Technical report:

Exchange Server 2003 on IBM System Storage N series

Jetstress best practices

Document NS3523-0

October 29, 2007

Table of contents

Abstract

This report establishes best-practices guidelines for testing and validating IBM System Storage N series with Microsoft Exchange Server workloads. It provides detailed and easy to follow steps on how to use Jetstress—a Microsoft tool—to validate storage system configurations using customer requirements, the Microsoft ESRP (Exchange Solution Reviewed Program) guidelines, and the IBM System Storage with N series Exchange Sizer. The intended audience includes IBM N series field personnel, engineers, and professional services representatives. This document is intended for internal use only.

Introduction

The Microsoft® Exchange Solution Reviewed Program (ESRP) – Storage was developed by Microsoft to help users validate Exchange storage system solutions prior to deployment into production Exchange messaging environments. ESRP uses a Microsoft tool – Jetstress – to exercise storage systems by simulating a configurable set of Exchange Server 2003 workload profiles. By following ESRP guidelines, users can quickly verify that the performance and reliability of proposed storage systems meet or exceed projected messaging workload requirements.

Background

Exchange Server 2003 can be a very I/O intensive application. Customers making a potentially large storage investment often use Jetstress to verify the storage systems they are going to purchase will support their unique Exchange messaging environment.

When correctly configured, Jetstress can showcase IBM® System Storage™ N series and help validate adequate levels of performance and stability for a production Exchange deployment. On the other hand, if incorrectly configured, Jetstress can produce misleading results.

To size and layout the storage environment correctly for use with Jetstress, the IBM System Storage N series with Exchange Sizer must be used. Furthermore, in order to properly configure and run Jetstress the ESRP methodology must be strictly followed.

This document is aimed at addressing, and ideally preventing, the vast majority of issues associated with mis-configured Jetstress test runs.

Purpose and scope

This report is composed with the following goals in mind:

- Outline the processes for conducting a Jetstress test on IBM N series storage systems
- Establish a set of Jetstress best practices for validating IBM N series storage systems
- Transfer expertise about Jetstress, ESRP and the Exchange Sizer to field personnel
- Provide guidance for customers in setting expectations for a Jetstress Proof of Concept (POC).

This report focuses on the important fundamentals of Jetstress, ESRP, and the Microsoft Exchange Sizer. It uses a specific configuration to demonstrate a set of processes that include:

- Pretest planning
- Test environment setup
- Configuring and running Jetstress

- Interpreting test results
- Troubleshooting.

This report does not cover all possible configurations or explain all the details of Jetstress, ESRP, or the Exchange Sizer. Readers are encouraged to review the appropriate reference documents to gain additional contextual knowledge about these tools and processes. It is assumed that readers are familiar with Exchange Server 2003 and related best practices, and are knowledgeable about IBM N series storage systems.

Jetstress overview

Jetstress is a tool developed by Microsoft to help verify the performance and stability of the disk subsystems used in Exchange Server environments. The latest Jetstress package can be downloaded from the Microsoft website.

In the download package, there are two flavors of Jetstress executables:

- JetstressUI.exe (also called Jetstress 2004) – the graphical user interface (GUI) version of the tool. The version number used in this report was: 06.05.7830.0.
- Jetstress.exe – the command-line version of the tool.

The use of JetstressUI.exe is recommended, primarily because of its simplicity compared to its command-line counterpart. Throughout this paper, the terms Jetstress and JetstressUI.exe are used interchangeably.

Also included in the package is Jetstress.doc. Reading this document is recommended before attempting a Jetstress test.

Three types of tests can be performed using Jetstress:

- **Performance** – The test is designed to verify the performance of the storage systems. It can be configured to run for 2 – 24 hours. By default, it runs for 2 hours.
- **Stress** – The test is designed to validate the reliability of the storage systems over a relatively long period. The default test runs for 24 hours.
- **Streaming Backup** – This test measures the storage systems' performance of streaming backup.

The Jetstress Stress and Performance tests work in three discrete phases: In the first phase, Exchange databases and mailboxes are created and initialized based on a set of user configurable parameters. The next phase that runs is either a Performance or a Stress test for a specified duration. The Performance and Stress tests run a workload that effectively simulates the I/O patterns observed in a real world Exchange environment during peak usage. The Stress and Performance tests generate mostly small block random reads and writes. The third and final phase is a process called Checksum verification which always runs at the completion of a Stress or Performance test. The Checksum verification works by reading database and log files in a sequential manner to validate them for consistency. In contrast to the random small block I/O patterns seen in Stress and Performance tests, the Checksum process uses 100% large block sequential I/O. In multiple server environments, the Checksum and Stress (or Performance) tests should not be run concurrently.

Based on field feedback that Performance testing is both most frequently used and that which has caused the most issues, this report will focus exclusively on the Jetstress Performance test.

Most IBM N series Exchange customers use IBM System Storage N Series with SnapManager® for Microsoft Exchange to backup and recover their Exchange environments. As such, the Streaming Backup test will not be covered in this report. Please refer to the Jetstress.doc for more details if required.

Aside from the three tests mentioned above, Jetstress has a number of other parameters with a broader range of options. If these parameters are incorrectly set the test may produce unusable or inaccurate results. It is essential to remember that the primary purpose of Jetstress tests is to validate the design of an Exchange storage solution ***based on customer requirements****, and* ***not to see how to saturate or break a storage system***, nor to see how fast a storage system can run. The next section will provide important guidelines for setting these Jetstress parameters properly based on the Microsoft ESRP methodology. To ensure success in your customer environment, these guidelines should be followed strictly.

ESRP methodology

ESRP – Storage is an Exchange Server program designed to facilitate third-party storage testing and solution publishing for Exchange Server. The program combines a storage testing harness (Jetstress) along with solution publishing guidelines. To date (at the time of this writing), certain models of the IBM System Storage N series (i.e., N5500) have passed the ESRP testing process.

ESRP test cases are designed to characterize the storage system's performance and reliability using Jetstress. The Performance test is geared at measuring I/O load (generated by using JET[1] transactional I/Os) the storage system can handle within a defined range of acceptable latency thresholds.

Jetstress installation

Jetstress can be installed on a server from the downloaded package. For Jetstress to run correctly, make sure .NET Framework v1.1 is installed. The following files must be copied to the Jetstress install location from an Exchange 2003 SP1 (or SP2) setup CD or from the Exchange website:

- Ese.dll
- Eseperf.dll
- Eseperf.hxx
- Eseperf.ini

Note that all these files must be from the same Exchange installation source. If for example, ese.dll is from SP1 and eseperf.dll is from RTM, then Jetstress will fail.

Jetstress input parameters

This section explains important input parameters along with corresponding best practices. For details about how to set these and other Jetstress parameters please review Section 6.

[1] The JET database refers to the underlying database engine used by Exchange. For more information about JET please see the Microsoft website.

PARAMETERS	DESCRIPTION	BEST PRACTICES
Number of storage groups	(Valid: 1-4). Each storage group has up to 5 database paths and one log path.	For enterprise customers, it is recommended to use 4 storage groups.
Number of disk drives per storage group	(Valid: 1-5). The drives refer to the logical volumes formatted to store the database files.	In the context of Jetstress, this parameter means that each storage group can have 1 – 5 logical volumes (LUNs) to store the databases. For simplicity, we recommend using 1 LUN for all databases within a storage group. So this value should be 1.
Database & log path	Specify the paths for mailbox stores and log paths. The available database paths are defined by the number of disk drives per storage group. The log path is one per storage group.	Given that we use 1 LUN for all databases within a storage group there should also be one database path and one log path. Note, databases and log files should always reside on different LUNs.
Database Operation Mix	The operation mix such as insert, replace and delete are parameters to change the database read, write ratio, and also log write ratio as compared to database write. The lazy commit is to modify the log write size.	As per ESRP guidelines, you should check the ***Lock Database Operation Mix*. IMPORTANT:** You will need to set the *Lock Database Operation Mix* parameter before each test run, as this setting does not persist from run to run.
Log Buffers	(Default: 9000). Exchange Log I/O's are written to the log buffers first, and then the buffer is cleared by either a non-lazy commit or a capacity commit. A non-lazy commit means that the log buffer is written to the disk immediately. A capacity commit means that the log buffer flushes to the disk when it is full. Increasing the log buffer size reduces the frequency of capacity flushes, increases the log write size, and subsequently reduces the overall log write latency.	Use the default number.
Threads	The threads directly control the amount of load generated by the server.	This will vary depending on a variety of factors discussed in depth in subsequent sections. Not calculating the optimal number of threads may result in test failure.
Mailbox size limit (MB)	The mailbox size per user.	Use customer input to assess mailbox size.
Number of mailboxes	Number of users supported on the server.	As per ESRP guidelines, you should not configure more than 4000 users hosted on a single Jetstress server.
Estimated IOPS per mailbox	• Heavy Knowledge Worker Profile= 1 IOPS • Average Knowledge Worker Profile= .5 IOPS • Light Knowledge Worker Profile= .2 IOPS	Use customer input to assess IOPS per mailbox.

Table 1. Important Jetstress input parameters.

Jetstress outputs

The Jetstress Performance test will generate the following outputs:

- A performance html report file contains the disk performance results as well as success or failure status of the test
- A database checksum html report file contains the status during the checksum phase of the test
- Both application and system event logs generated during the test
- A separate perfmon log for:
  - The performance phase of the test
  - The database checksum process
  - The log checksum

- A history of the current test in the status pane of the Jetstress GUI.

These outputs, together with outputs generated by using other tools (notably, perfstat—more on perfstat later in this report) should be carefully analyzed to interpret and validate the test results.

Criteria for passing a Jetstress test

For a Jetstress Performance test to pass, all of the following criteria must be met. If more than one server is used in the test, all servers must meet these criteria.

- The Performance html file must report "The test has been successful"
- Achieved IOPS must meet or exceed the Expected IOPS
- Both average read and write latencies must less than or equal to 20 ms
- Database Page Fault Stalls/sec must be 0
- No errors in any of the resultant html files
- No errors in the status pane of the Jetstress GUI
- No errors in any of the resultant event logs

Requirements/process to conduct a Jetstress POC

A Jetstress POC is often useful to help enterprise customers and prospects evaluate IBM N series storage systems. This section outlines the high level processes required to ensure a successful Jetstress POC.

Gathering requirements

The first step is to gather accurate requirements from the customer. In the requirement gathering phase, the following information is collected from the customer:

- Number of mailboxes – this number more than anything will decide the scale of the test and/or deployment.
- Mailbox sizes – this number is important in determining the required capacity of the storage system.
- User profiles – this is an important piece of information needed to estimate the target IOPS, and therefore, the number of spindles required in the storage system. Another question to ask is what percentage of users access email via wireless devices. Some wireless devices tend to generate substantially more I/O loads.
- Estimated growth rate – this is useful to help estimate any needed headroom for both IOPS and capacity.
- Deleted item retention – this will have an impact on the capacity of the storage system.
- Special requirements on performance, i.e. read/write latency – By ESRP standards, the average read and write latency should be no greater than 20 milliseconds. However, if the customer's IT organization has a more stringent requirement; you will need to know about it and size accordingly before any testing commences.
- Other special requirements or preferences – For example, protection levels: IBM System Storage N Series with RAID-DP™ (random array of inexpensive disks-double parity) or RAID4; protocols: fibre channel protocol (FCP) or internet small computing system interfact (iSCSI) protocol; hard drive types: serial ATA (SATA) or FC drive, 10K or 15K revolutions per minute (RPM), etc.

Using the Exchange Sizer

The Exchange Sizer was developed to simplify the sizing and storage layout of IBM N series storage systems used for Exchange deployments. The primary purpose of the Exchange sizer 3.5 is to provide accurate sizings and guide the Exchange storage group layout planning phase. It is essential to use the Exchange Sizer during the planning phase of a Jetstress POC, after you have gathered requirements from the customer. The Exchange Sizer helps you "right size" your customer environment while reducing the possibility of "under configuring" or "over configuring" the storage environment. It is also highly recommended that you also consult with your area certified systems engineer (CSE) when you generate the sizing and layout reports. You will likely need to run more than one sizing and layout iteration and obtain input from your regional CSE before the best configuration for your customer is determined.

In short, the sizing and layout for a Jetstress POC must follow Exchange Server 2003 best practices:

- Place storage group database files and transaction logs into separate aggregates. This provides flexibility for recovery in the unlikely event of the loss of an. This is also in alignment with Microsoft best practices.
- Use dedicated IBM System Storage N Series with FlexVol™ volumes for Storage Group databases.
- Use dedicated flexible volumes for transaction logs.
- Spindle counts for storage groups and log files should always be derived using the IBM N series Exchange Sizer.
- Use IBM System Storage N Series with SnapDrive® to provision LUNs. This reduces the complexity of LUN provisioning and eliminates the need of using Diskpart for LUN sector alignment. SnapDrive provides for correct LUN sector alignment and use of Diskpart should not be used.

Process and steps for doing a Jetstress POC

This section details a high level, 10-step process for conducting a Jetstress POC. This process was developed based on lessons and inputs from the field as well as knowledge and expertise of the Workload and Microsoft Alliance engineering teams.

STEPS	DESCRIPTION	ACTIVITIES
1	o Gather customer requirements	Meet with the customer, gather and understand their requirements. Specifically, obtain the number of mailboxes, mailbox size, and user profiles that clearly describe the expected IOPs per mailbox.
2	Consult with a FTSS and ask for Exchange sizing	With the help from your regional FTSS, do proper sizing and storage layout for the Jetstress POC. Have the FTSS review and make any necessary adjustments to the sizing and layout report.
3	Write test plan	o Specify the test cases. Outline the server and storage equipment needed for the test. Have the FTSS review and approve the test plan.
4	Configure test environment	o Configure server(s), storage system(s), and storage switch(s), if needed.
5	Setup Jetstress	o Configure Jetstress with appropriate input parameters.
6	Create Jetstress databases	o Create the databases as a separate process from actually running

	and use SnapDrive to create a snapshot of the clean databases	Jetstress performance tests. Create a snapshot for the baseline databases for reuse during Jetstress test iterations.
7	Run Jetstress test	(a) Start the N series performance tool, perfstat, on a server to monitor and log storage system performance during the test. See Appendix A for details about perfstat. (b) Start the Jetstress Performance test(s) on all participating servers at the same time. For the first assessment test, use dynamic thread tuning. If the dynamic tuning test does not pass, do a follow-on test using fixed number of threads evenly distributed across all instances. This process is described in section "6.5 Tuning Thread Counts" later in this document.
8	Analyze and Interpret the test results	o Check all Jetstress outputs to see if the test passed or failed, using the criteria in Section 3.4. Have the FTSS review the test results. In the case of a test failure, investigate Jetstress outputs and the perfstat output. Try to identify the root cause. Ask for help. The FTSS may need to interface with Workload Engineering when issues arise that cannot be resolved.
9	If required, restore the baseline databases and repeat Steps 7 & 8.	o When running additional tests, always use IBM System Storage N Series with SnapRestore® to put the environment back to the initial database creation state using the snapshot copy. Make manual necessary adjustments to thread count parameters and run another Jetstress test. It is not unusual to have to run several iterations of the test to produce the optimal results.
10	Summarize and the report test results	o Have the CSE review and approve the final results. Share the results and findings with the customer.

Table 2. Steps for conducting a Jetstress POC.

In the following 2 Sections, we will walk through this 10-step process to demonstrate how this process can be applied to a Jetstress POC. Section 5 will focus on test environment setup and cover Steps 1 – 4. Section 6 will focus on the details of Jetstress testing and cover Steps 5 – 10.

Overview of test environment

A Jetstress test environment setup should be based on customer requirements, ESRP methodology, Exchange Sizer outputs recommendations and Exchange Server 2003 best practices.

Test scenario

We will use a fictitious customer, with an Exchange deployment that supports 6000 heavy users (1.0 IOPS per user). The size of each mailbox is 100MB and the concurrency rate is 100%. Note, this is not a real world scenario; rather it is used to illustrate the Jetstress process and best practices.

Following the ESRP guideline, no more than 4000 users should be hosted on a single server. That means 2 servers are needed for this test, each server hosting 3000 users. A single filer will be used as the storage for these 6,000 users.

Exchange sizing and storage layout

In addition to the information mentioned prior, the following was also needed as inputs to the Exchange Sizer (version 3.5.1):

- 6000 Mailboxes
- User Profiles: 1

- IOPs per User: 1.0
- Servers: 2 (3000 per Server)
- RAID Type – RAID-DP
- RAID Group Size – 16
- Volume Type – FlexVol
- Drive Type – D10K72G
- Response Time (ms) – 20
- Deleted Item & Deleted mailbox Cache Space – 15%
- Snapshots Kept Online – 1 (snapshot copy captured by IBM System Storage N Series with Snapshot™)

Assuming a single aggregate, the Sizer suggested an initial configuration of 53 disks spread on 5 shelves. The Sizer also enables detailed layout planning with a simple button click. In the detailed layout stage, we separated databases and log files into their own aggregates (following IBM N series and Microsoft best practices). We also specified 4 Exchange Storage Groups (ESG) per server with each ESG using two LUNs, one for databases and one for log files. The Sizer suggested 2 aggregates: one for logs with 6+2 drives, and the other for databases with 39+6 drives (see Figure 1 in Section 5.5).

For this fictitious customer, we wanted to run a Jetstress Performance test using a single-head, multi-server configuration and the creation of the test plan was omitted. For real world customers, Step 3 (Table 2) should be followed to ensure test cases, objectives and expectations are clearly communicated and understood by the customer. Also, only high availability systems (WHQL logo'ed filer clusters) should be used for real world Exchange deployment.

Server configuration

Microsoft recommends, if possible, running Jetstress on servers and storage systems in an isolated Workgroup environment without Exchange Server software installed. Needless to say, Jetstress should never be run on production Exchange Servers.

In this test, we used two servers that are part of a Workgroup without Exchange Server software installed. After installing Jetstress and copying the needed files as described in Section 3.1, the following steps still needed to be completed in preparing the servers:

- Optimize Windows Server memory settings for Exchange..
- Install the Windows Host Utilities (version 3.0 was used for this test). .
- Install the latest Microsoft hotfixes that are required by Host Utilities.
- Update the Storport Driver (v9.1.2.16). Both servers have Qlogic host bus adaptors (HBAs), QLA2342, installed. This was downloaded from the Qlogic website. Note, if your server uses another brand of HBA, e.g. Emulex, you should check and ensure that proper storport driver is installed.
- Create a Local Administrator account. Create a user called "snapdrive" and make it a member of the Local Administrators Group. Also, set the "Password never expire" option.
- Install SnapDrive 5.0. Note: SnapDrive 5.0 requires various Microsoft hotfixes – these are identified in the release notes. Also, make sure SnapDrive service's LogOn Properties is set to "This Account" and uses the "snapdrive" account created in the previous step.

- Set HBA queue depth. Be sure to repeat these steps or similar steps for the HBA used in the customer environment. For this test we downloaded the Sansurfer command-line interface (CLI) for Windows from the Qlogic website and installed it on both servers. Then, we ran the Sansurfer CLI, scli.exe, in a command prompt and followed the steps below:
  1. At Main Menu, type 6, i.e. *Configure HBA Settings*, hit Enter key
  2. At Select an HBA Port, type 1, which should be the port that is online, hit Enter key
  3. At NVRAM Settings Menu, type11, i.e. *Execution Throttle*, hit Enter key
  4. Now you are at Enter Execution Throttle (1 – 65535) (current – 16), type **256**, hit Enter key. By doing this, you just changed the HBA queue depth from the default 16, to 256.
  5. At this point, you are back to NVRAM Settings Menu, type 20, *Commit Changes for this HBA*, hit Enter key. Note if you are using more than 1 HBA port, you need to repeat these steps for each port. After all ports are set, reboot the server.

IMPORTANT:

Setting HBA queue depth correctly is crucial for the success of a Jetstress test as well as ensuring the correct performance of the storage system.

Be aware that you will likely need to use HBA vendor's utility tool to set the HBA queue depth. For instance, if your server has Emulex HBAs installed, you need to use Emulex tool, *lputilnt*, to perform this task. Please reference Emulex or QLogic websites for more details.

Storage system configuration

The Exchange Sizer produced a detailed sizing and layout report that was used to configure the IBM N series filer. The following tasks were performed:

- Added licenses: common internet file system (CIFS), SnapRestore and FCP licenses are required.
- Added the Administrator account: the Local Administrator Account, "snapdrive", created on the servers in Section 5.3, was also added to the Storage Controllers local Administrator Group.
- Created 2 aggregates:
  - One for Exchange log files, aggr_sglogs, using 6+2=8 drives, total capacity 341GB.
  - One for Exchange databases, aggr_sgdata, using 42+6=48 drives, total capacity 2.33TB. Note the Exchange Sizer suggested 39+6=45 drives. We made minor adjustment by adding 3 more drives to round up to 3 full 16-disk RAID DP groups.
- Created 4 flexible volumes, 2 on aggr_sglogs for Exchange log files, and another 2 on aggr_sgdata for Exchange databases:
  - 2 flexible volumes on aggr_sglogs: fv1_sglogs (for server1) and fv2_sglogs (for server2). Each volume has total capacity of 102GB.
  - 2 flexible volumes on aggr_sgdata: fv1_sgdata (for server1) and fv2_sgdata (for server2). Each volume has total capacity of 896GB.
- Created 4 CIFS shares for the 4 flexible volumes just created:
  - fv1_logs at /vol/fv1_sglogs
  - fv2_logs at /vol/fv2_sglogs
  - fv1_data at /vol/fv1_sgdata
  - fv2_data at /vol/fv2_sgdata.

- Disabled scheduled snapshots and snap reserve for all 4 flexible volumes to be used by the Jetstress test.

Testing environment

To complete the testbed setup, SnapDrive was used to create the necessary LUNs for the ESGs and for the Log files. Each server hosts 4 ESGs. Each ESG requires 2 LUNs, one for databases and the other for log files. In total, 8 LUNs on each server were created. Note that when using SnapDrive to create LUNs, space was reserved for at least one snapshot. The resultant ESG and the filer storage layout is summarized in Table 3.

Controller	Aggregate	FlexVol	Server	Number of LUNs	NTFS Volume	Exchange Storage Group	RAID Type	Number of Data Drives	Number of Parity Drives
Filer Single Head	aggr_sgdata	fv1_sgdata	Fuji18	1x100GB	E:	ESG1 DBs	3x RAID-DP Groups	42	6
				1x100GB	F:	ESG2 DBs			
				1x100GB	G:	ESG3 DBs			
				1x100GB	H:	ESG4 DBs			
		fv2_sgdata	Fuji19	1x100GB	E:	ESG1 DBs			
				1x100GB	F:	ESG2 DBs			
				1x100GB	G:	ESG3 DBs			
				1x100GB	H:	ESG4 DBs			
	aggr_sglogs	fv1_sglogs	Fuji18	1x10GB	I:	ESG1 Logs	1x RAID-DP Group	6	2
				1x10GB	J:	ESG2 Logs			
				1x10GB	K:	ESG3 Logs			
				1x10GB	L:	ESG4 Logs			
		fv2_sglogs	Fuji19	1x10GB	I:	ESG1 Logs			
				1x10GB	J:	ESG2 Logs			
				1x10GB	K:	ESG3 Logs			
				1x10GB	L:	ESG4 Logs			

Table 3. Exchange Storage Group and IBM N series filer storage layout.

Before actually running tests top level folders were created per NTFS volume on both servers. In total, 16 top level folders were created as illustrated below:

- On Fuji18, E:\esg1, F:\esg2, G:\esg3, H:\esg4, I:\esg1, J:\esg2, K:\esg3, and L:\esg4
- On Fuji19, E:\esg1, F:\esg2, G:\esg3, H:\esg4, I:\esg1, J:\esg2, K:\esg3, and L:\esg4

The hardware and software used in our tests are briefly described below.

Servers:

- Two Fujitsu RX300 servers, each has 2 CPUs (3.6GHz) and 4GB of RAM
- One Qlogic FC HBA Ql2342 (dual 2 Gbps FC ports) per server
- Windows Server 2003, Enterprise Edition SP1
- IBM N series with SnapDrive 4.1 with MPIO
- IBM N series with FCP Windows Host Utilities
- The latest hotfixes required by SnapDrive and FCP Host Utilities

Storage:

- One filer with two clustered filer storage controllers. Only one is used in all of our tests. Note, for real world deployment, filer clusters should be used. These are WHQL Logo'ed storage platforms listed in the Windows Server Catalog.
- IBM System Storage N series with Data ONTAP® 7G storage operating system.
- A total of eight DS14 shelves consisting of 112, 72GB, 10K RPM drives were attached to the single filer controller. A total of 56 drives were used in our tests.

The test environment is shown in Figure 1.

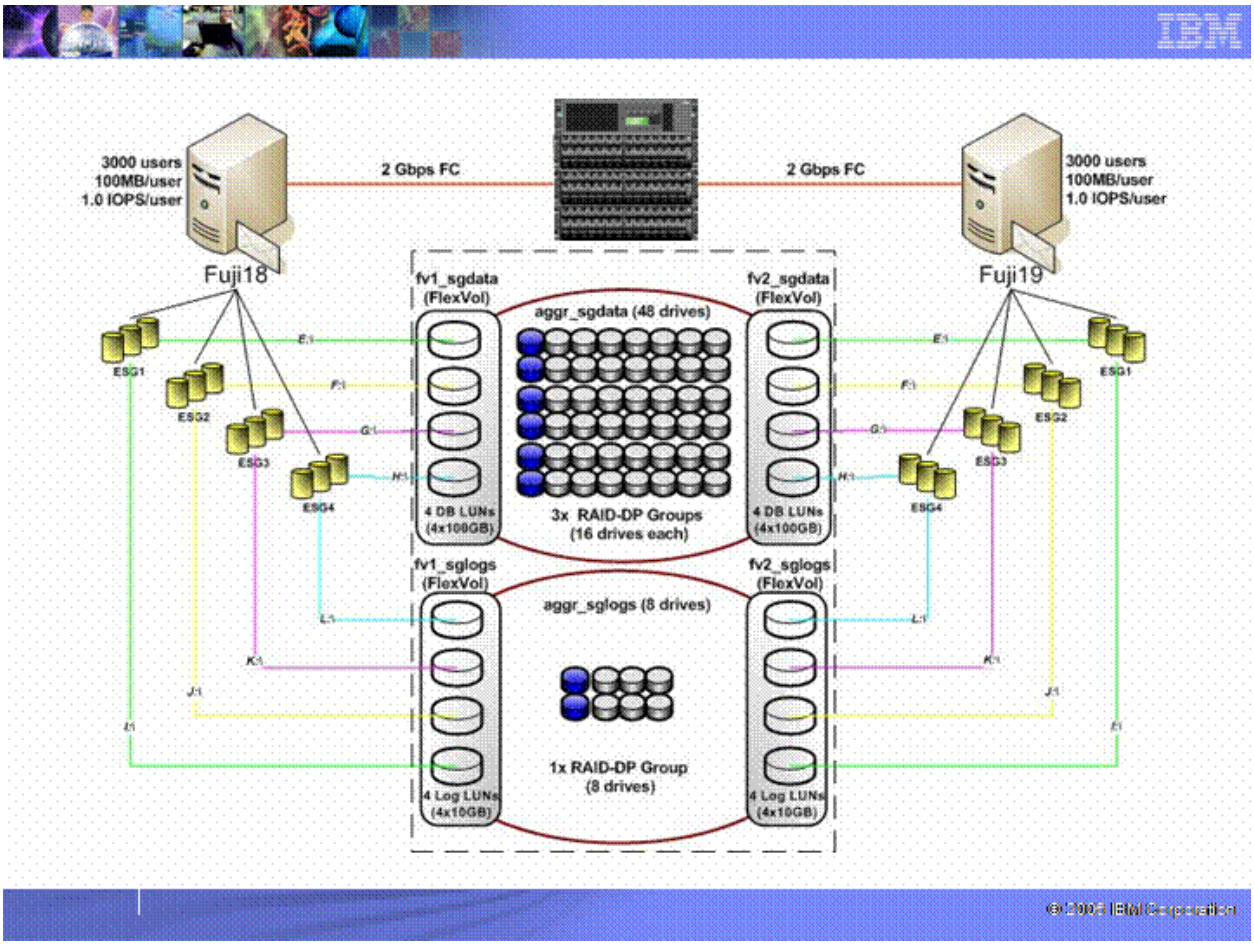

Figure 1. Jetstress test environment and storage layout.

Configuring Jetstress

This section provides in-depth and step-by-step tutorial on the how-to aspects of running Jetstress tests.

How to configure Jetstress

Jetstress groups input parameters into three categories: Storage Info, Test Run Info, and Database Info. Note these need to be done on both servers. ***Storage Info*** should be configured as follows:

- Number of storage groups: 4.
- Number of disk drives per storage group: 1. This means we used one LUN for all databases (.edb files) under a given storage group. Note that Log LUNs are not counted here.
- Use storage volumes on NAS: unchecked. This is because we are using FCP and block level access which is always the recommended and supported approach. This also applies when the storage is connected using iSCSI. .
- Filtered by: select Storage Group 1, then set Database path and Log path as below:
  - Database path: browse and select folder E:\esg1
  - Log path: browse and select folder I:\esg1
- Repeat the "Filtered by" step above, and select then set Database and Log paths for Storage Groups 2 – 4, respectively:
  - Storage Group 2, F:\esg2, J:\esg2
  - Storage Group 3, G:\esg3, K:\esg3
  - Storage Group 4, H:\esg4, L:\esg4

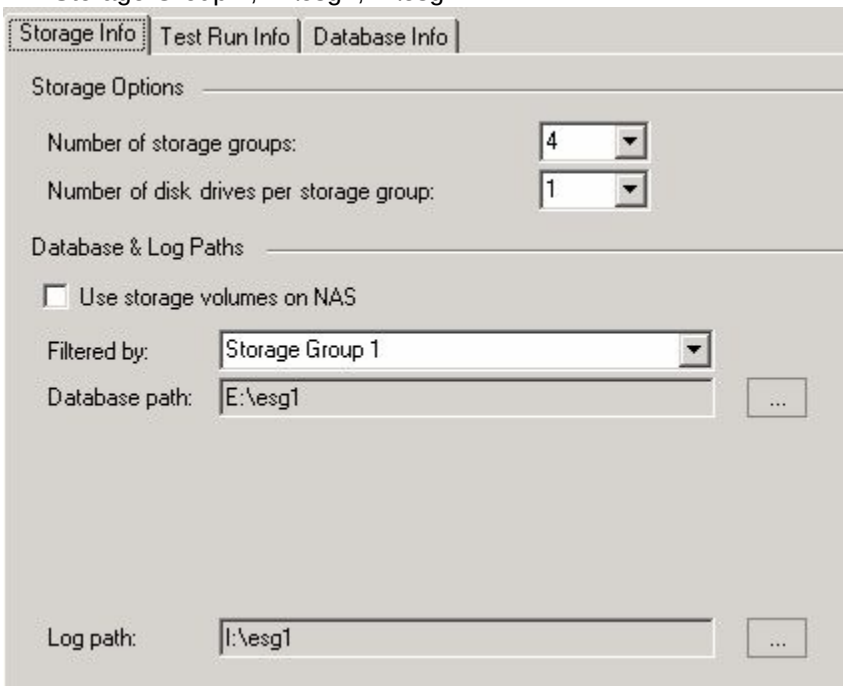

Figure 2. Jetstress storage info tab.

Test Run Info should be configured as follows:

- Test Run Type: choose Performance.
- Duration (hours): 2.
- Log Buffers: 9000, use the default value.
- Perf Log File Type: choose Binary File as it is easier to use with Perfmon.
- Output Path: C:\Program Files\Jetstress\results
- Tuning types::
  - For the first test, chose *Use tuning based on target I/O and disk latency* and let Jetstress attempt to determine the optimal number of threads. If either the dynamic tuning phase fails or the Jetstress test fails, review the Actual IOPs vs. Expected IOPs in the performance test html output. If Actual IOPs are much higher than Expected IOPs, manually reduce the thread

counts, check *Suppress tuning and use test I/O parameters* and rerun the test. If however the Actual IOPs are lower than Expected IOPs, increase the thread count, check *Suppress tuning and use test I/O parameters* and rerun the test. Thread counts should always be evenly distributed across each instance. **When using hard coded thread counts always check the *Suppress tuning and use test I/O parameters* option to ensure a fixed number of threads over the duration of the test.**

- *Lock Database Operation Mix*: checked.
- 4 instances of Operation Mix parameters are shown; one per Storage Group. For each instance, set *Inserts/Replaces/Deletes/Lazy Commits* to 17/70/5/90. As discussed in the previous section, for the initial run, leave the default thread settings and be sure to select the ***Use tuning based on target I/O and disk latency***, and set the ***Lock Database Operation Mix*** option, as shown in the figure below.

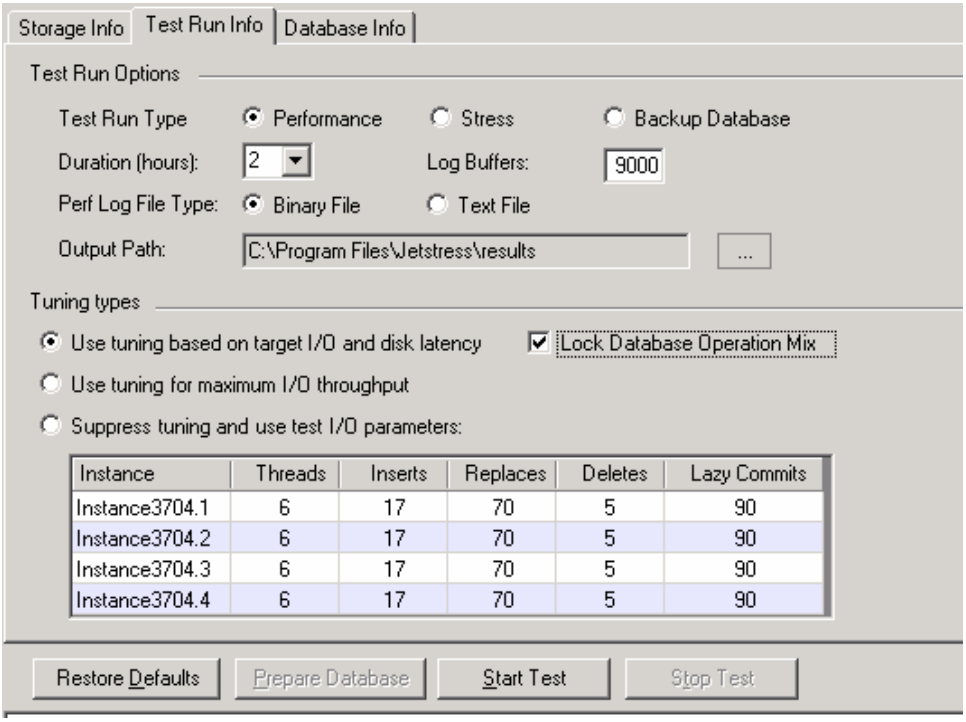

Figure 3. Jetstress test run info tab.

Database Info should be configured as follows:

- *Select database*: choose *Create new databases*, if you run Jetstress the very first time. Choose *Attach to existing databases*, if this is not the very first test and you have either created new databases or restored the databases' snapshot using snap restore. Note: leave the *Backup path* blank as SnapDrive is used to backup databases.
- *Mailbox size limit (MB):* 100.
- *Number of mailboxes*: 3000. i.e. 3000 per server, for a total of 6000.
- *Estimated IOPS per mailbox*: 1.
- *Storage hardware cache size (MB):* 8096 (cache size in the 980) for each server.

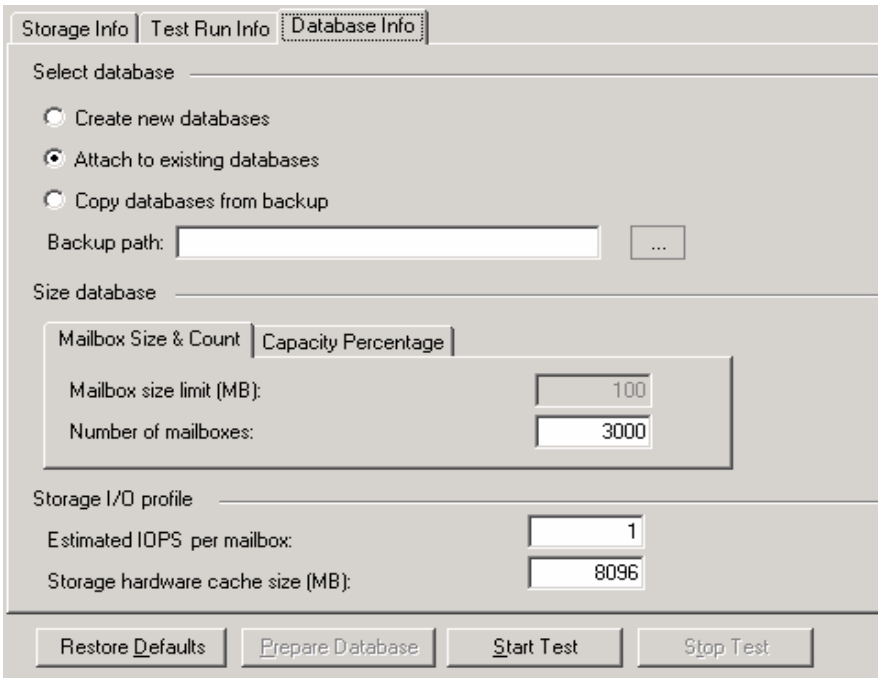

Figure 4. Jetstress database info tab.

At this point, use the File menu and select Save Configuration to save the configuration. **Note that opening the configuration file in Jetstress will not retain the "Lock Database Operation Mix" option setting. To retain this setting, you will be required to recheck it in subsequent runs. Failure to do this will almost certainly produce unfavorable results.**

How to run Jetstress

Follow the steps below, paying careful attention to their order. Note that if not explicitly stated, all steps below must be performed on both servers, one after another.

- **Create Jetstress databases.** Before you run Jetstress the first time, you must create new databases by selecting the *Create new databases* option on the Database info tab and clicking the *Prepare Database* button. Do not click the *Start Test* button during this step as you need to have a known baseline to go back to later on during test iterations.
- **After database initialization is complete, create a snapshot of the baseline Jetstress databases**. This is done using SnapDrive on both servers:
  - Close Jetstress.
  - Open the Computer Management console (Microsoft Management Console, MMC).
  - Navigate to Storage\SnapDrive\Disks\VirtualDisk[#,#,#,#] (E:\)\Snapshots. Note the 4 "#" signs represent 4 integer numbers (>=0).
  - Right click on *Snapshots*, and select *Create Snapshot*… in the context menu.
  - Enter a descriptive snapshot name and hit **OK**. You only need to do this once to snapshot all four ESGs' database LUNs on the server — a snapshot is per FlexVol and all four database LUNs were created on the same FlexVol.
  - Open Jetstress.
- Run a Jetstress test. The steps below should be followed.

- You need to make sure the command line tool perfstat.exe has been installed on server1. On server1, in our case, Fuji18, open a command prompt and run perfstat, as follows: `perfstat –f FilerName –t 5 –i 25 > jetstress.out`. See Appendix A for more details about perfstat. This step only needs to be done on just one of the servers.
- In the Jetstress GUI, switch to the Database Info page and choose *Attach to existing databases*. This assumes you have clean databases in place. If not, you need to restore databases first. See Section 6.4 for more details.
- Switch to the Test Run Info page and verify all Jetstress parameters are still configured according to the specifications in Section 6.1.
- Click *Start Test* button to start the Jetstress test. Ensure that the test starts on both servers more or less at the same time.

In actuality, the two-hour test will always run longer than 2 hours. In the case of the dynamic tuning test, Jetstress will first try to find proper thread counts by doing some auto tuning (less than 1 hour). Then it runs the performance test for 2 hours. After that, the checksum verification test runs on the databases and log files to ensure no data corruption has occurred. In our case, the checksum took between 1 and 2 hours to complete.

How to interpret the results

After the Jetstress test completes, the output files described in Section 3.2 will be generated on both servers. You should backup all test results during your Jetstress POC. Use the criteria listed in Section 3.3 (in that order) to review the output files and determine if the test passed or failed. All criteria must be met on all servers for a test to pass.

Furthermore, as a “sanity check”, it is recommended to validating the perfstat data by comparing to the perfmon data, and making sure they are in reasonable agreement. Here is how you can cross reference perfstat and perfmon data.

First upload the perfstat file to the perfstat viewer. Once the perfstat file has been uploaded, use the perfstat viewer to select a data point gathered during the middle of the two hour run. Once the data point has been selected, navigate to the lun_stats.out data point. Figure 5 below shows an example perfstat lun_stats.out data point including all 8 database LUNs from one of our tests, 4 from Fuji18 (fv2_sgdata/srv2_sg#data.lun), and 4 from Fuji19 (fv1_sgdata/srv1_sg#data.lun), where the # is: 1, 2, 3 and 4.

```
=-=-=-=-=-= PERF chakotay POSTSTATS =-=-=-=-=-= type ...\chakotay\lun_stats.out
 Mon 09/25/2006 17:47:23.87
 Read Write Other QFull  Read Write Average  Queue Partner Lun
  Ops   Ops   Ops          kB    kB Latency Length Ops kB
  885   193     0     0  3540   880   15.73  16.00   0  0 /vol/fv1_sgdata/srv1_sg1data.lun
  951   198     0     0  3808   876   14.69  16.00   0  0 /vol/fv1_sgdata/srv1_sg4data.lun
  876   208     0     0  3508   964   15.40  15.09   0  0 /vol/fv1_sgdata/srv1_sg3data.lun
  936   185     0     0  3756   832   15.18  16.01   0  0 /vol/fv1_sgdata/srv1_sg2data.lun
  952   174     0     0  3812  1800   14.99  16.01   0  0 /vol/fv2_sgdata/srv2_sg1data.lun
  904   155     0     0  3608   644   15.86  15.09   0  0 /vol/fv2_sgdata/srv2_sg2data.lun
  921   166     0     0  3680   712   15.37  15.08   0  0 /vol/fv2_sgdata/srv2_sg4data.lun
  942   173     0     0  3776   792   15.29  16.00   0  0 /vol/fv2_sgdata/srv2_sg3data.lun
```

Figure 5. perfstat’s lun_stats.out data for the 8 database LUNs on both servers.

To effectively compare perfstat vs. perfmon data, you need to know the mapping between LUNs and corresponding NTFS volumes. Table 4 below shows this mapping in our test environment.

SERVERS	NTFS VOLUMES	DATABASE LUNS
Fuji18	o E:\	/vol1/fv2_sgdata/srv2_sg1data.lun
	F:\	/vol1/fv2_sgdata/srv2_sg2data.lun
	G:\	/vol1/fv2_sgdata/srv2_sg3data.lun
	H:\	/vol1/fv2_sgdata/srv2_sg4data.lun
Fuji19	o E:\	/vol1/fv1_sgdata/srv1_sg1data.lun
	F:\	/vol1/fv1_sgdata/srv1_sg2data.lun
	G:\	/vol1/fv1_sgdata/srv1_sg3data.lun
	H:\	/vol1/fv1_sgdata/srv1_sg4data.lun

Table 4. Mappings between database LUNs and NTFS volumes.

Next, we loaded the perfmon logs generated during the tests into perfmon for both servers and picked a 5-minute time period that overlaps with the perfstat data point. We then added relevant disk counters and selected to display the results in the report view. The perfmon data is shown in Figures 3 and 4.

Note that perfstat and perfmon have different naming conventions for the various counters. To help simplify the comparison process, we have listed the mappings between counters in Table 5 below.

PERFSTAT COUNTERS (LUN_STAT.OUT)	PERFMON COUNTERS
o Read Ops	Disk Reads/sec
Write Ops	Disk Writes/sec
Read KB	Disk Read Bytes/sec
Write KB	Disk Write Bytes/sec
Average Latency	Avg. Disk sec/Transfer
Queue Length	Avg. Disk Queue Length

Table 5. perfstat and perfmon counter name mapping.

As can be seen from Figures 5, 6 and 7, the perfstat data and perfmon data are in reasonable agreement, except for Write Ops and Write kB. This delta can be attributed to how IBM System Storage N Series with WAFL® optimizes for writes by effectively sequentializing both sequential and random writes.

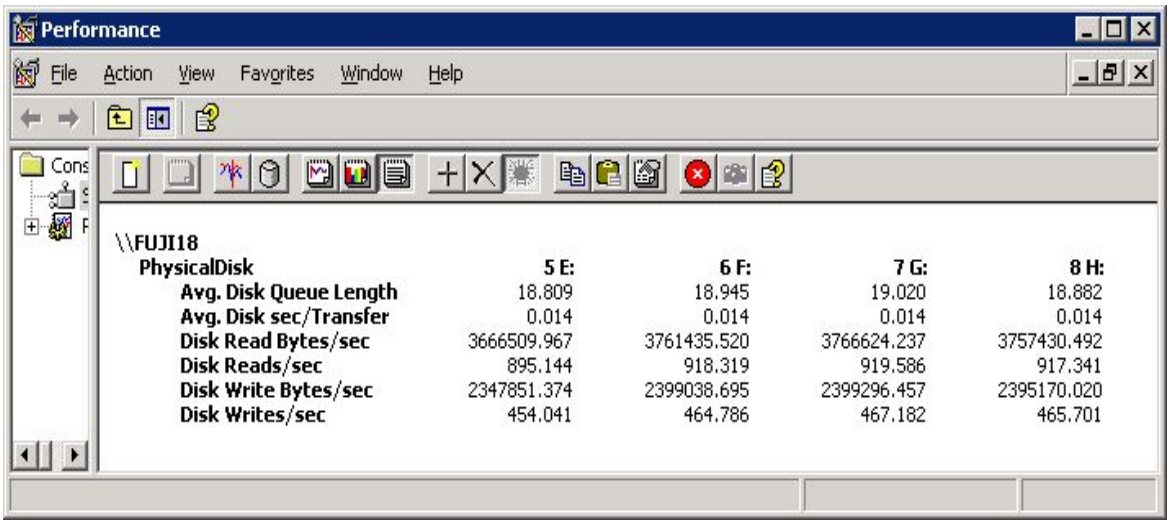

Figure 6. Fuji18 perfmon data averaged over ~5-minute time period: 09/25/2006 17:46:38 – 17:51:24.

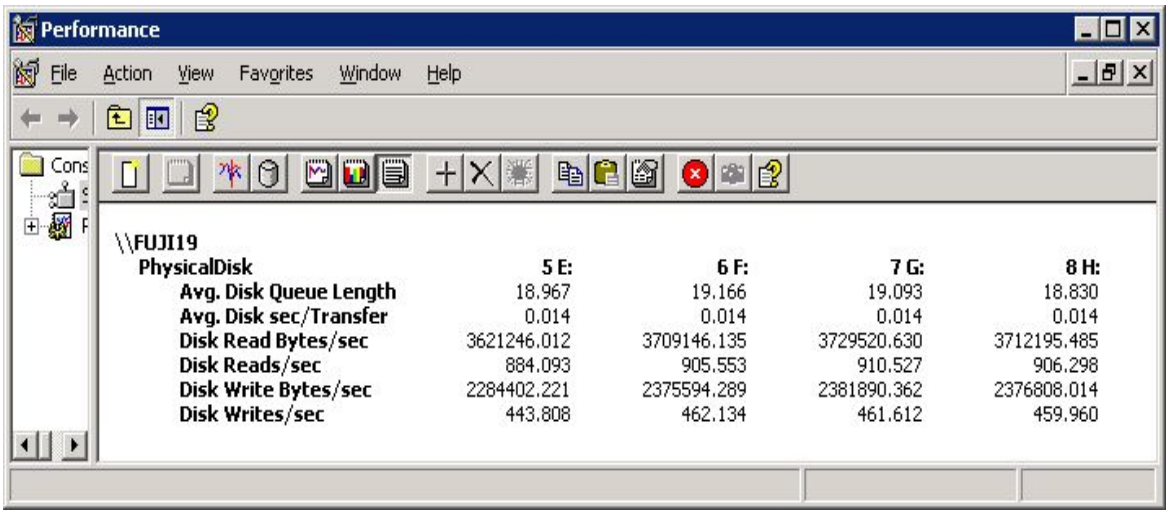

Figure 7. Fuji19 perfmon data averaged over ~5-minute time period: 09/25/2006 17:46:36 – 17:51:16.

Additionally, it is a good practice to always review the filer etc/messages file after every test run to make sure no extraordinary events, such as IBM System Storage N Series with SnapMirror®, scrubs or panics occurred during the test. Note that in ONTAP 7.2 and above there are two types of scrubs: continuous scrubs and scheduled scrubs and both are still required. The default scheduled scrub starts at 1:00AM every Sunday. We recommend leaving the continuous scrubs running and disabling scheduled scrubs if your test may overlap with the scheduled scrub time window. Note also that you should check to ensure that the clock and time on the storage system is set properly. Otherwise, your test may unexpectedly run into the scheduled scrub time window and the test results may be severely distorted.

If a test fails, try to analyze the root cause and/or engage your FTSS or IBM services resources in order to understand why it failed. In our test, the first trial failed because the Achieved I/O did not meet the Expected I/O. We tried to increase thread count then re-run the test (see Sections 6.4 and 6.5 below for more details).

How to rerun the test

As previously mentioned, it is likely that you may have to run Jetstress tests several times to achieve the most optimal configuration and to get successful results. One important point to remember during test iterations is that you want to start every test using the same set of Jetstress databases at the same point in time. This makes the comparison between test runs possible. Before re-running each test, we used the `Snap Restore` command on the storage system to restore the baseline Jetstress database snapshot created earlier.

Following steps are used (on both servers) to restore the baseline databases:

- Telnet and log on to the filer
- Shut down server1
- After the shutdown is complete, issue the `Snap Restore <flexvol>` command and confirm that you want to restore the correct snapshot.
- Restart server1
- Repeat the above steps for server2
- After the server is online again, start the test (Jetstress will automatically delete the logs).

Once the `Snap Restore` has completed, adjust Jetstress thread parameters and run another test cycle with the settings described in section 6.5 “Tuning Thread Counts”.

Tuning thread counts

Is it better to let Jetstress auto tune or to do manual tuning? During the testing it was determined both methods were complimentary and when used together, help create a positive outcome.

Auto tuning might be faster, however, there are two weaknesses. 1) Sometimes the thread count does not balance across Exchange storage groups and test servers. For example, it was observed that after Jetstress performed auto tuning, on server1, storage group 1 was automatically configured for 12 threads, while the other 3 storage groups were configured for 8 threads each. This type of asymmetrical thread tuning may result in one database LUN being overloaded while the other is starved. 2) In the 2-server or multi-server environment, the time Jetstress takes to auto tune was different on each server. For instance, server 1 may complete auto tuning and start performance test, while server2 continues to auto tune for an additional 15 minutes before starting the test. This phenomenon was observed in the test environment and it contributed to not only skewed test results but also Jetstress test failures.

On the other hand, manual tuning can be time consuming and *ad-hoc* in nature. However, it did give us a greater degree of control and enables a broader range of successful performance profiles while staying within acceptable latency targets.

To benefit from the strength of both methods, it is better to utilize a combined approach, summarized in the following 3 steps:

1. Start by using auto tuning combined with ESRP settings to get a rough estimate of the total thread count per server. Select *Use tuning based on target I/O and disk latency*. Ensure there are a total of 4 instances. For each instance, set *Inserts/Replaces/Deletes/Lazy Commits* to 17/70/5/90. Also be sure to set the “Lock workload operating mix”.
2. If the Jetstress test passes then it may be desirable to accept the results generated and conclude testing. If the test either failed or did not meet expectations, review the Actual IOPs vs. Expected

IOPs in the performance test html output. If Actual IOPs are much higher than Expected IOPs, reduce the thread counts and rerun the test. If Actual IOPs are lower than Expected IOPs, try increasing the thread count and rerunning the test. Try starting with a conservative number of threads and run a series of tests whereby thread count is progressively increased successive runs. For fixed thread runs, always be sure to check the ***Suppress tuning and use test I/O parameters*** option.

3. Repeat the steps in step 2 until the test passes.

Figure 8 below shows a range of successful results that were generated by manually varying the thread counts using the ESRP settings[3].

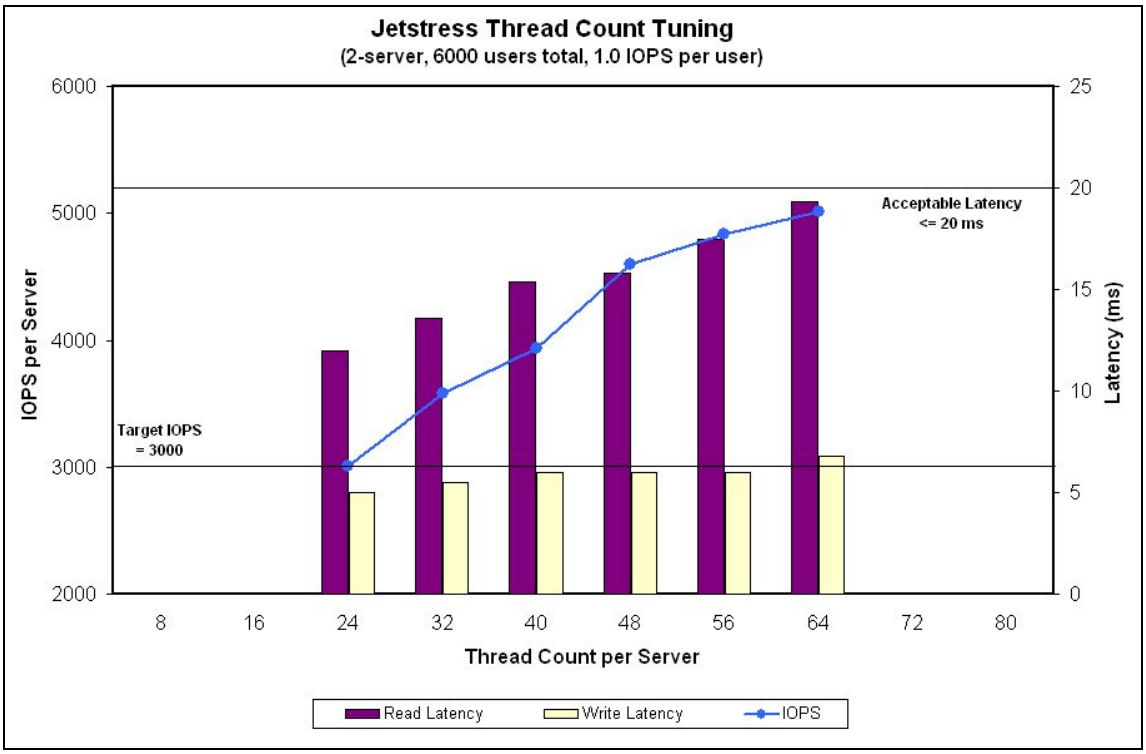

Figure 8. Threads vs. IOPS and read/write latencies for a series of two hour tests.

Thread counts per storage group were manually increased from 6 to 8, 10, 12, 14 and 16. There were 4 storage groups per server, the corresponding total thread counts per server were 24, 32, 40, 48, 56 and 64. This chart helps illustrate a variety of thread settings that can produce a range of successful results.

Troubleshooting

While Jetstress is a relatively simple test (when compared to the MMB3 benchmark tool, Loadsim), there are still several things that things can go wrong. Clearly, the best way to counter errors or test failures is to ensure proper prevention mechanisms are in place before starting the test. Following the process outlined in Section 4.3 and the best practices described in this report will ensure success. With that said,

[3] Instead of increasing thread count, one can also increase the number of users. To do so, Jetstress database needs to be reinitialized for each different user count.

it is also very helpful to know some signs of a successful Jetstress test. Table 6 below summarizes what should be expected from a good Jetstress test.

Performance Counter Instance	Guidelines for Performance Test	Guidelines for Stress Test
Database Avg. Disk sec/Read	The average value should be less than 20 ms (.020), and the maximum value should be less than 50 ms.	The maximum value should be less than 100 ms.
Database Avg. Disk sec/Write	This counter is not evaluated to determine whether a test passed or failed, but in an environment where storage replication is not being used, the average value should be less than 20 ms (.020).	
Log Avg. Disk sec/Read	The average value should be less than 20 ms, and the maximum value should be less than 50 ms.	The maximum value should be less than 50 ms.
Log Avg. Disk sec/Write	Log disk writes are sequential, so average write latencies should be less than 10 ms, with a maximum of no more than 50 ms.	The maximum value should be no more than 100 ms.
Database Disk Reads/Sec **Database Disk Writes/Sec** **Log Disk Writes/Sec**	The sum of the averages for these values gives you the total disk transfer I/O. The ratio between read and write should be approximately 3:2.	
Log Avg. Disk Bytes/write	This value should be between 6 to 8 K.	
%Processor Time	Average should be less than 80 percent and the maximum should be less than 90 percent.	
Available Mbytes	Minimum should be more than 50 MB.	
Free System Page Table Entries	Minimum should be more than 5000.	
Pages/sec	Average should be less than 100, and the maximum should be less than 1000.	
Pool Nonpaged Bytes	Maximum should be less than 75 MB.	
Pool Paged Bytes	Maximum should be less than 180 MB.	
Database Page Fault Stalls/sec.	Should never go above 0.	

Table 6: Guidelines for good Jetstress tests (Source: Microsoft Jetstress.doc, pp. 29 & 30).

In the case of a test failure where perfmon data does not provide enough information, perfstat output will be helpful in pinpointing the issues. System stats, protocol stats and LUN and disk stats, among other things, should be carefully investigated to help identify the root cause. See Appendix A for more details.

In our tests, we initially had difficulty to meet the target IOPS number. We examined the logs and determined that the “Queue Depth” counter in perfstat’s fcp_stats seemed stuck at 16. This led us to check the HBA queue depth setting on both our test servers. The HBA queue depth was left at 16 the default value. After setting the queue depth to 256 on both servers, all subsequent tests passed without any more problems.

When doing a Jetstress POC, always be sure to engage the expertise of your local Exchange CSE. If after analyzing Jetstress logs, results and perfstat data you and your CSE are still have difficulty resolving the issues; it may be time to engage the workload engineering.

Jetstress best practices for IBM N series

This section summarizes the best practices discussed in this report:

- Get the FTSS or IBM professional services involved at very beginning of the POC. It should be a paid for Professional Services engagement where appropriate.
- Follow ESRP guidelines and procedures.
- Use the Exchange Sizer.
- Follow the 10-step process outlined in Section 4.3.
- Separate storage group database files from all transaction logs into separate aggregates.
  - Use dedicated flexible volumes for Storage Groups databases.
  - Use dedicated flexible volumes for transaction logs.
- Spindle counts for storage groups and log files should always be derived using the Exchange Sizer.
- Use SnapDrive to provision LUNs. This reduces the complexity of LUN provisioning.
- Check and validate drivers, hotfixes, etc.
- Configure Jetstress properly – checkboxes, parameters, thread counts, and use the "Lock Database Operation Mix" for each test run. Be sure to always follow ESRP Guidelines.
- Configure server properly, set HBA queue depth to 128 or 256. For multiple servers, set the same queue depth on all servers.
- Configure storage systems properly, enable licenses, disable scheduled scrubs, etc.
- Create databases independently of the performance test.
- Snapshot baseline databases for future use.
- Reboot servers and storage systems between test runs.
- Restore baseline database snapshot before each new test to maintain consistency.
- Start with a dynamic thread tuning test. If required, use fixed thread settings for follow-on tests.
- Run perfstat, use "***perfstat –f FilerName –t 5 –i 25 > jetstress.out***" in a command prompt on one of the servers.
- Upload perfstat output to the perfstat viewer. .
- Save and backup test results.
- Results interpretation: Use the Jetstress reports and logs, event logs, perfmon logs, perfstat logs and the etc/messages file.
- Look for any inconsistencies, areas to validate include:
  - LUN read and write latency reported by perfstat vs. perfmon.
  - Read and write disk I/Os reported by perfstat vs. perfmon
- Multi server considerations – make sure the tests start at the same time or as close to one another as possible, particularly when auto-tuning is used.

Summary

Using the ESRP methodology combined with the best practices presented in this report will help ensure a successful Jetstress testing in your customer environment. If you have any questions regarding any of the information presented in the report, please contact your regional FTSS.

Appendix A - How to use perfstat

Perfstat is an performance tool that can collect a lot of useful performance data on filers as well as hosts.

Note, always verify that you are using the latest and greatest version of Perfstat. On Windows platform, after downloading the tool, perfstat.exe, to a local directory, for example C:\tools\perfstat.exe, you can open a command prompt to run the tool. Use **`perfstat /?`** for usage. When you run a Jetstress test, first run the following command on one of the server:

```
perfstat –f <filer> -t 5 –i 25 > jetstress.out
```

This command will collect data on the current server and the <filer>. It spends 5 minutes to collect data for each sample, and wait 10 minutes in-between data points. It will run 18 iterations and take roughly 3 hours, thus covers an entire Jetsteress test period. The output it generates will be about 30 MB. It is an ASCII text file and can be viewed with a plain old edit like Notepad.

It will take a few minutes to complete the upload of the perfstat output file. When it is successfully uploaded, it brings you to a resultant webpage. You can inspect the performance data online.

Perfstat captures tons of useful information about the filer's configuration, health, and performance statistics. Without diving into the filer operation theory, here are some tips about what to look for and where (all under Filer/Perf/PostStat).

- fcp_stats – check the "Queue Depth" counter, make sure it's not pegged to a small number (e.g. 16).
- perfstat_sys – check the "cpu_busy" and "average_processor_busy" counters, make sure the filer CPU is not saturated.
- perfstat_fcp (for FCP) or perfstat_iscsi (for iSCSI) – check the "read_latency", "read_ops", "write_latency" and "write_ops" counters, make sure they are reasonable with latencies at ~20 ms or less, and Ops near or above target.
- lun_stats.out – check the "Read Ops", "Write Ops" and "Avg Latency" for each LUN of interest. Make sure these numbers and the respective numbers from perfmon on the server are comparable. If abnormal discrepancies are found, further investigation is necessary.
- perfstat_disk – check the "user_read_latency", "user_write_latency", "user_reads", "user_writes", "disk busy" and "total_transfers", make sure these numbers (for individual physical disk) are in reasonable ranges.
- statit.out and sysstat.out – these two outputs give system wide overview. Check to see if there are any anomalies.

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

Network Appliance, the Network Appliance logo, Data-ONTAP, WAFL, RAID-DP, FlexVol, SnapDrive, SnapRestore. Snapshot, SnapMirror and SnapManager are trademarks or registered trademarks of Network Appliance, Inc., in the U.S. and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

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.