

Byungchul Tak

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/6321654/publications.pdf>

Version: 2024-02-01

14
papers

126
citations

1684188

5
h-index

1372567

10
g-index

14
all docs

14
docs citations

14
times ranked

111
citing authors

#	ARTICLE	IF	CITATIONS
1	BlackEye: automatic IP blacklisting using machine learning from security logs. <i>Wireless Networks</i> , 2022, 28, 937-948.	3.0	9
2	Privacy-Aware Collaborative Task Offloading in Fog Computing. <i>IEEE Transactions on Computational Social Systems</i> , 2022, 9, 88-96.	4.4	13
3	Fragmented Task Scheduling for Load-Balanced Fog Computing Based on Q-Learning. <i>Wireless Communications and Mobile Computing</i> , 2022, 2022, 1-9.	1.2	13
4	A Comprehensive Empirical Study of Query Performance Across GPU DBMSes. , 2022, , .		1
5	NoSQL Database Performance Diagnosis through System Call-level Introspection. , 2022, , .		0
6	A Comprehensive Empirical Study of Query Performance Across GPU DBMSes. <i>Performance Evaluation Review</i> , 2022, 50, 51-52.	0.6	0
7	Block-Level Storage Caching for Hypervisor-Based Cloud Nodes. <i>IEEE Access</i> , 2021, 9, 88724-88736.	4.2	4
8	A Survey of IoT Stream Query Execution Latency Optimization within Edge and Cloud. <i>Wireless Communications and Mobile Computing</i> , 2021, 2021, 1-16.	1.2	3
9	Green Fog Planning for Optimal Internet-of-Thing Task Scheduling. <i>IEEE Access</i> , 2020, 8, 1224-1234.	4.2	19
10	Fossil: Efficient Latency Reduction in Approximating Streaming Sensor Data. <i>Sustainability</i> , 2020, 12, 10175.	3.2	3
11	Priolog: Mining Important Logs via Temporal Analysis and Prioritization. <i>Sustainability</i> , 2019, 11, 6306.	3.2	3
12	LADRA: Log-based abnormal task detection and root-cause analysis in big data processing with Spark. <i>Future Generation Computer Systems</i> , 2019, 95, 392-403.	7.5	24
13	EDGESTORE: A Single Namespace and Resource-Aware Federation File System for Edge Servers. , 2018, , .		6
14	Failure Diagnosis for Distributed Systems using Targeted Fault Injection. <i>IEEE Transactions on Parallel and Distributed Systems</i> , 2016, , 1-1.	5.6	28