Measuring and Improving the Quality of Services Provided by Data Centers: a Case Study

Martin Zinner*, Kim Feldhoff*, Michael Kluge*, Matthias Jurenz*, Ulf Markwardt*, Daniel Sprenger*,
Holger Mickler*, Rui Song†, Andreas Tschipang*, Björn Gehlsen*, Wolfgang E. Nagel*
* Center for Information Services and High Performance Computing (ZIH)
Technische Universität Dresden
Dresden, Germany
E-mail: {martin.zinner1, kim.feldhoff, michael.kluge, matthias.jurenz}@tu-dresden.de,
{ulf.markwardt, daniel.sprenger, holger.mickler, andreas.tschipang}@tu-dresden.de,
{bjoern.gehlsen, wolfgang.nagel}@tu-dresden.de
† Technical Information Systems
Technische Universität Dresden
Dresden, Germany
E-mail: rui.song@tu-dresden.de

***Abstract*—As data centers become increasingly complex and deliver services of high importance, it is very important that the quality of the delivered services can be objectively evaluated and can fulfill the expectations of the customers. In this paper, we present a novel, general, and formal methodology to determine and improve the Quality of Services(QoS) delivered by a data center. We use a formal mathematical model and methodology in order to calculate the overall indicator of the service quality and discuss methods of improving the QoS. Since the considerations were conceived and results have been proved in a formal model, the considerations and results also hold in a more general case.**

Keywords–Quality of Services; QoS; Performance data center; Little's Law; Kingman's equation; Flow factor; Operating curve management; Customer satisfaction; Key performance indicators.

I. INTRODUCTION

A. Motivation and Short Overview

Nowadays, the services of a data center are indispensable for the good functioning of a company or a research institute. However, due to the advanced digitalization, data centers are becoming more and more complex and difficult to manage. According to a survey of Symantec [1], the main reasons are the raise of the Cloud Computing and the Virtualization. Basically, such complex infrastructures are more error-prone and require more maintenance efforts than simple ones. Thus, it is of crucial importance to measure the Quality of Services (QoS) provided by a data center, in order to detect which components / services are low performers and should be improved. This way, measuring the QoS also avoids service degradations. Services which underperform can be detected and measures can be taken (like relocation of resources) such that these services will perform better again. An optimized usage of the available resources does not only improve the QoS and thus, the image of the service provider, but also helps to save costs.

Furthermore, Butnaru [2] states that "quality has become a strategic element in companies dealing with services because it determines competitiveness at its highest level". Thus, by measuring and improving the QoS provided by companies / research institutes, the service providers can improve their ranking when compared to the competition.

Estimating the QoS of a data center is a complex endeavor. On the one hand, there are objectively measurable indicators like the duration and number of unplanned down times. On the other hand, the customer satisfaction has very important subjective components, which should not be neglected. Thus, if a customer has full confidence in the technical skills, seriousness, and professionalism of the operating staff, then his attitude is permissive and indulgent regarding possible malfunctions. For example, let us consider the scenario that a service has an unplanned downtime. If the operating staff can predict the time when the resumption of the service will occur with satisfying accuracy, the impression of the customer regarding the service provider will be very good. Otherwise, the customer will assume that the service provider does not have his processes under control and a failure of the system will sooner or later occur.

In this paper, we will focus on the perspective from the data center side. We will define and make use of different metrics in order to be able to establish objective criteria which characterize the Quality of Services and the performance of a data center.

B. Main Challenges and Objectives

If a customer is asked about the quality of the services of a data center he or she usually will answer: Yes, quality is good, but it could be better. This answer only describes the subjective perception of the customer. Our aim is to go further. Thus, searching for a positive response to the questions "Is the QoS measurable and if this is the case, how?" is one of the main challenges, we had to accept and take up.

Establishing and choosing meaningful performance indicators form the basis for improving the QoS.

The challenges described above lead us to the following main objectives:

1) Receive responses to the question whether the QoS is quantifiable / metrisable or not, i.e., whether the QoS can be expressed numerically in a reasonable non trivial way, such that this number is independent of the subjective perception of humans.

 ISBN: 978-1-61208-668-2

2) Establish a general approach on the modalities to quantify the QoS such that a single indicator (or only very few) expresses the service quality of the service provider.
3) Find possibilities to improve the QoS and implicitly the performance of the service provider.

C. Outline

The remainder of the paper is structured as follows: Section II gives a short overview of the state of the art and detail some difference of our approach. Section III introduces the proposed strategy for measuring the QoS, first in an informal way, afterwards formalized by introducing a mathematical model. Different metrics are defined and used in order to be able to establish objective criteria which characterize the quality of the services and the performance of a data center.
Section IV introduces the formal queuing model, makes the connection to the models used in practice, and discusses modalities to improve the performance of a data center by using the operation curve. In order to be able to balance the performance of the services of different departments of a data center, a formula to calculate the flow factor of the data center out of the flow factor of each department, is established. Section V gives some details of a use case and finally, Section VI concludes this paper and sketches the future work.

II. RELATED WORK

An important part of the existing approaches for quality improvement focus merely on the QoS from the user perspective – established through questionnaires (e.g., SERVQUAL and/or SERVPERF [3]) – and on the discrepancies between the user perception and the user expectation of the QoS.

Most approaches concerning the measurement of QoS have tended to avoid the use of pre-defined objective performance indicators and focus instead on the relationship between what consumers expect from a particular service and what they actually get [4]. The conclusion [3] is that customer satisfaction with services or perception of QoS can be viewed as confirmation or disconfirmation of customer expectations of a service offer. The role of emotions in customer-perceived service quality is analyzed [5] by widening the scope of service quality, i.e., by focusing on dimensions beyond cognitive assessment.

We concentrate our study primarily on the service provider perspective by using metrics to characterize the QoS and subsequently establish strategies on how those metrics can be combined together to generate a unique indicator, which characterizes the overall performance of the service provider.

Measuring and ranking service quality has been an issue for study for decades [4], whereby the difficulties lied in the development of the most suitable method of measurement. Approaches to the measurement of QoS are based on the analysis of the relationship between customer expectation of a service and their perceptions of it's the quality. Indices to provide measures of expectation, perceptions, and overall satisfaction from the customer side are set up and compared [4].

In [6], the authors report the insights obtained in an extensive exploratory investigation of quality in four business (retail banking, credit card, security brokerage, and product repair and maintenance) by developing a model of service quality. The most important insight obtained from analyzing the executive responses is the following: "A set of key discrepancies or gaps exists regarding executive perception of service quality and the tasks associated with service delivery to consumers. These gaps can be major hurdles in attempting to deliver a service which consumers would perceive as being of high quality".

Metrics in order to establish the QoS have been used for example by the Systemwalker [7], which supports "Information Technology Infrastructure Library" (ITIL) based IT service management. The focus in [7] is on the service delivery area, such as capacity, availability, and service level management. The composition of metrics is outside the scope of the Systemwalker.

In [8], a framework for the evaluation of QoS for Web Services within the OPTIMACS project is presented, such that Service Level Agreements (SLAs) are established in order to calculate / guarantee the QoS, then the properties are normalized by using statistical functions. The goal is to obtain a final Quality grade, that allows to rank the services. Finally, aggregation is performed using weighted sum of the different quality items.

As a final note, the studies regarding the normalization and composition of metrics considered for QoS for Web Services are straightforward and are based on statistics (min, max, mean value, standard deviation, Z-score) [8], the committed SLA time provides the QoS level. The metrics used to measure the QoS of a data center are so diverse that a case-by-case approach is necessary to determine the normalization and composition strategy. Moreover, statistical values as above are generally not a priori known for unconverted metrics such as "cycle time", etc.

III. MEASURING THE QOS

We describe the general strategy how to measure the QoS in an informal way in Subsection III-A and formalize this strategy in Subsection III-B.

A. Description of the Strategy

According to ITIL [9] (and similar), the (incomplete) list of processes comprises the following managements:

- "incident management",
- "problem management",
- "information security management",
- "service level management",
- "change management",
- "project management", and
- "release and deployment management".

A list of metrics is specified for each process according to ITIL [9] and / or to the "Key Performance Indicator (KPI) Library"(KPI Library) [10], see [11] regarding developing, implementing and using KPIs. Some of the most important metrics for the ITIL process “incident management” are given in the following:

- “Total number of incidents”,
- “Number of repeated incidents, with known resolution methods”,
- “Number of incidents escalated which were not resolved in the intended resolution time”,
- “Average cycle time associated to the subsequent responses”, i.e., including the average cycle time to resolve the incident,
- “Average waiting time from user side associated to the subsequent responses”,
- “Average work effort for resolving the incident associated to the subsequent responses”,
- “Average time between the occurrence of an incident and its resolution”,
- “Total number of incidents resolved within service level agreement (SLA) time divided by the total number of incidents”.

In order to establish objective criteria for measuring the QoS, it is not sufficient to consider simply one metric. Indeed, different metrics have to be combined. The following example illustrates this issue.

The metric “Total number of incidents” is a revealing metric regarding the performance of a service provider. Of course, this metric is important for reporting per se, as a non anticipated sharply increasing trend can be the cause for major concerns. Another important metric is "the average cycle time to solve an incident". If the metric “Total number of incidents” is increasing, but in the mean time the “Average cycle time to solve an incident” is decreasing, the balance is restored and the service provider will not face a total collapse of the service.

Hence, composition rules for metrics are needed, such that indicators that characterize the health of the services, can be established. Since we cannot directly compare the different metrics, we transform / normalize the metrics using relative values. By dividing the “Total number of incidents” by an artificially generated “Maximum number of incidents supported”, we receive a relative value between 0 and 1. Unfortunately, the value 1 is the worst value you can ever get. In order to circumvent this impediment, we subtract 1 and change the sign. Using the same considerations (by defining the “Minimum average cycle time”) analogue relative value for the cycle time can be established. In this case, this new indicator is directed in the sense, that the best cycle time is achieved when this value is equal to 1. This example is just to illustrate the technique. One may argue that an increase of the indicator value “Average waiting time of the incidents during processing” also indicates a congestion.

Thus, in order to combine different metrics, we will normalize them to the range $[0; 1]$ in such a way that the lowest value correspond to the poorest quality, the highest value to the best quality. Once, all the relevant metrics of a process are normalized, we can proceed with the composition such that for each process a single, composed metric is established.

The composed metric should also take values between 0 and 1, such that a greater value implies a better QoS. An example of a straightforward composing strategy is to establish weights for each metric, such that the sum of all weights is equal to 1 and important metrics have bigger weights. Hence, the decisive metrics are much better considered. Of course for practical purposes, we can define groups of incidents having the same importance and accordingly appropriate distribution functions (linear, exponential, etc.). The calculation of the associated weights is then immediate.

Normally, explicitly defining importance grouping and distribution functions is not always necessary. We can set up priority strategies regarding the QoS. As an example, under some circumstances, a fast but not necessarily very detailed answer is more helpful for IT professionals, who can elaborate the details themselves. In other cases, detailed and very accurate answers are necessary, especially for customers with little or no experience. Then, customers could return the ticket of the incident (if the answer is not accurate enough, e.g.) and ask for more information and assistance.

Hence, the development of an appropriate strategy for the quality improvement is essential, in some cases this strategy can be even customer dependent. For example, we can improve the quality:

- by improving only the accuracy of the responses, or
- by reducing only the response times, or
- by minimizing a metric which takes both accuracy and the response time into account.

In accordance with the improvement strategy, the grouping of metrics regarding their performance is more or less straightforward and easy to follow.

In effect, we can establish for each process a unique (abstract) indicator, which characterizes the quality of the process such that a greater value means better quality of the process according to the improvement strategy as above. The absolute value of this indicator has no particular interpretation, only the increment or decrement of this value in time is significant.

Same considerations using the indicators established for the processes lead to a unique indicator of the QoS for the whole service provider, i.e., the data center. By evaluating the time behavior of this indicator and / or the component indicators we can have a good overview which process and / or metrics performed better or worse.

This unique indicator can be deployed for example on daily bases, such that the performance of the service provider can be easily followed and appropriate measures can be taken if performance degradation occurs. Moreover, even if the overall unique indicator has improved in value, there can be some components, whose performance has degraded. By setting up appropriate Graphical User Interfaces (GUIs), and appropriate colors (for example red for degradation and green for improvement) the deviation with respect to the previous day can be visualized.

The only process through which the customers interact with the data center as the service provider is through the "incident management", the performance of the other processes is practically hidden for the regular customer. In order to improve the "incident management" we will analyze the impact of some important processes on the "incident management".

An important direct impact on the "incident management" has the "problem management" in the sense that by a very efficient "problem management" the number of repetitive incidents or the time to solve the repetitive incidents can be dramatically reduced. For this, each incident should be correctly assigned to the appropriate issue, have a correct and exhaustive root analysis, such that the causes of the incident are unambiguously elucidated. It seems a bit of common sense that all the detailed information regarding the incident including good ways of searching, finding, and retrieving the information should be stored in an appropriate knowledge database. Next, the probability of recurring should be estimated and if necessary, appropriate measures should be taken in order to avoid the next occurrence of the same incident.

Proactive methods are very efficient to avoid the occurrence of incidents, e.g., improving "change management", "release and deployment management", etc. By significantly reducing the impact of new releases on the services, the peaks on the QoS can be significantly reduced.

B. *Formalization of the Strategy*

We will formalize the strategy proposed in Subsection III-A by introducing a mathematical model in order to use the advantages of the rigor of a formal approach over the inaccuracy and the incompleteness of natural languages.

Let $\mathcal{A}$ be an arbitrary set. We notate by $2^{\mathcal{A}}$ the power set of $\mathcal{A}$, i.e., the set of all subsets of $\mathcal{A}$, including the empty set and $\mathcal{A}$ itself, and the cardinality of $\mathcal{A}$ by $\operatorname{card}(\mathcal{A})$.

We use a calligraphic font to denote index sets. We denote by $\mathcal{S} := \{S_i \mid i \in \mathscr{S} \text{ and } S_i \text{ is a service}\}$ the finite set of the services. Analogously, we denote by $\mathcal{P} := \{P_i \mid i \in \mathscr{P} \text{ and } P_i \text{ is a process}\}$ the finite set of processes and by $\mathcal{T} := \{[t_1; t_2] \mid t_1 \text{ and } t_2 \text{ are points in time, such that } t_1 \leq t_2\}$ time intervals.

A metric M is a measurement that monitors progress towards achieving the targeted objectives. We denote by $\mathcal{M} := \{M_i \mid i \in \mathscr{M} \text{ and } M_i \text{ is a metric}\}$ the finite set of metrics. Generally speaking, a metric M is defined for an environment containing subsets of $\mathcal{S}$ and $\mathcal{P}$.

For example, let us define the ratio between the "total number of incidents with known resolution method" and the "total number of incidents". Depending on the strategic orientation of the company, different goals can be pursued. On the one hand, having for most of the incidents corrective measures in place can be a targeted objective, one the other hand, avoiding repetitive incidents is crucial for the economic success of companies like fabs running 24x7 continuous manufacturing operations. A mixed strategy (for example 10% known errors) can be also targeted. Hence, the scope of a metric is most of the time business oriented.

In order to be able to compare and compose different metrics in a reasonable way, we introduce the value of a metric such that it is greater or equal 0 and lower or equal 1. A greater value of the metric means a closer value to the targeted business objectives. Formally, the range of values of the *possible* business values, including the targeted ones is $2^{\mathbb{R}}$. Hence, the progress towards achieving the targeted Business Objectives (BO) can be represented as a function.

$$\begin{aligned} BO : \mathcal{M} \times \mathcal{P} \times \mathcal{S} &\to BusinessObjectives, \\ (M, P, S) &\mapsto BO(M, P, S). \end{aligned}$$

Analogously, the value (V) of a metric is represented as:

$$\begin{aligned} V : \mathcal{M} \times \mathcal{P} \times \mathcal{S} \times \mathcal{T} &\to [0; 1], \\ (M, P, S, [t_1; t_2]) &\mapsto V(M, P, S, [t_1; t_2]). \end{aligned}$$

A greater value for $V(M, P, S, [t_1; t_2])$ means a closer value to the targeted business objectives for (M, P, S). The definition above highlights the fact that the same metric can have different business objectives and definition (values) depending on the environment (services and/or processes) it is used.

We illustrate the above considerations based on a simple example and consider the "average cycle time" of the incidents. The business demands short cycle times for all departments. In order to be able to compare the cycle times of different departments, we determine the minimal cycle time (i.e., the theoretical cycle time needed if there are no unplanned down times, etc.) and assign the ratio of minimal cycle time to the cycle time as the value of the metric. Hence, the performance of the different departments regarding the same metric (i.e., cycle time) can be easily compared, on the assumption that the respective minimal cycle time has been evaluated correctly.

Our aim is to establish a single indicator for the service performance (i.e., the QoS) of the service provider. In order to evaluate the performance of the different metrics of the same process (for example ITIL process), we set up a methodology to compose the different metrics in a reasonable way, such that the new metric (indicator) outlines the performance of the investigated process.

In order to simplify the notation, we will notate in the following the value of a metric M by $V(M)$, meaning that the metrics involved are defined on the same environment and the same time interval.

Definition III.1 (Composition of metrics) *Let* $\mathfrak{M} := \{M_i | i \in \{i_1, i_2, \ldots, i_k\} \subseteq \mathscr{M}\}$ *a subset of* $\mathcal{M}$. *We define*

$$\begin{aligned} COMP : 2^{\mathcal{M}} &\to \mathcal{M}, \\ \mathfrak{M} &\mapsto COMP(\mathfrak{M}), \end{aligned}$$

such that there is an aggregation function AGG

$$\begin{aligned} AGG : 2^{\mathcal{M}} &\to [0; 1], \\ V(COMP(\mathfrak{M})) &\mapsto AGG(V(M_{i_1}), V(M_{i_2}), \ldots, V(M_{i_k})) \end{aligned}$$

and

$$\begin{aligned} &v^1_{i_1} \leq v^2_{i_1}, v^1_{i_1} \leq v^2_{i_1}, \ldots, v^1_{i_k} \leq v^2_{i_k} \\ &\qquad \Rightarrow AGG(v^1_{i_1}, v^1_{i_2}, \ldots, v^1_{i_k}) \leq AGG(v^2_{i_1}, v^2_{i_2}, \ldots, v^2_{i_k}) \end{aligned}$$

Except for the case of trivial aggregations, the composition generates a new metric out of known ones.

In order to keep our notation simple and straightforward, we will not make any distinction in the formal representation of the initial metrics and those obtained by consolidation. Hence, $\mathcal{M}$ contains the initial metrics as well as the consolidated ones. Therefore, a consolidated metric can be finally set up for the entire service. We note:

Lemma III.2 (Composition properties) *Let $\mathfrak{M} := \{M_i | i \in \{i_1, i_2, \ldots, i_k\} \subseteq \mathcal{M}\}$ a subset of $\mathcal{M}$ arbitrarily chosen. Then, $COMP(\mathfrak{M})$ is a metric, i.e., fulfills the following properties:*

a) $0 \leq V(COMP(\mathfrak{M})) \leq 1$,

b) *A greater value for $V(COMP(\mathfrak{M}))$ means a closer value to the targeted business objectives for this metric.*

Hint *These properties are a direct consequence of Definition III.1.* ■

Next, we give a small example to illustrate the aggregation strategies. Let $\mathfrak{M} := \{M_{i_1}, M_{i_2}, \ldots, M_{i_k}\}$ be a subset of $\mathcal{M}$. We suppose, that the value of the new characteristic $COMP(\mathfrak{M})$ is a linear combination of the values of the components, i.e.,

$$V(COMP(\mathfrak{M})) := \sum_{i=i_1}^{i_k} \alpha_i \cdot V(M_i)$$

with weights $\alpha_i > 0 \; \forall i \in \{i_1, i_2, \ldots, i_k\}$ and $\sum_{i=i_1}^{i_k} \alpha_i = 1$. If the value of α_i is high, then the metric M_i within $COMP(\mathfrak{M})$ is important. In practice, it suffices to build weight groups $\{G_i | i \in \{i_1, i_2, \ldots, i_l\}\}$ out of $\mathfrak{M}$ such that each $M \in \mathfrak{M}$ belongs to a group G_i and all $M_j \in G_i$ are equally weighted. Furthermore, a weighting function W can be set up, such that all α_i can be explicitly determined, for example set

$$k_i := \frac{\alpha_i}{\alpha_{i+1}} \text{ for all } i \in \{i_1, i_2, \ldots, i_{l-1}\}.$$

The values k_i can be regarded as the “ratio of relevancy” of the corresponding metrics.

IV. Improving the QoS

In the last section, we proposed a strategy how to measure the QoS of a data center. In this section, we will establish metrics controlling the performance of a data center. Thus, we can determine in which cases the service of a data center collapses or the QoS substantially degrades.

A. Queuing Model and Basic Metrics

We model the processing line of a data center by introducing a queuing model and give some basic definitions related to it. In order to keep the presentation accessible and avoid technical complications, we will maintain our model as simple as possible. It is the task of the practitioners to map the real world onto this model according to their needs. We will analyze the entire processing line as well as subsystems of it.

A *queuing system* consists of discrete objects, termed *units* or *items* that arrive at some rate to the system. Within the system, the units may form one or more queues and eventually be processed. After being processed, the units leave the queue.

The finest granularity in our model is *unit*, *step*, *time stamp*, *section* and *classification*. For example, in practice, the unit can be a ticket, the section can be an employee of the service center, a group of employees having the same profile or a specific section of the service center, etc. The classification is the finest attribute which characterizes the unit (like bug, disturbance, project, etc.) and it can be distinguished in the processing phase.

In our model the unit enters the system (service center), is processed according to the specifications and leaves the system. The step is the finest abstraction level of processing which is tracked by the reporting system. When the material unit u enters the system, it is assigned to a classification c. This assignment remains valid till the unit u leaves the system. We will analyze the entire processing line as well as subsystems of it.

We denote by S the set of all steps of the processing line, by U the set of the units that entered the system, and by T the (ordered and discrete) points in time when events may occur in the system. Since we are merely interested in daily calculations, we will set D as the set of all points in time belonging to a specific day D, i.e., $D := \{t \in T| \text{ t belongs to day D}\}$.

Let $s \in S$ and $u \in U$. We denote by $TrInT_s(u)$ the *track in time* of u, i.e., the point in time when the processing of unit u is started at step s. Analogously, $TrOutT_s(u)$ is the *track out time* of u, i.e., the point in time when the processing of unit u has been finished at the step s.

We assume that for a step s, the function $succ_s(u)$, which identifies the succeeding step of s for the unit u is well defined. Analogously, we assume that the history of the production process is tracked, so the predecessor function $pred_s(u)$ of each step s is well defined. For formal reasons we set $succ_s(u) := s$ for the last step on the route and $pred_s(u) := s$ for the first step on the route.

By *cycle time* (CT), we generally denote the time interval a unit or a group of material units spent in the system / subsystem [12]. We do not make any restrictions on the *time unit* we use, but are merely interested on daily calculations. For formal reasons, – in order to be able to calculate average values – we denote by $24h$ the cardinality of an arbitrary day D. For $t \in T$ we denote by $t \pm 24h$ the point in time t shifted forward or backwards by 24 hours.

We assume that events in the system are repeated on a daily basis, i.e.,

$$\begin{aligned} &\forall u \in U \text{ and } \forall s \in S : TrInT_s(u) = t \\ &\Longrightarrow \exists v \in U : TrInT_s(v) = t + 24h \text{ and} \\ &\qquad TrOutT_s(v) = TrOutT_s(u) + 24h \end{aligned}$$

and

$$\begin{aligned} &\forall u \in U \text{ and } \forall s \in S : TrInT_s(u) = t \\ &\Longrightarrow \exists w \in U : TrInT_s(w) = t - 24h \text{ and} \\ &\qquad TrOutT_s(w) = TrOutT_s(u) - 24h. \end{aligned}$$

Under a *stable system* we mean a system according to the conditions above.

In practice, systems pass through a ramp up phase such that the above conditions are eventually reached, i.e., $\exists t_b \in T$ such that the above conditions are satisfied for all $t > t_b$. For our investigations, it is sufficient that the systems reach the stable state after some time (*eventually stable systems*). For reasons of a simple notation, we will use the term *stable system* or *system in a stable state*. However, the statements of this work are also valid for eventually stable systems.

The *raw process time / service time* of unit $u \in U$ related to step $s\ inS$ is the minimum processing time to complete the step s without considering waiting times or down times. We denote the raw process time of unit u related to step s by $RPT_s(u)$.

Let $u \in U$, let $\{s_1, s_2, \ldots, s_n\} \subset S$ be the complete list of steps according to the processing history to process unit u. Let $RPT_{s_i}(u)$ be the raw process times of unit u related to step s_i for all $i = 1, 2, \ldots, n$. Then the raw process time of unit u can be represented as follows:

$$RPT(u) = \sum_{i=1}^{n} RPT_{s_i}(u).$$

The work in progress is defined as the inventory at time $t \in T$ and will be denoted by $WIP(t)$. If the work in process is used in connection with Little's Theorem then it denotes the average inventory for a given period of time. We use the notation $avgWIP$ instead of WIP to denote the average inventory.

We denote by Th the throughput of the material units by. Usually, we consider the daily throughput and refer to it as Th^D for a specific day D.

The cycle time of a unit $u \in U$ spent at a step $s \in S$ in the system can be represented as:

$$CT_s(u) := TrOutT_s(u) - TrOutT_{pred_s(u)}(u).$$

Let $\{u_1, u_2, \ldots, u_n\}$ be the set of units that were processed at step s on a specific day $D \subset T$ i.e., $\forall i \in \{1, 2, \ldots, n\}\ \exists t_i \in D$ such that $TrOutT_s(u_i) = t_i$. Then, the average cycle time $avgCT_s^D$ the units u_i spent in the system at step s on a specific day D can be represented as:

$$avgCT_s^D = \frac{1}{n} \cdot \sum_{i=1}^{n} CT_s(u_i).$$

For $t \in T$, $u \in U$ we define the indicator function 1_s at a process step $s \in S$ as follows:

$$1_s : U \times T \to \{0, 1\},$$
$$(u,t) \mapsto 1_s(u,t) := \begin{cases} 1 & \text{if } t \geq TrOutT_{pred(s)}(u) \text{ and} \\ & \quad t < TrOutT_s(u), \\ 0 & \text{otherwise.} \end{cases}$$

Throughout this work we assume that T is discrete, i.e., units arrive and depart only at specific points in time, since the time is usually measured in seconds or milliseconds.

Lemma IV.1 (Representation of average inventory) *The average inventory $avgWIP_s^D$ for a process step $s \in S$ on a specific day D can be represented as follows:*

$$avgWIP_s^D = \frac{1}{\mathrm{card}(D)} \cdot \sum_{t \in D} \sum_{u \in U} 1_s(u,t) \tag{1}$$
$$= avgCT_s^D \cdot Th_s^D. \tag{2}$$

By interchanging the order of summation, we receive an expression for WIP_s^D, which is much easier to calculate in practice.

Hint *Let $U_{n,D} := \{u_1, u_2, \ldots, u_n\}$ be the set of units that left the step s on a specific day $D \subset T$, i.e., $\forall i \in \{1, 2, \ldots, n\}\ \exists t_i \in D$ such that $TrOutT_s(u_i) = t_i$. Then, in stable systems the following relation holds:*

$$avgWIP_s^D = \frac{1}{\mathrm{card}(D)} \cdot \sum_{t \in D} \sum_{u \in U} 1_s(u,t)$$
$$= \frac{1}{\mathrm{card}(D)} \cdot \sum_{t \in T} \sum_{u \in U_{n,D}} 1_s(u,t).$$

By interchanging the order of summation and considering that for $i \in \{1, 2, \ldots, n\}$ the average cycle time (measured in days) of the material unit u_i at step s is given by:

$$avgCT_s(u_i)$$
$$:= \frac{1}{\mathrm{card}(D)} \cdot \left(TrOutT_s(u_i) - TrOutT_{pred(s)}(u_i)\right)$$
$$= \frac{1}{\mathrm{card}(D)} \cdot \sum_{t \in T} 1_s(u_i, t).$$

Thus, we get:

$$avgWIP_s^D = avgCT_s^D \cdot Th_s^D.$$

Since in stable systems the variables above do not depend on the day chosen for their calculation, Little's Theorem follows. The consideration above do not hold in steady state systems used by Stidham and Sigman (see [13] or [14]). ■

Remark IV.3 (Case: Set of points in time is continuous) *Actually, in theoretical models it is not necessary to consider a discrete set T in order to be able to calculate $avgWIP_s^D$. Let Σ_U be the discrete σ-algebra on U (i.e., the power set 2^U of U). Let μ be the counting measure on Σ_U, i.e., $\mu(\mathcal{U}) := |\mathcal{U}|$ for $\mathcal{U} \in \Sigma_U$. Then, (U, Σ_U, μ) is a measure space. For $T \subset \mathbb{R}_+$ let Σ_T be the σ-algebra of all Lebesgue measurable sets on T and let λ the usual Lebesgue measure on T. Analogously (T, Σ_T, λ) is also a measurable space. Since both spaces are σ-finite, the product measure $\mu \otimes \lambda$ is well defined and for $\mathcal{U} \subset U$ and $\mathcal{T} \subset T$ the equality $\mu \otimes \lambda(\mathcal{U} \times \mathcal{T}) = \mu(\mathcal{U}) \cdot \lambda(\mathcal{T})$ holds. Since 1_s is a simple function (i.e., a finite linear combination of indicator functions of measurable sets) it is $\Sigma_U \times \Sigma_T$ measurable. Then, as expected $\mathrm{card}(D) = \int_D \mathrm{d}\lambda(t) = 24h$ and the theorem of Fubini-Tonelli gives:*

$$avgWIP_s^D = \frac{1}{\mathrm{card}(D)} \cdot \int_D \int_U 1_s(u,t)\,\mathrm{d}\mu(u)\,\mathrm{d}\lambda(t)$$
$$= \frac{1}{\mathrm{card}(D)} \cdot \int_{U \times D} 1_s(u,t)\,\mathrm{d}(\mu \otimes \lambda)(u,t)$$
$$= \frac{1}{\mathrm{card}(D)} \cdot \int_U \int_D 1_s(u,t)\,\mathrm{d}\lambda(t)\,\mathrm{d}\mu(u).$$

The last integral is much easier to evaluate.

In stable systems the value $avgWIP_s^D$ does not depend on the specific day D that was considered for the calculation.

ISBN: 978-1-61208-668-2

B. Expected Inventory

Next, we define one of the relevant metric for bottleneck control and present formulas to calculate them.

$\mathit{WIP24}_{s,c}(t)$ denotes the inventory which is expected in the next 24 hours at a specific step $s \in S$, classification c and $t \in T$. Usually, $\mathit{WIP24}_{s,c}(t)$ at midnight is considered. In this case, we will omit the time constraint and use the notation $\mathit{WIP24}_{s,c}$.

Let us suppose $\{s_1, s_2, \dots, s_n\}$ is the planned (ordered) list of steps as provided by the route for the classification c. There are of course different strategies to estimate $\mathit{WIP24}_{s_l,c}(t)$ for a specific $l \in \{1, 2, \dots, n\}$. One alternative supposes that the units moves across the line as planned by the route. Let $\mathit{refCT}_{s_i,c}$ be the target cycle time to process the unit at the step $s_i \in S$, let $\mathit{WIP}_{s_i,c}(t)$ be the inventory at the step s_i for the classification c and time $t \in T$. For l determine $j := min(k : k \leq l)$ such that $\sum\limits_{k \leq i \leq l} \mathit{refCT}_{s_i,c} \leq 24h$. Then the expected inventory can be written as follows:

$$\mathit{WIP24}_{s_l,c}(t) = \sum_{j \leq i \leq l} \mathit{WIP}_{s_i,c}(t).$$

Most of the time, the unit is not processed according to the specifications (route), reworks or alternative processing strategies are necessary. In this case, the formula as above does not hold, and other more complex approaches are necessary.

C. Little's Theorem

In the following, we will introduce Little's Theorem [15] [16]. Little's Theorem which is mostly called *Little's Law* is a mathematical theorem giving a rather simple relation between the average cycle time, the throughput, and the average work in process in the system. It will be used later on for calculating the flow factor and thus, controlling the performance of the data center. The relation of Little's Theorem is valid if some convergence criteria are fulfilled and if the underlying system is in steady state and non-preemptive. The latter means that the properties of the system are unchanging in time, there are no interrupts and later resumes. In many systems, steady state is not achieved until some time has elapsed after the system is started or initiated. In stochastic systems, the probabilities that some events occur in the system are constant. The result is entirely independent of the probability distribution involved and hence it requires no assumption whether the units are processed in the order they arrive or the time distribution they enter or leave the system.

We give now a formal definition for Little's formula. Our explanation is based on [14] slightly modified to use our notations. We consider the queuing system above where – unlimited but countable – units arrive, spend some time in the system, and then leave. Material units enter at most once the system, i.e., units that left do not enter the system again. Let $T := \{t_i | \, i \in \mathbb{N}\}$ be the countable set of points in time when those events occur. At any point in time $t \in T$ at most a finite number of units enter or leave the system. Let u_n denote the unit which enters the system at the time t^e_n. Upon entering the system, u_n spends CT_n time units in the system (the cycle time of u_n) and then leaves the system at time $t^d_n = t_n + CT_n$. The departure times are not necessary ordered in the same way as the enter times. This means that we do not require that the units leave the system in the same order as they arrived. Let $1^e_{u_i}(t) := 1$ if $t_i \leq t$ and 0 else. We denote by $N^e(t)$ the number of units which entered the system until time t, i.e.,

$$N^e(t) = \sum_{i=1}^{\infty} 1^e_{u_i}(t).$$

Analogously, we denote by $N^l(t)$ the number of units which have left until time t. Let $L(t)$ be the total number of the units in the system by time t. A unit u_n is in the system at time t if and only if $t_n \leq t < t_n + CT_n$. Hence $L(t) = N^e(t) - N^l(t)$. Let be (if the limit exists)

$$Th := \lim_{t \to \infty} \frac{N^e(t)}{t}$$

the arrival rate into the system,

$$avgCT := \lim_{n \to \infty} \big(\frac{1}{n} \cdot \sum_{j=1}^{n} CT_j\big)$$

the average cycle time the unit spends in the system,

$$avgWIP := \lim_{t \to \infty} \big(\frac{1}{t} \cdot \int_0^t L(s)\mathrm{d}s\big)$$

the average number of units in the system.

Theorem IV.4 (Little's theorem) *If both the arrival rate Th and the average cycle time $avgCT$ exist and are finite, then the above limit in the definition of the average inventory $avgWIP$ exists and it holds:*

$$avgWIP = avgCT \cdot Th. \tag{3}$$

Corollary IV.5 *If both Th and $avgCT$ exist and are finite, then the departure rate exists and equals the arrival rate:*

$$\lim_{t \to \infty} \frac{N^l(t)}{t} = Th.$$

Little used a stochastic framework to define and prove of what is known as Little's Law, the approach we are presenting makes no stochastic assumptions, i.e., the quantities and processes are deterministic. There are other versions of Little's Theorem that allow batch arrivals, see section 6.2 of [14].

D. Calculation of the Flow Factor

Next, we establish a formula for the calculation of the flow factor for the processing line. For this, we restrict to the following queuing model: The *adapted queuing model* is based on the one given in Subsection IV-A with the following modifications:

- Units can enter and leave the system only through a finite number of gates.
- Each gate on the entering side has its correspondence on the exit side.
- The entering and the corresponding exit gate are connected by a lane.
- Once, the person entered the system, he can move forward only on the lane set up by the entering gate.

 ISBN: 978-1-61208-668-2

He cannot switch the lane or leave the system except the exit gates.

- Each lane contains a number of clerks, not defined in detail, such that before each clerk an internal queue is formed and the clerk does not necessarily process the requests instantly.
- The sum of the time the clerks process the requests of a person during his voyage through a given lane (i.e., the raw process time) does not depend on the particular person involved. Hence, the system has a predefined raw process time (RPT^l) for each lane, i.e., the sum of the time the clerks process the requests of a person during his voyage through the lane.

Table I illustrates the queuing model.

Table I. ILLUSTRATION OF A QUEUING SYSTEM WITH 5 LANES $l_1, l_2, \ldots, l_5$ AND 8 PROCESSING STEPS.

l_1	⇒	■	■		■			■	■	⇒
l_2	⇒	■		■	■	■	■		■	⇒
l_3	⇒	■	■				■	■	■	⇒
l_4	⇒	■		■		■			■	⇒
l_5	⇒	■					■		■	⇒

We will denote by L the set of the lanes and by Th^l the throughput at lane $l \in L$.

Definition IV.6 (Flow factor) *Let $\{u_1, u_2, u_3, \ldots\}$ be the ordered list of units which enter the system, such that u_i enters the system at time t_i and $i < j \Rightarrow t_i \leq t_j$. The cycle time CT_i a unit $u_i \in U$ spent in the system can be split into the waiting time (WT_i) and raw process time RPT_i such that $CT_i = WT_i + RPT_i$. If the limit exists, then the (average) flow factor $avgFF$ is defined as:*

$$avgFF := \lim_{n\to\infty} \frac{\sum\limits_{i=1}^{n} CT_i}{\sum\limits_{i=1}^{n} RPT_i}. \tag{4}$$

Remark IV.7 *If $CT := \lim_{n\to\infty}\big(\frac{1}{n} \cdot \sum\limits_{i=1}^{n} CT_i\big)$ and $RPT := \lim_{n\to\infty}\big(\frac{1}{n} \cdot \sum\limits_{i=1}^{n} RPT_i\big)$ exists (and are finite), then the above limit exists and it holds:*

$$avgFF = \frac{CT}{RPT}.$$

Corollary IV.8 (Representation of raw process time) *Let $N_l^e(t)$ be the number of units which entered the lane $l \in L$ until time $t \in T$. Let $n := N^e(t_n) := \sum\limits_{l\in L} N_l^e(t_n)$. Then, the average raw process time RPT of the whole processing line can be represented as follows:*

$$RPT := \lim_{n\to\infty}\Big(\frac{1}{n} \cdot \sum_{i=1}^{n} RPT_i\Big) = \sum_{l\in L} \frac{Th^l}{Th} \cdot RPT^l. \tag{5}$$

Hint *We obtain:*

$$\frac{1}{n} \cdot \sum_{i=1}^{n} RPT_i = \frac{1}{n} \cdot \sum_{l\in L} \sum_{i_l=1}^{n} RPT_i^l = \sum_{l\in L} \frac{N_l^e(t_n)}{N^e(t_n)} \cdot RPT^l.$$

Since

$$\lim_{n\to\infty} \frac{N_l^e(t_n)}{t_n} \cdot \frac{t_n}{N^e(t_n)} = \frac{Th^l}{Th} \ \forall l \in L$$

it follows that

$$RPT := \lim_{n\to\infty}\Big(\frac{1}{n} \cdot \sum_{i=1}^{n} RPT_i\Big) = \sum_{l\in L} \frac{Th^l}{Th} \cdot RPT^l.$$

■

Corollary IV.10 (Representation of flow factor) *Assumed that the conditions of Little's Theorem are satisfied. Then, the flow factor can be represented as follows:*

$$\frac{1}{avgFF} = \sum_{l\in L} \frac{WIP^l}{WIP} \cdot \frac{1}{FF^l}. \tag{6}$$

Hint *Easy calculations using Little's Theorem for each lane $l \in L$ yields to the relationship between the flow factor for the whole system and the flow factors of its components / lanes as given in* (6). ■

We can calculate the average number of units in the system first by considering the whole system and secondly considering the reduced system with one lane. Little's formula is valid in both cases. Since units cannot switch to another lane, it follows that

$$WIP = \sum_{l\in L} WIP^l.$$

Using Little's formula and the definition of the average cycle time it follows that:

$$\lim_{n\to\infty}\Big(\frac{1}{n} \cdot \sum_{i=1}^{n} CT_i\Big) = CT = \sum_{l\in L} \frac{Th^l}{Th} \cdot CT^l.$$

Hence, as expected:

$$avgFF = \frac{\sum\limits_{l\in L} \frac{Th^l}{Th} \cdot CT^l}{\sum\limits_{l\in L} \frac{Th^l}{Th} \cdot RPT^l} = \frac{CT}{RPT}.$$

Let us suppose that the service center has different departments, such as for "incidents", "problems", "projects", "releases", etc., which operate independently. By abstracting those departments as lanes and calculating for each department the flow factor, the flow factor of the service center can be established as in (6).

Moreover, Equation (6) determines the correlation between the flow factors of each department and the flow factor of the data center. Thus, the flow factor of the data center can be improved within an existing budget, for example by resource reallocation, if the flow factor of some departments will be improved and the flow factor of some other departments will be degraded, see also the discussion regarding the operating curve.

A formula of the type given in (6) was proposed by Hilsenbeck in [17, p. 36]:

$$avgFF = \sum_{l\in L} \frac{Th^l}{Th} \cdot FF^l. \tag{7}$$

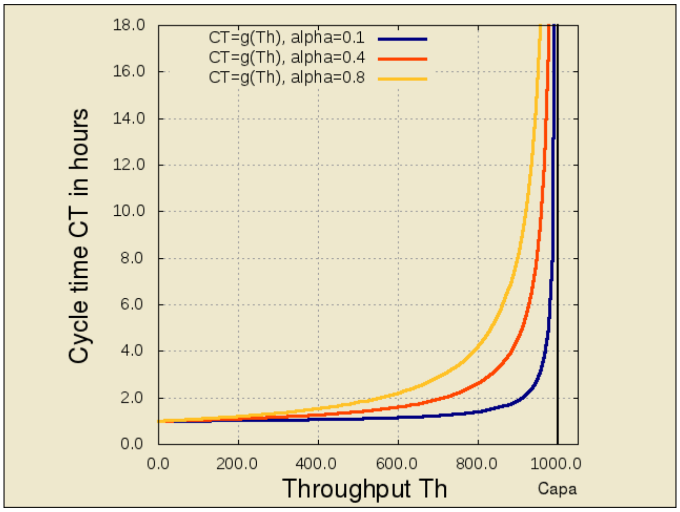

Figure 1. Graph of function g given in (10) (Operating curve). Throughput Th versus cycle time CT for four different values of α.

It seems that the formula (7) is empirical. In particular, no proof of the formula was given.

The flow factor plays an important role in the operating curve management. The operation curve follows from Kingman's equation [18]. One of the representation of the operating curve is based on the following formula (see [19, pp. 55, 58], [17, pp. 41, 44], [20] [21]).

$$avgFF = f(U) := \alpha \cdot \frac{U}{1-U} + 1. \tag{8}$$

U is the *utilization*, i.e., the percentage of the capacity $Capa$ of a tool or production segment (see [22] for a definition and [19, p. 57] for calculation). Introducing $avgFF$ and U in (8), the value for the coefficient α (variability) follows.

The operating curve as a function $avgFF(U)$ can be drawn. However, this relation is rather abstract. Since it holds

$$U = \frac{Th}{Capa}, \tag{9}$$

the flow factor in terms of a function $avgFF = f(U)$ can be easily transformed into a function of the type $CT = g(Th)$ (see [19, p. 40]):

$$CT = g(Th) := \alpha \cdot \frac{Th}{Capa - Th} \cdot RPT + RPT. \tag{10}$$

This relation is more practical as it shows how the throughput directly influences the cycle time. The self-generated graph of the function g is depicted in Figure 1. It is assumed that the average minimal cycle time RPT is 1 hour and that the maximal capacity is $Capa = 1000$. If Th is close to $Capa$, then the graph of g grows asymptotically. Hence, a point at the graph (named *operating point)* has to be chosen, such that a minimal increase of the number of items does not lead to dramatically increased cycle time. The operating curve has been used by Qimonda to improve overall fab performance.

V. Use case: An excerpt

We illustrate the principles of improving the QoS of a data center by means of a simplified example. Let us consider the department which provides the e-mail service of a data center. Firstly, we establish the conditions such that the providing the service is at all possible. Secondly, we set up metrics and compose them in order to be able to track the evolution of QoS.

One of the most sensible indicators whose value has to be estimated is the raw process time RPT which is the (average) minimal cycle time to process an incident. It contains only the effective time to process the incident, for example not including coffee breaks, private telephone calls, etc. Let us suppose that RPT is equal to 1 hour. In real systems (see [19, pp. 46, 48]) the cycle time CT corresponding to a specific throughput, denoted by Th is measured. Let us suppose that by considering the raw process time the maximum capacity $Capa$ is 1000 incidents per month.

Introducing CT and Th in (10), the value 0.4 for the coefficient α (variability) follows. As shown in Figure 1 we can easily follow that a slightly increase of the throughput (after leaving the linear part of the graph) considerably increase the cycle time. In order to avoid the flooding of the departments with tickets, the natural reaction of the employees is to reduce the raw process time and consequently reduce the QoS of the department. Hence, in our example, if the throughput exceeds 800 incidents per month appropriate measures should be taken in order to avoid the collapse of the service. On the contrary, if the throughput is equal to 400 tickets per month (being on the linear part of the graph), a part of the staff can be relocated to assist other services. The relation (6) shows the correlation between the flow factor of the individual departments and the data center and can be used to balance the individual departments.

In order to establish normalized / composite metrics, we consider those presented in Subsection III-A.

We describe below some of the metrics used in incident management, normalized and directed as described in Subsection III-B, i.e., each metric takes values in the closed interval $[0; 1]$ and a greater value for the metric implies a better accomplishment of the business requirements.

$$m_{01} := \frac{1}{\text{“Total No. of incidents”}}$$

$$m_{02} := 1 - \frac{\text{“No. of repeated incidents”}}{\text{“Total No. of incidents”}}$$

$$m_{03} := 1 - \frac{\text{“No. of repeated incidents with known solution”}}{\text{“No. of repeated incidents”}}$$

Unfortunately, the “Maximum No. of incidents” is not a priori known. Hence, generally speaking, it cannot be used in the formula. Furthermore, the business requires that corresponding measures are taken, such that repeated incidents are avoided. Therefore, “No. of repeated incidents” should be kept low. Further metrics, which are considered (SLA refers to Service Level Agreement):

$$m_{04} := \frac{\text{“No. of escalated incidents”}}{\text{Total No. of incidents”}}$$

$$m_{05} := \frac{1}{\text{“Average cycle time to resolve the incident”}}$$

$$m_{06} := \frac{1}{\text{“Average waiting time from user side”}}$$
$$m_{07} := \frac{1}{\text{“Average working time on the incident”}}$$

$$m_{08} := \frac{\text{“Total No. of incidents resolved within SLA time”}}{\text{“Total No. of SLA relevant incidents”}}$$
$$m_{09} := \frac{1}{\text{“First reaction time to repair the incident”}}$$
$$m_{10} := \frac{1}{\text{“No. of responses from service center side”}}$$

Unfortunately, defining metrics fulfilling the conditions as above, is not always straightforward. Let us consider $Incid_{out}$ as the total number of incidents closed and $Incid_{in}$ as the total number of incidents opened in the time frame considered. In order to avoid the flooding of the data center with incidents the metric $k := Incid_{out}/Incid_{in}$ could be tracked. Unfortunately, this metric does not fulfill our requirements, since it can take values outside the interval $[0; 1]$. In order to avoid this impediment, we define $k_{in} := Incid_{in}/$“Total No. of incidents” and $k_{out} := Incid_{out}/$“Total No. of incidents” and set

$$m_{11} := \frac{1 + k_{out} - k_{in}}{2}.$$

Then, m_{11} is normalized and satisfies the above conditions imposed for metrics. Generally speaking, the effective minimal and maximal value of a metric is not known, nor is the distribution a priori known. Thus, for example, a metric m_1 takes values in the interval $[0.5; 0.6]$ and another metric m_2 in the interval $[0.2; 0.7]$ with nearly uniform distribution. Hence, the metric m_2 varies more widely than m_1 and this should be considered – for example using the standard deviation – when setting up the groupings for the composition of the metrics, such that metrics having a low standard deviation should be assigned to more important groups.

Now, let us consider in our use case five composition groups, $G_0, G_1, \ldots, G_4$, such that G_1 is the most relevant group. Let us associate the weight w_i to group G_i and let us consider the weighting according to an exponential function, such that $k_0 := 1$ and $k_i := w_{i-1}/w_i$ for $i > 0$. This yields to the relation: $w_i = w_0/\prod_{l=0}^{l=i} k_l$. In our example $k_1 := e^1$, $k_2 := e^{0.75}$, $k_3 := e^{0.5}$, etc. The values for k_i are illustrated in Figure 2. Let us assign the above metrics to the composition groups, such that index set of the metrics assigned to the group G_l is equal to I_l and let n_l the number of metrics in the group G_l. Then, according to the composition rules: $\sum_{i=0}^{4} n_i \cdot w_i = 1$. Hence, the weight values follow. The value of the composition metric M is:

$$V(M) := \sum_{i=0}^{4} w_i \cdot \sum_{l \in I_i} V(m_l).$$

Examples of services at the ZIH of the Technische Universität Dresden (TU Dresden) are: “E-Mail Service”, “Backup and Archive Service”, “Data Exchange Service”, “Access to High Performance Computing Resources”, etc. [23] We conclude this section by presenting the assistance system for a data center in Figure 4 and by summarizing the composition strategy via the flow diagram given in Figure 3.

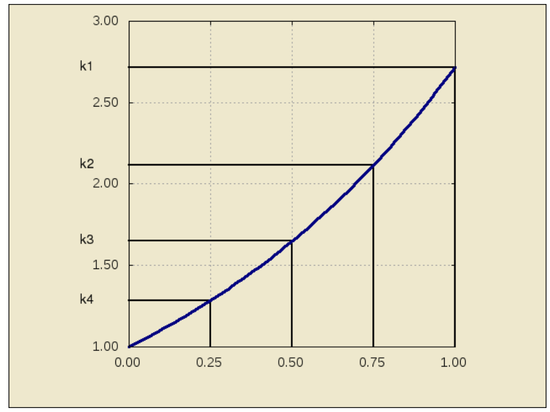

Figure 2. Graphical construction of values k_i such that metrics are weighted according to an exponential function, depicted for $f(x) = e^x$.

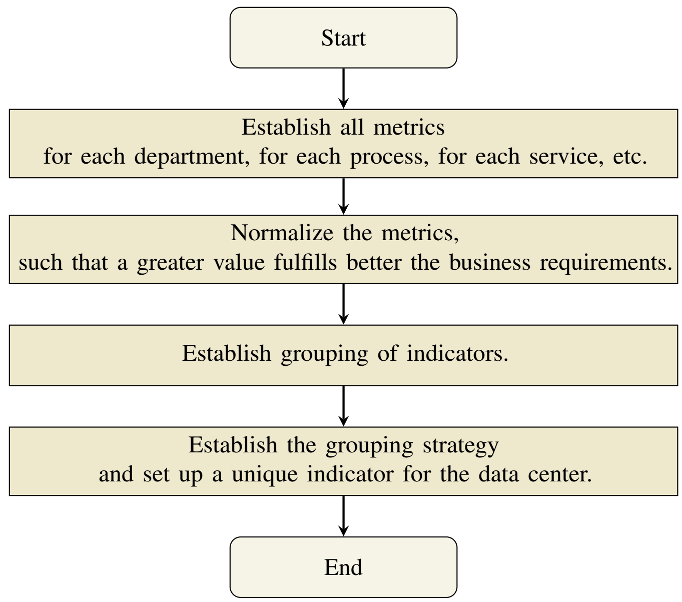

Figure 3. Simplified flow diagram regarding the composition strategy.

VI. CONCLUSION AND FUTURE WORK

We set up a formal, mathematical model and analyzed the QoS and the modalities to enhance it within this model. In that way, the QoS provided by the TU Dresden can be improved, which implicitly leads to a good ranking of the TU Dresden between the universities in Germany and world wide.

The main result of our research is that from a service provider perspective QoS can be characterized in mathematical terms pretty accurately, such that the improvement / degradation of the overall service or part of it can be tracked in IT systems and can be visualized through GUIs. According to Definition of ITU-T Rec. E.800 [24]: QoS is the “Collective effect of service performances which determine the degree of satisfaction of a user of a service”. The long term experience of the first author, working in the semiconductor industry is very similar to the definition above, such that it does not suffice to consider only the service provider perspective, but the user’s perception of the QoS should also be considered. Further research is necessary to establish the correlation (or lack of it) between the objectively

 ISBN: 978-1-61208-668-2

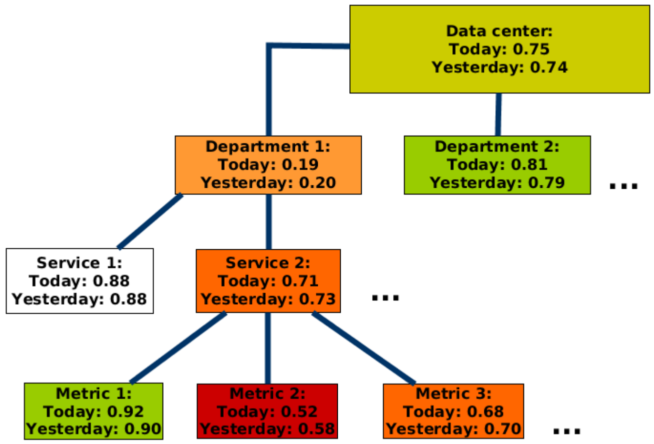

Figure 4. Hierarchical assistance system for a data center consisting of different departments, services, and metrics as levels. Changes will be depicted in green, white or red depending on the indicator's values of today and yesterday.

improved service and the subjective perception of the customer.

The composition strategy of various metrics to form an overall indicator can be a very complex endeavor. If all the metrics improve or degrade, then the overall indicator will improve or degrade accordingly. The question is in which direction will the overall QoS indicator swing if some metrics improve, some other degrade in time. We are not aware of any research in this direction. Similarly, how can the overall QoS be enhanced within the limited budget by improving some components and degrading others by resource reallocation.

The study has been accomplished for the data center of the ZIH, TU Dresden, but it can be used to improve the QoS by any service provider in the event that the real world can be mapped to the formal model used in this approach.

The similitude between a data center and a semiconductor fab regarding performance improvement cannot be denied. It would be then advantageous to identify the major differences, such that the theory developed to improve the performance of a semiconductor fab could be adapted for data centers. This work is a little step in this direction.

Acknowledgment

Part of this paper (including the study regarding Little's Theorem) completes an unfinished study of the first author regarding the performance of semiconductor fabs within the scope of the Cool Silicon Project (2012 - 2014). Part of this work was also supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) by funding the research project "EMuDig 4.0". Furthermore, we acknowledge the assistance and helpful comments of the anonymous referees.

References

[1] Symantec Corporation, "State of the data center survey – global results," September 2012, retrieved: September 2018. [Online]. Available: http://www.symantec.com/content/en/us/about/media/pdfs/b-state-of-data-center-survey-global-results-09_2012.en-us.pdf

[2] G. I. Butnaru, "The quality of services in tourism and in the romanian accommodation system," Analele Stiintifice ale Universitatii "Alexandru Ioan Cuza" din Iasi - Stiinte Economice, vol. 56, 2009, pp. 252–269. [Online]. Available: https://EconPapers.repec.org/RePEc:aic:journl:y:2009:v:56:p:252-269

[3] S. Jain and G. Gupta, "Measuring Service Quality: Servqual vs. Servperf Scales," vol. 29, 04 2004, pp. 25–38.

[4] C. Ennew, G. V. Reed, and M. Binks, Importance-Performance Analysis and the Measurement of Service Quality, 03 1993, vol. 27, pp. 59–70.

[5] B. Edvardsson, "Service Quality: Beyond Cognitive Assessment," vol. 15, 04 2005, pp. 127–131.

[6] A. P. Parasuraman, V. Zeithaml, and L. Berry, A Conceptual Model of Service Quality and its Implication for Future Research (SERVQUAL), 01 1985, vol. 49.

[7] K. Ishibashi, Maintaining Quality of Service Based on ITIL-Based IT Service Management, 08 2007, vol. 43, pp. 334–344.

[8] E. L. Hernandez, "Evaluation Framework for Quality of Service in Web Services: implementation in a pervasive environment," Master's thesis, INSA Lyon, France, 2010.

[9] itSMF UK, ITIL Foundation Handbook, 3rd ed. Norwich: The Stationery Office, 2012.

[10] ServiceNow, "Key Performance Indicators (KPI) Examples, Dashboard & Reporting," 2018, retrieved: September 2018. [Online]. Available: http://kpilibrary.com/

[11] D. Parmenter, Key Performance Indicators: Developing, Implementing, and Using Winning KPIs, ser. BusinessPro collection. Wiley, 2015.

[12] L. Turpin, "A note on understanding cycle time," International Journal of Production Economics, vol. 205, 2018, pp. 113 – 117. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925527318303748

[13] K. Sigman, "Notes on Little's Law," 2009, retrieved: September 2018. [Online]. Available: http://www.columbia.edu/~ks20/stochastic-I/stochastic-I-LL.pdf

[14] M. El-Taha and S. Stidham Jr, Sample-Path Analysis of Queueing Systems. Kluwer Academic Publishers, 01 1999, vol. 11.

[15] J. D. C. Little, "A Proof for the Queuing Formula $L = \lambda W$," Oper. Res., vol. 9, no. 3, Jun. 1961, pp. 383–387. [Online]. Available: http://dx.doi.org/10.1287/opre.9.3.383

[16] J. Shortle et al., Fundamentals of Queueing Theory, 5th ed., ser. Wiley Series in Probability and Statistics. Wiley, 2018.

[17] K. Hilsenbeck, "Optimierungsmodelle in der Halbleiterproduktionstechnik," Ph.D. dissertation, Technische Universität München, 2005, retrieved: September 2018. [Online]. Available: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss20050808-1721087898

[18] M. Holweg, J. Davies, and A. D. Meyer, Process Theory: The Principles of Operations Management, ser. BusinessPro collection. Oxford Univ. Press, 2018.

[19] W. Hansch and T. Kubot, "Factory Dynamics Chapter 7," retrieved: September 2018. [Online]. Available: http://fac.ksu.edu.sa/sites/default/files/Factory%20Dynamics.pdf

[20] S. S. Aurand and P. J. Miller, "The operating curve: a method to measure and benchmark manufacturing line productivity," in 1997 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop ASMC 97 Proceedings, Sep 1997, pp. 391–397.

[21] W. J. Hopp and M. L. Spearman, Factory Physics: Foundations of Manufacturing Management, Burr Ridge, IL, 2nd ed. Irwin/McGraw-Hill, 2001.

[22] D. Baur, W. Nagel, and O. Berger, "Systematics and Key Performance Indicators to Control a GaAs Volume Production," 2012, retrieved: September 2018. [Online]. Available: http://www.csmantech.org/Digests/2001/PDF/12_3_Berger.pdf

[23] TU Dresden, ZIH, "TU Dresden, ZIH, IT services," September 2018, retrieved: September 2018. [Online]. Available: https://tu-dresden.de/zih/dienste

[24] International Telecommunication Union/ITU Telecommunication Sector, "Standard ITU-T E.440: Terms and Definitions Related to Quality of Service and Network Performance Including Dependability – Telephone Network and ISDN Quality of Service, Network Management and Traffic," 1996, retrieved: September 2018. [Online]. Available: https://standards.globalspec.com/std/704295/itu-t-e-440/

