Online-Steering of HEP-Applications

Daniel Lorenz
University of Siegen

Darmstadt, 27. 4. 2006
HEPCG Workshop

Bundesministerium für Bildung und Forschung

Overview

- Why online steering?
- General goals
- Architecture
  - Communication layer
  - Data consistency layer
  - Application / visualization
- Further development
- Summary

Why Online Steering? (1)

- Typical cycle:
  - Configure job via job options
  - Submit job
  - Wait until job finished
  - Retrieve job output
  - Evaluation of the result
- Results may be meaningless because of
  - Improper job parameters (e.g cuts)
  - Semantic faults in job options, that are not found in test run

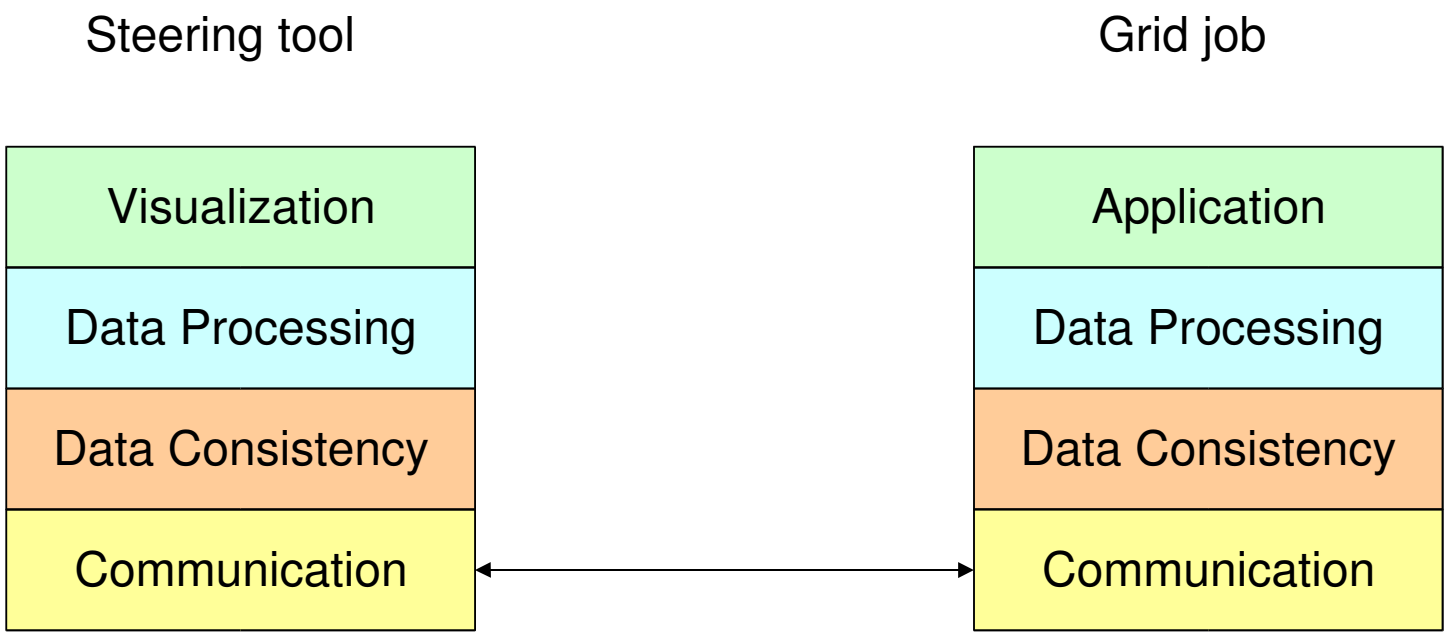
Why Online Steering? (2)

- Early detection of meaningless results
- Change and explore parameters online thus avoiding multiple runs
  - Early abortion of meaningless jobs
  - Faster and fewer cycles
  - Save time
  - Save resources

General Goals

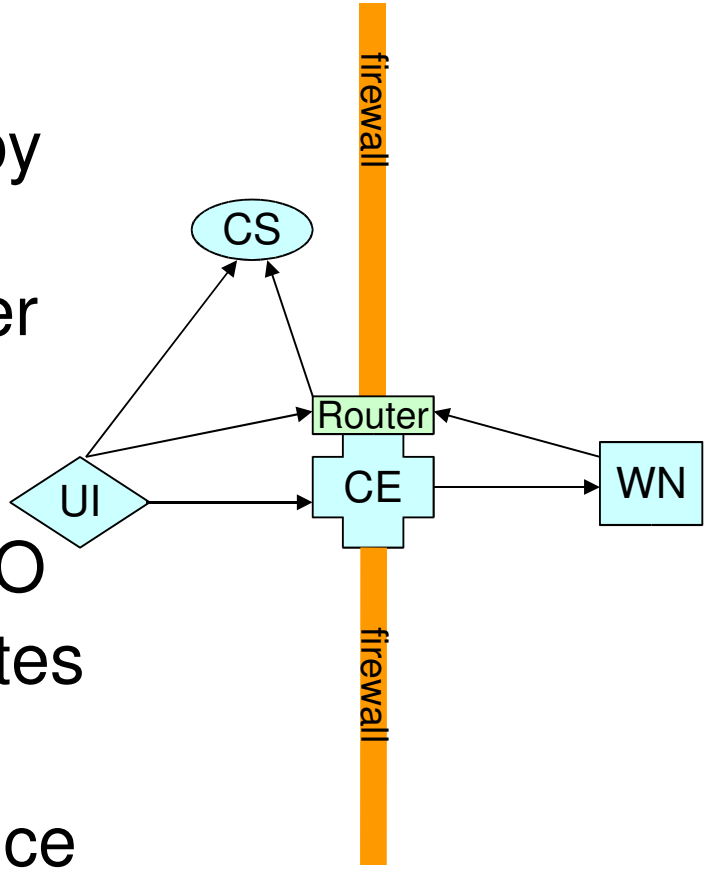
- Monitoring of user defined data (e.g. partially filled histograms)
- Steering of job parameters (e.g. algorithm properties, cuts)
- Restart of job with changed parameters without submitting a new job
- Termination of jobs
- Easy to integrate into existing systems (e.g. ATLAS software: Athena, ROOT)
- Secure interactive connection to a grid job

Overall Architecture



Communication Layer

- Connection to job invoked by steering tool
- Job runs on unknown worker node (WN)
- WN behind firewalls
- Secure router with Globus IO
- Authentication with certificates
- Connection by job name
- Needs communication service (CS)

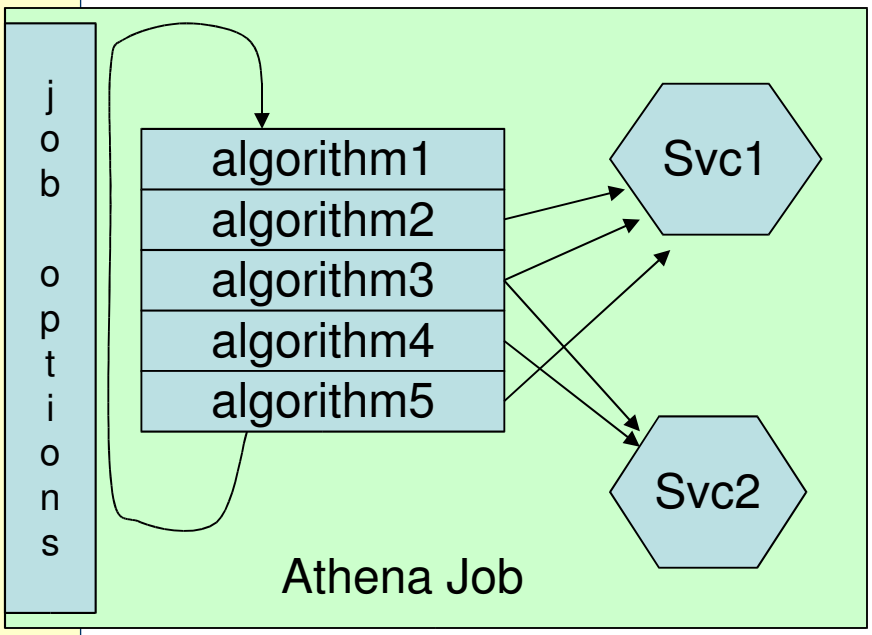


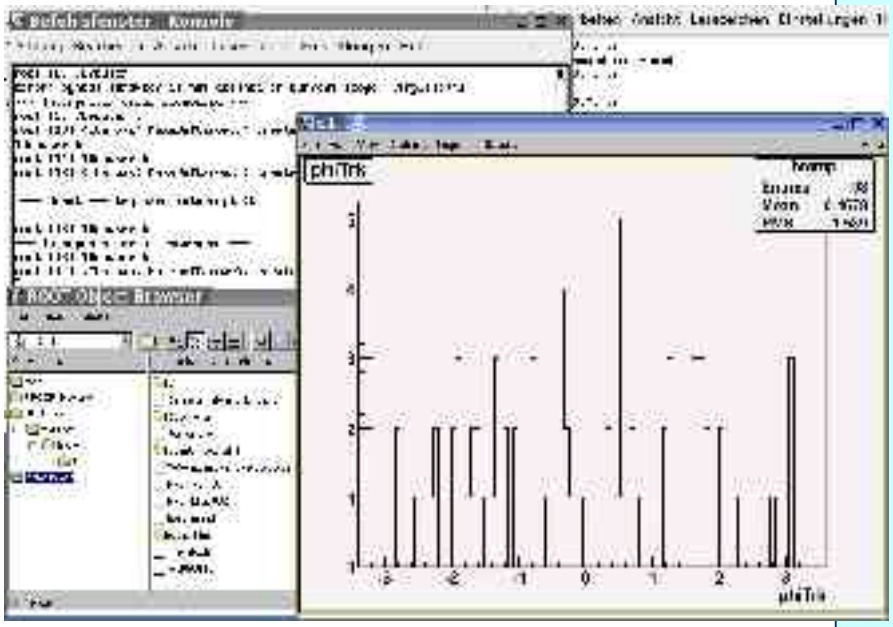
Data Consistency Layer

- Application independent
- Registration of variables and streams
- Registration of data with access methods
- Regular call of a check() method
  - Mapping of values with same name
  - Handling of data traffic
- Data exchange:
  - Pull on demand
  - On regular basis
  - Push steering values

Environment

- Integration into ATLAS experiment software Athena and ROOT





Integration into HEP Applications

- Simply adding one algorithm to a job options file provides:
  - Access to data stored in N-Tuple service
  - Histogram service
  - Possibility to stop, restart, single step
  - Change of job options and applying them after restart
- Steering service:
  - Instrumentation of own code
  - Registration of arbitrary data in memory or in streams
  - Algorithm for accessing special data (eg. ATLAS StoreGate objects)

Visualization

- Interface to ROOT
- Browser:
  - List available data
  - Download / upload streams
  - Monitors, steers simple data types

- Data mapping:
  - Interface for ROOT to data consistency layer
  - Registration of variables with a name
  - Mapping of variables of same name
  - Working with these data like with any other local data

Further development

- Implementation of a prototype with this functionality
- Further development depends on users
- Ideas for further development:
  - Automated evaluation and error report
  - Remote access of files instead of download (esp. ROOT-files)
  - Access to StoreGate objects
  - Extension for parallel programs

Summary (1)

Online steering saves time and resources

- Development of a general steering system

– Conceptual design	finished
– Communication layer	10 % (works locally)
– Data consistency layer	95 %
– Integration	
• Browser	95 %
• ROOT mapper	100 %
• Algorithm	95 %
• Service	0 %

Summary (2)

- First prototype (local communication) will be tested with particle physics group starting end of May
- Second prototype (grid communication) testing at Siegen starts in September
- First public release planned for November

- Integration into ATLAS software by changing job options
- Integration to ROOT by creating an interface to ROOT
- Possibilities for further development