

Modeling vs. Details

Sebastian Lehrig, Steffen Becker, Robert Horrox, Michal Kit, Miao Wang,
Murray Woodside, Takfarinas Saber

Interests

- How to model data centers?
 - VMs, PMs, SLA/SLOs, networks, software components & their interactions
 - How to audit data centers? Problem spotting, SLA/SLOs, etc.
- Analytical modeling technologies (“undefined environment”)
 - How to derive them? (data mining, regression, calibration, ...)
 - How to exploit them? (e.g., for cloud management, software design, ...)
 - How to cope with uncontrolled factors (e.g., delays, users, data intensiveness, ...)? How to adapt to them? Elasticity?
- Reusability
 - Models? Calibrations?
 - How to decompose models?
- Systems with highly dynamic architectures, e.g., wrt. loss of connection (“undefined system”)
- Multi-tenancy, multiple/new/time-depending workloads
- DevOps vs. Code vs. (Business-)Management: Interactions?

Use of Models

- Building/deriving Models
- Answer: What is the goal of the model?
 - Identify use-cases! Also business cases! Success stories!
 - Think about different stakeholders!
 - Static vs. dynamic? Steady-state? ...
 - How long does it take to model?
- Ideas
 - Modeling customers. E.g., personas
- Problems
 - Risks assessed too low at the beginning
 - Performance group too invisible
- Outlook
 - DevOps seems to improve situation

Reuse of Models

- Works good for
 - DevOps (e.g., RightScale)
 - Performance regression testing
 - Dynamic Spotter (SAP): Micro-benchmarks to detect performance anti-patterns
- Problem: Calibration reuse
 - Lots of parameters, complex models vs. simple model, complex system
- If you do performance regression, adding models is close
- Using models based on regressions seems to be acceptable by companies
- Start with easy models (SCRUM: 2 weeks!), don't do everything at once
- “Modeling has to be hidden behind an interface that everybody understands” (UML? ATAM? ...)