Department of **Natural Resources and Mines**

A decade of dmmove

Lindsay Brebber

DMFUG - December 2012

Queensland Government

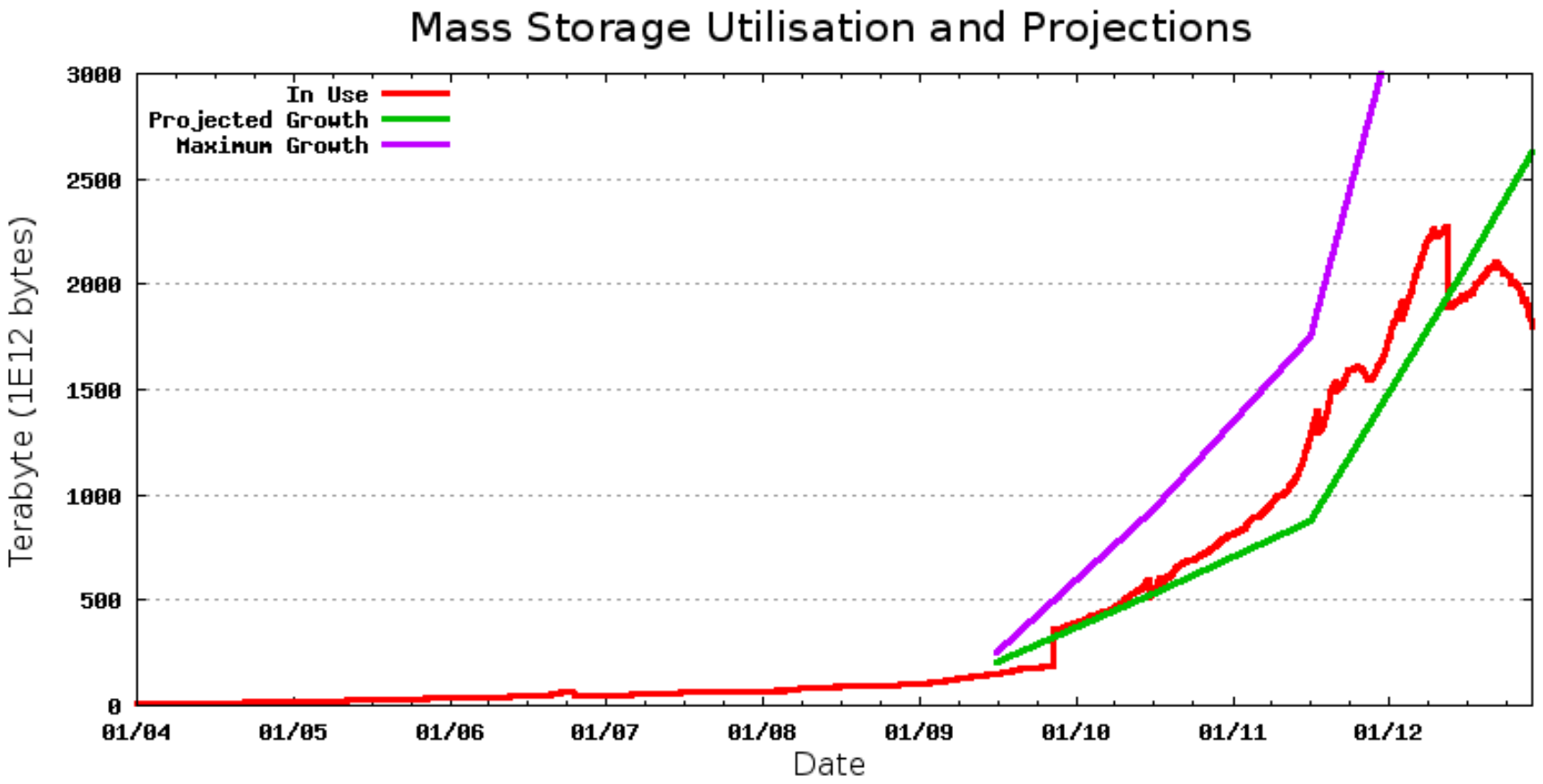
Overview

- DMF configuration
- Examples of how we use dmmove
- Lessons learnt

History

1996 - 1999 Cray J916 – Unicos	1999 - 2009 Cray SV1 - Unicos	2005 - 2009 SGI Origin 350 - IRIX	2009 - 2012 SGI Altix 450 - Linux
Timberline – 800MB	Redwood – 50GB	T9840A – 10GB	T9940B – 200GB
Redwood – 50GB	T9840A – 10GB	T9940B – 200GB	**T9840D – 75GB**
	T9840B – 20GB	T9840C – 40GB	**LTO4 – 800GB**
	T9940A – 60 GB	LTO3 – 400GB	**LTO5 – 1.5TB**
			T10000C – 5TB

Data Growth



History of dmmove

- Included with DMF since early versions
- Copies data from one media type to another via disk
- Enhancements to options as it evolved e.g. –d and -f
- man dmmove
- Also DMF Administrators Guide
- Robust
  - Copes with system crashes, kill
  - Still pays to check for single copies and too many copies
  - Won’t alter anything if bfid already moved
- Limited control over which volume group it reads from

DMF Configuration

- High availability configuration using SuSE HAE
- Two Altix 450 servers with SuSE SLES11SP1
- Each server has a IS220 disk array
- 4TB IS220 disk array for DMF filesytems
  i.e. /dmf/home, /dmf/spool, /dmf/journals
- 100TB IS4600 disk array for user filesystems
- Second IS220 used for DMF testing, dmmove

Two common uses of dmmove

- Media transition - move entire contents of one volume group to another
- Space management - move subset of a volume group to another
  - Create free space on existing volume group with limited resources .e.g. T9840D
  - User selected files

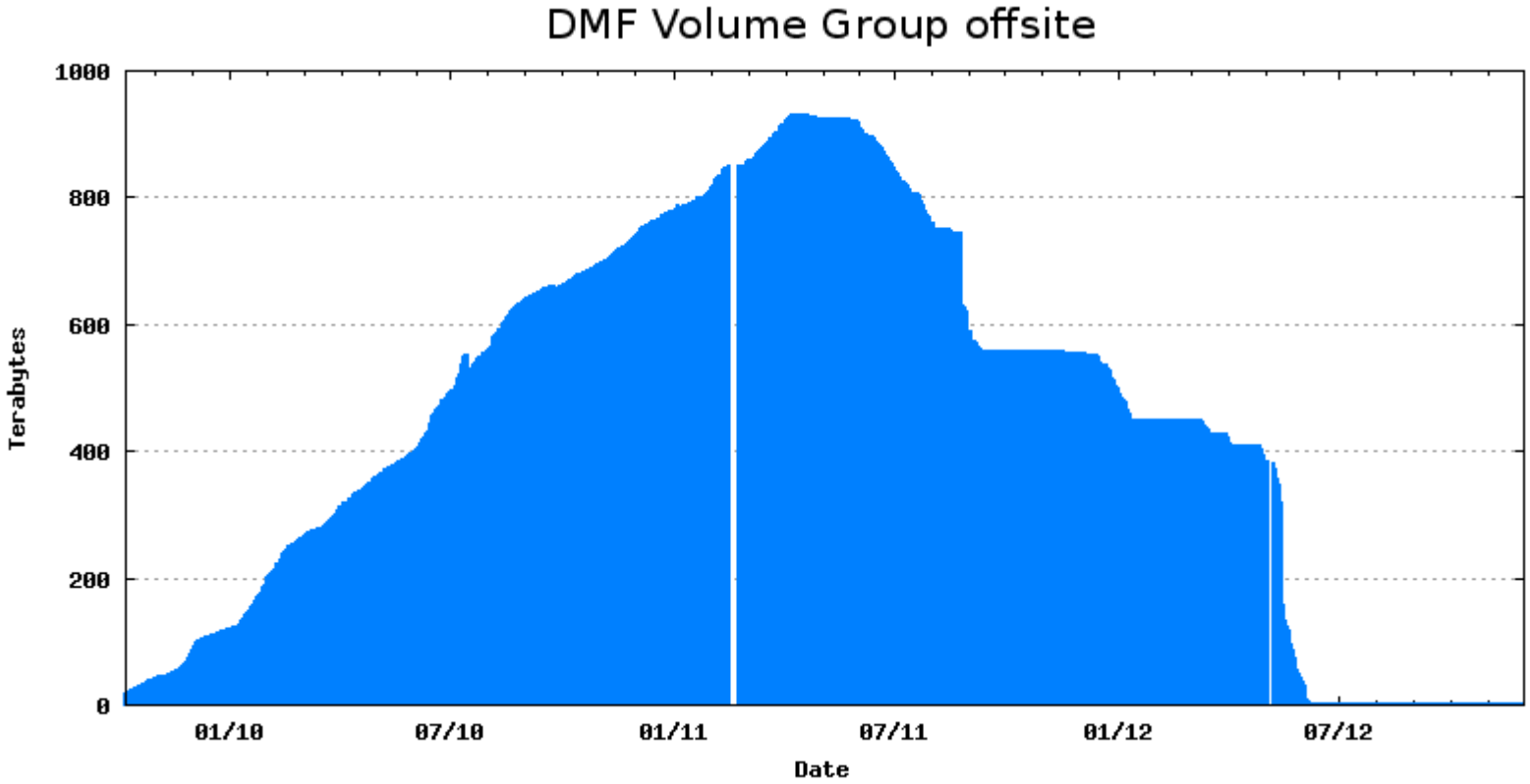
Time required to move a petabyte

Tape Type	Bandwidth (MB/s)	With no *compression* (TB/day)	With 1.5 : 1 compression (TB/day)	Time to move a petabyte (months)
T9840D – 75GB	30	2.6	3.9	8.5
LTO4 – 800GB	120	10.4	15.6	2.1
LTO5 - 1500GB	140	12.1	18.2	1.8
T10000C – 5000GB	250	21.6	32.4	1.0

Our experience - in real time to move offsite copy to LTO5

- About 1 year for 500TB
- Scripts ran for 8 months reading from different media types

Move offsite copy to LTO5



Optimising dmmove

- Managing resources
  - Number of tape drives – read
  - Number of tape drives – write
  - Disk space
- Minimising impact on users
  - Number of tape drives
- Minimising impact on system
  - dmmove –r (limit number of bfids)

Optimising dmmove - disk

- Disk configuration is critical to efficient dmmove
- Fast disk – bandwidth match or exceed that of tape drives
- Handle multiple read/writes – number tape drives
- Design move filesystem for maximum performance
- Size recommendations in DMF Admin Guide

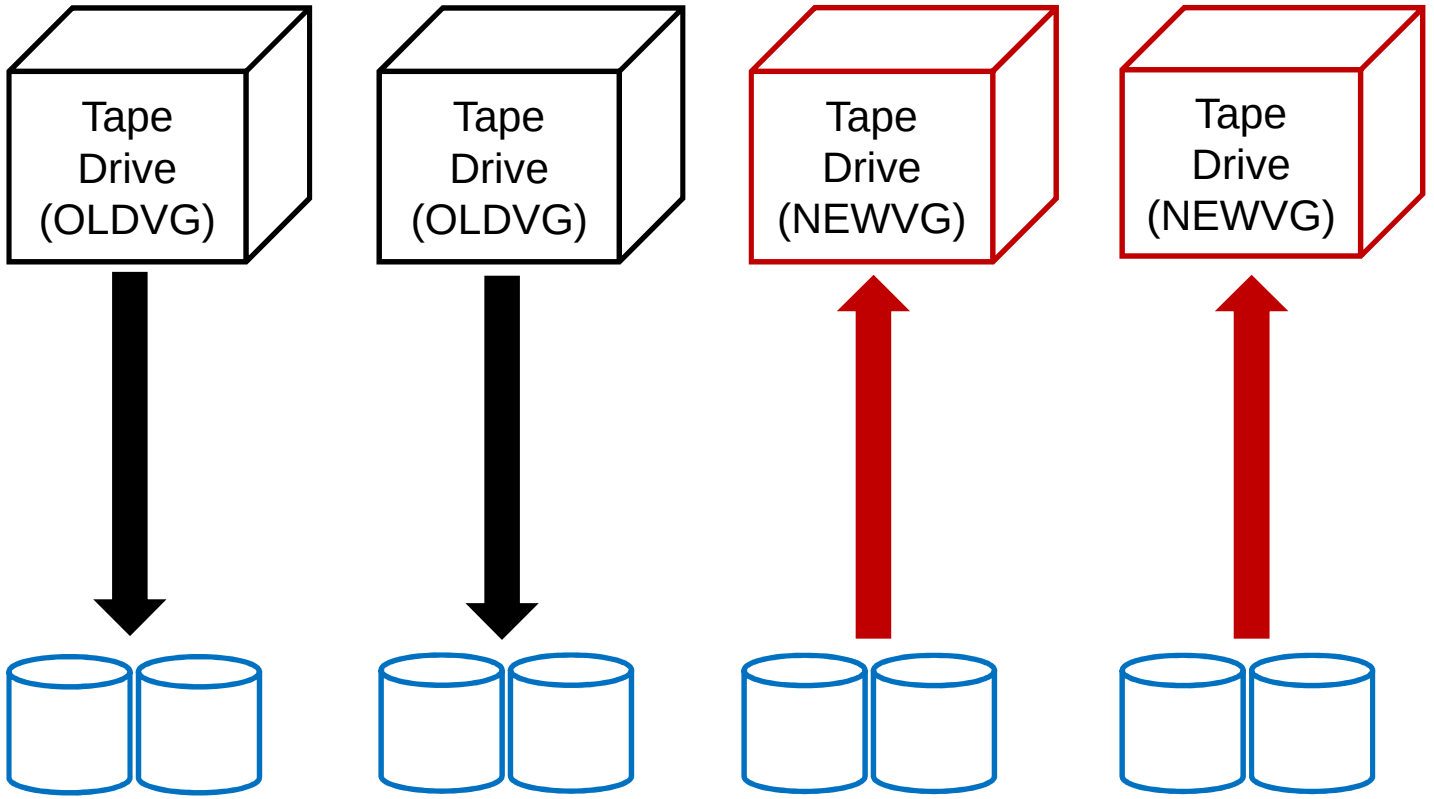
Optimising dmmove – tape read

- Most efficient method use a list of bfids for a tape/VSN with dmmove
  - Reduces number mounts/dismounts
  - Improves read performance, not jumping between tapes
  - Can be used to control the number of tapes drives used for reading

Optimising dmmove – tape write

- More difficult to control number of tape drives used
- Use DMF configuration settings for destination volume group
  - DRIVE_MAXIMUM and MAX_PUT_CHILDREN

dmmove in an ideal world



Aim: On average each read or write process streams to/from its own disk

Current disk configuration

- Disk
  - IS220 disk array
  - Eight x 300GB SAS disks
  - Four volumes using disks in a striped pair
  - Two x 4Gbit FC server connections, one to each controller
  - Two volumes per controller

Current disk configuration

- XVM – concat of four volumes

```
xvm:local> show -t prd_movefs1
vol/prd_movefs1                    0 online,open,accessible
    subvol/prd_movefs1/data  4679041024 online,open,accessible
        concat/concat0           4679041024 online,tempname,open,accessible
            slice/RXR0T0L20s0        1169760256 online,open,accessible
            slice/RXR0T0L22s0        1169760256 online,open,accessible
            slice/RXR1T0L21s0        1169760256 online,open,accessible
            slice/RXR0T0L18s0        1169760256 online,open,accessible
```

Current disk configuration

- mkfs options – standard for DMF filesystem
- xfs mount options
  - nobarrier,dmapi,logbufs=8,logbsize=256k,**agskip=3**
- pcp/pmchart to monitor disk I/O by volume

Ref: DMFUG Feb 2011 – Susheel Gokhale, SGI
http://hpsc.csiro.au/users/dmfug/Meeting_Feb2011/Presentations/DMF_Tuning_Examples.pdf

Media transition

1. Using dmvoladm list tapes for a volume group with hsite1=off sorted by dataleft in ascending order
2. From that list select the number of tapes to process, this controls the number of tape drives used for reading
3. Set hsite1 flag on each selected tape, prevents repeat attempts on tapes that failed a dmmove previously
4. Using dmcatadm generate a list of bfids for each tape

   dmcatadm –qc “list vsn=TEST01” | tail +4 | awk ‘{print $1}’ > TEST01.bfid
5. Run dmmove command using bfid list as input

   cat TEST01.bfid | dmmove –b –d OLDVG –r 50000 –f /move NEWVG

   OR

   cat TEST01.bfid TEST02.bfid > 2tapes.bfid

   cat 2tapes.bfid | dmmove –b –d OLDVG –r 50000 –f /move NEWVG

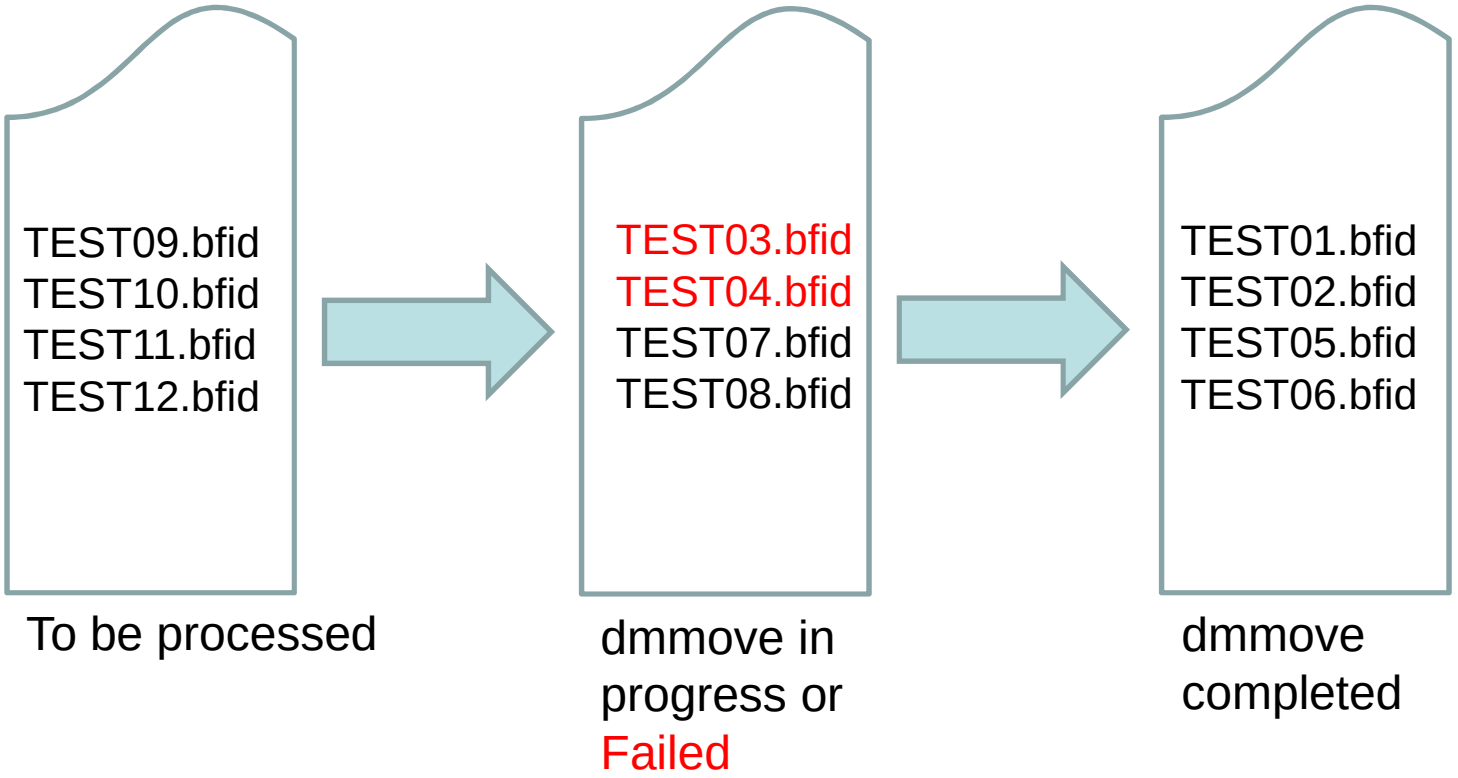
Making space in an existing volume group

1. Use dmattr to get bfid for each file in list supplied by user
2. Using dmcatadm to get VSN for each bfid
3. Cut list into text files based on VSN, one text file per VSN
4. Concatenate a selected number of these files
5. Set hsite1 flag on each selected VSN, prevents repeat attempts on tapes that failed a dmmove previously
6. Run dmmove command using bfid list as input

```
cat list.bfid | dmmove –b –d OLDVG –r 50000 –f /move NEWVG
```

Script processing steps using three directories

Each file contains the list of bfids for a tape/VSN



Automating the process

- Script
  - loops over a source of bfids
  - creates VSN related lists
  - uses hsite1 flag to prevent repeats
- Script creates unique log files for each invocation
- Has a simple mechanism to signal stop at next iteration so kill is not required, tests for existence of a certain file and exits if it exists
- Designed for simple restart, starts at next available bfid source

My general guidelines for use

- No matter how you select bfids, always aggregate by VSN before dmmove
- To “reduce the chance of costly mistakes”
  - Check you DMF database backups before starting
  - One copy of dmmove running at a time
  - One person managing any dmmove
  - Always use –d option, specify exactly what to delete
  - Script it, check it and check it again
  - Log everything the script does and keep the logs
  - Run dmaudit and verifymsp when done
  - Check for incorrect number of file copies (earlier DMFUG presentations)

Lessons

- Tape
  - Aggregate bfids by VSN for processing
- Disk
  - Use dedicated disks
  - Use a dedicated filesystem
    - Using /tmp is not ideal
- User impact
  - Set script to automatically run out of hours
    - e.g. 6pm Fri to 4am Mon or 6pm to 4am Tue/Wed/Thu
    - 4am allow time for a tape worth of dmmove to complete
  - Use holiday periods e.g. Christmas or Easter

Script improvements

- More complex sort of tapes to try and select tapes with similar data sets e.g. same size and number of chunks
- Aim: Want to keep all read tape drives busy for the same length of time

Questions or suggestions