FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Testing Idempotence and Convergence of Automatic Configuration Scripts

Masterstudium:
Software Engineering & Internet Computing

Oliver Hanappi

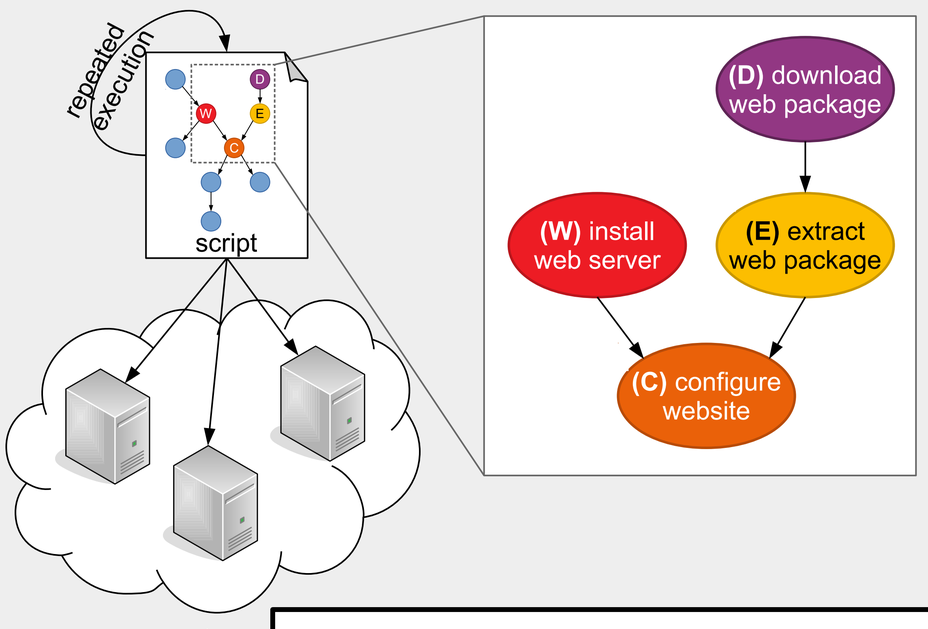
Distributed Systems Group

Technische Universität Wien
Institut für Informationssysteme
Arbeitsbereich: Distributed Systems Group
Betreuer: Univ.-Prof. Dr. Schahram Dustdar
Mitwirkung: Univ.-Ass. Dr. Waldemar Hummer

Motivation

Robust management of large scale IT systems

- Repeated execution of configuration scripts
- Desired state is eventually reached (convergence)
- Deviations are repaired automatically

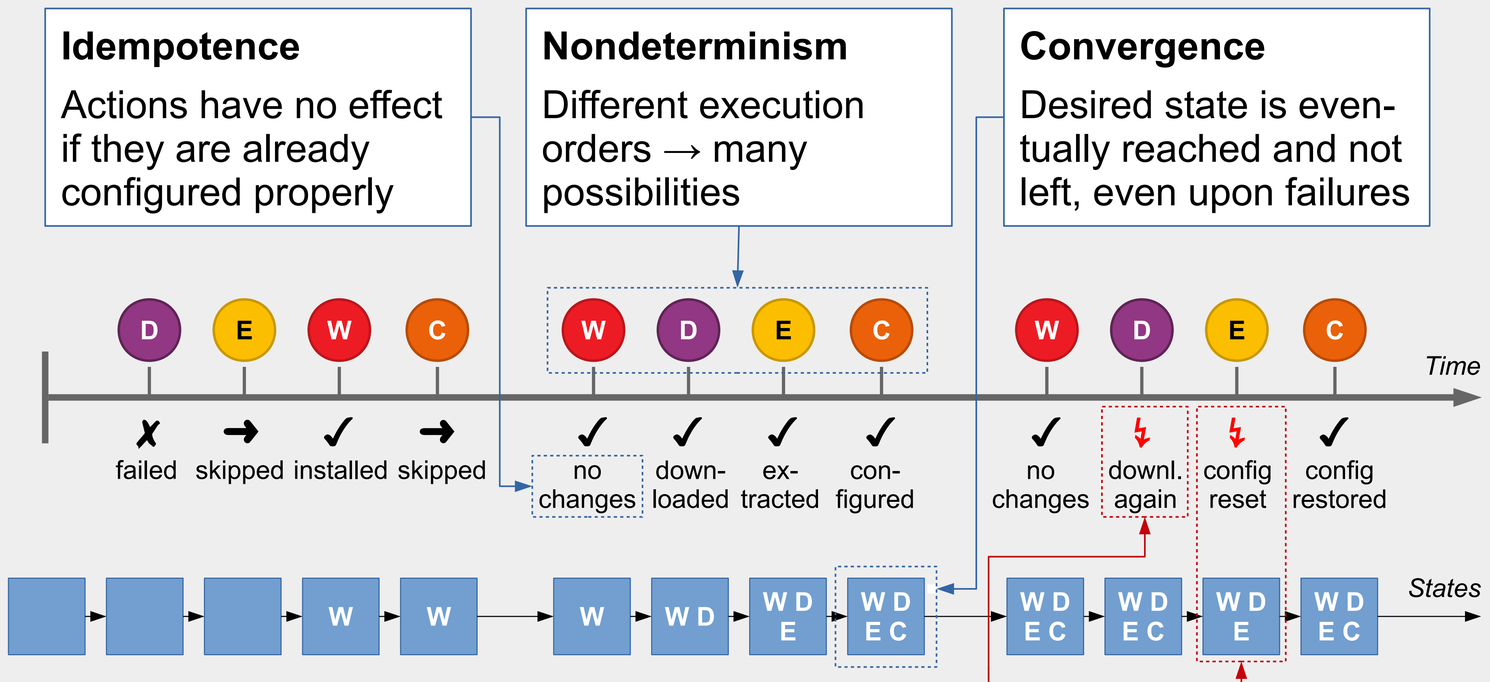


Goal: Systematic approach for automatically testing configuration scripts for idempotence and convergence.

Idempotence
Actions have no effect if they are already configured properly

Nondeterminism
Different execution orders → many possibilities

Convergence
Desired state is eventually reached and not left, even upon failures



Violation of Idempotence
Web package is downloaded again although it was already.

Violation of Convergence
Action E has undone the configuration of action C.

Formal Model

Precise definitions for the concepts introduced above were created.

Major results:

Resource Satisfaction
An action is *satisfied* if it is configured properly, hence if its re-execution has no effect.

Preservation Property
An action *preserves* another one if the former does not undo the effect of the latter.

Resource Preservation Theorem
Preservation of ancestors and non-related actions *implies* convergence.

Test Method

Testing preservation **pairwise** along certain partial script executions.

Foundation for pairwise testing:
Resource Preservation Theorem

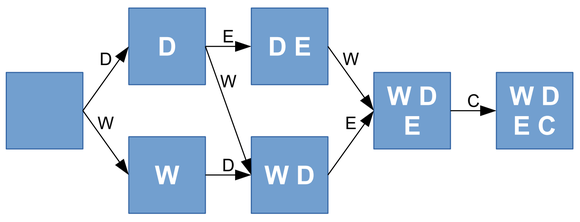
Testing Primitives

Satisfaction Assertion
Action is re-executed and changes tracked:
- no changes → action is satisfied
- changes detected → action is *not* satisfied

Idempotence of action A
1. execute A
2. assert A

Preservation of action B by A
1. execute B
2. execute A
3. assert B

State Transition Graph



Selected Paths (edge coverage)
1. D → E → W → C
2. W → D → E
3. D → W

Test Cases (i) … idempotence test, (p) … preservation test

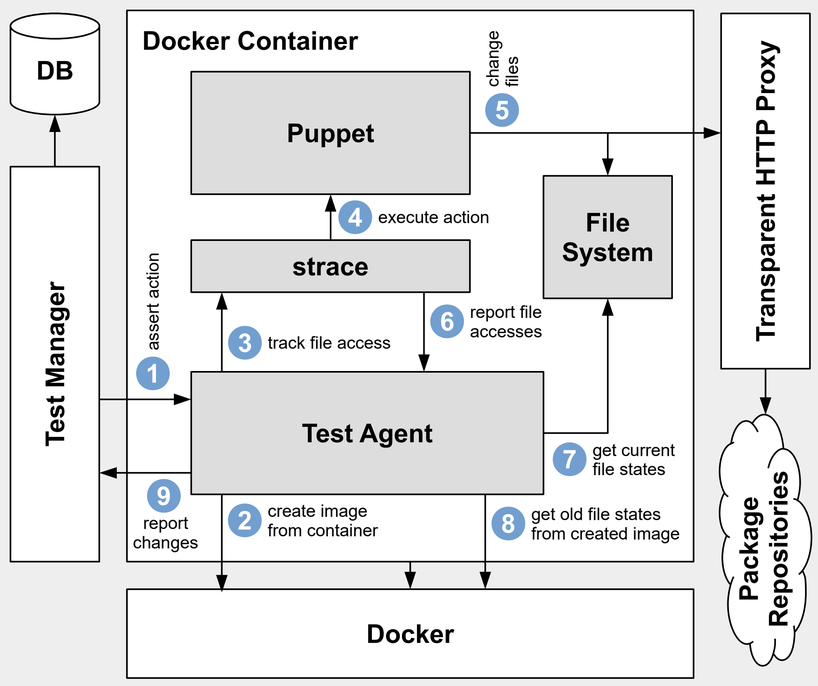
TC 1	TC 2	TC 3
1. exec D (i) assert D	1. exec W (i) assert W	1. exec D (i) assert D
2. exec E (i) assert E (p) assert D	2. exec D (i) assert D (p) assert W	2. exec W (i) assert W (p) assert D
3. exec W (i) assert W (p) assert D (p) assert E	3. exec E (i) assert E (p) assert D (p) assert W	
4. exec C (i) assert C (p) assert D (p) assert E (p) assert W		

Prototype

- Based on configuration management tool **Puppet**
- Tests are executed in Docker containers (lightweight virtualized environment)
- Test process is **fully automated**
- Squid (HTTP proxy) used to cache frequently downloaded packages

Change Tracking 1 – 9
- Mechanism is tool agnostic
- strace for tracking file access
- copy-on-write file system of Docker used to store old file copies

Architecture



Evaluation

✔ The **test method effectively detects issues in real world** community contributed scripts:
- all issues in a set of scripts with known issues (11 out of 11, 5 real world, 6 constructed)
- some hitherto unknown issues in a large set of real world scripts (5 out of 88)
- Evaluation Summary:

Test Data		Execution Statistics	
Real world scripts	88	Test Steps	132,507
Actions	858	Execution Time	125.15 hours

✔ The test method is **applicable** to a wide array of configuration management tools (evaluated by literature review).

Kontakt: oliver.hanappi@gmail.com