



KYUSHU UNIVERSITY 2011
100th Anniversary

Facilitating Consistency Check between Specification & Implementation with MapReduce Framework

Shigeru KUSAKABE, Yoichi OMORI, Keijiro ARAKI

Kyushu University, Japan



KYUSHU UNIVERSITY



Our expectation

Light-weight formal methods will become more popular.

One of key factors: platform to test specifications.

We are in “Cloud era”, and emerging cloud computing technologies are effective in developing platforms for LWFM?

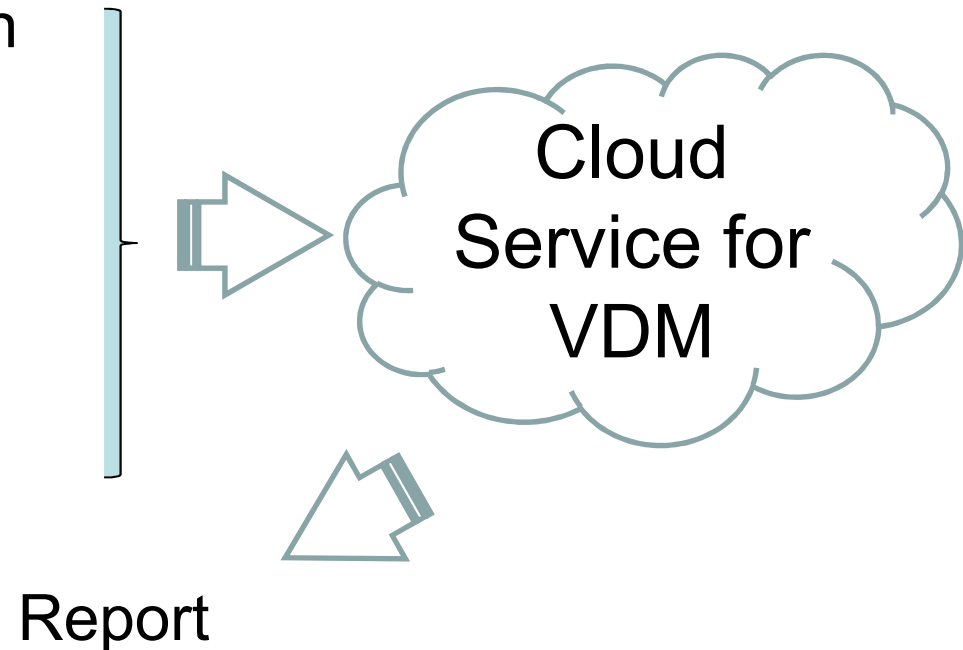
One of the points: elastic environment for various conditions

- increasing # nodes if time constraint gets tight
- enabling brute-force approaches if necessary
- ...

Conceptual scenario

Cloud Service for VDM:

- Executable specification
- Implementation
- Property to be checked
- Configuration
 - test data volume
 - resource scale

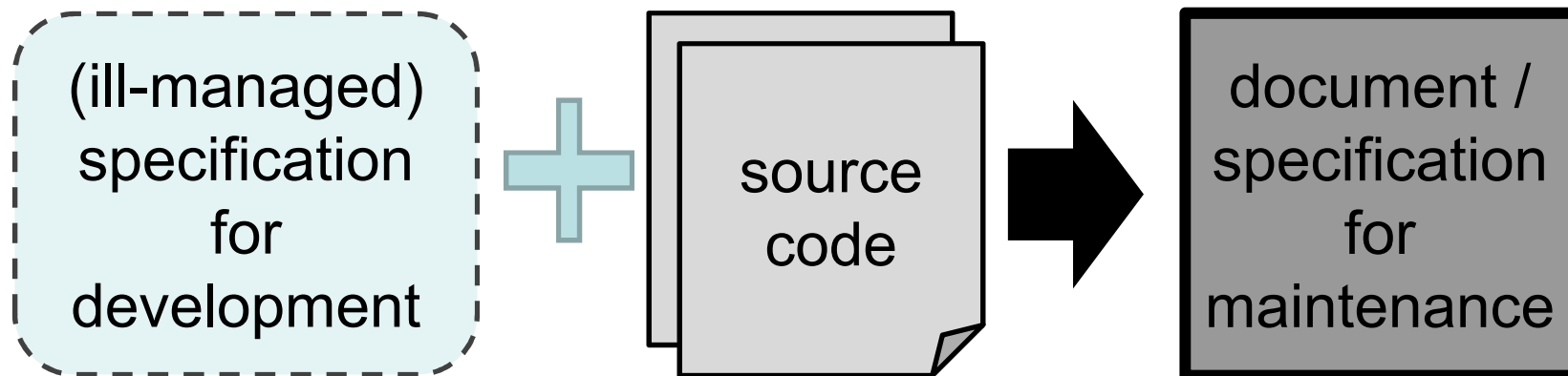


Conceptual scenario

Losing consistency between specification and implementation?

Checking and analyzing consistency (extensively)
between intentional world and extensional world to

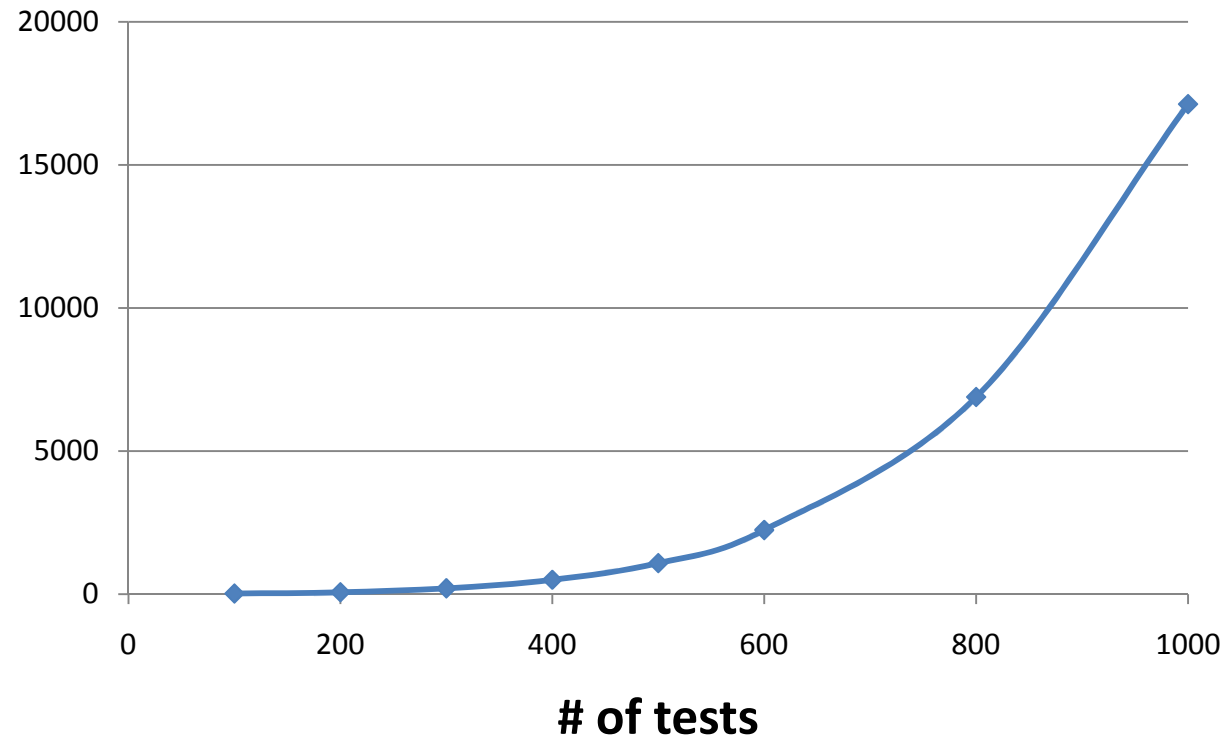
- understand the current status,
- fix the situation / establish the final specification if necessary.





Bad news for our extensive approach

execution time on single node [sec]





Introduce MapReduce to testing specification

Can we use cloud technologies for expensive tests?

MapReduce framework: Parallel & distributed processing without knowledge of

- parallelization,
 - fault tolerance,
 - data distribution,
 - load balancing
- Prepare input file, mapper and reducer functions, then MapReduce system splits input file, distributes over multiple nodes, gathers and folds results.



Testing property for a set of data

As principal abstractions, we often use *sets*, *sequences*, and *mappings*, in VDM.

e.g. test scenario : property for a set of data

exists x in set s & $P(x)$

mapper $P()$, reducer OR

forall x in set s & $P(x)$

mapper $P()$, reducer AND



Key components in our approach

- Test data generation: QuickCheck, Map in Hadoop
- Test execution: VDMTools, Map in Hadoop
- Making summary reports: VDMTools, Reduce in Hadoop

Hadoop: Open Source implementation of MapReduce in Java, including Distributed File System (HDFS), .

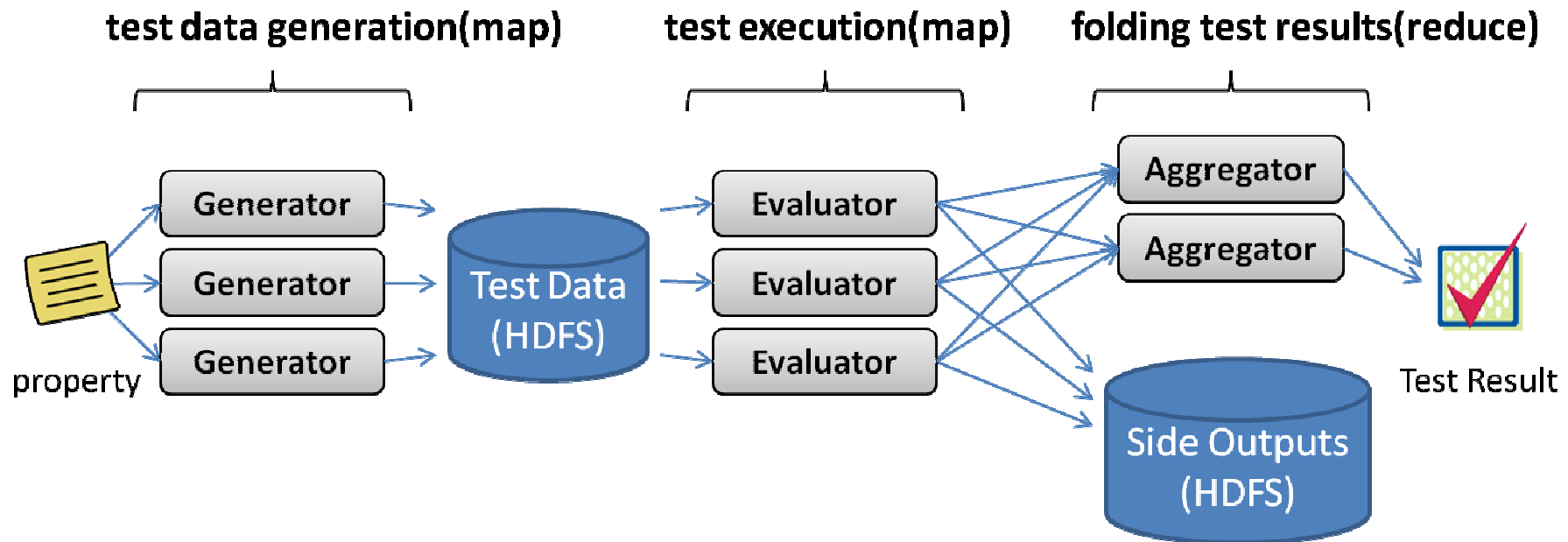
In default, we write mapper and reducer in Java, ...

Streaming allows other languages (scripts for VDM*)

QuickCheck: Test tool for Haskell

- write property expression system must keep
- generate test cases based on the type of property & constraints
- can use to generate test data in other language than Haskell

Test with property-based data generation



Generator: based on QuickCheck

– customized for distributed execution

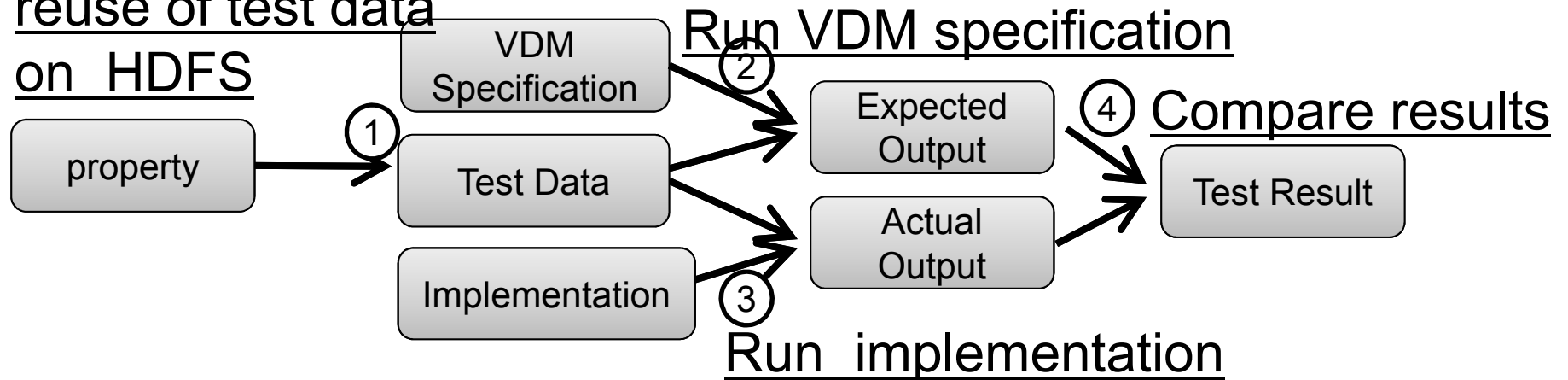
Side outputs: test coverage data, error info., ...

Model-based test

Compare the results of

VDM specification and corresponding implementation

Generation or reuse of test data on HDFS





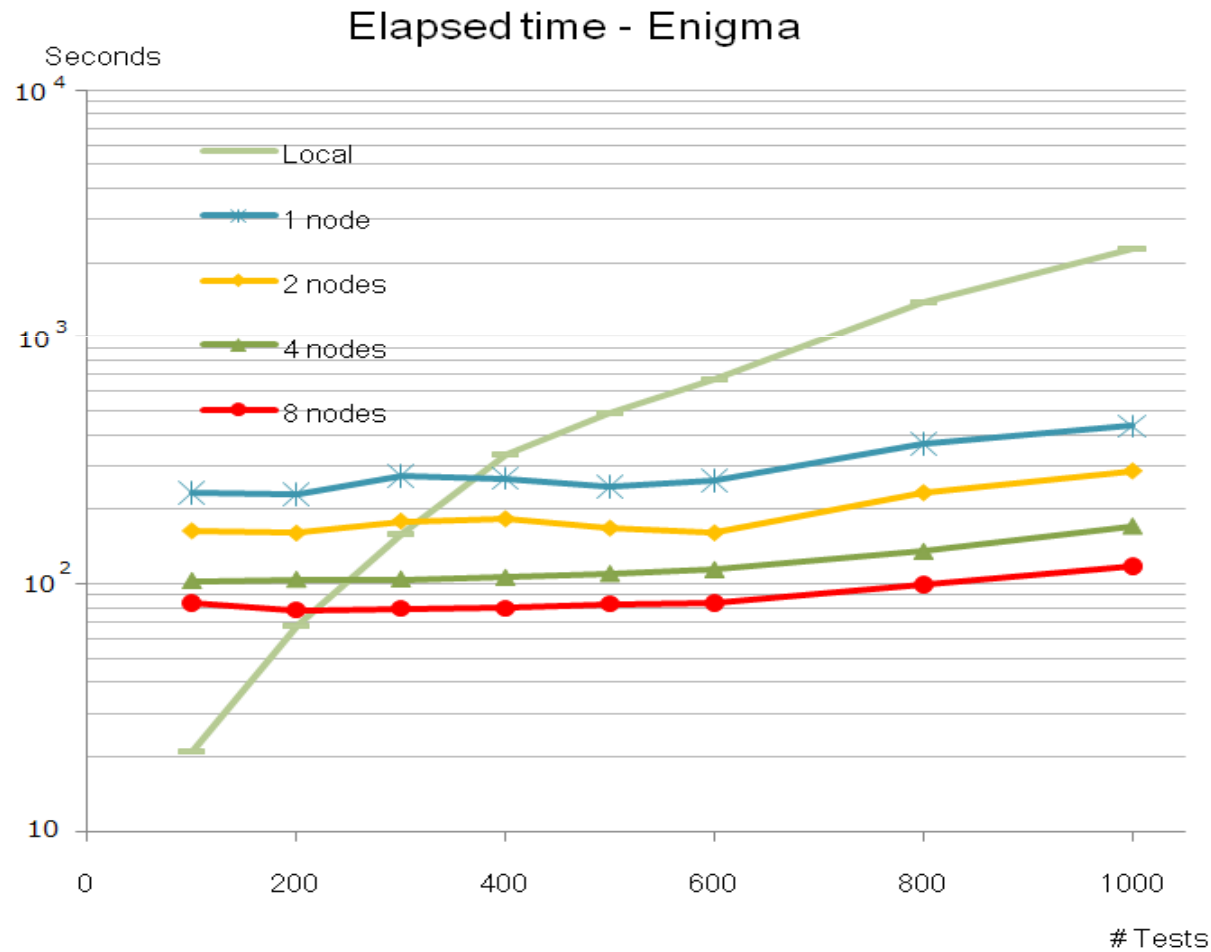
Performance evaluation

- Property-based test (with data generation)
- Model-based test (comparison of model & implementation)
- Configuration
 - Master nodes: JobTracker, NameNode
 - Slave nodes: (TaskTracker + DataNode) X 1 ... 8

	NameNode	JobTracker	TaskTracker+ DataNode
CPU	Xeon E5420 2.50GHz Quad Core	Xeon E5420 2.50GHz Quad Core	Xeon X3320 2.5GHz Quad Core
Memory	3.2GB	8.0GB	3.2GB
Disk	2TB	1TB	140GB
NIC	1Gbps	1Gbps	1Gbps

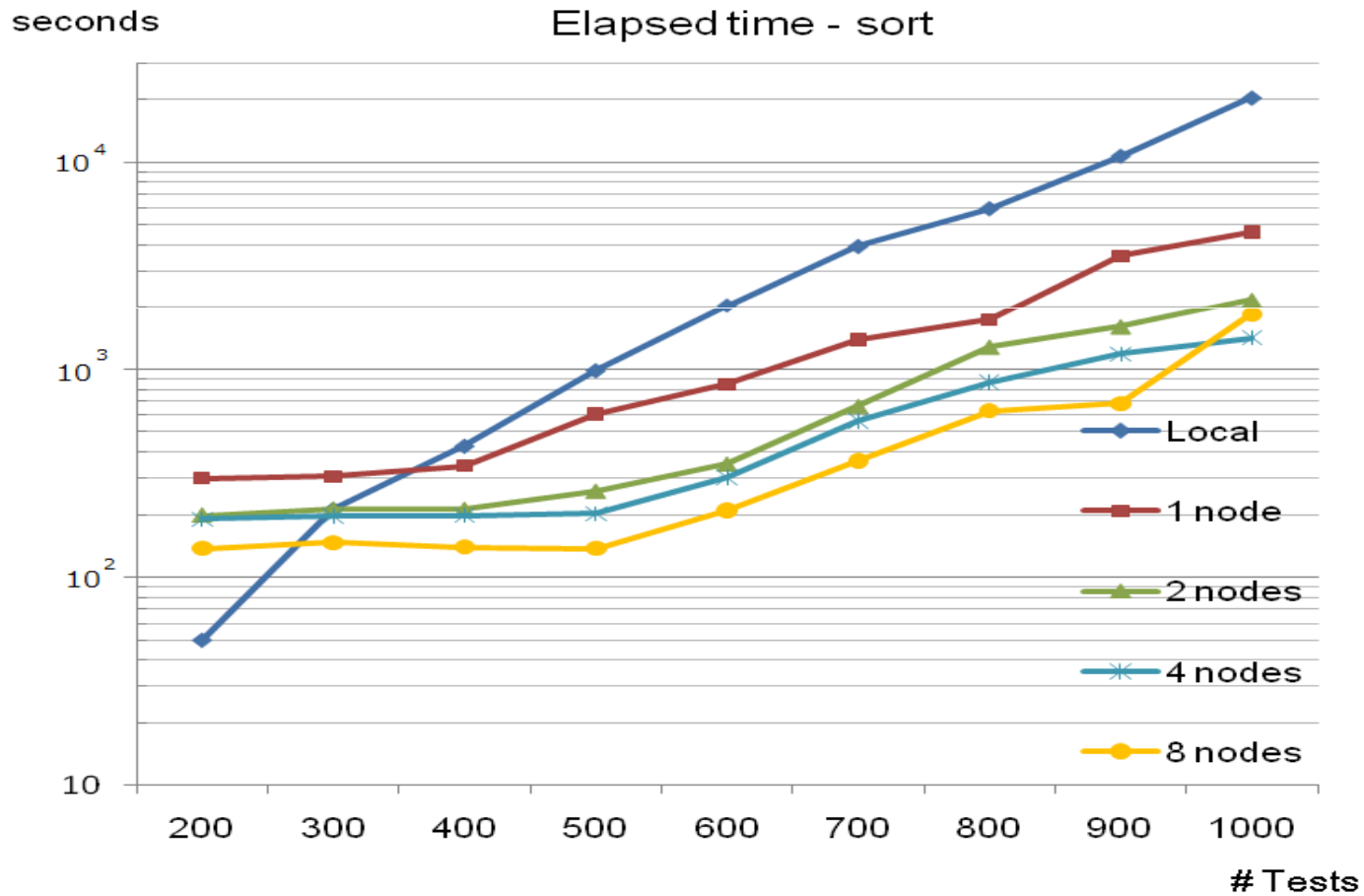


Property-based test - time





Model-based test - time





Concluding remarks

Our work-in-progress report

Parallel processing framework, Hadoop, seems useful for speeding up testing specifications in light-weight F.M.

- Develop specifications, and their extensive tests, in order to increase confidence (instead of proof)

Different testing schemes by combinations of map/reduce

- e.g. Property-based test, Model-based test, ...

Predictable performance in software projects, ...

- it seems hard to predict most cost-effective configuration
- performance can degrade in elastic cloud (virtual) environment