INFO SIZING

Performance Sizing Report

Sun Data Warehouse Reference Architecture for Structured and Unstructured Data

Petabyte Data Warehouse

using

Solaris™ 10 OS, Sybase® IQ

and

BMMsoft Server™

February 29, 2008

Cari Yuen
Director, Technical Marketing
Sun Microsystems
16 Network Circle
Menlo Park, CA 94025

Lisa Hopkins Dreyer
Director, Product Marketing
Sybase, Inc.
3665 Discovery Drive
Boulder, CO 80303

Paul Krneta
CTO
BMMsoft
225 Bush Street, 16 Floor
San Francisco, CA, 94104

February 29, 2008

In July 2007, at Sun Microsystem's request, **InfoSizing** verified the population and performance of the:

Sun Data Warehouse Reference Architecture
using Solaris 10 OS, Sybase IQ and BMMsoft Server

By defining and auditing the execution of a benchmark on a configuration assembled at Sun's Enterprise Technology Center (ETC) in Menlo Park, California.

The database was populated from a record-breaking **1 Petabyte of raw data.** In total, the database held **6 Trillion (or 6,000 Billion) rows** of transactional data and over **185 Million content-searchable documents**, such as emails, reports, spreadsheets and other multimedia objects.

The unprecedented volume of transactional data generated for this benchmark **represents a real-world workload scenario** comparable to the transactions processed across the worldwide financial trading networks over multiple years. This transactional data was then combined with a corresponding volume of multimedia documents representative of the electronic communication between half a million financial traders.

The configuration of the benchmark environment, the population of the data warehouse and the execution of the performance tests were **independently verified by InfoSizing** who also collected and analyzed the results of the measurements.

The attached report is an independent attestation of the capabilities and performance results detailed herein.

Respectfully Yours,

François Raab
President

INFO SIZING

Performance Sizing Report

Sun Data Warehouse Reference Architecture

using Solaris™ 10 OS, Sybase® IQ and BMMsoft Server™

Executive Summary

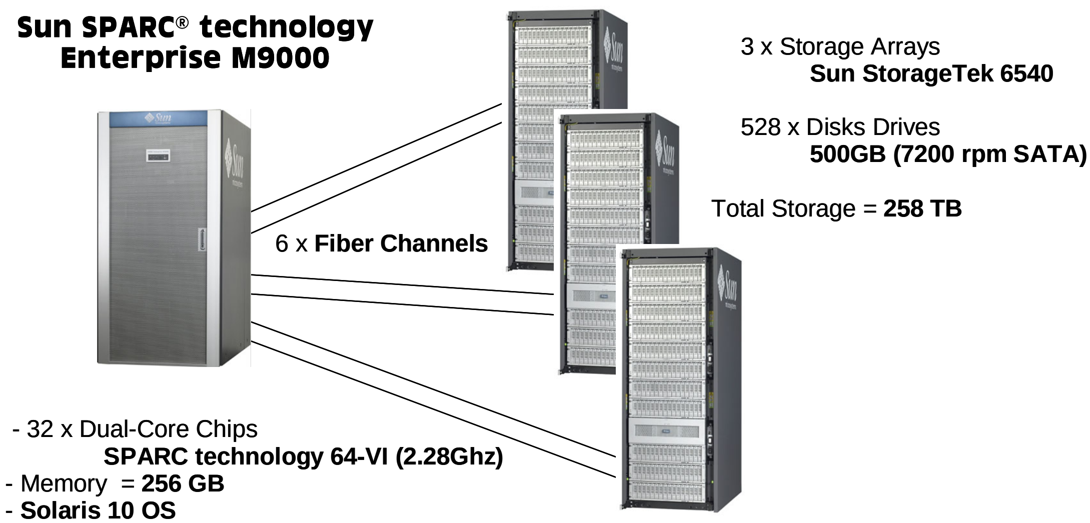
The **Sun Data Warehouse Reference Architecture**, powered by **Solaris™ 10 OS**, a **Sybase® IQ** analytic server and **BMMsoft Server™** on a **Sun SPARC®** technology **Enterprise M9000** server interconnected to three **Sun StorageTek™ 6540** arrays, demonstrated several significant achievements:

- It loaded **1 Petabyte of raw transactional data** (6 Trillion stock quote records) in a fully indexed star schema; creating a new, independently verified record for the **world's largest data warehouse**.
- It reached a **load speed of 285 billion rows per day** (3 Million rows per second) and sustained that database population pace for a period of over 3 weeks.
- It showed an **85% data compression ratio** by storing a Petabyte of raw transactional data in less than 260 Terabytes of actual disk space.
- It demonstrated an **average Ready-Time of less than 2 seconds** for data freshly added to the data warehouse.
- It replaced half of the "***T***" (Transactional) data with over 72 Terabytes of "***EDM***" (Emails, Documents and Multimedia) data, creating a data warehouse populated with **572 Terabytes of raw "*EDMT*" data**.
- It showed a **load speed of 26 Terabytes per day** when populating the data warehouse with 185 million documents (emails, attachments and other unstructured documents.)
- It reached **loading rates of 2 million emails per hour** and **6 million documents per hour** while consuming less than 7% of the available CPU power, leaving 93% of the M9000's CPUs available for other activities.
- It demonstrated a substantial reduction in the number of disk drives needed for storage, translating directly into at least **90% reduction in CO_2 emission** over the lifetime of the Sun Data Warehouse Reference Architecture using Solaris 10 OS, Sybase IQ and BMMsoft Server.

Configuration Overview

The "production-class" environment used in this Sun Data Warehouse Reference Architecture included a single-node Sun SPARC® technology Enterprise M9000 server running Solaris™ 10 OS, interconnected to three Sun StorageTek™ 6540 storage arrays using fiber channels and holding a single database image managed by Sybase® IQ version 12.7.

The following diagram details the platform used in this reference architecture:



Memory Allocation

- The Sybase IQ writer ran on 64 cores (for a total of 128 threads) and was allocated about 100 GB of main memory.
- About 45 GB of main memory was made available to the Sybase IQ load process and used to cache portions of input data files during the database population.
- The BMMsoft Server load server ran on 64 cores (for a total of 128 threads) and was allocated about 90 GB of main memory, of which 50GB was used in support of Sybase IQ and 4GB was used exclusively by BMMsoft Server.
- About 20 GB of main memory was allocated to Solaris 10 to optimize swapping and paging.

Growing to One Petabyte

Creating the database

This Sun Data Warehouse Reference Architecture was built using a star schema centered around 100 fact tables populated with "***T***" (transactional) data representing stock trading quotes.

Six dimensional tables were populated to provide for a wide range of table cardinalities (from 5 rows to nearly 40 million rows). Following are details on the population of the dimension tables:

Dimension Tables	Row Count
STOCK_HISTORY	37840000
RESIDENTIAL_CUSTOMER	100000
INSTRUMENT	2000
DOC_TYPES	10
CURRENCY	5
EXCHANGE	5

Populating the fact table

The 100 fact tables were created in Sybase IQ. Each table was then populated with over 120 of the load units described below:

Input Data File	Load Unit	
Raw File Size	*Byte*	102,814,873,296
Number of Records	*Million*	508
Record Width	*Column*	18
Size of Delimiters	*Byte*	9,146,722,464
Size with Delimiters	***Gbyte***	**95.8**
Pure Data Size	***Gbyte***	**87.2**

A set of real-life transactional data was used as a template to synthetically create the data in each load units. As a result the distributions, cardinalities and correlations of the resulting data warehouse population is believed to be representative of a real-life, production environment.

© InfoSizing, Inc.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

Reaching the Petabyte *milestone*

The population of the 100 fact tables took place over a period of about 6 weeks. Half of the population period was dedicated to system tuning and reconfiguration. The rest of the time (about 3 weeks) was dedicated to loading data into tables. On average, the population of these tables was done at the rate of 1.8 Terabytes of raw data per hour, representing a daily population rate of over 250 Billion rows.

In total, the tables were populated with over one Petabyte of ***pure raw data*** *(i.e.*, where bytes that do not contain actual data are excluded from the count.) In this case, *pure raw data* excludes any bytes that were used as separators between columns. While the database was populated from 1,130 Terabytes of input files including separators, these files contained a total of 1,029 Terabytes of *pure raw data* (i.e., excluding column separators).

In addition to the stock trading records, an initial 128 Gigabytes of email, text and multimedia documents, was populated in the data warehouse and managed by BMMsoft Server.

The record setting population of the one Petabyte Sun Data Warehouse Reference Architecture is detailed below:

Petabyte Data Population		
Raw Input Data		
Number of Load Units	*Data File*	12,081
Number of Stock Quote Records	***Trillion***	**6.1**
Raw Data Size (with delimiters)	*Tbyte*	1,130
Pure Data Size (w/o delimiters)	*Tbyte*	1,029
Pure Input Data Size	***Pbyte***	**1.01**
Average Document Size	*Kbyte*	**670**
Number of Unstructured Documents	***Million***	**0.2**
Unstructured Data Size	*Gbyte*	128
Total Data Size	*Tbyte*	1,157
	PetaByte	**1.13**
Sybase IQ Storage		
Space for ***T*** Data and Indices	*Gbytes*	162,639
	TeraBytes	**159**
	Compression	**84.57%**

Once the above population was completed, a global view was defined to form a "UNION ALL" of the 100 fact tables, creating a single access point for the entire population of over 6 Trillion rows. A series of performance evaluation tests were then executed against the Petabyte enterprise data warehouse. The results of these tests are documented in the next section of this report.

© InfoSizing, Inc.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

Petabyte Performance Measurements

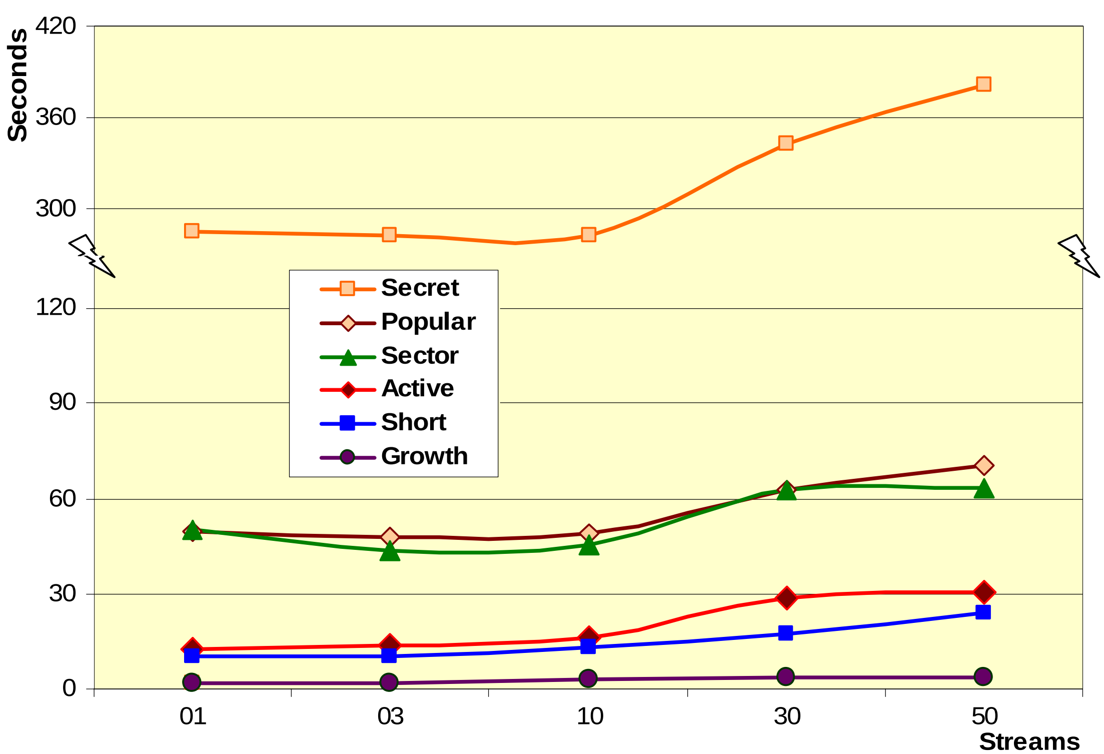
Query execution times

The Petabyte population was validated by executing multi-table queries involving the view representing the global fact table and some select dimension tables. These queries intended on verifying that the enterprise data warehouse was fully operational and that response times were kept within acceptable ranges.

In total, six different queries were executed from an increasing number (from 1 to 50) of concurrent streams. The queries were:

- The "Active" and "Sector" queries: simple queries against the "***T***" data.
- The "Short" and "Growth" queries: complex queries against the "***T***" data.
- The "Secret" and "Popular" queries: complex queries against mixed "***EDMT***" data.

The following graph illustrates the response times that were observed when running the six queries against the Petabyte data warehouse:



© InfoSizing, Inc.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

Ready-Time for new data

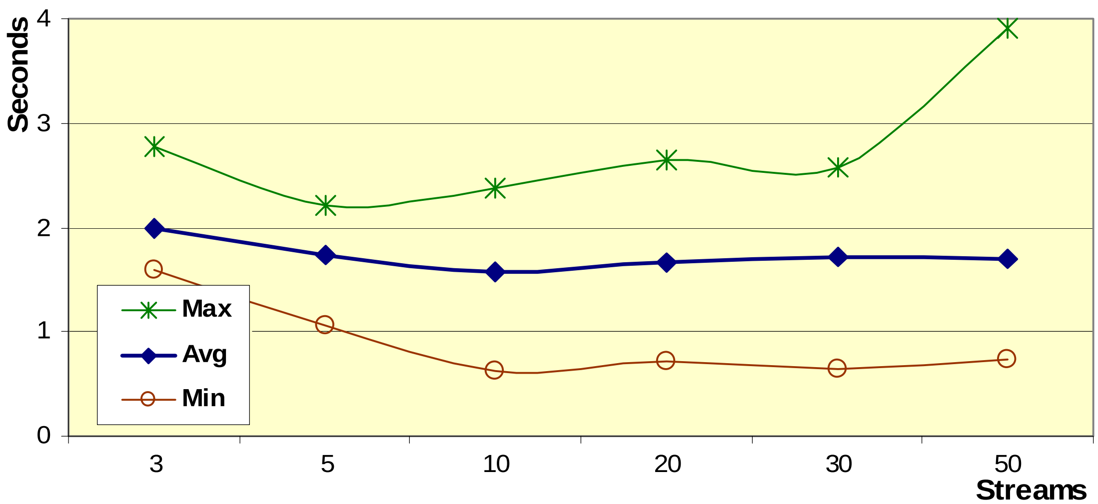
In an environment where new data is constantly generated in real time, the speed at which that new data can be added to the data warehouse and be seen by running queries is a crucial metric. The delay between the real-time generation of a new piece of data and it's availability to running queries is called the "Ready-Time".

Response-Time for simple queries– While the measurement of Ready-Times was the main focus of this performance test, it should be mentioned that the simple Response-Times for "pin-point" queries against transactions, email or documents in the Petabyte database was always sub-seconds.

Ready-Time for new email – The Ready-Time of a new email message was measured. Queries were executed from an increasing number (from 3 to 50) of concurrent streams. Each query was targeting an email containing a specific string, or tag, in its body.

While these queries were executing, new email messages were generated, captured by BMMsoft Server, and passed to Sybase IQ for storage in the data warehouse. The time between the creation of a new email message and its presence in the answer to a running query (the Ready-Time) was measured.

This test quantifies how the Sun Data Warehouse Reference Architecture operated with minimal delays between the generation of a new email message and its presence in the answer set of a query targeting it. The measurement of Ready-Time for new email messages was first conducted against the Petabyte database. The following graph illustrates the results obtained:



The average email Ready-Times measured against the Petabyte database was under 2 seconds. In fact, the maximum Ready-Time stayed under 4 seconds.

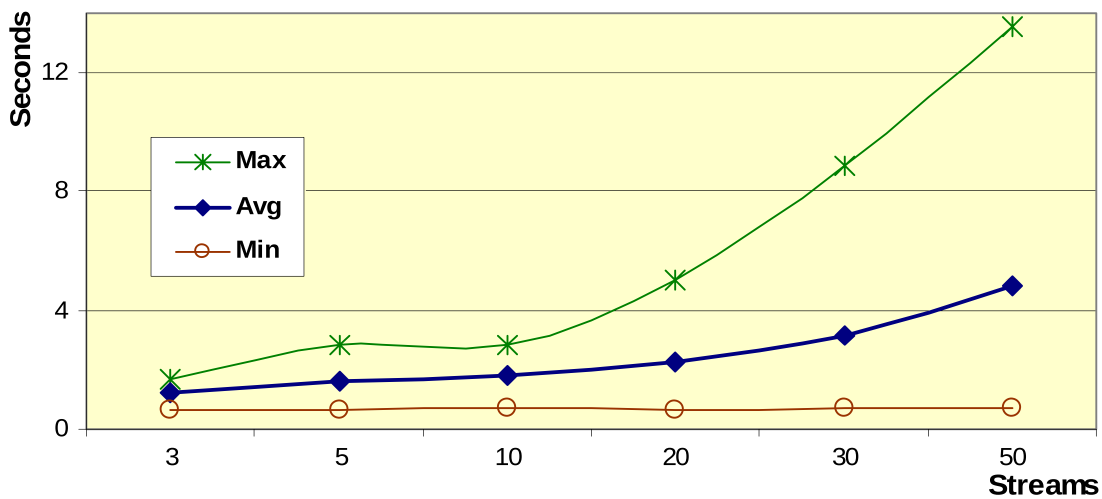
© InfoSizing, Inc.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

***Ready-Time for new stock quotes* –** The Ready-Time of a new completed stock transaction was measured. Queries were executed from an increasing number (from 3 to 50) of concurrent streams. Each query was targeting existing quotes for a specific stock. While these queries were executing, new stock transactions were completed and the resulting records were passed to Sybase IQ.

This test quantifies how the Sun Data Warehouse Reference Architecture using Solaris 10 OS, Sybase IQ and BMMsoft Server operated with minimal delays between the generation of a record reflecting a new transaction and its presence in the answer set of a query targeting it.

The time between the creation of a new transaction record and its presence in the answer to a running query (the Ready-Time) was measured against the EDMT data warehouse. The following graph illustrates the results that were obtained:



The average Ready-Times for new transactional data measured against the Petabyte data warehouse was under 5 seconds.

Populating Unstructured Data

Subsequently, 50% of the "***T***" (Transactional) data was dropped from the data warehouse and the available capacity was used to load a significant population of additional "***EDM***" (Email, Documents and Multimedia) data.

The "***EDM***" data was comprised of three types of unstructured data objects: emails, small documents (i.e., plain text) and large documents (i.e., video clips, graphic images and formatted text). The details of the populated unstructured data are shown below:

		Total	eMail	Small	Large
Average Object Size	*Kilobyte*		0.343	2.47	203,205
Objects Loaded	***Million***	**185**	59.0	125	0.374
Raw Size Loaded	***Gbyte***	**72,721**	19.3	295	72,406
Load Speed					
	Megabyte/sec.		0.189	4	317
	Object/hour		2 Million	6 Million	5,000

The new "***EDM***" data was loaded using only 7% of the available CPU power, or about 6 out of the 64 processor cores. The resulting population of mixed "***EDMT***" data, combining the remaining 50% of "***T***" data from the Petabyte data warehouse with the newly populated "***EDM***" data, is detailed below:

Mixed ***EDMT*** Data Population		
Raw Input Data		
Number of Load Units	*Data File*	5,476
Number of Stock Quote Records	***Trillion***	**2.8**
Total Pure ***T*** (transactional) Data Size	*Terabyte*	467
Number of *EDM* Documents	***Million***	**185**
Total Pure ***EDM*** (unstructured) Data Size	*Terabyte*	71.02
Total Mixed (*EDM* + *T*) Data Size	***Terabyte***	**537.5**

The unstructured "***EDM***" objects were managed by BMMsoft Server and stored in the data warehouse by Sybase IQ. Some documents were independent files of various formats and content. Other documents were email messages, 80% of which had attachments.

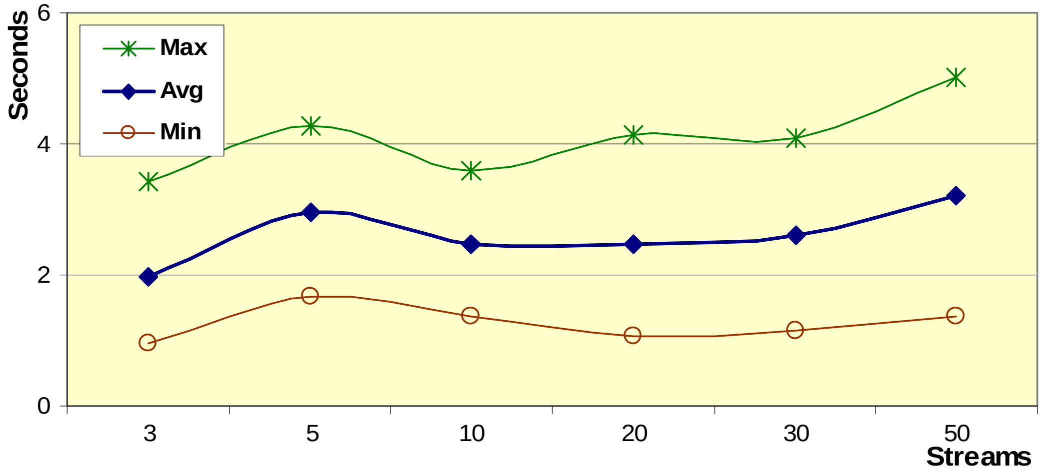
During the parsing of the email messages by BMMsoft Server, each attachment was separated from the body of its parent email to form two individual documents. These two documents were stored along with sufficient metadata to capture the relationship between the email's body and its attachment.

© InfoSizing, Inc.

EDMT Performance Measurements

Ready-Time for new email

The measurement of Ready-Time for new email messages was then repeated against the mixed ***EDMT*** data warehouse. The following graph illustrates the results that were obtained:



The average Ready-Time for new email, when measured against the mixed ***EDMT*** data warehouse, was under 3 seconds. In fact, after increasing the unstructured data population with 185 million new documents, the increase in average Ready-Time for new email stayed negligible, at about 5%.

The average transaction Ready-Time, when measured against the mixed ***EDMT*** database, was under 5 seconds. In fact, after increasing the population with 185 million documents, the average Ready-Time only increased by around 5%.

Reduction in CO_2 Emission

From data compression to CO_2 reduction

A reduction in the electrical power needed to operate a data warehouse can be directly translated into a reduction of global CO_2 emission. Given the measure of data compression achieved by the Sun Data Warehouse Reference Architecture using Solaris 10 OS, Sybase IQ and BMMsoft Server, a corresponding level of reduction in CO_2 emission can be computed.

The storage space needed by other, more conventional data warehouse solutions is typically greater than the size of the raw data populating it. Using a "row store" model with a medium level of indexing can results in storage requirements that are several times the size of the initial raw data.

In contrast, the measured Reference Architecture demonstrated a level of data compression of up to 85%. Because of this very substantial level of data compression, the Sun Data Warehouse Reference Architecture required less than 10% of the electricity consumed by other, more conventional solutions. Similarly, the floor space, size and weight of the configured storage devices were also reduced by at least 90% in the Reference Architecture.

Calculating life cycle CO_2 reduction

According to its technical specifications, and factoring a standard 50% power overhead for cooling, the power consumption of the tested configuration was 90KW, totaling over 780 MWh for a full year of operation. Conventional solutions with similar data capacity and performance levels would require about ten times more storage devices and consume over 7,800 MWh per year.

Based on the generally accepted "*pollution factor*" of 1.34 pounds of CO_2 per KWh, the Sun Data Warehouse Reference Architecture would **reduce CO_2 emission by 14,000 tons during its 3-year life cycle**.

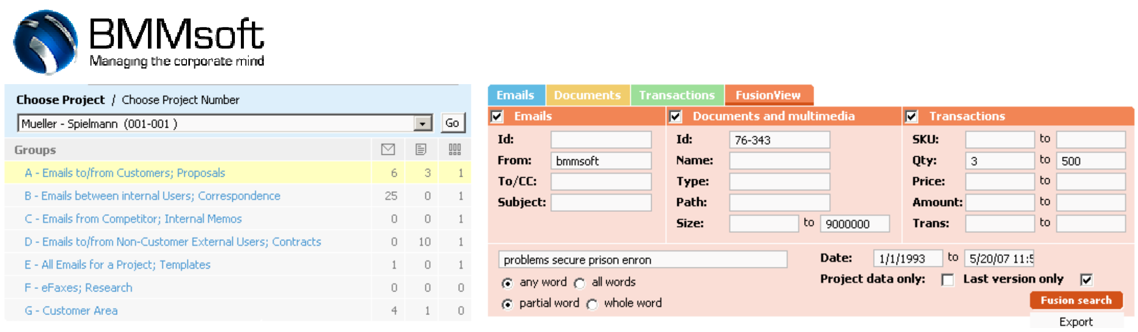
By the end of its life cycle, it is predicted that the Sun Data Warehouse Reference Architecture using Solaris 10 OS, Sybase IQ and BMMsoft Server can reduce by 90% the emission of all pollutants that would be generated by conventional solutions with similar data capacity and performance levels. It is expected to have also **reduced by 26 tons the volume of end-of-life storage devices to be discarded**.

© InfoSizing, Inc.

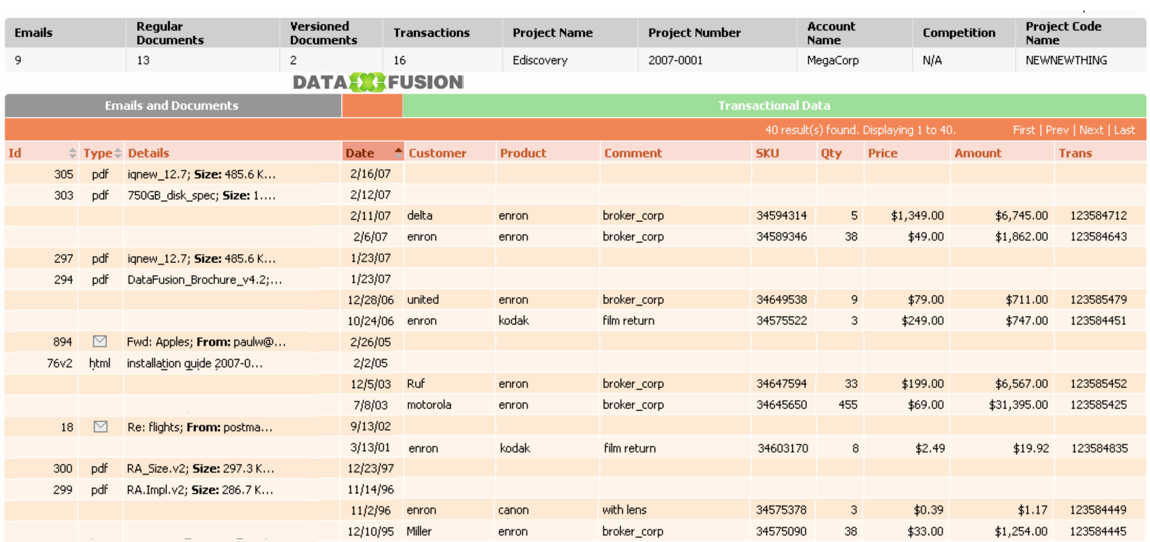
Cross-Correlation of Structured and Unstructured Data

Transactional and unstructured data can be cross-correlated through the use of BMMsoft Server. Unstructured objects are analyzed in real-time when inserted into the data warehouse, and the resulting metadata for each object is also stored in the data warehouse.

The screen-shot below illustrates how project folders (such as "Mueller-Spielmann" in this example) can be created to organize emails, attachments and other unstructured objects. These folders are automatically populated based on predefined and user-defined rules. A wide range of textual and numerical parameters for relational and full-text search, as well as cross-correlation of unified data can be entered on the right side of the screen.



The results of complex EDMT analysis are displayed in the unified view of EDMT data. Emails, documents and transactions are seamlessly analyzed, in real time, and integrated results can be displayed or exported to a file. The screen-shot below illustrates how potentially fraudulent stock trades, insider trading or other targeted events can be captured in real time.



Emails	Regular Documents	Versioned Documents	Transactions	Project Name	Project Number	Account Name	Competition	Project Code Name
9	13	2	16	Ediscovery	2007-0001	MegaCorp	N/A	NEWNEWTHING

DATA FUSION

Emails and Documents | Transactional Data

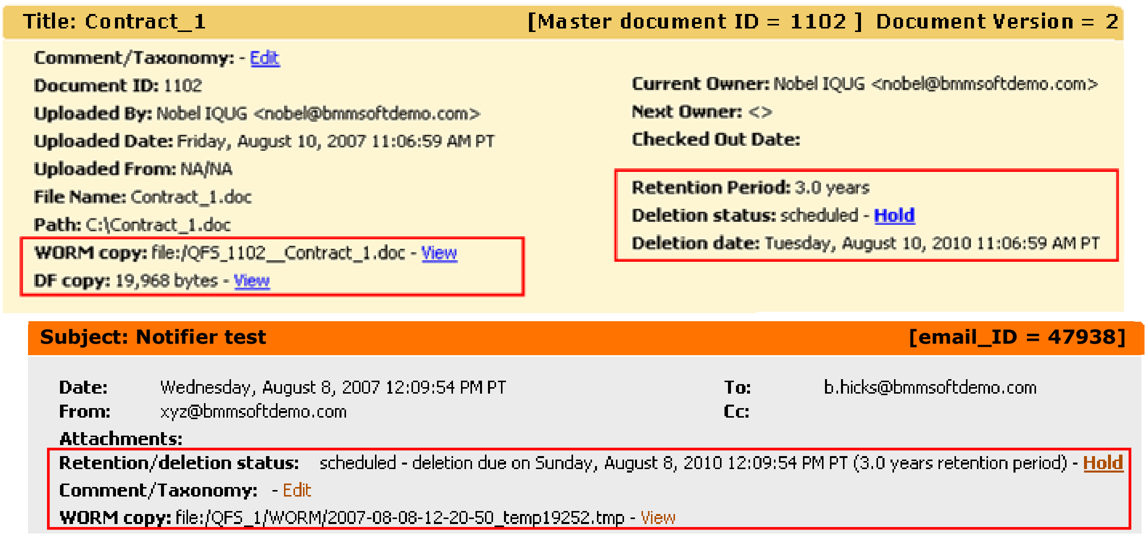
40 result(s) found. Displaying 1 to 40. First | Prev | Next | Last

Id	Type	Details	Date	Customer	Product	Comment	SKU	Qty	Price	Amount	Trans
305	pdf	iqnew_12.7; **Size:** 485.6 K...	2/16/07								
303	pdf	750GB_disk_spec; **Size:** 1....	2/12/07								
			2/11/07	delta	enron	broker_corp	34594314	5	$1,349.00	$6,745.00	123584712
			2/6/07	enron	enron	broker_corp	34589346	38	$49.00	$1,862.00	123584643
297	pdf	iqnew_12.7; **Size:** 485.6 K...	1/23/07								
294	pdf	DataFusion_Brochure_v4.2;...	1/23/07								
			12/28/06	united	enron	broker_corp	34649538	9	$79.00	$711.00	123585479
			10/24/06	enron	kodak	film return	34575522	3	$249.00	$747.00	123584451
894		Fwd: Apples; **From:** paulw@...	2/26/05								
76v2	html	installation guide 2007-0...	2/2/05								
			12/5/03	Ruf	enron	broker_corp	34647594	33	$199.00	$6,567.00	123585452
			7/8/03	motorola	enron	broker_corp	34645650	455	$69.00	$31,395.00	123585425
18		Re: flights; **From:** postma...	9/13/02								
			3/13/01	enron	kodak	film return	34603170	8	$2.49	$19.92	123584835
300	pdf	RA_Size.v2; **Size:** 297.3 K...	12/23/97								
299	pdf	RA.Impl.v2; **Size:** 286.7 K...	11/14/96								
			11/2/96	enron	canon	with lens	34575378	3	$0.39	$1.17	123584449
			12/10/95	Miller	enron	broker_corp	34575090	38	$33.00	$1,254.00	123584445

© InfoSizing, Inc.

Compliance with Regulatory Data Retention Requirements

Sun's StorageTek QFS Software with WORM-FS feature was configured as part of the platform used for the testing of the Petabyte data warehouse. The intent was to demonstrate the ability of BMMsoft Server to use external WORM storage subsystems to comply with the regulatory data retention requirements (e.g., SOX NASD, SEC 17a-4, HIPAA, GLBH and others.)



The above screen-shot illustrates how BMMsoft Server can be configured to automatically store new data (such as email, attachments or other unstructured documents) in external WORM devices, in BMMsoft Server's underlying storage server (such as Sybase IQ), or simultaneously in both.

© InfoSizing, Inc.

Query Set

The query set used for the performance test was made of the following 6 queries:

The "Active" Query

Determine the top 10 most active stocks for a specified date sorted by cumulative trade volume by considering all trades

```
SELECT
        TOP 10 TRADING_SYMBOL,
        SUM(BID_SIZE) AS TRADESIZE,
        DENSE_RANK OVER (ORDER BY SUM(BID_SIZE) DESC) AS RANKING
   FROM
       STOCK_QUOTE_VIEW
   WHERE
       QUOTE_DATE = '2007-07-04'
   GROUP BY
       TRADING_SYMBOL
   ORDER BY
       SUM(BID_SIZE) DESC;
```

The "Sector" Query

Find the most active stocks in the "COMPUTER" industry for the current day.

```
SELECT
        ST.TRADING_SYMBOL,
        SUM(BID_SIZE) TRADESIZE,
        DENSE_RANK() OVER (ORDER BY SUM(BID_SIZE) DESC) AS RANKING
    FROM
        STOCK_QUOTE_VIEW ST
    INNER JOIN
        INSTRUMENT II ON
            II.TRADING_SYMBOL = ST.TRADING_SYMBOL
    INNER JOIN
        SCND_IDST_CLS SC ON
            II.SCND_IDST_CLS_ID = SC.SCND_IDST_CLS_ID AND
            SC.SIC_NAME = 'COMPUTERS'
    WHERE
        ST.QUOTE_DATE = '2007-07-04'
    GROUP BY
        ST.TRADING_SYMBOL;
```


© InfoSizing, Inc.

The "Secret" Query

Determine if there is any correlation between quotes (BIDS) tendered and emails urging SECRECY about that STOCK.

```
SELECT
        TRADING_SYMBOL,
        QUOTE_DATE,
        SUM(BID_SIZE)
    FROM
        STOCK_QUOTE_VIEW,
        BMM_MESSAGEHEADER,
        BMM_MESSAGELONGTEXT
    WHERE
        TEXTTYP = 'P' AND
        MESSAGETEXT LIKE '%SECRET%' AND
        MESSAGE_ID = MESSAGE_ID AND
        SUBJECT = TRADING_SYMBOL AND
        QUOTE_DATE = '2007-07-04'
    GROUP BY
        TRADING_SYMBOL,
        QUOTE_DATE;
```

The "Popular" Query

Determine the STOCKS that are most discussed in eMail correspondence on a particular day.

```
SELECT
        COUNT(*),
        BM.MESSAGE_TIME,
        SQ.TRADING_SYMBOL,
        SUM(SQ.BID_SIZE) AS TRADESIZE,
        DENSE_RANK() OVER
             (ORDER BY SUM(SQ.BID_SIZE) DESC) AS RANKING
    FROM
        STOCK_QUOTE_VIEW SQ,
        BMM_MESSAGEHEADER BM
    WHERE
        SQ.QUOTE_DATE = '2007-07-04' AND
        BM.SUBJECT = SQ.TRADING_SYMBOL
    GROUP BY
        BM.MESSAGE_TIME,
        SQ.TRADING_SYMBOL
    ORDER BY
        SUM(SQ.BID_SIZE) DESC;
```


© InfoSizing, Inc.

The "Short" Query

Determine the top 10 percentage losers for the specified date on the specified exchanges sorted by percentage loss. The loss is calculated as a percentage of the last trade price of the previous day.

```
SELECT
        TOP 10 INSTRUMENT_ID,
        TRADING_SYMBOL,
        TRADE_PRICE,
        TRADE_DATE
    INTO #TEMP_TICK3A
    FROM
        DBA.STOCK_TRADE ST,
       (SELECT
                INSTRUMENT_ID AS IDX,
                MAX(TRADE_TIME) AS MAXTIME
            FROM
                DBA.STOCK_TRADE
            WHERE
                TRADE_DATE = '2005-11-14'
            GROUP BY
                INSTRUMENT_ID
       ) Y
    WHERE
        ST.TRADE_DATE = '2005-11-14' AND
        ST.TRADE_TIME = MAXTIME AND
        IDX = ST.INSTRUMENT_ID;

CREATE VARIABLE PREV_DAY DATE;
SET PREV_DAY = (SELECT MAX(TRADE_DATE)
                    FROM STOCK_TRADE
                    WHERE TRADE_DATE < '2005-11-14');

SELECT
        INSTRUMENT_ID,
        TRADING_SYMBOL,
        TRADE_PRICE,
        TRADE_DATE
    INTO #TEMP_TICK3B
    FROM
        DBA.STOCK_TRADE ST,
       (SELECT
                INSTRUMENT_ID AS IDX,
                MAX(TRADE_TIME) AS MAXTIME
            FROM
                DBA.STOCK_TRADE
            WHERE
                TRADE_DATE = PREV_DAY
            GROUP BY
                INSTRUMENT_ID
       ) Y
    WHERE
        ST.TRADE_DATE = PREV_DAY AND
        ST.TRADE_TIME = MAXTIME AND
        IDX = ST.INSTRUMENT_ID;
```

The "Short" Query (continued)

```
DROP VARIABLE PREV_DAY;

SELECT
        INSTRUMENT_ID,
        TRADING_SYMBOL,
        PER_LOSER,
        LOSER_RANK
    FROM
        (SELECT
                INSTRUMENT_ID,
                TRADING_SYMBOL,
                PER_LOSER,
                RANK() OVER (ORDER BY PER_LOSER ASC) LOSER_RANK
            FROM
               (SELECT
                       T.INSTRUMENT_ID,
                       T.TRADING_SYMBOL,
                      (T.MTP - Y.MTP)*100/Y.MTP PER_LOSER
                   FROM
                      (SELECT
                              INSTRUMENT_ID,
                              TRADING_SYMBOL,
                              TRADE_PRICE MTP
                          FROM
                              #TEMP_TICK3A) T,
                              (SELECT
                                      INSTRUMENT_ID,
                                      TRADING_SYMBOL,
                                      TRADE_PRICE  MTP
                                  FROM
                                      #TEMP_TICK3B
                              ) Y
                          WHERE T.INSTRUMENT_ID = Y.INSTRUMENT_ID
                      ) A
               ) B
            WHERE PER_LOSER < 0;
```


© InfoSizing, Inc.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

The “Growth” Query

Determine the value of $100,000 now if 1 year ago it was invested equally in 10 specified stocks (i.e. allocation for each stock is $10,000). The trading strategy is as follows: when the 20-day moving average crosses over the 5-month moving average, the allocation is invested; and when the 20-day moving average crosses below the 5-month moving average, the position is sold.

```
TRUNCATE TABLE HIST_TEMP;
TRUNCATE TABLE HIST7_TEMP;
COMMIT;

INSERT HIST_TEMP SELECT
        NUMBER(),
        B.INSTRUMENT_ID,
        B.TRADING_SYMBOL,
        B.TRADE_DATE, B.CLOSE_PRICE,
        IFNULL( SUM(A.SPLIT_FACTOR), 1, SUM(A.SPLIT_FACTOR) )
    FROM
        STOCK_HISTORY AS B
    LEFT OUTER JOIN
        SPLIT_EVENT AS A ON
            B.INSTRUMENT_ID = A.INSTRUMENT_ID AND
            B.TRADE_DATE < A.EFFECTIVE_DATE
    WHERE
        B.INSTRUMENT_ID BETWEEN 11 AND 20 AND
        B.TRADE_DATE >= DATEADD(DAY,-160,'2012-06-01') AND
        B.TRADE_DATE <= '2012-12-01'
    GROUP BY
        B.INSTRUMENT_ID,
        B.TRADING_SYMBOL,
        B.TRADE_DATE,
        B.CLOSE_PRICE
    ORDER BY
        B.INSTRUMENT_ID,
        B.TRADE_DATE;
```


© InfoSizing, Inc.

The "Growth" Query (continued)

```
INSERT HIST7_TEMP SELECT
        NUMBER(),
        X.INSTRUMENT_ID,
        X.TRADE_DATE,
        AVG_5MTH ,
        AVG_21DAY
    FROM
         (SELECT
                B.INSTRUMENT_ID,
                B.TRADE_DATE,
                AVG(C.CLOSE_PRICE * B.SPLIT_FACTOR) AVG_5MTH
            FROM
                HIST_TEMP AS B
            LEFT OUTER JOIN
                HIST_TEMP AS C ON
                    B.INSTRUMENT_ID = C.INSTRUMENT_ID AND
                    C.ROW_NBR BETWEEN B.ROW_NBR - 160 AND
                    B.ROW_NBR
            GROUP BY
                B.INSTRUMENT_ID,
                B.TRADE_DATE
         ) X,
         (SELECT
                B.INSTRUMENT_ID,
                B.TRADE_DATE,
                AVG(C.CLOSE_PRICE * B.SPLIT_FACTOR) AVG_21DAY
            FROM
                HIST_TEMP AS B
            LEFT OUTER JOIN
                HIST_TEMP AS C ON
                    B.INSTRUMENT_ID = C.INSTRUMENT_ID AND
                    C.ROW_NBR BETWEEN B.ROW_NBR - 21 AND B.ROW_NBR
            GROUP BY
                B.INSTRUMENT_ID,
                B.TRADE_DATE
         ) Y
    WHERE
        X.INSTRUMENT_ID = Y.INSTRUMENT_ID AND
        X.TRADE_DATE = Y.TRADE_DATE
    ORDER BY
        X.INSTRUMENT_ID,
        X.TRADE_DATE;
```


© InfoSizing, Inc.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

The "Growth" Query (continued)

```
SELECT
        Z.INSTRUMENT_ID,
        Z.TRADE_DATE,
        DIFF,
        TD2,
        DIFF2,
        PRE_DIFF
    INTO #HIST7_TEMP
    FROM
       (SELECT
                A.INSTRUMENT_ID,
                A.TRADE_DATE,
                B.AVG_21DAY - B.AVG_5MTH AS PRE_DIFF
            FROM
                HIST7_TEMP A,
                HIST7_TEMP B
            WHERE
                A.INSTRUMENT_ID = B.INSTRUMENT_ID AND
                B.ROW_NBR = A.ROW_NBR - 1
       ) X,
       (SELECT
                A.INSTRUMENT_ID,
                A.TRADE_DATE,
                B.TRADE_DATE AS TD2,
                B.AVG_21DAY - B.AVG_5MTH AS DIFF2
            FROM
                HIST7_TEMP A,
                HIST7_TEMP B
            WHERE
                A.INSTRUMENT_ID = B.INSTRUMENT_ID AND
                B.ROW_NBR = A.ROW_NBR + 1
       ) Y,
       (SELECT
                INSTRUMENT_ID,
                TRADE_DATE,
                AVG_21DAY - AVG_5MTH AS DIFF
            FROM
                HIST7_TEMP
       ) Z
    WHERE
        Z.INSTRUMENT_ID = X.INSTRUMENT_ID AND
        Z.TRADE_DATE = X.TRADE_DATE AND
        Z.INSTRUMENT_ID = Y.INSTRUMENT_ID AND
        Z.TRADE_DATE = Y.TRADE_DATE AND
        PRE_DIFF*DIFF <= 0 AND
        NOT (PRE_DIFF = 0 AND DIFF=0);
```


© InfoSizing, Inc.

125 West Monroe Street • Colorado Springs, CO 80907 • 719-473-7555 • www.sizing.com

The "Growth" Query (continued)

```
SELECT
        SUM(MP2.CLOSE_PRICE * (10000/MP1.CLOSE_PRICE)) AS S_VALUE
    FROM
        #HIST7_TEMP T7,
        STOCK_HISTORY MP1,
        STOCK_HISTORY MP2
    WHERE
        T7.INSTRUMENT_ID = MP1.INSTRUMENT_ID AND
        T7.INSTRUMENT_ID = MP2.INSTRUMENT_ID AND
        T7.TRADE_DATE = MP1.TRADE_DATE AND
        T7.TD2 = MP2.TRADE_DATE;
```

Sun, Sun Microsystems, Solaris, and StorageTek are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the US and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Sybase is a registered trademark of Sybase, Inc. in the United States and other countries.

BMMsoft Server is a trademark of BMMsoft, Inc. in the United States and other countries.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.