon.
metadata	Control information about the data stored under HPSS, such as location, access times, permissions, and storage policies. Most HPSS metafile contents are stored in a DB2 relational database.
migrate	To copy file data from a level in the file's hierarchy onto the next lower level in the hierarchy.
mount	An operation in which a cartridge is either physically or logically made readable and/or writable on a drive. In the case of tape cartridges, a mount operation is a physical operation. In the case of a fixed disk unit, a mount is a logical operation.
mount point	A place where a fileset is mounted in the XFS and/or HPSS namespaces.
Mover	An HPSS server that provides control of storage devices and data transfers within HPSS.
Name Service	The portion of the Core Server that providesa mapping between names and machine oriented identifiers. In addition, the Name Service performs access verification and provides the Portable Operating System Interface (POSIX).
name space	The set of name-object pairs managed by the HPSS Core Server.
NLS	National Language Support.
NSL	National Storage Laboratory.

Object	See Managed Object.
OSF	Open Software Foundation.
PB	Petabyte (2^{50}).
PM	Process Manager.
POSIX	Portable Operating System Interface (for computer environments).
RPC	Remote Procedure Call.
SCSI	Small Computer Systems Interface.
SNL	Sandia National Laboratories.
SSA	Serial Storage Architecture.
storage class	An HPSS object used to group storage media together to provide storage for HPSS data with specific characteristics. The characteristics are both physical and logical.
storage hierarchy	An ordered collection of storage classes. The hierarchy consists of a fixed number of storage levels numbered from level 1 to the number of levels in the hierarchy, with the maximum level being limited to 5 by HPSS. Each level is associated with a specific storage class. Migration and stage commands result in data being copied between different storage levels in the hierarchy. Each Class of Service has an associated hierarchy.
storage subsystem	A portion of the HPSS namespace that is managed by an independent Core Server and (optionally) Migration/Purge Server.
TB	Terabyte (2^{40}).
TCP/IP	Transmission Control Protocol/Internet Protocol.

Transaction	A programming construct that enables multiple data operations to possess the following properties: All operations commit or abort/roll-back together such that they form a single unit of work. All data modified as part of the same transaction are guaranteed to maintain a consistent state whether the transaction is aborted or committed. Data modified from one transaction are isolated from other transactions until the transaction is either committed or aborted. Once the transaction commits, all changes to data are guaranteed to be permanent.

Appendix B - REFERENCES

- ***HPSS Installation Guide***,
- ***HPSS Management Guide***
- ***HPSS User's Guide***
- ***HPSS Conversion Guide***
- ***GPFS Data Management API Guide***
- ***GPFS Administration and Programming Reference***
- ***GPFS Advanced Administration***
- ***HTAR:*** http://www.mgleicher.us/index.html/htar/htar_man_page.html,
- ***HSI:*** http://www.mgleicher.us/index.html/hsi,
- ***POSIX 1003.1-1990 Tar Standard***

Appendix C -TSM TO GHI CONVERSION

Prerequisites

The following table summarizes the pre-requisites for the deployment of the GHI 2.2.0 Patch3 TSM conversion feature:

Prerequisite	Description
GPFS 3 PTF 7	This is the level of GPFS tested for the TSM conversion feature.
HPSS 7.3.3 patch 1	See the HPSS documentation for HPSS 7.3.2 required patches
HSI/HTAR 3.5.7	Deploy the HSI Gateway Server on the HPSS Core Server. HTAR must be configured on all GHI IOM nodes.
GPFS Mirror Mount point	This is the target mount point where users are going to be putting/deleting files from
GPFS or NFS Mount point	This is the source TSM file system. **It must be available on all the GHI nodes as a remotely mounted GPFS file system. The mount point MUST be directly under "/".**

GHI and TSM compatibility

GHI and TSM can not run on the same cluster. The TSM file system must be remotely mounted on the destination GHI cluster.

Obtaining TSM Source Code

The TSM conversion source code was added to the GHI 2.2.0 patch 3.

Compiling GHI TSM conversion code

These instructions assume the HPSS and GHI code are placed into "/opt/hpss/src/" and "/opt/hpss/src/ghi."

To deploy the TSM code on a GHI cluster, follow these steps:

1. Compile the HPSS 7.3.3 patch 1 client source on:
   /opt/hpss

2. Copy the patch source code to:
   /opt/hpss/src/ghi

3. Open the following file for editing:
   /opt/hpss/src/ghi/Makefile.macros

4. Change the Makefile.macros SUPPORT_FOR_TSM setting as follows:

```
# Adds supports for TSM (on or off)
SUPPORT_FOR_TSM = on
```

5. Change to the following directory:
   /opt/hpss/src/ghi/tools/tsm_conversion

6. Open the following files for editing:
   /opt/hpss/src/ghi/tools/tsm_conversion/ghi_tsm_copy.c
   /opt/hpss/src/ghi/tools/tsm_conversion/ghi_tsm_build_fs.c
7.
8. The **ghi_tsm_copy.c** and **ghi_tsm_build_fs.c** program contains the settings for the TSM mount point and GHI mount points. Modify the source code as follows:

   a. The GHI source code is released with the **TSM_MNTPT** set to /ghi33. Change the **TSM_MNTPT** variable to be the TSM mount point of your system. The following is an example for a mount point on /ghi33
      #define TSM_MNTPT "/ghi33"
   b. The GHI source code is released with the **GHI_MNTPT** set to /ghi11. Change the **GHI_MNTPT** variable to be the GHI mount point of your system. The following is an example for a mount point on /ghi11
      #define GHI_MNTPT "/ghi11"

   Note: Recognizing that this is a non-ideal procedure for a customer product. A bug has been opened to request implementation of specifying the TSM parameters as a configuration option.

9. Change to the GHI source tree directory located at:
   /opt/hpss/src/ghi

10. Execute ***make*** to build the GHI source product

    make clean clobber; make 2>&1 | tee make.out

11. Copy the binaries to all other nodes of the GHI cluster from the compiling node. On this case, the ***ghi_iom*** program might fail to copy if it is running as a service. To accomplish this task delete the ***ghi_iom*** executable on the remote node prior to copying.
    a. ssh root@<remote_node> "rm –f /opt/hpss/bin/ghi_iom"
    b. scp /opt/hpss/bin/ <remote_node>:/opt/hpss/bin

Restarting GHI Services

Once GHI has been compiled to run on a TSM-enabled environment all the I/O Manager executables **must** to be restarted. To accomplish this task you can issue a ***kill -9 <ghi_iom process id>*** on each node.

In addition, GHI needs to be restarted. To restart GHI follow these steps:

1. On the command line, shutdown GPFS by issuing the following command:
   mmshutdown –a

2. Wait for all the GHI process to complete. You can find out the processes are still running by issuing the following command:
   ps –ef | grep ghi
3. Recycle I/O Managers as described at the beginning of this section. The script on the appendix can be used to assist on this task
4. Start GPFS by issuing the following command:
   mmstartup –a

The TSM file system

The TSM file system must be available locally on each of the GHI nodes. It can be a NFS mount that is available on all nodes or a GPFS remote mount point (different from the GFPS-TSM mirror). The mount point **must** be directly under the root path "/".

Building GPFS-TSM mirror file system

Before creating the GHI file system mirror of the TSM file system, start GPFS and GHI and make sure you have compiled and restarted GHI for the TSM mode as described on ***Compiling GHI TSM conversion*** code the ***Restarting GHI Services*** sections of this document.

Execute the following steps:

1. Build a copy of the TSM name space on the TSM managed cluster using the /var/hpss/ghi/policy/build_fs.policy. This must be run on the TSM cluster as GPFS does not support running policies on remotely mounted clusters.
   a. mmapplypolicy <file system> -P /var/hpss/ghi/policy/build_fs.policy –I defer
2. Run the conversion script
   /opt/hpss/bin/ghi_tsm_build <shell command> <file list>
   a. The shell command should be the method used to communicate between the GHI nodes. Either SSH or RSH
   b. The file list must be stored in a file system that is accessible from all the nodes in the cluster.

Backups and Restores of GPFS

There are no special procedures for backing up or restoring a GPFS file system that was built using a TSM conversion.

Upgrading GPFS

When GPFS is upgraded is require that you recompile the GHI source code to ensure it is compiled against the latest DMAPI libraries. The re-compiled code must be distributed to all the GHI nodes. After the GPFS upgrade follow the steps described in the ***Compiling GHI TSM conversion*** code and ***Restarting GHI Services*** sections of this document

Appendix D - DEVELOPER ACKNOWLEDGMENTS

The GHI feature of HPSS was developed by IBM Global Business Services – Federal. HPSS is jointly owned and developed by the HPSS Collaboration consisting of Lawrence

Berkeley National Laboratory, Lawrence Livermore National Laboratory, Los Alamos National Laboratory, Oak Ridge National Laboratory, Sandia National Laboratories, and IBM.

We would like to acknowledge IBM Almaden Research Center for the software development collaboration between GPFS and HPSS to implement the GPFS/HPSS Interface.

We would like to acknowledge NERSC, the National Energy Research Scientific Computing Center, for their help with initial design and development.

We would like to acknowledge Gleicher Enterprise, LLC, for providing a modified GHI-specific version of GHI-HTAR to support GHI file aggregation.

We would like to acknowledge SDSC, San Diego Supercomputing Center, for their help with the initial requirement review.

We would also like to acknowledge HLRS, High Performance Computing Center of the University of Stuttgart, and NCSA, National Center for Computing Applications, for providing testbeds for the initial GHI release.