Resource Defragmentation using Market-Driven Allocation in Virtual Desktop Clouds

Prasad Calyam, Sripriya Seetharam, Baisravan HomChaudhuri, Manish Kumar
University of Missouri-Columbia, USA; University of Toledo, USA.
Email: calyamp@missouri.edu, ssn68@mail.missouri.edu, baisravan.hc@gmail.com, manish.kumar2@utoledo.edu

Abstract—Similar to memory or disk fragmentation in personal computers, emerging “virtual desktop cloud” (VDC) services experience the problem of data center resource fragmentation which occurs due to on-the-fly provisioning of virtual desktop (VD) resources. Irregular resource holes due to fragmentation lead to sub-optimal VD resource allocations, and cause: (a) decreased user quality of experience (QoE), and (b) increased operational costs for VDC service providers. In this paper, we address this problem by developing a novel, optimal “Market-Driven Provisioning and Placement” (MDPP) scheme that is based upon distributed optimization principles. The MDPP scheme channelizes inherent distributed nature of the resource allocation problem by capturing VD resource bids via a virtual market to explore soft spots in the problem space, and consequently defragments a VDC through cost-aware utility-maximal VD re-allocations or migrations. Through extensive simulations of VD request allocations to multiple data centers for diverse VD application and user QoE profiles, we demonstrate that our MDPP scheme outperforms existing schemes that are largely based on centralized optimization principles. Moreover, MDPP scheme can achieve high VDC performance and scalability, measurable in terms of a ‘Net Utility’ metric, even when VD resource location constraints are imposed to meet orthogonal security objectives.

I. INTRODUCTION

User communities in retail, education and other economy sectors are increasingly transitioning “physical distributed desktops” that have dedicated hardware for traditional offline applications (e.g., MS Office, Media Player, Matlab) into “virtual desktop clouds” (VDCs) that are accessible via thin-clients. Cloud Service Providers (CSPs) have recognized this demand, and have recently started supporting on-demand, ‘pay-as-you-go’ Desktop-as-a-Service (DaaS) offerings (e.g., AWS WorkSpaces [1]) that augment their other common offerings such as Infrastructure-as-a-Service (IaaS), and Software-as-a-Service (SaaS).

One of the critical factors that determines success of a DaaS offering is the delivery of satisfactory user quality of experience (QoE). In order to comply with service level agreements (SLAs) that map to user QoE needs of users, CSPs provision resources in an ‘on-the-fly’ manner for every incoming virtual desktop (VD) request. The resource provisioning is performed across multiple data centers and particular placement of a VD request in a data center could be based upon considerations such as cost, load or latency. Similar to memory or disk fragmentation in personal computers, VDC services experience the problem of data center resource fragmentation, where ‘opportunistic’ resource placements over time lead to careless packing of VDs as in the “tetris effect” [2].

The opportunistic placements do not perform in-depth, computationally-intensive and time-consuming checks to determine whether every placement decision is globally optimal and leads to maximal-utility in a VDC; they use high-level information about resource status in data centers to rapidly make placement decisions. Such opportunistic VD resource allocations are desirable since they reduce user wait time for VD access after request initiation. However, due to workload dynamics, login/logout trends of user, and possible user mobility, opportunistic placements over time create irregular resource capacity holes in data centers, and also may cause many VDs to be placed in highly sub-optimal locations. For instance, users may be connected to VDs that are allocated only the minimum resources due to load issues, when additional resources are available at a different data center. In extreme cases, users VD requests may also be rejected when there are many resource capacity holes in one or more data centers. Such situations can thus impact user QoE for customers, and cause increased operational costs for CSPs due to their impact on performance and scalability.

In this paper, we address the above combinatorial problem in VDCs by developing a novel, optimal “Market-Driven Provisioning and Placement” (MDPP) scheme that is based upon distributed optimization principles. Our scheme uses virtual market economics and a consumer driven resource pricing strategy to optimize resource allocations, which has been successful in other large-scale complex networked systems such as power grids [3] [4]. Fig. 1 shows a simple illustration of how we apply such a cross-disciplinary strategy in the context of a VDC market. In our VDC system, VD users behave as buyers who require maximized utility for the...
resources allocated to them, while the CSPs behave as sellers trying to provide optimal resources at low cost. The thin-client VD requests individually bid for optimal resources based on the user QoE needs of an application. For e.g., a thin-client user may request high network bandwidth to be provisioned for high-definition video playback in a Media Player as part of a Distance Learning use case. Alternately, a thin-client user may request high memory to be provisioned for computer-aided design model manipulation with Matlab as part of an Engineering Site use case. With MDPP, bids only need price information of VDC provider and do not require detailed cost models knowledge, which is confidential in a VDC. Thus MDPP aims to satisfy the bids of all VDs requests in a distributed manner with minimal information exchange, while ensuring allocations do not exceed available VDC resources.

In our previous work [5], we showed that defragmentation is an important optimization step to improve VDC performance and scalability, measurable in terms of a ‘Net Utility’ metric. For a given online saturated state after opportunistic resource allocations, we showed how a centralized greedy heuristic with utility gain ordering (i.e., “Utility-ordered Provisioning and Placement” (UOPP)) coupled with a ‘migration cost-benefit analysis’ led to improvements in the Net Utility by up to 30%. However, due to the greedy heuristic approach, our UOPP scheme overcompensates for high-dimensionality in the inherent problem, and is still a sub-optimal solution. In contrast, the MDPP scheme has been designed to channelize inherent distributed nature of the resource allocation problem by capturing VD resource bids via a virtual market to explore soft spots in the problem space. Consequently, our MDPP scheme can optimally defragment a VDC through cost-aware utility-maximal VD re-allocations or migrations.

Through extensive simulations of VD request allocations to multiple data centers for diverse VD application and user QoE profiles, we demonstrate that our MDPP scheme outperforms existing schemes including our earlier UOPP scheme, which are largely based on centralized optimization principles. The VD application profiles of different user groups that we use in our simulations to determine resource provisioning requirements are obtained from a real VDC testbed [6] [7]. Moreover, we show that MDPP scheme can achieve high VDC performance measured in terms of Net Utility, even when VD resource location constraints are imposed to meet orthogonal security objectives in cases where for e.g., VDs with highly-confidential applications/data should be specifically placed on hosts that have none, or very few public-facing interfaces.

The remainder of this paper is organized as follows: In Section II, we describe related work on cloud resource allocation and optimization. In Section III, we formally describe the defragmentation solution objective, while reflecting on the involved complexity. In Section IV, we present details of our MDPP scheme. In Section V, we present performance comparison of our MDPP scheme along with UOPP and other centrally-driven schemes. Section VI concludes the paper.

II. RELATED WORK

There have been several works on opportunistic cloud resource allocation and migration optimization in the context of provisioning and placement of virtual machines [8] - [11]. A fair resource allocation scheme is proposed in [8], where user requests are handled centrally and are statically placed in a round-robin manner across data centers based on resource dominance levels. Works such as [9] and [10] propose dynamic placement solutions that involve virtual machine migrations between data centers to conserve overall power consumption. A genetic algorithm approach is also used in [11] to make resource allocations based on network topology and application requirements for data-intensive workloads. These works are targeted mainly for user workloads in virtual machines that can be modeled as transactions with predictable resource consumption characteristics. In comparison, our work considers utility functions of real-time application workloads of VD applications, which have distinctly bursty resource consumption trends [6].

Resource fragmentation is an extensively studied problem in storage and memory systems in traditional operating systems literature for many years. However, only in recent years have there been works emerging such as [5], [12] and [13] that have begun to address the defragmentation strategies in the context of cloud platforms. The authors in [12] developed a periodic defragmentation scheme that is triggered in a manner that minimizes the cost involved in migrations necessary to consolidate virtual machines (also resource capacity holes) onto a minimum number of servers. The authors in [13] seek to dynamically allocate resources to VMs including those that host virtual desktops using an iterative divvy algorithm. Their goal is to improve scalability and compliance with resource sharing policies within public clouds. Given that the general resource defragmentation problem is NP-hard in nature, heuristic and meta-heuristic based approaches in prior literature can suffer in terms of complexity as the total number of decision variables (i.e., resources, constraints) increase, and it is often challenging to define a single system-wide metric for the optimization process [3] [14].

In the distributed optimization approach used in our MDPP scheme, we resolve scalability issues by using a work sharing model that can handle large dimensions of decision variables. In addition, we use a Net Utility metric for a VDC system that not only measures the server-side utility for a given number of provisioned VDs, but also considers the level of network health that can impact the overall user QoE. The core idea of our work is the idea to decompose the overall Net Utility optimization into sub-problems. We solve the sub-problems in a decentralized manner using price-based bidding mechanisms seen in market-driven techniques that have been successfully applied for power grids [3] [4], which have resource allocation challenges that are comparable to the case of a VDC market. Decomposition methods can be broadly classified into primal decomposition [15] and dual decomposition [16]. The former method is based on decomposition of the original primal problem, while the latter is based on the decomposition of the Lagrangian dual problem [17]. In our work, we model resource defragmentation problem as a dual decomposition problem where the primal variables are the resources while the dual variables correspond to the prices of the resources.
III. PROBLEM MOTIVATION

In this section, we first describe the online opportunistic provisioning and placement scheme that can be used to saturate data center resources for a large-enough set of VD requests. Following this, we illustrate the problem of offline defragmentation formally and define the solution objective.

A. Online Opportunistic Resource Allocation

In our previous works [6] and [7], we developed and evaluated different opportunistic placement schemes that allocate resources on-the-fly for every incoming VD request. To perform provisioning of VDs, the VDC relies on information regarding user groups (e.g., Campus Computer Lab, Distance Learning Site, Engineering Site) and their application profiles that are obtained via toolkits for resource consumption and network health measurements. The profiles indicate the minimum and maximum resource amounts (i.e., \( R_{\text{min}} \) and \( R_{\text{max}} \)) that correlate to user QoE; resource provisioning below \( R_{\text{min}} \) violates SLAs (users find a VD unusable), and resource provisioning above \( R_{\text{max}} \) does not produce any user-perceivable performance benefit. The profiles thus allow ‘desktop pooling’ corresponding to the resource requirements of applications in different user groups. After addressing the provisioning requirements, a data center \( L_i \) location is selected with an aim to maximize Net Utility \( (U^{Net}) \) by using one of the four basic placement schemes: ‘Least Cost’, ‘Least Latency’, ‘Least Load’, and ‘Least Joint’, which are described in the following:

1) Least Cost: Least Cost scheme considers the costs involved in resource price and power consumption at data center \( L_i \) for reserving resources to a given user. The scheme places VDs on data centers such that every VD is assigned \( R_{\text{min}} \) amount of resources in order to decrease cost and increase the energy efficiency of a VDC. Though the user QoE perceived by VD user may not be optimum, the scheme increases the \( U^{Net} \) of the system since it reduces operational costs and allocates energy-conserving amount of resources.

2) Least Latency: Least Latency is a utility maximization scheme which maximizes user QoE by placing VDs with close to \( R_{\text{max}} \) resources on the nearest distance data centers from request locations. In other words, the perceived user QoE is inversely proportional to the function of latency (greater the latency, lesser is the user QoE). The scheme provides a high \( U^{Net} \) due to increased user QoE with latency-awareness between thin-client sites and data centers.

3) Least Load: Least Load scheme reduces the number of VDs served on a particular data center and every VD is allocated \( R_{\text{max}} \) resources in a greedy manner. Also, the scheme tries to avoid an imbalance on the data center by migrating a heavily burdened data center’s VD request load onto another data center which has low resource utilization. Though the user QoE is high through the load-awareness, the scheme results in a low \( U^{Net} \) value as the number of VD requests served are minimum due to \( R_{\text{max}} \) allocations.

4) Least Joint: Least Joint scheme combines the optimizations of the above three placement schemes and tries to achieve a higher online VD placement utility. The scheme considers all the factors that affect the VDC system’s performance such as server side utility, resource price, energy cost, network latency, and network/server-side load balancing.

The detailed pseudocode of the Least Joint algorithm is shown in Algorithm 1. Given that we do not know the nature of arrival of random VD requests, we provision the VDs sequentially in an online manner by executing the algorithm to determine the data center location \( L_i \) and the resource \( R_{\text{set}} \) \( (R_{\text{min}} < R_{\text{set}} < R_{\text{max}}) \) required by a VD request \( vd_i \). The algorithm chooses the network path providing maximum end-to-end bandwidth between the user site and a data center in an opportunistic manner. In a virtualized network aggregate within a VDC, a single physical network can be broken down into several logical networks \{\( P_{1,1}, P_{1,2} \ldots P_{k,k} \)\} available from a user edge-switch \( i \) to every data center where, each path is a set \{\( I_1, I_2 \ldots I_k \)\} of interconnects that forms the path. We choose the path that provides maximum standard deviation from the required bandwidth usage \( B_{\text{network}} \) as the candidate path from a user site to the preferred data center.

Such a data center \( L_i \) selection is similar to the Hill Climbing Algorithm implemented in [18] that also uses the concept of maximum standard deviation. \( U^{\text{new}} \) and \( U^{\text{cur}} \) represent the new and current utility values at a data center \( L_i \), and \( \Delta U_i \) represents the increase in utility at \( L_i \). The \( U^{\text{new}} \) at \( L_i \) is calculated using the utility function \( U_{\text{vd}} (R_{\text{vd}}, i) \) for path \( P_{i,k} \). If the \( U^{\text{new}} \) is greater than the cost involved in reserving the resources on the data center (expressed as a factor of amount of resources needed) and the new \( \Delta U_i \) is greater than its current local value, \( \Delta U_i \) is updated as the current maximum utility increase. Once the above steps are performed iteratively for all the available data centers, the data center \( L_i \) that gives rise to maximal \( \Delta U_i \) is selected for placing the VD request.

B. The Defragmentation Problem

As mentioned in Section I and as evident from Algorithm 1, the opportunistic placements do not perform in-depth, computationally-intensive and time-consuming checks to determine whether every placement decision is globally optimal and leads to maximal-utility in a VDC; they use high-level information about resource status in data centers to rapidly make placement decisions and provide lower response time for user access to a VD. We can envisage a case where the above online opportunistic scheme can be used to handle VD requests till the VDC system reaches a saturation state beyond which, any additional VDs allocated will violate SLAs.

Fig. 2 illustrates an example VDC saturation state scenario, where resource fragmentation can be seen to impact the performance and scalability within a VDC system. Fig. 2(a) shows a situation when a VD request \( vd_{15} \) belonging to a particular user group (uniquely identified by the size and color pattern) that is best suited for maximizing quality at Location \( L_1 \) is allocated by a VDC’s ‘Unified Resource Broker’ to Location \( L_1 \) due resource unavailability at \( L_2 \). This creates a situation where the placement is sub-optimal if the resources available at \( L_2 \) are not adequate for high VD performance. Also, Fig. 2(b) shows a situation where a VD request \( vd_{21} \) is rejected due to lack of resources at both \( L_1 \) and \( L_2 \) in the VDC saturation state (i.e., there is no resource capacity hole large enough to accommodate \( vd_{21} \)). At this point, the VD application workload dynamics, login/logout trends of user, and possible user mobility could get the VDC system out of the saturation state. For example, Fig. 2(c) shows a situation where one of the VDs \( vd_{17} \) already placed at \( L_2 \) departs (i.e.,
user is finished with application tasks and the VD is swapped out), creating a resource hole or fragment. In this scenario, we can solve the previous fragmentation problem by migrating \(v_{d15}\), that was originally placed at \(L_1\). In Fig. 2(d), since \(v_{d15}\) migrated to \(L_2\), defragmented resources are now available at \(L_1\) to accept the earlier rejected request \(v_{d21}\).

Although VDs can be migrated to new locations in order to improve the Net Utility as seen in previous scenario, we make a cautionary remark that migrating a VD costs energy and resources, and we need to consider the cost and benefits of migration before we make the decision for a VD to migrate. More specifically, we need to migrate only those VDs that generate a positive benefit after incurring migration costs that can be simply modeled as shown in Eqn. 1, where: (i) a snapshot image creation cost \((Sc_i(dc_i))\) is incurred at the current datacenter, (ii) a deployment cost \((Dc_i(dc_j))\) is incurred for the VD profile and data to be moved to the other (future) datacenter, and (iii) a network cost \((Nc_i(t))\) is incurred for the VD depending on for e.g., the time of the day.

\[
Mc_i = Sc_i(dc_c) + Dc_i(dc_f) + Nc_i(t) 
\] (1)

In general, the defragmentation is a combinatorial problem where optimal resource allocation depends on the order of the VD arrival and is a directed instance of the Generalized Assignment Problem [19]. To state the problem more formally, we provide the following definitions for a generic scenario comprising of \(n\) resource users (i.e., users with VD requests) and \(m\) resource providers (i.e., CSP with data centers):

- Let \(r_i\), where \(i = 1 \ldots n\), be the \(m\) dimensional vector presenting resource request from \(i^{th}\) resource user where \(r_{i\min} \leq r_i \leq r_{i\max}\)
- Let \(b_j\), where \(j = 1 \ldots m\), be the capacity available to the resource provider \(j\)
- Since the total amount of resource is fixed in the VDC, \(\sum_{i=1}^{n} r_{ij} \leq b_j \quad \forall j\)

The objective is then to find the optimal resource allocation \((r_1, r_2, r_3, \ldots, r_n)\) which will minimize the cost function:

\[
\min f(r) = \sum_{i=1}^{n} f_i(r_i)
\]

where

\[
f_i(r_i) = -Ur_i + Mc_i
\]

The optimization function can be further simplified by applying the Lagrangian relaxation with the help of dual variable \((\lambda)\) as follows:

\[
\min f(r) = \sum_{i=1}^{n} f_i(r_i) + \lambda^T \left(\sum_{i=1}^{n} r_{ij} - b_j\right)
\]

s.t \(\lambda \geq 0\)

We need to solve the above relaxed problem in a distributed fashion, where each resource user \(i\) solves \(f_i(r_i) + \lambda^T r_i\).
IV. MARKET-DRIVEN PROVISIONING AND PLACEMENT FOR DÉGRÄMENTATION

In this section, we present details of our MDPP scheme that we propose as the defragmentation solution within VDCs. We start with approach preliminaries statement, and next present the algorithm for distributed market-driven optimization to improve Net Utility of a VDC system that is in a saturated state, and is experiencing dynamic changes in supply and demand characteristics due to VD request arrivals as well as deallocations after completion of users’ VD application tasks.

A. Approach Preliminaries

Our solution approach is based upon the seminal result of economic theory viz., given proper conditions, a market will produce an optimal distribution of resources with a minimum of transaction costs. Our VDC market can be considered to be a system with locally interacting agents that are buying and selling with limited information exchange, in order to achieve an overall global behavior of fair resource allocation and stable prices. Moreover, our market-driven approach is additionally suitable for VDC systems due to its flexibility in handling cases where there are dynamic changes in demand/supply relation (i.e., VD requests are continuously arriving and getting swapped out). We remark that our primary objective in this paper is not to explore game-theoretic properties, but rather focus reasonably on realistic, intuitive and understandable behavior based on everyday experience in a VDC service environment. We thus make the simplifying assumption that agents do not speculate about how their own behavior or that of other agents will affect the market.

B. Distributed Market-Driven Optimization

In real-world market scenarios, we can view the utility maximization as an agent-level process of determining ideal resource quantities that provide maximum gain to the agent. Similarly, the cost minimization process captures the buying agent’s sentiment to explore the minimum amount it has to pay to maximize the profit out of the bargain for the identified ideal resources. The utility maximization step in a VDC system corresponds to a behavior in which every virtual desktop (vd) searches for the ideal resource provisioning in order to maximize its utility while the cost minimization step relates to determining a location where, the least cost of resource consumption is observed. We propose a “Bid” (B) metric which folds-in the utility maximization and cost minimization into one quantity. It allows for identification of ideal resource requirements ri for vd and finds an optimal data center l by selecting a data center which produces minimal “Bid” value, where “Bid” value per data center is computed by the formulation given below:

\[ B_{ij} = \min \left(-U_i(r_{ij}) + \lambda_j^T r_{ij} + \sum_{g=1}^{N_g} (Q_i - Q^g)^2 + M_{cf} + \sum_{i \in N_{vd}} (U_i - U_j)^2 \right) \]  

Hence, every vd computes its’ final “Bid” value (Bi = \min{B_{ij}}) by iterating over all data centers.

The above Eqn. 2 consists of several major components that are explained in detail below:

- **Utility (Ui):** We model the utility of vd as a measure of both the provisioning and placement decisions. The provisioning information is supplied via a resource request vector (ri) which is an m dimension vector; each dimension refers to one type of resource available in the system. The placement information is inherent since we are modeling “Bid” value for every data center.

- **Resource Usage Cost (\(\lambda_j^T r_{ij}\)):** The resource usage cost is a measure for the cost (Ci,j) associated with accessing resources given by a resource request vector (ri) from data center j. The price vector (\(\lambda_j\)) is also a m dimension vector where each dimension refers to the price of one unit of one type of resource at data center j. Hence, the dot product of price vector with resource request vector produces the total resource usage cost of accessing ri resources from data center j.

- **Quality Constraint (\(\sum_{g=1}^{N_g} (Q_i - Q^g)^2\)):** We use additional constraints for intra usergroup quality criteria which focuses on preserving equal user QoE levels or quality fairness for the VDs belonging to the same usergroup or desktop pool. We compute the quality as a composite function of resource provision vector (ri).

- **Migration Cost (M_{cf}):** The migration cost as depicted in Eqn. 1 captures the cost of migrating vd, from previously allocated data center c to the current data center under consideration f.

- **Utility Constraint (\(\sum_{i \in N_{vd}} (U_i - U_j)^2\)):** Utility constraint is similar to Quality constraint and enforces equal utility for all virtual desktops belonging to same usergroup where \(N_{vd}\) is the set of virtual desktops in i’s user group.

Each of the components contribute linearly to the final “Bid” value computed by a virtual desktop vd. Therefore, at the end of first step, we have a \(n \times m\) Bid matrix, where \(n\) is the number of virtual desktops and \(m\) is the number of resources, and reflects the resource provisioned vectors for all virtual desktops. The Bid matrix is passed to the VDC’s Unified Resource Broker (URB) (whose role to dispatch VD allocations is illustrated in Fig. 2) for further processing and resource allocations to particular data centers.

C. Determining the Maximum Net Utility and Cost Minimal Migration Set

Our market-driven approach in our MDPP scheme is an iterative approach and is strongly dependent on the terminating criteria. Our MDPP scheme implementation uses a complete satisfaction of the Bid matrix as a terminating criteria. The pseudocode of our MDPP scheme is shown in Algorithm 2, which iterates until bids (resource requests) from all virtual desktops are completely met. Note that the URB is the only entity which has complete knowledge of each and every incoming bid request. Hence, the terminating criteria is embedded in the URB’s core logic and thus, the URB works as a decision maker on final allocations. If the URB fails to satisfy all the requests from the Bid matrix, then it rejects the current Bid matrix and ask all virtual desktops to submit their bids again, until all bids are satisfied.
The URB also works as a liaison between data centers (or resource providers) and virtual desktops (or resource consumers). The important aspect of the URB being an intermediary is to communicate price per resource at every data center to all virtual desktops. In every iteration, the price associated to resources at every data center fluctuates based on the demand and supply relation for the resources at corresponding data centers (i.e., VD request arrivals as well as deallocations). The URB tracks all available resource at every data center and it also computes demand of the resources at every data center based on the Bid matrix. Hence, the URB is the sole judiciary to modify the price per unit of resource at every data center. We formulate the price update rule as a sub-gradient method based on the Lagrangian relaxed problem. The new price vector which is a \( m \) dimension vector for data center \( j \) is computed as follows:

\[
\lambda_j(t + 1) = \max \left( 0, \lambda_j(t) + \alpha \left( \sum_{i=1}^{n_j} r_{ij}^t - b_j \right) \right)
\]  

(3)

The update of the Lagrangian multipliers or the prices of the resources, as shown in the Eqn. 3 is in the sub-gradient direction which in economic terms, is the update according to the demand and supply of the resources. The \( \alpha \) term is introduced since the prices of the resources or the Lagrangian multipliers are always non-negative. The term \( \alpha \) in the above equation is the step size or the rate of price update which is chosen according to guidelines in [20]. The prices stabilize when \( \left( \sum_{i=1}^{n_j} r_{ij}^t - b_j \right) \rightarrow 0 \), which indicates balance in demand \( (r_{ij}^t) \) and supply \( (b_j) \).

**Algorithm 2 Market-Driven Placement and Provisioning**

1. **Input:** Set of VD requests placed in the system \( \{vd_1, ..., vd_n\} \)
2. **Output:** Resource reallocations \( R_{vd_i} \) for VDs \( vd_i, i \in \{1 \ldots n\} \) that maximize Net Utility \( \bar{U}_{net} \)
3. **begin procedure**
4. **While**
5. **Do**
6. Calculate Bids \( (B_{il}) \) for each \( vd_i \) using the Lagrangian Relaxation for each data center \( dc_l \)
   \[
   B_{il} = \min \{ -U_i(r_{il}^t) + \lambda r_{il}^t + \sum_{g=1}^{N_g} (Q_i - Q^g)^2 + Mc_f + \sum_{i \in N_{dc_l}} (U_i - U_l)^2 \}
   \]
7. Calculate the Bid \( (B_i) \) for each \( vd_i \)
8. **end While**
9. **Do**
10. Until all bids are satisfied
11. **end Do**
12. **end procedure**

**D. Comparison with Centrally-Driven Heuristic Schemes**

In this subsection, we compare our MDPP scheme’s resource allocation approach with centrally-driven heuristic optimization schemes, which include our previous UOPP scheme [5].

For a given online saturated state after opportunistic resource allocations, our previous UOPP scheme uses a centralized greedy heuristic with utility gain ordering coupled with a ‘migration cost-benefit analysis’, wherein the migration cost is the same as shown in Eqn. 1. The URB in the UOPP scheme pools resources across all data centers and identifies positive VD pairs for migration such that the new utility achieved is greater than the cost involved in migration. The core principle used in our UOPP scheme is the principle of minimizing the loss of utility due to sub-optimal placements. The UOPP scheme assumes we know through the utility functions of a VD user group (utility function values for different user groups shown later in Table I) that there exists an optimal upperbound of utility for each VD if the resources are not fragmented across data centers, and ideal resources are allocated to each VD. For further details regarding the UOPP scheme, the reader is suggested to refer [5].

Fig. 3 shows an illustrative example of the algorithmic implementation of the UOPP scheme. For the sake of illustration, we assume three arriving VD requests and three data centers which can provision \( R_{max} \) for \( j \in J \) resources to achieve maximum composite quality \( (Q_{max}) \) for every VD request. This can be extended to implementations with multiple data centers and multiple VD requests as shown later in Section V. Fig. 3 shows the placement performed by ‘online’ Least Joint placement scheme which produces Net Utility of 382 units until the system reaches a saturation state. The ‘offline’ (centrally-driven greedy heuristic based) UOPP defragmentation scheme performs optimized VD reallocations that results in increased Net Utility of 415 units. It is self-evident from the example that the UOPP algorithm achieves a better placement with cost-aware positive pair migration, as compared to the migration-free ‘on-line’ Least Joint scheme.

Fig. 4 shows an illustrative example to demonstrate the defragmentation performance of our ‘off-line’ MDPP scheme in comparison with the ‘on-line’ Least Joint scheme. In this
example, we assume a scenario of three data centers and three incoming VD requests similar to our previous UOPP scheme illustration. Also, resources available at each data center are sufficient to provision $R_{j\text{max}}$ for $\forall j \in J$ which results in maximum composite quality ($Q_{\text{max}}$) for every VD request. As seen from the figure, the Least Joint scheme achieves a Net Utility of 382 units until the system reaches a saturation state. Further, our proposed off-line MDPP defragmentation scheme captures the cost matrix across the data centers for providing the resource bid values of the bidding agents or VD requests across the data centers. Using much lesser number of iterations, it creates the optimized cost matrix along with bid values for every VD request. The decision matrix is based on the lowest bid values obtained for every VD request to every data center, and the VD request is placed at the data center which gives rise to the least bid value. The MDPP scheme thus achieves a significant increase in Net Utility summing to 430 units, compared to the previous 382 units obtained through the Least Joint scheme.

Fig. 5 shows another illustrative example to further demonstrate defragmentation advantages of the distributed optimization using the MDPP scheme over the UOPP scheme. In this example, we assume a scenario of four data centers and four incoming VD requests for the sake of study purposes. Also, resources available at each data center are sufficient to provision $R_{j\text{max}}$ for $\forall j \in J$ which results in maximum composite quality ($Q_{\text{max}}$) for every VD request. As seen from the figure, the placement of VD requests on data centers by the Least Joint scheme produces Net Utility of 570 units. The UOPP scheme provides relocations that results in Net Utility of 580 units, while the MDPP scheme produces a Net Utility of 600 units. It is self evident from the example that our proposed MDPP scheme achieves highest Net Utility supporting our defragmentation necessity claim based on the global optimization of virtual market of bidding agents. We are thus able to hypothesize improved performance gain with market-driven optimization, as compared to greedy heuristics because the MDPP scheme achieves a system balance by capturing the consumers interest for providing fair user QoE, while increasing the system scalability for CSPs.

In addition to UOPP, we can envision other greedy heuristic schemes such as ‘Round Robin’ and ‘Random Walk’ defragmentation that are relatively naive. In the Round Robin case, the relocations can be performed by placing the VD requests sequentially across data centers just with the constraint that one VD request per data center is allocated in every round. In the Random Walk case, the relocations can be performed by placing the VD requests randomly across data centers with no other additional constraints. Further, we can assume a variant of the MDPP scheme viz., ‘Security-Driven Provisioning and Placement’ (SDPP), where data centers can be labeled with security levels (e.g., least, medium and maximum-security) based on for e.g., the number of public-facing network interfaces or other such security considerations. In the SDPP case, VD resource location constraints are imposed on the MDPP scheme, and thus there will be sub-optimal placements that affect the Net Utility after defragmentation. In the following section, we compare the improved Net Utility obtained with MDPP defragmentation with relation to these other defragmentation schemes using extensive simulations.

V. PERFORMANCE RESULTS

In this section, we first describe the simulation setup along with the parameters considered for the defragmentation performance comparison of our MDPP scheme with other schemes.
Following this, we describe the metrics against which we evaluate the different schemes. Finally, we discuss the obtained results from simulation experiments.

A. Simulation Environment

Simulation Setup: Our simulation setup consists of a number of data centers, where each data center has several servers of fixed capacity. The number of data centers, number of servers per data center, and the capacity of each server are specified using a configuration file. An initialization script generates a 'VD-request-sequence', which is a sequence of allocation and de-allocation requests, belonging to different user groups, generated randomly based on a allocation-to-deallocation ratio parameter. Each data center and the VD-Client (from user site) request also has a physical location which can be specified with X and Y co-ordinates in a 2D-space. The location of the data center is generated either randomly or specified manually. The VD-clients are allocated a physical location randomly in the 2D-space. The physical locations are used to calculate the distance between data centers and VD-Clients, and these distances are used for estimating the network latency in the simulations.

Configuration Considerations: We considered different configurations of the VDC system in our simulation experiments. As a common configuration, we used 3 data centers based on our discussions with VDC infrastructure experts at IBM, Dell and VMware. Given the performance of thin-client protocols over long-distances, and challenges in synchronizing user profiles across many data centers for desktop delivery, it is common to have 3 data centers within the geographical size of USA. In some simulation configurations, we increased the number of data centers to show our scheme effectiveness at larger scale.

We consider different CPU, memory and network bandwidth capacities in the data centers such that they can support different VD request loads of different user groups based on their respective utility functions. We simulate VDs belonging to three different user groups, with different resource profiles that satisfy diverse user QoE needs. Table I shows the $R_{\min}$ and $R_{\max}$ values of the CPU, memory and network bandwidth resources considered in our simulation experiments that correspond to the VD application profiles of the user groups obtained from a real VDC testbed [6]. We used allocation to deallocation ratio of 3:1, such that for every 3 VD allocations, we assume there will be 1 VD deallocation in order to capture a dynamic VDC system with user login and logout trends.

<table>
<thead>
<tr>
<th>User Group</th>
<th>CPU (GHz)</th>
<th>Memory (GB)</th>
<th>Network (Mbps)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$R_{\min}$</td>
<td>$R_{\max}$</td>
<td>$R_{\min}$</td>
</tr>
<tr>
<td>Engineering Site</td>
<td>0.7</td>
<td>1.5</td>
<td>0.35</td>
</tr>
<tr>
<td>Distance Learning Site</td>
<td>0.5</td>
<td>1.3</td>
<td>0.2</td>
</tr>
<tr>
<td>Campus Computer Lab</td>
<td>0.3</td>
<td>1.4</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Utility versus Latency: We also consider the network latency while calculating the utility of the individual VDs. Network latency is an important factor that impacts interactive response time performance within thin-client protocols. We model the network latency as scaling factor that degrades the ideal user QoE, i.e., the user QoE that would be generated by the resources allocated to them, under ideal network conditions. Although the quality of all user groups decreases as latency increases, the rate of decrease is not necessarily the same for each user group. From our empirical studies in [6], we have observed that the Distance Learning group will have the steepest decline, since users in this group mainly use video streaming and web-conferencing applications that are more interactive in nature, and thus the video quality decreases sharply as latency increases. In contrast, the Engineering Site group has relatively slower decay, since users in this group mostly use computationally heavy applications for design tasks, without much network resource demands.

B. Metrics

We evaluate the different defragmentation schemes viz., MDPP, UOPP, SDPP, Round Robin and Random Walk using the following performance metrics:

1) Quality Gain: This metric reflects improvement in the quality of service after defragmentation. This metric is useful to differentiate improvements achieved from dual variable based provisioning in our proposed MDPP scheme with respect to the Least Joint scheme. The metric reflects percentage gain rather than the absolute gain.

$$Quality \text{ Gain} (Q_G) := \frac{Q_{\text{new}} - Q_{\text{old}}}{Q_{\text{old}}} \times 100$$

2) Utility Gain: This metric compares the Net Utility of the system after defragmentation. It is used to capture the gain achieved via a global, offline optimization process with the schemes being compared over the initial online placement that caused a saturation state. This metric also reflects the percentage gain rather than the absolute gain.

$$Utility \text{ Gain} (U_G) := \frac{\sum_{i=1}^{n} U_{\text{new}} - \sum_{i=1}^{n} U_{\text{old}}}{\sum_{i=1}^{n} U_{\text{old}}} \times 100$$

3) Migration Count: Migration count reflects the number of migrations suggested for the defragmentation. Since this metric computes the number of essential migrations dictated by a given scheme, it is an indirect representation of a scheme’s performance.

4) Migration Cost: For this metric, we use the formulation given in Section IV-D to model migration cost as a composite function of snapshot cost, deployment cost and network cost.

5) Benefit: This metric is the determining attribute which indicates whether a set of VD migrations will actually result in increased performance and scalability for the CSPs, and utility (user QoE level) increase for the consumer. The metric can be calculated by subtracting the utility rise across all data centers with the cost factor associated with the VD migrations.

C. Results

For our analysis, we generated different system saturation scenarios using different placement schemes. We simulated variation in demand for VDs in ascending fashion which indirectly forced provisioning to vary from $R_{\max}$ to $R_{\min}$.
TABLE II
SUMMARY OF USER GROUP MIGRATION ANALYSIS

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Campus Computer Lab</th>
<th>Engineering Site</th>
<th>Distance Learning Site</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defragmentation Schemes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Allocated</td>
<td>38</td>
<td>30</td>
<td>28</td>
</tr>
<tr>
<td>Total Deallocated</td>
<td>2</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Total Migrated</td>
<td>7</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>Net Utility Before Defragmentation</td>
<td>8.39</td>
<td>8.20</td>
<td>5.82</td>
</tr>
<tr>
<td>Net Utility After Defragmentation</td>
<td>10.23</td>
<td>12.10</td>
<td>7.10</td>
</tr>
<tr>
<td>Migration Cost</td>
<td>2.7</td>
<td>8.7</td>
<td>0.2</td>
</tr>
<tr>
<td>Net Utility Benefit</td>
<td>1.83</td>
<td>3.57</td>
<td>7.9</td>
</tr>
<tr>
<td>Net Utility Percent Increase</td>
<td>21.92</td>
<td>44.27</td>
<td>79.49</td>
</tr>
</tbody>
</table>

Fig. 6. Comparison of Net Utility results for individual user groups with increasing VD request loads

Fig. 7. Results showing benefits of defragmentation on Net Utility for the main schemes comparison

Fig. 8. System Net Utility results for increasing number of data centers

in observed saturation states. Figs. 6(a), 6(b) and 6(c) show the Net Utility results of various offline and online placement schemes for all three user groups.

As shown in Fig. 6(a) for the Campus Computer Lab user group, MDPP achieved maximum rise in Net Utility when compared to other offline schemes. This is attributed to the fact that MDPP ensures equal utility values for all users until the bids of all VDs are satisfied. Also, as the number of VD requests increases, the Net Utility of all VDs were increased for MDPP scheme, which is in contrast to other centralized heuristic schemes which decreased the system Net Utility as the number of VDs increased. In this scenario, the MDPP still maximizes the quality of every VD user and satisfied every VD’s bid as the load increased by taking advantage of the fact that the CPU, RAM and network requirements of campus computer lab users are comparatively less.
Fig. 6(b) shows the Net Utility results of Engineering Site users whose VD quality requirements are more demanding and user QoE expectations are higher. As shown in the figure, even though the MDPP Net Utility value is comparatively high when compared to other offline schemes, Net Utility values of all schemes decreased as the VD load increased. However, we note that the decrease in Net Utility value for MDPP was more graceful compared to the other schemes. The decreasing Net Utility is due to the fact that the combined CPU, RAM and network requirements of Engineering Site users are relatively high and providing the desired VD quality for every user as the load increases is a challenging task as the system exhausts resources quickly, compared to the other two user groups.

Fig. 6(c) shows the Net Utility results of Distance Learning Site users for the MDPP scheme evaluated against other centralized heuristic schemes. While the rest of schemes’ Net Utility consistently dropped for increasing VD load, our MDPP scheme maximized the Net Utility until the VD requests reached a load of 270. However, Net Utility slightly dropped for further increase in VD load. This is due to the fact that the Distance Learning Site user’s networking requirements are relatively more demanding, which in turn impacts the Net Utility as the VD load increases.

Fig. 7 captures the overall VDC System Net Utility combined for all the three user groups. We can observe that the MDPP scheme has the least decrease in Net Utility as compared to the UOPP and SDPP schemes. Additionally, Fig. 8 shows the Net Utility for increasing number of data centers plotted for varying VD request loads. For lower VD load settings such as 80, there is slight increase in Net Utility (shown by Slope-1) as the number of data centers increases. This is because, the optimal set of resources are provided for the VD load as the number of data centers is increased to 9. However, the highest VD load of 340 achieved a notable increase in Net Utility (shown by Slope-2) as the number of data centers increased. This is because the optimal set of VD resources is provisioned for all VDs only when more number of data centers are available to place the VD requests. Lastly, Table II summarizes the user group migration results of UOPP, SDPP and MDPP schemes for a VD request load of 110. We can observe that MDPP outperforms UOPP and SDPP obtaining as much as 44%, 85% and 79% in terms of Net Utility rise, respectively as compared to 21%, 6% and 21%, and 29%, 47% and 62% obtained using UOPP and SDPP schemes, respectively for the three user groups.

VI. CONCLUSION AND FUTURE WORK

In this paper, we identified an important problem of resource fragmentation that occurs in virtual desktop cloud (VDC) data centers similar to memory or disk fragmentation in traditional operating systems that can impact performance and scalability. We proposed a defragmentation solution that uses distribution optimization principles, and is called as the Market-Driven Provisioning and Placement (MDPP) scheme. At the core, MDPP performs resource allocation based on VDs’ bidding values in a virtual VDC market, and helps achieve fair allocation and stable prices.

We showed that MDPP out performs existing centralized heuristic schemes in achieving global optimization in a VDC system in terms of a Net Utility metric. MDPP scheme can also achieve high Net Utility, even when VD resource location constraints are imposed to meet orthogonal security objectives. Our evaluation has analytical rigor through application of the Lagrangian relaxation method. Moreover, our simulations build upon utility functions and user profiles obtained from a real-world testbed, and feature realistic application behavior based on everyday experience in a VDC. Similar to our UOPP implementation and optimization results in a realistic VDI testbed [7] built using VMware virtual desktop infrastructure technologies, MDPP can be implemented within URBs of actual VDC environments. Consequently, our evaluation and implementation readiness of MDPP provides insights to VDC service providers in terms of how to improve the scalability and performance of their Desktop-as-a-Service offerings.

Our future work involves extending our defragmentation approach in other cloud computing environments that are different from the VDC environment in user workload characteristics, user QoE needs, and service provider’s cost factors.

REFERENCES