Evaluation of Prioritization in Performance Models of DTP Systems

Christian Markl, Oliver Hühn
Department of Informatics, Boltzmannstr. 3
Technische Universität München
85748 Garching, Germany
{markl, oliver.huehn}@mytum.de

Abstract

Modern IT systems serve many different business processes on a shared infrastructure in parallel. The automatic request execution on the numerous interconnected components, hosted on heterogeneous hardware resources, is coordinated in distributed transaction processing (DTP) systems. While pre-defined quality-of-service metrics must be met, IT providers have to deal with a highly dynamic environment concerning workload structure and overall demand when provisioning their systems. Adaptive prioritization is a way to react to short-term demand variances. Performance models can be applied to predict the impacts of prioritization strategies on the overall performance of the system. In this paper we describe the workload characteristics and particularities of two real-world DTP systems and evaluate the effects of prioritization concerning supported overall load and resulting end-to-end performance measures.

1. Introduction

Today’s service industry makes use of distributed transaction processing (DTP) systems to run their day-to-day business. DTP systems coordinate the automated execution of business processes on shared IT infrastructures, often incorporating hundreds of basic service components hosted on numerous heterogeneous hardware resources. The automated request execution on such business processes without human interaction (called workflows) is controlled by a transaction processing (TP) monitor product.

Capacity planning and performance management issues of such applications are business-critical for IT service providers as the DTP systems have to satisfy quality-of-service criteria pre-defined in service level agreements (SLAs). Important performance measures include end-to-end workflow response times, throughput and availability. While proactive planning of such a complex meshwork of IT components is already difficult, the highly dynamic business environment of such systems makes it even more challenging. The request ratios on the single workflows (often referred to as workflow mix) as well as the overall demand coming in to the system in a certain time unit changes dynamically over the year, the week, and even over the day.

A telecommunication provider for example might deal with an increased number of new customers in the weeks before Xmas while such request ratios might be quite rare during the summer. While the number of user-initiated contract change or support requests might be relatively low during the night and increasing during the day, batch jobs on maintenance workflows might suffer from request ratios completely the other way round.

The huge number of independent workflows run on the same shared infrastructure together with this highly-dynamic demand environment makes the forecast of the workload mix and overall demand that the system has to serve extremely challenging. IT service providers deal with this problem by extremely over-provisioning their system capacities, aligning them even to very short demand peaks. To avoid SLA violations, huge capacity buffers are provisioned that are then unused most of the time.

Adaptive workflow prioritization is a way to deal with short-term demand variances. By giving workflows a higher priority that are at risk to violate an SLA due to a short-term demand peak, costly capacity buffers can be reduced as performance measures of these workflows will be improved. However, of course, the effect of prioritization comes along with harms to the performance of the other workflows run on the same infrastructure. It is therefore important for the providers to gain knowledge about the impacts of possible prioritization scenarios to the performance measures of the overall system proactively. Therefore
we evaluate the side effects of such prioritization scenarios on two real-world DTP systems of a telecommunication provider. These impacts are essential for the pro-active capacity planning of DTP systems.

In the remaining of this paper we first describe the particularities of two real-world DTP systems of a telecom provider. In chapter three, we then present the workload specifics of these systems in detail. After a general overview on performance modelling in this area in chapter four, we evaluate the effects of prioritization strategies on the overall system performance in an experiment in chapter five. Related literature is discussed in chapter six before we conclude in chapter seven.

2. DTP Systems

Modern enterprise IT infrastructures consist of a wide variety of applications and systems. To link and integrate these systems means moving data to the right place and transforming it correctly across different applications and databases. The concept of distributed transaction processing (DTP) supports the flexible, easily adaptable composition of distributed software services in such heterogeneous environments to so-called workflows, the IT representation of business processes.

Usually, several workflows are hosted on a shared IT environment, using the basic IT service components collectively. For example the validation of a credit card account might be such a basic service component that can be incorporated in several workflows of a shop system. Basic IT components might call others to fulfill their functionality, e.g. queries to a CRM or billing system. These interwoven call structures lead to a complex meshwork of interconnected IT services typical for a productive DTP system.

In practice, most automated business processes are executed on transaction processing (TP) monitors such as Oracle Tuxedo, IBM CICS, the Vitria suite, or TIBCO Business Works. The main purpose of the TP monitor is to ensure the ACID properties (atomicity, consistency, isolation, and durability) of the single transaction steps of a workflow while managing the automatic request execution across the shared IT infrastructure. Coordinating this process across numerous services, running on many different software components, hosted on heterogeneous hardware resources, and connected by possibly unreliable communication links, is an essential problem in distributed transaction management. TP monitors face this challenge by applying rollback mechanisms and the two-phase commit protocol.

DTP systems need to be designed for hundreds of thousands of requests a day, serving often hundreds of them in parallel. They need to meet high quality of service standards in terms of response times, throughput, and availability, defined in service-level agreements (SLAs).

The complex meshwork of interconnected IT services together with the restrictive quality-of-service requirements makes capacity planning of DTP systems a challenge. Provisioning decisions become even harder when considering the highly dynamic workload environment of typical DTP systems.

3. Workload

The workload of a typical DTP system consists of order-entry jobs as well as batch jobs. Requests of both sources are normally handled by the same IT infrastructure. As the order-entry jobs are initiated by human interaction e.g. on a web portal, their amount and arrival time can only be estimated but not be directly controlled by the IT service provider. In contrast most batch jobs, like database cleansing, are initiated by the service provider and therefore can be fed into the system anti-cyclical to the order-entry jobs, e.g. at night, for load-balancing issues.

Requests on the different workflows of a DTP system are typically independent from each other. We call the number of requests for the different workflows in a certain time period the workflow mix of the shared IT system.

The overall demand but also the workflow mix changes dynamically over the year, the week and even over the day. For our industry partner, for example, the weeks before Xmas are the top-selling weeks of the year. Thus, during this time the overall workload on the systems is significantly higher than throughout the rest of the year.

Figure 1 shows the daily workflow mix of a productive DTP system of a telecom provider over three weeks.

![Figure 1: Weekly workload mix of the productive DTP system of a telecom provider](image)

The lines represent the overall number of jobs for single workflows during the day between 11 am and 6 pm. As one can see, the workload has a recurrent
characteristic over the week. On weekends, especially on Sundays, the load on the system is relatively low. In average, the overall workload on weekdays is five times higher than that on weekends.

Figure 2 shows the number of requests per 60 seconds on a single workflow of the same DTP system of our industry partner over the day. The day-time workloads are generated by shop employees, customers using the web portal, and call center agents. During the night, batch jobs are executed on the system initiated e.g. by billing or CRM systems.

![Figure 2: Daily demand on a sample workflow of the DTP system of a telecom provider](image)

As Figure 2 shows, the daytime workload exhibits a certain usage pattern: increasing workload in the morning, a plateau phase throughout the day, followed by a decreasing workload in the evening hours.

Also the plotted workload of the workflow is representative for the other workflows of the system as well, each single workflow has varying peak demand times. This causes a dynamically changing workflow mix of the overall DTP system in addition to the varying load of the single workflows.

For capacity planning decisions it is important to know the expected overall load and workload mix of the DTP system in the planning period. Time series analysis (TSA) refers to those specific econometrical methods and techniques that are used for analyzing sequences of measurements over periods of time, with the purposes of understanding the structure of data and identifying patterns. TSA methods like regression methods [1] or triple exponential smoothing [2] can be applied to forecast long-term workload trends by analyzing historic data. However, the short-term variances of a real-world DTP system can hardly be included in such models.

The typical over-provisioning of DTP systems aligned even to short historic demand peaks can be improved by the application of adaptive workload prioritization. As peak demands on single workflows are typically short, adaptive prioritization is a possibility to react to short-term demand variances.

For an IT service provider it is important to gain knowledge about the performance impacts of prioritization strategies on the overall system. Various workload scenarios have to be analyzed making tests on a real system very time-consuming and expensive. Furthermore, we learned from our industry partner that despite enormous effort it is nearly impossible to reproduce all aspects of a productive DTP system on a test system infrastructure.

A possibility to evaluate the performance impacts of various prioritization scenarios on DTP systems is performance modelling.

### 4. Performance Modelling

Modelling real-world systems has two main goals: gaining insight into the actual system and predicting future system behavior [3]. Performance models support decision makers in their planning and optimization tasks e.g. by identifying possible bottleneck stations or predicting the resulting workflow response times of a certain load scenario.

We apply Queueing Theory (QT) to model productive DTP systems. QT is a well-studied methodology for the analysis of systems with service stations and waiting lines. Its applications range from manufacturing system planning over computer processor design to multi-tiered web services [4, 5].

The complex interdependencies of service stations in modern DTP systems results in end-to-end workflow response times that develop in a non-linear way up from a certain amount of load. The strength of QT is, that once a valid performance model is built, this non-linear response time behavior can be predicted. Thus, the impacts of many different load scenarios to the overall performance of the systems can be evaluated easily.

The main input parameters of Queueing Network Models (QNMs) in addition to the system configuration are the workflow arrival rates and the time at each service component that it takes to serve the requests there, known as service times. Stations might be able to serve more than one request in parallel. This is modelled as the number of parallel workers of this station.

Three types of queueing networks are distinguished: When the jobs enter the system from outside the system and leave it after being served, we speak of an Open Queueing Network Model. In Closed Queueing Networks, the jobs circulate inside the system without leaving it. If jobs of both characteristics are combined in a model, it is called a Mixed Queueing Network Model.

Productive DTP systems typically serve order-entry jobs as well as batch jobs on a shared IT infrastructure. Thus, such systems can be modelled as Mixed QNMs.
The characteristics of the model parameters, including its type, the assumed arrival rate and service time distributions, and the number of parallel servers, influence the applicability of analytic solution algorithms.

We evaluate prioritization strategies in QNMs. Only few analytic solution algorithms are able to consider workflow prioritization in their calculations. Their number gets even smaller when trying to study systems with prioritization and more than one parallel workers. To our knowledge no implementation of such an algorithm is available as open source project.

Another possibility to solve QNMs is by the means of simulation. In Discrete-Event Simulations (DES), the state of each station in the network changes to discrete points in time, e.g. when jobs are served and leave the station. As prioritization as well as stations with more than one parallel workers can be considered in a DES, we make use of a custom developed discrete-event simulation engine in our experiments. The simulation engine is part of our Performance Modelling Tool suite PerMoTo [6], specially designed for the evaluation of DTP systems.

In contrast to analytic algorithms, the simulation engine can calculate performance measures for models that include state-dependent real-world objectives like adaptive prioritization strategies or capacity adjustments. Prioritization strategies, for example, can be simulated by dispatching the jobs from the queues of service stations according to their prioritization level.

The usage of prioritization in a DTP system requires that priority levels are introduced and all workflows of the system are matched into one of those priority classes. Possible hints for this classification might be the business-criticality of the workflow, the visibility of performance measures of this single workflow to the customer or the risk, that an related SLA might be violated.

In general one distinguishes between preemptive and non-preemptive request scheduling. While in a non-preemptive strategy the execution of a single request is finished once it has started, the execution might be aborted in preemptive strategies in order to continue with the execution of an incoming request of a higher priority level. In the DTP systems of our industry partner only non-preemptive request scheduling is used.

A general problem with prioritization in DTP systems is the possible starvation of jobs with a lower priority level. If the arrival rate of higher prioritized jobs is high enough, lower prioritized jobs remain unserved in the service queue as always the higher prioritized ones are preferred by the scheduling strategy. Several advanced priority-based queueing approaches are discussed in chapter six that try to solve this problem.

Once a performance model of a DTP system is generated, several load scenarios can be evaluated. In the next chapter we present the results of an experiment concerning the evaluation of prioritization in performance models of two real-world DTP systems of our industry partner.

5. Experiment

In our experiment we evaluate the effect of the prioritization of one specific workflow on the whole system and if guaranteed SLAs for the whole system still hold. Therefore, we look at two productive DTP systems of a European telecom provider.

The first system under study, called System A, is the central IT backbone for workflows related to the management of the retail customer segment. The technical implementation is done based on the Transaction Monitor “Bea Tuxedo”™. Requests on the system are initiated in the point of sales system of our industry partner: internet portals, shop-based applications, and call centers.

System A serves 18 business critical workflows. The variable tasks of the workflows are achieved through the usage of 39 different item types, calling 51 services. The length of the workflows varies: a single one is very short as it consists of only one item step; the others are more complex and contain up to 17 items. A maximum of 3 different service types are called inside a single workflow, and up to seven service types are called within a single item. The length of the items vary as well - one contains only a single service step, the others call several services, up to a maximum of 12. Single item types are typically called by more than one workflow (up to three); single types are included in up to 25 different items and a maximum of 14 different workflows.

System B is the integration backbone for workflows related to the management of products hosted by our industry partner but originally sold as prepaid telecommunication or DSL packages by external third-party companies. The supported functionality includes tariff management, customer subscription and deactivation, SIM and phone number management, billing, and age verification. System B serves two main classes of workflows: order-entry workflows initiated by customers over a Voice Portal, and workflows used by internal IT systems of the partner companies like billing or tariff administration. Technically, B is based on a customized version of "Tibco Business Works"™.
Unlike in System A, the audit logs of the second DTP system under study contain only information on workflow and item level. Thus, we modelled the queueing network of B on item level granularity. B serves 15 workflows containing 35 different item types. Several workflows of System B call just one item type while others are more complex. A maximum of 19 items are called within a single workflow, belonging to a maximum of 17 different types. Most items are called in various workflows, one item type actually by up to seven different ones.

In order to estimate model parameters, we have analyzed log data of 30 weekdays in June and July 2008. Timestamps and request IDs in the log files allow for estimating the input parameters that are relevant for queueing network models and simulation, most notably arrival rates on a workflow level and service times on a service level, as well as end-to-end response times. Note that we have put considerable effort in data cleansing and outlier detection, which were due to technical problems of errors in the log data.

We model the two DTP systems as open multi-class queueing networks. Both are very large with 140 and 126 respectively stations. The main characteristics like number of classes and response times in different levels are described in Table 1.

### Table 1: Key characteristics of the Queueing Network Model of Systems A and B

<table>
<thead>
<tr>
<th>Queueing Network Model Characteristics</th>
<th>System A</th>
<th>System B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of classes (i.e., workflows)</td>
<td>18 / 18</td>
<td>15 / 15</td>
</tr>
<tr>
<td>Number of stations (i.e., services)</td>
<td>140</td>
<td>126</td>
</tr>
<tr>
<td>Number of parallel workers in station</td>
<td>1 - 8</td>
<td>1 - 6</td>
</tr>
<tr>
<td>Mean response time on a service level [sec]</td>
<td>0.17</td>
<td>-</td>
</tr>
<tr>
<td>Min - Max response time on a service level [sec]</td>
<td>0.01 - 4.41</td>
<td>-</td>
</tr>
<tr>
<td>Mean response time on item level [sec]</td>
<td>0.68</td>
<td>0.23</td>
</tr>
<tr>
<td>Min - Max response time on item level [sec]</td>
<td>0.01 - 5.46</td>
<td>0.051 - 2.53</td>
</tr>
<tr>
<td>Mean response time on workflow level [sec]</td>
<td>37.81</td>
<td>5.13</td>
</tr>
<tr>
<td>Min - Max response time on workflow level [sec]</td>
<td>4.41 - 248.26</td>
<td>0.01 - 36.06</td>
</tr>
</tbody>
</table>

First we validated our models. We calculated the response times of the workflows and compared them with the real-world response times from the log files. The predictive accuracy of our open models were at about 3% with one outlier at 11% for system A. The accuracy of system B was about 15%. Menasce et. al [4] state deviations of 10-20% for the predictive accuracy of response times as acceptable. Therefore, we used our models for further investigations.

In a first instance the base scenario is analyzed with respect to possible bottlenecks and the systems behavior when increasing the workload. This scenario is named increased load scenario in the further course of the paper. We increase the overall load on the system up to a certain load factor. Thereby the relative workload mix remains constant but the absolute number of requests is raised for all workflows in a linear way. This helps identifying the amount of workload the system is able to cope with and at which point in time problems might occur.

In the increased load scenario the arrival rate of each workflow is increased linear to 600% of the original value. Figure 3 presents the relative deviation of the mean response times for eight important workflows of system A.

![Figure 3: Response time deviation of an increased load of 600% on the system](image)

Three of the considered workflows exhibit a significantly rising response time development with deviations of plus 36.54%, 85.53%, and 118.21% respectively. The reason is that these workflows share some items.

![Figure 4: Utilization development of the items included in of workflow WF A2](image)

Figure 4 shows the utilization growth of all items of WF A2. The response time of item 4 increases...
dramatically. At 300% increased load its utilization is already at 100%. Item 4 and the second bottleneck station item 31 are included in the two workflows WF A1 and WF A5. These two items lead to the enormous growth of the response times for the three workflows shown in Figure 3.

Workflow WF A is of great importance for our research partner as it coordinates the activation of new customers. Therefore we use prioritization in our second scenario, the prioritization scenario, to improve the response time deviation of workflow WF A2 at increased load on the system.

We use two prioritization values in this scenario. Workflow WF A2 gets the value 2 as high-prioritized workflow; the others get the value 1. This prioritization improves the response time of workflow WF A2 at higher load as shown in Figure 5.

Figure 5: Response time development with and without prioritization of workflow WF A2

![Figure 5](image)

The workflow response time is at nearly the same level until the increased load of 650%. Then it increases in a non-linear way. Without prioritization the increase already begins at the load of 600%.

Table 2: Workflow response times (RT) of the base scenario, the increased load scenario and the prioritization scenario on System A

<table>
<thead>
<tr>
<th>Workflow</th>
<th>Base Scenario</th>
<th>Increased Load Scenario</th>
<th>Prioritization Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT [sec]</td>
<td>RT [sec]</td>
<td>RT [sec]</td>
</tr>
<tr>
<td>WF A1</td>
<td>226.69584</td>
<td>309.53597</td>
<td>1746.68921</td>
</tr>
<tr>
<td>WF A2</td>
<td>238.59284</td>
<td>442.65372</td>
<td>244.96256</td>
</tr>
<tr>
<td>WF A3</td>
<td>5.37035</td>
<td>5.34819</td>
<td>5.38824</td>
</tr>
<tr>
<td>WF A4</td>
<td>4.94283</td>
<td>4.94484</td>
<td>4.94589</td>
</tr>
<tr>
<td>WF A5</td>
<td>178.38016</td>
<td>389.23448</td>
<td>3811.83304</td>
</tr>
<tr>
<td>WF A6</td>
<td>10.14721</td>
<td>10.14642</td>
<td>10.14714</td>
</tr>
<tr>
<td>WF A7</td>
<td>9.32691</td>
<td>9.33009</td>
<td>9.32798</td>
</tr>
<tr>
<td>WF A8</td>
<td>14.88074</td>
<td>14.87481</td>
<td>14.88039</td>
</tr>
</tbody>
</table>

But the prioritization of workflow WF A2 comes along with harms. The response times of the other two workflows containing the bottleneck station increase dramatically. Table 2 gives an overview of the response times for the eight workflows under study.

Figure 6 shows the response time deviation of the load increased scenario and the prioritization scenario compared to the base scenario for system A. The lower number indicates the response time deviation of the load increased scenario, the upper number the value for the prioritization scenario. Workflow WF A2 has nearly the same response time for the prioritization scenario as in the base scenario with a little increase of 2.67%. In the load increased scenario the deviation was 85.5%. The impacts of the prioritization are visible on the other two workflows using the bottleneck item 4. WF A1 increases from 36.5% to 670.5% deviation, WF A5 from 118.2% to 1924.8%. The other workflows are not concerned by the prioritization of workflow WF A2.

Figure 6: Response time deviation of the prioritization scenario when compared to the increased load scenario on System A

A similar approach is used in the second system B. In this system we prioritize workflow WF B4 that coordinates the activation of new customers. Table 3 summarizes the response times for the different scenarios.

Table 3: Workflow response times of the base scenario, the increased load scenario and the prioritization scenario on System B

<table>
<thead>
<tr>
<th>Workflow</th>
<th>Base Scenario</th>
<th>Increased Load Scenario</th>
<th>Prioritization Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RT [sec]</td>
<td>RT [sec]</td>
<td>RT [sec]</td>
</tr>
<tr>
<td>WF B1</td>
<td>0.25756</td>
<td>0.25674</td>
<td>0.25674</td>
</tr>
<tr>
<td>WF B2</td>
<td>0.25401</td>
<td>0.25317</td>
<td>0.25317</td>
</tr>
<tr>
<td>WF B3</td>
<td>0.43748</td>
<td>0.44141</td>
<td>0.44141</td>
</tr>
<tr>
<td>WF B4</td>
<td>13.30888</td>
<td>15.72755</td>
<td>15.65380</td>
</tr>
<tr>
<td>WF B5</td>
<td>0.01870</td>
<td>0.01869</td>
<td>0.01869</td>
</tr>
<tr>
<td>WF B6</td>
<td>0.01843</td>
<td>0.01842</td>
<td>0.01842</td>
</tr>
<tr>
<td>WF B7</td>
<td>0.01868</td>
<td>0.01868</td>
<td>0.01868</td>
</tr>
<tr>
<td>WF B8</td>
<td>8.07691</td>
<td>10.40152</td>
<td>17.52653</td>
</tr>
</tbody>
</table>
In Figure 7 the response time deviations are visualized. Six of the shown workflows will not be concerned by such a load increase on the system. Without prioritization the workflow WF B4 has a response time deviation of 18.2% that can be decreased to 17.6% with the prioritization. Instead, the effect of this prioritization on workflow WF B8 is dramatically. In the increased load scenario the deviation is at 28.8%. This deviation increases in the prioritization scenario to the value of 116.9%. So the small advancement of workflow WF B4 leads to an enormous increase of the response time of workflow WF B8.

![Figure 7: Response time deviation of prioritization scenario compared to increased load scenario on system B](image)

As an example, figure 8 summarizes the positive effect of prioritization on workflow WF A2 of system A.

![Figure 8: Prioritization effect on workflow WF A2 has two dimensions: response time and workload benefit](image)

By assigning a higher priority value to workflow WF A2, the response time of this workflow was enhanced considerably. Figure 8 shows the two dimensions of the improvement. On the one hand, prioritization results in a response time benefit for the workflow. On the other hand, the workflow can serve a much higher load without a noticeable loss in performance.

### 6. Related Work

Performance Modelling is a common approach to gain insight into systems in Computer Science and Operations Research where load testing or benchmarking is either very expensive, time-consuming or even impossible at all. Target areas include file and memory systems, databases, computer networks, and operating systems [7, 8].

A well-studied methodology that can be applied in Performance Modelling is Queueing Theory. To our knowledge, most previous scientific work on Performance Modelling in the area of distributed IT infrastructures was restricted to rather small applications. For example Chen et al. focus in [9] on three-tier systems. We apply Queueuing theory to large-scale DTP systems including dozens of service components running on multi-tier systems.

Urgaonka et al. have recently applied queueing network models for predicting the performance of multi-tier internet services [10]. They solved their models by applying an exact analytic algorithm. In our work we focus on the impact of workflow prioritization strategies on the performance measures of the overall DTP system. Therefore, we solve the underlying queueing network simulative by using a custom-made discrete-event simulation engine.

Starvation is a main issue for prioritization in research in other areas. Most of the attempts use two queues for two different priority levels. For example in dynamic priority queueing [11] a counter guarantees that lower jobs are served at specific points of time. Threshold based priority queueing [12] uses the queue lengths as indicator for these points of time. Other queueing strategies use more than two queues [13].

Similar to the alpha scheduling approach in [14] we make use of a shared single queue per station with changing priority levels for the incoming jobs. For the experiments of this paper we implemented a priority-first dispatching strategy in our simulation engine. In the future we plan to explicitly consider starvation aspects in our models by applying dynamic priority queueing.

### 7. Conclusions

DTP systems are the IT backbone of today’s service industry. Proactive capacity and performance management of such systems is essential, as predefined quality-of-service metrics must be met. While the inherent complexity of DTP systems makes provisioning decisions already challenging, the highly dynamic workload mix and overall demand makes them even harder. Adaptive workload prioritization is a
way to reduce costly capacity over-provisioning. In this paper we described the particularities of two real-world DTP systems of a telecom provider including the workload specifics. We applied performance-modelling techniques, namely queueing theory and simulation, to evaluate the impacts of prioritization strategies on the performance measures of the overall system. This is important for IT service providers as it enables proactive capacity planning including workflow prioritization where the knowledge of resulting system performance is essential.

Our experiments based on real-world data showed that it is possible to react on short-term variances with adaptive prioritization. The side effects of this approach depend certainly on the DTP system and influence the pre-defined SLAs of all workflows. Therefore we evaluated the results of several prioritization scenarios on the supported overall system load and showed the gained end-to-end workflow response time benefits for the prioritized workflows.

8. References