LHC MC WG Kickoff Meeting

Dmitri Konstantinov

14.11.2024

EP-SFT Group: Key Directions and Projects

- Simulation group: Geant4: Development and maintenance of particle interaction simulation toolkit (focused on LHC and CERN experiments
- ROOT group: Core framework for data storage, analysis, and visualization.
- CVMFS group: development of CVMFS
- Software Stacks group: LCG Stack: stack of software libraries and tools supporting experiments' needs

EP-SFT Stack Project Overview

- Over 830 packages across architectures and compilers, with nightly builds and updates orchestrated by Jenkins (CVMFS, RPMs).
- Regular LCG releases are aligned with ROOT versions, with tailored sub-releases for ATLAS, LHCb, and SWAN.
- Primary Users: ATLAS, LHCb, CMS analytics, CERN-BE ("NXCALS"), SWAN, NA61, NA62, and more.
- Nightly builds (except Sundays) with continuous updates based on user requests, enabling early detection of bugs and inconsistencies in the software stack.

Extending Validation to Development Versions of MC Generators

- Current Validation
  - EP-SFT keeps MC tools up-to-date by following MC project announcements and addressing requests from ATLAS and LHCb.
  - Currently, ATLAS and LHCb incorporate LCG nightly builds, with ATLAS primarily testing ATHENA integration without specific MC testing
- Proposed Extension of Validation
  - Build MC generators directly from the latest GitHub/GitLab development versions, ensuring access to the newest updates.
  - Encourage ATLAS and LHCb to extend their validation to these development versions, providing faster feedback to MC generator authors and enhancing early testing of new features (in a similar way as CMS tests and provides feedback to Geant4 using their monthly development builds).
  - EP-SFT is open to running relatively quick tests, provided by MC authors, that will not significantly impact build time.

Facilitating MCPLOTS Testing with a Dedicated LCG Layer

- Current Challenges for MCPLOTS
  - MCPLOTS currently cherry-picks specific MC generators from various LCG releases, limiting access to all desired versions.
  - Selecting generators from multiple releases complicates MCPLOTS testing workflows, making consistent testing across versions challenging.
- Proposed Solution: Dedicated MCPLOTS Layer
  - Create a dedicated MCPLOTS layer within the LCG stack, centralizing all required MC generator versions for easier access.
  - Ensure the MCPLOTS layers include both stable and development versions, enabling better testing across different stages of generator development.
  - Streamline MCPLOTS testing by aligning generator availability directly with their validation needs (SPI JIRA), improving efficiency and consistency.

Static Analysis and Runtime Tools for Quality Assurance

- Current Use in Geant4
  - Coverity: Comprehensive static analysis for detecting bugs and vulnerabilities in C++ code, helping Geant4 maintain high-quality code standards.
  - PVS-Studio and Clang Static Analysis: Additional static analysis tools that detect memory leaks, security vulnerabilities, and performance issues.
  - clang sanitizers: Runtime tools (e.g., AddressSanitizer, UndefinedBehaviorSanitizer) used to catch memory and undefined behavior issues during code execution
- Extending Analysis and Testing to MC Generators
  - EP-SFT can extend static analysis (coverity, PVS-Studio, clang) and runtime sanitizers for MC generator projects, providing centralized quality assurance.
  - Making these results available to MC developers and provides valuable feedback on code quality clang sanitizers helps catch critical runtime issues early.

Summary

- EP-SFT provides a range of tools and resources that can help streamline MC generator workflows and improve code quality.
- We are happy to collaborate with the MC community to adapt and expand these resources. Your feedback and ideas are crucial to making these tools as useful as possible.

EP-SFT is also open to your vision on where our group could further contribute to MC development.