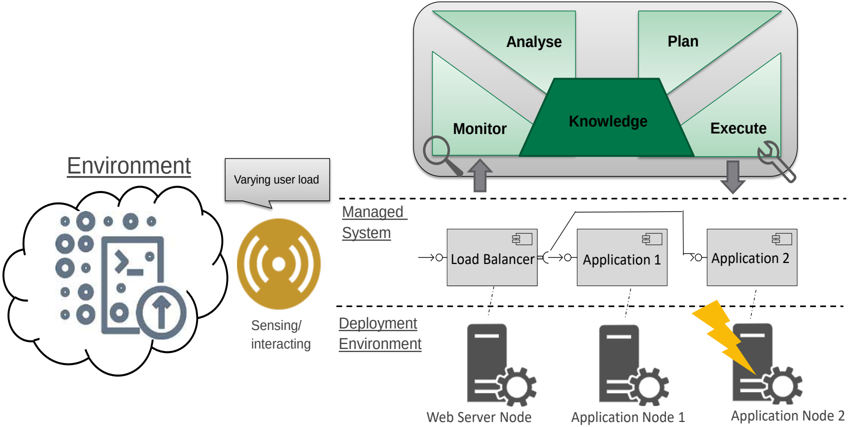
Design-time Performability Evaluation of Runtime Adaptation Strategies

ICPE 2023 - 14th ACM/SPEC International Conference on Performance Engineering

Martina Rapp, Max Scheerer, Ralf Reussner | April 15-19, 2023 Coimbra, Portugal

Motivation

Evaluate performability-oriented runtime adaptation strategies at design-time (DT) to support SW architects in the decision making process of self-adaptive software systems (SAS)s.



Contribution

- **Focus of previous work [1]:** framework development
  - Modeling the operating environment of an Self-adaptive software System (SAS)
  - Analysis approach that evaluates the quality of an adaptation strategy w.r.t. the modelled environment and specific quality objectives
- **Focus of current work:** reuse the concepts of [1] to evaluate adaptation strategies w.r.t. performability-specific quality objectives
- We contribute to the line of work by:

C1 **Using SAS to maintain performability attributes of a system:** We generalize formally and informally existing and well-known work from the performability domain to SAS.

C2 **Simulation of failure scenarios:** A simulation approach to predict performance attributes which reflect the quality objectives the SAS must maintain in the presence of system failures.

C3 **Performability-specific adaptation strategy evaluation:** We integrate performability metrics from literature as performance indicators to assess the quality of an adaptation strategy.

Approach - Big picture

FZI

Current state:
Performability evaluation traditionally applied for DT analyses of static systems [2]

→

Problem:
Structural changes of system configuration over time not covered

→

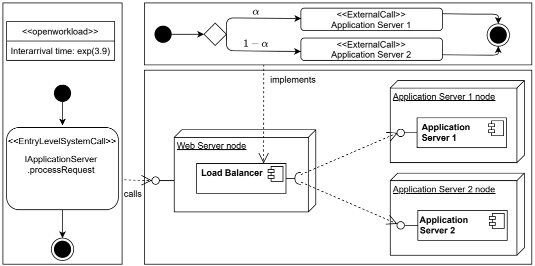
Idea:
Extend work on dynamic reconfigurable systems of [3] to SASs by extending MRMs to MDPs

Solution:

- i Formalize the extension of Markov Reward Models (MRM) to Markov Decision Process (MDP) in the context of SASs
- ii Identify uncertainty factors in the performability domain and their representation in the SASs environment
- iii Construct the reward function based on classic performability metrics
- iv Simulation of failure scenarios
- v Integrate (i) - (iii) into MBQA framework SimExp [1] and use (iv) for failure simulation

Proof of concept - Case study and setup

Case study: Znn.com system based loadbalancer [4]



Setup: We invested three adaptation strategies

- $\pi_\emptyset$ without any adaptation: static system
- π_s scaling: re-distributes load by adapting the distribution factor
- π_{sr} scaling with node recovery: re-distributes load while considering two factors:
  - (i) detected runtime threshold violations as described by π_s
  - (ii) detected node failures

- **Quality objective**: Performability -> performance under the impact of hardware failures
- **Goal**: Strategy π to keep system responsive under high load scenarios with a minimum number of resources even in case of failures
- **Expected result**: Comparison of adaptation strategies / Evaluating design decisions w.r.t. quality objectives

Proof of concept - Results

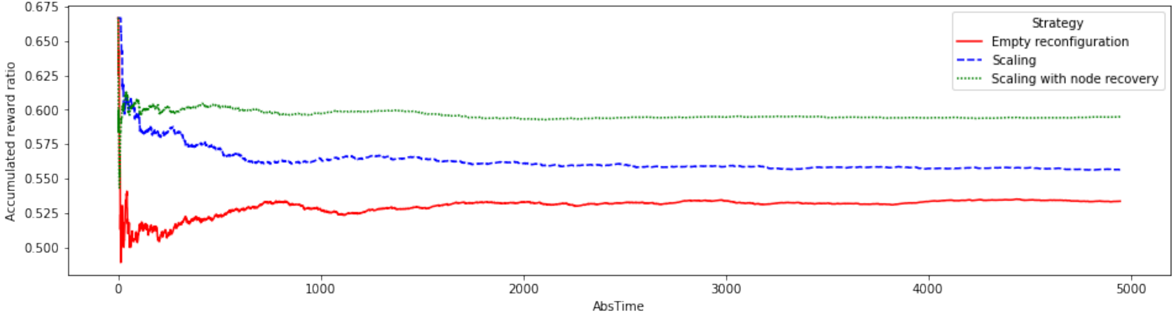


Figure: Total accumulated reward of strategies

Evaluation:

- Sampling 50 trajectories length 100:
  Total number of sampled states T = 5000
- Reward ratio: $ratio_{\pi}(t) = \frac{1}{2t} \cdot \sum_{i=0}^{t} r_i$

Results:

- $ratio_{\pi_{\emptyset}}(T) < ratio_{\pi_S}(T) < ratio_{\pi_{SR}}(T)$
- π_{SR} performs best and $\pi_{\emptyset}$ worst

Conclusion

FZI

Summary

- We presented a model-based approach to evaluate performability-oriented runtime adaptation strategies at DT
- Our contribution combines established performability concepts with an MBQA evaluation framework for runtime adaption strategies to support DT analysis of performability-oriented use cases
- This enables to:
  - evaluate DT decisions of runtime artifacts
  - gain knowledge from DT analysis applicable at runtime as initial system configuration

Future work

- Evaluate our approach with a case study to show the accuracy of DT results compared to runtime results
- Extend the SimExp framework for DT performability optimization of runtime adaptation strategies as presented in [5]

Thank you!

Your questions, comments and suggestions concerning our research activities are most welcome at any time.

Kindly write to:

rapp@fzi.de
scheerer@fzi.de

Literature I

[1] M. Scheerer, M. Rapp, and R. Reussner, “Design-time validation of runtime reconfiguration strategies: An environmental-driven approach,” in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, IEEE, Aug. 2020. DOI: 10.1109/acsos49614.2020.00028.

[2] B. R. Haverkort, *Performability modelling : techniques and tools*, B. R. Haverkort, Ed. Chichester [u.a.]: Wiley, Apr. 18, 2001, 338 pp.

[3] V. Grassi, R. Mirandola, and A. Sabetta, “A model-driven approach to performability analysis of dynamically reconfigurable component-based systems,” in *Proceedings of the 6th international workshop on Software and performance - WOSP ’07*, ACM Press, 2007. DOI: 10.1145/1216993.1217011.

[4] S.-W. Cheng, D. Garlan, and B. Schmerl, “Evaluating the effectiveness of the rainbow self-adaptive system,” in *2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*, IEEE, 2009. DOI: 10.1109/seams.2009.5069082.

[5] M. Rapp, M. Scheerer, and R. Reussner, “Design-time performability optimization of runtime adaptation strategies,” in *Companion of the 2022 ACM/SPEC International Conference on Performance Engineering*, ser. ICPE ’22, Bejing, China: Association for Computing Machinery, 2022, pp. 113–120. DOI: 10.1145/3491204.3527471.