An Empirical Comparison of Methods for Compressing Test Coverage Reports
Erik Ostrofsky and Gregory M. Kapfhammer
Department of Computer Science, Allegheny College
Presented at the 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM 2009), Lake Buena Vista, Florida

SOFTWARE TESTING CHALLENGES

- **Complex** source code, databases, files, and network communication
- **Defects** may exist in the individual components or their interactions
- **Testing isolates defects** and establishes confidence in correctness

REGRESSION TESTING PROCESS

- When software is **modified**, new tests run in addition to the old, thus reducing the risk of a regression in correctness while increasing the test suite size
- **Coverage reports** identify points in the source code and execution environment (e.g., files and databases) that are **covered** by each test case

TEST COVERAGE MONITORING CHALLENGES

- **Many testing and analysis** techniques (e.g., fault localizers, adequacy calculators, test prioritizers, debuggers) require a test coverage report
- **Coverage reports balloon** in size as the monitor includes many details about control flow, data definition and use, and environment interactions

FORMATION OF THE TEST COVERAGE MONITOR

- **Instrumentation**
- **Report Format**
- **Report Type**
- **Report Storage**
  - Static
  - Dynamic
  - Binary
  - XML
  - CCT
  - DCT
  - Standard
  - Compressed

Compressors: Gzip, Zip, XMll, and XMLPPM

**TIME OVERHEAD TO STORE THE COVERAGE REPORTS**

- **Storage Time (ms)**
  - CCT-Bin
  - DCT-Bin
  - CCT-XML
  - DCT-XML

**SIZE OF THE COMPRESSED COVERAGE REPORTS**

- **Report Size (KB)**
  - Uncompressed
  - Compressed

**CONTACT AUTHOR:** Gregory M. Kapfhammer (gkapfham@allegheny.edu)

http://www.cs.allegheny.edu/~gkapfham/research/kanonizo/