

7. REFERENCES

- [1] J. Jones, “Abstract syntax tree implementation idioms,” in *Proceedings of the 10th Conference on Pattern Languages of Programs (PLoP2003)*, 2003.
- [2] G. Hohpe and B. Woolf. “Enterprise integration patterns”. In *9th Conference on Pattern Language of Programs*, pages 1–9, 2002.
- [3] Andreas Haas, Andreas Rossberg, Derek L. Schuff, Ben L. Titzer, Michael Holman, Dan Gohman, Luke Wagner, Alon Zakai, and JF Bastien. “Bringing the web up to speed with WebAssembly”. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2017.
- [4] N. Maurer and M. Wolfthal, “Netty in Action”. Manning Publications, 2016.
- [5] Terence Parr and Russell Quong. “ANTLR: A predicated-LL(k) parser generator.” *Journal of Software Practice and Experience*, 25(7), 1995.
- [6] C. Lattner, “Introduction to the LLVM Compiler Infrastructure,” in *Itanium Conference and Expo*, April 2006.
- [7] J. Pombrio and S. Krishnamurthi. “Resugaring: Lifting evaluation sequences through syntactic sugar”. In *Programming Languages Design and Implementation*, 2014.
- [8] C. Lattner. “LLVM and Clang: Next Generation Compiler Technology”. In *The BSD Conference*, May 2008.
- [9] C. Lattner and V. Adve, “LLVM: A compilation framework for lifelong program analysis and transformation,” in *Proc. CGO*, 2004,
- [10] Gosling, J. “Java Intermediate Bytecodes”. in *SIGPLAN Workshop on Intermediate Representations (IR95)*. 1995.
- [11] D. Chisnall. “The Challenge of Cross-language Interoperability.” *Commun. ACM*, 2013.
- [12] Y. Shi, K. Casey, M. A. Ertl, and D. Gregg. “Virtual machine showdown: Stack versus registers.” *ACM Transactions on Architecture and Code Optimization*, 2008.
- [13] M. Wilde, M. Hategan, J.M. Wozniak, B. Clifford, D.S. Katz, I. Foster, “Swift: a language for distributed parallel scripting,” *Parallel Computing*, 2011.
- [14] W. Binder, J. Hulaas, and P. Moret. “Advanced java bytecode instrumentation.” In *Proc. of the International Symposium on Principles and Practice of Programming in Java*, 2007

- [15] B. Venners, “Inside the Java Virtual Machine, 2nd Edition”, McGraw Hill, 1999
- [16] A.-R. Ald-Tabatabai, M. Cierniak, G.-Y. Lueh, V. M. Parikh, and J. M. Stichnoth, “Fast, Effective Code Generation in a Just-in-Time Java Compiler,” *Proceedings, SIGPLAN '98 Conference on Programming Language Design and Implementation*, 1998.
- [17] “The Java® Virtual Machine Specification” [Online]. Available : <https://docs.oracle.com/javase/specs/jvms/se8/html/index.html> [Accessed: 16-March-2019].
- [18] JVM Compiler Backend Benchmark [Online]. Available : https://github.com/Kishanthan/ballerina/tree/bir_jvm_benchmark/benchmarks/bir-jvm-benchmark/src/main/java/org/ballerinalang [Accessed: 16-March-2019].
- [19] Ballerina Programming Language [Online]. Available : <https://ballerina.io/> [Accessed: 16-March-2019]
- [20] Ballerina Language Specification [Online]. Available : <https://ballerina.io/res/Ballerina-Language-Specification-v0.990-2019-01-16.pdf> [Accessed: 16-March-2019]
- [21] Ballerina Compiler Architecture [Online]. Available : <https://github.com/ballerina-platform/ballerina-lang/blob/master/docs/compiler/compiler-architecture.md> [Accessed: 16-March-2019]
- [22] Gary L. Schaps, “Compiler construction with antlr and java,” *Dr. Dobb's Journal*, 1999
- [23] Joe Groff and Chris Lattner. “Swift’s High-Level IR: A Case Study of Complementing LLVM IR with Language-Specific Optimization.” *2015 LLVM Developers’ Meeting*, 2015.
- [24] E. Bruneton. Asm 3.0, a java bytecode engineering library [Online]. Available : <https://asm.ow2.io/asm4-guide.pdf> [Accessed: 16-March-2019]
- [25] JVM Compiler Backend Benchmark Test Results [Online]. Available : <https://goo.gl/96W93J> [Accessed: 16-March-2019]