Sized Based Backup Scheduling with TiBS for AFS (and other topics)

Speaker: Kristen J. Webb

Overview

- Review of time based scheduling
- How sized based scheduling works
- Example production implementation (MIT-CSAIL)
- “Trapped” backups
- Media retention policies
- Delayed Consolidation
- Additional controls

Overview (other topics)

- **TiBS Documentation Project**
- **Kerberos 5 Update**
- **AFS-OSD Backups**
- **Common File Management**
- **AFS Backup Engine R&D**

Traditional Time Based Scheduling

- Simple full and incremental backup schedules
  - Full backup once a week
  - Daily incremental backups in-between
    - Differential: changes since last full backup
    - True incremental: changes since last backup
- More complex schedules use multiple dump levels
  - Typically defined as 0-9
  - Level 0 is defined as a full (complete) backup
  - Higher level backups copy changes since the most recent backup of any lower level

Traditional Time Based Scheduling

- For Example,
  - Level 1 copies changes since most recent level 0
  - Level 2 copies changes since the last level 1 or level 0 (whichever is the most recent)
- Adding additional levels reduces processing time and storage costs
  - Reduces frequency of larger, lower level backups
  - Doing a full once a month is cheaper than once a week!
  - Additional processing and storage costs dwarfed by savings
  - More backup volumes may be required for restores

Traditional Time Based Scheduling

- **Example 1: Basic two level backup**
  - **Level 0: once a week (14.7%/day)**
  - **Level 1: every day (1%/day)**
  - **Average daily load (15.7%)**
- **Example 2: Addition of a third backup level**
  - **Level 0: once every 4 week (3.6%/day)**
  - **Level 1: once a week (1.4%/day, assumes 10% average size)**
  - **Level 2: every day (1%/day)**
  - **Average daily load (6%)**
- **Result: almost a 3X reduction in processing and storage costs**

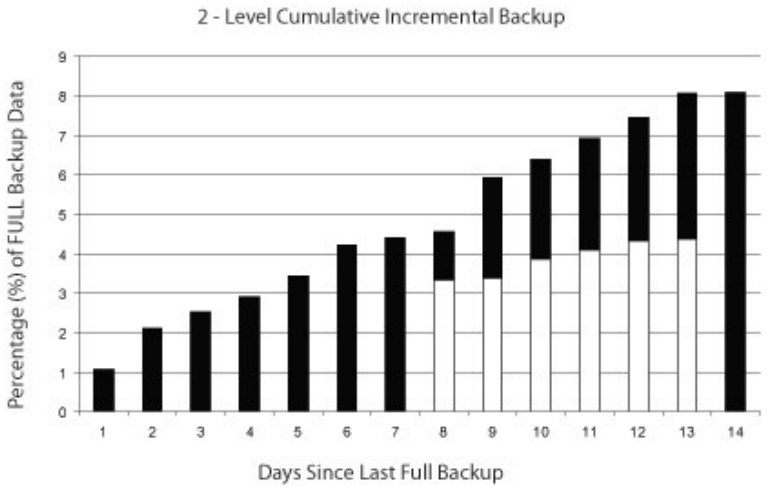
Traditional Time Based Scheduling

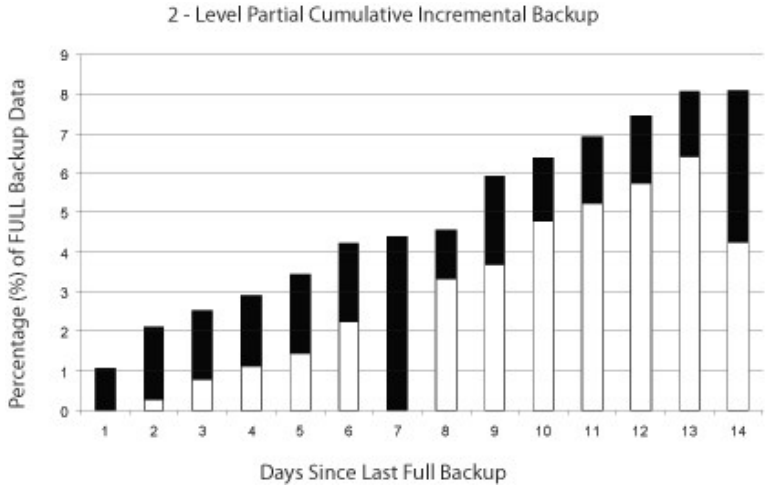
- Example schedule with 4 backup levels
  - Level 0: Full backup every 84 days
  - Level 1: Differential backup every 28 days
  - Level 2: Cumulative incremental backup every 7 days
  - Level 3: Cumulative incremental backup daily
- Processes a little over 1% new full backup each day on average
- About a 4% average daily workload (assuming 18% avg. level 1 size)
- Only a 33% reduction in processing and storage vs. a 3 level backup
- Requires up to 4 separate backup volumes to complete a restore
- Backups are scheduled regardless of the amount of data change

Time Based Scheduling with TiBS

- Example schedule with 4 backup levels
  - Level 0: Synthetic Full backup every 84 days
  - Level 1: Synthetic Differential backup every 28 days
  - Level 2: Synthetic Partial Cumulative incremental backup every 7 days
  - Level 3: True incremental backup daily
- Level 2 & 3 backups about 50% smaller than cumulative incrementals
- Average workload of 3% reduces processing and storage by 25%
- Synthetic processing removes 87% of network and client workload
- Level 0 & 1 backups now 60% of workload vs. 45% without partials

Comparison of Level 2 & 3 Workload





- **Cumulative Incremental Backup copies changes since most recent lower level backup**
- **TiBS Partial Cumulative Incremental Backup copies changes since most recent backup "at this level or lower"**
- **Increasing number of backups for restore mitigated by disk storage**

Full Disclosure

- Synthetic backup processing consumes more storage than traditional network backups
- For example, when a new full synthetic backup is generated a new True incremental backup is also taken on the same day
- Multiple level backups also create extra copies of incremental data
- This skews the data and percentages vs. network only backups
- This is a feature we call “Built-in Redundancy”
  - Synthetic backups are created using other backup volumes
  - Allows TiBS to rebuild any synthetic backup
  - Allows TiBS to restore data in the event of a failed backup volume
- Some percentages going forward do not include this overhead

For More Information

http://www.teradactyl.com/backup-knowledge/backup-definitions/backup-terminology.html

Different Types of Data

- Mature: little to no changes as a percentage of total data
  - Over a full backup cycle < 10% of data may have changed
  - A new full backup will copy > 90% of unchanged data
- Active: moderate data change between full cycles
  - Percentage change in the range 10-30%
  - Still copying 70-90% of unchanged data for a new full backup
- Growing: large percentage of data change relative to a previous full
  - Percentage change can easily exceed 100% of last full backup
  - Typical for new partitions as data is copied in by users
  - Large data changes may be reprocessed until next full backup

Example: Active Data

- **Assume 4 level backup with time schedules**
- **New 10 TB data partition added to backup**
- **Initial full backup taken with 1% partition utilization**
- **Over the next 4 weeks, 5TB of data is copied in**
- **First differential backup now 5TB**
- **Differential backup is 5000% of initial full backup!**
- **5TB+ will be recopied in new differential backups until the next full**

Computing Percentage Change

- TiBS processes a complete copy of meta data on every backup
  - Network retry phase catches files not picked up by modify time
  - Synthetic backups verify all files on the backup server
- Computing the percentage change is easy
  - Know how much data is in the backup just created (current_size)
  - Know the total data size of the partition (total_size)
  - current_size * 100 / total_size
  - Not based on size of previous full backup (always <= 100)
- If percentage exceeds a threshold (30%) schedule lower level backup

Level 0 Full and Level 1 Differential

- Turn of time based scheduling of Full Backups (every 84 days)
- Configure a size based threshold percentage for Full Backups
- Check percentage change of differential backups (every 28 days)
- If percentage change reaches threshold, then schedule new full
- Otherwise defer the full backup until the next differential
  - Verify all files referenced in the differential are backed up
    - The first time a full backup is deferred:
      - Scan the full backup and mark current files
      - Create an on disk file listing
    - Use file listing to mark files for future deferred full backups

Other Combinations of Backups

- Percentage calculations programmed into all incremental backups
- All percentages computed using size of the current full backup
- Cumulative Incremental Backups (Network or Synthetic)
  - Percentage includes all current data at this backup level
- Partial Cumulative Incremental Backups (includes True incremental)
  - Percentage only includes data for this backup (not all current data)
  - In development
    - Scan online file listing for all current backups at this level
    - Mark current files, compute effective cumulative percentage
    - Make size based scheduling decision

Multiple Level Size Based Backups

- Set a percentage threshold for all backups levels
- Any incremental change > percentage threshold is consolidated down to a new full backup
- Some data sets (especially smaller ones) run 4 levels each night
- More work needs to be done to make this practical
  - Addition of size limitations to scheduling
  - Addition of omit controls for other special cases
- Overall, the biggest gain is in deferring full backups for as long as possible and practical

Site Example: MIT-CSAIL

- 4 level backup, with new full backups generated every 84 days
- Backup policy:
  - Mirror full and differential backups, one copy sent offsite
  - Keep full and differential backup tapes forever
- As live data sizes increased, tape costs became unacceptable
- Developed size based scheduling for full/differential backups
- Deployed with an initial threshold of 33% for differential backups

Site Example: MIT-CSAIL

- Current statistics for AFS
  - Current full backup totals 38045 GB
  - Full backups generated in last 84 days total 3611 GB
  - Differential backups generated in last 84 days total 5130 GB
  - 23% of current full backup size generated in last 84 days
- Time based schedule for 84 days generates ~100%
- Not easy to estimate differential size for time based schedule
- From experience, size based differentials are smaller on average
- Estimated 5X reduction in full and differential processing and storage

Site Example: MIT-CSAIL

- For the entire site
  - Current full backup totals 200 TB
  - AFS represents about 20% of total live data
  - Full backups generated in last 84 days total 18 TB
  - Differential backups generated in last 84 days total 46.5 TB
  - 32% of current full backup size generated in last 84 days
  - Estimated 4X reduction versus time based scheduling
- Oldest current full backup volume is approximately 2.7 years old
- Average differential backup is 10% with 33% threshold

Site Example: MIT-CSAIL

- MIT changed tape technology from LTO-4 to LTO-5 in May 2011
  - Data is migrated from older tapes to make room in tape library
  - Volumes are selected based on two criteria
    - Amount of current full data left on older LTO-4 tapes
    - Accumulated amount of differential data since last full
- 25% of current full backups still reside on older LTO-4 tape
- Average less than 800 GB/day for new full and differential backups
- Average 1700GB/tape on LTO-5
- With mirroring, archive cost about 1 tape/day

Performance Summay

Schedule	Avg. Daily Workload	Network/Client Workload
1 level	100%	100%
2 level	16%	16%
3 level	6%	6%
4 level	4%	4%
4 level – TiBS	3%	.5%
4 level – TiBS (bysize fulls)	1.7%	.5%

Trapped Backups

- **Differential data change reaches a percentage, for example 25%**
- **Size based scheduling threshold set at 30%**
- **Data profile changes from Active to Mature (stops changing)**
- **Sized based schedule will not be able to consolidate to a new full**
- **New differential with 25% of data, generated forever!**

Trapped Backups

- **Solution: A secondary scheduler**
  - **Considers number of differential backups since last full**
  - **Allows average percentage to decrease as number of volumes increases**

Count	Total %	Average %
2	56	28
3	78	26
4	96	24
>8	150	N/A

- **Currently a prototype to be incorporated into the size based scheduler**

Site Example: MIT-CSAIL

- Secondary scheduler set up over 1 year ago
- Set to only process LTO-4 tapes as part of migration to LTO-5
- Now, LTO-5 needs to be included!
- Automation is being updated to catch up on trapped backups

- All processing done on a on a single Linux backup server
  - 32 GB RAM
  - 48 TB disk library
  - 2 LTO-5 tape drives
- Incremental backups scanning ~1 billion files nightly

Retention Polices

- Examples use permanent retention for full and differential backups
- Deferring backups for as long as possible gives the best result
- Most sites do not keep backups forever
  - Many sites only keep full backups for 1 year or less
  - Deferring backups forever is not an option
  - Size based scheduling must be integrated with time schedules
    - Extend time based requirements as long as possible
    - Size based scheduling captures large incremental changes
    - Helps to reduce differential storage costs
- Sized based scheduling effect diminishes as retention policy shortens

Retention Polices: Example

- Consider a 3 level backup strategy
  - Level 0: full backup every 4 weeks (retained 1 year)
  - Level 1: differential backup every week (retained 90 days)
  - Level 2: true incremental backup every day (retained 30 days)
- Technical note: better to describe policy as a desired restore point
  - If you want to restore data up to one year old, you actually need to keep the oldest full backup longer than that!
  - Restore policies are easier to define with TiBS
  - Remove a backup that is 365 days old only if a newer backup is at least 365 days old

Retention Polices: Example

- Updated 3 level backup strategy
  - Level 0: full backup every 4 weeks (restore up to 1 year)
  - Level 1: differential backup every week (restore up to 90 days)
  - Level 2: true incremental backup every day (restore up to 30 days)
- Now use size based scheduling
  - Level 0: full backup by size (restore up to 1 year)
  - Level 1: differential backup every week (restore up to 1 year)
  - Level 2: true incremental backup every day (restore up to 30 days)
  - Need to keep differential backups much longer!
  - Processing and storage costs will be lower, more granular restore

Retention Polices: Example

Schedule	Full Storage	Diff. Storage	Total Storage
Time Based	1300%	130%	1430%
Size Based	200-500%	530%	730-1030%

Very generalized results based on observations

Shows a potential 30-50% savings in storage costs

Backup processing of full backups reduced by 60-85%

Retention Polices: Example

- Still need to schedule full backups at some point
- Schedule new full backups annually
  - Mature data is not changing so weekly storage costs are low
  - Active and Growing data generates full backups more frequently
  - Keep full backups for a little more than 2 years
  - Once the newest full backup is 1 year old, older backups can be removed
  - 2-5 full copies stored on average instead of 12-13
  - Need redundancy? Mirror data to tape instead!
  - Don’t want to wait that long? Schedule fulls every 6 months

Delayed Consolidation

- **Problem: Minimize the number of full copies of data**
  - **Primarily driven by disk based solutions**
  - **Most polices require at least two different full copies**
    - **A current full copy, recently generated**
    - **An older copy to provide restores for times older than the current copy**
    - **Once current copy gets old enough, can delete older full**
    - **Now time to generate a new full copy, back to two copies**
- **Simple concept: create synthetic backups using combinations of older backups**

Delayed Consolidation

- Delayed consolidation allows for minimal backup storage
- Assume a restore policy of N (90) days
- Allow single full backup to age to N + Cycle (30) days
- Create a new synthetic backup using the full plus incremental data that is older than N (90) days
- New full generated is still older than N (90) days
- As soon as new full is generated, verified, etc, old full and older incremental data can be deleted
- Currently only works with disk based storage of incremental data
- Beta testing at CMU-H&SS, available in next TiBS release

Delayed Consolidation

- Review 3 level backup example with 1 year retention policy
- Without delayed consolidation, backups need to be retained 2+ years
- With delayed consolidations, older full backups brought forward
- For example on a 90 day cycle (to keep workload reasonable)
- Mature data may stay in this 90 day loop forever
- Can keep the older copy for redundancy or mirror data
- Current challenge is how to deal with keeping differential data on disk long enough
  - Use a pre-fetch to load older data from tape before performing consolidations
  - May require more than one tape drive to perform backups

Additional Controls

- Automatically generate synthetic full after initial network full backup
  - Network full backup represents a single copy of some data
  - Synthetic full makes second copy of all current data
  - Provides a redundant baseline for backups moving forward
  - Synthetic full can be repaired/regenerated from network full and first differential backup
  - May not need to do this if mirroring data to tape
- Forced migration when changing tape technologies
  - Skips size based scheduling for volumes on older tape technology
  - Not being used by MIT-CSAIL

Improved Tape Verification

- With reduced processing of full and differential backups, backup server has extra time to verify new full and differential backups
- Mirrored tapes sent offsite as soon as they are filled (or marked)
- Verification process creates online file listing
- Removes the need for first deferred full to read tape
- Rarely, if an onsite tape is found faulty, the offsite tape is recalled
- A repair process is performed and the offsite copy is sent back

Defer AFS backups using Last Update

```
# ./afs_profile.sh
YEAR/MO   VOLS  GB    TVOLS TGB
2004      4     50    4     50
2005      14    232   18    283
2006      30    363   48    647
2007      250   482   298   1129
2008      4344  1106  4642  2236
2009      309   1210  4951  3446
2010      566   3065  5517  6512
2011      1154  14318 6671  20830
2012/01   73    471   6744  21302
2012/02   102   1096  6846  22398
2012/03   91    1277  6937  23676
2012/04   62    554   6999  24230
2012/05   105   1434  7104  25665
2012/06   139   1411  7243  27076
2012/07   97    536   7340  27612
2012/08   118   797   7458  28410
2012/09   262   993   7720  29403
2012/10   96    251   7816  29655
```

2/3 of data unchanged since beginning of the year

Additional Topics

- **TiBS Documentation Project**
- **Kerberos 5 Update**
- **AFS-OSD Backups**
- **Common File Management**
- **AFS Backup Engine R&D**

TiBS Documentation Project

- Complete overhaul and update of existing online documentation
- User friendly and searchable
- More useful examples
- HTML and PDF formats
- Man pages! (available in next TiBS release)
- Designed to be
  - Multi-lingual
  - Mobile device compatible
- Ideas and feedback are always welcome!

Kerberos 5 Update

- Current implementation for UNIX
  - TLS using OpenSSL (dynamic link with OS libraries)
  - Mutual Authentication
    - Client/Server Certificates
    - Server certificate/GSSAPI
- Backup servers can communicate with TLS and Standard clients
- Running in production and CMU-SCS
- Requires Linux distribution specific builds (.rpm .deb, etc)
  - AppChecker from The Linux Foundation
  - Simplified build for current release, but not updated since 2011

AFS-OSD Backups

- **Preliminary prototype to scan sample vos dumps with OSD meta data**
- **Have not yet tested incremental backups**
- **Theoretically will work with TiBS synthetic backups**
- **Special considerations for restore may require additional updates**
- **Plan to finish backup engine updates and begin testing this year**

Common File Management

- TiBS stores files in a file stream on disk
  - Better processing for millions of small files
  - Larger files (> 1GB) stored on their own
- TiBS backup engine copies data from one stream to another
  - Remnant of pure tape based synthetic backup
  - Keeps files that are current, skips files no longer needed
- Experimental project stores each file on it's own on the backup server
  - Allows data copy to be replaced with hard linking
  - Not practical in production environments (billions of files)
  - May be useful in smaller, disk based environments

Common File Management

- TiBS backup engine copies data from one stream to another
- Some edge cases detected an FNAL did not work well
- Backup engine updated to detect large individual files and hard link
- A new file size threshold (for example 5MB) is introduced
- Backup engine detects files > 5MB and places them in a separate file stream file
- Still processes smaller files into 1 GB stream files
- Hard links can now be performed for files > 5MB when copying from one backup stream to another
- Makes concept practical for larger production environments
- Optimal threshold for larger files still being researched

Common File Management

- Current implementation running at CMU-ECE and CMU-H&SS
  - Preliminary results are promising
  - Takes time for hard links to take effect
  - Developed a compression calculator
    - Typical site measures 1.1:1
    - CMU-ECE currently at 1.3:1
  - Oct 16, 2012 CMU-ECE backups
    - Week/Month backup processing
    - 436 of 908 GB of new backup processes used hard links
- Future work to identify common files across volumes and block level

AFS Backup Engine R&D

- TiBS has always stored afs backups in vos dump format
  - Small backup header file used to integrate with UNIX/Windows
  - You can run vos restore –file from a disk library volume
- Generally, it works well
  - Size based scheduling: yes
  - Delayed consolidation: yes
  - Common file management: NO!
- Other scaling issues
  - Starting to see 10's of millions of files (memory intensive)
  - Lots of 2TB volumes (ND-CRC has 200TB afs cell)

AFS Backup Engine R&D

- **Good solutions for scale problems for UNIX/Windows**
- **Researching a transform from vos dump to our meta data and file stream format**
- **Track files using vnode/uniquifier, modify time and size**
- **Easier to include AFS in new features like common file management**
- **Possible to implement single file/subdirectory restore**
- **Possible to redirect AFS restores to other file systems (w/o ACLs)**

Thank You!

Enjoy the Conference!

kwebb@teradactyl.com