Eclipse OMR 2021 Highlights & 2022 Project Goals

Feb 3, 2022

https://github.com/eclipse/omr/issues/2316

OMR Project

2021

- 434 PRs merged
- 2 new committers (Irwin D’Souza, Devin Papineau), 13 total
- 7 beginner issues completed

Compiler

2021

- Re-design vector IL opcodes
- Significant reduction in compiler warnings
  - Enable warnings as errors on X, Z
- AArch64 code generator improvements
  - SIMD instructions, arraycopy, many perf improvements
- Documentation enhancements
- Ongoing code generator convergence
  - Consistent IL opcode evaluator nomenclature
- Initial contribution and demo of JitBuilder 2.0
- Progress on defining compiler API
- Numerous technical debt reduction PRs

Port, GC, DevOps

2021

- Port library initialization API for compiler and JitBuilder (#5798)
- GC parallelism and Adaptive Threading (#5829)
- Python CI dependency (#5916)

Compiler

2022

- Vector/SIMD enhancements
  - Re-design vector IL opcodes + Tril unit tests
  - Intel AVX/AVX2/AVX-512 support
  - Complete support on Z, Power, AArch64
- Improved documentation
- Continue progress on JitBuilder 2.0
- Compiler API
- Large refactoring epics decomposition into manageable tasks

Infrastructure

2022

- Automatic copyright verification on commit (#2254)
- Extensible class linter to enforce compiler API
- Move remaining builds to CMake (#4618), deprecate autotools
- CI builds and testing
  - Acquire native AArch64 hardware for testing (Linux and macOS)
  - Acquire native RISC-V hardware for testing
- Incorporate Python dependency into DevOps pipeline