White Paper

BACKUP AND RECOVERY OF THE EMC GREENPLUM DATA COMPUTING APPLIANCE

Greenplum Database, Greenplum HD, Greenplum Chorus, Greenplum UAP, EMC Data Domain Systems, Network File System, EMC Data Domain Boost

- Data warehouse protection
- Major storage savings with data deduplication
- Simple and fast restore

EMC Solutions Group

Abstract

This white paper provides insight into how EMC® Data Domain® deduplication storage systems effectively deal with the data growth, retention requirements, and recovery service levels that are essential to businesses. Data Domain's industry-leading technology provides a powerful backup, archiving, and disaster recovery solution that can scale with the most demanding data center requirements. This white paper explores the various practices and considerations for backing up EMC Greenplum® Data Computing Appliance data to Data Domain systems and how to effectively exploit Data Domain's leading-edge technology.

December 2012

EMC² PROVEN SOLUTIONS

GREENPLUM A DIVISION OF EMC

EMC²

Copyright © 2012 EMC Corporation. All Rights Reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided “as is.” EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All trademarks used herein are the property of their respective owners.

Part Number H8038.8

Table of contents

Executive summary

Business case

EMC® Greenplum® Unified Analytics Platform (UAP) combines the power of its classic data warehouse Greenplum Database™ (GPDB) for structured data management, with Greenplum HD for unstructured data management. The other critical software component of UAP is Greenplum Chorus™ for collaborative analytics. Greenplum UAP enables organizations to make strategic business and operational decisions based on advanced analytics, which in turn contribute to the overall success of an organization, both functionally and economically.

To ensure that an organization can make the right decisions in the shortest possible time, its data warehouse, including structured and unstructured data, must be reliable, secure, high performing, and extremely flexible to support the growing amounts of data, concurrency of many users, and increasing complexity of analysis.

Therefore, an organization's data needs to be loaded into the system so that it can be queried intelligently and quickly. The data also needs to be backed up, recovered, and restored efficiently and cost-effectively.

In today's business environments, data analytical systems are:

- Growing exponentially over time, often leading to many terabytes of storage requiring protection.
- Frequently becoming unmanageable, lacking the ability to back up or provide effective disaster recovery (DR).

EMC has created a purpose-built analytics platform—Unified Analytics Platform—using the Greenplum Data Computing Appliance (DCA). The DCA addresses essential business requirements and ensures predictable functional, performance, and scalability results. This eliminates the guesswork and unpredictability of a highly customized in-house solution.

Data protection for a data warehouse is a critical IT discipline. Businesses have historically chosen simple approaches, such as periodically creating a full backup to tape. With the rapid growth in the volume of data stored, a simple, periodic full backup to tape or to a non-deduplicating, disk-based storage system is no longer economical. In addition, tape or non-deduplicated disk-based backups do not provide the recoverability or reliability of next-generation backup solutions like EMC Data Domain® deduplication storage systems.

The DCA, combined with Data Domain systems, provides a total solution for data warehousing deployment that addresses all these key challenges. This white paper illustrates how to back up data from the DCA to a Data Domain system, and how to subsequently recover and restore the data to the DCA. EMC recommends that you read this paper in conjunction with the white paper *EMC Greenplum Data Computing Appliance: Architecture, Performance and Functions—A Detailed Review*.

Solution overview

Data warehouse environments today demand a more comprehensive strategy for data protection, security, and high availability than ever before. Data recovery options must align with application and business requirements to yield the highest availability. Organizations with traditional backup and recovery systems face many challenges and require:

- Efficient use of both infrastructure and people to support the business by:
  - Improving IT efficiency—save hours of staff time and boost user productivity
  - Reducing operational and infrastructure costs by eliminating the requirements of tape and reducing data center footprint requirements to a single rack
  - Reducing the need for massive backup storage
  - Correcting and reducing costs—matching infrastructure costs with changing information value through efficient, cost-effective storage
- Simplicity through:
  - Ease of integration
  - Ease of management
- Faster backup and restores to:
  - Minimize backup windows to reduce the impact on application and system performance
  - Meet more aggressive recovery time objective (RTO) service-level agreements (SLAs) set by business owners by maximizing end-to-end recovery performance

The DCA provides an end-to-end data warehousing/business intelligence (DW/BI) solution packaged within a manageable, consolidated, self-contained data warehouse appliance that can be easily integrated into an existing data center. This white paper describes a backup and recovery environment for the DCA using EMC Data Domain deduplication storage system.

Many DCA product options are available. This solution uses two of the configuration options available:

- A full-rack DCA with four GPDB modules
- A full-rack DCA set up in a UAP configuration with one GPDB module and three Greenplum HD modules

The backup and recovery solution for the GPDB modules uses the Greenplum **gpcrondump** backup utility. The utility was deployed using the following two methods:

- Over a network file system (NFS) protocol
- Using EMC Data Domain Boost

Both methods back up the data to a Data Domain deduplication storage system.

The **gpcrondump** backup utility automates the parallel backup of distributed Greenplum databases across multiple Segment Servers. For NFS, each of the Segment Servers has a directory mapped to a Data Domain system. For Data Domain Boost, data is backed up to a dedicated storage unit on the Data Domain system.

The backup and recovery solution for the Greenplum HD modules uses the **DistCp** (distributed copy) utility. **DistCp** was deployed over NFS only. The **DistCp** tool is used for large inter-/intra-cluster copying. It uses MapReduce to achieve its distribution, error handling and recovery, and reporting. It expands a list of files and directories into input to map tasks, each of which copies a partition of the files specified in the source list. For more information, visit: http://hadoop.apache.org/common/docs/stable/distcp.html.

New data protection solutions led by Data Domain's architecture improve data recoverability, while increasing the probability for a business to survive most types of outages. This is increasingly critical since most businesses cannot survive now without their BI function. A backup solution that provides an easy and fast method of data recovery is a necessary requirement rather than an optional one.

EMC has ensured that leading-edge technology is available to support the backup and recovery of its DCA. This white paper demonstrates how the Data Domain system achieves this in the most simple, efficient, and cost-effective way to provide greatly increased storage savings.

Key results

With data deduplication, organizations can reduce the amount of data that needs to be stored over an extended period of time. This offers cost savings in terms of the number of disks or tapes required for backup. In addition, organizations can fundamentally change the way they protect their backup and archive data. Only unique, new data is written to disk, eliminating the restore penalties associated with incremental backups.

This solution describes a backup and recovery environment for the DCA using the Data Domain DD890. The test results show that this solution meets the business challenges faced by many organizations today through:

- Operational ease and efficiency through the utilization of massively parallel processing (MPP) architecture for GPDB backups to efficiently back up, in parallel, across the network to the Data Domain system.
- Major space saving advantages using Data Domain inline deduplication—in this case, test results with GPDB DD Boost demonstrate savings of 34.5x on repetitive nightly backups and an aggregate 11.3x cumulative storage reduction over a week of uncompressed backups to the Data Domain system. One of the key benefits of Data Domain deduplication storage systems is a reduction in the need for excessive amounts of backup storage.
- Fast restore times for returning uncompressed and deduplicated data to the DCA. The test results with GPDB DD Boost demonstrate that the combination of **gpcrondump** backups and Data Domain deduplication technology provides an average backup of 13.08 TB/hour and a restore of 5.90 TB/hour. This enables the backup and recovery of a Greenplum full-rack DCA (36 TB uncompressed database) in 2.75 and 6.1 hours respectively.

- Efficient backup and recovery of the Hadoop Distributed File System (HDFS). The test results for Greenplum HD NFS backups demonstrate that the combination of **DistCp** and Data Domain deduplication technology provides an average backup of 5.37 TB/hour and a restore of 3.62 TB/hour.

The Data Domain DD890 in this solution provides an effective, disk-based backup target that significantly minimizes storage while providing long-term retention.

Introduction

Purpose

The purpose of this white paper is to illustrate how Data Domain's industry-leading technology provides a powerful backup, archiving, and disaster recovery solution that can scale with the most demanding data center requirements, such as data warehousing. This white paper also explores the various practices and considerations for backing up DCA data to the Data Domain system and how to effectively exploit this technology.

Scope

The scope of this white paper is to document:

- DCA features and benefits (at a high level)
- Data Domain system features, as used in this solution
- Installing and configuring the Data Domain system for the DCA
- The solution test objectives, scenarios, and results

Audience

This white paper is intended for:

- Field personnel who are tasked with implementing a backup and recovery solution for the DCA or a data warehouse
- Customers, including IT planners, storage architects, and database administrators involved in evaluating, acquiring, managing, operating, or designing a backup and recovery solution for the DCA or a data warehouse
- EMC staff and partners, for guidance and for the development of proposals

Terminology

Table 1 defines key terms used in this document.

Table 1. Terminology

Term	Definition
Addressable capacity	The amount of physical space available on a Data Domain system to store deduplicated and compressed backup images.
Apache Hadoop	Allows for the distributed processing of large data sets across clusters of computers using a simple programming model. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than relying on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. For more information, visit: http://hadoop.apache.org.

Term	Definition
Business intelligence (BI)	The effective use of information assets to improve the profitability, productivity, or efficiency of a business. IT professionals use this term to refer to the business applications and tools that enable such information usage. The source of information is frequently the data warehouse.
Cumulative compression factor	The ratio of the logical storage size to the physically stored space.
Data Domain	EMC deduplication storage system.
Data Domain Boost (DD Boost)	Extends the backup optimization benefits of Data Domain deduplication storage solutions by distributing parts of the deduplication process to the backup server or application client. DD Boost dramatically increases throughput speeds, minimizes backup LAN load, and improves backup server utilization.
Data warehousing (DW)	The process of organizing and managing information assets of an enterprise. IT professionals often refer to the physically stored data content in some databases managed by database management software as the data warehouse. They refer to applications that manipulate the data stored in such databases as DW applications.
Deduplication	Deduplication is similar to data compression, but it looks for redundancy of very large sequences of bytes across very large comparison windows. Long sequences (greater than 8 KB) are compared to the history of other such sequences and, where possible, the first uniquely stored version of a sequence is referenced rather than stored again. In a storage system, this is all hidden from users and applications, so the whole file is readable after being written. Eliminating redundant data can significantly shrink storage requirements and improve bandwidth efficiency.
GPDB	Greenplum Database.
Greenplum HD	Greenplum HD is a 100-percent open-source certified and supported version of the Apache Hadoop stack. It includes HDFS, MapReduce, Hive, Pig, HBase, and Zookeeper. Greenplum HD's packaged Hadoop distribution removes the pain associated with building out a Hadoop cluster from scratch, which is required with other distributions. Greenplum has also incorporated a pluggable storage layer to Hadoop, enabling customers to exploit the best storage options without requiring changes to their existing applications.

Term	Definition
HDFS	Hadoop Distributed File System (HDFS) is the primary storage system used by Hadoop applications. HDFS creates multiple replicas of data blocks and distributes them on compute nodes throughout a cluster to enable reliable, extremely rapid computations.
Local compression	Standard lossless compression algorithms. The algorithms available on a Data Domain system include **lz** (Lempel-Ziv), **gz**, and **gzfast**: • **lz**: The default algorithm that gives the best throughput. EMC recommends the **lz** option. • **gz**: A zip-style compression that uses the least amount of space for data storage (10 percent to 20 percent less than **lz** on average; however, some datasets get much higher compression). This also uses the most CPU cycles (up to five times as much as **lz**). The **gz** compression type is commonly used for near-line storage applications in which performance requirements are low. • **gzfast**: A zip-style compression that uses less space for compressed data, but more CPU cycles (twice as much as **lz**). **gzfast** is the recommended alternative for sites that want more compression at the cost of lower performance.
Logical storage size	The total size of all backup images in all pools on a Data Domain system. This total size includes all pools mapped to a Data Domain system instance, which can include primary disk pools and clone storage pools.
Massively Parallel Processing (MPP)	A type of distributed computing architecture where tens to hundreds of processors team up to work concurrently to solve large computational problems.
NFS	Network file system.
Redundant Array of Independent Disks (RAID)	A method of organizing and storing data distributed over a set of physical disks, which logically appear to be one single storage disk device to any server host and operating system performing I/O to access and manipulate the stored data. Frequently, redundant data is distributed and stored inside this set of physical disks to protect against loss of data access, should one of the drives in the set fail.
Scale out	A technique that increases the total processing power of a system by adding additional independent computational nodes, as opposed to augmenting a single, large computer with incremental disk, processor, or memory resources.
Shared-Nothing Architecture	A distributed computing architecture made up of a collection of independent, self-sufficient nodes. This is in contrast to a traditional central computer that hosts all information and processing in a single location.

Term	Definition
Storage unit	The Data Domain system exposes pre-made disk volumes called storage units to a DD Boost-enabled media server. Multiple media servers, each with the Data Domain Boost plug-in, can use the same storage unit on a Data Domain system as a storage server.
UAP	Unified Analytics Platform that includes Greenplum Database, Greenplum HD, Greenplum Chorus, and Greenplum DCA.
VLAN overlay	A VLAN overlay is used to separate network traffic from the DCA internal network.

Overview of components

Introduction to the components

This section identifies and briefly describes the components deployed in the solution environment. The components used are:

- EMC Greenplum DCA
- EMC Data Domain deduplication storage system

EMC Greenplum DCA

The DCA is a purpose-built, highly scalable, parallel DW appliance that architecturally integrates database, compute, storage, and network into an enterprise-class, easy-to-implement system. The DCA brings in the power of MPP architecture, delivers the fastest data loading capacity and the best price/performance ratio in the industry without the complexity and constraints of proprietary hardware.

The DCA can also be set up in a UAP configuration that is capable of managing, storing, and analyzing large volumes of structured and unstructured data. Greenplum UAP includes Greenplum Database, Greenplum HD, and Greenplum Chorus.

The DCA is offered in multiple-rack appliance configurations to achieve the maximum flexibility and scalability for organizations faced with terabyte to petabyte scale data opportunities.

This solution uses two of the configuration options available:

- A full-rack DCA with four GPDB modules for GPDB backups
- A full-rack DCA with one GPDB module and three Greenplum HD modules for Greenplum HD and Greenplum Chorus backups

Figure 1 illustrates the architectural layout of the two DCA configurations used.

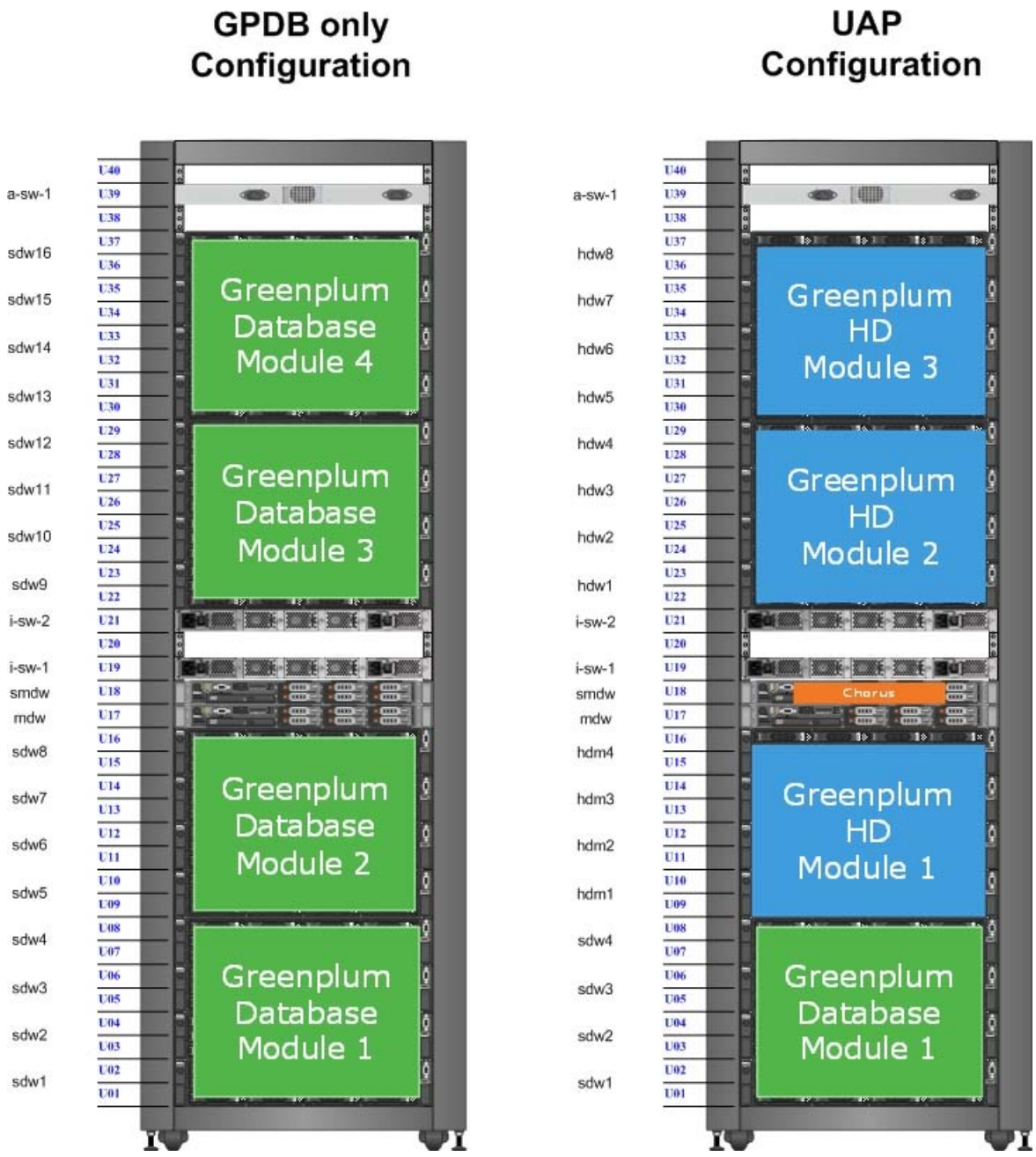

Figure 1. DCA configurations

Table 2 briefly describes the main components of DCA.

Table 2. Main components of DCA

Item	Description
Greenplum Database	Greenplum Database is an MPP database server, based on PostgreSQL open-source technology. It is explicitly designed to support BI applications and large, multi-terabyte data warehouses.
Greenplum Database system	An associated set of Segment Instances and a Master Instance running on an array, which can be composed of one or more hosts.
GPDB Master Servers	The servers responsible for the automatic parallelization of queries.
GPDB Segment Servers	The servers that perform the real work of processing and analyzing the data.
Greenplum HD Master Servers	The servers that perform administration of the Greenplum HD cluster.
Greenplum HD Worker Servers	The servers that perform the real work of processing, storing, and analyzing the data in the Greenplum HD cluster.
Greenplum Chorus	Agile analytics productivity platform.

For more information about the DCA, refer to the following white paper *EMC Greenplum Data Computing Appliance: Architecture, Performance and Functions—A Detailed Review.*

EMC Data Domain deduplication storage system

EMC Data Domain deduplication storage systems dramatically reduce the amount of disk storage needed to retain and protect enterprise data. By identifying redundant data as it is being stored, Data Domain systems reduce backup storage requirements by 10 to 30 times. Backup data can then be efficiently replicated and retrieved over existing networks for streamlined disaster recovery and consolidated tape operations. This allows Data Domain appliances to integrate seamlessly into database architectures, maintaining existing backup strategies with no changes to scripts, backup processes, or system architecture.

The Data Domain appliance is the industry's fastest, most cost-effective and scalable single-controller deduplication storage solution for disk-based backup and network-efficient DR.

The Data Domain Stream-Informed Segment Layout (SISL™) scaling architecture enables the fast-inline deduplication throughput of the Data Domain system. A CPU-centric approach to deduplication delivers a high throughput while minimizing the number of disk spindles required.

Solution architecture

Overview of the DCA and Data Domain solution

This section illustrates the architectural layout of the DCA and Data Domain solution. It also provides details of the hardware and software resources that were used in the solution.

Physical environment

Figure 2 illustrates the architectural layout of the solution used for both DCA configurations.

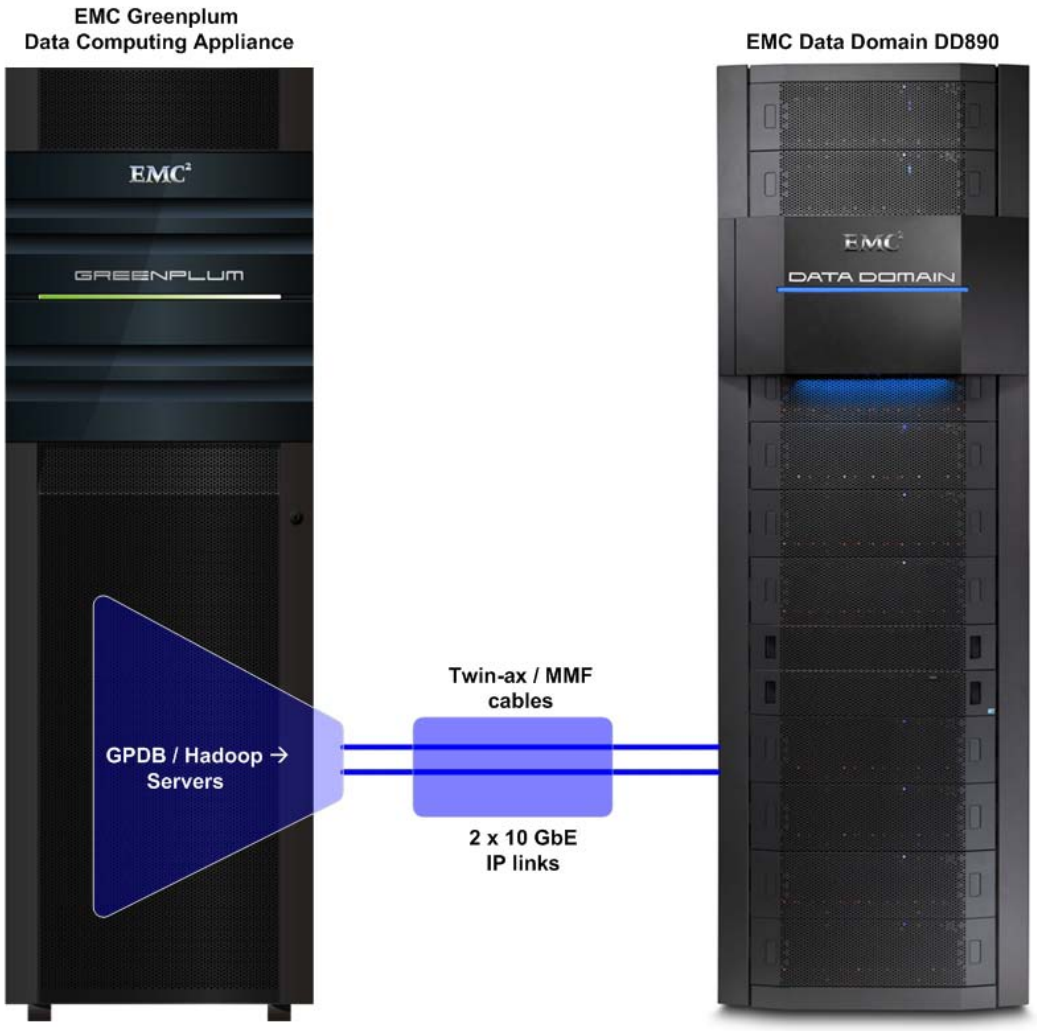

Figure 2. Solution architecture layout

Hardware resources

Table 3 lists the hardware used to validate this solution.

Table 3. Data Domain DD890 system

Equipment	Specification	Quantity
DD890	96 GB memory 4 x internal SATA drives 2 x 1 GbE network interfaces 1 x dual-port 10 GbE optical network interface card	1
Expansion shelves (32 TB)	ES20 disk shelves with 16 x 2 TB SATA disks	6

Table 4 lists the specifications of the GPDB Master Servers used in this solution.

Table 4. GPDB Master Host specifications

Hardware	Specification	Quantity
Processor	3.33 GHz (6-core)	2
Memory	DDR3 1333 MHz	48 GB
Dual-port converged network adapter	2 x 10 Gb/s	1
Quad-port network adapter	4 x 1 Gb/s	1
RAID controller	Dual-channel 6 Gb/s SAS	1
Hard disks	600 GB 10k rpm SAS (one RAID5 volume of 4+1 with one hot spare)	6

Table 5 lists the specifications of the GPDB Segment Servers and Greenplum HD servers used in this solution. For the UAP configuration, we repurposed Segment Servers as Greenplum HD servers.

Table 5. GPDB Segment and Greenplum HD Host specifications

Hardware	Specification	Quantity
Processor	2.93 GHz (6-core)	2
Memory	DDR3 1333 MHz	48 GB
Dual-port converged network adapter	2 x 10 Gb/s	1
Quad-port network adapter	2 x 1 Gb/s	1
RAID controller	Dual-channel 6 Gb/s SAS	1
Hard disks	600 GB 15k rpm SAS (two RAID5 volumes of 5+1 disks)	12

Table 6 details the backup network specification.

Table 6. Backup network

Equipment	Ports	Specification
DCA Interconnect Bus*	2	10 GbE

*The existing Greenplum DCA Interconnect Bus is designed to accommodate the backup network port requirements.

Software resources

Table 7 lists the software used to validate this solution.

Table 7. Software resources

Software	Version	Comment
Data Domain OS	5.1.1.0-291218	Data Domain operating system
Red Hat Enterprise Linux	5.5	DCA Servers operating system
EMC Greenplum Database	4.2.1.0	4.2.2.0 was used for tests with Direct I/O
Greenplum HD	1.1	Greenplum HD Modules
Greenplum Chorus	2.2	Greenplum Chorus for collaborative analytics
Fabric OS	6.3.2a1	CEE/FCoE network

Data Domain system features, integration, and administration

Introduction to Data Domain systems

EMC Data Domain deduplication storage systems provide a next-generation backup and recovery solution for big data that allows users to enjoy the retention and recovery benefits of inline deduplication as well as the offsite disaster recovery protection of replication over the wide area network (WAN). Data Domain systems reduce the amount of disk storage needed to retain and protect data by 10x to 30x. Data on disk is available online and onsite for longer retention periods, and restores become fast and reliable. Storing only unique data on disk also means that data can be cost-effectively replicated over existing networks to remote sites for DR.

EMC further extends these benefits through EMC Data Domain Boost software (DD Boost). DD Boost enables advanced integration between Data Domain systems and Greenplum Databases for faster, more efficient backup and recovery.

This section also provides details on Data Domain system integration and administration.

Faster, more efficient backup

- Distributed deduplication process dramatically increases throughput
- Reduced network bandwidth utilization

Network-efficient replication

- Cost-efficient disaster recovery
- Encrypted replication
- Up to 99 percent bandwidth reduction
- Faster “time-to-DR” readiness

Seamless integration

- Configured using native Greenplum Database backup and restore utilities

Ultra-safe storage for fast and reliable recovery

- Data Invulnerability Architecture
- Continuous recovery verification, fault detection, and healing
- End-to-end data integrity

Scalable deduplication storage

EMC Data Domain is the industry’s fastest deduplication storage system for enterprise backup and archiving workloads. With a throughput of up to 31 TB/hour, Data Domain systems can protect up 28.5 petabytes of logical capacity, enabling more backups to complete sooner while putting less pressure on limited backup windows.

Easy integration

Data Domain is qualified with all leading enterprise backup software and archiving applications. It easily integrates into existing software infrastructures without change for either data center or distributed office data protection.

Data Domain systems integrate easily into existing data centers. All Data Domain systems can be configured as storage destinations for leading backup and archiving applications using NFS, common internet file system (CIFS), Data Domain Boost, or virtual tape library (VTL) protocols. Consult the compatibility matrices for information about the applications that work with the different configurations. Multiple backup servers can share one Data Domain system.

Integration into an existing Greenplum DCA

The DCA architecture is designed to help you easily integrate Data Domain systems in a nondisruptive, seamless manner. You can reserve Port 19 in each of the Interconnect switches for Data Domain connectivity. All that is required is to connect the Data Domain system directly into the DCA environment and start the configuration steps for NFS or DD Boost. Refer to the Data Domain and DCA connectivity section for more options.

Operational simplicity

Data Domain systems are simple to install and manage. Connect an appliance to the backup server either as a file server via Ethernet or as a VTL via Fibre Channel. All three interfaces can be used simultaneously. Data Domain Boost is also compatible with many other backup applications. For more information, see the EMC Data Domain Boost Compatibility Matrix at the Data Domain support portal.

Data integrity

The Data Domain Data Invulnerability Architecture provides ultra-safe storage for reliable recovery and continuous protection. It provides the industry's best defense against data integrity issues. Continuous recovery verification, along with extra levels of data protection, continuously detect and protect against data integrity issues during the initial backup and throughout the data life cycle. Unlike any other enterprise array or file system, each appliance ensures recoverability is verified and then continuously re-verified.

The Data Domain operating system (DD OS) includes extra levels of data protection to protect itself against storage-related faults that threaten data recoverability. Dual disk parity RAID 6 is part of the foundation for continuous fault detection and healing on DD OS. RAID 6 protects against two simultaneous disk faults, can rebuild a failed disk even if there are read errors on other sectors, and can detect and correct errors on-the-fly during reads. This added protection ensures the highest levels of data availability.

In determining global uniqueness, DD OS leverages very strong cryptographic hashing for speed and security. But it does not stop there—a universal hash ensures against random and malicious hash collisions. An append-only write policy guards against overwriting valid data.

After a backup is completed, a validation process looks at what was written to disk to check that all file segments are logically correct within the file system and that the data is the same on the disk as it was before being written to disk. In the background, the Online Verify operation continuously checks that the data on the disks is correct and unchanged since the earlier validation process.

The back-end storage is set up in a double parity RAID 6 configuration (two parity drives). Additionally, hot spares are configured within the system. Each parity stripe has block checksums to ensure that the data is correct. The checksums are constantly used during the online verify operation and when data is read from the

Data Domain system. With double parity, the system can fix simultaneous errors on up to two disks.

To keep data synchronized during a hardware or power failure, the Data Domain system uses non-volatile RAM (NVRAM) to track outstanding I/O operations. An NVRAM card with fully-charged batteries (the typical state) can retain data for a minimum of 48 hours.

When reading data back on a restore operation, the DD OS uses multiple layers of consistency checks to verify that restored data is correct.

Data compression

The DD OS stores only unique data. Through Global Compression™, a Data Domain system pools redundant data from each backup image. The storage of unique data is invisible to backup software, which sees the entire virtual file system. DD OS data compression is independent of a data format. Data can be structured, such as databases, or unstructured, such as text files. Data can be from file systems or raw volumes.

Typical compression ratios are 20:1 on average over many weeks. This assumes weekly full and daily incremental backups. A backup that includes many duplicate or similar files (files copied several times with minor changes) benefits the most from compression. Depending on backup volume, size, retention period, and rate of change, the amount of compression can vary.

The best compression happens with backup volume sizes of at least 10 mebibytes (MiB—a unit of data storage that is exactly 1,048,576 bytes, the base 2 equivalent of MB). To take full advantage of multiple Data Domain systems, a site that has more than one Data Domain system should consistently back up the same client system or set of data to the same Data Domain system. For example, if a full backup of all sales data goes to Data Domain system A, the incremental backups and future full backups for sales data should also go to Data Domain system A.

A Data Domain system compresses data at two levels:

- **Global compression**–compares received data to data already stored on disk. Duplicate data does not need to be stored again, while new data is locally compressed before being written to disk.
- **Local compression**–a Data Domain system uses a local compression algorithm developed specifically to maximize throughput as data is written to disk. The default algorithm (**lz**) allows shorter backup windows for backup jobs but uses more space. Local compression options provide a trade-off between performance and space usage.

For more information on how to change compression, refer to the *Data Domain Operating System (DD OS) Administration Guide*. Changing the algorithm immediately affects any new data written to the system. Any data already stored on the system will be recompressed during the next cleaning run, which may take much longer to run than usual.

SISL

Data Domain SISL enables high throughput, inline deduplication. SISL identifies 99 percent of the duplicate segments in RAM, inline, before storing to disk. In addition, it stores related segments and fingerprints together, so large groups can be read at once. With these patented techniques, Data Domain can utilize the full capacity of large SATA disks for data protection and minimize the number of disks needed to deliver high throughput. In the long term, SISL allows dramatic Data Domain system performance improvements as CPU speeds increase.

Multipath and load-balancing configuration

Data Domain systems that have at least two 10 GbE ports can support multipath configuration and load balancing. In a multipath configuration on the Data Domain system, each of the two 10 GbE ports on the system is connected to a separate port on the backup server.

DD Boost

EMC Data Domain Boost significantly increases performance by distributing parts of the deduplication process to the backup server, simplifies disaster recovery procedures, and serves as a solid foundation for additional integration between backup applications and Data Domain systems.

Design considerations

Retention of data, frequency, rate of change, and backup policies influence the decision when determining the amount of storage required in the Data Domain system. For this solution, the initial capacity was chosen to accommodate a simulated 10 weeks of backup of the DCA.

Data Domain Enterprise Manager

All Data Domain systems run the DD OS, which includes Data Domain Enterprise Manager, a simple web-based rich Internet application for managing Data Domain systems. DD System Manager provides both a GUI and a command line interface (CLI) for configuration management and monitoring all system operations. The web-based GUI, available through Ethernet connections, can manage up to 20 Data Domain systems (depending on the model) at any location. DD System Manager provides a single, consolidated management interface that allows for the configuration and operation of many system features and settings.

DD System Manager also provides real-time graphs and tables that enable users to monitor the status of system hardware components and configured features. Additionally, a command set that performs all system functions is available to users through the CLI. Commands configure system settings and provide displays of system hardware status, feature configuration, and operation.

The CLI is available through a serial console when a keyboard and monitor are directly attached to the Data Domain system, or remotely through an Ethernet connection using SSH or Telnet. For more information on Data Domain EnterpriseManager, refer to the *Data Domain Operating System (DD OS) Administration Guide*.

Data Domain file system

Data Domain systems are designed to be a highly reliable “storage of last resort” to provide longer-term onsite retention of backups. As new backups are added to the system, old backups are aged out. Such removals are normally done under the control of backup software (on the backup server) based on the configured retention

period. This process is similar to configuring tape retention policies in which older backups are retired and the tapes are reused for new backups.

When backup software removes an old backup from a Data Domain system, the space on the Data Domain system becomes available only after the Data Domain system cleans the retired disk space. A good way to manage space on a Data Domain system is to retain as many online backups as possible, with some empty space (about 20 percent of the total space available) to comfortably accommodate backups until the next scheduled cleaning run.

Space utilization on a Data Domain system is primarily affected by:

- The backup policy and redundancy in the data
- The size, redundancy, and rate of change of the backup data
- The retention period specified in the backup software

High levels of compression result when backing up datasets with many duplicates and retaining them for long periods of time.

The Data Domain file system supports the following interfaces:

- NFS
- CIFS
- Data Domain Boost
- VTL

For more information on the file system, refer to the *Data Domain Operating System (DD OS) Administration Guide*.

Creating and working with shares on the Data Domain system

When creating shares, assign client access to each directory separately and remove access from each directory separately.

Note If replication is to be implemented, a single destination Data Domain system can receive backups from both CIFS clients and NFS clients as long as separate directories are used for each. Do not mix CIFS and NFS data in the same directory.

Greenplum Chorus

Greenplum Chorus is a collaboration tool that data science teams use to share a variety of information from the data they work on. When deployed, Greenplum Chorus provides an agile and analytic infrastructure for data science teams to participate and collaborate on data sets, methods, and workflows, providing valuable insights to each other.

Greenplum Chorus maintains its own database. To protect this valuable data, this solution includes the backup of Greenplum Chorus to Data Domain.

For this solution, Greenplum Chorus is installed on the DCA Standby Master Server (smdw). For more information, refer to the *Greenplum Chorus 2.2 Installation Guide*.

Installing and configuring the Data Domain system with the DCA

Overview

Data Domain storage systems integrate seamlessly into existing DCA deployments. Data Domain system storage can be exported as NFS shares that can easily be mounted on the DCA GPDB and Greenplum HD Servers and utilized as a backup target. With Greenplum Database support for DD Boost, the Data Domain system also integrates easily in this case. Depending on the DCA configuration, other Data Domain system models are also supported, for example, DD670, DD860, and DD990. To determine the correct Data Domain model, your local Backup and Recovery Specialist (BRS) should complete a sizing exercise. For reference information on compatibility matrices, visit the Data Domain Support Portal my.datadomain.com. Although the DD990 would also have been suitable, the DD890 was used in this solution. Connectivity to the DCA was via NFS or using Data Domain Boost—both configurations were tested.

Installing the Data Domain system

Install the Data Domain hardware, including a system console, as described in the *Data Domain Installation and Setup Guide*, which is shipped with the Data Domain system. The *Installation and Setup Guide* provides instructions for installing the Data Domain system, connecting it to an administrative console, and powering it on. After completing the installation and powering on the system, refer to the *Data Domain Operating System (DD OS) Initial Configuration Guide* for additional information.

Configuring the Data Domain system

When the installation is complete, the Data Domain Configuration Wizard starts automatically. The Configuration Wizard performs an "initial" configuration of the Data Domain system, configuring only what is needed for the most basic system setup, including licenses, network, file system, CIFS, and NFS. For more information about using the Configuration Wizard and changing or updating the configuration, refer to the *Data Domain Operating System (DD OS) Initial Configuration Guide*.

Data Domain and DCA connectivity

Two options are available when connecting the Data Domain system to the DCA:

- Direct connection via Port 19 on both the DCA Interconnect switches.
- Alternatively, Ports 18 and 19 can be configured as a two-port link aggregation on both the DCA Interconnect switches. A separate switch can then be connected to increase the connectivity options. In this case, if Ports 18 and 19 are not configured in a link aggregation group (LAG), they will act as independent links and can be configured for active or passive redundancy.

Note By default, Ports 18 and 19 are set up in a two-port link aggregation. To convert Ports 18 and 19 to switch ports, refer to the Supporting information section.

A VLAN overlay can also be used to separate network traffic from the DCA internal network. For more information on the DCA physical connectivity options and setting up a VLAN overlay, see:

- *Greenplum Data Computing Appliance Getting Started Guide*
- *Greenplum Data Computing Appliance Administration Guide*

For this solution, the connection was made using two Multi-Mode Fibre (MMF) cables through optical Small Form-Factor Pluggable (SFP) devices, as shown in Table 8.

Table 8. DCA and Data Domain connectivity

DCA Interconnect Bus	Data Domain system interface
i-sw-1, Port 19	Slot 4, Port 1/eth4a
i-sw-2, Port 19	Slot 4, Port 2/eth4b

The hardware used for this connectivity option is listed in Table 9.

Table 9. Hardware used for DCA and Data Domain connectivity

Hardware	Part number	Quantity
Data Domain:10 GbE card	C-10G-L2PO	1
10 GbE SFP	BRSFP-10GSW1P	2
OM3 multimode optical cable	CTX-OM3-10M	2

A connectivity kit that consists of the cables and required SFPs is also available under part number DCA1-10GBEXT.

Depending on distance, connectivity can be provided through either MMF cables and optical SFP, or twin-ax cables. For more information about options and internal connectivity, refer to the white paper *EMC Greenplum Data Computing Appliance: Architecture, Performance and Functions—A Detailed Review*.

GPDB, NFS: Configuring the Data Domain system and the DCA

For NFS backups, the Data Domain system must be configured to enable the DCA to access the NFS directories that are used as a target for the **gpcrondump** utility to store the backups.

The following example shows how the system was configured using two 10 GbE interfaces with the IP addresses 172.28.8.209 and 172.28.12.209, respectively:

On Data Domain:

```
sysadmin@dd_890_1# net show settings
port    enabled   DHCP   IP address     netmask         type  additional setting
-----   -------   ----   -------------  -------------   ----  ------------------
eth0a   yes       no     172.28.4.209   255.255.248.0   n/a
eth0b   no        n/a    n/a            n/a             n/a
eth4a   yes       no     172.28.8.209   255.255.252.0   n/a
eth4b   yes       no     172.28.12.209  255.255.252.0   n/a
eth5a   no        n/a    n/a            n/a             n/a
eth5b   no        n/a    n/a            n/a             n/a
-----   -------   ----   -------------  -------------   ----   -----------------
* Value from DHCP
```

It is very important also to configure the host files on both the DCA and the Data Domain system. The following example shows how the Data Domain host entries look on the Master and Segment Servers.

On the DCA:

```
[gpadmin@mdw ~]$ grep "Data Domain" /etc/hosts
### Data Domain - Backup
172.28.8.209    dd_889_1_1    # Data Domain interface eth4a, Network 172.28.8.0
172.28.12.209   dd_890_1_2    # Data Domain interface eth4b, Network 172.28.12.0
```

To add the hosts to the Data Domain system, type:

```
net hosts add <ipaddr> <host-list>
```

The following example adds both 10 GbE interfaces of the host mdw:

On the DCA:

```
sysadmin@dd_890_1# net hosts add 172.28.8.204 mdw mdw1-1
Added "mdw mdw1-1" -> "172.28.8.204" mapping to hosts list.
sysadmin@dd_890_1# net hosts add 172.28.12.204 mdw1-2
Added "mdw1-2" -> "172.28.12.204" mapping to hosts list
```

Repeat with both 10 GbE interfaces for all Master and Segment Servers.

Perform the following steps to configure NFS:

1. Add the NFS share **/backup** and set the access level to the DCA network.

   The network used in this case is **172.28.8.0/22** and **172.28.12.0/22**.

   On Data Domain:

   ```
   sysadmin@dd_890_1# nfs add /backup 172.28.8.0/22,172.28.12.0/22
   ```

   NFS export for **/backup** is added.

2. Ensure that the network was added correctly as shown in the following example:

   On Data Domain:

   ```
   sysadmin@dd_890_1# nfs show clients
   path            client          options
   -------------   -------------   ---------------------------------------
   /backup         172.28.8.0/22   (rw,no_root_squash,no_all_squash,secure)
   /backup         172.28.12.0/22  (rw,no_root_squash,no_all_squash,secure)
   ```

3. Enable NFS on the Data Domain system as follows:

   On Data Domain:

   ```
   sysadmin@dd_890_1# nfs enable
   ```

   NFS server is enabled.

4. On the DCA Master Server, mount the **/backup** NFS as shown in the following example:

   On the DCA:

   a. Create a temporary mount point for the **/backup**.

      ```
      [root@mdw ~]# mkdir /backup_tmp
      [root@mdw ~]# ls -la /backup_tmp
      ```

```
total 12
drwxr-xr-x  2 root root 4096 Aug 18 13:57 .
drwxr-xr-x 30 root root 4096 Aug 18 13:57 ..
```

b. Mount the **/backup** directory on the Master Server (all one line):

```
[root@mdw ~]# mount -t nfs –o
hard,intr,nfsvers=3,tcp,rsize=32768,wsize=32768
dd_890_1_1:/backup /backup_tmp/
```

5. Create the DCA directory and all the server subdirectories.

Note In this case, use the user **gpadmin** to avoid any issues with permissions while running the **gpcrondump** utility.

In the following example, the directory **DCA-01** and the subdirectories **Master01**, **Master02**, and **Seg01** to **Seg16** were created using **mkdir** as follows (all one line):

On the DCA:

```
[gpadmin@mdw ~]$ mkdir /backup_tmp/DCA-01;mkdir
/backup_tmp/DCA-01/Master{01,02};mkdir /backup_tmp/DCA-
01/Seg{01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16}
```

After all the directories are created, unmount the /**backup_tmp** directory as follows:

On the DCA:

```
[root@mdw ~]# umount /backup_tmp
```

The **gpcrondump** utility creates a directory named **db_dumps** when the backup is started for the first time.

Each server should use a different mount point for this purpose.

We created an NFS share for each server on the Data Domain and the corresponding mount point created on each server was **/backup/DCA-01**. Table 10 provides an overview of the mapping of the directories in this solution.

Note: A single NFS share on Data Domain can also be used by all Master and Segment Servers. A single NFS share is recommended if the gpdbrestore option to restore individual tables from full backups will be required.

Table 10. NFS shares and mount points

DCA server	Host name	NFS share on Data Domain	Mount point on DCA
Master Server 01	mdw	/backup/DCA-01/Master01	/backup/DCA-01
Master Server 02	smdw	/backup/DCA-01/Master02	/backup/DCA-01
Segment Server 01	sdw1	/backup/DCA-01/Seg01	/backup/DCA-01
Segment Server 02	sdw2	/backup/DCA-01/Seg02	/backup/DCA-01
Segment Server 03	sdw3	/backup/DCA-01/Seg03	/backup/DCA-01
Segment Server 04	sdw4	/backup/DCA-01/Seg04	/backup/DCA-01

DCA server	Host name	NFS share on Data Domain	Mount point on DCA
Segment Server 05	sdw5	/backup/DCA-01/Seg05	/backup/DCA-01
Segment Server 06	sdw6	/backup/DCA-01/Seg06	/backup/DCA-01
Segment Server 07	sdw7	/backup/DCA-01/Seg07	/backup/DCA-01
Segment Server 08	sdw8	/backup/DCA-01/Seg08	/backup/DCA-01
Segment Server 09	sdw9	/backup/DCA-01/Seg09	/backup/DCA-01
Segment Server 10	sdw10	/backup/DCA-01/Seg10	/backup/DCA-01
Segment Server 11	sdw11	/backup/DCA-01/Seg11	/backup/DCA-01
Segment Server 12	sdw12	/backup/DCA-01/Seg12	/backup/DCA-01
Segment Server 13	sdw13	/backup/DCA-01/Seg13	/backup/DCA-01
Segment Server 14	sdw14	/backup/DCA-01/Seg14	/backup/DCA-01
Segment Server 15	sdw15	/backup/DCA-01/Seg15	/backup/DCA-01
Segment Server 16	sdw16	/backup/DCA-01/Seg16	/backup/DCA-01

Note The mount point can be created on each DCA using the **gpssh** utility. The **/backup** directory is created by the **root** user but **/backup/DCA-01** must be created by the **gpadmin** user.

The following example describes one way of creating the mount points. For more options, refer to the *Data Domain Operating System (DD OS) Administration Guide*.

On the DCA:

```
[root@mdw ~]# cd ~gpadmin/gpconfigs

[root@mdw gpconfigs]# gpssh -f hostfile_gpdb
=> mkdir /backup
=> mkdir /backup/DCA-01
=> chown gpadmin:gpadmin /backup/DCA-01
=> exit
```

After the mount points are created, mount the Data Domain NFS shares according to Table 10. The following command provides an example:

On the DCA:

```
[gpadmin@mdw ~] $ gpssh -h sdw1 -v -e mount -t nfs –o
rw,hard,tcp,rsize=1048576,wsize=1048576 dd_890_1_1:/backup/DCA-
01/Seg01 /backup/DCA-01
```

To get the best balance across both Data Domain 10 GbE interfaces (eth4a and eth4b), servers mdw and sdw1 to sdw8 should use host **dd_890_1_1** (network 172.28.8.0, interface eth4a), while smdw and sdw9 to sdw16 should use host **dd_890_1_2** (network 172.28.12.0, interface eth4b).

Check that the directories are mounted correctly as follows:

On the DCA:

```
[gpadmin@mdw ~]$ for NODE in `cat ~/shosts`; do gpssh -h $NODE -v
-e df -h /backup/DCA-01 ; done;

[Reset ...]
[INFO] login sdw1
[sdw1] Filesystem            Size  Used Avail Use% Mounted on
[sdw1] dd_890_1_1:/backup/DCA-01/Seg01
[sdw1]                       130T  4.9T  125T   4% /backup/DCA-01
[INFO] completed successfully

[Cleanup...]

[Reset ...]
[INFO] login sdw2
[sdw2] Filesystem            Size  Used Avail Use% Mounted on
[sdw2] dd_890_1_1:/backup/DCA-01/Seg02
[sdw2]                       130T  4.9T  125T   4% /backup/DCA-01
[INFO] completed successfully

[Cleanup...]

[ some text removed ]

[Reset ...]
[INFO] login sdw16
[sdw16] Filesystem            Size  Used Avail Use% Mounted on
[sdw16] dd_890_1_2:/backup/DCA-01/Seg16
[sdw16]                       130T  4.9T  125T   4% /backup/DCA-01
[INFO] completed successfully

[Cleanup...]
```

If all the NFS shares are mounted successfully, the Data Domain system is ready to receive any backups run by the **gpcrondump** utility.

GPDB, NFS: Backing up to Data Domain from the DCA

To automate routine NFS backups, Greenplum provides the **gpcrondump** utility, which can be called directly or from a crontab entry. The **gpcrondump** utility is a wrapper for the **gp_dump** command. **gpcrondump** also enables the backup of objects other than databases and data, such as database roles and server configuration files.

By default, **gpcrondump** creates data dump files in the Master and each Segment instance's data directory in **‹data_directory›/db_dumps/YYYYMMDD**. By default, the segment data dump files are compressed using **gzip**.

For NFS backups, the **gpcrondump** utility with the **-u** parameter is required. Without the **-u** parameter, the backups are stored locally in the DCA array.

For the NFS solution, the intention was to send the backups to the Data Domain system so the directory used was the Data Domain NFS share created in **/backup/DCA-01** as follows:

On the DCA:

```
[gpadmin@mdw db_dumps]$ pwd
```

```
/backup/DCA-01/db_dumps
[gpadmin@mdw db_dumps]$ ls -la
total 12
drwxrwxr-x 7 gpadmin gpadmin  382 Aug 20 15:01 .
drwxrwxr-x 4 gpadmin gpadmin  158 Aug 16 17:10 ..
drwx------ 2 gpadmin gpadmin 1342 Aug 16 19:37 20100816
drwx------ 2 gpadmin gpadmin 2608 Aug 17 19:19 20100817
drwx------ 2 gpadmin gpadmin 1563 Aug 19 23:57 20100819
drwx------ 2 gpadmin gpadmin 3328 Aug 20 15:21 20100820
```

We generated all the NFS backups by running the following command as **gpadmin** user:

For **gpcrondump** uncompressed backups:

```
gpcrondump -x tpch -g -G -u /backup/DCA-01 -b –a –z
```

Log files for the Greenplum Database management utilities are written to ~/gpAdminLogs by default. The naming convention for management log files is:

`<script_name>_<date>.log`

For example: `gpcrondump_<date>.log`

Table 11 lists the gpcrondump options used in NFS.

Table 11. NFS – gpcrondump options used

Parameter	Description
-x	Database name
-g	Copy config files
-G	Dump global objects
-u	Backup target directory
-a	Do not prompt
-b	Bypass disk space check
-z	Do not use compression

Note The **gpcrondump** compressed options **--rsyncable** are also available, but may not be suitable for environments with a high rate of random data changes. To gain the maximum benefit from EMC Data Domain deduplication technology, the **gpcrondump** uncompressed option **–z** is recommended.

You can automate parallel backups with **gpcrondump** by adding an entry to the crontab of the Master Server:

```
SHELL=/bin/bash
GPHOME=/usr/local/greenplum-db- 4.2.0.1
MASTER_DATA_DIRECTORY=/data/gpdb_master/Data-1
PATH=$PATH:$GPHOME/bin
01 0 * * * gpadmin gpcrondump -x tpch -c -g -G -a -q >>
gpcrondump_Backup.log
```

For more information, refer to the *Greenplum Database 4.2 Administrator Guide*.

GPDB, NFS: Backups with Direct I/O

In some cases, GPDB backups over NFS can fail when the backup target becomes overwhelmed with the rate the DCA is sending data over the network. GPDB 4.1.2.3, 4.2.2.0, and versions above enable more control over the data size and writing mechanism to the NFS backup target using Direct I/O. In standard operating conditions, file systems use memory for file cache and databases use memory for database cache. With Direct I/O enabled, the file system cache is bypassed and data is transferred directly from disk to the application buffer. This effectively avoids the double caching effect and reduces CPU utilization.

Note This feature is only supported with Red Hat Enterprise Linux platforms for GPDB backups over NFS.

To enable Direct I/O, enter:

```
$ gpconfig -c gp_backup_directIO -v on
```

To activate the configuration setting, enter:

```
$ gpstop -u
```

To verify Direct I/O is enabled, enter:

```
$ gpconfig –s gp_backup_directIO
```

To set the chunk size to 10 MB, enter:

```
$ gpconfig -c gp_backup_directIO_read_chunk_mb -v 10
```

To activate the configuration setting, enter:

```
$ gpstop -u
```

To verify the data chunk size, enter:

```
$ gpconfig –s gp_backup_directIO_read_chunk_mb
```

The default chunk size is 20 which should be optimal for most environments.

GPDB, NFS: Restoring data on the DCA from Data Domain

For restoring a database, Greenplum provides the **gpdbrestore** utility, which is a wrapper for the **gp_restore** command.

The prerequisites for restoring data using **gpdbrestore** are:

- Backup files are created by a **gpcrondump** operation.
- The Greenplum Database system is up and running.
- The Greenplum Database system has the exact same number of primary segment instances as the system that was backed up.
- The database being restored to is already created in the system.

Note **gpdbrestore** is the powerful and relatively new wrapper for the **gp_restore** command and is still under active qualification. This solution uses the

simpler **gp_restore** utility. This is adequate for most customer needs, as shown in the following example.

The NFS shares on the Data Domain should be available and mounted on both Master Servers and on each Segment Server for the restore to proceed.

For example, to restore the **tpch** database backup generated by the **gpcrondump** on **20/08/2010**:

1. Ensure that the Data Domain system can be reached from the Master Server:

   On the DCA:

   ```
   [gpadmin@mdw ~]$ ping dd_890_1_1
   PING dd_890_1_1 (172.28.8.209) 56(84) bytes of data.
   64 bytes from dd_890_1_1 (172.28.8.209): icmp_seq=1 ttl=64
   time=0.105 ms
   64 bytes from dd_890_1_1 (172.28.8.209): icmp_seq=2 ttl=64
   time=0.103 ms
   [gpadmin@mdw ~]$ ping dd_890_1_2
   PING dd_890_1_2 (172.28.12.209) 56(84) bytes of data.
   64 bytes from dd_890_1_2 (172.28.12.209): icmp_seq=1 ttl=64
   time=0.755 ms
   64 bytes from dd_890_1_2 (172.28.12.209): icmp_seq=2 ttl=64
   time=0.109 ms
   ```

2. Ensure that the database being restored to has already been created in the system:

   On the DCA:

   ```
   [gpadmin@mdw ~]$ createdb tpch
   ```

   This command fails if the database already exists.

3. Ensure that the database contains no objects as follows:

   On the DCA:

   ```
   [gpadmin@mdw ~]$ psql -d tpch
   psql (8.2.14)
   Type "help" for help.

   tpch=# \d
   No relations found.
   ```

4. Run the **gp_restore** utility:

   On the DCA:

   ```
   [gpadmin@mdw ~]$ gp_restore -d tpch  --gp-d=/backup/DCA-
   01/db_dumps/20100820 --gp-k=20100820120142
   ```

   Log files for the Greenplum Database management utilities are written to ~/gpAdminLogs by default. The naming convention for management log files is:

   <script_name>_<date>.log

   For example: *gpdbrestore_<date>.log*

Table 12 lists the gp_restore options used in NFS.

Table 12. NFS – gp_restore options used

Parameter	Description
-d	The database name
--gp-d	The backup set path
--gp-k	The timestamp (key) of the backup set to be restored

For more options, see the *Greenplum Database 4.2 Administrator Guide*.

GPDB, DD Boost: Configuring Data Domain and the DCA

For DD Boost backups, the Data Domain system must be configured to enable the DCA to access the Data Domain storage unit that is used as the target for the **gpcrondump** utility to store the backups.

The following example shows how the system was configured using two 10 GbE interfaces with the IP addresses 172.28.8.209 and 172.28.12.209, respectively:

On Data Domain:

```
sysadmin@dd_890_1# net show settings
port    enabled   DHCP   IP address     netmask         type  additional setting
-----   -------   ----   -------------  -------------   ----  ------------------
eth0a   yes       no     172.28.4.209   255.255.248.0   n/a
eth0b   no        n/a    n/a            n/a             n/a
eth4a   yes       no     172.28.8.209   255.255.252.0   n/a
eth4b   yes       no     172.28.12.209  255.255.252.0   n/a
eth5a   no        n/a    n/a            n/a             n/a
eth5b   no        n/a    n/a            n/a             n/a
-----   -------   ----   -------------  -------------   ----   -----------------
* Value from DHCP
```

Also, it is very important to configure the host files on both the DCA and on the Data Domain system. The following example shows how the Data Domain host entries look on the Master and Segment Servers:

On the DCA:

```
[gpadmin@mdw ~]$ grep "Data Domain" /etc/hosts
### Data Domain - Backup
172.28.8.209    dd_889_1_1    # Data Domain interface eth4a,
Network 172.28.8.0
172.28.12.209   dd_890_1_2    # Data Domain interface eth4b,
Network 172.28.12.0
```

To add the hosts to the Data Domain system, type:

```
net hosts add <ipaddr> <host-list>
```

The following example adds both 10 GbE interfaces of the host mdw:

On Data Domain:

```
sysadmin@dd_890_1# net hosts add 172.28.8.204 mdw mdw1-1
Added "mdw mdw1-1" -> "172.28.8.204" mapping to hosts list.
sysadmin@dd_890_1# net hosts add 172.28.12.204 mdw1-2
```

```
Added "mdw1-2" -> "172.28.12.204" mapping to hosts list
```

Repeat these instructions on both 10 GbE interfaces for all Master and Segment Servers.

Perform the following steps to configure DD Boost:

1. Add DD Boost license.

   On Data Domain:

   ```
   sysadmin@dd_890_1# license show
   Feature licenses:
   ##   License Key           Feature
   --   -------------------   -----------
   1    XXXX-YYYYY-XXXX-YYYY   DDBOOST
   --   -------------------   -----------
   ```

   If necessary, use the following command to add the license:

   On Data Domain:

   ```
   sysadmin@dd_890_1# license add <license-code>
   ```

   Where ‹license-code› is the DD Boost license key.

2. Enable NFS on the Data Domain system as follows:

   Note: The access list for the DD Boost clients is an NFS export, so NFS must be enabled.

   On Data Domain:

   ```
   sysadmin@dd_890_1# nfs enable
   NFS server is enabled
   ```

3. Enable DD Boost access for the DCA Master and Segment Servers

   The following example enables DD Boost access for mdw:

   On Data Domain:

   ```
   sysadmin@dd_890_1# ddboost access add clients mdw mdw-1 mdw-
   2

   mdw : Added
   mdw-1 : Added
   mdw-2 : Added
   ```

   Repeat this step for all Master and Segment Servers.

   Ensure that all Master and Segment Servers were added correctly, as shown in the following example:

   On Data Domain:

   ```
   sysadmin@dd_890_1# ddboost access show
   DD Boost access allowed from the following clients:
   mdw
   mdw-1
   mdw-2
   ```

```
sdw1
sdw1-1
sdw1-2
……output is truncated……
sdw16
sdw16-1
sdw16-2
```

4. Create a DD Boost user.

   By default, the DD Boost user is **sysadmin**. If an individual user is required for DD Boost, refer to the *Data Domain Operating System (DD OS) Administration Guide*. This solution uses the **sysadmin** user.

5. Set up the DD Boost credentials to give the DCA Master and Segment servers login access to Data Domain for DD Boost backups.

   Use option a or b only:

   a. For GPDB versions **above 4.2.0.0 but** prior to 4.2.2.0:

   Create a credentials file

   On the DCA as the **gpadmin** user, create a configuration file named .ddconfig in the **/home/gpadmin** directory of the Master Server (mdw). The file should have three lines using the parameters shown in Table 13.

   Table 13. Parameters for the credentials file

Parameter	Description
Host name	IP address of DD target interface (10 GbE)
User name	Data Domain Boost User Name
Password	Data Domain Boost Login Password

   The following three lines show the content of a sample .ddconfig file:

```
172.28.8.209
sysadmin
pa55w0rd#
```

   Copy the credentials file to the Segment Servers

   To get the best balance across both Data Domain 10 GbE interfaces (eth4a and eth4b), servers mdw and sdw1 to sdw8 should use the 172.28.8.0 network (interface eth4a), while smdw and sdw9 to sdw16 should use the 172.28.12.0 network (interface eth4b).

   For example:

To backup via eth4a:	To backup via eth4b:
172.28.8.209 sysadmin PA55W0RD#	172.28.12.209 sysadmin PA55W0RD#

   Use the **gpscp** command to distribute the appropriate DD Boost credentials file from the Master Server to the secondary Master and Segment Servers.

For example:

```
$ gpscp -f hostfile /home/gpadmin/.ddconfig_
=:/home/gpadmin/.ddconfig
```

b. For GPDB versions above 4.2.2.0:

Use the gpcrondump command to setup the login credentials.
This process will create the encrypted login credentials on the Master and Segment servers to allow access to the Data Domain system.

```
For one Data Domain network interface:

gpcrondump --ddboost-host <ddboost_hostname> --ddboost-user
<ddboost_user>

For two Data Domain network interfaces:

gpcrondump --ddboost-host <ddboost_hostname1> --ddboost-host
<ddboost_hostname2> --ddboost-user <ddboost_user>
```

For example:

On the DCA:

```
[gpadmin@mdw ~]$ gpcrondump --ddboost-host dd_890_1_1 --ddboost-
host dd_890_1_2 --ddboost-user sysadmin
Password:
```

Enter the password for user **sysadmin**.

Example of the output from one server:

```
 [sdw16] 20121118:20:55:50|ddboost-[DEBUG]:-Libraries were loaded
successfully
[sdw16] 20121118:20:55:50|ddboost-[INFO]:-creating LB on
/home/gpadmin/DDBOOST_CONFIG
```

6. Enable distributed segment processing:

On Data Domain:

```
sysadmin@dd_890_1# ddboost option show
Option                          Value
------------------------------  -------
distributed-segment-processing  enabled
------------------------------  -------
```

If disabled, enable as follows:

On Data Domain:

```
sysadmin@dd_890_1# ddboost option set distributed-segment-
processing enabled
DD Boost option "distributed-segment-processing" set to
enabled.
```

Note Ensure that the interface groups on Data Domain are disabled or at least not used for the DCA interfaces.

On Data Domain:

```
sysadmin@dd_890_1# ddboost ifgroup show config
The 'ifgroup' list is empty.
sysadmin@dd_890_1# ddboost ifgroup status
Status of ifgroup is "disabled".
```

7. Enable DD Boost.

   On Data Domain:

   ```
   sysadmin@dd_890_1# ddboost enable
   DD Boost enabled.
   ```

   Alternatively, as with all other DD Boost options, you can also enable DD Boost using the Enterprise Manager GUI with the Data Management › DD Boost view option.

   The Data Domain system is now ready to receive DD Boost backups run by the **gpcrondump** utility.

GPDB, DD Boost: Backing up to Data Domain from the DCA

To automate routine DD Boost backups, Greenplum provides the **gpcrondump** utility that can be called directly or from a crontab entry. The **gpcrondump** utility is a wrapper for the **gp_dump** command. **gpcrondump** also enables the backup of objects other than databases and data, such as database roles and server configuration files.

By default, **gpcrondump** creates data dump files in the Master and each Segment instance's data directory in **‹data_directory›/db_dumps/YYYYMMDD**. The Segment data dump files are compressed using **gzip**.

For DD Boost backups, **gpcrondump** with the **--ddboost** parameter is required. Without the **--ddboost** parameter, the backups are stored locally in the DCA array.

For the DD Boost solution, the deduplication is performed on the Segment Servers before sending the data to the storage unit on the Data Domain system. This reduces network traffic and speeds up the backup.

Note A storage unit called GPDB is automatically created on the Data Domain system during the initial DD Boost backup.

All the DD Boost backups were generated by running the following command as **gpadmin** user:

For **gpcrondump** uncompressed backups with DD Boost:

```
gpcrondump -x tpch --ddboost -z –a
```

Log files for the Greenplum Database management utilities are written to ~/gpAdminLogs by default. The naming convention for management log files is:

<script_name>_<date>.log

For example: *gpcrondump_<date>*.log

Table 14 lists the **gpcrondump** options used in DD Boost.

Table 14. DD Boost – gpcrondump options used

Parameter	Description
-x	Database name
--ddboost	Use Data Domain DD Boost for this backup
-z	Do not use compression
-a	Do not prompt

Note The **gpcrondump** compressed options --rsyncable are also available, but may not be suitable for environments with a high rate of random data changes. In order to gain the maximum benefit from EMC Data Domain deduplication technology, the **gpcrondump** uncompressed option **-z** is recommended.

It is also possible to automate parallel backups with **gpcrondump** by adding an entry to the crontab of the Master Server:

```
SHELL=/bin/bash
GPHOME=/usr/local/greenplum-db- 4.2.0.1
MASTER_DATA_DIRECTORY=/data/gpdb_master/Data-1
PATH=$PATH:$GPHOME/bin
01 0 * * * gpadmin gpcrondump -x tpch -c -g -G -a -q >>
gpcrondump_Backup.log
```

For more information, refer to the *Greenplum Database 4.2 Administrator Guide*.

GPDB, DD Boost: Restoring data on the DCA from Data Domain

For restoring a database using DD Boost, Greenplum provides the **gpdbrestore** utility, which is a wrapper for the **gp_restore** command.

The prerequisites for restoring data with DD Boost using **gpdbrestore** are:

- Backup files were created by a **gpcrondump –ddboost** operation.
- The Greenplum database system is up and running.
- The Greenplum database system has the exact same number of primary and segment instances as the system that was backed up.
- The database to which the data is being restored is already created in the system.

For example, to restore the **tpch** database backup generated by the **gpcrondump** on **14/12/2011**:

1. Ensure that the Data Domain system can be reached from the Master Server:

   On the DCA:

```
[gpadmin@mdw ~]$ ping dd_890_1_1
PING dd_890_1_1 (172.28.8.209) 56(84) bytes of data.
64 bytes from dd_890_1_1 (172.28.8.209): icmp_seq=1 ttl=64
time=0.105 ms
64 bytes from dd_890_1_1 (172.28.8.209): icmp_seq=2 ttl=64
time=0.103 ms
[gpadmin@mdw ~]$ ping dd_890_1_2
PING dd_890_1_2 (172.28.12.209) 56(84) bytes of data.
```

```
64 bytes from dd_890_1_2 (172.28.12.209): icmp_seq=1 ttl=64
time=0.755 ms
64 bytes from dd_890_1_2 (172.28.12.209): icmp_seq=2 ttl=64
time=0.109 ms
```

2. Make sure the database to which the data is being restored has already been created in the system:

   On the DCA:

   ```
   [gpadmin@mdw ~]$ createdb tpch
   ```

   Note This command fails if the database already exists.

3. Ensure that the database contains no objects as follows:

   On the DCA:

   ```
   [gpadmin@mdw ~]$ psql -d tpch
   psql (8.2.14)
   Type "help" for help.
   tpch=# \d
   No relations found.
   ```

4. Run the **gpdbrestore** utility:

   On the DCA:

   ```
   [gpadmin@mdw ~]$ gpdbrestore -t 20111214224212 –ddboost -a
   ```

   Log files for the Greenplum Database management utilities are written to ~/gpAdminLogs by default. The naming convention for management log files is:

   `<script_name>_<date>.log`

   For example: *gpdbrestore_<date>*.log

   Table 15 lists the **gpdbrestore** options used in DD Boost.

Table 15. DD Boost – gpdbrestore options used

Parameter	Description
-t	Timestamp key
--ddboost	Use Data Domain DD Boost for this restore
-a	Do not prompt

For more options, see the *Greenplum Database 4.2 Administrator Guide*.

Backup schedule

Data Domain's SISL scaling architecture ensures balanced backup and restore speeds.

The backup schedule for this solution uses full backups every day. The advantage of this schedule is that when a recovery is required, just a single restore is required, as incremental or differential restores are not needed. This greatly improves the RTO.

Data Domain systems support both full and incremental backups. However, the **gpcrondump** command can perform a full backup only. Therefore, EMC recommends that you perform a full backup to Data Domain since the recovery and restore process is simple and fast. This is because only a single restore is required, regardless of where in the schedule the data restore is required.

Note With Data Domain systems, it is also possible to do a hot backup while running query and ingest loads. For more information, refer to the *Greenplum Database 4.2 Administrator Guide*.

Greenplum HD, NFS: Configuring Data Domain and the DCA for backups

For backups of the DCA Greenplum HD modules, the Data Domain system must also be configured to enable the Greenplum HD servers to access the NFS directories that are used as a target for the **DistCp** utility to store the backups.

The following example shows how the system was configured using two 10 GbE interfaces with the IP addresses 172.28.8.209 and 172.28.12.209, respectively.

On Data Domain:

```
sysadmin@dd_890_1# net show settings
port     enabled   DHCP   IP address      netmask          type  additional setting
-----    -------   ----   -------------   -------------    ----  ------------------
eth0a    yes       no     172.28.4.209    255.255.248.0    n/a
eth0b    no        n/a    n/a             n/a              n/a
eth4a    yes       no     172.28.8.209    255.255.252.0    n/a
eth4b    yes       no     172.28.12.209   255.255.252.0    n/a
eth5a    no        n/a    n/a             n/a              n/a
eth5b    no        n/a    n/a             n/a              n/a
-----    -------   ----   -------------   -------------    ----   -----------------
* Value from DHCP
```

It is very important also to configure the host files on both the DCA Greenplum HD servers and the Data Domain system. The following example shows how the Data Domain host entries look on the Greenplum HD Servers.

On the DCA Greenplum HD modules:

```
[gpadmin@hdm1 ~]$ grep "Data Domain" /etc/hosts
### Data Domain - Backup
172.28.8.209    dd_889_1_1    # Data Domain interface eth4a, Network 172.28.8.0
172.28.12.209   dd_890_1_2    # Data Domain interface eth4b, Network 172.28.12.0
```

To add the hosts to the Data Domain system, use the following command:

```
net hosts add <ipaddr> <host-list>
```

The following example adds both 10 GbE interfaces for hosts hdm1 and hdw1.

On Data Domain:

```
sysadmin@dd_890_1# net hosts add 172.28.9.250 hdm1-1
Added "hdm1-1" -> "172.28.9.250" mapping to hosts list.
sysadmin@dd_890_1# net hosts add 172.28.13.250 hdm1-2
Added "hdm1-2" -> "172.28.13.250" mapping to hosts list
```

```
sysadmin@dd_890_1# net hosts add 172.28.10.1 hdw1-1
Added "hdw1-1" -> " 172.28.10.1" mapping to hosts list.
sysadmin@dd_890_1# net hosts add 172.28.14.1 hdw1-2
Added "hdw1-2" -> " 172.28.14.1" mapping to hosts list
```

Repeat with both 10 GbE interfaces for all Greenplum HD Servers.

Perform the following steps to configure NFS:

1. Add the NFS share **/backup** and set the access level to the DCA network.

   The networks used in this case are as follows:

   - For the Greenplum HD Master Servers, **172.28.9.0/22** and **172.28.13.0/22**
   - For the Greenplum HD Worker Servers, **172.28.10.0/22** and **172.28.14.0/22**

   On Data Domain:

   ```
   sysadmin@dd_890_1# nfs add /backup 172.28.9.0/22,172.28.13.0/22
   sysadmin@dd_890_1# nfs add /backup 172.28.10.0/22,172.28.14.0/22
   ```

   NFS export for **/backup** is added.

2. Ensure that the network was added correctly as shown in the following example:

   On Data Domain:

   ```
   sysadmin@dd_890_1# nfs show clients
   path             client           options
   --------------   --------------   ----------------------------------------
   /backup          172.28.10.0/22   (rw,no_root_squash,no_all_squash,secure)
   /backup          172.28.12.0/22  (rw,no_root_squash,no_all_squash,secure)
   /backup          172.28.13.0/22  (rw,no_root_squash,no_all_squash,secure)
   /backup          172.28.14.0/22   (rw,no_root_squash,no_all_squash,secure)
   /backup          172.28.8.0/22  (rw,no_root_squash,no_all_squash,secure)
   /backup          172.28.9.0/22  (rw,no_root_squash,no_all_squash,secure)
   ```

   Note **172.28.8.0/22** and **172.28.12.0/22** were added previously for the Greenplum Database and the Data Domain network.

3. Enable NFS on the Data Domain system as follows:

   On Data Domain:

   ```
   sysadmin@dd_890_1# nfs enable
   ```

   The NFS server is enabled.

4. On the DCA Master Server, mount the **/backup** NFS as shown in the following example:

   On the DCA:

   a. Create a temporary mount point for the **/backup** directory.

   ```
   [root@mdw ~]# mkdir /backup_tmp
   [root@mdw ~]# ls -la /backup_tmp
   total 12
   drwxr-xr-x  2 root root 4096 Aug 18 13:57 .
   drwxr-xr-x 30 root root 4096 Aug 18 13:57 ..
   ```

b. Mount the **/backup** directory on the Master Server (all one line):

```
[root@mdw ~]# mount -t nfs –o
hard,intr,nfsvers=3,tcp,rsize=32768,wsize=32768
dd_890_1_1:/backup /backup_tmp/
```

5. Create the DCA directory and all the server subdirectories.

   Note In this case, use the user **gpadmin** to avoid any issues with permissions while running the **DistCp** utility.

   In the following example, the directory DCA-01 and the subdirectory HD were created using **mkdir** as follows (all one line):

   On the DCA:

```
[gpadmin@mdw ~]$ mkdir /backup_tmp/DCA-01;mkdir
/backup_tmp/DCA-01/HD;
```

6. After all the directories are created, unmount the **/backup_tmp** directory as follows:

   On the DCA:

```
[root@mdw ~]# umount /backup_tmp
```

We created a single NFS share on Data Domain and the mount point created on each server was **/backup/DCA-01**. Table 16 outlines the mapping of the directories in this solution.

Table 16. NFS share and mount points

DCA server	Host name	NFS share on Data Domain	Mount point on DCA
Greenplum HD Master 01	hdm1	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Master 02	hdm2	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Master 03	hdm3	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Master 04	hdm4	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 01	hdw1	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 02	hdw2	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 03	hdw3	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 04	hdw4	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 05	hdw5	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 06	hdw6	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 07	hdw7	/backup/DCA-01/HD	/backup/DCA-01
Greenplum HD Worker 08	hdw8	/backup/DCA-01/HD	/backup/DCA-01

Note You can create the mount point on each DCA using the **gpssh** utility. The owner of the **/backup** directory must be **root**. The owner of the **/backup/DCA-01** directory must be **gpadmin**.

The following example describes one way of creating the mount points. For more options, refer to the *Data Domain Operating System (DD OS) Administration Guide*.

On the DCA Master Server (mdw):

```
[root@mdw ~]# cd ~gpadmin/gpconfigs

[root@mdw gpconfigs]# gpssh -f hostfile_hadoop
=> mkdir /backup
=> mkdir /backup/DCA-01
=> chown gpadmin:gpadmin /backup/DCA-01
=> exit
```

After the mount point is created, mount the Data Domain NFS share according to Table 16. The following command provides an example.

On the DCA:

```
[root@mdw gpconfigs]# gpssh -h hdm1 -v -e mount -t nfs -o
rw,hard,intr,tcp,rsize= 1048576,wsize= 1048576
dd_890_1_1:/backup/DCA-01/HD /backup/DCA-01
```

To get the best balance across both Data Domain 10 GbE interfaces (eth4a and eth4b), servers hdm1, hdm3, hdw1, hdw3, hdw5, and hdw7 should use host dd_890_1_1 (network 172.28.8.0, interface eth4a), while hdm2, hdm4, hdw2, hdw4, hdw6, and hdw8 should use host dd_890_1_2 (network 172.28.12.0, interface eth4b).

Check that the directories are mounted correctly.

On the DCA:

```
[root@mdw gpconfigs]# for NODE in `cat hostfile_hadoop`; do gpssh
-h $NODE -v -e df -h |grep DCA-01; done;
[hdm1] dd_890_1_1:/backup/DCA-01/HD
[hdm1]                          130T   63M  130T   1% /backup/DCA-01
[hdm2] dd_890_1_2:/backup/DCA-01/HD
[hdm2]                          130T   63M  130T   1% /backup/DCA-01
[hdm3] dd_890_1_1:/backup/DCA-01/HD
[hdm3]                          130T   63M  130T   1% /backup/DCA-01
[hdm4] dd_890_1_2:/backup/DCA-01/HD
[hdm4]                          130T   63M  130T   1% /backup/DCA-01
[hdw1] dd_890_1_1:/backup/DCA-01/HD
[hdw1]                          130T   63M  130T   1% /backup/DCA-01
[hdw2] dd_890_1_2:/backup/DCA-01/HD
[hdw2]                          130T   63M  130T   1% /backup/DCA-01
[hdw3] dd_890_1_1:/backup/DCA-01/HD
[hdw3]                          130T   63M  130T   1% /backup/DCA-01
[hdw4] dd_890_1_2:/backup/DCA-01/HD
[hdw4]                          130T   63M  130T   1% /backup/DCA-01
[hdw5] dd_890_1_1:/backup/DCA-01/HD
[hdw5]                          130T   63M  130T   1% /backup/DCA-01
[hdw6] dd_890_1_2:/backup/DCA-01/HD
[hdw6]                          130T   63M  130T   1% /backup/DCA-01
```

```
[hdw7] dd_890_1_1:/backup/DCA-01/HD
[hdw7]                        130T   63M  130T   1% /backup/DCA-01
[hdw8] dd_890_1_2:/backup/DCA-01/HD
[hdw8]                        130T   63M  130T   1% /backup/DCA-01
```

If all the NFS shares are mounted successfully, the Data Domain system is ready to receive any backups run by the **DistCp** utility.

Greenplum HD, NFS: Backing up to Data Domain from the DCA

DistCp, which was designed for Hadoop inter- and intra-cluster copying, can be used for Greenplum HD backups via NFS. The **DistCp** utility can be called directly or from a **crontab** entry.

DistCp uses a distributed copy from the source Greenplum HD cluster to the destination. For the Greenplum HD backup solution, the intention was to send the backups to Data Domain so the directory we used was the Data Domain NFS share created on **/backup/DCA-01/HD**.

We generated all the Greenplum HD NFS backups by running the following command as user **gpadmin** on the Greenplum HD Master server (hdm1):

```
[gpadmin@hdm1 ~]$ hadoop distcp  hdfs://hdm1:8020/user/gpadmin
file:///backup/DCA-01/<YYYYMMDDhhmm>
```

To take advantage of Data Domain's deduplication technology, we used a script to create a new target directory ‹YYYYMMDDhhmm› on the Data Domain NFS share for each individual backup. This way, Data Domain takes care of any duplication in subsequent backups.

Note Attempting to send subsequent backups to the same target directory may result in **DistCp** skipping previously copied files even if they have changed since the previous backup.

Table 17 lists some of the **DistCp** options.

Table 17. Some distcp options

Parameter	Description
-p	Preserve status
-i	Ignore failures
-log ‹logdir›	Write logs to ‹logdir›
-m ‹num_maps›	Maximum number of simultaneous copies
-overwrite	Overwrite destination
-update	Overwrite if the source size is different from the destination size
-skipcrccheck	Do not use CRC check to determine if source is different from destination. Relevant only if **–update** is specified

We observed task failing errors during the **DistCp** jobs as in the following example:

```
INFO mapred.JobClient: Task Id :
attempt_201206060707_0009_m_000007_0, Status : FAILED
```

```
Task attempt_201206060707_0009_m_000007_0 failed to report status
for 601 seconds. Killing!
```

This only becomes a problem if the same **distcopy** task fails more times than the maximum attempts allowed. Should this occur, the remaining map tasks are cancelled resulting in an incomplete backup. The following options can be considered for the initial backup to avoid failing tasks.

The **-m** parameter can be used especially for the initial backup to Data Domain. The **-m** parameter limits the maximum number of simultaneous copies. This is useful to ensure the backup target is not overwhelmed as all data needs to be written to Data Domain during the initial backup. If **-m** is not specified, **DistCp** will attempt to schedule work, the minimum being (total_bytes / bytes.per.map OR 20 * num_task_trackers), where bytes.per.map defaults to 256 MB.

For example:

```
[gpadmin@hdm1 ~]$ hadoop distcp –m 40
hdfs://hdm1:8020/user/gpadmin file:///backup/DCA-01/<YYYYMMDDhhmm>
```

The Greenplum HD configuration parameter **mapred.task.timeout** can also be changed for the initial backup. The **mapred.task.timeout** determines the number of milliseconds before a task is terminated if it does not read an input, write an output, or update its status string.

Note The **mapred.task.timeout** is a global parameter. Before making this change, you need to understand the potential impact on other tasks. Set **mapred.task.timeout** back to the default value as soon as possible.

The **mapred.task.timeout** can be changed as follows:

1. As user **gpadmin**, connect to the DCA master server:

   ```
   [gpadmin@mdw ~]$ scp hdm1:/opt/dca/etc/hadoop_conf/mapred-
   site.xml .
   ```

2. Change these values from:

   ```
   <property>
   <name>mapred.task.timeout</name>
   <value>600000</value>
   </property>
   ```

   To the following:

   ```
   <property>
   <name>mapred.task.timeout</name>
   <value>1800000</value>
   </property>
   ```

3. Update mapred-site.xml on all nodes in the Greenplum HD cluster:

   ```
   [gpadmin@mdw ~]$ cd gpconfigs
   [gpadmin@mdw gpconfigs]$ gpscp -f hostfile_hadoop
   /home/gpadmin/mapred-site.xml
   =:/opt/dca/etc/hadoop_conf/mapred-site.xml
   ```

4. Stop and restart the Greenplum HD cluster.

As user **gpadmin**, connect to the DCA Greenplum HD namenode server and stop and restart Greenplum HD as follows:

```
[gpadmin@hdm1 ~]$ dca_hadoop –stop
[gpadmin@hdm1 ~]$ dca_hadoop –start
```

The new **mapred.task.timeout** setting is now active.

Log files are written to /opt/dca/libexec/hadoop-1.0.0-gphd-1.1.0.0/logs by default. The naming convention for management log files is:

```
<script_name>_<date>.log
hadoop-gpadmin-<service_name>-<server_name>.log
```

For example: *hadoop-gpadmin-datanode-hdw1.log*

For more information, refer to http://hadoop.apache.org/.

Greenplum HD, NFS: Restoring data to the DCA using Data Domain

This solution also uses **DistCp** for the restore. We reversed the source and destination fields so the source became the NFS share provided by the Data Domain system and the Greenplum HD cluster became the target.

We generated all the Greenplum HD NFS restores by running the following command as user **gpadmin**:

```
hadoop distcp file:///backup/DCA-01/<source-folder>
hdfs://hdm1:8020/<destination-folder>
```

Backup of Greenplum Chorus using Data Domain

Perform the following steps to back up Greenplum Chorus to Data Domain:

1. Create a backup target for Chorus on the Data Domain system:

   a. On the DCA Standby Master Server (smdw), mount the Data Domain **/backup** NFS share as shown in the following example:

   On the DCA:

   Create a temporary mount point for **/backup**:

   ```
   [root@smdw ~]# mkdir /backup_tmp
   ```

   Mount the Data Domain **/backup** directory on the Standby Master Server (all one line):

   On the DCA:

   ```
   [root@smdw ~]# mount -t nfs –o
   hard,intr,nfsvers=3,tcp,rsize=32768,wsize=32768
   dd_890_1_1:/backup /backup_tmp/
   ```

   b. Create a backup directory on Data Domain for Chorus.

   In the following example, the directory DCA01-Chorus was created by user chorus using **mkdir** as follows:

   On the DCA:

   ```
   [chorus@smdw ~]$ mkdir /backup_tmp/DCA01-Chorus
   ```

After the Chorus backup directory is created, unmount the **/backup_tmp** directory as follows:

On the DCA:

```
[root@smdw ~]# umount /backup_tmp
```

c. Create a mount point to mount the Chorus backup share on the Standby Master Server.

On the DCA:

```
[root@smdw ~]# mkdir /backup
[root@smdw ~]# mkdir /backup/DCA01-Chorus
[root@smdw ~]# chown chorus:chorus /backup/DCA01-Chorus
```

d. Mount the **/backup/DCA01-Chorus** share on the Standby Master Server (all one line):

```
[root@smdw ~]# mount -t nfs –o
hard,intr,nfsvers=3,tcp,rsize=32768,wsize=32768
dd_890_1_1:/backup/DCA01-Chorus /backup/DCA01-Chorus
```

2. On the DCA Standby Master Server, change the user to **chorus**:

```
[root@smdw ~]# su - chorus
```

3. Change the directory to **<chorus install path>**:

```
[chorus@smdw ~]$ cd  /usr/local/greenplum-chorus
```

4. Source the path **chorus_path.sh**:

```
[chorus@smdw chorus]$ source chorus_path.sh
```

5. Back up Greenplum Chorus to Data Domain using the following syntax:

```
chorus_control.sh backup [-d backup_dir] [-r rolling_days]
```

Notes

- Chorus should be running when you run the backup.
- Chorus binary backup file greenplum_chorus_backup_YYYYMMDD_HHMMSS.tar is dumped to the specified backup directory.
- You can control the expiry time by setting **rolling_days**.
- If no backup directory is specified, a default backup directory **/data/greenplum-chorus/bak** is created and used.

The following example backs up Chorus to **/backup/DCA01-Chorus** (the Data Domain share) and automatically removes all Chorus backup files more than 30 days old:

```
[chorus@smdw greenplum-chorus]$ chorus_control.sh backup -d
/backup/DCA01-Chorus -r 30
Backing up chorus data...
```

```
/usr/local/greenplum-chorus/releases/2.2.0.0.1379-
5e2c3529e/vendor/bundle/jruby/1.9/gems/jdbc-postgres-
9.0.801/lib/jdbc/postgres.rb:4 warning: already initialized constant
VERSION
Dumping database contents...
Compressing assets...
Created backup archive file: /backup/DCA01-
Chorus/greenplum_chorus_backup_20121122_150141.tar
Removing backups more than 30 days old...
```

Restore of Greenplum Chorus using Data Domain

To restore Greenplum Chorus, complete the following steps.

Note The restore process has three main steps.

1. Stop Chorus:

   a. If Chorus is running, stop Chorus as follows:

   ```
   [root@smdw ~]# su – chorus
   ```

   b. Change directory to the Chorus installation directory:

   ```
   [chorus@smdw ~]$ cd /usr/local/greenplum-chorus
   [chorus@smdw chorus]$ source chorus_path.sh
   [chorus@smdw chorus]$ chorus_control.sh stop
   ```

2. Restore Greenplum Chorus to Data Domain using the following syntax:

   ```
   chorus_control.sh restore backup_filename
   ```

   The following example restores Chorus from **/backup/DCA01-Chorus** (the Data Domain share) using a Chorus backup from date `20121122` and time `_150141`:

```
[chorus@smdw greenplum-chorus]$ chorus_control.sh restore /backup/DCA01-
     Chorus/greenplum_chorus_backup_20121122_150141.tar
Restoring chorus data...
Setting maximum database connections to 75
[production] postgres started as pid 26839
/usr/local/greenplum-chorus/releases/2.2.0.0.1379-
     5e2c3529e/vendor/bundle/jruby/1.9/gems/jdbc-postgres-
     9.0.801/lib/jdbc/postgres.rb:4 warning: already initialized
     constant VERSION
** Invoke backup:restore (first_time)
** Execute backup:restore
Continuing will overwrite existing assets and data. It is strongly
     advised that
you have a recent backup available before performing a restore.

Are you sure you want to continue? (Y/N):
y
Deleting existing assets...
Restoring backed up assets...
Restoring database...

[production] stopping postgres  ( Stopped )

Restore of /backup/DCA01-
     Chorus/greenplum_chorus_backup_20121122_150141.tar completed.
To start Chorus, run the following commands:

    source /usr/local/greenplum-chorus/chorus_path.sh
```

```
    chorus_control.sh start
[chorus@smdw greenplum-chorus]$ chorus_control.sh start
Setting maximum database connections to 75
[production] postgres started as pid 27284
[production] Worker started as pid 27407
[production] Scheduler started as pid 27594
[production] Solr started as pid 27743 on port 8983
[production] Writing nginx config...
[production] nginx started as pid 27958
[production] updating jetty config...
[production] starting jetty...
```

3. Start Chorus with the recovered instance:

```
[root@smdw ~]# su – chorus
[chorus@smdw ~]$ cd /usr/local/greenplum-chorus
[chorus@smdw chorus]$ source chorus_path.sh
[chorus@smdw chorus]$ chorus_control.sh start
```

Greenplum Chorus is now restored and you can log in to Chorus as usual. For more information, refer to the *Greenplum Chorus 2.2 Installation Guide*.

GPDB, NFS test results

Test objectives

The test objectives were to validate the success of data backup and restore over NFS by:

- Comparing Data Domain compression rates
- Quantifying the Data Domain deduplication ratio using increased data loads
- Quantifying the impact of running read queries and write queries on the database while running backups
- Quantifying the successful restoration of data with the time to complete the restores

Test scenarios

Table 18 outlines the data backup and restore test scenarios in this solution.

Table 18. GPDB, NFS backup and restore test scenarios

Test	Description
1	Perform full backup to Data Domain via NFS with Greenplum backup compression off, and with Data Domain using **lz**, **gzfast**, and **gz** local compression options, to determine the best compression option to use.
2	Perform full backup to Data Domain via NFS with Greenplum backup compression off and Data Domain **lz** compression over a simulated week of full backups.
3	Perform full backup to Data Domain via NFS with Greenplum backup compression off and Data Domain **lz** compression with: • Query load running • Ingest load running
4	Perform a restore of the Greenplum database via NFS from a Greenplum uncompressed backup.

Test 1 results

In Test 1, we performed a full backup to Data Domain via NFS with Greenplum compression off and with Data Domain using **lz**, **gzfast**, and **gz**.

Figure 3 shows the comparison of **lz**, **gzfast**, and **gz** types of NFS test results.

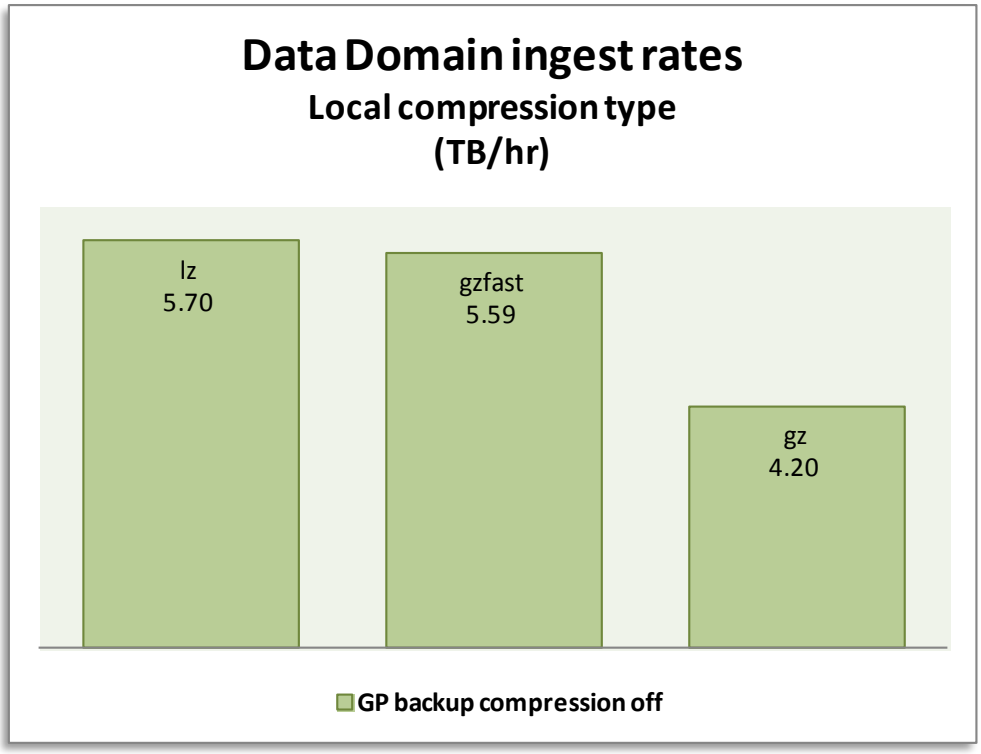

Figure 3. GPDB, NFS - Comparison of lz, gzfast, and gz

Note that the Data Domain ingest rate refers to the speed at which the DD890 was able to accept incoming backup data.

For the NFS solution, the choice was to use a lighter compression type. **lz** was selected as it gives the best performance. We performed the test on a 2.59 TB database, where an incremental data load of 5 percent was applied since the previous backup.

Test 2 results

In Test 2, we performed a full backup to Data Domain with Greenplum backup compression off and Data Domain **lz** compression.

Figure 4 illustrates the 5 percent daily increase in the source database size over seven days on the DCA with an uncompressed backup of an uncompressed database. To simulate the real-life expectancies and data growth of a data warehouse, an incremental data load of 5 percent was applied between backups.

Note The Direct I/O feature was enabled for this test and the chunk size was set to 20. For more information, see section GPDB, NFS: Backups with Direct I/O.

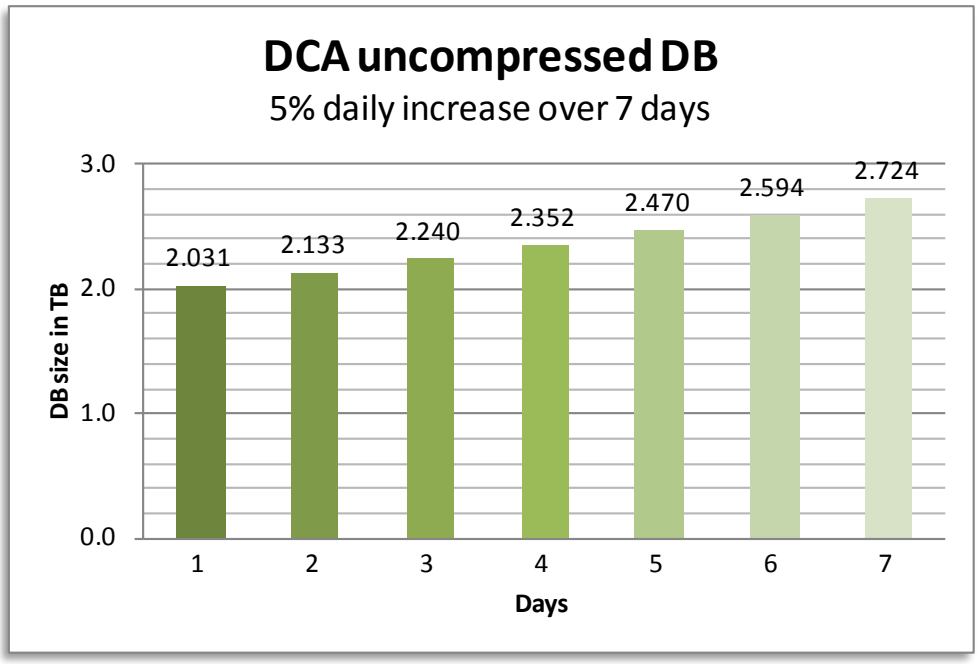

Figure 4. GPDB, NFS - Five percent daily increase over a simulated seven days with an uncompressed database

Figure 5 illustrates the backup duration in minutes on the DCA with an uncompressed backup of an uncompressed database, based on the 5 percent daily increase. The first backup takes more time to complete because it is the first time the data is being written to the Data Domain system.

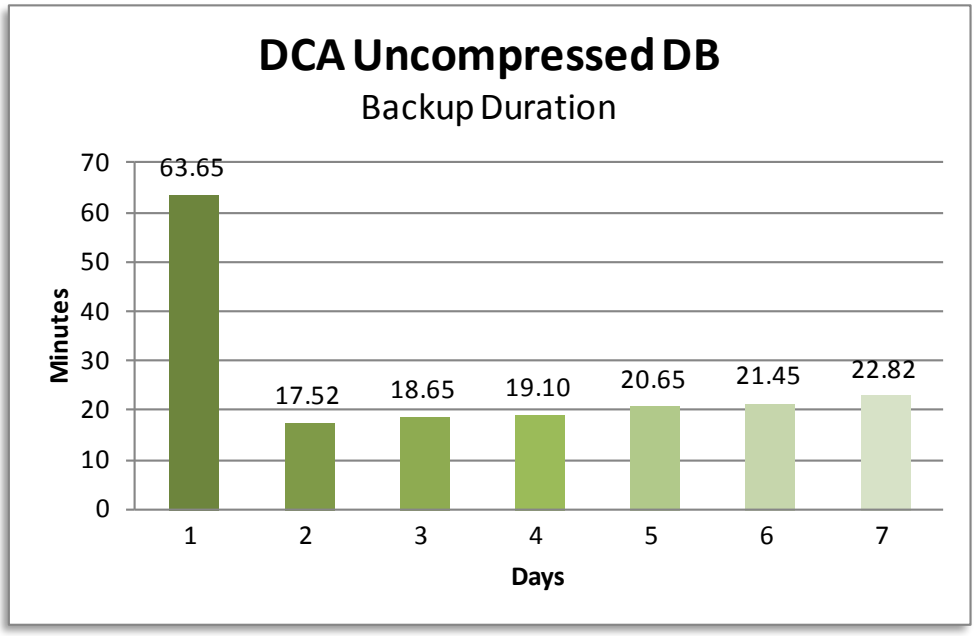

Figure 5. GPDB, NFS - Backup duration with gpcrondump uncompressed backup

Figure 6 illustrates the backup rate in TB/hour with an uncompressed backup of an uncompressed database. As seen previously in the backup duration test, the throughput is slower on the first backup. Using **gpcrondump** uncompressed backups with Data Domain's deduplication technology results in an average backup speed of 6.91 TB/hour for backups 2 to 7.

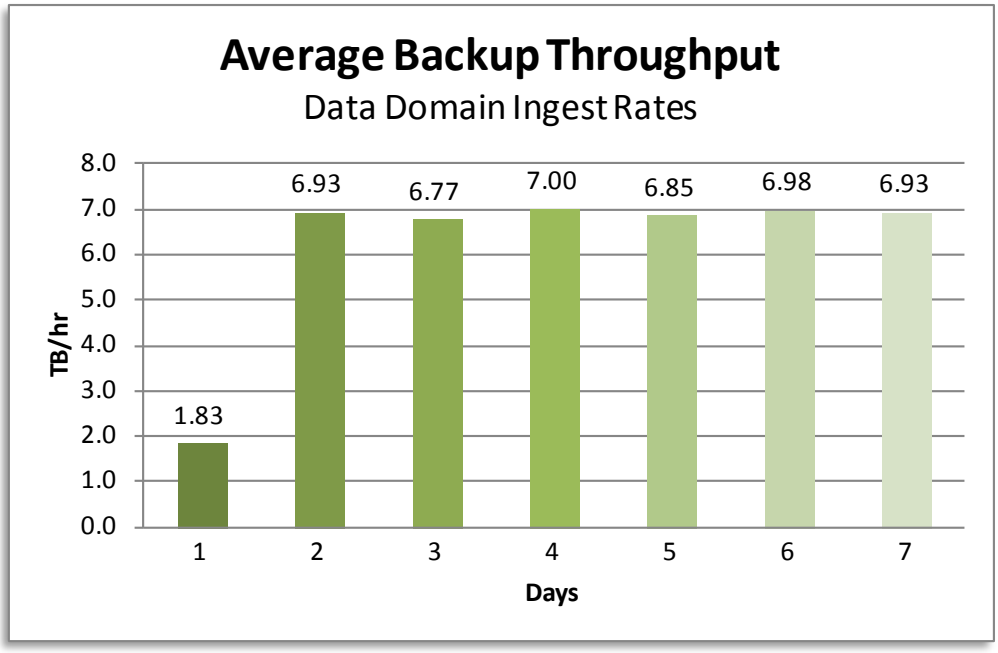

Figure 6. GPDB, NFS -Data Domain ingest rate with gpcrondump uncompressed backup

Due to the strength of Data Domain deduplication technology, there was an average 34.2x storage saving for each nightly backup. Shortly after the nightly database backups began, the savings were significant. For example, on the second day, a full backup of 1,971 GiBibytes (GiB) used 58.6 GiB, resulting in a 33.6x storage reduction.

Figure 7 illustrates the tremendous incremental savings that can be achieved on a daily basis. Over time, the savings are even greater.

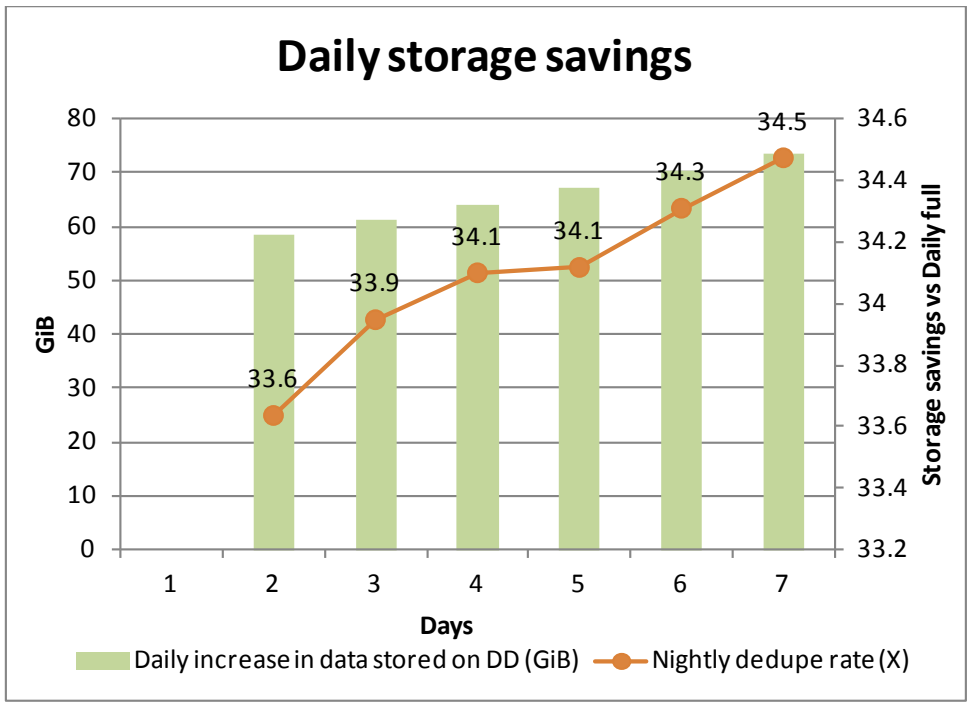

Figure 7. GPDB, NFS - Storage savings after seven days

Note During all the backup tests, the DCA was idle.

Figure 8 illustrates the cumulative effect of this storage savings over a seven-day backup cycle. After seven days of running the backup, 15,362 GiB of data was backed up; however, only 1,361.6 GiB of storage was needed on the Data Domain deduplication storage system, resulting in a 11.3x storage saving. This 91.1 percent storage saving represents a significant saving in backup infrastructure and facility costs. Regular backups of large data warehouses are far more sustainable and much less costly than before.

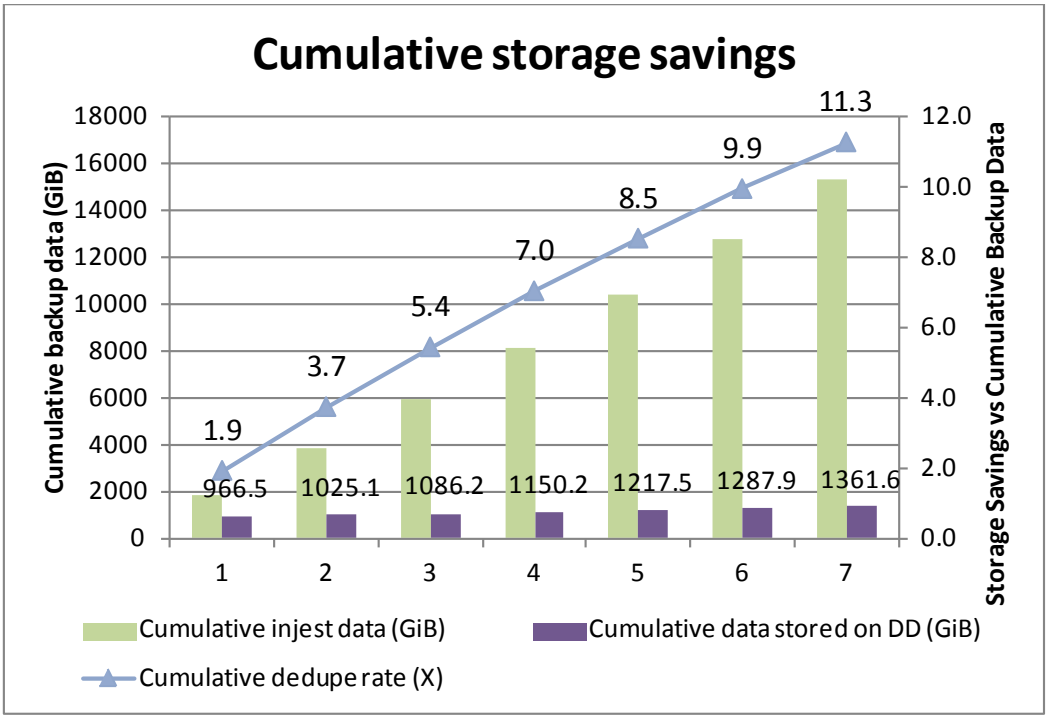

Figure 8. GPDB, NFS - Cumulative storage savings over seven days

Note There is roughly a 5.5 percent difference between the daily database sizes reported using Greenplum utilities in Figure 4 versus the actual data saved to the Data Domain system, as noted in Figure 8. The reason for this difference is that **gpcrondump** extracts the actual data from the Greenplum database for backing up to the Data Domain system. The 5.5 percent is additional Greenplum overhead that is not required for a successful backup or restore.

Test 3 results

In Test 3, we performed a full backup to the Data Domain system with a query load and an ingest load running against the DCA. No incremental data was applied since the previous backup. The query and ingest loads were not performed at the same time. The intention was to demonstrate the impact while a backup job was running.

Figure 9 illustrates the DCA database size before and after the ingest load. The amount of data backed up was 2.03 TB and the database went from 2.03 TB to 2.29 TB while the backup was performed.

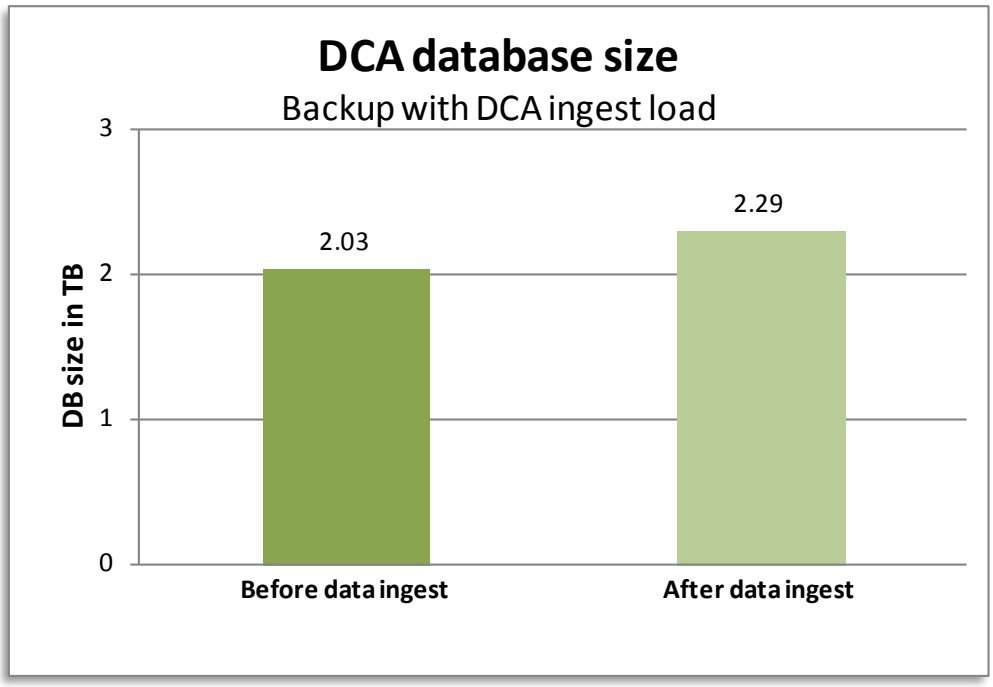

Figure 9. GPDB, NFS - Database size before and after the ingest load

Figure 10 compares the DCA backup duration in minutes with no load, query load, and ingest load running. As illustrated, under a concurrent query load, the NFS backup performance is somewhat degraded. NFS backup performance under a full ingest load, which is a write-intensive process, is less affected, with no operational issues.

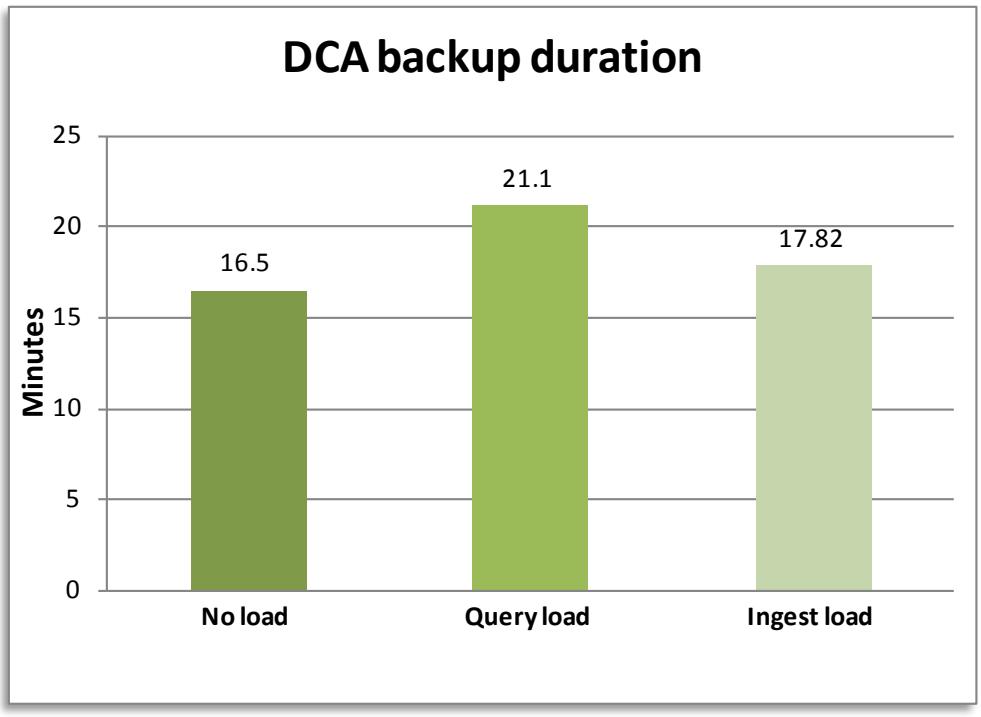

Figure 10. GPDB, NFS - Backup duration under varying conditions

Figure 11 compares the Data Domain average ingest rates with no load, query load, and ingest load running.

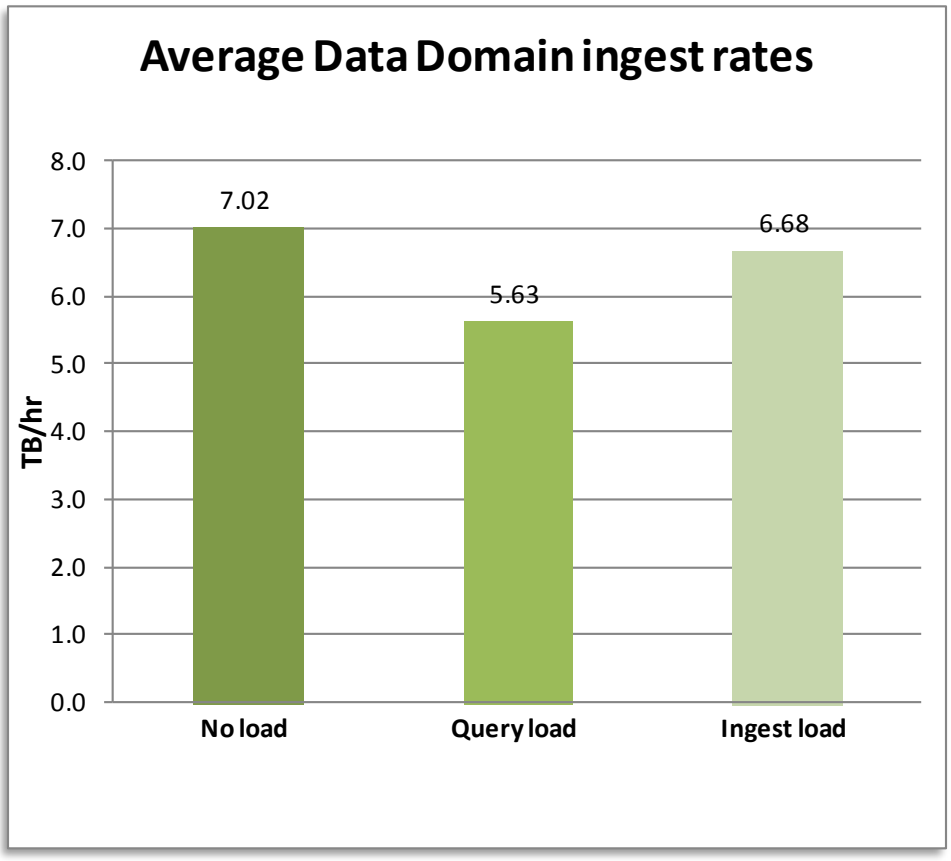

Figure 11. GPDB, NFS – Average Data Domain ingest rate under varying conditions

Test 4 results

In Test 4, we performed a restore of the Greenplum database using the Greenplum uncompressed backup set from previous tests.

The Test 4 DCA restore with uncompressed data produced the following results:

- The restore of a 2.72 TB database from an uncompressed backup was achieved in 30.12 minutes.
- The average restore throughput of a Greenplum uncompressed backup was 5.43 TB/hour.

GPDB, DD Boost test results

Test objectives

The test objectives were to validate the success of a DD Boost data backup and restore by:

- Comparing Data Domain compression rates
- Quantifying the Data Domain deduplication ratio using increased data loads
- Quantifying the impact of running read queries and write queries on the database while running backups
- Quantifying the successful restoration of data with the time to complete the restores

Test scenarios

Table 19 outlines the data backup and restore test scenarios in this solution.

Table 19. GPDB, DD Boost backup and restore test scenarios

Test	Description
1	Perform a full backup to Data Domain using DD Boost with Greenplum backup compression off and with Data Domain using **lz**, **gzfast**, and **gz** local compression options to determine the best compression option.
2	Perform a full backup to Data Domain using DD Boost with Greenplum backup compression off and Data Domain **lz** compression over a simulated week of full backups.
3	Perform a full backup to Data Domain using DD Boost with Greenplum backup compression off and Data Domain **lz** compression with: • Query load running • Ingest load running
4	Restore the Greenplum database using DD Boost from a Greenplum uncompressed backup.

Test 1 results

Test 1 comprised a full backup to Data Domain using DD Boost, with Greenplum compression off and with Data Domain using **lz**, **gzfast**, and **gz.**

Figure 12 shows the **lz**, **gzfast**, and **gz** DD Boost test results.

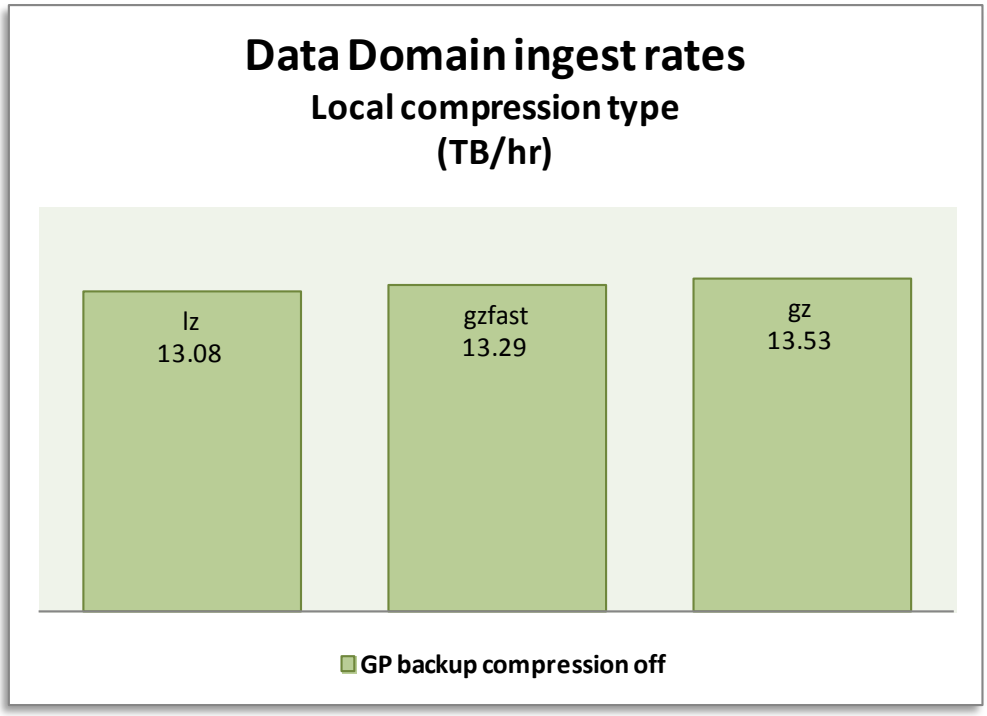

Figure 12. GPDB, DD Boost: Comparison of lz, gzfast, and gz

The Data Domain ingest rate refers to the speed at which the DD890 was able to accept incoming backup data.

For the DD Boost solution, **gz** provides the best backup and restore performance as well as increased storage savings. Because this is a global setting, the choice was to use the default and lighter compression option **lz**. In most cases, the Data Domain system is also used for other backup jobs and using **gz** can have a significant effect on their performance. We performed the test on a 2.59 TB database, where an incremental data load of 5 percent was applied since the previous backup.

Test 2 results

Test 2 comprised a full backup to Data Domain, with Greenplum compression off, using Data Domain **lz** compression.

Figure 13 illustrates the 5 percent daily increase in the source database size over seven days on the DCA with an uncompressed backup of an uncompressed database. To simulate the real-life expectancies and data growth of a data warehouse, we applied an incremental data load of 5 percent between backups.

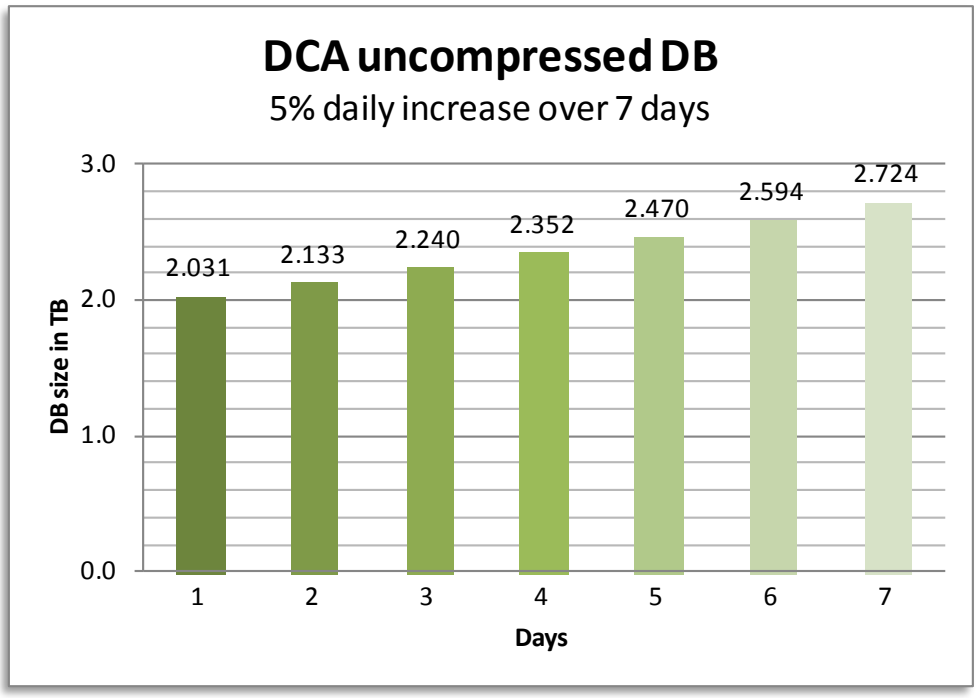

Figure 13. GPDB, DD Boost: Five percent daily increase over a simulated seven days with an uncompressed database

Figure 14 illustrates the backup duration in minutes on the DCA with an uncompressed backup of an uncompressed database, based on the 5 percent daily increase. The first backup takes more time to complete because it is the first time the data is being written to the Data Domain system.

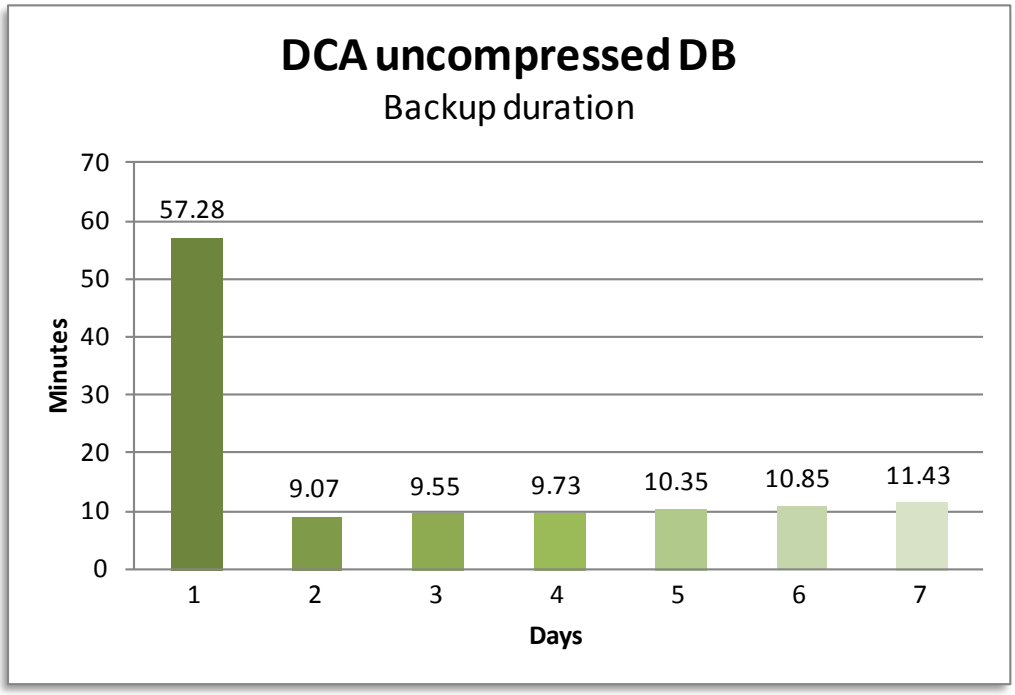

Figure 14. GPDB, DD Boost: Backup duration with gpcrondump uncompressed backup

Figure 15 illustrates the backup rate in TB/hour with an uncompressed backup of an uncompressed database. As seen previously in the backup duration test, the throughput is slower on the first backup. The combination of **gpcrondump** uncompressed backups and Data Domain's deduplication technology results in an average backup speed of 13.08 TB/hour for backups 2 to 7.

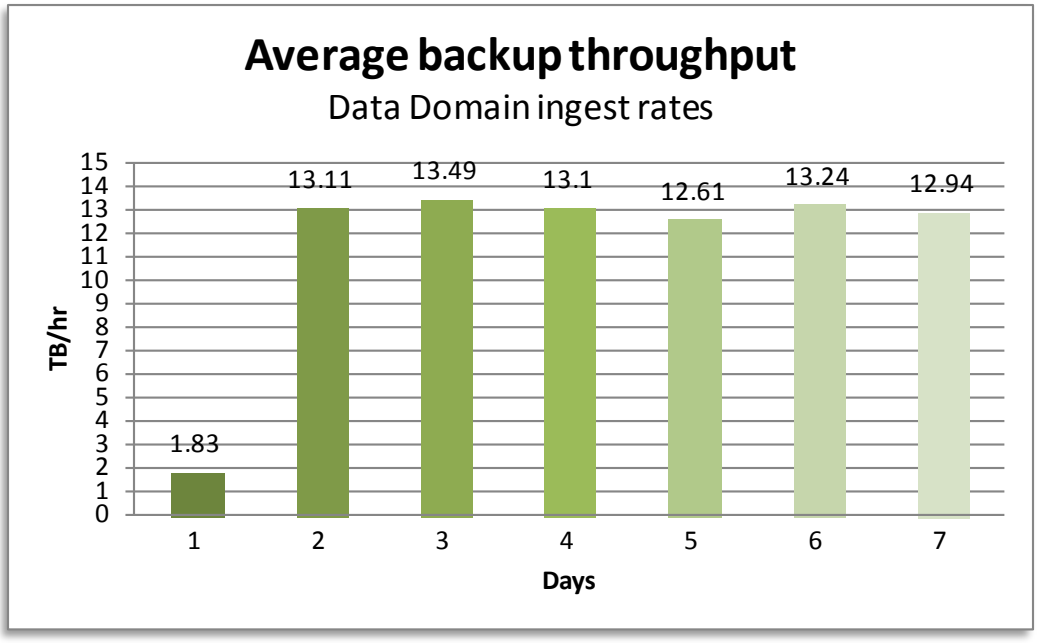

Figure 15. GPDB, DD Boost: Data Domain ingest rate with gpcrondump uncompressed backup

Due to the strength of Data Domain's deduplication technology, there was an average 34.5x storage saving for each nightly backup. Shortly after the nightly database backups began, the savings were significant. For example, on the second day, a full backup of the 1,971.5 GiB used 59 GiB, resulting in a 33.4x storage reduction.

Figure 16 illustrates the tremendous incremental savings that can be achieved on a daily basis. Over time, the savings are even greater.

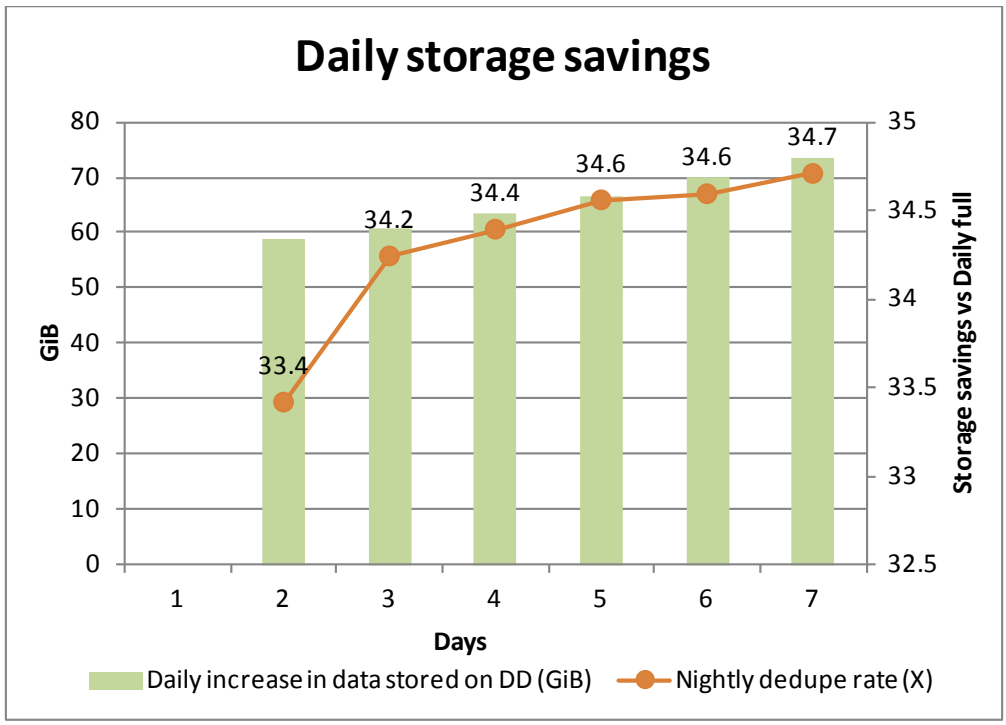

Figure 16. GPDB, DD Boost: Storage saving after seven days

Note During all the backup tests, the DCA was idle.

Figure 17 illustrates the cumulative effect of this storage saving over a seven-day backup cycle. After seven days of running the backup, 15,363.2 GiB of data was backed up; however, only 1,354.9 GiB of storage was needed on the Data Domain deduplication storage system, resulting in a 11.3x storage saving. This 91.2 percent storage saving represents a significant saving in backup infrastructure and facility costs. Regular backups of large data warehouses are far more sustainable and much less costly than before.

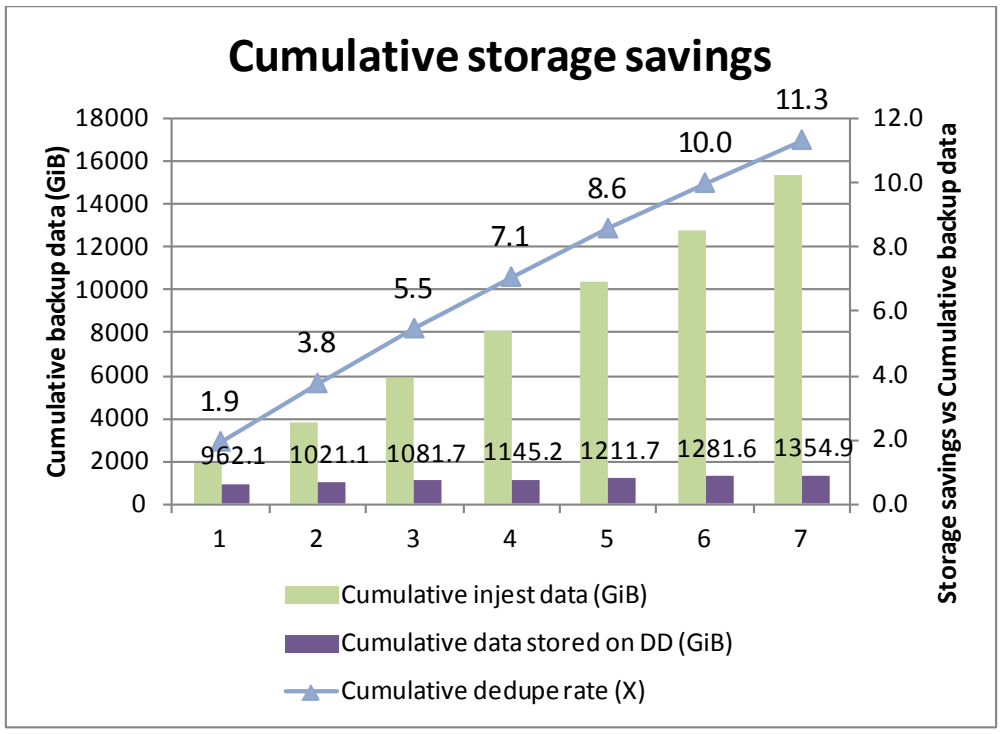

Figure 17. GPDB, DD Boost: Cumulative storage savings over seven days

There is roughly a 5.5 percent difference between the daily database sizes reported using Greenplum utilities in Figure 13 versus the actual data saved to the Data Domain system as noted in Figure 17. This is because **gpcrondump** extracts the actual data from the Greenplum database for backing up data to the Data Domain system. The 5.5 percent is additional Greenplum overhead that is not required for a successful backup or restore.

Test 3 results

Test 3 comprised a full backup to the Data Domain system with a query load and an ingest load running against the DCA. No incremental data was applied since the previous backup. The query and ingest loads were not performed at the same time. The intention was to demonstrate the impact while a backup job was running.

Figure 18 illustrates the DCA database size before and after the ingest load. The amount of data backed up was 2.03 TB and the database went from 2.03 TB to 2.42 TB while we performed the backup.

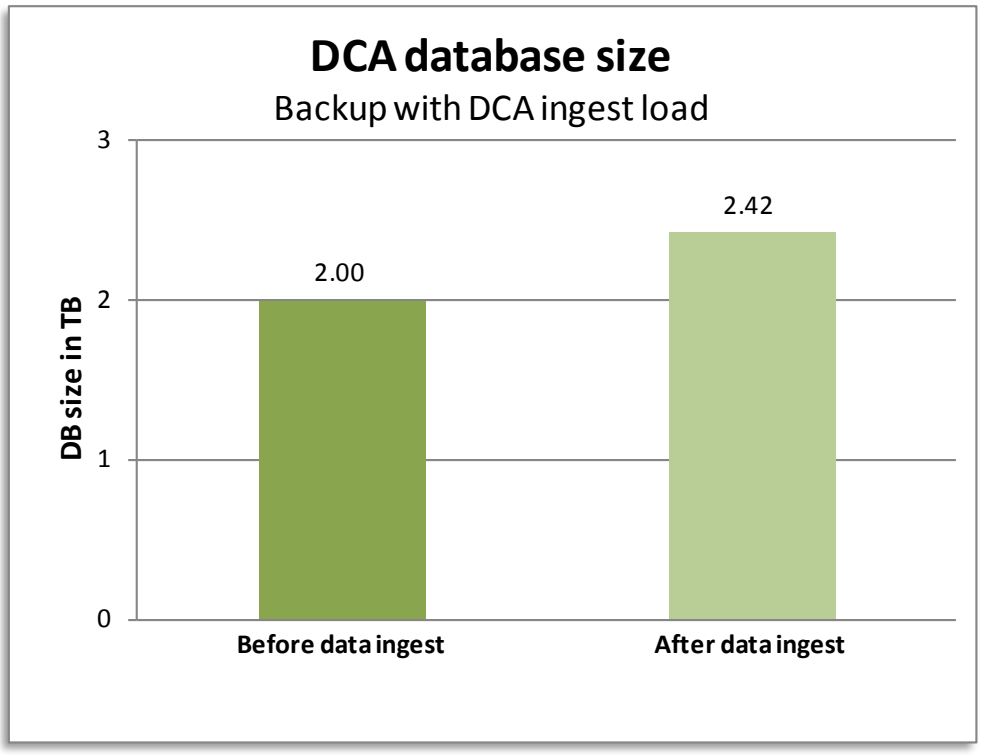

Figure 18. GPDB, DD Boost: Database size before and after the ingest load

Figure 19 compares the DCA backup duration in minutes with no load, query load, and ingest load running. As illustrated, under a concurrent query load, DD Boost backup performance is degraded. DD Boost backup performance under a full ingest load, which is a write-intensive process, is less affected, with no operational issues.

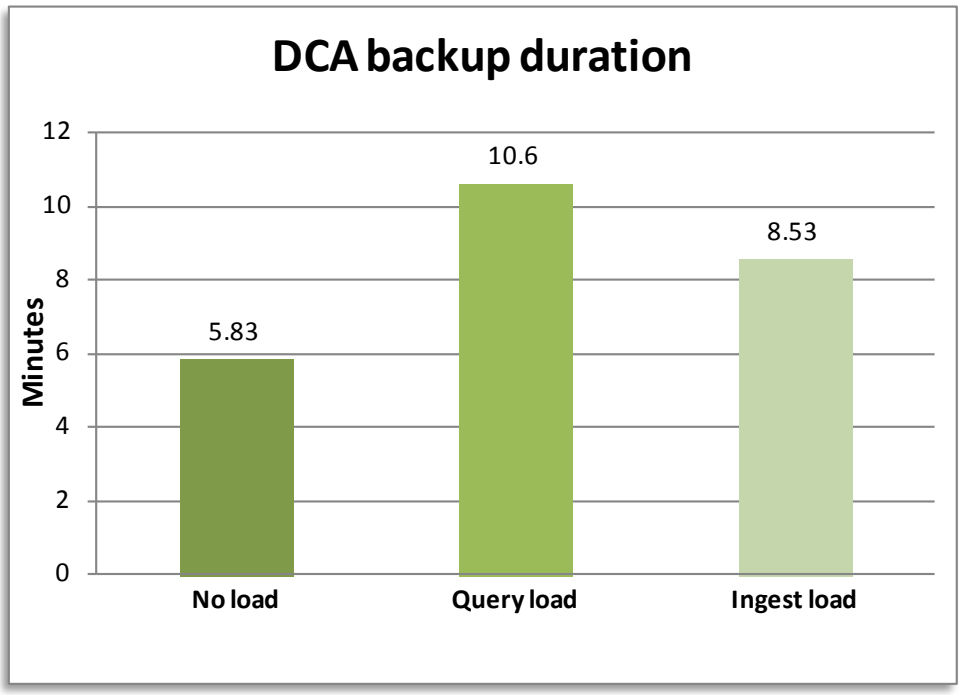

Figure 19. GPDB, DD Boost: Backup duration under varying conditions

Figure 20 compares the Data Domain average ingest rates with no load, query load, and ingest load running.

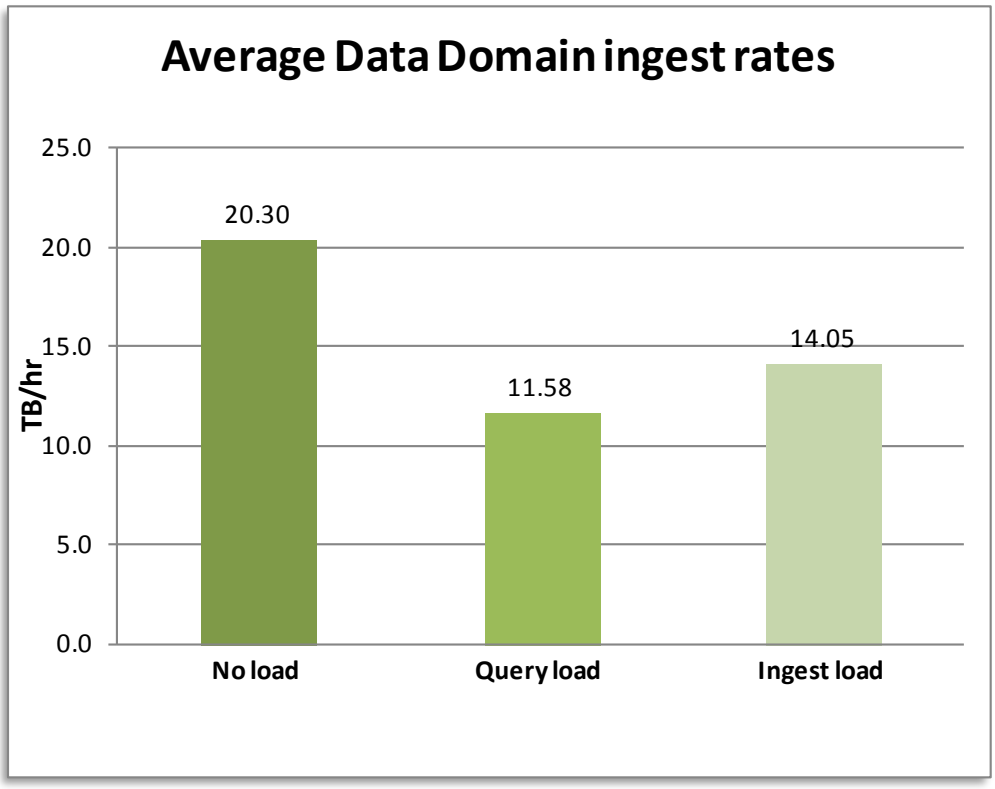

Figure 20. GPDB, DD Boost: Average Data Domain ingest rate under varying conditions

Test 4 results

In Test 4, we performed a restore of the Greenplum database using the Greenplum uncompressed backup set from previous tests.

The Test 4 DCA restore with uncompressed data produced the following results:

- The restore of a 2.72 TB database from an uncompressed backup was achieved in 27.72 minutes.
- The average restore throughput of a Greenplum uncompressed backup was 5.9 TB/hour.

Greenplum HD, NFS test results

Test objectives

The test objectives were to validate the success of data backup and restore over NFS by quantifying:

- The Data Domain deduplication ratio using increased data loads
- The successful restoration of data with the time to complete the restores

Test scenarios

Table 20 outlines the data backup and restore test scenarios in this solution.

Table 20. NFS – Greenplum HD backup and restore test scenarios

Test	Description
1	Perform a full backup to Data Domain via NFS with Hadoop file system (HDFS) uncompressed data and Data Domain **lz** compression over a simulated week of full backups.
2	Perform a restore of HDFS uncompressed data via NFS.

Test 1 results

In Test 1, we performed a full backup to Data Domain using **DistCp** with the default number of simultaneous copies and Data Domain **lz** compression. We also set the Greenplum HD configuration parameter **mapred.task.timeout** to the default value.

Figure 21 illustrates a 5 percent daily increase in the source data over seven days on the DCA HDFS for backups of uncompressed data. To simulate the real-life expectancies and data growth of a data warehouse, we applied an incremental data load of 5 percent between backups.

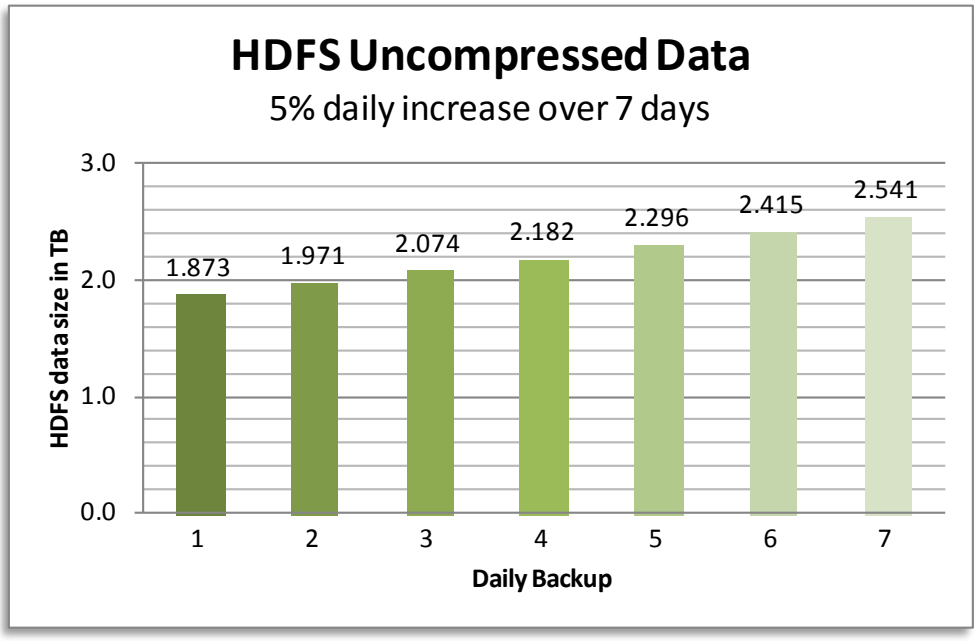

Figure 21. Greenplum HD, NFS: Five percent daily increase over a simulated seven days with uncompressed data

Figure 22 illustrates the backup duration in minutes on the DCA with a backup of uncompressed data, based on the 5 percent daily increase. The first backup takes more time to complete because it is the first time the data is being written to the Data Domain system.

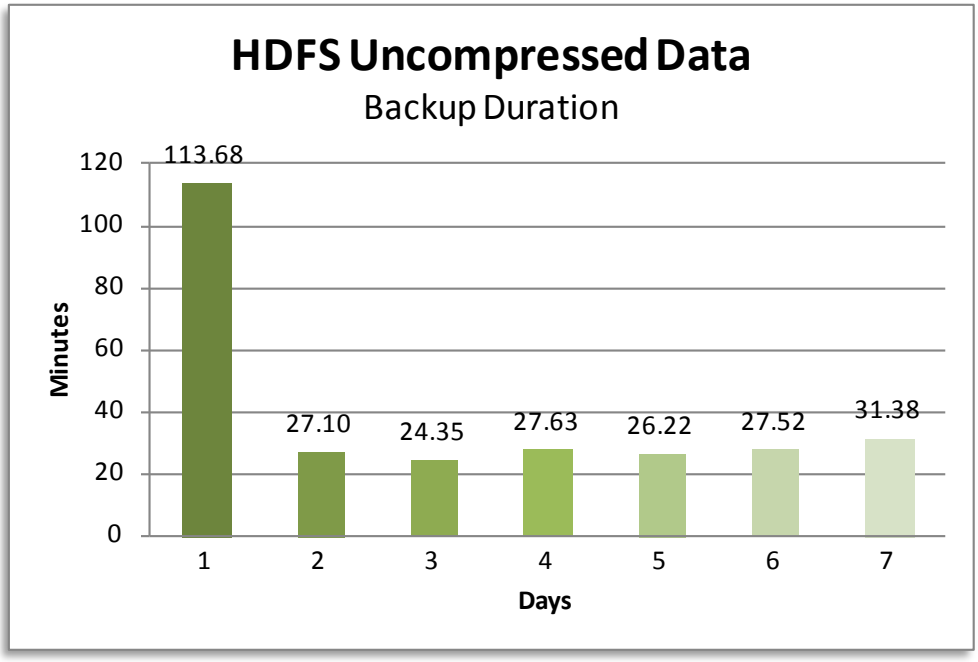

Figure 22. Greenplum HD, NFS: Backup duration with DistCp

Figure 23 illustrates the backup rate in TB/hour with uncompressed data. As seen previously in the backup duration test, the throughput is slower on the first backup. Using **DistCp** with Data Domain's deduplication technology results in an average backup speed of 5.37 TB/hour for backups 2 to 7.

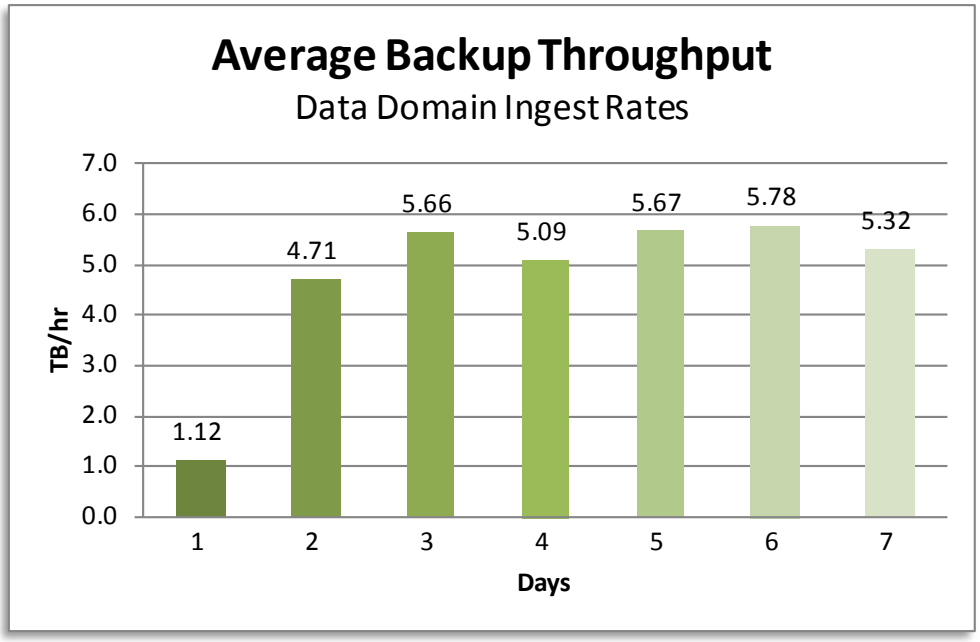

Figure 23. Greenplum HD, NFS: Data Domain ingest rate with DistCp

Due to the strength of Data Domain deduplication technology, there was an average 32.9x storage saving for each nightly backup. Shortly after the nightly backups began, the savings were significant. For example, on the second day, a full backup of 1,986.3 GiB used only 62.4 GiB, resulting in a 31.8x storage reduction.

Figure 24 illustrates the tremendous incremental savings that can be achieved on a daily basis. Over time, the savings are even greater.

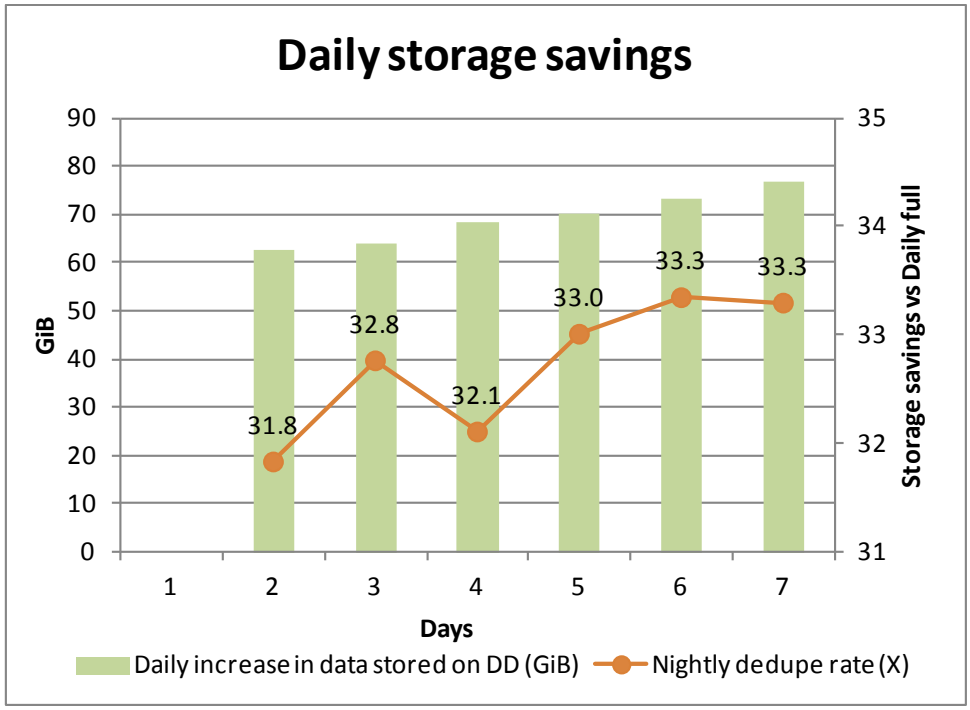

Figure 24. Greenplum HD, NFS: Storage saving after seven days

Note During all the backup tests, the DCA was idle.

Figure 25 illustrates the cumulative effect of this storage saving over a seven-day backup cycle. After seven days of running the backup, 15,471.8 GiB of data was backed up; however, only 1,382.7 GiB of storage was needed on the Data Domain deduplication storage system, resulting in a 11.2x storage savings. This 91.1 percent storage saving represents a significant saving in backup infrastructure and facility costs. Regular backups of large data warehouses are far more sustainable and much less costly than before.

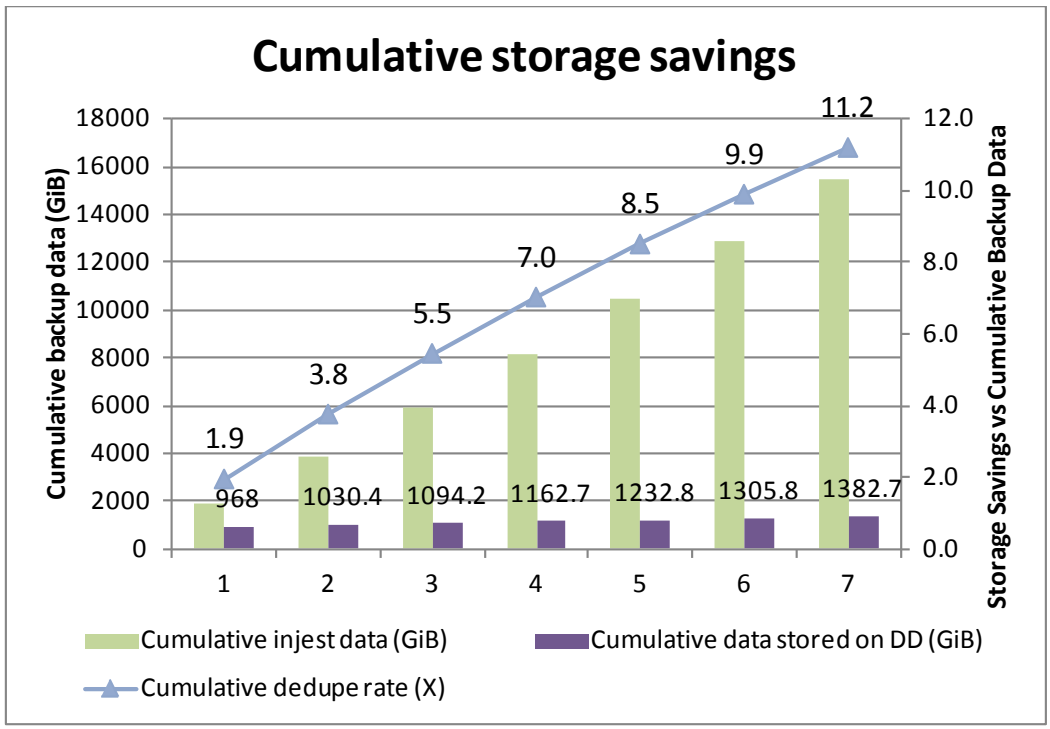

Figure 25. Greenplum HD, NFS: Cumulative storage savings over seven days

Test 2 results

In Test 2, we performed a restore of the Greenplum HD data using the backup data set from Test 1.

The DCA restore with uncompressed data produced the following results:

- The restore of a 2.54 TB uncompressed data set was achieved in 31.38 minutes.
- The average restore throughput of the uncompressed data set was 3.62 TB/hour.

Conclusion

Summary

EMC provides leading-edge technology to support the backup and recovery of the Greenplum DCA. This white paper demonstrates how a Data Domain deduplication storage system achieves this in the most simple, efficient, and cost-effective manner.

With this backup and recovery solution for the DCA, organizations can achieve:

- Faster backup and restores with minimized backup windows and maximized end-to-end recovery performance
- Efficient use of both infrastructure and people to support the business, leading to reduced operational and infrastructure costs
- Operational simplicity through ease of integration and management

Findings

The EMC Data Domain deduplication storage system provides cost-effective, long-term, onsite retention and protection of business-critical data and enables WAN-efficient networked disaster recovery.

With this solution, companies can expect to achieve:

- Operational ease and efficiency through the use of MPP architecture to efficiently back up, in parallel, across the network to the Data Domain system.
- Major space saving advantages using Data Domain inline deduplication—in this case, test results with DD Boost demonstrate a saving of 34.5x on nightly backup, and 11.3x cumulative saving over a week of uncompressed backups to the Data Domain system. One of the key benefits of Data Domain deduplication storage systems is to reduce the need for excessive amounts of backup storage.
- Fast restore times for returning uncompressed and deduplicated data to the DCA. The test results with DD Boost demonstrate that the combination of **gpcrondump** and Data Domain deduplication technology results in a backup of 13.08 TB/hour and a restore of 5.90 TB/hour. This enables the backup and recovery of a Greenplum full-rack DCA (36 TB uncompressed database) in under 2.75 and 6.1 hours respectively.
- Efficient backup and recovery of the Hadoop Distributed File System. The test results for Greenplum HD NFS backups demonstrate that the combination of **DistCp** and Data Domain deduplication technology provides an average backup of 5.37 TB/hour and a restore of 3.62 TB/hour.

Other observations from the test cycle:

- The **gpcrondump** compressed backup option **--rsyncable**, especially for NFS backups, will provide greater backup and restore throughput. However, this is only suitable for environments that make little or no changes to existing data. To gain the maximum benefit of EMC Data Domain deduplication technology, the **gpcrondump** uncompressed option **-z** is recommended.
- For gpcrondump NFS backups, when mounting the NFS shares on the DCA, EMC recommends using an rsize= 1048576 and wsize= 1048576 for best performance.

- When running DD Boost backups of the Greenplum DCA, a significant drop in backup throughput was observed when running a query load or an ingest load in parallel.
- For DD Boost backups, if the Data Domain system is dedicated to DCA backups, you should consider Data Domain **gz** compression.
- The **gpcrondump** uncompressed DD Boost backups have a significant performance improvement over **gpcrondump** uncompressed NFS backups.
- The rate of change directly affects the average Data Domain ingest rate. For this solution the backup performance varied as follows

gpcrondump - NFS	gpcrondump - DDBoost
1.37 TB/hour - initial backup	1.83 TB/hour - initial backup
7.02 TB/hour – Subsequent Backup (no new data or changes to existing data)	20.30 TB/hour– Subsequent Backup (no new data or changes to existing data)

For this solution the Data Domain ingest rate varied between these values depending on the rate of change to existing data and the percentage of new data added since the previous backup.

- For NFS backups with DirectIO enabled, no significant improvements or degradation to the backup performance are observed.
- A DCA expansion causes a reseed of the data so the first **gpcrondump** backup performed after the expansion should be treated as new backup regardless of previous backups of the old DCA configuration. It should be highlighted that a longer backup window will be required.
- Although the DD890 with DDOS 5.1 was used for this solution preliminary backup tests with the DD890 (DD0S 5.2) and the DD990 (DDOS 5.2) were also performed and the following performance gains were observed.

Backup type	DD890 with DDOS 5.1 v 5.2 Initial backup	DD890 v DD990 both with DDOS 5.2 Initial backup
NFS	54% increase with DDOS 5.2	With DirectIO enabled in both cases 55% increase with DDOS 5.2
DDBoost	59% increase with DDOS 5.2	68% increase with DDOS 5.2.

Subsequent backups, however, will not see as substantial a difference as the performance increase is only observed for changed or new data added since the previous backup. Data previously backed up will see a similar performance on both the DD890 and DD990. Therefore the larger the rate of change, the more the performance difference is observed.

- For Greenplum HD backups using the **DistCp** utility, task failures may be observed, especially during the initial backup. As long as the failed tasks complete successfully on subsequent retries, there is no impact. If necessary, the number of simultaneous copies and the **mapred.task.timeout** can be used to prevent task failures.

As the key findings show, the Data Domain DD890 has been used in this solution to provide an effective, disk-based backup target that significantly minimizes storage usage while providing long-term retention. Retention of many backups can be managed more easily and cost effectively when using the Data Domain deduplication storage system.

Note This white paper provides one example of results from using a Data Domain deduplication storage system. However, customer environments differ considerably in terms of backup polices and backup windows, so the rate of data change and data retention and deduplication ratios can vary. Therefore, each environment must be reviewed individually to size the Data Domain system to address each customer's particular requirements. Interested customers should contact their EMC representative or partner for more information.

References

White papers

For additional information, see *EMC Greenplum Data Computing Appliance: Architecture, Performance, and Functions —A Detailed Review*.

Product documentation

For additional information, see the product documents listed below.

- *Greenplum Database 4.2 Administrator Guide*
- *Greenplum Chorus 2.2 Installation Guide*
- *Greenplum Data Computing Appliance Getting Started Guide*
- *Greenplum Data Computing Appliance Administration Guide*
- *Data Domain Operating System (DD OS) Administration Guide*
- *Data Domain Operating System (DD OS) Initial Configuration Guide*
- *Data Domain Installation and Setup Guide*

Supporting information

Interconnect 1: Converting ports 18 and 19 from a LAG to switch ports

After logging into the switch, run the following commands:

```
i-sw-1:admin> cmsh
i-sw-1#show run
```

Ports 18 and 19 will look like this:

```
interface TenGigabitEthernet 0/18
 channel-group 2 mode active type brocade
 no shutdown
 lacp timeout short
!
interface TenGigabitEthernet 0/19
 channel-group 2 mode active type brocade
 no shutdown
 lacp timeout short
```

To setup ports 18 and 19 as switch ports, run the following commands:

```
i-sw-1#config t
Enter configuration commands, one per line.  End with CNTL/Z.
i-sw-1(config)#no interface Port-channel 2
i-sw-1(config)#interface TenGigabitEthernet 0/18
i-sw-1(conf-if-te-0/18)#switchport
i-sw-1(conf-if-te-0/18)#switchport mode access
i-sw-1(conf-if-te-0/18)#vlan classifier activate group 2 vlan 199
i-sw-1(conf-if-te-0/18)#no shutdown
i-sw-1(conf-if-te-0/18)#spanning-tree edgeport
i-sw-1(conf-if-te-0/18)#spanning-tree edgeport bpdu-guard
i-sw-1(conf-if-te-0/18)#interface TenGigabitEthernet 0/19
i-sw-1(conf-if-te-0/19)#switchport
i-sw-1(conf-if-te-0/19)#switchport mode access
i-sw-1(conf-if-te-0/19)#vlan classifier activate group 2 vlan 199
i-sw-1(conf-if-te-0/19)#no shutdown
i-sw-1(conf-if-te-0/19)#spanning-tree edgeport
i-sw-1(conf-if-te-0/19)#spanning-tree edgeport bpdu-guard
i-sw-1(conf-if-te-0/19)#exit
i-sw-1(config)#exit
i-sw-1#write mem
Overwrite the startup config file (y/n): y
Building configuration...
i-sw-1#show run
```

Ports 18 and 19 will now look like this:

```
interface TenGigabitEthernet 0/18
 switchport
 switchport mode access
 vlan classifier activate group 2 vlan 199
 no shutdown
 spanning-tree edgeport
 spanning-tree edgeport bpdu-guard
!
interface TenGigabitEthernet 0/19
 switchport
 switchport mode access
```

```
 vlan classifier activate group 2 vlan 199
 no shutdown
 spanning-tree edgeport
 spanning-tree edgeport bpdu-guard
```

Plug the sfp into Port 19; then connect the fiber cable from Port 19 to Data Domain eth4a, for example.

Interconnect 2: Converting ports 18 and 19 from a LAG to switch ports

After logging into the switch, run the following commands:

```
i-sw-2:admin> cmsh
i-sw-2#show run
```

Ports 18 and 19 will look like this:

```
interface TenGigabitEthernet 0/18
 channel-group 2 mode active type brocade
 no shutdown
 lacp timeout short
!
interface TenGigabitEthernet 0/19
 channel-group 2 mode active type brocade
 no shutdown
 lacp timeout short
```

To setup ports 18 and 19 as switch ports, run the following commands:

```
i-sw-2#config t
Enter configuration commands, one per line.  End with CNTL/Z.
i-sw-2(config)#no interface Port-channel 2
i-sw-2(config)#interface TenGigabitEthernet 0/18
i-sw-2(conf-if-te-0/18)#switchport
i-sw-2(conf-if-te-0/18)#switchport mode access
i-sw-2(conf-if-te-0/18)#vlan classifier activate group 2 vlan 299
i-sw-2(conf-if-te-0/18)#no shutdown
i-sw-2(conf-if-te-0/18)#spanning-tree edgeport
i-sw-2(conf-if-te-0/18)#spanning-tree edgeport bpdu-guard
i-sw-2(conf-if-te-0/18)#interface TenGigabitEthernet 0/19
i-sw-2(conf-if-te-0/19)#switchport
i-sw-2(conf-if-te-0/19)#switchport mode access
i-sw-2(conf-if-te-0/19)#vlan classifier activate group 2 vlan 299
i-sw-2(conf-if-te-0/19)#no shutdown
i-sw-2(conf-if-te-0/19)#spanning-tree edgeport
i-sw-2(conf-if-te-0/19)#spanning-tree edgeport bpdu-guard
i-sw-2(conf-if-te-0/19)#exit
i-sw-2(config)#exit
i-sw-2#write mem
Overwrite the startup config file (y/n): y
Building configuration...
i-sw-2#show run
```

Ports 18 and 19 will now look like this:

```
interface TenGigabitEthernet 0/18
 switchport
 switchport mode access
 vlan classifier activate group 2 vlan 299
 no shutdown
 spanning-tree edgeport
```

```
 spanning-tree edgeport bpdu-guard
!
interface TenGigabitEthernet 0/19
 switchport
 switchport mode access
 vlan classifier activate group 2 vlan 299
 no shutdown
 spanning-tree edgeport
 spanning-tree edgeport bpdu-guard
```

Plug the sfp into Port 19; then connect the fiber cable from Port 19 to Data Domain eth4b, for example.