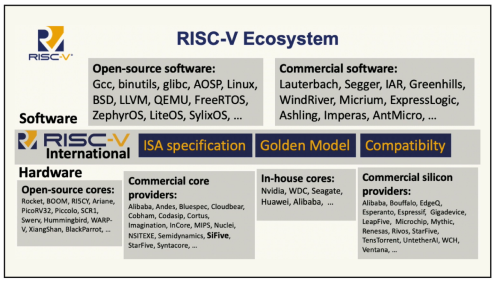
XUANTIE 玄铁

Leveraging Container Technology for Streamlined RISC-V ACT

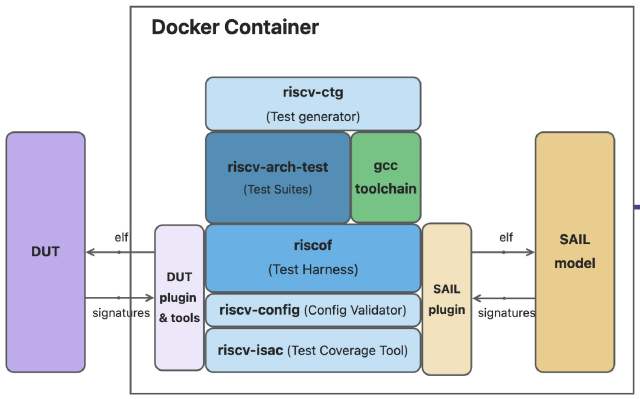
Author: Qinghao Shi, Alibaba Damo Academy
E-mail: shiqinghao.sqh@alibaba-inc.com

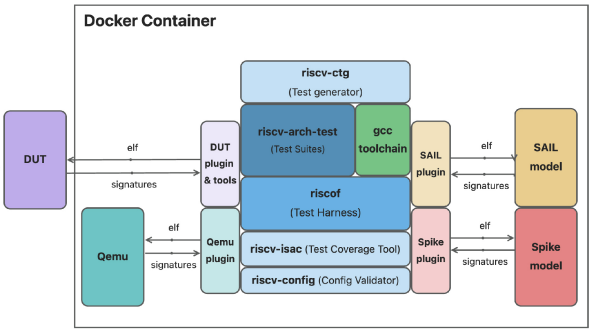
ACT (Architecture Compatibility Testing) plays a critical role in the development of the RISC-V ecosystem. As an endorsed compatibility testing standard by the RISC-V International, ACT provides a fundamental measuring tool for the vendors that implements RISC-V processors. With the growing number of RISC-V cpu implementers and the increasing number and complexity of RISC-V ISA extensions, ensuring a smooth running of ACT for numerous users to validate the defined Specifications has become a big challenge.

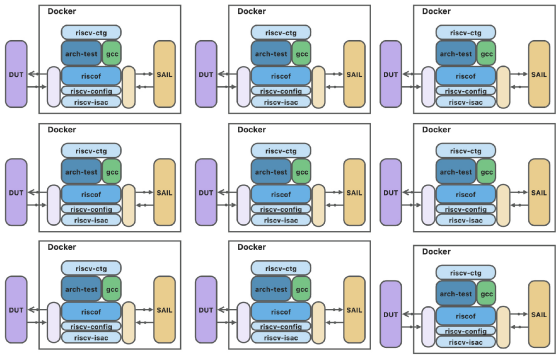


Major Challenges

- **Complexities in Test Environment Configuration:** Establishing a test environment for ACT presents considerable challenges. The intricate setup process poses significant barriers to entry for developers, and the complex dependence requirements likly to cause components incompatiability, thereby affecting the efficiency and efficacy of ACT adoption.
- **Inconsistency in Development Environments:** The absence of a standardized development framework for ACT has been identified as a major issue for replicating bugs across different systems, as differences in the environments configuration hinder the process of bug reproduction and resolution.
- **Deficiencies in ACT CI Infrastructure:** Currently lack of a reliable and scalable CI system dedicated to the verification of ACT test cases, this exposed big risks when ACT is in active development
- **Verification of Golden Reference Model:** There is a noticeable gap in methodologies conducive to the effective verification of the SAIL model. This deficiency undermines the capacity to authenticate the integrity and functionality of the SAIL model systematically.





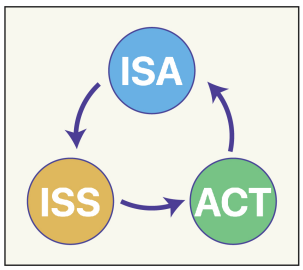


Solution

- **Unified Containerization Approach:** Consolidate all requisite components of ACT into one cohesive container. This strategy ensures the encapsulation of all necessary dependencies and components within a single, manageable entity, facilitating ease of deployment and dependency management.
- **Synchronized Release Protocol:** Concurrently with each release of ACT, a companion container will be published within the official repository. This container will encapsulate the latest version of ACT along with all associated components, thereby ensuring that users have access to a complete, ready-to-deploy ACT package.
- **Extensive Component Testing:** Prior to release, all components housed within the same container will undergo rigorous testing. This meticulous validation process is designed to ensure that all components function harmoniously and meet the predefined quality standards, thereby guaranteeing the reliability and robustness of the ACT within its operational environment.
- **Comprehensive Documentation Update:** Commit to maintaining thoroughly up-to-date documentation for all aspects of the ACT, including the latest functionalities, component integrations, and deployment methodologies. The aim is to provide clear, accurate, and accessible guidance to all end-users and developers, thereby enhancing user experience and facilitating easier adoption and troubleshooting.

Benefits

1. **Streamlined Deployment Process:** Achieve a significant reduction in manual intervention by condensing the process from over 30 steps to a mere 2, thereby substantially decreasing setup time and enhancing operational efficiency.
2. **Consistent Testing Environments:** Foster the creation of reproducible test environments that ensure reliability and consistency in testing outcomes across different instances and platforms.
3. **Guaranteed Compatibility:** Ensure seamless interoperability among all tools within the ecosystem, with rigorous version control checks in place to verify the compatibility of all integrated components.
4. **Facilitation of Continuous Integration:** Empower the establishment of a robust CI system for the ACT, thereby streamlining development workflows and enhancing product robustness through automated testing.
5. **Models Cross-Verification:** Enable extensive cross-verification across diverse emulators, ensuring that ACT systems function effectively and consistently, regardless of the underlying emulator technology.
6. **Scalable and Extensible Architecture:** Design the system with scalability and extensibility in mind, enabling future expansion to accommodate a wide range of services, such as microservices tailored for educational purposes or the development of an extensive ACT farm.



ACT Container

RISC-V Cloud Lab

For more information, please visit ***xrvm.com***