Hybrid Network- and User-Centric Scalable Cell-Free Massive MIMO for Fronthaul Signaling Minimization

Phu Lai, Wei Xiang, *Senior Member, IEEE*,
William Damario Lukito, Khoa Tran Phan, Peng Cheng, Chang Liu,
Guoqiang Mao, *Fellow, IEEE*

***Abstract*—Cell-free massive multiple-input multiple-output (CFmMIMO) coordinates a great number of distributed access points (APs) with central processing units (CPUs), effectively reducing interference and ensuring uniform service quality for user equipment (UEs). However, its cooperative nature can result in intense fronthaul signaling between CPUs in large-scale networks. To reduce the inter-CPU fronthaul signaling for systems with limited fronthaul capacity, we propose a low-complexity online UE-AP association approach for scalable CFmMIMO that combines network- and user-centric clustering methodologies, relies on local channel information only, and can handle dynamic UE arrivals. Numerical results demonstrate that compared to the state-of-the-art method on fronthaul signaling minimization, our approach can save up to 94% of the fronthaul signaling load and 83% of the CPU processing power at the cost of only up to 8.6% spectral efficiency loss, or no loss in some cases.**

***Index Terms*—Cell-free massive MIMO, user association, access point selection, fronthaul, online algorithm, scalability.**

I. Introduction

Cell-free massive multiple-input multiple-output (CFmMIMO) is a promising architecture for 6G and beyond. The main novelty of CFmMIMO includes two aspects: the spectral efficiency (SE) analysis with imperfect channel state information [1], and the integration of the physical layer of mMIMO, ultra-dense network deployment, and coordinated multipoint (CoMP) methods in a scalable manner [2]. In CFmMIMO, user equipment (UE) are served by a much larger number of geographically distributed access points (APs), which are coordinated by central processing units (CPUs) through wired or wireless fronthaul (Fig. 1). These APs cooperate phase-coherently to serve users in the same time-frequency resource through time-division duplex (TDD), thereby eliminating the concept of cell boundaries [3]. Most CFmMIMO research focuses on *user-centric* approaches [4], in which each UE is served by its selected APs (based on some criteria), rather than *network-centric* approaches, which divide APs into separate clusters serving distinct sets of UEs [5]. Those user-centric approaches that do not take into account inter-CPU coordination often incur very intense inter-CPU signaling, which is not ideal for systems with limited fronthaul capacity. In this correspondence, we propose a hybrid method that combines network- and user-centric clustering and show that its performance is on par with user-centric clustering in terms of SE and fairness (measured by the SE/service uniformity among all UEs) while being superior in terms of scalability with significantly reduced inter-CPU fronthaul signaling and CPU processing power.

Many studies on CFmMIMO do not take system scalability into account and often assume that the fronthaul capacity is unlimited or all APs are connected to a single CPU [3], [6], [7], [8], [9], which is clearly unscalable. As such, researchers have started investigating the scalability aspects of CFmMIMO [1], [10], which involves multiple CPUs. However, the fronthaul signaling between CPUs (or network topology and the links between APs and CPUs) is overlooked in those studies. In a practical CFmMIMO system, a UE may be served by APs associated with multiple CPUs, which exchange signals and data to decode the uplink (UL) data and precode the downlink (DL) data. In a large network, this exchange could be very intense and may overload the fronthaul [2], [11]. Realizing the limited fronthaul constraint, several studies [11], [12], [13] have taken this constraint into consideration. In [13], the UE-AP association is predetermined and the fronthaul is assumed to be flexible, allowing dynamic routing of fronthaul traffic across APs and CPUs through intermediate routing-capable nodes called routers. The authors also consider the finite computing capacity of CPUs. In [11], the authors categorize UEs into local UEs (served by a single CPU, which incurs no inter-CPU signaling) and edge UEs (served by several CPUs, thus inter-CPU signaling is required) based on their geographical distance to the borders of network-centric clusters, which is not robust and ineffective in many scenarios as will be demonstrated in our numerical results. Later on, a cluster refinement method is introduced in [12] which limits inter-CPU coordination by fine-tuning any arbitrary UE-AP association solutions. We aim to design a new UE-AP association scheme (a.k.a AP selection or UE/AP clustering in the literature) that is more scalable and requires much lower amount of inter-CPU fronthaul signaling and CPU computing power with little to no loss in SE and fairness.

Many existing user association methods are *offline*, or based on network-wide optimization and thus unscalable as the number of UEs grows very large [14], [15], [16]. They (implicitly) require a large amount of fronthaul signaling and a central coordinator, which is aware of all UEs' information in the network, to make a global decision; and their computational complexity grows exponentially with the total number of UEs. It is often unclear how such methods would handle the dynamic situations where there are new UE arrivals. They would either have to re-associate existing UEs or wait for a period of time to associate a new batch of UEs. In contrast, we aim to design a scalable *online* mechanism that associates UEs to APs at the time of their arrivals. This paper focuses on the DL transmission. Our key contributions in this correspondence are:

- A hybrid network- and user-centric clustering scheme for user association that is highly scalable and can handle dynamic UE arrivals. The computational complexity of finding an association decision for a UE is independent of the total number of UEs as the scheme only relies on local channel information.
- Numerical results showing that the proposed scheme can save up to 94% of fronthaul signaling and 83% of CPU compute

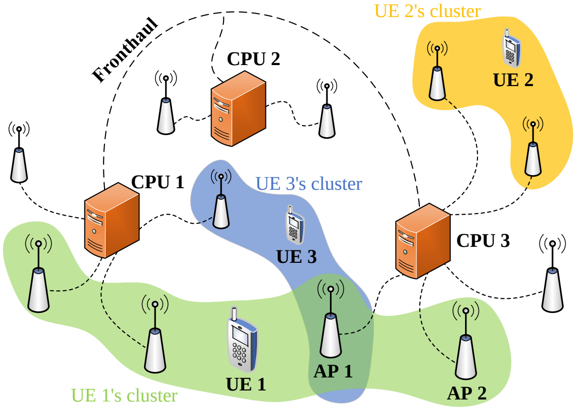

Fig. 1: A cell-free massive MIMO system.

Manuscript received Dec 23, 2021; revised 2022; accepted 2022.
Date of publication 0 . 0000; date of current version 0 . 0000.
(*Corresponding authors: Wei Xiang, Phu Lai*)

P. Lai, W. Xiang, W. D. Lukito, K. T. Phan, P. Cheng, and C. Liu are with Cisco-La Trobe Centre for AI and IoT, La Trobe University, Australia. E-mail: {p.lai, w.xiang, w.lukito, k.phan, p.cheng, c.liu6}@latrobe.edu.au.

G. Mao is with Xidian Research Institute of Smart Transportation, Xidian University, China. E-mail: g.mao@ieee.org.

Digital Object Identifier no.

power with at most 8.6% SE loss (or no loss in some cases) compared to the state-of-the-art (SOTA) on fronthaul signaling minimization.

Our source code (in Python) is accessible at https://github.com/phulai/hybrid-cell-free for reproducibility.

II. System Model

A. Network topology

Access points (APs) and central processing units (CPUs): The system consists of a geographically distributed set $\mathcal{U}$ of U CPUs and set $\mathcal{L}$ of L N-antenna APs. Each CPU is connected to a set of APs via ultra-reliable and error-free fronthaul links, which also connect all the CPUs together [5]. There are two types of CFmMIMO implementations categorized by various levels of cooperation between APs/CPUs, namely centralized and distributed implementations [1]. This paper adopts the centralized implementation (which has significantly higher SE than the distributed implementation [5]). In the centralized operation, the APs only function as relays that forward their received UL signals to their associated CPUs and forward DL signals generated by CPUs to the UEs. The CPUs are responsible for channel estimation, receive combining, and data detection. We denote by $\mathcal{L}_u \subseteq \mathcal{L}$ the set of APs connected to CPU u.

User equipment (UEs): The set of K single-antenna UEs is denoted by $\mathcal{K}$. The diagonal matrix decision variable $\mathbf{D}_{kl}$ decides which antennas of AP l may serve UE k. In this paper, UE k is served by all antennas of AP l if the AP is assigned to serve that UE, i.e., $\mathbf{D}_{kl} = \mathbf{I}_N$, and $\mathbf{D}_{kl} = \mathbf{0}_N$ otherwise. We let block-diagonal matrix $\mathbf{D}_k = \text{diag}(\mathbf{D}_{k1}, ..., \mathbf{D}_{kL})$. We use $\mathcal{L}_k \subseteq \mathcal{L}$ and $\mathcal{U}_k \subseteq \mathcal{U}$ to denote the sets of APs and CPUs serving UE $k \in \mathcal{K}$. Among $\mathcal{U}_k$, a CPU is selected as the *master CPU* of UE k to handle all the signal processing, including channel estimation, receive combining, and data detection. The other CPUs will just act as relays that forward their received signals to the master CPU (on the UL) and to their associated APs serving the UE (on the DL).

B. Communication model

The channel $\mathbf{h}_{kl} \in \mathbb{C}^N$ between the k-th UE and the l-th AP is a correlated Rayleigh fading channel, $\mathbf{h}_{kl} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{R}_{kl})$, where $\mathbf{R}_{kl} \in \mathbb{C}^{N \times N}$ is the correlation matrix describing the large-scale fading (LSF), which includes spatial channel correlation, pathloss, antenna gains, and shadowing. We considers the absence of dominant line-of-sight (LoS) paths between APs and UEs, which is the case in scenarios such as heavily built-up city centers with rough terrain, many tall buildings and obstacles, hence the use of Rayleigh fading channel model in the system model. For other settings where there is a strong, direct LoS path between the transmitter and receiver (e.g., industrial settings), Rician fading can be used. For a comprehensive performance analysis on Rician fading in CFmMIMO, readers can refer to [17], [18]. The average channel gain between the k-UE and an antenna at the l-th AP can be expressed by the normalized trace:

$$\beta_{kl} = \frac{\text{tr}(\mathbf{R}_{kl})}{N} \tag{1}$$

The average channel gain β_{kl} is also referred to as LSF coefficient (LSFC) and modeled as [5], [19]:

$$\beta_{kl} = \Upsilon - 10\alpha \log_{10}(d_{kl}) + F_{kl} \tag{2}$$

where Υ is the median channel gain at 1 km reference distance, α is the path loss exponent, d_{kl} is the propagation distance between UE k and AP l, and $F_{kl} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{\text{SF}})$ is the shadow fading with σ^2_{SF} being the shadow fading variance. The collective channel between UE k and all the APs is denoted by $\mathbf{h}_k = [\mathbf{h}^{\text{T}}_{k1} \ldots \mathbf{h}^{\text{T}}_{kL}]^{\text{T}} \in \mathbb{C}^{NL}$. The APs' channel vectors are assumed to be independently distributed, thus $\mathbf{h}_k \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \mathbf{R}_k)$, with $\mathbf{R}_k = \text{diag}(\mathbf{R}_{k1}, \ldots, \mathbf{R}_{kL}) \in \mathbb{C}^{NL \times NL}$ being the block-diagonal spatial correlation matrix.

The system operates in TDD mode where each τ_c-sample frame (coherence interval) is split into three phases: pilot transmission of τ_p samples for UL channel estimation (UL training), UL data transmission of τ_u samples, and DL data transmission of τ_d samples. In UL training, all the UEs transmit pilot data to the APs to enable channel estimation at the CPUs. There are τ_p mutually orthogonal τ_p-length pilot signals. We use $t_k \in \{1, ..., \tau_p\}$ to denote the index of the pilot allocated to UE k[1]. The set of UEs sharing the same pilot as UE k, including UE k, is $\mathcal{K}_{t_k} \subset \mathcal{K}$. When those UEs transmit the pilot, the signal received at AP l during the entire pilot transmission is:

$$\mathbf{y}^{\text{pilot}}_{t_k l} = \sum_{i \in \mathcal{K}_{t_k}} \sqrt{\tau_p p_i} \mathbf{h}_{il} + \mathbf{n}_{t_k l} \tag{3}$$

where p_i is UE i's transmit power, $\mathbf{n}_{t_k l} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \sigma^2_{\text{UL}} \mathbf{I}_N)$ is the additive white Gaussian noise (AWGN). This will then be forwarded to the associated CPUs (and ultimately gathered at one master CPU) via the fronthaul links for channel estimation. Using the received pilot signals, the CPU can calculate all the channel estimate $\{\hat{\mathbf{h}}_{kl} : l \in \mathcal{L}_k\}$. $\mathbf{h}_{kl}$'s minimum mean squared error (MMSE) estimate, $\forall k \in \mathcal{K}_{t_k}$, is given by [5]:

$$\hat{\mathbf{h}}_{kl} = \sqrt{\tau_p p_k} \mathbf{R}_{kl} \mathbf{\Psi}^{-1}_{kl} \mathbf{y}^{\text{pilot}}_{t_k l} \tag{4}$$

with $\mathbf{\Psi}_{kl} = \mathbb{E}\{\mathbf{y}^{\text{pilot}}_{t_k l} (\mathbf{y}^{\text{pilot}}_{t_k l})^{\text{H}}\} = \sum_{i \in \mathcal{K}_{t_k}} \tau_p p_i \mathbf{R}_{il} + \sigma^2_{\text{UL}} \mathbf{I}_N$ being the correlation matrix of $\mathbf{y}^{\text{pilot}}_{t_k l}$. The mutual interference in $\mathbf{y}^{\text{pilot}}_{t_k l}$ caused by UEs sharing the same pilot ($\mathcal{K}_{t_k}$) is referred to as the pilot contamination. Pilot contamination reduces the channel estimation accuracy and makes the channel estimates $\hat{\mathbf{h}}_{kl}$ of co-pilot UEs $\mathcal{K}_{t_k}$ become correlated, causing coherent interference [19]. As combining/precoding vectors rely on channel estimates, pilot contamination could reduce the signal-to-interference-plus-noise ratio (SINR) and SE in both UL and DL.

After the UL training phase, the i-th UE transmits intermediary data symbol $s_i \in \mathbb{C}$ to the APs. Let $\mathbf{n}_l \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}_N, \sigma^2_{\text{UL}} \mathbf{I}_N)$ be the noise term, the UL data signal received at the l-th AP is:

$$\mathbf{y}^{\text{UL}}_l = \sum_{i \in \mathcal{K}} \mathbf{h}_{il} s_i + \mathbf{n}_l \tag{5}$$

This will be relayed to its associated CPU for centralized data detection. We adopt the partial MMSE (P-MMSE) combining scheme proposed in [1]. Compared to the conventional MMSE combining scheme, the calculation of P-MMSE combining vector for a UE k only involves the UEs partially served by the same APs as UE k. Its complexity is independent of the total number of UEs K, thus it is fully scalable and suitable for large-scale networks, which is the aim of our paper. In terms of performance, the average SE achieved by P-MMSE is within 89% of the optimal MMSE combining [1].

In DL data transmission phase, the CPUs compute the normalized precoding vectors $\bar{\mathbf{w}}_{il} \in \mathbb{C}^N$ for UE i at AP l using the UL channel estimates, generating AP l's DL signal:

$$\mathbf{x}^{DL}_l = \sum_{i \in \mathcal{K}} \sqrt{\rho_i} \mathbf{D}_{il} \bar{\mathbf{w}}_{il} \varsigma_i \tag{6}$$

where ρ_i is UE i's allocated transmit power and $\varsigma_i \in \mathbb{C}$ is UE i's intended data signal ($\mathbb{E}\{\|\varsigma_i\|^2\} = 1$). This DL signal is transmitted to AP l via fronthaul, which then transmits to its associated UEs. Let

[1] Pilot assignment is out of the scope of this paper. There are existing pilot assignment methods that are scalable and suitable for our scenario, such as the heuristic proposed in [1], which will be used in our numerical evaluation.

$n_k \sim \mathcal{N}_{\mathbb{C}}(0, \sigma^2_{\text{DL}})$ be the receiver noise, the DL signal received by UE k is:

$$y_k^{DL} = \sum_{l \in \mathcal{L}} \mathbf{h}_{kl}^{\text{H}} \mathbf{x}_l^{DL} + n_k \tag{7}$$

The SINR of UE k in the DL is (Theorems 6.1 in [5]):

$$\text{SINR}_k^{\text{DL}} = \frac{\rho_k |\mathbb{E}\{\mathbf{h}_k^{\text{H}} \mathbf{D}_k \bar{\mathbf{w}}_k\}|^2}{\sum\limits_{i \in \mathcal{K}} \rho_i \mathbb{E}\{|\mathbf{h}_k^{\text{H}} \mathbf{D}_i \bar{\mathbf{w}}_i|^2\} - \rho_k |\mathbb{E}\{\mathbf{h}_k^{\text{H}} \mathbf{D}_k \bar{\mathbf{w}}_k\}|^2 + \sigma^2_{\text{DL}}} \tag{8}$$

where $\mathbf{w}_k \in \mathbb{C}^{NL}$ is UE k's precoding vector, which is computed by P-MMSE precoding scheme [1] in this paper. The above communications steps take place after the UE-AP association has been carried out. We use this well-known communication model as representative, our UE-AP association approach can also be applied to other communication models as it only requires the information on $\mathbf{R}_{kl}$.

C. Problem formulation

Our objective is to assign UEs to APs such that 1) the fronthaul signaling load is minimized, 2) while maximizing the sum SE. The fronthaul signaling load, quantified by the number of complex scalars transmitted between CPUs, is measured as follows. In each coherence block, in the UL, the l-th AP transmits $\tau_p N$ and $\tau_u N$ complex scalars representing pilot signal $\mathbf{y}_{t_k l}^{\text{pilot}}$ and received data signal $\mathbf{y}_l^{\text{UL}}$, respectively, to its associated CPU. In the DL, the CPU returns $\tau_d N$ complex scalars to AP l representing DL signal $\mathbf{x}_l^{DL}$ [1], [5]. This CPU will then need to relay some of its received signals (complex scalars) to other CPUs, which incurs inter-CPU fronthaul signaling. Take Fig. 1 for instance, UE 1 is served by APs linked to CPU 1 and CPU 3 with CPU 1 being the master CPU, which handles all the signal processing. CPU 3 acts as a relay that forwards the signals received by AP 1 and AP 2 to CPU 1 via fronthaul. UE 3 is served by APs also linked to CPU 1 and CPU 3, with CPU 1 also being the master CPU. As the signals previously relayed to CPU 1 by CPU 3 (for UE 1) already include UE 3's signals, no extra inter-CPU signaling is required for UE 3. UE 2 is served by APs linked to one single CPU and does not cause any inter-CPU fronthaul signaling. Formally, the fronthaul load minimization objective can be expressed as follows:

$$\underset{\{\mathbf{D}_k:\forall k\}}{\text{minimize}} \sum_{u \in \mathcal{U}} \sum_{l \in \mathcal{L}_u^{\dagger}} N\tau_c \tag{9a}$$

$$\text{subject to } \mathcal{K}_u = \{k \in \mathcal{K} : \text{tr}(\mathbf{D}_{kl}) \geq 1, l \in \mathcal{L}_u\}, \forall u \in \mathcal{U} \tag{9b}$$

$$\mathcal{K}_u^{\dagger} = \{k \in \mathcal{K}_u : \arg\max_{i \in \mathcal{U}_k} |\mathcal{L}_k \cap \mathcal{L}_i| = u\}, \forall u \in \mathcal{U} \tag{9c}$$

$$\mathcal{L}_u^{\dagger} = \{l : l \in \mathcal{L}_k \setminus \mathcal{L}_u, k \in \mathcal{K}_u^{\dagger}\}, \forall u \in \mathcal{U} \tag{9d}$$

Objective (9a) minimizes the inter-CPU fronthaul signaling load, measured by the number transmitted complex scalars. Set $\mathcal{K}_u$ (9b) denotes the set of UEs served by APs associated with CPU u. Set $\mathcal{K}_u^{\dagger}$ (9c) denotes the set of UEs whose master CPU is CPU u. The master CPU of a UE is defined as the CPU that has the most number of associated APs serving the UE. Set $\mathcal{L}_u^{\dagger}$ (9d) denotes the set of APs serving UEs whose master CPU is CPU u, excluding APs associated with master CPU u. Note that $\mathcal{L}_u^{\dagger}$ may contain APs associated with multiple CPUs. APs associated with the master CPU are excluded from this set because the signals received by those APs do not have to be relayed to any other CPU.

Following the primary objective of minimizing the fronthaul signaling load, we aim to maximize the sum-SE of all UEs.

$$\underset{\{\mathbf{D}_k:\forall k\}}{\text{maximize}} \sum_{k \in \mathcal{K}} \frac{\tau_d}{\tau_c} \log_2 \left(1 + \text{SINR}_k^{\text{DL}}\right) \tag{10a}$$

III. Proposed Hybrid Network- and User-centric Online User Association

Our proposed approach is summarized in Algorithm 1. When UE k joins the network, it measures its channel quality to every network-centric cluster, which is a set of APs associated with the same CPU (Line 2). The channel quality β_{ku} between UE k and CPU u's cluster is approximated by the sum of large-scale fading coefficients (LSFC) β_{kl} between the UE and all APs in the cluster: $\beta_{ku} = \sum_{l \in \mathcal{L}_u} \beta_{kl}$.

The next step is the key part of our solution - identify if this UE would be likely to receive sufficient SE from the APs linked to the same CPU, or if the UE would be better off being served by APs linked to multiple CPUs, which requires inter-CPU fronthaul signaling. To do so, we calculate z-scores z_{ku} for each $\beta_{ku}, \forall u \in \mathcal{U}$ (Line 3): $z_{ku} = \frac{\beta_{ku} - \mu_k^{\text{LSFC}}}{\sigma_k^{\text{LSFC}}}$, where $\mu_k^{\text{LSFC}} = \frac{\sum_{u \in \mathcal{U}} \beta_{ku}}{U}$ and $\sigma_k^{\text{LSFC}} = \sqrt{\frac{\sum_{u \in \mathcal{U}} (\beta_{ku} - \mu_k^{\text{LSFC}})^2}{U}}$.

Z-score is a statistical measure measuring how far a value is from the mean of a group of values; e.g., a z-score of 1 means that β_{ku} is one standard deviation away from the mean of all ${\beta_{ku}}^2$. We set a threshold ϵ for the z-scores. If there is only one β_{ku} exceeding this threshold and also being the highest ${\beta_{ku}}^3$ (Line 4), then this cluster u stands out from all the other network-centric clusters, i.e., channel quality well above the others, and this UE is likely to be sitting in the middle of this network-centric cluster u, and likely to receive good service from APs belonging to just this cluster (*network-centric clustering*). In this case, we adopt the technique proposed in [20] to determine the selective best APs, which collectively contribute at least $\delta\%$ of the total LSFC of all APs connected to this cluster u, to serve UE k.

Algorithm 1 Hybrid online user association (HybridUA)

```
 1: for each UE $k \in \mathcal{K}$ do
 2:     Measure channel quality $\beta_{ku}, \forall u \in \mathcal{U}$
 3:     Calculate z-scores $z_{ku}, \forall u \in \mathcal{U}$
 4:     if $|\{z_{ku} | z_{ku} \geq \epsilon, \forall u \in \mathcal{U}\}| = 1$ and $u = \arg\max_u \beta_{ku}$
    then
 5:         $\mathcal{U}_k^{\text{top}} = \arg\max_u \beta_{ku}$
 6:     else
 7:         $\mathcal{U}_k^{\text{top}}$ is top $\upsilon$ CPUs with highest $\beta_{ku}$
 8:     end if
 9:     Find APs $\mathcal{L}_k$ such that $\frac{\sum_{l \in \mathcal{L}_k} \beta_{kl}}{\sum_{u \in \mathcal{U}_k^{\text{top}}} \sum_{l \in \mathcal{L}_u} \beta_{kl}} \geq \delta\%$
10:     $\mathbf{D}_{kl} = \mathbf{I}_N, \forall l \in \mathcal{L}_k$
11: end for
```

In other cases where multiple β_{ku} exceed ϵ, does it means that those clusters have equally good channel quality? Not necessarily. It just means that this UE is possibly sitting in the middle/intersection of those clusters. The same applies to the cases where no β_{ku} exceeds ϵ. In such cases, it would more beneficial for the UE to receive service from APs linked to different CPUs/clusters (*user-centric clustering*). We define a parameter υ to enforce the maximum number of CPUs/clusters a UE can be assigned to. Again, we apply the technique proposed in [20] to select APs that collectively contribute at least $\delta\%$ of the total LSFC of all APs connected to the top-υ clusters, which are the clusters with highest sum LSFC, to serve UE

[2] In a very strict statistical sense, calculating z-scores makes more sense for normally distributed data. While we cannot guarantee the distribution of β_{ku} being a normal distribution, our numerical results demonstrate that this method is quite effective.

[3] Checking if β_{ku} is the highest is required because exceeding this threshold could also mean that β_{ku} is ϵ-standard deviation *below* the mean, which infers poor channel quality.

k. Among those υ CPUs, one with more APs selected to serve the UE will be assigned as the master CPU. The other CPU(s) simply just relay their received UE signals to this master CPU for signal processing. The time complexity of finding an association decision for each UE (Lines 2-10 of Algorithm 1) is $\mathcal{O}(U \log L)$, which is independent of the total number of UEs K. Its complexity does not grow with the number UEs in the network and thus is scalable.

Algorithm 1 involves three thresholds, ϵ, υ, and δ, controlling the trade-off between SE and fronthaul signaling load. Decreasing ϵ, increasing υ and δ will increase the SE at the cost of extra fronthaul signaling. However, our simulations show that this SE improvement is often very marginal (up to 2%) and not worth the extra fronthaul signaling load (up to 70%). It appears there is a sweet spot for these thresholds (LSFC contribution threshold $\delta = 95\%$ and per-UE CPU/cluster limit $\upsilon = 2$ CPUs/clusters) that is robust and consistent across all simulation settings (Sect. IV).

In summary, the aspects of fronthaul load minimization and SE maximization are interconnected throughout the proposed algorithm via channel quality (LSFC). First, each UE identifies if there is any network-centric cluster whose sum LSFC is significantly higher than that of all other clusters (Lines 2-5 of Algorithm 1). In this case, inter-CPU fronthaul signaling is not required while the UE is still likely to achieve high SE. If there is no such cluster, the UE will opt to be served by APs linked to top-υ clusters with the strongest channels in order to achieve better SE.

IV. Numerical Results

A. Simulation settings and benchmark approaches

Our CFmMIMO network contains K UEs, L N-antenna APs, and U CPUs. The AP-CPU association is generated via k-means clustering based on geographical locations. The UEs and APs are uniformly randomly distributed in a square area. The median channel gain at 1 km reference distance $\Upsilon = -148.1$ dB, path loss exponent $\alpha = 3.76$, shadow fading standard deviation $\sigma_{\text{sf}} = 10$. The bandwidth is 20 MHz with receiver noise power $\sigma^2 = -94$ dBm consisting of thermal noise and 7 dB noise figure. We set $\tau_p = 10$ and $\tau_d = 190$ when evaluating DL. All UEs have equal transmit power $p_k = 100$mW in the UL for all approaches in the benchmark. In the DL, we adopt a scalable centralized fractional power allocation ([5], Sec. 7.2.2). For pilot allocation, all approaches apply the scalable heuristic proposed in [1]. In all simulations, z-score threshold ϵ is set to 0.4, the number of antennas per AP (N) is 4, the total LSFC contribution threshold δ is 95%, and per-UE CPU limit υ is 2 CPUs.

We evaluate our proposed approach (HybridUA) against the following user association schemes. Note that we only consider schemes that are designed to be scalable.

- SCF1: Scalable cell-free (SCF) scheme proposed in [1], based on dynamic cooperation clustering framework [21] and does not take into account the cooperation between CPUs.
- SCF2: Scalable cell-free scheme proposed in [10]. Each UE is served by all APs associated with a fixed number of CPUs, which is 2 in our simulations.
- SCF1lim: The SOTA on inter-CPU coordination minimization, proposed in [12]. It fine-tunes any arbitrary UE-AP association solutions, which are produced by SCF1 in this paper, by dropping the UEs coordinated by multiple CPUs and have the worst channels.
- Border: It limits the inter-CPU coordination based on UEs' distance to the borders (100m in this paper) of network-centric clusters, proposed in [11]. UEs near the border are served by multiple CPUs and require inter-CPU signaling while the other UEs do not.

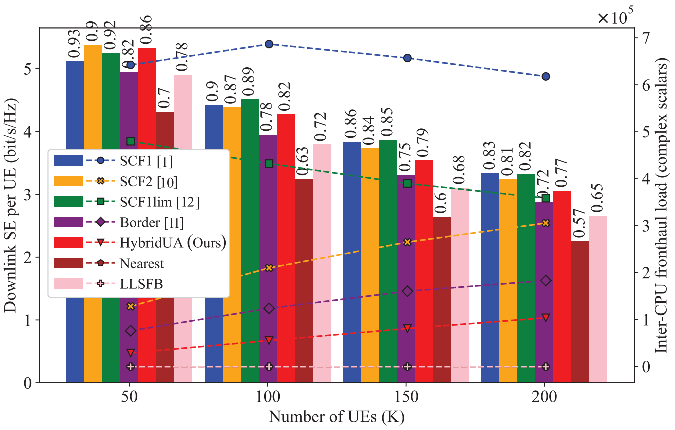

Fig. 2: The bars correspond to the left y-axis, which represents per-UE DL SE. The lines correspond to the right y-axis, which represents inter-CPU fronthaul signaling load. The number above each bar is the Jain's fairness index of the UEs' DL SE.

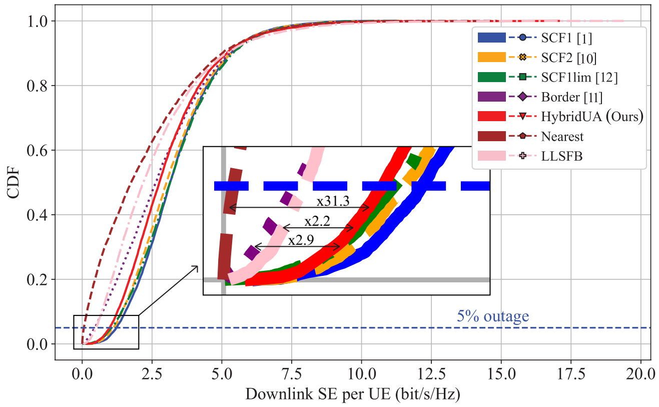

Fig. 3: CDF of per-UE DL SE.

- LLSFB (Largest-Large-Scale-Fading-Based): A network-centric variant of the LLSFB user association scheme proposed in [20] that we came up with. For each UE, we first identify which network-centric cluster has the greatest sum LSFC, then select APs that collectively contribute at least δ% of the total LSFC of all APs in this cluster to serve the UE.
- Nearest: This naive network-centric scheme assigns each UE to all APs associated with the UE's nearest network-centric cluster. Nearest and LLSFB do not incur inter-CPU fronthaul signaling.

Beside the key performance metrics, which are inter-CPU fronthaul signaling load and per-UE SE, we also report the numbers of APs serving a UE, the numbers of UEs served by an AP or a CPU, and fairness. The fairness indicates whether the UEs are receiving equal SE. To quantify the fairness, we utilize Jain's fairness index, calculated as $(\sum_{k\in\mathcal{K}} \text{SE}_k)^2/(K\sum_{k\in\mathcal{K}} \text{SE}_k^2)$, which ranges from 0 to 1, with 1 being the fairest. UL results, which are fairly similar to DL results due to channel reciprocity, are omitted for brevity.

B. Impact of network traffic (number of UEs)

This simulation consists of 200 APs and 40 CPUs over a 8km^2 area (100 antennas/km^2, 5 APs/CPU). The number of UEs is varied between 50 and 200 (6.25-25 UEs/km^2). As shown in Fig. 2, compared to SCF1lim (the latest work on inter-CPU signaling minimization), at the cost of a small loss in per-UE DL SE (up to 8.6% loss), HybridUA can save 71-94% fronthaul signaling. In some cases ($K = 50$), we even observe no loss in SE. As the number of UEs increases, the fronthaul load of HybridUA also increases at a relatively slow rate. Interestingly, the fronthaul load of SCF1 and SCF1lim decreases,

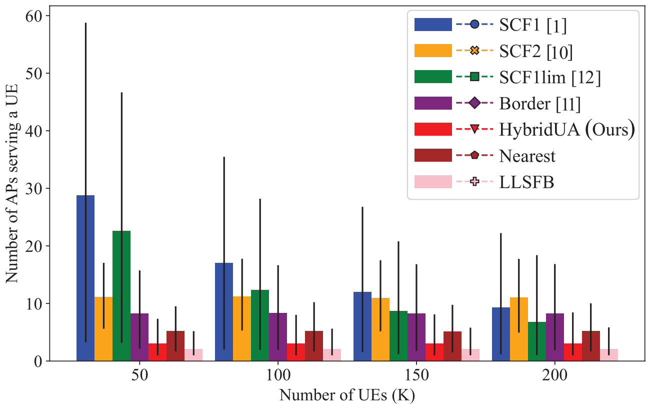

Fig. 4: Average number of APs serving a UE (the colored bars). The error bars illustrate the lowest/highest numbers.

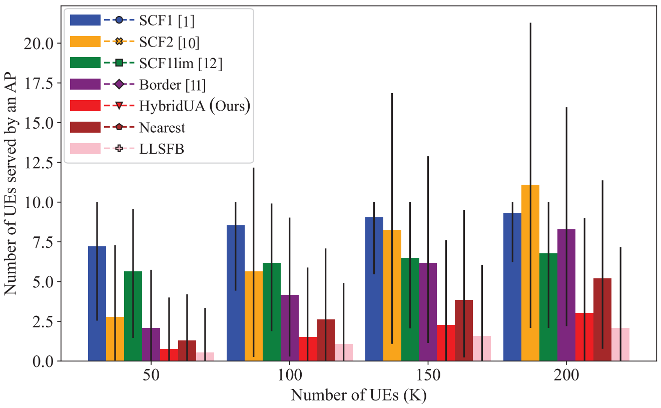

Fig. 5: Average number of UEs served by an AP (the colored bars). The error bars illustrate the lowest/highest numbers.

which happens because those schemes explicitly limit the number of UEs per AP; the more UEs, the fewer APs (and CPUs) serving each UE. Nevertheless, the fronthaul load of SCF1 and SCF1lim are still significantly higher than HybridUA. SCF2 does help with inter-CPU fronthaul load as each UE is served by a fixed number of network-centric clusters. However, its fronthaul load is still three times higher than HybridUA while only offering a very marginal SE improvement. LLSFB and Nearest, although incur no inter-CPU signaling, losing 9-15%, 24-36% SE compared to HybridUA, respectively. Border achieves 6-8% lower SE than HybridUA while uses 43-62% higher fronthaul signaling.

Jain's fairness index (the number above each bar in Fig. 2) provides a quantitative measurement of the fairness/uniformity of the UEs' SE. HybridUA's fairness is at most 0.08 lower than the fairest scheme, and 0.06 lower than SCF1lim. Nearest's and LLSFB's fairness are up to 0.2 and 0.12 lower than HybridUA, respectively. This shows that HybridUA can provide a uniform service to the UEs.

Fig. 3 illustrates the cumulative distribution function (CDF) of per-UE DL SE with 200 UEs, which confirms HybridUA's service uniformity. Compared to HybridUA, LLSFB, Border, and Nearest have 2.2×, 2.9×, and 31.3× lower per-UE SE in the 5% outage (top 5% most unfortunate UEs), respectively. This shows how HybridUA can make a substantial improvement to the most unfortunate UEs.

Fig. 4 depicts that we do not need a very large number of APs to serve each UE in order to achieve satisfactory SE. In fact, on average, HybridUA requires only 3 APs (8 APs at most) per UE regardless of the number of UEs being simulated - significantly fewer than all other schemes except LLSFB. This is the reason why our scheme performs so well. The other schemes may require up to 10× more APs per UE on average only to achieve a 8.6% increase in SE. Fig. 5 shows that while HybridUA does not explicitly limit the number of UEs per AP like SCF1 and SCF1lim do, the number of UEs per AP by HybridUA is much lower than that of SCF1 and SCF1lim. Thus, the effect of pilot contamination is minimized as the UEs are less likely to share a pilot. This also proves the scalability of our approach with regards to processing power of APs. Fig. 6 shows that on average, HybridUA requires each CPU to serve just a few UEs - 49-83% fewer UEs per CPU than SCF1lim, saving a substantial amount of computing power on CPUs (for channel estimation, receive combining, and data detection) and making HybridUA even more scalable than SCF1 and SCF1lim.

C. Impact of the number of CPUs and area size

The first three groups of bars in Fig. 7 shows the results of various AP/CPU ratios, ranging from 10 to 5 APs per CPU. Compared to the SOTA (SCF1lim), HybridUA only loses up to 1.7% SE while saves up to 33% fronthaul load. HybridUA's fronthaul load remains virtually unchanged as the number of CPUs increases; whereas the load incurred by SCF1 and SCF1lim grows steadily. SCF2 and Border's fronthaul load also remains relatively unchanged. Nevertheless, SCF2's fronthaul load is still twice as high as HybridUA's load only to achieve a 2-6% SE improvement. Again, Border's SE and fairness are lower than HybridUA while requiring more fronthaul signaling.

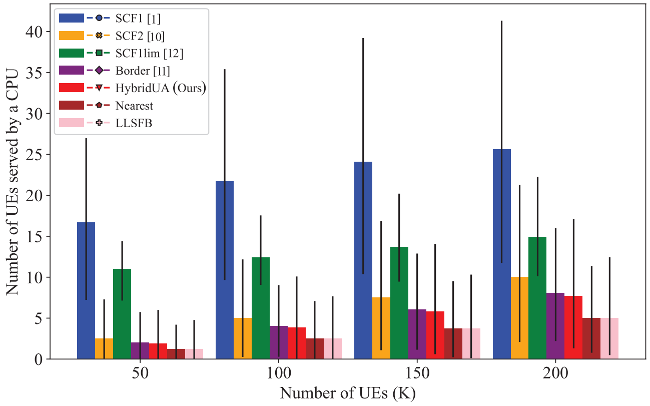

Fig. 6: Average number of UEs served by a CPU (the colored bars). The error bars illustrate the lowest/highest numbers.

We also evaluate the impact of area size in the third and last groups of bars in Fig. 7, where we double the area from 8 to 16km^2 to make the density of APs and UEs more sparse. When the area size increases, HybridUA suffers the second least performance degradation (11%, after LLSFB at 8%). As a result, HybridUA's SE is now on par with other methods while still consuming the least fronthaul signaling. SCF2 and Border suffer the most with 22% SE loss.

D. Discussion

We found that a only very small number of APs make a strong contribution to the service of a UE (Fig. 4). This makes our hybrid network- and user-centric clustering approach a viable option that is particularly attractive to systems with limited fronthaul capacity. The other benchmark approaches have not taken advantage of this observation. The network-centric schemes (Nearest and LLSFB) are not terribly bad and may even be acceptable for extreme cases with little to no inter-CPU fronthaul capacity; but their SE, fairness, and 5% outage (Fig. 3) can be significantly improved by partially incorporating user-centric clustering like we do.

In terms of scalability, our approach meets all the criteria of a scalable CFmMIMO system as defined in [1]. It is even much more scalable than SCF [1] and SCF1lim [12] measured by the processing

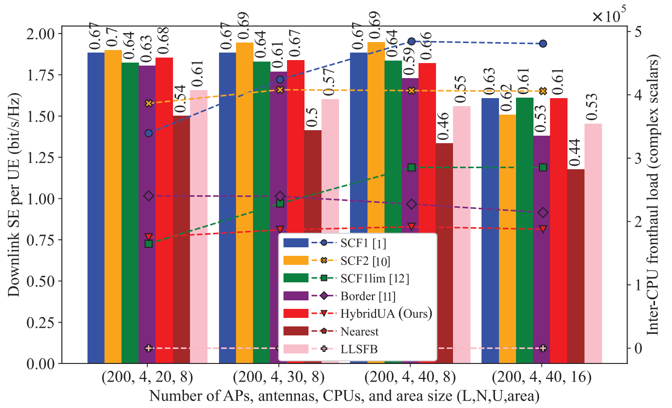

Fig. 7: The bars correspond to the left y-axis, which represents per-UE DL SE. The lines correspond to the right y-axis, which represents inter-CPU fronthaul signaling load. Each four-number tuple on the x-axis represents (total number of APs, number of antennas per AP, total number of CPUs, area size in km^2). There are 200 APs, 400 UEs, and 20/30/40 CPUs in a 8km^2 area (first three tuples), and 40 CPUs in a 16km^2 area (last tuple). The number above each bar is the Jain's fairness index of the UEs' DL SE.

power required on CPUs (Fig. 6), which is quantified by the number of UEs that a CPU has to serve. Our approach is also highly flexible in handling the trade-off between SE and fronthaul load through adjusting the z-score threshold ϵ - the higher the ϵ, the fewer "cell-edge" UEs, the lower the SE and the greater the saving of fronthaul signaling, and vice versa.

The robustness of our approach is also highlighted in Fig. 7 when we increase the geographical area size. While other approaches experience a big SE loss, our approach still fares very well and becomes the best approach.

V. Conclusion and Future Work

We proposed a hybrid network- and user-centric online user association scheme for CFmMIMO that is highly scalable and robust. We showed that it is a viable option worth considering as we can save a considerable amount of inter-CPU fronthaul signaling and CPU processing power with a small loss, or no loss in some cases, of spectral efficiency and fairness.

Following this work, there are several potential research directions. With the distributed implementation of CFmMIMO, there might exist a correlation between the load per AP to its CPU and the fronthaul loads between the CPUs. A theoretical quantification of this correlation could be an interesting contribution. Some other avenues of interest for CFmMIMO include the integration of machine learning techniques for adaptive clustering or the impact of mobility on UE-AP association in dynamic environments.

References

[1] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, 2020.

[2] H. Q. Ngo, G. Interdonato, E. G. Larsson, G. Caire, and J. G. Andrews, "Ultra-dense cell-free massive MIMO for 6G: Technical overview and open questions," *arXiv preprint arXiv:2401.03898*, 2024.

[3] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-free massive MIMO versus small cells," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, 2017.

[4] S. Buzzi and C. D'Andrea, "Cell-free massive MIMO: User-centric approach," *IEEE Wireless Commun. Lett.*, vol. 6, no. 6, pp. 706–709, 2017.

[5] Ö. T. Demir, E. Björnson, L. Sanguinetti *et al.*, "Foundations of user-centric cell-free massive MIMO," *Foundations and Trends in Signal Processing*, vol. 14, no. 3-4, pp. 162–472, 2021.

[6] M. Guenach, A. A. Gorji, and A. Bourdoux, "Joint power control and access point scheduling in fronthaul-constrained uplink cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2709–2722, 2020.

[7] M. Zaher, Ö. T. Demir, E. Björnson, and M. Petrova, "Learning-based downlink power allocation in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 174–188, 2022.

[8] J. Kassam, D. Castanheira, A. Silva, R. Dinis, and A. Gameiro, "Joint Decoding and UE-APs Association for Scalable Cell-Free Systems," *IEEE Trans. Commun.*, 2023.

[9] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "Downlink resource allocation in multiuser cell-free MIMO networks with user-centric clustering," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1482–1497, 2021.

[10] G. Interdonato, P. Frenger, and E. G. Larsson, "Scalability aspects of cell-free massive MIMO," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[11] V. Ranjbar, A. Girycki, M. A. Rahman, S. Pollin, M. Moonen, and E. Vinogradov, "Cell-free mMIMO support in the O-RAN architecture: A PHY layer perspective for 5G and beyond networks," *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 28–34, 2022.

[12] M. M. Freitas, D. D. Souza, D. B. da Costa, A. M. Cavalcante, L. Valcarenghi, G. S. Borges, R. Rodrigues, and J. C. Costa, "Reducing Inter-CPU Coordination in User-Centric Distributed Massive MIMO Networks," *IEEE Wireless Commun. Lett.*, 2023.

[13] Z. Li, F. Göttsch, S. Li, M. Chen, and G. Caire, "Joint Fronthaul Load Balancing and Computation Resource Allocation in Cell-Free User-Centric Massive MIMO Networks," *arXiv preprint arXiv:2310.14911*, 2023.

[14] C. Wei, K. Xu, X. Xia, Q. Su, M. Shen, W. Xie, and C. Li, "User-centric access point selection in cell-free massive MIMO systems: A game-theoretic approach," *IEEE Commun. Lett.*, vol. 26, no. 9, pp. 2225–2229, 2022.

[15] F. Guo, H. Lu, and Z. Gu, "Joint power and user grouping optimization in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 991–1006, 2021.

[16] T. X. Vu, S. Chatzinotas, S. ShahbazPanahi, and B. Ottersten, "Joint power allocation and access point selection for cell-free massive MIMO," in *IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[17] Y. Zhang, H. Zhao, Y. Mao, W. Xia, W. Lu, and H. Zhu, "Rate-Splitting Multiple Access in Cell-Free Massive MIMO-URLLC Systems: Achievable Rate Analysis and Optimization," *IEEE Trans. Commun.*, 2024.

[18] Y. Zhang, W. Xia, H. Zhao, Y. Zhu, W. Xu, and W. Lu, "Performance analysis of cell-free massive MIMO-URLLC systems over correlated Rician fading channels with phase shifts," *IEEE Trans. Wireless Commun.*, 2024.

[19] E. Björnson, J. Hoydis, L. Sanguinetti *et al.*, "Massive MIMO networks: Spectral, energy, and hardware efficiency," *Foundations and Trends® in Signal Processing*, vol. 11, no. 3-4, pp. 154–655, 2017.

[20] H. Q. Ngo, L.-N. Tran, T. Q. Duong, M. Matthaiou, and E. G. Larsson, "On the total energy efficiency of cell-free massive MIMO," *IEEE Transactions on Green Communications and Networking*, vol. 2, no. 1, pp. 25–39, 2017.

[21] E. Bjornson, N. Jalden, M. Bengtsson, and B. Ottersten, "Optimality properties, distributed strategies, and measurement-based evaluation of coordinated multicell OFDMA transmission," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6086–6101, 2011.