IBM® DB2® for Linux®, UNIX®, and Windows®

Best Practices

DB2 databases and the IBM General Parallel File System™

Aslam Nomani
DB2 Quality Assurance Manager

Jeremy Brumer
DB2 Quality Assurance

Scott Fadden
IBM GPFS Technical Marketing

Yvonne Chan
DB2 pureScale Principal, IBM PureData EcoSystem

Last updated: **March 2013**

Executive summary

In today's highly competitive marketplace, it is important to deploy a data processing architecture that not only meets your immediate tactical needs, but that also provides the flexibility to grow and change to adapt to your future strategic requirements. To help reduce management costs, add flexibility, and simplify the storage management of your DB2® for Linux®, UNIX®, and Windows® installation, you need to choose a file system that is designed to provide a dynamic and scalable platform. The IBM® General Parallel File System™ (GPFS™) is a powerful platform on which to build this type of relational database architecture. This paper describes why GPFS is the right file system to use with DB2 databases by outlining the benefits and providing best practices for deploying GPFS with DB2 software. In addition, a section has been added to this paper to describe the DB2 pureScale feature, and how it configures and uses GPFS.

Introduction

The IBM General Parallel File System™ (GPFS™) file system is a high-performance shared disk file management solution that provides fast, reliable access to a common set of file data from two computers up to hundreds of systems. A GPFS file system integrates into your environment by bringing together mixed server and storage components to provide a common view of enterprise file data.

Pairing GPFS file systems and DB2 for Linux, UNIX, and Windows databases creates a powerful platform for your high-performance database systems. Using GPFS with your DB2 deployment offers you the following benefits:

- **Enhanced availability** - With the advanced replication capabilities of GPFS, you can mirror data within a single site or across multiple locations. The file system can be configured to remain available automatically in the event of a disk or server failure.

- **Flexibility through dynamic infrastructure** – By deploying your DB2 database on GPFS, you have the ability to easily deploy DB2 data to different machines without moving data across storage devices or modifying existing file systems. Instead of having to copy the data to another machine to allow data access, that machine can be added as part of the GPFS cluster allowing the data to be mounted and accessible from the new machine without requiring any data movement.

- **Continued high performance** – GPFS lets you expand your storage with automatic data rebalancing that is transparent to your applications. GPFS also delivers performance comparable to best of breed files systems like JFS2.

- **Improve operational efficiency** – The simplified storage administration provided by GPFS can help reduce your total cost of ownership. Leading edge file system technologies such as integrated policy based storage management can help automate many storage management tasks.

IBM has optimized DB2 9.7 and DB2 10.1 for use with GPFS file systems and has developed a set of best practices that you can apply when deploying DB2 on GPFS. These best practices help you achieve the best results for your DB2 databases that run on GPFS.

This paper describes the benefits of using GPFS with DB2 databases, including best practices for deploying DB2 databases on GPFS. These best practices are used by DB2 pureScale when the GPFS cluster is installed and configured for you.

The target audience for this paper is people responsible for deploying DB2 software and the systems on which DB2 resides. This paper assumes you have moderate knowledge of DB2 and UNIX system administration. This paper also assumes you are not using DB2 pureScale. With DB2 pureScale, the recommendation is to allow DB2 to create, configure and tune the GPFS cluster for you. By doing this, you can take full advantage of the integration between DB2 and GPFS. In addition to the items presented in this paper,

there are some other cluster integration points that are required for the fully automated failure recovery system to work properly. Those items have been left out of this paper as it is beyond the scope of the topic at hand.

The examples in this paper are based on DB2 V10 fix pack 2 and GPFS 3.5.0.4 efix13 installed on AIX® 6.1 TL6 SP5 but can be extended to more recent versions and other supported platforms. All versions of GPFS are supported with DB2 for Linux, UNIX, and Windows, however, the latest supported fix packs are recommended to ensure the best quality experience.

Further details on DB2 and GPFS software can be found on the following websites:

DB2 software

http://www.ibm.com/software/data/db2/

General Parallel File System

http://www.ibm.com/systems/gpfs

Why GPFS?

GPFS delivers reliable, high performance

It is important that the file system you choose for your database has very strong performance characteristics and a proven history of reliability. A GPFS file system performs like a local file system, but with the advantage of the flexibility and scalability of a clustered file system. In tests, IBM has shown that DB2 on GPFS delivers performance characteristics comparable to local file systems. In addition to high performance, you can dynamically add hosts and storage when using GPFS, so that you can keep pace with ever growing processing demands.

GPFS helps reduce your total cost of ownership

A GPFS file system is easy to deploy, easy to manage, improves hardware utilization, and can streamline your data workflow, all of which can improve operational efficiency. GPFS provides a standard file system interface that you can use for many different applications across your environment. Large file system support, storage pooling, and automated file management allows you to better utilize existing hardware. Rolling upgrades and support for mixed hardware and operating system means that you can quickly integrate new server and storage hardware to meet ever-changing demands and budgets.

A GPFS storage infrastructure can help you reduce data copy operations and data processing times as the data can be accessed directly from multiple machines, potentially resulting in lower storage costs and fewer administrators required per terabyte of storage managed.

GPFS provides leading edge technology

GPFS provides a global namespace which means that data can be accessed from multiple machines simultaneously while still maintaining data integrity. This allows you the flexibility of accessing the data from any machine in your environment, as if it were local. GPFS provides leading-edge technology compared to other file systems with features that give you greater ability to meet the demands of your business. Examples of these features include:

- High performance file system
    - **Direct I/O -** Direct input/output allows DB2 to use its own internal buffering and to bypass buffering within the file system. The ability for the DB2 database to leverage this capability of GPFS eliminates double buffering, which improves performance significantly.

- **Wide striping –** GPFS stripes all data over all disks in a storage pool. This allows for optimal performance from a single pool.

- **Process workflow optimization –** Your database storage can share a common namespace with application data. This allows for highly efficient workflows when data is processed in steps by multiple machines. There is no need to copy data between each processing step since all hosts see all data.

- Policy-based storage

  - **Storage pools** – You can create multiple storage pools to provide multiple levels of performance or reliability characteristics within a single file system.

  - **Logical administrative containers** – You can create filesets to provide a means to logically partition the file system namespace for ease of management.

  - **Rule-based file management** – You can use rule-based policies based on file attributes to automatically direct the initial file placement and process transparent file migrations.

GPFS offers advanced high availability features

Continuous data availability is imperative in today's highly competitive environment where data is a key asset. GPFS provides numerous features that are designed to allow you to achieve your availability goals:

- Data consistency

  - **Disk leasing protocol** - Only active hosts can get a disk lease to read and write to a disk. If the host fails, another host in the cluster waits for the granted disk lease to expire, and then performs recovery for the affected host. This protocol prevents a failed host from changing data, allowing for data integrity while still providing ultra-fast recovery from failures.

  - **Distributed lock management** – There is no need to worry about concurrent file access from multiple nodes; GPFS has a proven distributed lock management solution capable of handling thousands of concurrent accesses to a single file system or file.

- Snapshots and data replication

  - **File system snapshots** – Snapshots allow you to make a point–in-time copy of the GPFS file system. This allows for exceptionally fast deployments of point-in-time copies of the data while using a minimal amount of disk space.

- **Data replication** – In GPFS, data replication is software-based with synchronous mirroring across multiple storage arrays within a site or across the WAN. This replication provides an additional level of redundancy to meet availability requirements of mission critical data.

- **Backup and disaster recovery** - Snapshots, replication, and integration with IBM Tivoli® Storage Manager can be used for backup and disaster recovery purposes. Tivoli Storage Manager can be leveraged for integrated backups of the GPFS file system to further enhance a data protection strategy.

- High availability design

  - **Fast failover** – GPFS support for SCSI-3 Persistent Reserve reduces the time required for data operations to continue in the event of a host failure. In the event of a host failure, a surviving host suspends the writing ability of the failed host as soon as the failure is detected. By evicting the failed host immediately, normal processing can continue quicker on the surviving hosts.

  - **Integrated cluster management** - Cluster management features are integrated into the GPFS software. There is no need for separate clustering software to manage the hosts in the cluster.

  - **Cluster quorum** – With GPFS, both majority node set and tie-breaker disk quorum options are available. Quorum ensures data consistency by providing a guaranteed way of determining when a host has the ability to read from or write to the file system.

Usage models for DB2 and GPFS

There are multiple ways that GPFS can enhance a DB2 database environment. Here are just a few.

Dynamic system growth

Let's say you start your deployment with a single host DB2 installation. You quickly realize that you are going to need more processing power to support your application and have to buy a new server. Using GPFS, you can expand your file system to the new machine without a database outage. You can then configure the DB2 instance on the new machine to use the database stored on GPFS, stop the DB2 instance on the old machine, and then start the DB2 instance on the new machine. Finally, you can now point your DB2 clients to the new machine or leverage automatic client rerouting technology.

Process workflow optimization

Your organization relies on data analytics to keep you ahead of the competition. You want to reduce the time it takes to process the data and provide these vital statistics.

Assume you have a transaction processing system that is collecting customer transactions and a separate environment that is doing extract, transform, and load (ETL) processing. When this ETL processing is done, the data is loaded into your DB2 data warehouse. Using GPFS with DB2 allows you to tie this processing together, eliminating the need to copy the data between systems for processing.

Best practices for deploying GPFS

These basic recommendations are based on extensive testing of DB2 databases with GPFS file systems. These tips are intended to help you achieve maximum benefits in terms of features and performance.

GPFS cluster design

- **Exploit quorum design** - With up to an eight node cluster, define all nodes as quorum nodes and use tiebreaker disks. With more than eight nodes, use node quorum (no tiebreaker disks).

- **Ensure communication bandwidth is sufficient for multisite operations** - You can deploy a cluster within a single site or across multiple locations. If you need to replicate data across sites, the communication bandwidth and latency between sites needs to be sufficient to support your application response requirements.

Storage configuration

- **Quorum with tiebreaker disks** - When defining tiebreaker disks you can utilize any of the network shared disks (NSD) in the file system.

- **Storage RAID level** - Choosing a RAID level is typically a decision made based on capacity versus availability. Raid 5 and Raid 1+0 are typically used when deploying a DB2 database.

- **Select storage that supports SCSI-3 PR** - For optimum failover performance you should use a storage server that supports the SCSI-3 protocol Persistent Reserve feature. GPFS uses Persistent Reserve to provide fast cluster failover in the event of a node failure. Enable this feature in GPFS using the **mmchconfig** command with the **usePersistentReserve=yes** option.

Configuring the GPFS file systems

- **Where performance and reliability is a concern, use two GPFS file systems** - Create two GPFS file systems, one for database data and the other for the database logs due to the widely different access patterns between data and log access.

- **Where simplicity of administration is the priority, use a single GPFS file system** - Using a single GPFS file system for data minimizes the administrative overhead. A single file system combined with DB2 automatic storage allows you to dynamically add storage capacity and even additional hosts without having to create another file system.

- **Use a block size of 1 MB** - Use a GPFS file system block size of 1 MB. Larger block sizes use the pagepool more efficiently.

GPFS tuning parameters

- **Use direct I/O (DIO)** - DB2 version 9.7 and DB2 V10 can directly exploit direct I/O in GPFS. As of version 9.7, all DB2 files, including the active log files, are opened with DIO. This allows for faster access to the disks and avoids double buffering of data in DB2 and the file system cache. This way, DB2 chooses what files need to be accessed and what data should be placed in cache. Allowing DB2 to open the files using DIO is preferred over mounting the entire file system in DIO mode.

- **Enable higher concurrency** - We suggest that you increase the value of the GPFS **worker1Threads** parameter from the default of 40 to 256. This parameter controls the maximum number of concurrent file operations on a GPFS file system. The sum of the **prefetchThreads** parameter value and the **worker1Threads** parameter value must be less than 550.

- **Increase File System Cache** - The GPFS pagepool is used to cache user data, file system metadata and other internal data. We suggest that you increase the size of the pagepool by changing the value of the **pagepool** parameter from the default of 64 MB to at least 256 MB. The pagepool is pinned in memory, so make sure that there is enough of free memory available on the system, otherwise this change could result in system paging and negatively effect overall performance.

Changing GPFS configuration parameters

Use the **mmchconfig** command to change these GPFS configuration parameters. For example, to change the size of the pagepool to 256MB by setting the **pagepool** parameter, issue the following command:

```
mmchconfig pagepool=256m
```

You can use the **mmlsconfig** command to view the current values. The **mmlsconfig** output shows only parameters changed from their default.

For these changes to take effect, you must first unmount the GPFS file systems and then restart the GPFS daemon.

Sample configuration example

What follows is a sample set of instructions on how to deploy GPFS following the best practices outlined within this paper. All of the GPFS commands shown are in the /user/lpp/mmfs/bin directory.

1. Create the GPFS Cluster on machine *db2server*:
   ```
   mmcrcluster –A –N db2server:manager-quorum –p db2server
   ```

2. Start the GPFS Cluster:
   ```
   mmstartup –a
   ```

3. Create a file called nsd_mappng that contains a list of disks (LUNS) that will be used to create the GPFS file systems. The sample below uses six disk partitions. The colons separating blank columns indicate that the default value be used. (For more information on the file format, see the documentation for the **mmcrnsd** command in the GPFS documentation) The choice of network shared disk (NSD) name is up to the users.

   ```
   /dev/hdisk1:::::Data1:
   /dev/hdisk2:::::Data2:
   /dev/hdisk3:::::Data3:
   /dev/hdisk4:::::Data4:
   /dev/hdisk10:::::DBLog1:
   /dev/hdisk11:::::DBLog2:
   ```

4. Create the GPFS network shared disks using the nsd_mapping file you just created:

   ```
   mmcrnsd –F nsd_mapping
   ```

5. Stop the GPFS cluster and specify tiebreaker disks:

   ```
   mmshutdown –a
   mmchconfig tiebreakerDisks=”Data1;Data2;Data3”
   ```

6. Set the GPFS configuration parameters according to DB2 on GPFS best practices:

   ```
   mmchconfig worker1Threads=256
   mmchconfig pagepool=256M
   mmchconfig usePersistentReserve=yes
   ```

7. Start the GPFS Cluster:

```
mmstartup -a
```

8. Create the GPFS file systems for the data and logs with a one megabyte block size

```
mmcrfs –T /db2data db2data “Data1;Data2;Data3;Data4” –B
1M
mmcrfs –T /db2log db2log “DBLog1;DBLog2” –B 1M
```

   Ensure that the mount points /db2data and /db2log are created.

9. Mount the GPFS file systems

```
mmmount db2data -a
mmmount db2log -a
```

DB2 pureScale and GPFS

The DB2 pureScale feature provides clustering technology that helps deliver high availability and scalability in an application transparent way. It contains both a cluster manager and a cluster file system. The cluster file system used by DB2 pureScale is GPFS. As part of the setup for DB2 pureScale, a GPFS cluster is created, along with the initial file systems used by DB2. The cluster is optimized for fast failure detections and the file systems are optimized for DB2 data and log access. Command line tools are also provided to create more file systems using an optimal configuration for DB2. These are the same optimizations provided in this paper. In addition to the optimizations, the GPFS binary code is also updated and maintained alongside the DB2 binaries. When a DB2 pureScale fix pack is installed, any necessary GPFS fixes are also applied at the same time.

Conclusion

Flexible storage management is a key component of the next-generation dynamic data center. The next-generation dynamic data center provides cost efficiency though optimal hardware utilization and reduced administration costs, and the ability to quickly deploy data processing and storage infrastructure where and when required. Combining GPFS advanced storage management with the scale-out architecture of DB2 for Linux, UNIX, and Windows provides an industry leading solution well suited to the next-generation dynamic data center.

Appendix A – Moving data to GPFS

There are many methods of moving data between file systems. This section identifies some mechanisms of moving your DB2 data from a given file system to GPFS. You can move existing data to the new GPFS file systems using many methods including:

- Database backup and restore
- Table space rebalance across file systems in conjunction with the db2relocatedb tool
- Operating system **copy** or cp command
- DB2 High Availability Disaster and Recovery (HADR) feature

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Contacting IBM

To provide feedback about this paper, write to db2docs@ca.ibm.com

To contact IBM in your country or region, check the IBM Directory of Worldwide Contacts at http://www.ibm.com/planetwide

To learn more about IBM Information Management products, go to http://www.ibm.com/software/data/