Jeff Huang

List of Publications by Year
in descending order

Source: https://exaly.com/author-pdf/11726427/publications.pdf

Version: 2024-02-01

32
papers

706
citations

1307594
7
h-index

1125743
13
g-index

32
all docs

32
docs citations

32
times ranked

251
citing authors

#	Article	IF	Citations
1	Maximal sound predictive race detection with control flow abstraction. , 2014, , .		111
2	CLAP. , 2013, , .		92
3	LEAP. , 2010, , .		70
4	Stateless model checking concurrent programs with maximal causality reduction. , 2015, , .		64
5	Persuasive prediction of concurrency access anomalies. , 2011, , .		63
6	Maximal sound predictive race detection with control flow abstraction. ACM SIGPLAN Notices, 2014, 49, 337-348.	0.2	52
7	Maximal causality reduction for TSO and PSO. , 2016, , .		28
8	ECHO: instantaneous in situ race detection in the IDE. , 2016, , .		27
9	LEAN. , 2012, , .		21
10	LEAP. , 2010, , .		18
11	CLAP. ACM SIGPLAN Notices, 2013, 48, 141-152.	0.2	16
12	Scaling predictive analysis of concurrent programs by removing trace redundancy. ACM Transactions on Software Engineering and Methodology, 2013, 22, 1-21.	6.0	13
13	GPredict: Generic Predictive Concurrency Analysis. , 2015, , .		11
14	OMPRacer: A Scalable and Precise Static Race Detector for OpenMP Programs. , 2020, , .		11
15	RDIT: race detection from incomplete traces. , 2015, , .		10
16	Maximal causality reduction for TSO and PSO. ACM SIGPLAN Notices, 2016, 51, 447-461.	0.2	10
17	Debugging Concurrent Software: Advances and Challenges. Journal of Computer Science and Technology, 2016, 31, 861-868.	1.5	9
18	Verifying Synchronization for Atomicity Violation Fixing. IEEE Transactions on Software Engineering, 2016, 42, 280-296.	5.6	9

#	Article	IF	Citations
19	D4: fast concurrency debugging with parallel differential analysis. ACM SIGPLAN Notices, 2018, 53, 359-373.	0.2	9
20	SafeCheck: Safety Enhancement of Java Unsafe API. , 2019, , .		8
21	Understanding JavaScript Vulnerabilities in Large Real-World Android Applications. IEEE Transactions on Dependable and Secure Computing, 2020, 17, 1063-1078.	5.4	8
22	An Efficient Static Trace Simplification Technique for Debugging Concurrent Programs. Lecture Notes in Computer Science, 2011, , 163-179.	1.3	8
23	LEAN. ACM SIGPLAN Notices, 2012, 47, 451-466.	0.2	7
24	When threads meet events: efficient and precise static race detection with origins. , 2021, , .		7
25	Precise and maximal race detection from incomplete traces. ACM SIGPLAN Notices, 2016, 51, 462-476.	0.2	7
26	ReCBuLC: Reproducing Concurrency Bugs Using Local Clocks. , 2015, , .		6
27	Distributed Networked Real-Time Learning. IEEE Transactions on Control of Network Systems, 2021, 8, 28-38.	3.7	3
28	Execution privatization for scheduler-oblivious concurrent programs. ACM SIGPLAN Notices, 2012, 47, 737-752.	0.2	3
29	Towards Incremental Static Race Detection in OpenMP Programs. , 2018, , .		2
30	Using Local Clocks to Reproduce Concurrency Bugs. IEEE Transactions on Software Engineering, 2018, 44, 1112-1128.	5.6	1
31	FastCFI: Real-Time Control Flow Integrity Using FPGA Without Code Instrumentation. Lecture Notes in Computer Science, 2019, , 221-238.	1.3	1
32	Automated Use-After-Free Detection and Exploit Mitigation: How Far Have We Gone?. IEEE Transactions on Software Engineering, 2022, 48, 4569-4589.	5.6	1