Microsoft SQL Server
Configuration Guide for HP IO Accelerators

Part Number 647096-001
December 2010 (First Edition)

Intended audience

This document is for the person who installs, administers, and troubleshoots servers and storage systems. HP assumes you are qualified in the servicing of computer equipment and trained in recognizing hazards in products with hazardous energy levels.

Contents

Installing and preparing the IO Accelerator for Microsoft Windows SQL Server

Initial setup

After the IO Accelerator is installed, follow these steps to prepare it for use with Microsoft® SQL Server:

1. Start the HP IO Accelerator Management Tool.
2. In the Device Tree, select the **IO Accelerator**.
3. Click **Format Low-level**.
4. In the dialog box, select **Advertised Capacity**, and select a 4K block size.
5. Select **Disk Management in Windows**.
6. If you are using a RAID configuration on multiple IO Accelerators, convert the drives to **Dynamic**.
7. Format the drive for a filesystem, setting the **File Allocation Unit Size** to 64K.

Installing Microsoft SQL Server

NOTE: When you install Microsoft® SQL Server, install the executables, usually found in the **Program Files** folder, to your local drive rather than to the IO Accelerator drive.

Install the following system databases on the IO Accelerator:

- Master
- Tempdb
- MSDB
- Model

If heavy paging occurs, install the page files on the IO Accelerator, which provides a significant increase in performance (when DRAM is available).

For more information, see the latest Windows® user guide.

Additional performance tips

- Have as many user database data files as there are processor cores in the server. This ratio enables multiple threads to operate at the same time and significantly improves the IO Accelerator performance.
- Keep all the data files the same size.
- Allow the database to use only one log file.

RAID options

RAID 0

RAID 0 offers the full capacity of the IO Accelerator without any disk-level redundancy.

NOTE: Windows® Software RAID configurations with IO Accelerators have not demonstrated performance deterioration. Due to the high performance potential of the IO Accelerators, physical RAID controllers are discouraged since they might become a performance bottleneck.

RAID 1

RAID 1 offers half the raw capacity with disk-level redundancy.

RAID 5

HP does not recommend configuring IO Accelerators in RAID 5 because of the parity calculations required to manage the RAID 5 array.

RAID 10

You can use Symantec Storage Foundation to configure RAID 10 with IO Accelerators.

- You must have a minimum of four IO Accelerators.
- For more details, request the *Symantec Storage Foundation Configuration Guide* from your HP representative.

High availability considerations

Database mirroring

NOTE: Database mirroring requires Windows® SQL Server 2005 or 2008.

To mirror the servers on the same local network, HP recommends synchronous mirroring for best data loss protection. If the servers are on different networks, HP recommends asynchronous mirroring to reduce the performance impact on the primary server.

NOTE: The performance impact on the primary server is not due to any IO Accelerator inadequacy. In a synchronous mirroring, every transaction must be committed first on the mirror server before it is committed on the primary server. This requirement can add high latency if the primary and the mirror servers are separated by a non-dedicated public Internet.

Database mirroring occurs at the database level so you can choose which databases to mirror. System databases cannot be mirrored, so you must create scripts using SSIS to automate the syncing of these databases.

With SQL Server 2008, the transaction log stream is automatically compressed before being sent to the mirror server. Database mirroring also automatically fixes any corrupted pages without user intervention.

You can use a witness server to keep a heartbeat and create an automatic failover scenario. You can even use SQL Server Express (free version) to act as the witness server.

If an environment consists entirely of a .net codebase, then add the following command to the connection string to make the failover completely automatic:

```
Failover Partner=myMirrorServerAddress;
```

When the failover occurs, the clients (Web servers, application servers, and so on) automatically re-point to the mirror server.

If an environment consists of different Web servers, application servers, and so on, that cannot take advantage of the connection string feature, a script can be used to push alternate versions of host files and re-map the IP address to name mapping. This script makes the mirror server the primary server in the event of a failover.

The enterprise edition of SQL Server 2005/2008 enables you to create a read-only snapshot on the mirror server to be used as a standalone reporting server in addition to being a failover server. This creates an efficient SQL Server infrastructure.

Multi-site clustering

With Windows® 2008, Microsoft® offers multi-site clustering that does not require shared storage. As long as the operating system is Windows® 2008, the SQL Server version can be 2005 or 2008. This feature provides an automatic and seamless failover framework. Clients talk to a virtual IP address. When one node goes down, another one starts.

Third-party software is required to keep the data in sync between the different cluster nodes. HP recommends the following:

- DataKeeper by Steel Eye (http://www.steeleye.com/DataKeeper_Cluster_Edition_144.htm)
- *Clustering For Mere Mortals, Step-by-Step: Configuring a 2-node multi-site cluster on Windows Server 2008 R2 – Part 1* (*http://clusteringformeremortals.com/2009/09/15/step-by-step-configuring-a-2-node-multi-site-cluster-on-windows-server-2008-r2—part-1/*)
- Double Take Availability (http://www.doubletake.com/english/products/double-take-availability/Pages/default.aspx)
- Neverfail (http://www.neverfailgroup.com/windows-apps/sql-server-high-availability.html)

The different cluster nodes can be in the same or separate locations.

Backup considerations

General recommendations

Backing up large SQL Server databases often takes a long time and ends up affecting the performance of the primary systems. HP recommends the following for performance improvement:

- If you are using SQL Server 2008, use the built-in feature to compress backups.
- If you are using SQL Server 2005, various third-party tools are available to compress backups:
  - LiteSpeed (http://www.quest.com/litespeed-for-sql-server/)
  - SQL Backup (http://www.red-gate.com/products/SQL_Backup/index.htm)
  - SQL Safe Backup (http://www.idera.com/Products/SQL-Server/SQL-safe-backup/)
- Using compression for backups has the following advantages, especially when using IO Accelerators:
  - Significantly reduces completion time
  - Significantly reduces drive capacity needs
  - Blocks databases for much smaller periods of time
- To realize the full performance benefit of the IO Accelerators for backup operations, load the databases on IO Accelerators and, at the same time, create the backup file on the IO Accelerators. This feature accelerates both read and write operations.

Testing IO Accelerator-powered SQL performance

Methodology

Generally, benchmarks such as SQLIO and ioMeter are used to measure the I/O profile of the storage sub-systems. However, with IO Accelerators, these tests often give false results since the tests were designed for measuring spinning disk media and do not completely provide the performance of the IO Accelerators.

Workload examples

HP recommends that you use a realistic a load when testing the IO Accelerator-powered SQL Server. For example:

1. Restore a copy of the production database onto the test IO Accelerator server.
2. Create a profiler trace on the production server for approximately one hour.
3. Replay this trace on the test server.

For the baseline from production and the baseline from test, perform the following steps:

1. Calculate the average duration for SQL Statements completed using the SQL Profiler.
2. Observe the following counters using PERFMON:
   - Processor ——% Processor Time
   - Paging file—% Usage [Page file Activity]
   - SQLServer:Databases—Data File(s) size (kb) [Database Activity for tempdb]
   - SQLServer:Databases—Log File(s) size (kb) [Database Activity for tempdb]
   - Physical Disk—Avg. Disk sec/Read [Read Latency]
   - Physical Disk—Avg. Disk sec/Write [Write Latency]
   - Physical Disk—Disk Reads/sec [Read IOPS]
   - Physical Disk—Disk Writes/sec [Write IOPS]
   - Physical Disk—Disk Read Bytes/sec [Read throughput]
   - Physical Disk—Disk Write Bytes/sec [Write throughput]
   - Physical Disk—% Disk Read Time
   - Physical Disk—% Disk Write Time
   - Physical Disk—% Disk Time
   - Physical Disk—% Idle Time
   - Physical Disk—Average Disk Read Queue Length
   - Physical Disk—Average Disk Write Queue Length

- Physical Disk—Average Disk queue Length
- Logical Disk—Avg. Disk sec/Read [Read Latency]
- Logical Disk—Avg. Disk sec/Write [Write Latency]
- Logical Disk—Disk Reads/sec [Read IOPS]
- Logical Disk—Disk Writes/sec [Write IOPS]
- Logical Disk—Disk Read Bytes/sec [Read throughput]
- Logical Disk—Disk Write Bytes/sec [Write throughput]
- Logical Disk—% Disk Read Time
- Logical Disk—% Disk Write Time
- Logical Disk—% Disk Time
- Logical Disk—% Idle Time
- Logical Disk—Average Disk Read Queue Length
- Logical Disk—Average Disk Write Queue Length
- Logical Disk—Average Disk queue Length

These data points provide information on how your SQL Server is performing.

If you cannot restore your entire database, you can evaluate performance by creating a sample database and doing the following:

- Conduct index maintenance.
- Conduct DBCC maintenance.
- Conduct full backups.
- Conduct full restores.
- Monitor the PERFMON counters previously stated while these operations occur. The counters provide a profile of your SQL Server performance.

Architectural scenarios

Scenario 1: The entire database fits on the IO Accelerators

1. Place the data files on one IO Accelerator.

   NOTE: You can use one IO Accelerator or a RAID 0 set of multiple IO Accelerators.

2. Place the `tempdb` file for both data files and log files, the log files, and backup files on another IO Accelerator.
3. If available, give the `tempdb` file its own dedicated IO Accelerator.

Scenario 2: The database is too big to fit on the IO Accelerators

1. If the database is too large to fit the entire database on the IO Accelerators, then place specific database elements that use the most I/O resources on the IO Accelerators.
2. Query the `sysprocesses` system table. If the `waitresource` file appears as `2:1:1` (PFS Page) or `2:1:3` (SGAM Page), then you can successfully move the `tempdb` file to the IO Accelerator.

tempdb considerations

The following SQL operations require high performance from the `tempdb` file:

- Repeated create and drop of temporary tables (local or global).
- Table variables that use the `tempdb` file for storage purposes.
- Work tables associated with CURSORS.
- Work tables associated with an ORDER BY clause.
- Work tables associated with an GROUP BY clause.
- Work files associated with HASH PLANS.

For optimal `tempdb` file performance, use these guidelines:

- Dedicate a separate IO Accelerator to the `tempdb` file.
- Create multiple `tempdb` data files. The number of files must be equal to the CPU cores (not hyperthreaded). For example, two Quad Core CPUs must use eight data files, even if the CPUs are hyperthreaded.
- You must have only one `tempdb` log file.

- Determine the maximum size of the `tempdb` file you need and then create each `tempdb` data file to be the same size.
- Turn off auto-grow on the `tempdb` file.

For more detailed information, see the Microsoft® website (http://support.microsoft.com/default.aspx?scid=kb;en-us;328551).

Index considerations

Indexes help retrieve data from the tables, so consider moving all indexes to a separate file group and then placing the file group on an IO Accelerator. For more information, see the following article on creating file groups for indexes (http://deepakrangarajan.blogspot.com/2008/12/moving-index-to-seperate-filegroup.html).

HP recommends that you delete indexes that are not being used. They cause unnecessary I/O overhead. For more information, see the following article on detecting which indexes are being used by the database (http://blogs.msdn.com/b/sqlserverstorageengine/archive/2007/04/20/how-can-you-tell-if-an-index-is-being-used.aspx).

Frequently accessed tables

Placing frequently accessed tables in individual file groups and then placing the file groups on your IO Accelerator can greatly improve performance. To identify the most frequently accessed tables:

1. Run a profile trace.
2. Review the stored procedures and statements that run most frequently.

Optimize the tables accessed by these stored procedures and statements.

Large tables

Large tables create an I/O bottleneck for most activities. Index maintenance and reads on large tables can be time-consuming. To relieve stress, move large tables to a separate file group and then place the file group on IO Accelerators.

For a method of listing all the tables and sizes, see the following article (http://searchsqlserver.techtarget.com/tip/Find-size-of-SQL-Server-tables-and-other-objects-with-stored-procedure).

Log files

Log files consume a lot of write performance. Every committed transaction results in date being written to the log file. IO Accelerators improve sequential write performance compared to traditional media, so putting log files on IO Accelerators improves the speeds of committing data transactions.

Partition tables

Microsoft® added partition tables to SQL Server 2005 and enhanced them in SQL Server 2008. This feature enables you to split the data in one large table into many small tables based on certain criteria. For example, you can split order data, based on data ranges. For example, you can devote one table to

data for each year. In this scenario, you can keep the most recent data on IO Accelerators and keep older data on more traditional media.

Acronyms and abbreviations

CPU

central processing unit

DBCC

database console commands

IOPS

input/output operations per second

IP

Internet Protocol

SQL

structured query language

SSIS

SQL Server Integration Services