Middle East University for Graduate Studies

DATABASE FRAGMENTATION TECHNIQUE FOR NEWS WEBSITES

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Information Systems

Wael Sulaiman Hamad Al-murafi

Supervisors

Prof. Mohammad A. Al Fayoumi

Dr. Ezz Hatab

DEDICATION

This is dedicated to my mother, my family, and their love and encouragement.

COMMITTEE DECISION

This Thesis **(DB Fragmentation Technique for News Websites)**
Was successfully defended and approved on 31 May 2009.

Examination Committee Signature

Prof. Mohammad Ahmad Al Fayoumi
Professor / Department of Computer Information System
(Middle East University for Graduate Studies)

Dr. Hazim Farhan
Assistant Professor / Department of Computer Information System
(Middle East University for Graduate Studies)

Dr. Ezz Hatab
Associate Professor / Department of Computer Information System
(The Arab Academy for Banking and Financial Sciences)

Dr. Hussein Al-Bahadili
Associate Professor / Department of Computer Science.
(The Arab Academy for Banking and Financial Sciences)

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to Professor Mohammad Al Fayoumi and Dr. Ezz Hatab for their guidance, support and motivation throughout my Master's Thesis.

I would further like to acknowledge all of the Information Technology faculty members at the Middle East University for Graduate Studies and I am particularly grateful to Professor Musbah Aqel, for helping and encouraging my efforts especially at the beginning of the thesis.

Then I would like to especially thank my family for supporting me during the time I was writing this thesis.

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

- **UDI** **Update, delete and insert.**
- **DB** **Databases.**
- **CERN** **The European Organization for Nuclear Research.**
- **HTML** **Hyper Markup Language**
- **ASP** **Active Server Page.**
- **XML** **Extensible Markup Language.**
- **IDE** **An integrated development environment**
- **CLR** **Common Language Runtime**
- **RDBMS** **Relational database management system**
- **DDBS** **Distributed database system**
- **SQL** **Structured Query Language**
- **ACID** **Atomicity, Consistency, Isolation, Durability**
- **HAMP** **Hierarchical Synchronized Multilevel consistency Protocol**
- **CPU** **Central processing unit**
- **WAN** **Wide area network**
- **MMC** **Microsoft management console**

ABSTRACT

The continued growth of the Web and its increasing role in our daily life has created new technical and social challenges. A large number of these Web applications increasingly use a database to generate customized and personalized responses to users' requests. Due to of the widely varying load, currently there is no economical way to provision infrastructure for many of these applications in which the DB system is the bottleneck. On the social side, Web applications increasingly collect sensitive data, which must be kept private.

Response time is essential to many Web applications. Consequently, many DB-driven Web applications rely on data centers that host applications and database contents. Such IT infrastructure enables generation of requested pages at locations much close to the end-users, avoiding network latency. However, it then has additional challenges of DB/data center synchronization and data freshness. In this thesis, discuss in DB fragmentation for news website and DB consistency by using programming language.

In practical test, we have taken ammonnews.net website as sample to measure the technique and take results.

الملخص باللغة العربية

ادى النمو المستمر للمواقع الالكترونية على شبكة الإنترنت الى تحديات جديدة على مستوى التقنية وادارة الكم الهائل من المواقع الاكترونية التي تستخدم قواعد البيانات. ولذا فان استخدام قواعد البيانات بشكل رئيسي يؤدي الى زيادة الضغط على الحاسوب الرئيسي وبالتالي الى تاخير استرجاع البيانات و تباطؤ الموقع بشكل عام.

ونتيجة للتنافس الشديد للمواقع الالكترونية عامة والاخبارية منها خاصة ، فانه لا بد من تقديم خدمات تقنية عالية الجودة من اهمها سرعة استرجاع البيانات لتلك المواقع.

ومن الطرق تحسين قواعد البيانات للمواقع الاخبارية هي استخدام التجزئية للبيانات، بحيث يتم استرجاع البيانات من اكثر من قاعدة بيانات حسب تصنيف الخبر.وبذلك يتم تخفيف الضغط على قاعدة البيانات الرئيسية. وهي موضوع دراستنا في هذه الرسالة. ونحن هنا سنناقش قواعد البيانات للمواقع الاخبارية الالكترونية وتم اختيار موقع عمون الاخباري كنموذج للبحث والاختبار.

CHAPTER 1

INTRODUCTION

1. BACKGROUND

Recently, news Websites viewed as an important media to deliver the contents as soon as possible to the intended audience because there are fierce competitors in the news battlefield. The number of news websites is growing quickly. So the scoop (be the first to publish an article) plays a major role in the electronic-based media.

Reliability and response time are two key points for news websites. The need for accounting for users' quality perception in designing Web servers for dynamic website systems has been highlighted by [12]. Snafus (problems) and slow-downs in any website at major websites during special events or peak times demonstrate the difficulty of scaling up dynamic website. Slow response times and down times can be devastating for dynamic website as reported in a study by Zona Research [33] on the relationship between Web page download time and user abandonment rate*. The study showed that only 2% of users will leave a Web site (i.e. abandonment rate) if the download time is less than 7 seconds. However, the abandonment rate jumps to 30% if the download time is around 8 seconds. The abandonment rate reaches 70% as download times exceed 12 seconds. This study clearly establishes the importance of fast response times to a news website to retain its customers.

In any case, many of systems for scalable hosting of Web applications have been proposed. These systems typically cache (fragments of all pages) the generated pages [7],

---

* Percentage of the calls made by an automatic dialing device to prospective customers

cache the results of database queries [8] or distribute the computation across multiple application servers [17, 11]. Although these techniques can be very effective depending on the application, however in many cases their ultimate scalability bottleneck resides in the throughput of the origin database where the authoritative version of the application state is stored.

DB fragmentation would be helpful, the system throughput is therefore limited to the point where the quantity of UDI (update, delete and insert) queries alone is sufficient to overload one server, regardless of the number of machines employed [2]. The solutions to this problem are to increase the performance of individual server or to use DB fragmentation.

In technical terms, ensuring the timely delivery of fresh dynamic content to end-users and engineering highly scalable news Websites for special peak access times put heavy demand on IT staff. This load is compounded by the ever-changing complexity of dynamic website applications. For many news applications, Web pages are created dynamically based on the current state of news, such as latest news or news search system. This characteristic requires Websites deploy cache servers, Web servers, application servers, and database systems [15].

In this thesis, we will focus on DB fragmentation, especially in transparency and consistency using programming language such as asp.net with c#, Ammon news website has been taken as an experimental study.

1.1. BEGINNINGS OF WEB CONTENT

While the Internet began with a U.S. Government research project in the late 1950s, the Web in its present form did not appear on the Internet before Tim Berners-Lee and colleagues at the European laboratory (CERN) proposed the concept of linking documents with hypertext. But it was not available till Mosaic (earlier web browser), then the Internet become more than a file serving system [31]. The use of hypertext, hyperlinks and a page-

based model of sharing information, introduced with Mosaic and later Netscape, helped to define web content, and the formation of websites.

1.1.1. THE PAGE CONCEPT

Web content is dominated by the "page" concept. Having its beginnings in academic settings, and in a setting dominated by type-written pages, the idea of the Web was to link directly from one academic paper to another academic paper. This was a completely revolutionary idea in the late 1980s and early 1990s when the link could be made to cite a reference in the midst of a written paper and names that reference either at the bottom of the page or on the last page of the academic paper [31].

At the time any person could write and own a Mosaic page, the concept of a "Home Page" blurred the idea of a page. It was possible for anyone to own a "Web page" or a "Home Page" which in many cases the website contained many physical pages in spite of being called "a page". People often designed their "Home Page" to provide credentials, links to anything that a person supported, or any other individual content a person wanted to publish [24].

Even though "the web" may be the resource it is commonly use "get to" particular locations online, many different protocols [18] are invoked to access embedded information. Giving us a link, such as http://www.youtube.com, we expect to see a range of web pages, but in each page we have embedded tools to watch "video clips".

1.1.2. CONTENT IN THE NEWS WEBSITES

A news site is a website with the primary purpose of reporting news. There are two main types of news site: general news and subject-specific.

The articles or subject in news website belong to sections such as sports, political, life style, and local news …etc.

Each page or part in news websites will be connected with database by "connection string". A connection string is specifies information about a data source and the means of connecting it. It is passed in code to an underlying driver or provider in order to initiate the connection. The connection string may include attributes such as the name of the driver, server and database, as well as security information such as user name and password [27].

In many cases, a web page must be able to deliver content of those changes, either over time (automatic) or by a user. For example, the following scenarios require dynamic processing in news website:

- User wants the latest news.
- User search about specific article or statement in such category.
- Customer need to send article by news website.
- Admin user insert, update or delete news.

1.2. DATABASE FRAGMENTATION

Fragmentation is a relation which partitioned into several fragments stored in distinct sites. Division of relation R into fragments R1, R2… Rn which contain sufficient information to reconstruct relation R. There are three types of DB fragmentation [5]:

1-Horizontal fragmentation: each tuple of R is assigned to one or more fragments.

2-Vertical fragmentation: the schema for relation R is split into several smaller schemas.

3-Hybrid fragmentation: mixed two type above.

1.3. STATEMENT OF PROBLEM

The central database model simplifies implementation and configuration as there is a single source for using graphs or reports, where is the main problem in one big database or control them from one central location that barriers imposed by communications bandwidth and speed-bound central hardware continue to fall. Growths of DB volume in news websites and Use central DB technique cause low speed for the website downloading. The strongest challenges that encountered news websites designer are how to design optimal DB fragmentation to increase response time in dedicated hosting. From other side, how to keep consistency for each DB.

1.4. THESIS MOTIVATION

- The growing number of complaints about news websites speed.
- How to make DB fragmentation and consistency for news websites DB.

1.5. SIGNIFICANCE OF THE STUDY

The main objectives of this thesis may include:

- Reduced respond time for news websites in dedicated hosting: A DB fragmentation system should be designed to store data at the location where it is used more often.

- Availability and reliability. Reliability is defined as how to keep DB consistency when errors occurred. The availability is the probability that the system will be up continuously during a given time period. In the centralized DB, if any component of the DB goes down, the entire system will be down, whereas in the DB fragmentation, only the effected fragment is down, and the rest of the system will not be affected.

- Performance improvement. When large DB is fragmented into a number of DB, the local subset of the database is smaller, which will improve the size of transactions and the processing time.

1.6. CONTRIBUTION OF THE THESIS

The most important contributions of this thesis are the following:

- Design and implement system with shortest possible response time in dedicated hosting news websites.
- Design and build the DB fragmentation for news website as a dynamic Web content, with fully decentralized cache consistency management.
- Design and build integrated system to keep the DB consistency.
- How can handle the consistency errors to transfer data from DB to other.

1.7. RELATED WORKS

1. **[On Data Fragmentation and Allocation in Distributed Object Oriented Databases,1997]**

   This research presented by A. Koreichi and B.Le Cun.

   The objectify of object oriented databases is to respond to the needs of new applications as engineering and multimedia which manipulate complex data.

   Furthermore, most of these applications need to execute in a distributed environment, making necessary the distribution of data on different sites.

   This must guarantee a minimum cost of inter-site communication and minimum access to irrelevant data distribution has largely been studied in the relational model, but the complex structure of objects and their relationships make it difficult for object oriented databases.

In this study, both fragmentation and allocation problems are tackled. Hybrid fragments are defined by applying horizontal fragmentation followed by vertical fragmentation to each database class.

Resulting fragments are disjoint regarding to data but overlapping regarding to methods. An optimization model for a non replicated data allocation in the form of a non linear integer zero-one programming problem is also developed.

The objective function proposed aims to minimize communication and storage costs.

2. [AI Clustering Techniques: a New Approach in Horizontal Fragmentation of Classes with Complex Attributes and Methods in Object Oriented Database , 2004]

This study presented by Adrian Darabant, Alina Câmpan, Grigor Moldovan, Horea Grebla.

Horizontal fragmentation plays an important role in the design phase of Distributed Databases. Complex class relationships: associations, aggregations and complex methods, require fragmentation algorithms to take into account the new problem dimensions induced by these features of the object oriented models.

In this work proposed a new technique for horizontal fragmentation in object-oriented databases with complex attributes and methods. Fragmentation in complex OO hierarchies is usually performed in two steps: primary fragmentation and derived fragmentation. Primary fragmentation groups class instances according to a set of class conditions, imposed on their simple attributes.

Derived fragmentation takes into account the class relationships (aggregation, association, complex methods). It groups instances of a class in fragments according to the fragmentation of the related classes.

There are generally two approaches in derived fragmentation: left order derived fragmentation (parent first) and right order derived fragmentation. They differ in the order in which two related classes are fragmented.

In the left order derived fragmentation, the referring class is fragmented first and determines a partitioning of the instance set of the referred class. In the right order derived fragmentation, the referred class is fragmented first and determines the partitioning of the instances of the referring class.

Provide quality and performance evaluations using a partition evaluator function and proved that fragmentation methods handling complex interclass links produce better results than those ignoring these aspects.

3. [Fragmentation Solution Used in The Projection of the Distributed DB Systems, 2004]

This research presented by I. Lungu , T. Vatuiu, and A. G. Fodor. Presents the main methods of database fragmentation in distributed systems. The steps of the distributed database design are presented. the fragmentation rules and the fragmentation types is described followed by the presentation of a mixed fragmentation methodology for initial distributed database design based on a graphical algorithm. The research ends with presentation of some aspects of fragmentation methods.

There are three rules which must be respected in order to accomplish a correct fragmentation of a relation from the data base:

1. The complete character.
2. The reconstruction.
3. the disjoint character.

They two main types of fragmentation: horizontal and vertical. Horizontal fragments represent subsets of tuplets, and the vertical fragments attributes subsets. In case there are

two more types of fragmentation: mixed and derivate (which represents a type of horizontal fragmentation).

For initial distributed database design based on a graphical algorithm. This approach also has the advantage of studying simultaneous the vertical and horizontal fragments. It uses the top-down strategy for creating the grid and the bottom-up strategy for its optimization, balancing between ensemble and detail and heading to the best solution of the performance-cost relation.

4. [Transparent Caching with Strong Consistency in Dynamic Web Content, 2005]

This study presented by C.Amza, G.Soundararajan, E.Cecchet. Considers that a cluster architecture in which dynamic content is generated by a database back-end and a collection of Web and application server front-ends. Then they study the effect of transparent query caching on the performance of such a cluster. With Based on observed workload characteristics, enhance the cache with the necessary dependency tracking and invalidations at the finer granularity of columns.

The main steps in this experience first observe that in compute-intensive (computer application that demands a lot of computation) queries that involve table joins and sorting or grouping, only a couple of attributes are of interest to the user Next reduce the impact of automatic invalidations by either making them unnecessary or reducing the penalty of a cache misses through full or partial coverage of query results. Then maximize the probability of significant partial coverage for read-only queries from the cache, by keeping track of newly inserted rows in separate small temporary tables.

Their work evaluated the tradeoffs between the different cache designs and the system's architectural trade-offs between the various locations at which such cache(s) can be positioned.

Figure 1.1 depicts the main cache data structures. The primary structure is a large hash table containing cache entries. Each entry contains the SQL query string, the state of the entry, the query result, the query's dependencies, and some additional fields to implement the cache replacement algorithm.

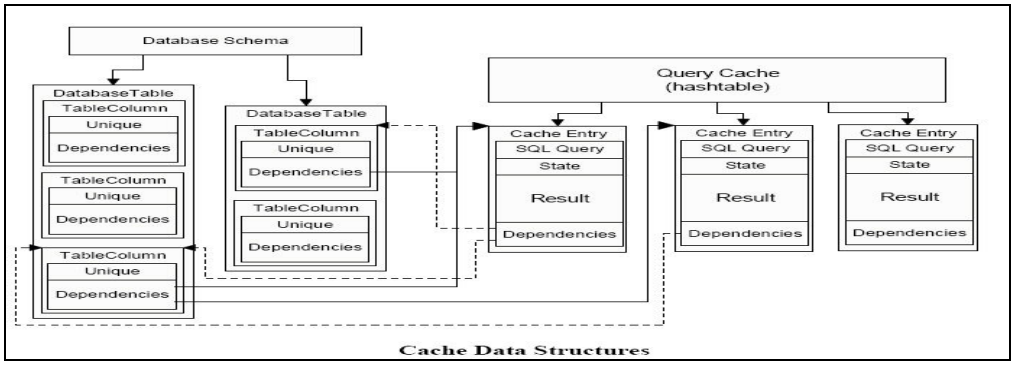


Figure 1.1 cache data structures
Source: Transparent Caching with Strong Consistency in Dynamic Web Content paper

1.8. METHODOLOGY

News websites database usually consists of multiple tables, especially articles tables (subject), which contains a huge amount of records either, published or archived articles, and has sections table such as arts, sports, local news. If there are need to retrieve data from article table, database engine will be order all records by particular parameter such as Date then retrieve the specific records from table. Practically, the process of querying the news is relatively slow if DB size is very big. However, most of the underlying design guidelines behind the model were taken from online papers and guidebooks related to the effective DB fragmentation in news websites.

This thesis mainly based on how to find DB fragmentation technique for news website to ensure DB consistency and how to decrease the number of records to the minimum that could be reached in news DB.

Thesis suggests DB Fragmentation for each active page in news website by fragmented original DB to subsets of DBs. And discuss data structure for news websites by showing

ER diagram and DB schema. Then discuss how to handle read and write queries by keeping consistency for each DB. Finally discuss practical measurement to ammon news websites.

Model measurement result showed that the advantage of fragmentation is to decreases CPU usage especially when the DB size is enormous with high server hits.

CHAPTER 2

DB FRAGMENTATION

2.1. WHAT IS DB FRAGMETATION

DB fragmentation means the breakup of an entity into smaller logical units of distribution. Consider a gathering of fifty people seated in a row that we would like to divide evenly into two even groups so that no two people adjacent to each other are from the same group. The simple approach would be to number them alternately one and two in the order in which they are seated. All of the "ones" would fall in one group, and all of the "twos" would fall in the other group. This is a simple illustration of a horizontal fragmentation using a round robin strategy. Each group is a fragment, or partition [20].

DB fragmentation is the highest level of distribution transparency, and the one that should be strived for in a grid environment, as location and fragmentation levels are likely to change within the grid as the demand fluctuates. DB access is based on a global schema and the user (either human user directly accessing the data in batch or other manner) does not have to be able to supply a fragment name or location to access that data. A query by user needs to be able to access the fragments or the distribution means in the same manner that it would access a centralized DB, without having to be rewritten [4].

2.2. THE PROJECTION OF THE FRAGMENTATION

In the process of projecting the distributed data bases, the relations from a data base's scheme are, usually, decomposed in smaller fragments. There are three rules which must be respected in order to accomplish a correct fragmentation of a relation from the data base [5]:

1. **The complete character**. If an R instance of a relation is decomposed in the fragments $R_{i,\ i=1,2,\ldots,n,}$ then each data article which exists in the R relation must

appear in at least one fragment R_i. This rule guarantees that no data are lost in the fragmentation.

2. **The reconstruction.** There must be possible to define a relational operation through which the R relation can be reconstructed from fragments. This rule guarantees the keeping of the functional dependencies.

3. **The disjoint character.** If an article appears in the fragment R_i, "i=1, 2… n, then it must not appear in any other article. There is only one exception from this rule, in the case of the vertical fragmentation, when every fragment must contain the attributes of the primary key, so that the reconstruction of the relation can be accomplished. Through this rule a minimal and necessary redundancy of data is assured.

2.3. RELIABILITY

Reliability (consistency) guarantees that a transaction never leaves the database in a semi finished state. If one part of the transaction fails, all of the pending changes are rolled back, leaving the database as it was before you initiated the transaction. As example, when you delete a customer record, you should also delete all of that customer's records from associated tables. A properly configured database wouldn't let you delete the customer record, if that meant leaving remains of a customer's records in a table such as invoices [16].

2.4. ADVANTAGES OF DB FRAGMENTATION

Advantages of DB fragmentation are as follows*:*

- Fragmentation can help to better the performance and reduce the I/O contention. If you are looking to increase the parallelization and throughput, fragmentation could be the answer.
- DB fragmentation allows parallel processing of data [20].

2.5. DB FRAGMENTATION TYPES

There are three types of fragmentation:

1. **HORIZONTALLY FRAGMENTED DATA**

A horizontal fragment is a subset of the relation's tuple. It results through the specification of a predicate which imposes a restriction upon the relation's tuples. Horizontal fragmentation is obtained through the application of the selection operation from the relational algebra, by grouping the tuples which have a common priority [5]. The fragmentation strategy supposes finding a minimal set (complete and relevant) of predicates which can be used as base for the fragmentation scheme [6]. A set of predicates is complete if and only if any application makes reference to any two tuples from the same fragment with the same probability. A predicate is relevant if there is at least one application which accesses in different ways the resulted fragments [5].

2. **VERTICLALLY FRAGMENTED DATA.**

The vertical fragment of a relation is defined with the help of the projection operation from the relational algebra and is formed from a subset of its attributes. Any fragment of the R relation, which results from a process of vertical fragmentation, contains attributes that compose the primary key, necessary for the re-establishment of the relation. Vertical fragmentation is much more complicated that horizontal fragmentation. This type of fragmentation groups some attributes depending on the way in which they are accessed by the user applications. This grouping is maid taking in account the "affinity" of one attribute for another. A method to determine it is through creating a matrix which shows the number of accesses referring to each attribute [5].

3. **MIXED FRAGMENTED DATA.**

Mixed fragmentation as a process of simultaneously applying the horizontal and vertical fragmentation on a relation. It can be achieved in one of two ways: by performing

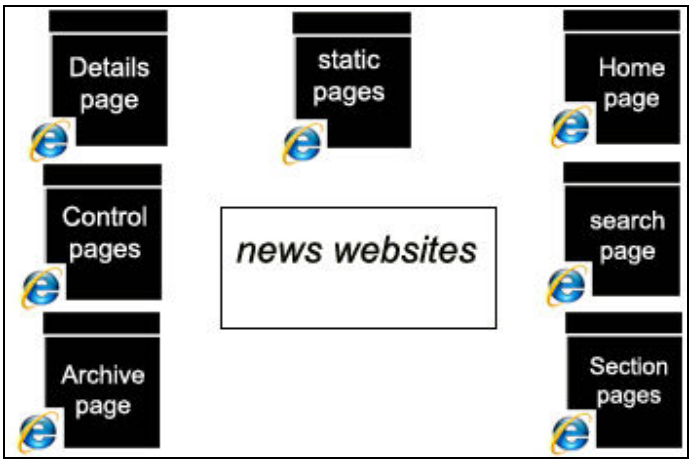
horizontal fragmentation followed by vertical fragmentation or by performing vertical fragmentation followed by horizontal fragmentation. The need for mixed fragmentation arises in distributed DB because DB users usually access subsets of data which are vertical and horizontal fragments of global relations and there is a need to process queries or transaction that would access these fragments optimally [13].

CHAPTER 3

PROPOSED ARCHITECTURE

3.1. FRAGMENTATION IN NEWS WEBSITE

The proposed news websites structure has seven basic parts (figure 3.1):



3.1 basic parts model of news websites

1. Home page: is the starting point or front page of a Web site. This page usually has some sort of table of contents on it and often describes the purpose of the site. For example, http://www.apple.com/index.html is the home page of Apple.com. When you type in a basic URL, such as "http://www.cnet.com," you are typically directed to the home page of the Web site. In news website the home page is the web page has the most recent news from each section.
2. Sections: web pages have related news in website such as sports, political, life style.
3. Archive: web pages of Expired news in the website.
4. Search: web page has program that searches documents for specified keywords and returns a list of the documents where the keywords were found.
5. Details: web page has full information about the article or news.

6. Control panel: control all pages such as: insert, update or delete news. And can configure nearly anything in the website using privileged username.
7. pages: such as “about us”, “contact us” …etc. this part is not of our concern in this thesis.

Dynamic content Web server consists of a front-end, containing the web server and the application logic, and back-end databases. In the proposed system original DB will be fragmented to 4 DBs, each one has news as will be needed in the various parts of news website (**Figure 3.2**) on a read query, and the web server decides which database connection is needed. After that will execute query from specific DB and checks whether the results of the query reside in the DB, and, if so, returns them immediately to the application. On an update query, the application performs the necessary update in the original DB then forwards the update to the other DB (consistency part below).

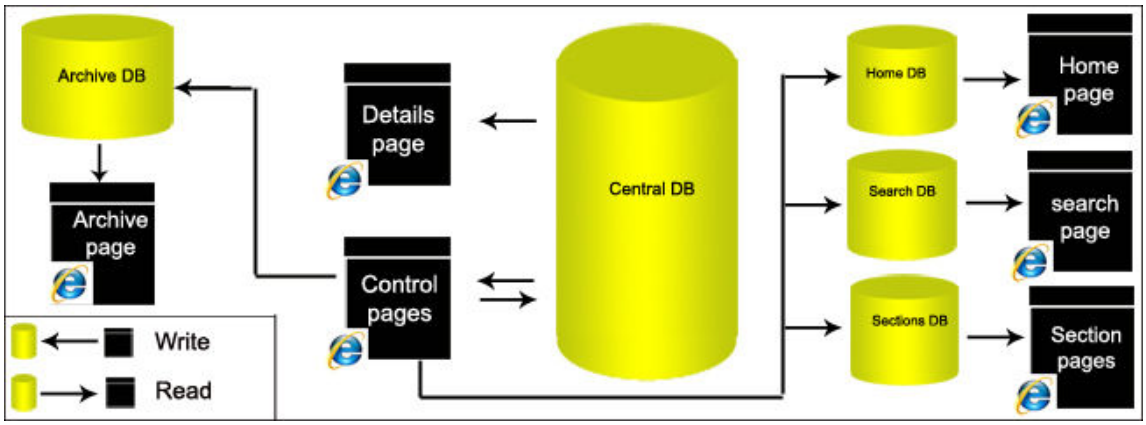


Figure 3.2 General model of DB fragmentation for news website

3.1.1. DATA STRUCTURES

Different news websites have several DB schemas, but in general it has a lot of attributes (fields) such as **subject_id**, title, details…etc. Figure 3.3 shows schema diagram as standard DB for news website.

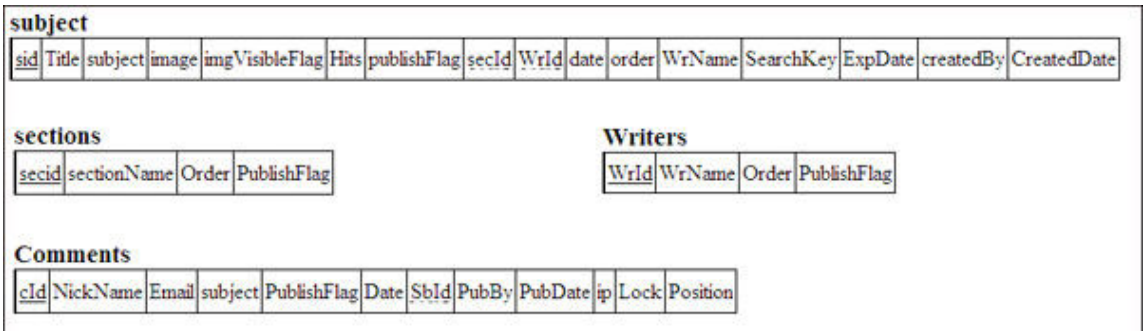


Figure 3.3 Data structure for news website.

TYPE OF TABLES:

1. Subject table.
2. Section table.
3. Writer table.

TABLES STRUCTURE:

Figure 3.4 shows ER diagram for news table

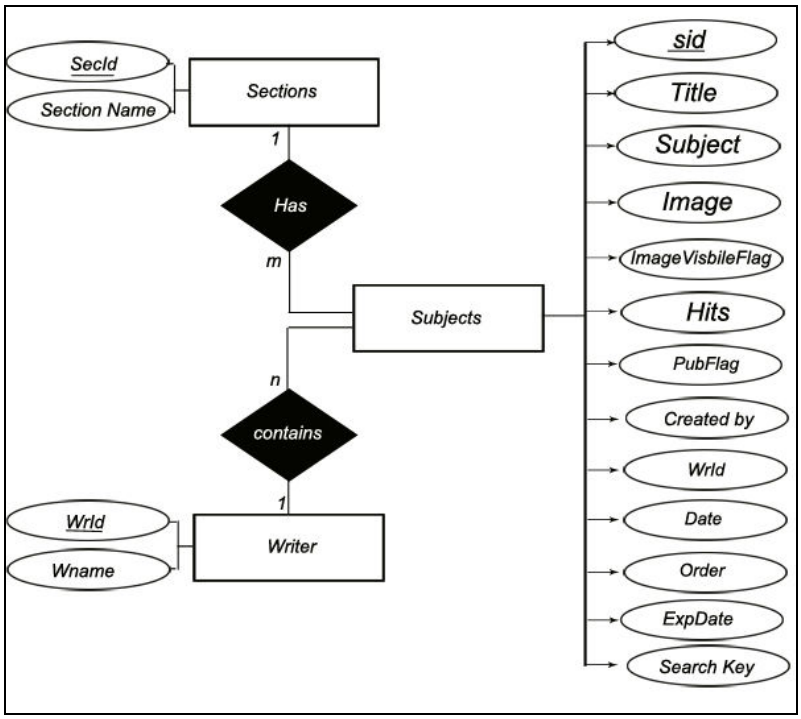


Figure 3.4 ER diagram model for news websites

1. Subject table structure:

- **Sid**: field contains integer number as primary key. We use this field to return news from DB. And later use it to know which row has been changed.
- **Title**: field contains news title.
- **Subject**: field contains news details.
- **Image**: field contains image path from web server to return it.
- **ImageVisibleFlag**: it is Boolean number (0 or 1) to show or hide main image attached to the news.
- **Hits**: field contains read counts for news.
- **PublishFlag**: integer number denotes news state as follow:

PublishFlag=1, denoted to current news (recently news).
PublishFlag=2, denoted to never been published news.
PublishFlag=3, denoted to archived news.
PublishFlag=4, denoted to deleted news.

- **SecId**: field contains an integer number as foreign key to section table. And it denotes to the section in news website as sport section =1 …etc.
- **WrId**: field contains an integer number as foreign key to writer table. This field is used in many positions in website such as return name of a writer or gets the entire writer's articles…etc.
- **Date**: news insertion date.
- **Order**: field contains an integer number; used in order to classify articles according to importance.
- **WrName**: field contains writer's name and we use it to avoid joining between subject table and writer table.
- **Searchkey**: field contains a rich summary about news details. We will use this field in search system in news website.

- **ExpDate**: integer number denoted to how much days news are valid before to be moved to archive.
- **Createdby** and date: save information about user that inserts news with date.

2. Section table structure:

- **Secid**: integer number as primary key.
- **SectionName**: section caption such as sport, local, world…etc.

3. Writer table structure as follow:

- **WrId**: integer number as primary key.
- **WName**: name of the writer.

3.1.2. DB NORMALIZATION

In case, it will be use DB normalization for each DB as show in figure 3.5

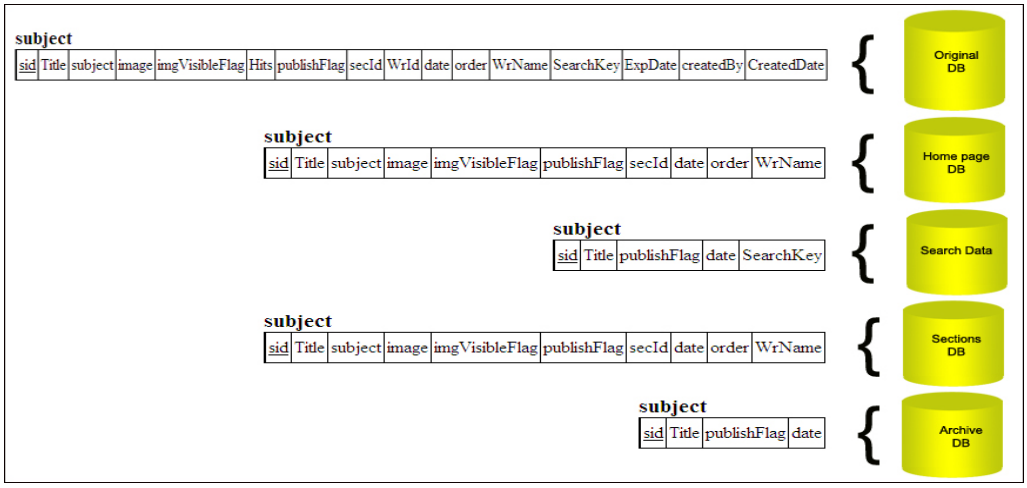


Figure 3.5 DB normalization model for news website

Each DB has functional dependency as follow:

1. HOME PAGE DB:

In general, read query in home page determined by SELECT clause and WHERE clause in SQL queries for example:

```
Select top 6 Sid, Title, subject, image, imgVisibleFlag, publishFlag, date, wrName
from HomePageDB.subject Where secId=x and publishFlag=1 order by [order] Desc
```

Let x= {1, 2, 3, 4, 5, 6…etc} as section table
We assume that each section in home page has 6 articles.

To keep homeDB more efficient ,it must delete all news in subject table that do not appear in home page (out of date) by using scheduled task, works over a period of time. This task SQL query such as:

```
While (Reader.Read(x)*)
{
Delete HomePageDB.subject where Sid not in (Select top 6 Sid from HomePageDB.subject
Where secId=x and publishFlag=1 order by [order] Desc)
}
```

** Read secID from sections table as x=secID*
Let x= {1, 2, 3, 4, 5, 6…etc} PK of section table

2. SEARCH DB:

General SQL query:

```
Select Sid, Title, date from SearchDB.subject Where  publishFlag=1 and ((Title LIKE
'%'+Text+'%') or (SearchKey LIKE '%'+ Text +'%')) Order by [date] desc
```

3. SECTION DB:

General SQL query:

```
Select Sid, Title, subject, image, imgVisibleFlag, publishFlag, date, wrName From
SectionDB.subject Where secId=x and publishFlag=1 order by [order] Desc
```

Let x= {1, 2, 3, 4, 5, 6…etc} as section id

4. ARCHIVE DB:

General SQL query:

Select Sid, Title, date from ArchiveDB.subject Where publishFlag=3 order by [order] Desc

ORIGINAL DB: HAS FULL INFORMATION SUCH AS NEWS TABLE, SECTION TABLE, COMMENTS TABLE, USER TABLE...ETC WITH FULL DATA.

3.2. READ QUERY HANDLING

If an incoming **read query** to web server (**Figure** 3.6), the web server response is as follows:

1. When a Web server receives a read request by web browser, it examines the file-name extension of the requested page such as home.aspx.

2. ASP.NET framework compiles the page.

3. During compilation, the system decides which DB needed by DB connection driver.

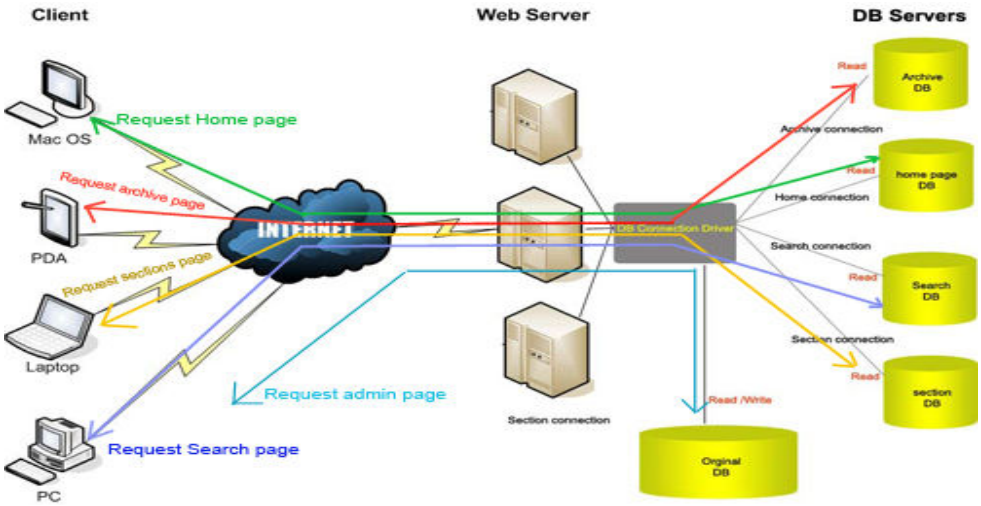


Figure 3.6 Read query handling model

4. If read query matches a valid entry, the web server simply returns the result query from DB. E.g. if user requests the home page from news website, the browser requests the page from the web server. The web server processes the unique page extension and connection driver determines homeDB connection.

5. Completes compilation process and all read query in page.

6. Finally, web server return data to front-end.

3.3. HANDLING UDI QUERIE(CONSISTENCY)

We consider that save all combination of the web server changes in original database (**Figure 3.7**).

3.3.1. CONSISTENCY POLICY

The results of recent UDI queries are saved in original database, and broadcast data from original DB to other DBs (home, Search, Section and Archive DB).

TECHNIQUE STEPS:

1. When a web server receives a write query request by web browser, it examines the file-name extension of the requested page.
2. ASP.NET framework compiles the page.
3. In write queries, the system decides original DB by connection driver to do write operation.
4. When execute query UDI, history thread will be run and save all data changes in history DB as a temporary data. (Policy in section 3.3.2).
5. Each website will have a unique period of time that will automatically run consistency thread (policy in section 3.3.3). And will be responsible of reading all the changes from history DB and copy them from original DB to other (homeDB, SectionDB, SearchDB, ArchiveDB).
6. Then all subs DB begin same original DB.

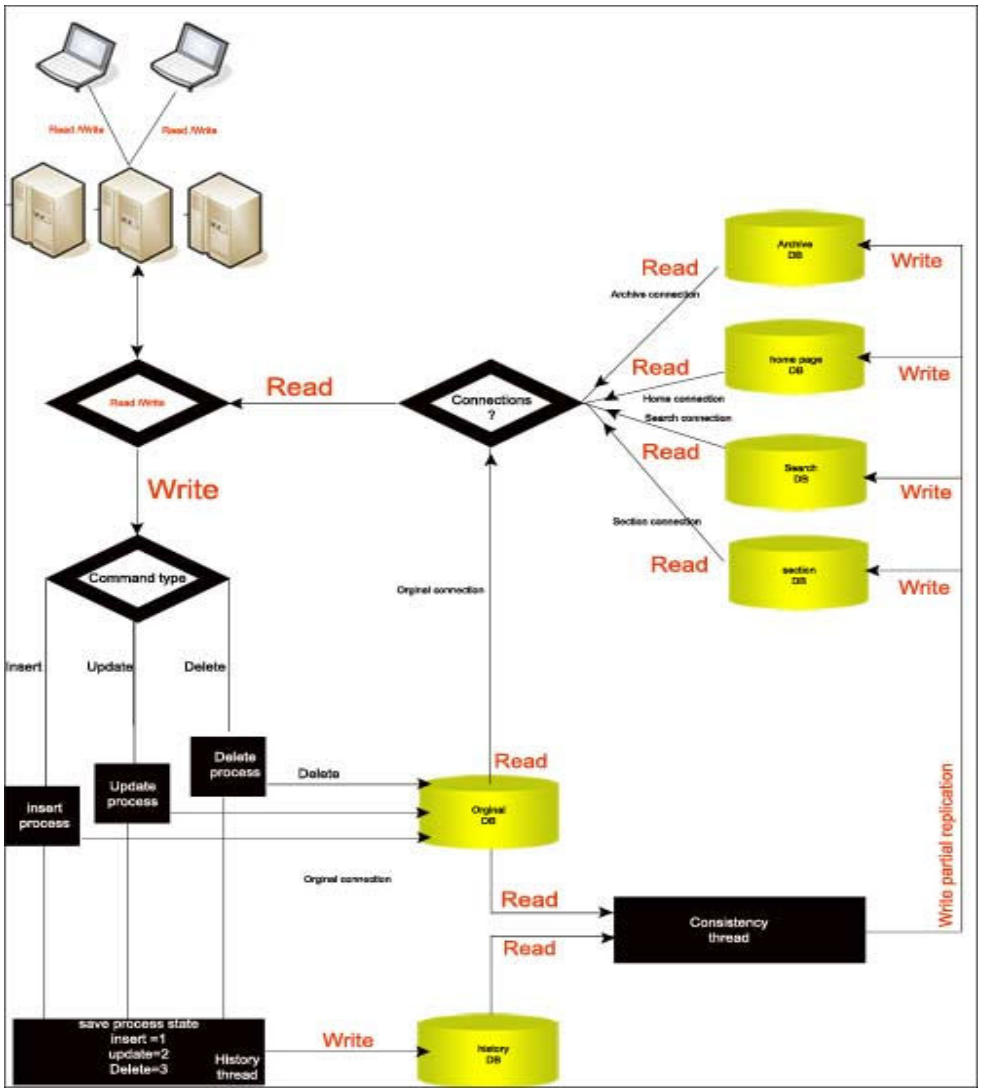


Figure 3.7 General fragmentation steps model in news website

3.3.2. HISTORY THREAD POLICY

History thread is a DB trigger run when any change happens in DB Instance especially in subject table data in original DB. History DB (**Figure 3.8**) contains a one table (history) with 9 columns as follow:

1. **Id**: sequence number as primary key.

2. **TableName**: have a table name that changed its instance e.g. Subject table in original DB.
3. **SubjectId**: the primary key value that points to the changed row in table.
4. **HomeCheckFlag**: is a flag to ensure that data was copied correctly from original DB to Home Page DB.
5. **SearchCheckFlag**: -same as above- there is a flag to ensure that data was copied correctly from original DB to search Page DB.
6. **SectionCheckFlag**: flag to ensure that data was copied correctly from original DB to sections Page DB.
7. **ArchiveCheckFlag**: flag to ensure that data was copied correctly from original DB to Archive Page DB.
8. **CommandType**: an integer number that has been denoted to type of query.
9. Date: event date.

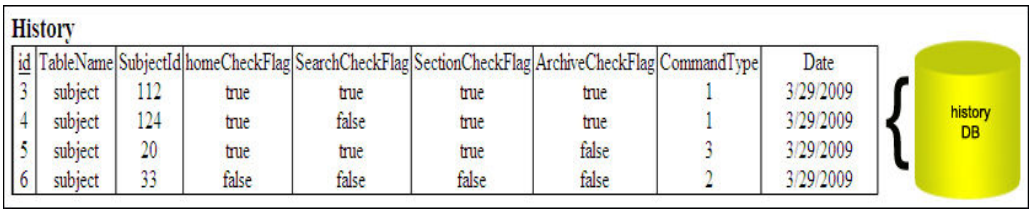


Figure 3.8 History table in history DB

When any changes made to the data in the original Database Instance, DB trigger will be lunched. And saves all information in history DB.

Table name and ***subject Id*** used to specify which DB table instance was changed. This information make consistency thread later able to copy data from original DB to other DBs.

Home, search, section and archive check flags. The consistency thread uses this information to ensure that data write from original DB to Sub DBs was going correctly (true =write correct / false= write is failure).

Command Type: consistency threads use it to know which query type was occurred (insert =1 / update=2 / delete=3).

EXAMPLE: if the user updates 1324th article by website control panel. Then the scenario would be shown as follows:

1. Update the article in original DB.
2. Save information in history DB(**Figure 3.9**)

History

id	TableName	SubjectId	homeCheckFlag	SearchCheckFlag	SectionCheckFlag	ArchiveCheckFlag	CommandType	Date
6	subject	**1324**	false	false	false	false	2	3/29/2009

Figure 3.9 Example of history table.

3. In history table, insert "subject" as table name changed his instance. Then save number of article in subjectId=1324, then insert default value "false" in checks flags. Then insert number 2 in command type column to denote update query, finally insert event date.

3.3.3. CONSISTENCY THREAD POLICY

1. When consistency thread runs, the first step is open history DB connection and read history table with order by command type column .because the priority in query is to "insert" query then "update" then "delete" (**figure 3.10 Step 1**).
2. Determined command type in order to know which DB query of system will use to make data the same. For each query type, special functions have code to ensure copy data from original DB to other DBs (Step 2).
3. Read data from original DB and send this data to next function (step 4) to execute query type. The Processes have functions to each command type even if it is "insert","delete" or "update" query from original DB to other sub DB.

4. In Step 5 checks if query execution correctly. If incorrect execution happened then query run “wait thread” (step 8) to try later. If correct execution, go to next process.
5. In Step 6 a process to make DB check flag true if process success to ensure that copy data correct and complete (**Figure 3.8**).
6. Step 7 process ensures that copy data for all subs DBs was correctly.
7. Step 9 ensures that delete of data was successful copy from original DB to other DBs.
8. Step 10 goes to next row in history DB. If there are no rows then end thread execution.

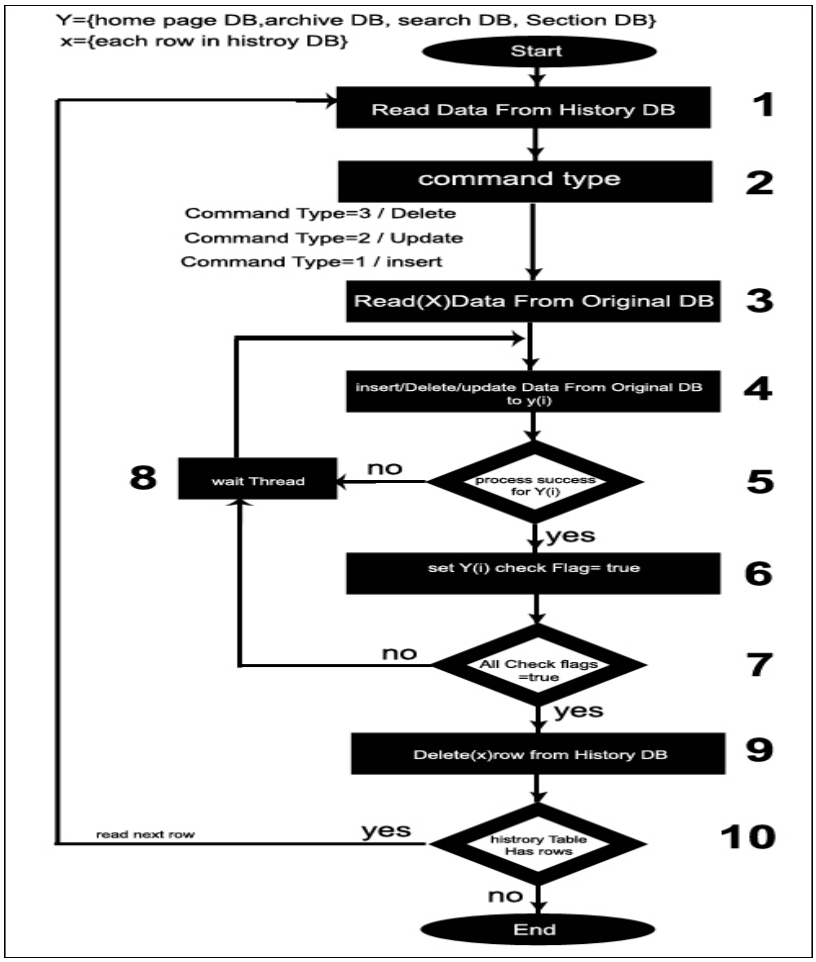


Figure 3.10 consistency thread policy model

3.4. MODEL MEASUREMENT

By taking a real life example of high count of hits as news website, it made a practical measurement to www.AmmonNews.net website by a set of CPU usage tests.

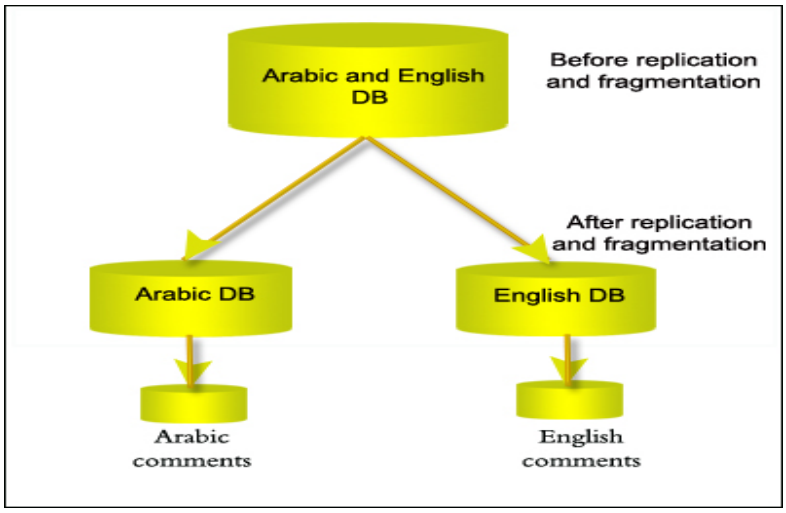
www.Ammonnews.net specification:

- Website with both Arabic and English languages
- One of the most visited websites in Jordan and one of 17 websites in Jordan measured by alexa.com
- About 270,000 average daily hits
- Arabic website has about 37,000 articles with more than 1,500,000 comments
- English website has about 2000 articles with more than 20,000 comments
- 2GHZ X 8 CPU, and 4 GB RAM

In this experiment, need to be assured that all the other factors on the server is not interfering with the CPU time readings, terminating unnecessary programs process such as automatic backup system and antivirus update schedule.

Knowing the various hits result along one week, it was found that Sunday is the highest day for hits and Friday is the least, which required us to implement the test twice per day.

This experiment focuses on two tables (subject and comments) because of the large number of records as mentioned in the www.ammonnews.net specification above (Figure 3.11)



3.11 General test

The experiment took place for one hour for each scenario as follows:

1. **Scenario one:**

By using central DB for both Arabic and English containing subject and comments tables. Also directing all the requests to the central DB using connection strings [21], so we measured the CPU usage readings (as in testing procedure below) for central DB without any fragmentation (Figure 3.12).

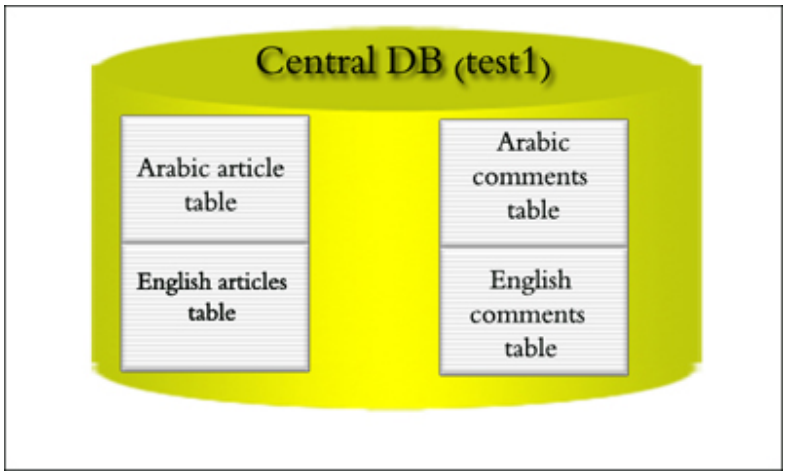


Figure 3.12 Central DB for www.ammonnews.net

The results is as shown in table 3.1

Table 3.1 Scenario one result

	Saturday	**Sunday**	**Monday**	**Thursday**	**Friday**
Daily hits	205,430	324,510	288,150	265,320	123,141
CPU usage without fragmentation DB(scenario 1)	19.140%	31.487%	27.430%	25,459%	9.989%

2. **Scenario two:**

First fragmented DB into 4 DBs (Figure 3.13) as follows:

a) Arabic subject table as DB1 with 37,000 records

b) English subject table as DB2 with 2,000 records

c) Arabic comments table as DB3 with 1,500,000 records

d) English comments table as DB4 with 20,000 records

Then directing all the requests to the 4 DBs using four connection strings instead one connection as in scenario 1**.** Then measuring the CPU usage after performing the fragmentation data and the results is as shown in Table 3.2

Table 3.2 Scenario 2 result

	Saturday	**Sunday**	**Monday**	**Thursday**	**Friday**
Daily hits	205,430	324,510	288,150	265,320	123,141
CPU usage with fragmentation DB(scenario 2)	17.432%	25.543%	22.926%	21.965%	9.022%

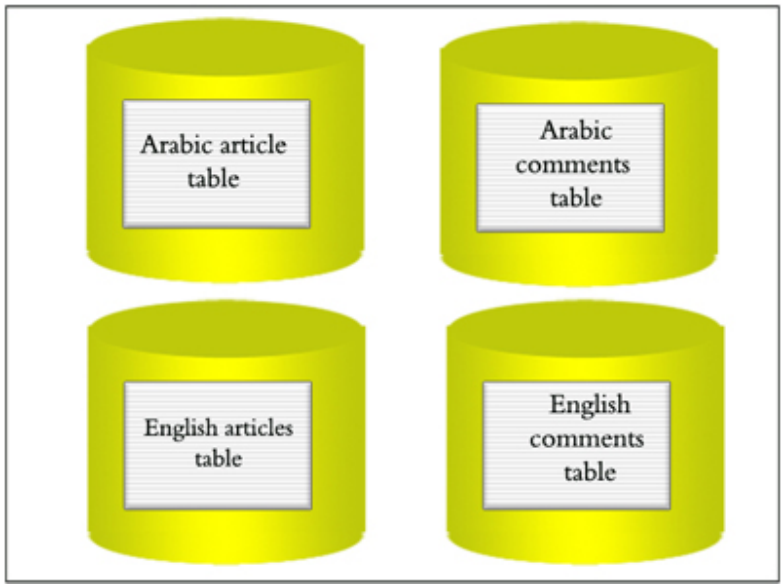


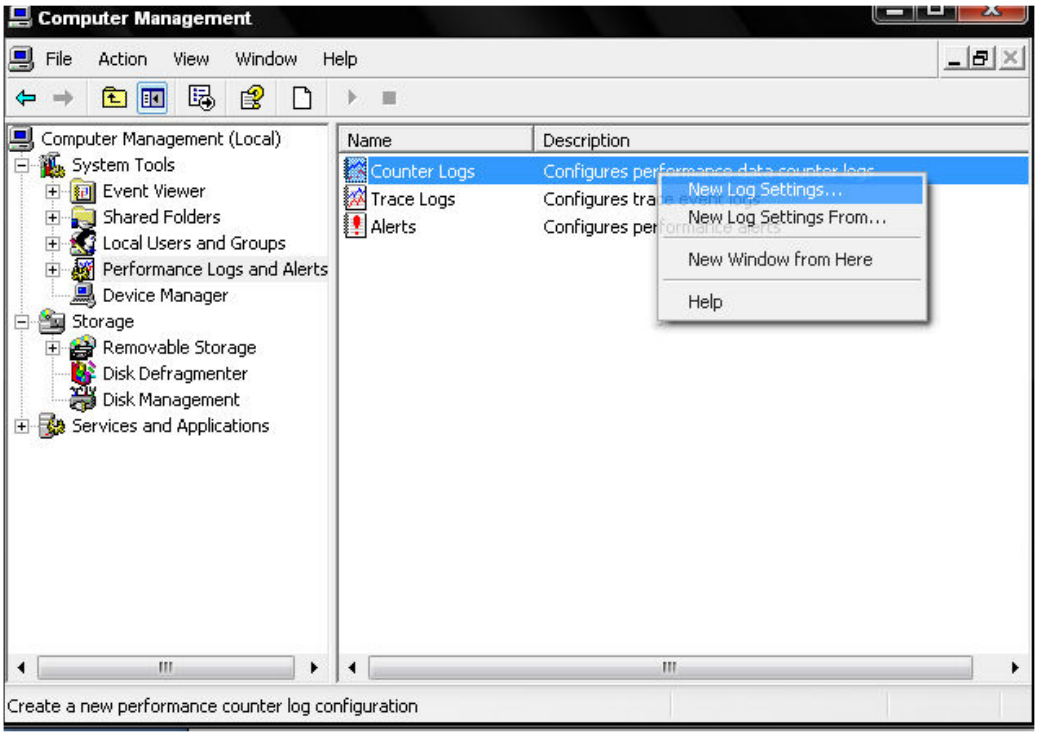
Figure 3.13 DB Fragmentations for ammon news

TESTING PROCEDURE

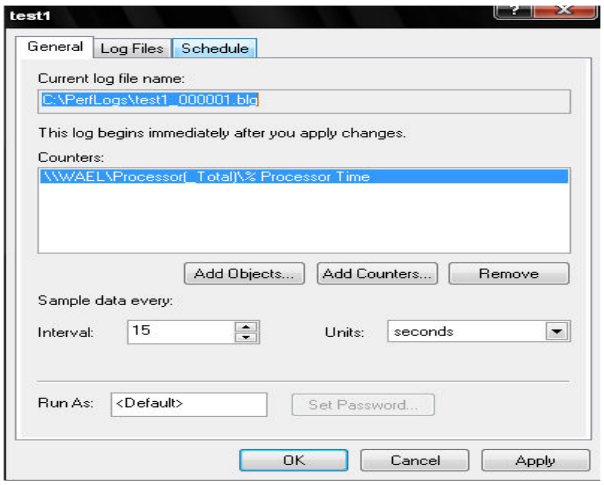
In testing will use the Microsoft **performance logs** tool in windows server 2003 which calculates average readings of CPU time, page/second and many of countering logs. The performance monitor, or system monitor, is a utility used to track a range of processes and give a real time graphical display of the results, on a Windows 2003 system. This tool can be used to assist you with the planning of upgrades, tracking of processes that need to be optimized, monitoring results of tuning and configuration scenarios, and the understanding of a workload and its effect on resource usage to identify bottlenecks. Bottlenecks can occur on practically any element of the network and may be caused by a malfunctioning resource, the system not having enough resources, a program that dominates a particular resource. In fact, 40% network utilization is considered a bottleneck. Using performance will help to identify these bottlenecks and allow you to take action [32].

1- Adding counters

Open Microsoft management console (MMC) then select performance logs tool and choose Add CPU processor from Counters list. The Add Counter steps shown below (Figures 3.14 and 3.15)



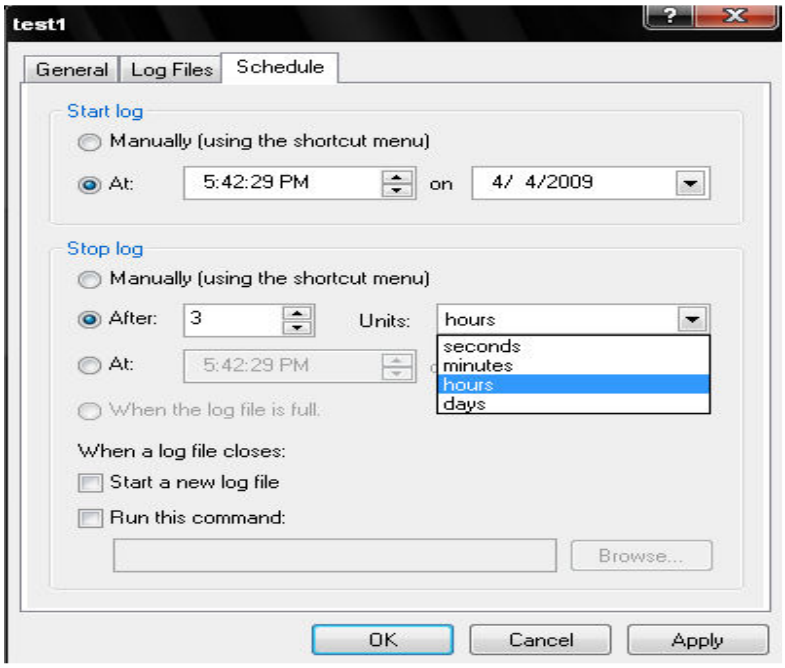
3.14 Adding a counter (step 1)



3.15Adding a counter (step 2)

2- Using the monitor for CPU usage vs. performance.

Figure 3.16 shows how to add a schedule for test to be run at a particular time.



3.16Adding a counter (step 3)

For example Figure 3.17 displays graph of CPU activities that took a place within two minutes. The green line represents the CPU usage per second; the other lines display several measurement factors such as main memory space with time.

3.17 Adding a counter (step 5)

We will monitor the CPU usage for www.ammonnews.net server for five days by comparing between daily hits and CPU usage. The table 3.3 shows the result of CPU usage with daily hits before and after the use of fragmentation data in ammonnews.net DB.

Table 3.3 CPU usage with daily hits

	Saturday	Sunday	Monday	Thursday	Friday
Daily hits	205,430	324,510	288,150	265,320	123,141
CPU usage without fragmentation DB(scenario 1)	19.140%	31.487%	27.430%	25,459%	9.989%
CPU usage with fragmentation DB(scenario 2)	17.432%	25.543%	22.926%	21.965%	9.022%

Below is the chart of the resulting readings (Figures 3.18 and 3.19)

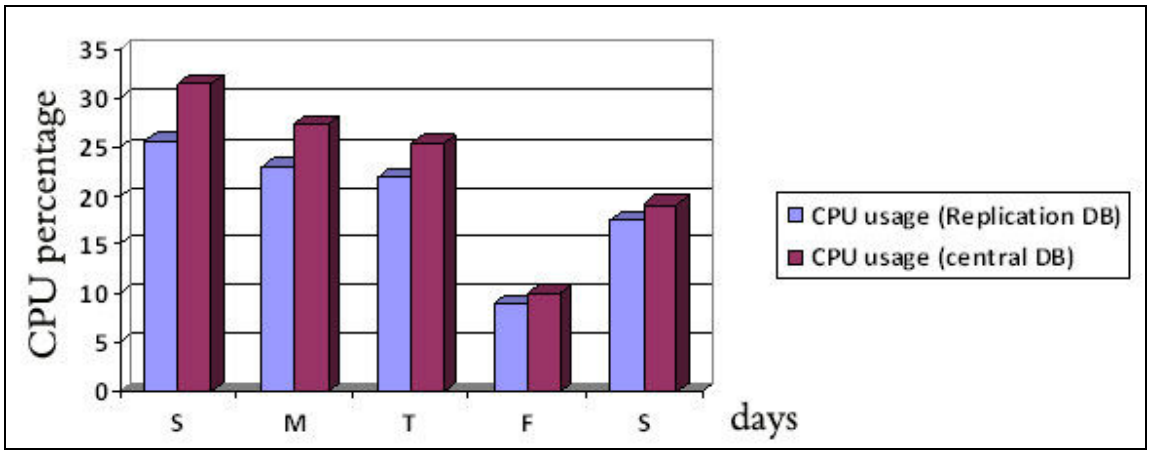


Figure 3.18 CPU usage comparisons

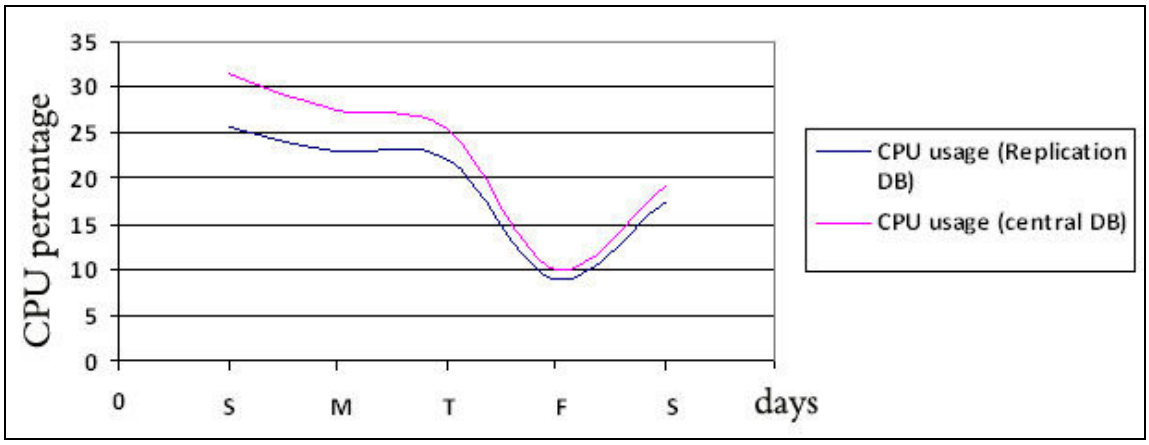


Figure 3.19 CPU usage comparisons

Take Sunday as an example where the hits are very high, so the CPU usage with the presence of fragmentation is 17,432%, unlike the absence of fragmentation which is 19.140% for the same day.

Note that the variance of CPU usage time between using fragmentation or not is decreasing in case of low day time hits.

The result showed that if fragmentation is used it will decrease CPU usage time and allows to select Web servers with low cost consideration.

TESTING OBSTACLES

1. Persuasion of the website owner to let use the website resources in order to perform the experiment
2. The risk of system unexpected critical failures
3. The measurements may lack the trueness because of other programs and processes affecting the measured CPU usage.
4. For the testing specimen it is needed to turn off admin pages (control panel)to increase the realty of the testing environment which disables update website contents.

CONCLUSIONS

1. The assumed fragmentation process decreases CPU usage especially when the DB size is enormous with high server hits.
2. The results returns no big difference on CPU usage during low hits time especially on Friday and Saturday.
3. For the case of fragmentation data scenario, it is capable to reduce the resource needed (server cost) in hosting.
4. If CPU usage decreased then increase the response time.

CHAPTER 4

FUTURE WORK AND CONCLUSIONS

4.1. CONCLUSIONS

The proposed model 1-designed 2-built 3-evaluated prototype of a fragmentation data in news websites to solve fragmentation problem in shared hosting, and avoid using central DB in big system which comprised the centerpiece of scalability service architecture for dynamic applications. By using prototype service to scale two benchmark applications (fragmentation and consistency data) – for news application increased the scalability by a factor of at least 2. Our thesis mainly based on how to use programming language to ensure DB consistency and how to decrease the number of records to the minimum that could be reached in news database by using DB fragmentation . Thesis way suggests replicating DB for each active page in website with keeping consistency for all DB replicas such as home page DB containing the recent news.

4.2. FUTURE WORK

There are many ideas to improve fragmentation technique for news website as Future work, we mention:

1. Comparing the proposed model for designing and adapting the web content with some other models such as: web services and e-commerce system.
2. Implementation of the proposed model for designing and adapting full system for news website by using DB fragmentation and refinement fragmentation data.
3. Adjust the suggested model and enhancing its features by adding other cashing technique evaluation at the end of design iterations.
4. Develop complete libraries for DB fragmentation by using programming language such as c# to support shared hosting (to save money).
5. Another interesting direction would be to analyze if and how possible is to define all the context factors as well as the fragmentation automatically.

REFERENCES

[1] [A.Caloiaro, Database Replication Techniques, College of Arts and Science at Stetson University DeLand, Florida,2007]

[2] [B. Fitzpatrick. Inside LiveJournal's backend, or holy hell that's a lot of hits! Presentation at the O'Reilly Open Source Convention, 2004]

[3] [C. Olston, A. Manjhi, C. Garrod, A. Ailamaki, B. Maggs, and T. Mowry. "A scalability service for dynamic web applications". In Proc. Conf. on Innovative Data Systems Research, pages 56–69, Asilomar, CA, USA, 2005]

[4] [Grid Database Design investigates the origin, background, and components of this new computing model by Tom Nadeau, 2007]

[5] [I. Lungu, T. Vatuiu,A. G. Fodor ,FRAGMENTATION SOLUTIONS USED IN THE PROJECTION OF THE DISTRIBUTED DATA BASE SYSTEMS ,2004]

[6] [I. Lungu, C. Bodea, G. Badescu, C.Ionita, Data bases. Organization, Design and Implementation, Ed. ALL, 1995]

[7] [J. Challenger, P. Dantzig, A. Iyengar, and K. Witting.A fragment-based approach for efficiently creating dynamic web content. ACM Transactions on Internet Technologies, 5(2):359–389, 2005]

[8] [K. Amiri, S. Park, R. Tewari, and S. Padmanabhan.DBProxy: A dynamic data cache for Web applications. In Proc. Intl. Conf. on Data Engineering, pages 821–831, 2003]

[9] [K. Amiri, S. Park, R. Tewari, and S. Padmanabhan. DBProxy: A dynamic data cache for Web applications. In Proc. Intl. Conf. on Data Engineering, pages 821–831, 2003]

[10] [KEMME, B., BARTOLI, A., AND BABAOGLU, ¨O . Online reconfiguration in replicated databases based on group communication. In Proceedings of the International Conference on Dependable Systems and Networks (DSN), IEEE Computer Society, pp. 117–130 ,2001]

[11] [M. Rabinovich, Z. Xiao, and A. Agarwal. Computing on the edge: A platform for replicating internet applications. In Proc. Intl. Workshop on Web Content Caching and Distribution, pages 57–77, Hawthorne, NY, USA, 2003]

[12] [N. Bhatti, A. Bouch, and A. Kuchinsky. Integrating user-perceived quality into web server design. In Proceedings of the 9th World-Wide Web Conference, pages 1–16, Amsterdam, The Netherlands, 2000]

[13] [S.B.Navathe,a mixed fragmentation methodology for initial distributed DB design,1995]

[14] [S. Sivasubramanian, G. Pierre, M. van Steen, and G. Alonso. GlobeCBC: Content-blind result caching for dynamic web applications. Technical Report IR-CS-022, Vrije Universiteit, Amsterdam, The Netherlands, 2006]

[15] [Wen-Syan ,Li Oliver Po, Wang-Pin Hsiung , K. Selc¸uk Candan, Divyakant Agrawal. Acceleration of Data Center-Hosted Distributed Database-driven Web Applications, Association for Computing Machinery, 2002]

WEB REFERENCES

[16] ACID - Atomicity, Consistency, Isolation, Durability? - Information Systems Essay
URL:http://www.freeonlineresearchpapers.com/acid-atomicity-consistency-isolation-durability

[17] Akamai EdgeSuite.
URL: http://www.akamai.com/en/html/services/edgesuite.html.

[18] A Quick Tour of a Typical Web Site
URL: http://www.centerspan.org/tutorial/www.htm#tour

[19] ASP.NET (2008)
URL: http://en.wikipedia.org/wiki/ASP.NET

[20] An Introduction to Data Fragmentation in IDS(2002)
http://www.ibm.com/developerworks/data/zones/informix/library/techarticle/0207noronha/0207noronha.html

[21] Connection String Syntax
URL: http://msdn.microsoft.com/en-us/library/ms722656(VS.85).aspx

[22] Database Engine XML Enhancements
URL: http://msdn.microsoft.com/en-us/library/ms170809.aspx

[23] Dedicated hosting.
URL: http://www.bytehouse.co.uk/hosting_services/dedicated_hosting/definition-of-dedicated-hosting.php

[24] History of the Internet
URL: http://www.netvalley.com/archives/mirrors/davemarsh-timeline-1.htm

[25] I3 UPDATE / Entovation International News
URL: http://www.skyrme.com/updates/u59_f2.htm

[26] 55 Keys to Improving Your Rankings(2007)
URL: http://www.searchengineguide.com/searchbrief/senews/009767.html

[27] Microsoft Visual Studio(2008)
URL: http://en.wikipedia.org/wiki/Visual_Studio_.NET_2005

[28] Microsoft SQL Server(2008)
URL: http://en.wikipedia.org/wiki/Microsoft_SQL_Server

[29] shared hosting for websites(2001)
URL:http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci514044,00.html

[30] Shared Hosting – Definition.
URL: http://www.real-knowledge.com/shared-hosting.htm

[31] Web content
URL: http://en.wikipedia.org/wiki/Web_content#cite_note-0

[32] Windows 2003 Performance Monitor(2004)
URL:http://www.windowsnetworking.com/articles_tutorials/Windows_2003_Performance_Monitor.html

[33] Zona Research(1999).
URL: http://www.zonaresearch.com/.
URL: http://ecommerce.hostip.info/pages/942/Site-Response-Time.html

APPENDIX

Curriculum Vitae

WAEL SULIMAN AL-MURAFI
Computer Science

Personal Information

Nationality:	Jordanian
Resident of:	Jordan – Amman
Date of Birth:	April/25/ 1985
Gender:	Male
Marital Status:	Single

Connection Information

Address:	Jordan – Amman
Mobile Phone:	+962785908843 OR +962799064263
E-Mail:	Wael_almarafi@yahoo.com

Education

University name	AL-Hussein Bin Talal University
Degree:	Bsc in Computer Science
Uni Average	81.6%
Tawjeehi Average	81.8 (Scientific)

Programming Languages

- C# Web Applications- ASP. NET (VS2005).
- C# windows Applications (VS2005).
- HTML , JAVA SCRIPT

Databases

- Oracle10g DB.
- Microsoft SQL server 2005.

Websites experiences

- http://ammonnews.net/
- http://jordantimes.net
- http://en.ammonnews.net
- http://shabab.addustour.com.
- http://www.nepco.com.jo
- http://www.cigre.org.jo
- http://www.aqarac.com
- http://www.areejnews.net
- http://www.speedmethods.com
- http://nashamanews.com

Software Packages

- Photoshop cs.
- Sothink Glanda 2005 for web Applications.
- Sothink DHTML Menu.

Professional Experience:

Job Title	System analyst and Developer.
Organization	National Electric Power Company (http://www.nepco.com.jo)
Date	[August / 2007] – [present].
Location	**Jordan, Amman [HQ offices].**

Job Details:	• Have Developed two websites for NEPCO company with **content management system**(CMS):
	  ○ http://www.nepco.com.jo
	  ○ http://www.cigre.org.jo
	• Have used VS2005, SQL2005, oracleDB-10g and some knowledge in VS2003.

- Responsible administrating those two sites and training user to publish and edit data.
- Update company website if there is any change.
- Develop some windows application for NEPCO Company.
- Analysis and develop finance system for CIGRE conference.

Other certification

Certification :

- Seven honoring certification from Alhussain Ben Talal University.