Statistical static capacity management in virtualized data centers supporting fine grained QoS specification

Talk at e-Energy 2010

Speaker: Marko Hoyer

Marko Hoyer
OFFIS Institute for Information Technology
marko.hoyer@offis.de

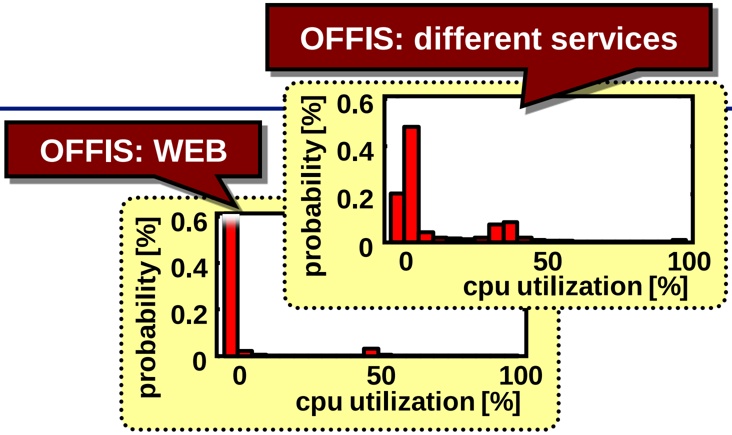
Kiril Schröder
C.v.O. University of Oldenburg
kiril.schroeder@informatik.uni-oldenburg.de

Wolfgang Nebel
C.v.O. University of Oldenburg
nebel@informatik.uni-oldenburg.de

2 Motivation 1 / 4

Wasted energy through low server utilization

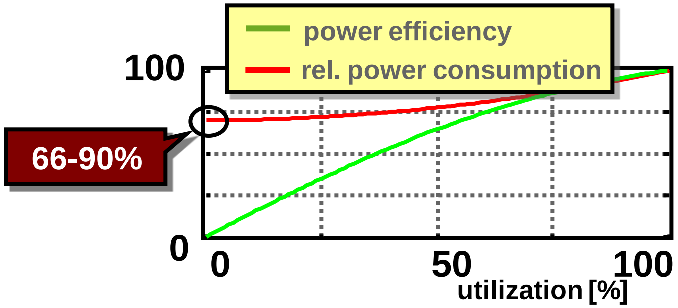
- Server utilization in small or medium enterprises (e.g. OFFIS ≈ 50 servers)
  - **90th percentile** at the web server: **2% CPU utilization**
  - **90th percentile** at another server deploying different services: **39% CPU utilization**



OFFIS: different services

OFFIS: WEB

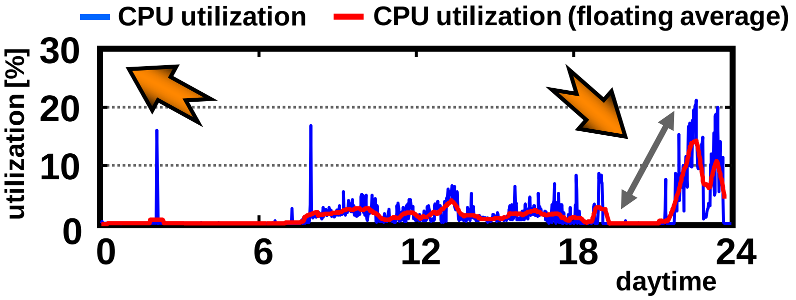
- Server utilization in large efficient server clusters
  - **90th percentile** of 5000 Google servers averaged[1]: **50-60% CPU utilization**



66-90%

- Problem: **Utilization ≁ power consumption**

- Main reasons for under utilization
  - **Hardware to powerful** for single services
  - **Daytime dependent variation** of workload (and thus resource demand)



---

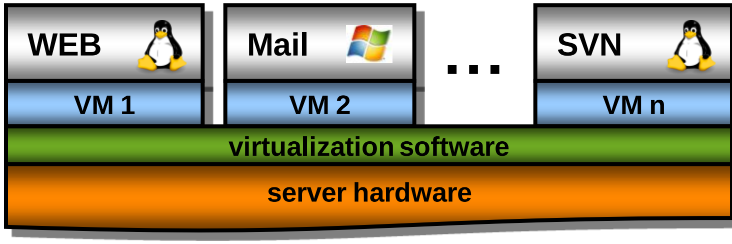
[1] estimated from: Barroso, L. A.; Hölzle, U.: *The case of energy-proportional computing*. IEEE Computer 40(12).2007

3 Motivation 2 / 4

Basic idea of static statistic capacity management

- Optimizing utilization using **virtualization technique**
  - Different services deployed at one server

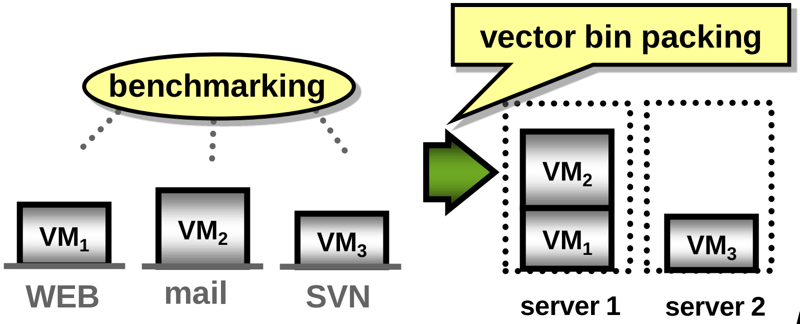
- Questions to be answered
  - **How much resources must be reserved for a service?**
  - **Which services are deployed together at one server?**



WEB | Mail | … | SVN
VM 1 | VM 2 | VM n
virtualization software
server hardware

pessimistic approach

- Determine maximal required resources
- Distribute services to servers **guaranteeing maximal required** resources all the time



vector bin packing

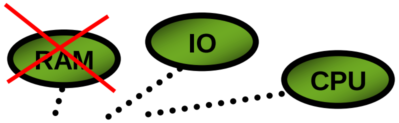
benchmarking

VM$_1$ WEB
VM$_2$ mail
VM$_3$ SVN

VM$_2$, VM$_1$ server 1
VM$_3$ server 2

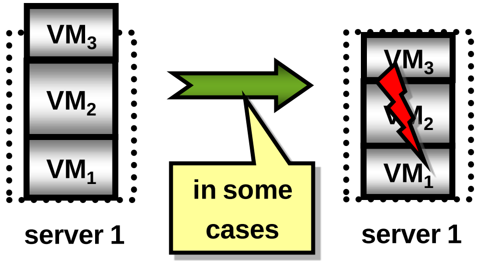
optimistic approach

- **Overbook resources**



~~RAM~~
IO
CPU

- **Resource shortage** in some cases
  - For resource types that allow trading off between amount of resources and performance
- **Limits** that can be set for each service
  - **Probability** of resource shortage
  - **Strength** of resource shortage

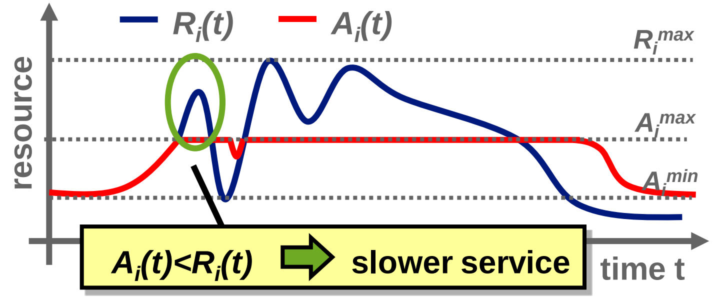


VM$_3$, VM$_2$, VM$_1$ server 1
in some cases
VM$_3$, VM$_2$, VM$_1$ server 1

4 Motivation 3 / 4

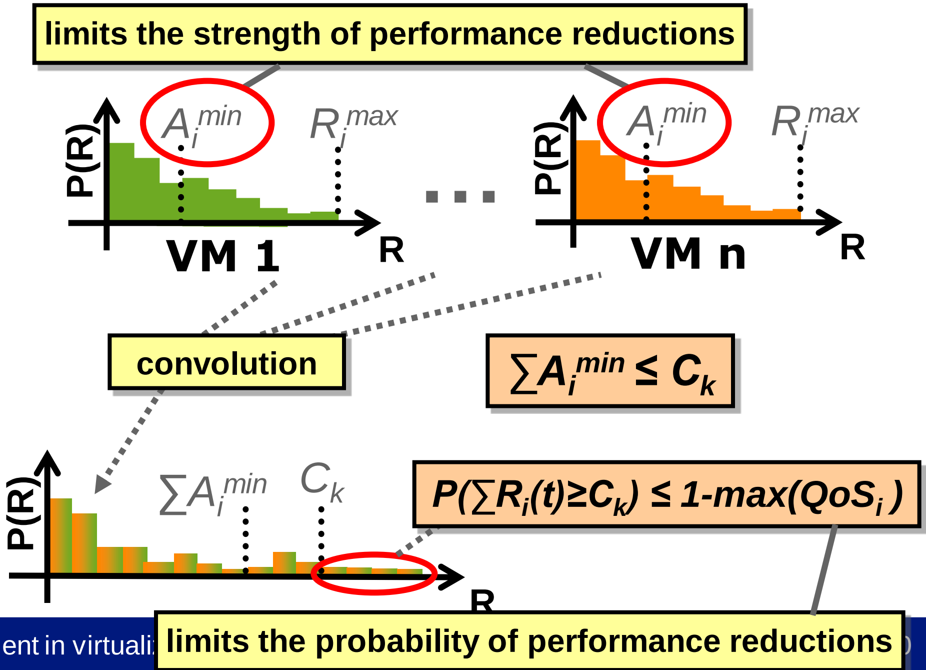
Known static statistical approaches

- Demand, allocated resources and server capacity
  - For each VM *i* (virtual machine)
    - **resource demand:** $R_i(t) \leq R_i^{max}$
    - **allocated resources:** $A_i^{min} \leq A_i(t) \leq A_i^{max}$
  - For each Server *k* (host)
    - **Resource capacity:** C_k



$A_i(t) < R_i(t)$ ⇨ **slower service**

- Statistical allocation planning
  - first pessimistically **plan by R_i^{max}** (vector bin packing)
  - create **histograms** from observed demand
  - create optimized allocation plan w. r. t. the demand behavior
    - **ensuring minimal resources A_i^{min}**
    - **ensuring minimal probability QoS_i** of not having performance problems
  - vector bin packing considering both conditions



limits the strength of performance reductions

convolution

$\sum A_i^{min} \leq C_k$

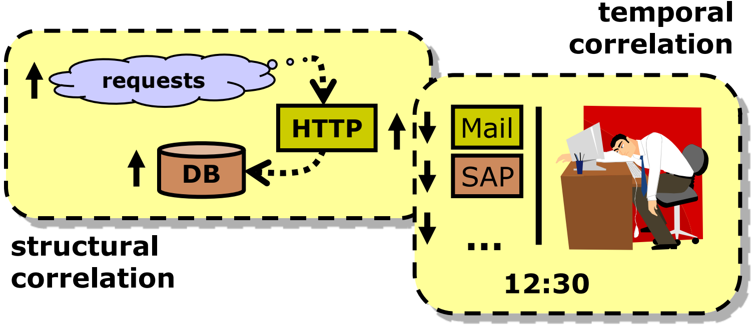
$P(\sum R_i(t) \geq C_k) \leq 1\text{-}max(QoS_i)$

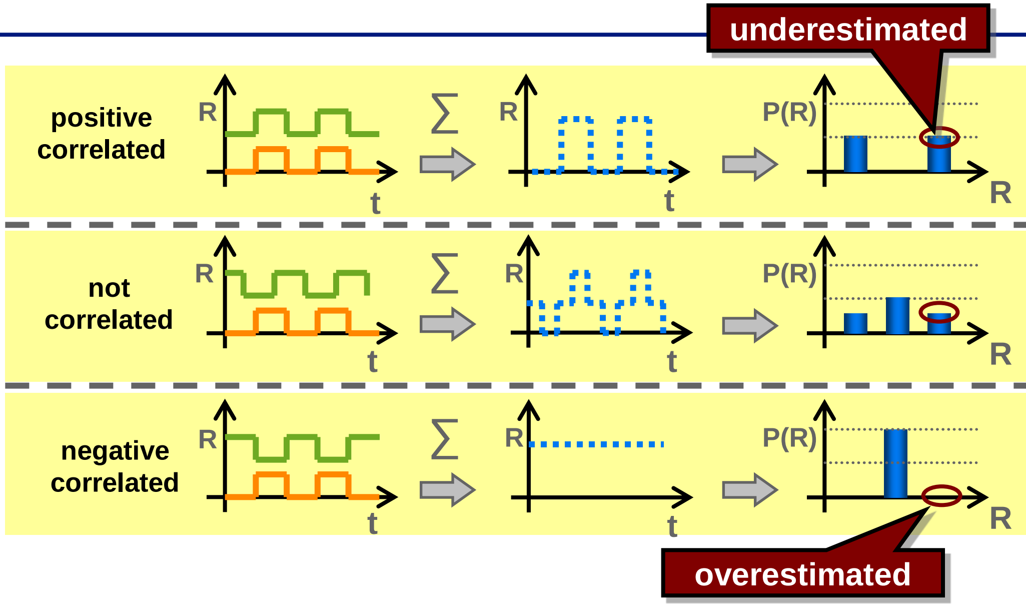
limits the probability of performance reductions

Motivation 4 / 4

Problems of known approaches

- **Convolution requires statistical independence**





- **Inflexible QoS** specification
  - Only one pair of QoS_i and A_i^{min} possible

example

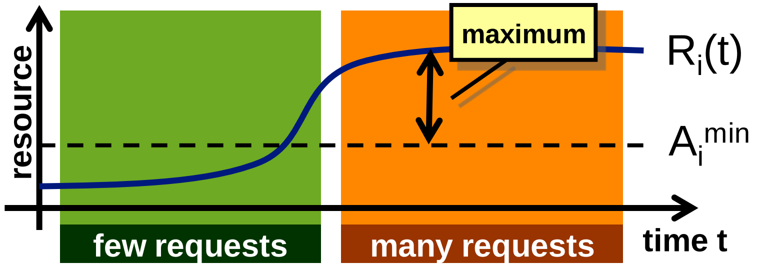
response time	≤100ms	≤200ms	≤500ms	≤800ms
A_i^{min} (e.g. rel. cpu)	0.5	0.25	0.1	0.06
QoS_i	0.50	0.80	0.90	0.99

strongly limit strength

OR

strongly limit occurrence

- **Strength of resource shortage depends on resource demand**

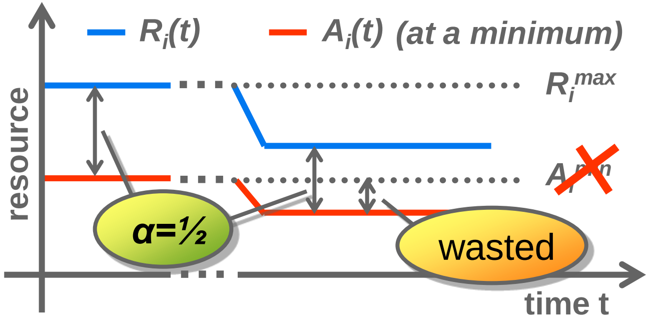


6 Concept 1 / 3

Demand independent fine grained QoS specification

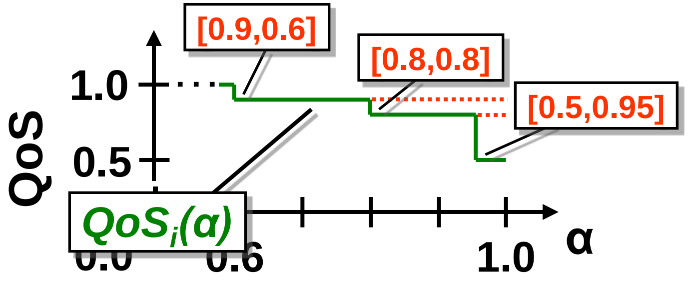
- Independence of resource demand
  - Describe resource shortage as **ratio α between allocable and required resources**
  - Not reserving fixed resources but **guaranteeing** a **minimal α**

$$\alpha_i(t) = \frac{A_i(t)}{R_i(t)}$$



- More flexible QoS specification
  - **Old** one: **1 pair** of *QoS* and *α*
  - **New** one: **function** *QoS* of *α*
    - **For each *α***, an **independent QoS** value can be defined
    - **More optimistic planning** possible
  - For each VM it must hold that:

$[\cancel{QoS_i, \alpha_i}] \Rightarrow QoS_i(\alpha)$



$$\forall \alpha < 1.0 : P\left(\frac{A_i(t)}{R_i(t)} < \alpha\right) < 1 - QoS_i(\alpha)$$

- Two **allocation approaches** in the paper
  - one for **uncorrelated workload**
  - one for **correlated workload**

7 Concept 2 / 3

Dealing with correlations – pessimistic approach

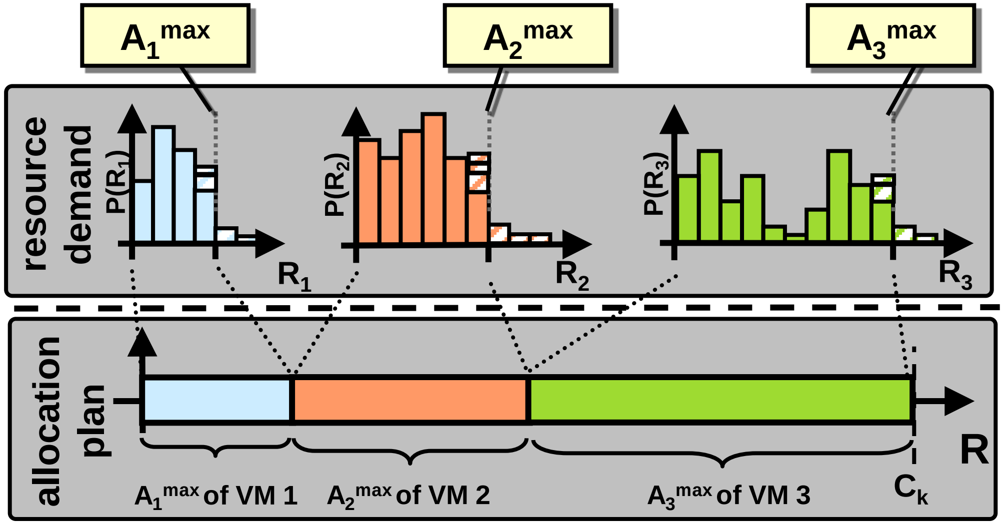
- Reserve **individual resource capacity** for each VM
  - No interactions between VMs possible
  - **Correlations** can be **neglected**

$$\forall \alpha < 1.0 : P\left(\frac{A_i^{\max}}{R_i(t)} < \alpha\right) < 1 - QoS_i(\alpha)$$

- **Reserved capacity** can be **lower than** the **maximum demand** of the VM ($R_i^{max} > A_i^{max}$)
  - Consequences: resource shortage in some cases
- Approach
  - **Select A_i^{max}** so that the **QoS specification** of the respective VM **is met**
  - Bin packing by A_i^{max}

$$\sum A_i^{\max} \leq C_k$$

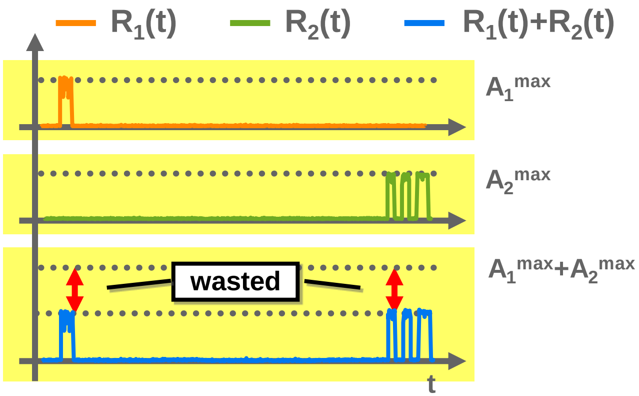
- Disadvantage
  - **Assume** having **fully positively correlated** resource **demand**
  - Wasted resources



Concept 3 / 3

Dealing with correlations – optimistic approach

- More optimistic approach
  - **Joint resource demand** of all VMs must **never exceed** the **servers' capacity**



$$\cancel{\sum A_i^{max} \le C_k}$$

$$\forall t : \sum R_i(t) = JR_k(t) \le C_k$$

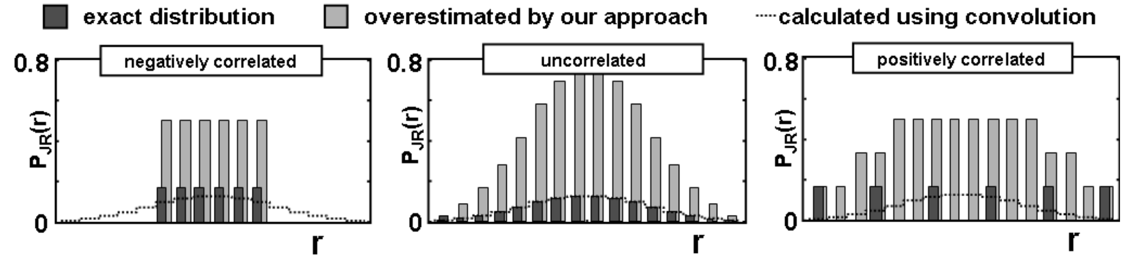
$$P(JR_k(t) > C_k) = 0$$

$$\int_{C_k}^{\infty} P_{JR_k}(s)ds = 0$$

probability distribution of joint resource demand

- Some slides ago …
  - **Probability distribution of joint resource demand** is derived from the individual distributions using **convolution**
  - **Problem:** Statistical independence not given when having **correlations!!**

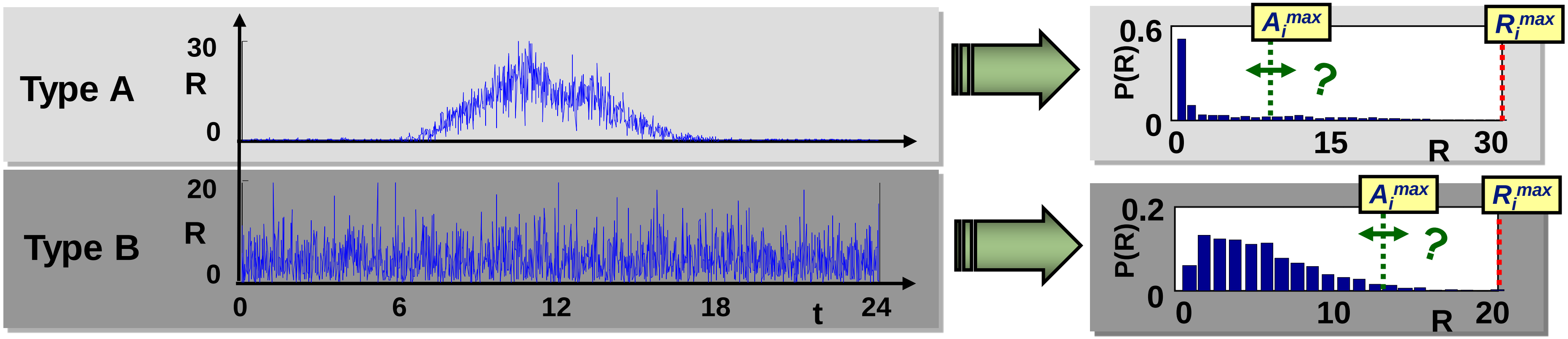
- Solution
  - **Overestimation**
  - Details in the paper …



9 Evaluation results 1 / 3

Resource saving when using fine grained QoS specification

Evaluation data

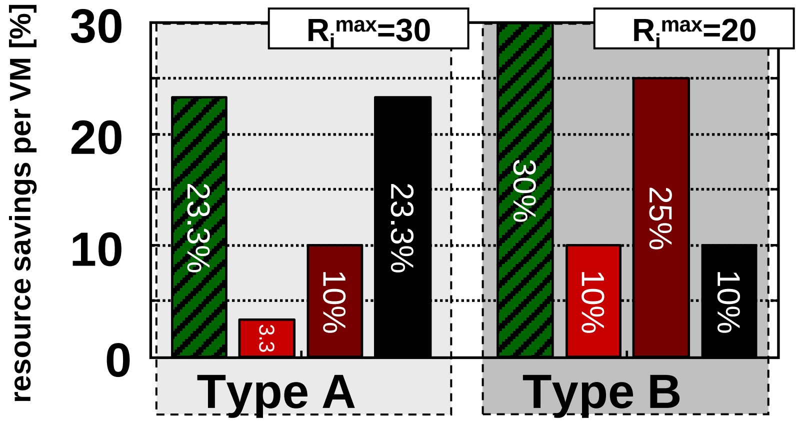


Fine grained QoS specification

	Type A		Type B	
	A_i^{min} / α	QoS_i	A_i^{min} / α	QoS_i
🟥	*28 / 0.95*	*0.5*	*18 / 0.90*	*0.85*
🟫	*27 / 0.90*	*0.9*	*14 / 0.70*	*0.95*
⬛	*15 / 0.50*	*0.99*	*10 / 0.50*	*0.99*

▨ *fine grained QoS (contains all three QoS / α pairs from above)*

Savings



Evaluation results 2 / 3

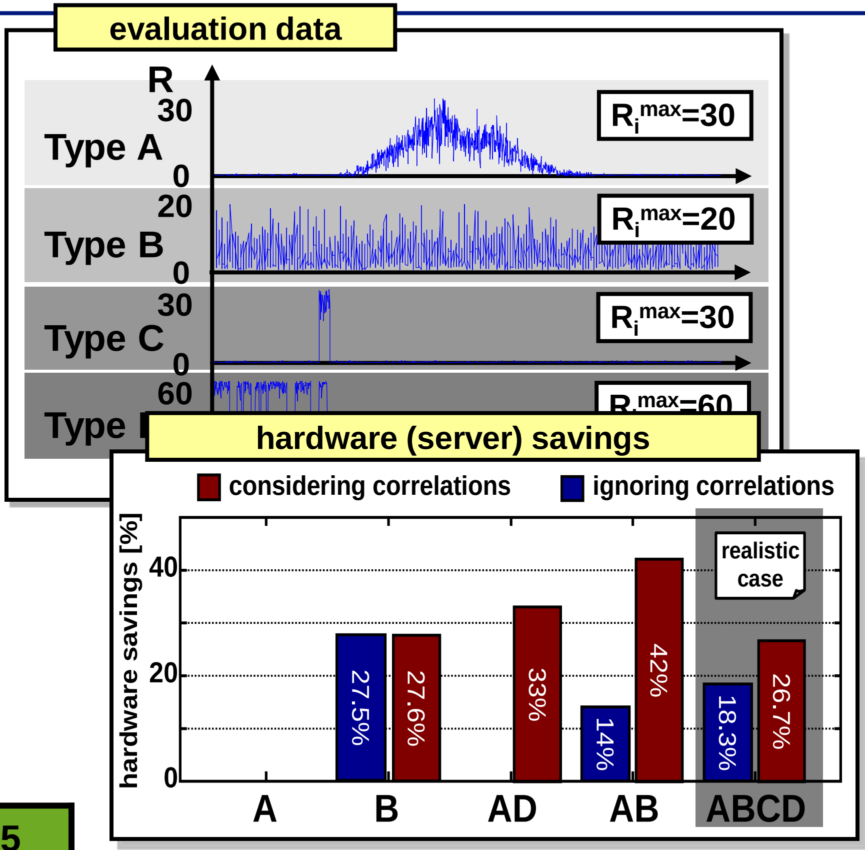
Resource saving when using fine grained QoS specification

- **Simulation based evaluation**
  - 4 workload types, 200 VMs
- **Relative hardware savings**
  - Count of servers **compared to pessimistic approach** (guaranteeing max. demand all the time)
- **Energy savings** in data center
  - Best case (PUE remains constant)
    - **Hardware savings = Energy savings**

≈ 27%

  - Worst case (only server power is saved)

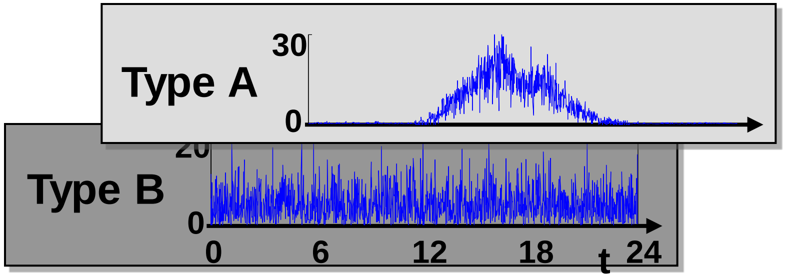
PUE (pessimistic approach)	**1.5**	**2.0**	**2.5**
PUE (statistical approach)	*1.7*	*2.3*	*3.0*
Energy savings	**19%**	**13%**	**11%**



11 Evaluation results 3 / 3

QoS violations when ignoring correlations

- **Methodology**
  - As many **VMs of same type** as possible placed at one server
  - **Statistical approach using convolution**
  - Demand traces of one day workload
  - Strength and frequency of **resource shortage measured** and compared to QoS
- **Analyzed cases**
  - Fully positively correlated workload (**cor. coef. 1.0**)
    - **Workload Type A**, exactly the same time series are used
  - Mainly positively correlated workload (**cor. coef. 0.87**)
    - **Trend of workload Type A**, random noisy part
  - Uncorrelated workload (**cor. coef. 0.0**)
    - **Workload Type B** (random)
- **Results**
  - **Violations of up to 4x the specified** one when having positively correlated workload



P(α(t)<a) for VM of Type A (cor. coef.: 1

down to a=0.45

	α ≥0.95	α≥0.9	α≥0.5	α<0.
must is	≤0.5 0.1	≤0.1 0.1	≤0.01 0.08	=0 0.01

8x

P(α(t)<a) for VM of Type A (cor. coef.: 0.87)

	α ≥0.95	α≥0.9	α≥0.5	α<0.5
must is	≤0.5 0.06	≤0.1 0.04	≤0.01 0.04	=0 0.0

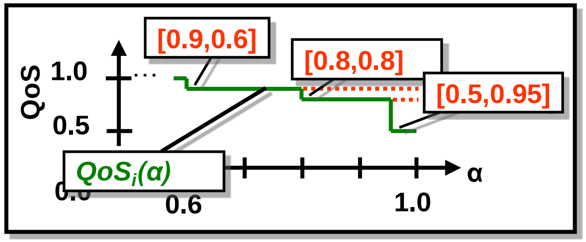
4x

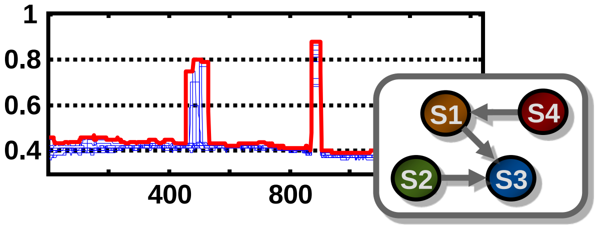
P(α(t)<a) for VM of Type B (cor. coef.: 0.0)

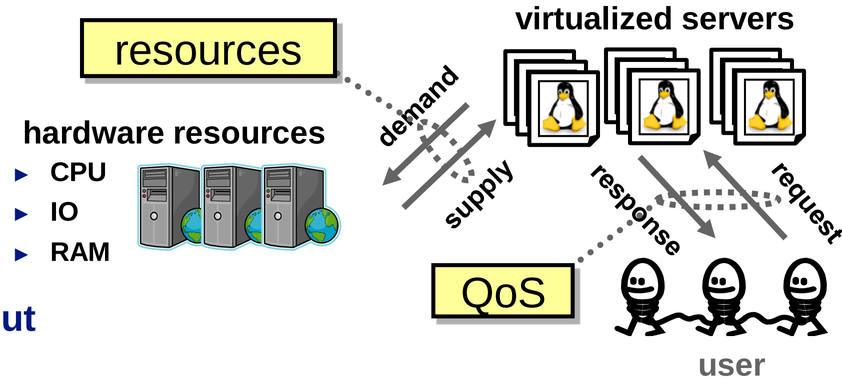
	α ≥0.9	α≥0.7	α≥0.5	α<0.5
must is	≤0.15 ≈0.0	≤0.05 ≈0.0	≤0.01 ≈0.0	=0 =0.0

Summary & Ongoing and required future work …

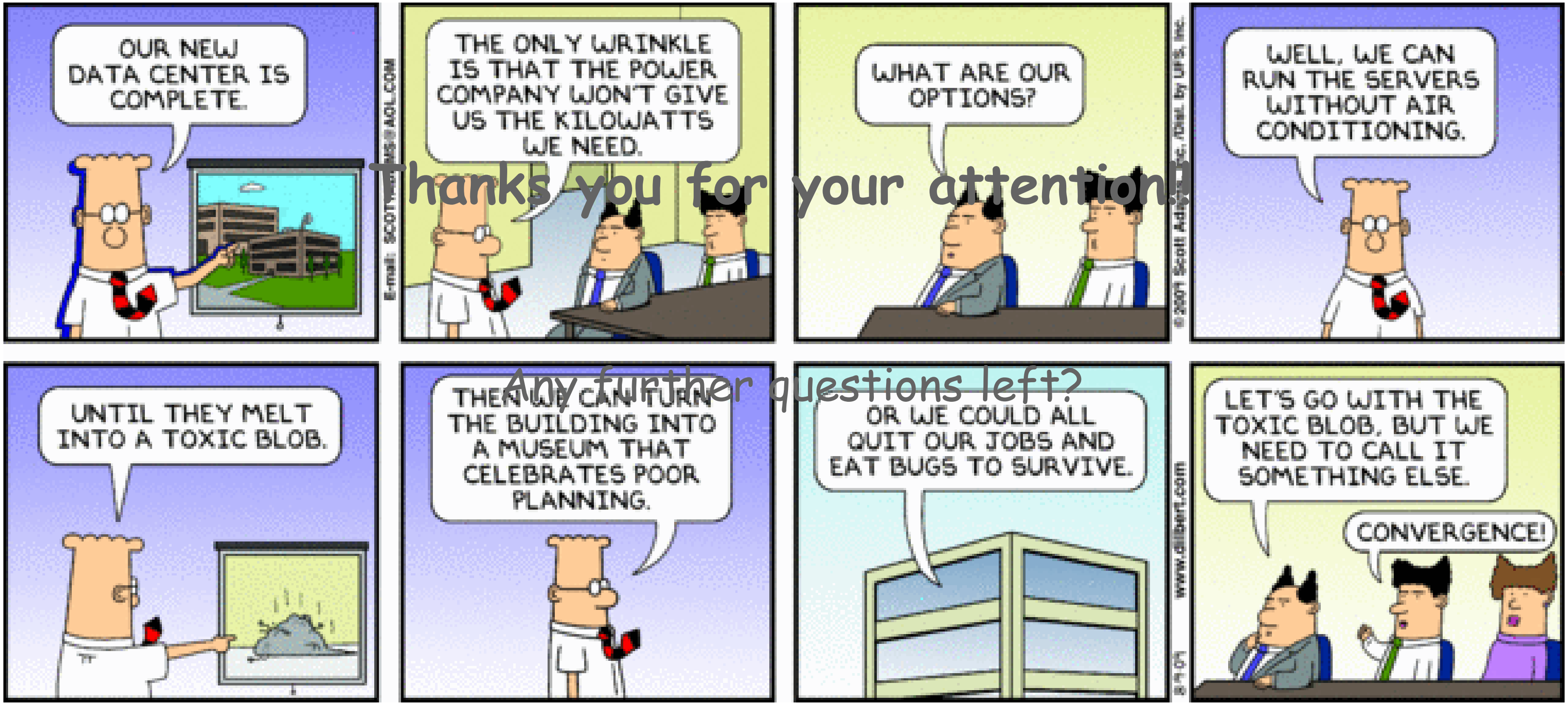
- **Main outcomes in this paper**
  - Statistical static allocation approach
    - Fine grained **trade off between performance and hardware resources** (energy)
    - Pessimistically and optimistically **dealing with correlations**
  - Evaluation results
    - Up to **27% resource (energy) savings** in our example
    - Ignoring correlation will lead to significant QoS violations
- **Ongoing work**
  - Just finished an **dynamic allocation approach** for VMs
    - Uses **life migration and server standby**
    - Ensures meeting fine grained QoS at any time
    - Ensures **redistributing the VMs right in time**
- **Required Future Work**
  - Until now, resource demand and supply are adjusted as good as possible
  - **Real QoS Attributes** are **response time or throughput**
    - Mapping them to resource demand is required







13 Questions & Discussion?



Thanks you for your attention!

Any further questions left?