# Machine-Learning based IoT Data Caching

Marc-Oliver Pahl, Stefan Liebald, Lars Wüstrich
Technische Universität München, Email: {pahl, liebald, wuestrich}@s2o.net.in.tum.de

***Abstract*—The Internet of Things (IoT) continuously produces big amounts of data. Data-centric middleware can therefore help reducing the complexity when orchestrating distributed Things. With its heterogeneity and resource limitations, IoT applications can lack performance, scalability, or resilience. Caching can help overcoming the limitations.**

**We are currently working on establishing data caching within IoT middleware. The paper presents fundamentals of caching, major challenges, relevant state of the art, and a description of our current approaches. We show directions of using machine learning for caching in the IoT.**

***Index Terms*—Data-centric, Internet of Things, caching, machine learning**

## I. Introduction

The Internet of Things (IoT) consists of distributed data sources and sinks. The interacting components are software services. They federate dynamically for implementing complex applications such as a heating control [1].

The central element of the IoT is data discovery and exchange. The availability of data, e.g. from sensors, is crucial for the operation of the IoT. Missing data can lead to a decrease in functionality, e.g. when a heating system does not know the room temperature, it cannot regulate the temperature suitably. More severe, missing data can also lead to safety, e.g. smoke detectors, and security threats, e.g. window sensors.

**Providing IoT data timely and continuous is a central challenge.** This challenge grows with the IoT's pervasion of the world. Timeliness can be affected by slow links and multiple network hops between services. The continuity can be affected by link or node failures. Both factors can be affected by a high rate of concurrent accesses to single data sources.

The IoT is a network of heterogeneous nodes that are connected over heterogeneous links. Due to the heterogeneity *bottlenecks* or even *failures* happen more likely than in classic, more homogeneous systems. Bottlenecks impact the *performance* of IoT systems: workflows take longer than expected as data cannot be exchanged in an ideal way. An example is pressing a physical light switch and having to wait seconds until the room gets alighted. Failures impact the *resilience* of IoT systems: workflows cannot be executed anymore. An example is closing the shutters when it gets dark, which depends on the availability of a light sensor.

Bottlenecks can also emerge from popularity that results in frequent accesses to certain services. Data access bottlenecks can therefore also impact the *scalability* of an IoT system. Finally, the dynamic access patterns that happen through the diversity of devices and use cases can affect the *energy efficiency* of an IoT system.

A newer paradigm for orchestrating the IoT is using reusable microservices to implement scenarios [1], [2]: Complex tasks are divided into multiple reusable microservices that get dynamically mashed-up at runtime [3], [4]. Data-centric middleware enables such mash-ups [3]–[5].

All previously identified problems happen through *insufficient availability of IoT data*. A suitable mitigation strategy for them is **data caching** [6]. Caching is most efficient as a core building block of the used communication middleware. We are currently working on this.

In this work, we introduce fundamental caching principles for the IoT. We detail our current approaches for using machine learning to improve the cache efficiency. The resulting increase in performance, resilience, scalability, and energy efficiency can increase the robustness of IoT systems significantly. This is especially relevant as IoT software systems more and more replace classic hardware controls, e.g. in heating systems, or dedicated systems such as alarms.

The failure of IoT systems can therefore have fatal consequences in all domains from private environments over hospitals to factories. Our work on IoT data caching therefore contributes to *increased security and safety of IoT systems*.

Section II introduces challenges of caching algorithms, putting them in the context of the IoT. Section III presents relevant state of the art. Section IV introduces our approach for introducing caching to a data-centric IoT.

## II. Caching Strategies and Challenges

Caching is utilized since the early days of computing, mainly for increasing *performance* [6]. There, memory hierarchies with caches enhance the local data access performance. For distributed systems, caches offer additional benefits. *Scalability* can be increased by storing data on multiple or faster nodes. *Resilience* can be increased by storing data on more or more reliable nodes [7].

The IoT is a complex distributed system with heterogeneous compute nodes, links, and data. Still, the fundamental challenges of caching are equally relevant for it. In this section we introduce basic principles and challenges of caching.

A fundamental problem with caching is a *limited cache size*. In case of the IoT, available storage on a node is limited mainly for price and energy reasons. Due to their limited availability, cache resources have to be managed suitably. Cache management has two main components, the *cache decision* and the *cache replacement strategy*.

Which data should be cached for optimizing a system is determined by the *cache decision strategy*. Though it is fundamental, there is not much research about caching decision strategies [8]. Instead, often all data passing through a node is added to its cache.

Support: German Federal Ministry of Economic Affairs and Energy (BMWI) in DECENT (0350024A); German-French Academy SCHEIF.

*Cache replacement strategies* are needed when the size of a new data object exceeds the amount of free space left in the cache. Consequently, content of the cache has to be replaced. It is desirable to delete items that are least-likely needed again in the future. As this is usually not known in advance, items are replaced based on a *usage prediction*.

For the prediction of the likelihood of future use, different metrics can be applied. They can be categorized into *recency of access* on an item, *frequency of access* on an item, *function based*, and *random* [9], [10].

Another issue with copies of information is *consistency*. Cached data is inconsistent when its source is updated but the cached copy is not. Such inconsistencies are especially relevant for the IoT. There, data is distributed, continuously generated, and often subject to change (e.g. sensor readings). Therefore, the IoT needs mechanisms to ensure *cache consistency*.

There are two main cache decision strategies: reactive and proactive caching. With *reactive caching*, data is cached after it was queried. A typical example is a web proxy. With *proactive* or *predictive caching*, data is cached before it is requested. The base is a prediction on which data is likely requested soon.

An optimal cache always has the requested data cached when needed. The cache *hit ratio* then is 100%. The IoT is a dynamically changing environment. This makes pushing the cache hit ratio towards 100% especially challenging.

## III. State of the Art

The IoT is a distributed system. Therefore, we identified *Information Centric Networking* (ICN) and *web caching* as the two most promising technologies.

ICN offers a disruptive (Inter-) networking architecture with a focus on data. Caching data on each node is a core principle of ICN [11].Web caching schemes on the other hand are widely deployed in the Internet since its early days, as they became required for handling the rapid growth of Internet resources [12]. The IoT also has to handle large amounts of data, e.g. periodic sensor readings.

Common to both technologies and the IoT is the focus on semantically tagged data. In ICN identifiers address data chunks. In the WWW Unified Resource Identifiers (URI) take this role. In the IoT, with suitable middleware the same principles apply [3], [5], [13]–[15]. We believe it is promising to carry parts of the caching approaches from both, web caching and ICN, over to the IoT.

For handling the complex, dynamically changing data exchange behavior of IoT services, we apply machine learning on caching. Therefore, we also survey relevant work following this approach.

*Information Centric Networking* aims to replacing host-based routing with content-based routing in the Internet. ICN assumes a distributed multi-hop architecture. Data is generated at data sources in the network, e.g. a sensor in an IoT scenario. It then travels through multiple intermediate hops until it reaches the data sink. ICN nodes have a content store that caches data reactively [16], [17].

A multitude of ICN caching approaches have been evaluated over time, differing in caching decision strategy, cache location or cache collaboration [18]. These approaches vary from caching everything to selecting the data that should be cached based on a calculated likelihood (prediction) for an item [19].

ICN caching often implements *on-path caching*: the reply of a request is kept in the local content store to answer similar requests in the future [16]. Alternatively, *off-path caching* is used: dedicated nodes in the network are used for caching in order to utilize each nodes cache more efficiently and to reduce redundancy.

An approach that makes use of queries to physically close devices is proposed in [20]. A history-based popularity index for data locations is calculated, successfully improving the deployed caching mechanism. and achieving high cache hit ratios for popular areas by prefetching data from these regions.

Data-centric IoT systems can also be designed as a peer-to-peer architectures with autonomously federating nodes [1], [5]. However, the IoT has specific data characteristics. Often data chunks are small and update frequently. The IoT is highly dynamic with frequently changing data sources and sinks.

Especially IoT sensor readings require a high cache recency. This is relevant for addressing the *consistency* issue. Some implementations solve this problem by introducing a *time to live* (TTL), indicating how long data from a cache can be used to answer queries [20].

The authors of [21] present caching in an ICN-based IoT scenario. Similarly, but without ICN, the authors of [22] look at sensor networks. Different to us, both look at resource constraint mobile devices with a focus on energy efficiency. In addition, they do not employ machine learning.

[10], [23] focus on ICN-specific mechanisms, including the flooding of an IoT system with information. We do not consider this aspect of ICN as suitable for the IoT, as it requires too much resources. Instead we focus on the data-centric addressing scheme with targeted caching in our work.

An older, but similarly related area for IoT caching are *web caches*. Web caching schemes have proven their scalability [12]. By strategically placing caches between a consumer and a data source, different goals of caching (i.e. latency reduction, increase of availability) can be achieved [24], [25].

For Web Caches, schemes utilizing machine learning exist. Depending on the goal, different approaches have been taken. Often neural networks improve existing *cache replacement strategies* like LRU or LFU [26], [27].

For proactive caching, some approaches propose the use of *log mining* in order to detect correlations between data accesses, and associations between data sources and sinks. It helps to *prefetch* data accordingly, improving the *Cache Decision Strategy* [28]–[30].

ICN and web caching both deal with annotated data. IoT middleware can also provide detailed data semantics [15]. In addition, many IoT applications have recurring data exchange behavior such as periodic sensor queries. This makes it a promising domain for caching.

A strength of machine learning is predicting complex correlations. For learning these correlations, annotated data is helpful. Therefore we consider combining both, machine learning and caching, highly promising. We build on the existing state of the art to improve caching in IoT systems.

## IV. Approach

Our goal is introducing caching as a central building block for the IoT. Middleware channels the data exchange between IoT services. Introducing caching functionality in IoT middleware is therefore promising to reach our goal.

Especially promising is data-centric middleware that does not only enable data communication between IoT services, but manages data on behalf of services [5], [14], [31], [32]. For illustrating our approach, we use the data-centric Virtual State Layer middleware (VSL). It is strong in managing IoT data [15], and it enables a dynamic discovery [3] and coupling [4] of IoT services.

In addition, the VSL provides persistent data items. Each sensor reading has a unique VSL ID. This facilitates caching as cached items are never incoherent. However, the problem is only shifted to knowing what is the most recent VSL data item as it may not have been propagated to a cache yet [5].

The VSL enforces tagging data items with type and functional identifiers [15]. This facilitates the analysis and decorrelation of inter-service communication [33], and can be useful for applying machine learning.

Relevant though not in the focus here, in addition the VSL provides security mechanisms such as access control and transport encryption directly in its core [34], [35]. Such security-by-design [36] is important as the IoT inherently processes privacy-critical personal data. By including caching in the VSL, the security mechanisms can directly be used for data protection and access control of cached data.

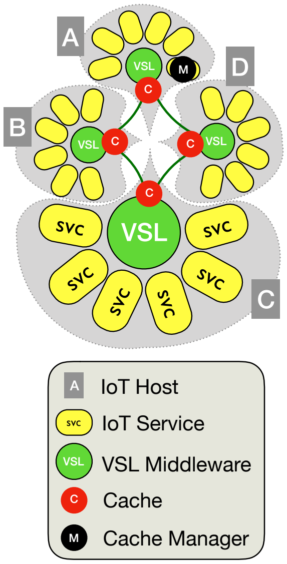

Fig. 1. IoT system with the VSL middleware and data caches.

Figure 1 shows an IoT site with many microservices that mash-up for implementing complex workflows. A complex example workflow is an intelligent climate control that orchestrates different services on distributed IoT hosts.

The illustration in fig. 1 consists of four IoT compute nodes (A-D). All nodes run multiple IoT (micro-) services (yellow on the outside). Each service is connected to the VSL self-organizing peer-to-peer IoT middleware [1] that handles all inter-service data exchange [4].

In its current implementation, the VSL stores data only at the source IoT host. This results in strong consistency, but can oppose performance, resilience, and scalability (section I). Therefore, we add caching.

To reduce the dependencies between the IoT nodes, we install a *node-local cache* (C) at each IoT host that has enough resources. It caches data requested by locally running services on-path. See fig. 1.

The independent nodes only have a local view on the IoT system. Therefore, we introduce a *site-local cache manager* (M). It uses the knowledge about the available resources on each node and its communication characteristics. The central cache manager informs all decentralized caches about relevant site-global data, enabling them to optimize their strategies even more. This implements off-path caching.

To implement near-optimal caching for the IoT, we are currently developing and implementing

- different suitable caching strategies,
- a data exchange analyzer that classifies different inter-service communication settings,
- a chooser that selects the best caching strategy on a per-service-interaction level,
- an overall cache manager that orchestrates the other components taking the restrictions of the local IoT node such as resources like CPU, storage, and network bandwidth into account, and
- an IoT-site wide analyzer and optimizer that has a global view on the entire IoT system and passes partial views for the local optimizations to the independent nodes.

As novel aspect compared to traditional caching in distributed systems we introduce *machine learning* for predicting the behavior of services' data exchanges. Machine-to-Machine (M2M) communication often leads to periodic patterns that can be detected using machine learning [33]. We use machine learning in two ways:

1) Choosing a suitable caching strategy per data item
2) Improving the prediction of data accesses

For *choosing a suitable replacement strategy per data item* we want to use Deep Learning as a black box technology. We want to feed the communication endpoint descriptors and the service IDs into our Neural Network. The output will be a number that indicates the most suitable replacement strategy for the data item such as LRF or LFU.

For *improving the prediction of data accesses*, we want to use and improve our communication models [33]. The more detailed we can locally model the future behavior of another service, the better we can predict its future data accesses. This knowledge enables us to optimize our cache decision and replacement strategies.

The decoupling of inter-service communication from an IoT site instance, the gained models can be exchanged between local IoT nodes, and globally between sites [33].

For *evaluating* our approaches, we will compare standard strategies with our two innovative uses of machine learning. We will measure the performance, scalability, resilience, and energy efficiency in different scenarios.

## V. Conclusion

Introducing caching to the IoT is highly promising. Due to the characteristics of the IoT it can be expected to have a significant effect on the performance, scalability, and resilience of IoT systems. With the VSL IoT middleware we have context data at hand that provides us with a comfortable base for complex caching decisions. Our preliminary tests with the VSL IoT middleware are promising towards a significant improvement in caching performance.

In this paper we introduced challenges and possible strategies for introducing data caching in the IoT. We started with the basic caching methodologies (section II), the cache decision strategy, and the cache replacement strategy. For the relevant state of the art (section III) we focused on ICN and web caching. We presented our current and planned approach towards caching in a data-centric IoT. Finally, we introduced our planned use of machine learning for choosing the best caching strategy per data item, and for predicting node behavior better (section IV).

Through the increased performance, scalability, and resilience we hope to make future IoT systems readier for a real-world use, where timely data availability can be critical for life-relevant applications such as fire detection or the control of heavy machinery in Industrial Internet of Things approaches.

## References

[1] M.-O. Pahl, G. Carle, and G. Klinker, "Distributed Smart Space Orchestration," in *NOMS - Dissertation Digest*, 2016.

[2] D. Lu, D. Huang, A. Walenstein, and D. Medhi, "A Secure Microservice Framework for IoT," in *IEEE Symposium on Service-Oriented System Engineering (SOSE)*. IEEE, 2017.

[3] M.-O. Pahl and S. Liebald, "A modular distributed iot service discovery," in *International Symposium on Integrated Network Management (IM)*, Washington DC, USA, 2019.

[4] M.-O. Pahl, "Data-Centric Service-Oriented Management of Things," in *Integrated Network Management (IM), IFIP/IEEE International Symposium on*, Ottawa, Canada, 2015.

[5] M.-O. Pahl and S. Liebald, "Designing a Data-Centric internet of things," in *International Conference on Networked Systems (NetSys)*, Garching b. München, Germany, 2019.

[6] J. Mellor-Crummey, D. Whalley, and K. Kennedy, "Improving memory hierarchy performance for irregular applications using data and computation reorderings," *Internat. Journal of Parallel Programming*, 2001.

[7] C. Carvalho, "The gap between processor and memory speeds," in *Proc. of IEEE International Conference on Control and Automation*, 2002.

[8] G. Rossini and D. Rossi, "Evaluating ccn multi-path interest forwarding strategies," *Computer Commun.*, 2013.

[9] S. Podlipnig and L. Böszörmenyi, "A survey of web cache replacement strategies," *ACM Computing Surveys*, 2003.

[10] M. A. M. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "On the performance of caching and forwarding in information-centric networking for the iot," in *International Conference on Wired/Wireless Internet Communication*. Springer, 2015.

[11] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *5th international conference on Emerging networking experiments and technologies*. ACM, 2009.

[12] J. Wang, "A survey of web caching schemes for the Internet," *ACM SIGCOMM Computer Comm. Review*, 1999.

[13] M. Amadeo, C. Campolo, J. Quevedo, D. Corujo, A. Molinaro, A. Iera, R. L. Aguiar, and A. V. Vasilakos, "Information-centric networking for the internet of things: Challenges and opportunities," *IEEE Netw.*, 2016.

[14] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information Centric Networking in the IoT: Experiments with NDN in the Wild," in *1st ACM Conf. on Information-Centric Networking*, 2014.

[15] M.-O. Pahl and G. Carle, "Crowdsourced Context-Modeling as Key to Future Smart Spaces," in *NOMS*, 2014.

[16] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. B. Ohlman, "A Survey of Information-Centric Networking," *IEEE Communications Magazine*, no. July, 2012.

[17] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "A Survey of information-centric networking research," *IEEE Communications Surveys and Tutorials*, 2014.

[18] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Comm. Surveys and Tut.*, 2015.

[19] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *2nd edition of ICN workshop on Information-centric networking*. ACM, 2012.

[20] Z. Zhou, D. Zhao, X. Xu, C. Du, and H. Sun, "Periodic query optimization leveraging popularity-based caching in wireless sensor networks for industrial iot applications," *Mobile Networks and Applications*, 2015.

[21] M. Meddeb, A. Dhraief, A. Belghith, T. Monteil, K. Drira, and S. Alahmadi, "Cache Freshness in Named Data Networking for the Internet of Things," *The Computer Journal (Oxford)*, 2018.

[22] D. Niyato, D. I. Kim, P. Wang, and L. Song, "A novel caching mechanism for Internet of Things (IoT) sensing service with energy harvesting," *IEEE International Conf. on Communications (ICC)*, 2016.

[23] F. Song, Z. Y. Ai, J. J. Li, G. Pau, M. Collotta, I. You, and H. K. Zhang, "Smart collaborative caching for information-centric IoT in fog computing," *Sensors (Switzerland)*, 2017.

[24] S. Podlipnig and L. Böszörmenyi, "A survey of Web cache replacement strategies," *ACM Comp. Surveys*, 2003.

[25] G. Barish and K. Obraczke, "World wide web caching: Trends and techniques," *IEEE Comm. magazine*, 2000.

[26] H. ElAarag, "Web proxy cache replacement scheme based on backpropagation neural network," in *SpringerBriefs in Computer Science*, 2013.

[27] W. Tian, B. Choi, and V. V. Phoha, "An adaptive web cache access predictor using neural network," in *Developments in Applied Artificial Intelligence*. Berlin, Heidelberg: Springer, 2002.

[28] Q. Yang, H. H. Zhang, and T. Li, "Mining web logs for prediction models in www caching and prefetching," in *7th ACM SIGKDD international conf. on Knowledge discovery and data mining*. ACM, 2001.

[29] Q. Yang and H. H. Zhang, "Web-log mining for predictive web caching," *IEEE Transactions on Knowledge and Data Engineering*, 2003.

[30] W. Ali, S. M. Shamsuddin, and A. S. Ismail, "A survey of web caching and prefetching," in *International Journal of Advances in Soft Computing and its Applications*, 2011.

[31] M.-O. Pahl and G. Carle, "The Missing Layer - Virtualizing Smart Spaces," in *10th IEEE International Workshop on Managing Ubiquitous Comm.s and Services (MUCS, PerCom adjunct)*, San Diego, USA, 2013.

[32] V. Raychoudhury, J. Cao, M. Kumar, and D. Zhang, "Middleware for pervasive computing: A survey," *Pervasive and Mobile Comp.*, 2012.

[33] M.-O. Pahl and F.-X. Aubet, "All eyes on you: Distributed Multi-Dimensional IoT microservice anomaly detection," in *14th International Conf. on Network and Service Management (CNSM)*, Rome, Italy, 2018.

[34] M.-O. Pahl and L. Donini, "Giving iot edge services an identity and changeable attributes," in *International Symposium on Integrated Network Management (IM)*, Washington DC, USA, 2019.

[35] ——, "Securing IoT Microservices with Certificates," in *Network Operations and Management Symposium (NOMS)*, 2018.

[36] A. Cavoukian, "*Privacy by Design*: Leadership, Methods, and Results." *European Data Protection*, 2013.