

Benchmarking in The Dark: On the Absence of Comprehensive Edge Datasets

Oleg Kolosov and Gala Yadgar
Computer Science Department
Technion

Sumit Maheshwari and Emina Soljanin
Department of Electrical and Computer Engineering
Rutgers University

Abstract

Application workloads are used for system analysis, design, optimization, and evaluation. As systems and their applications evolve, new workloads are required to reflect their characteristics and bottlenecks. Edge-computing systems are a new model with unique characteristics, including heterogeneous and geodistributed components and complex failure patterns. Their workload characteristics are also unique, including unpredictable load and user mobility, combined with stringent latency and bandwidth requirements. Supporting these workloads would require resource management and allocation policies that take into account a rich set of attributes.

In this paper, we show that currently available datasets include only partial subsets of these attributes. Moreover, having been collected on existing systems, they do not reflect the unique characteristics of edge systems and applications. At the same time, current edge systems in early deployment stages are not ready for the collection of representative application workloads. We show how to bridge the gap between required and available datasets by *workload composition*: we combine attributes from several available datasets to create realistic representations of edge systems, their users, and their workloads.

1 Introduction

Application workloads play an important role in systems research, design, and optimization. Characterizing these workloads helps define system design objectives, identify optimization goals, and make appropriate tradeoffs [45, 52, 60, 67, 74, 84, 85, 95, 101, 106]. Examples include storage caching and tiering designed according to data access frequencies [100, 108], scheduling and resource allocations based on job sizes, interdependencies, and latency sensitivity [50, 54, 70, 82, 89, 96], and consistency mechanisms chosen according to the degree of data sharing and the ratio between reads and writes [36, 48]. Workload traces are further used to evaluate the resulting systems with respect to alternative designs. Industry-standard benchmarks, such as YCSB [43], TPC [9], and SPEC [26], are based on representative workloads and are often used as a synthetic substitute for real application traces.

New edge computing systems promise to meet the stringent latency requirements of emerging real-time applications, such as augmented and virtual reality [55, 56], by bringing compute, storage and networking resources closer to user devices [46]. Thus, the edge infrastructure is a collection of interoperating

edge nodes, each of limited size and resources, located one or two network hops away from the end user [38, 41, 71, 88].

Edge systems represent a new, arguably disruptive, system model. They are inherently different from existing systems, such as cloud datacenters or geodistributed systems, in two major aspects [46, 102]. The first is their workloads: clouds aggregate service for large numbers of users, thus experiencing mostly predictable traffic volumes. In contrast, edge services are intrinsically local and are thus subject to significantly larger workload fluctuations due to correlated events and user mobility. The second aspect is the way their resources are managed. Clouds can optimize user-perceived service as well as their resource utilization via centralized routing and load balancing. Edge systems, however, are by definition distributed across multiple edge networks, possibly involving multiple service providers. They are associated with considerable heterogeneity in bandwidth and compute resources, for which the centralized resource management model of datacenters is not applicable.

Thus, to design, optimize, and evaluate resource allocation and management policies for edge systems, we need workloads that (1) reflect the unique characteristics of edge applications and users, and (2) incorporate the attributes required for optimizing their service. These attributes are a superset of the attributes used in various optimizations today. For example, within a data center, the optimal allocation of jobs to nodes is orthogonal to the physical location of their user. Similar allocation of user jobs to edge nodes will likely attempt to minimize latency by minimizing the physical distance between the node and the user. As another example, strategies for call handovers in a cellular network are based on user mobility and service level, as well as the available resources in the new cell. A similar handover of an edge-based session or function might also depend on the availability of the data or code, and require transferring non-negligible state and context information. In both examples, the edge-based policies rely on attributes that are not used in current policies for datacenters or even networks.

Unfortunately, application workloads that were collected on existing systems rarely reflect the combination of aspects required for optimizing the edge infrastructure, nor the low-latency and high bandwidth requirements of edge applications. At the same time, collecting edge-based workloads is currently impossible—there are no large-scale deployed and operational edge systems, and existing small-scale systems still do not host user-facing applications. Without relevant and available

workloads, current studies on the design and optimization of the edge infrastructure resort to basic synthetic distributions for important workload attributes [57, 64, 75, 99, 105]. Such limited evaluation settings inevitably limit the advancement of research and optimization of the edge infrastructure.

We here turn the light to this (hopefully temporary) gap between edge-systems research and realistic input from the field. We characterize the workload attributes required for informed research into four categories: *storage*, *compute*, *user/application*, and *geolocation*, and then describe respective categories for datasets describing the edge systems themselves: *architecture*, *availability*, and *business*. We demonstrate that attributes from these categories are individually included in various datasets, but no available dataset includes the combination of attributes required to represent edge application workloads.

Our approach for bridging the current gap is *workload composition*, in which existing datasets are joined to obtain attribute combinations that are otherwise unavailable. We describe such composition in detail to show the advantage of this approach compared to using entirely synthetic workloads: it allows us to model and evaluate edge systems and designs with workloads that are more representative of how users move and access data, and how real applications interact with systems. We are yet to learn how representative these behaviors are of future edge applications and users. In the meantime, this approach can serve as a temporary solution to “bootstrap” research in edge computing systems. By demonstrating the importance of real application workloads for the edge, we hope to also encourage collection and release of such workloads.

2 Dataset Attributes We Need

The first set of attributes refers to the information we are interested in with respect to the work done by the system. These attributes would ideally describe each request in the workload. We characterize them into four categories: *storage*, *compute*, *user/application*, and *geolocation*.

Storage-related attributes describe the way data is accessed by users and applications. Standard attributes are the timestamp, operation (PUT or GET), object ID, and possibly its size. These attributes help understand (as in previous studies) object popularities over time, if the system is read-heavy or write-heavy, the request rate and how it fluctuates according to diurnal or weekly patterns, etc. Previous studies addressing edge-based storage services obtained these attributes from various non-edge resources. In one example, the authors used a peer-to-peer workload, to which they added synthetic request types [59]. Others used entirely synthetic request distributions [47, 75].

Compute-related workloads describe jobs and their tasks, and are used to understand how resources should be consolidated, scheduled, and allocated. Their standard attributes include job submission time, its tasks, dependencies between them, and their resource requirements, such as cores, GPUs, FPGAs, and memory and storage size and bandwidth. Some workloads specify task deadlines that are used for setting scheduling

priorities. In the edge, this information could be used to determine whether a task can be offloaded to a neighboring node or to the cloud. Another useful attribute is whether a submitted task is a resubmission (retry) of a previously failed task, in which case its resource requirements depend on the success or failure of the original task. The need for compute-related workloads has become evident in several recent studies. One used synthetic values for evaluating node placement and resource allocation [73], while others used non-edge applications to evaluate their edge systems [57, 68].

Information linking requests to the **user and application** that issued them is important for aggregating requests into logical streams that may access shared data, code, and resources, while having different (possibly conflicting) objectives, service levels, and progress. The relevant attributes are user ID, application name or ID, and possibly the device type. The latter is relevant for applications such as web browsing that require different responses (e.g., image resolution) to requests originating from different devices. In the context of edge computing, user permissions might imply constraints on request forwarding and collocation. Many of these attributes were modeled by synthetic values and distributions in studies addressing the allocation of resources to users sharing the edge system [31, 57, 64, 73, 99].

In wide-area networks, **geolocation** information is used to map requests to available nearby resources, to identify opportunities for load-balancing, and to specify restrictions on possible routing and forwarding optimizations. The most important attribute is the location from which a request originated, which can be absolute (e.g., global coordinates), or relative (e.g., system-specific zones). The location is particularly valuable when it can be correlated to other attribute categories. For example, when combined with storage attributes, it can help understand how object popularities vary between locations, and when combined with compute related attributes it can help identify areas with high resource requirements. One recent study used the locations of edge clients to evaluate the benefit from their collaboration [58], and another relied on their location for the choice of service provider [104]. The first study used locations from Twitter, while the second simulated uniformly distributed clients.

The second set of attributes refers to datasets that describe the infrastructure of the system itself. They can be categorized into: *system architecture*, *availability*, and *business*.

The edge **system architecture** is defined by two main components: (1) the nodes, characterized by their locations and resources, and (2) the network, characterized by the properties of the connections between nodes and their peers, their users (the edge devices), and their cloud backend. These attributes are important for correlating user requests to nearby available resources and for optimizing and consolidating these requests via offloading and peer-to-peer collaboration. For example, when evaluating models for allocating resources and contracting with providers, recent studies generated synthetic

node locations, resources, and network topologies [73, 99], or used topologies of non-edge networks [104, 105].

As the **availability** of the system’s components (nodes and network connections) may change over time, we are interested in logs specifying their state at fixed time intervals, or logs of failure events. Ideally, we wish to distinguish between two main causes of unavailability: (1) excessive load, which is a result of the workload and the way the system was managed at the time of serving it. For example, high load in certain areas, combined with suboptimal scheduling decisions, may result in nodes becoming unresponsive. These failures might be avoided by an alternative resource management policy. (2) External failures such as power outages, node hardware problems, or network partitions. Alternative management of the edge infrastructure cannot avoid such failures, but it should minimize their effect. For example, recent approaches for ensuring service availability with nodes whose availability is limited used synthetic datasets of node availability and failures for their evaluation [35, 109].

Business-related attributes are important for identifying opportunities for sharing or consolidating resources, as well as respective constraints on such optimizations. Relevant attributes include the operator a node or a connection belongs to, and the business agreements which specify who is entitled to using its resources and at which service level. These agreements are important for identifying potential collaborative optimizations. To the best of our knowledge, these attributes were not considered in previous edge-related studies.

3 Available Datasets

The systems community has been using a wide range of datasets, each containing a different set of workload attributes or system characteristics. We describe representative available datasets from different domains (summarized in Table 1).

Storage workloads. The SNIA IOTTA [7] repository contains traces collected at various levels of the storage hierarchy. System-call traces include file-system operations and attributes, e.g., the inode number, the requested byte range, and whether the request hit or missed in the page cache [39]. Block-level traces include the device, byte range, and sometimes the file and process associated with the I/O request [1, 53, 62, 63, 77, 107]. The traces were collected in university servers, production clusters, and even smartphones. None of them contains location attributes, and only limited information is available regarding their users and applications. Datasets of file-system snapshots include file sizes and their organization into directories and logical volumes [8, 51, 74], but this information is not correlated to the way files are read and written by users.

Several studies address object popularity and size in large-scale systems [34, 60]. Although the datasets are not available, their distributions are used to generate synthetic workloads in simulations and benchmarks [9, 43]. Popularity can also be extracted from independent datasets of social networks graphs, collaborations, citations, links, and web accesses [27, 65].

Mobility datasets. Several available datasets provide location and mobility information. We distinguish between *location-centric* datasets, such as camera traffic counts [12] or entrance and exit information from subway stations or parking lots [16, 23], and *user-centric* datasets, such as logs of taxi and bike rides [14, 17, 18, 79], and location-based online social networks [65]. Most of these datasets include only user and geolocation attributes [15]. One exception is the Mobile Data Challenge (MDC) dataset, with smartphone location and activity of 182 individuals in the Lake Geneva region, collected between 2009 and 2011 [61]. This dataset includes most of the properties we are interested in. However, we are not aware of a similar, more recent, dataset, most likely due to the unusual effort involved in its collection.

Cluster workloads. Several large companies have released traces of jobs and VMs on their clusters. These traces include information about jobs and their tasks, the dependencies between them, their resource utilization, the machine they were running on, and sometimes also the user that deployed them [44, 91, 98]. Available HPC cluster traces are less detailed, presenting only aggregate job statistics [32]. These datasets do not include any geolocation information, and their storage-related attributes, such as I/O bandwidth and memory size, are insufficient for correlating jobs with their datasets.

System architecture. Some cluster traces also include information about the cluster nodes and their resources [44, 98]. This information can be correlated with the jobs scheduled to run on them, but these allocations are location independent, as they take place within the datacenter. Various datasets describe different aspects of network architectures, such as graphs of autonomous systems and P2P networks [65], as well as official datasets of hotspots and cellular towers [19, 61]. Some of these datasets include information about the nodes’ resources or ownership, but additional business or availability attributes are unavailable.

Node availability. There are two major sources of information regarding node availability in large-scale networks. The first is the Internet, whose connections are constantly monitored for latency, round-trip, and connectivity. Detailed as well as aggregate information is available in several public repositories [11, 25]. The second source is datacenters that maintain detailed information about their storage devices. For example, the Backblaze dataset includes detailed SMART statistics and failure information of hard drive [30]. Similar information was used to analyze drive failures at EMC [69] and Google [86, 87], but their datasets have not been made public.

Edge-related datasets. Smart cities are one of the “killer apps” of edge computing. Cities around the world are preparing or have already started to collect data and make it available through various platforms [12, 13, 16–21, 23, 29, 33, 79, 93]. Available datasets include some of the mobility and system architecture datasets described above, as well as datasets related to transportation infrastructure, air quality, small businesses, and many others. We describe how these datasets can be used

Datasets	Storage	Compute	User/App.	Location	Arch.	Avail.
Storage workloads: FIU [39] UMass [1] MSR [77] FIUC [62] Nexus [107] Syster17 [63] Slacker [53]	V	X	•	X	X	X
FS snapshots: ECMWF [51] UBC [74] FSL [8]	•	X	X	X	•	X
Object popularity: Facebook [34] SNAP [65] Alexa [27] YCSB [43] TPC [9]	V	X	X	X	X	X
Mobility: Austin [12, 23] NYC [16–19] SFO [79]	X	X	•	V	V	X
Cluster: BORG [98] Azure [44] Alibaba [91] LANL [32]	X	V	•	X	V	X
Network architecture: RIPE [25] CAIDA [11]	X	X	X	V	V	V
Device failures: Backblaze [30]	X	X	X	X	X	V

Table 1: Available datasets and their attributes. V – available, X – not available, • – partially available.

for edge-systems research in the next section.

Edge-related compute and storage attributes are more difficult to obtain. E.g, data collected by autonomous-vehicle sensors is available [28], but primarily intended to improve machine learning applications. The interesting attributes in the context of the infrastructure are the storage and compute requests such applications will eventually forward to the edge. Similarly, a recently introduced benchmark for autonomous vehicles [97] models the workload the devices on the vehicle have to serve, and cannot be directly applied to the infrastructure.

Deployed edge systems. Current commercial systems are deployed as an orchestration layer on top of existing cellular and network infrastructure [2–6]. Data from these systems is not publicly available. Several academic initiatives are deploying large-scale edge testbeds to facilitate related research and evaluation. Some are extensions of systems originally deployed for global-scale network and cloud-based research, such as PlanetLab [42] and CloudLab [83]. Others are designed with the edge as their primary research objective, either by deploying dedicated physical infrastructure [72, 78, 80, 81], by creating a virtual edge-layer on top of existing wireless and mobile infrastructure [24, 37, 40, 90], or both [10].

Current research and development efforts related to those prototypes focus on porting applications to the edge and on exporting edge resources to applications [22]. In the future, these testbeds could be used for collecting the workloads of edge applications. At the same time, workloads collected on commercial deployments would be valuable input for evaluations executed on these testbeds.

4 What We Can Do

To bridge the gap between available datasets and the edge workloads we need, we can join attributes from several available datasets. We refer to this method as *workload composition*. The resulting composite workload represents edge workloads better than any of its individual components. Workload composition is inevitably ad-hoc—the original workloads, their attributes, and the way they are composed are chosen according to the purpose for which the workload is created. Similar techniques were used in previous studies, but to a much smaller extent: those examples that we are aware of augment one available dataset with a synthetic distribution for an additional attribute. For example, [47] used a Zipf distribution for accesses to an

available image dataset, while [58] accelerated a Twitter feed trace to simulate real world behavior. To demonstrate the need for more complex compositions, we describe a representative use case of workload requirements and generation.

Consider the design of an edge-based container store or caching service [47, 76], and the evaluation of object-placement and eviction policies for such a service. Ideally, we would use a trace of client requests that includes storage as well as geolocation attributes, with some dataset describing the physical architecture of the edge system. In the absence of such a workload, we generate a composite workload from the following available datasets:

- NYC taxi zones [18]: 263 polygons partitioning the city into neighborhood-sized zones.
- NYC yellow taxi [17]: 112M records of taxi rides, including their start and end times and pick-up and drop-off zones.
- NYC hotspots [19]: detailed coordinates of 3319 hotspots throughout the city.
- Wikipedia [66]: 28.5M graph edges representing links from one Wikipedia page to another for 1.8M pages.

Workload generation. In our composite workload, the hotspots represent edge-node locations, passenger drop-off zones represent the location of the passengers when they issue their requests, and the Wikipedia links represent page popularity. We first join the taxi-zones and hotspots workloads, to attribute each hotspot to the taxi zone it is located in. We list the Wikipedia pages in descending order of the number of links pointing to them, thus ranking them by their popularity. We then generate a trace of object GET requests of the form $\langle \text{time}, \text{node}, \text{page} \rangle$ as follows.

Each taxi ride represents a ‘browsing session’ that starts at the drop-off time and takes place at the drop-off zone. We choose a random hotspot from the drop-off zone as the node receiving the request, node_h . The first page of the session, page_0 , is chosen at random with a probability proportional to its static page rank. With some probability p_{exit} (*exit rate*), the session ends. With probability $1 - p_{\text{exit}}$ the session continues, and the next page is chosen from the pages pointed to by the current page, with a probability proportional to its rank. The session further continues with probability $1 - p_{\text{exit}}$, with additional pages chosen in the same manner. For a session with n pages starting at time drop-off T , our trace will include n requests of the form $\langle T + i \times \epsilon, \text{node}_h, \text{page}_i \rangle$, for $0 \leq i < n$. ϵ

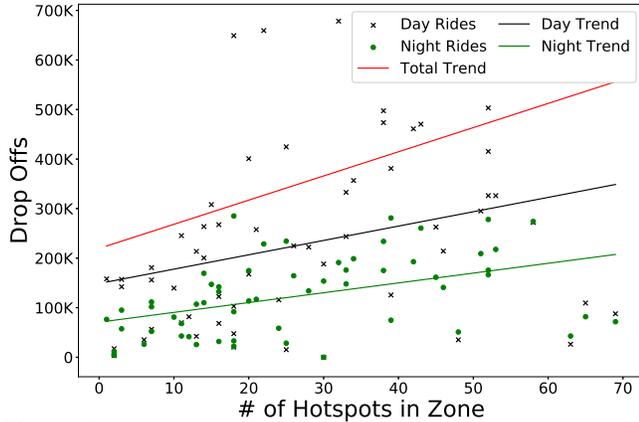


Figure 1: Taxi zones in Manhattan, ordered by their number of hotspots. Trend lines show correlation between drop-offs and hotspots.

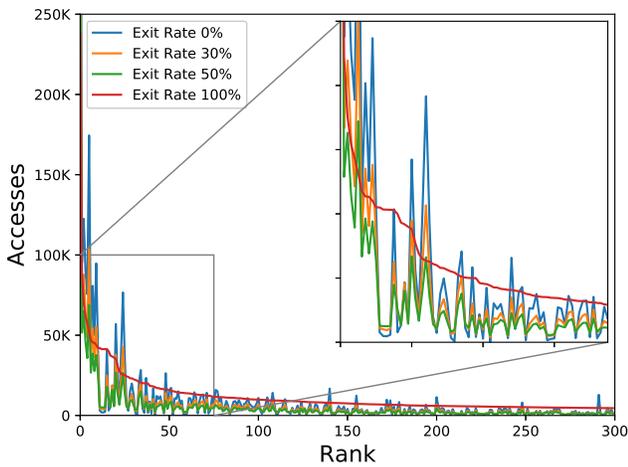


Figure 2: Wikipedia pages ordered by their static page ranks, and the number of accesses in our composite workload. Different exit rates result in visibly different distributions.

is a parameter representing the request rate within a session.

Workload properties. In addition to its use for evaluation purposes, our composite workload can also help characterize the system and its users. For example, Figure 1 shows the taxi zones in Manhattan, ordered by the number of hotspots in them (zones without hotspots were removed), and the number of rides that ended in them in the course of three months (May–July 2018), separated into daytime and nighttime. The trend lines, depicting linear regression, show the different mobility patterns throughout the day and indicate that the number of drop-offs is lightly correlated with the number of hotspots in the zone.

To understand the difference between static page ranks and dynamic access frequencies, we generated the composite workload with four different exit rates. Figure 2 shows the accesses to the 300 highest ranking pages in the first 4M requests in the composite trace. When the exit rate is 100%, the requests are distributed exactly according to the page ranks. With smaller rates, the request popularities differ from the static ones. We can distinctly notice that some high-ranking pages are accessed

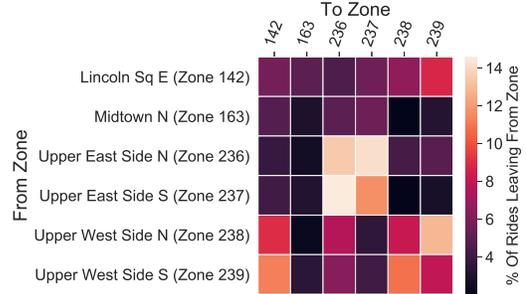


Figure 3: Rides between zones around Central Park, according to pick-up (‘from’) and drop-off (‘to’) zones. The distribution is non-uniform and mostly asymmetric.

very infrequently—these pages belong to a small category with few links from outside pages. To best represent user behavior, the exit rate should be chosen according to the application whose accesses are simulated.

Refinements and additional compositions. More detailed workloads could be used to generate comprehensive and possibly more accurate compositions. For example, if the pick-up and drop-off locations were given in finer granularity, we could assign sessions directly to the nearest hotspot rather than to a random hotspot in the zone. A detailed trace of the rides’ path, as in the SFO taxi dataset [15], could be used to generate sessions during the actual ride. At the same time, we note that even the aggregate information in the yellow taxi dataset can be used to represent user mobility. For example, Figure 3 depicts the most crowded zones bordering Central Park, and for each two zones $\langle Z_{from}, Z_{to} \rangle$, the probability that a ride starting at Z_{from} will end in Z_{to} . Similar characterization is often used for resource allocation and reservation in networked environments [92, 94, 103].

The composition we described can easily be done with alternative datasets. For example, user presence around a hotspot can be represented by the number of exits from nearby subway stations [16]. Similarly, object requests can be generated from any object distribution, or taken directly from a trace of requests. Datasets from location-dependent recommendation platforms, such as TripAdvisor, could be used to generate requests according to the user’s location. Finally, the number of sessions for each user and the inter-arrival times of requests within a session could be derived from a representative distribution or a separate workload, rather than a static parameter.

5 Conclusions

We described seven categories of attributes that are important for edge-system analysis, design, and optimization. We showed how, in the absence of up-to-date datasets with all, or even most of these attributes, partial datasets can be used to compose a workload with the required set of attributes. While these composite workloads provide an approximate representation of edge systems, applications, and user behaviors, they can help this field move forward by facilitating informed research, design, and management of the edge infrastructure.

6 Discussion topics

Is the absence of datasets really temporary? Looking forward, it is not clear how easy it would be to obtain edge workloads with all the relevant attributes, even when edge systems are fully deployed and widely used. Storage, Internet, and content providers are already reluctant to share their datasets, and the research community relies on a small variety of datasets in each domain. How realistic is it to expect more comprehensive workloads to be available in the future?

On the positive side, many datasets related to the edge infrastructure, such as locations of cellular towers and traffic surveillance cameras, are already public. Many mobility datasets originate from municipal repositories committed to open data, and this trend is likely to continue [93]. On the other hand, one of the most challenging aspects of workload collection is ensuring that they do not leak private information about the platform users, a challenge that is increased in edge-related workloads, where user and locality are major attributes.

Collecting detailed user and locality information requires either explicit user opt-in, aggregation, or both. In either case, this might become a critical issue preventing commercial operators from making their data public. In this case, the following questions arise: how important is detailed user information, compared to aggregate attributes? And, what is the best way to protect personal data while making the most detailed information available in the dataset?

Will workload composition become standard practice?

Datasets that include less attributes might be easier to anonymize, and thus easier to make public. Thus, it is possible that, while full and detailed edge workloads will remain proprietary and unavailable to the public, partial datasets will become increasingly available. In this scenario, workload composition might be the best way of incorporating the available datasets into the process of system research, design, and optimization.

How relevant are the basic workloads? Workload traces are inevitably limited in their ability to represent future demand for system resources. This limitation is bigger when the system itself changes, as in our case. In the absence of real edge workloads, how can we verify that our composite workloads are representative of future edge systems and applications? Which of the basic workloads described in this paper are most likely to represent future edge workloads? Are other relevant workloads available? Finally, can some predictable access patterns be generated synthetically for the purposes we described? Synthetic distributions of specific aspects can facilitate a more systematic coverage of a large parameter space, including its extremities. They can also be adapted as observed characteristics of edge systems change over time. Which aspects are better modelled by synthetic distributions?

How to generate realistic and useful compositions? By composing existing workloads into new ones we are making assumptions about correlations between workload attributes. For example, in our composite workload, we are assuming that

drop-off locations are correlated with user presence in an area. What type of analysis can we perform, and on which workloads, to determine which attributes are indeed independent of other attributes, which are correlated to others, and what is the nature of these correlations? These questions are part of the general challenge of establishing validity, reliability, and replicability of evaluations, which are addressed in the context of experimental design [49].

References

- [1] <http://traces.cs.umass.edu/>.
- [2] <https://mobiledegx.com/>.
- [3] <https://www.edgemicro.com/>.
- [4] <https://www.clearblade.com/>.
- [5] <https://www.edgeintelligence.com/>.
- [6] <https://www.edgeconnex.com/>.
- [7] SNIA IOTTA Repository. <http://iotta.snia.org/tracetypes/6>. Accessed: 2018-10-24.
- [8] Traces and snapshots public archive. <http://tracer.filesystems.org/>. Accessed: 2018-10-24.
- [9] Active TPC benchmarks. <http://www.tpc.org/information/benchmarks.asp>, February 2020.
- [10] AERPAW: Aerial experimentation and research platform for advanced wireless. <https://aerpaw.org/>, February 2020.
- [11] CAIDA data - overview of datasets, monitors, and reports. <https://www.caida.org/data/overview/>, February 2020.
- [12] Camera traffic counts dataset in city of AUSTIN. <https://data.world/cityofaustin/sh59-i6y9s>, February 2020.
- [13] City of Columbus GIS open data and map portal. <http://opendata.columbus.gov/>, February 2020.
- [14] CoGo bike share system data. <https://www.cogobikeshare.com/system-data>, February 2020.
- [15] CRAWDAD: A community resource for archiving wireless data at dartmouth. <https://crawdad.org/index.html>, February 2020.
- [16] MTA turnstile data. <http://web.mta.info/developers/turnstile.html>, February 2020.
- [17] NYC open data: 2018 yellow taxi trip data. <https://data.cityofnewyork.us/Transportation/2018-Yellow-Taxi-Trip-Data/t29m-gskq>, February 2020.

- [18] NYC open data: Nyc taxi zones. <https://data.cityofnewyork.us/Transportation/NYC-Taxi-Zones/d3c5-ddgc>, February 2020.
- [19] NYC open data: Nyc wi-fi hotspot locations. <https://data.cityofnewyork.us/Social-Services/NYC-Wi-Fi-Hotspot-Locations/a9we-mtpn>, February 2020.
- [20] oneTRANSPORT data marketplace. <https://onetransport.io/>, February 2020.
- [21] Open data DK. <https://www.opendata.dk/>, February 2020.
- [22] The OpenAPI specification: a broadly adopted industry standard for describing modern APIs. <https://www.openapis.org/>, February 2020.
- [23] Parking lot entrances and exits dataset in city of AUSTIN. <https://data.world/cityofaustin/ij6a-fwpi>, February 2020.
- [24] POWDER. <https://powderwireless.net/index>, February 2020.
- [25] RIPE network coordination center. <https://www.ripe.net/analyse/raw-data-sets>, February 2020.
- [26] Standard performance evaluation corporation. <https://www.spec.org/>, February 2020.
- [27] The top 500 sites on the web. <https://www.alexa.com/topsites>, February 2020.
- [28] Waymo open dataset. <https://waymo.com/open/>, February 2020.
- [29] Welcome to SELECT for cities knowledge hub. <https://www.select4cities.eu/>, February 2020.
- [30] Welcome to the Backblaze hard drive data and stats. <https://www.backblaze.com/b2/hard-drive-test-data.html>, February 2020.
- [31] Marcelo Abranches, Sepideh Goodarzy, Maziyar Nazari, Shivakant Mishra, and Eric Keller. Shimmy: Shared memory channels for high performance inter-container communication. In *2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19)*, Renton, WA, July 2019. USENIX Association.
- [32] George Amvrosiadis, Jun Woo Park, Gregory R. Ganger, Garth A. Gibson, Elisabeth Baseman, and Nathan DeBardeleben. On the diversity of cluster workloads and its impact on research results. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 533–546, Boston, MA, July 2018. USENIX Association.
- [33] Hamidreza Arasteh, Vahid Hosseinneshad, Vincenzo Loia, Aurelio Tommasetti, Orlando Troisi, Miadreza Shafie-khah, and Pierluigi Sian. IoT-based smart cities: A survey. In *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, 2016.
- [34] Berk Atikoglu, Yuehai Xu, Eitan Frachtenberg, Song Jiang, and Mike Paleczny. Workload analysis of a large-scale key-value store. *SIGMETRICS Perform. Eval. Rev.*, 40(1):53–64, June 2012.
- [35] C. Avasalcari, C. Tsigkanos, and S. Dustdar. Decentralized resource auctioning for latency-sensitive edge computing. In *2019 IEEE International Conference on Edge Computing (EDGE)*, pages 72–76, July 2019.
- [36] Peter Bailis, Alan Fekete, Ali Ghodsi, Joseph M. Hellerstein, and Ion Stoica. Scalable atomic visibility with RAMP transactions. *ACM Trans. Database Syst.*, 41(3), July 2016.
- [37] Mark Berman, Jeffrey S Chase, Lawrence Landweber, Akihiro Nakao, Max Ott, Dipankar Raychaudhuri, Robert Ricci, and Ivan Seskar. GENI: A federated testbed for innovative network experiments. *Computer Networks*, 61:5–23, 2014.
- [38] Charles C Byers. Architectural imperatives for fog computing: use cases, requirements, and architectural techniques for fog-enabled IoT networks. *IEEE Communications Magazine*, 55(8):14–20, 2017.
- [39] Daniel Campello, Hector Lopez, Ricardo Koller, Raju Rangaswami, and Luis Useche. Non-blocking writes to files. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*, 2015.
- [40] Justin Cappos, Matthew Hemmings, Rick McGeer, Albert Rafetseder, and Glenn Ricart. EdgeNet: a global cloud that spreads by local action. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 359–360. IEEE, 2018.
- [41] M. Chiang and T. Zhang. Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, Dec 2016.
- [42] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. PlanetLab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12, 2003.
- [43] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. Benchmarking cloud serving systems with YCSB. In *1st ACM Symposium on Cloud Computing (SoCC)*, Indianapolis, Indiana, USA, 2010.

- [44] Eli Cortez, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. In *26th Symposium on Operating Systems Principles (SOSP '17)*, 2017.
- [45] J. Courville and F. Chen. Understanding storage I/O behaviors of mobile applications. In *32nd Symposium on Mass Storage Systems and Technologies (MSST)*, 2016.
- [46] Jim Davis, Philbert Shih, and Alex Marcham. State of the edge 2018: A market and ecosystem report for edge computing. Report, State of the Edge, 2018.
- [47] U. Drolia, K. Guo, J. Tan, R. Gandhi, and P. Narasimhan. Cachier: Edge-caching for recognition applications. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS 17)*, 2017.
- [48] Jose M. Faleiro and Daniel J. Abadi. Rethinking serializable multiversion concurrency control. *Proc. VLDB Endow.*, 8(11):1190–1201, July 2015.
- [49] Ronald A Fisher. The design of experiments (9th ed.). [1935] (1971). MacMillan.
- [50] Robert Grandl, Srikanth Kandula, Sriram Rao, Aditya Akella, and Janardhan Kulkarni. Graphene: Packing and dependency-aware scheduling for data-parallel clusters. In *12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16)*, 2016.
- [51] Matthias Grawinkel, Lars Nagel, Markus Mäsker, Federico Padua, Andre Brinkmann, and Lennart Sorth. Analysis of the ECMWF storage landscape. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*, 2015.
- [52] Ajay Gulati, Chethan Kumar, and Irfan Ahmad. Storage workload characterization and consolidation in virtualized environments. In *VPACT*, 2009.
- [53] Tyler Harter, Brandon Salmon, Rose Liu, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. Slacker: Fast distribution with lazy Docker containers. In *14th USENIX Conference on File and Storage Technologies (FAST 16)*, 2016.
- [54] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. Mesos: A platform for fine-grained resource sharing in the data center. In *8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*, 2011.
- [55] Marco C Jacobs, Mark A Livingston, and Andrei State. Managing latency in complex augmented reality systems. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 49–ff, 1997.
- [56] Puneet Jain, Justin Manweiler, and Romit Roy Choudhury. Overlay: Practical mobile augmented reality. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 331–344, 2015.
- [57] A. Jonathan, M. Ryden, K. Oh, A. Chandra, and J. Weissman. Nebula: Distributed edge cloud for data intensive computing. *IEEE Transactions on Parallel and Distributed Systems*, 28(11):3229–3242, Nov 2017.
- [58] Albert Jonathan, Abhishek Chandra, and Jon Weissman. Locality-aware load sharing in mobile cloud computing. In *10th International Conference on Utility and Cloud Computing, UCC '17*, page 141–150, New York, NY, USA, 2017. Association for Computing Machinery.
- [59] Sudarsun Kannan, Ada Gavrilovska, and Karsten Schwan. Cloud4Home—enhancing data services with home clouds. In *2011 31st International Conference on Distributed Computing Systems*, pages 539–548. IEEE, 2011.
- [60] S. Kavalanekar, B. Worthington, Qi Zhang, and V. Sharda. Characterization of storage workload traces from production Windows servers. In *IISWC*, 2008.
- [61] Niko Kiukkonen, Jan Blom, Olivier Dousse, Daniel Gatica-Perez, and Juha Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. In *7th ACM International Conference on Pervasive Services (ICPS 10)*, 2010.
- [62] Ricardo Koller and Raju Rangaswami. I/O deduplication: Utilizing content similarity to improve I/O performance. *Trans. Storage*, 6(3):13:1–13:26, September 2010.
- [63] Chunghan Lee, Tatsuo Kumano, Tatsuma Matsuki, Hiroshi Endo, Naoto Fukumoto, and Mariko Sugawara. Understanding storage traffic characteristics on enterprise virtual desktop infrastructure. In *10th ACM International Systems and Storage Conference (SYSTOR '17)*, 2017.
- [64] Adisorn Lertsinsruttavee, Anwaar Ali, Carlos Molina-Jimenez, Arjuna Sathiseelan, and Jon Crowcroft. Pi-Casso: A lightweight edge computing platform. In *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, pages 1–7. IEEE, 2017.
- [65] Jure Leskovec. Stanford large network dataset collection. <https://snap.stanford.edu/data/index.html>, February 2020.

- [66] Jure Leskovec. Wikipedia network of top categories. <https://snap.stanford.edu/data/wiki-topcats.html>, February 2020.
- [67] Andrew W. Leung, Shankar Pasupathy, Garth Goodson, and Ethan L. Miller. Measurement and analysis of large-scale network file system workloads. In *USENIX (ATC)*, 2008.
- [68] D. Loghin, L. Ramapantulu, and Y. M. Teo. Towards analyzing the performance of hybrid edge-cloud processing. In *2019 IEEE International Conference on Edge Computing (EDGE)*, pages 87–94, July 2019.
- [69] Ao Ma, Fred Douglass, Guanlin Lu, Darren Sawyer, Surendar Chandra, and Windsor Hsu. RAIDShield: Characterizing, monitoring, and proactively protecting against disk failures. In *13th USENIX Conference on File and Storage Technologies (FAST '15)*, 2015.
- [70] Sumit Maheshwari, Dipankar Raychaudhuri, Ivan Seskar, and Francesco Bronzino. Scalability and performance evaluation of edge cloud systems for latency constrained applications. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 286–299. IEEE, 2018.
- [71] Redowan Mahmud, Ramamohanarao Kotagiri, and Rajkumar Buyya. Fog computing: A taxonomy, survey and future directions. In Beniamino Di Martino, Kuan-Ching Li, Laurence T. Yang, and Antonio Esposito, editors, *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, pages 103–130. Springer Singapore, Singapore, 2018.
- [72] Joe Mambretti, Jim Chen, and Fei Yeh. Next generation clouds, the chameleon cloud testbed, and software defined networking (SDN). In *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*, pages 73–79. IEEE, 2015.
- [73] A. Mehta, W. Tärneberg, C. Klein, J. Tordsson, M. Kihl, and E. Elmroth. How beneficial are intermediate layer data centers in mobile edge networks? In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, 2016.
- [74] Dutch T. Meyer and William J. Bolosky. A study of practical deduplication. In *FAST*, 2011.
- [75] Nitinder Mohan, Pengyuan Zhou, Keerthana Govindaraj, and Jussi Kangasharju. Managing data in computational edge clouds. In *Workshop on Mobile Edge Communications*, pages 19–24, 2017.
- [76] Seyed Hossein Mortazavi, Mohammad Salehe, Carolina Simoes Gomes, Caleb Phillips, and Eyal de Lara. Cloudpath: A multi-tier cloud computing framework. In *Second ACM/IEEE Symposium on Edge Computing (SEC 17)*, 2017.
- [77] Dushyanth Narayanan, Austin Donnelly, and Antony Rowstron. Write off-loading: Practical power management for enterprise storage. *Trans. Storage*, 4(3):10:1–10:23, November 2008.
- [78] Katerina Pechlivanidou, Kostas Katsalis, Ioannis Igoumenos, Dimitrios Katsaros, Thanasis Korakis, and Leandros Tassiulas. NITOS testbed: A cloud based wireless experimentation facility. In *2014 26th International Teletraffic Congress (ITC)*, pages 1–6. IEEE, 2014.
- [79] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. CRAWDAD dataset epfl/mobility (v. 2009-02-24). <https://crawdad.org/epfl/mobility/20090224/cab>, February 2009.
- [80] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejcki, M. Sherman, Z. Kostic, X. Gu, H. Krishnaswamy, S. Maheshwari, P. Skrimponis, and C. Gutterman. Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless. *Mobicom*, 2020 (to appear).
- [81] Dipankar Raychaudhuri, Ivan Seskar, Max Ott, Sachin Ganu, Kishore Ramachandran, Haris Kremos, Robert Siracusa, Hang Liu, and Manpreet Singh. Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 3, pages 1664–1669. IEEE, 2005.
- [82] Andreas Reiter, Bernd Prünster, and Thomas Zefferer. Hybrid mobile edge computing: Unleashing the full potential of edge computing in mobile device use cases. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 935–944. IEEE, 2017.
- [83] Robert Ricci, Eric Eide, and CloudLab Team. Introducing CloudLab: Scientific infrastructure for advancing cloud architectures and applications. ; *login.: the magazine of USENIX & SAGE*, 39(6):36–38, 2014.
- [84] Alma Riska and Erik Riedel. Disk drive level workload characterization. In *USENIX (ATC)*, 2006.
- [85] Drew Roselli, Jacob R. Lorch, and Thomas E. Anderson. A comparison of file system workloads. In *USENIX (ATC)*, 2000.
- [86] Bianca Schroeder and Garth A. Gibson. Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you? In *5th USENIX Conference on File and Storage Technologies (FAST 07)*, 2007.

- [87] Bianca Schroeder, Raghav Lagisetty, and Arif Merchant. Flash reliability in production: The expected and the unexpected. In *14th USENIX Conference on File and Storage Technologies (FAST 16)*, 2016.
- [88] Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016.
- [89] Pieter Simoens, David Griffin, Elisa Maini, T Khoa Phan, Miguel Rio, Luc Vermoesen, Frederik Vandeputte, Folker Schamel, and Dariusz Burstzynowski. Service-centric networking for distributed heterogeneous clouds. *IEEE Communications Magazine*, 55(7):208–215, 2017.
- [90] M Hibler R Ricci L Stoller, Jonathon Duerig, Shashi Guruprasad, Tim Stack, Kirk Webb, and Jay Lepreau. Large-scale virtualization in the EmuLab network testbed. In *USENIX Annual Technical Conference, Boston, MA*, 2008.
- [91] Huangshi Tian, Yunchuan Zheng, and Wei Wang. Characterizing and synthesizing task dependencies of data-parallel jobs in Alibaba cloud. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC '19)*, 2019.
- [92] Tuyen X Tran and Dario Pompili. Joint task offloading and resource allocation for multi-server mobile-edge computing networks. *IEEE Transactions on Vehicular Technology*, 68(1):856–868, 2018.
- [93] Jane Wakefield. Google given green light for Toronto smart city. Technical report, BBC News, October 2019.
- [94] Chenmeng Wang, Chengchao Liang, F Richard Yu, Qianbin Chen, and Lun Tang. Computation offloading and resource allocation in wireless cellular networks with mobile edge computing. *IEEE Transactions on Wireless Communications*, 16(8):4924–4938, 2017.
- [95] Feng Wang, Qin Xin, Bo Hong, Scott Brandt, Ethan Miller, and Darrell Long. File system workload analysis for large scale scientific computing applications. In *MSST*, 2004.
- [96] Shiqiang Wang, Rahul Uргаonkar, Murtaza Zafer, Ting He, Kevin Chan, and Kin K Leung. Dynamic service migration in mobile edge-clouds. In *2015 IFIP Networking Conference (IFIP Networking)*, pages 1–9. IEEE, 2015.
- [97] Yifan Wang, Shaoshan Liu, Xiaopei Wu, and Weisong Shi. CAVBench: A benchmark suite for connected and autonomous vehicles. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018.
- [98] John Wilkes. More Google cluster data. Google research blog, November 2011. Posted at <http://googleresearch.blogspot.com/2011/11/more-google-cluster-data.html>.
- [99] Jinlai Xu, Balaji Palanisamy, Heiko Ludwig, and Qingyang Wang. Zenith: Utility-aware resource allocation for edge computing. In *2017 IEEE international conference on edge computing (EDGE)*, pages 47–54. IEEE, 2017.
- [100] Gala Yadgar, Michael Factor, Kai Li, and Assaf Schuster. Management of Multilevel, Multiclient Cache Hierarchies with Application Hints. *ACM Transactions on Computer Systems (TOCS)*, 29:5:1–5:51, 2011.
- [101] Gala Yadgar and Moshe Gabel. Avoiding the streetlight effect: I/O workload analysis with SSDs in mind. In *8th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage)*, 2016.
- [102] Gala Yadgar, Oleg Kolosov, Mehmet Fatih Aktaş, and Emina Soljanin. Modeling the edge: Peer-to-peer reincarnated. In *2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19)*, 2019.
- [103] Changsheng You, Kaibin Huang, Hyukjin Chae, and Byoung-Hoon Kim. Energy-efficient resource allocation for mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, 16(3):1397–1411, 2016.
- [104] Aleksandr Zavodovski, Nitinder Mohan, Suzan Bayhan, Walter Wong, and Jussi Kangasharju. ExEC: Elastic extensible edge cloud. In *2nd International Workshop on Edge Systems, Analytics and Networking, EdgeSys '19*, page 24–29, New York, NY, USA, 2019. Association for Computing Machinery.
- [105] Lei Zhao, Jiajia Liu, Yongpeng Shi, Wen Sun, and Hongzhi Guo. Optimal placement of virtual machines in mobile edge computing. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.
- [106] D. Zhou, W. Pan, W. Wang, and T. Xie. I/O characteristics of smartphone applications and their implications for eMMC design. In *IEEE International Symposium on Workload Characterization (IISWC)*, 2015.
- [107] Deng Zhou, Wen Pan, Wei Wang, and Tao Xie. I/O characteristics of smartphone applications and their implications for EMMC design. In *IEEE International Symposium on Workload Characterization (IISWC '15)*, 2015.
- [108] Yuanyuan Zhou, James Philbin, and Kai Li. The Multi-Queue replacement algorithm for second level buffer caches. In *USENIX Annual Technical Conference (ATC)*, 2001.

- [109] H. Zhu and C. Huang. Availability-aware mobile edge application placement in 5G networks. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pages 1–6, Dec 2017.