**A**

**Report On**

**“Implementation of Multi-Tenant Environment in cloud computing”**

****

**Prepared by**

Srishu Chintakindi(20CE012)

Nidhi Batra(20CE002)

Yash Bhalgamiya(20CE005)

**Under the supervision of**

Prof. Arpita Shah

A Report Submitted to

Charotar University of Science and Technology

for Partial Fulfillment of the Requirements for the

Degree of Bachlors of Technology

in Computer Engineering

(4th Semester CE255 )

**Submitted at**

22nd February, 2022



**U & P U. Patel Department of Computer Engineering**

**Chandubhai S. Patel Institute of Technology (CSPIT)**

**Faculty of Technology & Engineering (FTE), CHARUSAT**

**At: Changa, Dist: Anand – 388421**

# ABSTRACT

Heat scintillation mainly leads to atmospheric turbulence which causes the image distortion due to the propagation of light through the volatile environment. The change in the refractive index due to variation in wind velocity is also the reason for causing turbulence in the atmosphere. Since last few year`s atmospheric turbulence has been a significant research topic and various approaches have already been proposed to mitigate it. Traditional image registration approach lags as it is computationally expensive and need post-processing algorithm for sharpening the image. A non-registration based Sobolev Gradient and Laplacian (SGL) algorithm removes turbulence but results in ghost artifacts in moving objects. This paper proposes a novel approach based on weighted average SGL. The proposed method mitigates atmospheric turbulence as well as restores the moving object in the scene. Performance metrics like SSIM, MSE and PSNR prove that the proposed algorithm outperforms the state of the art algorithms in terms of restoring the geometric distortion as well as restoring the object of interest.

**TABLE OF CONTENTS**

[ABSTRACT II](#_Toc96469468)

[CHAPTER 1. INTRODUCTION 1](#_Toc96469469)

[1.1 What is IMDBMS?? 1](#_Toc96469470)

[1.2 How does an In-Memory Database Work? 2](#_Toc96469471)

[1.3 Why use an In-Memory Database? 2](#_Toc96469472)

[1.4 In-Memory Database vs Traditional Database? 3](#_Toc96469473)

[1.5 Properties of In-Memory Database System 3](#_Toc96469474)

[1.6 Tools and Technology 4](#_Toc96469475)

[1.7 Virtual Storage Compression 4](#_Toc96469476)

[CHAPTER 2. LITERATURE SURVEY AND ANALYSIS 5](#_Toc96469477)

[2.1 Dimensions of IMDBMS supported by Oracle Times Ten 5](#_Toc96469478)

[2.2 Dimensions of IMDBMS supported by SAP HANA 6](#_Toc96469483)

[2.3 Dimensions of IMDBMS supported by Gridgrain 7](#_Toc96469484)

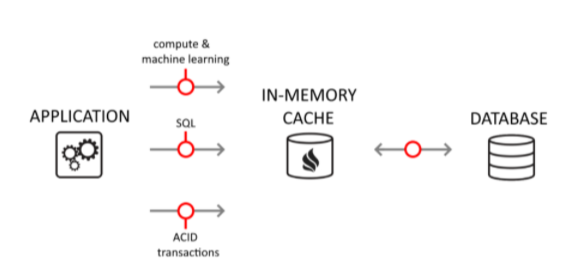
[References 9](#_Toc96469485)

# INTRODUCTION

The introduction of cloud computing has allowed applications to run on shared resources. As a result, the hosting world become subject to economies of scale. In order to truly benefit from this shared resource model, many software providers are faced with the question of implementing multi-tenancy. This model contrasts with traditional software approaches, where each grouping of customers is hosted on their own customized application and database instances. Multi-tenancy promotes sharing of resources all the way from a single application instance to the database. This allows applications to better fit the cloud computing paradigm. All the while, providing increased maintainability, scalability and reducing hosting costs. This thesis attempts to create an architecture description that addresses the design concerns related to the implementation of a multi-tenant that utilizes Microsoft Azure.

## What is IMDBMS??

An in-memory database (IMDB) stores computer data in a computer’s main memory instead of a disk drive to produce quicker response times. Accessing data stored in memory eliminates the time needed to query data from a disk. In-memory databases are used by applications that depend on rapid response times and real-time data management. Industries that benefit from in-memory databases include telecommunications, banking, travel and gaming. An in-memory database is also referred to as a main memory database (MMDB), real-time database (RTDB) or in-memory database system (IMDS).



## How does an In-Memory Database Work?

Data storage in an in-memory database relies on a computer’s random access memory (RAM) or main memory instead of traditional disk drives. Data is loaded into an in-memory database in a compressed and non-relational format. The data is in a directly usable format without the barrier of compression or encryption. It allows for direct navigation from index to row or column and is a read-only system.

The speed of an in-memory database is made possible by lack of translation and caching. The data is used in the same form as the application that contains it. Data access is managed by an in-memory database management system.

An in-memory database system can also act as an read-only analytic database that stores historical data on metrics for business intelligence (BI) applications. This eliminates data indexing, which can reduce IT costs. Multi-core servers, 64-bit computing and lower RAM prices have made in-memory analytics more common.

## Why use an In-Memory Database?

Applications that manage vast quantities of data and require rapid response times can benefit from in-memory database architecture. The data analytics industry increasingly relies on in-memory database systems.

The major advantage of systems using in-memory databases vs traditional database systems is: its performance speed.

The internal optimization algorithms for in-memory databases are simpler and execute fewer CPU instructions, thereby facilitating faster response times than disk-optimized databases.

Source data is loaded into the system memory in a compressed format. Therefore, in-memory processing reduces disk seek time for accessing data and streamlining the work involved in processing queries.

An in-memory database system streamlines processing by eliminating multiple data transfers, reduces memory consumption by removing multiple copies of data, and simplifies processing by minimizing CPU demands.

In-memory databases are commonly used for:

* Real-time banking, retail, advertising, medical device analytics, machine learning and billing/subscriber applications
* Online interactive gaming
* [Geospatial](https://www.omnisci.com/learn/geospatial)/[GIS](https://www.omnisci.com/technical-glossary/gis) processing
* Processing of streaming sensor data
* Developing embedded software systems
* Applications in transport systems, network switches and routers
* Fulfilling the requirements of e-commerce applications

## In-Memory Database vs Traditional Database?

An in-memory database comparison to a traditional disk database includes speed, volume and volatility. An in-memory database keeps all data in the main memory or RAM of a computer. A traditional database retrieves data from disk drives.

In-memory databases are faster than traditional databases because they require fewer CPU instructions. They also eliminate the time it takes to access data from a disk.

In-memory databases are more volatile than traditional databases because data is lost when there is a loss of power or the computer’s RAM crashes. Data can be more easily restored from the disks of a traditional database.

Traditional databases are formatted by the disk drives on which the data is read and written. When one part of a traditional database refers to another part, a different block must be read from the disk. With an in-memory database, different parts of the database can be managed through direct pointers.

In-memory databases allow for real-time analysis and reporting of data.

Traditional databases store redundant data as the system creates a copy of the data for each component added to the system.

## Properties of In-Memory Database System

Most of the In-memory database systems exhibit ACID properties—Atomicity, Consistency, Isolation and Durability.

**Atomicity:** In-memory database entails a single data transfer and avoids multiple data transfer streamline processing. Each transaction is “all or nothing”: where the entire file is left unchanged even if one part of the transaction fails.

**Consistency:** The consistency property ensures that data exists in a consistent and valid state which abides to all defined rules of data existence.

**Isolation:** The isolation property of in-memory database ensures that each transaction is executed in total isolation and is totally independent of any other concurrent transaction.

**Durability:** In-memory database ensures the completion of a transaction once assigned to it, even in the event of power loss, crashes or errors.

## Tools and Technology

The tools and technology required for executing the experiment are described as follows:

AWS/ASP.Net/Azure/GCP/Python

## Virtual Storage Compression

Virtual memory compression (also referred to as RAM compression and memory compression) is a memory management technique that utilizes data compression to reduce the size or number of paging requests to and from the auxiliary storage.[1] In a virtual memory compression system, pages to be paged out of virtual memory are compressed and stored in physical memory, which is usually random-access memory (RAM), or sent as compressed to auxiliary storage such as a hard disk drive (HDD) or solid-state drive (SSD).

# LITERATURE SURVEY AND ANALYSIS

## Dimensions of IMDBMS supported by Oracle Times Ten

Satisfy your need for speed

Oracle based IMDB works on state-of-the-art algorithm.

These optimizations enable Oracle Database In-Memory to run queries at the astounding rate of billions of rows per second for each CPU core.

Oracle based IMDB does not require all database data to fit in memory. Users can choose to populate only performance sensitive tables or partitions into memory.

Queries execute transparently on data residing on all three tiers—memory, flash and disk—enabling Oracle Database In-Memory to be used with databases of any size.

Enabling Oracle based IMDB is as easy as setting the size of the in-memory column store and identifying tables to bring into memory.

No changes are required to use it with any application or tool that runs against the Oracle Database.

Oracle based IMDB inherits all the proven functionality of Oracle Database, including the sophisticated and robust high availability solutions embodied in Oracle’s popular Maximum Availability Architecture (MAA). Oracle based IMDB is fully compatible with Oracle’s Multitenant database architecture, allowing consolidated databases to take advantage of a combination of fast in-memory and low-cost storage technologies.

* Timesten is world’s fastest OLTP database software.
* Timesten IMDB is an [in-memory](https://en.wikipedia.org/wiki/In-memory_database), [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) with persistence and high availability.
* It provides standard relational database APIs and interfaces such as the [SQL](https://en.wikipedia.org/wiki/SQL) and [PL/SQL](https://en.wikipedia.org/wiki/PL/SQL) languages.
* TImesten’s most popular feature is it’s fast response time.
* **API Support:**

      The run time architecture of TimesTen supports connectivity through the ODBC, JDBC, OCI, Pro\*C/C++ Precompiler and ODP.NET APIs.TimesTen also provides built-in procedures, utilities and the TTClasses API (C++) that extend ODBC, JDBC and OCI functionality for TimesTen-specific operations, as well as supporting PL/SQL. API support is described in subsequent section

* **Access Control:**

TimesTen and TimesTen Cache are installed with access control to allow only users with specific privileges to access particular TimesTen features. TimesTen Access Control uses standard SQL operations to establish user accounts with specific privileges. TimesTen offers object-level access control as well as database-level access control.

* **Database connectivity:**

TimesTen and TimesTen Cache support direct driver connections for higher performance, as well as connections through a driver manager. TimesTen also supports client/server connections.

### Durability:

### TimesTen and TimesTen Cache achieve durability through a combination of transaction logging and periodic refreshes of a disk-resident version of the database. Log records are written to disk asynchronously or synchronously to the completion of the transaction and controlled by the application at the transaction level.

### Concurrency:

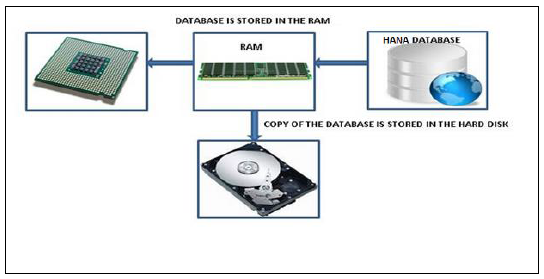
### TimesTen and TimesTen Cache provide full support for shared databases. Options are available so users can choose the optimum balance between response time, throughput and transaction semantics for an application.

## Dimensions of IMDBMS supported by SAP HANA

The main features of SAP HANA in-memory database are −

* SAP HANA is Hybrid In-memory database.
* It combines row based, column based and Object Oriented base technology.
* It uses parallel processing with multicore CPU Architecture.
* Conventional Database reads memory data in 5 milliseconds. SAP HANA In-Memory database reads data in 5 nanoseconds.

It means, memory reads in HANA database are 1 million times faster than a conventional database hard disk memory reads.



**Advantages of In-Memory Database**

* HANA database takes advantage of in-memory processing to deliver the fastest data-retrieval speeds, which is enticing to companies struggling with high-scale online transactions or timely forecasting and planning.
* Disk-based storage is still the enterprise standard and price of RAM has been declining steadily, so memory-intensive architectures will eventually replace slow, mechanical spinning disks and will lower the cost of data storage.
* In-Memory Column-based storage provides data compression up to 11 times, thus, reducing the storage space of huge data.
* This speed advantages offered by RAM storage system are further enhanced by the use of multi-core CPUs, multiple CPUs per node and multiple nodes per server in a distributed environment.

## Dimensions of IMDBMS supported by Gridgrain

Apache Ignite (Ignite) is the leading Apache Software Foundation (ASF) project for in-memory computing. It is one of the top five ASF projects in terms of commits and email list activity. Ignite is an in-memory computing platform that includes an in-memory data grid (IMDG), in-memory database (IMDB), support for streaming analytics, and a continuous learning framework for machine and deep learning.

The source code for Apache Ignite was originally contributed to the Apache Software Foundation by GridGain Systems.

GridGain's In-Memory Computing Platform is used in a broad range of applications around the world including:

* Financial trading systems
* Online gaming
* Bioinformatics
* Hyperlocal advertising
* Cognitive computing
* Geospatial analysis

GridGain's complete In-Memory Computing Platform enables organizations to conquer challenges that traditional technology can't approach.

GridGain's product line delivers all the high performance benefits of in-memory computing in a simple, intuitive package. From high performance computing, streaming and data grid to an industry first in-memory Hadoop accelerator, GridGain provides a complete end-to-end stack for low-latency, high performance computing for each and every category of payload and data processing requirements

.

# References

**Papers:**

[1] Application Development with Oracle Database 12C

ORACLE WHITE PAPER | JANUARY 2017

[2] SAP HANA® Database for Next-Generation Business Applications and Real-Time Analytics

[3] In-Memory Data Grid White Paper Grid Gain Systems, 2013

**Websites:**

[1] <https://www.omnisci.com/technical-glossary/in-memory-database>

[2]<https://searchdatamanagement.techtarget.com/definition/in-memory-database-management-system-IMDBMS>

[3]<https://paristech.com/blog/in-memory-data-advantages/#:~:text=II.-,Properties%20of%20In%2Dmemory%20Database%20Systems,multiple%20data%20transfer%20streamline%20processing>.

[4] <https://www.sciencedirect.com/topics/computer-science/compression-technique>

[5]<https://www.google.com/search?q=Architecture+of+disk+base+dbms&oq=Architecture+of+disk+base+dbms&aqs=chrome..69i57j0l3.11377j0j7&sourceid=chrome&ie=UTF-8>

[6] <https://www.tutorialspoint.com/sap_hana/in_memory_computing_engine.htm>

[7] https://devopedia.org/in-memory-database#qst-ans-6