StorHouse.com –

The Data Management Service Provider

Felipe Cariño, Jr. and John Burgess

FileTek, Inc.

360 N. Sepulveda Blvd. Suite 1080

El Segundo, California 90245

{FCARINO, JGB} @FileTek.com

Abstract

FileTek's StorHouse.com is a data management service that provides intelligent SQL database retrieval to an enterprise's historical data. The data is duplexed for availability and performance reasons and stored on a storage hierarchy (I.e. RAID, JBOD and Tape Farm Libraries). StorHouse.com provides solution and strategy unicity to the business and technical problems associated with storage and databases issues in DOT.com enterprises and applications. StorHouse.com is the enabling business solution while StorHouse/Relational Manager is the enabling technology to address very large database and large-scale storage problems for Web and generic applications. StorHouse and StorHouse.com are unique in their ability to store, manage, and access data volumes ranging from relatively small (100 GBs) to very large (over a petabyte).

Specifically, this paper describes and analyzes StorHouse.com proposed solution for

- Cost-effective use of active storage hierarchy
- Using SQL to retrieve and query up to petabytes (and more) of historical data
- An economic model to capture, preserve mine and convert historical data into strategic information.
- Analysis of the service provider business model with real world cost justification of StorHouse.com.
- Step-wise refinement, usage and incorporation of existing database systems into a central data store or federated database architecture.

In summary, StorHouse.com is a service that uses StorHouse/Relational Manager to cost-effectively exploit an active storage hierarchy. Oracle, SQL Server 7 and DB2 transparent federation can be use to develop and deploy web applications. We include an analysis StorHouse.com economic model since we believe this addresses the hidden economic, frequently unarticulated barrier that applications face when developing an enterprise data management solution where all historical data is stored, analyzed and converted into useful information.

1.0 Introduction

FileTek created and developed a StorHouse.com service to address the storage and intelligent retrieval of massive volumes of historical data and information. It is available for enterprise's that want to:

- Reduce IT infrastructure costs (total cost of ownership)
- Focus corporate resources on core competencies and strategy
- Improve the performance of critical operational databases
- Respond more quickly to markets (time to market)

StorHouse.com has both business and technical innovation that provides a cost-effective and economical solution to managing up to petabytes of storage. StorHouse.com uses StorHouse to provide database technology that manages a storage hierarchy. The storage hierarchy consists of RAID, JBOD, high performance tape and high capacity tape. In [CB 00] we analyzed how StorHouse/Relational Manager (RM) provides direct SQL query access to an active storage hierarchy and why this is a superior database and storage solution, as compared to conventional hierarchical storage management (HSM) systems, for terabytes to yottabytes databases.

Data is loaded to StorHouse.com from flat files on a user's server. The service manages data transfer to StorHouse.com servers, including *compression and encryption.* It manages multiple concurrent transfers to consume a fixed amount of Internet bandwidth, acquired in 100-megabit increments. Files are then loaded in parallel into segments (partitions) of StorHouse relational tables. Each user's application may have its own database. Each user's data is stored on its own physical set of volumes. Data storage is duplexed onto separate tape libraries so that downed libraries do not impact user access. Disaster recovery is also provided; duplicate data are also stored in a geographically separate server.

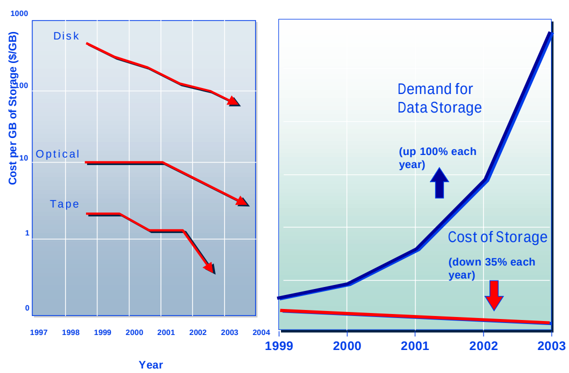
StorHouse.com can be used for both operational applications and fueling informational systems. Users access StorHouse.com data through bulk transfers (fueling), or common gateways such as ODBC, JDBC or ADO. Fueling delivers a flat file to the user's server, suitable for the batch load into a RDBMS or the processing of a data mining application. All Internet access data transfers are compressed and encrypted. Users define a fueling dataset through a web interface that generates SQL, unloads data, optionally summarizes and then transports the results to a user's server define fueling.

Load and Fueling can be integrated into user processes (e.g., daily loads). ODBC, JDBC and ADO can be connected to the user's RDBMS to provide an RDBMS hierarchy with StorHouse.com being the history layer. Non-scalar data is also supported through the use of an optional Relational File System interface. This provides for relational selects on scalar meta-data of non-scalar entities, which can then be accessed through NFS, or similar file system interfaces. StorHouse.com can also support complex database architectures where data from various data sources can be migrated to an active storage hierarchy on StorHouse/RM. These data sources include Oracle, SQL Server 7 and DB2. Applications on these databases can then transparently query the migrated data on StorHouse/RM.

This paper is organized as follows: Section 2 provides a technical analysis of StorHouse/Relational Manager (RM) database system approach to managing data on a storage hierarchy. Section 3 describes operational and federated database architectures for storing data in StorHouse/RM. Section 4 provides an analysis of StorHouse.com service that provides a unified solution for managing historical data. To validate our explicit and implicit assertions about the StorHouse.com approach we developed and describe an economic model based on total cost-of-owner (TCO) and other related components (e.g. storage). This addresses the cost and business obstacles for storing and mining all historic data. We conclude by describing future StorHouse.com and StorHouse/RM work in progress.

2.0 StorHouse/Relational Manager

Commercial Database Management Systems (DBMS) have evolved to support a diverse range of applications. DBMSs have been based on hierarchical, network, relational, object-oriented and the emerging object/relational database models. With few exceptions, these database systems and applications use disk media as their primary storage. Hierarchical Storage Management (HSM) is used by some of these applications to exploit the benefits of cost-effective optical and tape storage systems. However, as discussed later in this paper, these systems have limitations that make their use impractical for actively retrieved data by multiple applications. We propose and discuss the idea that database systems may use a complete active storage hierarchy (i.e. tape, optical, and disk). The key proposal is that active data be stored, queried and analyzed on tape farm libraries and optical jukeboxes, as well as magnetic disk. In this paper, we use the term active storage hierarchy when (say, SQL) queries execute against data stored on diverse media. This should not be confused with the research term “active disks”.



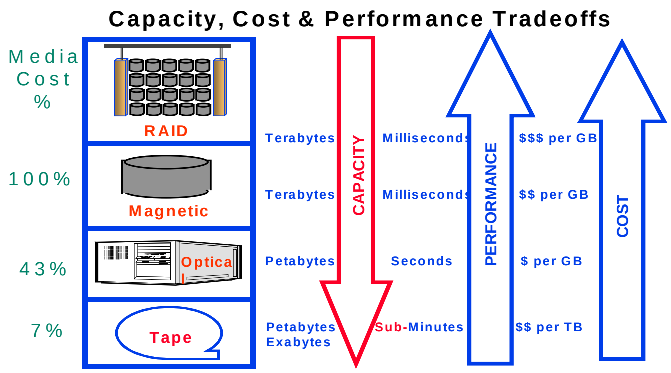
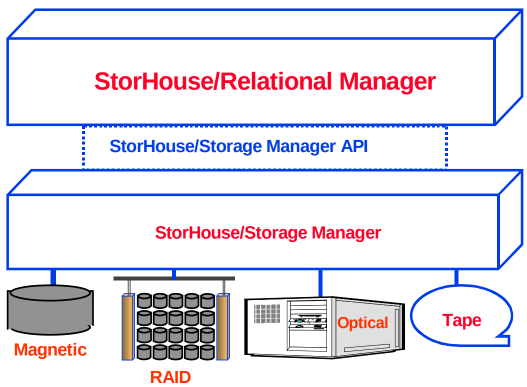


Figure 1: Storage Cost & Usage Trend **Figure 2: Cost, Capacity & Performance Tradeoffs**

StorHouse/Relational Manager (RM) [CB00] is a commercial relational database system that supports SQL queries for data stored in an active storage hierarchy. StorHouse/RM storage hierarchy includes RAID, Optical Jukeboxes and Tape Farm libraries. Again, we use the term *active* storage hierarchy because StorHouse/RM *directly* stores and data from a storage hierarchy (e.g. Tape Farm Library).

StorHouse/RM was designed and optimized to store atomic or historical data on diverse media. StorHouse/RM works in conjunction with StorHouse/SM, which specifically administers the storage, access and movement of relational data. SQL access is available from different platforms through a variety of industry-standard protocols. StorHouse/RM runs on Sun Microsystems Ultra Enterprise platforms. StorHouse/SM, FileTek’s comprehensive data management software, controls a hierarchy of storage devices comprised of cache, redundant array of independent disk (RAID), erasable and write-once-read-many (WORM) optical disk jukeboxes, and automated tape libraries. StorHouse/SM is also responsible for critical system management tasks, like data migration, backup, and recovery. StorHouse/SM provides system-managed storage that optimizes media usage, response time, and storage costs for each application. StorHouse/SM runs on Sun® Microsystems Ultra Enterprise Servers and comes standard with all StorHouse systems.

Figure 3 shows StorHouse/RM architecture, which is different from Hierarchical Storage Management (HSM) systems as shown in Figure 4. The major differences between the StorHouse/RM and HSM approach are summarized in the table below.



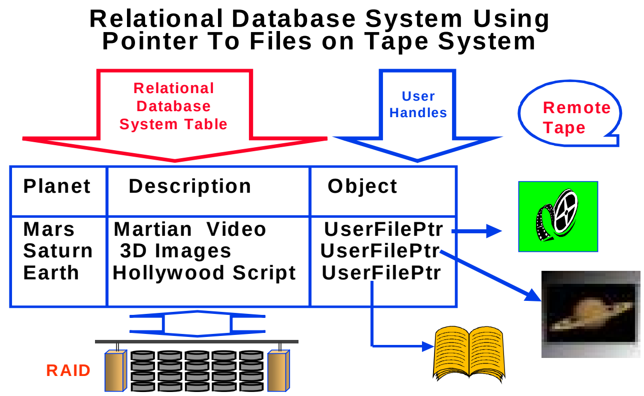


Figure 3: Relational Manager Architecture

Figure 4: HSM and/or File Pointer Approach

Key StorHouse/RM Versus HSM Solutions – Benefits Summary		
StorHouse/RM	**HSM Solution**	**Comment**
Direct data store and SQL query retrieval of data stored on say Tape Farm Library	HSM software places a marker in the database tuples, thus query must wait for data usually an entire partition to be found/restored.	HSM software is not as robust and in some cases requires database values restore and delete. The sometimes labor intensive process reduces historical archived data usage.
Database handles duplexing, recovery and internal pointers to values (not files).	Most HSM systems return a pointer to a file and not a specific tuple value; thus the receiving application may have to perform file post-processing to get desired tuple value.	Like many other database operations, we agree that the database management system should place, duplex and retrieve tuple values (automatically) via SQL (without DBA intervention).
Database provides ANSI SQL Large Object (LOB) and LOCATOR capability to LOB columns.	Application creates a pointer (i.e. OID) scheme for LOB values it stores in files.	Direct ANSI SQL LOB tuple/column values stored and retrieved via LOCATORs reduces application development complexity.
Heterogeneous Active Storage Supported	Burden falls on application to manage (and duplex) data from different storage types (and vendors).	Historical data may be stored and retrieved from several different types of storage (i.e. storage hierarchy).
Heterogeneous and Legacy database supported. In most cases without modifying the current application (i.e. transparent access to data from existing application)	Again burden is on application developer and DBAs to migrate, locate, place, restore and query data from heterogeneous database systems.	In depth analysis can be found in Section 3.0.

3.0 Database Models

Relational database systems, like DB2, Oracle and SS7, are used to store and manage data in Data Warehouse, Data Mining and/or operational enterprise systems. In section 2, we described the benefits of StorHouse/RM direct SQL access to data stored in the storage hierarchy. In section 4, we provide a concise analysis of the direct and indirect cost assertions made regarding StorHouse/RM and StorHouse.com. This section deals with the Data Warehouse, Data Mart and database architecture options for organizing and accessing large volumes of historic atomic (and LOB) data.

3.1 Data Warehousing

A major byproduct of cost-effective use of active storage hierarchy described is a new Atomic Data Store (ADS) Data Warehouse (DW) concept whereby *all* detailed (atomic) historical data is stored and used for information mining , decision support, customer relationship management or to support other operational requirements for data. Bill Inmon spawned an information revolution with what is now know as Data Warehousing [Imn92]. There are several potential DW and Data Mart architectures (see Figure 5) [Gar98] and philosophies [BZ98]. One such architecture is an enterprise-wide DW where detail data is stored and used for strategic reasons [Arm98]. Implicit in [Arm98] – an Enterprise-wide DW - is the idea that the value of detailed data is greatest when it is immediately available. Industry changing DW case studies are documented in [CK92] and [CJ98].

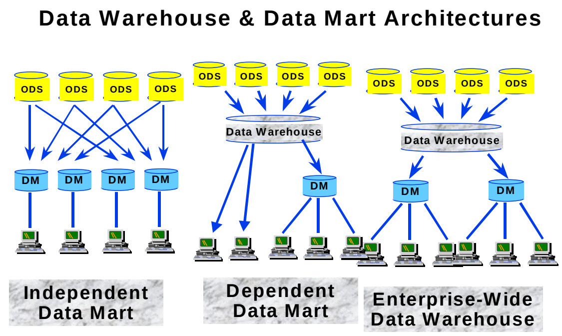
The economics of magnetic disk storage, however, force users to decide what data to maintain online and what to migrate out; migrated data is frequently replaced online with less useful summaries. This creates a "*Stealth Storage Cost-Barrier*" whereby application developers limit the amount of (disk-based) online data. It is clear that the majority of data does not need to be maintained on costly high-speed media. The next section on federation describes strategies for migrating rows (i.e. aging or predicate) or columns (i.e. large object) based on inactive or infrequently accessed data. A central ADS is used to store and manage all of the detail data and the ADS data can be used to fuel/load data into operational systems (e.g. using predictive triggers) or a DM/DW (e.g. using data summaries or frequent data access).

There are three major ways database systems use storage hierarchies:

(1) Hierarchical Storage Management (HSM) is used to migrate data to secondary storage (optical or tape). An HSM implementation strategy is to place a marker in the file system and move the data to secondary storage. When the query engine accesses the file system at the marker, the HSM will reload the data to disk, usually an entire file or partition. The query engine is generally unaware of the HSM data movement processing, so continues to view storage as disk.

(2) Data Backup and Archival copies or moves data to tape or optical storage. Data backup is used to make a complete or partial copy of the database in order to restore databases in case of disasters or data corruption problems. Data archival writes the data to optical or tape (tape is most likely) and then removes the data from the database. A customer may decide to keep "N periods" of data and archive (i.e. migrate) data at the "N+1 period". In order to use the archive information, users must restore the data and then later delete it. This is a cumbersome process that for many practical reasons is rarely done.

(3) Atomic Data Store (ADS) is where all the historical data is stored on some or all of the storage hierarchy media. A Hub-and-Spoke Architecture (Figure 6) can be used by operational data stores to store all their atomic/historical data into a central DW (with cost-effective active storage). Then, Data Marts can be feed cleanse and aggregated data (e.g. also solves island-of-automation data problem). A central data store can also be used by federated database/data warehouse architectures as shown in Figure 8.



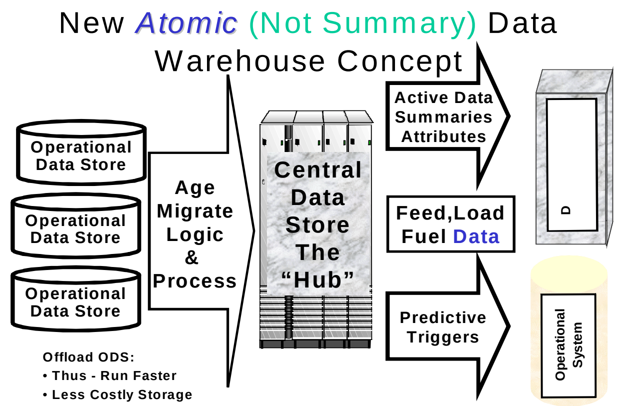


Figure 5: Data Warehouse & Data Mart Architectures **Figure 6: Hub-and-Spoke Architecture**

3.2 Federated Architectures

Federated Databases [Car86] can provide a "single system image" for diverse data sources; these products integrate the sources and provide uniform access to the data. In [CKK 00] we documented how database sizes will grow to yottabyte size and two database modeling techniques. We described and expand database and data warehouse architectures within the context of federated databases below.

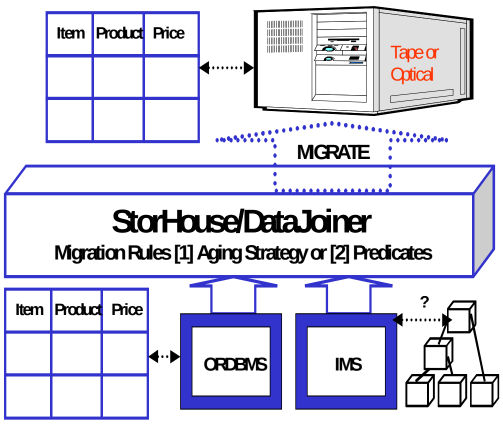
Enterprises have and use heterogeneous databases systems for a variety of reasons. StorHouse/RM can be transparently federated to Oracle, DB2 DataJoiner/UDB and SQL Server 7. Thus, some or all the data from these RDBMSs can be migrated and/or duplexed, in StorHouse/RM. A key point is that the current application can then transparently access data from the RDBMS and/or the StorHouse storage hierarchy. This architecture also provides location transparency.

The two main data modeling techniques that involve Oracle or DB2 are:

(1) Logical Horizontal Partitioning (LHP) – as shown in Figure 7 – is a horizontal partition of rows, whereby different rows from the same logical table are assigned to different data sources. The common strategies for migrating rows from the RDBMS to StorHouse would be to: (i) "age out" tuples based on a date, say a 6th quarter of data when you keep five quarters on-line in the disk-based RDBMS; (ii) "least frequently accessed" tuples – like Pine Cone System data and content Tracker approach to reducing disk-storage needs and making RDBMS run faster; and (iii) "complex predicate" where a predicate is used to migrate values to the active storage hierarchy. With LVP, different columns from the same (logical, federated) table may be assigned to different data sources (or storage media).

(2) Logical Vertical Partitioning (LVP) is a vertical partition of column across the RDBMS and StorHouse/RM. Figure 8 shows an LVP data model where text and video columns are stored on tape or optical and scalar values are stored on disk.

Like data modeling for relational databases there is a trade-off between performance, data replication and cost that each application must make. The LHP and LVP techniques above are new options available to existing and new web application developers. A future third technique will rely on Automatic Summary Table (AST) capability being added to relational database systems, like DB2 UDB.



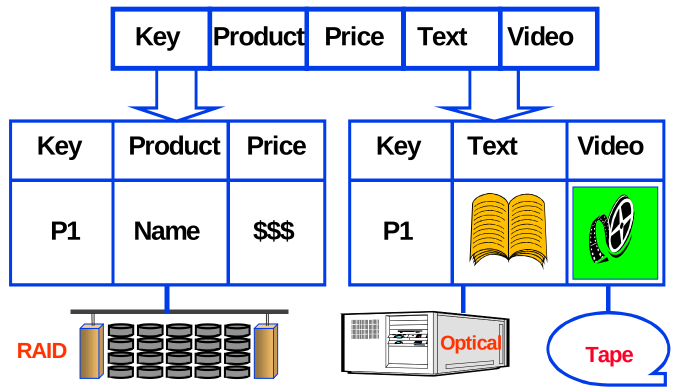


Figure 7: Logical Horizontal Partitioning **Figure 8: Logical Vertical Partitioning**

A critical part of the Data Warehouse is integrating all the information from diverse operational (i.e. Data Mart or Data Warehouse) systems. Figure 9 shows that a common central data store used by heterogeneous databases, in this case Oracle and DB2, provides a mechanism for Oracle applications to access DB2 historical data and vice-versa.

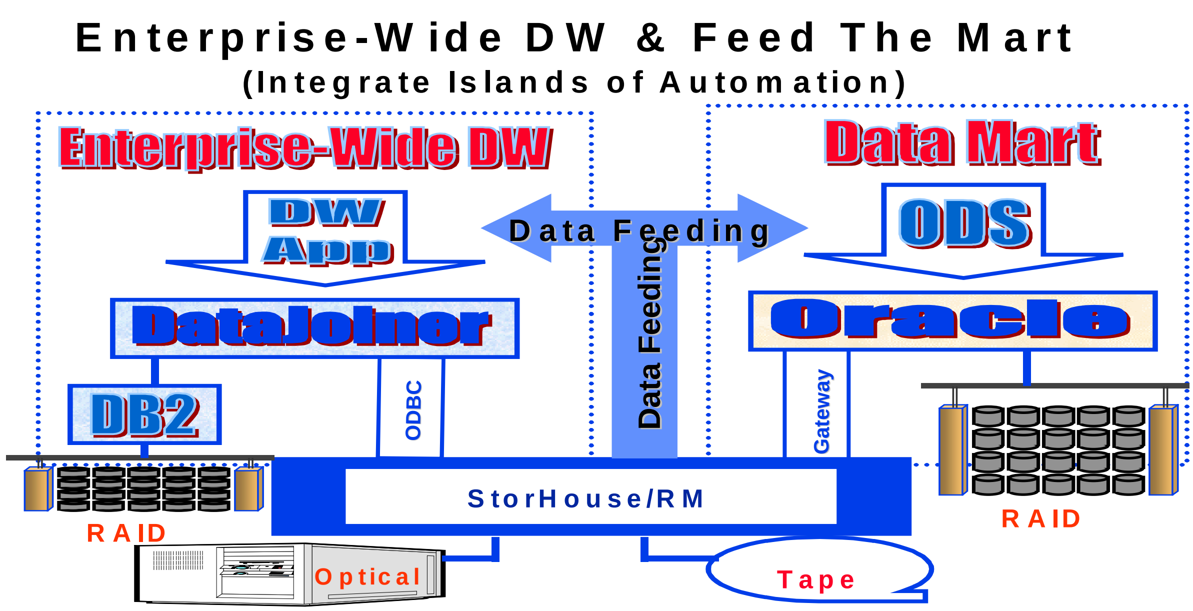


Figure 9: Enterprise-Wide StorHouse Central Data Store

4.0 StorHouse.com – Data Management Service Provider Model (for the Web)

Service provider model can be best explained by an analogy, we don't need to buy a 777 from Boeing, negotiate landing rights and hire and train airline staff to start flying. The same principle applies in the computer field whereby enterprises can avoid or delay large capital investments until application volume growth justifies an in-house system purchase and maintenance. The service provider model addresses these for diverse markets, like Storage Service Provider (SSP), Application Service Provide (ASP), Business Service Provide (BSP) and Host Service Provider (HSP). StorHouse.com – we believe is an industry first – *data management service provider (DMSP)* – provides a mechanism to outsource the management of persistent data such as telecommunications call event detail, web clickstream, IP telephony, customer encounters, digital certificates, and so on - in a way that keeps it all online and accessible.

The detailed historic data can then support many different applications, including telecommunications mediation, large-scale data mining, decision support, customer relationship management (CRM), multimedia file management, and business intelligence. StorHouse transparent access to Oracle, DB2 and SQL Server 7 can then be used as described in the section 3 as well as native SQL access to data stored in StorHouse/RM.StorHouse.com DMSP model was developed to reduce to initial cost (e.g. platform, storage, database, DBA, and so on) that experience shows is a major barrier to storing and querying web historical data. Figure 10 shows a four-quadrant model for a storage (and database) strategy.

- Active data must be kept current and available. Supportive data may be derived and refreshed on a periodic basis.
- High-Concurrency storage services high amounts of simultaneous accesses to small amounts of data (i.e. transactions per second).
- High-Volume storage services small amounts of simultaneous requests for large volumes of data (i.e. gigabytes per minute).

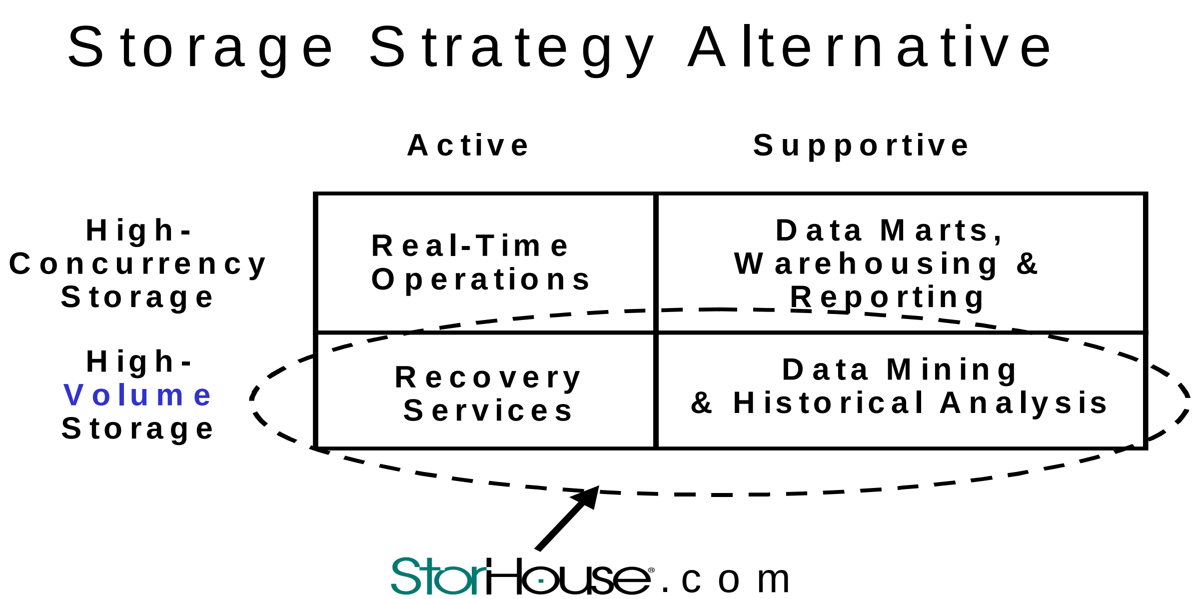


Figure 10: Four-Quadrant Storage Strategy Matrix

We validate our claims in the paper with a cost model. But, it is intuitively clear that the best "metanoia" (mental conversion) arguments for developing and mining data over the web will be ignored when a high initial cost is associated with the application, thus the data management service provider model was developed to provider greater StorHouse technology ubiquity. Once mining of historical data has proven its worth, then enterprises, including DOT.coms, can consider in-house StorHouse/RM solutions. Others may find the outsourcing benefits of the DMSP model are appropriate for their long-term use.

StorHouse.com DMSP – key benefits summary (for a complete analysis see www.StorHouse.com)

- Leverages Internet to provide an affordable and scalable data management solution.
- Initial and custom setup and configuration by experts (reduce expert staffing needs).
- Out-source development and testing of data loading (e.g. programs, tools and scripts).
- Data loaded to hosting center and can be securely uploaded.
- Pay-as-you-grow makes it easy to change service levels (at any time).
- Of course, no large upfront capital outlays (i.e. don't need to buy the 777 to fly from San Juan to New York)

Can deploy and discover new applications needed faster by concentrating on application development and not on database and storage infrastructure development.

4.1 Total Cost of Ownership - Data Management Costs

According to Gartner Group, the average Total Cost of Ownership (TCO) for disk is 3 to 7X the capital cost of the hardware and they also state that most organizations replace their DASD every 24 months. Remember, though, that storage management costs shown here do not fall at the rate of the disk hardware. Many of these costs actually increase over time, so the management overhead 'factor' will actually increase with time (many analysts would suggest it is already at 10X the capital cost of disk hardware).

Cost Factors		
Cost	**Open Systems**	**Mainframe**
Management	.5 - 2 %	2 - 3%
Environmental	1 - 25%	3 - 15%
Back-up	4 to 9%	32 - 54%
Capital	12 to 28%	19 - 22%
Operations	17 to 23%	16 - 26%
Other	17 to 61%	10 - 13%

Elements of the life cycle cost				
Management	Operations	Environmental	Back-up	Other
Formal Education	Disk & Remote B/U	Space / RE	HSM	Outages
Informal Training	Allocation factors	Electrical	Off-site tape store	- Scheduled
Admin. Labor	End user support	Power / cooling	Tape back-up	-Unscheduled
Other	Maintenance	Fire suppression/Security		Restore success rates

Note: Restore success rate for open system is poor – 25% to 75% vs. 99% for mainframes

Capital Cost of Data Storage

Industry analysts suggest that a 30 to 35% per megabyte server disk storage cost improvement will continue for the next few years. Remember, the capital cost of the raw disk storage is merely one component in the total cost of ownership of disk storage. Companies do not use all of the raw storage they purchase. Disk space is unutilized to improve performance. It is also used for temporary storage that certain processes require. Database indices can also consume large chunks of disk space. Therefore, the true capital cost of disk must account for unused space and system requirements.

Data Volume Growth

Industry analysts predict increasing data volume growth rates over the next 3 to 5 years. Driving this growth are E-Commerce, Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and Data Warehousing initiatives. While growth estimates vary, they surpass historical trends. The range of estimates falls between 75% and to 400% CAGR for DOT.com enterprises. As Figure 1 shows, the 35% disk improvement coupled with 100% to 400% data growth implies the need for cost-effective storage hierarchy.

Database Administrator Costs

For the purposes of this model, the total number of DBAs allowed for a single disk-centric database is eight. This is a conservative estimate since large IT shops may have dozens of DBAs and Systems Administrators caring for its databases. However, storage management costs are already built into the model (storage management costs factors). The annual burdened cost is the annual salary plus corporate overheads such as insurance, facilities, etc.

Total Cost of Ownership - Examples

To validate our cost claims we developed a TCO model that captures all the real costs associated with deploying a large web application. The tables below show the 5-year TCO for 100TB and 1PB web database application for (1) Disk-storage, (2) StorHouse/RM and (3) StorHouse.com.

Scenario 1: 500 GBs of base data: 100% CAGR					
	Current	Yr 1	Yr 2	Yr 3	
Base data (GBs)	500	1,000	2,000	4,000	
Total DB (GBs)	1,000	2,000	4,000	8,000	
Annual TCO		Yr 1	Yr 2	Yr 3	**Total**
Disk only		$1,040,000	$1,492,000	$2,160,800	**$4,692,800**
DMSP		$121,500	$225,650	$451,325	**$798,475**
Scenario 2: 10,000 GBs of base data; 50% CAGR					
	Current	Yr 1	Yr 2	Yr 3	
Base data (GBs)	10,000	15,000	22,500	33,750	
Total DB (GBs)	20,000	30,000	45,000	67,500	
Annual TCO		Yr 1	Yr 2	Yr 3	**Total**
Disk only		$14,900,000	$15,660,000	$19,482,000	**$50,042,000**
DMSP		$1,041,500	$1,562,250	$2,138,580	**$4,742,330**
Scenario 3: 200,000 GBs of base data; 25% CAGR					
	Current	Yr 1	Yr 2	Yr 3	
Base data (GBs)	200,000	300,000	450,000	675,000	
Total DB (GBs)	400,000	600,000	900,000	1,350,000	
Annual TCO		Yr 1	Yr 2	Yr 3	**Total**
Disk only		$144,960,000	$108,960,000	$102,210,000	**$356,130,000**
DMSP		$10,935,935	$16,403,900	$24,584,850	**$51,924,685**

5.0 Future Work

Future StorHouse.com DMSP capability enhancement includes: (1) Object/Relational database functionality as new SQL-99 User-Defined Types (UDTs) and User-Defined Functions (UDFs) are added to StorHouse/RM; and (2) Wider federated database integration within StorHouse.com.

Conclusion

We analyzed StorHouse.com – which we believe is the first commercially available Data Management Service Provider. The paper described the StorHouse/RM relational database approach to using SQL to directly store and retrieve data from an active storage hierarchy. We compared and document the benefits of direct SQL to a storage hierarchy versus an HSM solution. We also described how StorHouse/RM has the capability to federate as a data source to Oracle, SS7 and DB2, thus transparently adding new storage hierarchy capabilities to these disk-only database systems. We described two new database modeling techniques – vertical and horizontal partitioning of columns and rows - so that Oracle, SS7 and DB2 users can use a cost-effective storage hierarchy. We also described a new Data Warehouse concept and how a StorHouse/RM system can be used as an Atomic Data Store and to integrate islands-of-automation (i.e. Oracle, DB2, SS7). Finally, we described the benefits of StorHouse.com DMSP and validated our claims with a TCO (Total Cost-Ownership) model discussion and specific examples.

Acknowledgements

The authors want to thank Rob Simonds (developed the TCO model), Dave Clements and Pekka Kostamaa for comments and contributions.

References

[Arm98] Armstrong, B., "Data Warehousing: Dealing with Growing Pains", Data Engineering, March 1999, pp. 199 – 205.

[BZ98] Bontempo, C. and Zagelow, G., "The IBM Data Warehouse Architecture", CACM September 1998, Volume 41, No. 9, pp. 38 –48.

[Car86] Cariño, F. "HETERO – Heterogeneous DBMS Frontend", Proceedings Conference on Methods and Tools for Office Systems, Pisa, Italy, October 1986, pp. 159 – 172.

[CB00] Cariño, F. and Burgess, J., "StorHouse/Relational Manager (RM) –*Active* Storage Hierarchy Database System and Applications", Proceedings of 17th IEEE Mass Storage Systems Symposium, March 2000, College Park, Maryland, pp. 179 – 186.

[CJ98] Cariño, F. and Jahnke, M., "Bank Of America Case Study: The Information Currency Advantage", Proceedings of 24th VLDB Conference, New York City, pp. 641 – 644.

[CK92] Cariño, F. and Kostamaa, P., "Exegesis of DBC/1012 and P-90 – Industrial Supercomputer Database Machine", Proceedings of PARLE '92, Springer-Verlag, Paris, France, pp. 877 - 892.

[CKK00] Cariño, F., Kaufmann, A. and Kostamaa, P., "Are you ready for Yottabytes? StorHouse – Federated and Object/Relational Solution", 17th IEEE Mass Storage Systems Symposium, March 2000, College Park, Maryland, Vendor Presentation.

[Gar98] Gardner, S. R., "BUILDING the Data Warehouse", CACM September 1998, Vol. 41, No. 9, pp. 52 – 60.

[Imn92] Inmon, W., "Building the Data Warehouse", QED Technical Publishing Group, Wellesley, MA, 1992.