A Regression Testing Framework for Non-catastrophic Failure Detection

Lonnie D. Crosby
Justin L. Whitt, Patricia A. Kovatch and George Herring
University of Tennessee, NICS
Chris Buckley – University of Tennessee

Non-catastrophic Failure Detection

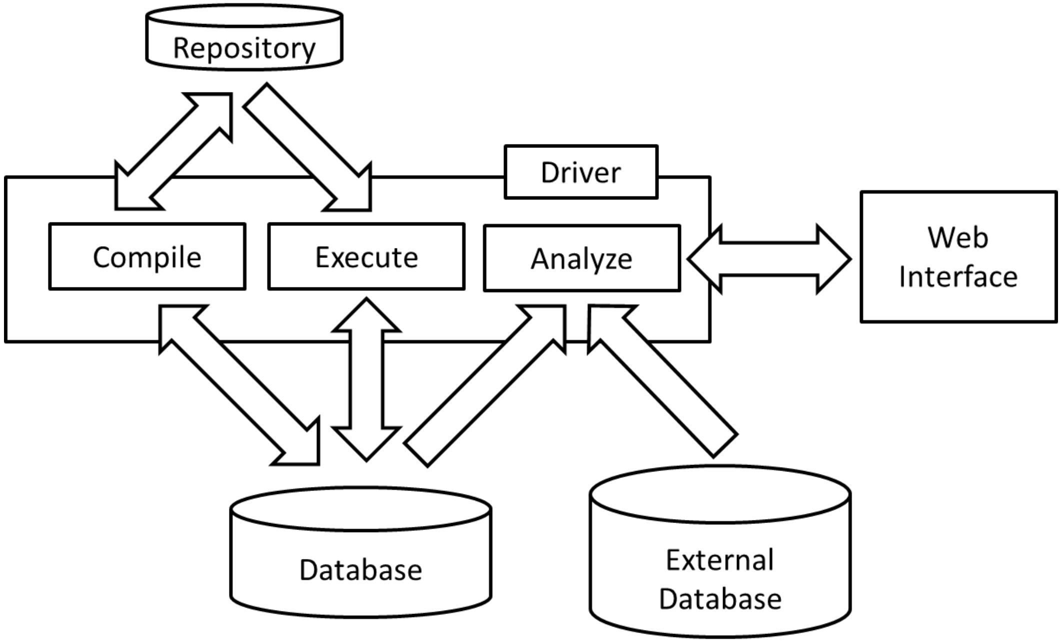
- Benchmark or diagnostic tests are good for the detection of catastrophic failures.
- Non-catastrophic failures:
  - adversely affect application execution without system instability or notification.
  - unknown frequency and can have long durations
  - user facing
- Appropriate first-response methods could quickly detect these failures and reduce impact.
- Detection of non-catastrophic failures may be performed by observing the historical behavior of applications.

Applications as Indicators of System Performance

- Scientific applications interact with computing resources in unique, varied, and complex ways.
  - Use, store, and alter data in RAM
  - Perform mathematical operations and comparisons on data
  - Communicate data between processes
  - Write data to and read data from file systems
  - Utilize and are integrated with system and third party libraries and code
- Changes in application performance can indicate a hardware/software problem.

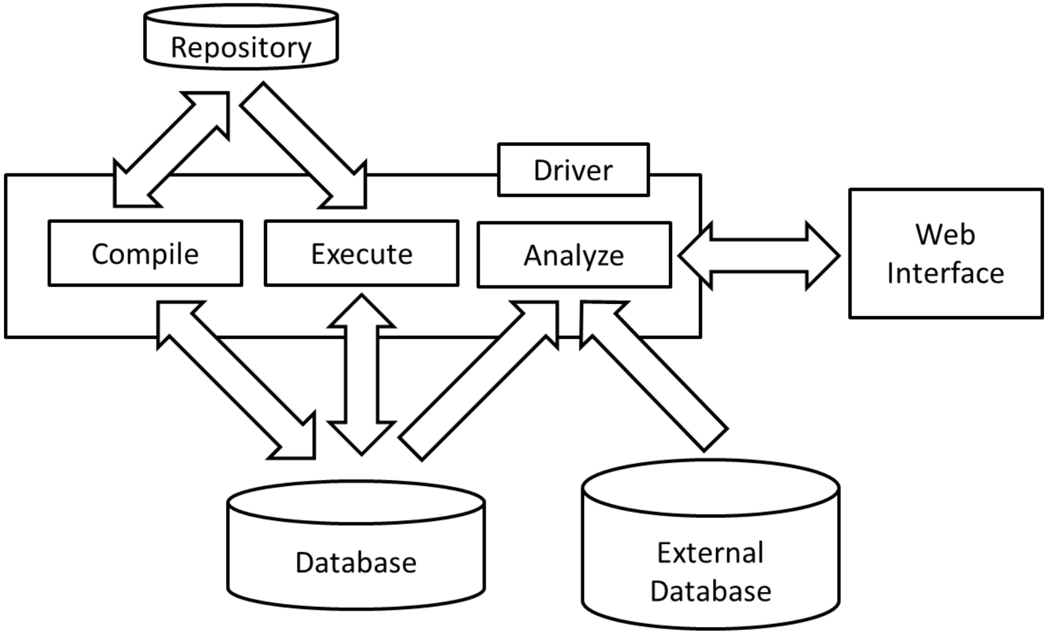
Regression Testing Framework

- Applications (source, executables, and tests) are stored in a software repository.
- The "compile" component produces executable files utilizing system compilers and libraries.



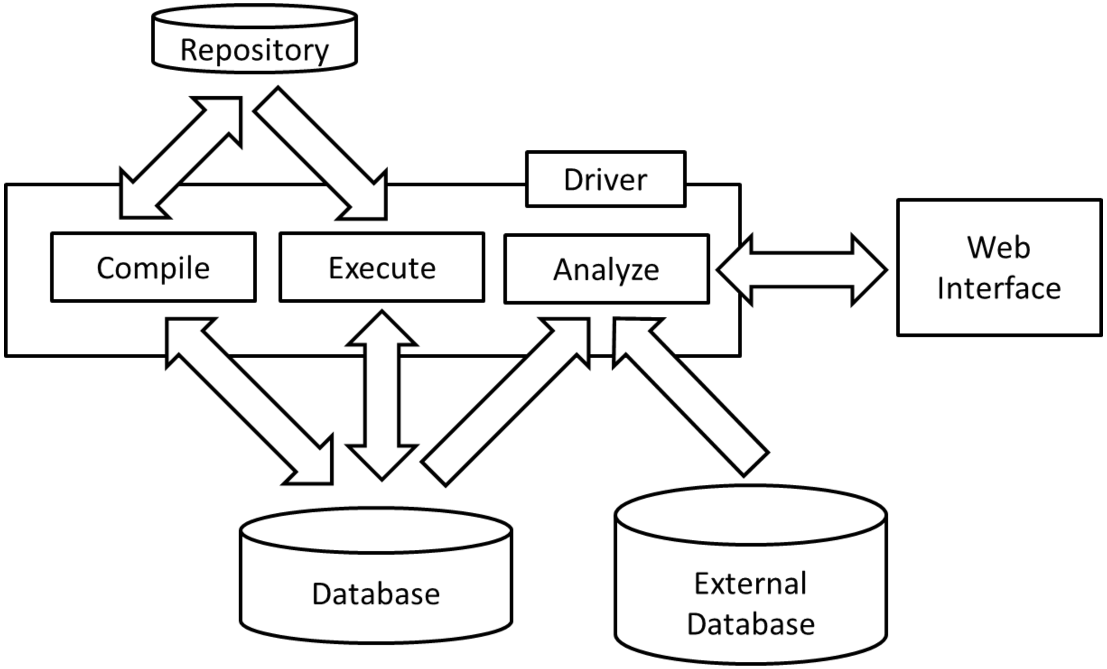
Regression Testing Framework

- The “execute” component uses executables to run defined test cases.
- Data collected through the compile and execute steps are collected in a dedicated database.



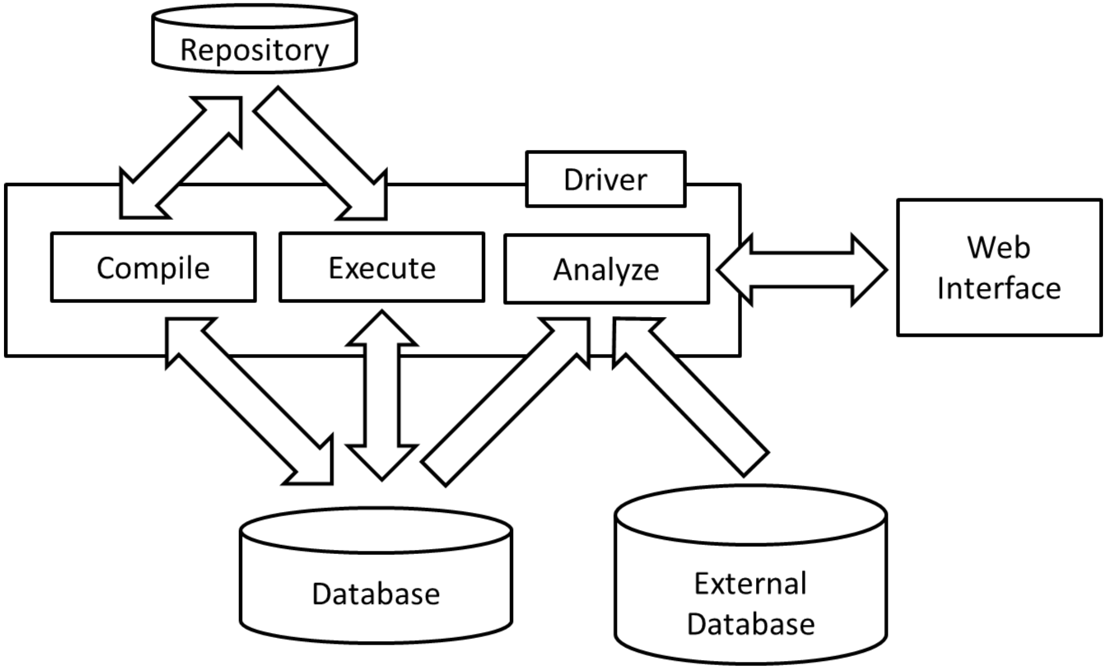
Regression Testing Framework

- The “analyze” component gathers results of test cases and compares against historical performance. Data may be correlated from external databases. Reports may be visualized through a web interface.



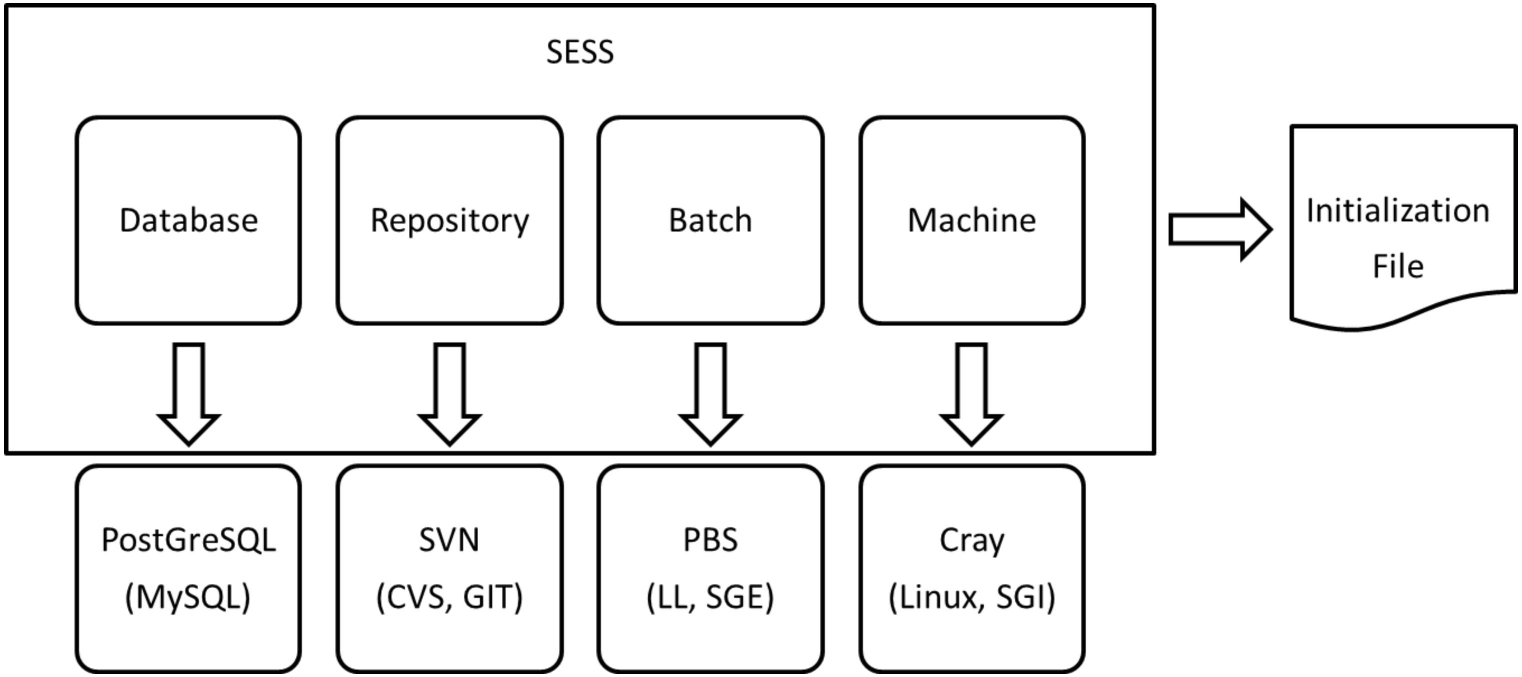
Regression Testing Framework

- The “driver” component orchestrates complex operations requiring compilation, execution, and analysis.



Modularity of Components

- The compile, execute, and analyze components utilize the System Environment Specification System (SESS) to interact with specific system components. Interactions are configurable per system via an initialization file.



Current Work/Future Directions

- Chris Buckley – University of Tennessee Student
  - Developed the database design and PostgreSQL interface.
- George Herring – NICS Summer Intern
  - Developed the SESS interface for the Database, Repository, and Machine classes.
  - Developed the PostgreSQL, Subversion, Cray, and Linux modules for SESS.
  - Developed the “compile” component of the Regression Testing Framework.
- Future Work
  - Develop the “execute” and “analyze” components.
  - Develop the Batch class and additional SES modules.

Contact

Lonnie D. Crosby

University of Tennessee, NICS
lcrosby1@utk.edu