Data-locality-aware User Grouping in Cloud Radio Access Networks

Weng Chon Ao and Konstantinos Psounis, Fellow, IEEE

Abstract—Cellular base band units of the future are expected to reside in a cloud data center which provides computation resources, content storage and caching, and a natural place to perform multi-user precoding, thus addressing both cost and performance concerns of cellular systems. Multi-user precoding relies on efficient user grouping schemes to maximize multiplexing gains. However, traditional user grouping schemes are unaware of data center constraints, and may induce a large number of data transfers across racks when fetching requested data to a certain rack for precoding. When congestion occurs in the data center network, the delay of data transfers across racks may exceed the channel coherence time. This would kill multi-user MIMO transmissions as channel state information becomes outdated.

In this paper we design novel data-locality-aware user grouping schemes which preferentially group users whose requested data are located under the same rack. We also design user grouping algorithms which adapt to the congestion level in the cloud data center. Specifically, a regularized spectral efficiency maximization problem is proposed where the number of data transfers across racks is introduced as a regularization term. By adjusting the weight of the regularization term according to the congestion level, we gradually suppress data transfers across racks in forming user groups when congestion occurs. We reduce the above problem to a soft-capacitated facility location problem and we devise a 2-approximation user grouping algorithm. Last, we conduct simulations which show that our proposed algorithm performs close to the optimal in practical scenarios, and study the tradeoff between higher spectral efficiency and lower data transfer cost.

Index Terms—cloud radio access network, data center, cellular network, multi-user MIMO, user grouping, approximation algorithm

I. INTRODUCTION

In the cloud radio access network (C-RAN) architecture, base band units (BBUs) are centralized as a cloud data center (called BBU pool) separated from the remote radio heads (RRHs) deployed at base stations (BSs) to reduce energy consumption and better utilize computation resources [1], [2], see Fig. 1a. The centralized BBUs can not only provide computation resources to serve a large number of BSs but also enable coordination among BSs to increase spectral efficiency by using techniques like Coordinated MultiPoint (CoMP) [3]. In addition to providing computation resources, content data can be stored or cached at the BBU pool [4], [5]. For example, the content data can be recent movies and TV series hosted and organized by Netflix.

A cloud data center consists of a large number of racks. See, for example, the Telco cloud proposed by Nokia [6], [7]. In each rack, there is a top-of-rack switch that connects to a large number of servers (for both computing and data storage). Moreover, a core switch connects to the top-of-rack switches to enable data transfers across different racks, see Fig. 1b. To perform computation over data such as precoding, the data needs to be first fetched to a local drive. It is typically more expensive to fetch data across different racks than under the same rack because data transfers across racks are more susceptible to congestion and more prone to delay [8], [9].

In the BBU pool, a major computation task is to perform precoding over different data streams requested by different users to enable spatial multiplexing via multi-user MIMO [10] or massive MIMO [11]–[13]. Since most of today’s BSs are equipped with at most 16 antennas, e.g., in the current LTE standard [14] codebooks for precoding are defined for up to 16 antenna ports, we focus the discussion in the multi-user MIMO case, enabled via Zero-Forcing BeamForming (ZFBF) precoding, though our analysis can be readily extended to the massive MIMO setting as well where other precoders are more popular, e.g. conjugate beamforming.

ZFBF requires to select a set of users to be served concurrently, an operation commonly referred to as user grouping, and it also requires instantaneous channel state information (CSI) to be collected to perform the precoding. Traditional user grouping schemes are not aware of the data locality in data centers, that is, of where the data reside. Instead, their task is to minimize the overhead from collecting CSI and/or to maximize the multiplexing gain. For example the so-called randomized and round-robin schemes select users randomly or in a round-robin fashion, aiming to minimize overhead since CSI is collected only for the selected users once the group is formed. In contrast, in CSI-based user grouping [15]–[17] it is

Fig. 1: (a) C-RAN architecture. (b) BBU pool: a cloud data center for efficient computation and content caching.
required to collect CSI from a large number of users and then select a subset of users with semi-orthogonal channel vectors to form a group, aiming to maximize the multiplexing gain.

Multi-user ZFBF precoding with CSI-based user grouping has higher chances to achieve the maximum spectral efficiency. That said, it yields a higher data rate only if the turn-around time between CSI feedback and the actual data transmission is smaller than the channel coherence time. Otherwise, the CSI becomes outdated, causing significant performance degradation. And, the turn-around time might be large because after collecting CSI and selecting the users, we then have to fetch the requested data of the users to a local disk in a certain rack to perform precoding, as these data may be located across different racks in the data center. Depending on the congestion level in the data center, the data fetching time ranges from hundreds of microseconds to tens of milliseconds [8], [19]. This can be larger than the channel coherence time, since, depending on the user speed and the carrier frequency, the channel coherence time ranges from milliseconds (high mobility users) to tens or hundreds of milliseconds (low mobility or static users) [14].

Therefore, when congestion occurs in the cloud data center network, a reduction in the number of data transfers across racks is required to guarantee that the overall turn-around time is smaller than the channel coherence time and the wireless transmissions are successful. However, all the above traditional user grouping schemes are independent of where the requested data is located, potentially inducing a large number of data transfers across racks. With this in mind, we propose user grouping schemes under the C-RAN architecture which take into consideration where the data is located in the data center. Thus, we preferentially group together users whose requested data are located under the same rack, and only transfer data across racks if the ensuing rate increase is sizable. This significantly reduces the number of data transfers across racks, while still providing high wireless spectral efficiency through spatial multiplexing.

The wireless spectral efficiency becomes higher as we allow more data transfers across racks to form user groups with better-conditioned channel matrix as long as the CSI does not become outdated. On the other hand, when congestion occurs in the cloud data center, the inter-rack data fetching time increases such that the overall turn-around time may become a large fraction of (or even larger than) the channel coherence time and the CSI may become outdated, reducing the spectral efficiency. To devise a user grouping algorithm that is able to adapt to the congestion level in the cloud data center network, we propose a regularized spectral efficiency maximization problem where the number of data transfers across racks is introduced as a regularization term. When there is no congestion in the cloud data center, one can simply set the weight of the regularization term to zero, allowing arbitrary number of data transfers across racks to maximize the wireless spectral efficiency (in this case our problem reduces to the traditional CSI-based user grouping). On the other hand, when congestion is high, one may impose a large weight to regulate the inter-rack data transfers and settle for “local” user groups. By adjusting the weight of the regularization term according to the congestion level, data transfers across racks can be controlled to achieve the desired system operating point.

The regularized spectral efficiency maximization problem is NP-hard. Motivated by this, we first introduce a simplified form of the problem (referred to as the regularized resource block minimization problem), where we use the number of resource blocks needed to serve a fixed number of users as a proxy for the spectral efficiency. Then, we reduce this problem to a soft-capacitated facility location problem [20], [21], and devise a 2-approximation algorithm to efficiently solve it.

The remainder of this paper is organized as follows. We present related work in Section II. Section III describes the system model and presents motivating examples. The problem formulation and performance analysis are given in Section IV. We extend our formulation to various practical scenarios of interest in Section V. Section VI presents numerical and simulation results. Last, Section VII concludes the paper.

II. Prior Work

An introduction of the C-RAN architecture and the advantages of BBU centralization can be found in [1], [2] and in references therein. Such BBU centralization facilitates base station coordination like CoMP [3]. The concept of the BBU pool acting as a cloud data center for both computation and content caching or storage is introduced in [4], [22], [23]. Multiple recent works have evaluated the performance of the C-RAN architecture using simulations, real-world data, software-defined radios, and even prototypes, see, for example, [6], [7], [24]–[27].

A major task of the BBU pool is to perform multi-user ZFBF precoding over users’ requested data streams to enable multi-user MIMO [10]. A central piece of multi-user ZFBF is user grouping. The analysis of the performance of randomized user grouping for user scheduling in multi-user MIMO and massive MIMO wireless networks can be found in [12], [13]. CSI-based user grouping for multi-user beamforming was investigated in [15]–[17], where the users are grouped according to the instantaneous CSI to exploit the multi-user diversity. User grouping can also be based on the channel distribution information, the second-order channel statistics or some combination of the above approaches. For example, in [28], a two stage precoding scheme has been presented, where the users are first grouped according to the second-order channel statistics, and then, for the users that have the same such statistics, randomized user grouping is used to schedule a subset of users for multi-user ZFBF precoding.

CSI-based beamforming has been considered in a C-RAN with a focus on fronthaul optimization [10], [18]. Specifically,
in [10], the authors designed a sparse beamformer to minimize the fronthaul power consumption. In [18], the authors considered enhanced RRHs that are also embedded with caches and baseband processing units. By using superposition coding, the precoding can be decomposed, performed partly at the BBU pool and partly at the enhanced RRH subject to the fronthaul capacity constraint. As mentioned earlier, in this work we operate under scenarios where the fronthaul connections are fast enough [6], [7] and the fronthaul delay does not become a bottleneck in the turn-around time. Instead, we focus on the case where the data fetching time for precoding in the cloud data center becomes a bottleneck when congestion occurs.

To the best of our knowledge, this is the first work that takes into consideration the data locality information and the congestion level in the cloud data center in forming user groups for multi-user MIMO ZFBF precoding in a C-RAN.

### III. System model

Consider a BS in the context of the C-RAN architecture. (Section V discusses how to extend the discussion and results for multiple BSs.) Suppose that the BS has $\eta$ antennas and there are $M$ users under the coverage of the BS, each equipped with a single antenna. (See Section V for the case with multi-antenna users.) Denote the users by the set $M = \{1, 2, \ldots, M\}$.

Suppose that each of the $M$ users makes a single data request. We denote the data requested by user $i$ as $d_i$, $i = 1, \ldots, M$, which is stored in the BBU pool cloud data center. Suppose that there are $N$ racks (denoted by the set $N = \{1, 2, \ldots, N\}$) in the data center and assume that the requested data of a user is equally likely to be stored in any one of the $N$ racks. The event that the data requested by user $i$ is in rack $j$ is represented as $d_i \in r_j$ (we use the shorthand notation $r_j$ to refer to rack $j$). We denote the set of users whose requested data are in rack $j$ as $M_j = \{i : d_i \in r_j\}$. Note that $M_j$, $j = 1, 2, \ldots, N$ form a partition of the user set, where $M \cap M_j = \emptyset$, $i \neq j$ and $\bigcup_{j=1}^{N} M_j = M$. Our formulation can be extended to the case with data replication across different racks ($M \cap M_j \neq \emptyset$, $i \neq j$), see Section V for a detailed discussion.

#### A. User grouping

Since the base station has $\eta$ antennas, it can provide $\eta$ degrees of freedom for spatial multiplexing ($\eta$ is also called the spatial multiplexing gain) and can serve at most $\eta$ users in a resource block (time-frequency slot). For simplicity, we assume that the unit size of a user’s requested data is one resource block. User grouping refers to partitioning the set of users $M$ into $T$ groups, i.e., $S_1, S_2, \ldots, S_T$, where they are disjoint $S_i \cap S_j = \emptyset$, $i \neq j$, and the cardinality of each group is less than or equal to the degrees of freedom $|S_i| \leq \eta$. By using multi-user ZFBF precoding, the users in the same group will be served simultaneously in one resource block, so a total number of $T$ resource blocks will be used.

Assuming $\eta$ divides $M$, we partition the set of users $M$ into $\frac{M}{\eta}$ groups each of size $\eta$, to fully utilize the degrees of freedom for data transmission and minimize the number of resource blocks needed to serve all users. The partition criteria (i.e., how to partition the $M$ users into $\frac{M}{\eta}$ groups) for the randomized/CSI-based user grouping scheme depends on the randomization strategy or the instantaneous CSI of the users and will be discussed in Section IV-A. The partition criteria for the data-locality-aware user grouping scheme depends on the data locality (encoded by the sets $M_j$, $j = 1, 2, \ldots, N$): given a fixed number of $\frac{M}{\eta}$ resource blocks, we form $\frac{M}{\eta}$ user groups while minimizing the number of induced data transfers across racks, see Section IV-B. Since the resulting wireless rates depend not only on utilizing the maximum multiplexing gain (and thus the minimum number of resource blocks) but also on the degree of orthogonality among the users’ channel vectors and thus on the CSI [15], [16], in Section IV-C we extend the above scheme to also take into consideration the CSI when forming those $\frac{M}{\eta}$ user groups, and term the scheme a joint CSI- and data-locality-aware user grouping scheme. In Sections IV-D and IV-E, to further reduce the number of induced data transfers, we relax the constraint of using only $\frac{M}{\eta}$ resource blocks (i.e., we allow to form $T > \frac{M}{\eta}$ user groups). Since transferring data across racks takes time, if the associated delay is larger than the channel coherence time due to high congestion, the CSI will be invalidated and the resulting spectral efficiency will be significantly reduced.

Motivated by this, we devise two user grouping algorithms that can adapt to the congestion level in the cloud data center. First, we propose a regularized resource block minimization problem where we minimize a weighted sum of the number of resource blocks used and the number of induced data transfers across racks (see Section IV-D). Second, we propose a regularized spectral efficiency maximization problem where we consider the data rate directly (see Section IV-E). By adjusting the weight of the regularization term according to the congestion level in the cloud data center, we can suppress data transfers across racks in forming user groups when congestion occurs. To simplify the analysis we assume $\eta$ divides $M$ in all analytical derivations and relax this in Section VI where we present simulation results.

#### B. Multi-user ZFBF precoding and the need to transfer data

Let $h_i$ be the $\eta \times 1$ channel coefficient vector between the BS and user $i$ and let $H \doteq [h_1 \cdots h_M]^T$ be the $M \times \eta$ channel matrix. Fix a typical user group $S$. Let $H(S)$ be the $|S| \times \eta$ submatrix of $H$ where the rows correspond to the channel vectors of the users in $S$. The ZFBF precoding matrix is defined as $V \doteq H(S)^{\dagger}$, where $H(S)^{\dagger}$ is the pseudoinverse of $H(S)$. Let $d(t)$ denote the $|S| \times 1$ data vector corresponding to the data streams requested by the users in $S$ and $x(t)$ denote the $\eta \times 1$ symbol vector to be transmitted by the BS. We have $x(t) = Vd(t) = H(S)^{\dagger}d(t)$. Under the ZFBF precoding, the $|S| \times 1$ received signal $y(t)$ at the users is

$$ y(t) = H(S)x(t) = H(S)H(S)^{\dagger}d(t), $$

(1)

where we ignore the background noise. Note that when $|S| \leq \eta$ and $H(S)$ has full row rank, we have $H(S)H(S)^{\dagger} = I$ and thus $y(t) = d(t)$. In this case, we can see that by using ZFBF each user gets its own requested data stream without interference from other streams. Unless otherwise stated, we
will assume that user groups of cardinality up to \( \eta \) yield a well-conditioned channel matrix and ZFBF can be used to serve all the users in the same group concurrently.

In a C-RAN, the ZFBF precoding is performed at the BBU pool, where we jointly precode all data streams requested by the users in the same group \( S \) to get a spatial multiplexing gain of order \(|S|\). Before the actual computation for ZFBF precoding happens, the requested data streams for users in \( S \) need to be fetched to the same rack, which may induce a large number of data transfers across racks depending on whether data locality is taken into consideration upon the formation of the user groups or not.

### C. Simple motivating examples

Suppose that the base station has \( \eta = 2 \) antennas that can provide two degrees of freedom for spatial multiplexing, sending two data streams in a single resource block. Suppose that there are \( M = 4 \) single antenna users. Last, suppose that the data requested by user 1 and user 3 are in the first rack and the data requested by user 2 and user 4 are in the second rack, i.e., \( M_1 = \{1,3\} \) and \( M_2 = \{2,4\} \), see Fig. 2.

If we group user 1 and 2 together and user 3 and 4 together for ZFBF precoding, i.e., \( S_1 = \{1,2\} \) and \( S_2 = \{3,4\} \), then we need to first transfer data \( d_2 \) from \( r_2 \) to \( r_1 \) and data \( d_3 \) from \( r_1 \) to \( r_2 \), resulting in a total of two data transfers across racks. This is because for ZFBF precoding, the coded symbols to be transmitted by the BS antenna array are weighted linear combinations of all streams. On the other hand, if we group user 1 and 3 together and user 2 and 4 together, i.e., \( S_1 = \{1,3\} \) and \( S_2 = \{2,4\} \), then obviously there is no need to transfer any data across racks for precoding since, for example, for users 1 and 3 \( d_1 \in r_1 \) and \( d_3 \in r_1 \).

Suppose now that \( d_1 \in r_1, d_3 \in r_1, d_2 \in r_2, \) and \( d_4 \in r_N \) as shown in Fig. 3. Consider the following two user grouping schemes. The first one is to have two groups with \( S_1 = \{1,3\} \) and \( S_2 = \{2,4\} \), in which case two resource blocks and one data transfer (say, moving \( d_4 \) from \( r_N \) to \( r_2 \)) are required. The second one is to have three groups with \( S_1 = \{1,3\}, S_2 = \{2\}, \) and \( S_3 = \{4\} \), in which case three resource blocks are required (the degrees of freedom are not fully utilized) and no data transfer is needed. It is easy to see that to fully utilize the available degrees of freedom provided by the system more data transfers across racks may be required. We shall see in later sections that the number of data transfers across racks in forming user groups can be gradually suppressed by adjusting the weight of the regularization term according to the congestion level in the cloud data center.

### IV. Problem formulation and analysis

We start by deriving expressions for the number of data transfers induced by traditional user grouping schemes (Section IV-A) and the proposed data-locality-aware user grouping scheme (Section IV-B) under the assumption that any user group of size \( \eta \) yields a well-conditioned channel matrix and thus \( \frac{M}{\eta} \) resource blocks suffice to serve all users. Note that the derived analytical expressions can easily quantify the behavior of the system and thus offer intuition. We then present the details of how to jointly use the CSI and data locality information for user grouping (Section IV-C). To design a user grouping algorithm that can adapt to the congestion level in the cloud data center, we also consider the use of more than \( \frac{M}{\eta} \) resource blocks. Thus, we formulate a regularized resource block minimization problem and introduce an efficient approximation algorithm (Section IV-D). Finally, we discuss how to exploit CSI to further increase the spectral efficiency and formulate a regularized spectral efficiency maximization problem (Section IV-E).

#### A. Randomized/CSI-based user grouping

In this subsection, we compute for randomized and CSI-based user grouping the number of induced data transfers, and derive closed-form lower and upper bound expressions under the regime where the number of racks is a lot larger than the degrees of freedom (antennas) of the BS, which is trivially satisfied in practice.

In randomized user grouping, we form the first group by selecting \( \eta \) users uniformly at random from the \( M \) users. Then, we form the second group by selecting \( \eta \) users uniformly at random from the remaining \( M - \eta \) users, and we continue this process until we have \( \frac{M}{\eta} \) groups. In CSI-based user grouping, user groups are formed by lumping together \( \eta \) users with semi-orthogonal channel vectors [15]. Since the performance metric under study is the number of induced data transfers, the analysis is the same for both randomized and CSI-based user grouping schemes.
Under such traditional user grouping schemes, the sets 
$S_i$, $i = 1, \ldots, M$ are formed independently of the sets 
$M_j$, $j = 1, \ldots, N$ since users are grouped independently of 
the location of their requested data. As a result, the requested 
data by the $\eta$ users in a group will be randomly and uniformly 
distributed among the $N$ racks. This is similar to the “balls into 
bins” problem where we throw $\eta$ balls into $N$ bins.

To perform ZFBF precoding over the $\eta$ data streams re-
quested by the users in a group we need to first select a rack 
as the designated rack to perform the precoding. Intuitively, 
we select the rack that stores the largest number of requested 
data as the designated rack (ties are randomly resolved). Then, 
we transfer all remaining requested data (located outside the 
designated rack) to the designated rack for precoding. Thus, 
the number of induced data transfers equals $\eta$ minus the 
number of requested data in the designated rack. For example, 
if $\eta = 6$, rack 1 stores two, rack 2 stores one, and rack 3 stores 
three requested data files, then rack 3 will be selected as the 
designated rack and a total of 3 data transfers will take place 
towards rack 3 (two from rack 1 and one from rack 2). As 
a result, the average number of induced data transfers under 
the randomized/CSI-based user grouping scheme (denoted as 
$D_{\text{rand/CSI}}$) is equal to

$$\mathbb{E} \left[ \eta - \text{the no. of requested data files in the designated rack} \right].$$

Since the number of requested data files in the designated 
rack is the same as the load of the maximum loaded bin of the 
“$\eta$ balls into $N$ bins” problem, we proceed the analysis by 
defining $X_j$ to be the number of requested data files (balls) in 
rack (bin) $j$. By using the concentration results for the balls 
to bins problem [29], when $\frac{N}{\log \log N} \leq \eta \ll N \log N$, we 
have 
$$\Pr \left\{ \max_{j=1, \ldots, N} X_j > \frac{\log N}{\log \frac{N}{\log N}} \left( 1 + \frac{\log \log \frac{N}{\eta}}{\log \frac{N}{\eta}} \right) \right\} = o(1) \text{ and } \Pr \left\{ \max_{j=1, \ldots, N} X_j > \frac{\log N}{\log \frac{N}{\log N}} \right\} = 1 - o(1).$$

As a result, the asymptotic upper and lower bound on the 
average number of induced data transfers are

$$D_{\text{rand/CSI}} \leq \frac{M}{\eta} \left( \eta - k \right) \Pr \left\{ \max_{j=1, \ldots, N} X_j = k \right\},$$

and

$$D_{\text{rand/CSI}} \geq \frac{M}{\eta} \left( \eta - \frac{\log N}{\log \frac{N}{\log N}} \right).$$

We compare both bounds with simulation results in Section VI. 
From Fig. 7, we can see that both bounds are quite tight.

**B. Data-locality-aware user grouping**

Consider a data-locality-aware user grouping scheme whose 
goal is to minimize the number of induced data transfers by 
properly grouping users into $\frac{M}{\eta}$ groups.

To find the user groups that minimize data transfers we 
introduce the following optimization problem. Let $x_{i,j}$ be a 
binary variable, where $x_{i,j} = 1$ if the requested data $d_i$ is 
assigned to rack $j$ for processing, that is, it will be jointly 
precoded with another $\eta - 1$ requested data assigned to rack 
j. Let $c_{i,j}$ be the associated cost of transferring data file $d_i$ 
to rack $j$. Since we aim to count the number of induced data 
transfers across racks, we set $c_{i,j} = 0$ if $d_i \in R_j$ and $c_{i,j} = 1$ 
otherwise. (In Section IV-D, we generalize to non-binary costs 
to take into account different routing paths.) Let $y_j$ be the 
number of resource blocks required to serve all the requested 
data assigned to rack $j$. The data-locality-aware user grouping 
problem can be formulated as follows:

$$\begin{align*}
\text{minimize} & \quad \sum_{i \in M} \sum_{j \in N} c_{i,j} x_{i,j} \\
\text{subject to} & \quad \sum_{i \in M} x_{i,j} = 1, \forall i \in M \\
& \quad \sum_{i \in M} x_{i,j} \leq \eta y_j, \forall j \in N \\
& \quad \sum_{j \in N} y_j = \frac{M}{\eta} \\
& \quad x_{i,j} \in \{0, 1\}, \forall i \in M, j \in N \\
& \quad y_j \in \{0, 1, 2, \ldots \}, \forall j \in N.
\end{align*}$$

The objective function above is equal to the total number of 
induced data transfers across racks. The first constraint ensures 
that the requested data $d_i$ has to be assigned to some rack. 
The second constraint ensures that we allocate enough resource 
blocks to rack $j$ to serve all the requested data files assigned 
to it. Recall that the number of requested data files that can 
be served in one resource block by using ZFBF precoding is 
less than or equal to the number of BS antennas $\eta$. The third 
constraint ensures that the total number of resource blocks that 
we use is equal to $\frac{M}{\eta}$ like in the randomized/CSI-based case.

**A polynomial time algorithm for Problem (5):** The above 
problem can be solved in polynomial time as follows. Recall 
that $M_j$ denotes the set of users whose requested data are 
in rack $j$. First, for each rack $j = 1, \ldots, N$ with $|M_j| \geq \eta$, 
we group batches of $\eta$ users in $M_j$ until $|M_j| \mod \eta$ users 
are left ungrouped. Note that in this step we form a total of 
$\sum_{j=1}^{N} \left\lfloor \frac{|M_j|}{\eta} \right\rfloor$ groups, inducing no data transfers across racks. 
The number of ungrouped requested data files in rack $j$ at
the end of this step is $|M_j| \mod \eta$. Second, we sort the set of numbers $\{m_j \mod \eta, j = 1, \ldots, N\}$ in descending order (note that $0 \leq m_j \leq \eta - 1$). We denote the sorted numbers as $m_1(1) \geq m_2(1) \geq \cdots \geq m_N(1)$. Third, to ensure that we jointly pre-code and serve $\eta$ requested data files at each resource block, we find the number $t$ such that

$$t = \sum_{j=1}^{N} (\eta - m_j)$$

and transfer the requested data files stored in the racks corresponding to the set $\{m_{j+1}, \ldots, m_N\}$ to the racks corresponding to the set $\{m_1, \ldots, m_\eta\}$, as shown in Fig. 4.\footnote{Note that the complexity of the above user grouping algorithm is $2M + N \log N$ since we perform two linear passes in the first and third steps, and we sort $N$ numbers in the second step.} Intuitively, the equation above finds the right way to partition racks such that the minimum number of data files are transferred from racks with a few ungrouped requests into racks with a few “empty” slots to reach $\eta$ ungrouped requests and then group them together. As a result, the average number of induced data transfers under the data-locality-aware user grouping scheme is equal to $D_{\text{locality}} = E\left[\sum_{j=1}^{N} m_{j}\right]$. For $M \gg N\eta$, that is, when the number of resource blocks required to serve one request of each user is a lot larger than the number of racks, we derive an approximation formula for $D_{\text{locality}}$. When $M \gg N\eta$, $\eta > 1$, the number of requested data files left ungrouped in rack $j$, i.e., $m_j \mod \eta = \eta$, will be approximately uniformly distributed in $\{0, 1, 2, \ldots, \eta - 1\}$. As a result, we have on average $\frac{\eta}{N}$ racks left with $\eta$ requested data files ungrouped, for $i = 0, 1, \ldots, \eta - 1$. Under this deterministic approximation, we may transfer the only ungrouped data file of $\frac{\eta}{N}$ racks to the $\frac{\eta}{N}$ racks which have $\eta - 1$ ungrouped data files, and so on and so forth till we form groups each of size $\eta$ as usual. Specifically, when $\eta$ is odd, the total number of data file transfers equals

$$D_{\text{locality}} \approx \frac{N}{\eta} \left[1 + 2 + \cdots + \left(\frac{\eta - 1}{2}\right)\right] = \frac{N(\eta + 1)(\eta - 1)}{8\eta},$$

and, when $\eta$ is even it equals

$$D_{\text{locality}} \approx \frac{N}{\eta} \left[1 + 2 + \cdots + \left(\frac{\eta}{2} - 1\right) + \frac{\eta}{2}\right] = \frac{N\eta}{8},$$

where the term $\frac{\eta}{2}$ comes from the fact that we have $\frac{N}{\eta}$ racks with $\frac{\eta}{2}$ ungrouped data files, and we transfer from half of these racks these ungrouped data files to the other half. Note that these asymptotic expressions offer intuition, for example, they imply that under the data-locality-aware scheme the number of transfers does not depend on the number of users/requests, whereas under the CSI and randomized schemes it grows linearly with the number of users/requests, see Eq. (3) and (4).

The above results can be obtained formally using order statistics [30] when $m_j$ is uniformly distributed in $[0, \eta - 1]$ and $N \to \infty$. Specifically, it can be shown that when $\eta$ is odd, we have $D_{\text{locality}} = E\left[\sum_{j=1}^{N} m_{j}\right] \to \frac{N^2(\eta + 1)(\eta - 1)}{8\eta}$, and, when $\eta$ is even, we have $D_{\text{locality}} = E\left[\sum_{j=1}^{N} m_{j}\right] \to \frac{N^2\eta}{8}$.

where $t = \frac{N(\eta - 1)}{2\eta}$ since $t\eta = E\left[\sum_{j=1}^{N} m_{j}\right] = E\left[\sum_{j=1}^{N} m_{j}\right] = N\frac{\eta - 1}{2}$. We compare the above approximations with simulation results in Section VI, see Fig. 7.

C. Joint CSI- and data-locality-aware user grouping

The proposed data-locality-aware user grouping scheme in the previous subsection can be generalized to a joint CSI- and data-locality-aware user grouping scheme that also takes CSI into account when forming user groups. Let us consider again the problem of minimizing the number of induced data transfers by properly grouping $M$ users into $M/\eta$ groups (Problem (5)) and its corresponding polynomial time algorithm. We are going to modify the polynomial time algorithm such that not only the optimal in Problem (5) is reached (i.e. we have the same minimum number of induced data transfers) but also all user groups of cardinality $\eta$ yield a well-conditioned channel matrix.

The polynomial time algorithm consists of three steps. We modify the first and last step as follows: Recall that in the first step of the polynomial time algorithm, we fix a rack $j$ and group batches of $\eta$ users in $M_j$ until $|M_j| \mod \eta$ users are left ungrouped. Unlike before where users are selected at random, here we select users based on CSI. Note that it is well known that CSI-based user grouping is NP-hard, and no tight approximation algorithms exist. Thus, we resort to a greedy approach which forms what is commonly referred to as semi-orthogonal user groups, based on prior work, see, for example, [15]–[17]. To form the first group (of size $\eta$), we begin with choosing an arbitrary user $i_1^1 \in M_j$ and insert it into the first group. Then, we choose another user $i_2^1$ whose channel vector is the most orthogonal to that of user $i_1^1$ and insert $i_2^1$ into the first group. After that, we choose another user $i_3^1$ whose channel vector is the most orthogonal to the signal space spanned by the channel vectors of users $i_1^1$ and $i_2^1$, and so on and so forth until we have $\eta$ users in the first group, namely, $\{i_1^1, \ldots, i_\eta^1\}$. We repeat the above procedure to form the second group considering the candidate user set $M_{j1} \backslash \{i_1^1, \ldots, i_\eta^1\}$. We continue this process until we form the $\left\lceil \frac{|M|}{\eta} \right\rceil$-th user group consisting of users $\left\{i_1^1, \ldots, i_\eta^1 M_{j1} \backslash \{i_\eta^1\}\right\}$.

We modify the last (third) step of the polynomial time algorithm as follows. When we transfer requested data from the racks in the tail portion of Fig. 4 to the racks in the head portion, we again use CSI and the above greedy algorithm repeatedly to decide the formation of batches of $\eta$ users. Specifically, let us fix the rack with the most remaining ungrouped users $m_{(1)}$. We select a total number of $\eta - m_{(1)}$ users one by one (from the set of ungrouped users whose requested data files are stored in the racks corresponding to the set $\{m_{(1)}, \ldots, m_N\}$) by using the aforementioned orthogonality principle to form a group of size $\eta$. Then, we fix the rack with the second most remaining ungrouped users $m_{(2)}$, and so on and so forth. We continue this procedure till we fix the rack with $m_{(N)}$ ungrouped users and group them together with the last remaining $\eta - m_{(N)}$ users, forming the last group.

Last, we characterize the complexity of the above user grouping process in terms of the number of comparisons (inner
achieve this goal by relaxing the constraint of using exactly \( \eta \) user groups. We can see that to form a group of \( \eta \) users we need at most \( \eta^2(M - \eta) \) such comparisons. We have here a total of \( \frac{M}{\eta} \) user groups, so the total number of comparisons is upper bounded by \( M\eta(M - \eta) \).

D. Regularized resource block minimization

In this subsection, we aim to design a user grouping algorithm that can adapt to the congestion level in the cloud data center. The idea is to suppress data transfers across racks in forming user groups when congestion occurs. We achieve this goal by relaxing the constraint of using exactly \( \frac{M}{\eta} \) resource blocks to serve \( \frac{M}{\eta} \) user groups of \( \eta \) users each (i.e., allowing forming a larger number of smaller user groups) and by introducing a regularization term.

Specifically, we propose a regularized resource block minimization problem in which we minimize a weighted sum of the number of resource blocks needed to serve all user requests \( \sum_{j \in N} y_j \) and the cost associated with the data transfers across racks \( \sum_{i \in M} \sum_{j \in N} c_{i,j} x_{i,j} \) that acts as a regularization term with weight \( \rho \). The regularized resource block minimization problem can be written as:

\[
\begin{align*}
\text{minimize} & \quad \sum_{j \in N} y_j + \rho \sum_{i \in M} \sum_{j \in N} c_{i,j} x_{i,j} \\
\text{subject to} & \quad \sum_{i \in M} x_{i,j} = 1, \ \forall i \in M \\
& \quad \sum_{i \in M} x_{i,j} \leq \eta y_j, \ \forall j \in N \\
& \quad x_{i,j} \in \{0, 1\}, \ \forall i \in M, j \in N \\
& \quad y_j \in \{0, 1, 2, \cdots\}, \ \forall j \in N.
\end{align*}
\]

Note that here we allow a general cost \( c_{i,j} \) for transferring data file \( d_i \) to rack \( j \) (instead of restricting to the binary cost as in Section IV-B) to account for more complicated data center topologies [31], [32], as shown in Fig. 5. For example, \( c_{i,j} \) may be the cost of the shortest path to transfer data file \( d_i \) to rack \( j \). Note that \( c_{i,j} = 0 \) if \( d_i \not\in r_j \).

To solve the above problem, we first map it to the well-known metric soft-capacitated facility location problem (SCFLP). Then, we follow the approach presented in [20] to solve it with a 2-approximation algorithm.

Mapping Problem (8) to a metric SCFLP: Under a general cost \( c_{i,j} \), Problem (8) is NP-hard since it can be mapped to the SCFLP [20]. In the SCFLP, soft capacity means that we can open multiple copies of the same facility at the same location and each copy has a hard capacity for servicing customers. In the SCFLP, we minimize the sum of the costs for opening facilities and the costs for servicing customers by deciding which facilities to open, how many copies of each facility to open, and which customers are served by which facilities.

The mapping between Problem (8) and the SCFLP is established as follows. Performing ZFBF precoding at rack \( j \) for \( y_j \) resource blocks is equivalent to opening facility \( j \) with \( y_j \) copies. The degrees of freedom \( \eta \) (the maximum number of requested data files that can be jointly precoded to be transmitted in one resource block) is equivalent to the hard capacity of a copy of a facility (one resource block can only serve up to \( \eta \) requested data files). The user data requests are equivalent to customers. Processing/precoding data \( d_i \) at rack \( j \) with the associated data transfer cost \( c_{i,j} \) is equivalent to servicing customer \( i \) by facility \( j \) with the associated servicing cost \( c_{i,j} \).

In the SCFLP, when the costs live in a metric space and thus satisfy the triangular inequality, we call it a metric SCFLP. Our problem is a metric SCFLP since our costs satisfy the triangular inequality. Indeed, let \( r(i) \) denote the rack that stores data \( d_i \) and denote by \( \hat{c}_{i,j} \) the cost for transferring data \( d_i \) via the shortest path between rack \( r(i) \) and rack \( j \). Then, denote \( \hat{c}_{i,j} \) the cost for transferring a data file from rack \( k \) to rack \( j \). Then, we have \( c_{i,j} = \hat{c}_{i,j} \) for any other rack \( k \) and, more general, \( \hat{c}_{k,j} \), the cost for transferring a data file from rack \( k \) to rack \( j \). Then, we have \( \hat{c}_{i,j} \).

A 2-approximation algorithm for Problem (8): The metric SCFLP is NP-hard. The authors in [20] map the metric SCFLP to a linear facility location problem (LFLP) and then to an uncapacitated facility location problem (UFLP) which they solve using the Jain-Mahdian-Saberi (JMS) algorithm [33], yielding a 2-approximation solution to the original metric SCFLP problem. We follow the same procedure to solve Problem (8), which yields a 2-approximation solution [20].

### Algorithm 1 The 2-approximation algorithm for Problem (8)

1. Let \( \tilde{c}_{i,j} = \frac{1}{\eta} + \rho \hat{c}_{i,j} \).
2. At the beginning, all requests are unassigned, all racks are open, and the budget of every request \( i \), denoted by \( B_i \), is initialized to 0. At every moment, each request \( i \) offers some money from its budget to each unopened rack \( j \). The amount of this offer is equal to \( \max(B_i - \tilde{c}_{i,j}, 0) \) if \( i \) is unassigned, and \( \max(\tilde{c}_{i,j} - \tilde{c}_{i,j}, 0) \) if it is assigned to some other rack \( j' \).
3. While there is an unassigned request, increase the budget of each unassigned request at the same rate, until one of the following events occurs:
   - For some unopened rack \( j \), the total offer that it receives from requests is equal to \( 1 - \frac{1}{\eta} \). In this case, we open rack \( j \), and for every request \( i \) (assigned or unassigned) which has a non-zero offer to rack \( j \), we assign request \( i \) to rack \( j \).
   - For some unassigned request \( i \), and some rack \( j \) that is already open, the budget of request \( i \) is equal to \( \tilde{c}_{i,j} \). In this case, we assign request \( i \) to rack \( j \).

To evaluate the performance of the 2-approximation algorithm, we use CVX with Gurobi to solve Problem (8), which is
a mixed integer linear programming problem. Gurobi reports the distance of its output to the optimal and often this distance is zero. Thus, we are able to compare the 2-approximation algorithm against the optimal for small-scale scenarios, see Section VI for the results.

Last, we can employ a bisection approach to search for the weight $\rho^*$ that results in a desired amount of data transfers across racks (say, $\tau$), i.e., $\sum_{i \in M} \sum_{j \in N} c_{i,j} x_{i,j}^*(\rho^*) \approx \tau$.

Algorithm 2 The bisection approach for finding $\rho^*$

1: Initialize $\rho_U = \rho_{max}$, $\rho_L = 0$, $\epsilon = \epsilon_0$, $\rho = 0$
2: while $|\sum_{i \in M} \sum_{j \in N} c_{i,j} x_{i,j}^*(\rho) - \tau| > \epsilon$ do
3: \hspace{1em} if $\sum_{i \in M} \sum_{j \in N} c_{i,j} x_{i,j}^*(\rho) > \tau + \epsilon$ then
4: \hspace{2em} $\rho_L \leftarrow \rho$
5: \hspace{1em} else
6: \hspace{2em} $\rho_U \leftarrow \rho$
7: \hspace{1em} end if
8: \hspace{1em} $\rho \leftarrow \frac{\rho_U + \rho_L}{2}$
9: end while
10: return $\rho^* \leftarrow \rho$

Note that $\epsilon_0 > 0$ is a small positive number (the tolerable deviation from the desired amount of data transfers across racks $\tau$), $\rho_{max}$ is chosen to be large enough such that there is no data transfers across racks, and $x_{i,j}^*(\rho)$ is the optimal solution of the regularized resource block minimization problem (8) under a given weight $\rho$.

E. Regularized spectral efficiency maximization

Similar to what we did in Section IV-C, for the regularized resource block minimization problem, we can further increase the spectral efficiency by exploiting CSI and the aforementioned greedy algorithm to form semi-orthogonal user groups whenever there are more than $\eta$ requests assigned to the same rack at the end of the 2-approximation algorithm. Specifically, we greedily add users to user groups such that the channel vector of each added user is the most orthogonal among the ungrouped users to the signal space spanned by the channel vectors of the users already in the group, see Section IV-C.

The procedure described above uses CSI to form better groups whenever there are more than $\eta$ requests assigned to the same rack at the end of the 2-approximation algorithm. However, it might be the case that upon transferring a request from a rack with $\eta$ or less requests to another rack, the corresponding channel matrices might be such that the spectral efficiency of the resulting user groups is improved. And, depending on the weight $\rho$, the marginal gain from such a move might be larger than the marginal cost of the additional data transfer. Motivated by this, in the following we introduce a general formulation referred to as the regularized spectral efficiency maximization problem, by using as a metric for the efficiency of the wireless channel not the number of resource blocks required, but the actual data rates achieved on these resource blocks.

We optimize over the set of all possible partitions of the user set $M$, i.e., we optimize over $S_1, S_2, \ldots, S_T$, where $S_k \cap S_l = \emptyset, k \neq l$, $\bigcup_{k=1}^T S_k = M$, and the cardinality of each group is less than or equal to the degrees of freedom $|S_k| \leq \eta$. Let $R_{ZFBF}(S_k)$ denote the sum rate of the users in group $S_k$ under ZFBF precoding, where $R_{ZFBF}(S_k) = \sum_{i \in S_k} \log(1 + SNR)$ and $\gamma_i = 1/\left[ \left( H(S_k)^H H(S_k) \right)^{-1} \right]_{i,i}$ [15]. The regularized spectral efficiency maximization problem can be written as

$$\begin{align*}
\text{maximize} & \quad \frac{\sum_{k=1}^T R_{ZFBF}(S_k)}{T} - \rho \sum_{k=1}^T \min_{i \in S_k} \sum_{j \in N} c_{i,j} \\
\text{subject to} & \quad T \in \{1, 2, \ldots, M\} \\
& \quad \bigcup_{k=1}^T S_k = M \\
& \quad |S_k| \leq \eta, k = 1, \ldots, T. 
\end{align*}$$

(9)

Note that all the data files requested by the users in group $S_k$ will be transferred to a rack $j_k$ for ZFBF precoding. It is easy to see that $j_k = \arg \min_{i \in N} (\sum_{i \in S_k} c_{i,j})$ since this rack designation results in the minimum data transfer cost for users in group $S_k$. Like before, $\rho$ is the weight for the regularization term.

As already mentioned, the pure spectral efficiency maximization problem ($\rho = 0$ in Problem (9)) is well known to be NP-hard and no tight approximation algorithms exist. Thus, we devise a greedy algorithm to solve Problem (9).

Let us define the utility of a user group $S_k$ as $U(S_k) = R_{ZFBF}(S_k) - \rho \min_{i \in N} (\sum_{i \in S_k} c_{i,j})$, which is the sum rate minus the induced data transfer cost. In addition, we define the marginal utility of including a new user $i$ into a group $S_k$ as $U(i|S_k) \pm U(S_k \cup \{i\}) - U(S_k)$.

Algorithm 3 Greedy algorithm for Problem (9)

1: Initialize $M \leftarrow \{1, \ldots, M\}, k \leftarrow 0$
2: while $M \neq \emptyset$ do
3: \hspace{1em} $k \leftarrow k + 1$
4: \hspace{1em} $S_k \leftarrow \emptyset$
5: \hspace{1em} while ($|S_k| < \eta$ and $\max_{i \in M} U(i|S_k) > \alpha$) or $S_k = \emptyset$ do
6: \hspace{2em} $S_k \leftarrow S_k \cup \{i^*\}$, where $i^* = \arg \max_{i \in M} U(i|S_k)$
7: \hspace{1em} $M \leftarrow M \setminus \{i^*\}$
8: end while
9: end while

In the greedy algorithm, when forming the $k$-th user group $S_k$, we include a new user $i^*$ that has the highest marginal utility into $S_k$ if the number of users in $S_k$ is less than the degrees of freedom $\eta$ and that highest marginal utility is larger than some threshold $\alpha$. Otherwise, we proceed to form the next group $S_{k+1}$. At the end of the algorithm, the sets $S_1, S_2, \cdots$ form a partition of the user set $M$.

Note that larger groups will allow for more concurrent transmissions and thus better spectral efficiency, which Problem (9) takes into account by dividing the sum rate by $T$, the total number of groups, and optimizing the average group rate rather than the total sum rate. Since the greedy algorithm does not know a priori the optimal number of user groups, the user
group utility and marginal utility defined above work with the group rate, and, the parameter $\alpha$ is used to parameterize over different group sizes (the smaller the $\alpha$ the larger the groups tend to be and thus the less their number and vice versa).

Last, to achieve a desired amount of data transfers across racks, the optimal weight $\rho^*$ can be obtained by using a bisection approach similar to Algorithm 2.

V. EXTENSIONS

A. The case with replicas

In data centers it is common to have multiple replicas of the same data file in different racks to reduce the read/write latency and increase system reliability [34]. Let us define the set of racks that store a requested data file $d_i$ by user $i$ as $R_i$. Recall that $\hat{c}_{k,j}$ denotes the cost of transferring a data file from rack $k$ to rack $j$. Since we have $|R_i|$ replicas of data file $d_i$, the associated cost of transferring data file $d_i$ to rack $j$ would be the minimum cost between transferring the data file from any rack $k \in R_i$ to rack $j$, that is, $c_{i,j} = \min_{k \in R_i} \hat{c}_{k,j}$. Note that $c_{i,j} = 0$ if one of the replicas of data file $d_i$ is stored in rack $j$ already.

From the discussion above it is easy to see that the presence of replicas yields the same formulation as before.

B. The case with multiple BSs

We consider the scenario as shown in Fig. 1a, where there are multiple BSs/cells (say, $G$ cells), each associated with a number of users. For independent cells (no inter-cell interference and no coordination among cells like in CoMP [3]) it is easy to see that the global problem can be decomposed into $G$ independent instantiations leading to the same formulation as before.

When cells are dependent, the problem is coupled. In the case of Problem (8), this is so because two users belonging to two different cells may or may not use the same resource block depending on inter-cell interference, and the problem cannot be mapped to the SCFLP. In the case of Problem (9), this is so because the rate of users depends on both inter-cell interference and on whether CoMP is used or not. That said, the greedy algorithm can still be used.

C. The case with multi-antenna users

When a user has more than one antennas, it can receive more than one data streams at the same resource block. Let $n_i$, $i \in M$ be the number of antennas of user $i$. By replacing user $i$ with $n_i$ users of one antenna each, and serving different requests of the original user $i$ with each new user, we can map the multi-antenna problem to the same formulation as before.

D. The case with different requested data sizes

We consider the case where the requested data of users can have different sizes, consuming different numbers of resource blocks (note that we assume that the unit size of a user’s requested data is one resource block). Specifically, let $a_i$ be the size of the requested data of user $i$ (which is equal to the number of resource blocks to be consumed by user $i$) and $\Omega_i = \{0, 1, 2, \ldots, a_i\}$ be a binary variable, where $\Omega_i = 1$ if the $k$-th portion of the requested data $d_i$ is assigned to rack $j$ for processing and $\Omega_i = 0$ otherwise. The regularized resource block minimization problem (8) can be generalized as:

$$\text{minimize } \sum_{j \in N} y_j + \rho \sum_{i \in M} \sum_{k \in R_i} \sum_{j \in N} c_{i,j} \Omega_{i,j,k}$$

subject to

$$\sum_{j \in N} \Omega_{i,j,k} = 1, \quad \forall i \in M, k \in A_i$$

$$\sum_{i \in M} \sum_{k \in A_i} \sum_{j \in N} \Omega_{i,j,k} \leq \eta y_j, \quad \forall j \in N$$

$$\sum_{k \in A_i} \Omega_{i,j,k} \leq y_j, \quad \forall i \in M, j \in N$$

$$\Omega_{i,j,k} \in \{0, 1\}, \quad \forall i \in M, k \in A_i, j \in N, \quad y_j \in \{0, 1, 2, \ldots, \}.$$  

where the third constraint guarantees that any two portions of the requested data $d_i$ are not multiplexed in the same resource block (since each user has only a single antenna and can only receive one stream at each resource block). Problem (10) is a mixed integer linear program which can be solved by using Gurobi.

VI. SIMULATION AND NUMERICAL RESULTS

A. Randomized/CSI-based vs data-locality-aware user grouping

Fig. 6 compares the data-locality-aware user grouping scheme with the randomized/CSI-based user grouping scheme in terms of the total number of induced data transfers. We assume that the number of racks $N$ equals 10, the number of antennas in a BS $\eta$ equals 4, and the number of user data file requests $M$ varies from 1 to 50. The data files requested by the $M$ users are distributed among the $N$ racks independently and uniformly at random.

Under all schemes, we plot the number of induced data transfers needed to fully exploit the spatial multiplexing gain, i.e., we group the $M$ users into $\lceil \frac{M}{\eta} \rceil$ groups of size $\eta = 4$. Fig. 6
In Fig. 7, under the randomized/CSI-based user grouping scheme, we compare the asymptotic upper bound (Eq. (3)) and lower bound (Eq. (4)) for the number of induced data transfers with simulation results. These bounds hold for \( N \gg \eta \) and we vary \( N \) from 10 till 50 in Fig. 7a-c. We can see that both the upper and lower bounds give a good approximation.

In addition, under the data-locality-aware user grouping scheme, we compare the approximation formula (Eq. (7)) for the number of induced data transfers with simulation results. The approximation holds for \( M \gg N\eta \), and, indeed, it is accurate in this regime. Last, Fig. 7 illustrates that the number of induced data transfers for the data-locality-aware user grouping scheme does not increase with the number of user data file requests \( M \), while that for the randomized/CSI-based user grouping scheme increases linearly with \( M \).

Last, in Fig. 8, we consider the case that the data requests of a user arrive according to a discrete-time arrival process where the inter-arrival time follows a geometric distribution with parameter \( p \) (traffic intensity). We assume that \( M = 50, N = 10, \eta = 4 \), and the user grouping decisions are made per time slot. We plot the average number of data transfers as we vary the traffic intensity from \( p = 0.1 \) (the average inter-arrival time of requests of a user is 10 slots) to \( p = 1 \) (each time slot there is a request from a user). We can see that the performance gain increases as the traffic intensity increases.

### B. Joint CSI- and data-locality-aware user grouping

We compare the performance of the data-locality-aware user grouping scheme with that of the joint CSI- and data-locality-aware user grouping scheme in terms of the average spectral efficiency measured in bits/s/Hz. Note that as discussed in Section IV-C, both schemes induce the same number of data transfers. We assume that the number of antennas in a BS is \( \eta = 4 \) and assume i.i.d. Rayleigh fading channel coefficients with unit power. The spectral efficiency is calculated based on ZFBF precoding at an SNR value of 3dB [15]. For the joint CSI- and data-locality-aware user grouping scheme, we use the greedy algorithm introduced in Section IV-C to form semi-orthogonal user groups.

3For the randomized/CSI-based user grouping scheme, since the number of data transfers to form a group of \( \eta \) users is a fixed constant \( \frac{\log N}{\eta} \) (independent of \( M \)) and we have a total of \( \frac{M}{\eta} \) groups, the total number of data transfers increases with the number of users \( M \).
Fig. 9 shows that, as expected, the joint CSI- and data-locality-aware user grouping scheme achieves a higher spectral efficiency due to the better-conditioned channel matrix formed by the users in the same group. Note that the rate gains can be significant. For example, when the average number of requests per rack is 50, there is a 1.4x improvement in data rate over the data-locality-aware scheme. In addition, we can see that the gap between the two curves gradually increases as the number of requests per rack increases. The reason is that we have progressively better-conditioned channel matrix due to increasing multiuser diversity.

C. Regularized resource block minimization and regularized spectral efficiency maximization

We study the Pareto optimal curves for the regularized resource block minimization problem (8) in Figs. 10–13. We vary the number of user data file requests $M$ from 50 to 150, the number of racks $N$ from 5 to 20, and the number of antennas in a BS $\eta$ from 2 to 8. The cost $\hat{c}_{k,j}$ for transferring data from rack $k$ to rack $j$ is assumed to be distributed in the range $[0, 50]$. We plot the number of resource blocks needed to serve all requests against the cost associated with the induced data transfers as the weight $\rho$ varies between 0
The achieved cost for the induced data transfers across racks and preferentially group together users whose requested data are located under the same rack. Operating the system at the leftmost point in Fig. 14 corresponds to the case of using a very large weight for regularization (the data center network is heavily congested), resulting in no data transfers across racks in forming user groups.

Last, Fig. 15 shows the increase in the spectral efficiency by exploiting CSI. The blue curve corresponds to Problem (8), the regularized resource block minimization problem. We refer to this as the CSI-unaware scheme. The red curve corresponds to Problem (8) with the addition of using CSI to form semi-orthogonal groups within a rack with more than \( \eta \) requests assigned to it, see the first paragraph of Section IV-E. We refer to this as the CSI-partially-aware scheme. Last, the green curve corresponds to Problem (9), the regularized spectral efficiency maximization problem, where CSI is fully exploited. We refer to this as the CSI-fully-aware scheme. To create the blue and red curves we use the Shannon rate formula with ZFBF precoding to obtain the corresponding rates for different number of resource blocks used. Since the CSI-partially-aware scheme is using CSI for some group formations, it has better channel gains, yielding about 1.2x improvement. To create the green curve and compare it with the other two, we use Fig. 13 to map resource blocks to the cost of data transfers for the first two schemes, and then use Fig. 14 to directly get the corresponding rates for the CSI-fully-aware scheme for the aforementioned cost of data transfers. Since this scheme uses CSI for all group formations it yields a higher gain, about 1.3x. This shows how the increasing use of CSI can lead to increasing spectral efficiency without causing additional data transfers at the cloud data center.

D. Accuracy of the 2-approximation algorithm

In Fig. 16, we compare the achieved objective value of the 2-approximation algorithm for the regularized resource block minimization problem with that we get by using CVX with Gurobi, as we vary the weight \( \rho \) from 0 to 1. We can see that the 2-approximation algorithm behaves as expected, i.e., the achieved objective value lies within two times of the optimal.

VII. Conclusion

In this paper, we introduced a novel data-locality-aware user grouping scheme for multi-user MIMO beamforming under...
the cloud radio access network architecture. By grouping users based on both the CSI and the locality of their requested data, we significantly reduced the number of induced data transfers across racks in the cloud data center, while providing the same level of spatial multiplexing gain. To design a user grouping algorithm that can adapt to the congestion level in the cloud data center, we proposed a regularized spectral efficiency maximization problem where the number of data transfers across racks serves as a regularization term whose weight can be controlled by a system operator. When there is no congestion, a small weight can be used, resulting in the traditional CSI-based user grouping scheme. When congestion occurs, a large weight can be used, suppressing data transfers across racks and preferentially grouping together users whose requested data are located in the same rack. Last, under some simplifications, we casted the above problem as a soft-capacitated facility location problem, developed a 2-approximation user grouping algorithm to solve it, and studied via simulations and numerical analysis the tradeoff between spectral efficiency and data transfer cost.

REFERENCES


Konstantinos Psounis (S’97–M’02–SM’08–F’17) received the bachelor’s degree from the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1999 and 2002, respectively. He is currently a Professor of electrical engineering and computer science with the University of Southern California. He models and analyzes the performance of a variety of wired and wireless networks. He also designs methods, algorithms, and protocols to solve problems related to such systems. He is a senior member of ACM.