DIASER beta-2 - Technical Manual v 1.0.9

Damian L Brasher - 02/01/2011

This work by Damian L Brasher is licensed under a Creative Commons Attribution-Share Alike 2.0 UK: England & Wales License.

Index

1 Introduction
1.1 Feature overview

2 Explanation of the overall design
2.1 Design philosophy
2.2 The storage architecture
2.3 Integrated approach
2.4 Limitations
2.5 Why Linux?

3 The package and contents
3.1 Downloading and unpacking
3.2 Main source file
3.3 Configuration files
3.4 Example backup software configuration
3.5 Licence
3.6 Documentation

4 Requirements
4.1 Hardware
4.2 Software
4.3 Skills

5 Primary scripts
5.1 diaser
5.2 tab_$.pl
5.3 hvautoc_$.pl
5.4 fill_diaser.pl

6 Explanation of features
6.1 Geographical distribution
6.2 Security
6.3 SE Linux and AppArmor
6.4 Upgrade and modify
6.5 Filling or loading
6.6 Non distinct binary volumes
6.7 Logging
6.8 Archive retrieval
6.9 Data and node migration
6.10 Reporting and monitoring
6.11 Multiple instances
6.12 Extending operation
6.13 Pruning old volumes
6.14 Time zone compensation and leap years
6.15 Digital volume check-sum or stamp
6.16 Complete removal

7 Configuration
7.1 diaser.conf
7.2 Number of years of expected operation
7.3 First year of operation
7.4 Start time of phases
7.5 Node IP address's
7.6 OpenSSH ports
7.7 Dry run mode
7.8 Lowest maximum bandwidth (LMB)
7.9 Time zone compensation
7.10 Working diaser account name
7.11 Time out
7.12 Home directories
7.13 Fill start time
7.14 Volume directory
7.15 Differential or constant name prefix
7.16 Collect Full volume or not
7.17 Collect Full volume on which day
7.18 Full volume prefix
7.19 More than one configuration file

8 Installation

9 Command Line Options
9.1 --help
9.2 --bandwidth
9.3 --configure
9.4 --extend
9.5 --install
9.6 --list
9.7 --lock
9.8 --logs
9.9 --migrate
9.10 --modify
9.11 --pause
9.12 --recreate
9.13 --remove
9.14 --resume
9.15 --retrieve
9.16 --stats
9.17 --stop
9.18 --upgrade
9.19 --version

10 Operation
10.1 Stop
10.2 Pause
10.3 Resume
10.4 Hard Lock
10.5 Migrate node

11 The Code
11.1 Why Perl?
11.2 Style
11.3 Modules

11.4 Error handling
11.5 Contribute

12 On-line resources
12.1 Website
12.2 SourceForge
12.3 Mailing list
12.4 DIAP/LTASP and early project memory

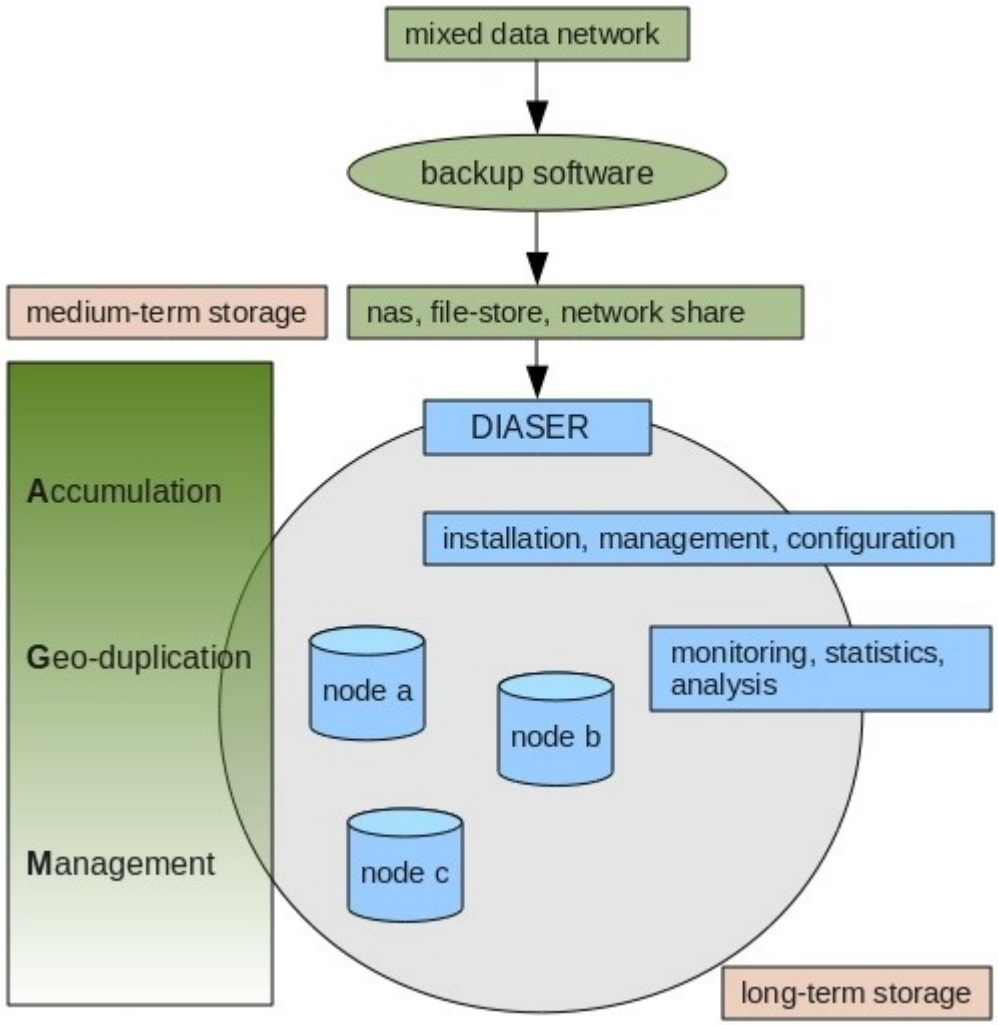
APPENDIX

A Tables and calculations
B Glossary of terms

C Appliances **1 Introduction**

DIASER is for long term digital archive storage, it securely...

1) Accumulates
2) Geo-Duplicates
3) Manages



DIASER has been created to solve mid-range and below, long term archiving requirements of the SME, a data vault application. Where tape has been deployed in the past DIASER now offers an alternative solution designed to be more robust and manageable in the long term than simple NAS devices or disk based storage alone. This manual is designed to assist the systems administrator providing; a detailed technical overview of the system and it's components parts, how to plan deployment, installation, storage

space calculations, an overview of the code base and other available resources.

1.1 Feature overview

- Engineered storage architecture - for high performance, quality and reliability
- Exists and operates in dedicated user accounts - self-contained, sealed and easy to migrate environment
- Flat, human readable storage structure - to ensure data is retrievable and code is comprehensible
- Highly resilient and robust - to minimise the risk of data loss over many years
- Large volume capacity (TB's) - extremely low cost storage
- Low operational and maintenance overheads - reduced cost of ownership
- Manage independently from a Perl enabled workstation - for a highly manageable solution
- Manage long-term archives - software that will guarantee retrieval over time
- Migratable nodes - replace hardware without changing the backup software infrastructure
- Multiple configuration files for multiple installations - to simplify multiple installation management
- Perl installer and configurator - for a stable and mature cross-platform environment
- Powered by rsync and OpenSSH - to utilise the powerfull rsync data transfer algorithm
- Repair tool - allowing broken nodes to be rebuilt
- Scalable - grows with your disk and network capacity
- Secure design - to prevent data compromises and minimise the risk of vulnerabilities
- Simple configuration file and format - to ease installation and maintenance overheads
- Standards compliant - allowing tight systems integration and interoperability
- Stats and analysis tools built-in - assisting the deployment manager and administrators
- Straightforward upgrade procedure - to allow new features, enhancements and fixes to be deployed quickly
- Use commodity disks for robust storage - reduce long-term storage costs
- UTC Time Zone compensation mechanism - nodes can exist across time zones
- Works with existing backup infrastructures - seamlessly integrate without duplicating deployment costs
- 3 replicating storage nodes - for optimal performance vs maximum data redundancy

Cloud based computing has taken off the last few years. DIASER is an ideal application for cloud computing deployment as well as an archiving framework solution. Once implemented the system is invisible to users but allows them to do more. Cloud computing is a popular term, a useful way of communicating a complex collection of technologies. The use of virtual machines in a distributed environment has many advantages. The problem many people foresee with cloud computing is lock-in-in and loss of control of data and increased cost of services. DIASER allows an organisation to build private storage clouds using existing resources as you will see in this technical manual. The result is control over your long term could based storage in terms of administration and resources as soon as the system is deployed and beyond. This means that data can be migrated when you want to without penalties from a 3rd party provider.

With security in mind at all times DIASER is based on a carefully designed robust storage architecture called LTASP, Long Term Archive Storage Protocol. This means consistency is ensured now and in the future. The design phase involved four years of careful evaluation and testing. DIASER is open source software using GPL the GPL v3 licence model so users can enjoy the benefits the of open development methodology. Simplicity of design and reuse of code and readily available resources is key to power of this system. A strong design philosophy has been cultivated and adhered to for the benefit of all users. DIASER is written by a systems administrator for systems administrators but potential benefits to an SME, it's IT manager, CEO and committee have been the highest priority throughout all stages of the design process. The DIASER implementation is targeted primarily at education, hence the name Distributed Internet Archive System for Educational Repositories however the system can be downloaded and deployed by any SME. DIASER is designed to be extremely future proof. As an Open Source product

minimise the risks associated with vendor lock-in and data retrieval.

More features are planned for the future and the most current development road-map can be viewed here: http://diaser.svn.sourceforge.net/viewvc/diaser/ROADMAP_DEV.

2 Explanation of the overall design

2.1 Design philosophy

Archiving and backup is art and science. For me a philosophy has evolved over the years I have been a systems administrator and I applied them to the design of LTASP and DIASER:

Maximise: Storage capacity, availability of data, data restoration and recovery speed, scalability, modularity, cross-platform deployment, resilience and robustness.
Minimise: Operating bandwidth overhead, impact of network outages, management overheads, support costs.
Simplify: Development cycle, deployment, data recovery, operation, integration with existing systems.

2.2 The storage architecture

To maintain archives over a number of years requires organisation. For this reason DIASER builds a set of slots/directories on each node in advance which correspond to date. This is done in advance and not generated as required for a number of reasons. As the system operates across networks and network connections can have variable performance or be down completely creating a year or more slots (slots roughly equate to a single tape) of storage upon installation ensures that the directories are named and therefore dated correctly. This ensures if data is not copied correctly we can identify failure even without log data. Log data may or may not be created on a node but empty slots are indicative of copy or network failure. Computers are not the best time keepers left to their own devices. If the storage structure creation is undertaken when all nodes are known to be synchronised then accuracy of the storage structure is ensured. If slots were created on the fly and node time was not synchronised for any reason, bios changes, other software changes the time inadvertently and so on, inaccuracies could occur. The structure is human readable too and simply put; empty slots are easier to read and parse than missing slots.

Storing old archives in a well defined data storage structure is very important. This means DIASER can be deployed in the past, i.e. 2007 onwards. Then the system can be manually filled with old data like a filling cabinet and default automatic operation simply continues.

The system is optimised to store a combination of Full and differential volumes. Fulls created at the beginning of the month and Diffs during the month. However this does not preclude storage of constant volume sizes, i.e. the storage of CCTV video footage but calculations must reflect this kind of storage mode. The recommended data vault operation will make use of certain directory structures in each month; Full01 and Full02. Full01 will store a Full volume at the beginning of the month and skip d1. Full02 is there for additional redundancy and to cope with the scenario where the current month is the last (this is not default behaviour).

There are two parts to the architecture, that described above and the data transfer mechanism. The data transfers are initiated by an internal structure called the hyper virtual auto-changer, a virtual concept drawn from the mechanical tape changer. The well used tool rsync is a key component of this mechanism and it's features are utilised fully. DIASER installs onto three Linux nodes for optimal data storage resilience. No parity is used, this means complete data can be stored and retrieved if a single node is isolated from others.

DIASER can be managed from any Perl 5.8.8 and network enabled workstation or from a node if preferred.

This section can be skipped and here for the very technically minded. Taking a deeper look at the

architecture, also see section 6.4 filling or loading. Nodes A and B both contain d0's. This structure allow copy phases to simply and accurately span different days, if data was set to be copied directly from Node A d5 then midnight passed +-1 day will have to be factored depending on the point of reference - node$. Filling of DIASER can then occur well in advance thus keeping the copy phases operationally contained and therefore greater control over operation, implementation and readability. The filling occurs outside the LMB calculations and can be at a much slower rate. This means LMB calculations remain applicable to both phases. d0 also acts as a buffer; original copies exists if an internal copy fails, allows simultaneous copies i.e. A; d0->d5 and B; d0->d0 otherwise the second copy would have to wait and begin safely only after completion. d0 can be tested for a successful fill before phases begin.

The concept of node role assists towards an optimised architecture, which differs depending on the node role. To allow the roles to be practically changed and for simplified fail-over implementation the directory structure is identical on each node, whether it is node A, B or C. The difference between roles is subtle but important:

Role A: uses d0 contain in each month, designed to be closest to original backup volume source. Utilised in phase1 only.
Role B: Utilised in phase1 and phase2. Only accepts data during phases.
Role C: Utilised in phase2 only. Only accepts data during phases.

2.3 Integrated approach

DIASER makes use of existing resources where possible. This results in streamlined software tightly integrated with the POSIX, Linux computing environment. Using Perl for this task ensures GNU tools are used for tasks instead of re-writing functionality unnecessarily. Use of the the common Linux home directory environment, cron, OpenSSH and rsync. Perl is commonly installed on most Linux operating systems by default and only the core is required on the storage nodes. This allows for very simple installation and management. By using user space the system is contained and a layer away from it's host root environment which has many positive implications not least better security and deployment modularity. DIASER will store backup volumes generated by most backup software products, at least all those that can write volumes to disk lessening operation, integration and installation overheads. Volumes are are defined as resembling a single tape entity.

2.4 Limitations

Storage space is limited by bandwidth. At my reference installation site I spent half an hour with the IT manager to decide the relative importance of the organisations data. To this end we managed to select about 30% of all data generated on a regular basis and pipe this into DIASER. This practical approach coupled with compression, data-data de-duplication may be available, means that the organisations critical data is stored using DIASER. Node A is a single point of failure. This is the node in network terms closest to the backup server and if it failed data will cease to transfer. However plans exist to allow node A bypass. Even if node A did prevent data transfer it is expected the systems administrator has the skills and access to resolve any issues.

2.5 Why Linux

Linux should not be underestimated for its appropriateness as a storage platform for many reasons. The cost of obtaining Linux is very low and essentially free as in libre and to obtain and use, supported versions can be very good value too. Linux is widely available and has lightweight resource requirements. Licence issues are avoided. Organisations that need the flexibility of deployment with low initial purchase costs can do so when they deploy Linux. Linux is extremely robust under most circumstances, i.e. the ext3 file system under normal circumstances does not require regular de-fragmentation which makes it ideally suited to storage environments. Many of the tools required to enable DIASER are included in standard distributions, even small installations without a GUI or a windowing system. This means DIASER is streamlined, lightweight and does not attempt to needlessly duplicate existing code, i.e. rsync.

3 The package and contents

3.1 Downloading and unpacking

DIASER is currently supplied by anonymous download from SourceForge as a diaser-1.0.$.tar.gz (this contains everything in the subversion repository), rpm, dist-tarball or deb installation. rpm dependencies will be automatically installed with yum. Makefile as root will allow installation; make, make install. deb package still requires extra dependencies. See INSTALL and section 8 of this manual.

3.2 Main source file

diaser - this file unpacks more embedded scripts which are sent to the nodes upon installation, modification and upgrade.

3.3 Configuration files

diaser.conf - this is the main configuration file. See section 7 for configuration guidance. A second configuration file can be created manually for development or second deployments. Keep your configuration files in separate directories or rename them. If no configuration file is present then the default values set in diaser will be used, this will not lead to successful deployment.

Also see section 7.19 for use of more than one configuration file.

3.4 Example backup software configuration

helper_scripts/bacula-dir.conf.extract

To fill DIASER with backup volumes created by backup software you need to name volumes in a certain way. This example configuration comes from the Open Source backup software called Bacula. If you use Bacula you can implement volume creation is an identical fashion. If not then use this file as a guide. The scripts generated by the installer residing on node A are called fill_diaser.pl. As the names suggest these collect volumes generated by your backup software, perhaps stored on a share mounted by node A or directly backed up to node A, and fill DIASER with pre-defined named volumes.

3.5 Licence

This software is licenced under GPL V3 - gpl.txt and fdl-1.2.txt. The website is licenced under fdl-1-2.txt

The manual, DiaserSystem.png and DiaserDocsv1.1.pdf are licenced under Creative Commons Attribution-Share Alike 2.0 UK: England & Wales Licence.

3.6 Documentation

Located in directory docs. This includes this technical manual docs/manual.txt .html or .pdf and diagrammatic overview docs/overview.png.

Importantly INSTALL contains a quick start guide.

More theoretical documentation is available from http://www.diap.org.uk and don't forget to check http://www.diaser.org.uk for up to project date news and other information.

A man page is also installed.

4 Requirements

4.1 Hardware

Workstation, 1GHz CPU or above, 500MB Ram and network connection. You can also use a node as as the installation platform but you need to ensure all the Perl modules listed below for the workstation are available.

3 x Linux storage nodes (can use VM's) with root access for initial setup. Anything above 1GHz 32bit or 64bit with 500MB Ram. Enough disk space. I'll make all this much simpler to calculate when I have finished subroutine calculate_lmb, see appendix A, tables and calculations.

LAN or WAN connection between each server and workstation, the 3 machines must be able to, at least notionally, ping one another. Nodes can be connected across a Virtual Private Network if necessary.

4.2 Software

Minimum Perl v5.8.8 enabled (Perl v5.10.0 is recommended for best performance) **workstation** with Perl modules:

Net::SSH::Perl, Net::SFTP, Getopt::Long, AppConfig, Term::ReadKey and Data::Password. Optional for the --bandwidth tool gnuplot v4.2.

Install modules i.e. as root]#yum -y install perl-Net-SSH-Perl or cpan>install Net::SSH::Perl

Automatic module installation occurs when installing using the rpm release.

Nodes Perl Core (v5.8.8 or above) File::Find (installed as default with most distributions). SSH server on each node, not necessarily port 22.

Each node must run services; sshd, crontab, iptables ssh port open, ntpd, rsync (non daemon).

4.3 Skills

It is recommended the administrator have at least these skills:

Bash command line - ability to move around directories, create files and directories, set permissions and add and remove user accounts. Knowledge of SSH logins, text editor and adding and removing software. Basic knowledge of rsync and the ability to effectively use scp. Use of commands less and cat. Ability to install Perl modules and check versions.

Less important are some Perl scripting abilities, Basic bash scripting skills may also help.

5 Primary scripts

5.1 diaser

The primary script containing most of the DIASER code. Code embedded within diaser is unpacked and copied to nodes with variables set by the user. For upgrades and configuration changes code is again unpacked and copied over to nodes as required.

5.2 tab_$.pl

One for each node and contains the crontab definitions which trigger the internal diaser data copies managed by the scripts hvauto_$.pl. The cron job run every hour i.e. 0 * * * * ~/hvautoc_a.pl and the script reads the local system time, compares the the user set copy phase and if there is a match will initiate data transfer. The script logs to the node, log_$, as does rsync.

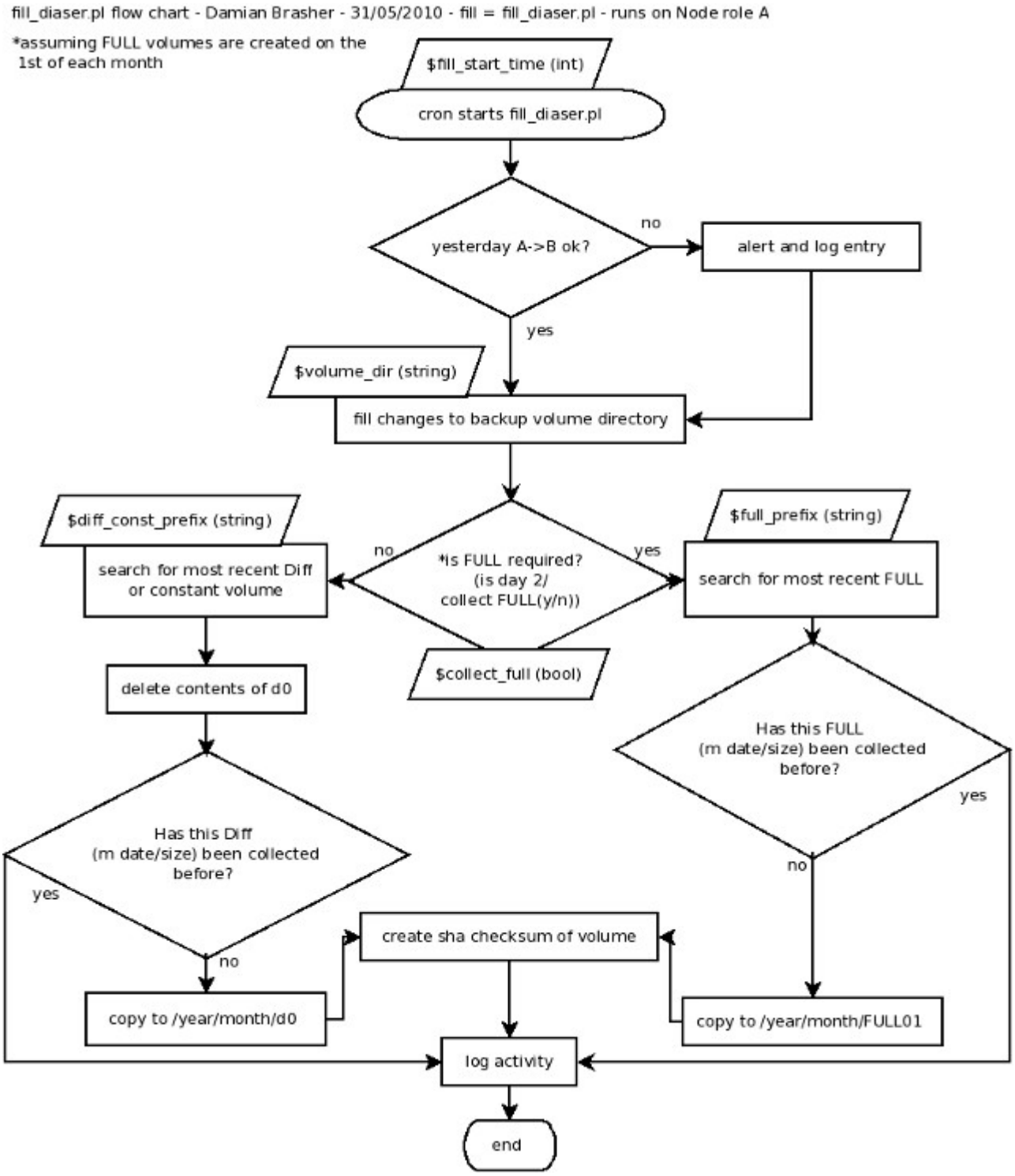
5.3 hvautoc_$.pl

Each node has a single hvautoc_$.pl script. This script is triggered every hour and depending on the times set by the user variable, HOUR1 and HOUR2 they initiate the rsync data transfers. If the user modifies variables then these updates can be copied to the nodes by replacing the hvautoc_$.pl scripts.

5.4 fill_diaser.pl

This script resides only on node A. This is responsible for filling the correct slot with data fed into DIASER by the user. The script is called by cron job set when configuring or modifying DIASER. The script copies the latest created of either Full, Differential or constant volume types to the DIASER directory to either Full01 or d0. Aside from the cron job time there are a number of variables that can be user configured including the volume directory, that is where your backup software stores volumes and the volume prefix, i.e. fullbackup... for Full volumes.

Filling is designed to be as simple as possible. Volumes on your file store are assumed to be read/write by user id: $your_diaser_uid. This flow chart provides a detailed overview of the fill process, everything apart from the node A->B copy check has been implemented:



fill_diaser.pl automatically clears out the drop off directory ad0 after the contents of which would normally have been transferred to other slots as specified by the architecture.

6 Explanation of features

6.1 Geographical distribution

Tapes can be moved from site to site and often are. To emulate this ability distributing data provides geographical redundancy. A simple mirror of a NAS device is one way to achieve this but to spread over three nodes can be difficult to manage. DIASER is a self contained wrapper around the long term archiving across three nodes. We believe the extra resilience provided by storing in three geographical locations give your archives the protection needed for long term planning and data retrieval. Ensuring your archives are safe means a better chance of recovering data when you need it. Being a disk based

solution will help render your data more accessible in may scenarios. Planning your installation is important and as the system may run for years spending time before deployment will pay off. DIASER is ready for trail and evaluation. Your chosen storage nodes may also be equipped with RAID. This is highly recommended.

6.2 Security

These security precautions have been implemented: The primary script, diaser, does not store any passwords on file. Passwords are stored in memory temporarily while the script runs. When a password is requested the entry view is hidden. New DIASER account passwords are quality checked and a warning given if not secure. Root passwords are only requested when the system is installed and removed. DIASER exists and runs in user space. All network communication is handled by OpenSSH. A unique RSA certificate is generated so the nodes can use password-less logins to transfer data and communication during normal operation. Password-less login certificates can be regenerated using the modify switch --upgrade. A kind of emergency account lock can be initiated with the switch --lock.

The perl module Net::SSH::Perl and Net::SFTP are used for all SSH communications and file transfers initiated by the system. Rsync uses SSH to transfer data. It is possible to use different port to the standard SSH port 22 and individually set these for each node.

An sha256sum checksum and a date stamp file is created a every volume enters DIASER in a format similar to:

4865c5bdf3cf64709acd797688db5b337e7c8643
2009/mth7/Full01/fullbackup7
Tue Jul 21 07:10:28 BST 2009

For extra security DIASER can run within a Virtual Private Network. It is recommended encrypted partitions are used for DIASER, i.e. when deploying an external USB hard drive.

/dev/sdb can be an externally attached USB2 hard disk drive i.e. replace with the disk chosen on your system.

```
# Create a new partition on the disk

fdisk /dev/sdb

# Generate a mapping and LUKS partition

cryptsetup --verbose --verify-passphrase luksFormat /dev/sdb1

cryptsetup luksOpen /dev/sdb1 sdb1

# Format the partition

mkfs.ext3 -j /dev/mapper/sdb1

# Mount the partition for the first time

mount /dev/mapper/sdb1 /mnt/crypt/

df -h

# Open and mount the device after reboot or disk removal

cryptsetup luksOpen /dev/sdb1 sdb1

mount /dev/mapper/sdb1 /mnt/crypt/

# Umount and close
```

```
umount /mnt/crypt/
cryptsetup luksClose sdb1
```

6.3 SE Linux and AppArmor

No problems observed during either installation or operation.

6.4 Upgrade and modify

Currently modify switch, see below, is still under review. For now the upgrade switch sends modifications and upgrades to the nodes. This does not and will not modify the archive storage directory structure. Changes to settings and development improvements can be sent using this option. If you use newer version than your previous then follow these steps:

1) rename your current diaser_rel
2) unpack the download, see section 3.1
3) copy your previous diaser.conf to the new diaser_rel
4) run]$diaser --upgrade to update your DIASER installation

6.5 Filling or loading

See section 5.4.

The initial entry point for data, d0 (node A, directory 0), resides in each monthly segment and not a single d0 in the root directory. This lessens the risk of deleting or overwriting archive data that may not, for whatever reason, have been successfully transferred to the other nodes. If connection to node B fails there will be at least two copies of the file in d0 and d30 or whatever the last day of the month happens to be, before another Full is generated and the next months d0 is cleared and filled. This adds more resilience at little extra cost. Also, if copies are only set to occur once a month and the copy failed as before and this was not noticed until after the next copy last months data will have been deleted and only a single copy stored.

6.6 Non distinct binary volumes

The volumes which have been described are binary files, like those created by Bacula. Other backup software generate directories which need to some processing before they can be collected by DIASER.

There are a number of problems to avoid to ensure DIASER operates non-destructively, so instead of manipulating the directories in your data store I suggest you use a script to create tar volumes of the archives you want to be collected. Here is a psudo code suggestion of how this might be achieved.

```
    # non distinct binary volume alternative collection
    # run as a cron job independently of DIASER

    sub non_full_binary {
        look for directories, if directories
        ls
        if($directories) {

        check for a previous tar Full

        -> if no Full this month then tar/shasum/date
        any directories collected for Full -> Full01 slot and
        name with the chosen Full volumes prefix.

        check for a previous tar Diff

        -> if Full this month then create a
        tar/shasum/date Diff against it for the day slot
        name with the chosen Diff volumes prefix.
```

```
}
```

6.7 Logging

Log files are kept on all nodes and named log_$ where $ is the node; a, b or c. The scripts hvautoc_$.pl, fill_diaser.pl and all rsync transfers log to these files. The log files are created automatically as soon as the system begins operation. All entries are contain [diaser_hvautoc_$] or [diaser_fill] where $ is the node; a, b or c.

6.8 Archive retrieval

Either use the simple tool provided using the --retrieve option, which also has additional command line options or login to nodes directly and use scp. The retrieval tool will walk you through a set of questions then list files for you to pick and transfer. The file will retain it's name and be located in the diaser_rel directory.

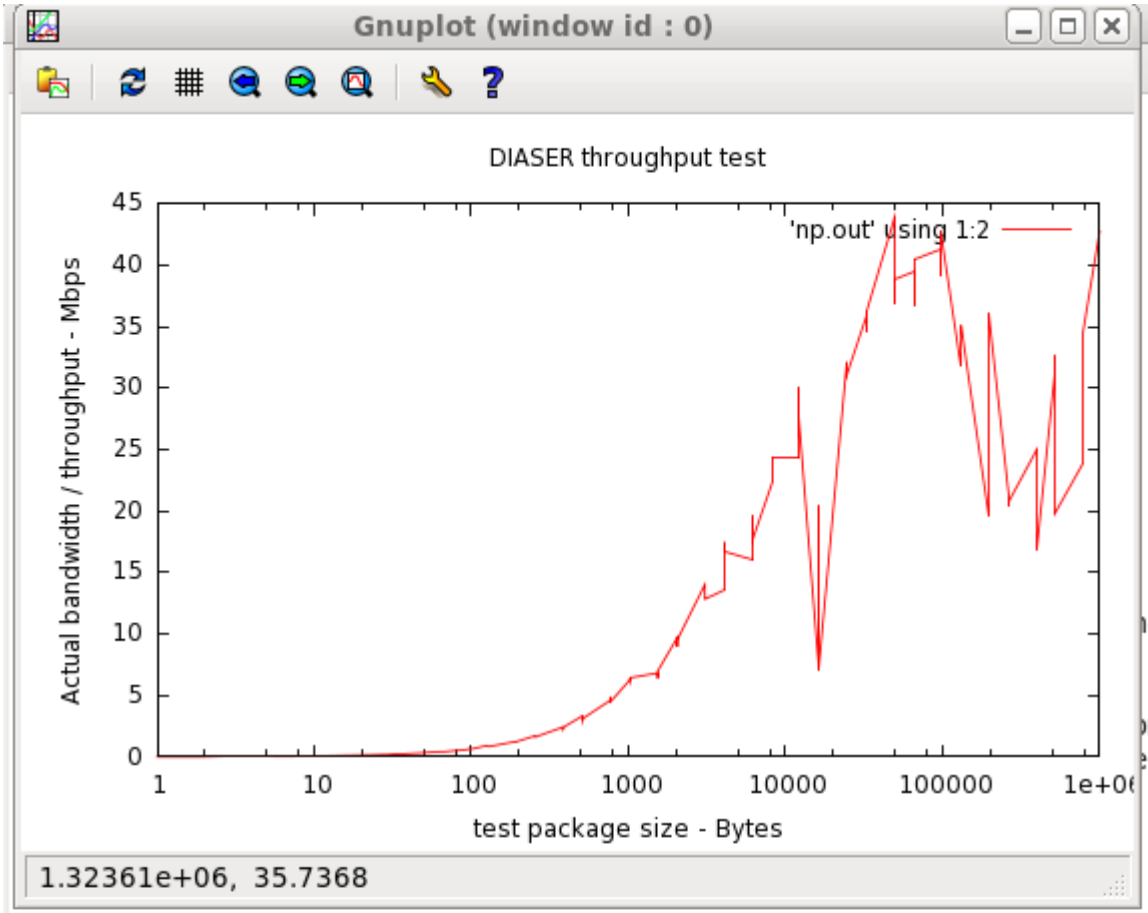
If using cp, scp, rsync or other native tools. The directory structure is human readable and matching the required date to directories can be easily achieved i.e on node $ the archives stored on date June 25th 2009 can be found in ../diaser/2009/mth6/d26. Navigate to the directory and copy the contents to the required recovery destination. It is assumed you have the tools to extract your data provided by your backup software vendor. It is recommended you also archive any backup catalogues or tools generated and provided with your usual backup software.

6.9 Data and node migration

Node migration can be achieved using the --migrate tool.

6.10 Reporting and monitoring

Bandwidth throughput calculations can been made using the --bandwidth tool. See section 9.3 for more details. This is an example screenshot of the ouput:




```
Please enter (y)yes or (n)no
y
NetPIPE already downloaded
cc -O -g ./src/netpipe.c ./src/tcp.c -DTCP -o NPtcp -I./src
Which node will be the listener? [A, B or C]
A
Which node will be the receiver? [A, B or C]
B
Please enter a password for all the diaser accounts
Please enter the password again
Thank you

NPtcp and run listener.pl transferred to listener
NPtcp and run_sender.pl transferred to sender
NPtcp initiated on listener node
NPtcp initiated on sender node
|||||||||||||||||||| <-- 99.5% Complete -->
```

6.11 Multiple instances

Share disk space with other organisations or groups by using a different account name and staggering or alternating the transfer times (phases) or lowering the LMB - lowest maximum bandwidth between nodes. See diaser.conf. diaser will allow the use of more than one configuration file. See section 7.19.

Also if more than one pair of phases is required, i.e. a morning session and an night session than two instances on the same nodes will archive at alternative phase times. If one instance contains FULL volumes then the second does not necessarily need to archive these as well thus saving disk space.

6.12 Extending operation

Operation can be extended. Minimum recommended is two years. You can set DIASER to install to 10 or even 20 years, which means 10-20 years of archive directory structure will be created. Deployment can represent the past if required then manually filled with previously generated archive data.

6.13 Pruning old volumes

Not yet implemented. This will allow the user to remove old archives from DIASER freeing up disk space.

6.14 Time zone compensation and leap years

Time zone compensation allows all the nodes to work together across time zones. The user is asked for the time zone in UTC+(integer). UTC +/- integer value for node A, B and C; if node A is BST = UTC+1, so use 0 as daylight saving is usually automatic on most systems. For three servers in the same time zone use the same offset integer value for each node.

The scripts hvautoc_$.pl all contain an algorithm that will ensure proper interpretation of leap year occurrences.

6.15 Digital volume check-sum or stamp

Generating a unique check-sum or stamp and date stamp as a volume enters DIASER to be stored along side the volume.

6.16 Complete removal

This will completely remove all DIASER components and all archive data stored within the system. Data recovery is not possible after this operation has been performed.

7 Configuration

7.1 diaser.conf

This supplied configuration can be adjusted to suit your deployment requirements. Each parameter is in uppercase the name of which must not change. Change the values to the right of each parameter with a space in between. The default values are there to guide you for your choice. i.e. NODE_A 0.0.0.0 can be interpreted as NODE_A 192.168.2.1. Use the same case and value type for your chosen values as the defaults.

7.2 Number of years of expected operation

NUM_YEARS

Minimum recommended 2 the default is 3.

7.3 First year of operation

START_YEAR

This is the year when DIASER begins operation. Would usually be the current year.

7.4 Start time of phases

HOUR1

HOUR2

DIASER operates in two phases. Phase one identified by HOUR1 and phase two identified by the variable HOUR2. The operation is split into two phases, these can be at any time over a 24 hour period. It

is assumed that the start time is based on your local timezone, i.e. BST or UTC+1. It is recommended to set the phases to early in the morning to avoid using day time bandwidth resources. Once set the operation can be reset by sending a new configuration from diaser. The operation is fixed for at the same time every day once set. Using two phases optimises the use of resources when transferring internally on a node and between nodes and prevents simultaneous transfers from interfering with each other as well as simplifying the management and tracking of transfers.

7.5 Node IP address's

NODE_A

NODE_C

NODE_B

7.6 OpenSSH ports

PORT_A

PORT_B

PORT_C

Change from the default port 22.

7.7 Dry run mode

DRY_RUN

Copies are initiated but no archive data is transferred. This can be used for testing, debugging and trails.

Can be toggled at any time and the new setting transferred as for all settings in this section.

7.8 Lowest maximum bandwidth (LMB)

LOW_MAX_BW

BANDWIDTH control, please enter the Maximum speed in KBPS of your slowest network connection between either A->B or B->C or C->B. I recommend you run some test transfers between nodes using scp, also don't assume the bandwidth will remain constant throughout the cycle so you may need to run some long term viability tests. This feature will be implemented automatically with the subroutine calculate_lmb(). Adjust if you install more than one diaser instance on a single disk or machine. Default is 12500 KBytes per second / 100 Mbits per second

7.9 Time zone compensation

For deployments that span different time zones. UTC +/- integer value for node A, B and C; if node A is BST = UTC+1, so use 1.

TZONE_A

TZONE_B

TZONE_C

7.10 Working diaser account name

USER_ACC

Choose a name for your DIASER user accounts. The same name will be used on all three nodes. Limit this to between 5-10 lower case characters for simplicity. I use diasertest for example.

7.11 Time out

TOUT

The copy timeout used by rsync for transfers. Set lower than phase periods.

7.12 Home directories

DIR_A

DIR_B

DIR_C

Home directory of diaser account, you may need to adjust if a large partition is not in the usual home directory place i.e. /mnt/big/ will evaluate as /mnt/big/diaser.

7.13 FILL_START_TIME

Time to initiate the daily filling script this should be set in advance of the DIASER archive transfer phases to ensure DIASER is filled before the phases begin.

7.14 VOLUME_DIR

Location of volume storage directory is where you store backup volumes created by your backup software.

7.15 DIFF_CONST_PREFIX

Differential or constant volume name prefix.

7.16 COLLECT_FULL

Choose whether full volumes are collected or not you want to simply collect constant sized volumes, like CCTV footage.

7.17 COLLECT_FULL_DAY

Day of moth when full volumes are collected.

7.18 FULL_PREFIX

Full volume name prefix

7.19 More than one configuration file

It is possible to force diaser to read a particular configuration file by executing]$diaser diaser.conf --opts

The configuration file can named as the user chooses i.e.

]$diaser my.config --opts

Currently, changes will always be written to diaser.conf from the directory diaser was executed in. The user is free to change the name of the configuration file and read it into diaser as described above. This feature is particularly useful when there us more than one installation being managed from a single user account.

8 Installation

]$./diaser --install

Use after you have configured diaser.conf as a normal user. As each task is completed you will be informed. At the end of installation you will need to one time only - you will need to login from the diaser account on each node to accept the certificates between nodes, like the 1st time you SSH into a box. A->B, A->C, B->A, B->C, C->A and C-B. Afterwards logins between nodes are password-less, this step will allow DIASER to begin work. This step may be automated depending on user feedback.

9 Command Line Options

Please note, not all of these operations have been implemented. Please view the most current development road-map: http://diaser.svn.sourceforge.net/viewvc/diaser/ROADMAP_DEV. As such some of these items may change or be removed altogether or others added. Later in the development cycle I plan to extend command line options so configuration changes can be set using the diaser command.

Run all commands from a prompt as a normal user, i.e.

]$diaser --install

9.1 --help

Display menu and command line options.

```
DIASER Usage: diaser_setup.pl

    --help                  help|-?

    --bandwidth             calculate real bandwidth throughput between nodeX-Y
    --configure             question driven configuration tool
    --extend                extend maximum storage structure date
    --install               install
    --list                  list all volumes in storage
    --lock                  lock all DIASER node accounts
    --logs                  condensed log readings from nodes
    --migrate               migrate node
    --modify   [opts]       send modified configuration to nodes either
                            from conf file or command options or both
    --pause                 pause operation
    --recreate              recreate a single node from scratch
    --remove                remove from nodes, all data will be lost
    --resume                resume operation
    --retrieve [opts]       retrieve archive data
    --stats                 generate statistics
    --stop                  stop operation
    --upgrade               apply upgrades
    --version               show version

 For more information please use man diaser or the more detailed
 online manual: http://diaser.org.uk/manual.html

 Please send any FEEDBACK to dbrasher@interlinux.co.uk.
 I'm especially interested in how DIASER may be of use to you now or in the future.
```

```
Thank you.
```

9.2 --bandwidth

This option will allow you to view the real, not theoretical, data throughput between two of your chosen storage nodes. You will need to have the OpenSource tool, gnuplot, installed on the system from which you are running this application.

This tool will attempt to download and compile the binary NPtcp from the NetPIPE utility suite: http://bitspjoule.org/netpipe/. The tool operates over port 5002 and stats will be collected from the sender.

9.3 --configure

Question driven configure tool for new and existing diaser deployments with input validation.

9.4 --extend

Extend maximum storage structure beyond the currently installed year.

9.5 --install

Install DIASER. See the section 8 Installation above.

9.6 --list

This option lists all volumes stored in DIASER.

9.7 --lock

Lock all DIASER node accounts. The systems administrator will need to reset the passwords for each diaser user account manually.

9.8 --logs

Condensed log readings from nodes.

9.9 --migrate

Migrate node to a different server.

9.10 --modify

Apply modified settings to the running DIASER on your designated nodes. Any changed settings will also be written to diaser.conf.

9.11 --pause

Pause any currently running data transfers on all nodes. Sends kill -STOP.

9.12 --recreate

In case you need to rebuild a node. You should only need to rebuild a node in the event of a disk failure or other non-recoverable node loss. In all other cases please consider using the --migrate (node) option.

```
--numyear   years of operation required
--startyear year to begin storing archives, this can be in the past
--phase1    hour between 0 and 23
```

```
--phase2    hour between 0 and 23
--nodea     ip address in format 0.0.0.0
--nodeb     ip address in format 0.0.0.0
--nodec     ip address in format 0.0.0.0
--dryrun    boolean 1(y) or 0(n)
--lmb       lowest maximum bandwidth, KBytes per second
--tzone     [not yet implemented]
--tout      copy time out in seconds
--fillstarttime     time to run DIASER fill operation, hour between 0 and 23
--volumedir         the directory where your backup volumes reside
--diffconstprefix   prefix given to your Differential or constant volumes
--collectfull   are Full volumes to be collected or not, boolean 1(y) or 0(n)
--fullprefix    prefix given to your Full volumes
```

9.13 --remove

Completely remove DIASER from your previously designated nodes. Please use with caution as all archive data stored in DIASER will be permanently deleted.

9.14 --resume

Resume paused data transfers. Sends kill -CONT.

9.15 --retrieve

Fetch archived data volumes.

A simple tool provided which also has additional command line options. The retrieval tool will walk you through a set of questions then list files for you to pick and transfer. The file will retain it's name and be located in the diaser_rel directory.

```
--r_year    which year
--r_month   which month
--r_day     which day
--r_full    if not a day name a full directory - leave as default
--nodea     ip address in format 0.0.0.0
--nodeb     ip address in format 0.0.0.0
--nodec     ip address in format 0.0.0.0
--porta     int
--portb     int
--portc     int
--user_acc  user account name, usually default set previously
```

9.16 --stats

Displays for each node in GiB; disk space, total daily volumes, total full volumes and total data stored on each node and average differential volume size.

9.17 --stop

Discontinue data transfers. Sends kill -9.

9.18 --ugrade

Apply product upgrades to an existing nodes with a DIASER installation. Your DIASER account password will be requested.

9.19 --version

Show current DIASER and currently installed Perl version.

10 Operation

10.1 Stop

This option will stop DIASER copies currently in operation, until the next set of transfer operations are initiated. This will kill any rsync processes.

10.2 Pause

This option will pause DIASER copies currently in operation, until the resume options is used.

10.3 Resume

This option will resume DIASER copies currently in operation.

10.4 Hard Lock

Lock all DIASER node accounts. This is a security feature. Enables the operator with root access to lock all DIASER node accounts immediately. Only by logging in to the nodes as root and re-enabling the DIASER account password will access from node to node and hence operation resume.

10.5 Migrate node

Migrate will assist you in moving an existing node from the current machine, server or workstation, to a new one. This may be located anywhere as long as it satisfies the requirements for DIASER inter-node-visibility. The procedure may take anywhere from minutes to hours depending on the amount of data stored on the existing node and network bandwidth available.

11 The Code

11.1 Why Perl?

The language is very well suited to the Linux POSIX environments. It is well supported, has good network programming capabilities. Perls is very flexible and allows a simple yet robust coding environment. Cross platform properties are extremely valuable and ensures the code base is portable. Perls inherent text parsing abilities are also valuable and set the language apart from many other contenders.

11.2 Style

Style is based as much as possible on the excellent O'Reilley Perl Best Practises by Damian Conway. A modular approach is used to code DIASER. All subroutines take parameters derived from the configuration mechanisms. Only three global variables are used, the rest are passed directly to subroutines and returns read back.

11.3 Modules

Popular modules are used where possible. Only modules that are shipped with popular Linux distributions. The installer use a number of modules, the code deployed on nodes only use File::Find (shipped as default with most distributions) and the core Perl shipped as default by most Linux distributions.

11.4 Error handling

Under review.

11.5 Contribute

Please see http://www.diaser.org.uk/contribute.html. All contributions are received under MIT/X licence terms.

12 Online resources

12.1 Website

http://www.diaser.org.uk

12.2 SourceForge

http://sourceforge.net/projects/diaser

12.3 Mailing list

https://lists.sourceforge.net/lists/listinfo/diaser-devel

12.4 DIAP/LTASP and early project memory

http://www.diap.org.uk

APPENDIX

A Tables and calculations

```
Bandwidth and capacity lookup table
===================================
BW      Hours GB (Decimal)
Mbit/s  1    2   3    4    5     6
1       0.45 0.9 1.35 1.8  2.25  2.7
10      4.5  9   13.5 18   22.5  27
100     45   90  135  180  225   270
1000    450  900 1350 1800 2250  2700

Disk space lookup table
=======================
BW      Month   1xYr    2xYr
Mbit/s
1       20GiB   240GiB  480GiB
10      67GiB   804GiB  9.6TiB
100     542GiB  6.5TiB  78TiB
1000    5.2TiB  62.4TiB 748.8TiB
```

For more calculations information please use the --bandwidth tool.

Include more calculation examples.

B Glossary of terms

Under review

C Applicances

DIASER-appliance-3node-OVF-test-pak

Getting started:

Welcome to this 3 node pre-configured DIASER appliance, test pack.

Unzip and import the three appliances into your virtual machine hypervisor. The network is internal only. Images were created using the freely available, cross-platform, VirtualBox. You can also test DIASER whilst using Windows.

Things to try:

Test data is read from /mnt/backup on nodeA and generated by a cron job, then distributed. You can view logs and other activity by running #diaser diaser.conf --logs from nodeA (logged in as diaser-user with password diaser-user.) Use diasertest when the node password is requested. Also run $man diaser for more options. Explore the working accounts too.

Leave the system running for a few days and watch the test data inside DIASER using --list.

Pack contents:
---
3 x OVF images; based on Ubuntu 32bit 10.04.1 LTS
diaser-appliance-nodeA
diaser-appliance-nodeB
diaser-appliance-nodeC
diaser.conf - node construction is based on this config file
appliance_instructions.txt
manual.pdf
--list screenshot

General node specs:

256MiB Ram (PAE CPU mode)
Upto 2TB dynamically expanding disk
Internal network intnet
Hostname - diaser
DIASER working account/pass - diasertest/diasertest

Node specific:

A) IP 10.20.0.1
DIASER user account/pass, diaser-user/diaser-user

B) IP 10.20.0.2

C) IP 10.20.0.3

Security precautions:

This is a test pack. Please, if you do decide to put the appliance into a production environment you must change all user account passwords.

NB: The nodeA Perl build has not been performance tuned.

Index