Extending search-based software testing techniques to big data applications

ERIK M. FREDERICKS
REIHANEH H. HARIRI
MAY 17TH, 2016
Big Data?
Big Data?

- **Volume**: Petabytes of information
- **Velocity**: Speed of changing information
- **Variety**: Data comes in all shapes and sizes
- **Veracity**: Trustworthiness / reliability of data (uncertainty)
Techniques for Managing Big Data

Hadoop / MapReduce

Apache Spark

NOSQL, BigTable, etc.

Google BigTable

Apache CouchDB

neo4j
Position

**SBST techniques can enhance testing techniques for big data applications.**

- Focus on automated test suite generation
- Reduce enormous search space generated by big data

- Isn’t reducing the search space the entire point of SBST?
  - Of course!
  - Big data is simply the next obstacle to be overcome using SBST!
    - Extend our techniques to this new paradigm
Issues and Possible Solutions

Nearly all facets of software testing can be impacted by big data!

Issues that concern the SBST community...

- **Test suite generation**
- Combinatorial testing
- Mutation testing
- etc.
Test Suite Generation

Test suite
- Typically comprise a set of test cases
- Generally concerned with validating a particular operating context
  - Combination of parameters that specify system and environmental configuration
  - Well-studied problem in SBST community [Fraser.2011]

However...
- Big data adds a new wrinkle!
- How can we possibly generate enough test suites to adequately cover the 4 V’s?
Impact of Big Data

Test suites provide measure of coverage for known operating contexts

Consider a nation-wide medical records network (MRN)

- Patient data recorded in Detroit, MI
- Immediately available in Austin, TX
- Patients, doctors, nurses, etc. all interface using heterogeneous devices
  - Network supported by heterogeneous devices
- Data such as patient records, medical imaging, video, etc. ALL available

Deriving test suites to cover entire application becomes quickly non-trivial!

- More reasonable to focus on subsets of application
- E.g., Android/iOS/WinPhone application that interfaces with network
Applications of SBST

SBST techniques now needed more than ever!

Explore a **massive** solution space

Augment existing big data approaches to support SBST
Applications of SBST

Hadoop/MapReduce, for example

- Comprises, at its core, Map and Reduce functions

The overall MapReduce word count process

Input: Deer Bear River, Car, Car, River, Deer, Car, Bear

Splitting: Deer Bear River, Car, Car, River

Mapping: Deer, 1 Bear, 1 River, 1 Car, 1

Shuffling: Bear, 1 Bear, 1 Car, 1 Car, 1 Deer, 1 River, 1

Reducing: Bear, 2 Car, 3 Deer, 2 River, 2

Final result: Bear, 2 Car, 3 Deer, 2 River, 2

For more information, visit: http://www.cs.uml.edu/~jlu1/doc/source/report/img/MapReduceExample.png
Applications of SBST

Hadoop/MapReduce, for example
- Comprises, at its core, Map and Reduce functions

MAP to operating contexts

The overall MapReduce word count process
- Mapping:
  - Deer, 1 Bear, 1 River, 1
  - Car, 1 Car, 1 River, 1
  - Deer, 1 Bear, 1
  - Car, 1 Car, 1
  - Deer, 1
  - River, 1

- Shuffling:
  - Bear, 1 Bear, 1
  - Car, 1 Car, 1 Car, 1
  - Deer, 1 Deer, 1
  - River, 1 River, 1

- Reducing:
  - Bear, 2
  - Car, 3
  - Deer, 2

Final result:
- Bear, 2 Car, 3 Deer, 2 River, 2

REDUCE to minimal coverage criteria

http://www.cs.uml.edu/~jlu1/doc/source/report/img/MapReduceExample.png
Applications of SBST

1 $\rightarrow$ BLOB data

2 $\rightarrow$ Network reliability

$n \rightarrow$ Video playback
Applications of SBST

Parallelized genetic algorithm (GA) for generating test suites with Hadoop [Geronimo.2012]
- Each GA generation is a MapReduce job
  - Fitness evaluation performed by Mappers
  - Reducer collects results and performs evolutionary operations
  - Extend paradigm to manage big data – mappers concerned with operating contexts

Automated test generation using relational databases [McMinn.2015]
- Testing integrity constraints on relational database schema
  - Constraint and column coverage
- Augmented random search and alternating variable method
  - Generate test suites
- Highly-relevant to big data, as big data is typically schema-less!
Acknowledgements

The authors would like to thank Oakland University for supporting this work.
Discussion

Testing applications that interface with big data

Dealing with unstructured data

Extending search-based techniques to the big data (testing) domain
References

