Tivoli Storage Product Directions

Mark Blunden
28 February 2008
Agenda

- Tivoli Storage

- TSM
  - 5.4
  - 5.5

- News

- Tivoli Storage Roadmap
  - TSM 6.1
  - TPC

- Questions?
Tivoli Storage

- **Tivoli Storage Manager (TSM)**
  - Not just a backup tool…
    - Data backup
    - Data archive
    - Data recovery
    - Space management
    - Disaster recovery planning
    - Media

- **TotalStorage Productivity Centre (TPC)**
  - Data Management
  - Device Management
  - Fabric Management

- **Continuous Data Protection (CDP)**

- **SAN Volume Controller (SVC)**
  - Market leading storage virtualisation

"Talking brings an audience - doing brings a profit"
TSM 5.4 – Software availability 1Q07
Collocation of Active Data – Active-Data Pool

- Active backup versions still reside on client machine (have not been modified or deleted)

- Active-data pools are sequential-access (typically sequential-access disk). Can also be on tape

  - Benefits of active-data pools on sequential-access disk
    - Optimized access to active versions for fast restore
    - Reduced size of disk pools if only active versions are stored
    - Reduced data movement in preparation for restore of active data

  - Benefits of active-data pools on tape
    - Reduced storage requirement while protecting against media failure or disaster
    - Simplified tape management because fewer tapes
TSM Transparent encryption

- TSM provides simple key management (TSM 5.3)
  - Available for TSM API applications; e.g. DP Exchange, DP SQL,…
  - TSM API Client generates a random encryption key password
    - Per data object selected by “include. encrypt” and “exclude. encrypt” criteria
  - TSM API Client passes encrypted data object encryption key password to TSM Server across “the wire”
    - TSM Client-Server “Session Key” used to encrypt/decrypt
    - AES128 encryption used if BOTH Client and Server support it

- HW Tape Drive Encryption (IBM TS1120) (TSM 5.4)
  - Device Class specification
  - One encryption key for each tape volume/cartridge
  - TSM Server generates, encrypts, and stores the encryption key in the TSM database along with other tape volume metadata
  - TSM supports HW provided encryption modes, Application, System and Library
  - TSM manages key in Application mode, other modes transparent to TSM

- Future – Backup-Archive client simple key management support (TSM 5.5)
Overwrite of Deleted Data – Data Shredding

- Allows disk storage pools to be designated as “shreddable”
  - Random-access (not sequential access disk (devtype=file), not tape or other media types)
  - Primary pools
- When a data object is moved or deleted from a shreddable pool, TSM server overwrites the object (number of overwrites can be set)
- Sensitive data objects are destroyed when deleted/moved, preventing undesirable data discovery
Snapshot Exploitation Topology (TSM for CS/ACS)

Support for multiple, persistent snapshots
- Persistent snapshots retained locally
- Very fast restore from the snapshot

Snapshot backup to TSM server
- Transfer outboard of application server to minimize impact to application
- Copies on TSM server provide long-term retention and disaster recovery

Policy-based management of local, persistent snapshots
- Retention policies may be different for local snapshots and copies on TSM server
- Automatic reuse of local snapshot storage as older snapshot versions expire

Restore can be performed from
- Local snapshot versions
- TSM storage hierarchy
NDMP Filer-to-Server Configuration

- Allows centralization of tape resource
- Exploits full capability of TSM storage hierarchy
- Data flow over the LAN and through TSM server
- IBM N series, Network Appliance, EMC Celerra, and other NAS devices certified for NDMP operations with TSM
Backup sets enhancements

Provides self describing tapes for off-site vaulting, data retention and tape rotation

- Generation of Backup Sets to Point in Time
- File Selection During Backup Set Restore
- Generation of Backup Sets for List of Nodes
- Image Data can be placed on Backup Sets
Generation of Backup Sets to Point in Time

- **Allows generation of backup set to specified point in time**
  - Files must not already be expired or deleted from the server
  - Default is date/time of backup set generation

- **Benefits**
  - Increases operational flexibility because backup sets need not be generated before next client backup
  - Allows retroactive generation of full backup if unanticipated needs arise
GENERATE BACKUPSET bob,fred,sue

- Allows generation of multiple backup sets with a single command
  - A distinct backup set is generated for each node
  - Backup sets generated with a single command are written sequentially to the same media

- Improves media utilization

- Physical security may be important for backup set restore at the client (without the TSM server)
Other Enhancements

- **Improved Memory utilization on incremental backup of very large file systems**
  - Option to use disk cache
  - Memory cache is default as today

- **Improved security for scheduling the Tivoli Storage Manager client** (Ability to block the following actions on the client machines)
  - Executing a scheduled Operating System command operation (action=cmd)
  - Executing a scheduled Operating System command operation via the pre/post-schedule or pre/post-snapshot command defined in the Tivoli Storage Manager server client option set
  - Executing a scheduled restore or retrieve operation (action=restore or action=retrieve)

- **Mac OS X Intel Client**

- **Microsoft Vista Client**

- **TSM HSM for Windows – MSCS cluster failover, globalization**

- **TSM Express upgrade to TSM enterprise**

- **SharePoint (1H2007) backup via partnership**
TSM 5.5 – Software available November 2007

Snapshot Exploitation Topology (TSM for CS/ACS)

Enhanced Snapshot support

**TSM 5.5**

- Snapshot support for MS SQL
- Additional Device and OS support for DB2 snapshots
  - Enhanced to support DB2 on NFS attached N-series / NetApp (Already supports SAN)
  - Enhanced to support DB2 on Linux (x and p Series) (Already supports AIX)

**Benefits:**

- Eliminates backup window for large databases
- Backup load is shifted to a second server
- Production Server full speed again after minutes
- No database or application shutdown required
- Provides instant ‘flash back’ restores

**Snapshot image backup of AIX JFS2 filesystem**
**Snapshot-based file level backup and archive of AIX JFS2 file system**
Ease of Use Enhancements

**VMWare Consolidated Backup**

- Consolidated Backup is a backup solution for ESX Server + SAN
- Uses a single agent running on the proxy server rather than an agent on every virtual machine.

**TSM 5.4**

- Basic integration using the VCB integration module
- Recovery view of the data is centered around the VCB proxy

**TSM 5.5**

- Does not require integration module from VMware
- Better recovery and management of the data on the TSM Server
  - User is removed from the fact that the backup was performed from a VCB backup host
  - Manage a virtual machine's backup data as if it had been backed-up by a TSM client running on the virtual machine

**Benefit**

- Integrated approach to protecting VMWare
- Reduce load on ESX Server
- Eliminates LAN traffic
Overview of Random- and Sequential-Access Disk

- **Benefits of disk storage**
  - Faster access (no delays for tape mount and positioning)
  - Eliminates management cost for tape handling

- **TSM supports two methods for storing and accessing data on magnetic disk**
  - Random-access storage pools (also known as DISK pools)
  - Sequential-access storage pools (also known as FILE pools)

- **Random- and sequential-access disk pools differ in how TSM manages disk storage and the operations that are supported**

- **TSM development views sequential-access disk as strategic, especially for long-term storage of data on disk**

**Random-access disk**

Objects stored in random blocks

**Sequential-access disk**

Objects stored sequentially in file volumes
Sequential-Access Disk: Concurrent Volume Access

- Sequential-access disk is based on tape paradigm
- Disk volume is locked by a single process or session using that volume
- Other operations block while volume is in use

Concurrent read access for the same sequential-access disk volume
Multiple operations will be able to retrieve data on the same volume without blocking

Benefit
Improved access to sequential-access disk

Planned 5.5
Sequential-Access Disk: Migration Thresholds

### Sequential-Access Disk Today

- Migration from sequential-access disk is based on tape paradigm
- Migration begins when percentage of volumes containing data reaches the high migration threshold
- **Example**
  - High migration threshold is 80%
  - 5 volumes in pool, each 30% occupied
  - Percent migratable is 100%
  - Migration begins even though pool is only 30% occupied

### Enhanced Sequential-Access Disk

- Migration thresholds for sequential-access disk similar to random-access disk
- Migration begins when percentage occupancy for the entire pool reaches the high migration threshold
- **Example**
  - High migration threshold is 80%
  - 5 volumes in pool, each 30% occupied
  - Percent migratable is 30%
  - Migration does not begin until the entire pool is 80% occupied

### Benefit

More data can be stored on sequential-access disk before migration

Planned 5.5
Export and Import Enhancements

Tivoli Storage Manager enables you to move backup and meta data from one server to another.

- Transfer clients between servers for load balancing
- Move from one server operating system platform to another

**Restartable Export/Import**

- Export/import operation is interrupted
- Export/import restarts from checkpoint (some files may be resent)

**Export Todate/Totime Options**

- Fromdate/Fromtime: Objects stored prior to this time are not exported
- Todate/Totime: Objects stored after this time are not exported
- Todate/Totime: Objects stored in this time window are exported

Allows an administrator to specify a time window for object data that will be exported

**Benefit**

- Reduced time to complete interrupted export/import
- Fault tolerance

**Benefit**

- Improved control over what gets exported
IBM TSM for Microsoft SharePoint

**Product Highlights:**
- Granular restore of SharePoint items
- Easy to install / Quick time to value
- Full, incremental, or differential backup
- Single Pane of Glass - Centralized administration
- Policy based management
  - Automatic migration to lower pools in hierarchy
- Automatic space reclamation
- Copying of data for disaster protection
- Migration of data to new storage technologies

**Key Benefits**

**Support for SharePoint 2003 3/07 & SharePoint 2007 5/07**
- Full backup and recovery of MS SharePoint Portal 2003 and MS Office SharePoint Server 2007 environments

**Robust Backup and Restore capability**
- Item level, site-level and sub-site level backup and restore

**Integrated with the TSM Server**
- All the benefits of TSM:
  - hierarchical storage management
  - support for hundreds of devices
  - automatic policy based management of stored objects
HSM for Windows Enhancements

<table>
<thead>
<tr>
<th>TSM HSM for Windows Reconciliation function and backup integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ Reconciliation Process</td>
</tr>
<tr>
<td>▪ Checks to prevent orphan stubs and files</td>
</tr>
<tr>
<td>▪ Backup Integration</td>
</tr>
<tr>
<td>▪ Backup both file contents and stubs</td>
</tr>
<tr>
<td>▪ GUI Enhancements- Configuration Editor, Progress Bar</td>
</tr>
<tr>
<td>▪ Configure all options with GUI</td>
</tr>
<tr>
<td>▪ Provide current state of the migration job and estimate on the duration</td>
</tr>
<tr>
<td>▪ Interoperability with Other File System Filter based Software</td>
</tr>
<tr>
<td>▪ Symantec/Norton Anti Virus, McAfee AV</td>
</tr>
<tr>
<td>▪ Journal based backup with TSM B/A, Open File support w/TSM B/A</td>
</tr>
</tbody>
</table>

**Benefits**

- Improve HSM storage pool utilization
- Reduce customer hardware/software costs
- Aligned with customer requirements/retention policies
TSM for Space Management

- Increase performance and use memory more efficiently in searches of candidate files for migration
- Process more files in a single namespace
Other Enhancements

- **Long File Name Support**
  - Approximately 4k-8k bytes in length

- **IPv6**
  - IPv6 is the "next generation" protocol to replace the current version IP, IP Version 4 ("IPv4").
  - Coexist with IPv4, Larger number of IP addresses
  - Unix and Windows HSM support 2H2008

- **Exchange 2007 support**
  - TSM for CS – Exchange Cluster support
  - TSM for CS - Individual Mailbox Restore (IMR) using VSS Snapshots
  - TSM for CS – TSM for Mail (Exchange) exploits 2007 VSS features
  - ‘TSM for ACS - Exchange 2007 64-bit support

- **Open File support on Windows 64-bit operating systems**

- **Online Image Backup Windows for 64-bit operating systems**
  - Use snapshots create point-in-time volume image copies to be backed up to TSM
AIX Enhancements

- **Backup, Archive and Space Management support for AIX Workload Partitions (AIX 6.1 WPAR)**
  - Backup and restore of local partition file data within the global partition using the local partition namespace available within the global partition.
  - Migration and transparent recall of local partition file data - Journaled File System (JFS2) and General Parallel File System (GPFS) - within the global partition.
  - Storage and retrieval of application data using Tivoli Storage Manager APIs supported from global partition.

- **Support for AIX Encrypted File System (EFS) backup**
  - In raw (encrypted) format, the data is not decrypted on backup, and the keys must be available to use the data after restore.
  - In clear text format, the data is decrypted (by EFS) as it is being read on backup and can optionally be encrypted by Tivoli Storage Manager using encryption options available within Tivoli Storage Manager.
  - Using the option to decrypt and back up with Tivoli Storage Manager encryption can make it easier to manage keys for long term data archival using Tivoli Storage Manager key managed encryption.
AIX Enhancements..cont

- **AIX filesystems enhancements**
  - AIX JFS2 Extended Attribute (EAV2) support
  - Used block-only image backup of AIX JFS2 filesystem
  - NFSV4 ACL support for AIX and AIX JFS2 filesystem

- **TSM for ACS**
  - Ships with DB2’s High Availability Features Pack 9.5
TSM for ERP 5.5 Enhancements

- Support for Red Hat Enterprise Linux 5.5
- Support for DB2 V9.5
- Support SAP with Oracle on HP-UX 11iv3 on IA64
TSM for Databases enhancements

- Data Protection for SQL operates in a Veritas Cluster Server environment
Enhanced Security

TSM Management of encryption keys for backup/archive client encryption

- Prior to TSM 5.5 backup / archive client required client to retain the encryption key to restore or retrieve the data.

- Enhancement:
  - TSM Client generates a random encryption key password
    - Per data object selected by “include. encrypt” and “exclude. encrypt” criteria
  - TSM Client passes encrypted data object encryption key password to TSM Server across “the wire”
    - TSM Client-Server “Session Key” used to encrypt/decrypt
    - AES128 encryption used if BOTH Client and Server support it

- Previously available TSM server-managed encryption
  - TSM 5.3 API applications; e.g. DP Exchange, DP SQL,...
  - TSM 5.4 HW Tape Drive Encryption (IBM TS1120,IBM LTO4)

Benefits

- Ease administrator and user burden - Key managed by the TSM server not the administrator or user.
- Authorized users have access to their data without having to remember the encryption password
News
Tivoli News

- **Acquisitions**
  - FileNet
    - Enterprise Content Management, Document management
  - Princeton Softech (Optim)
    - Data archival for Information Management environments (Oracle, SQL, Sybase, DB2, Informix, Adabas, DB400...)
  - Arsenal Digital Solutions
    - BCRS Storage backup/recovery service offerings
  - XIV Nextra
    - Storage capacity appliance

- **Announcements**
  - 5 + 3 product support
  - TSM for ACS without TSM Server as prerequisite (UDB 9.5 for SAP)
  - Beta programs
    - TSM for Replication
    - TSM V6.1
TSM 6.1 – Software availability 2H08
TSM 6.1 highlights

- This presentation describes future enhancements to IBM Tivoli Storage Manager family of products

- Information in this presentation does not constitute a commitment to deliver the described enhancements or to do so in a particular timeframe

- IBM reserves the right to change product plans, features, and delivery schedules according to business needs and requirements
## Tivoli Storage Manager (TSM) Roadmap

<table>
<thead>
<tr>
<th>Current Attributes</th>
<th>2H07</th>
<th>1H08</th>
<th>2H08</th>
<th>1H09</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TSM 5.4 Highlights</strong></td>
<td><strong>TSM 5.5</strong></td>
<td><strong>TSM 6.1</strong></td>
<td><strong>TSM 6.1</strong></td>
<td><strong>TSM 6.1</strong></td>
</tr>
<tr>
<td><strong>Core functions</strong></td>
<td><strong>Core functions</strong></td>
<td><strong>Core functions</strong></td>
<td><strong>Core functions</strong></td>
<td><strong>Core functions</strong></td>
</tr>
<tr>
<td>• Active data pools</td>
<td>• VMware Consolidated backup</td>
<td>• Improve server install and config</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Data shredding on random access disk</td>
<td>• Sequential disk pool concurrent retrieve access</td>
<td>• Data Deduplication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NDMP Off site vaulting</td>
<td>• Sequential disk pool migration thresholds</td>
<td>• Scalability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• NDMP filter to TSM server</td>
<td>• Long File name support</td>
<td>• Server scalability – upgrade TSM DB</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>• Restartable Export/Import</td>
<td><strong>Administration</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• B-A client scaling – memory usage</td>
<td>• IPv6 compliance</td>
<td>• Administration Center Updates</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Administration</strong></td>
<td><strong>Application Recovery</strong></td>
<td><strong>Reporting / Monitoring enhancements</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Administration Center Updates</td>
<td>• DB2 snapshot for IBM N series and NetApp NAS</td>
<td><strong>Application Recovery</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Application Recovery</strong></td>
<td>• DB2 FlashCopy for IBM DS8000 and SVC</td>
<td>• MS Exchange Mailbox Restore</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• mySAP / DB2-Oracle snapshot support for IBM N series and NetApp FAS</td>
<td>• MS SQL Server VSS snapshot</td>
<td>• MS Active Directory Item Restore</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• MS Exchange instant restore for IBM DS8000 / DS6000</td>
<td>• MS Exchange 2007 VSS backup</td>
<td>• MS Exchange VSS instant restore for IBM N series and NetApp FAS, offload backup validation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• MS SharePoint backup</td>
<td><strong>HSM</strong></td>
<td>• MS SQL Server VSS instant restore for IBM N series and NetApp FAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>HSM</strong></td>
<td>• HSM for Windows reconciliation</td>
<td>• File system snapshot for IBM N series of NetApp FAS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• HSM for Windows (MSCS cluster failover, globalization)</td>
<td>Security</td>
<td><strong>Interoperability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>• TSM managed key for B/A client encryption</td>
<td><strong>Win AIX64 GFS</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• TSM-managed tape encryption for IBM TS1120 / LTO-4</td>
<td><strong>Interoperability</strong></td>
<td><strong>Windows IA64, x864 online image backup / restore</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Interoperability</strong></td>
<td>• Win AIX64, x64 online image backup / restore</td>
<td><strong>Novell OES V2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• MAC OS X Intel</td>
<td>• Support for AIX Workload Partitions (AIX 6.1 WPAR)</td>
<td><strong>Support for AIX Encrypted File System (EFS) backup</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• MS Windows Vista basic support</td>
<td>• Support for AIX Encrypted File System (EFS) backup</td>
<td>• Snapshot/image backup of AIX JFS2 filesystem</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Snapshot/image backup of AIX JFS2 filesystem</td>
<td>• Snapshot-based file level backup and archive of AIX JFS2 file system</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# TSM Database Upgrade Benefits

## Database upgrade goals
- Increased Scalability
- More concurrent operations
- Store more objects, manages more data
- Larger maximum database size
- May reduce required number of TSM servers
- Increased Availability:
  - On-line, automated reorganization of database
  - Real-time database integrity validation
- Performance
- Potential for faster backups and restores

## Upgrading the database
- TSM server will be offline during the upgrade
- Will check for referential integrity during the upgrade process, and correct or log detected errors
- Upgrade requirements:
  - Upgrade from v5.5 only (not previous releases)
  - Source and target systems must be on the same platform, but not necessarily the same system
- Goal - Upgrade largest customer database over a weekend (will validate with actual customer database)

## Benefits
- Increased TSM Server Availability
- Increased Scalability
- Increased Performance
- No increase in Administrative Complexity

## Benefits
- Accurate upgrade
- Minimal administrator intervention
- Minimal TSM downtime
Data De-duplication

- Common files (operating system, application) are not duplicated in the TSM storage hierarchy for each node

- Client can transparently restore unique and common files

Benefits
- Reduced space requirements in storage pools
  - Reduced redundancy for:
    - Identical objects from same or different client nodes (even if names are different)
    - Common data extents (subfiles or “chunks”) in objects from same or different nodes
- Allows more objects to be stored on disk for fast access
- Especially attractive for disk-only TSM configurations
- Data migrated or copied to tape will be re-duplicated to avoid excessive mounting and positioning
Administration Center: Infrastructure Improvements

- Upgrade to new, lighter-weight version of Integrated Solutions Console (ISC) infrastructure
  - Significant reduction in footprint
    - Disk footprint ~1/2 the size of previous version
    - Initial memory footprint ~1/2 the size of previous version
  - Faster installation

- Improved Performance
  - Reduced time for initial page load and page switches
  - Enhanced processing of large queries
  - Progress indicators
  - Improvements in overall response time

Benefits:
- Improved administrative experience
- Faster installation
- Smaller footprint
- Better performance
Reporting overview

- Improved capability for out-of-the-box historical reporting and operational monitoring
- Graphical representation of trends and current status
- Complement and integrate with the Administration Center
- Sufficiently lightweight for managing one TSM server
- Scalable to allow management of servers (potentially hundreds) within an enterprise
- Allow customers to define their own reports
- Integration with reporting tools from other IBM products

Benefits: Improved reporting and analysis
TSM 6.1 monitoring

- **Scheduled Client Activity**
  - Scheduled client based activities

- **Scheduled Server Activity**
  - Scheduled expiration, migration activities

- **Client Current Activity**
  - Current client operations, backup/restore etc

- **Server Current Activity**
  - Current server operations, restoration, migration etc

- **Current Errors**
  - Recent errors and operation failure details

- **TSM Database Status**
  - Health and size of database and logs

- **Tape Device Status**
  - Status of volumes and usage of devices

- **Storage Pool Status**
  - Capacity used within storage pool

- **Client Backup Status**
  - Last backup of clients

Customizable, personalized views

Generic screen shots for example purposes only. TSM views will be different.
Data Deduplication and Tivoli Storage Manager
Disclaimer

- This presentation describes potential future enhancements to the IBM Tivoli Storage Manager family of products.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information in this presentation does not constitute a commitment to deliver the described enhancements or to do so in a particular timeframe.
- IBM reserves the right to change product plans, features, and delivery schedules according to business needs and requirements.
- This presentation uses the following designations regarding availability of potential product enhancements:
  - Planned 5.5: Planned for delivery in TSM v5.5 (2007)
  - Next Release Candidate: Candidate for delivery in the next release after v5.5
  - Future Candidate: Candidate for delivery in future release
Topics

- Deduplication technology
- Data reduction and deduplication in TSM
# Data Reduction Methods

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Compression</strong></td>
<td>▪ Encoding of data to reduce size</td>
</tr>
<tr>
<td></td>
<td>▪ Typically localized, such as to a single file, directory tree or storage volume</td>
</tr>
<tr>
<td><strong>Single instance store (SIS)</strong></td>
<td>▪ A form of compression, usually applied to a large collection of files in a shared data store</td>
</tr>
<tr>
<td></td>
<td>▪ Only one instance of a file is retained in the data store</td>
</tr>
<tr>
<td></td>
<td>▪ Duplicate instances of the file reference the stored instance</td>
</tr>
<tr>
<td></td>
<td>▪ Also known as redundant file elimination</td>
</tr>
<tr>
<td><strong>Data deduplication</strong></td>
<td>▪ A form of compression, usually applied to a large collection of files in a shared data store</td>
</tr>
<tr>
<td></td>
<td>▪ In contrast to SIS, deduplication often refers to elimination of redundant subfiles (also known as chunks, blocks, or extents)</td>
</tr>
<tr>
<td></td>
<td>▪ Only one instance is stored for each common chunk</td>
</tr>
<tr>
<td></td>
<td>▪ Duplicate instances of the chunk reference the stored instance</td>
</tr>
</tbody>
</table>

---

This terminology is not used consistently throughout the industry. In particular, the terms SIS and deduplication are sometimes used interchangeably.
Deduplication Concept

Data at source locations

Data source 1
A
B
C
D
E
F

Data source 2
C
G
E
H
I
J

Data source 3
J
H
L
B
K
C
E

Target data store (backup server)

D
I
B
J
F
H
G
E
C
A
L
K

Unique subfiles
Duplicate subfiles
How Deduplication Works

1. Data chunks are evaluated to determine a unique signature for each

2. Signature values are compared to identify all duplicates

3. Duplicate data chunks are replaced with pointers to a single stored chunk, saving storage space
Data Deduplication Value Proposition

**Potential advantages**

- Reduced storage capacity required for a given amount of data
- Ability to store significantly more data on given amount of disk
- Restore from disk rather than tape may improve ability to meet recovery time objective (RTO)
- Network bandwidth savings (some implementations)
- Lower storage-management cost resulting from reduced storage resource requirements

**Potential tradeoffs/limitations**

- Significant CPU and I/O resources required for deduplication processing
- Deduplication might not be compatible with encryption
- Increased sensitivity to media failure because many files could be affected by loss of common chunk
- Deduplication may not be suitable for data on tape because increased fragmentation of data could greatly increase access time
Deduplication Design Considerations

- Source-side vs. target-side
- In-band vs. out-of-band
- Method used for data chunking
- How redundant chunks are identified
- Avoiding false matches
- How redundant chunks are eliminated and tracked
## Where Deduplication is Performed

<table>
<thead>
<tr>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source-side (client-side)</strong></td>
<td>▪ Deduplication before transmission conserves network bandwidth</td>
<td>▪ Deduplication consumes CPU cycles on the file/application server</td>
</tr>
<tr>
<td>Deduplication performed at</td>
<td>▪ Awareness of data usage and format may allow more effective data reduction</td>
<td>▪ Requires software deployment at source (and possibly target) endpoints</td>
</tr>
<tr>
<td>the data source (e.g., by a</td>
<td>▪ Processing at the source may facilitate scale-out</td>
<td>▪ Depending on design, may be subject to security attack via spoofing</td>
</tr>
<tr>
<td>backup client), before transfer to target location</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Target-side (server-side)</strong></td>
<td>▪ No deployment of client software at endpoints</td>
<td>▪ Deduplication consumes CPU cycles on the target server or storage device</td>
</tr>
<tr>
<td>Deduplication performed at</td>
<td>▪ Possible use of direct comparison to confirm duplicates</td>
<td>▪ Data may be discarded after being transmitted to the target</td>
</tr>
<tr>
<td>the target (e.g., by backup software or storage appliance)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Source-side and target-side deduplication are not mutually exclusive.
## When Deduplication is Performed

<table>
<thead>
<tr>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In-band</strong></td>
<td><em>Immediate data reduction, minimizing disk storage requirement</em></td>
<td><em>May be bottleneck for data ingestion (e.g., longer backup times)</em></td>
</tr>
<tr>
<td>Deduplication</td>
<td><em>No post-processing</em></td>
<td><em>Only one deduplication process for each I/O stream</em></td>
</tr>
<tr>
<td>perform during data</td>
<td></td>
<td><em>No deduplication of legacy data on the target server</em></td>
</tr>
<tr>
<td>processing on the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>source or target</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Out-of-band</strong></td>
<td><em>No impact to data ingestion</em></td>
<td><em>Data must be processed twice (during ingestion and subsequent deduplication)</em></td>
</tr>
<tr>
<td>Deduplication</td>
<td><em>Potential for deduplication of legacy data</em></td>
<td><em>Storage needed to retain data until deduplication occurs</em></td>
</tr>
<tr>
<td>performed after</td>
<td><em>Possibility for parallel data deduplication processing</em></td>
<td></td>
</tr>
<tr>
<td>data ingestion at</td>
<td></td>
<td></td>
</tr>
<tr>
<td>the target</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In-band and out-of-band deduplication are not mutually exclusive
Generalized Deduplication Processing

1. Chunk the object
   - Divide object into logical segments called chunks

2. Identify duplicate chunks
   - Hash each chunk to produce unique identifier
   - Compare each chunk identifier with index to determine whether chunk is already stored

3. Eliminate redundant chunks
   - Update index to reference matching chunks
   - Deallocate space for redundant chunks

Object → Chunks → Chunk identifiers → Index

5
91
17
25

5
91
17
25

Chunk identifiers → Index
Data Chunking Methods

**Whole file chunking**
- Each file is treated as a single chunk
- No detection of duplicate data at subfile level

**Fixed-size chunking**
- Chunk boundaries occur at fixed intervals, irrespective of data content
- Method is unable to detect duplicate data if there is an offset difference
  - Because redundant data has shifted due to insertion/deletion
  - Because redundant data is embedded within another file or contained in a composite structure

**Variable-size chunking**
- Rolling hash algorithm is used to determine chunk boundaries to achieve an expected average chunk size
- Can detect redundant data, irrespective of offset differences
- Often referred to as fingerprinting (e.g., Rabin fingerprinting)

**Format-aware chunking**
- In setting chunk boundaries, algorithm considers data format/structure
- Examples: awareness of backup stream formatting; awareness of PowerPoint slide boundaries; awareness of file boundaries within a composite structure
Identification of Redundant Chunks

- Unique identifier is determined for each chunk
- Identifiers are typically calculated using a hash function that outputs a digest based on the data in each chunk
  - MD5 (message-digest algorithm)
  - SHA (secure hash algorithm)
- For each chunk, the identifier is compared against an index of identifiers to determine whether that chunk is already in the data store
- Selection of hash function involves tradeoffs between
  - Processing time to compute hash values
  - Index space required to store hash values
  - Risk of false matches
False Matches

- Possibility exists that two different data chunks could hash to the same identifier (such an event is called a collision)
- Should a collision occur, the chunks could be falsely matched and data loss could result
- Collision probability can be calculated from the possible number of unique identifiers and the number of chunks in the data store
  - Longer digest -> More unique identifiers -> Lower probability of collisions
  - More chunks -> Higher probability of collisions
- Approaches to avoiding data loss due to collisions
  - Use a hash function that produces a long digest to increase the possible number of unique identifiers
  - Combine values from multiple hash functions
  - Combine hash value with other information about the chunk
  - Perform byte-wise comparison of chunks in the data store to confirm matches
Hash Functions

Hash functions take a message of arbitrary length as input and output a fixed length digest of L bits. They are published algorithms, normally standardized as RFC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Output size L (bits)</th>
<th>Performance (cycles/byte) Intel Xeon: C / assembly*</th>
<th>Collision chance 50% (or greater) when these many chunks (or more) are generated **</th>
<th>Chance of one collision in a 40 PB archive*** (using 4KB / chunk)</th>
<th>Year of the standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD5</td>
<td>128</td>
<td>9.4 / 3.7</td>
<td>$2^{64} \approx 10^{20}$</td>
<td>$0.5 \times 10^{-20}$</td>
<td>1992</td>
</tr>
<tr>
<td>SHA-1</td>
<td>160</td>
<td>25 / 8.3</td>
<td>$2^{60} \approx 10^{24}$</td>
<td>$0.5 \times 10^{-28}$</td>
<td>1995</td>
</tr>
<tr>
<td>SHA-256</td>
<td>256</td>
<td>39 / 20.6</td>
<td>$2^{128} \approx 10^{40}$</td>
<td>$0.5 \times 10^{-60}$</td>
<td>2002</td>
</tr>
<tr>
<td>SHA-512</td>
<td>512</td>
<td>135 / 40.2</td>
<td>$2^{256} \approx 10^{80}$</td>
<td>$0.5 \times 10^{-140}$</td>
<td>2002</td>
</tr>
<tr>
<td>Whirlpool</td>
<td>512</td>
<td>112 / 36.5</td>
<td>$2^{512} \approx 10^{80}$</td>
<td>$0.5 \times 10^{-140}$</td>
<td>2003</td>
</tr>
</tbody>
</table>

** The probability of one collision out of k chunks is $p = 1 - e^{-k^2/2N}$, where $N = 2^L$; when $p = 0.5$, we get $k = N^{1/2} = 2^{L/2}$ (from birthday paradox).
*** The probability of one hard-drive bit-error is about $10^{-14}$.

Probability of collision is extremely low and can be reduced at the expense of performance by using hash function that produces longer digest.
Elimination of Redundant Chunks

- For each redundant chunk, the index is updated to reference the matching chunk
- Index is updated with metadata indicating how to reconstruct the object from chunks, some of which may be shared with other objects
- Any space occupied by the redundant chunks can be deallocated and reused
- Deduplication index is critical
  - Integrity
  - Performance
  - Scalability
  - Protection
Deduplication Ratios

- Used to indicate compression achieved by deduplication
- If deduplication reduces 500 TB of data to 100 TB, ratio is 5:1
- Deduplication vendors claim ratios in the range 20:1 to 500:1
- Ratios reflect design tradeoffs involving performance and compression
- Actual compression ratios will be highly dependent on other variables
  - Data from each source: redundancy, change rate, retention
  - Number of data sources and redundancy of data among those sources
  - Backup methodology: incremental forever, full+incremental, full+differential
  - Whether data encryption occurs prior to deduplication
- Beware of hype
Deduplication and Encryption

1. Three data sources have the same text file
2. After encryption, text files do not match
3. Deduplication processing does not detect redundancy
4. Text files are stored without data reduction
Topics

- Deduplication technology
- Data reduction and deduplication in TSM
Data Reduction with TSM Today

**Client compression**
- Files compressed by client before transmission
- Conserves network bandwidth and server storage

**Subfile backup**
- Only changed portions of files are transmitted
- Conserves network bandwidth and server storage

**Device compression**
- Compression performed by storage hardware
- Conserves server storage

**Incremental forever**
- After initial backup, file is not backed up again unless it changes
- Conserves network bandwidth and server storage

**Appliance deduplication**
- Deduplication performed by storage appliance (VTL or NAS)
- Conserves server storage
Native Data Deduplication in TSM

- TSM’s incremental forever methodology greatly reduces data redundancy as compared to traditional methodologies based on periodic full backups.
- Consequently, there is less potential for data reduction via deduplication in TSM as compared to other backup products.
- Nevertheless, deduplication is an important function to TSM because it will allow more data objects to be stored on a given amount of disk for fast access.
- Native deduplication is a key product enhancement in TSM.
TSM Deduplication Overview

Client 1: A

Client 2: B

Client 3: C

Files A, B and C have common data

Deduplication

Server

Deduplicated disk storage pool stores unique chunks to reduce disk utilization

TSM Database

Node | File
---|---
Client 1 | A
Client 2 | B
Client 3 | C

Tape copy pool stores A, B, and C individually to avoid performance degradation

Allows more objects to be stored on disk for fast access

Next Release Candidate
# Design Points for Initial TSM Deduplication Solution

<table>
<thead>
<tr>
<th>Design point</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Server-side                                       | ▪ Avoids need for deployment of client software  
▪ Effective for all types of stored data           |
| Out-of-band                                       | ▪ Allows deduplication of legacy data in addition to new data  
▪ Minimizes impact to backup windows  
▪ Concurrent processing to identify duplicate data |
| Index maintained in TSM server database (DB2)    | Transactional integrity, scalability, performance, disaster protection    |
| Variable-size chunking                            | ▪ Rabin fingerprinting with awareness of TSM data format                 |
| SHA-generated identifiers for detection of duplicate chunks | ▪ Probably SHA-1 or SHA-256  
▪ Longer identifiers of SHA-256 would reduce collision probability, at the expense of increased processing and database space usage |
| Average chunk size to be determined              | ▪ Larger chunks require less database overhead  
▪ Larger chunks reduce the total number of chunks required for given amount of data and therefore reduce collision probability  
▪ Smaller chunks improve compression                |
| Space occupied by redundant chunks will be recovered during reclamation | ▪ Allows coordinated recovery of space occupied by deleted objects and redundant chunks |
Expected Deduplication Behavior

- Disk storage requirement reduced via optional data deduplication for FILE storage pools
- Deduplication processing performed on TSM server and tracked in database
- Reduced redundancy for
  - Identical objects from same or different client nodes (even if names are different)
  - Common data chunks (subfiles, extents) in objects from same or different nodes
- Post-ingestion (out-of-band) detection of duplicate data on TSM server to minimize impact to backup windows
- Space occupied by duplicate data will be removed during reclamation processing
- Allowed for all data types: backup, archive, HSM, TDP, API applications
- Transparent client access to deduplicated objects
Expected Deduplication Behavior

- Deployment of new clients or API applications not required
- Legacy data stored in or moved to enabled FILE storage pools can be deduplicated
- Data migrated or copied to tape will be reduplicated to avoid excessive mounting and positioning during subsequent access
- Ability to control number, duration and scheduling of CPU-intensive background processes for identification of duplicate data
- Reporting of space savings in deduplicated storage pools
- Deduplication will not be effective for client-encrypted data, but should work with storage-device encryption
- Native TSM implementation, with no dependency on specific hardware
Deduplication Example

1. Client1 backs up files A, B, C and D. Files A and C have different names, but the same data.

Client1

Vol1

Vol2

Server

A B C D

A B C D

2. Client2 backs up files E, F and G. File E has data in common with files B and G.

Client2

Vol1

Vol2

Server

E F G

E F G

3. Server process “chunks” the data and identifies duplicate chunks C1, E2 and G1.

Server

Vol1

Vol2

A1 B1 B2 C1 D1

E1 E2 E3 F1 G1

4. Reclamation processing recovers space occupied by duplicate chunks.

Server

Vol3

A1 B1 B2 D1 E1 E3 F1
## Comparison of TSM Data Reduction Methods

<table>
<thead>
<tr>
<th>How data reduction is achieved</th>
<th>Client compression</th>
<th>Incremental forever</th>
<th>Subfile backup</th>
<th>Deduplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client compresses files</td>
<td>Client only sends changed files</td>
<td>Client only sends changed subfiles</td>
<td>Server eliminates redundant data chunks</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conserves storage pool space?</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Conserves network bandwidth?</th>
<th>Yes</th>
<th>Yes</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Data supported</th>
<th>Backup, archive, HSM, API</th>
<th>Backup</th>
<th>Backup (Windows only)</th>
<th>Backup, archive, HSM, API</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Scope of data reduction</th>
<th>Redundant data within same file on client node</th>
<th>Files that do not change between backups</th>
<th>Subfiles that do not change between backups</th>
<th>Redundant data from any files in storage pool</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avoids storing identical files renamed, copied, or relocated on client node?</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

| Removes redundant data for files from different client nodes? | No | No | No | Yes |
Considerations for Use of TSM Deduplication

- Consider deduplication if
  - Data recovery would improve by storing more data objects on limited amount of disk
  - Data will remain on disk for extended period of time
  - Much redundancy in data stored by TSM (e.g., for common operating-system or project files)
  - TSM server CPU and disk I/O resources are available for intensive processing to identify duplicate chunks

- Deduplication might not be indicated for
  - Mission-critical data, whose recovery could be delayed by accessing chunks that are not stored contiguously
  - TSM servers that do not have sufficient resources
Potential Follow-on Enhancements

- The initial TSM deduplication solution is designed to allow extensibility.
- Depending on business priorities, possible future extensions to this solution could include:
  - Option to perform inline deduplication during data ingestion (to achieve immediate compression)
  - Client-side deduplication (to distribute processing and conserve network bandwidth)
  - Option to control which hash function is used (tradeoff between performance and probability of false match)
  - Deduplication support for random-access disk or tape storage pools
  - Policies to control deduplication based on node, filespace, file size, or other criteria
Summary

- Data deduplication can reduce storage requirements, allowing more data to be retained on disk for fast access.
- Deduplication involves tradeoffs relating to degree of compression, performance, risk of data loss and compatibility with encryption.
- TSM’s incremental forever method avoids periodic full backups, reducing the potential for additional data reduction via deduplication.
- Server-side deduplication is a key enhancement in TSM.
Thank You

Questions?????