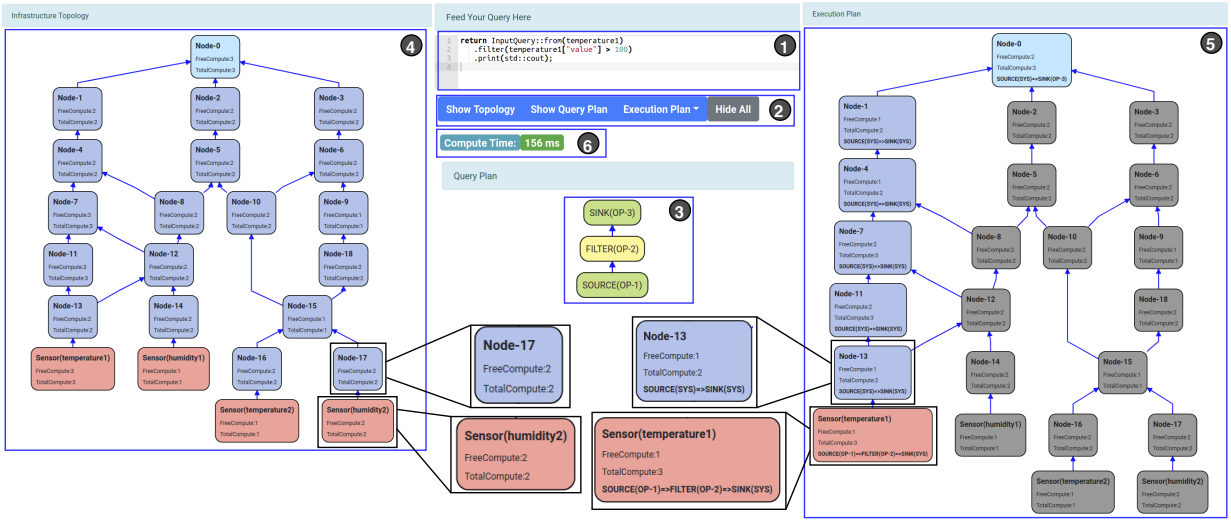

Figure 4: Demonstration system web interface.

in Table 1, i.e., Low-Latency, Fault-Tolerance, High-Throughput, Low Energy Consumption, and Low Resource Consumption. The display panel ⑤ will show the operator placement plan returned by Governor. The operator placement plan contains pinned, unpinned, as well as system-generated operators (e.g., forward operators) and their corresponding mappings on infrastructure nodes. Additionally, the time taken for the plan computation using the selected GP is shown in the UI ⑥.

Overall, attendees of our demonstration can explore the operator placement in a unified fog-cloud infrastructure. We will point out different options and their impact on particular characteristics. Using this demonstration, we present a first step towards operator placement designed for upcoming IoT infrastructures with millions of sensors and thousands of queries.

4 RELATED WORK

In this section, we categorize related work from different research areas. One line of research covers operator placement approaches for a centralized computing infrastructure. Jiang et al. offer a heuristic-based operator placement approach for optimizing with latency and throughput [8]. Kafil et al. consider heterogeneity within a distributed environment to find a optimum compute node to place all operators together [10]. Chatzistergiou et al. optimize the operator placement for inter-node network bottlenecks by reducing the number of nodes required to running a query [5]. In contrast, Governor allows a flexible and controlled operator placement strategy by enabling administrators to specify different optimization objectives within the same infrastructure.

Another line of research examines placement approaches for a unified fog-cloud environment. The approach presented by Veith et al. focuses mainly on minimizing total data processing latency [6]. However, this approach does not consider the node and link volatility within the fog as a factor that can significantly impact query execution. Cardellini et al. present a multi-objective operator placement approach [4]. However, their approach is not optimized for computing operator placements for a large scale IoT infrastructure. In contrast, Governor considers various characteristics of the fog-cloud infrastructure, including the high volatility that is commonly present in a fog infrastructure. Furthermore, Governor's two-phase approach using heuristics reduces the computation time for operator placement.

Overall, none of the previous approaches support defining flexible operator placement objectives. In contrast, Governor allows the administrator to specify different optimization goals using GPs for each query submission.

5 CONCLUSION

In this demonstration, we present Governor, a first step towards operator placement on IoT infrastructures with millions of sensors and thousands of queries. We demonstrate Governor's ability to accept custom Governor Policies with different optimization objectives for operator placement. In addition, we present five example policies and showcase their placement for five example application scenarios. In our demonstration, we will present the challenges as well as early solutions for operator placement in the future IoT era.

6 ACKNOWLEDGEMENT

This work was funded by the German Ministry for Education and Research as BIFOLD - Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref 01IS18037A).

REFERENCES

[1] Bonomi et al. 2012. Fog computing and its role in the internet of things. In *MCC*. ACM.
[2] Carbone et al. 2015. Apache flink: Stream and batch processing in a single engine. *IEEE Computer Society TCDE* 36, 4.
[3] Cardellini et al. 2016. Optimal operator placement for distributed stream processing applications. In *DEBS*. ACM.
[4] Cardellini et al. 2017. Optimal operator replication and placement for distributed stream processing systems. *SIGMETRICS* 44, 4 (2017).
[5] Chatzistergiou et al. 2014. Fast heuristics for near-optimal task allocation in data stream processing over clusters. In *CIKM*. ACM.
[6] da Silva Veith et al. 2018. Latency-aware placement of data stream analytics on edge computing. In *ICSOC*. Springer.
[7] de Assuncao et al. 2018. Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. *Journal of Network and Computer Applications* 103 (2018).
[8] Huang et al. 2011. Operator placement with QoS constraints for distributed stream processing. In *CNSM*. IEEE.
[9] Jaspal et al. 1993. Exploiting task and data parallelism on a multicomputer. In *ACM SIGPLAN Notices*, Vol. 28. ACM.
[10] Kafil et al. 1998. Optimal task assignment in heterogeneous distributed computing systems. *IEEE concurrency* 6, 3 (1998).
[11] O'Keeffe et al. 2018. Frontier: Resilient edge processing for the internet of things. In *VLDB*.
[12] Zaharia et al. 2012. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *NSDI*. USENIX Association.
[13] Zeuch et al. 2020. The NebulaStream Platform: Data and Application Management for the Internet of Things. In *CIDR*.