ConfValley: A Systematic Configuration Validation Framework for Cloud Services

Peng Huang†, William J. Bolosky‡, Abhishek Singh*, and Yuanyuan Zhou†
UCSD†, Microsoft Research‡, Microsoft*

Background

Configuration in cloud-scale systems
- Wide variety
  - Format: XML, INI, YAML, REST API, etc.
  - Control: feature, security, fault tolerance, performance, etc.
- Interwoven impact
  - Config. in one component can impact others
- Duplication & customization

Problem & Motivation

Configuration validation
- What: checking if configuration satisfies some constraints
- When: configuration edit-time, integration, production
- Benefits: prevent damages from production, save diagnosis
- Current practice: inefficient, ad-hoc, and late
  - Manual configuration review
  - Bulky ad-hoc scripts/code in different places
  - Invoked late in runtime

Solution

ConfValley: a config. validation framework
- Language: a declarative language (CPL) to write config. specs
- Abstraction: unified config. representation
- Inference: infer basic specs automatically
- Validation: support different scenarios,
  - Edit-time, in IDE
  - On the fly, an interactive console
  - Continuous service, run specs when config. changes

Evaluation

Quick comparison

Example CPL specs

Rewrite existing validation code

Prevent misconfiguration

Performance