GraalVM, a polyglot multi-platform and multi-architecture VM, combines more than a decade of research in compiler optimizations, language compositions and ahead-of-time (AOT) compilation into a single high-performance JDK distribution. In this presentation we will take an overview look into GraalVM covering its polyglot nature and deployment scenarios before we take a quick but deep dive into the heart of the VM: the compiler. We will discuss its graph-based intermediate representation, the compilation pipeline as well as its optimization plan. The compiler is utilized to produce high performance machine code for a multitude of programming languages and offers additional capabilities for AOT compilation in native image. We will cover the general layout of the compiler before taking one of this year's internship projects to explain interesting research challenges we want our future interns to tackle. Interested candidates will get a feeling for the large impact of their work. This year we offer multiple 3 to 6 months internships in the entire GraalVM project, ranging from compiler optimization work to language implementation and code generation tasks. Interns can work on bleeding edge research questions in a production quality VM supervised by a global team of industry and domain experts.

Graph processing is an invaluable tool for data analytics. In particular, pattern-matching queries enable flexible graph exploration and analysis, similar to what SQL provides for relational databases. Graph queries focus on following connections in the data; they are a challenging workload because even seemingly trivial queries can easily produce billions of intermediate results and irregular data access patterns. In this presentation, we will describe Oracle Labs PGX.D/aDFS [published in USENIX ATC'21]: A distributed graph-querying system that can process practically any query fully in memory, while maintaining bounded runtime memory consumption. To achieve this behavior, PGX.D/aDFS relies on (i) almost depth-first (aDFS) graph exploration with some breadth-first characteristics for performance, and (ii) non-blocking dispatching of intermediate results to remote edges. Additionally, we will briefly describe recent machine-learning (ML) developments in Oracle Labs, including highlights from our graph and autonomous ML projects.