An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs)

Ali A. Amer a,*, Adel A. Sewisy b, Taha M.A. Elgendy c

a Computer Science Department, Science College, Taiz University, Yemen
b Computer Science Department, Computers and Information Faculty, Assuit University, Egypt
c Mathematics Department, Science College, Assuit University, Egypt

* Corresponding author.
E-mail addresses: aliaaa2004@yahoo.com, aliaaa2004@gmail.com (A.A. Amer).

Abstract

With the substantial ever-upgrading advancement in data and information management field, Distributed Database System (DDBS) is still proven to be the most growingly-demanded tool to handle the accompanied constantly-piled volumes of data. However, the efficiency and adequacy of DDBS is profoundly correlated with the reliability and precision of the process in which DDBS is set to be designed. As for DDBS design, thus, several strategies have been developed, in literature, to be used in purpose of promoting DDBS performance. Off these strategies, data fragmentation, data allocation and replication, and sites clustering are the most immensely-used efficacious techniques that otherwise DDBS design and rendering would be prohibitively expensive. On one hand, an accurate well-architected data fragmentation and allocation is bound to incredibly increase data locality and promote the overall DDBS throughputs. On the other hand, finding a practical sites clustering process is set to contribute remarkably in reducing the overall Transmission Costs (TC). Consequently, consolidating all these strategies into one single work is going to undoubtedly satisfy a massive growth in DDBS influence. In this paper, therefore, an optimized heuristic horizontal fragmentation
and allocation approach is meticulously developed. All the drawn-above strategies are elegantly combined into a single effective approach so as to an influential solution for DDBS productivity promotion is set to be markedly fulfilled. Most importantly, an internal and external evaluations are extensively illustrated. Obviously, findings of conducted experiments have maximally been recorded to be in favor of DDBS performance betterment.

Keywords: Computer science

1. Introduction

In this modern time of rapidly-paced data, and information, feeds and an ever-growing connection with data usage through constantly-advanced technology, it is therefore being more than an obligation for all DDBS-based organizations/individuals to search for creatively-designed DDBS of highly-appreciated throughputs. Nevertheless, in DDBS, a well-designed system is still considerably demanding task as it is of continuous uncontrollable urge to reach the satisfactory level of DDBS performance. The intended level of DDBS throughputs, however, could be simply achieved through: proposing a proper data fragmentation; presenting precise data allocation, and designing practical algorithm for sites clustering. As a matter of fact, these techniques combined were successfully proven to be super effective in boosting DDBS productivity (Sewisy et al., 2017; Hababeh et al., 2015). Therefore, this work comes with the aim of introducing a newly-designed approach by combining them all into a single efficacious work. Off these techniques, (Abdalla, 2014) is being selected and optimized to be then re-introduced as an optimized fragmentation technique for this work. Additionally, this optimized technique is being finely integrated with proposed sites clustering algorithm and cost model for data allocation. It is worth mentioning that the obtained results confirm emphatically that the proposed (optimized) approach outperforms (Abdalla, 2014) to the great extent, and proves to be a potential progress not just in lessening TC substantially, but also in promoting DDBS performance significantly. To sum up, contributions along with motivations of this work are clearly featured as follows;

1. Objective function of (Abdalla, 2014) had not delved communication costs into distributed query costs which leads to its being inefficient on either mitigating costs of communications, as distributed query being processed, or even evaluating the whole technique. Communication costs, however, is the prime rational for which (Abdalla, 2014) and also present work basically come to find a practical solution capable of minimizing these costs to the most greatest extent. In other words, practically reducing communication costs has been the major concern of present work providing that these costs are being carefully reflected in the intrinsically-amended objective function. Moreover, having communication costs involved within objective function would satisfy: (1) the
reflection of actual, or at least near-optimal, reality of Transmission Costs (TC) (Sewisy et al., 2017); (2) the precipitation of accurately evaluating overall DDBS performance as distributed queries processed and maximally verified rate of data locality growth and DDBS throughput as a result.

2. On contrary to (Abdalla, 2014), to further minimizing communication costs, present work aims at delicately proposing clustering algorithm for sites as sites clustering has significantly proven to be considerably efficient on lessening communication costs (Sewisy et al., 2017; Hababeh et al., 2015; Abdel Raouf and Badr, 2017).

3. Data allocation has widely proven to be super effective factor in DDBS productivity promotion, specifically as it has been done appropriately. So, unlike (Abdalla, 2014), data allocation (including replication over clusters/sites) has neatly been made using precise mathematical model with considering communication costs between both sites and clusters of sites. This model is meant to be applicable in both works while its being completely complied with present work’s concepts including sites grouping.

4. In full contrast with (Abdalla, 2014), in which just data replication scenario for data is adopted and data replication was bound to be permanently met, present work seeks to adopt replication when it is just necessary, and non-replication scenario as well for both works which would contribute in avoiding unnecessary replication of demoralizing effects.

5. In (Abdalla, 2014), technique’s evaluation was not provided to measure effectiveness of proposed technique as distributed queries under processing. Present work, however, seeks to substantively draw evaluation process for both works while strictly maintaining their circumstances. This evaluation has been expressively conducted with considering the precisely-modified objective function in mind.

6. To prove proposed concepts of the present work, many different experiments under varied circumstances have been conducted that an internal and external evaluations are extensively drawn in self-explanatory frame. Both works are being exposed on TC function to measure their quality and grade DDBS performance accordingly.

The rest of this paper is elegantly organized as follows; section (2) profoundly covers earlier works, which are closely relevant to this work. In section (3), technique’s methodology, including architecture, is presented. Site clustering algorithm is stated in section (4). In section (5), the proposed data allocation and replication models are elaborately given. In section (6), pseudo code algorithm is briefly provided. In section (7), experimental results are extensively drawn. Section (8) illustrates works’ evaluation thoroughly, and gives comparative theoretical study. Finally, conclusions and future work directions are included in section 9.
2. Related work

As per earlier studies of DDBS design, a remarkable progress is being recorded in form of successive steps to improve DDBS rendering. As one important aspect of this progress, several Horizontal Fragmentation (HF) techniques/methods have been presented. As example, in (Ceri et al., 1982; 1986) used a min-term predicate as a measure to split relations so that primary HF was produced assuming that previously-specified predicates set satisfied properties of disjoint-ness and completeness. On the same line, (Zhang and Orlowska, 1994) draw two-phase HF method. In first phase, relations were fragmented by primary HF using predicate affinity and bond energy algorithm. Secondly, relations were further divided using derived HF. For initial stage of DDBS design, (Surmsuk and Thanawastien, 2007) proposed a Create, Read, Update and Delete Matrix (CRUD) so that attributes used as rows of CRUD and applications locations used as columns. Fragmentation and data allocation was considered together as well. (Amer and Abdalla, 2012) Presented a cost model to find an optimal HF in which two scenarios for data allocation were considered so that no supplemental complexity was added to data placement. In follow-up work, this model was further extended in (Abdalla, 2014) and mathematically shown to be an effective at reducing costs of communication. Experimental results, performance analysis and model practicality were not provided, though.

On the other hand, a hybridized fragmentation was proposed in (Harikumar and Ramachandran, 2015) to reduce database access time based on subspace clustering algorithm. Data fragments were generated with respect to tuple and attribute patterns that the closely correlated data were assembled together. In the meantime, (Hauglid et al., 2010) evolved a decentralized approach for dynamic Table fragmentation and allocation in DDBS (DYFRAM) to maximize data locality based on recorded access history. Moreover, approach feasibility was experimentally demonstrated. By the same token, (Abdel Raouf and Badr, 2017) gave an enhanced system to perform initial-stage fragmentation and data allocation along with replication at run time over cloud environment. Site clustering was addressed as well to enhance DDBS performance through increasing local accesses.

Meanwhile, (Lin et al., 1993) addressed data allocation problem in DDBS through developing two algorithms with the aim of lessening the entire costs of communication. On the same page, a model to approach queries behavior in DDBS was drawn in (Huang and Chen, 2001). In terms of reducing communication costs, two heuristic algorithms were then given to find a near-optimal allocation scenario. This algorithms was proven to be close enough from being an optimal compared to (Lin et al., 1993). In (Tâmbulea and Horvat, 2008), a dynamic data allocation method was presented to decrease transmission cost with considering database catalog as the only storing place for required data. In (Amita
Goyal Chin, 2002), as a new of its kind, a partial data reallocation and full reallocation heuristics were approached to minimize costs and keep complexity controlled. Moreover, to find an optimal data allocation technique, (Abdalla et al., 2014) presented a non-replicated dynamic data allocation algorithm (POEA). Actually, POEA was originally sought to integrate some previously-proposed concepts used in its earlier peers including (Mukherjee, 2011), (Dejan Chandra Gope, 2012), on its turn, developed a dynamic non-replicated data allocation algorithm (named, NNA). Data reallocation was done with respect to the changing pattern of data access along with time constraints.

By the same token, (Singh, 2016) approached data allocation framework for non-replicated dynamic DDBS using threshold algorithm (Ulus and Uysal, 2007), and time constraint algorithm (Singh and Kahlon, 2009). Furthermore, this work was shown to be most efficient in terms of long-term performance than threshold algorithms when access frequency pattern changes in rapid paces. Nevertheless, (Kumar and Gupta, 2016) evolved an extended allocation approach capable of dynamically assigning fragment in redundant/non-redundant DDBS. The problem of having more than one site qualified to have data was also discussed. Lastly and most importantly, in (Wiese, 2015), Data Replication Problem (DRP) was formulated to have precise horizontal fragmentation of overlapping fragments. This work aimed at placing N-copy replication scheme of fragments into M distinct sites with ensuring that overlapping is being precluded. Most of all, replication problem was looked at as an optimization problem to achieve intended goal that fragments’ copies and sites kept minimized. In follow-up work, (Wiese, 2015) was further being extended in (Wiese et al., 2016) and DRP problem was then re-formalized as an integer linear program. Runtime performance was analyzed and data insertion and deletion were addressed as well.

3. Methodology

This work is ultimately driven by different quests with Transmission Costs reduction has been the top primacy. On the other hand, introducing a mathematically-based data allocation, integrating non-replication scenario for data allocation as well as proposing site clustering algorithm have been further aspects of evolving this approach of this paper. Therefore, motivations could be presented in the next a few lines.

3.1. Motivations

Motivations of this work are delicately identified to be either general or particular motivations as follows;
3.1.1. General motivations

After a comprehensively-done investigation for related work, in context of horizontal method of relations in relational distributed database, and to best of our acknowledge that there has never been any single work seeks to integrate several communication costs-reducing techniques (horizontal fragmentation, data allocation and replication, site clustering, mathematical models, etc) into one single work. Therefore, this work meant to be promising, leading and distinguishable approach (able to be valid as mathematical-based general solution for most problems of DDBS performance). In the sense that all these techniques are set to be combined together in purpose of finding creative sustainable solution for DDBS performance improvement. Moreover, this work has successfully been shown to be highly promising either theoretically, experimentally or mathematically. On the other hand, it is worth referring that (Hababeh et al., 2015; Abdel Raouf and Badr, 2017) are spotted as they were proposed so that all these technique were combined together, but these approaches were implemented and designed in a completely different circumstances of our present work of this paper as well as they were meant to solve a particular cases of telecommunication databases and cloud environment. Both of references were adequately cited on our paper, though.

3.1.2. Particular motivations

After a thoroughly-made investigation and carefully-raised questioning for all closely-relevant earlier studies in terms of DDBS performance enhancement, (Abdalla, 2014) has been found to be an interesting well-designed technique to be examined and significantly extended. The major purpose of (Abdalla, 2014) was focused on DDBS rendering promotion through decreasing communication costs. However, given the importance of this work’s objective function, intrinsic flaws has been recorded as this function had never considered communication costs, which is largely responsible for either DDBS performance boost or deterioration. Data allocation process also was not clearly elaborated as it was given merely theoretically. Moreover, no evaluation for technique was provided so that objective function could be identified as being effective. So, based on this flaws and as per this work’s drawn-above contributions, the present work of this paper is built on to close these gaps for the sake of producing super effective approach. In other words, those flaws and gaps make (Abdalla, 2014) to be somehow appearing unable of achieving the acquired goal of reducing communication costs as well as promoting DDBS performance, which it was originally intended to be satisfied in the first place. This claims, nevertheless, are significantly confirmed to be indisputable facts via discussion and evaluation presented in this work. To sum up, using (Abdalla, 2014) as essential and initial cornerstone, this work comes to fully and intrinsically perfect and optimize technique of (Abdalla, 2014).
### 3.2. Horizontal fragmentation model

In this work, fragmentation model is set out to be entirely depending upon predicates set, Pr [Pr₁, . . . ,Prₚ]. In its turn, these predicates are supposedly assigned to all (NA) attributes under consideration, A [A₁, . . . ,Aₙ]. While numerical attributes are assumed to have one of three states: (Prᵢ > Value₁), (Prᵢ < Value₂) or (Prᵢ = Value₃), alphabetical attributes are destined to have only one state: (Prᵢ = Alphabetical Value). On the other hand, for each attribute, retrieval and update frequencies would be extracted/given by Database Administrator (DBA) for all predicates in form of “Prᵢ.RFᵢ or Prᵢ.Rᵢ “and “Prᵢ.UFᵢ or Prᵢ.Uᵢ“.

Meanwhile, these attributes are observed to be constantly required by most-frequently used queries (Qs), which are supposedly released from several sites, and each query has its own RF/UF (or, R/U) frequency over data in each site. In the sense that if query (Q) is launched from several sites (M), that query would be treated as a different query in each site with different RF/UF frequency. These frequencies have to be precisely saved for all queries in Query Frequency Matrix (QFM). Based on these requirements, this work is set to substantially utilized fragmentation procedure drawn in (Abdalla, 2014) along with slightly-made modifications (Fig. 1) as follows:

1. Relations under consideration are set to be defined and their predicates are bound to be identified.

2. Individually, all most-constantly used queries that are observed to reach each relation would be kept and considered regardless of their type (either retrieval or update). Query Frequencies over sites, Retrieval and Update Frequencies of queries over data in all sites would carefully be drawn into QFM, QRM and QUM matrices respectively. Using these matrices along with fragmentation cost model, data fragmentation is set to be activated.

3. Based on Eq. (4) along with the mentioned-above matrices, Attribute Frequency Accumulation (AFAC) would be introduced.

![Fig. 1. The proposed Technique Architecture.](http://dx.doi.org/10.1016/j.heliyon.2017.e00487)
4. As per Eq. (5), Communication Cost Matrix (CSM) would be converted into Distance Costs Matrix (DCM) by applying Minimum Algorithm as presented in (Abdalla, 2014). Then, using Eq. (6), DCM is multiplied by AFAC to yield Total Frequencies of Attributes predicate Matrix (TFAM).

5. After that, TFAM is used for attributes individually to compute the entire pay (total) of access costs and then sort all attributes according to their pays.

6. Finally, among these attributes, attribute of highest pay would be selected to be Candidate Attribute “CA” by which fragmentation process is to be successfully conducted, Eq. (7).

### 3.3. Objective function

\[
TC_1 = \sum_{j=1}^{m} \sum_{i=1}^{m} \sum_{k=1}^{q} \left(1 - X_{kj}\right) \times (QRM_{kj}) \times F_{size} \times CMS_{ij}
\]  

(1)

\[
TC_2 = \sum_{j=1}^{m} \sum_{i=1}^{m} \sum_{k=1}^{q} \left(1 - X_{kj}\right) \times (QUM_{kj}) \times F_{size} \times CMS_{ij}
\]  

(2)

\[
TC_{total} = TC_1 + TC_2
\]  

(3)

It is worth mentioning that objective function is basically taken from (Abdalla, 2014) and intrinsically evolved to be best befitting present work’s circumstances so as to actual reality of transmission cost (TC) is being precisely reflected (Sewisy et al., 2017). While the first equation is set to be used to measure costs incurred as distributed retrieval queries are being processed, Eq. (2) is going to measure costs yielded as a result of performing distributed update queries running over DDBS. Transmission Costs in Total would therefore be accurately calculated with the use of Eq. (3). The effects of drawn objective function, however, are conspicuously illustrated in the demonstrated-below discussion section. Meanwhile, TC\(_1\) and TC\(_2\) (Eqs. (1) and (2)) represents Transmission costs in terms of retrieval operations and Transmission costs in terms of Update operations respectively. While CMS stands for either costs matrix between sites (CSM) or costs matrix between (Cn) clusters of sites (CCM). Finally, \(F_{size}\) stands for considered fragment’s size, and \(X_{ij}\) is binary variable drawn to indicate fragment’s allocation over sites.

### 3.4. Fragmentation cost functions

\[
AFAC_{jih} = \sum_{j=1}^{m} \sum_{i=1}^{m} \sum_{h=1}^{A} (RF_{jihp} \times QF_{ji} + UF_{jihp} \times QF_{ji})
\]  

(4)

\[
DCM_{ij} = \text{Min}(CMS_{ij}), 1 \leq i; j \leq m
\]  

(5)

\[
TFAM_{jh} = \sum_{j=1}^{m} \sum_{i=1}^{m} (AFAC_{jih} \times DCM_{ij}), 1 \leq i; j \leq m
\]  

(6)
\[ CA_h = \text{MAX} \left( \sum_{j=1}^{m} \sum_{h=1}^{A} \text{TFAM}_{jh} \right) \] (7)

Where QF, RF and UF are just abbreviations for elements of matrices of QFM, QRM and QUM respectively.

### 3.5. Site clustering algorithm

The presented algorithm of site clustering has precisely been designed based on the proved-to-be-efficient concept of Least Difference Value (LDV) proposed in (Sewisy et al., 2017) which was essentially used to cluster DDBS queries. In this work, however, clustering algorithm behaves differently from that of (Sewisy et al., 2017) which was basically done based on threshold values. Compared to proposed algorithm of this work, this threshold value-based algorithm seems (in some cases) to either minimize number of site clusters to inaccurate extent or maximize clusters to an excessive undesirable rang that in both cases bounds to adversely come at the expense of DDBS performance as shown in discussion section. Threshold-based algorithm is slightly shown to better behave than (Abdalla, 2014), though. Therefore, instead of using threshold algorithm to cluster sites in (Sewisy et al., 2017), LDV concept is fully utilized to initiate first clusters of sites using communication costs. After that, to keep clustering the remaining sites, the least average of communication cost between sites would be used as metric to delicately pull each site into its relative cluster (among those already initiated). Consequently, this clustering procedure is set to keep proceeding in such pattern to decide all sites’ belonging. Number of clusters, on the other hand, is subjected to nothing but behavior of algorithm. This algorithm is carefully drawn so that cluster of sites would be kept at minimum, though. Communication costs within and between clusters (CCM) thus are of key importance to be taken for data allocation and performance evaluation alike, particularly in non-replication scenario. Finally, as per (Ceri et al., 1982; Al-Sayyed et al., 2014), the cost matrix is assumed to be a symmetric between sites (and between clusters), and costs between the same sites are considered to be a zero or, Tables 1 and 2).

In this work, two experiments are separately conducted, and Tables 1, 2, 3 and 4 exhibit communication costs matrices, CSM, of two different network (four sites and six sites respectively) and their relative already-generated clusters. As a matter of fact, Table 2 is deliberately taken from (Sewisy et al., 2017) in purpose of confirming superiority of clustering algorithm of this work (Tables 3 and 4). This superiority has further been supported by results drawn in evaluation section.
3.6. Proposed data allocation and replication model

3.6.1. Problem description

Generally speaking, in DDBS, the optimal solution to boost DDBS performance is to intelligently fragment data, and delicately distribute them over cluster/site in where they are heavily reached (Ozsu and Valduriez, 2011). This problem, on the other hand, counts deeply on the complexity embedded in choosing cluster/site for targeted data. As matter of fact, one solution is believed to contribute highly in achieving intended performance; so that the number of update and retrieval accesses of each cluster/site for a specific data is being accumulated and considered for performing data assignment.

3.6.2. Data allocation requirements

Provided that having a set of N disjoint fragments $F = \{F_1, F_2, \ldots, F_n\}$ required by set of K queries $Q = \{Q_1, Q_2, \ldots, Q_k\}$, are to be assigned to a set of M network sites $S = \{S_1, S_2, \ldots, S_m\}$ which are grouped into Cn clusters $C = \{C_1, C_2, \ldots, C_{cn}\}$ in a fully connected network. Allocation model seeks naturally to find the optimal distribution of each fragment (F) over clusters, and on cluster’s own sites. Simply, allocation problem could be mathematically debriefed by a function from set of fragments to set of clusters/sites, Eq. (8).

Table 1. Communication Cost Matrix between Sites (CSM); Four Sites.

| Site/Site | S1 | S2 | S3 | S4 |
|-----------|----|----|----|----|
| S1        | 0  | 5  | 9  | 18 |
| S2        | 5  | 0  | 16 | 4  |
| S3        | 9  | 16 | 0  | 11 |
| S4        | 18 | 4  | 11 | 0  |

Table 2. Communication Cost Matrix between Sites (CSM); Six Sites.

| Site/Site | S1 | S2 | S3 | S4 | S5 | S6 |
|-----------|----|----|----|----|----|----|
| S1        | 0  | 10 | 8  | 2  | 4  | 6  |
| S2        | 10 | 0  | 7  | 3  | 5  | 4  |
| S3        | 8  | 7  | 0  | 3  | 2  | 5  |
| S4        | 2  | 3  | 3  | 0  | 11 | 5  |
| S5        | 4  | 5  | 2  | 11 | 0  | 5  |
| S6        | 6  | 4  | 5  | 5  | 5  | 0  |
3.6.3. Data allocation scenarios

Scenario 1 (Phase 1); (over clusters, replication adopted)

Data fragments are set to be individually assigned to all clusters of sites. Such procedure is believed to contribute overwhelmingly at decreasing TC and increasing data locality, chiefly as retrieval operations are outnumbered update operations.

Scenario 2 (Phase 1); (over clusters, no replication adopted)

Each fragment would be placed to cluster of highest access cost. Such mathematically-based process is shown to have undeniable positive effects on DDBS performance, specifically as update operations are outnumbered retrieval operations (Sewisy et al., 2017; Bellatreche and Kerkad, 2015)

Scenarios 1 and 2 (Phase 2); (over sites of clusters)

In each cluster, fragments are lined up to be placed into sites of each cluster as follows; firstly, like (Abdalla, 2014), a threshold would be tacitly calculated based on Average of Update Cost (AUC) and Average of Retrieval Cost (ARC) of each fragment. Then, whenever F’s AUC is higher than F’s ARC, the triggered fragment is to be assigned to site of highest update cost inside its relative cluster providing that cluster/site’s constraints have never been violated. On the other hand, as

Table 3. Communication Cost Matrix between Clusters (CCM); Four Sites.

| Cluster/Cluster | C1(S1S3) | C2(S2S4) |
|-----------------|----------|----------|
| C1(S1S3)        | 0        | 5        |
| C2(S2S4)        | 5        | 0        |

Table 4. Communication Cost Matrix between Clusters (CCM); Six Sites.

| Cluster/Cluster | C1(S2S6) | C2(S1S4) | C3(S3S5) |
|-----------------|----------|----------|----------|
| C1(S2S6)        | 0        | 3        | 5        |
| C2(S1S4)        | 3        | 0        | 3        |
| C3(S3S5)        | 5        | 3        | 0        |
constraints violation being recorded, fragment would automatically be placed into site of the next highest AUC inside the same cluster. On the contrary, whenever F’s ARC is greater than F’s AUC, that fragment would be replicated over all requesting sites as done in (Abdalla, 2014). As a result of strictly following this procedure, the ideal case is set to be satisfied and DDBSs’ response time, disk access and overall performance are bound to have significantly got reinforced, as shown in evaluation section.

3.6.4. Data allocation cost functions

\[
TFRS = \sum_{l=1}^{m} \sum_{f}^{f} \sum_{i=1}^{m} \sum_{q}^{q} \frac{QF_{ik} \times RF_{ki} \times CSM_{ij}}{C3}
\]  
(9) 

\[
TFUS = \sum_{l=1}^{m} \sum_{f}^{f} \sum_{i=1}^{m} \sum_{q}^{q} \frac{QF_{ik} \times UF_{ki} \times CSM_{ij}}{C3}
\]  
(10) 

\[
TFRUS = \sum_{l=1}^{m} \sum_{f}^{f} TFRS_{ij} + TFUS_{ij}
\]  
(11) 

\[
TFRC = \sum_{l=1}^{Cn} \sum_{f}^{f} \sum_{i=1}^{m} \sum_{q}^{q} \frac{QF_{ik} \times RF_{ki} \times CCM_{ij}}{C3}
\]  
(12) 

\[
TFUC = \sum_{l=1}^{Cn} \sum_{f}^{f} \sum_{i=1}^{m} \sum_{q}^{q} \frac{QF_{ik} \times UF_{ki} \times CCM_{ij}}{C3}
\]  
(13) 

\[
TFRUC = \sum_{l=1}^{P} TFRC + TFUC
\]  
(14) 

In brief, Eqs. (9) and (10) are used to compute Total Frequency of Retrieval and Update activities (TFRS and TFUS) over network sites in non-replication scenario, on the other hand, Eq. (11), as the summation of TFRS and TFUS, would be used to find matrix by which fragment is to be assigned to sites based on maximum cost concept. In contrary to above-given equations, whilst flowing the same pattern of Eqs. (9), (10) and (11), Eqs. (12), (13) and (14) are used to distribute fragments over clusters of sites.

For Data Replication, data replication model, which is drawn in (Sewisy et al., 2017) based on (Wiese et al., 2016), is expertly utilized as it has been proven to have huge positive impact on overall DDBS performance. However, this model is slightly modified to have it skilfully complied with proposed work of this paper. It worth noting that X_{ik} points to fragment F_{i} located in cluster C_{k} or (site S_{k}), and Y_{k} indicate to that cluster/site M already in use. Thus, an integer linear program (ILP) to represent this problem presented as follows;

\[
\text{Minimize } \sum_{k=1}^{m} y_{k}
\]  
(15)
\[ \sum_{i=1}^{N} X_{ik} = 1k = 1, \ldots, m \]  
(16)

\[ \sum_{k=1}^{m} C_i X_{ik} \leq C_{jk}, i = 1, \ldots, m \]  
(17)

\[ X_{ik} \in \{0, 1\}, k = 1, \ldots, m; i = 1, \ldots, n \]  
(18)

\[ Y_{ik} \in \{0, 1\}, k = 1, \ldots, m; i = 1, \ldots, n \]  
(19)

To keep number of clusters/sites at minimum, Eq. (15) is used. For non-replicated scenario, Eq. (16) seeks to control fragment replica placement as each fragment set to be given to only one single cluster. However, to maintain Cluster/Site constraints, Eq. (17) aims at evading capacities being overflowed. Finally, Eqs. (18) and (19) indicate that variables are binary (0, 1) linear program.

Lastly, since two experiments have exceptionally been performed in a clear demonstration to show work’s mechanism and superiority; tables 5 and 6 describe two constraints of sites (for both considered network), represented in virtual capacity (in bytes), limit of fragments allowed to be kept at each site.

**Table 5.** Network Sites with Constraints (Four Sites); where site’s capacity calculated hypothetically in bytes.

| S # | Capacity (C) in byte | Fragment Limit (FL) |
|-----|----------------------|---------------------|
| S1  | 1000                 | 6                   |
| S2  | 900                  | 1                   |
| S3  | 250                  | 3                   |
| S4  | 870                  | 4                   |

**Table 6.** Network Sites with Constraints (Six Sites) where site’s capacity calculated hypothetically in bytes.

| S # | Capacity (C) in byte | Fragment Limit (FL) |
|-----|----------------------|---------------------|
| S1  | 1000                 | 6                   |
| S2  | 900                  | 1                   |
| S3  | 250                  | 3                   |
| S4  | 870                  | 4                   |
| S5  | 950                  | 2                   |
| S6  | 710                  | 2                   |
3.7. The proposed fragmentation and allocation algorithm

The proposed algorithm is properly designed in such procedure so that cluster/site constraints are set to be maintained. As mentioned earlier, while fragmentation is the first process, sites clustering have been done ahead of data allocation.

3.7.1. Data fragmentation process

Using fragmentation cost model along with QFRUM, AFAC, DCM matrices, TFAM is to be constructed.

From TFAM, CA attributes is set to be drawn based on equation (7). //like (Abdalla, 2014)

Using CA attribute, specify its predicates to activate fragmentation process as follows:

1. Set of CAs predicates is to be produced, \( P_s = \{P_1, P_2, \ldots, P_n\} \), where \( n \) is number of predicates under consideration, and \( P_i \) is in the form of selection statement.

2. Using \( P_s \), the targeted relation is set to be fragmented as per fragmentation cost model.

3.7.2. Sites clustering before data allocation

Clustering process pseudo code

```
{ 

Select pair of sites of lowest cost (Least Difference Value) to initial first cluster \( C_1 \) with selecting representative

For each site \( S_j \) compare distance1(\( C_1, S_j \)) and distance2 (\( S_j, S_k \)) where distance of \( S_j \) and \( S_k \) is the next lowest cost (next LDV).

If (distance1 > distance2) add \( S_j \) to \( C_1 \) else construct new cluster with \( S_j \) and \( S_k \) as newly-added members for newly-formed cluster, select representative//this step comes in favour of satisfying key//-goal of TC so that TC is to be maximally reduced as distributed query being processed.

Repeat steps (2–4) till all clusters formed and all sites have been involved in clustering process

} //end procedure
```
3.7.3. Communication costs calculation as evaluation process being initiated

{Identifying places of targeted fragment and queries accessing it.

For each cluster/site

For each query and fragment

If Query and fragment at the same site communication costs = zero

If Query and fragment at different sites of the same owner cluster

Communication costs = average distance of all sites within same cluster

If Query and fragment at different clusters

Communication costs = distance cost from site of query/fragment to its counterpart with taking communication costs between clusters representatives into consideration.

}

3.7.4. Data allocation process

Allocation Phase (1):

Procedure assign fragments into Clusters (replication adopted)

{ For i = 1 to n//n is number of obtained fragments

For c = 1 to Cn//Cn number of clusters

{Place (Fi, Cc)

If Cc. constraints. Violation = True//constraint is either capacity or fragment-limit

Take Fi off Cc

} //end for c

} //end for i

Assign-fragments-into-sites procedure (fragment, clusters of sites, sites)// to activate allocation
phase 2
}

**Procedure assign fragments into Clusters (No replication adopted over clusters) // Phase (1)**

{For i = 1 to n // n is number of obtained fragments

{Flag = True

For c = 1 to Cn // Cn number of clusters

{find Cc of highest access value (update cost + retrieval cost)

Place (Fi, Cc)

If Cc. constraints. Violation = True // constraint is either capacity or Cc.fragment-limit

{Take Fi off Cc

Flag = false} // end if

If Flag = True exit Loop (cluster) // to guarantee placing F into one single cluster
else

{find cluster of next highest access cost, let say Cj.

Insert (fi, Cj) } // else

Flag = True // to go back loop and probe cluster constraints

} // end for c

} // end for i

**Assign-fragments-into-sites procedure (fragment, clusters of sites, sites) // to activate allocation**

phase 2
}

} // end procedure

Allocation Phase 2
Procedure Assign-fragments-into-Sites(\text{fragment, clusters of sites, sites})

\{For \textit{i} = 1 to n \ //\text{fragment}

\{If (F.UAC \gt F.RAC) then

\{insert (\textit{fi}, S\textit{j} of highest update cost)

\text{Flag} = \text{True}

\text{For j = 1 to m} \ //\text{m number of sites}

\{If (S\textit{j}. Constraints. Violation = \text{True})

\{take \textit{Fi} off S\textit{j}

\text{Flag} = \text{false} \ //\text{end if}

\text{If Flag} = \text{True} \text{exit Loop} \ //\text{to guarantee placing F into one single site}

\text{else}

\{find site of next highest update cost, let say S\textit{j}.

\text{Insert} (\textit{fi}, S\textit{j}) \ //\text{else}

\text{Flag} = \text{True} \ //\text{to go back loop and probe site constraints}

\} \ //\text{for}

\text{End if}\}

\text{End if}\}

\text{else}

\{Identify all sites that access F and draw them into set of S; Sf

\text{For j = 1 to Sf.length Insert} (F, S\textit{j})

\} \ //\text{end else}

\} \ //\text{end for}

\} \ //\text{end procedure}

4. Results

This work has been implemented on the proposed relation “Student” (Table 7) as per description given in Table 8. As per (Abdalla, 2014), data requirements of dataset can be explicitly provided by administrator of DDBSs or generated
(adopted in this implementation) using a generator for a given attributes predicates and applications over network sites. It is worth indicating that to implement this work, the same environment, including Software and Hardware, in which (Abdalla, 2014) was implemented, is sought to be deliberately created. Therefore, in this work, the same environment along with the same data for both work are purposefully used to proof proposed concepts and confirm present work’s distinction. In short, for the sake of showing both works’ behavior, and simplicity as well as demonstrating DDBS performance, two experiments has separately been conducted with assuming a fully-connected networks of four sites and six sites respectively, (Figs. 2 and 3).

Table 7. Initial datasets (Student Relation) which is set to be collected before running the separately-done implementation, of this work, on all experiments of problems (1) and (3). These datasets can be either explicitly provided by DBA or generated (adopted in this implementation) using a generator program for the given metadata (Table 8).

| Stud-no | Stud-Name | In-date | Position | Fund | Proj-Place | Proj-Id |
|---------|-----------|---------|----------|------|------------|---------|
| 1       | Anna      | 11/01/2015 | Leader   | 15000 | P1         | 112     |
| 2       | Ingrid    | 01/06/2014 | Follower | 12000 | P2         | 113     |
| 3       | Diana     | 29/03/2016 | Follower | 11500 | P3         | 112     |
| 4       | Nadeem    | 21/11/2015 | Follower | 10000 | P1         | 111     |
| 5       | Michel    | 11/01/2015 | Follower | 11000 | P1         | 113     |
| 6       | Amber     | 05/05/2016 | Follower | 9500  | P2         | 114     |
| 7       | Brown     | 29/03/2015 | Leader   | 9000  | P1         | 112     |
| 8       | Sid       | 11/01/2016 | Follower | 11000 | P2         | 111     |
| 9       | Danial    | 01/06/2014 | Follower | 10000 | P3         | 111     |

Table 8. Student Dataset Description (Relation Metadata).

| Attribute | Type   | Length (Bytes) |
|-----------|--------|----------------|
| Stud-no   | Nominal| 3              |
| Stud-Name | Categorical | 30           |
| In-date   | Categorical | 36           |
| Position  | Categorical | 4            |
| Fund      | Numerical | 6             |
| Proj-Place| Categorical | 7            |
| Proj-Id   | Nominal  | 4             |
4.1. First experiment (Network of four sites)

4.1.1. Fragmentation process

Execution steps have partly been illustrated as follows (all tables and pictures are taken from real implementation). Firstly, all requirements information needed are to be accurately recorded as given in Table 9.

As mentioned earlier, for first four sites in both separately-done experiments, only the same frequencies of queries drawn in (Abdalla, 2014) are intentionally used. For process to be begun, suppose that Database Administrator (DBA) gives QRM, QUM and QFM (Tables 10, 11 and 12) as well as the predicates of considered

---

**Fig. 2.** Network Sites (Four sites).

**Fig. 3.** Network Sites (Six sites).
Table 9. Four-Site network Information that is supposed to be given by DBA before running implementation.

| Directive                      | Response |
|--------------------------------|----------|
| Enter no of queries            | 5        |
| Enter no of sites              | 4        |
| Enter no of attributes         | 6        |
| For attribute 1, enter no of predicates | 0    |
| For attribute 2, enter no of predicates | 3    |
| For attribute 3, enter no of predicates | 0    |
| For attribute 4, enter no of predicates | 3    |
| For attribute 5, enter no of predicates | 0    |
| For attribute 6, enter no of predicates | 3    |

Table 10. Query Retrieval Matrix (QRM), which gives how many time each retrieval query is to be running over each Predicate and access data.

| Q#/P# | P1 | P2 | P3 |
|-------|----|----|----|
| Q1    | 1  | 1  | 2  |
| Q2    | 2  | 3  | 1  |
| Q3    | 2  | 5  | 0  |
| Q4    | 0  | 1  | 1  |
| Q5    | 2  | 0  | 1  |

Table 11. Query Update Matrix (QUM), which gives how many time each update query is to be running over each Predicate and access data.

| Q#/P# | P1 | P2 | P3 |
|-------|----|----|----|
| Q1    | 1  | 0  | 0  |
| Q2    | 2  | 2  | 0  |
| Q3    | 2  | 1  | 0  |
| Q4    | 1  | 2  | 0  |
| Q5    | 3  | 1  | 0  |
attributes (Table 9) as follows: In-date (P1: In-date > 2015, P2: In-date < 2015, P3: In-date = 2015), Fund (P1: Fund > 11000, P2: Fund < 11000, P3: Fund = 11000) and Project place (P1: Proj-place = “P1”, P2: Proj-place = “P2”, P3: Proj-place = “P3”).

After that; based on these requirements along with applying fragmentation cost model on Student dataset, “Fund” is deservedly selected to be CA and its predicate set is introduced as follows, Ps = {P1: Fund > 11000, P2: Fund < 11000, P3: Fund = 11000}. It is worth repeating that resulted fragments are equally achieved for both works (Tables 13, 14 and 15).

Finally, it is worth assuring that fragments information like sizes and cardinalities are indispensable for data allocation and performance evaluation.

Table 12. Query Frequency Matrix (QFM), which gives how many time each query (in both cases, retrieval + update) is to be released from each site over network.

| S#/Q# | Q1 | Q2 | Q3 | Q4 | Q5 |
|-------|----|----|----|----|----|
| S1    | 3  | 5  | 0  | 0  | 0  |
| S2    | 0  | 2  | 4  | 0  | 0  |
| S3    | 6  | 0  | 0  | 8  | 0  |
| S4    | 0  | 0  | 0  | 9  | 3  |

Table 13. First Data Fragment (F1).

| Stud-no | Stud-Name | In-date    | Position | Fund | Proj-Place | Proj-Id |
|---------|-----------|------------|----------|------|------------|---------|
| 1       | Anna      | 11/01/2015 | Leader   | 15000| P1         | 112     |
| 2       | Ingrid    | 01/06/2014 | Follower | 12000| P2         | 113     |
| 3       | Diana     | 29/03/2016 | Follower | 11500| P3         | 112     |

Table 14. Second Data Fragment (F2).

| Stud-no | Stud-Name | In-date    | Position | Fund | Proj-Place | Proj-Id |
|---------|-----------|------------|----------|------|------------|---------|
| 4       | Nadeem    | 21/11/2015 | Follower | 10000| P1         | 111     |
| 6       | Amber     | 05/05/2016 | Follower | 95000| P2         | 114     |
| 7       | Brown     | 29/03/2015 | Leader   | 9000 | P1         | 112     |
| 9       | Danial    | 01/06/2014 | Follower | 10000| P3         | 111     |
4.1.2. Data fragments allocation

Based on allocation cost model presented in section 5, and the given matrices of QFM, QRM, QUM, CSM and CCM, data allocation process would be running as follows;

**Phase (1):** QFM, QRM, QUM and CCM would be used along with Eqs. (12), (13) and (14) to give TFRUC matrix, Table 16

**Phase (2):** QFM, QRM, QUM and CSM would be used along with Eqs. (9), (10) and (11) to produce matrices, as shown in Tables 17, 18 and 19. These matrices

### Table 15. Third Data Fragment (F3).

| Stud-no | Stud-Name | In-date     | Position   | Fund | Proj-Place | Proj-Id |
|---------|-----------|-------------|------------|------|------------|---------|
| 5       | Michel    | 11/01/2015  | Follower   | 11000| P1         | 113     |
| 8       | Sid       | 11/01/2016  | Follower   | 11000| P2         | 111     |

### Table 16. Total Frequency of Retrieval and Update Query Frequencies between clusters (TFRUC), for network of four sites. This matrix is used to assign fragment to clusters based on maximum cost concept (Eq. (14)).

| Q#/F# | F1 | F2 | F3 |
|-------|----|----|----|
| C1    | 240| 320| 70 |
| C2    | 230| 265| 155|

### Table 17. Total of Query Retrieval Frequencies (TFRS), Eq. (9).

| Q#/F# | F1 | F2 | F3 |
|-------|----|----|----|
| S1    | 222| 418| 406|
| S2    | 185| 350| 423|
| S3    | 375| 677| 263|
| S4    | 348| 582| 426|

### Table 18. Total of Query Update Frequencies (TFUS), Eq. (10).

| Q#/F# | F1 | F2 | F3 |
|-------|----|----|----|
| S1    | 510| 562| 0  |
| S2    | 361| 390| 0  |
| S3    | 507| 449| 0  |
| S4    | 436| 388| 0  |
would be used to individually assign fragments into sites of clusters. TFRS and
TFUS would be used to determine the precisely-calculated threshold of fragments’
allocation over sites (inside each cluster).

As per constraints of sites, the allocation process, which is accomplished while site
constraints are maintained, for fragments over network sites (for this experiment of
four sites) is shown in Tables 20, 21, 22 and 23. Thus, Tables 20, 21 show final
fragments’ allocation according to (Abdalla, 2014) and the newly-proposed data
allocation scenario for (Abdalla, 2014). Tables 22, 23 display final fragments’
allocation as per newly-proposed data allocation scenarios for present work.

Table 19. Total of Query Retrieval and Update Frequencies (TFRUS) over
network sites. This matrix is used to assign fragments to sites, in each cluster,
based on maximum cost concept. Eq. (11).

| Q#/F# | F1  | F2  | F3  |
|-------|-----|-----|-----|
| S1    | 732 | 980 | 406 |
| S2    | 546 | 740 | 423 |
| S3    | 882 | 1126| 263 |
| S4    | 784 | 970 | 426 |

Table 20. Final Fragments’ Allocation (Abdalla, 2014). Fragments distribution
over sites in replication scenario.

| Fragment/Site | S1 | S2 | S3 | S4 |
|---------------|----|----|----|----|
| F1            | 1  |    |    |    |
| F2            | 1  |    | 1  | 1  |
| F3            | 1  | 0  | 1  | 1  |

Table 21. Final Fragments’ Allocation (Abdalla, 2014). Fragments distribution
over sites in no replication. This scenario however is newly proposed in this work,
in the sense that it was not drawn in (Abdalla, 2014).

| Fragment/Site | S1 | S2 | S3 | S4 |
|---------------|----|----|----|----|
| F1            |    |    |    | 1  |
| F2            |    |    |    |    |
| F3            |    |    |    | 1  |
4.2. Second experiment (Network of six sites)

4.2.1. Fragmentation process

On the other hand, for network of six sites, information needed would be recorded as shown in Table 24 below:

| Directive                                      | Response |
|-----------------------------------------------|----------|
| Enter no of queries                          | 5        |
| Enter no of sites                            | 6        |
| Enter no of attributes                       | 6        |
| For attribute 1, enter no of predicates      | 0        |
| For attribute 2, enter no of predicates      | 3        |
| For attribute 3, enter no of predicates      | 0        |
| For attribute 4, enter no of predicates      | 3        |
| For attribute 5, enter no of predicates      | 0        |
| For attribute 6, enter no of predicates      | 3        |
Like first experiment, only the same query frequencies of first four sites drawn in (Abdalla, 2014) are purposefully taken. So, for process to be begun, suppose that DBA gives QRM, QUM and QFM (Tables 10, 11 and 12) as well as these data of newly-added sites (Table 25) below.

By applying fragmentation cost model along with using information given in QFM, QRM, QUM and QFRUM, “Fund” is also accidentally filtered as CA and the same predicate set (introduced in first experiment) is set to be used. As a result, both works would have the same fragments presented in Tables 13, 14 and 15.

### 4.2.2. Data fragments allocation

Depending on data allocation cost model presented in section 5, and the given matrices of QFRUM, QFM, QRM, QUM, CSM and CCM, data allocation process would be running as follows;

**Phase (1):** QFRUM, QFM, QRM, QUM and CCM would be used along with Eqs. (12), (13) and (14) to give TFRUC matrix, Table 26.

**Phase (2):** QFRUM, QFM, QRM, QUM and CSM would be used along with Eqs. (9), (10) and (11) to produce matrices; as shown in Tables 27, 28 and 29. These matrices would be used to individually assign fragments into sites of clusters.

### Table 25. Query Frequency of Retrieval and Update operation Matrix (QFRUM) for six-site network. This matrix draws how many time each retrieval or update query is to be running over each Predicate and access data. Moreover, Query Frequency Matrix (QFM) is given to depict how many time each query (in both cases, retrieval + update) is to be released from each site over network.

| S# | Q# | Frequency | Activity Mod | In-date | Fund |
|----|----|-----------|--------------|---------|------|
|    |    |           | R/U          | P1 | P2 | P3 | P1 | P2 | P3 | P1 | P2 | P3 |
| S5 | Q1 | 2         | R            | 1  | 0  | 0  | 1  | 1  | 2  | 0  | 0  | 1  |
| S5 | Q1 |           | U            | 2  | 0  | 1  | 1  | 0  | 0  | 2  | 2  | 0  |
| S5 | Q3 | 3         | R            | 0  | 2  | 1  | 2  | 5  | 0  | 1  | 3  | 1  |
| S5 | Q3 |           | U            | 1  | 0  | 0  | 2  | 1  | 0  | 2  | 0  | 1  |
| S5 | Q5 | 2         | R            | 1  | 0  | 1  | 2  | 0  | 1  | 2  | 2  | 1  |
| S5 | Q5 |           | U            | 1  | 0  | 2  | 3  | 1  | 0  | 0  | 0  | 3  |
| S6 | Q2 | 2         | R            | 3  | 1  | 0  | 2  | 3  | 1  | 1  | 2  | 0  |
| S6 | Q2 |           | U            | 0  | 1  | 1  | 2  | 2  | 0  | 1  | 2  | 0  |
| S6 | Q3 | 1         | R            | 0  | 2  | 1  | 2  | 5  | 0  | 1  | 3  | 1  |
| S6 | Q3 |           | U            | 1  | 0  | 0  | 2  | 1  | 0  | 2  | 0  | 1  |

Like first experiment, only the same query frequencies of first four sites drawn in (Abdalla, 2014) are purposefully taken. So, for process to be begun, suppose that DBA gives QRM, QUM and QFM (Tables 10, 11 and 12) as well as these data of newly-added sites (Table 25) below.
Table 26. Total Frequency of Retrieval and Update Query Frequencies between clusters (TFRUC), for network of six sites. This matrix is used to assign fragment to clusters based on maximum cost concept (Eq. (14)).

| C #/F# | F1  | F2  | F3  |
|--------|-----|-----|-----|
| C1     | 380 | 434 | 189 |
| C2     | 246 | 306 | 84  |
| C3     | 330 | 424 | 89  |

Table 27. Total of Query Retrieval Frequency (TFRS), Eq. (9).

| S#/F# | F1  | F2  | F3  |
|-------|-----|-----|-----|
| S1    | 264 | 524 | 224 |
| S2    | 274 | 434 | 310 |
| S3    | 260 | 442 | 160 |
| S4    | 242 | 398 | 158 |
| S5    | 220 | 384 | 232 |
| S6    | 246 | 412 | 254 |

Table 28. Total of Query Update Frequency (TFUS), Eq. (10).

| S#/F | F1  | F2  | F3  |
|------|-----|-----|-----|
| S1   | 360 | 300 | 0   |
| S2   | 376 | 320 | 0   |
| S3   | 300 | 234 | 0   |
| S4   | 288 | 172 | 0   |
| S5   | 368 | 368 | 0   |
| S6   | 356 | 302 | 0   

TFRS and TFUS would be used to determine the precisely-calculated threshold of fragments’ allocation over sites (inside each cluster).

For data allocation process, site constraints are preserved for fragments to be assigned to network sites (for this experiment of six sites) as illustrated in Tables 30, 31, 32 and 33. While Tables 30, 31 give final fragments’ allocation according to (Abdalla, 2014) and the newly-proposed scenario of data allocation
for (Abdalla, 2014); Tables 32, 33 show final fragments’ allocation as per newly-proposed data allocation scenario for present work.

5. Discussion

In light of above-addressed contributions of this meticulously-designed work, it can be concluded that this optimized work has come with remarkable progress (comparing with (Abdalla, 2014)) on DDBS performance enhancement. This

Table 29. Total of Query Retrieval and Update Frequencies (TFRUS) over network sites. This matrix is used to assign fragments to sites, in each cluster, based on maximum cost concept. Eq. (11).

| S#/F# | F1  | F2  | F3  |
|-------|-----|-----|-----|
| S1    | 624 | 824 | 224 |
| S2    | 650 | 754 | 310 |
| S3    | 560 | 676 | 160 |
| S4    | 530 | 570 | 158 |
| S5    | 588 | 752 | 232 |
| S6    | 602 | 714 | 254 |

for (Abdalla, 2014); Tables 32, 33 show final fragments’ allocation as per newly-proposed data allocation scenario for present work.

Table 30. Final Fragments’ Allocation (Abdalla, 2014). Fragments distribution over sites in replication scenario.

| Fragment/Site | S1 | S2 | S3                  | S4 | S5 | S6 |
|---------------|----|----|---------------------|----|----|----|
| F1            | 0  | 1  | 0                   | 0  | 0  | 0  |
| F2            | 1  | 0  | fragment limit violation | 0 | 1  | 1  |
| F3            | 1  | 0  | fragment limit violation | 1 | 1  | 1  |

Table 31. Final Fragments’ Allocation (Abdalla, 2014). Fragments distribution over sites in no replication. This scenario however is newly proposed in this work, in the sense that it was not drawn in (Abdalla, 2014).

| Fragment/Site | S1 | S2 | S3 | S4 | S5 | S6 |
|---------------|----|----|----|----|----|----|
| F1            | 1  |    |    |    |    |    |
| F2            | 1  |    |    |    |    |    |
| F3            | 1  | 0  | fragment limit violation so to site of next max |    |    |    |
progress is technically supported with experimental results and performance evaluation of this section. Data locality promotion and transmission costs (TC) reduction are the main factors by which this work is critically measured. According to procedure by which this work is produced, it is believed that data would be as local as possible leading to maximal minimization on TC as consequences. On the other hand, to verify these claims, TC is clearly expressed by objective function of this work, and an internal and external evaluation are made for both works. Needless to say that performance is measured with respect to how much costs have been incurred inside network as distributed queries being processed.

In brief, Table 34 shows that five problems (each of which has its own experiments, parameters and variables) are carefully addressed to evaluate both works, (Abdalla, 2014) and the present optimized work of this paper. In the sense that all results of present work of this paper (for both scenarios of data allocation) are evaluated against work of (Abdalla, 2014) including its newly-proposed data allocation scenario (given in Tables 21, 31). Last but not least, for the sake of enforcing optimized approach’s superiority under several circumstances, many different experiments, along with diversifying number of sites of network, dataset cardinalities and number of queries running against DDBS, are precisely conducted (Table 34).

### Table 32. Final Fragments’ Allocation [present work- replication adopted].
Fragments distribution over sites and clusters.

| Fragment/Cluster | C1 | C2 | C3 |
|------------------|----|----|----|
| Fragment/Site    | S2 | S6 | S1 | S4 | S3 | S5 |
| F1               |    | 1  | 1  |    | 0  |    |
| F2               | 0  |    | 1  | 1  | 0  |    |
| F3               | 0  |    | 1  | 1  | 1  | 1  |

### Table 33. Final Fragments’ Allocation [present work- no replication adopted].
Fragments distribution over sites and clusters.

| Fragment/Cluster | C1 | C2 | C3 |
|------------------|----|----|----|
| Fragment/Site    | S2 | S6 | S1 | S4 | S3 | S5 |
| F1               |    | 1  |    |    |    |    |
| F2               | 0  |    | 1  |    |    |    |
| F3               | 0  |    | 1  |    |    |    |
Finally, before discussing evaluation process, it has to be referring that three data allocation scenario are considered while conducting evaluation to stress optimized work’s distinction. These scenarios are: Scenario (1): (Abdalla, 2014) with Present Work so that Replication Scenario, for both works, is imposed; Scenario (2): (Abdalla, 2014) with Present Work so that No-Replication Scenario for both works; and Scenario (3): (Abdalla, 2014) with replication scenario and Present Work with no replication scenario. All information concerning evaluation process is concisely displayed in Table 33. Additionally, it is most important to indicate that original queries running against dataset (for all problems) are five queries. However, as per above-discussed methodology, each query released from each site would be treated as a different with different frequency. In other words, each query in each site is set to be processed independently of its replica at other sites. As a result, entire number of actual considered queries is bound to be significantly increased based on the rate in which queries are released over sites. Hence, for each problem in Table 34, column which is titled “queries#” gives number of actual launched queries including queries’ replica rather than original queries.

Table 34. All Problem Addressed in this work. Each problem has been investigated through conducting three experiments within its own unique dataset cardinality, queries number and number of sites of network considering all data allocation scenarios.

| Problem# | Experiment# | Dataset Cardinality | Allocation Scenario | Actual Queries# | Original Queries# | Sites# | Clusters# |
|---------|-------------|---------------------|---------------------|-----------------|-------------------|-------|-----------|
| P_1     | 1           | 9                   | Scenario (1)        | 16              | 5                 | 4     | 2         |
|         | 2           | -                   | Scenario (2)        | -               | -                 | -     | -         |
|         | 3           | -                   | Scenario (3)        | -               | -                 | -     | -         |
| P_2     | 4           | 50                  | Scenario (1)        | 16              | -                 | 4     | 2         |
|         | 5           | -                   | Scenario (2)        | -               | -                 | -     | -         |
|         | 6           | -                   | Scenario (3)        | -               | -                 | -     | -         |
| P_3     | 7           | 9                   | Scenario (1)        | 26              | -                 | 6     | 3         |
|         | 8           | -                   | Scenario (2)        | -               | -                 | -     | -         |
|         | 9           | -                   | Scenario (3)        | -               | -                 | -     | -         |
| P_4     | 10          | 50                  | Scenario (1)        | 26              | -                 | 6     | 3         |
|         | 11          | -                   | Scenario (2)        | -               | -                 | -     | -         |
|         | 12          | -                   | Scenario (3)        | -               | -                 | -     | -         |
| P_5     | 13          | 200                 | Scenario (1)        | 40              | -                 | 12    | 5         |
|         | 14          | -                   | Scenario (2)        | -               | -                 | -     | -         |
|         | 15          | -                   | Scenario (3)        | -               | -                 | -     | -         |
For this demonstration, four experiments (1, 2, 7 and 8) have exclusively been illustrated in this paper as per section 7. All experiments (including these separately-done experiments) are made in self-explanatory frame as they seek to find which work is the best fitting for DDBS design.

To begin, for first experiment of first problem, Figs. 4 and 5 draw the results as data replication scenario is imposed. Obviously, present work outperforms (Abdalla, 2014) in term of TC1 (Fig. 4). However, both works are observed to be close to each other regarding TC2 (Fig. 5), with a very subtle leading is being recorded for (Abdalla, 2014). Nevertheless, for TC in total, these results substantially come in favor of present work (Fig. 6). Obviously, present work produces less communication costs when compared to (Abdalla, 2014). All in all, for this scenario, present work is observed to contribute significantly at highly improving DDBS performance. It is worth repeating that performance is mathematically weighed-up by how much costs (in bytes) are being yielded as distributed query under processing.

Fig. 4. Problem 1; Experiment 1- Within four-site network, present work is being evaluated against (Abdalla, 2014) in data replication scenario, as they are both exposed on TC1 Eq. (1).

Fig. 5. Problem 1; Experiment 1- Within four-site network, present work is being evaluated against (Abdalla, 2014) in data replication scenario, as they are both exposed on TC2 Eq. (2).
By the same token, experiment (7) in Table 34 reinforces the hypothesis proved in last experiment. Figs. 7, 8 and 9 simply display findings so that replication scenario is elaborated. As per these findings, present work is recorded to be by far the best (in terms of TC1, TC2 and TC in total). In other words, present work yields very much less communication costs comparing with (Abdalla, 2014).

Fig. 6. Problem 1; Experiment 1- Within four-site network, present work is being evaluated against (Abdalla, 2014) in data replication scenario, as they are both exposed on TC-in-total Eq. (3).

Fig. 7. Problem 1; Experiment 2-Within six-site network, present work is being evaluated against (Abdalla, 2014) in data replication scenario, as they are both exposed on TC1 Eq. (1).

Fig. 8. Problem 1; Experiment 2- Within six-site network, present work is being evaluated against (Abdalla, 2014) in data replication scenario, as they are both exposed on TC2 Eq. (2).
To sum up, **according to this scenario (1)**, present work contributes remarkably at promoting overall DDBS throughputs. The second experiment (2) in Table 34, on the other hand, has been conducted with considering no data replication in both works. The purpose of this scenario is sought to prove optimized technique’s effectiveness in all circumstances. Like experiment (1), this experiment is accomplished so that network is fully connected of four sites. Figs. 10, 11 and 12 clearly demonstrate that (Abdalla, 2014) behave worse in terms of producing TC than present work. In the sense that present work also proves its superiority at highly enhancing DDBS performance, as shown in Fig. 12.

Last but not least, experiment (8) has been conducted with considering no data replication in both works. The purpose of this scenario is aimed at backing hypothesis proved in experiment (2) so that optimized present work is much more effective than (Abdalla, 2014). Like experiment (7), this experiment is accomplished in network of fully-connected six sites. Therefore, Figs. 13, 14 and 15 vividly support that present work behaves much better, in terms of increasing DDBS productivity, than (Abdalla, 2014). In the sense that, **according to this scenario**, present work also proves its superiority at substantially
decreasing transmission costs and overwhelming boosting DDBS rendering, as shown in Figs. 16 and 17.

Lastly, all experiments (drawn in Table 34), for all five problems, have been carried out in the same pattern for both works taking all scenarios of data allocation into account. Moreover, as mentioned earlier in section (3.3). Most importantly, to distinguish the supremacy of site clustering algorithm of this work and to prove present’s work primacy, clustering sites algorithm drawn in (Sewisy et al., 2017) is also involved to be conducted in the same pattern for all these experiments so that present work is to be conducted twice. While the first conduction is set to be made with clustering algorithm of this paper, the second conduction is set to be done using clustering algorithm of (Sewisy et al., 2017). In fact, as another outstanding contribution of this work, both conductions have been considered just to further emphasize the superiority of present work’ algorithm of site clustering over that of

![Graph showing TC values for various scenarios](image1)

**Fig. 11.** Problem 3; Experiment 6- Within four-site network, present work is being evaluated against (Abdalla, 2014) in data non-replication scenario, as they are both exposed on TC2 Eq. (2).

![Graph showing percentage distribution](image2)

**Fig. 12.** Problem 3; Experiment 6- Within four-site network, present work is being evaluated against (Abdalla, 2014) in data non-replication scenario, as they are both exposed on TC-in-total Eq. (3).
Both algorithm have been observed to perform much better than (Abdalla, 2014), though. So, according to the summed-up findings of these experiments, present work has tremendously proved to outperform (Abdalla, 2014) in terms of lessening communication costs as shown in Figs. 16 and 18 as well as effectively promoting overall DDBS performance as given in Figs. 17 and 19. Furthermore, all results of present work has undoubtedly demonstrated that sites’ clustering process has huge impact on satisfying transmission costs reduction and promoting DDBS performance as well. On the other hand, in spite of that present work’s sites’ clustering algorithm is shown to substantially outweigh that clustering algorithm of (Sewisy et al., 2017), both works, however, basically endeavor to accentuate the undeniable impact of sites clustering over (Abdalla, 2014), as it comes to the issue of enhancing DDBS rendering.

According to the above-drawn contributions and the well-analyzed results, it can be technically analyzed that the proposed work (of this paper) aims at perfectly optimizing (Abdalla, 2014) in a clear bid to professionally meet the needs of finding an optimal data fragmentation and allocation solution in DDBSs. In the
following, a simple theoretical comparison is tersely put for this work along with some previous relevant works including (Abdalla, 2014) to highlight the strength and weakness points of present work.

Theoretically, on the other hand, to conduct data allocation in both replicated and non-replicated scenario, (Hababeh et al., 2005) proposed two-approach model. While the first approach is “best fit” which is used to perform non-replicated based data allocation, the second approach is “all beneficial strategy” accomplish replicated-based data allocation. Therefore, like (Hababeh et al., 2005), present work of this paper has considered both scenarios. Meanwhile, (Huang and Chen, 2001) sought to exploit the concept proposed in (Hababeh et al., 2005) to distribute relations in replicated manner using “best fit” concept and in non-replicated manner using “all beneficial strategy” concept. Additionally, operation allocation was considered along with data allocation that they are treated at the same time. To
make this work capable of dynamically allocating data, initial allocation and re-allocation was addressed as well.

In the meantime, model proposed in (Abdalla, 2014) was aimed to perform fragmentation and allocation on the fly that no additional complexity for data allocation was needed. Moreover, it considered two scenarios for site constraints as data being allocated. In first scenario, data allocation had been done while constraints being relaxed. In the second scenario, sites constraints had been maintained, though. However, technique’s evaluation and performance analysis were not supported. Finally, the proposed work of this work, as mentioned earlier, comes with the aim of integrating some of previously proposed techniques.
including (Abdalla, 2014) to significantly satisfy the continuously needs of data fragmentation and allocation optimality.

To sum up, this optimized work of this paper seeks to introduce an efficient approach capable of outstandingly performing the following:

1. Like (Abdalla, 2014), queries and sites information are used to perform non-replicated data allocation using strategy similar to “best fit” approach, but with different calculations methods. By the same token, strategy similar to “all beneficial approach” is used to conduct data allocation over clusters in replicated manner.

2. A heuristic approach, for synchronized horizontal fragmentation and allocation, is evolved using optimized cost model of this work. Data locality increasing and TC reduction were the key motivations. Adversely, this work consumes more storage space compared to previous methods including (Abdalla, 2014).

3. On contrary to (Abdalla, 2014), present work integrates site clustering algorithm in a step ahead of data allocation to reinforce communication costs lessening.

4. Unlike (Abdalla, 2014), this work proposed mathematically-based data allocation and replication models.

5. Fragments replication was considered so that replication decision is automatically done over all clusters using the “all beneficial approach” of (Hababeh et al., 2005) in first scenario. However, for sites within each cluster, replication decision is taken based on precise calculation of threshold values and their adaptation to the changes in access pattern making the proposed work much more efficient than (Abdalla, 2014) as it comes to transmission costs minimization.

Fig. 19. Performance in percentage (No-Rep Scenario). This measure has been calculated, for data non-replication scenario, based on obtained results of objective functions as shown in Eqs. (1), (2) and (3).
6. Opposed to (Abdalla, 2014), both cluster and site constraints were equally taken into account. However, like (Abdalla, 2014) “Dijkstra” algorithm is adopted to further reduce communication costs by producing Distance Cost Matrix (DCM) which used for fragments allocation and performance evaluation as distributed query process.

7. Most importantly, internal and external evaluation for present work along with (Abdalla, 2014) has been expertly drawn to demonstrate work’s superiority in both scenarios of replication and non-replication.

6. Conclusions

Recently, DDBS is becoming unstoppable growing swell of the current technology-based era since it best fits the real-world enterprises and organizations of different sites at different places worldwide. Consequently, DDBS performance is heavily being of key paramount to be desperately considered. However, DDBS performance is basically dependent on procedure by which DDBS is set to be designed (fragmented and allocated over all sites of its network). For fragmentation process, data should be carefully divided so that a match between targeted data and considered queries is to be maximally satisfied. On the other hand, the major motivation for data allocation is to store data fragments at several sites in such way that delicately seeks to minimize overall transmission costs considerably as a set of queries under consideration is being processed.

Therefore, as an optimization setup for (Abdalla, 2014), in this paper, an optimized approach for horizontal fragmentation is perfectly suggested along with proposing newly-made mathematical-based data allocation and replication models. This work moreover aims to carefully develop a site clustering algorithm that similar sites (in terms of communication costs) are to be grouped together in step ahead of conducting data allocation. Data allocation, on the other hand, is unanimously known to result in a significant role in both DDBS design and performance, too. In this work, data allocation is fully done using proposed cost-effective model with the aim of simultaneously increasing data locality and decreasing remote access. Experimental results therefore records undeniable growing enhancement with regard to DDBSs performance through minimizing transmission costs among sites of network. For both works, a different data allocation (including data replication) scenarios (over clusters and sites) are being proposed and evaluated against each other. However, like (Abdalla, 2014), a threshold of retrieval and update costs has been used to decide whether or not to replicate fragments over sites inside clusters individually. Cluster and site constraints are also considered to stimulate the real-world DDBS as well as to strengthen proposed approach efficiency which has been confirmed by presented experimental results. The present approach is cautiously evaluated against (Abdalla, 2014) on the basis of drawn objective function of
present work which is originally taken from (Abdalla, 2014) and significantly amended to reflect actual reality of transmission costs (Sewisy et al., 2017). The evaluation process, on the other hand, is primarily sought to manifest that present work behaves much better (If not the best) in terms of TC and DDBS performance alike.

Last but not least, as per results obtained and given in discussion section, it can be confidently said that experimental results have concretely demonstrated by every indication that present optimized work of this paper has undisputable leading over (Abdalla, 2014) in terms of hugely mitigating transmission costs and massively promoting overall DDBS throughputs. To sum up, it has been indisputably shown that the proposed optimized approach is close enough to be an optimal, and technically behave much better (if not the best) compared to (Abdalla, 2014). Finally, it is worth pointing that the upcoming work is completely devoted to conduct more experiments on a real-world DDBS so that a general framework (for data fragmentation and allocation) is set to be ingeniously created.

Declarations

Author contribution statement

Ali A. Amer: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Adel A. Sewisy: Performed the experiments; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Taha M. A. Elgendy: Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Competing interest statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.
Acknowledgements

The authors would like to heartily express their sincere appreciation to the research unit of computer science department at Taiz University for the physical materials dedicated to help accomplish this work successfully. Additionally, authors’ thanks is to be sincerely extended to both Editors of HELIYON along with those respected unknown reviewers for their invaluable guidance, comments and suggestions that otherwise this research paper would not be coming out.

References

Abdalla, Hassan I., 2014. A synchronized design technique for efficient data distribution. Comput. Human Behav. 30, 427–435.

Abdalla, Hassan I., Amer, Ali A., Mathkour, Hassan, 2014. Performance Optimality Enhancement Algorithm in DDBS (POEA). Comput. Human Behav. 30, 419–426.

Abdel Raouf, Ahmed E., Badr, Nagwa L., 2017. Mohamed Fahmy Tolba, Distributed Database System (DSS) Design Over a Cloud Environment. Springer International Publishing AG Multimedia Forensics and Security, pp. 97–116.

Al-Sayyed, R., Al Zaghoul, F., Suleiman, D., Itriq, M., Hababeh, I., 2014. A New Approach for Database Fragmentation and Allocation to Improve the Distributed Database Management System Performance. Journal of Software Engineering and Applications 7, 891–905.

Amer, Ali A., Abdalla, Hassan I., 2012. Dynamic Horizontal Fragmentation, Replication and Allocation Model in DDBSs. IEEE International Conference on Information Technology and e-Services, Sousse, Tunisia. http://ieeexplore.ieee.org/xpl/login.jsp?p=&arnumber=6216603&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6216603.

Amita Goyal Chin, 2002. Incremental Data Allocation and Reallocation in Distributed Database Systems Data warehousing and web engineering. IRM Press Hershey, PA, United States, pp. 137–160.

Bellatreche, L., Kerkad, A., 2015. Query interaction based approach for horizontal data partitioning. International Journal of Data Warehousing and Mining, (IJDWM) 11, 44–61.

Ceri, S., Negri, M., Pelagatti, G., 1982. Horizontal data partitioning in database design. ACM SIGMOD international conference on Management of data, 128–136. http://dl.acm.org/citation.cfm?id=582376.

Ceri, S., Pernici, B., Wiederhold, G., 1986. Optimization Problems and Solution Methods in the Design of Data Distribution. J. Manag. Inf. Syst. 14 (3), 261–272.
Dejan Chandra Gope, 2012. Dynamic Data Allocation Methods in Distributed Database System. American Academic & Scholarly Research Journal 4 (6).

Hababeh, I., Bowring, N., Ramachandran, M., 2005. A method for fragment allocation design in the distributed database systems. The Sixth Annual UAE University Research Conference. https://www.researchgate.net/publication/228345279_A_method_for_fragment_allocation_design_in_the_distributed_database_systems.

Hababeh, I., Khalil, I., Khreishah, A., 2015. Designing High Performance Web-Based Computing Services to Promote Telemedicine Database Management System. IEEE Transactions on Services Computing 8 (1), 47–64.

Harikumar, S., Ramachandran, R., 2015. Hybridized fragmentation of very large databases using clustering. IEEE Signal Processing, Informatics Communication and Energy Systems (SPICES), 1–5.

Hauglid, Jon Olav, Ryeng, Norvald H., Norvag, Kjetil, 2010. Dynamic Fragmentation and Replica Management in Distributed Database Systems. Journal of Distributed and Parallel Databases 28 (3), 1–25.

Huang, Yin-Fu, Chen, Jyh-Her, 2001. Fragment Allocation in Distributed Database Design. Journal of Information Science and Engineering.

Kumar, Raju, Gupta, Neena, 2016. An Extended Efficient Approach to Dynamic Fragment Allocation in Distributed Database Systems. IJCTA, 473–482 International Science Press.

Lin, Xuemin, Orlowska, M., Zhang, Yanchun, 1993. On data allocation with the minimum overall communication costs in distributed database design. Computing and Information. Fifth International Conference.

Mukherjee, Nilarun, 2011. Synthesis of Non-Replicated Dynamic Fragment Allocation Algorithm in Distributed Database Systems. ACEEE Int. J. Inform. Technol. 1 (1).

Ozsu, M. Tamer, Valduriez, Patrick, 2011. Principles of Distributed Database Systems, 3rd edition Prentice-Hall, New Jersey.

Sewisy, Adel A., Amer, Ali Abdullah, Abdalla, Hassan I., 2017. A Novel Query-Driven Clustering-Based Technique for Vertical Fragmentation and Allocation in Distributed Database Systems. Int. J. Semantic Web Inf. Syst. 13 (2). http://www.igi-global.com/article/a-novel-query-driven-clustering-based-technique-for-vertical-fragmentation-and-allocation-in-distributed-database-systems/176732.
Singh, Arjan, Kahlon, K.S., 2009. Non-replicated Dynamic Data Allocation in Distributed Database Systems. IJCSNS International Journal of Computer Science and Network Security 9 (9).

Singh, Arjan, 2016. Empirical Evaluation of Threshold and Time Constraint Algorithm for Non-replicated Dynamic Data Allocation in Distributed Database Systems, Proceedings of the International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing, 439.

Surmsuk, P., Thanawastien, S., 2007. The integrated strategic information system planning Methodology. 11th IEEE International Conference, Enterprise Distributed Object Computing.

Tâmbulea, Leon, Horvat, Manuela, 2008. Dynamic Distribution Model in Distributed Database. International Journal of Computers, Communications & Control 3 (3), 512–515.

Ulus, T., Uysal, M., 2007. A Threshold Based Dynamic Data Allocation Algorithm- A Markov Chain Model Approach. J. Appl. Sci. 7 (2), 165–174.

Wiese, L., 2015. Horizontal fragmentation and replication for multiple relaxation attributes. Data Science (30th British International Conference on Databases), Springer, pp. 157–169.

Wiese, L., Waage, T., Bollwein, F., 2016. A Replication Scheme for Multiple Fragmentations with Overlapping Fragments. The Computer Journal 60 (3), 308–328.

Zhang, Y., Orlowska, Maria E., 1994. On Fragmentation Approaches for Distributed Database Design. Information Sciences Applications 1 (3), 117–132.