Utilizing ARACHNE runtime environments packaging know-how in preparation for running network studies

Konstantin Yaroshovets[1], Adam Black[2], Alexey Manoylenko[3], Gregory Klebanov[4]
[1, 3, 4] Odysseus, an EPAM company
[2] Erasmus Medical Center

Background

Reacting on a need to adapt to miscellaneous infrastructural requirements and an ability to rapidly reference necessary changes in runtime environments with their transparent distribution after ARACHNE Execution Engine 2.0.0[1] and its subsequent released versions extended a possibility to support multiple runtime environments allowing runtime environments being prepared as Docker images so that an analysis execution can be performed as a dedicated Docker container

During deployment an administrator configures if ARACHNE Execution Engine should work with runtime environments based on Docker images and then handle the appropriate requests from a client which specifies which Docker image should be explicitly used to run an analysis creating a separate Docker container for that. An analysis execution log available at runtime, realtime analysis execution status and a possibility to cancel a running execution are features being supported

HADES[2] packages are usually released each six months and an “umbrella” renv.lock file is distributed based on which a runtime environment is built and then available for ARACHNE Execution Engine. For instance, DARWIN EU®[3] has a need to have proprietary runtime environments which can be easily maintained and in case of changes requested modified by engineers so that they are available after for debugging and convenient analysis execution both in the underlying organization infrastructure and on an engineer’s host

Methods

The existing approach[4] of building a runtime environment as a tarball with an operating system, R and Java runtimes and R packages installed is supplemented with a possibility to additionally build a Docker image so that both techniques can be used depending on operational requirements. The build scripts are versioned so that a particular runtime environment can be built for a specific use case and retrospectively

Results

A version of ARACHNE Execution Engine starting from release 2.0.0 supports both configuration options when an analysis execution is performed by a schroot command or by a Docker container. The scripts which allow building runtime environments for both execution modes should be available for their further extensions on OHDSI GitHub. An additional system improvement has been implemented (after version 2.2.1) so that ARACHNE Execution Engine if configured correspondingly can serve requests from clients in the dual mode performing an analysis execution whether in a Docker container or in an isolated environment with the help of schroot depending on the client’s request parameters

Conclusion

A new 2.x series of ARACHNE Execution Engine gives a flexibility on choosing an approach how to build runtime environments and update them when necessary and not neglecting any functional requirements defined for a secured reproducible study execution. In some institutions allowing Docker might be cumbersome though when it is not so some time which is required to unpack a tarball could be already spent on an analysis execution. Though these aspects don’t sound like a major disadvantage of a particular mode sometimes it might be useful for a decision maker

An important topic for the future solution development should be standardization of approaches on how runtime environments are centrally stored and distributed even though an existing build-on-demand method is rather straightforward and less error prone

References

1. https://github.com/OHDSI/ArachneExecutionEngine
2. https://ohdsi.github.io/Hades/packages.html
3. https://www.darwin-eu.org/
4. https://github.com/odysseusinc/DockerEnv