A new Scheme for ATLAS Trigger Simulation using Legacy Code

Gorm Galster

University of Copenhagen

Joerg Stelzer

Michigan State University

Werner Wiedenmann

University of Wisconsin

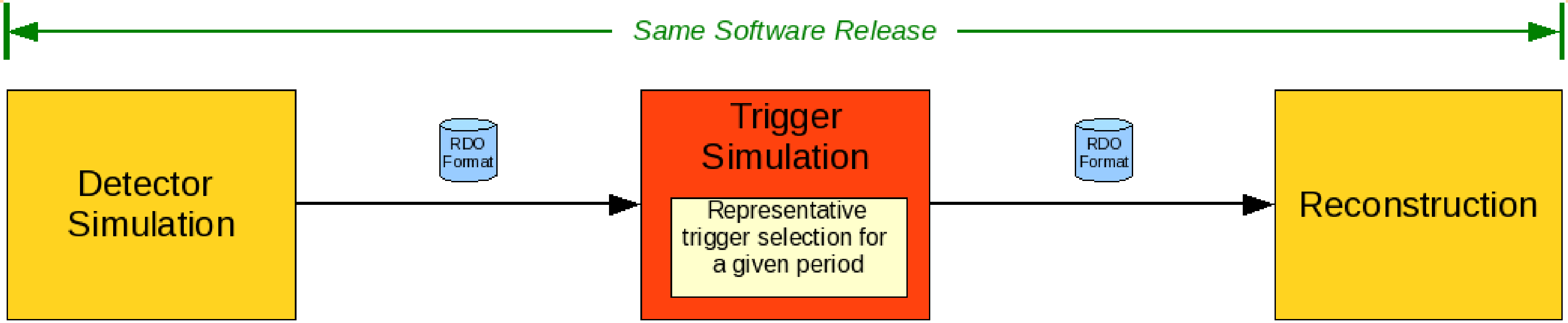
CHEP 2013

Amsterdam, 14-18 October 2013

Motivation

- Precise Monte Carlo simulation of data taking periods also includes the accurate simulation of the trigger response
  - Ideally use the same algorithms and selection criteria as during data taking
  - Follow important changes in the trigger selection over time
    - Changed selection criteria e.g. due to higher luminosities
    - New trigger algorithms and new trigger lines
- How to achieve this for past data taking periods ?
- Re-simulation of the trigger response may be necessary because
  - improved event generators are deployed
  - an improved detector description or event reconstruction is available
  - the statistics of the MC samples needs to be increased
  - the response of new physics processes to the trigger selection needs to be studied

Present Monte Carlo Simulation Chain



One software release for all simulation steps

- Event reconstruction and detector simulation normally reflect the best knowledge about the detector and use the most recent software developments
- The trigger simulation needs to match the online selection of the respective data taking period → For accurate simulation of all data periods the selection configurations and all algorithms which were ever used are kept in the simulation release
- Data exchange between the simulation steps happens through the POOL/ROOT based “*Raw Data Object*” (RDO) format

Trigger Simulation Options

- Porting of old trigger selection code to new simulation releases
  - Requires a maintenance and manpower effort
  - Old selection lines have to be kept operational alongside the most recent trigger selections → sometimes conflicting requirements
  - Old and obsolete algorithms have to be kept operational → need to preserve knowledge about setup of these algorithms over long time periods
  - Old algorithm configurations and conditions data have to be made available in new releases → need to maintain or port necessary infrastructure services
- Parameterizing the trigger response
  - Keep trigger efficiency curves
  - Correlations difficult to handle
  - Maybe not accurate enough
- Re-running legacy trigger selection code "as is" from old releases
  - Drastically reduces the maintenance effort for porting old trigger code to newer releases
  - Effort is shifted to more technical aspects on how to create and maintain an environment for re-running legacy code

Trigger Simulation with Legacy Trigger Code

- Proposal: Use for the trigger simulation of past data taking periods the same trigger software releases which were deployed for these data taking periods
- To realize such a scheme the following issues need to be addressed
  - Split of simulation chain into sub-steps which can use different software releases
  - Long term conservation of important trigger software releases, of configuration data and of conditions data
  - Choice of a data format which can be used for data exchange between legacy trigger releases and the latest detector simulation and event reconstruction releases
  - Ability to run a legacy trigger software release “as is” on evolving hardware and software platforms → abstraction of hardware and software layer
- Proposed strategy should require minimal to no work on already existing releases

Trigger Simulation with Legacy Trigger Code

- The feasibility of the proposal was studied with a prototype
- Split of simulation chain and use of different software releases in each sub-step:
  - The simulation team has already fulfilled this requirement and provided a new set of scripts for running the event simulation chain
  - Different software releases can be used for each simulation step
- Long term release conservation
  - For software distribution ATLAS uses the *cvmfs* file system
  - *cvmfs:* HTTP based file system with local caches
  - *cvmfs* is a write once then read only file system → fulfills requirements for release conservation
  - ATLAS publishes on the *cvmfs* file system
    - Software releases together with all required external software
    - Development tools (e.g. compilers)
    - Conditions and configuration data in *SQLite* files

Data Format Issues : RDO Data Format

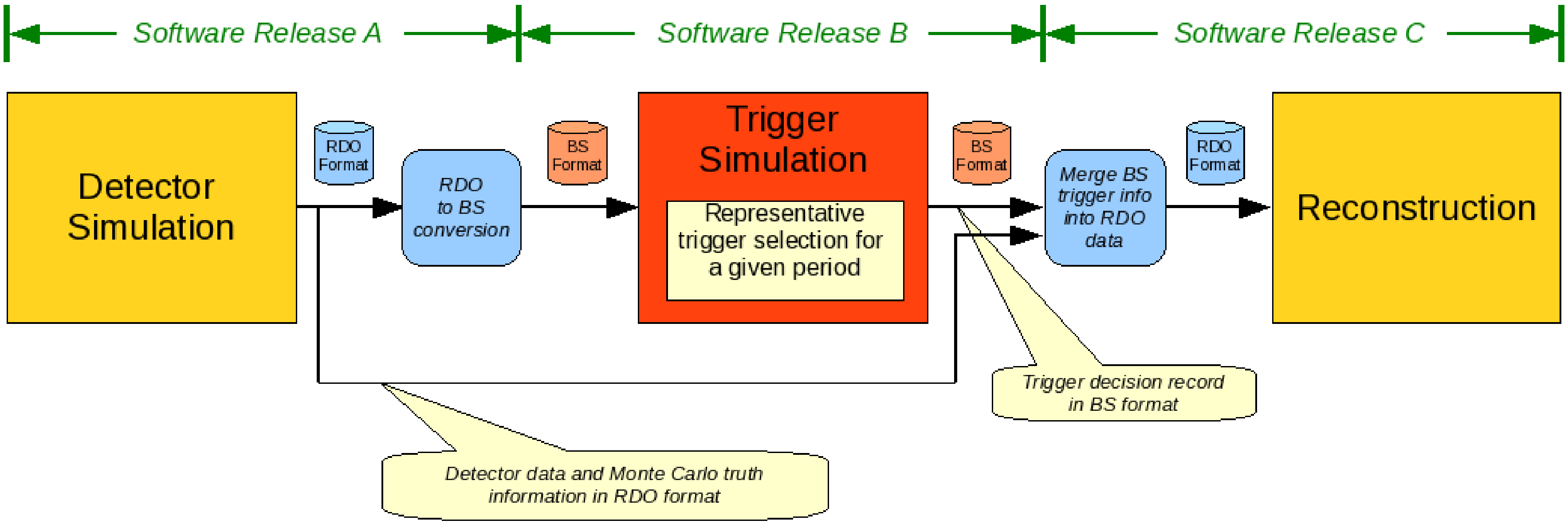
- **In the current simulation chain data are exchanged between the simulation modules in the RDO format**
  - **Container format based on ROOT technology.**
  - **Payload format is community dictated and may change with new releases**
  - **RDO data from simulation contain also containers with information about Monte Carlo truth information and data processing parameters**
- **For retrospective trigger simulation the exchanged data have to be compatible with the different release versions**
  - **Output data from the newer detector simulation release need to be readable by the old trigger simulation release → *Forward compatibility***
  - **Output data from the old trigger simulation release need to be readable by the newer reconstruction release → *Backward compatibility***
- **Release dependent changes in the RDO format make it difficult to reach compatibility with many old trigger releases → substantial amount of work required**

Werner Wiedenmann, A new Scheme for ATLAS Trigger Simulation…, CHEP 2013, Amsterdam

Data Format Issues: Detector Raw Data Format

- **Raw data directly read out from the ATLAS detector are provided in the "*Byte Stream*" (BS) format**
  - **Container format based on *uint32* arrays**
  - **Format is tightly coupled to detector readout hardware → payload version changes slowly**
  - **Simple structure for payload data**
- **Backwards compatibility is guaranteed: All ATLAS event reconstruction releases are required to be able to read byte stream data from all important data taking periods → the byte stream output from an old trigger release is readable by newer reconstruction releases**
- **Forward compatibility is easier to provide → new requirement: detector simulation releases need to be able to convert the detector raw data to byte stream format with a given payload version**
  - **Conversion scripts to byte stream format are already available**
  - **Some detectors already allow the specification of the payload version**

Modified Simulation Chain

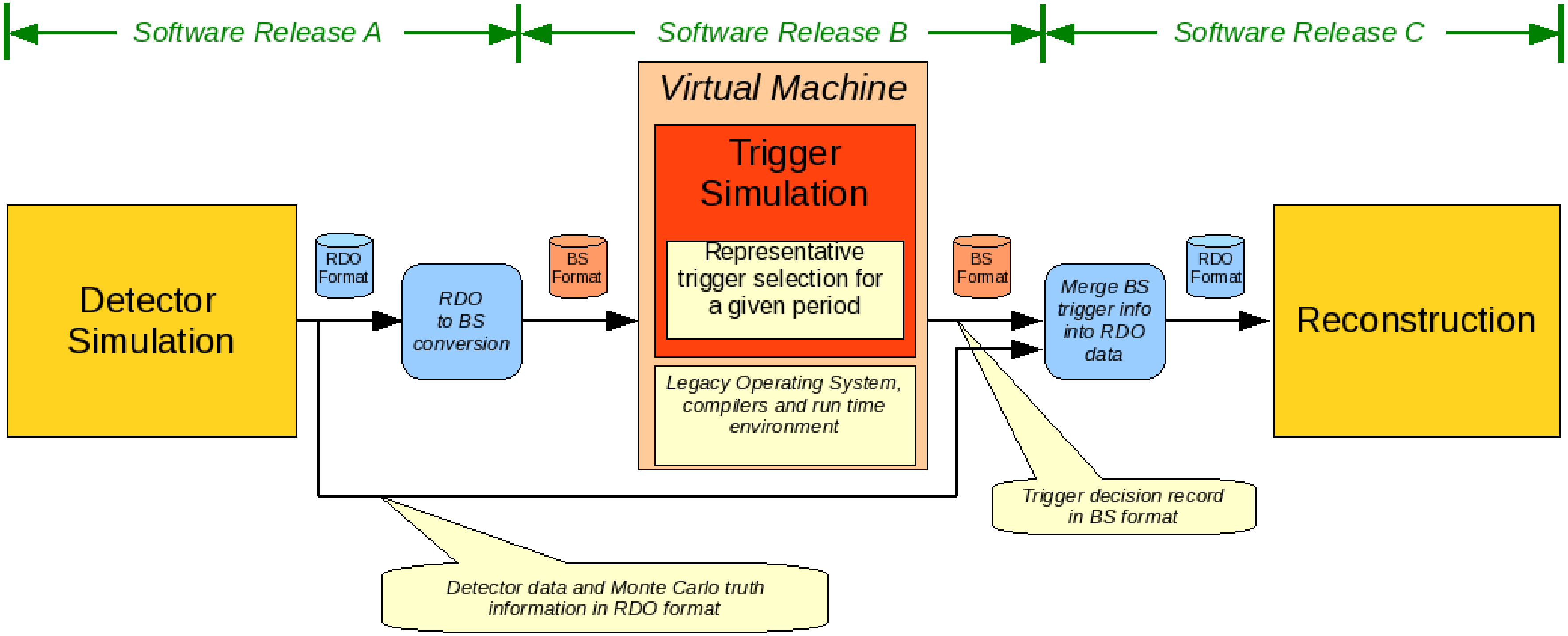

Software Release A
Software Release B
Software Release C
Detector Simulation
RDO Format
RDO to BS conversion
BS Format
Trigger Simulation
Representative trigger selection for a given period
BS Format
Merge BS trigger info into RDO data
RDO Format
Reconstruction
Trigger decision record in BS format
Detector data and Monte Carlo truth information in RDO format

- Use byte stream format for input/output in the trigger simulation module
- The byte stream format has no equivalent containers for simulation meta data and Monte Carlo truth information →
  - Write out only trigger decision record in BS from trigger simulation
  - Use RDO data as input to reconstruction
  - Add data merging step for trigger decision record before reconstruction step

Use of Virtualization

- For the medium term older releases can be used on a newer operating system with a compatibility layer
  - E.g. releases built on Scientific Linux 5 can still run on Scientific Linux 6
- In the long term it will be impossible to use legacy code "as is" and run it unmodified
  - New computing hardware technologies will be introduced
  - Operating systems will change
  - Compiler and core libraries will change
- For long term use virtualization to preserve the complete runtime and development environment for trigger legacy code
  - Hardware abstraction and preservation of software environment
  - Introduces computational and resource overhead
  - Need to foresee patch releases to adapt to changing external infrastructure services → e.g. for data input/output services or for changing database technologies

Modified Simulation Chain with Virtual Machine


Software Release A
Software Release B
Software Release C
Virtual Machine
Trigger Simulation
Representative trigger selection for a given period
Legacy Operating System, compilers and run time environment
Detector Simulation
RDO Format
RDO to BS conversion
BS Format
BS Format
Merge BS trigger info into RDO data
RDO Format
Reconstruction
Trigger decision record in BS format
Detector data and Monte Carlo truth information in RDO format

- **Further changes to simulation chain required**
  - **Virtual image for trigger simulation can not run in a batch queue which runs itself in a virtual machine**
  - **Requires infrastructure for data input/output e.g. with *xrootd* or connection to virtual disks, which can be exchanged between simulation modules**

Proof of Concept Implementation

- Modified simulation chain, no virtualization
  - Proof of concept implemented as 5 standalone jobs
  - Release change supported for each sub-step
  - Versioned byte stream from detector simulation for some detectors
  - Work is presently being done
    - to merge with latest ATLAS simulation scripts
    - to provide versioned byte stream writing for all sub-detectors
- Trigger simulation in virtual image
  - Use *CernVM* appliance (*http://cernvm.cern.ch*) for virtualization
  - Fully integrated with *cvmfs*
  - ATLAS uses *cvmfs* for software and configuration/conditions data distribution
  - A proof of concept script is available which uses the *libvirt* API with the Linux native hypervisor, KVM, as underlying infrastructure to
    - create, start and contextualize a fresh *CernVM* instance.
    - setup ATLAS release software from *cvmfs*
    - run the trigger simulation code
    - dispose the virtual machine and ship out the results over *xrootd* daemon

Future Developments

- Choice of virtual image setup
  - Conserve configuration of production image and recreate it when needed again → *CernVM* model
  - Monolithic virtual image with all necessary software in it
  - Depends also on the experiment’s choices for a data preservation strategy
- Integration into a cloud computing environment
  - Adaptation of simulation scripts
  - Choice of data exchange technology
- Development of a test environment which allows to verify the correct operation of the archived virtual images and a comparison with references over long time periods

Conclusions

- With a proof of concept implementation it was possible to simulate the ATLAS trigger by using older software releases for the trigger simulation while maintaining modern releases for detector simulation and event reconstruction
- The proposed model introduces only little extra maintenance for legacy trigger simulation code
- By using the detector raw data format it was possible to achieve compatibility for data exchange between a large range of different release versions
- Virtualization solutions allowed running old trigger selection code unchanged even when the required running environment was not any more compatible with the used production environment
- More work is still needed before production scalability can be assessed with this “mixed” simulation chain
- Final setup also depends on the experiment’s long term data preservation strategy