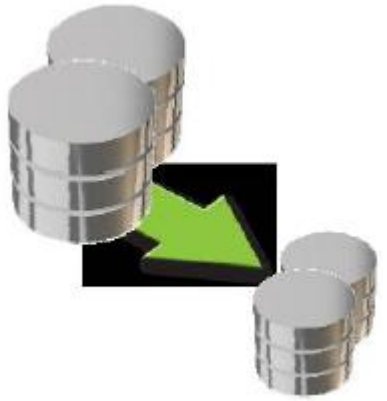
Honey I Shrunk the Data Warehouse! (Oracle Advanced Compression)

NYOUG March 19, 2009



Shyam Varan Nath
President, Oracle BIWA SIG
ShyamVaran@Gmail.com

Agenda

The Storage Problem

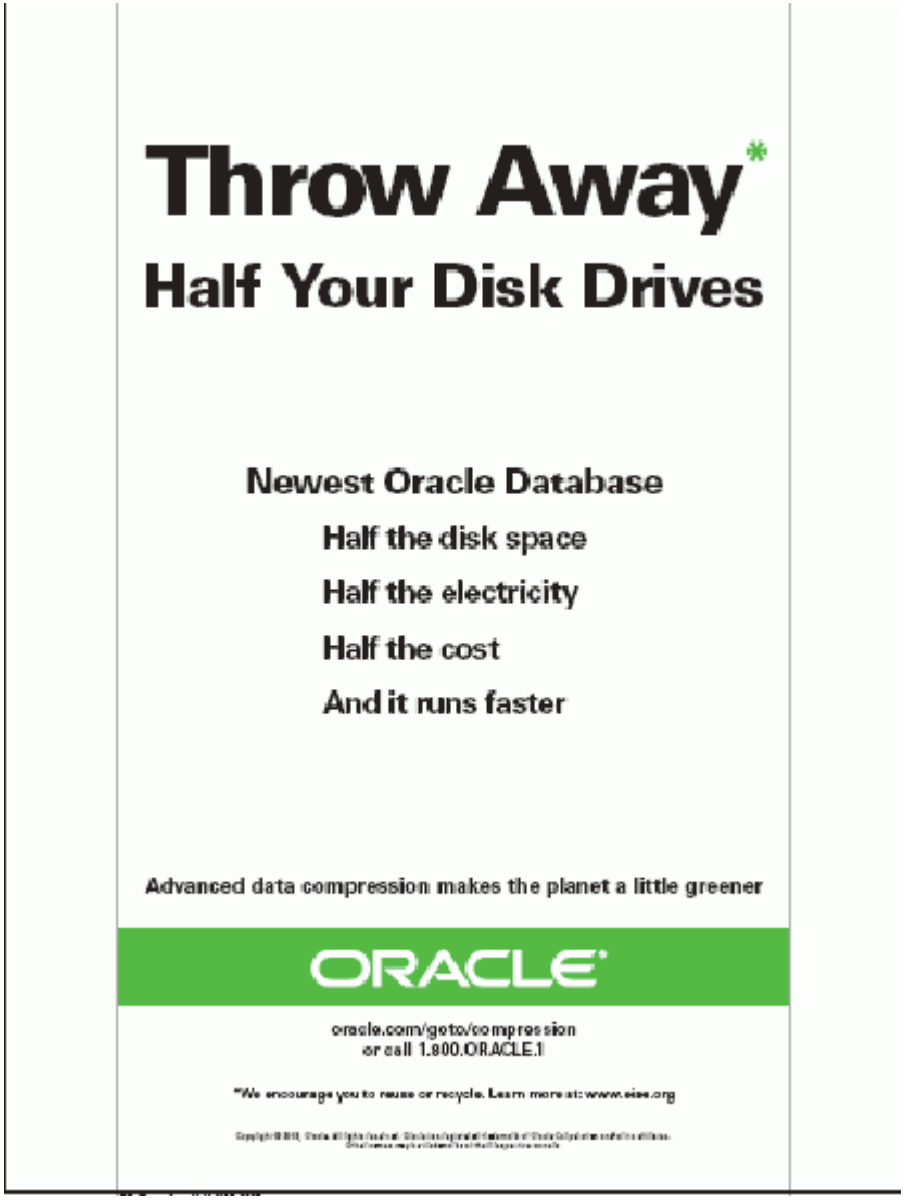
Introduction to Data Compression

Data Compression for Data Warehouse

Data Compression for OLTP

Next Steps

Q & A



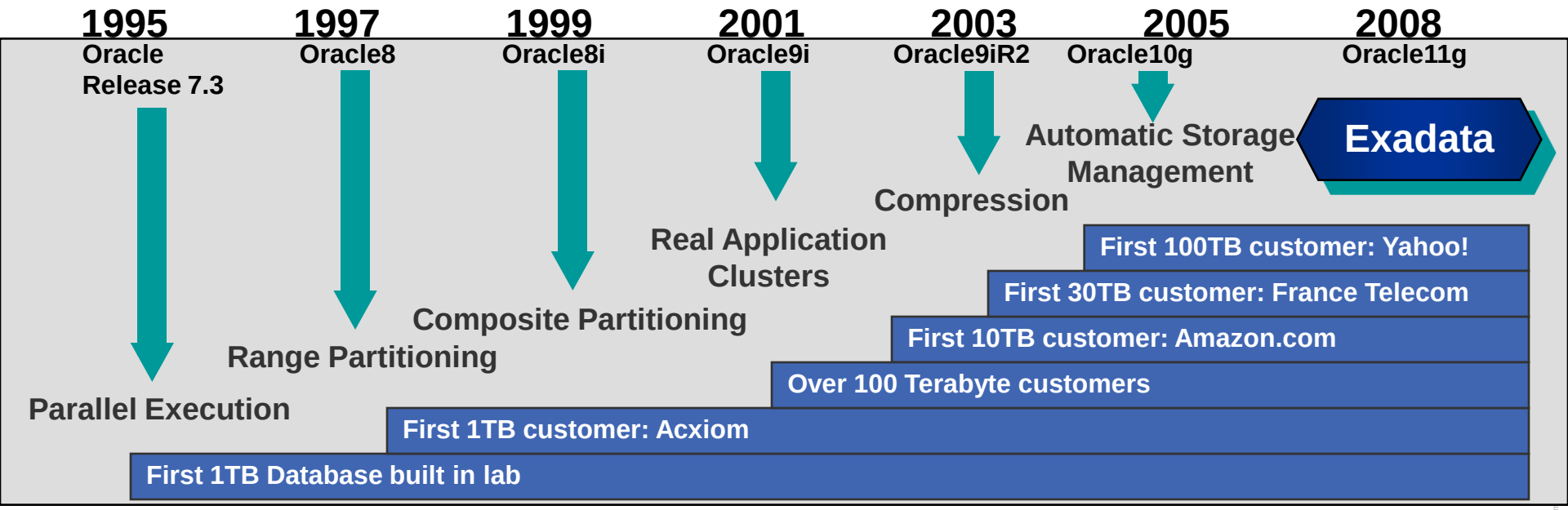
About Myself….

Word of Thanks to NYOUG – for this talk today

- A Certified DBA (OCP) on 4 different Database versions – since 1998
- Former member of Oracle Corporation - BI Consulting Practice
- Experience in Oracle Data Warehousing, Business Intelligence (OBIEE) and Data Mining
- Founder and President of Oracle BIWA SIG (http://OracleBIWA.org)
- Received IOUG Oracle Contribution Award in 2007
- Frequent speaker in Oracle Openworld (2003, 06, 07, 08), NYOUG (June 06, Sep 06, Sep 08), IOUG/Collaborate (2005, 06, 08), NOUG (2006), SFOUG (2007), ODTUG (2008) on topics ranging from Database to BI.
- Bachelors from Indian Institute of Technology (IIT), MBA and MS from Florida Atlantic University

The Growth of Databases

Here is the historic perspective of the growth of the databases in the industry and how Oracle has responded to that.



Source: Oracle Presentation by A Muley

The Storage Problem

ABC Inc.'s Data Warehouse is approaching 12 terabytes in size and growing by 100% every year! Storage and backup of data alone is costing 20% of the IT budget.

Today

How much is ABC Inc. spending in Storage?

Annual cost $1.0 m

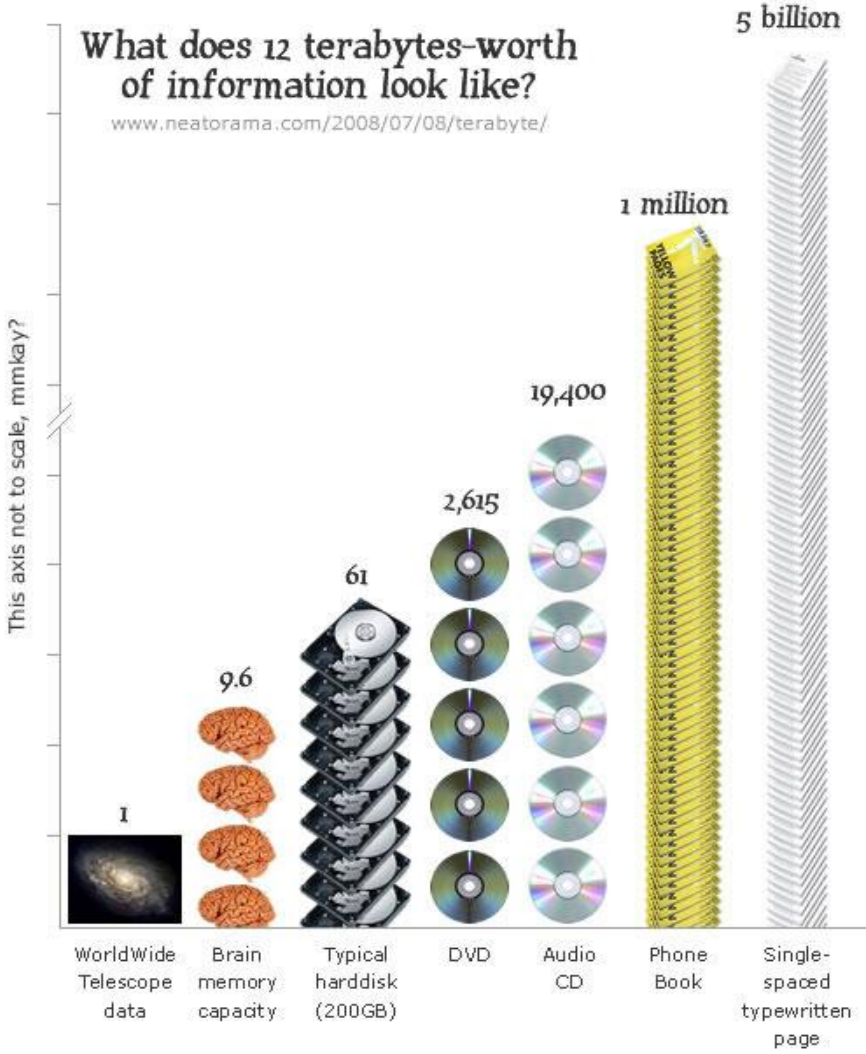
- Total IT budget is $5m, and storage cost is expected to double next year at the given rate

Tomorrow

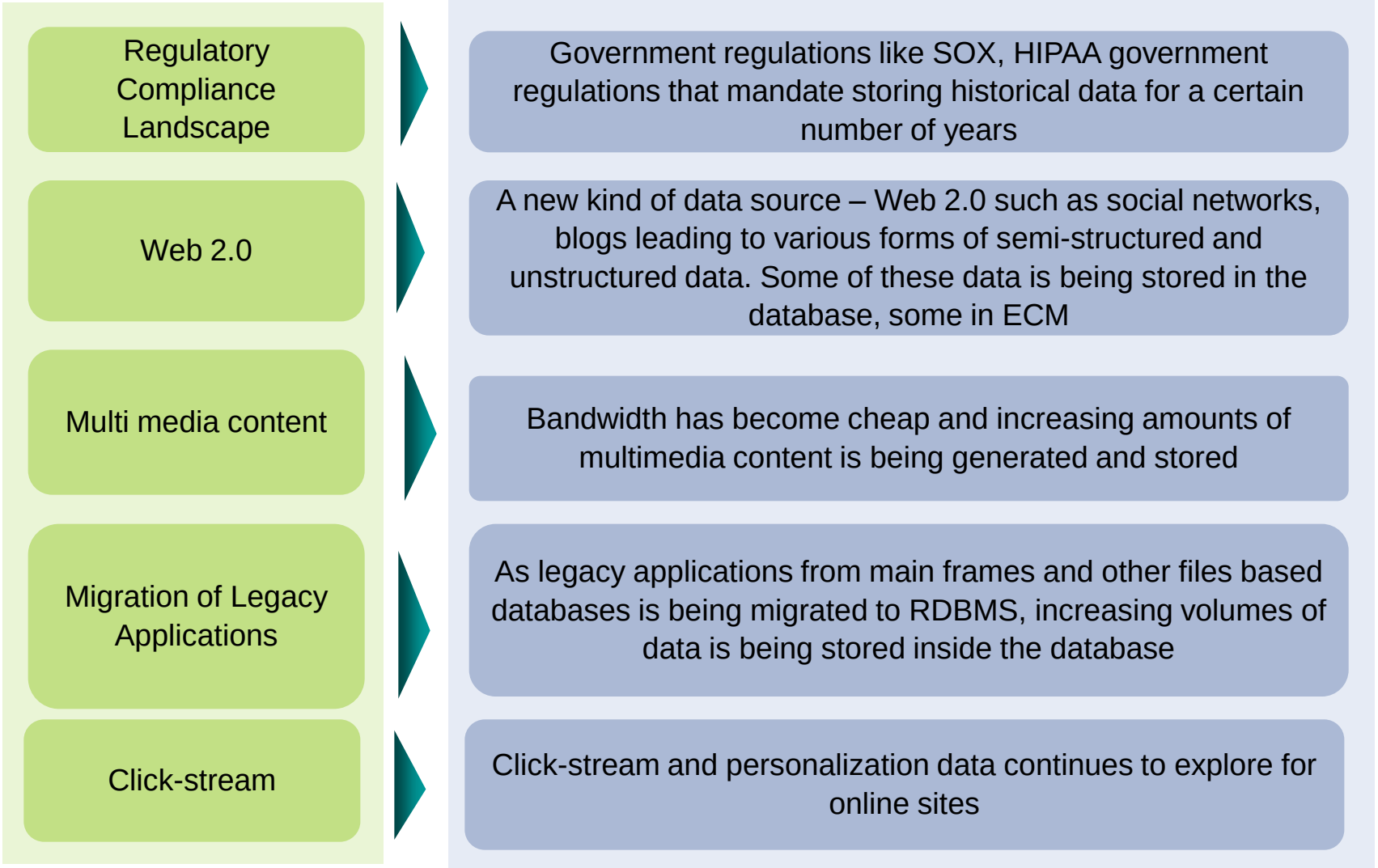
What are the other impacts of huge storage needs?

Information Retrieval is slow

- Not only is the Data Warehouse growing unmanageable in size, information query is slowing down leading to lost orders



What is causing the explosion of data in most enterprises?



Some Large Databases in use Today

•Yahoo's data needs are substantial.
•According to Hasan, VP of Data, the travel industry's Sabre system handles 50m events / day, credit card company Visa handles 120m events / day, and the New York Stock Exchange has handled over 225 m events / day.
•Yahoo, he said, handles **24 billion events / day**, fully two orders of magnitude more than other non-Internet companies.

Yahoo claims 2-petabyte database is world's biggest, busiest

2-petabyte, specially-built data warehouse used to analyze the behavior of Yahoo's half a billion Web visitors each month

Eric Lai (Computerworld) 23 May, 2008 08:20:43

Page: 1 2 next >

The petabyte is the new petaflop.

Interest in raw computational speed waned -- sorry, IBM -- after data center managers began turning away from super-expensive supercomputers toward massive grids comprised of cheap PC servers.

Meanwhile, the rise of business intelligence and its even more technical cousin, business analytics, has spurred interest in super-large data warehouses that boost profits by crunching the behavior of millions of consumers at a time.

Take Yahoo's 2-petabyte, specially-built data warehouse, which it uses to analyze the behavior of its half a billion Web visitors a month. The Sunnyvale, Calif. firm makes a

Have your say! Write a comment

Print this story

Digg this story

More by Eric Lai

ARN Directory | Distributors relevant to this article

ACA Pacific , Alloys , ASI Sol
Australasian PC Distributors
(APCD) , Avnet Technology
Solutions , Banksia Software
Cellnet , Computergate , Data
Technology Solutions , Dicke
Express Data , HYKA , Inforte
Distribution , Ingram Micro
Australia , itX , LDS Internati
Leader Computers , Leading
Australia , Lynx Technologie
Systems , Phoenix Toner , S
Consulting , Synnex Australi

Types of Compression

Loss-less compression

- Original content can completely be recovered
- E.g. gzip, winzip

Not the same as for example

- JPEG (images), MP3 (audio)
- MPEG (video)

	Adaptive Huffman	Lempel-Ziv
LaTeX file	66%	44%
Speech file	65%	64%
Image file	94%	88%

Size of compressed file as percentage of the original file

Compression Algorithms

- Huffman coding by David Huffman
- The basic idea in Huffman coding is to assign short codewords to those input blocks with high probabilities and long codewords to those with low probabilities.
- Same principle is used in Morse code (A ._ E .)

Statisticians study the frequency of occurrence of English alphabets

Physical v/s Logical Database Compression

Physical Database Compression

- Hardware assisted compression
  - IMS, the first commercially available database offered Hardware Assisted Data Compression (HDC)
  - It compressed the IMS blocks at the hardware level, completely transparent to the database engine.
- Block/page level compression
  - Historical database compression uses external mechanisms that are invisible to the database.
  - As block are written from the database, user exits invoke compression routines to store the compressed block on disk.
  - Examples include CLEMCOMP, Presspack and InfoPak

Logical Database Compression

- Table segment and Block Compression
  - Examples Oracle 9i R2, 10g, 11g

Physical Compression

Hardware assisted compression

- IMS, the first commercially available database offered Hardware Assisted Data Compression (HDC)
- It compressed the IMS blocks at the hardware level, completely transparent to the database engine.

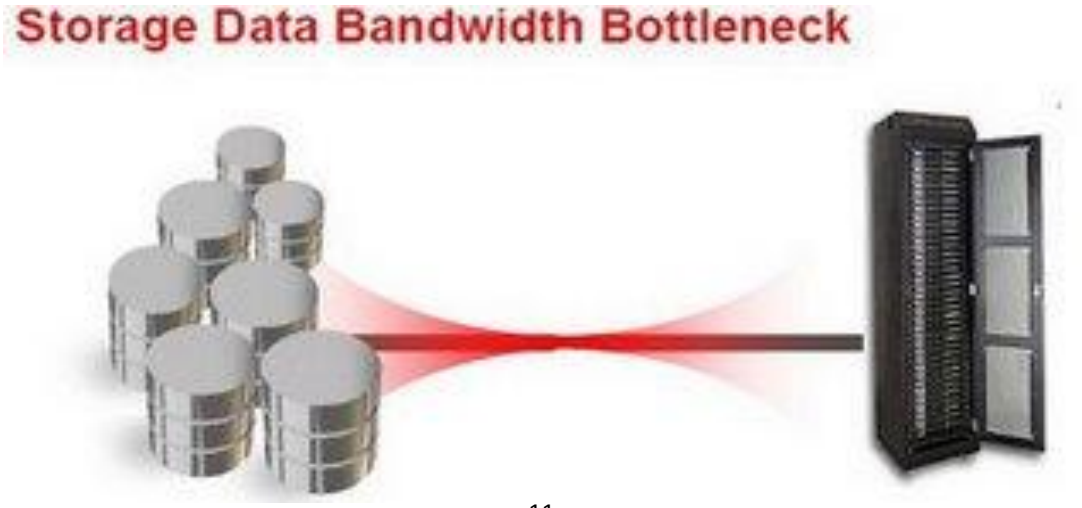
Block/page level compression

- Historical database compression uses external mechanisms that are invisible to the database.
- As block are written from the database, user exits invoke compression routines to store the compressed block on disk.
- Examples include CLEMCOMP, Presspack and InfoPak

Quote from TDWI

In any BI application, it's always disk I/O that slows performance.

•Data compression works hand in hand with partitioning
•Data in business intelligence environments typically remains active for a certain period, after which it does not change any more
• For example, a retailer may allow a 30- day return period after the purchase, after which the records will not change
•Once data is "frozen," it can be compressed and made available for query-only purposes
•A time-based partitioning scheme helps in identifying the data set that can be compressed. [TDWI]



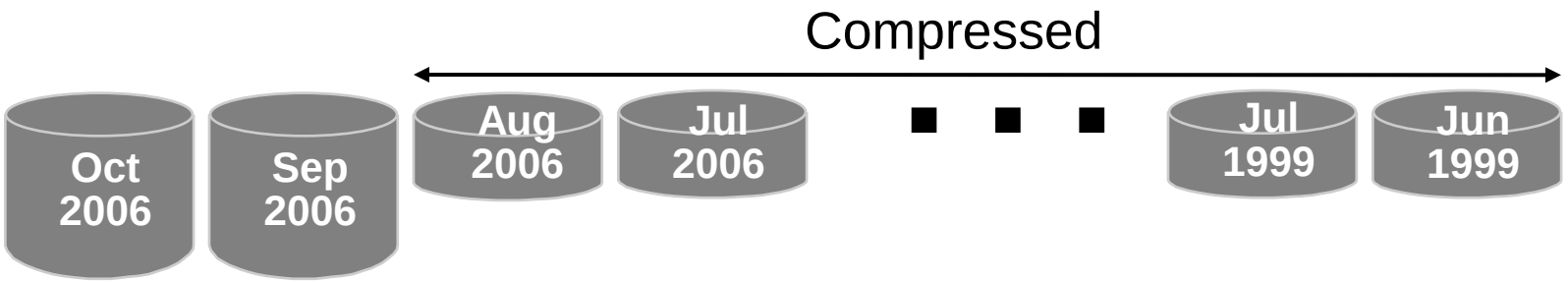
Read-only Data

Data Warehouse (reporting database)

- Monthly nature of data changes
  - E.g. sales and returns
  - E.g. call detail records

Information Lifecycle Management scenarios

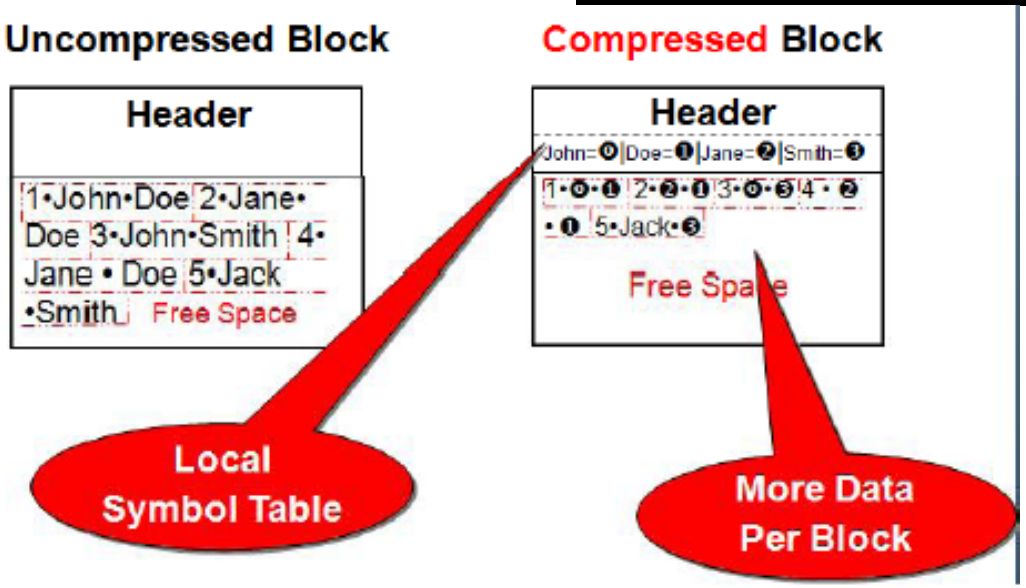
- Older data may not be subject to changes
- Remember compression by partition



How does Data Compression Work?

Let is look at the internal workings of data compression

ID	FIRST_NAME	LAST_NAME
1	John	Doe
2	Jane	Doe
3	John	Smith
4	Jane	Doe



The compressed block uses a symbol table and holds more data, leading to reduction in space usage by up to 5 times.

The compressed block uses a symbol table and holds more data, leading to reduction in space usage by up to 5 times.

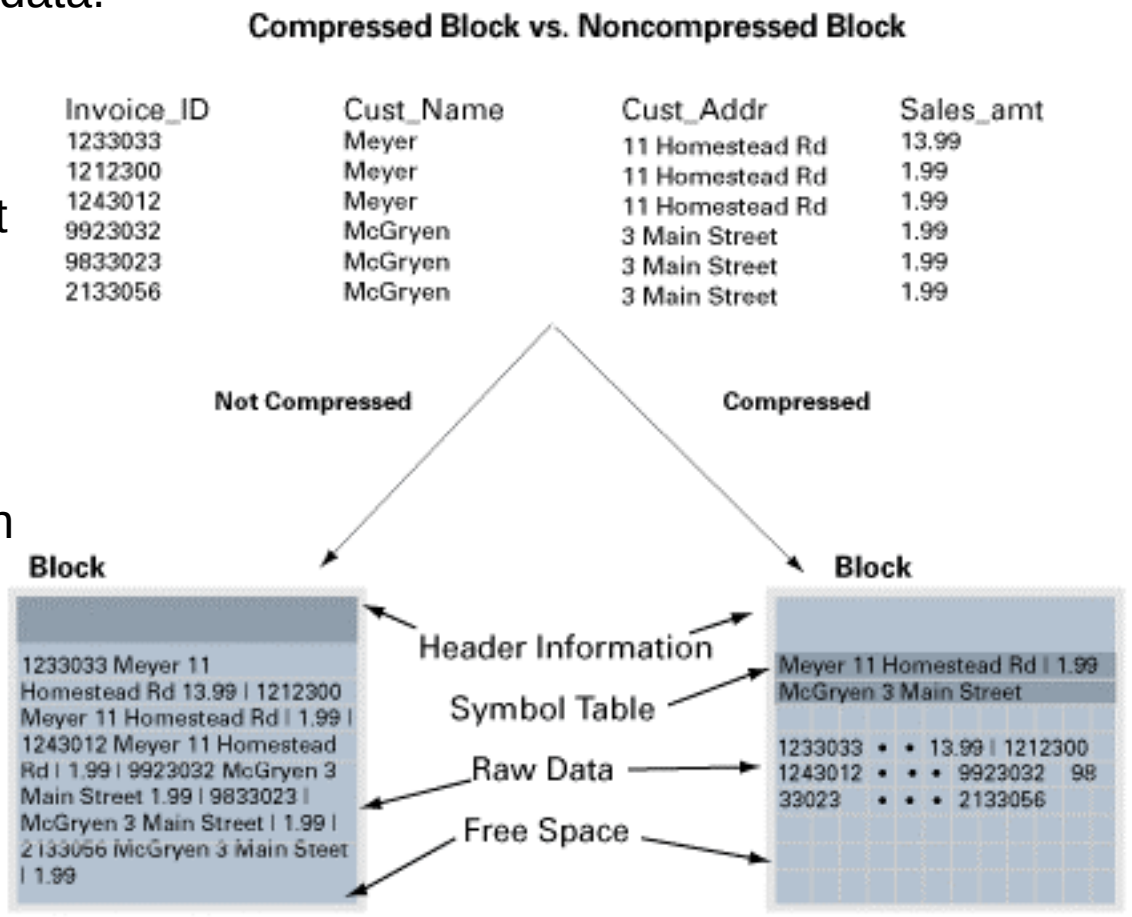
In other words…

•The algorithm works by eliminating duplicate values within a database block, even across multiple columns.

•Compressed blocks contain a structure called a symbol table that maintains compression metadata.

•When a block is compressed, duplicate values are eliminated by first adding a single copy of the duplicate value to the symbol table

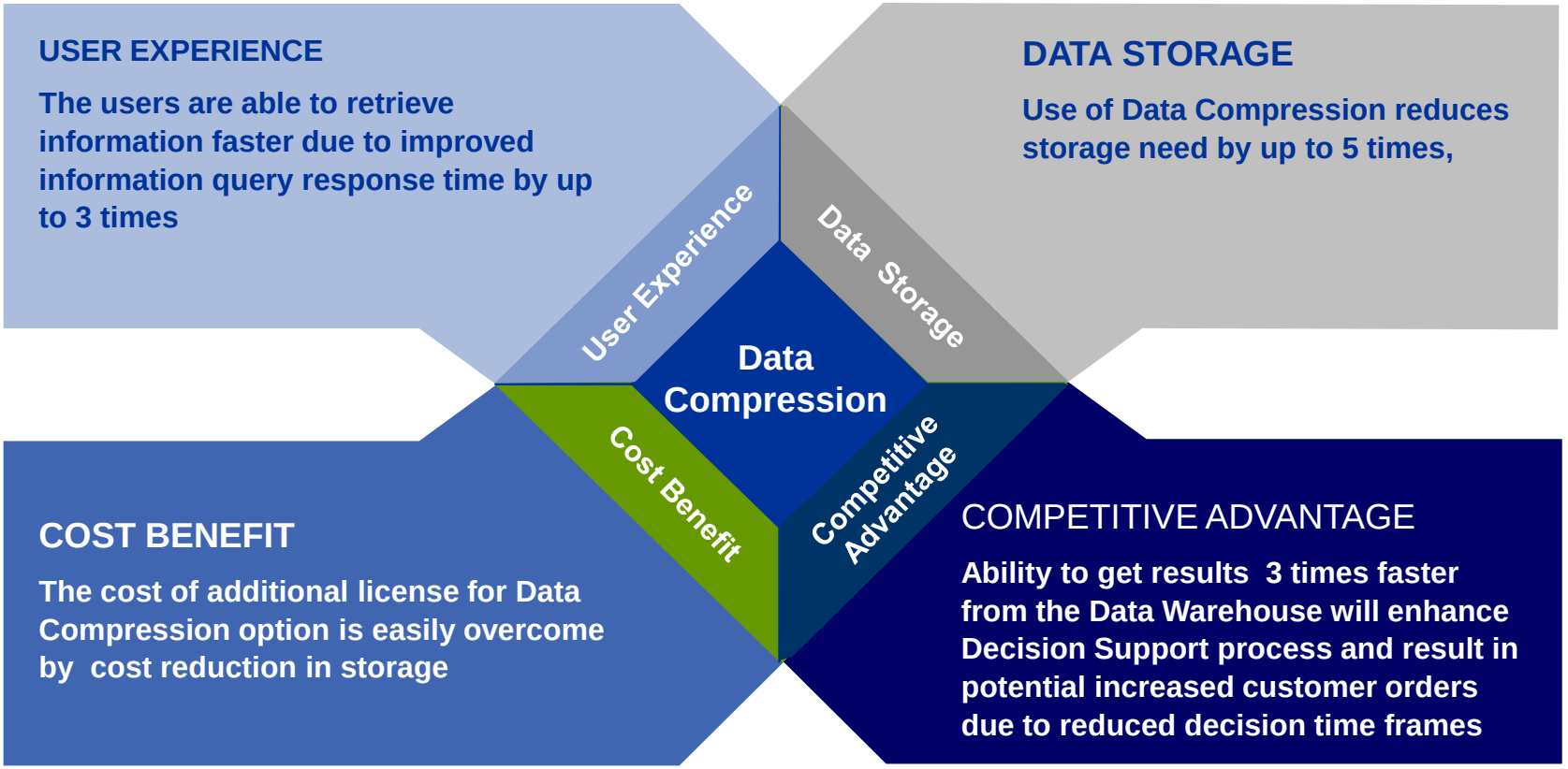
•Each duplicate value is then replaced by a short reference to the appropriate entry in the symbol table



Data warehouse compression solves the key problem at ABC Inc.

Data Compression has a four-fold role in an enterprise

- Enhanced user experience inside the organization
- Net cost benefit
- Handle data warehouse storage needs
- Create competitive advantage



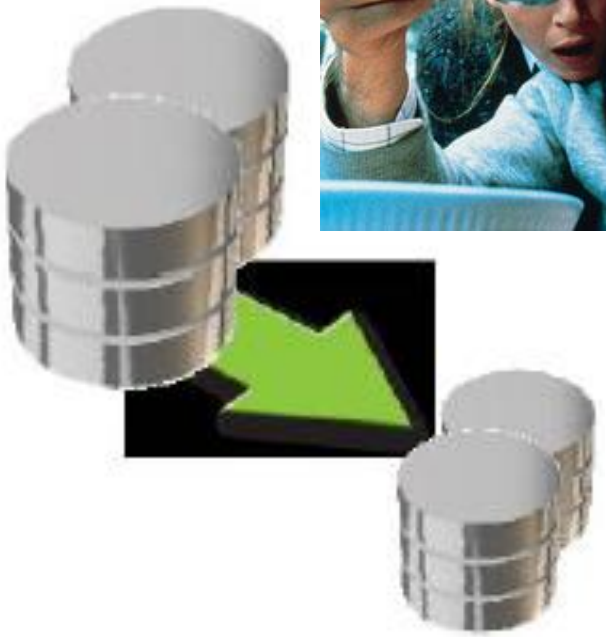
The Goal of Data Compression

Table Compression Syntax (OLTP):

CREATE TABLE emp (emp_id NUMBER, first_name VARCHAR2(128), last_name VARCHAR2(128))
COMPRESS FOR ALL OPERATIONS;

Direct Load Compression Syntax (default):

CREATE TABLE emp (emp_id NUMBER, first_name VARCHAR2(128), last_name VARCHAR2(128)
) ***COMPRESS [FOR DIRECT_LOAD OPERATIONS];***



Historical Perspective of Data Compression

Oracle 8.15 (Feb 1999)

- Index key compression: duplicate keys are combined to save space

Data Compression introduced in Oracle Database 9i Release 2

- Compression during bulk load operations (Direct Load, CTAS)
- Data modified using conventional DML not compressed

LOB compression (utl_compress) in Oracle 10g

Oracle 11g – Advanced Compression

Advanced Compression

Comprehensive set of data compression capabilities

- Structured, unstructured, backup, network transport

- Reduces storage consumption by 2 – 3 times
  - Savings cascades into test, QA, backup, DR environments
- Improves application performance
- Enhances memory and network efficiency
- Complete application transparency
- Benefits diverse application workloads
  - Transaction Processing
  - Data Warehousing
  - Content Management

Syntax for Compression

COMPRESS keyword works for tables, table partitions and entire tablespaces, within the "*create table*" DDL:

```
create table NYOUG (col1 number) NOCOMPRESS;
create table NYOUG (col1 number) COMPRESS FOR DIRECT_LOAD
OPERATIONS;
create table NYOUG (col1 number) COMPRESS FOR ALL OPERATIONS;
```

CREATE TABLESPACE MYSTUFF . . . DEFAULT{ COMPRESS [FOR { ALL | DIRECT_LOAD } OPERATIONS] | NOCOMPRESS }

Several DBA views have been enhanced in 11g to show compression attributes.
The *dba_tables* view has added the new columns COMPRESSED (enabled, disabled) and
COMPRESSED_FOR (nocompress, compress for direct_load operations, compress for all operations.

OLTP Table Compression

Oracle Database 11g extends table compression for OLTP data

- Support for conventional DML Operations

  (INSERT, UPDATE)

New algorithm significantly reduces write overhead

- Batched compression ensures no impact for most OLTP transactions

No impact on reads

- Reads may actually see improved performance due to fewer IOs and enhanced memory efficiency

Features of Compression

- Specify table compression with the COMPRESS clause of the CREATE TABLE statement. You can enable compression for an existing table by using this clause in an ALTER TABLE statement.
- In this case, the only data that is compressed is the data inserted or updated after compression is enabled.
- Similarly, you can disable table compression for an existing compressed table with the ALTER TABLE...NOCOMPRESS statement. In this case, all data the was already compressed remains compressed, and new data is inserted uncompressed.
- To enable compression for conventional DML, you must set the COMPATIBLE initialization parameter to 11.1.0 or higher.
- The keyword COMPRESS by itself is the same as the clause COMPRESS FOR DIRECT_LOAD OPERATIONS, and invokes the same compression behavior as previous database releases.

Compression - Features

- **Adding and Dropping Columns in Compressed Tables**
  - When you enable compression for all operations on a table, you can add and drop table columns.
  - If you enable compression for direct-path inserts only, you cannot drop columns, and you can add columns only if you do not specify default values.
- **Compression and Partitioned Tables**
  - You can enable or disable compression at the partition level. You can therefore have a table with both compressed and uncompressed partitions.
  - If the compression settings for a table and one of its partitions disagree, the partition setting has precedence for the partition.

```
SQL> SELECT table_name, compression, compress_for FROM user_tables;

TABLE_NAME     COMPRESS     COMPRESS_FOR
---------------  --------  ------------------
T1             DISABLED
T2             ENABLED      DIRECT LOAD ONLY
T3             ENABLED      FOR ALL OPERATIONS
```

Partitioned and Compressed Table

Here all partitions except the northeast partition are compressed.

CREATE TABLE sales

(saleskey number, quarter number, product number, salesperson number,

amount number(12, 2), region varchar2(10)) **COMPRESS**

PARTITION BY LIST (region)

(PARTITION northwest VALUES ('NORTHWEST'),

PARTITION southwest VALUES ('SOUTHWEST'),

PARTITION northeast VALUES ('NORTHEAST') **NOCOMPRESS**,

PARTITION southeast VALUES ('SOUTHEAST'),

PARTITION western VALUES ('WESTERN'));

…continued

- The following simple statement parallelizes the creation of a table and stores the result in a compressed format, using table compression:

  CREATE TABLE hr.admin_emp_dept PARALLEL **COMPRESS**

  AS SELECT * FROM hr.employees WHERE department_id = 10;

- You can also use the ALTER TABLE...MOVE statement with a COMPRESS clause to store the new segment using table compression.

- Compressing the index

  CREATE INDEX cust_dup_idx ON customer(sex, hair_color, customer_id) PARALLEL NOLOGGING **COMPRESS 2;**

- Key Compression for index organized table, you cannot specify COMPRESS for a bitmap index

  CREATE TABLE admin_iot5(i INT, j INT, k INT, l INT, PRIMARY KEY (i, j, k)) ORGANIZATION INDEX **COMPRESS**;

Improve Performance – Reduce IO

Query large table (500 GB) – full table scan

Throughput 1 GB/s

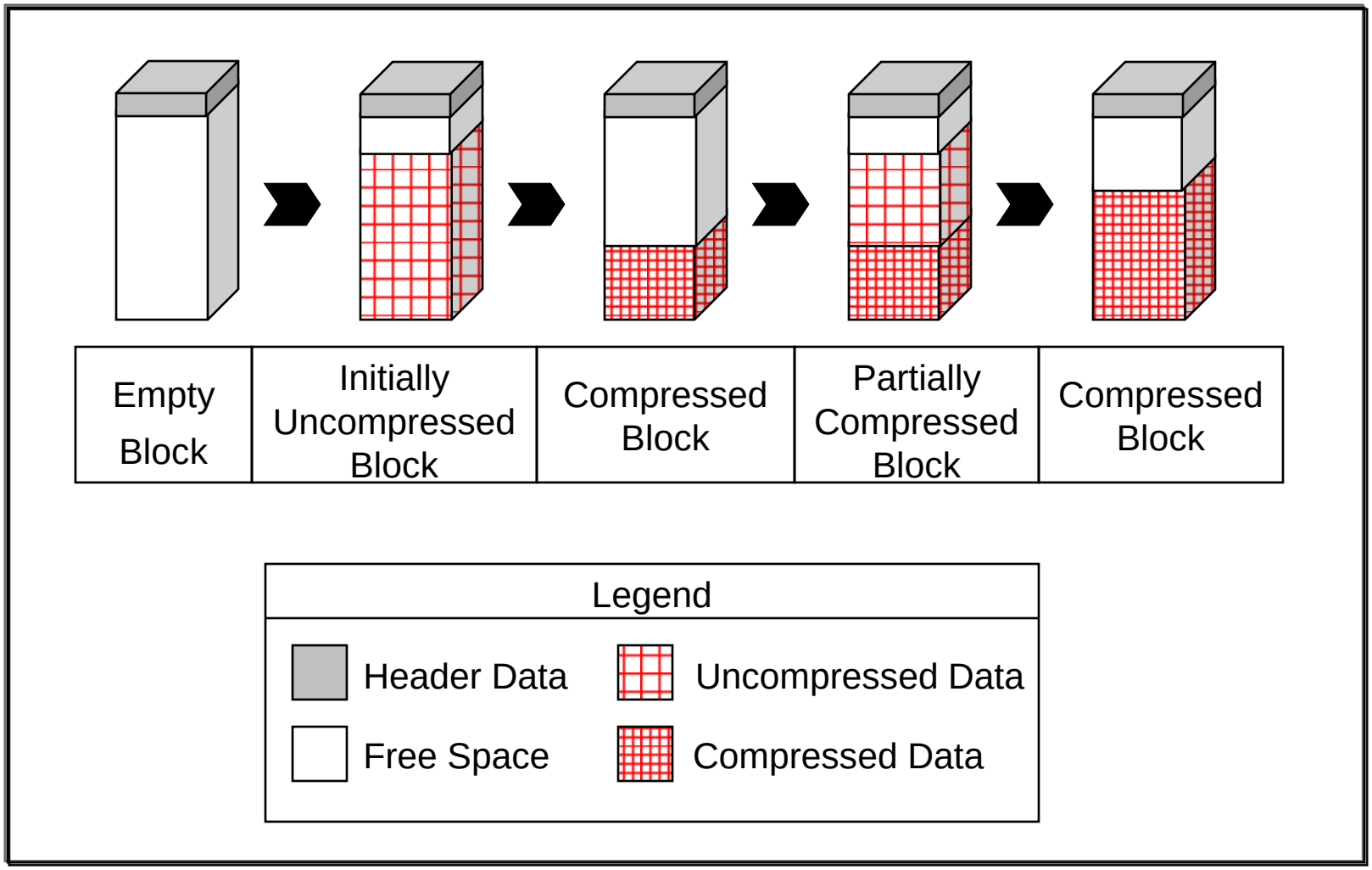
- 8 minutes, 20 seconds to read table
- Compression 2:1 – 4 minutes, 10 seconds to read table

IO-bound operations can get better performance

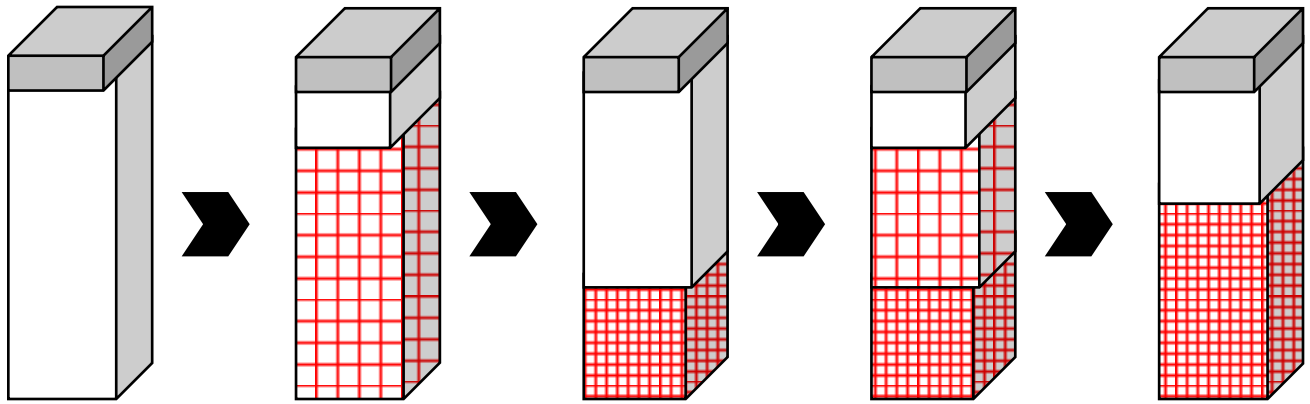
- Assume the CPU cost is less than the IO performance gain

Find the bottleneck in your system

OLTP Table Compression Process



Block-Level *Batch* Compression



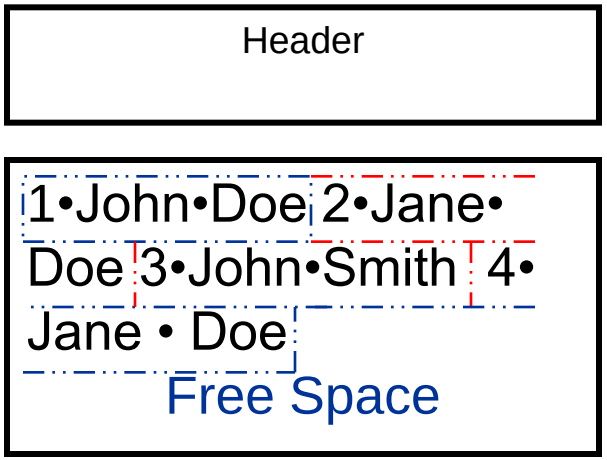
- Individual INSERT and UPDATEs do not cause recompression
- Compression cost is spread over several DML operations
- Block-level (Local) compression keeps up with frequent data changes in OLTP environments
- Compression utilities such as GZIP and BZ2 use similar adaptive, block level compression

OLTP Table Compression

Employee Table

ID	FIRST_NAME	LAST_NAME
1	John	Doe
2	Jane	Doe
3	John	Smith
4	Jane	Doe

Initially Uncompressed Block




```
INSERT INTO EMPLOYEE
    VALUES (5, 'Jack', 'Smith');
COMMIT;
```

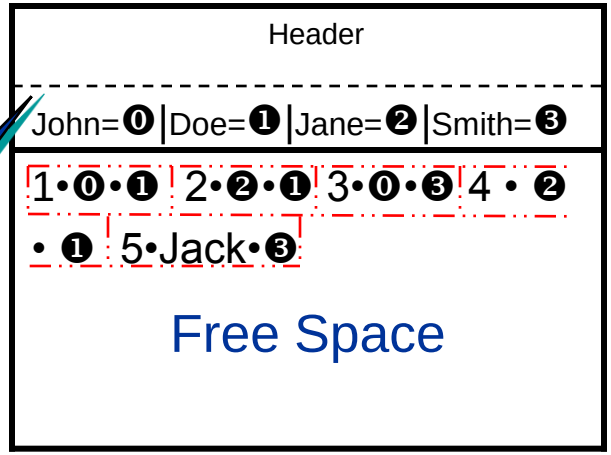
OLTP Table Compression

Employee Table

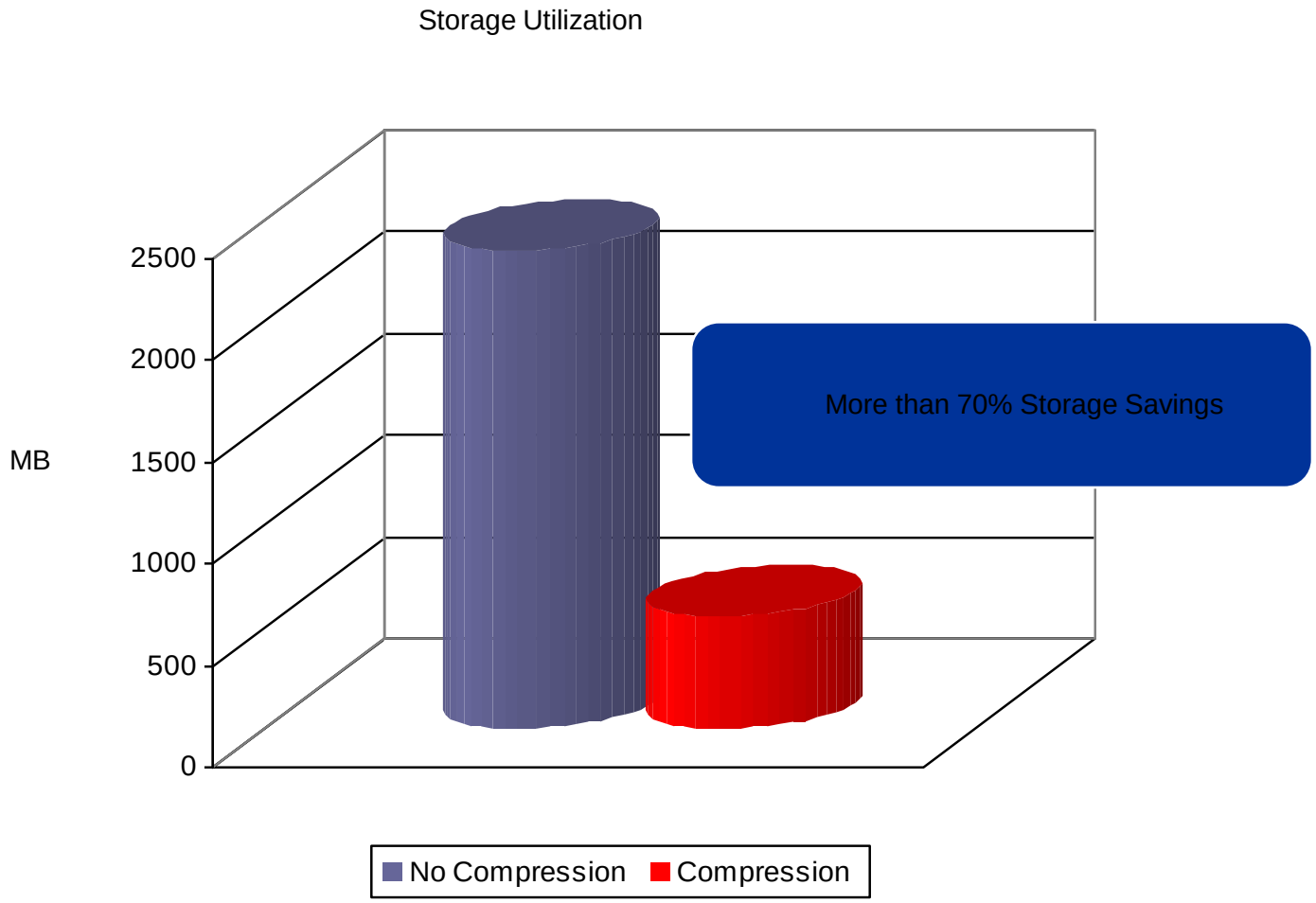
ID	FIRST_NAME	LAST_NAME
1	John	Doe
2	Jane	Doe
3	John	Smith
4	Jane	Doe
5	Jack	S

Local
Symbol Table

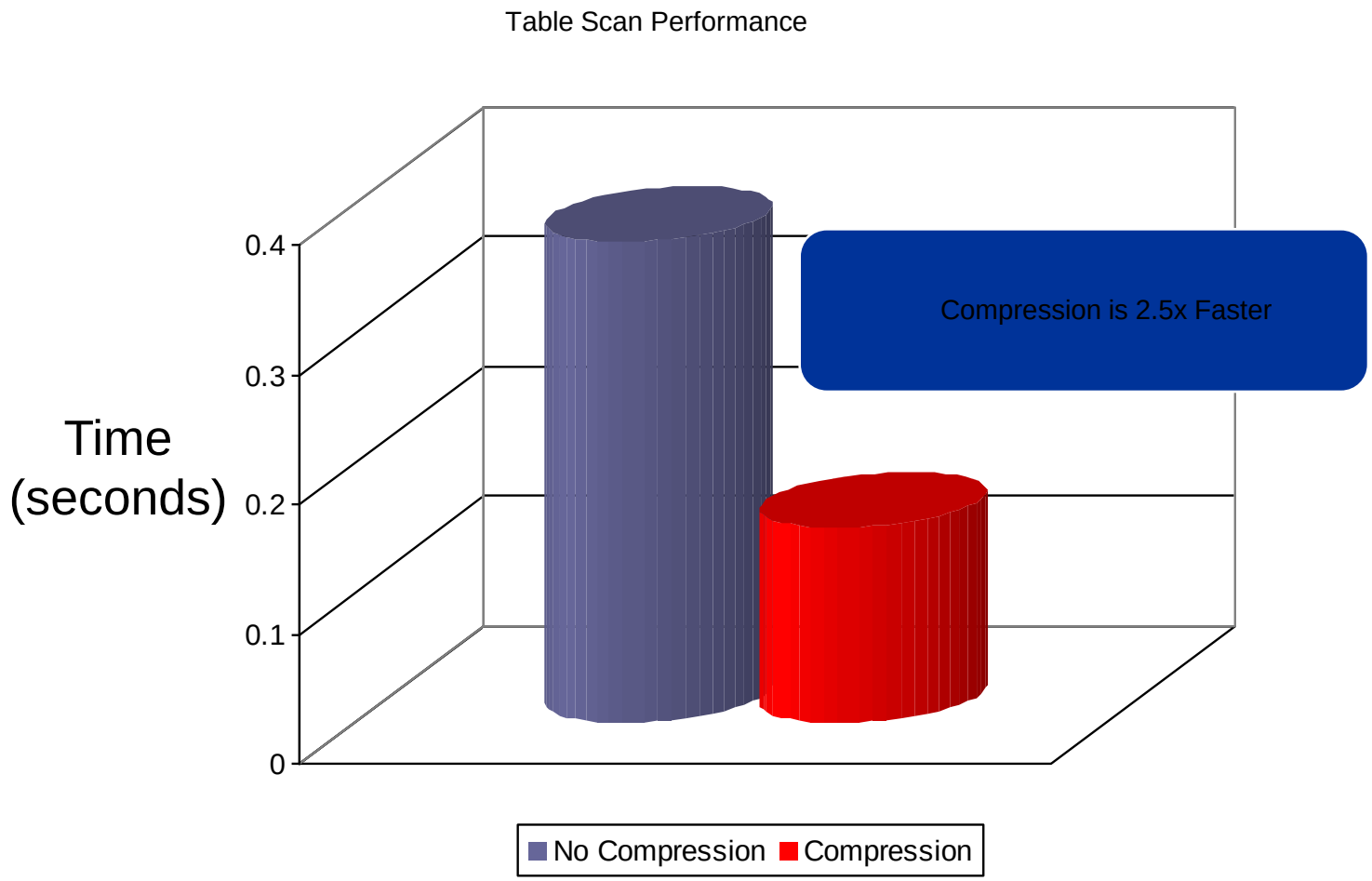
Compressed Block



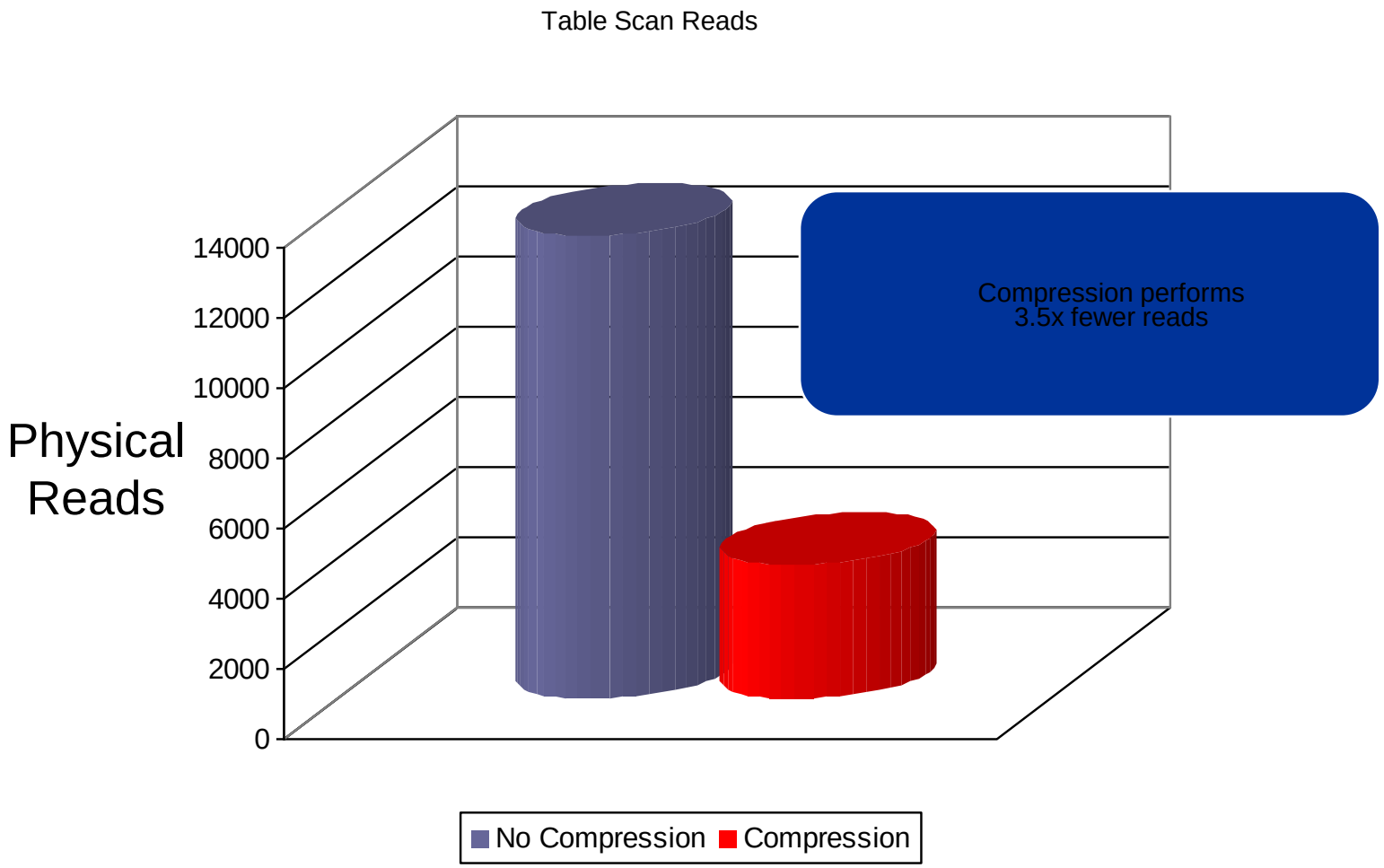
Real World Compression Results



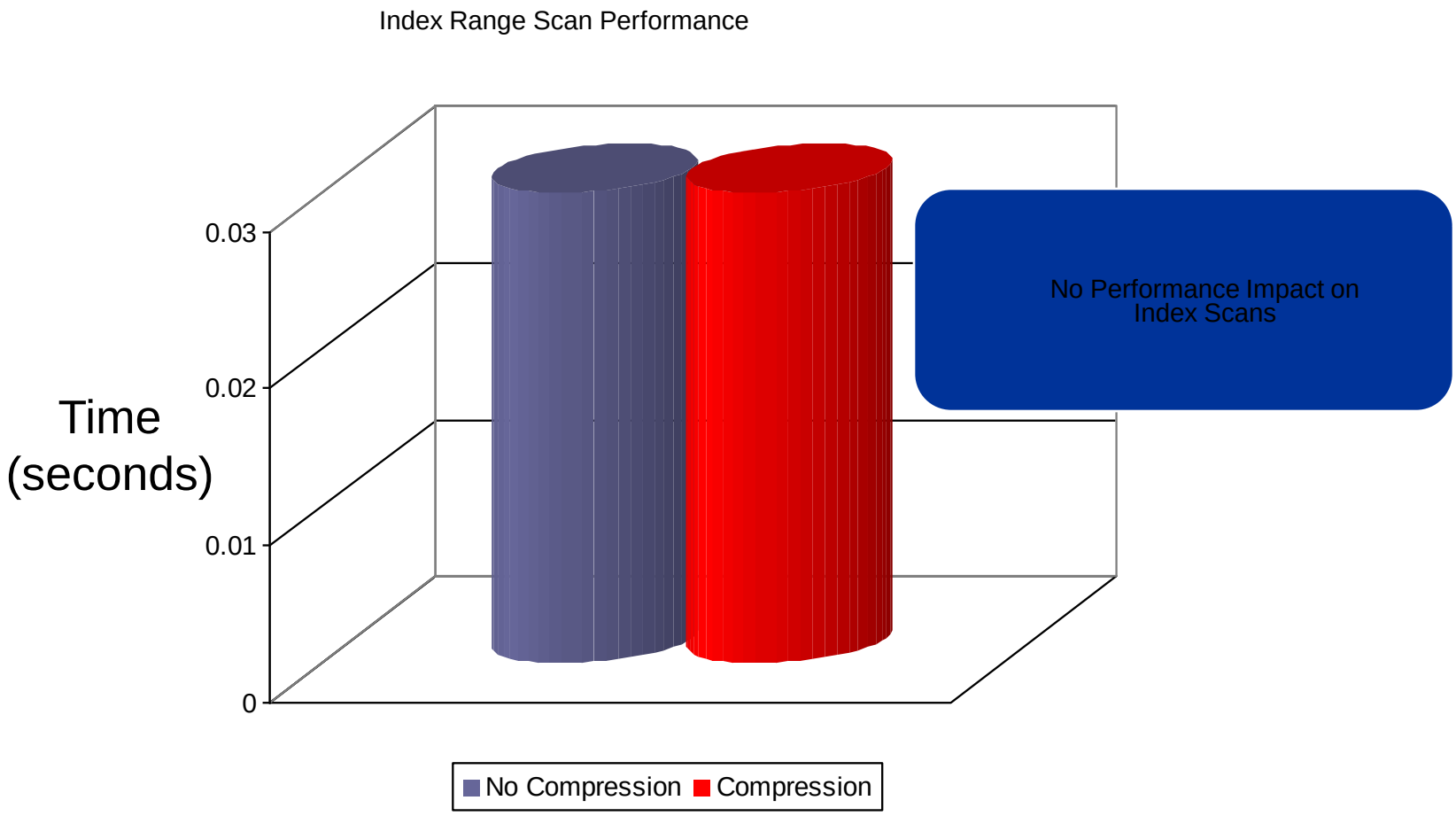
Real World Compression Results



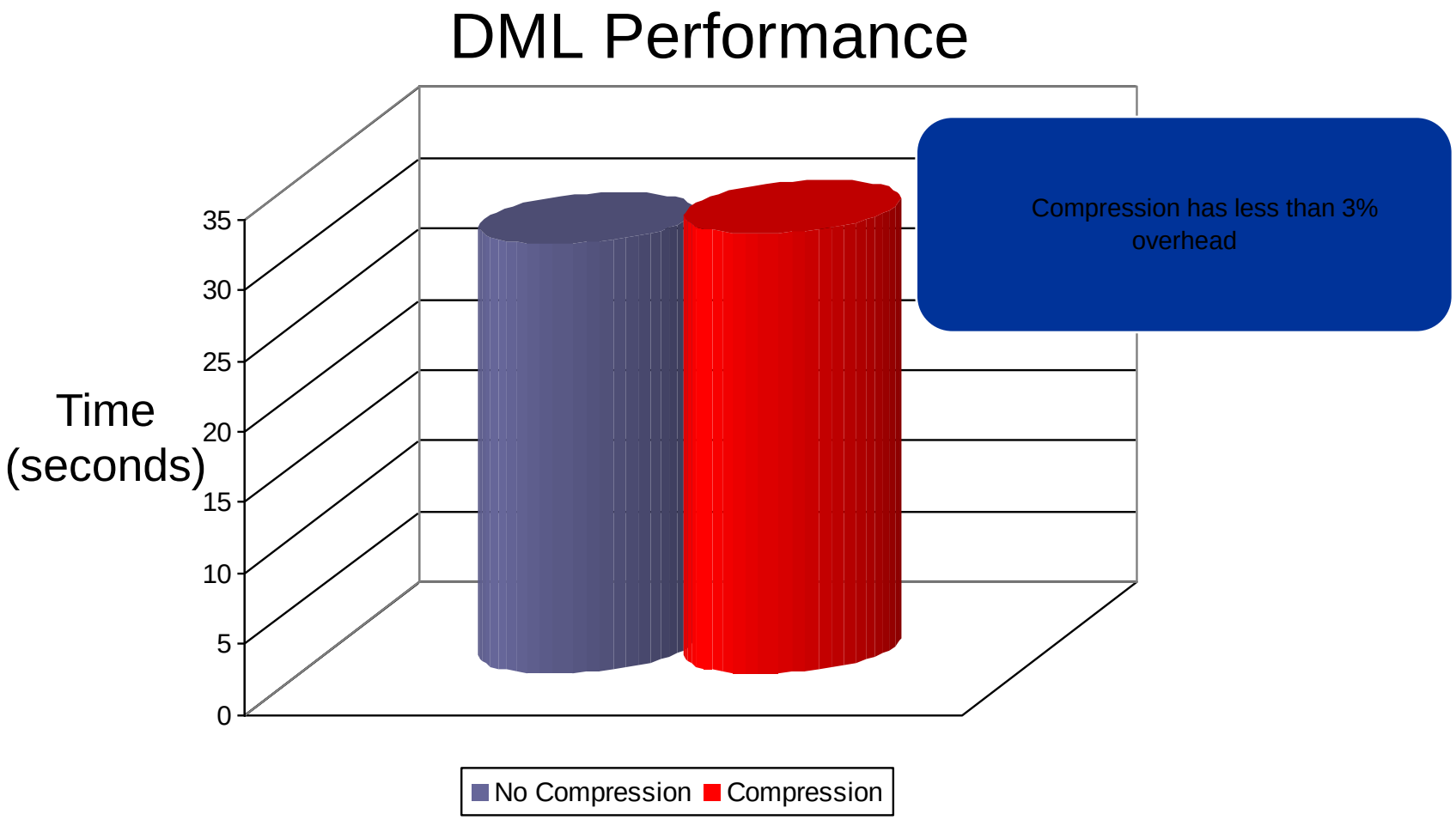
Real World Compression Results



Real World Compression Results



Real World Compression Results



Best Practices

Compress the top few largest tables only

80/20 Rule – 20% of your Tables Consume 80% of your Space

Better Compression with Bigger Block Sizes

Higher Probability of Duplicate Values

B-Tree Index Compression

Validate Index and Review INDEX_STATS

– INDEX_STATS.OPT_CMPR_COUNT

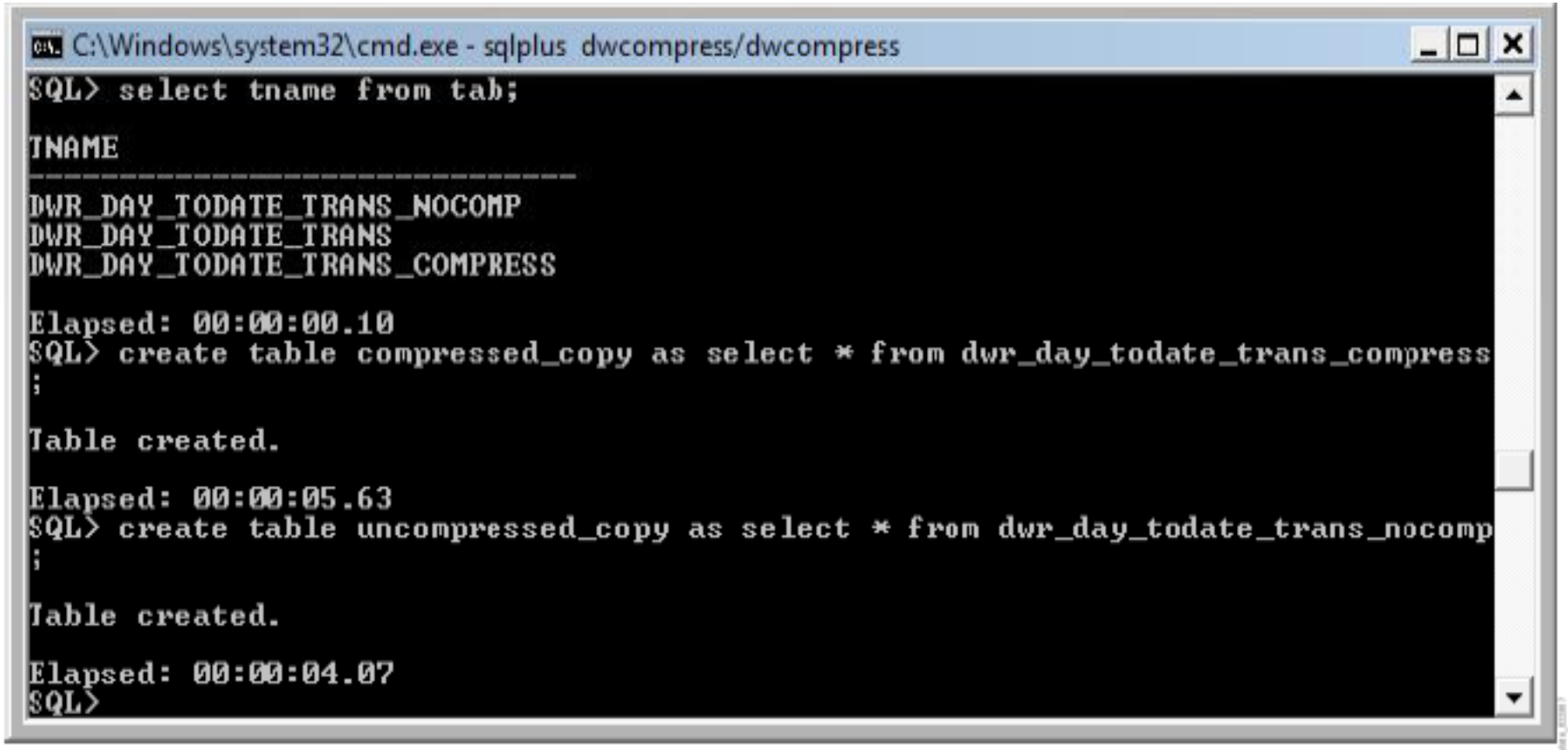
– INDEX_STATS.OPT_CMPR_PCTSAVE

Bitmap Indexes are Highly Compressed

Good for low and medium cardinality

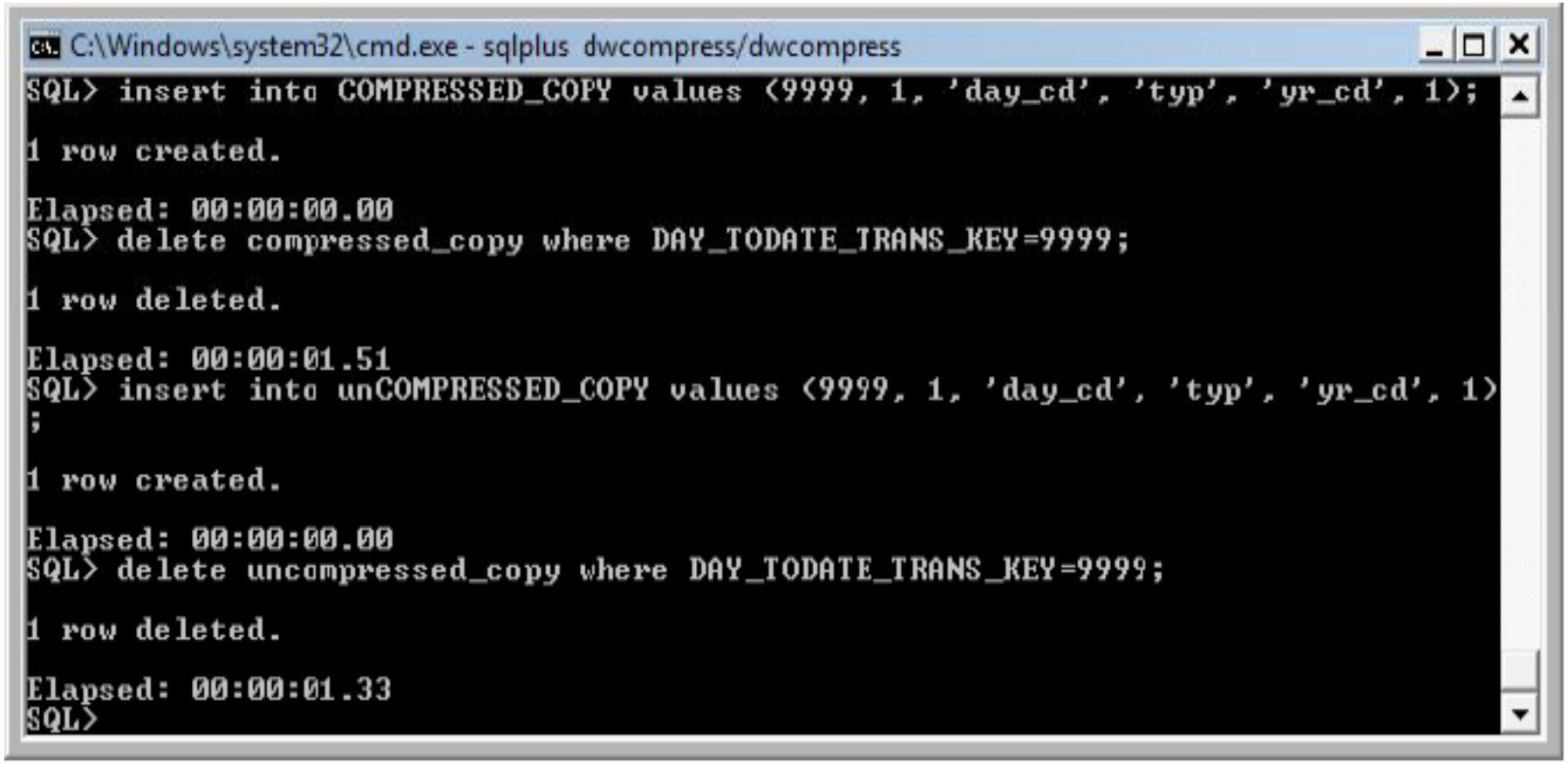
Hands on Testing 10 g

Used CTAS with compressed and uncompressed table as source



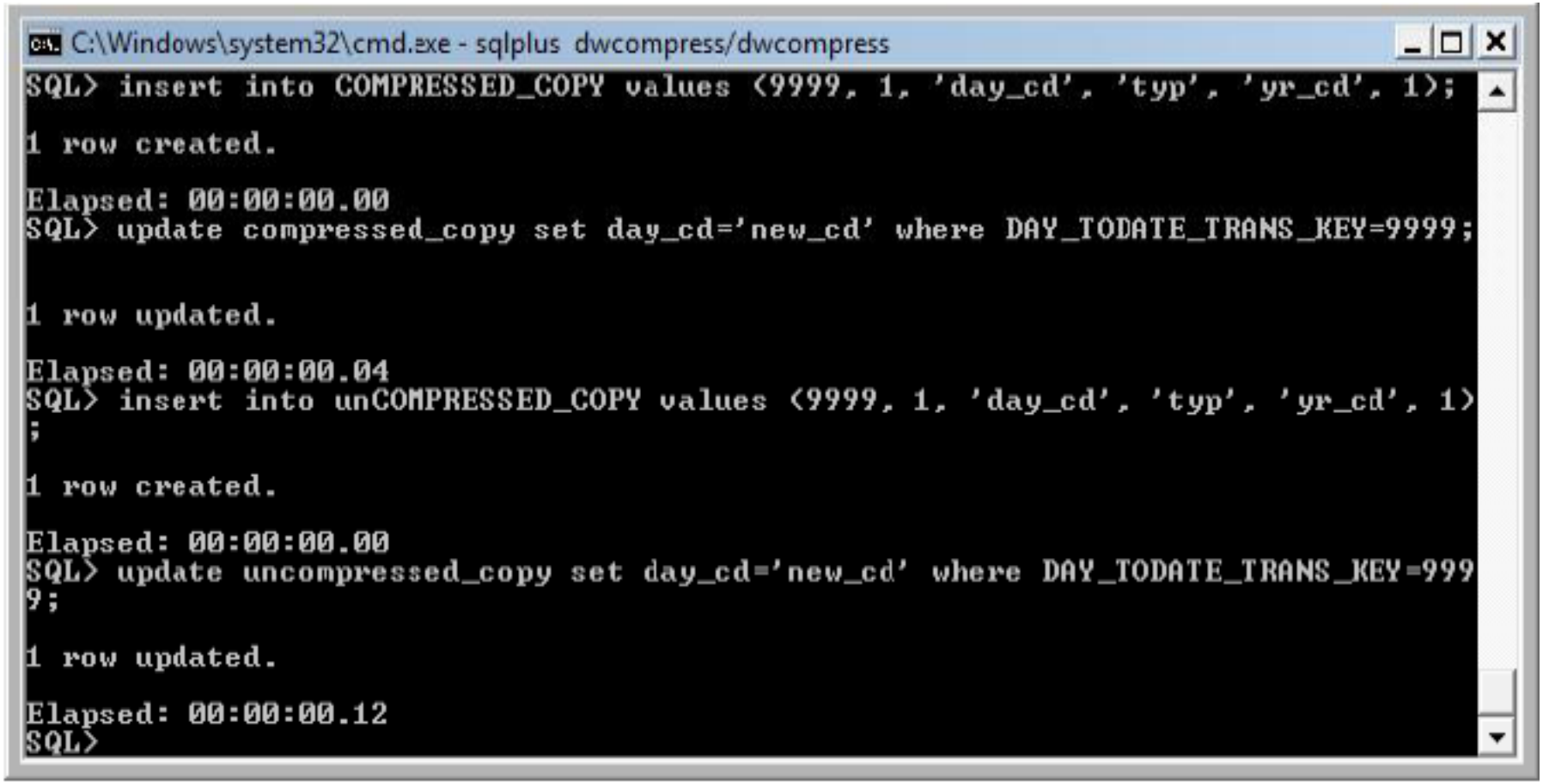
Hands on Testing 10 g

Insert / Delete Performance test



Hands on Testing 10 g

Update Performance test



Advanced Compression Advisor

Compression Advisor estimates potential storage savings using OLTP Table Compression feature
This estimate is based on analysis of a sample of data and provides a good estimate

Using Compression Advisor - **DBMS_Compression** package containing the following procedure:

```
getratio(
   ownername          IN     varchar2,
   tabname            IN     varchar2,
   sampling_percent   IN     number
 );
where
```

- 'ownername' is the schema that the table belongs to
- 'tabname' is name of the table for which compression ratio is to be estimated
- 'sampling_percent' is any value between 0.000001 and 100

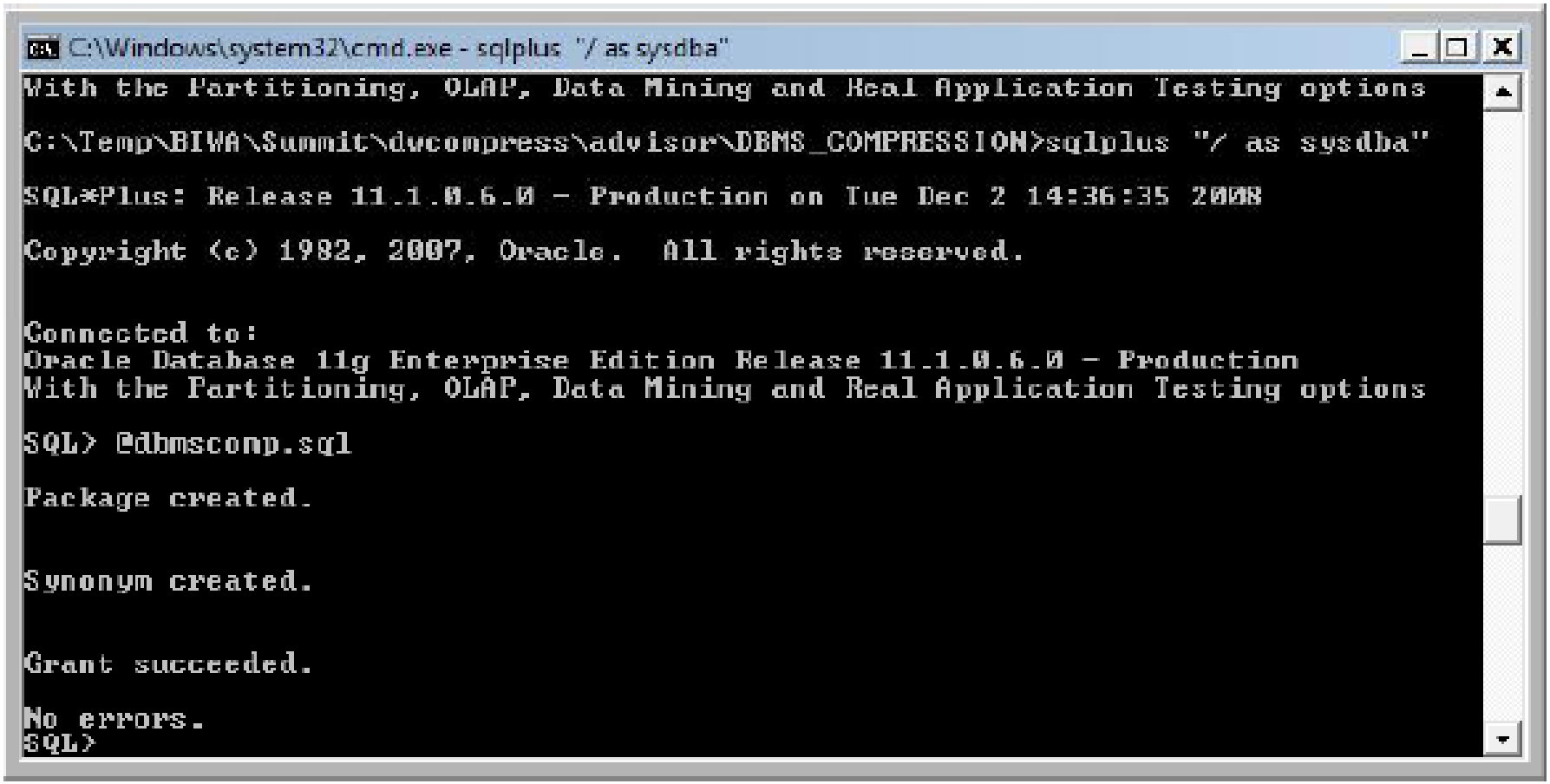
The output of this procedure is the estimated compression ratio.

```
SQL>  exec DBMS_COMPRESSION.getratio('SH','SALES',10);

Sampling table: SH.SALES
Sampling percentage: 10%
Compression ratio: 2.99
PL/SQL procedure successfully completed.
```

Installing Advanced Compression Advisor

Download and install it by running the SQL package



Real World Testing of Compression (sean_mccown@infoworld.com)

- Note that the two tables with 0% compressed (C_District and C_Item) are lookup tables; they have no duplicate data, so wouldn't be expected to benefit from Advanced Compression.

•The TPC-C data reflects the operations of a company with 4,000 warehouses that supply a total of 40,000 districts (10 per warehouse) and 120 million customers (3,000 per district).
•The company sells 100,000 different items, and each warehouse maintains stocks for each item.
•The C_New_Order table I generated contained 36 million rows, and the C_Order_Line and C_Stock tables had 1,578,942,568 and 186,338,813 rows, respectively.

Table Name	Original Size (MB)	Compressed Size (MB)	% of Original	% Compressed
C_District	5	5	100	0
C_Item	11	11	100	0
C_New_Order	624	448	72	28
C_Customer	93,882	78,249	83	17
C_History	7,680	3,327	43	**57**
C_Order_Line	104,528	72,703	70	30
C_Stock	66,304	56,065	85	15

Ref: http://weblog.infoworld.com/tcdaily/archives/2008/06/oracle_advanced.html

Relational Table Compression Vendor T

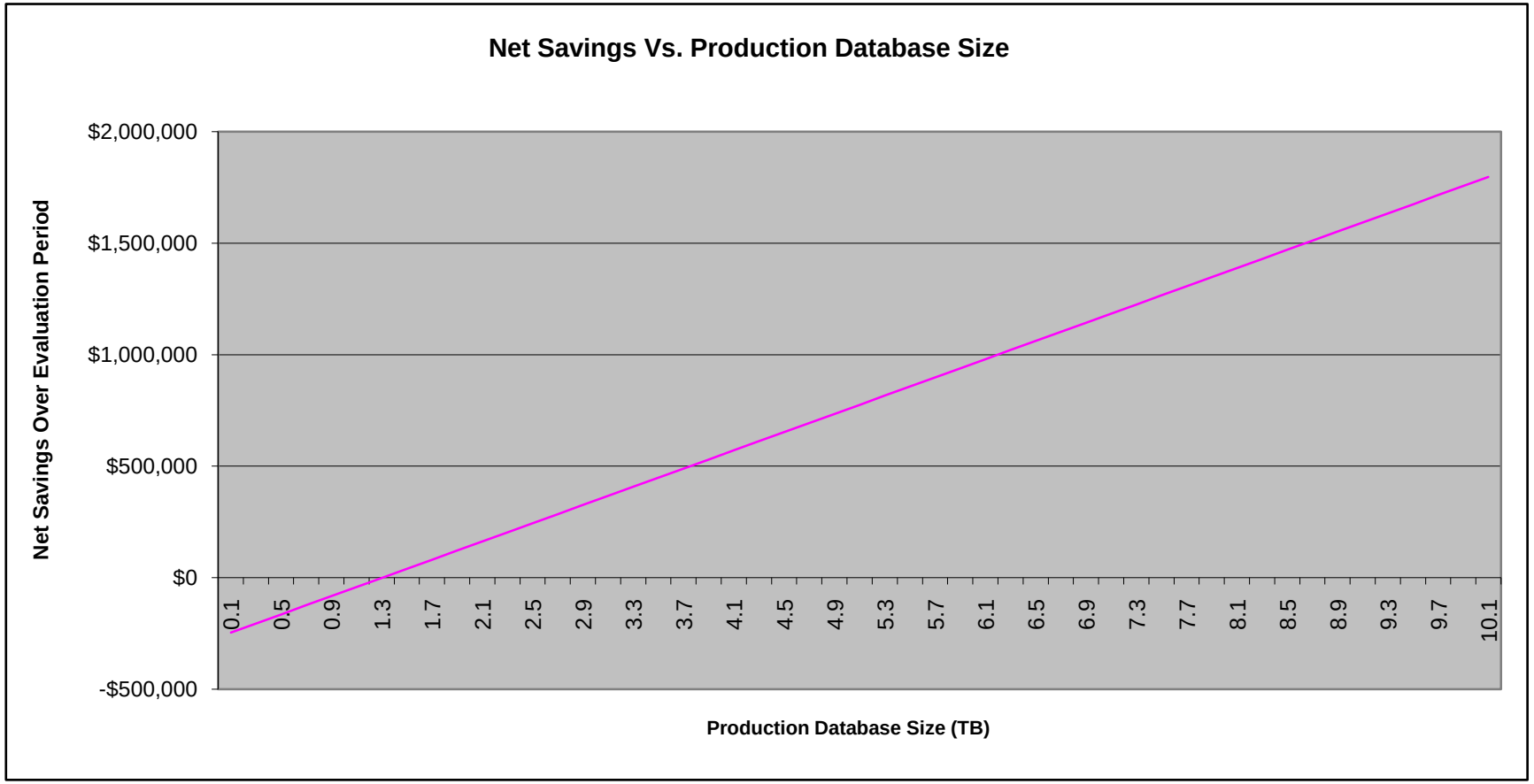
Advantages

- Efficient – decompression is very cheap

Disadvantages

- Requires advance data knowledge
  - Re-compress implies recreate table – DDL becomes ugly
- Real world has more than 255 repetitive values
  - E.g. ORDER_DATE
- Hard to tune for optimal compression
  - Choose how many bits to use
  - Take into account CHAR vs. VARCHAR
  - May require several iterations
- No multi-column compression

The Pay-Off for Data Compression



Index Compression

B-tree indexes

- Can be compressed
- May have prefix columns
  - Compressing a unique index makes no sense

Bitmap indexes

- Naturally compressed bitmap structure

Introduction to SecureFiles

SecureFiles is a new 11g feature designed to break the performance barrier keeping file data out of databases

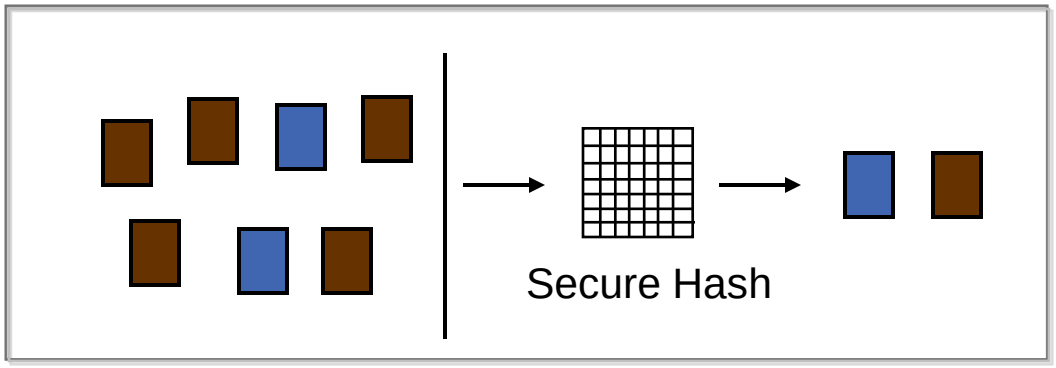
Next-generation LOB

- Superset of LOB interfaces allows easy migration from LOBs
- Transparent deduplication, compression, and encryption
- Leverage the security, reliability, and scalability of database

Enables consolidation of file data with associated relational data

- Single security model
- Single view of data
- Single management of data
- Scalable to any level using SMP scale-up or grid scale-out

SecureFiles Deduplication



SecureFiles is a new 11g feature designed to break the performance barrier keeping file data out of databases

- Enables storage of a single physical image for duplicate data
- Significantly reduces space consumption
- Dramatically improves writes and copy operations
- No adverse impact on read operations
- May actually improve read performance for cache data

Data Pump Compression

Performance cost: ~10% overhead

Compression Ratio: comparable to gzip

Compression Method	OE/SH Schemas	Spatial Table	Spatial Schema
exdp compression=none	6.0 MB	26.6 MB	443 MB
expdp compression=all	1.5 MB (74.7%)	9.9 MB (62.7%)	140 MB (68.4%)
gzip –cv1	1.1 MB (82.7%)	11.4 MB (57.1%)	162 MB (63.4%)
gzip –cv6	835 KB (86.2%)	10.2 MB (61.7%)	142 MB (68.0%)
gzip –cv9	818 KB (86.5%)	10.1 MB (62.0%)	141 MB (68.2%)
compress	1.6 MB (74.2%)	13.8 MB (48.1%)	198 MB (55.3%)

Backup Compression

Fast RMAN Compression

- Compresses the backup set contents before writing them to disk or tape
- No extra decompression steps are required during recovery when you use RMAN compression
- High performance, industry standard compression algorithm
  - 40% faster backup compression versus Oracle Database 10g
- Suitable for fast, incremental daily backups
- Reduces network usage

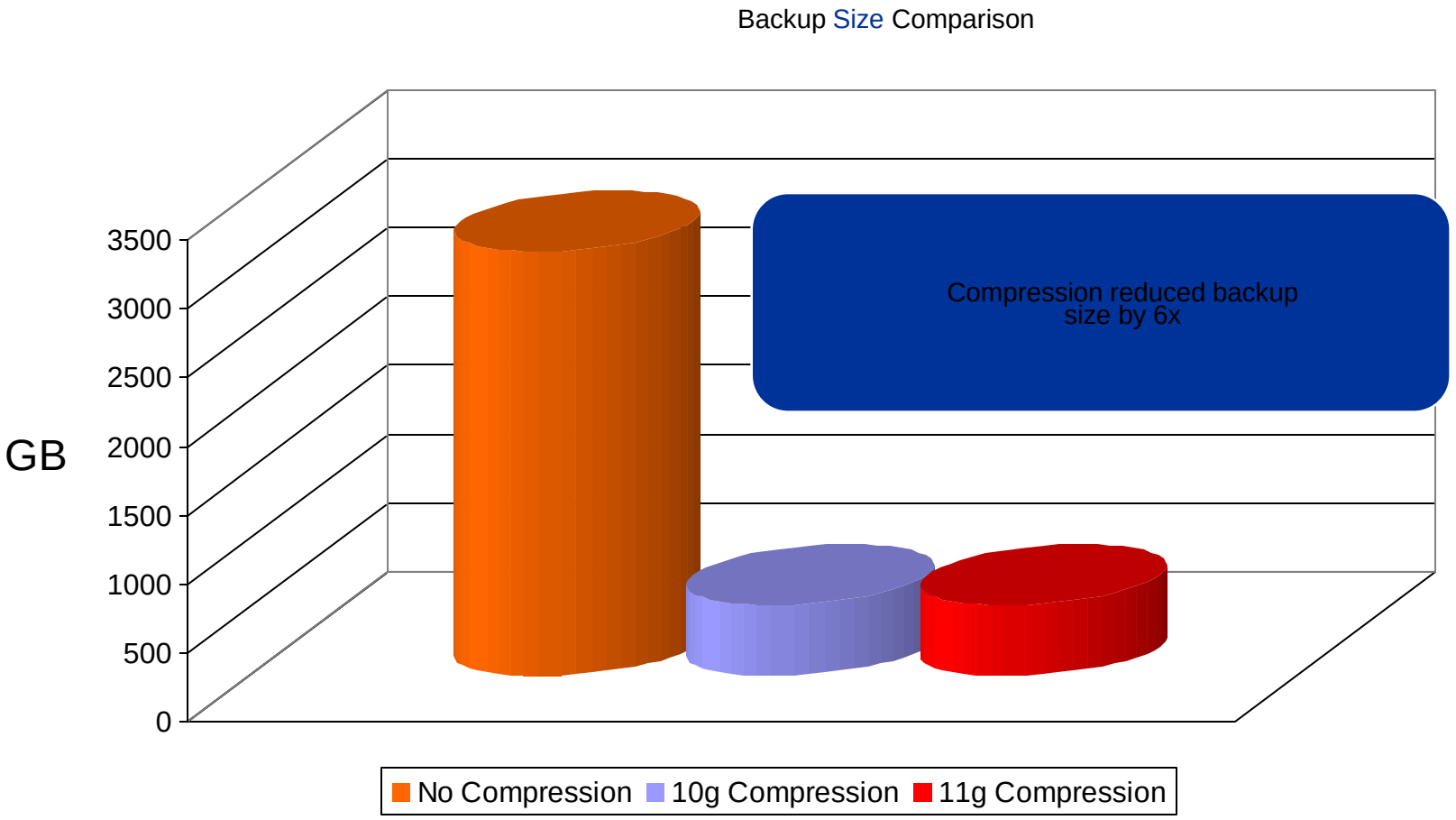
RMAN Compression Syntax

*RMAN> backup **as COMPRESSED BACKUPSET** database*
archivelog all

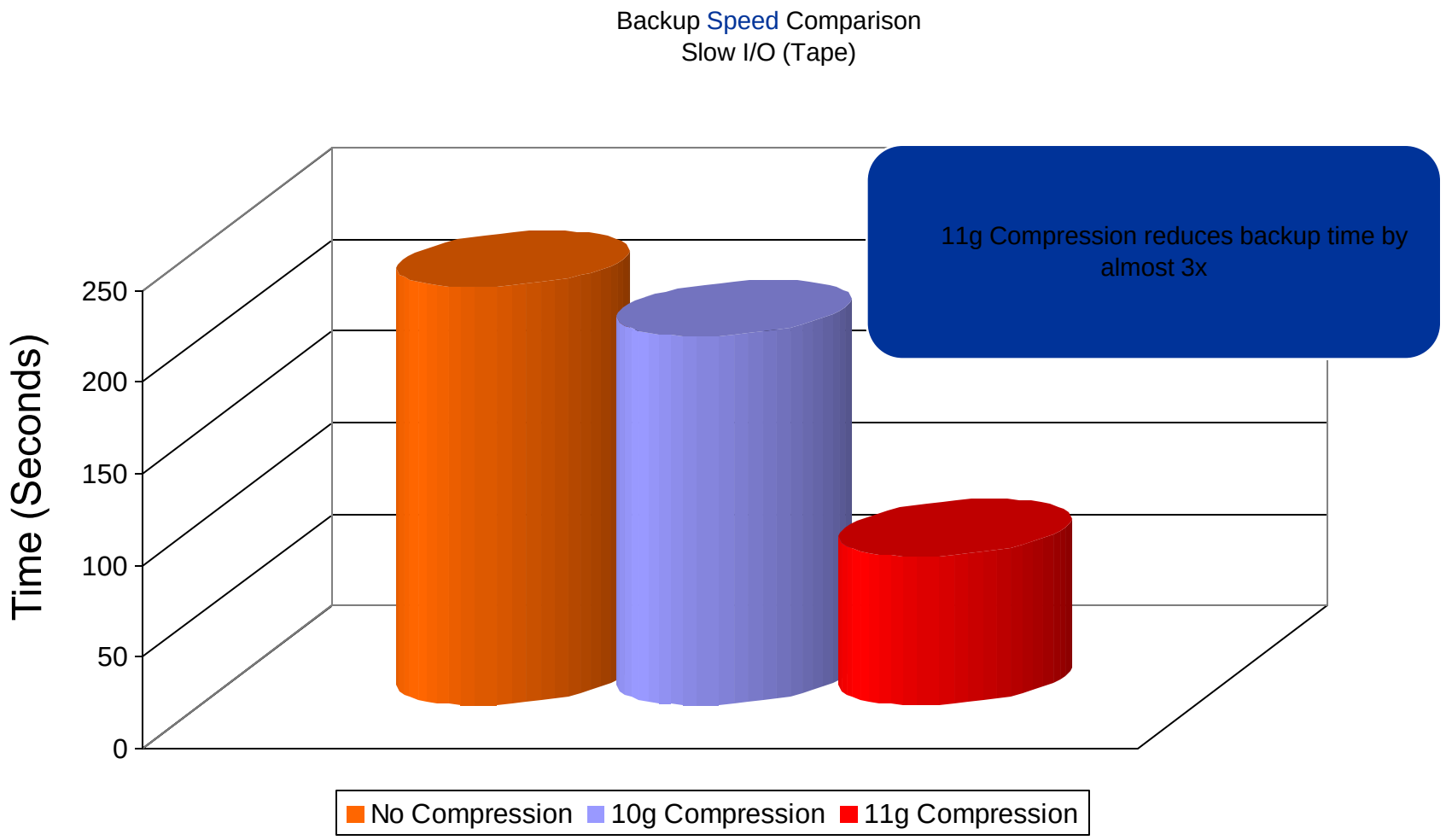
DataPump Syntax

PROMPT> expdp hr FULL=y DUMPFILE=dpump_dir:full.dmp
COMPRESS

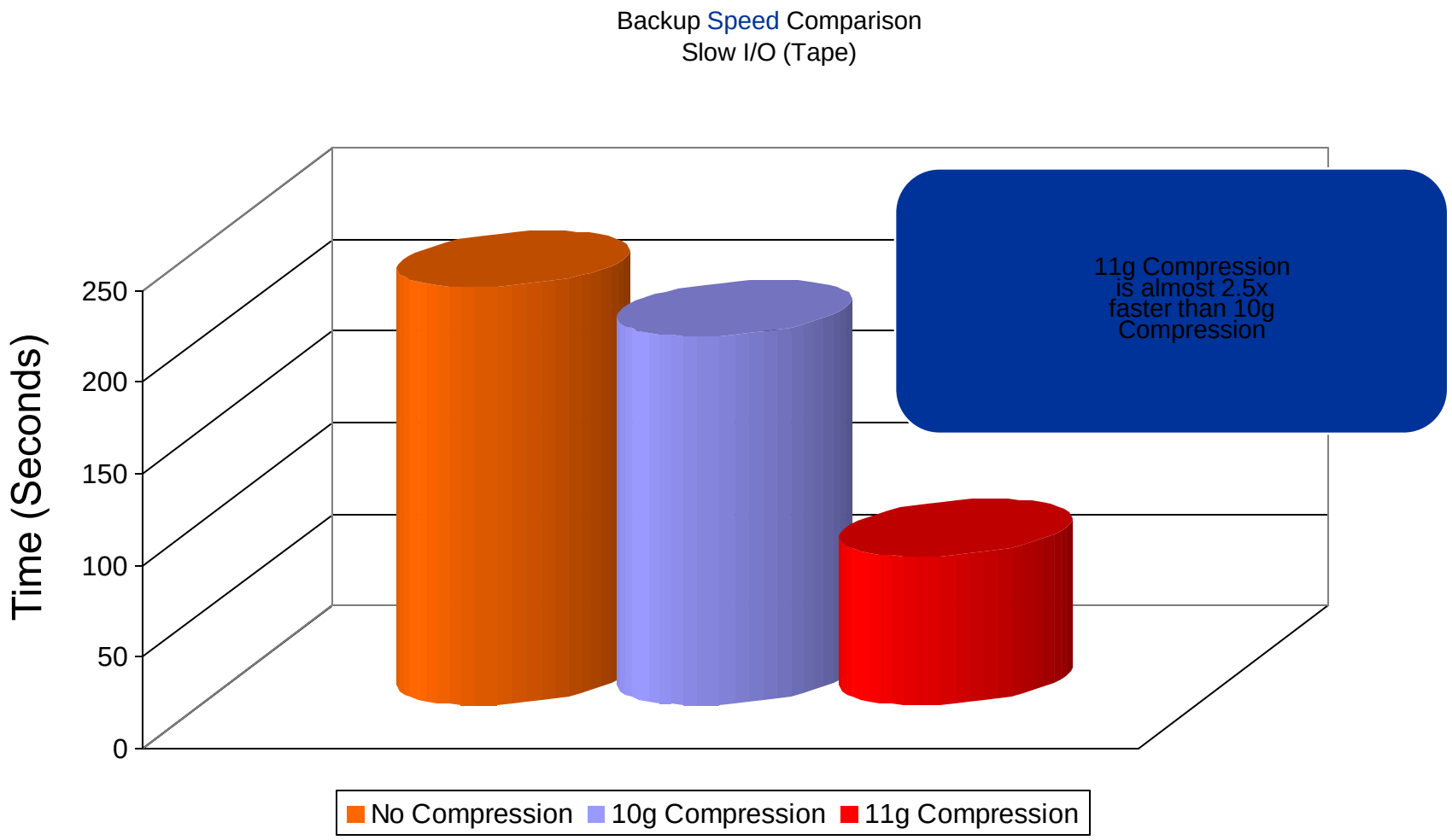
Backup Compression Results



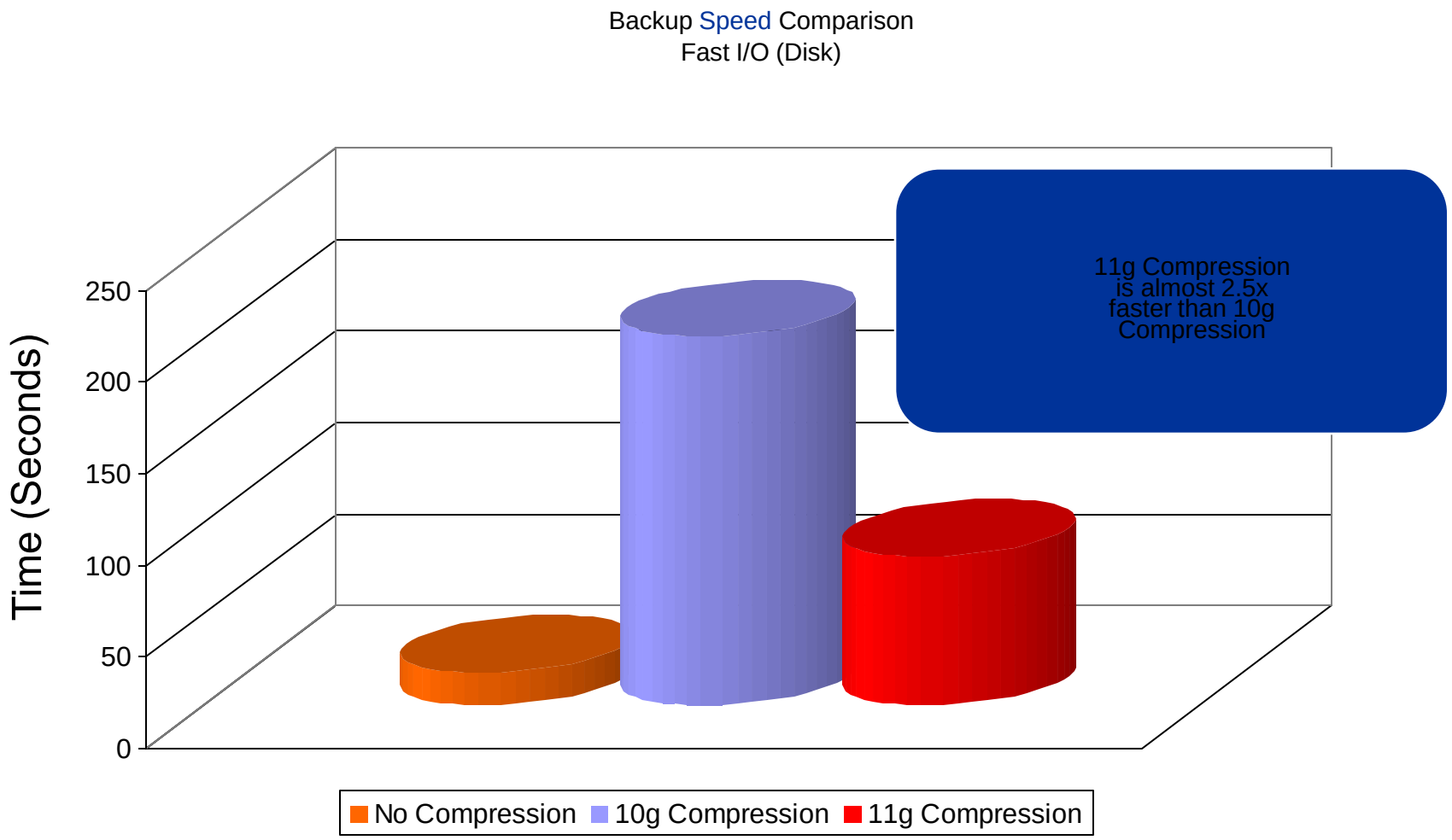
Backup Compression



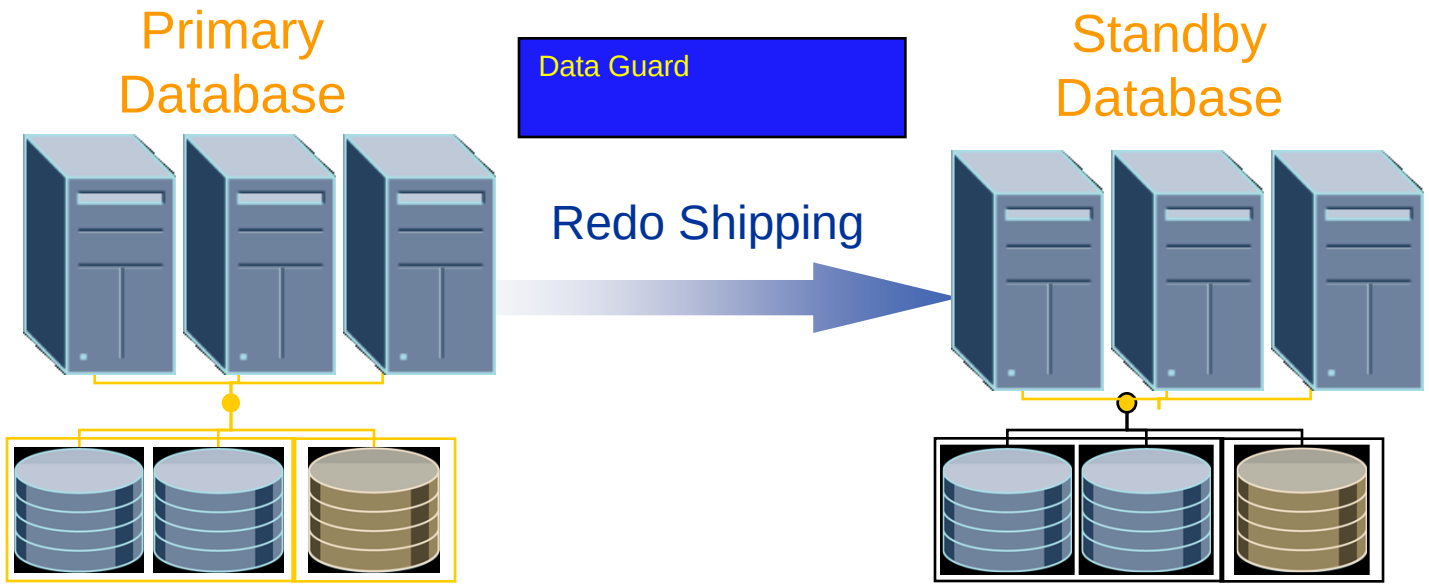
Backup Compression



Backup Compression



Oracle Data Guard



Redo Shipping

- Send redo data over network from primary to standby
- Size of redo data typically small (transactional) and not network-bound

Gap Resolution

- After network outage – resynchronize standby
- Size of redo data much larger

Oracle Data Guard Redo Transport Services

Fast re-sync of standby database after network outages

Lower bandwidth networks (<100Mbps)

- 15-35% less time required to transmit 1 GB of data
- Bandwidth consumption reduced up to 35%

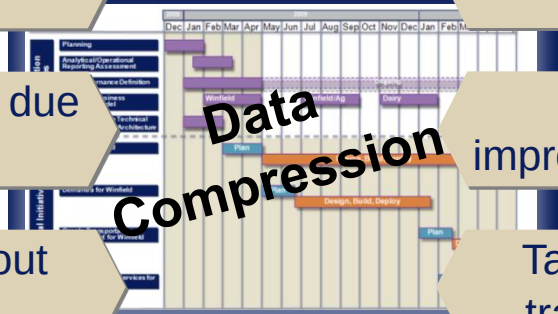
High bandwidth networks (>100 Mbps)

- Compression will not reduce transmission time
- But will reduce bandwidth consumption up to 35%

What can data compression do for you?

Let us look at why data compression needs to be in the BIDW roadmap of the companies to address common issues

Issues	Opportunities
Explosion on Data Volumes	Absolute volumes of growth can be reduced by using compression
Cost of licensing Advanced Data Compression	Total cost of ownership is reduced
Reduciing Query Performance due to large database size	Compression leads to improvement in query performance
DB is compressed, what about backup?	Table, index, backup, network traffic, all can be compressed
Compatibility with other 11g features like DB m/c or Partitioning	Compression can be used with Exadata storage
Fear of adoption and learning curve of data compression	No impact to app developers/end-users, minimal impact for DBA's



Questions

Reminder: Please complete the speaker survey

Contact Info: **ShyamVaran@Gmail.com**

(954) 609 – 2402 cell

http://OracleBIWA.org

About Oracle BIWA SIG

A world wide body of 1600+ professionals to share interests in the below areas:

•Oracle Database, Oracle OLAP Option, Oracle Data Mining Option, Partitioning Option, statistical functions, SQL Analytics, Oracle Text, Oracle *inter*Media, regular expression searches, text mining, XML Publisher, etc.

• Query and reporting tools, BI tools and BI dashboards including Oracle BI SE, Oracle BI EE, and similar tools from Oracle ISVs and partners.