Lovén *et al. J Wireless Com Network* (2022) 2022:86
https://doi.org/10.1186/s13638-022-02170-y

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

A dark and stormy night: Reallocation storms in edge computing

Lauri Lovén[1*], Ella Peltonen[1], Leena Ruha[2], Erkki Harjula[3] and Susanna Pirttikangas[1]

*Correspondence: lauri.loven@oulu.fi

[1] Center for Ubiquitous Computing, University of Oulu, Oulu, Finland
[2] Natural Resources Institute Finland, Oulu, Finland
[3] Center for Wireless Communication, University of Oulu, Oulu, Finland

Abstract

Efficient resource usage in edge computing requires clever allocation of the workload of application components. In this paper, we show that under certain circumstances, the number of superfluous workload reallocations from one edge server to another may grow to a significant proportion of all user tasks—a phenomenon we present as a reallocation storm. We showcase this phenomenon on a city-scale edge server deployment by simulating the allocation of user task workloads in a number of scenarios capturing likely edge computing deployments and usage patterns. The simulations are based on a large real-world data set of city-wide Wi-Fi network connections, with more than 47M connections over ca. 560 access points. We study the occurrence of reallocation storms in three common edge-based reallocation strategies and compare the latency–workload trade-offs related to each strategy. As a result, we find that the superfluous reallocations vanish when the edge server capacity is increased above a certain threshold, unique for each reallocation strategy, peaking at ca. 35% of the peak ES workload. Further, while a reallocation strategy aiming to minimize latency consistently resulted in the worst reallocation storms, the two other strategies, namely a random reallocation strategy and a bottom-up strategy which always chooses the edge server with the lowest workload as a reallocation target, behave nearly identically in terms of latency as well as the reallocation storm in dense edge deployments. Since the random strategy requires much less coordination, we recommend it over the bottom-up one in dense ES deployments. Moreover, we study the conditions associated with reallocation storms. We discover that edge servers with the very highest workloads are best associated with reallocation storms, with other servers around the few busy nodes thus mirroring their workload. Further, we identify circumstances associated with an elevated risk of reallocation storms, such as summertime (ca. 4 times the risk than on average) and on weekends (ca. 1.5 times the risk). Furthermore, mass events such as popular sports games incurred a high risk (nearly 10 times that of the average) of a reallocation storm in a MEC-based scenario.

Keywords: Edge computing, Reallocation storm, Load balancing

1 Introduction

Edge computing has been foreseen as an integral part of future 5G and beyond computing environments [1, 2]. Particularly computing tasks of spatio-temporal environments, spanning from smart cities, healthcare units and logistic centres to large-scale transportation operators and autonomous vehicles, are especially dependant on local and

efficient computational capabilities. Edge computing should allow such environments to utilize local ubiquitous intelligence [3] and data processing [4] close to the data production, with or without continuous cloud support.

However, edge computing architectures come with several already envisioned challenges, including computational optimization and physical placement of the edge servers in dynamic scenarios with mobile users [5, 6]. Particularly load balancing has seen as a mission-critical challenge for any computing service from cloud to local networking capabilities [7]. In everyday language known as a "rock festival phenomena", certain events—whether predictable or not—can cause serious lacks in the quality of service or even exterminate the service for an amount of time. In such a situation, if considering critical environments such as logistics, automatic driving or computer-assisted surgery, can circumstances fast forward to fatal if service gaps are not addressed appropriately and apropos.

Even though certainly studied in the cloud computing context [8, 9], it is not clear if similar load-balancing strategies fit for the resource-constrained edge computing world. This is due to many differences between virtually centralized cloud architectures and physically distributed edge computing devices. For example, coherence of the network topology cannot be trusted to be completely fixed in the edge computing environment where some of the key devices or at least clients are mobile. Typically, edge environments consist of heterogeneous devices that vary in available capabilities, resources and dedication to serving edge clients, compared to the cloud setting, where systems and services are dedicated to their primary task of serving clients and are at least virtually corresponding to each other, and their pre-requirements are known.

To further study edge environments and to *characterize the spatio-temporal phenomena* they introduce, we have explored both *horizontal and vertical workloads* with real-life data transmission. This paper focuses on the phenomena we call *reallocation storms*, expanding our earlier work [10]. We highlight how these horizontal and vertical workloads become bottlenecks of edge computing services and how *reallocation strategies* designed for clouds fail in the edge computing environment. Finally, we discuss how these overloading situations can be predicted and avoided.

1.1 Definitions and scope

Workload is the computational burden on edge servers (ES) or cloud, caused by user tasks accumulating on those servers [11, 12]. Workload may be the result of, for example, user applications or their components offloaded on the edge servers [13]; by edge applications following users, migrating from one ES to another [14, 15]; or by cloud applications being onloaded to edge servers for low-latency interaction with users or environment [13, 16].

In offline edge server placement, expected workload from user tasks is allocated on the edge servers as a part of the process of finding the optimal placement for the servers. Workload allocation designates each access point (AP) (at least) one edge server, which acts as the default server for the workload from the users of that AP when the edge deployment is online [12]. ESs serve two types of workload: *vertical* workload is caused by user tasks on the APs allocated to the ES, while *horizontal* workload is

reallocated from other ESs. The grid computing paradigm can be seen analogous to edge computing in the sense that the grid data centres serve both vertical and horizontal workload [17].

Reallocation is an online process, where the allocation of an AP is changed [18]. Reallocation may be caused by the allocated edge server being unavailable due to, for example, exceeded capacity. Depending on the edge computing architecture, reallocation may be the responsibility of an orchestrator [19].

Figure 1 illustrates an example of reallocation. ES 1 is over capacity and cannot accept the workload from AP 1.1. Consequently, AP 1.1 has to reallocate new user workload to either ES 2 or ES 3 or else offload the workload to cloud. Depending on the type of the edge application, reallocation may require the migration of data from one ES to another. For example, if the edge application in question maintains connection-specific session or state, that state needs to migrate.

Reallocating workload, the target ES needs to be decided (Fig. 1). While the details vary, proposed strategies often fall into the following broad categories: cloud strategy always offloads the workload to cloud, if the allocated ES is over capacity [20, 21]. This is the default strategy the edge strategies may revert to if they, for some reason, fail. Edge strategies try to find another ES in the edge deployment as a reallocation target. There are a number of subcategories:

- Proximity strategy reallocates workload to the nearest ES with available capacity [22, 23].
- Bottom-up strategy reallocates workload to the ES with the lowest current workload [17, 18, 24]. This strategy does not minimize communication latency. Depending on the edge architecture, it may require constant communication between the ESs and

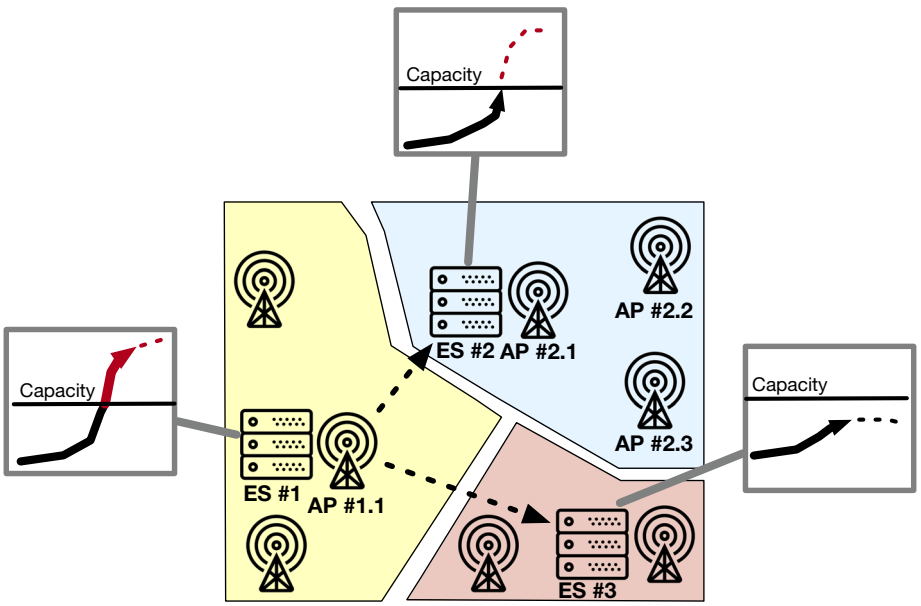

Fig. 1 Reallocation of workload. ES 1 is over capacity, so AP 1.1 may decide to reallocate new workload from its users to ES 2, the nearest ES (proximity strategy), ES 3, with the lowest workload (bottom-up strategy), or offload the workload to cloud (cloud strategy). (Figure originally in [10].)

an orchestrator to maintain up-to-date book-keeping of ES workloads, queue sizes, or minimum estimated task completion times for each ES.

- Random strategy reallocates workload to a random ES with available capacity. This strategy is also a possible fallback if another strategy fails or produces a number of possible candidates [22].

A poor choice of target ES with the chosen edge strategy may trigger another reallocation. In Fig. 1, if AP 1.1 chooses (or is orchestrated) to follow the proximity strategy and reallocate workload to ES 2, the nearest one, ES 2 will exceed its capacity. Any subsequent vertical workload from any of ES 2's allocated APs (2.1, 2.2 or 2.3) must thus also be reallocated.

Under certain conditions, the number of superfluous reallocations may grow to a high proportion of all user connections, a condition we refer to as a reallocation storm. This happens when a high proportion of ESs are above or nearly above capacity, i.e. when the ES network as a whole is under high workload. In such conditions, superfluous reallocations may trigger yet another round of reallocations if other nearby ESs are also near capacity, etc. In this article, we simulate popular reallocation strategies, analyse the workload–latency trade-off for each of them, study how the reallocation storm is realized for each strategy, and discuss their merits and pitfalls. Further, we study the conditions leading to reallocation storms, their frequency, duration, and impact and find ways to avoid them.

1.2 Related work

Edge computing has become the de facto strategy for bringing computational capabilities and distributed intelligence into the local environments to reduce latency between clients and traditional cloud services [25]. Harnessing the local computational capabilities does not only remove networking load but also enables several real-time applications characterized by hard time constraints and other critical resources, such as data privacy and system-level trust. Application areas for edge computing, or computing in the edge–cloud continuum, include various modern networking and computing areas with diverse requirements for high real-time quality of service. The Internet of things [26, 27], and drones [28], with tens of billions of heterogeneous end devices with sometimes low computing power, call for e.g. low data processing costs and energy savings; hybrid reality [29] calls for low latency to match the human sensory system processing speed; vehicular computing [30, 31] targets safety with low latency; edge for connected healthcare [32] requires privacy and security; and the various robotics, HCI, digital production and sustainable supply chain services for the Industry 4.0 systems [18] all have their own performance prerequisites. With increasing interest in edge intelligence, or EdgeAI, bringing artificial intelligence and machine learning on the edge [3, 33], there will be even more novel applications critically dependent on trustworthy edge resources and their on-demand availability.

Workload management is a widely studied topic in the context of server clusters of cloud computing [8, 9], data centres [34], and grid computing [35]. In edge computing, workload management must, however, deal with user mobility and higher variance in server and network topologies and capacities, thus making it a distinct research topic.

Workload management on the edge can be handled with different strategies, such as the physical placement of edge servers [5, 12, 36] or reallocating services on the software-side with different optimization algorithms [18, 37, 38]. Reallocation can rely on known edge server features, such as capacity, or their current state, such as load or even price [39]. The simplest option may be to reallocate workload to the nearest available edge server, or if a number of candidates are produced, one selected randomly amongst these [22]. More detailed hybrid models can combine edge server placement, resource allocation, and run-time reallocation, also optimizing for proximity [40]. Some of these heuristic reallocation algorithms can minimize both computing and network delay, penalizing for longer computing times of over-capacity edge servers [23, 41].

Of course, cloud offloading from the edge environment [20, 21] is sometimes a viable option, especially if the spatio-temporal dependencies are predictable or applications less critical in terms of the user experience or well-being. However, there are cases where cloud offloading cannot take place due to, for example, requirements for low latency, such as on autonomous driving with the vehicular edge [42] or with mobile augmented reality [29], where either critical services or user's quality of experience (or even physical wellbeing) cannot be compromised by increased latency.

1.3 Contributions

In this paper, we identify a phenomenon we name as the reallocation storm, where reallocations trigger new reallocations, and the number of these superfluous reallocations grows to a significant proportion of all user tasks executed on the edge. Further, we estimate spatial and spatio-temporal dependencies behind the reallocation storms and identify patterns leading to reallocation storms. This paper extends our previous work [10] by expanding overall on the study as well as including new sections on spatio-temporal analysis and further discussion on the results.

Our contributions can be summarized as follows:

- We analyse edge server workload using a real-world data set of city-wide Wi-Fi network connections with more than 47M connections over 550 access points. We simulate a number of scenarios, some roughly corresponding MEC with a small number of high-capacity servers, and others corresponding Fog with a high number of low-capacity servers.
- We demonstrate how popular reallocation strategies can lead to reallocation storms, with up to 15–20% of all tasks during the observation period reallocated needlessly.
- We show that the reallocation storm is highly linked to the capacity of the edge servers and vanishes when the edge server capacity is increased above a certain threshold, unique for each reallocation strategy, peaking at ca. 35% of top workload.
- We find that when looking at the workloads in a deployment of edge servers, the few edge servers with the very highest workloads (i.e. the 95% quantile of workload) are in both scenarios best associated with reallocation storms. Apparently, other servers around the few busy nodes tend to mirror their workload, increasing the risk of reallocation storms.

- Finally, we identify circumstances which are associated with an elevated risk of reallocation storms. For example, spatial dependency is high in summertime and on weekends. Furthermore, mass events such as popular sports games incur a high risk of reallocation storms in the MEC scenario.

The article is organized as follows. Section "Data and methods" describes the origins and composition of the data used in the analysis, as well as the assumptions behind the augmentations on the data set. Further, the section details the methods used in the analysis. Section "Results" describes the analysis results, while section "Discussion" discusses the results, their impact, and limitations. Finally, the article ends with the conclusions.

2 Data and methods

2.1 Base data

We use the connection log of the PanOULU public Wi-Fi network in the city of Oulu, Finland, in 2007–2015 [43], as a basis for the analysis. We select the starting timestamps and the APs of all the connections during the observation period of 2013–2014, for a total number of 47.656.939 individual Wi-Fi connections. There are a total of 898 APs in the data set, 559 of which were active in the city of Oulu in the observation period (Fig. 2).

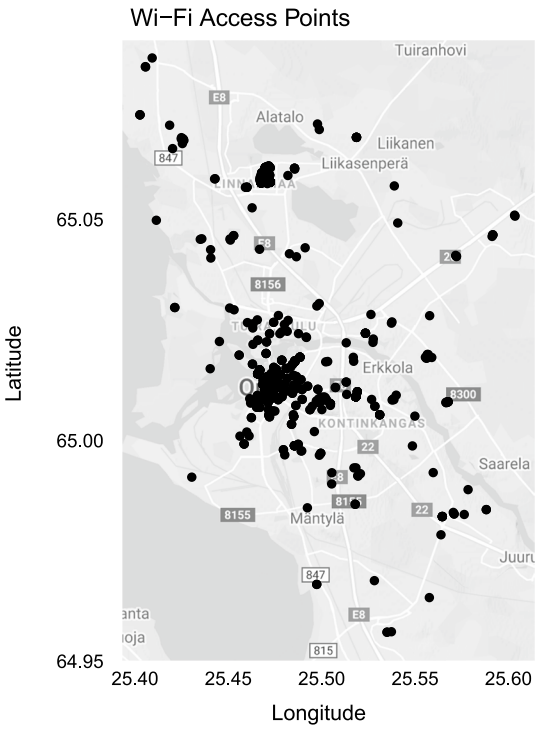

Fig. 2 PanOULU Wi-Fi access points. 559 were active during the observations period of 2013–2014

2.2 Assumptions

Following many of the related work (see [12]), we assume the underlying network is homogeneous in relation to AP density and approximate the latency between the APs with the Euclidean distance. Furthermore, we assume an edge deployment where edge servers (ES) of balanced capacity [12] are deployed at some of the access points in Fig. 2, serving user tasks of variable workload, one per user connection.

To simplify reallocation analysis, we use a predetermined numeric value for the *capacity* of an ES, compared online to the combined workload of the tasks allocated to it. An ES may either be below the capacity, where it is assumed to function normally, or have exceeded its capacity, where it is no longer functioning normally, and will reallocate vertical workload (i.e. the workload of its allocated APs) and refuse horizontal workload (i.e. workload reallocated by other ESs). This reflects a scenario where the workload of an ES exceeds a threshold such that the processing time of a task becomes unacceptable. For an analysis of such a model, see, for example, [38].

2.3 Data augmentation

We enrich the base data set with simulated data on the duration of the user tasks and the normalized workload they incur on the edge servers during that time. The workloads are sampled from the log-normal distribution, the maximum entropy distribution on $(0, \infty)$ with given mean and variance [44, 45], setting the mean and variance as 1 under the assumption of standardized workload.

The task durations are sampled from the exponential distribution with three alternative means ($1/\lambda$): namely, $1/\lambda = 10$s, $1/\lambda = 100$s, and $1/\lambda = 1000$s. We choose the exponential distribution as it is the maximum entropy distribution on $[0, \infty)$ on a given mean with no separate variance parameter, with the standard deviation equal to the mean $1/\lambda$ [44, 45]. The underlying assumption here is that using the three different means, the variances of the distributions follow the means: the larger the mean, the higher the variance.

Moreover, the means correspond to three different dominant application usage patterns: 10s reflects, e.g., dominant messaging application usage, 100s reflects, e.g., dominant mobile game usage, and 1000s reflects, e.g., dominant AR/VR or other constant or near-constant app usage. Indeed, Hintze et al. [46] measure comparable session durations on mobile phone and tablet.

For reallocation analysis, we split the enriched data to a training set we use for offline edge server placement, and a testing set we use for simulation of the online operation. The training–testing split is set to 0.8:0.2, respectively, along the temporal axis, with the larger training set comprising ca. 38M user connections between 1 January 2013 and 29 July 2014 and the testing set comprising ca. 9.5M connections between 29 July 2014 and 31 December 2014.

We select two edge server placement schemas for serving the user workload aggregated through the APs: 20 edge servers, roughly corresponding to MEC or edge–cloud, with a sparse deployment of high-capacity edge servers serving all users, and 150 edge servers, roughly corresponding to a dense Fog deployment with a large number of low-capacity servers. We employ the PACK placement method, based on capacitated clustering, and an R-based `rpack` software tool implementing the method [12]. We use the offline training data set to find the placement and allocation for the edge servers in both deployment

scenarios. The employed placement method interprets the clustering task as an optimization problem, minimizing an objective function comprising AP allocation and their distances from edge servers, with constraints for an upper and a lower limit for edge server capacity. Each AP is assigned a weight in relation to its maximum workload within the training set time period. The resulting ES placement and allocation for 20 edge servers and a user task duration mean of 100s are shown as an example in Fig. 3.

For spatio-temporal analysis, we further aggregate the connection data to hourly workload maximums and include information on the weekday, the season, the availability daylight, and the possible occurrence of rainfall, rush hour, and a mass event for each hour (see Table 1). The resulting hourly workloads on all edge servers in the MEC and Fog schemas, with duration mean $1/\lambda = 1000s$, on the first month of the observation period are depicted in Fig. 4.

2.4 Reallocation analysis

For each combination of duration mean (10s, 100s, 1000s) and edge deployment schema (20 ESs, 150 ESs), totalling 6 scenarios, we run a number of simulations. Each simulation assumes the ESs all have identical capacity, and that capacity is increased for each subsequent simulation. Further, for each scenario and each capacity, we simulate all of the strategies, namely cloud, proximity, bottom-up, and random. Further, we set the cloud strategy as the fallback for the edge strategies (proximity, random, and bottom-up), in case all of the ESs are over capacity. For all of those strategies, we analyse the resulting workloads on the ESs, the number and workload of cloud offloads and reallocations, as well as the relative latency in the ES network.

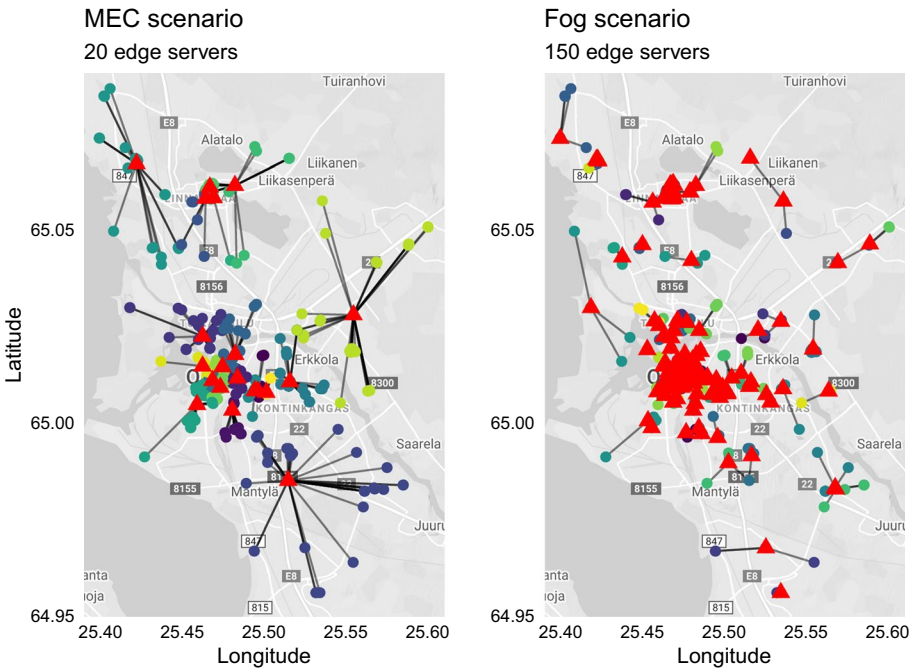

Fig. 3 Location of edge servers and the allocation of APs to edge servers

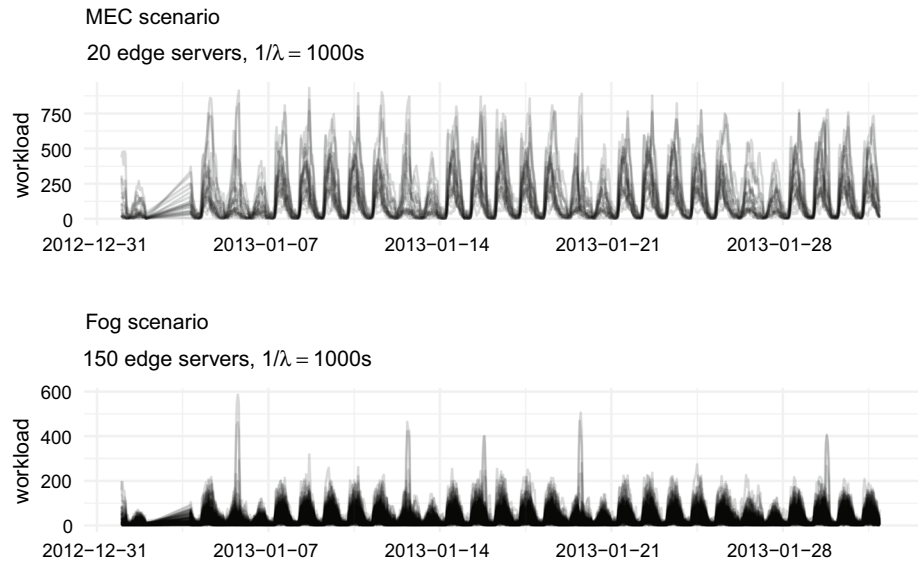

Fig. 4 Workload on edge servers for MEC and Fog, with duration mean 1000s, during first weeks of the observation period 2013–2014. A service break is clearly visible in the first week

Table 1 Independent categorical variables, augmented in the data for each hour

Variable	Values	Notes
Rain	Rain, no rain	Source: Linnanmaa weather station [47].
Season	Winter, spring, summer, fall	Winter in Oulu, Finland, in months 11–12 and 1–3, spring 4-5, summer 6–8, fall 9–10.
Rush hour	No, morning, afternoon	Morning: weekdays at 7–8; afternoon: Monday–Thursday 16–17, Friday 15–16.
Daylight	Dark, light	Sun above horizon, according to the R package `suncalc` [48].
Mass event	Event, no event	Occurrence of an ice hockey game (source: [49]).

The difference between the number of cloud offloads for the cloud strategy and the sum of the number of cloud offloads and the reallocations for the edge strategies (proximity, bottom-up, random) gives us the number of superfluous reallocations:

$$P_E = R_E + O_E - O_C, \tag{1}$$

where P is the number of superfluous reallocations, R is the number of reallocations, and O is the number of cloud offloads, while the subscript E designates an edge strategy and C the cloud strategy.

Indeed, since the cloud offloads in the cloud strategy affect only the offloading ES, they give a baseline of the times when that ES was over capacity due to vertical workload from its allocated APs. Since reallocation transfers workload horizontally from one ES to another, that extra workload on the target ES may cause there superfluous reallocations, which manifest as a number of reallocations exceeding the baseline. Further, the number of cloud allocations for the edge strategies, i.e. the times when an edge strategy had to revert to cloud offloading due to over capacity in all edge servers, is also included (Fig. 5). As proxies for latency in the ES network, we measure the distances between ESs along the great circles connecting them.

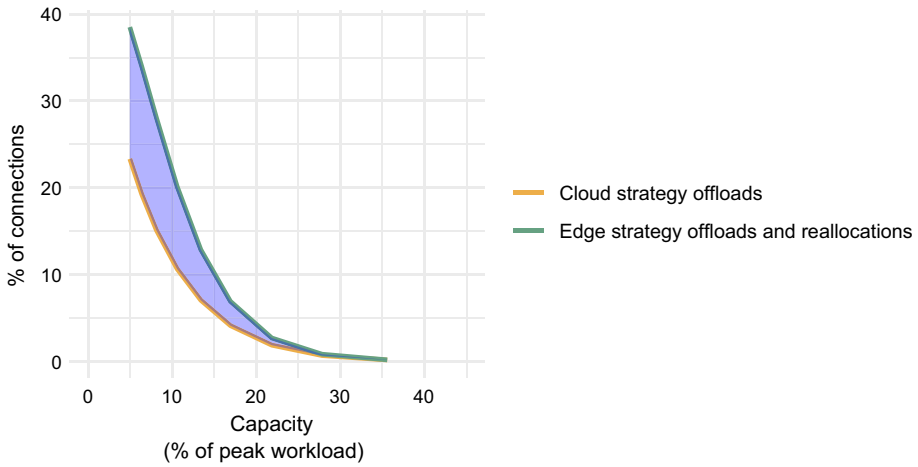

Fig. 5 Superfluous reallocations. The difference (blue area) between the number of offloads for cloud strategy (orange line) and the number of reallocations and offloads for an edge strategy (green line) is the number of superfluous reallocations, shown here as a proportion of all connections. (Figure originally in [10].)

2.5 Spatio-temporal analysis

Spatial dependency in the workloads of the edge servers means that the workloads of two servers, close to each other, are more alike than those of servers far removed. We measure spatial dependency with Moran's index (Moran's I) [50], defined as

$$I = \frac{n}{\sum_i^n \sum_j^n w_{ij}} \frac{\sum_i^n \sum_j^n w_{ij}(x_i - \bar{x})(x_j - \bar{x})}{\sum_i^n (x_i - \bar{x})^2},$$

where x_i and x_j are the observed response at locations i and j, $\bar{x}$ is the mean of observations, and w_{ij} is the spatial weight between locations i, j, while n is the total number of observations. We use the maximum workload during an hour as the response x, reflecting the dependence of the peak loads on the edge servers. Moreover, we use the inverse of the geodesic distance between two locations as the spatial weight. This assumes spatial dependency decays in direct proportion to the distance between the edge nodes, which is a reasonable assumption as the workloads on edge servers are point-based.

We focus on the $1/\lambda = 1000$s duration mean with both deployment schemas (MEC and Fog) for the spatio-temporal analysis, totalling in two scenarios for spatio-temporal analysis. We assume here that with the duration mean of 1000s for analysis (reflecting for example dominant AR/VR usage), there is a higher possibility of spatio-temporal dependency due to the long connection times, thus justifying the focus.

We first compare the association of spatial dependency with the workload of the edge servers. For each hour in the data, we take the minimum (min), the maximum (max), the mean and the median, as well as quantiles 10%, 20%, ⋯ 90% across the observations, plus additional 95% and 99% quantiles at the very highest end. We compare these statistics, in turn, to the significance of the Moran's I value (i.e. whether $p < 0.05$ is true or false) during that hour, calculated with x as the maximum workload on each edge server. To quantify the association, we fit a logistic regression model for each of the workload statistic, with

$$\log \frac{q_t}{1-q_t} = \alpha_s + \beta_s w_{s,t},$$

where t corresponds to the index of the time point and s that of the workload quantile, $q_t = \Pr(Y_t = 1)$ is a probability, with $Y_t \in \{0, 1\}$ the significance of the Moran's I estimate, one value for each hour during the observation period. $w_{s,t}$ stands for one of the workload statistics for that hour. Further, α is the intercept, and $\beta_{w,s}$ is the regression coefficient for the workload statistic. We call these models the *workload models*.

A logistic regression model measures the effect of the independent variables on the logarithm of the odds ratio of spatial dependency. Accordingly, to get the effect of one unit increase in workload w on the odds ratio of spatial dependency vs. no spatial dependency, we must thus exponentiate the corresponding coefficient.

The difference in workload model goodness-of-fit values (measured with AIC and BIC [51]) provides us information on whether the spatial dependency is particularly dependent on the highest (max) or lowest (min) workload of all edge servers, the mean or the median, or some particular quantile.

Furthermore, we explore the association of spatial dependency with a number of independent variables in addition to workload x_t. We thus fit a multiple logistic regression model between the significance of the Moran's I estimate of each hour in the data set and the variables in Table 1, each with its own regression parameter.

These additional independent variables include, in addition to the momentary workload statistic in the edge server network, availability of daylight, the current day of week, season, rush hour, rainy weather, and the occurrence of a large sports event (namely a game of the local ice hockey team *Oulun Kärpät*, a popular event in Oulu). We refer this model as the *variable model*. With the variable models, we thin the hourly data set to roughly a third, taking a random sample of 5000 rows, and also including all rows where a mass event was present, to reduce the effect of temporal autocovariance.

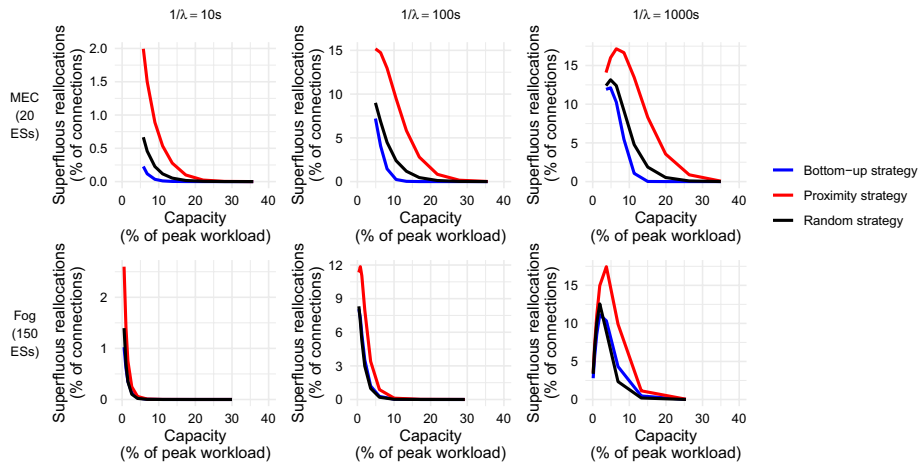

Fig. 6 Superfluous reallocations in each scenario. Proximity strategy (red) produces the severest reallocation storm in each scenario, while bottom-up (blue) and random (black) strategies result in much lower numbers of superfluous reallocations. (Figure originally in [10].)

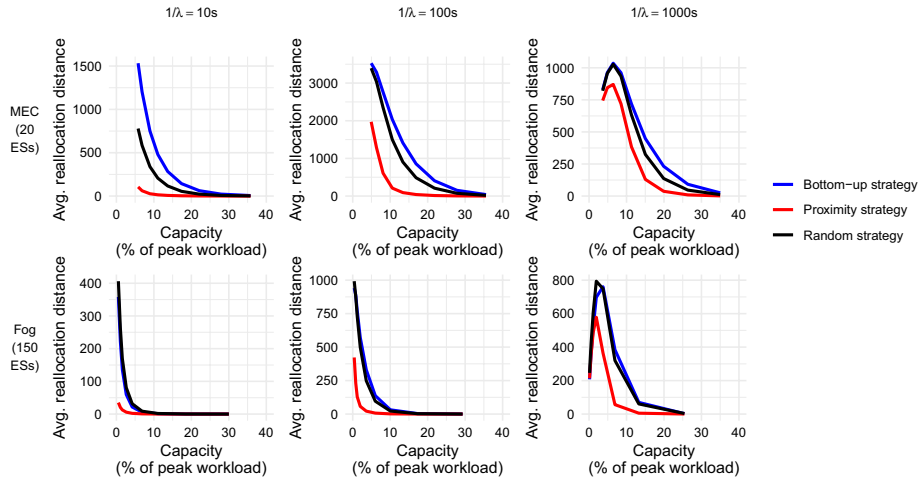

Fig. 7 Average reallocation distances in each scenario. Proximity strategy (red) results, as expected, in the shortest distances in each scenario, while bottom-up (blue) and random (black) strategies result in somewhat higher distances. (Figure originally in [10].)

3 Results

3.1 Reallocation analysis

Results are depicted in Figs. 6 and 7. The superfluous reallocations for proximity strategy (in red) peak at ca. 12–17% of all connections for durations with mean 100 and 1000 seconds, and ca. 2% and 3% of connections for 20 and 150 edge servers for task durations with mean 10, respectively. Bottom-up (blue) and random (black) strategies consistently result in a lower number of superfluous reallocations.

The higher the mean of the task duration, the wider the storm. For the proximity strategy, the capacity where the reallocation storm ends, that is, where the proportion of superfluous reallocations drops below 0.1% of all connections, peaks at ca. 35% of top ES workload for task duration means 100s and 1000s, with 20 ESs.

The effect of the number of edge servers is inverse. The more the edge servers, that is, the denser the ES deployment, the narrower the storm, with the proportion of superfluous reallocations always lower with 150 ESs than with 20 ESs.

On the other hand, average reallocation distances (Fig. 7), as a proxy to communication latency in the network connecting the ESs, show that the proximity strategy results in consistently shorter distances than the other two strategies.

3.2 Spatio-temporal analysis

Out of all hours in the observation period, ca. a quarter of the hours (25% with MEC and 28% with Fog) indicated significant spatial dependency, with the p value of the Moran's I below 0.05. These episodes of spatial dependency may take several hours, with the medians at 7 (MEC) and 14 (Fog) hours (Fig. 8).

Results of the workload models are collected in Table 2, with the goodness of fit further plotted in Fig. 9. All workload statistics, with the exception of the model using workload minimum as an independent variable, produce positive coefficients with significant estimates: as workload grows, by and large, odds of spatial dependency also grow. However, the 95% quantile for MEC and the 99% quantile for Fog (both corresponding, usually, to

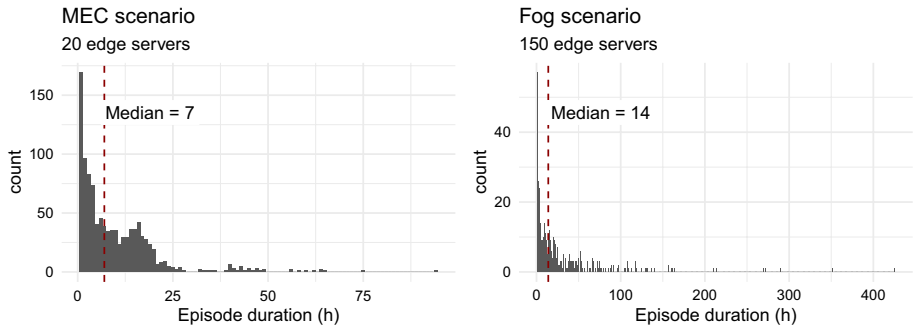

Fig. 8 Duration of episodes of spatial dependency

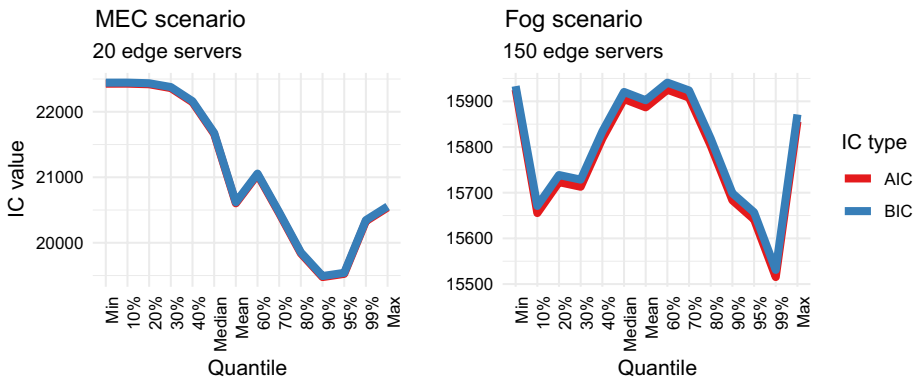

Fig. 9 Workload model goodness-of-fit measures

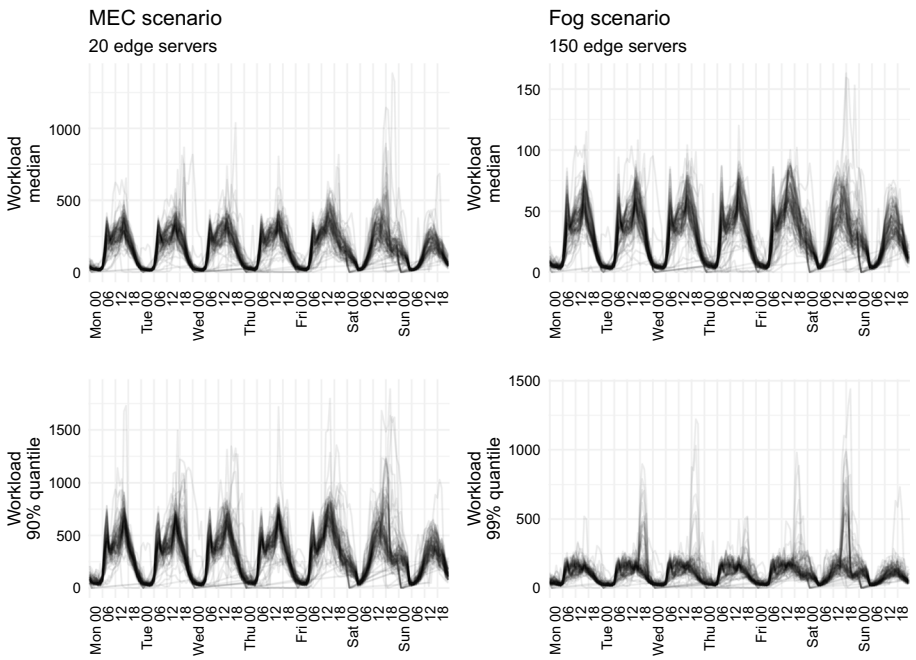

Fig. 10 Workload medians and 95% quantiles across all edge servers, during each week of the observation period 2013–2014

Table 2 Coefficients for the workload models

Scenario	Min	30% quantile	Mean	Median	70% quantile	95% quantile	99% quantile	Max
MEC	− 0.003	0.007***	0.024***	0.018***	0.019***	0.011***	0.009***	0.008***
Fog	NA	− 0.163***	0.264***	0.027*	0.096***	0.109***	0.075***	0.013***

$+p < 0.1$, $*p < 0.05$, $**p < 0.01$, $***p < 0.001$

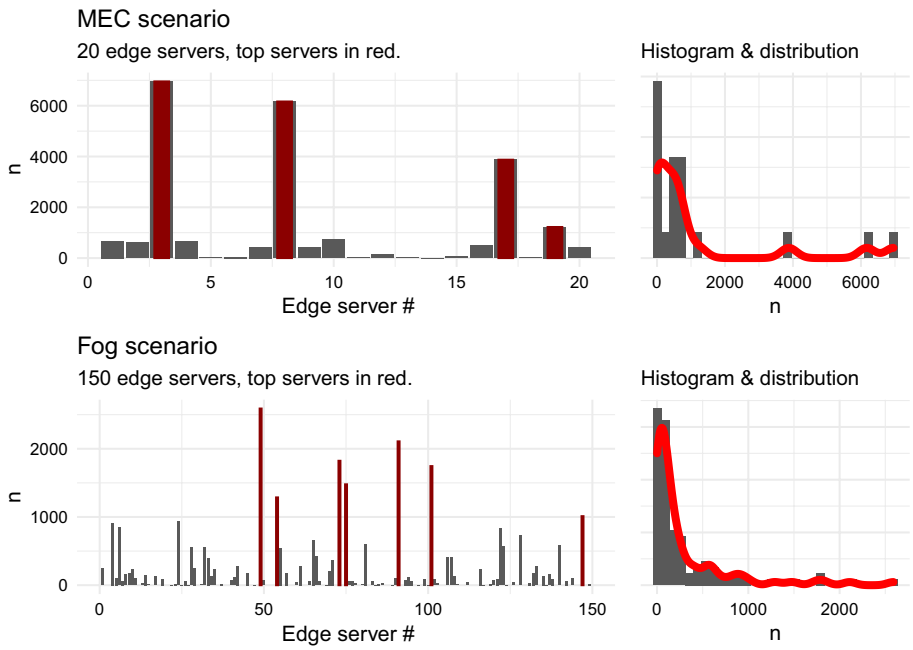

Fig. 11 Edge servers with high load during episodes of high spatial dependency. On the left, the number of such occurrences for each edge server. On the right, the histogram and distribution of such occurrences

just 1 or 2 servers) produce the best goodness-of-fit measures with both AIC and BIC. As depicted in Fig. 10, those quantiles have much more variation than the median in both scenarios, and according to the goodness-of-fit criteria, this high variation better explains the level of spatial dependency occurring in the network.

Figure 11 shows, on the left panel, the number of times each edge server experiences occurrences of high workload (i.e. above the designated quantile) during episodes of significant spatial dependency, and on the right panel, the histograms and distributions of such occurrences. In both scenarios, a handful of servers feature much more prominently than the others. Figure 12 shows the location and frequency of such high load edge servers during episodes of significant spatial dependency. Indeed, in both scenarios, the areas shown are in the city centre and nearby sports arena, both highly likely to experience the highest workloads. The results indicate that, during episodes of spatial dependency, other servers around these few are also subject to relatively high load.

The variable models indicate the importance of the independent variables (Table 3, Fig. 13) on spatial dependence. Comparing the variable model to a baseline model, which includes only workload, the variable models have indeed better value on both goodness-of-fit criteria (AIC, BIC). Accordingly, in the MEC scenario, a normal working day (Tuesday–Friday) reduces the odds of spatial dependency to ca. 0.5–0.6

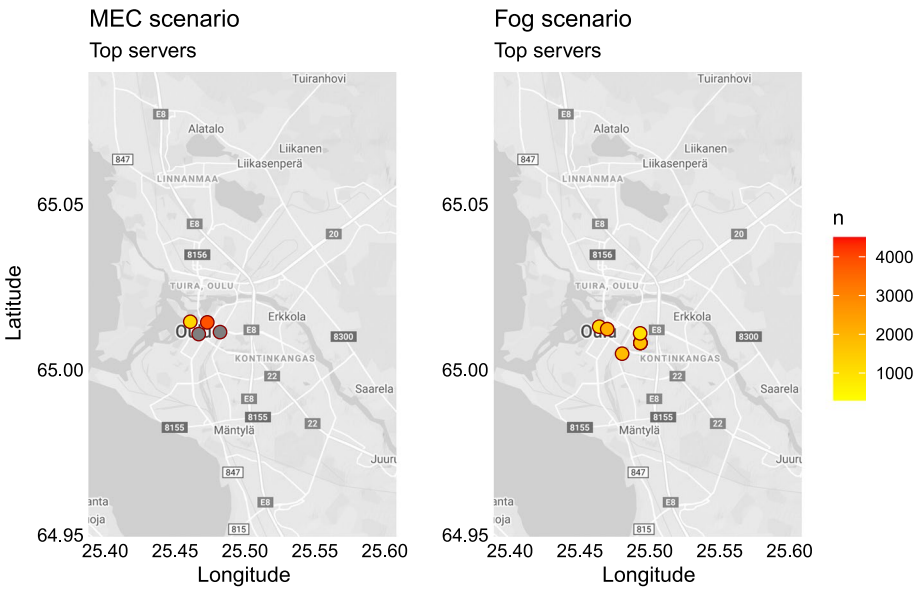

Fig. 12 Edge servers with high load during high spatial dependency episodes. The more often a server occurs in such an occasion, the more red its dot

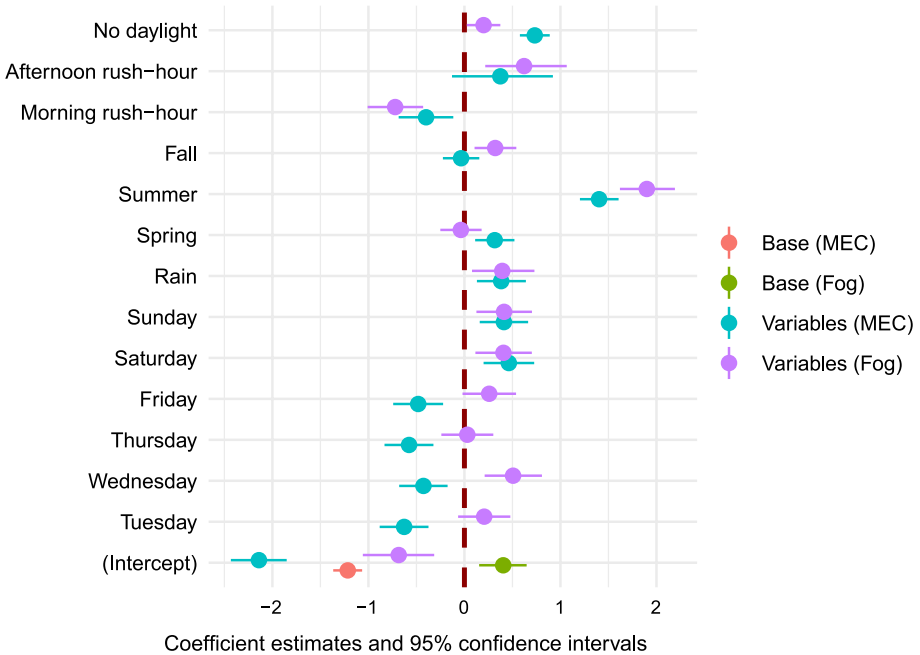

Fig. 13 Variable model coefficients (on the log scale) with confidence intervals

times as likely as on average, on a Monday, in winter, in daylight, with no rain, no rush hour, and no mass event. Also, morning rush hour reduces the risk (0.7 times as likely), while summer (4.1 times as likely) and springtime (1.4), weekend (Saturday 1.6, Sunday 1.5), and the absence of daylight (2.1) increase it. For Fog, morning rush

Table 3 Exponentiated coefficients and goodness-of-fit measures for the variable models

	Base (MEC)	Base (Fog)	Variables (MEC)	Variables (Fog)
(Intercept)	0.30***	1.50**	0.12***	0.50***
Workload, high quantile	1.01***	1.01***	1.01***	1.01***
Tuesday			0.53***	1.23
Wednesday			0.65***	1.66***
Thursday			0.56***	1.03
Friday			0.62***	1.29+
Saturday			1.59***	1.50**
Sunday			1.51**	1.51**
Rain			1.47**	1.48*
Spring			1.37**	0.96
Summer			4.07***	6.68***
Fall			0.97	1.38**
Morning rush hour			0.67**	0.49***
Afternoon rush hour			1.45	1.86**
No daylight			2.08***	1.22*
Mass event			9.65***	3264805.95
Num.Obs.	5168	5168	5168	5168
AIC	5541.9	4427.9	5101.8	4078.8
BIC	5555.0	4441.0	5206.6	4183.6

$+ p < 0.1$, $*p < 0.05$, $**p < 0.01$, $***p < 0.001$

hour (0.5 times as likely) reduces the odds, while summertime (6.7), afternoon rush hour (1.9), weekend (1.5), and the absence of daylight (1.2) again increase the risk of spatial dependency. Further, the effect of a mass event is enormous (9.7) in the MEC scenario, but, while positive, is not significant in the Fog scenario.

4 Discussion

4.1 Reallocation analysis

Reallocation storms are associated with the spatial dependency of the user workload submitted on edge servers. Indeed, results show that reallocating workload to spatially nearby edge servers greatly exacerbates the storm. This follows the very definition of spatial dependency, as in Tobler's first law of geography: "everything is related to everything else, but near things are more related than distant things" [52]. While the actual onset of a reallocation storm depends heavily on the capacity of the edge servers, spatial dependency thus increases the risk of storms. The factors affecting spatial dependency are further discussed in the following section.

Clearly, the proportion of superfluous reallocations is high, indicating a reallocation storm, when edge servers have low capacity. Indeed, a lower capacity is exceeded more quickly, resulting in more superfluous reallocations. However, at very low capacity values, we see the number of superfluous reallocations suddenly dropping in many scenarios. This is likely the result of the number of regular (i.e. not superfluous) reallocations growing so high that they start to dominate over the superfluous ones.

Further, superfluous reallocations are caused by user workload that is reallocated to a server, which exceeds its capacity within the duration of the task. Such an event is less likely to happen when there are more ESs around.

Interestingly, with a high number of ESs, random strategy and bottom-up strategy accumulate nearly identical reallocation distances. Since they also all but agree on the number of superfluous reallocations, the results suggest employing the random strategy instead of the bottom-up strategy in dense ES deployments due to its lower communication overhead and simpler decentralization. In sparser deployments, and with short average task duration, the bottom-up strategy may, however, provide a slight edge over the random strategy in terms of the number of superfluous reallocations, at the cost of some additional latency.

4.2 Spatio-temporal analysis

Results indicate that spatial dependency, and thus reallocation storms, is best associated with the workload on a small number of edge servers with the highest workloads. In other words, reallocation storms do not require an overall high load on all servers (as indicated by the min statistic), on half of them (median), or on average (mean). Instead, only a few ESs with a high load at each given moment—1 in the MEC scenario, and 8 in the Fog scenario—are associated with the triggering of the storm.

The results identified a number of conditions increasing the risk of a reallocation storm. The largest of these were weekends, summer, lack of daylight, and afternoon rush hour. Barring lack of daylight, these suggest increased mobility of users, transferring workload between spatially close edge servers. Further, in the MEC scenario, a mass event increased the risk of spatial dependency substantially, likely the result of attendants amassing to and from the event venue. Unfortunately, in the Fog scenario, there was not enough evidence to support a similar result; more such events should be identified and labelled in the data set for further study.

Nevertheless, edge operators can watch the identified risky conditions, together with the workload of top servers, to mitigate or prevent the occurrence of reallocation storms, e.g. by switching to a random reallocation strategy.

4.3 Impact

In general, the value of these findings is threefold. First, avoiding the reallocation storm improves QoS/QoE and resource efficiency, as superfluous reallocations causing network burden and increasing latency are minimized. Second, edge operators have more information for selecting the most suitable reallocation strategy based on the requirements of their customers' business scenarios, as well as the changing conditions in the edge network. In addition to improved performance, the improvement in resource efficiency also contributes towards more sustainable future and green transition. Finally, the results point the path towards relevant future studies.

4.4 Limitations

While the study considered the Wi-Fi deployment of one geographical area, our earlier studies [5, 12] have shown the deployment is representative of an edge deployment spanning urban areas with a high AP density as well as suburban areas with a low AP density.

Furthermore, the study simplified edge server capacity to one scalar, exceeding which leads to ES unavailability. In reality, capacity is a more complex concept, with execution

slowing down with increasing workload. Particularly for edge applications where reallocation is a lightweight operation (i.e., reallocation does not require the transfer of application data from one ES to another), reallocation could be triggered well before the actual capacity of the ES is reached. However, this can be reflected in the study by employing a lower capacity value.

Moreover, we chose the Moran's I as the model for spatial dependency, with inverse geodesic distance as the spatial weight. This was convenient as the focus here was on analysing the occurrence and causes of spatial dependency. Other dependency and decay models, such as spatial and spatio-temporal variography with different covariance functions, may provide more precise information on the nature and structure of the dependency; we plan to pursue these studies in future works.

We focussed on long task durations (with the mean at 1000s) for the spatio-temporal analysis. The spatio-temporal results thus apply for applications with such durations; however, it can be argued that similar results could be obtained with shorter durations if the number of tasks were increased in proportion, or the capacity of the edge servers reduced.

Finally, this study focused on large edge networks with user-generated workload. This reflects, for example, in the prominence of the circadian rhythm in the results. While such underlying conditions are not present in many edge deployments serving IoT sensor networks, such deployments may exhibit other forms of spatio-temporal dependencies which cause similar reallocation episodes.

5 Conclusions

We employed a real-world large-scale Wi-Fi connection data set for the simulation of workload in a number of edge computing scenarios. While the study considered the Wi-Fi deployment of one geographical area, earlier studies [5, 12] have shown that the deployment is representative of an edge deployment spanning urban areas with a high AP density as well as suburban areas with a low AP density. Further, while the studied period was as early as 2013–2014, we augmented the data to accommodate potential future edge use cases with usage patterns ranging from predominantly short to predominantly long, sampling the actual task execution times and workloads from realistic distributions.

We studied four distinct strategies, namely the cloud, proximity, bottom-up, and random strategies, for reallocating workload when edge servers exceed their capacity. We discovered a reallocation storm with a large number of superfluous reallocations, triggered when a task is reallocated to an ES whose capacity is exceeded within the duration of the task.

The proximity strategy, aiming for low latency, resulted in the highest number of superfluous reallocations in all conditions. However, the superfluous reallocations vanished when capacity was increased above a certain threshold, unique for each strategy. Moreover, we discovered that the few edge servers with the very highest workloads are in both scenarios best associated with reallocation storms.

Finally, we identified circumstances associated with an elevated risk of reallocation storms. For example, spatial dependency is high in summertime, and on weekends. Mass

events such as popular sports games incurred a high risk of a reallocation storm in the MEC scenario.

As future work, we plan to study in more detail the nature of the spatial dependency, as well as find refined methods for avoiding the storm, while optimizing ES capacity by means of novel reallocation strategies.

Abbreviations

AP	Access point
ES	Edge server
MEC	Multi-access edge computing

Acknowledgements
The authors would like to acknowledge Tero Lähderanta, Ilkka Launonen, and Mikko Sillanpää from the Research Unit of Mathematical Sciences at the University of Oulu, Finland, for their valuable advice with the spatio-temporal analytics methods used in this study.

Authors Contributions
LL designed the experiments and simulations, analysed and interpreted the results, and directed the writing of the manuscript. EP, EH, and SP were major contributors to writing the manuscript. LR contributed substantially to the statistical analysis methods and the interpretation of the results. All authors read and approved the final manuscript.

Funding
This research is supported by Academy of Finland 6Genesis Flagship and DigiHealth programs (grants 318927, 326291); the ECSEL JU FRACTAL (grant 877056), receiving support from the EU Horizon 2020 programme and Spain, Italy, Austria, Germany, France, Finland, Switzerland; and the Infotech Oulu research institute.

Availability of data and materials
The data set analysed during the current study is not publicly available due to the high privacy requirements of Wi-Fi user connection logs.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 8 November 2021 Accepted: 1 September 2022
Published online: 15 September 2022

References

1. W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges. IEEE Internet Things J. **3**(5), 637–646 (2016)
2. W. Shi, S. Dustdar, The promise of edge computing. Computer **49**(5), 78–81 (2016)
3. E. Peltonen, L. Lovén, *et al.* 6G White Paper on Edge Intelligence, pp. 1–27. 6G Flagship, University of Oulu, Oulu, Finland (2020)
4. S.P. Singh, A. Nayyar, R. Kumar, A. Sharma, Fog computing: from architecture to edge computing and big data processing. J. Supercomput. **75**(4), 2070–2105 (2019)
5. L. Lovén, T. Lähderanta, L. Ruha, T. Leppänen, E. Peltonen, J. Riekki, M.J. Sillanpää, Scaling up an edge server deployment. In: IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 1–7. IEEE, Austin, TX, US (2020)
6. L. Lovén, T. Lähderanta, L. Ruha, E. Peltonen, I. Launonen, M.J. Sillanpää, J. Riekki, S. Pirttikangas, EDISON: an edge-native method and architecture for distributed interpolation. Sensors **21**(7), 1–20 (2021). https://doi.org/10.3390/s21072279
7. Y. Dai, D. Xu, S. Maharjan, Y. Zhang, Joint load balancing and offloading in vehicular edge computing and networks. IEEE Internet Things J. **6**(3), 4377–4387 (2018)
8. E.J. Ghomi, A.M. Rahmani, N.N. Qader, Load-balancing algorithms in cloud computing: a survey. J. Netw. Comput. Appl. **88**, 50–71 (2017)
9. K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, J. Al-Jaroodi, A survey of load balancing in cloud computing: Challenges and algorithms. In: 2012 second symposium on network cloud computing and applications, pp. 137–142 (2012). IEEE
10. L. Lovén, E. Peltonen, E. Harjula, S. Pirttikangas, Weathering the reallocation storm: Large-scale analysis of edge server workload. In: 2021 Joint EuCNC & 6G Summit, pp. 1–6. IEEE, Virtual (Porto, Portugal) (2021)
11. J. Qadir et al., Towards mobile edge computing: taxonomy, challenges, applications and future realms. IEEE Access **8**(October), 189129–189162 (2020). https://doi.org/10.1109/ACCESS.2020.3026938
12. T. Lähderanta, L. Lovén, T. Leppänen, L. Ruha, E. Harjula, M. Ylianttila, J. Riekki, M.J. Sillanpää, Edge computing server placement with capacitated location allocation. J. Parallel Distributed Comput. **153**, 130–149 (2021)

13. A. Yousefpour et al., All one needs to know about fog computing and related edge computing paradigms: a complete survey. J. Syst. Archit. **98**, 289–330 (2019)
14. T. Ouyang, Z. Zhou, X. Chen, Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing. IEEE J. Selected Areas Commun. **36**(10), 2333–2345 (2018)
15. T. Taleb et al., Mobile edge computing potential in making cities smarter. IEEE Commun. Magaz. **55**(3), 38–43 (2017). https://doi.org/10.1109/MCOM.2017.1600249CM
16. K. Bhardwaj, et al. Fast, scalable and secure onloading of edge functions using Airbox. IEEE/ACM Symposium on Edge Computing (2016). doihttps://doi.org/10.1109/SEC.2016.15
17. Y. Caniou, G. Charrier, F. Desprez, Analysis of tasks reallocation in a dedicated Grid environment. In: IEEE Int. Conf. on Cluster Computing (ICCC), pp. 284–291. IEEE, Heraklion, Crete, Greece (2010). https://doi.org/10.1109/CLUSTER.2010.39
18. Y. Zhang, C. Pang, G. Yang, A real-time computation task reconfiguration mechanism for industrial edge computing. In: The Annual Conf. of the IEEE Industrial Electronics Society (IECON), pp. 3799–3804. IEEE, Singapore (2020). https://doi.org/10.1109/IECON43393.2020.9255395
19. Group Report: GR MEC 031 - V2.1.1 - Multi-access Edge Computing (MEC) MEC 5G Integration. Technical report, ETSI (2020)
20. Y. Miao et al., Intelligent task prediction and computation offloading based on mobile-edge cloud computing. Fut. Gener. Comput. Syst. **102**, 925–931 (2020). https://doi.org/10.1016/j.future.2019.09.035
21. C. Wu, Y. Zhang, Y. Deng, Toward fast and distributed computation migration system for edge computing in IoT. IEEE Internet Things J. **6**(6), 10041–10052 (2019). https://doi.org/10.1109/JIOT.2019.2935120
22. W. Chang et al., An offloading scheme leveraging on neighboring node resources for edge computing over fiber-wireless (FiWi) access networks. China Commun. **6**(11), 107–119 (2019). https://doi.org/10.23919/JCC.2019.11.009
23. Q. Fan, N. Ansari, Application aware workload allocation for edge computing-based IoT. IEEE Internet Things J. **5**(3), 2146–2153 (2018). https://doi.org/10.1109/JIOT.2018.2826006
24. D. Puthal, M.S. Obaidat, P. Nanda, M. Prasad, S.P. Mohanty, A.Y. Zomaya, Secure and sustainable load balancing of edge data centers in fog computing. IEEE Commun. Mag. **56**(5), 60–65 (2018). https://doi.org/10.1109/MCOM.2018.1700795
25. M. Satyanarayanan, The emergence of edge computing. Computer **50**(1), 30–39 (2017)
26. W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, X. Yang, A survey on the edge computing for the internet of things. IEEE Access **6**, 6900–6919 (2017)
27. H. Li, K. Ota, M. Dong, Learning iot in edge: deep learning for the internet of things with edge computing. IEEE Netw. **32**(1), 96–101 (2018)
28. J. Chen, S. Chen, S. Luo, Q. Wang, B. Cao, X. Li, An intelligent task offloading algorithm (itoa) for uav edge computing network. Digital Commun. Netw. **6**(4), 433–443 (2020). https://doi.org/10.1016/j.dcan.2020.04.008
29. T. Braud, P. Zhou, J. Kangasharju, P. Hui, Multipath computation offloading for mobile augmented reality. IEEE Int. Conf. on Pervasive Computing and Communications (2020)
30. J. Lin, W. Yu, X. Yang, P. Zhao, H. Zhang, W. Zhao, An edge computing based public vehicle system for smart transportation. IEEE Trans. Veh. Technol. **69**(11), 12635–12651 (2020)
31. K. Zhang, *et al.*: Optimal delay constrained offloading for vehicular edge computing networks. In: IEEE Int. Conf. on Communications (ICC), pp. 1–6 (2017). IEEE
32. P. Pace et al., An edge-based architecture to support efficient applications for healthcare industry 4.0. IEEE Trans. Indus. Inf. **15**(1), 481–489 (2019)
33. L. Lovén, T. Leppänen, E. Peltonen, *et al.* EdgeAI: A vision for distributed, edge-native artificial intelligence in future 6G networks. In: The 1st 6G Wireless Summit, Levi, Finland, pp. 1–2 (2019)
34. T.C. Chieu, A. Mohindra, A.A. Karve, A, Segal, Dynamic scaling of web applications in a virtualized cloud computing environment. In: 2009 IEEE International Conference on e-Business Engineering, pp. 281–286 (2009). https://doi.org/10.1109/ICEBE.2009.45
35. A. Rahman, X. Liu, F. Kong, A survey on geographic load balancing based data center power management in the smart grid environment. IEEE Commun. Surv. Tutorials **16**(1), 214–233 (2014). https://doi.org/10.1109/SURV.2013.070813.00183
36. L. Ruha, T. Lähderanta, L. Lovén, T. Leppänen, J. Riekki, M.J. Sillanpää, Capacitated spatial clustering with multiple constraints and attributes. arXiv preprint arXiv:2010.0633v3 (2021). arXiv:2010.06333v3
37. X. Li, A computing offloading resource allocation scheme using deep reinforcement learning in mobile edge computing systems. J Grid Comput. (2021). https://doi.org/10.1007/s10723-021-09568-w
38. J. Edinger, M. Breitbach, N. Gabrisch, D. Schafer, C. Becker, A. Rizk, Decentralized low-latency task scheduling for Ad-Hoc computing. Proceedings - 2021 IEEE 35th International Parallel and Distributed Processing Symposium, IPDPS 2021, (2021). https://doi.org/10.1109/IPDPS49936.2021.00087
39. H. Liu, S. Li, W. Sun, Resource allocation for edge computing without using cloud center in smart home environment: a pricing approach. Sensors (Basel) **20**(22), 6545 (2020). https://doi.org/10.3390/s20226545
40. X. Niu et al., Workload allocation mechanism for minimum service delay in edge computing-based power internet of things. IEEE Access **7**, 83771–83784 (2019)
41. S. Wang et al., A machine learning approach for task and resource allocation in mobile edge computing based networks. IEEE Internet Things J. (2020). https://doi.org/10.1109/JIOT.2020.3011286
42. Y. Chen, J.P. Walters, S.P. Crago, Load balancing for minimizing deadline misses and total runtime for connected car systems in fog computing. In: IEEE Int. Symp. on Parallel and Distributed Processing with Applications and IEEE Int. Conf. on Ubiquitous Computing and Communications (ISPA/IUCC), pp. 683–690 (2017). https://doi.org/10.1109/ISPA/IUCC.2017.00107
43. V. Kostakos, T. Ojala, T. Juntunen, Traffic in the smart city. Internet Comput. IEEE **17**(6), 22–29 (2013)
44. J. Lisman, M. Van Zuylen et al., Note on the generation of most probable frequency distributions. Stat. Neerlandica **26**(1), 19–23 (1972)

45. S.Y. Park, A.K. Bera, Maximum entropy autoregressive conditional heteroskedasticity model. J. Econ. **150**(2), 219–230 (2009)
46. D. Hintze, P. Hintze, R.D. Findling, R. Mayrhofer, A large-scale, long-term analysis of mobile device usage characteristics. ACM Interactive Mobile Wearable Ubiquitous Technol. **1**(2), 1–21 (2017). https://doi.org/10.1145/3090078
47. VTT: Linnanmaa weather station. http://weather.willab.fi Accessed 2021-11-04
48. B. Thieurmel, A. Elmarhraoui, Suncalc: Compute sun position, sunlight phases, moon position and lunar phase. (2019). R package version 0.5.0. https://CRAN.R-project.org/package=suncalc
49. Kärpät games 2013-2014. https://fi.wikipedia.org/wiki/Oulun_Kärppien_SM-liigakausi_2013-2014 Accessed 2021-11-04
50. P. Moran, Notes on Continuous Stochastic Phenomena Published by : Biometrika trust stable. Biometrika **37**(1), 17–23 (1950)
51. K.P. Burnham, D.R. Anderson, Multimodel inference: understanding AIC and BIC in model selection. Soc. Methods Res. **33**(2), 261–304 (2004). https://doi.org/10.1177/0049124104268644
52. W.R. Tobler, A computer movie simulating urban growth in the Detroit region. Econ. Geogr. **46**(Supplement), 234–240 (1970)

Publisher's Note