Your Deduplication is not your type after all?
Understanding Data Types and Their Impact on Deduplication

Jeff Tofano  |  Chief Technology Officer  |  Sepaton, Inc.
Data growth continues to skyrocket

• **+35% to 50%** compounded annual growth

• More data from new sources and new applications are being treated as critical
  – Structured data still prevalent
  – Unstructured data growing very fast
  – Semi-structured data also increasing fast

• Moving data is getting more and more costly
  – Basic cost of large volume moves and copies
  – Protocol and replication challenges

• Efficient deduplication is even more essential going forward but current models must evolve
Today’s deduplication

- Deduplication is critical for data protection but almost all current deduplication implementations create “islands of data”:
  - Only handle certain data sets and workloads well
  - Required matching endpoints for data reduced movement (i.e. Replication)
  - Are integrated into limited sets of applications
  - Deduplicated data can't be efficiently transported to heterogeneous architectures
  - Sizing and storage provision often very complicated
  - Current solutions provide much needed relief but are they really what customers want or need going forward?

Offering many benefits but is still very young technology. Most offerings fall into one of two classes:

**Inline (hash-based) Deduplication**
- reduction attempted BEFORE written to disk

**Post or concurrent Deduplication**
- reduction *performed AFTER* data written to disk
Many data types not handled well by some deduplication methods

Challenges

• Duplicate items come in all sizes and alignments
• Still have to deal with legacy multistreamed/multiplexed data
• Workloads and change rates vary
• Some data doesn't have a lot of deduplication potential
• Very large data sets often tax deduplication engines

Databases
• Big Data Analytics
  – Real-time Customer Insight
• Business Databases
  – Oracle, SQL, DB2
• ‘Big Science’
  – Astronomy, LHC, Genomics, Climate

Imaging and Digitized Media
• Cell phone cameras
• Medical imaging, ebooks, TV, Music, movies, news
• Mobile devices
  – Sensors, smart phones, surveillance cameras, RFID etc
Deduplication efficiency is all about trade-offs

- Granular Deduplication
- Backup Window
- Multiplexed/Multistreamed Performance
- Scalability
- Deduplication Efficiency
- Replication Efficiency
# Deduplication method details

## How to reduce

<table>
<thead>
<tr>
<th>Inline Deduplication</th>
<th>Post-processing Deduplication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong></td>
<td></td>
</tr>
<tr>
<td>• Removing dups in memory reduces IO load (fewer spindles and potentially better performance)</td>
<td>• Ingest rate is constant and tied directly to disk throughput</td>
</tr>
<tr>
<td>• All data dups against all other data</td>
<td>• Amount of data stored doesn't affect performance</td>
</tr>
<tr>
<td>• Large dup ranges can be handled very efficiently</td>
<td>• Small dup ranges can be handled efficiently...</td>
</tr>
<tr>
<td>• Replication model is a conceptually simple extension. Overlapped scheme can improve time to safety</td>
<td>• Easy model to scale</td>
</tr>
<tr>
<td><strong>Cons</strong></td>
<td></td>
</tr>
<tr>
<td>• Capacity often restricted because of index size issues</td>
<td>• Data typically only compared to other selected similar data – virtue and vice</td>
</tr>
<tr>
<td>• Performance varies by size of index</td>
<td>• System IO throughput needs to be larger than required for ingest (compares do IO)</td>
</tr>
<tr>
<td>• Non-sequential workload much harder to deal with</td>
<td>• Some capacity must be reserved as a staging area</td>
</tr>
<tr>
<td>• Very hard to scale</td>
<td>• Replication more difficult but isn't necessarily required for time to safety</td>
</tr>
<tr>
<td>• Data with low or no deduplication potential goes through same inline overhead</td>
<td>• Fragmentation can adversely affect read operations</td>
</tr>
<tr>
<td>• Fragmentation can adversely affect read operations</td>
<td></td>
</tr>
</tbody>
</table>
### Deduplication method details – *When* to reduce

<table>
<thead>
<tr>
<th>Source Deduplication</th>
<th>Target Deduplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Allows portion of deduplication process to be distributed to client systems</td>
<td>• Biggest benefit is transparent integration via standard protocols</td>
</tr>
<tr>
<td>• Typically embedded in backup apps but doesn’t have to be</td>
<td>• Doesn't require specialize client-side software</td>
</tr>
<tr>
<td>• Has ability to significantly reduce ingest time but can tax clients unexpectedly</td>
<td>• Allows for well sized/tuned solutions</td>
</tr>
<tr>
<td>• Doesn’t solve indexing issues in hash based systems and often makes them worse</td>
<td>• Offloads all deduplication processing so client systems are unaffected but does not reduce ingest transfer requirements</td>
</tr>
<tr>
<td>• Only benefits ingest</td>
<td>• Less effected by data types</td>
</tr>
<tr>
<td>• Heavily effected by data types</td>
<td>• Straight forward way to build scalable system</td>
</tr>
<tr>
<td>• Easier to implement for inline hash-based systems</td>
<td></td>
</tr>
</tbody>
</table>
Data types affect deduplication!

• Data types and access methods are often related and often imply a R/W footprint that can adversely affect some deduplication engines
• Different data types tend to have different levels of dup granularity. Databases and other structured data types regularly defeat many deduplication engines
• Some data types just don’t deduplicate well – optimal handling that balances reduction/performance cost is important (i.e. rich media)
• Some data types are just naturally large and require very large capacity (i.e. rich media, big data) with constant access time
• Data types can affect where you deduplicate – compressed or encrypted data is very hard for target mode deduplication technologies
• Data types that don’t deduplication well can actually make source deduplication solutions slower

**Bottom line:** **Today, the product you choose and the deduplication methods it employs must match the data types you’re protecting or you’re experience may be “sub-optimal”**

But what if the data types you’re protecting are changing as fast as the volume is growing?
Environment affects deduplication!

• Several major trends are affecting deduplication:
  – Virtualization
  – Cloud
  – Mobile
  – Big Data
  – Unified Storage

• Each of these is creating new challenges in a couple of areas:
  – Capacity and Performance Scaling
  – The need to address transfer costs as much as persistence costs
  – Many new and varied data types
  – Multiple new ingest/”out-gress” methods
  – Differing requirements for what data should and shouldn’t be protected
Deduplication must evolve

• All the above mentioned issues beg some interesting questions IT shops must ask...
  – Do you want ALL your protected data in the same unified deduplication repository?
  – How big will repositories get and how important is performance scaling?
  – Can any one deduplication methodology handle all the data types?
  – Are the access methods changing along with the data types?
  – Can you afford to create silos based on what deduplicates well and what doesn’t?
  – Can you afford to manage multiple silos just for capacity limitation reasons?
  – Do you want to have to “tune” your deduplication environments to leverage them?
  – Do you know what data types are being protected – now and in the future?
  – Do you want to extend the benefits of deduplication to reduce ingest and replication costs regardless of data types you have?

For deduplication to be effective in the coming years, it must provide much wider benefit, across all the emerging data types in a much more scalable transparent way
Hybrid deduplication – Can we merge the values?

In-line/Hash Benefits
- Resource efficient (disk) - Designed to find duplicates before data is written to disk
- Efficient for low volume, single-node systems
- Efficient for many data types

Post Process Benefits
- Moves massive data volumes to safety faster
- Not bound by hash table growth or limitations
- Excellent deduplication of structured databases and progressive incremental backups
- Scales across multiple nodes and disk trays
Hybrid deduplication requirements

- Must transparently handle multiple different data types
- Must transparently handle scale and load
- Must appear to the users as a single coherent environment
- It must also transparently embed in systems at least as well as existing conventional deduplication engines do
- At end of day, a hybrid system must present a simple, globally accessible, easy to use repository for all data protection needs
Hybrid deduplication needs

- Multiple deduplication methods (inline and post-process) that handle different data types
- Multiple options on where to deduplication (source vs. target) preferably based on data type and load
- Ability to bypass deduplication for data sets with no redundancies
- Common scalable repository format and retrieval methods
- Some "smarts" that select the right "where" and "how" and avoids unnecessary overhead
- Ability to define the rules for balancing performance and optimal reduction
- Replication and migration schemes that work off the common repository
### How hybrid might handle different data types

<table>
<thead>
<tr>
<th>Inline</th>
<th>Post Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Unstructured data:</td>
<td>• Multiplexed/multistreamed databases</td>
</tr>
<tr>
<td>– Office files</td>
<td>– Oracle, SQL, DB2</td>
</tr>
<tr>
<td>– Text Files</td>
<td>• Databases, sensor data, etc.</td>
</tr>
<tr>
<td>– Vmdk files</td>
<td>– Data stored in small segments</td>
</tr>
<tr>
<td>• Oracle Archive &amp; Redo Logs</td>
<td>• CAT scans, X-Rays, MRIs</td>
</tr>
<tr>
<td>• Medium-to-large daily backup volumes</td>
<td>• Large backup volumes with high change rates</td>
</tr>
<tr>
<td></td>
<td>• Progressive incremental backups</td>
</tr>
</tbody>
</table>
**Benefits of a Hybrid scheme**

- Ideal for enterprise environments because they handle all data types better than traditional deduplication schemes.

- Can be configured to support mid-range and low-end systems that mimic current offerings.

- Ideal for virtualized and cloud environments because they offer scale, control on where to deduplication and are more resistant to varied workloads.

- Can be mated to big data systems for DP or staging purposes.
Summary

• Data volumes are exploding and many new data types are showing up in DP environments

• Most IT environments must handle the growing requirements without being forced to separate or silo data

• Traditional deduplication environments are not designed to handle scale and varied data types well and will increasingly add to the complexity

• Deduplication technology must evolve to keep pace with the evolving needs and directly address prior shortcomings

• Hybrid deduplication offerings are an interesting (and necessary?) next step in deduplication evolution