Modeling The Edge: Peer-to-Peer Reincarnated

Gala Yadgar and Oleg Kolosov
Computer Science Department
Technion

Mehmet Fatih Aktaş and Emina Soljanin
Department of Electrical and Computer Engineering
Rutgers University

Abstract

The rise of edge computing as a new storage and compute model has already motivated numerous studies within the systems community, focusing on the choices and mechanisms of task offloading from end devices to the edge infrastructure, pricing, consistency, indexing and caching. However, it is not yet entirely clear how the edge infrastructure itself will be deployed, and, more importantly, managed. A common point of view considers the edge as an extension of traditional *content distribution networks (CDN)*, due to its hierarchical layout, centralized ownership, and cloud back-end.

In this paper, we consider a different view of the edge, as a "reincarnation" of the well-known *peer-to-peer (P2P)* model. We show how the edge is similar to P2P systems in many aspects, including the number, heterogeneity and limited availability and resources of its nodes, their central role in performing the system's storage and computation, and the vulnerabilities related to tight interoperability with user end devices. We describe the similarities of the edge to both CDNs and P2P systems, the challenges that arise from these similarities, and the previous approaches to address them in both contexts. We show that the challenges that remain in applying these approaches may be addressed by viewing the edge as a larger and smarter reincarnation of P2P systems.

1 Introduction

Edge computing is drawing increasing amounts of attention as the infrastructure facilitating the IoT revolution, as well as the promise of autonomous vehicles and advances in online video gaming, virtual and augmented reality, and numerous additional machine learning applications. The main advantage of edge systems is the close proximity of storage and compute capabilities to the end user. This proximity is achieved by deploying a large number of edge nodes which are placed one or two network hops, approximately several milliseconds, from the end user. These nodes differ from traditional data center nodes not only in their wide geographical distribution, but also in their limited resources [30].

Current research efforts to improve the availability and performance of edge-based services address various aspects of the service, including pricing [102], consistency [42, 60], decoupling the application's tasks from one another [52, 65, 72], orchestration of edge and cloud resources [60, 90], indexing, and caching [23, 35, 41]. Industry efforts cover similar aspects, which can be roughly categorized into delivery, platforms, infrastructure, hardware, connectivity, routing, and software [30]. All these efforts address the edge infrastructure to various extents, each making different assumptions about its layout, centralization of management, costs, and objectives.

Despite the differences between them, all these studies assume, implicitly or explicitly, that the edge infrastructure presents a unified and even centralized interface to the application. Thus, a common view is that *"edge computing generalizes and extends the CDN concept by leveraging cloud computing infrastructure [81]."* Indeed, the hierarchical layout, centralized ownership, and cloud-based backend of edge systems are very similar to those of content distribution networks (CDNs). At the same time, edge systems differ from CDNs in several major aspects that require special attention.

Interestingly, these differences all correspond to similarities between edge systems and peer-to-peer (P2P) systems:

- The lowest level of the edge hierarchy has an exceptionally high fan-out, with a number of nodes that may prevent centralized management.
- Edge nodes are heterogeneous and their availability may be limited by occasional excessive load.
- Edge devices are not only heterogeneous but also privately owned, and operate according to local, "selfish", objectives.
- A large portion of storage and computation takes place at the bottom layers of the edge hierarchy.
- Untrusted edge devices and unsecure connections present a plethora of security threats.

P2P has not prevailed as a robust economic model due to a combination of reasons: the cost of addressing the above challenges is high, while, approximately a decade ago, cloud services started providing an appealing alternative. In this paper, we make the following observation: in spite of the decline of P2P, the increased focus on edge systems must bring the challenges of P2P systems back to our attention. In a sense, we view the edge as a reincarnation of the P2P model: larger in scale, with enhanced capabilities, and, most importantly, without a feasible alternative.

In the following, we describe the similarities between the edge and CDNs as well as P2P systems. We discuss the resulting challenges and revisit the approaches taken to address these challenges in the context of both models, and examine their applicability to the edge. We believe that now, when operational edge systems are only beginning to emerge, it is especially crucial to bring forward these challenges and to focus especially on those that were not fully addressed in the "previous incarnation" of the model.

2 Background

We begin by briefly describing the three system models—content distribution networks, peer-to-peer, and the edge—focusing on the characteristics and challenges that distinguish them from one another.

Content distribution networks (CDNs) constitute tens of thousands of servers organized hierarchically as a "virtual network" [46]. The CDN's customers are *content providers*, typically website or application owners. The CDN's servers are used to cache their customers' popular content, route user requests to the nearest copy of the content, and provide alternate network paths in case of congestion or connectivity problems. Some servers support operations for application acceleration, such as content prefetching, compression, data collection and aggregation, and dynamic page generation [64].

Since all the servers belong to the CDN operator[1], their resources can be managed in a way that optimizes a global performance or business objective. The challenges in designing such systems stem from the need to achieve these objectives in very large scale and in a distributed manner. Thus, issues related to CDN optimization include caching, prefetching and refreshing of content, authentication and cryptographic key management, path optimization, system mapping and monitoring, and fault tolerance.

Akamai was the first to offer a commercial CDN service, and remains one of the largest players in this market to date. It delivers daily Web traffic reaching more than 30 Terabits per second, using more than 240,000 servers, with "*85% of the world's Internet users within a single network hop of an Akamai CDN server*" [4]. Other major players include Google, Level 3 Communications, Limelight Networks, and Amazon Web Services. Some CDN providers engage in peering contracts with other providers or ISPs, to improve routing and to avoid bottlenecks [61].

Peer-to-peer (P2P) systems (networks) are composed of individual peers' end devices, such as smartphones, desktop or laptop computers, or even servers. The peers cooperate by contributing their resources to serve one another's requests. Common applications of P2P systems include file download and sharing, multicast and message passing in ad-hoc networks, and crypto-currency computation and attestation.

Most of the challenges in designing a P2P system can be attributed to its nodes belonging to different owners, who may or may not be online and cooperative at different times. Thus, these systems may suffer high churn, low availability, and arbitrary (and dynamic) geographical distributions. Furthermore, since peers rarely share a global objective, some incentive mechanism is necessary to induce cooperation. Typical examples include BitTorrent's tit-for-tat policy [29], reputations [39, 71], and credit based (virtual currency) systems [51, 67, 87, 105]. Other issues addressed in P2P systems include overlay and state maintenance [38, 76, 78], controlling membership to avoid attacks by malicious nodes [34, 95], ensuring availability, durability, and integrity of data [53, 71], and minimizing cross-ISP traffic [26, 82].

P2P systems started to gain popularity and interest approximately two decades ago, peeking approximately one decade ago, and have been gradually declining ever since. P2P has not prevailed as a robust economic model due to the limited utilization and robustness it provides. In addition, while many solutions were proposed to the challenges described above, varying in their degrees of optimality and applicability, their cost was too high, and their benefit not high enough. At the same time, cloud-based solutions have become available, providing an attractive alternative. For example, Spotify has relied on its P2P network since its launch in 2008, but has completed a transition to relying on its own servers several years ago [92]. On the other hand, BitTorrent, which maintained its P2P model, prevailed mainly as a platform for unlicensed file sharing.

Hybrid CDNs, sometimes referred to as *peer-assisted CDNs*, complement the traditional CDN hierarchy by offloading some of the load from the CDN servers to the user's end devices. For example, in Akamai's NetSession [103], the server may redirect download requests to nearby peers who are known to store the data. Although peer participation is not mandatory, NetSession deliberately avoids incentives, relying on the server to absorb traffic that is not handled by available peers. Similarly, other hybrid-CDN designs show that even limited peer participation within an ISP's service region can considerably reduce server load [19, 47, 66, 70, 103].

Edge systems provide storage and compute infrastructure at interoperating *edge nodes* located one or two network hops from the end user [22, 25, 57, 85]. An edge node is an aggregation of storage and compute servers limited to a typical capacity of 50-150 kW and a diameter of approximately 10 feet. These limitations come from non-traditional locations in which such nodes are deployed, such as the base of cellular network towers. These locations also dictate the limited network resources available to the nodes [30]. We distinguish between edge nodes, which are part of the edge infrastructure, and *edge devices* owned and operated by the end user, e.g., smart phones, wearable devices, sensors, etc. Nevertheless, we include in our discussion the number, availability, and heterogeneity of edge devices and their resources, as some systems rely on those resources for augmenting the service provided by the edge infrastructure.

Despite the large involvement of individual companies

---

[1] In this paper, we refer to peer-to-peer CDNs as peer-to-peer systems.

and joint initiatives [9, 10, 98], edge-based services currently consist of small-scale ad-hoc research prototypes and initial testbeds and platforms [1, 3, 5, 7, 8, 11, 44, 93]. The absence of full-scale operational systems makes it hard to construct an analytical model of the edge, specifically in terms of the costs and objectives of such a model. Most theoretical studies that address placement and scheduling of jobs model the edge system as a global queue managing a hierarchy of servers between the cloud and the end users [33, 36, 37, 55, 58, 59], while some address only specific layers.

For example, the study of (i) *fog computing* considers computation within mini-clouds located at the network edge, close to users [21], (ii) *mist computing* considers collaborative computation over the devices between the mini-cloud and the end users (e.g., switches, wireless access points) [48, 91], and (iii) *dew computing* involves the end devices in computation together with the cloud [88, 96]. Each of those studies makes different assumptions on the connectivity and interoperability of the edge system's components. This makes it difficult to compare their contributions and to apply their results in a broad applicable context.

In the following, we discuss the major aspects relevant to modelling a general edge system, and argue that in most of these aspects, the edge will have much in common with traditional P2P systems. As a result, we may have to revisit challenges previously addressed in the context of these systems.

3 Layout and membership

In this section, we address the edge system's layout and the challenges related to tracking the current system's state and managing its resources accordingly.

Aspects in common with CDNs. The State of the Edge report defines the practical aspect of edge computing as *"distributing new resources and software stacks along the path between today's centralized data centers and the increasingly large number of devices in the field, concentrated, in particular, but not exclusively, in close proximity to the last mile network, on both the infrastructure side and the device side"* [30]. Thus, when considering an edge system, we view the edge nodes as the lowest level of a well-defined tree-like hierarchy whose root is the cloud-based data center. Since the edge nodes are deployed at static locations, their membership and geographical location and coverage are easily identified and maintained, like those of a CDN.

Aspects in common with P2P. To achieve the low latency required by edge services, edge nodes must be deployed as close as possible to the users, implying their number would be orders of magnitude *larger* than that of user-facing servers in a single CDN. Furthermore, the storage, network, and compute resources available at each edge node are expected to be at least one or two orders of magnitude *smaller* than those of their CDN counterparts [86]. Combined with highly skewed demand and mobility of users, this may lead to non-negligible probability of individual nodes becoming unavailable. Since each node will likely participate in several edge services, high load generated by users of one service may reduce the availability of the node for other services. The availability and membership of edge devices is even more dynamic, and their large numbers make them additionally challenging to track.

Implications. Thanks to the hierarchical nature of edge systems, edge nodes may be grouped to subtrees and managed by servers at the higher levels of the hierarchy. These higher-level servers may also be leveraged for coordination between nodes of adjacent subtrees. Indeed, current CDN architectures already route requests to alternative servers and via alternative paths based on similar principals [46]. However, the high number of edge nodes presents a new challenge in maintaining their state and balancing their load, which must be done in a highly distributed manner. This is even more challenging for systems that rely on collaborative resources of edge devices.

Membership and layout have been extensively studied in the context of P2P systems, many of which utilize distributed topology protocols for lookups and request routing. Pastry [78] and Bamboo [76] implement a *distributed hash table (DHT)* with a lookup cost of $O(logN)$ hops, but do not deal well with high churn. Coral [38] adds locality to DHTs by mapping nodes to hierarchical clusters, and ChunkCast [27] optimizes lookup for large objects. In gossip-based protocols [67], peers have knowledge of their neighbors, through which they can connect to the "swarm". Though completely distributed, the high lookup costs may be too high for latency critical applications. Server-assisted lookup and routing [6, 19, 26, 83, 103] is significantly more efficient and may be more appropriate for edge systems. For example, Skype servers have replaced supernodes in the Skype P2P system for improved scalability [97].

The second challenge in such a dynamic system layout is ensuring its availability. To that end, edge-based services must not rely on individual nodes for a specific service, and should ensure sufficient redundancy of data as well as compute capabilities. In theory literature, redundant storage schemes and optimized allocation of redundant data have been proposed in the context of distributed storage systems. They ensure that, even if some nodes become unavailable, a user can likely access data and get service jointly from other nodes [12, 16, 17, 54, 63, 69, 79]. Finding such schemes is not easy, and is connected to certain long time open math problems [14].

Traditional replication techniques incur unacceptably high overheads: n-way replication multiplies the storage capacity by n, and task cloning increases the contention on compute resources [15]. Lower redundancy can be achieved by using erasure coding of both data objects and tasks. However, the amount of redundancy required to ensure availability in systems with low availability is still high. For example, the OceanStore P2P prototype uses 10-way replication and (32,16) erasure coding [53, 75]. Similarly, in distributed computing, although erasure coding has been shown to greatly reduce the extra load incurred by the redundant tasks [13], redundancy still incurs additional load and the sufficient degree of redundancy that would not overburden the system is still unknown. Thus, redundancy is used with great care in today's compute systems [15, 31, 74, 100, 101]. Edge systems are fundamentally different from traditional distributed storage, and redundancy techniques for the edge have not been explored.

4 Data and service "center of mass"

In this section we consider the different locations in which online and latency-critical computation occurs, and where the relevant data objects are stored.

Aspects in common with CDNs. One premise of edge-based applications is that they require data and/or computational power unavailable at the edge devices on which they run. Thus, a common use-case in machine-learning applications involves training a model over a large data set and storing it in the cloud. At the application side, effort is made to split the inference logic between the edge devices and the edge servers [52, 65, 90]. At the infrastructure side, current efforts focus on managing the content of the edge nodes (and possibly additional servers) as a cache hierarchy [35, 60]. In this use case, the edge nodes serve as an 'accelerator', minimizing the number of round-trips to the cloud, but not replacing them altogether. If the data is unavailable at the edge, its latest copy is fetched from the cloud, like in a CDN.

Aspects in common with P2P. Another premise of edge-based applications is the large amount of data that is generated and collected at the end devices. Examples include states in online games, sensor data in video surveillance and autonomous computing, and various virtual and augmented reality applications [30, 41, 73, 90]. In the case of IoT devices, the magnitude of data collection means that most of the data must be processed and aggregated as close as possible to its source, i.e., at the end devices and edge nodes, to reduce the amount of data transferred to the cloud [84]. In all these scenarios, data flows bidirectionally between the edge nodes and the cloud, and is processed at both sides. For latency-critical applications, such as autonomic vehicles and real-time surveillance, processing by edge nodes is critical to achieve the required response times, and relaying tasks to the cloud is not an option. Instead, adjacent edge nodes and even devices may provide alternative service at acceptable latency [41].

Implications. In edge-based applications, the edge is not only where the application's output is redirected, it is where its data and compute 'center of mass' are located. At the same time, the cloud remains responsible for long-term data storage and analysis, and will often hold the most up-to-date state (e.g., machine learning models based on the most recent input and statistics). Thus, to provide real-time service, potentially even when temporarily disconnected from the cloud infrastructure, edge systems must combine the caching functionalities of CDNs with collaboration principles of P2P systems.

Many optimizations have been proposed for managing the content of CDN servers as a caching hierarchy [18, 64, 80], where neighboring servers may be leveraged for more efficient large-file transfer [24] or request re-routing. Hybrid CDNs are particularly relevant to edge systems, because their servers re-route requests to nearby *users* that are likely to store the requested object [19, 47, 66, 70, 103]. Another interesting model is that of federated CDNs, where otherwise competing CDNs collaborate to increase their coverage in terms of geographical location, content, and peak loads [62]. These models only address the caching and routing of objects that originate from the content provider.

P2P systems provide solutions to data originating from the users themselves, and for the allocation of their compute resources. Examples include the Na Kika [40] and CoBlitz [68] P2P-CDNs, the OceanStore [53] distributed P2P object storage system, severless distributed file systems [20, 32], and ad-hoc P2P networks [71, 105]. The design of these systems addresses load balancing and utilization of distributed heterogenous resources. As mentioned in Section 2, their main limitation is the high overhead of their protocols, especially in very large systems. Thus, their applicability to large-scale latency-critical edge-based applications must be considered carefully.

5 Ownership and objectives

In this section, we examine the ownership of the edge system's components, and the effect of ownership as well as other factors on the different objective functions in the system.

Aspects in common with CDNs. Given current market trends and main players [7, 11, 30, 102], it is reasonable to assume that Internet, cloud, and wireless service providers will each deploy their own edge system, consisting of a large number of geographically dispersed edge nodes. Each operator will likely manage its resources according to its own *global* objectives, in terms of performance guarantees, availability, and business costs. For example, ETSI describes mobile edge computing as a platform owned by the mobile operator and managed in the benefit of its customers [45]. Other models of managing edge resources address such global objectives [35, 81, 90], and some extend this global management to the entire path between the edge and the cloud [60, 84].

Aspects in common with P2P. Nodes owned by multiple providers may collaborate under certain circumstances. For example, if excessive load or a cloud outage [81] are experienced within a certain geographical area, collaborative caching and request forwarding can allow several providers to meet their required service levels. As another example, a virtual collaborative edge may provide a unified interface to collections of data owned by different organizations and served by different physical systems [84]. Even within a single provider's edge system, the need to deploy a large number of nodes with the best possible geographical coverage will result in diverse deployment scenarios: while some nodes will be deployed at mobile base stations or wifi hot-spots, others may be deployed within central businesses, such as coffee or food chains. The location of an edge node may affect its *local* objectives and costs. For example, the contract between an edge provider and a coffee chain might specify the rent and electricity costs paid by the edge provider, as well as the SLA class it should provide to the store's customers. Thus, this node will have to prioritize requests originating from within the store over requests forwarded from other nodes. Finally, storage and compute resources of edge devices may be leveraged to mask outages at lower levels of the network. Collaboration between such devices follows traditional P2P collaboration models.

Implications. In the collaboration scenarios described above, edge nodes, groups of nodes, or edge devices should be viewed as *selfish* entities, in the game-theoretical sense. In other words, each node has local objectives, and will collaborate with other nodes only if this collaboration will help it achieve its own objectives. While all participating nodes are expected to benefit from long-term collaboration, some may not benefit from it in the short-term, and may thus refuse to cooperate. Collaboration between such selfish entities is typically achieved by the introduction of an *incentive* mechanism.

Incentives were extensively studied in the context of P2P systems. BitTorrent's tit-for-tat strategy is sufficient to induce cooperation between peers [29], although 'free-riders' may reduce its overall system's utilization [49]. Incentive mechanisms based on reputations [71] or virtual currency [50] are generally more robust, but they are also susceptible to exploitation by colluding peers and by hoarders [105] or have limited scalability [51, 87]. The main limitation of virtual currency is the overhead of verifying the virtual coins, which requires a dedicated and trusted central server or a costly distributed mechanism. These overheads must be addressed in order to apply similar mechanisms in resource-constraint environments and latency-critical applications in edge systems. At the same time, centrally owned servers at higher levels of the edge system's hierarchy may be leveraged to improve the efficiency of such traditionally distributed mechanisms [99].

Another limitation of existing incentive mechanisms is their tight coupling with their applications. These mechanisms are designed for systems serving one type of requests, such as serving a cached block, passing a message, or serving a forwarded request. Collaboration between heterogenous edge nodes, however, may be more efficient if it combines applications and request types. For example, one node may have more available threads than its peer, which, in turn, may have more available memory for caching. These collaboration scenarios require more complicated mechanisms, and, possibly, complex *valuation functions*, for nodes to compare the costs of collaboration with its benefits.

6 Security, privacy and trust

In this section, we discuss the security issues that arise from the tight coupling of the edge infrastructure and the applications running on the edge nodes and devices.

Aspects in common with CDNs. The servers comprising the edge infrastructure are deployed by the system's provider. As such, their identity is static and can be securely verified. We can also assume, to some extent, that physical access to the edge nodes is secured, despite their geographically dispersed locations, which may include third-party premises.

Aspects in common with P2P. Edge nodes are exposed to several types of security threats. In the "infrastructure as a service" model, clients offload computation to edge nodes, which makes them vulnerable to malicious code. Each edge node must allocate its resources between the numerous edge devices that it serves (whose identity is unknown) and is thus vulnerable to DDoS attacks. Edge nodes are more vulnerable to such attacks than CDN nodes because each request can potentially consume larger portions of the node's resources. In application scenarios where the edge node is responsible for aggregating, summarizing, and analyzing input streams from a large collection of edge devices, malicious devices can compromise the result of this process by injecting adversarial input data. These threats are common to edge and P2P systems, which both involve unsecure end devices and networks [84, 89].

Implications. Symantec's Internet security threat report [28] predicts a continued increase in IoT attacks, which *"will likely diversify as attackers seek new types of devices to add to botnets"*. Indeed, the high vulnerability of the edge infrastructure entails extensive security measures. Nodes must verify the identity of the edge devices and the authenticity of their data over untrusted communication paths. These challenges have been studied extensively in the context of large-scale data centers as well as P2P systems.

Isolating collocated applications and workloads from one another is addressed in shared servers by running them within virtual machines or containers [2]. Oblivious computations allow query processing without access pattern and information leakage in centralized or distributed environments [56, 104]. For protection against malicious or non-cooperative nodes, P2P systems rely on Byzantine fault tolerance [53, 94], encryption [53], message authentication [39, 77], and tracking peers' history [43, 94]. The applicability of such resource intensive solutions to edge systems is limited by their high overheads [89] and possible scalability issues, and remains a major challenge that must be addressed.

7 Conclusions

In this short paper, we examined the common view of the edge as an extended CDN, and argued that in many crucial aspects, the edge has many challenges in common with P2P systems. At the same time, the resemblance to CDNs may facilitate more effective solutions to these challenges than those available in traditional, completely decentralized, P2P systems. The best way to utilize the infrastructure's hierarchical structure and global perspective remains to be found.

8 Discussion topics

This paper describes a series of challenges that must be overcome for the edge to fulfill its disruptive potential. We hope that our new point of view will inspire and motivate new approaches for addressing these challenges. Specifically, in the context of the workshop, we hope it will stimulate discussion of several open issues regarding the applicability of existing approaches to the edge:

- How big is the advantage of the hierarchical layout and centralized ownership of the edge? In other words, how much can we expect to improve on the P2P solutions by utilizing centrally managed components before they become the bottleneck?
- What is the "price of anarchy"—the difference in the utilities of systems with and without cooperation between edge nodes and devices? Will this cost justify the complexity of potential incentive mechanisms and security threats?
- What is the level of availability that we should expect from edge nodes? Can it be low enough to justify the high redundancy used in P2P systems, or will providers simply deploy additional nodes if their availability drops below an acceptable level?

Some of our claims may be considered controversial, and we are looking forward to hearing different points of view on these issues:

- Should we even try to model and address the edge system in the context of a general model? Perhaps most of the challenges we describe can be solved in the context of specific use-cases and applications.
- Can any of the challenges described in this paper be considered solved? Specifically, hybrid-CDNs and P2P-CDNs have utilized central servers to deal with some of the challenges of P2P systems. Are those solutions directly applicable to the edge?
- Have we overlooked technological advances or hardware solutions that trivialize some of the challenges we described?

Finally, perhaps the most interesting question is whether any of these challenges going to be the "deal breaker" for the edge vision. Since most of the challenges that we describe have alreay been encountered in various forms in the context of P2P systems, will the cost of addressing them be too high for the edge as well?

References

[1] IBM and Nokia Siemens Networks announce world's first mobile edge computing platform. https://www-03.ibm.com/press/us/en/pressrelease/40490.wss, February 2013.

[2] Introduction to container security. White paper, Docker, August 2016.

[3] Akamai Cloudlets. https://cloudlets.akamai.com/, March 2019.

[4] Akamai facts & figures. https://www.akamai.com/uk/en/about/facts-figures.jsp, February 2019.

[5] Amazon Lambda@Edge. https://aws.amazon.com/lambda/edge/, March 2019.

[6] Apache ZooKeeper^{TM}. https://zookeeper.apache.org/, February 2019.

[7] Azure IoT Edge. https://azure.microsoft.com/en-us/services/iot-edge/, February 2019.

[8] The federated raspberrypi μ-infrastructure testbed. https://fruit-testbed.org/, February 2019.

[9] Open Edge Computing Initiative. http://openedgecomputing.org/, February 2019.

[10] Open Fog Consortium. https://www.openfogconsortium.org/, February 2019.

[11] Securely connect, collect and start processing IoT data quickly and easily with Watson IoT Platform. https://www.ibm.com/internet-of-things/solutions/iot-platform/watson-iot-platform, February 2019.

[12] Mehmet Aktas, Sarah E. Anderson, Ann Johnston, Gauri Joshi, Swanand Kadhe, Gretchen L. Matthews, Carolyn Mayer, and Emina Soljanin. On the service capacity region of accessing erasure coded content. In *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton 17)*, 2017.

[13] Mehmet Fatih Aktas, Pei Peng, and Emina Soljanin. Effective straggler mitigation: Which clones should attack and when? *SIGMETRICS Perform. Eval. Rev.*, 45(2):12–14, October 2017.

[14] Noga Alon, Peter Frankl, Hao Huang, Vojtech Rödl, Andrzej Rucinski, and Benny Sudakov. Large matchings in uniform hypergraphs and the conjectures of Erdős and Samuels. *J. Comb. Theory, Ser. A*, 119:1200–1215, 2012.

[15] Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, and Ion Stoica. Effective straggler mitigation: Attack of the clones. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013.

[16] Sarah E. Anderson, Ann Johnston, Gauri Joshi, Gretchen L. Matthews, Carolyn Mayer, and Emina Soljanin. Service rate region of content access from erasure coded storage. In *IEEE Information Theory Workshop, ITW 2018, Guangzhou, China, November 25-29, 2018*, pages 1–5, 2018.

[17] Iryna Andriyanova and Pablo M Olmos. On distributed storage allocations for memory-limited systems. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2015.

[18] Daniel S. Berger, Ramesh K. Sitaraman, and Mor Harchol-Balter. AdaptSize: Orchestrating the hot object memory cache in a content delivery network. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017.

[19] Danny Bickson and Dahlia Malkhi. The Julia content distribution network. In *2nd Conference on Real, Large Distributed Systems (WORLDS 05)*, 2005.

[20] William J. Bolosky, John R. Douceur, David Ely, and Marvin Theimer. Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs. In *2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 00)*, 2000.

[21] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[22] 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.

[23] Ashish Chawla, Benjamin Reed, Karl Juhnke, and Ghousuddin Syed. Semantics of caching with SPOCA: A stateless, proportional, optimally-consistent addressing algorithm. In *USENIX Annual Technical Conference (USENIX ATC 11)*, 2011.

[24] Ludmila Cherkasova and Jangwon Lee. FastReplica: Efficient large file distribution within content delivery networks. In *4th Conference on USENIX Symposium on Internet Technologies and Systems (USITS 03)*, 2003.

[25] M. Chiang and T. Zhang. Fog and iot: An overview of research opportunities. *IEEE Internet of Things Journal*, 3(6):854–864, Dec 2016.

[26] David R Choffnes and Fabián E Bustamante. Taming the torrent: a practical approach to reducing cross-ISP traffic in peer-to-peer systems. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 363–374. ACM, 2008.

[27] Byung-Gon Chun, Peter Wu, Hakim Weatherspoon, and John Kubiatowicz. ChunkCast: An anycast service for large content distribution. In *5th International workshop on Peer-To-Peer Systems (IPTPS 06)*, 2006.

[28] Gillian Cleary, Mayee Corpin, Orla Cox, Hon Lau, Benjamin Nahorney, Dick O'Brien, Brigid O'Gorman, John-Paul Power, Scott Wallace, Paul Wood, and Candid Wueest. Symantec Internet security threat report. Report 23, Symantec, March 2018.

[29] Bram Cohen. Incentives Build Robustness in BitTorrent. In *Workshop on Economics of Peer to Peer Systems (P2PECON)*, 2003.

[30] 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.

[31] Jeffrey Dean. Achieving rapid response times in large online services. Talk given at Berkeley AMPLab Cloud Seminar, March 2012.

[32] Michael J Demmer, Bowei Du, and Eric A Brewer. Tierstore: A distributed filesystem for challenged networks in developing regions. In *7th USENIX Conference on File and Storage Technologies (FAST 03)*, 2008.

[33] Ruilong Deng, Rongxing Lu, Chengzhe Lai, Tom H Luan, and Hao Liang. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet of Things Journal*, 3(6):1171–1181, 2016.

[34] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[35] 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.

[36] Mohammad Etemad, Mohammad Aazam, and Marc St-Hilaire. Using devs for modeling and simulating a fog computing environment. In *2017 International Conference on Computing, Networking and Communications (ICNC)*, pages 849–854. IEEE, 2017.

[37] Qiang Fan and Nirwan Ansari. Towards workload balancing in fog computing empowered iot. *IEEE Transactions on Network Science and Engineering*, 2018.

[38] Michael J. Freedman, Eric Freudenthal, and David Mazières. Democratizing content publication with Coral. In *1st Symposium on Networked Systems Design and Implementation (NSDI 04)*, 2004.

[39] Yun Fu, Jeffrey Chase, Brent Chun, Stephen Schwab, and Amin Vahdat. SHARP: An architecture for secure resource peering. *ACM SIGOPS Operating Systems Review*, 37(5):133–148, 2003.

[40] Robert Grimm, Guy Lichtman, Nikolaos Michalakis, Amos Elliston, Adam Kravetz, Jonathan Miller, and Sajid Raza. Na Kika: Secure service execution and composition in an open edge-side computing network. In *3rd Conference on Networked Systems Design & Implementation (NSDI 06)*, 2006.

[41] Harshit Gupta, Zhuangdi Xu, and Umakishore Ramachandran. Datafog: Towards a holistic data management platform for the iot age at the network edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[42] Zijiang Hao, Shanhe Yi, and Qun Li. Edgecons: Achieving efficient consensus in edge computing networks. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[43] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on Bitcoin's peer-to-peer network. In *24th USENIX Security Symposium (USENIX Security 15)*, 2015.

[44] Pia Hettinger. Deutsche Telekom and partners build Edge Computing testbed with low-latency vRAN technology: The Living Edge Lab. https://www.telekom.com/en/media/media-information/archive/living-edge-lab-515082, February 2018.

[45] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile edge computing—a key technology towards 5G. White paper 11, ETSI, September 2015.

[46] Cheng Huang, Angela Wang, Jin Li, and Keith W Ross. Measuring and evaluating large-scale CDNs. In *ACM IMC*, volume 8, pages 15–29, 2008.

[47] Cheng Huang, Angela Wang, Jin Li, and Keith W. Ross. Understanding hybrid CDN-P2P: Why Limelight needs its own red swoosh. In *18th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 08)*, 2008.

[48] Michaela Iorga, Larry Feldman, Robert Barton, Michael J Martin, Nedim S Goren, and Charif Mahmoudi. Fog computing conceptual model. Technical report, 2018.

[49] Seung Jun and Mustaque Ahamad. Incentives in BitTorrent induce free riding. In *2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 116–121. ACM, 2005.

[50] Kfir Karmon, Liran Liss, and Assaf Schuster. Gwiq-p: an efficient decentralized grid-wide quota enforcement protocol. In *14th IEEE International Symposium on High Performance Distributed Computing (HDPC 05)*, 2005.

[51] Ian A. Kash, Eric J. Friedman, and Joseph Y. Halpern. Optimizing scrip systems: Efficiency, crashes, hoarders, and altruists. In *8th ACM Conference on Electronic Commerce (EC '07)*, 2007.

[52] Ana Klimovic, Yawen Wang, Christos Kozyrakis, Patrick Stuedi, Jonas Pfefferle, and Animesh Trivedi. Understanding ephemeral storage for serverless analytics. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, 2018.

[53] John Kubiatowicz, David Bindel, Yan Chen, Steven Czerwinski, Patrick Eaton, Dennis Geels, Ramakrishna Gummadi, Sean Rhea, Hakim Weatherspoon, Westley Weimer, Chris Wells, and Ben Zhao. OceanStore: An architecture for global-scale persistent storage. In *Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 00)*, 2000.

[54] Derek Leong, Alexandros G Dimakis, and Tracey Ho. Distributed storage allocations. *IEEE Trans. Information Theory*, 58(7):4733–4752, 2012.

[55] Wei Li, Igor Santos, Flavia C Delicato, Paulo F Pires, Luci Pirmez, Wei Wei, Houbing Song, Albert Zomaya, and Samee Khan. System modelling and performance evaluation of a three-tier cloud of things. *Future Generation Computer Systems*, 70:104–125, 2017.

[56] C. Liu, X. S. Wang, K. Nayak, Y. Huang, and E. Shi. ObliVM: A programming framework for secure computation. In *2015 IEEE Symposium on Security and Privacy*, 2015.

[57] 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.

[58] Yuyi Mao, Jun Zhang, SH Song, and Khaled B Letaief. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems. *IEEE Transactions on Wireless Communications*, 16(9):5994–6009, 2017.

[59] Tomoya Mori, Yoichi Utsunomiya, Xuejun Tian, and Takashi Okuda. Queueing theoretic approach to job assignment strategy considering various inter-arrival of job in fog computing. In *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2017.

[60] 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.

[61] Limelight Networks. Limelight realtime streaming: Sub-second latency for live online streaming video. White paper, Limelight Networks, September 2018.

[62] Ben Niven-Jenkins, Francois Le Faucheur, and Nabil Bitar. Content Distribution Network Interconnection (CDNI) Problem Statement. Internet-draft, IETF, January 2012.

[63] Moslem Noori, Emina Soljanin, and Masoud Ardakani. On storage allocation for maximum service rate in distributed storage systems. In *2016 IEEE Internat. Symp. on Inform. Theory (ISIT)*, 2016.

[64] Erik Nygren, Ramesh K. Sitaraman, and Jennifer Sun. The akamai network: A platform for high-performance internet applications. *SIGOPS Oper. Syst. Rev.*, 44(3):2–19, August 2010.

[65] Samuel S. Ogden and Tian Guo. MODI: Mobile deep inference made efficient by edge computing. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[66] Venkata N. Padmanabhan and Kunwadee Sripanidkulchai. The case for cooperative networking. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS 01)*, 2002.

[67] Vinay Pai, Kapil Kumar, Karthik Tamilmani, Vinay Sambamurthy, and Alexander E. Mohr. Chainsaw: Eliminating trees from overlay multicast. In *4th International Conference on Peer-to-Peer Systems (IPTPS 05)*, 2005.

[68] KyoungSoo Park and Vivek S. Pai. Scale and performance in the CoBlitz large-file distribution service. In *3rd Conference on Networked Systems Design & Implementation (NSDI 06)*, 2006.

[69] Pei Peng and Emina Soljanin. On distributed storage allocations of large files for maximum service rate. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 784–791. IEEE, 2018.

[70] Ryan S. Peterson and Emin Gün Sirer. Antfarm: Efficient content distribution with managed swarms. In *6th USENIX Symposium on Networked Systems Design and Implementation (NSDI 09)*, 2009.

[71] Michael Piatek, Tomas Isdal, Arvind Krishnamurthy, and Thomas Anderson. One Hop Reputations for Peer to Peer File Sharing Workloads. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI 08)*, 2008.

[72] Arun Ravindran and Anjus George. An edge datastore architecture for latency-critical distributed machine vision applications. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[73] Arun Ravindran and Anjus George. An edge datastore architecture for latency-critical distributed machine vision applications. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[74] Xiaoqi Ren, Ganesh Ananthanarayanan, Adam Wierman, and Minlan Yu. Hopper: Decentralized speculation-aware cluster scheduling at scale. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 379–392. ACM, 2015.

[75] Sean Rhea, Patrick Eaton, Dennis Geels, Hakim Weatherspoon, Ben Zhao, and John Kubiatowicz. Pond: The OceanStore prototype. In *2Nd USENIX Conference on File and Storage Technologies (FAST 03)*, 2003.

[76] Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiatowicz. Handling churn in a DHT. In *USENIX Annual Technical Conference (ATC 04)*, 2004.

[77] Rodrigo Rodrigues and Peter Druschel. Peer-to-peer systems. *Communications of the ACM*, 53(10):72–82, 2010.

[78] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 01)*, 2001.

[79] M. Sardari, R. Restrepo, F. Fekri, and E. Soljanin. Memory allocation in distributed storage networks. In *2010 IEEE Internat. Symp. on Inform. Theory (ISIT)*, 2010.

[80] Stefan Saroiu, Krishna P. Gummadi, Richard J. Dunn, Steven D. Gribble, and Henry M. Levy. An analysis of

Internet content delivery systems. In *5th Symposium on Operating Systems Design and implementation (OSDI 02)*, 2002.

[81] Mahadev Satyanarayanan. The emergence of edge computing. *Computer*, 50(1):30–39, 2017.

[82] Guobin Shen, Ye Wang, Yongqiang Xiong, Ben Y Zhao, and Zhi-Li Zhang. HPTP: Relieving the tension between ISPs and P2P. In *6th International workshop on Peer-To-Peer Systems (IPTPS 07)*, 2007.

[83] R. Sherwood, R. Braud, and B. Bhattacharjee. Slurpie: a cooperative bulk data transfer protocol. In *IEEE INFOCOM 2004*, 2004.

[84] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[85] Weisong Shi and Schahram Dustdar. The promise of edge computing. *Computer*, 49(5):78–81, 2016.

[86] Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, and Mahadev Satyanarayanan. Scalable crowd-sourcing of video from mobile devices. In *11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys 13)*, 2013.

[87] Michael Sirivianos, Jong Han Park, Xiaowei Yang, and Stanislaw Jarecki. Dandelion: Cooperative content distribution with robust incentives. In *USENIX Annual Technical Conference (USENIX ATC 07)*, 2007.

[88] Karolj Skala, Davor Davidovic, Enis Afgan, Ivan Sovic, and Zorislav Sojat. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open Journal of Cloud Computing (OJCC)*, 2(1):16–24, 2015.

[89] Hui Suo, Jiafu Wan, Caifeng Zou, and Jianqi Liu. Security in the internet of things: a review. In *2012 international conference on computer science and electronics engineering*, volume 3, pages 648–651. IEEE, 2012.

[90] Nisha Talagala, Swaminathan Sundararaman, Vinay Sridhar, Dulcardo Arteaga, Qianmei Luo, Sriram Subramanian, Sindhu Ghanta, Lior Khermosh, and Drew Roselli. ECO: Harmonizing edge and cloud with ML/DL orchestration. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[91] Minoru Uehara. Mist computing: linking cloudlet to fogs. In *International Conference on Computational Science/Intelligence & Applied Informatics*, pages 201–213. Springer, 2017.

[92] Ernesto Van der Sar. Spotify starts shutting down its massive P2P network. TorrentFreak, April 2014.

[93] Deepak Vasisht, Zerina Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. Farmbeats: An IoT platform for data-driven agriculture. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017.

[94] Dan S Wallach. A survey of peer-to-peer security issues. In *International symposium on software security*, 2002.

[95] Kevin Walsh and Emin Gün Sirer. Experience with an object reputation system for peer-to-peer filesharing. In *3rd Conference on Networked Systems Design & Implementation (NSDI 06)*, 2006.

[96] Yingwei Wang. Definition and categorization of dew computing. *Open Journal of Cloud Computing (OJCC)*, 3(1):1–7, 2016.

[97] Zack Whittaker. Skype ditched peer-to-peer supernodes for scalability, not surveillance. Zero Day, June 2013.

[98] Xiaopei Wu, Robert Dunne, Qingyang Zhang, and Weisong Shi. Edge computing enabled smart firefighting: opportunities and challenges. In *5th ACM/IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb 17)*, 2017.

[99] Gala Yadgar, Michael Factor, and Assaf Schuster. Cooperative caching with return on investment. In *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST 13)*, 2013.

[100] Neeraja J Yadwadkar and Wontae Choi. Proactive straggler avoidance using machine learning. *White paper, University of Berkeley*, 2012.

[101] Matei Zaharia, Andy Konwinski, Anthony D Joseph, Randy H Katz, and Ion Stoica. Improving MapReduce performance in heterogeneous environments. In *8th USENIX Conference on Operating Systems Design and Implementation (OSDI '08)*, 2008.

[102] Shizhe Zang, Wei Bao, Phee Lep Yeoh, Branka Vucetic, and Yonghui Li. Paying less for more? combo plans for edge-computing services. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[103] Mingchen Zhao, Paarijaat Aditya, Ang Chen, Yin Lin, Andreas Haeberlen, Peter Druschel, Bruce Maggs, Bill Wishon, and Miroslav Ponec. Peer-assisted content distribution in Akamai NetSession. In *2013 Conference on Internet Measurement Conference (IMC 13)*, 2013.

[104] Wenting Zheng, Ankur Dave, Jethro G Beekman, Raluca Ada Popa, Joseph E Gonzalez, and Ion Stoica. Opaque: An oblivious and encrypted distributed analytics platform. In *14th USENIX Symposium on*

Networked Systems Design and Implementation (NSDI 17), 2017.

[105] Sheng Zhong, Jiang Chen, and Yang Richard Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc networks. In *IEEE International Conference on Computer Communications (INFOCOM)*, 2003.