

Antony L Hosking

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/4544147/publications.pdf>

Version: 2024-02-01

67
papers

2,640
citations

687363

13
h-index

526287

27
g-index

68
all docs

68
docs citations

68
times ranked

945
citing authors

#	ARTICLE	IF	CITATIONS
1	The DaCapo benchmarks. , 2006, , .		1,119
2	The DaCapo benchmarks. ACM SIGPLAN Notices, 2006, 41, 169-190.	0.2	189
3	Open nesting in software transactional memory. , 2007, , .		133
4	Wake up and smell the coffee. Communications of the ACM, 2008, 51, 83-89.	4.5	90
5	Nested transactional memory: Model and architecture sketches. Science of Computer Programming, 2006, 63, 186-201.	1.9	80
6	A multidisciplinary approach towards computational thinking for science majors. SIGCSE Bulletin, 2009, 41, 183-187.	0.1	71
7	Barriers. , 2004, , .		70
8	Schism. , 2010, , .		70
9	A comparative performance evaluation of write barrier implementation. , 1992, , .		58
10	A multidisciplinary approach towards computational thinking for science majors. , 2009, , .		57
11	Transactional Monitors for Concurrent Objects. Lecture Notes in Computer Science, 2004, , 518-541.	1.3	47
12	A transactional object calculus. Science of Computer Programming, 2005, 57, 164-186.	1.9	45
13	Protection traps and alternatives for memory management of an object-oriented language. , 1993, , .		33
14	Seed selection for successful fuzzing. , 2021, , .		32
15	Barriers reconsidered, friendlier still!. , 2012, , .		30
16	A Semantic Framework for Designer Transactions. Lecture Notes in Computer Science, 2004, , 249-263.	1.3	26
17	Object fault handling for persistent programming languages. , 1993, , .		24
18	Transparently Reconciling Transactions with Locking for Java Synchronization. Lecture Notes in Computer Science, 2006, , 148-173.	1.3	24

#	ARTICLE	IF	CITATIONS
19	Reducing generational copy reserve overhead with fallback compaction. , 2006, , .		19
20	The Truth, The Whole Truth, and Nothing But the Truth. ACM Transactions on Programming Languages and Systems, 2016, 38, 1-20.	2.1	19
21	Safe futures for Java. ACM SIGPLAN Notices, 2005, 40, 439-453.	0.2	18
22	Relaxing safely: verified on-the-fly garbage collection for x86-TSO. , 2015, , .		18
23	Partial redundancy elimination for access path expressions. Software - Practice and Experience, 2001, 31, 577-600.	3.6	17
24	Penumbra. , 2003, , .		16
25	Scheduling Hard Real-Time Garbage Collection. , 2009, , .		16
26	Hierarchical real-time garbage collection. , 2007, , .		15
27	Schism. ACM SIGPLAN Notices, 2010, 45, 146-159.	0.2	14
28	Software prefetching for mark-sweep garbage collection. , 2004, , .		13
29	Fine-grained adaptive biased locking. , 2011, , .		13
30	Scheduling real-time garbage collection on uniprocessors. ACM Transactions on Computer Systems, 2011, 29, 1-29.	0.8	13
31	Stop and go: understanding yieldpoint behavior. , 2015, , .		12
32	A comparative performance evaluation of write barrier implementation. ACM SIGPLAN Notices, 1992, 27, 92-109.	0.2	11
33	Mostly-copying reachability-based orthogonal persistence. , 1999, , .		11
34	Value-Based Partial Redundancy Elimination. Lecture Notes in Computer Science, 2004, , 167-184.	1.3	10
35	Portable, mostly-concurrent, mostly-copying garbage collection for multi-processors. , 2006, , .		9
36	Hierarchical real-time garbage collection. ACM SIGPLAN Notices, 2007, 42, 123-133.	0.2	7

#	ARTICLE	IF	CITATIONS
37	Impact of GC design on power and performance for Android. , 2015, , .		7
38	Rust as a language for high performance GC implementation. , 2016, , .		7
39	Pervasive Load-Time Transformation for Transparently Distributed Java. Electronic Notes in Theoretical Computer Science, 2009, 253, 47-64.	0.9	6
40	X10 on the single-chip cloud computer. , 2011, , .		6
41	Relaxing safely: verified on-the-fly garbage collection for x86-TSO. ACM SIGPLAN Notices, 2015, 50, 99-109.	0.2	6
42	Hybrid STM/HTM for nested transactions on OpenJDK. , 2016, , .		6
43	Protection traps and alternatives for memory management of an object-oriented language. Operating Systems Review (ACM), 1993, 27, 106-119.	1.9	5
44	Closed and open nested atomic actions for Java. , 2014, , .		5
45	Don't race the memory bus: taming the GC leadfoot. , 2015, , .		5
46	Software prefetching for mark-sweep garbage collection. Operating Systems Review (ACM), 2004, 38, 199-210.	1.9	4
47	Class Transformations for Transparent Distribution of Java Applications.. Journal of Object Technology, 0, 10, 9:1.	0.9	4
48	Anticipation-based partial redundancy elimination for static single assignment form. Software - Practice and Experience, 2004, 34, 1413-1439.	3.6	3
49	Hop, Skip, & Jump. , 2018, , .		3
50	Object fault handling for persistent programming languages. ACM SIGPLAN Notices, 1993, 28, 288-303.	0.2	3
51	Stop and go: understanding yieldpoint behavior. ACM SIGPLAN Notices, 2016, 50, 70-80.	0.2	3
52	Mostly-copying reachability-based orthogonal persistence. ACM SIGPLAN Notices, 1999, 34, 382-398.	0.2	2
53	Persistent object systems. Software - Practice and Experience, 2000, 30, 293-294.	3.6	2
54	Revocation techniques for Java concurrency. Concurrency Computation Practice and Experience, 2006, 18, 1613-1656.	2.2	2

#	ARTICLE	IF	CITATIONS
55	Barriers reconsidered, friendlier still!. ACM SIGPLAN Notices, 2013, 47, 37-48.	0.2	2
56	One Process to Reap Them All. , 2017, , .		2
57	Classifying Java class transformations for pervasive virtualized access. , 2009, , .		2
58	The locality of concurrent write barriers. , 2010, , .		2
59	Don't race the memory bus: taming the GC leadfoot. ACM SIGPLAN Notices, 2016, 50, 15-27.	0.2	2
60	Extending OpenJDK to support hybrid STM/HTM: preliminary design. , 2016, , .		1
61	A Framework for Persistence-Enabled Optimization of Java Object Stores. Lecture Notes in Computer Science, 2001, , 4-17.	1.3	1
62	Software prefetching for mark-sweep garbage collection. ACM SIGPLAN Notices, 2004, 39, 199-210.	0.2	0
63	One Process to Reap Them All. ACM SIGPLAN Notices, 2017, 52, 171-186.	0.2	0
64	Classifying Java class transformations for pervasive virtualized access. ACM SIGPLAN Notices, 2010, 45, 75-84.	0.2	0
65	The locality of concurrent write barriers. ACM SIGPLAN Notices, 2010, 45, 83-92.	0.2	0
66	Hybrid STM/HTM for nested transactions on OpenJDK. ACM SIGPLAN Notices, 2016, 51, 660-676.	0.2	0
67	Designing a low-level virtual machine for implementing real-time managed languages. , 2019, , .		0