

Stefan Marr

List of Publications by Year in descending order

Source: <https://exaly.com/author-pdf/1945057/publications.pdf>

Version: 2024-02-01

54
papers

326
citations

1937685

4
h-index

1474206

9
g-index

55
all docs

55
docs citations

55
times ranked

122
citing authors

#	ARTICLE	IF	CITATIONS
1	Naïve transient cast insertion isn't (that) bad. , 2021, , .		0
2	Fully Reflective Execution Environments: Virtual Machines for More Flexible Software. IEEE Transactions on Software Engineering, 2019, 45, 858-876.	5.6	0
3	Which of my transient type checks are not (almost) free?. , 2019, , .		4
4	A flexible framework for studying trace-based just-in-time compilation. Computer Languages, Systems and Structures, 2018, 51, 22-47.	1.4	0
5	Few versatile vs. many specialized collections: how to design a collection library for exploratory programming?. , 2018, , .		4
6	Efficient and deterministic record & replay for actor languages. , 2018, , .		15
7	Parallelization of dynamic languages: synchronizing built-in collections. , 2018, 2, 1-30.		7
8	Introduction to the Meta [™] 16 Workshop Special Issue.. Journal of Object Technology, 2018, 17, 1.	0.9	1
9	Cross-language compiler benchmarking: are we fast yet?. ACM SIGPLAN Notices, 2017, 52, 120-131.	0.2	2
10	Applying Optimizations for Dynamically-typed Languages to Java. , 2017, , .		4
11	A Metaobject Protocol for Optimizing Application-Specific Run-Time Variability. , 2017, , .		1
12	Kā̂î¼îcoĩ, , 2017, , .		5
13	Building efficient and highly run-time adaptable virtual machines. ACM SIGPLAN Notices, 2017, 52, 60-71.	0.2	0
14	Fully-reflective VMS for ruling software adaptation. , 2017, , .		1
15	A concurrency-agnostic protocol for multi-paradigm concurrent debugging tools. , 2017, , .		9
16	Garbage collection and efficiency in dynamic metacircular runtimes: an experience report. ACM SIGPLAN Notices, 2017, 52, 39-50.	0.2	0
17	A concurrency-agnostic protocol for multi-paradigm concurrent debugging tools. ACM SIGPLAN Notices, 2017, 52, 3-14.	0.2	0
18	Cross-language compiler benchmarking: are we fast yet?. , 2016, , .		35

#	ARTICLE	IF	CITATIONS
19	GEMs: shared-memory parallel programming for Node.js. , 2016, , .		2
20	Generic messages. , 2016, , .		0
21	Building efficient and highly run-time adaptable virtual machines. , 2016, , .		7
22	Domains: Sharing state in the communicating event-loop actor model. Computer Languages, Systems and Structures, 2016, 45, 132-160.	1.4	10
23	Efficient and thread-safe objects for dynamically-typed languages. , 2016, , .		18
24	Efficient and thread-safe objects for dynamically-typed languages. ACM SIGPLAN Notices, 2016, 51, 642-659.	0.2	3
25	GEMs: shared-memory parallel programming for Node.js. ACM SIGPLAN Notices, 2016, 51, 531-547.	0.2	2
26	Towards fully reflective environments. , 2015, , .		9
27	A formal foundation for trace-based JIT compilers. , 2015, , .		2
28	Just-in-time data structures. , 2015, , .		16
29	Partitioned Global Address Space Languages. ACM Computing Surveys, 2015, 47, 1-27.	23.0	36
30	Zero-overhead metaprogramming: reflection and metaobject protocols fast and without compromises. , 2015, , .		22
31	Domains: Safe sharing among actors. Science of Computer Programming, 2015, 98, 140-158.	1.9	4
32	Parallel gesture recognition with soft real-time guarantees. Science of Computer Programming, 2015, 98, 159-183.	1.9	0
33	Zero-overhead metaprogramming: reflection and metaobject protocols fast and without compromises. ACM SIGPLAN Notices, 2015, 50, 545-554.	0.2	1
34	Tracing vs. partial evaluation: comparing meta-compilation approaches for self-optimizing interpreters. , 2015, , .		22
35	Tracing vs. partial evaluation: comparing meta-compilation approaches for self-optimizing interpreters. ACM SIGPLAN Notices, 2015, 50, 821-839.	0.2	1
36	Fork/join parallelism in the wild. , 2014, , .		11

#	ARTICLE	IF	CITATIONS
37	Data interface + algorithms = efficient programs. , 2014, , .		2
38	Are We There Yet?: Simple Language Implementation Techniques for the 21st Century. IEEE Software, 2014, 31, 60-67.	1.8	9
39	Tanks. , 2013, , .		5
40	Cloud PARTE. , 2013, , .		2
41	Parallel gesture recognition with soft real-time guarantees. , 2012, , .		3
42	Synchronization views for event-loop actors. , 2012, , .		0
43	Modularity and conventions for maintainable concurrent language implementations. , 2012, , .		0
44	Synchronization views for event-loop actors. ACM SIGPLAN Notices, 2012, 47, 317-318.	0.2	0
45	Identifying a Unifying Mechanism for the Implementation of Concurrency Abstractions on Multi-language Virtual Machines. Lecture Notes in Computer Science, 2012, , 171-186.	1.3	5
46	Which problems does a multi-language virtual machine need to solve in the multicore/manycore era?. , 2011, , .		0
47	Many-core virtual machines. , 2010, , .		0
48	The SOM family. , 2010, , .		14
49	Encapsulation and locality. , 2010, , .		1
50	Insertion Tree Phasers: Efficient and Scalable Barrier Synchronization for Fine-Grained Parallelism. , 2010, , .		3
51	Towards an actor-based concurrent machine model. , 2009, , .		15
52	Intermediate language design of high-level language virtual machines. , 2009, , .		7
53	Towards Composable Concurrency Abstractions. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 155, 54-60.	0.8	2
54	Virtual Machine Support for Many-Core Architectures: Decoupling Abstract from Concrete Concurrency Models. Electronic Proceedings in Theoretical Computer Science, EPTCS, 0, 17, 63-77.	0.8	4