Predicting Web Service Levels During VM Live Migrations

5th International DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standards and the Cloud

Helmut Hlavacs, **Thomas Treutner**

24. 10. 2011

Table of Contents

Abstract

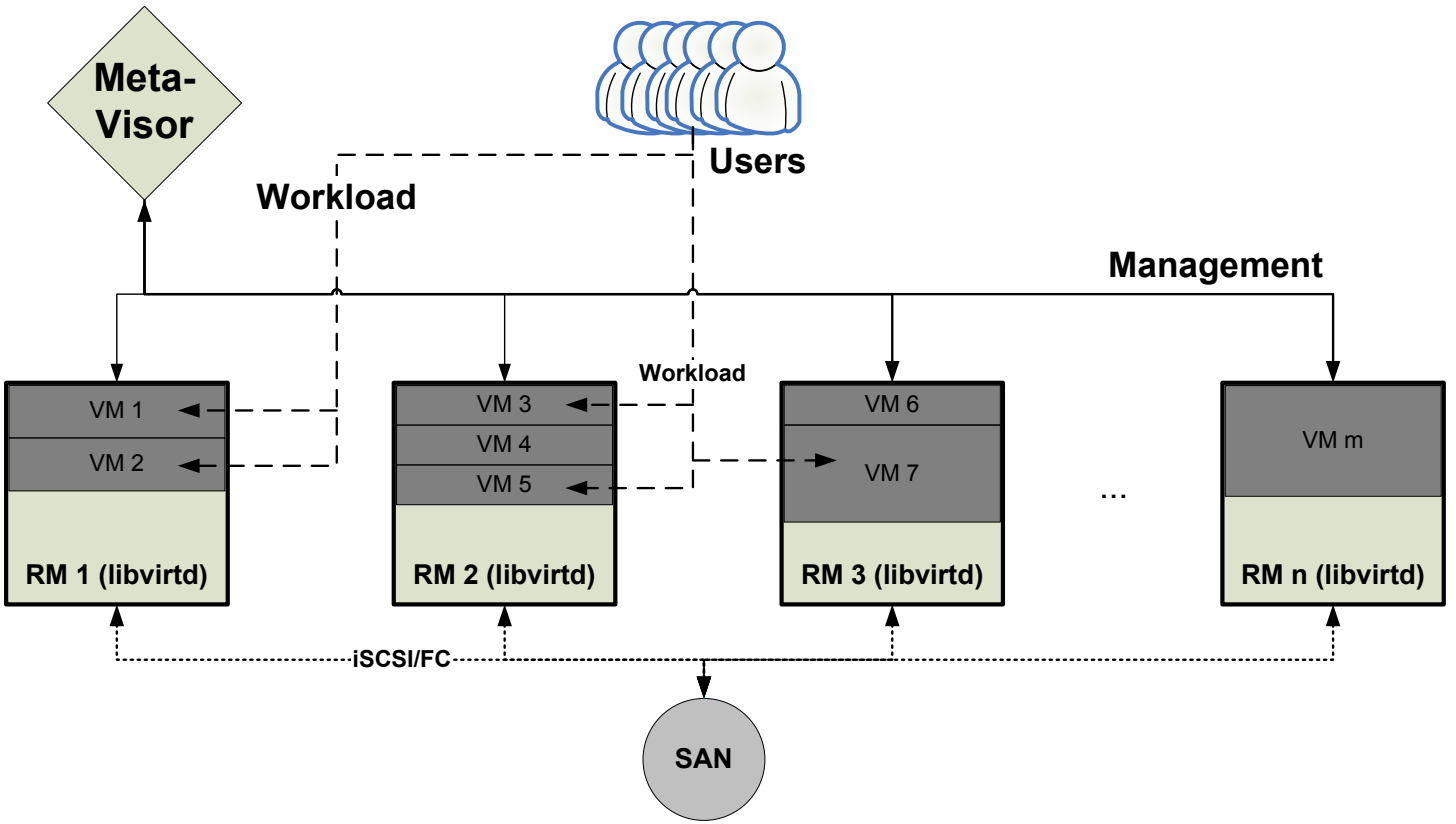
- **VM live migration important for energy efficiency**
- Establish energy efficient target distribution of VMs
- No perceivable downtime while live migrating?
- **Live migration is resource intensive** (iterative page copying)
- **Experiments: Influence on service levels while migrating?**
- **Modelling: Predict service levels based on utilization?**

Scenario

Scenario

- Virtualized data center, static consolidation (P2V)
- Provisioning for peak load, still bad energy efficiency
- E.g., 9-5-cycles, very low utilization at night
- Live migration enables dynamic consolidation
- But: Seldom used, fear of possible side effects
- $\Rightarrow$ Identify and quantify effects on (web) service levels

Scenario

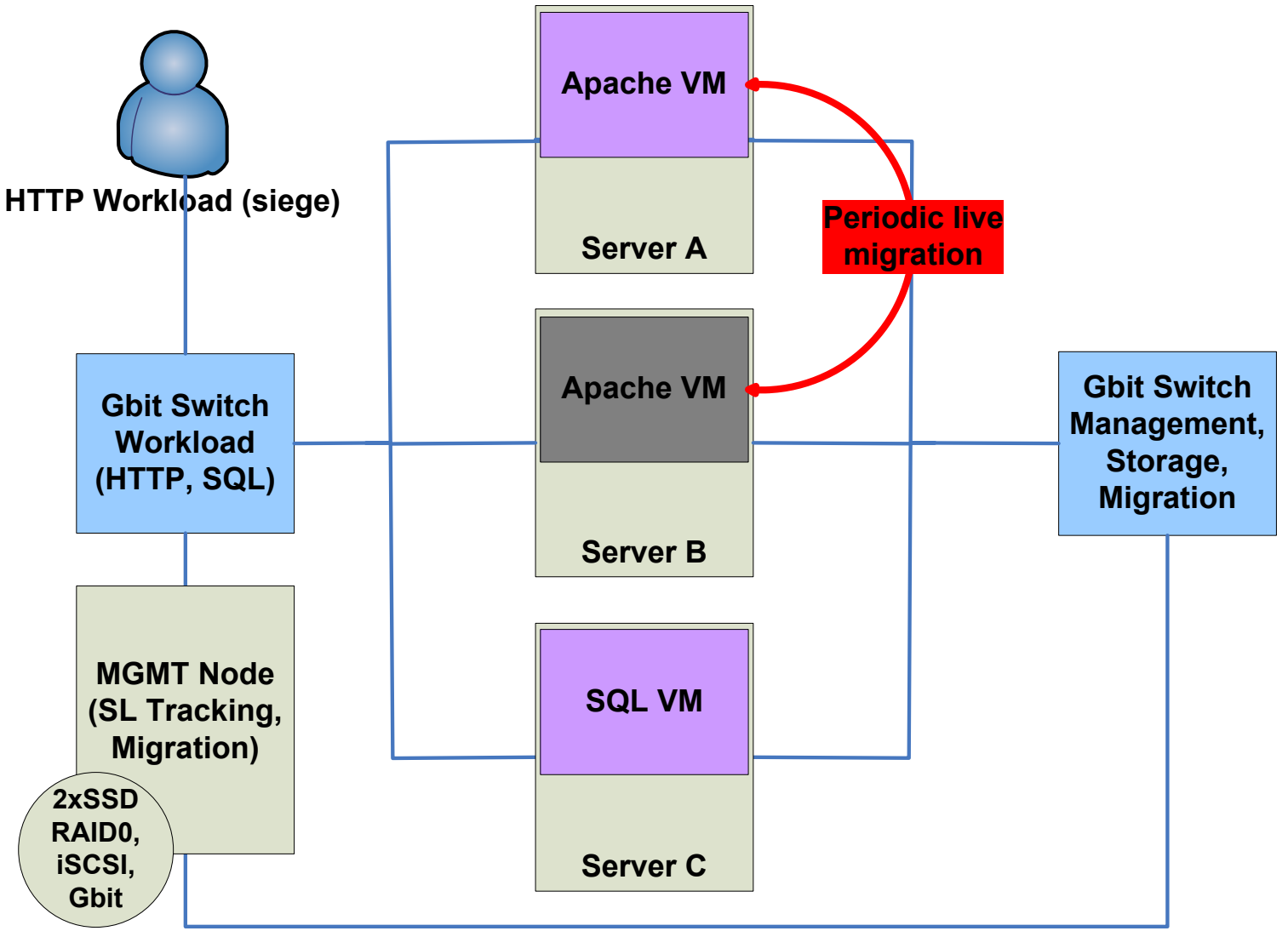

Meta-
Visor
Users
Workload
Management
Workload
VM 1
VM 2
RM 1 (libvirtd)
VM 3
VM 4
VM 5
RM 2 (libvirtd)
VM 6
VM 7
RM 3 (libvirtd)
...
VM m
RM n (libvirtd)
iSCSI/FC
SAN

Experiment

Experiment

- 2 servers, 1 VM, migrating forth and back
- VM disk image on central node (Gbit, open-iscsi)
- qemu-kvm VM: Linux, Apache2, PHP5, MediaWiki
- SQL VM and load generation on an extra nodes
- Logging utilization of servers and VM, $>$100 variables
- Rise load from 50 to 600 concurrent virtual users and back
- Migrate every 15min, track response time of last 5min (SLA=1s)

Experiment


HTTP Workload (siege)
Gbit Switch Workload (HTTP, SQL)
MGMT Node (SL Tracking, Migration)
2xSSD RAID0, iSCSI, Gbit
Apache VM
Server A
Apache VM
Server B
SQL VM
Server C
Periodic live migration
Gbit Switch Management, Storage, Migration

CPU utilization

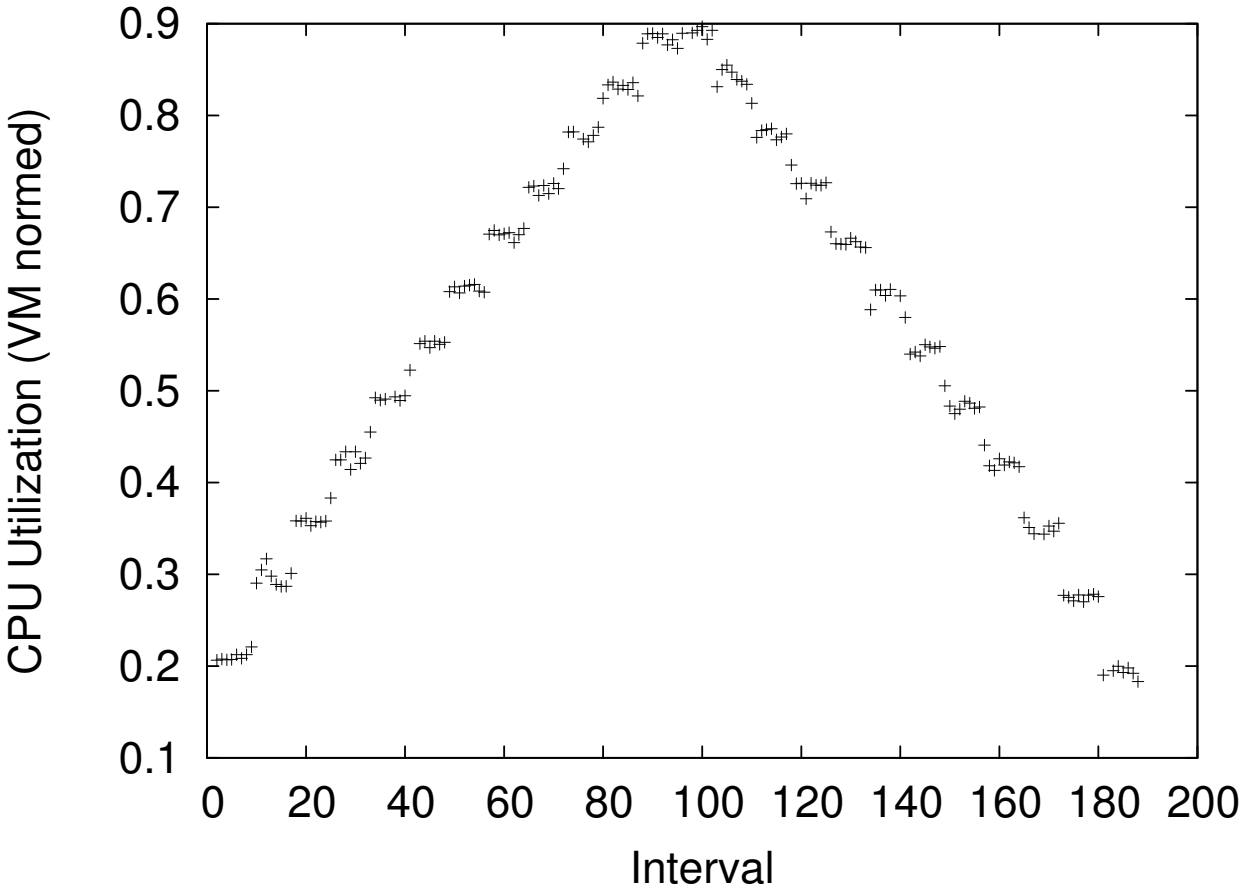


Abbildung: Effect of increasing/decreasing HTTP workload in equal steps on the VM's CPU utilization

Load average

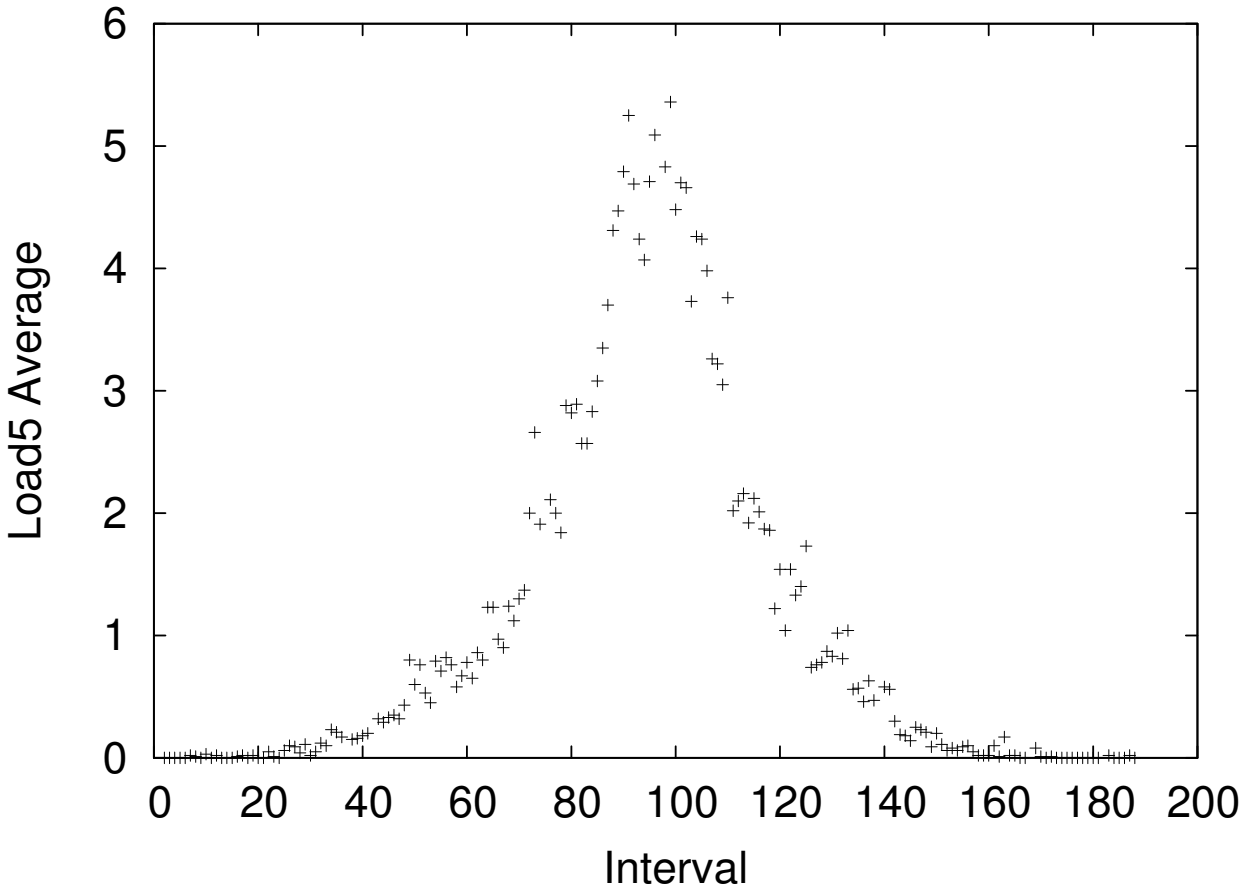


Abbildung: Effect of increasing/decreasing HTTP workload in equal steps on the VM's UNIX load average.

Markov Queues

- UNIX `load` as approximation to $\overline{Q}$, the average number of jobs in an M/M/1 queue, CPU utilization as ρ, the system utilization
- $\overline{Q}_{M/M/1} = \frac{\rho^2}{1-\rho}$
- UNIX `load` exponentially averaged by definition, service times are not necessarily exactly exponentially distributed, a systematic deviation from the theoretical solution is expectable.
- $\overline{Q}_{M/G/1} = \frac{\rho^2}{1-\rho} \cdot \frac{(1+c_B^2)}{2}$, c_B: coefficient of service time variation
- Exponentially distributed service times: $c_B^2 = 1$
- Deterministic service times: $c_B^2 = 0$
- Simple linear regression: $c_B^2 = 0.42$

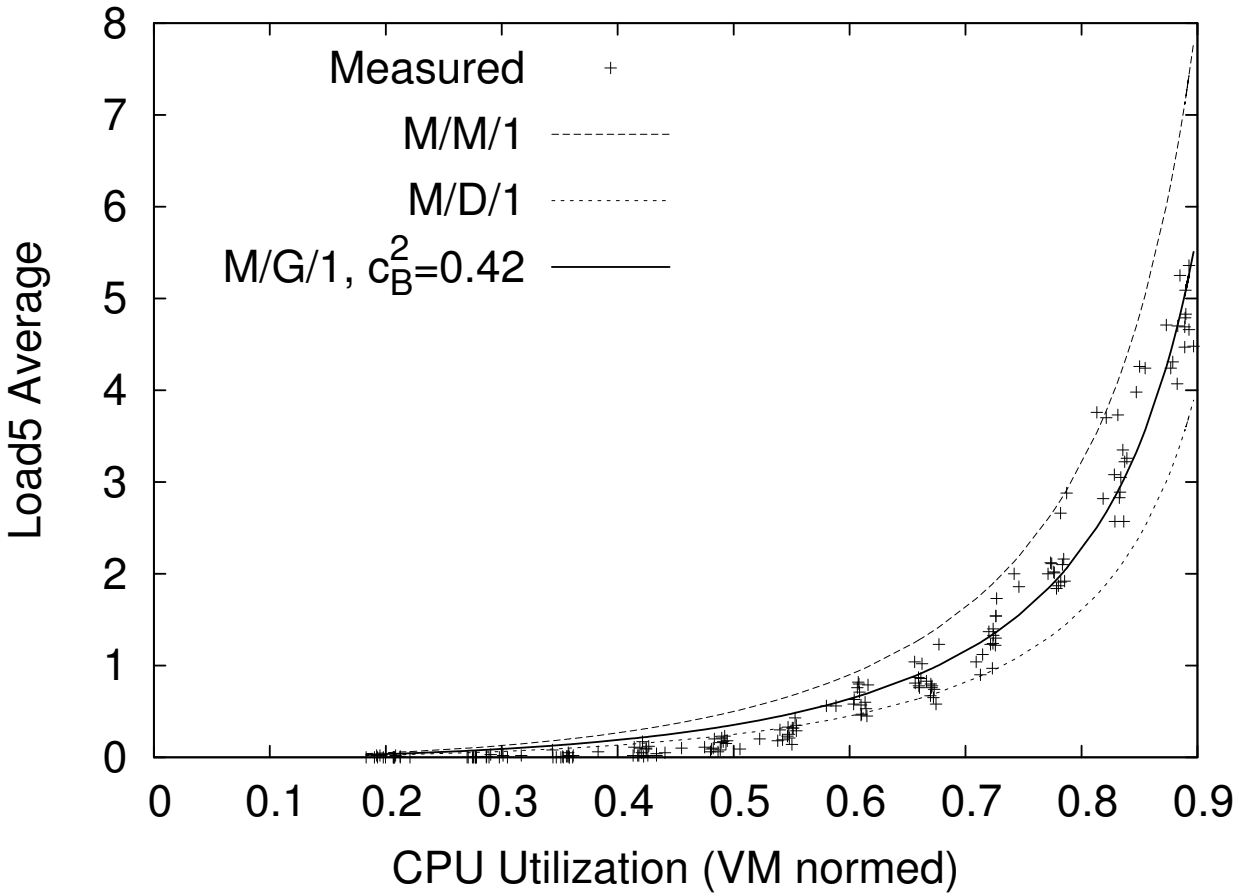


Abbildung: Queuing issues are rapidly increasing at 60-70% CPU utilization. The measured data follow the trend of the theoretical solution very closely.

Response Time

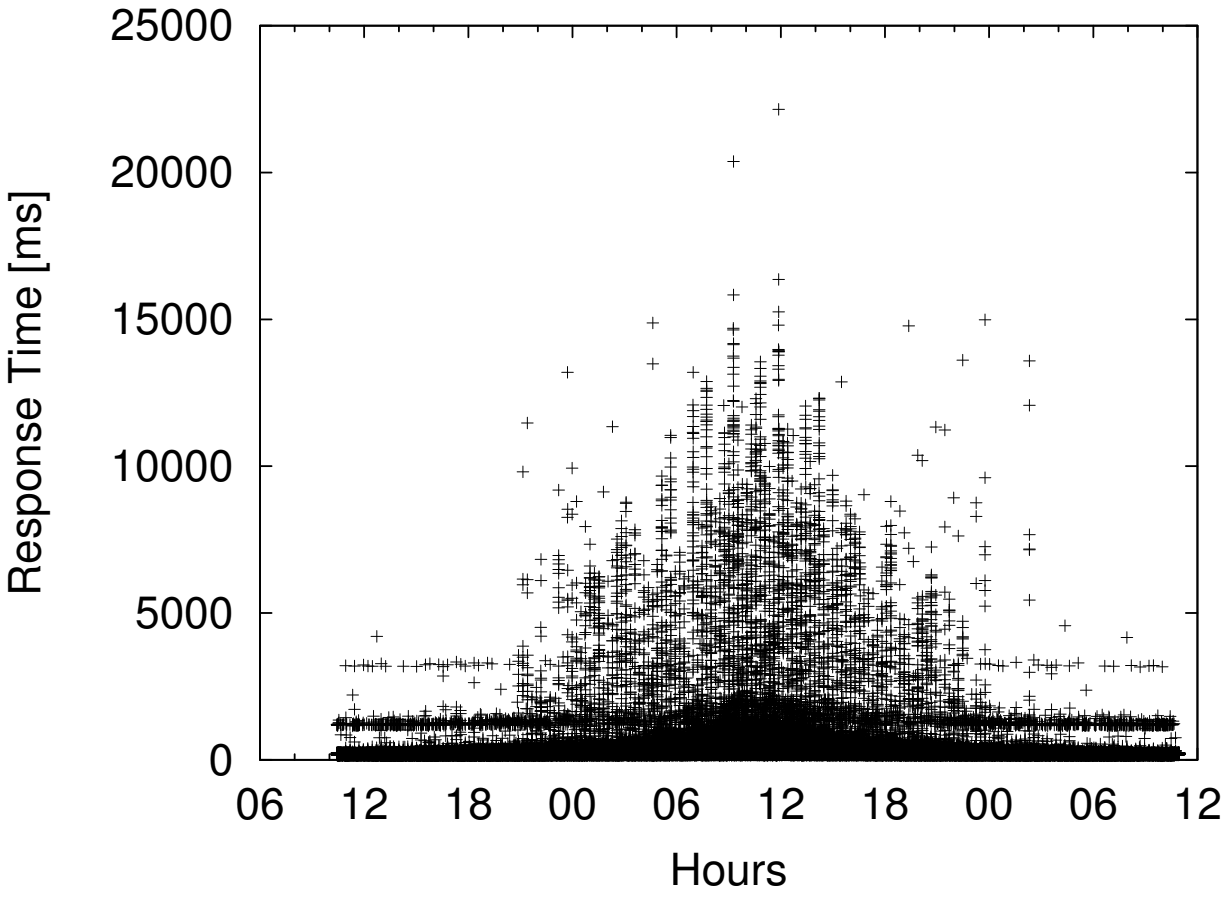


Abbildung: HTTP response times are massively increased during phases with higher workload.

Service Level

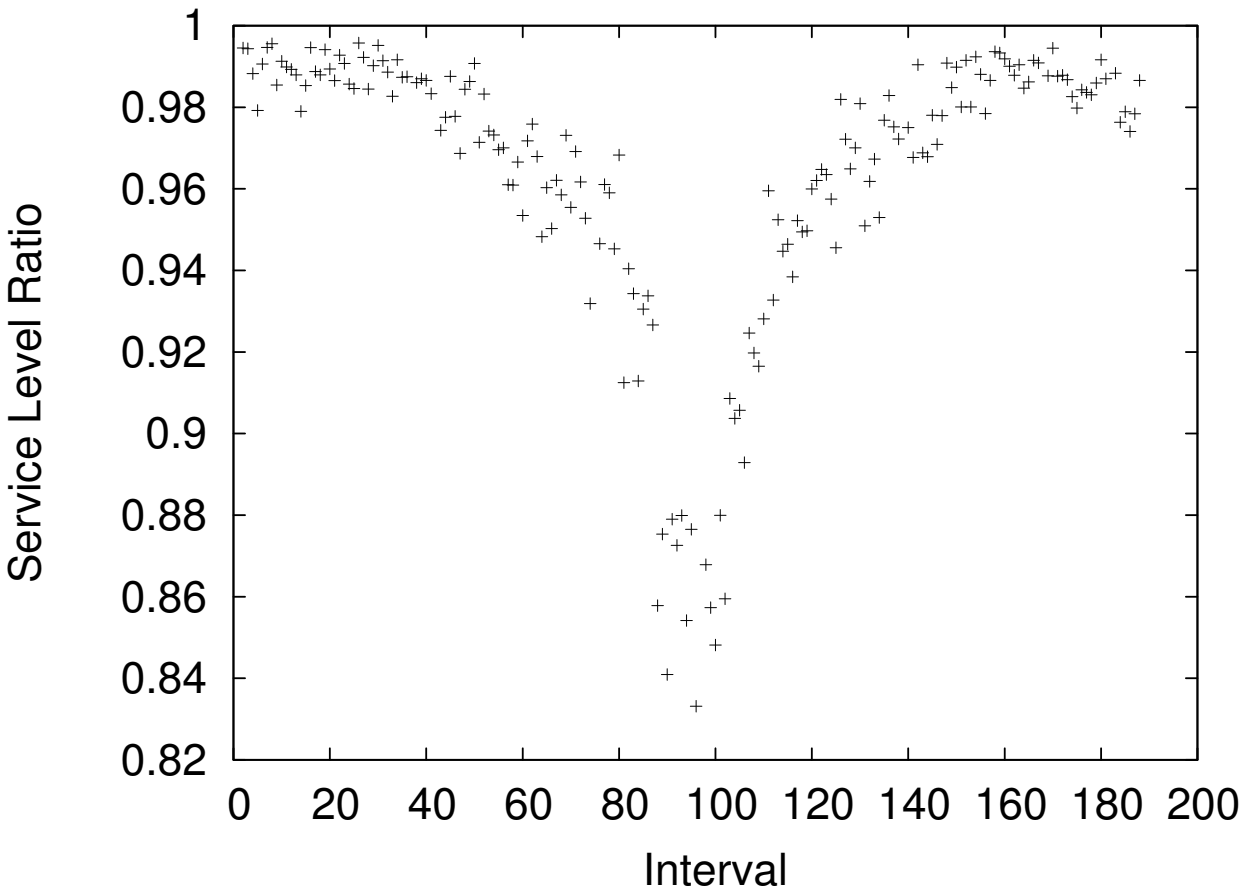


Abbildung: Service levels decrease significantly during phases of high workload. The maximum allowed response time is 1 s, which is extremely conservative.

Load average vs Service Level

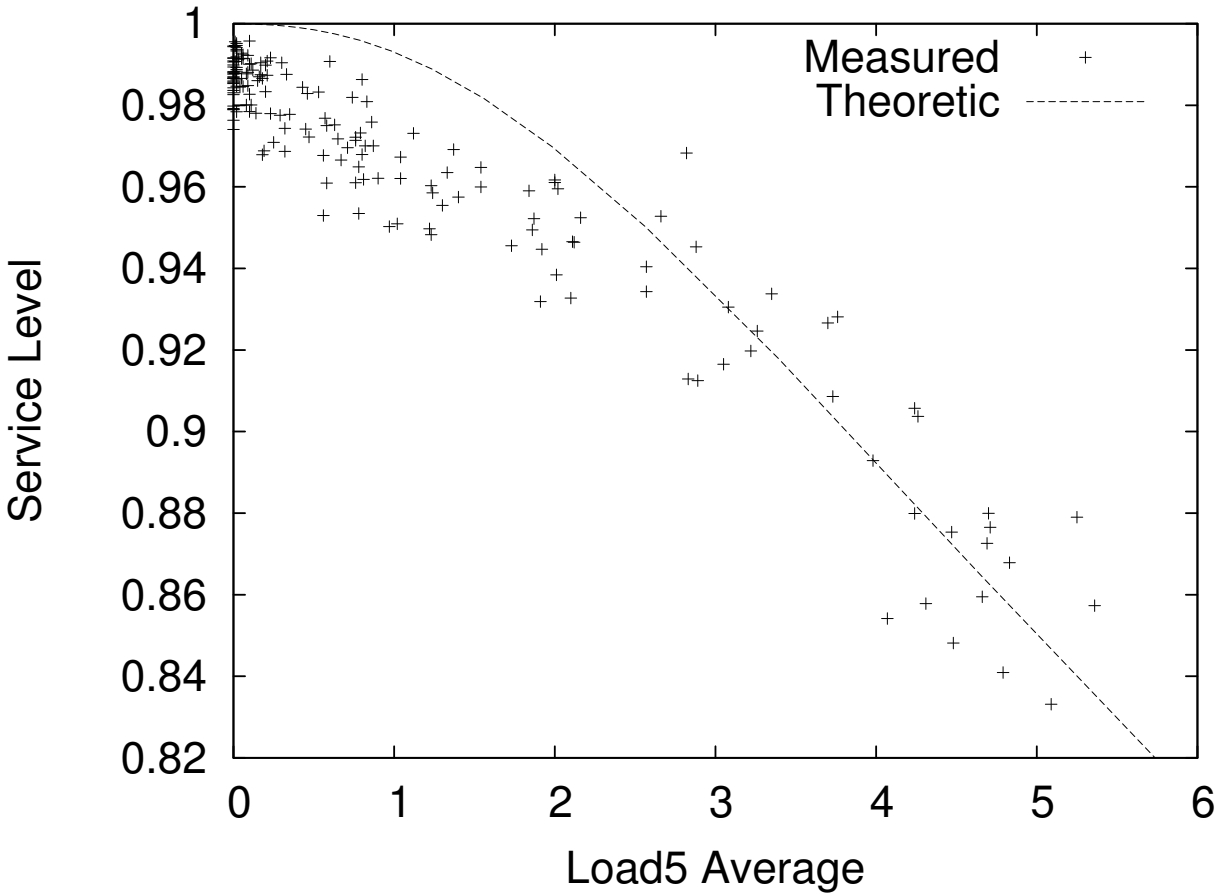


Abbildung: Higher load average is an indicator for queuing issues which are causing decreased service level ratios. The measured data follow the theoretical solution closely for higher `load` values.

Response Time: Low Load

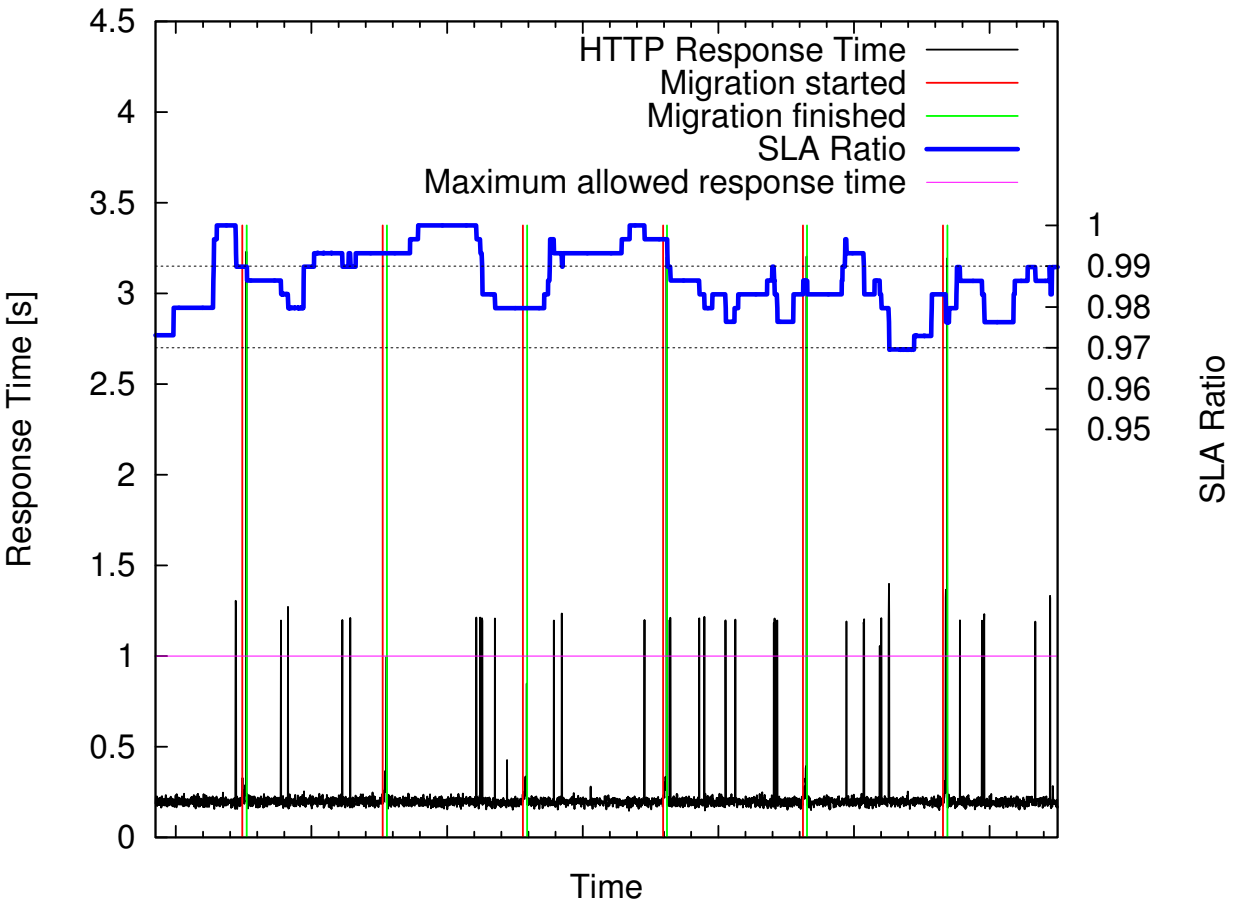


Abbildung: Influence of live migration on HTTP response time and service level during low load.

Response Time: Medium Load

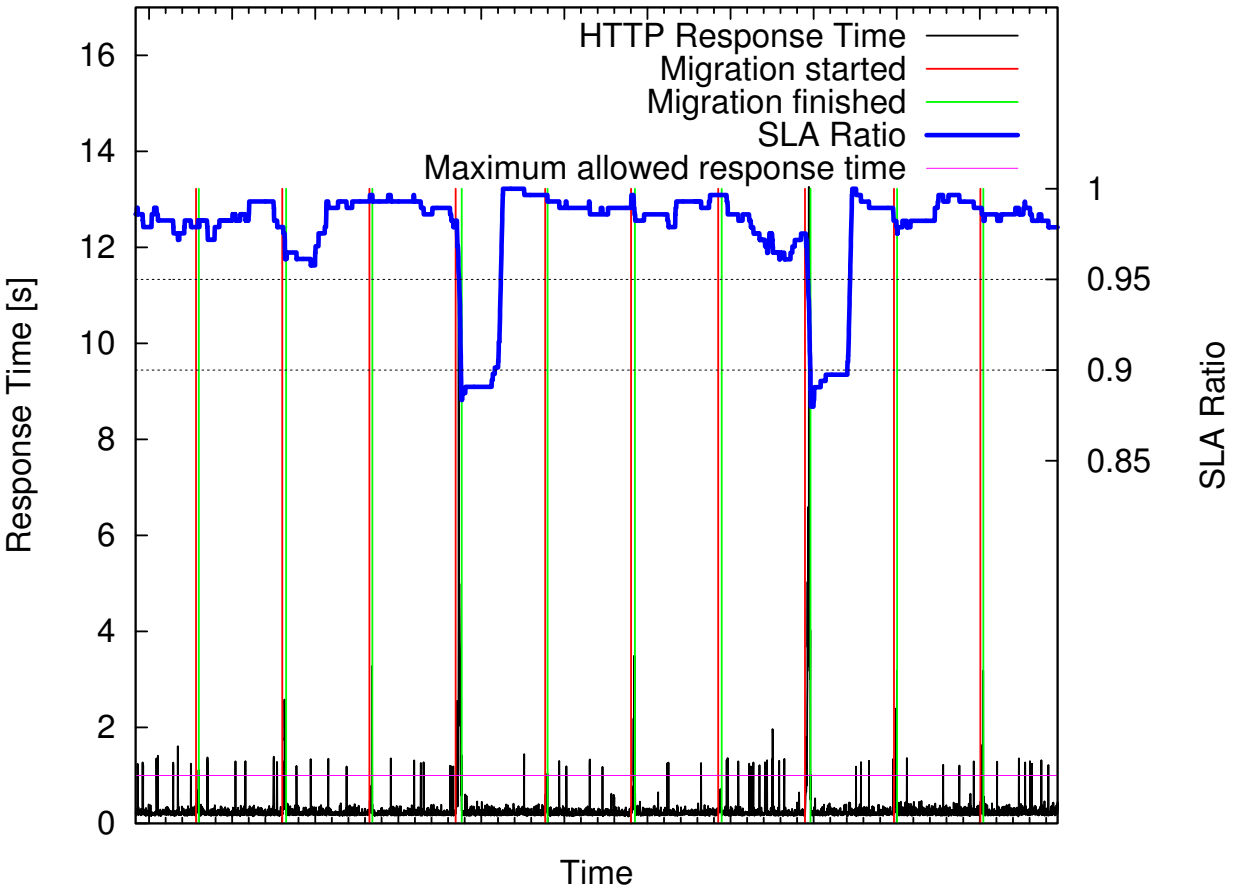


Abbildung: Influence of Live Migration on HTTP Response Time and service level during medium load.

Response Time: High Load

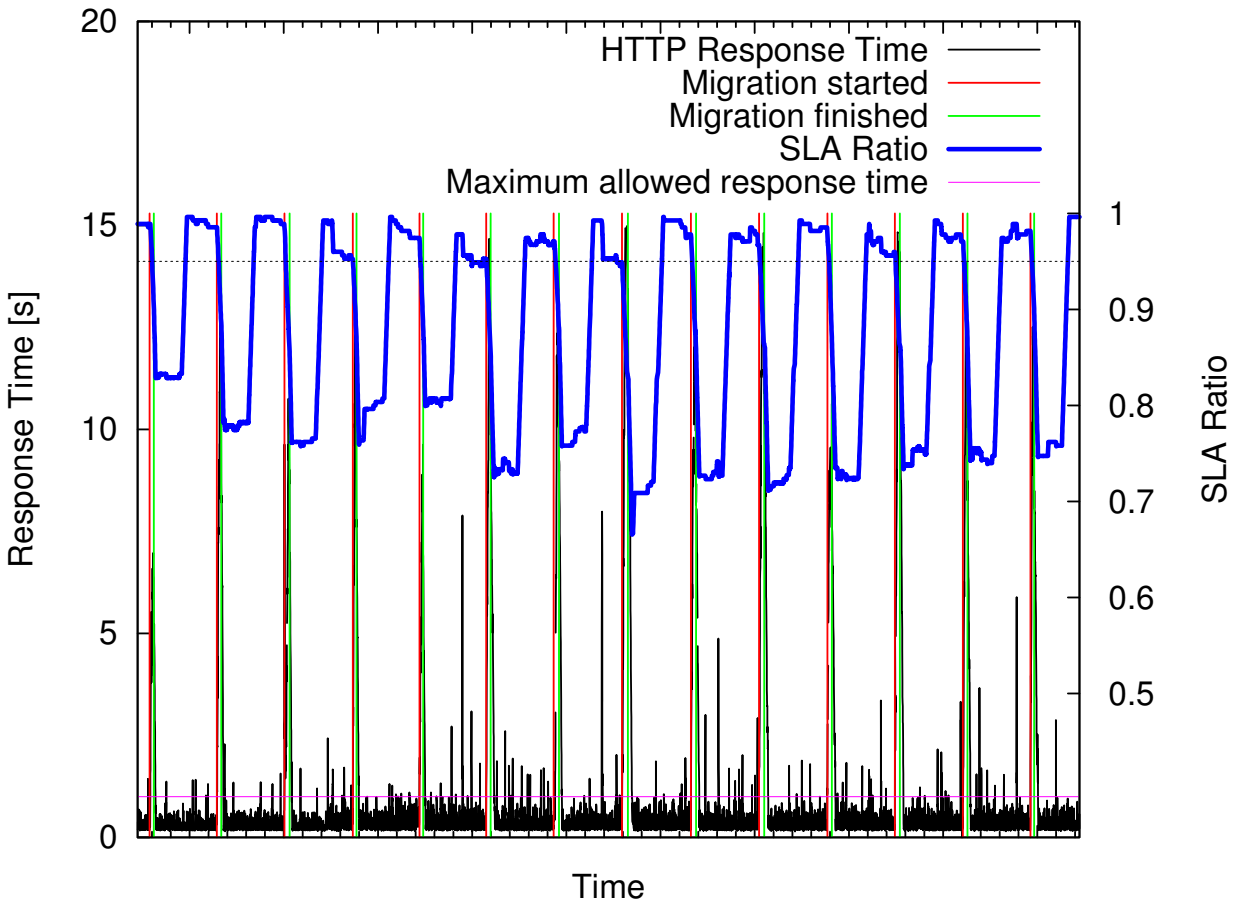


Abbildung: Influence of Live Migration on HTTP Response Time and service level during high load.

Model selection

Stepwise: AIC

- Akaike Information Criterion (AIC), lower values are better
- $AIC = 2k - 2\ln(L)$
- k: the number of independent parameters of a model
- L: the maximum likelihood
- Stepwise model selection approach
- Find trade-off between number of parameters (model size) and goodness of fit (model quality)
- Start with empty model, in each step a variable can be added/removed, until the AIC can not be decreased further.
- Does not guarantee to find a global optimum, but typically gives a near-optimum result within reasonable time.

Model selection

Exhaustive: LEAPS

- Comparison with the stepwise AIC approach
- Exhaustive all-subsets-regression
- Find best of all possible models for given range of model size
- Computationally intensive even if number of variables is limited

AIC

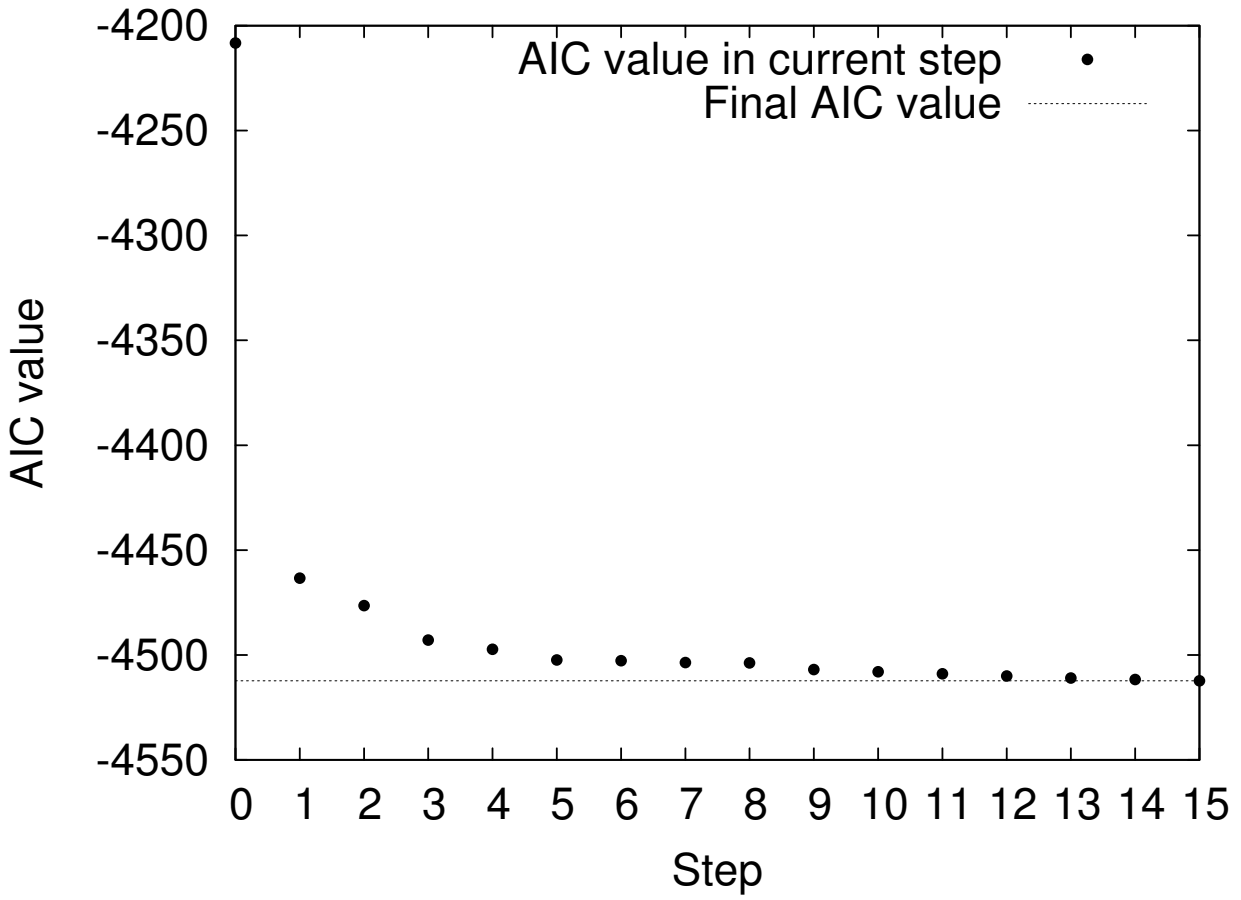


Abbildung: The AIC value quickly improves during the first steps and then slowly converges to the final value, thus allowing a trade-off between model complexity and precision.

AIC

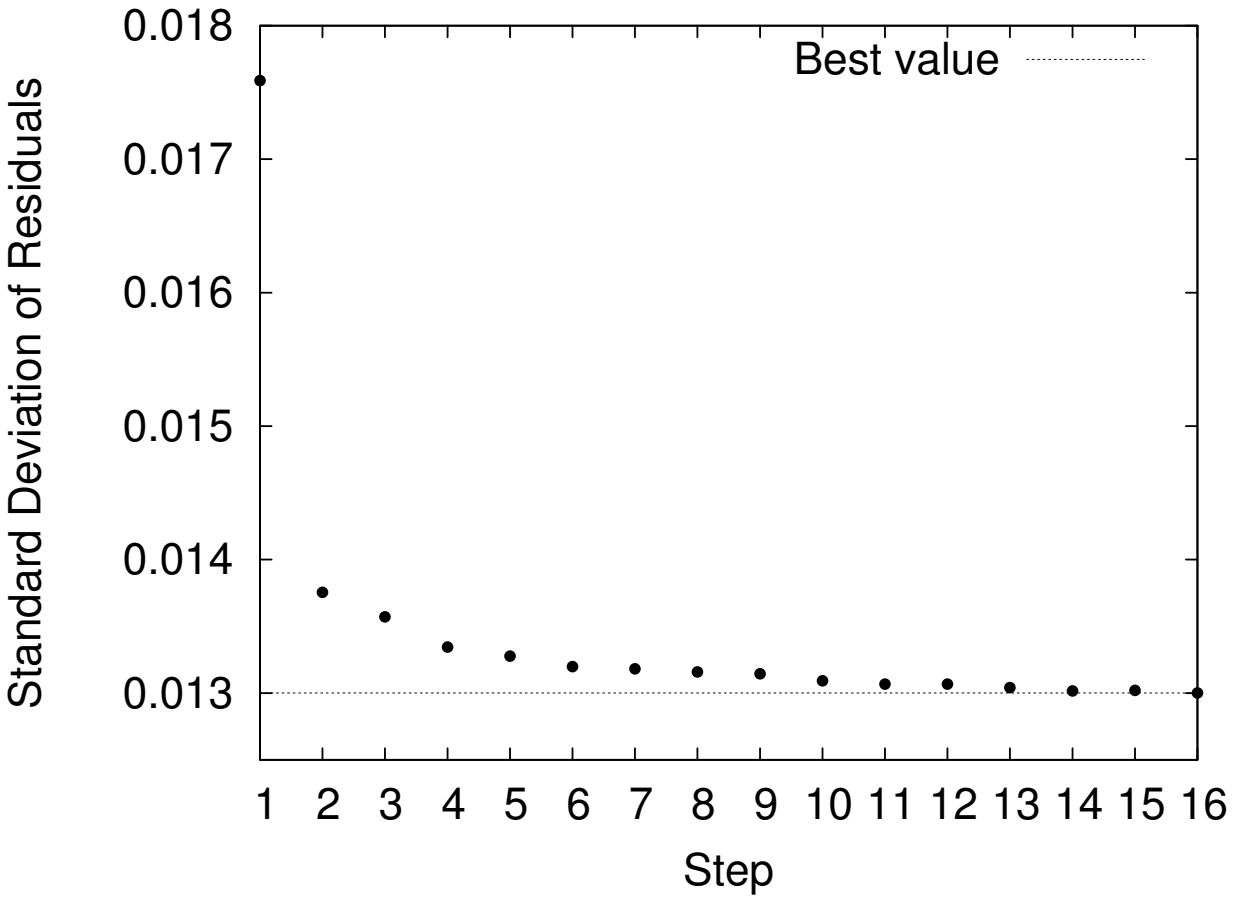


Abbildung: The residuals' standard deviation quickly improves during the first steps and then slowly converges to the final value, when using stepwise AIC.

AIC

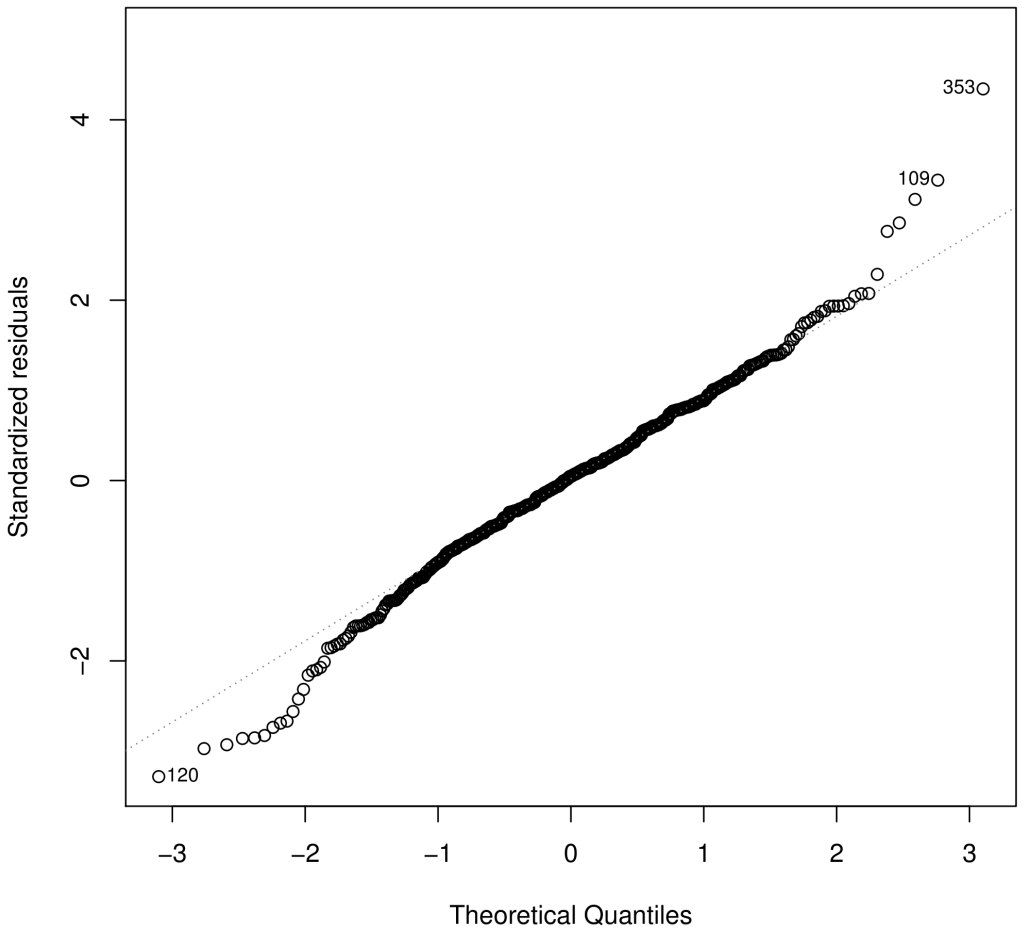


Abbildung: QQ-Plot of standardized residuals for the model produced by stepwise AIC.

AIC

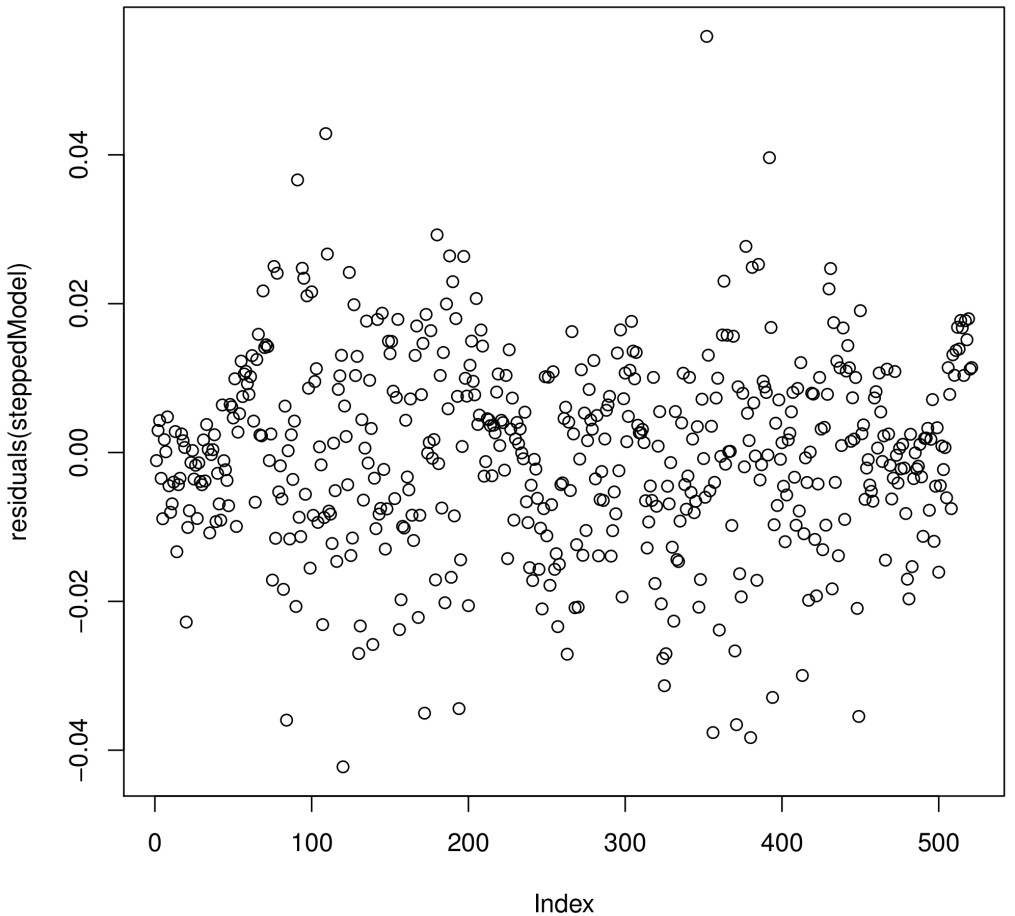


Abbildung: There is no visible correlation between the residuals' variance and the time series.

LEAPS

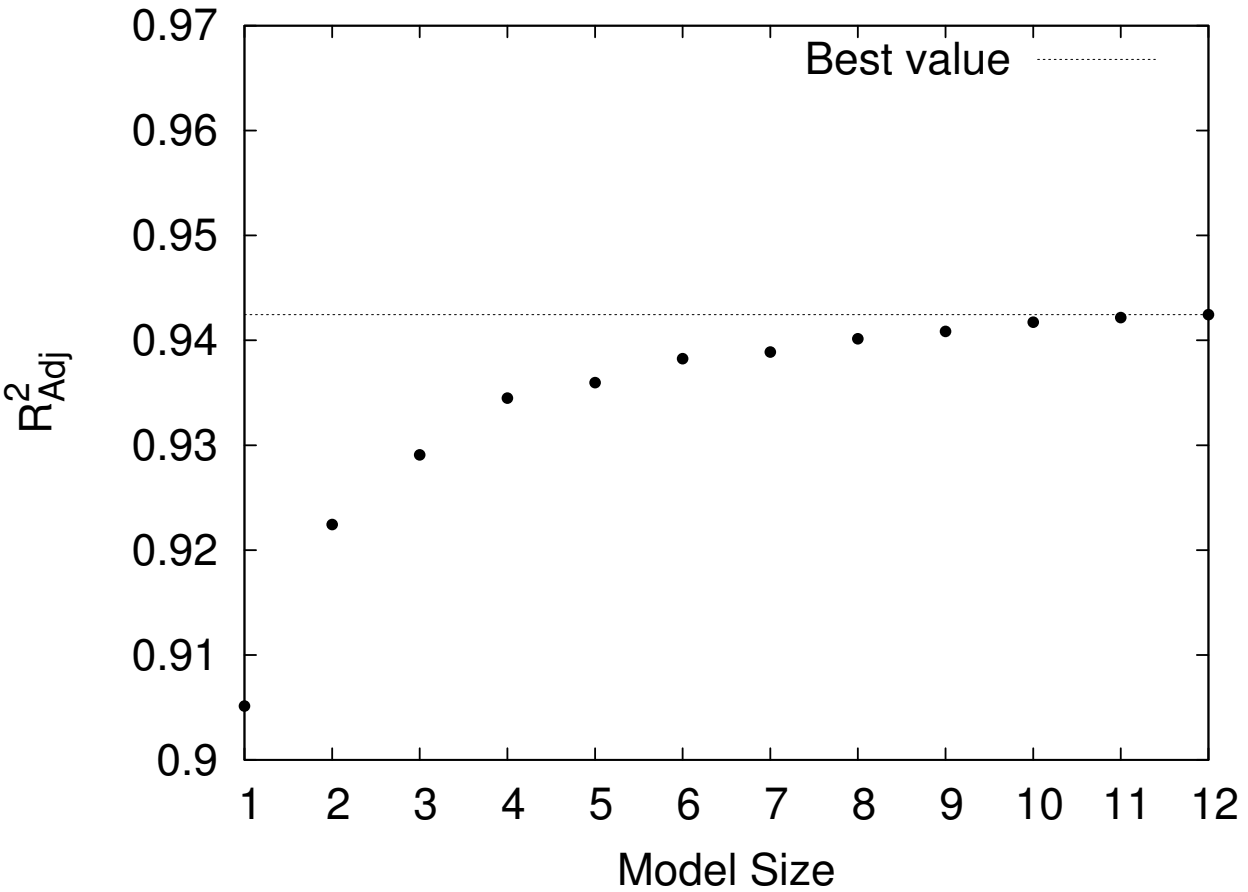


Abbildung: Higher model complexity yields better model quality when using LEAPS, but the increase is rather small.

LEAPS

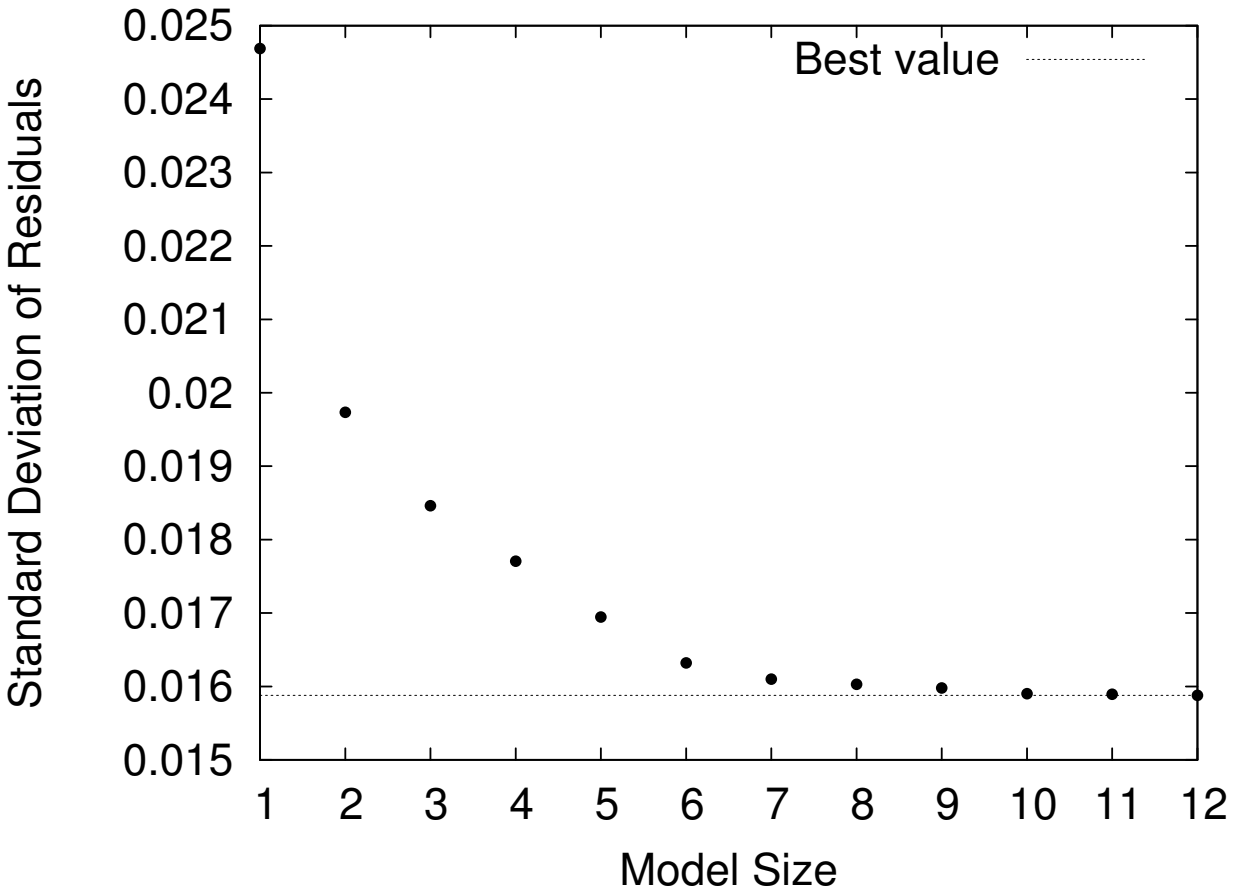


Abbildung: Higher model complexity yields better regression error when using LEAPS.

AIC: Most influencing variables of final model

Variable	Meaning	Estimate	Std. Error	$Pr(>\|t\|)$
Intercept		2.395e+00	5.069e-01	3.00e-06
wp01_load5	VM `UNIX load5`	-1.871e-02	2.627e-03	3.67e-12
wp01_swapUsed	VM swap used	-7.656e-07	8.809e-08	$<$ 2e-16
wp01_residentSize_SQ	The squared amount of resident memory used by the `qemu-kvm` process.	-4.652e-14	1.652e-14	0.00506
src_host_cpu_proc_s	Tasks created/s, source host.	-2.475e+00	9.166e-01	0.00716
src_host_cpu_proc_s_SQ		1.091e+00	4.133e-01	0.00856
wp01_cpu_util_vmnorm_SQ		-1.328e-01	3.187e-02	3.63e-05
wp01_cpu_util_vmnorm	CPU util measured inside VM	9.517e-02	2.316e-02	4.64e-05
wp01_load5_SQ	The squared `UNIX load5` of the VM.	-1.140e-03	1.918e-04	5.22e-09
wp01_freeMemRatio_SQ	The squared ratio of free memory inside the VM.	1.976e-02	9.462e-03	0.03727

Tabelle: The most influencing and significant variables of the final model produced by stepwise AIC.

Conclusions

Qualitative

1. Impact of live migration on SL depends on amount of workload
2. Tighter SLAs can be fulfilled for low workload situations
3. High workload: SL decreases massively
4. $\Rightarrow$ Live migrating when VM has only medium load
5. Avoid multiple live migrations within short time
6. $\Rightarrow$ Inertia effect for recently migrated VMs
7. Avoid senseless flip-flop migrations and stabilize service level

Conclusions

Quantitative

1. SL variance during a live migration to 90% predictable using only a single variable, the `UNIX load5 average`.
2. Models with 12 variables can explain 95% of variance
3. Models selected by AIC/LEAPS are of comparable quality and robust to statistical tests
4. AIC is way faster, so LEAPS is not really necessary

Conclusions

Technical

1. Systems using live migration as a mechanism to realize a more energy efficient target distribution need to consider the `UNIX load average`, at least for web servers and comparable
2. Typical hypervisors do not collect and export this information
3. Needs to be done by VM introspection
4. Related efforts by `qemu-kvm` and `libvirt` developers to pass-through the VMs' memory utilization (free mem)
5. Should be extended to export `load` information

Conclusions

Future Work

1. Influence of additional VMs
   - idle
   - utilized
   - and a mixed scenario
2. `Linux` and `qemu-kvm`: Kernel Samepage Merging (`KSM`)
3. Database VM migration
4. Bandwidth limits, maximum allowed downtime
5. Migration delay, energy consumption, service downtime

Q&A

Thank you for your attention!