TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Sharing Software Configuration via Specified Links and Transformation Rules

Markus Raab

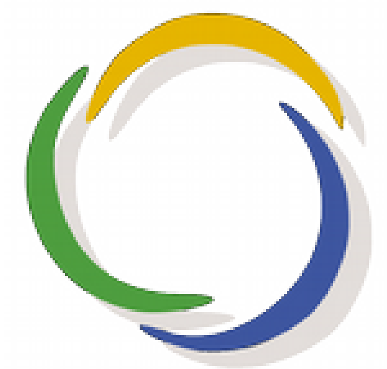
Vienna University of Technology

Institute of Computer Languages, Austria

Email: markus.raab@complang.tuwien.ac.at

Outline

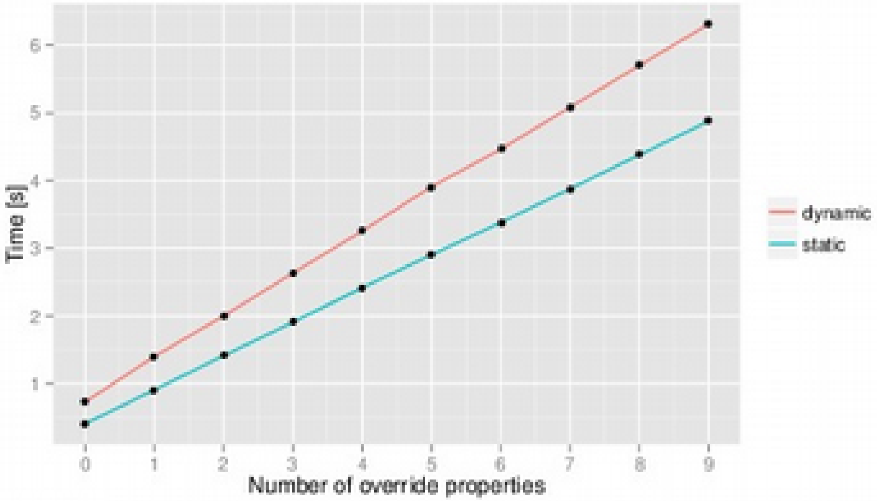
- **Motivation**
- **Demo**
- **Evaluation**
- **Conclusion**



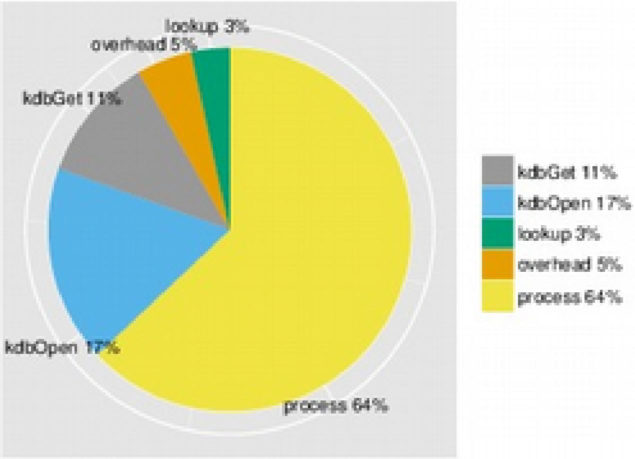
Elektra's Logo

Evaluation

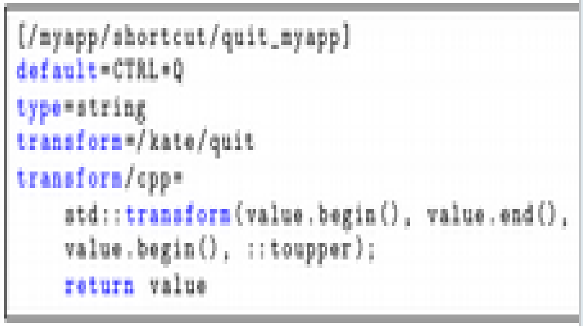
- two implementations
  - dynamic
  - static
- result: dynamic
  - slower by factor 1.8
  - time grows 22% faster



comparision



static variant



```
[/myapp/shortcut/quit_myapp]
default=CTRL+Q
type=string
transform=/kate/quit
transform/cpp=
    std::transform(value.begin(), value.end(),
    value.begin(), ::toupper);
    return value
```

Conclusion

- integration of existing configuration files
- links between any configuration items
- specification is only key/value data
    - access by tools
    - enforced on access
    - configurable specification
- contributions
    - access to others application configurations
    - implementations available
    - performance measurements for static/dynamic
    - case study

Thank you for your attention!

Markus Raab

Vienna University of Technology

Institute of Computer Languages, Austria

Email: markus.raab@complang.tuwien.ac.at

Benchmark Setup

- Laptop: hp ® EliteBook 8570w ™
  - CPU Intel ® Core i7-3740QM @ 2.70GHz
  - 7939 MB Ram
- GNU/Linux Debian Wheezy 7.5
- gcc compiler Debian 4.7.2-5
  - with the options -std=c++11, -O2
- measured the time using **`gettimeofday`**
- Median of eleven executions