Measuring the Combinatorial Coverage of Software in Real Time

Zachary Ratliff

Computer Security

Security Components & Mechanisms

August 4th, 2016

What is Combinatorial Testing?

- Design of Experiments (D.O.E.) for software testing
- Can significantly reduce testing time and costs without sacrificing effectiveness
- Offers a partial solution for showing that a particular program will work for all given inputs

Intractable Nature of Software Testing

- The input domain space of software grows exponentially to the number of input parameters

- 10 binary inputs: $2^{10} = 1{,}024$ configurations

- 20 binary inputs: $2^{20} = 1{,}048{,}576$ configurations

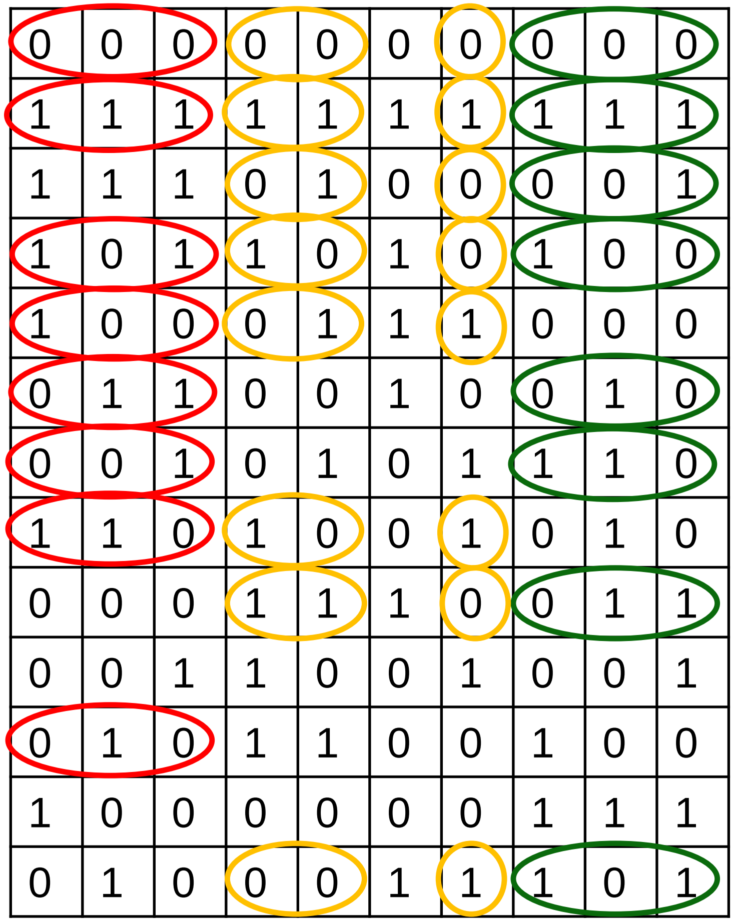
Folding a piece of 0.01cm thick paper 42 times will get you to the moon…
$(0.01 \times 2^{42}) = 439{,}804$km

***Note: You can only fold paper in half about 7 times…**

Covering Arrays

- Mathematical object representing all t-way combinations of n parameters.

- Every combination between t parameters appears at least once



0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	0	0	0	1
1	0	1	1	0	1	0	1	0	0
1	0	0	0	1	1	1	0	0	0
0	1	1	0	0	1	0	0	1	0
0	0	1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0	1	0
0	0	0	1	1	1	0	0	1	1
0	0	1	1	0	0	1	0	0	1
0	1	0	1	1	0	0	1	0	0
1	0	0	0	0	0	0	1	1	1
0	1	0	0	0	1	1	1	0	1

Efficiency of Covering Arrays

- Total variable value configurations for a given system is given by:

$$v^t \binom{n}{t}$$

$n =$ number of parameters

$t =$ level of t-way coverage

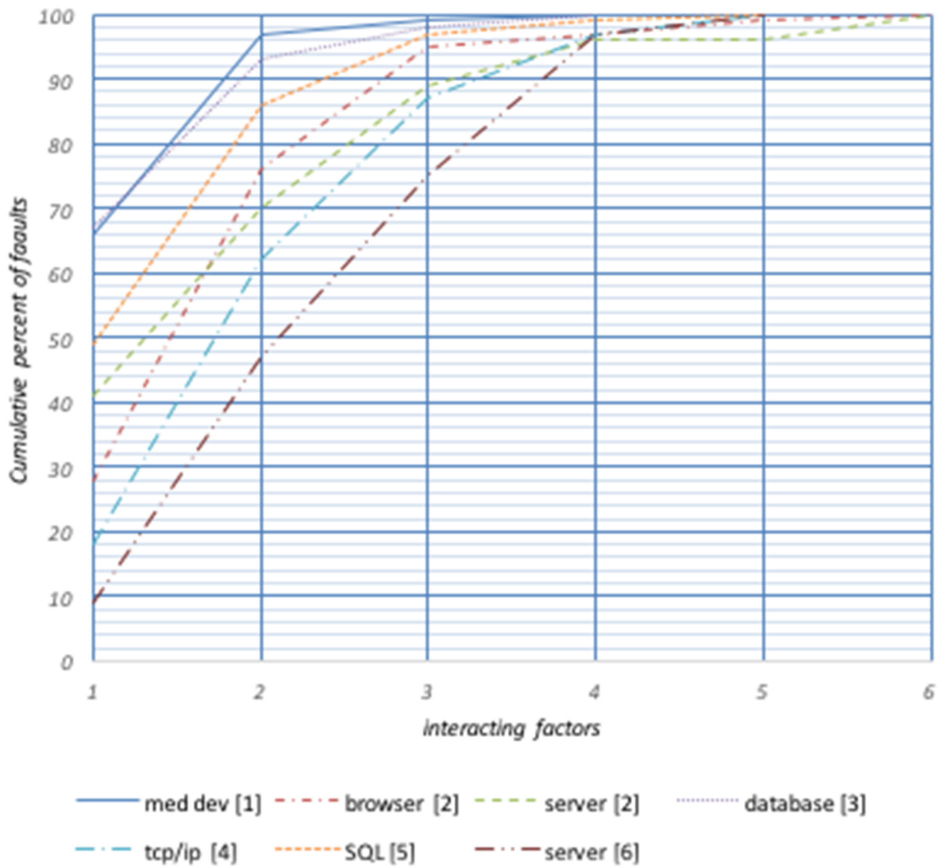
For Mixed Level variable configurations:

$$\sum_i v_{i1} \times \cdots \times v_{it}, \forall \ \mathrm{i} = 1 \ldots \binom{n}{t} \text{ combinations}$$

- In practice, covering arrays grow exponentially to t and logarithmically to n

$$\text{Number of tests} \approx v^t \log(n)$$

The Interaction Rule



- Most failures are induced by one or two factors with progressively fewer faults induced by more than two factors
- No failure involving more than 6 factors has been reported
- Covering all 4 to 6-way combinations provides strong testing

The Problem

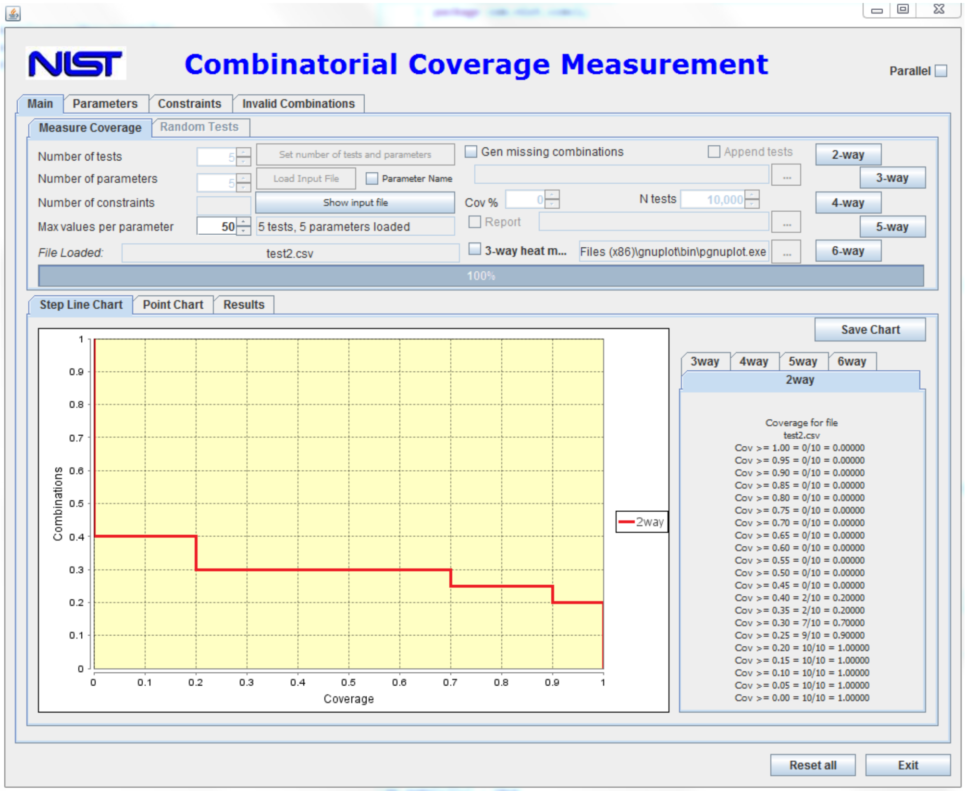
- Most organizations do not fully understand the benefits of switching to combinatorial testing methods
- Time, money, and other resources may not be available to alter testing practices
- Lack of Combinatorial testing software tools and training available

CCM: Combinatorial Coverage Measurement Tool

- Cross platform tool written in Java

- Measured combinatorial coverage of static .csv files

- Features:
  - Generate missing combinations
  - Constraint support
  - Display invalid combinations



Number of tests: 5

Number of parameters: 5

Number of constraints:

Max values per parameter: 50

5 tests, 5 parameters loaded

File Loaded: test2.csv

Cov %: 0

N tests: 10,000

Files (x86)\gnuplot\bin\pgnuplot.exe

100%

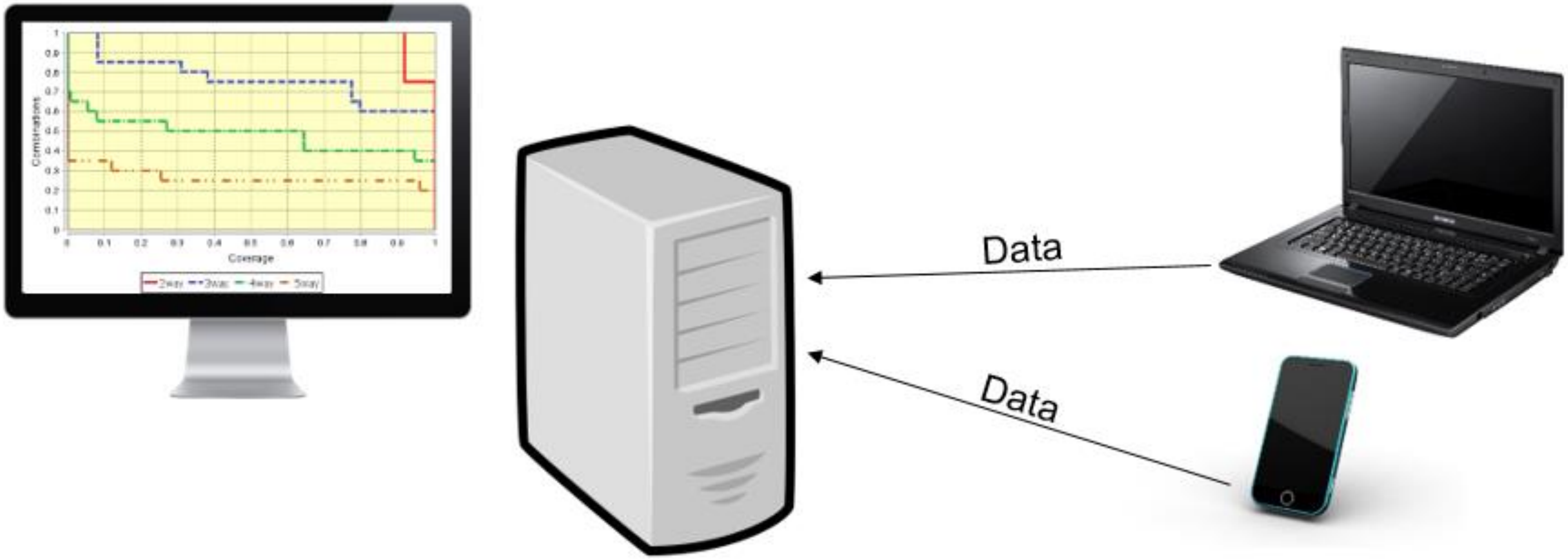
```
Coverage for file
test2.csv
Cov >= 1.00 = 0/10 = 0.00000
Cov >= 0.95 = 0/10 = 0.00000
Cov >= 0.90 = 0/10 = 0.00000
Cov >= 0.85 = 0/10 = 0.00000
Cov >= 0.80 = 0/10 = 0.00000
Cov >= 0.75 = 0/10 = 0.00000
Cov >= 0.70 = 0/10 = 0.00000
Cov >= 0.65 = 0/10 = 0.00000
Cov >= 0.60 = 0/10 = 0.00000
Cov >= 0.55 = 0/10 = 0.00000
Cov >= 0.50 = 0/10 = 0.00000
Cov >= 0.45 = 0/10 = 0.00000
Cov >= 0.40 = 2/10 = 0.20000
Cov >= 0.35 = 2/10 = 0.20000
Cov >= 0.30 = 7/10 = 0.70000
Cov >= 0.25 = 9/10 = 0.90000
Cov >= 0.20 = 10/10 = 1.00000
Cov >= 0.15 = 10/10 = 1.00000
Cov >= 0.10 = 10/10 = 1.00000
Cov >= 0.05 = 10/10 = 1.00000
Cov >= 0.00 = 10/10 = 1.00000
```

***Created by Itzel Mendoza while working as a guest researcher at N.I.S.T.**

Limitations of CCM

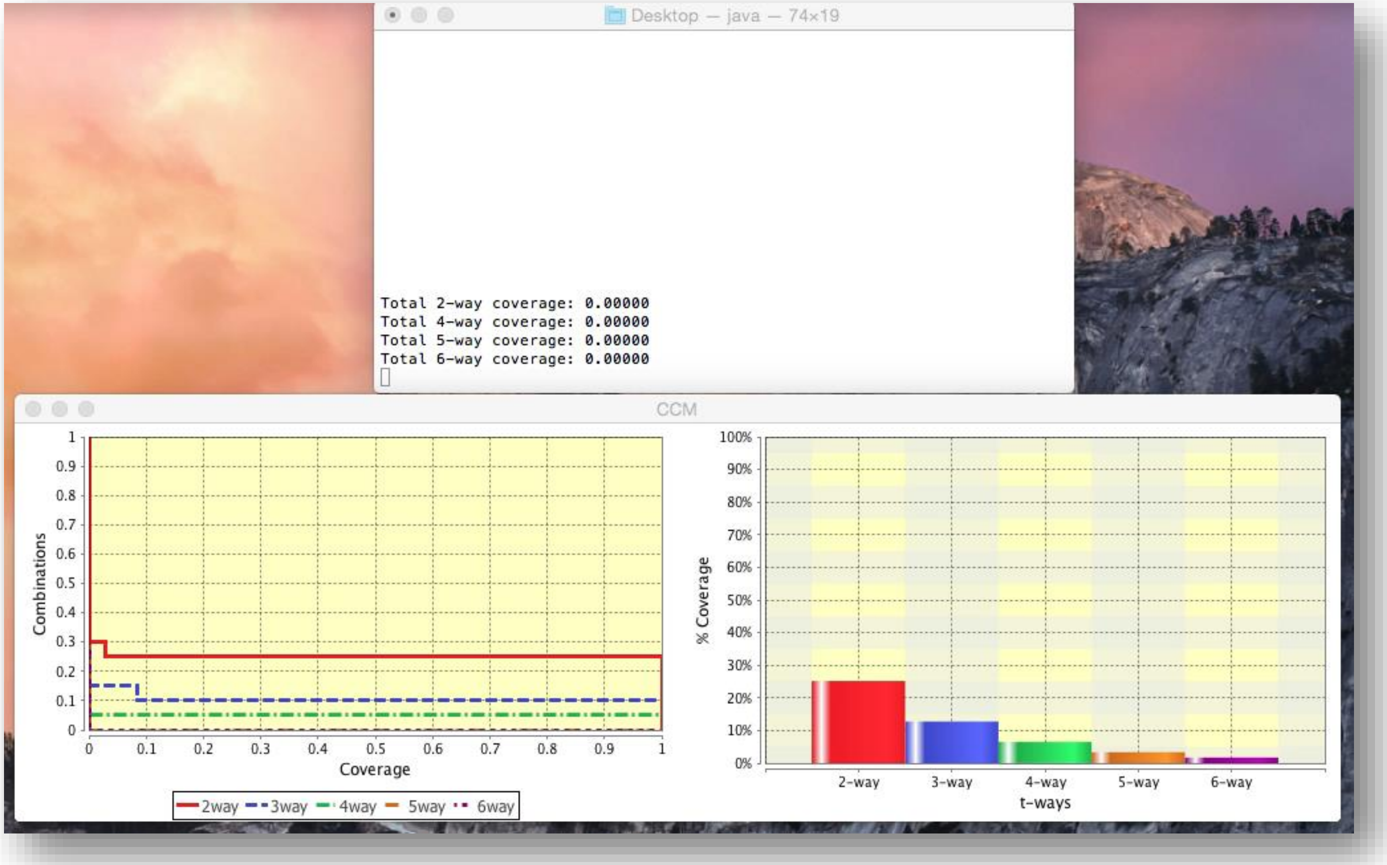
- Could only accept .csv files for test case input
    - No ability to hook other tools in
    - Had to be ran on a local machine
- Limited to static analysis of data
    - Very inefficient for when measuring multiple times as new data is added

Interest was generated in various industries for a new combinatorial measurement tool with capabilities to measure coverage in real time.



Introducing CCM Command Line

Real time combinatorial coverage measurement tool



Desktop — java — 74×19

```
Total 2-way coverage: 0.00000
Total 4-way coverage: 0.00000
Total 5-way coverage: 0.00000
Total 6-way coverage: 0.00000
```

New Capabilities

- Can read multiple file types
  - .csv test case files
  - .txt test case files
  - ACTS .xml configuration files
  - ACTS .txt configuration files

- Added support for equivalence classes and groups within ACTS configuration files
  - Ranges and boundary values defined by interval notation
    - (*,5],[6,*) – creates two range classes: -∞ to 5, 6 to ∞
  - Groups are specified in brackets
    - {“Debian”, “Ubuntu”, “Red Hat”},{“Windows XP”, ”Windows 7”}

- Real time measurement functionality
  - Incrementally measures combinatorial coverage as new test cases are added to the data set
- Accepts input from various sources
  - Files
  - Standard Input
  - External Programs
  - Internet / TCP
- More robust constraint definitions
  - !employee => !grant_permission

***Older version of CCM had issues processing constraints in this notation**

Time Complexity

- The time complexity of initial measurement of static test case files remains the same:

$$\theta\left(n^t(v^t + m)\right)$$

- Incremental measurements while adding test cases:

$$\theta(n^t v^t)$$

In both static and real time measurements, the algorithm is tractable in real world situations

Applications of CCMCL

- Product Readiness
  - Determining if a pre-release version has been tested enough by Beta users.
- Monitoring IV&V Performance
  - Is the IV&V company providing quality tests to meet the software assurance standards?
- Measuring current test suite implementations
  - Do current test suite implementations already provide significant combinatorial coverage?
- Internet of Things Reliability
  - Measuring how reliable a system of interconnected components likely is.

Acknowledgements

- Rick Kuhn, National Institute of Standards & Technology
- Raghu Kacker, National Institute of Standards & Technology
- Dylan Yaga, National Institute of Standards & Technology
- Itzel Mendoza, Centro Nacional de Metrologia
- SURF Undergraduate Research Program, National Institute of Standards & Technology

References

- D.R. Kuhn, R.N. Kacker, Y. Lei, J. Hunter, *Combinatorial Software Testing*, IEEE Computer Society, August 2009.
- D.R. Kuhn, D.R. Wallace, A.M. Gallo, Jr., *Software Fault Interactions and Implications for Software Testing*, IEEE Transactions of Software Engineering, June 2004.
- Kuhn, D. Richard, Raghu N. Kacker, and Yu Lei. *Introduction to combinatorial testing*. CRC press, 2013.