IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 August 2023, accepted 6 September 2023, date of publication 11 September 2023, date of current version 21 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3314201

RESEARCH ARTICLE

Comparing the Scalability of Communication Networks and Systems

TOBIAS HOSSFELD[1], **(Senior Member, IEEE), POUL E. HEEGAARD**[2], **(Senior Member, IEEE), AND WOLFGANG KELLERER**[3], **(Senior Member, IEEE)**
[1]Chair of Communication Networks, University of Würzburg, 97074 Würzburg, Germany
[2]Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), 7034 Trondheim, Norway
[3]Chair of Communication Networks, TU Munich, 80333 Munich, Germany

Corresponding author: Tobias Hossfeld (tobias.hossfeld@uni-wuerzburg.de)

This work was supported by the German Research Foundation [Deutsche Forschungsgemeinschaft (DFG)] under Grant 316878574 (DFG SDN-App 2).

ABSTRACT Scalability is often mentioned in literature, but a stringent definition is missing. In particular, there is no general scalability assessment which clearly indicates whether a system scales or not or whether a system scales better than another. The key contribution of this article is the definition of a scalability index (SI) which quantifies if a system scales in comparison to another system, a hypothetical system, e.g., linear system, or the theoretically optimal system. The suggested SI generalizes different metrics from literature, which are specialized cases of our SI. The primary target of our scalability framework is, however, benchmarking of two systems, which does not require any reference system. The SI is demonstrated and evaluated for different use cases, that are (1) the performance of an IoT load balancer depending on the system load, (2) the availability of a communication system depending on the size and structure of the network, (3) scalability comparison of different location selection mechanisms in fog computing with respect to delays and energy consumption; (4) comparison of time-sensitive networking (TSN) mechanisms in terms of efficiency and utilization. Finally, we discuss how to use and how not to use the SI and give recommendations and guidelines in practice. To the best of our knowledge, this is the first work which provides a general SI for the comparison and benchmarking of systems, which is the primary target of our scalability analysis.

INDEX TERMS Communication networks, performance, availability, scalability.

I. INTRODUCTION

The evaluation of systems focuses on different aspects: performance, efficiency, elasticity, flexibility, and scalability. Especially, scalability is often used in literature with statements like *"The system scales well."* or *"Our approach scales better than previous ones."* However, such statements are imprecise and do not give meaningful insights. To this end, a stringent definition of scalability is provided which allows quantifying scalability and to compare the scalability of communication networks and systems.

In the context of software engineering and cloud computing, there are several definitions of scalability, e.g., [1], and [2]. The closest work to ours is the definition of scalability metrics in [1] for cloud computing. They define the quality scalability metric of the system as follows: For a system, the target measure of interest $f(x)$ is measured depending on a certain parameter. In the case of cloud computing [1], the target measure is, e.g., the average service response time and the parameter is the demand level x. The obtained area F under the average service response time function is then compared to the area H of an ideal system function $h(x)$ (i.e., ideal service response time) depending on the demand. The quality scalability metric is the ratio of the two areas under the curve, i.e. H/F, which gives a value between 0 and 1. This is visualized in Figure 1.

In the realm of communication networks and systems, scalability is frequently acknowledged but lacks a precise definition, analysis, and quantification in existing research and literature. We generalize the definition in [1] by considering:

The associate editor coordinating the review of this manuscript and approving it for publication was Cesar Vargas-Rosales.

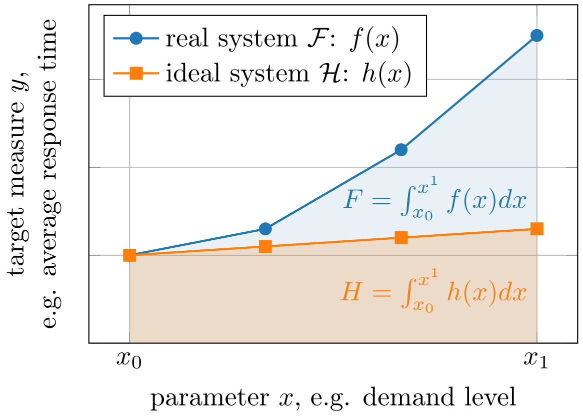


FIGURE 1. Quality scalability metric as ratio H/F of the areas under the target measure curves in the context of cloud computing according to [1]. The real system $\mathcal{F}$ is related to an ideal system $\mathcal{H}$.

1) A system function $f(x)$ which quantifies *an arbitrary target measure of interest*, e.g., average response time, e.g., 95% quantile of response time, e.g., packet loss ratio.
2) An *arbitrary reference system* (with target reference function $h(x)$), which *may* be the optimal system behavior. However, the optimal system may be unknown in practice, and we may want to consider if the system, e.g., scales linearly. Then this can be done with a proper reference function.
3) Instead of a parameter range $[x_0; x_1]$ to be considered in the scalability analysis, we focus on a *weighted parameter range* $w(x)$ which allows defining the importance of some parameter settings in the scalability analysis. Or we may also exclude some parameter settings in the analysis.

Our definition is more general, and several definitions of scalability metrics in literature are a special case of ours. The closest definition to ours is [1] which considers the optimal reference system; however, the optimal system behavior may be unknown for real-world communication networks and systems. Furthermore, the arbitrary target measure needs to be adapted to communication networks and systems, e.g., considering service-level agreements as we demonstrate in our use cases. Our introduced weighted parameter range is thereby of utmost importance to appropriately include the network and system configurations or parameter values of interest. The weighted parameter range may also consider costs, see the availability use case later. Thus, all ingredients are generalized to the needs of communication networks.

A. CONTRIBUTION

The key contribution of this paper is a general framework to quantify *whether* a system (or communication network) is scaling in comparison to a reference system. This also allows comparing two systems and to rank them, i.e., which system is scaling better. We additionally provide an overview of related concepts (performance, efficiency, elasticity, flexibility) and how they differ from scalability. This conceptual difference is important, since related work is partly mixing terms and not providing measures for scalability. To this end, we conduct an in-depth literature study and show the differences of existing measures to our framework. We show that our scalability index generalizes approaches from the literature by the introduction of arbitrary reference systems and target measures, as well as the introduction of a weighted parameter range. Finally, we demonstrate the usage of the scalability framework for different use cases: IoT load balancer, availability in communication systems, location selection in fog computing, comparison of time-sensitive networking (TSN) mechanisms. Those use cases show different aspects which need to be considered in a scalability analysis and are summarized as lessons learned. We discuss practical guidelines on how to use the scalability index, especially regarding the definition of reference systems, target functions, weighted parameter ranges. Our contributions in a nutshell:

- scalability framework generalizing existing approaches;
- overview of related concepts: performance, efficiency, elasticity, flexibility, scalability;
- detailed analysis of related work wrt. scalability definitions: identification of misleading usage of the term ‘scalability’ due to missing scalability definition;
- complementary use cases which demonstrate how to use the scalability index in practice and which indicate the need for the suggested generalization.

B. ORGANIZATION

The remainder of this paper is structured as follows. Related work is revisited in Section II to get an overview of existing scalability definitions and to differentiate it from aspects like performance, efficiency, elasticity. The literature study serves as the basis for our definition of a scalability index (SI) in Section III that generalizes the existing approaches. To demonstrate the SI, different use cases are analyzed in Section V: (1) scalability of an IoT load balancer depending on system load, which is modeled with queueing theory; (2) availability of a communication system depending on the number of nodes and system structure, which is modeled by probability theory; (3) scalability comparison of different location selection mechanisms in fog computing with respect to delays and energy consumption based on existing experimental results; (4) comparison of time-sensitive networking (TSN) mechanisms in terms of the number of deployed streams while guaranteeing upper delay bounds based on measurement results, which are investigating an unequally spaced parameter range (number of requested stream). The intention of those use cases is to demonstrate how to compare systems, the relevance of the target parameter under investigation, how to cope with positive (e.g., availability) and negative target functions (e.g., waiting times), the impact of the parameter range under investigation, how the target measure influences the scalability result. This will be summarized in Section VI which provides additional discussions and recommendations for the practical usage of the SI and lessons learned from the use cases.

II. EXISTING FRAMEWORKS AND METRICS ON SCALABILITY

In the context of cloud computing, several definitions of scalability are provided, which are revisited and summarized below. This summary will show that our proposed definition of scalability is more general and abstracts the existing definitions. However, before that, we want to differentiate several terms related to scalability to clarify the different scopes.

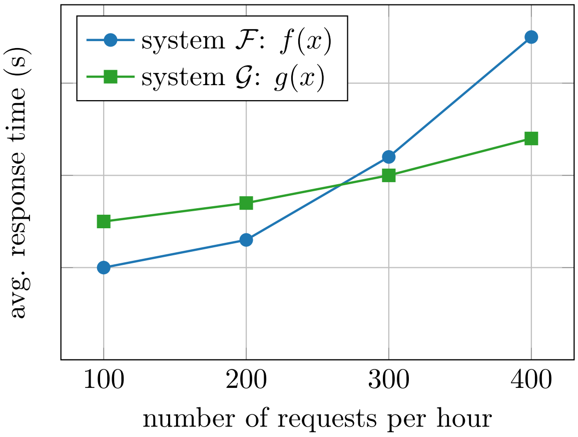


FIGURE 2. Performance curves of two systems $\mathcal{F}$ and $\mathcal{G}$.

A. DIFFERENTIATION: PERFORMANCE, EFFICIENCY, ELASTICITY, FLEXIBILITY, SCALABILITY

Figure 2 shows the *performance* of two different systems $\mathcal{F}$ and $\mathcal{G}$. In the example, the average response time is plotted depending on the number of requests per hour to be served by the system. Thereby, the system is considered for a given request rate over a longer time, i.e., under quasi steady-state assumptions[1] of the system. It can be seen that the system $\mathcal{F}$ has a better performance for 100 and 200 requests per hour than $\mathcal{G}$. However, $\mathcal{G}$ outperforms $\mathcal{F}$ for 300 or more requests per hour. From Figure 2, it is unclear which system scales better, but it seems that $\mathcal{G}$ has better scalability properties. For such statements, a proper definition of scalability is required.

Efficiency relates to the costs (or consumption of resources in general) required to complete a request or a given amount of work in a system. For example, energy efficiency would be the ratio of the number of completed requests in a system compared to the maximum number of requests which could have been completed in an ideal system with the same amount of energy [3]. As for the quantification of performance, the system is considered for a particular parameter setting over longer time, e.g., 100 requests per hour. Of course, the request arrivals are a stochastic process, but a quasi stationary system is considered where the request arrival rate over longer time is quasi constant. Other measures of efficiency in the context of distributed systems consider the work rate per processor, while "Scalability means not just the ability to operate, but to operate efficiently and with adequate quality of service, over the given range of configurations" [4].

In contrast, *elasticity* considers the dynamic changes of a system and quantifies the ability of the system to adapt itself during shorter time scales. Elasticity is defined as "the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible" [5]. The dynamic adaptation of capacity, e.g., by altering the use of computing resources, to meet a varying workload is called *elastic computing* [6]. In communication networks, elasticity means that the network adapts its operation and reallocates or redistributes resources (resource supply) according to temporal and spatial traffic fluctuations and service demands (resource demand). This may include computational and communications resources, e.g., for the management of computational resources in softwarized and virtualized networks, e.g., in 5G systems [7]. Figure 3 illustrates the elasticity of a system over time by comparing the resource demand and the resource supply of that system. In contrast, efficiency would relate the resource supply and the resulting costs to the resource demand. Mathematical definitions for elasticity are provided in [5].

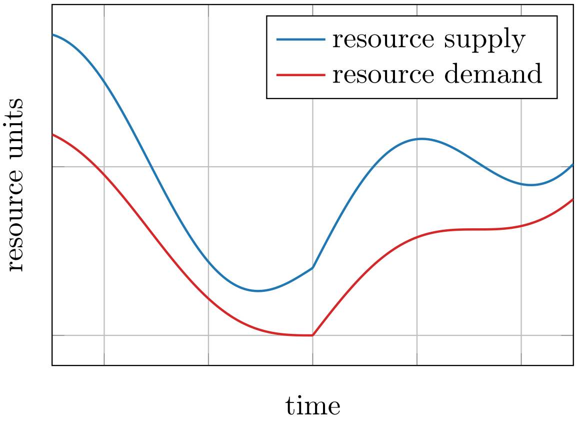


FIGURE 3. Elasticity of a system relating resource supply and resource demand [5].

It is worth noting that the time dynamics of the resource demand in Figure 3 are summarized as a single parameter setting x in the performance curve in Figure 2. E.g., the average resource demand, expressed as number of requests per hour, is the considered parameter in the quasi stationary system. The performance of that system is then characterized in that quasi steady state, e.g., by considering the average response time. Thus, the results from Figure 3 are a single point in Figure 2, as visualized in Figure 4.

[1] The system conditions are varying slowly enough such that the system acts over a longer period of time as in equilibrium.

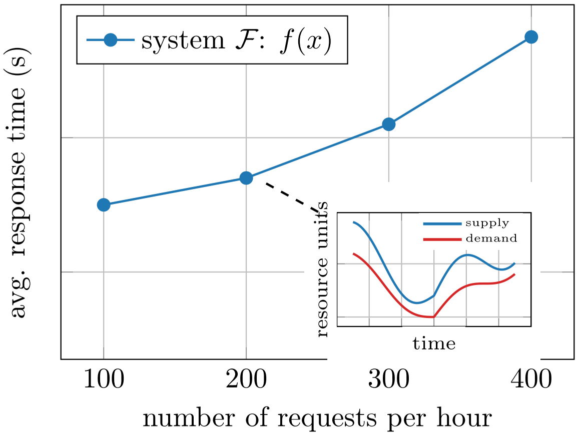


FIGURE 4. Performance and elasticity of a system are considered on longer and shorter time scales.

Finally, *flexibility* is a key property of systems that are surveyed for communication networks in [8]. A model for measuring network flexibility is proposed in [9] which quantifies network flexibility as the achievable subset of the set of all possible demand changes. It is defined in [9] as: "Given the demands the communication network has to respond, network flexibility is the ability of the network to adapt its state to satisfy the new demands promptly and with little effort." In contrast to elasticity, flexibility considers all possible demand changes in the quantification.

B. DEFINITIONS OF SCALABILITY IN LITERATURE

In general, *scalability is seen as the ability of the system to sustain increasing workloads of a quasi stationary system*, i.e., on a longer timescale, e.g., by making use of additional resources, e.g., by adapting its configuration, e.g., by adapting or reducing QoS. In the context of cloud computing, scalability is typically seen as the ability of the cloud-based system to increase the capacity of the software service delivery by expanding the quantity of the software service that is provided when such increase is required by increased demand for the service over a longer period of time [1], [2]. In contrast, short-term flexible provision of the resources is captured by elasticity of the service provision which means scaling up or down at a specific time; hence, elasticity is the measurement of the instantaneous behavior of the service in response to changes in service demand [1], as depicted in Figure 3. Scalability is scaling up by adding resources in the context of a given time frame and considers the behavior of the service over a (longer) period of time. Thus, scalability does not aim at quantifying how fast, how often, and at what granularity scaling actions can be performed [5], but considers a longer period of time. However, scalability needs to consider the system behavior (e.g., performance or any other desired target function) of all different and relevant demand scenarios. Hence, Figure 2 is the required general input for a scalability measure. We follow this understanding of scalability and provide a general framework to quantify a scalability index and show how to compare the scalability of systems.

For various use cases and scenarios, particular definitions of scalability are provided. For software defined networks, scalability issues are arising due to logically centralized control planes, which may need to be physically distributed. Such scalability issues as well as concrete solution approaches are discussed, e.g., in a special issue [10]. However, still, a concrete scalability metric is required which may be utilized for quantifying such solution approaches. To this end, [11] defines a *scalability metric for control planes in software defined networks*. Thereby, the authors propose to use another target function beyond throughput and latency to address control planes properly. In particular, they consider for a network the ratio of workload over overhead. Workload is thereby quantified as the number of flows entering the network through the data plane. Overhead is quantified as the number of messages processed in the control plane. However, this definition of scalability lacks the consideration of changed demands and is more related to efficiency. To fit this into a general scalability framework, the ratio of workload over overhead may be considered as a target function depending on the system load. The resulting curves which are provided in [11] are then similar to Figure 1 and may serve as input for a scalability index. In general, literature quite often considers such performance curves (e.g., Figure 2) and authors qualitatively argue that a system is scaling without properly defining what this really means.

For communication networks, coping with dynamicity, heterogeneity of demands, diversity of communication mechanisms, and the scale leads to significant challenges, while stringent and dynamic quality requirements need to be fulfilled. Such advancements in the field were considered in [12] focusing on adaptive and scalable communication networks. One of the core concepts is *transitions*, aiming at increasing the flexibility and scale at which communication networks can be adapted. Thereby, possibilities to change existing protocols, technologies, or their configuration while the network is in operation are realized by transitions. Still, for benchmarking different implementations of transitions, a framework for comparing the scalability of different transition solutions is missing. Complementing the existing flexibility framework [9], our scalability framework fulfills a need in literature and provides a novel approach which can be applied in the domain of communication networks.

The *scalability of big data processing systems in clouds* is investigated in [13]. They consider some performance curves depending on load and consider linear scalability, sublinear scalability and super-linear scalability as defined in previous work [14]. Thereby, scalability considers a reference system, which is a theoretical system with a linear relation between the target measure and the parameter under investigation. This will be a relevant scenario in practice, since scalability is often interpreted as comparison to linear relations. However, a precise definition is also needed here, since the constant offset of a linear function must be considered, see Section III-C. [13] also defines a higher level target measure of the system performance for a dedicated workload. This target measure is referred to as 'scalability' (which is misleading and wrong), but it does not provide a single measure quantifying how the system behaves for different workloads.

Reference [15] coined the term *'stochastic scalability'* and considered as an example P2P-based information sharing platform. As target function of their analysis, they consider the 99%-quantiles of search delays in such a platform depending on the number of customers of the network and the amount of information to be stored in it (parameter under consideration). This target function goes beyond typical works, since quantiles instead of averages are considered, which may be relevant for business. Reference [15] uses the term 'functional scalability' to analyze whether the functionality of a system (quantified by the target function) also works for many customers (parameter under investigation). Reference [15] advances the scalability analysis and defines: "Stochastic scalability, on the other hand, tries to verify whether a system can sustain the stochastic behavior of its components." To be more precise, the influence of the coefficient of variation of the interarrival times of service requests is considered as another parameter on the performance. Hence, stochastic scalability changes the parameter under investigation (here: coefficient of variation of interarrival times). In our general framework, the target function and the stochastic characterization of the system load as parameter under consideration are well reflected. Hence, our framework also includes stochastic scalability.

A metric to predict software scalability is suggested in [16]. However, the authors simply define another target function, which is referred to as Performance Non-Scalability Likelihood (PNL). For a given load in a system, the PNL metric reflects the probability that the system's performance objective will not be met. Hence, again, the authors do not define a scalability index considering various load situations and the system's performance. The PNL can be used in our framework as a target function.

Other definitions of 'scalability' consider the ratio of efficiency for two different load scenarios [4], although this is more related to speedup and does not generalize scalability for various load scenarios and in comparison to arbitrary reference systems. Similarly, *isospeed scalability* relates the workload capacity of the system at two different scales [14]. The initial and scaled problem size (workload) and the resources (number of processors) are considered. The target function is the ratio of workload per processor. Isospeed scalability quantifies then the ratio of the target functions for the initial and scaled problem size. The *Iso-efficiency scalability* [17] is described as the "ability of parallel machines to keep the parallel efficiency constant when the system and problem size increase" [11]. This is advanced to *H-isoefficiency* for heterogeneous systems [18]. Similar critics as above are observed, since those quantities do not reflect scalability as intended.

In the software engineering domain, it is often differentiated between *resource scalability* and *demand scalability*, see e.g., [19]. A resource demand metric indicates the resource demand depending on the actual load. It is analyzed whether the resource demand increases linearly, sub-linearly, or super-linearly (i.e., using an appropriate reference function). This analysis may also consider whether a system only scales up to a certain point of load, while additional load cannot be handled even though further resources are added. This is also referred to a *strong scaling* in software engineering. The load capacity metric indicates how processing capabilities increase with increasing resources. Similarly, a certain point of capacity may be reached when with additional resources do not increase the processing capabilities anymore. The additional resources may even decrease the capabilities due to signaling and coordination overhead. This is also referred to as *weak scaling* in software engineering. Both aspects are included in our scalability framework by using appropriate parameters under investigation and target functions.

III. GENERAL FRAMEWORK FOR QUANTIFYING AND COMPARING SCALABILITY

From the definitions and understanding of scalability in literature, we provide now a general framework to quantify scalability in terms of a scalability index, which allows comparing the scalability of different systems. To this end, exact definitions of scalability and a scalability index (SI) are proposed. The ingredients of the scalability index are discussed and the fundamental characteristics of the SI are analyzed.

A. DEFINITION OF SCALABILITY INDEX

A tempting definition of scalability is seen as the ability of the system to sustain increasing workloads of a quasi stationary system. However, this definition is not precise enough, since it is unclear what "sustaining" really means here. It is not specified what is the target measure of interest, what is the parameter under consideration, and how to evaluate if a system "sustains" increasing workloads. Furthermore, workload is only one specific parameter (probably the most important one in practice), but also the size of networks (e.g., IoT mesh networks), or stochastic variations (i.e., stochastic scalability) may be of interest in a scalability analysis. Another critical point is that 'sustaining' means that we need a (theoretical) reference system for comparison. This has also been suggested in literature for specific theoretical systems: linear, sublinear, super-linear scaling and target functions [13], [14] or the comparison to the optimal/ideal reference system [1].

Furthermore, the analysis of scalability needs to consider all relevant parameter settings. By weighting the importance or relevance of a parameter setting, the target measure then needs to accumulate the weighted target measure. As a consequence, the quantification of scalability results in an integral measurement of the (weighted) target measure over the entire parameter range. If the system may not work properly above a certain load, then this needs to be captured in the target function, while the relevance or importance of such a scenario can be adjusted with the weighting function.

Then, the ingredients of scalability and its quantification are as follows:

1) A system function $f(x)$ quantifies *an arbitrary target measure of interest* for the system $\mathcal{F}$, see Figure 1.
2) An *arbitrary reference system* $\mathcal{H}$ with a target reference function $h(x)$, like the ideal system behavior in Figure 1, is used for comparison. In practice, the optimal system may be unknown. Or we may want to investigate if the system scales linearly, i.e., the reference function is a linear function.
3) *Integral measurements* F and H of the target measure for the system and the reference consider the parameter range under investigation. This is simply the area under the corresponding target function curve in Figure 1.
4) A *weighted parameter range* $w(x)$ allows defining the importance of some parameter settings as weighted integral. Thereby, we may also exclude some parameter settings in the analysis.

We propose the following definition of the term 'scalability'.

Definition 1 (Scalability): The ability of a system to perform as well as a reference system regarding a target measure within a defined weighted parameter range.

The quantification of scalability is then the weighted integral measurement F of the system function using the desired target measure of interest in relation to the weighted integral measurement H of a reference system over a defined parameter range $\mathbb{X}$. The parameter range is potentially weighted $w(x)$ according to the relevance/importance of a parameter setting in the scalability analysis.

$$F = \int_{x\in\mathbb{X}} w(x)\cdot f(x)\,dx \qquad \text{(integral system meas.)}$$

$$H = \int_{x\in\mathbb{X}} w(x)\cdot h(x)\,dx \qquad \text{(integral reference meas.)}$$

Definition 2 (Scalability Index): Quantification of the scalability of a system $\mathcal{F}$ with respect to a reference system $\mathcal{H}$ as the ratio of the integral measurements F and H.

$$SI = H/F \text{or } SI = F/H \qquad \text{(scalability index)}$$

Depending on the target measure, we may use the ratio H/F or the inverse ratio F/H. In the case of a target reference function, for which an increase means increasing 'badness', e.g., average response time, then the ratio H/F is used. If H is the optimal reference system, then SI is normalized between 0 and 1. Note that in other reference systems, e.g., a linear system, the SI may also achieve values larger than 1. In the case of a target reference function, for which an increase means increasing 'goodness', e.g., throughput or availability, then the ratio F/H is used, see Table 1. We will discuss this further for the two use cases in the later sections.

TABLE 1. Concrete definition of the scalability index as ratio of the integral measurements *F* and *H* depending on goodness/badness of an increase in the target measure and parameter, respectively, with some (examples) given in brackets.

Parameter increase indicating	*Target measure's increase indicating system's increase of* badness $\gamma=-1$ (e.g. response time)	 goodness $\gamma=1$ (e.g. throughput)
badness (load)	H/F (response vs. load)	F/H (throughput vs. load)
goodness (#servers)	H/F (response vs. #servers)	F/H (throughput vs. #servers)
any parameter	H/F (response time)	F/H (throughput)

To have a general definition of the SI, we may use the auxiliary variable γ with $\gamma = 1$ indicating 'goodness' and $\gamma = -1$ indicating 'badness' of the target measure.

Definition 3 (General Scalability Index): Quantification of the scalability of a system $\mathcal{F}$ with respect to a reference system $\mathcal{H}$ as the ratio of the integral measurements F and H and the goodness indicator γ.

$$SI = (F/H)^{\gamma} \qquad \text{(general scalability index)}$$

With the definition of the SI, we can test if a system is scaling. To be more precise, we need to provide the reference target function $h(x)$ and the weighting function $w(x)$.

Definition 4 (Testing Scalability): A system $\mathcal{F}$ is scaling with respect to a reference system $\mathcal{H}$, a well-defined target measure $f(x)$ and $h(x)$, respectively, a weighting function $w(x)$, and a parameter range $\mathbb{X}$ if the scalability index SI is less or equal to 1.

$$SI \leq 1 : \textit{System scales wrt. h(x) and w(x) for } x \in \mathbb{X}.$$

$$SI > 1 : \textit{System does not scale wrt. h(x), w(x), } x \in \mathbb{X}.$$

Thus, testing scalability simply means comparing the integral measurements F and H. Note that a system is never scaling in relation to the optimal system. But the quantification SI shows how close a system gets to an optimal one.

Our definition of scalability and the scalability index generalizes definitions of scalability metrics in the literature, which are a special case of ours. In particular, [1] uses an optimal reference system $\mathcal{H}$ with equal importance of all parameter settings, i.e. $w(x) = 1$. However, we may be interested in quantifying if a system scales linearly, which we discuss later.

B. COMPARING THE SCALABILITY OF SYSTEMS

With the definitions above, we can now compare the scalability of two different systems $\mathcal{F}$ and $\mathcal{G}$ wrt. a reference system $\mathcal{H}$, well-defined target measure and weighting function. The integral measurement is F and G and the corresponding scalability index is SI_F and SI_G, respectively.

$\mathcal{F}$ is scaling better than $\mathcal{G}$ wrt. a well-defined target measure and weighting function, if the scalability index SI_F is larger than the scalability index SI_G. For a target reference function indicating 'badness' (e.g., response times), this means:

$$SI_F = H/F > H/G = SI_G \implies F < G \qquad \text{(badness)}$$

if $\mathcal{F}$ is scaling better than $\mathcal{G}$. For a target reference function indicating 'goodness' (e.g., throughput), this means

$$SI_F = F/H > G/H = SI_G \implies F > G \qquad \text{(goodness)}$$

if $\mathcal{F}$ is scaling better than $\mathcal{G}$. Hence, there is no need for an additional reference system $\mathcal{H}$.

Definition 5 (Comparing Scalability of Systems): A system $\mathcal{F}$ scales better than a system $\mathcal{G}$ with respect to a well-defined target measure and a weighting function $w(x)$ defined for the parameter x if

$$(F/G)^\gamma > 1\,. \qquad \text{(scalability comparison: } \mathcal{F} \succ \mathcal{G})$$

System $\mathcal{G}$ is scaling better than $\mathcal{F}$ if

$$(G/F)^\gamma > 1\,. \qquad \text{(scalability comparison: } \mathcal{G} \succ \mathcal{F})$$

Hence, we have the same structure and definition for comparing a system to a reference system (Definition 4) or to any other system (Definition 5).

C. LINEAR SCALING

In practice, linear scaling is important and a good reference system for comparison. Statements like *"The system is scaling."* are not precise due to the missing reference system and weighting function. However, often a linear function is implicitly assumed while for comparison a parameter range $[x_0; x_1]$ is considered where all parameter settings are equally important ($w(x) = 1$ for $x \in \mathbb{X} = [x_0; x_1]$). Thus, the weighting function is $w(x) = \mathbb{1}_{\mathbb{X}}(x)$.

Nevertheless, it is important to clearly define the linear function. Consider a simple example of a square relationship between the request rate and the avg. response time in a system $\mathcal{F}$. The parameter range of interest is $1\,\text{s}^{-1}$ to $3\,\text{s}^{-1}$. The reference system $\mathcal{H}_1$ has a larger constant offset than another reference system $\mathcal{H}_2$, while the gradient of the reference function $h_1(x)$ is less than the gradient of $h_2(x)$. Depending on the reference system, our conclusion would be that the system is linearly scaling ($\mathcal{F} \succ \mathcal{H}_1$) or is not linearly scaling ($\mathcal{F} \prec \mathcal{H}_2$). This becomes even more obvious when using constant functions, cf. dashed lines in Figure 5.

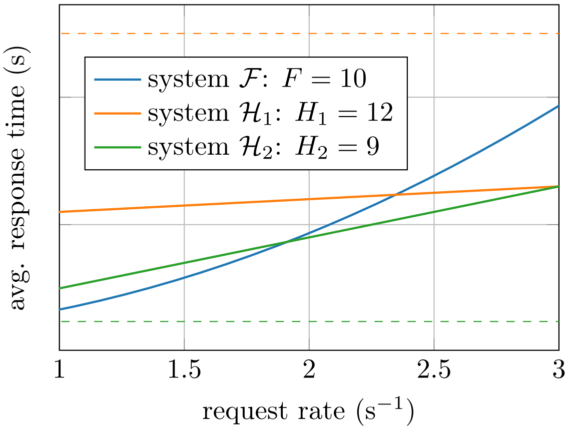


FIGURE 5. Linear scaling also needs to define a reference system and the linear reference target function. The scalability index is $SI_1 = 1.2$ and $SI_2 = 0.9$ wrt. system $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively.

D. FEATURES OF THE SCALABILITY INDEX

1) COMPARISON TO IDEAL SYSTEM

The scalability index is in the range $[0; 1]$ if a system is compared to an ideal system. Then, the SI shows how close the system gets to the scalability behavior of the perfect system.

2) CONSTANT REFERENCE TARGET MEASURE

The average $\bar{f}$ of the target measurement over the parameter range is $\bar{f} = F/(x_1 - x_0)$, which is identical to the scalability index $(F/H)^\gamma$ with a constant reference system $h(x) = 1$ and equal weights $w(x) = 1$. The integral measurement of the reference system is $H = \int_{x_0}^{x_1} w(x) \cdot h(x)\, dx = x_1 - x_0$.

3) CONSTANT WEIGHTS AND RELATION OF MEANS

Constant weights $w(x) = c$ have no influence on the SI.

$$SI = \left(\frac{\int_{x\in\mathbb{X}} f(x) \cdot c\, dx}{\int_{x\in\mathbb{X}} h(x) \cdot c\, dx} \right)^\gamma = \left(\frac{\int_{x\in\mathbb{X}} f(x)\, dx}{\int_{x\in\mathbb{X}} h(x)\, dx} \right)^\gamma \tag{1}$$

It may be useful to define weights, such that the integral measurements can be interpreted accordingly. For example, we consider the mean of the system function $f(x)$ and the reference function $h(x)$. The ratio of means corresponds to the SI with constant weighting function $w(x) = \frac{1}{x_1 - x_0}$.

$$\bar{f} = F = \int_{x\in\mathbb{X}} f(x) \frac{1}{x_1 - x_0}\, dx \tag{2}$$

$$\bar{h} = H = \int_{x\in\mathbb{X}} h(x) \frac{1}{x_1 - x_0}\, dx \tag{3}$$

$$SI = (F/H)^\gamma = (\bar{f}/\bar{h})^\gamma \tag{4}$$

It is tempting to compute the average function $\bar{f}$ by just computing the system function $f(\bar{x})$ of the average value of the parameter range $\bar{x} = \frac{x_1 + x_0}{2}$. This would require only the derivation of a single value of the system at the parameter $\bar{x}$ instead of deriving the entire parameter range. Especially, when measurements of the system $\mathcal{F}$ are conducted, this would save significant efforts.

However, Jensen's inequality, see for example [3], shows that the two quantities are different in general.

$$\bar{f} = \frac{1}{x_1 - x_0} \int_{x\in\mathbb{X}} f(x)\, dx \neq f(\bar{x}) = f(\frac{x_1 + x_0}{2}) \tag{5}$$

Only for linear systems, both quantities are identical. Hence, for testing linear scalability with $h(x) = mx + c$, the mean $\bar{h}$ of the function $h(x)$ and the function of the mean $\bar{x}$ are identical.

$$\bar{h} = \frac{1}{x_1 - x_0} \int_{x\in\mathbb{X}} h(x)\, dx = \frac{m}{2}(x_1 + x_0) + c \tag{6}$$

$$= h(\frac{x_1 + x_0}{2}) = h(\bar{x}) \tag{7}$$

4) STOCHASTIC PARAMETER RANGE AND EXPECTED TARGET MEASURE

A typical weight function $w(x)$ is the probability or probability density function of the parameter x for a discrete and continuous parameter range, respectively, see for example Section IV-C. Hence, the parameter range X is a random variable, described by $w(x)$. The integral measurement is then simply the expected value of the target function. The scalability index relates the expected system target measure $E[f(X)]$ to the expected reference target measure $E[h(X)]$.

$$SI = \left(\frac{\int_{x\in\mathbb{X}} f(x)\cdot w(x)\,dx}{\int_{x\in\mathbb{X}} h(x)\cdot w(x)\,dx}\right)^{\gamma} = \left(\frac{E[f(X)]}{E[h(X)]}\right)^{\gamma} \tag{8}$$

5) DIFFERENCE OF SYSTEM CURVES

The SI can also be interpreted as the relative difference between the integral measurements of the system function $f(x)$ and the reference function $h(x)$.

$$SI = (F/H)^{\gamma} = (H/H - H/H + F/H)^{\gamma} \tag{9}$$

$$= \left(1 - \frac{H-F}{H}\right)^{\gamma} \tag{10}$$

The area $H - F$ between the two curves is normalized by the area H. This relative difference indicates how far away the system $\mathcal{F}$ is from $\mathcal{H}$. The SI is then the difference between 1 and this relative difference.

6) MULTI-DIMENSIONAL PARAMETERS

Note that the integral measurement is defined for a parameter x which may also reflect a vector of different parameters, i.e. $x = (\xi_1, \ldots, \xi_n)$. The measurement integral is then a multiple integral over all parameter variables with a corresponding multidimensional weighting functions.

$$F = \int_{x\in\mathbb{X}} w(x)\cdot f(x)\,dx \qquad \text{(integral system meas.)}$$
$$= \int_{\xi_1\in\mathbb{X}_1}\cdots\int_{\xi_n\in\mathbb{X}_n} w(\xi_1,\cdots,\xi_n)\cdot f(\xi_1,\cdots,\xi_n)\,d\xi_1\cdots d\xi_n$$

The computation of the scalability index is not changed by considering multidimensional parameters.

7) SCALING IN NON-OPERATIONAL AREA

Consider a system which is not operating, e.g., due to overload. For example, a single server waiting queue cannot serve more requests than the service rate of the single processing unit. Thus, the arrival rate λ must be smaller than the service rate μ for a stable system. If this stability condition is violated, the quasi steady-state is not reachable. Then, the target measure like the waiting time is ∞. The corresponding integral measurement is then $F = \infty$ and the scalability index is $SI = H/F = 0$.

8) ATTRIBUTES OF SCALABILITY

Scalability is an integrative concept that encompasses the following basic attributes. Those attributes are considered in our definition of the scalability index accordingly.

- The target measure defines the major interest and scope of the scalability analysis, e.g., performance of the system, e.g., availability of the system, and if the system scales with respect to that particular target measure. Considering several target measures lead to different SI values or may be combined accordingly, see the example in Section V-B4.
- Costs or importance of configurations or parameter settings need to be considered. This may be achieved through a proper weighting function.
- Elasticity is the ability of a system to automatically scale resources up or down based on demand, and is a need for scalability (see also Figure 4).
- Fault Tolerance and Redundancy are essential. Scalable systems should be designed with fault tolerance in mind. Redundancy and replication of critical components can ensure that failures in one node do not disrupt the entire system. This behavior is included through the target measure function, as system reaction to faults.

The scalability index has the following characteristics to quantify the scalability of a system under test $\mathcal{F}$.

- A system function $f(x)$ quantifies an arbitrary target measure of interest for the system $\mathcal{F}$.
- A weighted parameter range w(x) allows defining the importance or costs of parameter / configuration settings.
- An arbitrary reference system $\mathcal{H}$ with a target reference function $h(x)$ is required to quantify how well the system $\mathcal{F}$ scales in comparison to an ideal, optimal, linear, or arbitrary system.

In practice, there are several means to attain scalability. Some common examples are reflected here. *Vertical Scalability (Scaling Up)* is the ability to handle increased load by adding more resources to a single node, such as increasing the CPU, RAM, or storage capacity of a server. Vertical scalability is often limited by the hardware limitations of a single machine. *Horizontal Scalability (Scaling Out)* handles increased load by adding more nodes to a distributed system, such as adding more servers to a cluster. Horizontal scalability is typically achieved through load balancing and partitioning of data and tasks across multiple nodes. *Load Balancing* means distributing incoming workloads (evenly) across multiple resources or nodes, ensuring that no single node is overwhelmed while others are underutilized. For distributed systems, *data partitioning and sharding* involve breaking large datasets into smaller, manageable subsets and storing them across different nodes. This strategy helps improve data access and distribution, making it easier to scale horizontally. In particular, *caching* frequently accessed data can significantly improve system performance and reduce the load on devices. Effective caching mechanisms can enhance scalability by reducing the need for repeated data processing. Finally, breaking down a monolithic system into smaller, loosely coupled *modular microservices* can enhance scalability. Each microservice can be scaled independently based on its specific demands.

IV. COMPARISON OF GENERAL SI WITH EXISTING SCALABILITY FRAMEWORKS

Table 2 summarizes relevant approaches from related work which investigate scalability. The ingredients of a scalability index are analyzed, that are the target measure and the parameter of interest. It is considered if a reference system is considered, e.g., a linear system, to investigate linear scalability. Or if the approach can be used for comparing the scalability of two systems. Our introduction of weights is only partly addressed in literature by taking into account costs.

The column 'scope' in Table 2 indicates what kind of concept is really considered, e.g., efficiency, performance, scalability. Some works consider only performance curves, e.g., a performance target measure depending on a parameter like load (i.e., performance curve). Based on that curve, scalability is qualitatively analyzed without providing any scalability index. The column 'index' indicates whether a scalability index is provided. The column 'application' shows the example domain investigated in the presented approach.

A detailed description of the approaches is discussed in Section II-B, while a brief summary is provided at the bottom of Table 2. The comparison shows that literature misuses the term scalability and partly considers different aspects like efficiency or performance. The related performance- or efficiency-curves give only qualitative insights. Those system functions are the main ingredient of a scalability analysis, and it is one possibility to report the system function as *curve measure* concerning scalability. To relate this curve measure to linear scalability, additionally a linear curve is then provided as a reference function. Or when comparing two systems, the two system functions are provided as curve measures for scalability.

However, as already discussed, just providing those system curves is typically not sufficient to identify which system scales better, see as an example Figure 2 or Figure 5. Especially when the two system functions are crossing each other.

Our goal is to provide a *single-value* scalability index which allows comparing the scalability of the two systems. Therefore, we need to aggregate the system function into a single value, which is done through the integral measurement. To relate the two systems, the ratio of the integral measurements is considered, reflecting the SI. As shown previously, this is identical to the ratio of the means of the two functions.

Such a desired single-value scalability index is only provided in a few works. Our framework generalizes those works and allows quantifying, e.g., linear, scalability, or to benchmark the scalability of two systems. In the following, we will show for some selected scalability measures from the literature how they fit into our generalized framework. The approaches from literature are a special case of our SI.

A. SI OF CLOUD SOFTWARE SERVICES IN RELATION TO IDEAL SYSTEM

A single-value scalability measure for cloud-based software services is provided in [1]. As target measure, the response time of cloud services or volume of available software instances is considered. They compare the system $\mathcal{F}$ under investigation with an ideal system $\mathcal{H}$. The single-value metric J is defined by comparing the areas under the curve. It is $0 \leq J \leq 1$ with corresponding $\gamma \in \{-1; 1\}$.

$$J = \left(\frac{\int_x f(x)\, dx}{\int_x h(x)\, dx} \right)^{\gamma} \tag{11}$$

Bringing this approach into our scalability framework yields the following instances of the SI constituents.

- Reference [1] "Scalability analysis comparisons of cloud-based software services" by Al-Said Ahmad and Andras
- target measure: $f(x)$
- reference system: ideal system with $h(x)$
- parameter range: $x \in [x_0; x_1]$
- weight: $w(x) = 1$

$$SI = \left(\frac{\int_x f(x) \cdot w(x)\, dx}{\int_x h(x) \cdot w(x)\, dx} \right)^{\gamma} = J \tag{12}$$

B. LINEAR SCALABILITY OF BIG DATA PROCESSING SYSTEMS (BDPS)

Linear scalability is a common term in literature, e.g., [13], and [14] divide scalability into three categories that are linear scalability, sublinear scalability and super-linear scalability. For the quantification, the system's performance $v(x)$ is divided by a linear function $l(x)$. As a result, a performance curve $f(x) = v(x)/l(x)$ is obtained, which indicates if the performance is better ($f(x) > 1$) or worse than that of the linear system.

As a concrete example, $f(x)$ is the speedup of the system, when the number of processing nodes is x, while $l(x)$ indicates linear speedup. Then, the system function is $f(x) = v(x)/l(x)$. Furthermore, a more advanced measure for BDPS is provided which goes beyond speedup. Still, the basic concept is to relate the measure to a linear system $l(x)$.

For evaluating scalability, [13] provides the curve measure $f(x)$. Then, $f(x) > 1$ and $f(x) < 1$ shows super-linearity and sub-linearity, respectively. To obtain a single value J from the curve measure, we may use the average value. Then, the single value J indicates whether the system scales linearly or better ($J \geq 1$).

$$J = \frac{1}{n} \sum_{x=1}^{n} f(x) = \frac{1}{n} \sum_{x=1}^{n} \frac{v(x)}{l(x)} \tag{13}$$

Bringing this approach into our scalability framework yields the following instances of the SI constituents.

- Reference [13] "Scalability and performance analysis of BDPS in clouds" by Li, Ou, Zhou, et al.
- target measure: $f(x) = v(x)/l(x)$

TABLE 2. Comparison of the suggested scalability index and the framework for comparing the scalability of two systems. The 'scope' indicates what is considered in the different papers and if a single metric for scalability is provided. 'Index' shows whether a single scalability index is provided to quantify the scalability of a system. The 'application' shows the example domain investigated in the presented approach.

Ref.	Target Measure	Reference System	Parameter	Weights	Index	Scope	Application
[1]	response time (system quality scalability), volume of software instances (provisioning)	ideal, optimal system	demand	—✗—	✓	scalability	cloud software services, see Section IV-A
[2]	assigned resources as major measure, performance, QoS	—✗—	workload, demand	—✗—	✓	scalability	cloud computing
[20]	performance to used resources ratio (PRR)	—✗—	workload (changes)	price factor	—✗—	scalability	cloud applications
[11]	ratio of workload (flows) over overhead (messages)	—✗—	traffic load	—✗—	—✗—	efficiency	control planes in software defined networks
[13]	performance	linear	#nodes	cost	—✗—	scalability	big data processing systems in clouds, see Section IV-B
[14]	performance	linear	load	—✗—	—✗—	scalability	(sub-/super-)linear scalability; *isospeed scalability*
[15]	performance	—✗—	stochastic variation	—✗—	—✗—	scalability	stochastic scalability
[21]	success ratio (SLO)	—✗—	load	probability of load	—✗—	scalability	domain-based scalability [21], domain-specific scalability [22], see Section IV-C
[16]	Performance Non-Scalability Likelihood (PNL)	—✗—	offered load	probability of system state	—✗—	performance	customer care database system
[4]	productivity per cost	—✗—	demand	cost	—✗—	efficiency	distributed computing applications, see Section IV-E
[23]	power metric: throughput per response time	—✗—	demand	cost	—✗—	efficiency	p-scalability for distributed computing applications, see Section IV-D
[17]	speedup	—✗—	#processors	—✗—	—✗—	Isoefficiency	parallel algorithms and architectures
[18]	efficiency	—✗—	workload	—✗—	—✗—	H-isoefficiency	heterogeneous parallel systems
[19]	required resources/ max. load the system can process	linear, exponential	load/ provisioned resources	—✗—	—✗—	resource/ demand scalability	stream processing
Our	any ✓	any ✓	any ✓	any ✓	SI ✓	scalability benchmarking	generalization of scalability metrics above ✓

[1] The definition is closest to ours, but is limited to an optimal reference system which may be unknown in practice. Only dedicated target measures for the software-as-a-service domain are considered. This is particular use case of our SI.

[2] Surveys literature on scalability definitions and provides an overview on scalability concepts concerning target measures and parameters. Scalability means are horizontal scaling (adding computing nodes) and vertical scaling (adding computing power to single node).

[20] The ratio of performance to used resources (PRR) is measured. The scalability of the system under test is measure by the performance change (PC) when workload changes. The variance of the performance change is an indicator for the scalability.

[11] Another target function appropriate for the scalability of control planes is introduced. Workload is the number of flows in the data plane. Overhead are the processed messages in the control plane. For data plane scalability, throughput and latency are mentioned. However, an efficiency metric is defined, not scalability.

[13] Linear scalability is considered for big data processing systems in clouds, based on the definition in [14]. A higher level target measure of the system performance for a dedicated workload is defined as scalability curve, but no single value is provided.

[14] By using a theoretical system with linear relationship between the parameter and the target measure, linear scalability, sublinear scalability and super-linear scalability are qualitatively compared. This is particular use case of our SI.

[14] Relates the workload capacity of the system at two different scales.

[15] The influence of stochastic variations of the parameter under investigation is considered for arbitrary target measures. No scalability metric is provided. We use this parameter as example to quantify stochastic scalability with our SI.

[16] As target measures the Performance Non-Scalability Likelihood (PNL) is introduced. No scalability measure is provided.

[4] Productivity is using a Power metric, but expresses efficiency. The throughput in responses weighted with value per response is the goodness. The costs are the badness. The ratio of efficiency for two different load scenarios is defined as scalability, but it quantifies efficiency. If productivity is maintained as the scale changes, the system is regarded as scalable.

[23] P-scalability uses the power metric as target measure. Then the ratio of the power metric for two systems is quantified as P-scalability. Efficiency curves are considered, but no scalability index of a system is provided.

[17] Scalability is described as the "ability of parallel machines to keep the parallel efficiency constant when the system and problem size increase". Efficiency curves are considered, but no scalability index of a system is provided.

[18] Efficiency curves are considered, but no scalability index of a system is provided.

[19] Performance curves are considered and compared to linear or exponential curves. No scalability index is provided.

- reference system: $h(x) = 1$
- parameter range: number of nodes $x \in \{1, 2, \ldots, n\}$
- weight: $w(x) = 1$

$$SI = \frac{\sum_{x=1}^{n} f(x) \cdot w(x)}{\sum_{x=1}^{n} h(x) \cdot w(x)} = \frac{\sum_{x=1}^{n} f(x)}{n} = J \tag{14}$$

Instead of using the relative target measure, the two curves $v(x)$ and $l(x)$ may be directly used for comparison. This leads to an alternative definition of the target measure and the reference system.

- alternative comparison to [13]
- target measure: $\hat{f}(x) = v(x)$
- reference system: $\hat{h}(x) = l(x)$
- parameter range: number of nodes $x \in \{1, 2, \ldots, n\}$
- weight: $w(x) = 1$

$$\hat{SI} = \frac{\sum_{x=1}^{n} \hat{f}(x) \cdot w(x)}{\sum_{x=1}^{n} \hat{h}(x) \cdot w(x)} = \frac{\sum_{x=1}^{n} v(x)}{\sum_{x=1}^{n} l(x)} \tag{15}$$

$$\neq \frac{1}{n} \sum_{x=1}^{n} \frac{v(x)}{l(x)} = \frac{1}{n} \sum_{x=1}^{n} f(x) = SI \tag{16}$$

The two definitions SI and $\hat{SI}$ lead to different values, but similar behavior of the scalability index and conclusions are observed. Figure 6 shows exemplary the scalability index SI and $\hat{SI}$ when considering the throughput of a system depending on the number of processing nodes x. The throughput of the system is $\hat{f}(x) = v(x) = x^{\beta}$ with $\beta = 0.9$. The reference system yields $\hat{h}(x) = l(x) = x$. The scalability index $\hat{SI}$ is computed for the parameter range $\{1, \ldots x_1\}$, and x_1 is varied in Figure 6. In contrast, for the computation of SI, the target measure $f(x) = \frac{v(x)}{l(x)} = x^{\beta-1}$ and the reference function $h(x) = 1$ are used. Both definitions of SI lead to similar results in practice.

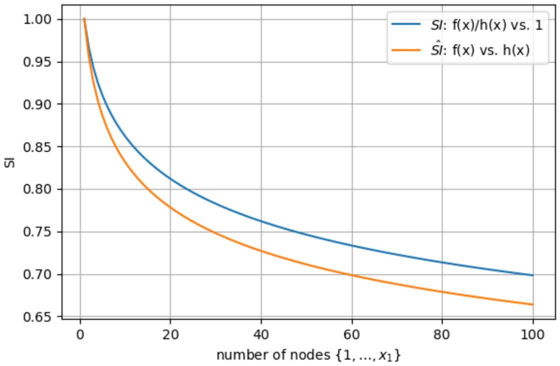


FIGURE 6. Different definitions of the target measure and the reference system lead to the scalability index $\hat{SI}$ and SI as defined in Eq.(15) and Eq.(16), respectively. It is $\hat{f}(x) = x^{\beta}$ with $\beta = 0.9$ and $\hat{h}(x) = v(x) = x$ yielding $\hat{SI}$. In contrast, SI uses $f(x) = v(x)/l(x) = x^{\beta-1}$ and the reference function $h(x) = 1$.

C. DOMAIN-BASED SCALABILITY FOR MICROSERVICE ARCHITECTURES

A single value scalability metric is provided for the assessment of microservice architectures in [21]. Thereby, the response time of the architecture is considered for a given workload. This workload in the system is modeled as a random variable X with a probability density function $p(x)$ for the parameter range $x \in [x_0; x_1]$. Thus, $\int_{x_0}^{x_1} p(x)\, dx = 1$. Different services are tested under a certain load test specification. The fraction of successful executions of all services for a given load x while keeping the response time below a threshold is the target measure of interest $f(x)$. The scalability metric is defined as follows in [21].

$$J = \int_{x_0}^{x_1} f(x) \cdot p(x)\, dx = E[f(X)] \tag{17}$$

Hence, the scalability index J is the expected success ratio $E[f(X)]$ over all load conditions X, i.e., the expected value of a function $f(x)$ of the random variable X.

Bringing this approach into our scalability framework yields the following instances of the SI constituents.

- Reference [21] "Scalability assessment of microservice architecture deployment configurations: A domain-based approach leveraging operational profiles and load tests" by Avritzer, Ferme, Janes, et al.
- target measure: $f(x)$ is the fraction of successful executions of all services for a given load x while keeping the response time below a threshold
- reference system: ideal system $h(x) = 1$
- parameter range: $x \in [x_0; x_1]$
- weight: $w(x) = p(x)$ with probability density function $p(x)$ for the random variable X of the load in the system

$$SI = \frac{\int_{x_0}^{x_1} f(x)w(x)\, dx}{\int_{x_0}^{x_1} h(x)w(x)\, dx} = \frac{\int_{x_0}^{x_1} f(x)p(x)\, dx}{\int_{x_0}^{x_1} p(x)\, dx}$$

$$= \frac{\int_{x_0}^{x_1} f(x)p(x)\, dx}{1} = E[f(X)] = J \tag{18}$$

Besides the domain-based metric by [21], a domain-specific metric was proposed by [22]. The differences between the works is not the quantification of scalability as a single-value measure, but whether resources may be added to satisfy specified service-level objectives (SLOs) requirements.

D. P-SCALABILITY OF DISTRIBUTED SYSTEMS

The P-scalability metric [23] depicts a measure curve which combines capacity and response time with cost. Thereby, the so-called 'power' measure $P(x)$ is used when the system is considered with a scale factor x. This power measure is the ratio of the throughput of a distributed system and the response time of the system. It follows Kleinrock's power metric [24] which considers the ratio of 'goodness' (here: throughput) to 'badness' (here: response time). We discuss Kleinrock's approach in more detail for the use case of availability in communication networks in Section V-B4.

The power measure $P(x)$ is combined with the costs $C(x)$ and reflects the system function $f(x) = P(x)/C(x)$. The scale factor x reflects a certain number of active users or jobs x.

As a simple example, a database system is considered in [23], where the number of processors and the database size depend on that scale factor x.

The P-scalability is then defined as the ratio between $f(x)$ (system with scale factor x) and a reference system $f(x_1)$ (system with scale factor x_1). However, [23] does not provide a single-value scalability metric. Therefore, we consider the system for all scale factors $x \in [x_0; x_1]$ and compute the average system function $\overline{f}$. Then, a scalability index may be given depending on the reference scale factor x_1, i.e., $h(x) = f(x_1) = P(x_1)/C(x_1)$.

$$J = \frac{\overline{f}}{f(x_1)} = \frac{\int_{x_0}^{x_1} f(x)\,dx}{x_1 - x_0} \cdot \frac{1}{f(x_1)} \tag{19}$$

Bringing this approach into our scalability framework yields the following instances of the SI constituents.

- Reference [23] "A scalability metric for distributed computing applications in telecommunications" by Jogalekar and Woodside
- target measure: $f(x)$
- reference system: $h(x) = f(x_1)$
- parameter range: $x \in [x_0; x_1]$
- weight: $w(x) = \frac{1}{x_1 - x_0}$

$$SI = \frac{\int_{x_0}^{x_1} f(x) \cdot w(x)\,dx}{\int_{x_0}^{x_1} h(x) \cdot w(x)\,dx} = \frac{\int_{x_0}^{x_1} \frac{f(x)}{x_1 - x_0}\,dx}{\int_{x_0}^{x_1} \frac{f(x_1)}{x_1 - x_0}\,dx}$$
$$= \frac{\int_{x_0}^{x_1} \frac{f(x)}{x_1 - x_0}\,dx}{f(x_1)\int_{x_0}^{x_1} \frac{1}{x_1 - x_0}}\,dx = \frac{\int_{x_0}^{x_1} f(x)\,dx}{(x_1 - x_0) f(x_1)} = J \tag{20}$$

E. PRODUCTIVITY AND VALUES AS TARGET MEASURE

In a follow-up work of [23], another target measure was defined [4], which is quite interesting. For a distributed system, the *productivity* is considered as the value delivered per second, divided by the cost per second at a scale factor x. For a throughput $t(x)$ (in responses per second at scale k) and average value $v(x)$ of each response, calculated from its quality of service at scale x, the productivity $f(x)$ is

$$f(x) = t(x) \cdot v(x)/c(x) \tag{21}$$

for the costs $c(x)$ at scale x, expressed as a running cost per second to be inline with the definition of $t(x)$.

Similarly, as above [23], the scalability measure is a curve measure and defined as the ratio of productivity figures. Over the entire parameter range x of scale factors, we consider therefore the average productivity and relate it to the reference point $f(x_1)$.

$$J = \frac{\overline{f}}{f(x_1)} = \frac{\int_{x_0}^{x_1} f(x)\,dx}{x_1 - x_0} \cdot \frac{1}{f(x_1)} \tag{22}$$

Bringing this approach into our scalability framework yields the following instances of the SI constituents.

- Reference [4] "Evaluating the scalability of distributed systems" by Jogalekar and Woodside
- target measure: $f(x)$
- reference system: $h(x) = f(x_1)$
- parameter range: $x \in [x_0; x_1]$
- weight: $w(x) = \frac{1}{x_1 - x_0}$

$$SI = \frac{\int_{x_0}^{x_1} f(x) \cdot w(x)\,dx}{\int_{x_0}^{x_1} h(x) \cdot w(x)\,dx} = \frac{\int_{x_0}^{x_1} \frac{f(x)}{x_1 - x_0}\,dx}{\int_{x_0}^{x_1} \frac{f(x_1)}{x_1 - x_0}\,dx}$$
$$= \frac{\int_{x_0}^{x_1} \frac{f(x)}{x_1 - x_0}\,dx}{f(x_1)\int_{x_0}^{x_1} \frac{1}{x_1 - x_0}}\,dx = \frac{\int_{x_0}^{x_1} f(x)\,dx}{(x_1 - x_0) f(x_1)} = J \tag{23}$$

Note that Eq.(20) and Eq.(23) are identical, just the definition of the target measure differs: power measure vs. productivity.

F. USING SI WITHOUT REFERENCE SYSTEM: GENERAL SYSTEM'S REFERENCE POINT

Some examples above used a single reference point x_r and related the target measure of the system $f(x)$ to the target measure at the reference point $f(x)$, like in Section IV-E.

In general, instead of a reference system $\mathcal{H}$, a single reference point x_r and its target measure $f(x_r)$ of the system $\mathcal{F}$ can be utilized to indicate scalability. To be more precise, the target measure of the system is related to that reference point, how the system develops. This reference point may lead to the best target measure, but any arbitrary reference point can be used. It is just used to relate the target measure accordingly.

We consider the speedup as an example. The response time of the system $\mathcal{F}$ is $T(x)$ if x servers are used. The speedup is then the factor $S(x) = T(1)/T(x)$, i.e., the improvement of the response time with respect to a reference system with $x_r = 1$ server and response time $T(1)$. As weight, we consider the probability $p(x)$ that the system has x active servers. The discrete random variable X models the number of active servers with a probability function $p(x)$. It is $\sum_{x=x_0}^{x_1} p(x) = 1$.

In general, for a discrete or continuous random variable X, we have the following ingredients for the SI, respectively.

- scalability index computation by only using the system function without reference system
- target measure: $f(x)$
- reference system: $h(x) = f(x_r)$ with reference point x_r
- parameter range: $x \in \{x_0; \ldots; x_1\}$ for a discrete parameter x; $x \in [x_0; x_1]$ for a continuous parameter x
- weight: $w(x) = p(x)$ with probability mass function $p(x)$ of the discrete random variable X; $w(x) = p(x)$ with probability density function $p(x)$ of the continuous random variable X

The SI for a discrete parameter x is as follows.

$$SI = \frac{\sum_{x=x_0}^{x_1} f(x) \cdot w(x)}{\sum_{x=x_0}^{x_1} h(x) \cdot w(x)} = \frac{\sum_{x=x_0}^{x_1} f(x)p(x)}{f(x_r)\sum_{x=x_0}^{x_1} p(x)} \tag{24}$$
$$= \frac{E[f(X)]}{f(x_r)} \tag{25}$$

Hence, the expected target measure $E[f(X)]$ is normalized by the target measure at the reference point. For the example of speedup, we may use as reference point the maximal speedup when x_1 servers are available. Then, the SI indicates how far the system is away from the performance at the reference point. However, in practice, it may be more interesting to understand if the speedup linearly scales or not. Then, the SI is the relation between the expected target measure of the system $\mathcal{F}$ and the linear reference system $\mathcal{H}$: $SI = E[f(X)]/E[h(X)]$ with $E[h(X)] = m \cdot X + c$.

Similarly, the SI for a continuous parameter x is computed by using the probability density function $p(x)$.

$$SI = \frac{\int_{x=x_0}^{x_1} f(x) \cdot w(x)\, dx}{\int_{x=x_0}^{x_1} h(x) \cdot w(x)\, dx} = \frac{\int_{x=x_0}^{x_1} f(x)p(x)\, dx}{f(x_r)\int_{x=x_0}^{x_1} p(x)\, dx} \tag{26}$$

$$= \frac{E[f(X)]}{f(x_r)} \tag{27}$$

It is the expected target measure of the system $\mathcal{F}$, which is normalized by the target measure at the reference point.

V. USE CASES FOR DEMONSTRATION

The application of the SI is demonstrated for four use cases: (1) performance of an IoT load balancer depending on the system load, (2) availability of a communication system depending on the size and structure of the network, (3) scalability comparison of different location selection mechanisms in fog computing with respect to delays and energy consumption; (4) comparison of time-sensitive networking (TSN) mechanisms in terms of efficiency and utilization. The use cases show the need for proper selection of reference systems and target measures. The selected reference system and target measure aim at analyzing the scalability with a concrete question in mind. We also show the impact of the weighting of the parameter space, e.g., according to its occurrence in practice.

A. WAITING TIME OF AN IoT LOAD BALANCER

An IoT scenario is considered where data arrives from sensor nodes and is aggregated at a load balancing gateway. This IoT load balancer then forwards the data to the backend cloud servers according to some load balancing strategy. This IoT load balancer may be the performance bottleneck of the IoT architecture [25] and we can model it as a single server queueing system, e.g., to dimension the load balancer. An appropriate model is an M/GI/1 waiting queue [3], for which analytical formulas are well known and used here to produce numerical results. IoT messages arrive at the load balancer at rate λ and are served at rate μ. The mean service time to process a single message is $E[B] = 1/\mu$. The offered load and the utilization of the load balancer is $\rho = \lambda/\mu$ which is the key quantity defining the waiting time of messages in the queue before they are served. The system is stable when $\lambda < \mu$ ($\rho < 1$). The variance of the service time is described by the coefficient of variation c_B.

1) SCALABILITY OVER THE ENTIRE PARAMETER RANGE

It is well known that a waiting system is not scaling if the load gets close to 1. Then, the system is not stable anymore and waiting times get bigger and bigger, thus for $\rho \to 1$, the expected waiting times are $E[W] = \infty$. The average waiting time is as follows, see, e.g., [3].

$$E[W] = E[B]\frac{\rho(1 + c_B^2)}{2(1 - \rho)} \tag{28}$$

Figure 7 plots the expected waiting time (the target measured) depending on the load (the parameter under investigation $x = \rho$). We compare it to a linear reference system with $h(\rho) = \rho$. For $\rho > 0.5$, the IoT load balancer has a worse performance in terms of expected waiting time than the linear system. The strong decay indicates that the system gets unstable when approaching $\rho = 1$.

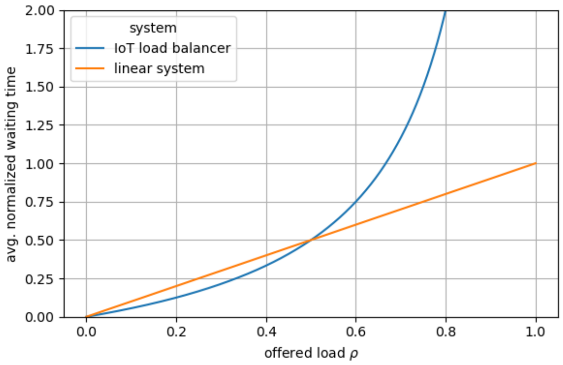


FIGURE 7. IoT load balancer – Linear scalability wrt. mean waiting time of messages normalized by message processing time. Deterministic message processing times are assumed.

Let us formally analyze that *a waiting system is not scaling if the load gets close to* 1. We are interested in analyzing scalability with respect to the expected waiting time, a linear reference system with $h(x) = x$ and we consider the entire parameter range $\mathbb{X} = [0; 1]$. Then, the integral measurement diverges: $F = \int_{\mathbb{X}} f(x)\, dx = \infty$ with $f(x) = E[W]$. Hence, the scalability index is $SI = H/F = 0$.

2) WEIGHTING THE PARAMETER RANGE

In practice, additional mechanisms like admission control may be implemented to guarantee that the load is below 1, e.g., $\rho \leq 0.8$ such that the expected waiting time is below a certain threshold and service-level agreements (SLAs) can be met. Therefore, we limit the parameter range accordingly. Figure 8 shows the integral measurement of the IoT load balancer and the linear system, while the considered parameter range of the scalability analysis is $[0; \rho]$. We observe that the integral measurement H of the linear system is slightly above the integral measurement F of the load balancer when the load is below 0.68. At the intersection point $\rho^* \approx 0.68$, the integral measurements are identical and $SI = 1$.

It is important to understand that the performance of the IoT load balancer is worse than the linear system at the load ρ^*. However, the scalability index considers the entire range of parameter settings, i.e. $\rho \in [0; \rho^*]$. Over the entire range, the accumulated expected waiting times are

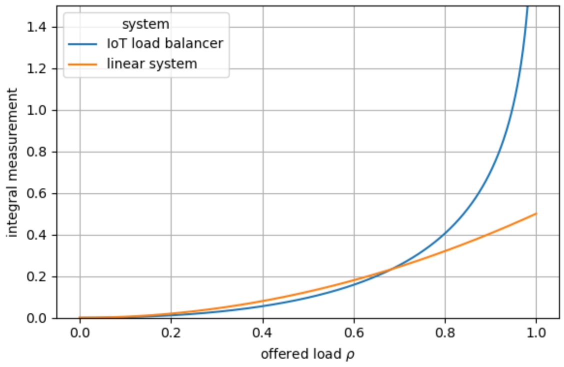


FIGURE 8. IoT load balancer – Integral measurement over the parameter range [0; ρ] for the IoT load balancer and the linear system regarding the mean waiting time of messages.

identical for both systems and the scalability behavior is the same, i.e. $SI = 1$. This is also visualized in Figure 9.

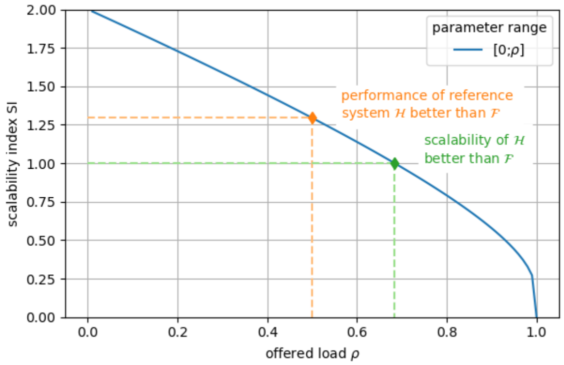


FIGURE 9. IoT load balancer – Scalability index over the parameter range [0; ρ] with respect to linear function $h(\rho) = \rho$ and mean waiting time of messages.

In general, it is important to consider the parameter range of interest to draw conclusions. Furthermore, the importance of the particular parameter settings may be adjusted. For example, with higher offered load, more IoT sensors and their messages are fed to the cloud. Therefore, this scenario and higher loads may be more important in the evaluation ('high load importance'). On the other hand, this load situation may not be so relevant in practice, since it does not occur with high probability. Assume a scenario where the less the load, the more likely the scenario occurs. Accordingly, the parameter weights may be adjusted ('load and medium load'). If such weights are unknown, then all parameter settings should be equally weighted ('equal importance'). Figure 10 indicates the scalability index for those three different scenarios. The resulting scalability index depending on the upper parameter $x_1 = \rho$ to be considered in the integral measurement is visualized. We see similar behavior for the three different weight functions. Nevertheless, with low and medium load having a higher importance, the SI is higher, e.g., when considering

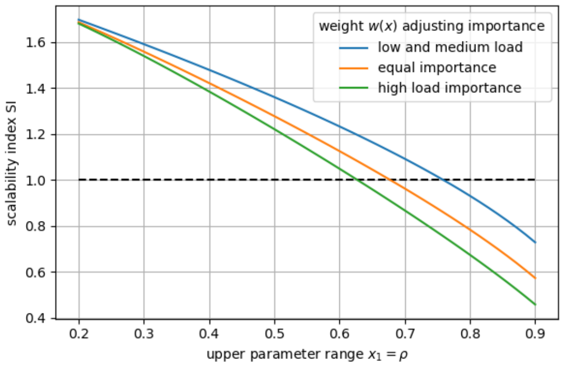


FIGURE 10. IoT load balancer – Scalability index over a weighted parameter range [0.1; ρ] with respect to linear function $h(\rho) = \rho$ and mean waiting time of messages. Three scenarios are considered: low and medium load importance ($w(x) = 1/x$), equal importance of all parameter settings ($w(x) = 1$), high load importance ($w(x) = (x + 1)^4$).

the parameter range [0.1; 0.9], the SI is 0.73, 0.57, 0.45 for low load, equal, high load importance, respectively.

3) STOCHASTIC SCALABILITY

Instead of considering how the system scales with respect to load, it may also be interesting to analyze the scalability in terms of variance of the service process due to different IoT message sizes. To this end, the coefficient of variation c_B of the service demand (i.e., message size) is considered as the parameter under investigation in the scalability analysis. Hence, the stochastic scalability of the IoT load balancer is investigated. Figure 11 shows the expected waiting time depending on c_B.

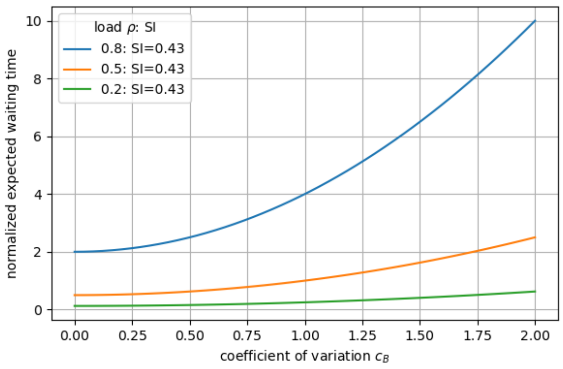


FIGURE 11. Stochastic scalability of the IoT load balancer – Expected waiting time of the IoT load balancer for varying service demands, expressed as coefficient of variation c_B of the service time of a single IoT message. The expected waiting time is normalized with the mean service time $E[W]/E[B]$.

The scalability analysis uses the best case as a reference system $\mathcal{H}$, i.e., deterministic service times ($c_B = 0$), and considers the expected waiting time over the coefficient of variation in the range [0; 2.0]. The resulting scalability index is $SI = 0.43$ for any load ρ.

$$F = \int_{x_0}^{x_1} \tfrac{E[B]\rho}{2(1-\rho)}(1 + c_B^2)\, dc_B = \frac{14}{3}\tfrac{E[B]\rho}{2(1-\rho)}$$

$$H = \int_{x_0}^{x_1} \tfrac{E[B]\rho}{2(1-\rho)}(1 + 0^2)\, dc_B = 2\tfrac{E[B]\rho}{2(1-\rho)}$$

$$SI = H/F = \frac{6}{14} \approx 0.43$$

4) DIFFERENT TARGET MEASURES: MEAN, QUANTILE, PROBABILITY

So far, we have changed the parameter range, the parameter weights, and the reference system. Next, we consider different target measure functions. Instead of looking at average waiting times, the $\alpha\%$-quantile q_α of the waiting times or the probability w_y that the waiting time is below $y = 100\,\text{ms}$ are considered, which may be more relevant SLAs in practice.

We assume that the server operates with exponentially distributed service times (M/M/1-∞). Then, the cumulative distribution function of the waiting time is

$$W(t) = 1 - \rho \cdot e^{-(1-\rho)\mu t} \tag{29}$$

and the $\alpha\%$-quantile follows as

$$q_\alpha = -\frac{\log(\frac{1-\alpha}{\rho})}{(1-\rho)\mu} \,. \tag{30}$$

With exponentially distributed service times ($c_B = 1$), the expected waiting time is

$$E[W] = E[B] \cdot \frac{\rho}{(1-\rho)} \,. \tag{31}$$

Those different key characteristics may be considered as SLAs and are depicted in Figure 12 depending on the system load.

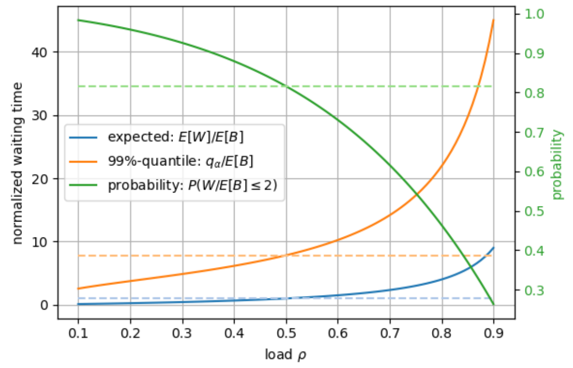


FIGURE 12. IoT load balancer – Different SLAs are now considered in the scalability analysis.

In the scalability analysis, we consider those SLAs as different target measures over the range [0.1; 0.9] with equal weights. However, the question arises what is a good reference system. To this end, we define the reference system to always operate like the original IoT load balancer at load $\rho = 0.5$, see the dashed lines in Figure 12. For all target measures, the SI is below 1. Thus, the original system is less scalable than the reference system. In particular, the SI is $SI_{E[W]} = 0.5726$, $SI_{q_{0.99}} = 0.7016$, $SI_{W(2E[B])} = 0.9239$, respectively.

5) ECONOMY OF SCALES – BUNDLING SERVERS

The term *"economies of scale"* is well known in the analysis of queueing systems and refers to the fact that larger-scale operation has advantages over smaller ones. In the particular use case of the IoT load balancer, there may be several load balancers which are independently operating. Let us assume that there are n independent load balancers, i.e., a single processing unit operating with service rate μ and a single waiting queue for each of the n load balancers. The arrival rate of requests to each of the n load balancers is λ.

Bundling all the n independent load balancers into a single one results in a single entity with n processing units, but a single waiting queue for all incoming requests. This single entity then needs to serve all requests, i.e. $n \cdot \lambda$. Economies of scale means that the bundled servers will result in a better performance like the waiting probability of incoming requests or the expected waiting time.

However, we are interested in quantifying the scalability of the bundled servers, which is modeled as M/M/n-∞ waiting queue. Formulas for the expected waiting time $E[W]$ or the probability p_W that a request has to wait (known as Erlang-C formula) are given in literature, see e.g., the Python implementation for the numerical calculation of the M/M/n-∞ in [3].

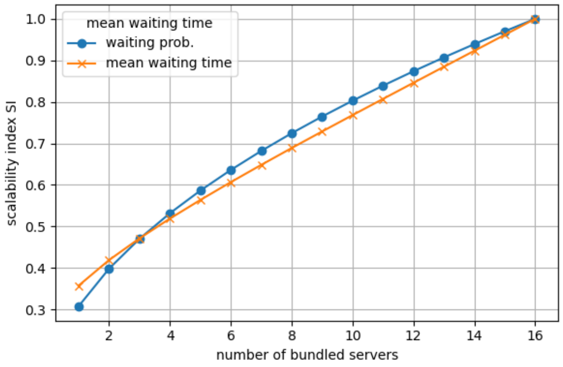


FIGURE 13. Economies of scales of the IoT load balancer – The scalability analysis uses as reference system an M/M/16 waiting queue with n processing units, service rate μ per unit, arrival rate $n\lambda$. As target measure, the waiting probability and the expected waiting time are considered. The parameter under investigation is the load over the range [0.1; 0.9].

Figure 13 shows the scalability index depending on the number of bundled servers. If all IoT load balancers are operating independently ($n = 1$), the scalability index is only 0.3. Bundling the servers increases the scalability and the SI reaches 1 for the reference system.

6) HIGH-PERFORMANCE SERVER VS. COMMODITY SERVERS

Note that Figure 13 also shows the results to answer if it is better to have a single high-performance server or n servers bundled. The high-performance server operates at a rate μ and serves an arrival rate λ. When having n servers with less performance μ/n, the same total service rate is achieved. However, with several smaller servers bundled into one entity, the head-of-line blocking is reduced in comparison to the single high-performance server.

In Section V-A5, bundling servers means increasing the arrival rate $n\lambda$ with service rate μ per processing unit. Thus, the offered load is $n\lambda/\mu$ and the utilization per processing unit is λ/μ. Here in Section V-A6, the high-performance server has a service rate μ and serves an arrival rate λ, while the low-end servers have a service rate μ/n and serve the same arrival rate λ. Hence, the offered load is $n\lambda/\mu$ and the utilization per processing unit is λ/μ. Thus, we get the same results.

B. AVAILABILITY IN A COMMUNICATION SYSTEM

1) SYSTEM STRUCTURES/TOPOLOGIES

Communication systems have by design different physical (and logical) topologies and structures, e.g., given by the trade-off between the cost of network elements and the required system availability. The availability of connected peers will depend on the grade of redundancy provided in the physical structure available. In this example, we define four topologies with n (network) nodes:

(1) *Bus topology* (serial): a serial structure with no redundancy for the connected peers, and with only one peer-to-network link per peer, and n - 1 intermediate links.

(2) *Ring topology* (serial-parallel): a serial structure in two parallels which provides two node disjoint redundant paths for the peer to peer connection between X and Y. It has two peer-to-network links per peer, and n - 2 intermediate links, see Figure 14 for an illustration.

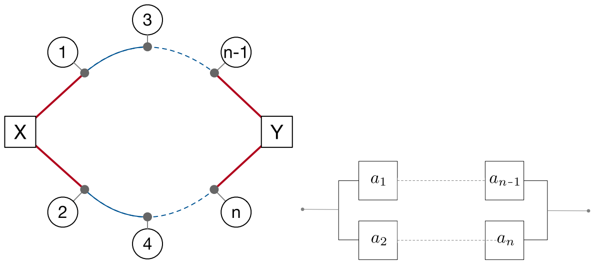


FIGURE 14. System structures and Reliability Block Diagrams (RBD) for serial-parallel structure.

(3) *Ring topology with cross-connects* (parallel-serial): two component parallels in a serial structure, which provides n non-disjoint redundant paths for the peer to peer connection between X and Y. It has two peer-to-network links per peer, and 2(n - 2) intermediate links, see Figure 15 for an illustration.

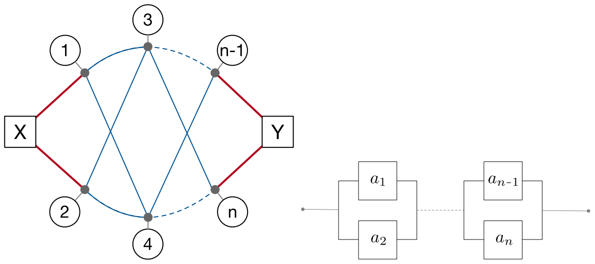


FIGURE 15. System structures and Reliability Block diagrams for parallel-serial structure.

(4) *Parallel topology* (parallel): a parallel structure with n node and link disjoint redundant paths for the connected peers, with n peer to network links, and no intermediate links.

The parallel structure has the best system availability and will be used as a reference system for the system availability, but is regarded as too expensive to be a practical alternative because the number of peer-to-network links (access links) increases with the network size. To reflect the cost of the alternatives, we add cost related to the number of links required in a network of size n nodes in the four different cases.

In Figures 14 and 15, the serial-parallel and parallel-serial systems are shown with their corresponding reliability block diagrams (RBD). From the RBDs, the system availability of the four system structures can be determined:

$$A_{\mathrm{s}}(n,a) = a^n \qquad \text{(serial)}$$
$$A_{\mathrm{sp}}(n,a) = 1 - (1 - a^{n/2})^2 \qquad \text{(serial-parallel)}$$
$$A_{\mathrm{ps}}(n,a) = (1 - (1 - a)^2)^{n/2} \qquad \text{(parallel-serial)}$$
$$A_{\mathrm{p}}(n,a) = 1 - (1 - a)^n \qquad \text{(parallel)}$$

where $n = 2, 4, 6, \cdots$ is the number of nodes, and a is the (homogeneous) node availability.

The costs of the two peer-to-network links and the cost (c) for each of the n intermediate links are considered for each topology:

$$c_{\mathrm{s}}(n) = (2c_a + (n-1)c) \qquad \text{(serial)}$$
$$c_{\mathrm{sp}}(n) = (4c_a + (n-2)c) \qquad \text{(serial-parallel)}$$
$$c_{\mathrm{ps}}(n) = (4c_a + 2(n-2)c) \qquad \text{(parallel-serial)}$$
$$c_{\mathrm{p}}(n) = (2nc_a) \qquad \text{(parallel)}$$

2) SCALABILITY INDEX: SYSTEM AVAILABILITY AS TARGET MEASURE

We want to study the scalability of the different structures. In this example, we focus on how the system availability scales with fixed node availability, a, when the number of nodes, n, increases in the range (n_0, n_1). The target function is *system availability*, $A_i(n, a)$. We define the parallel system as our *reference* system, and the other three as *target* systems.

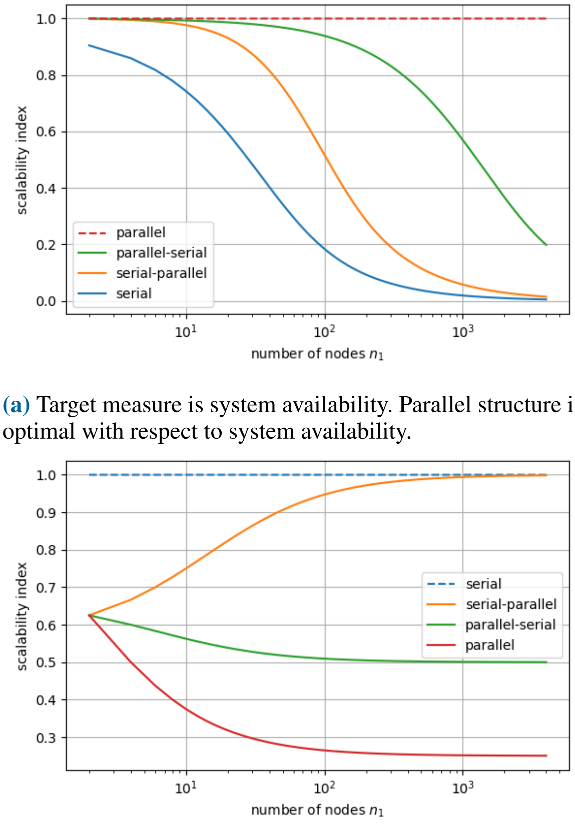
The scalability index is then

$$\mathtt{SI}_i(n|a) = F_i/H = \mathcal{A}_i(n|a)/\mathcal{A}_{\mathrm{p}}(n|a) \qquad (32)$$

where $i = \mathtt{s}, \mathtt{sp}, \mathtt{ps}$, and

$$\mathcal{A}_i(n|a) = \sum_{k=2}^{n} A_i(k,a)$$

Figure 16a shows the SI wrt. system availability and assumes a node availability $a = 0.95$ and the number of nodes in the range $n \in (2; 4000)$. Concerning the system availability, the parallel structure is optimal. Individual nodes may fail, still the communication is possible over other nodes. As expected, the parallel-serial is second best. The serial structure is the worst when considering the system availability and achieves the lowest SI.



(a) Target measure is system availability. Parallel structure is optimal with respect to system availability.

(b) Target measure are costs. Serial structure is optimal with respect to costs.

FIGURE 16. Scalability index for different network topologies in communication networks, depending on the number of nodes $n \in (2; n_1)$ with node availability $a = 0.95$. The reference system is the optimal one wrt. target measure and plotted as a dashed line.

3) SCALABILITY INDEX: COSTS AS TARGET MEASURE

The system availability of the parallel structure results however in much higher costs. Therefore, we consider now the scalability wrt. costs of the different structures. The serial structure is the optimal one regarding costs and used as a reference system. The link costs are $c_a = 2\,c$ with $c = 1$, i.e., the access links have double the cost compared with the intermediate links. We see significant differences of the SI for the structures wrt. costs, see Figure 16b.

Furthermore, we recognize that the ranking of the structures changes when considering system availability and costs. Depending on the target measure, the decision which topology to use in practice may vary. In general, the target measure of interest determines the scalability index. As we have seen, a system may scale with respect to one measure (here: system availability), but not for another measure (here: costs).

4) SCALABILITY OF STRUCTURES: COMBING AVAILABILITY AND COSTS

In practice, we may want to consider both, system availability and costs, to decide which topology to use regarding their scalability. There are several approaches how to tackle this multi-objective problem, e.g., Pareto optimization approaches and exploration strategies of the Pareto front, e.g., multi-objective ranking methods, e.g., the weighted sum method by converting the multi-objective problem into a single objective problem by linearly combining the objectives with weights, e.g., the constraint method by introducing constraints that reflect the importance of each objective.

For the scalability analysis of structures, we follow here Kleinrock's approach [24] and use the so-called *Power metric* ψ as a transformation into a one-dimensional utility metric. The power metric is the ratio of 'goodness' (i.e., system availability) divided by 'badness' (i.e., costs). Then, higher values of the power metric indicate a better system with respect to system availability and costs. Thus, we are combining system availability and costs appropriately and use the power metric in the scalability analysis.

$$\psi_i(n,a) = A_i(n,a)/c_i(n) \qquad \text{(power metric)} \tag{33}$$

As reference system, we use the parallel structure $\mathcal{H}$ and the corresponding integral reference measurement H. The scalability index is then

$$\mathtt{SI}_i(n|a) = H/F_i = \mathcal{A}^w_{\mathtt{p}}(n|a)/\mathcal{A}^w_i(n|a) \tag{34}$$

where $i = \mathtt{s}, \mathtt{sp}, \mathtt{ps}, \mathtt{p}$, and

$$\mathcal{A}^w_i(n|a) = \sum_{k=2}^{n} \psi_i(k,a) = \sum_{k=2}^{n} A_i(k,a)/c_i(k)\,.$$

In other words, we are weighting the target measure $A_i(k,a)$ in the integral measurement with a weight $w_i(k,a) = 1/c_i(k)$. Thus, a weight function $w_i(n)$ is added to each structure to reflect the cost (c_a) of the two peer-to-network links and the cost (c) for each of the n intermediate links:

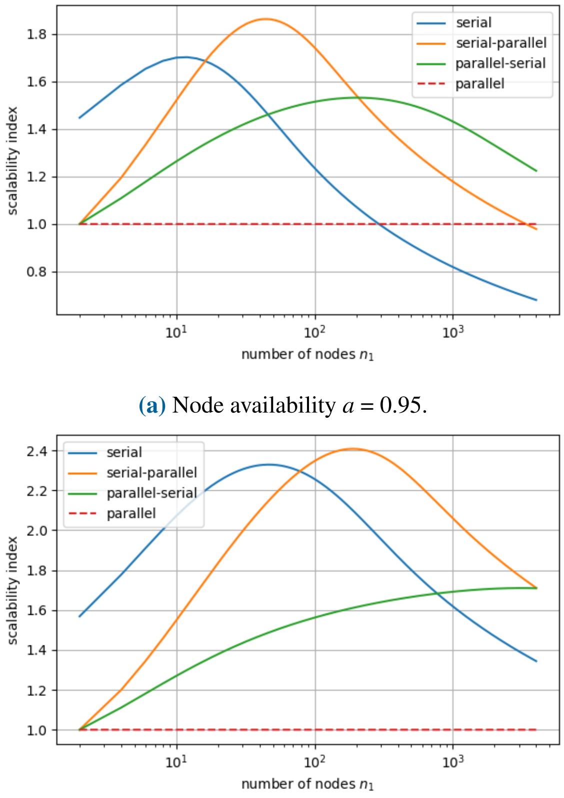
$$w_i(n) = 1/c_i(n) \qquad \text{(weight)}$$

The integral measurement of the reference is also weighted with the costs of the reference structure:

$$H = \mathcal{A}^w_{\mathtt{p}}(n|a) = \sum_{k=2}^{n} A_{\mathtt{p}}(k,a)/c_{\mathtt{p}}(k)\,.$$

Thus, the scalability analysis with Kleinrock's power metric as target measure is the same as the scalability analysis with system availability as target measure, but weighted with the inverse of the cost function.

In Figure 17, $SI_i(n|a)$ from Eq.(34) is plotted for $i = \mathtt{s}, \mathtt{sp}, \mathtt{ps}$, for node availability, $a = 0.95$ and $a = 0.99$, and number of nodes in the range $n \in (2; 400)$ and $n \in (2; 400)$. Again, the link costs are $c_a = 2\ c$ with $c = 1$, i.e., the access links have double the cost compared with the intermediate links.



(a) Node availability $a = 0.95$.

(b) Node availability $a = 0.99$.

FIGURE 17. System availability scalability of $n \in (2; n_1)$ with node availability $a = 0.95$ and $a = 0.99$, respectively.

The plots in Figure 17 show that all structures scale better than the parallel due to the cost of access links relative to the intermediate links when the node availability is high, $a = 0.99$. For lower node availability, $a = 0.95$, parallel outranks (crosses 1) the others, first serial, then serial-parallel, and finally parallel-serial. For $a = 0.95$, the serial structure scales best in the range (2; 16), and in the range (16; 210) the serial-parallel, and then the parallel-serial. Observe that even the serial scales better than the parallel in (2; 270). For $a = 0.95$, the same is observed; serial scales best in (2; 70), serial-parallel in (70; 4000), but now all scales better than the parallel structure.

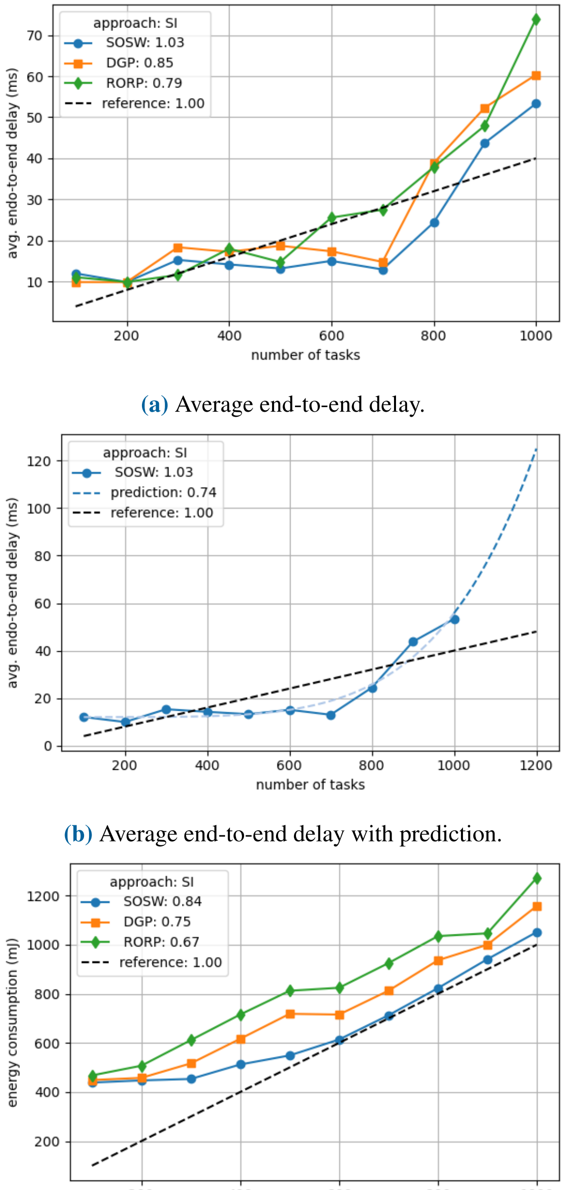
The main observation is that the scalability index as defined in this section gives can be useful to gain insight in how to structure the network when it is expected that the number of nodes will grow. The index used here takes both the cost of links (and distinguished between access and intermediate links), the node availability, and the system availability that is provided.

C. LOCATION SELECTION FOR FOG NODE DEPLOYMENT

As a concrete example from literature how to use the SI to compare the scalability of different solution approaches, the location selection for fog node deployment and routing in SDN-based wireless networks for IoT systems is investigated [26]. The interesting aspect of that use case is that we need to consider two different target measures, which are the average end-to-end delay as well as the energy consumption. The scalability of both aspects is thereby investigated for three different solution approaches. The fog computing architecture involves relocating services such as computing, processing, and storage from the centralized cloud to the network edge or nearby devices, where these services are deployed. [26] introduces a novel approach called "Scalable and Optimal Near-Sighted Location Selection" (SOSW) to address two key issues in fog computing architecture with software-defined networking (SDN). (i) Fog nodes are strategically deployed for optimal performance. (ii) A new heuristic-based traffic engineering algorithm computes the best paths for data flows based on constraints like end-to-end delay and link utilization, which are deployed in an SDN environment. The goal is to minimize both the energy consumption and end-to-end delay of IoT devices during task offloading.

The more fog nodes are deployed, the better the performance measures. Nevertheless, increasing the number of fog nodes comes with a significant cost, making it preferable to achieve optimal performance using a limited number of these nodes. For a given number of IoT nodes, the scalability of the SOSW solution is therefore investigated in terms of end-to-end delay, averaged over the number of tasks to be offloaded, and the energy consumption of the IoT nodes. Therefore, the scalability index is computed for both target measures. As parameter, the number of tasks is considered, which is varied from 100 to 1000. Equal weights are selected for the entire parameter range. As reference function, we consider linear scalability with the suggested linear functions in Figure 18.

Figure 18a shows that the suggested heuristic even leads to a better scalability index than the reference system with $SI = 1.03 > 1$. However, the shape of the SOSW curve is not following a linear curve. It has to be noted that the scalability index only considers the parameter range of interest with the corresponding weights. Hence, for computing the SI, the parameter range of interest must be specified and the corresponding target measure (here: end-to-end delay) must be known. Otherwise, some prediction curves of the end-to-end delay need to be assumed to compute the SI for larger parameter range and a larger number of tasks. A power law prediction of the delay curve is provided in Figure 18b.



(a) Average end-to-end delay.

(b) Average end-to-end delay with prediction.

(c) Energy consumption.

FIGURE 18. Example of location selection for fog node deployment and routing in SDN-based wireless networks for IoT systems. The data is taken from [26]. In addition, the SI is computed for different approaches with respect to a linear reference system. As target measures, energy consumption as well as the average end-to-end delay are considered. The parameter range is equally weighted.

Now, we want to investigate the parameter range from 100 to 1200 tasks. While the measurement points are taken from 100 to 1200 tasks, the predicted delays are used for 1100 and 1200 tasks. Now, the SI is only 0.74 indicating that the system is not scaling linearly over the parameter range up to 1200 tasks. This example again demonstrates the importance of setting the relevant parameter range for the scalability investigation.

Figure 18a also shows two other approaches from literature [27]: (a) the random offloading random path (RORP) model, (b) the delay-aware greedy path (DGP). Their scalability index is provided in the legend in relation to the linear reference system. Considering the scalability of the delay, the SOSW heuristic demonstrates significant improvements over the two existing methods. Hence, we may conclude that SOSW scales linearly and better than DGP and RORP, which have SI values smaller than the SI of SOSW.

We may also directly compute the SI when using another system as reference. Each row in Table 3 computes the scalability index for different reference systems (as indicated in the columns). For example, the SI of SOSW with DGP as reference is $SI_{\text{SOSW,DGP}}$=1.20. The SI of DGP with SOSW as reference system is the inverse: $SI_{\text{DGP,SOSW}}=1/SI_{\text{SOSW,DGP}}=1/1.20=0.83$. However, this is not required, since we may simply compare the SI of SOSW and DGP in comparison to the linear reference system: $SI_{\text{SOSW,lin.}}=1.03$ and $SI_{\text{DGP,lin.}}=0.85$. The SI shows that SOSW outperforms DGP. The corresponding SI follows directly: $SI_{\text{SOSW,DGP}}=SI_{\text{SOSW,lin.}}/SI_{\text{DGP,lin.}}=1.20$. In practice, we are typically interested in linear scalability, i.e., a linear reference function, and then a ranking of the methods. Hence, the computation of the SI as indicated in the legend of Figure 18a is sufficient.

TABLE 3. Fog computing example: The SI is computed for any combination of the different approaches (SOSW, DGP, RORP) and the linear reference.

SI wrt. delay	*using reference system* SOSW	 DGP	 RORP	 linear	 rank
SOSW	1.00	1.20	1.30	1.03	1
DGP	0.83	1.00	1.08	0.85	2
RORP	0.77	0.93	1.00	0.79	3
linear	0.97	1.17	1.27	1.00	*ref.*
SI wrt. energy	*using reference system* SOSW	 DGP	 RORP	 linear	 rank
SOSW	1.00	1.13	1.26	0.84	1
DGP	0.89	1.00	1.11	0.75	2
RORP	0.80	0.90	1.00	0.67	3
linear	1.19	1.34	1.49	1.00	*ref.*

Similarly, other target measures may be investigated in terms of scalability. Figure 18c shows the energy consumption of the IoT nodes depending on the number of tasks for the different approaches and a linear reference system. Now, the scalability of the SOSW is not linear when compared with an appropriate linear system (SI<1). This is caused by a certain amount of energy consumption in idle mode or when the load in terms of number of tasks is low. Hence, we get different conclusions when using different target measures, as already discussed in Section V-A4 for the IoT load balancer. In the fog computing example, the energy and delay scalability lead to the same ranking of the approaches, see Table 3. The next example will show that also the ranking may change itself depending on the measure of interest.

D. NUMBER OF SUPPORTED FLOWS IN TSN

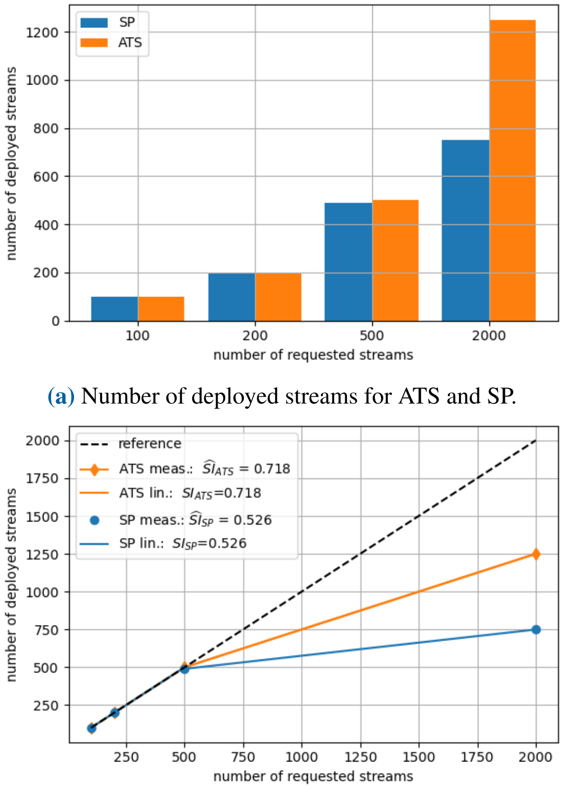
As a final use case, time sensitive networking (TSN) is considered with two different mechanisms, that are the IEEE 802.1Q strict priority transmission selection algorithm (SP) [28] and the IEEE 802.1Qcr Asynchronous Traffic Shaping (ATS) mechanism [29]. The scalability analysis is based on measurement results, for which the parameter values are not evenly spaced. Such a situation may happen in practice, when comparing the scalability of an own system with another system, where only limited measurement results are available.

An IEEE 802.1Q Strict Priority switch supports different traffic classes and uses a FIFO transmission selection algorithm for all data frames within the same queue (i.e., traffic class). Reference [30] shows that deterministic latency with priority queuing is feasible without the need for network-wide information or reshaping and timed gates in the SP forwarding devices. The mechanism relies on a resource reservation process that communicates necessary information for the resource reservation of each stream along its path, e.g., based on the resource allocation protocol (RAP) developed in IEEE 802.1Qdd [31] which provides stream reservation and quality of service capabilities.

In contrast, the ATS mechanism deploys a per-hop reshaping of streams based on IEEE standard draft P802.1Qcr in an ATS switch. To be more precise, ATS applies per-stream leaky bucket shaping with interleaved queuing to keep the burstiness of streams low. The ATS and the SP mechanism are proven to guarantee latency bounds with a proper reservation protocol [30], [32].

An evaluation of the impact of SP and ATS on stream reservations is provided in [30]. In the experiments, a varying number of streams is deployed with different traffic characteristics for low-priority and high-priority traffic. A new stream attempts to reserve network resources, but if the new stream reservation leads to violations of any delay guarantees for any stream in the network, the new stream is declined. Otherwise, the new stream is accepted.

Figure 19a shows the number of deployed streams depending on the number of requested stream reservations. The study involves conducting multiple experiments with various configurations of attempted reservations. These configurations range from 100 to 2000 streams, with four different steps in between. For each of these configurations, 20 repetitions are conducted, each time using random stream traffic specifications. Figure 19a shows the average numbers of accepted reservations for ATS and SP. It can be seen that ATS could accept more streams than SP and is therefore preferred. However, an ATS switch is more complex than an SP switch, which is a lightweight solution that is available in current switches. Note that SP and ATS guarantee the upper delay bounds for the traffic of the different classes (which are low- and high-priority traffic in the numerical example).



(a) Number of deployed streams for ATS and SP.

(b) Scalability index for ATS and SP, respectively.

FIGURE 19. Time sensitive networking (TSN) using the strict priority (SP) and the asynchronous traffic shaping (ATS) mechanism, respectively. The experimental results are taken from [30] for two different traffic classes with the following delay guarantees: $\delta_{\text{high}} = 2\,\text{ms}$, $\delta_{\text{low}} = 8\,\text{ms}$.

The question arises how to compute the scalability index. We use as a reference system the optimal system, i.e., the number $h(x)$ of accepted and deployed streams is the number x of requested streams: $h(x) = x$. The parameter range x is from 100 to 2000. The measurement points are discrete and not evenly spaced. Instead, we have the measurement values for the parameter $x \in \{100, 200, 500, 2000\}$.

In general, we may have n measurement values and therefore tuples (x_i, y_i) for $i = 1, \ldots, n$. Then, the SI can be computed by a piecewise linear function between the measurement points. The area under that function is

$$\begin{aligned}\widehat{A}_y &= \sum_{i=2}^{n} (x_i - x_{i-1}) y_{i-1} + \frac{1}{2}(x_i - x_{i-1})(y_i - y_{i-1}) \\ &= \frac{1}{2} \sum_{i=2}^{n} (x_i - x_{i-1})(y_i + y_{i-1}) \end{aligned} \tag{35}$$

for the parameter range $x \in \{100, 2000\}$. In our example, the area A_h under the reference function $h(x) = x$ is

$$A_h = \frac{1}{2} x_n y_n - \frac{1}{2} x_0 y_0 \tag{36}$$

and the SI is

$$\widehat{SI} = \widehat{A}_y / A_h \,. \tag{37}$$

Note that a piecewise linear function fitting $f(x)$ leads to the same area

$$A_y = \sum_{x=x_1}^{x_n} f(x) = \widehat{A}_y \tag{38}$$

and consequently the same SI.

Figure 19b visualizes the piecewise linear function between the measurement points and provides the scalability index using the computation of the area in Eq.(36) and Eq.(38). The scalability of ATS is $SI_{ATS}/SI_{SP} = 1.36$ times better than the scalability of SP. The relation of the SI values simply means to compare the integral measurements of the two systems, i.e., without explicitly defining a reference system like the optimal system. This is a very useful feature of the SI in practice. In general, there may be system $\mathcal{F}$ and $\mathcal{G}$ with the corresponding integral measurement F and G, respectively. Then, the scalability index SI_F^H and SI_F^H can be computed based on a reference system $\mathcal{H}$ and integral measurement H.

$$SI_F^H = \left(\frac{F}{H}\right)^\gamma \quad SI_G^H = \left(\frac{G}{H}\right)^\gamma \tag{39}$$

The relation of these two SI values is however the SI of the $\mathcal{F}$ using $\mathcal{G}$ as reference system.

$$SI_F^G = SI_F^H / SI_G^H = \left(\frac{F}{H}\right)^\gamma / \left(\frac{G}{H}\right)^\gamma = \left(\frac{F}{G}\right)^\gamma \tag{40}$$

Thus, the scalability improvement or deterioration of $\mathcal{F}$ in relation to $\mathcal{G}$ is quantified by the SI relation.

VI. CONCLUSION AND DISCUSSIONS

Scalability is often mentioned in literature, but a stringent definition is missing. In particular, there is no general scalability assessment which clearly indicates whether a system scales or not or whether a system scales better than another. Furthermore, it is often unclear what is meant by statements like *"A system scales."* To this end, we survey literature and differentiate scalability from aspects like performance, efficiency, elasticity.

A. KEY CONTRIBUTIONS

The key contribution of this paper is the definition of a *"general scalability index"* that generalizes existing approaches from the literature, which are a special case of ours. Our general framework allows quantifying whether a system or communication network is scaling in comparison to a reference system. This also allows, e.g., a comparison with an optimal system or benchmarking systems and ranking them. With our numerical results, we demonstrate the use of the scalability index and emphasize the relevance of the key components of the scalability index, which are as follows.

(1) The *system function* (or target measure function) quantifies the *target measure* of interest for the system, depending on a certain parameter. The researcher needs to link the scalability question about an appropriate target measure. Diverse target measures can yield varying outcomes regarding the scalability of a system in practical scenarios. Defining the target measure, such as considering SLAs, is crucial and should be prioritized as the initial step in assessing system scalability. By modifying the system function, it becomes possible to analyze additional aspects, such as stochastic scalability. This opens up new opportunities for examining the system under probabilistic scenarios.

(2) A *reference system* is used for comparison with the system under test. The reference system serves as a benchmark for comparing and evaluating two or more systems. It provides a standard against which the target measure of the systems can be measured. Often, the ideal system serves as a desirable benchmark; however, it may not always be achievable or known in real-world applications. In practice, a typical question is to investigate if the system scales linearly. Then, the reference function is a linear function. Still, the slope and offset of such a linear function needs to be determined.

(3) The analysis of scalability needs to consider all relevant parameter settings. Scalability means not just the ability to operate, but to operate efficiently and with adequate quality of service, over the given range of configurations. Thus, the scalability index must consider the *parameter range of interest* to draw conclusions.

(4) By weighting the importance or relevance of a parameter setting or configuration, the quantification of scalability results in an integral measurement of the (weighted) target measure over the entire parameter range.

B. LESSONS LEARNED FROM THE USE CASES

The use cases for demonstrating the SI are an IoT load balancer, availability in communication systems, node selection in fog computing, and benchmarking of TSN mechanisms. They differ in the parameter range (continuous load of IoT balancer, discrete number of nodes, number of fog computing tasks, number of requested TSN streams), the goodness indicator (response times, availability, costs, delay, energy consumption, deployed TSN stream), and the focus of the scalability analysis (functional and stochastic scalability, structure of networks, combination of availability and costs, benchmarking of existing fog computing or TSN mechanisms).

The use case of availability in communication systems demonstrates that the scalability index can be useful to gain insight in how to structure the network when it is expected that the number of nodes will grow. The scalability of the system differs depending on the target measure (availability, costs), which are mutually contradicting. Therefore, it is shown how to combine different target measures with the power metric or using costs as weights. It is important to understand that the scalability index and the scalability analysis change depending on the target measure of interest. In our example, the parallel system scales wrt. availability, but not wrt. costs.

For the use case of an IoT load balancer, our results show how the scalability index helps to quantify aspects like "economies of scale", i.e., larger-scale operation has advantages over smaller ones. Bundling the servers increases the scalability, as quantified by the scalability index. It also allows quantifying how much better it is to have a single high-performance server or smaller servers bundled.

The fog computing use case shows how to benchmark different mechanisms concerning their scalability. Thereby, the results show linear scalability of one approach. This means that the system behavior in terms of delay as target measure of a linear system is similar to the mechanism under investigation. The performance curve shows however that the curve is following an exponential increase instead of a linear one. It is important to understand the scalability index needs a defined parameter range of interest. If it is expected that the system may also need to cope with larger parameter values (number of tasks) in the future, then the performance curve needs to be extended to the entire (future) parameter range. This may be done based on predictions.

We clearly want to mention that the scalability index is computed for a system where the behavior is known. This means the target measure function must be known or estimated for future developments. This is the required input for the integral measurement. In other words, the scalability index of a deployed system is constant. If the system is changed, after it is built (e.g., by adding more resources), then the scalability index must be re-computed, since this would change the target measure function. If the system behavior is unknown, then the scalability index cannot be computed.

Finally, the TSN use case shows how to deal with unequally spaced measurements and limited configurations under test. Piece wise linear functions may be used, if no more system knowledge and more fine-grained target measure-parameter curves are available. For benchmarking different mechanisms, we may use the optimal reference system. Then, the SI relates the scalability of a mechanism to the optimal system and the absolute value of the SI provides meaningful insights. Similarly, linear systems (i.e., linear target measure functions) as reference allow quantifying to which degree a system linearly scales.

The comparison of two mechanisms means to compute the SI for one system in relation to another system. The relation of the two SI values shows how much better a system scales in comparison to another.

C. PRACTICAL GUIDELINES AND LIMITATIONS

For the scalability analysis of a system, there are a couple of aspects to be considered in practice. We provide some practical guidelines by revisiting the ingredients of the SI.

The target measure: The system function quantifies the system behavior depending on a certain parameter and the desired target measure of interest. This system function is the input for the integral measurement of the SI. In practice, several target functions may be of interest, e.g., delays or energy consumption. Then, it is recommended to compute the SI for all the interesting target functions. This gives a detailed understanding how the system scales in different dimensions. If there is the possibility to combine some target measures into a single-dimensional utility function, then the SI regarding that utility function as target measure function may lead to different results and conclusions. Kleinrock's power metric or appropriate weighting functions, e.g., for costs, are recommended and may give advice, which systems are scaling better in practice.

The parameter range: The scalability analysis requires a defined parameter range to be investigated. The parameter range may also be unbounded, but the system function, which is the target measure depending on a certain parameter over the entire range, is the necessary input for the SI computation. In practice, only limited information, how the system behaves, may be known for extended parameter ranges, e.g., future system size and more nodes in a future system. The SI computation must get the system function as input. If the system behavior is not known for the parameter range of interest, then the SI cannot be computed. However, in practice, interpolation of measurement points, e.g., piecewise linear functions, as well as prediction of future system behavior are possibilities how to obtain the system function.

The weighting function: The weighting function gives a powerful way how to include additional aspects like the importance of parameter / configuration settings, the probability for such settings, or the costs resulting from such settings. The weighting function requires deep expert knowledge of the system, e.g., costs may be difficult to measure or to estimate in practical environments. If not known, all parameter settings should be treated equally, i.e., same weight.

The reference system: : If the optimal system behavior is known, the optimal system should be defined as a reference system. The SI shows then how far a system is away from the optimal system. In practice, the optimal system is often unknown or too complex to be derived. Linear reference functions are a proper mean to test for linear scalability, which is a key comparison in practice. Still, a proper linear function needs to be considered, see Section III-C. However, knowledge in the domain or expert knowledge of the system allows defining the slope as well as the constant offset of the linear function. The offset may be derived from the system in idle mode. The slope may reflect the desired or acceptable behavior of the system.

However, we want to emphasize that the scalability analysis provides a framework to compare two different systems $\mathcal{F}$ and $\mathcal{G}$. Then, the scalability index can be computed with a linear system as reference. Then resulting SI values of $\mathcal{F}$ and $\mathcal{G}$ provide then a ranking, independent of which reference system is used. The relation of the two SI values shows the scalability improvement or deterioration of $\mathcal{F}$ in relation to $\mathcal{G}$, see the TSN example in Section V-D. For benchmarking of two or more systems, the SI can be directly computed by just using the system functions of $\mathcal{F}$ and $\mathcal{G}$, see also the fog computing example and Table 3. Thus, benchmarking of two systems does not require a reference system – and is the primary target of our SI framework. Probably most often used in practice is the scalability comparison of a system with another similar system. Only in case there is no comparison targeted, we use a reference system for the computation of SI.

In practice, additional issues may arise. To account for situations where the system may not function correctly with specific parameter settings or configurations, it is crucial to incorporate this information into the target function. Additionally, the relevance or importance of such scenarios can be adjusted using a weighting function. This ensures that the scalability evaluation captures the impact of problematic parameter settings or configurations appropriately.

If we compute the scalability index of a deployed system, then the SI is constant. If the system is changed, after it is built (e.g., by adding more resources), then the scalability index must be re-computed, since the system function is changed. If that system function is not known, the SI cannot be computed. As discussed above, predictions or interpolations may be useful in that case. This also means that the SI will remain unchanged, i.e., constant, when a system is not changed.

In general, the SI measures the potential of a system to scale. If the system scales up/down over time, then this must be captured in the system function. Hence, aspects like elasticity are part of a scalability analysis.

Knowing how the system works and the system functions are a prerequisite for calculating the SI. If we have a black box system at hand, then we need to learn the system behavior to derive the system function. This may be done via experiments in a test bed or via simulations, or alternatively, we can do some stress tests with the running system if possible. In practice, we typically want to compare a system with another similar system. If no other system for comparison is available, we can compare to a linear or optimal system.

REFERENCES

[1] A. Al-Said Ahmad and P. Andras, "Scalability analysis comparisons of cloud-based software services," *J. Cloud Comput.*, vol. 8, no. 1, pp. 1–17, Dec. 2019.

[2] S. Lehrig, H. Eikerling, and S. Becker, "Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics," in *Proc. 11th Int. ACM SIGSOFT Conf. Quality Softw. Architectures (QoSA)*, May 2015, pp. 83–92.

[3] P. Tran-Gia and T. Hoßfeld, *Performance Modeling and Analysis of Communication Networks*. Würzburg, Germany: Würzburg University Press, 2021. [Online]. Available: https://modeling.systems

[4] P. Jogalekar and M. Woodside, "Evaluating the scalability of distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 6, pp. 589–603, Jun. 2000.

[5] N. R. Herbst, S. Kounev, and R. H. Reussner, "Elasticity in cloud computing: What it is, and what it is not," in *Proc. ICAC*, vol. 13, 2013, pp. 23–27.

[6] J. Perez, C. Germain-Renaud, B. Kégl, and C. Loomis, "Responsive elastic computing," in *Proc. 6th Int. Conf. Ind. Session Grids Meets Autonomic Comput.*, Jun. 2009, pp. 55–64.

[7] D. M. Gutierrez-Estevez, M. Gramaglia, A. D. Domenico, N. D. Pietro, S. Khatibi, K. Shah, D. Tsolkas, P. Arnold, and P. Serrano, "The path towards resource elasticity for 5G network architecture," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2018, pp. 214–219.

[8] M. He, A. M. Alba, A. Basta, A. Blenk, and W. Kellerer, "Flexibility in softwarized networks: Classifications and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2600–2636, 3rd Quart., 2019.

[9] P. Babarczi, M. Klügel, A. Martínez Alba, M. He, J. Zerwas, P. Kalmbach, A. Blenk, and W. Kellerer, "A mathematical framework for measuring network flexibility," *Comput. Commun.*, vol. 164, pp. 13–24, Dec. 2020.

[10] O. Hohlfeld, J. Kempf, M. Reisslein, S. Schmid, and N. Shah, "Guest editorial scalability issues and solutions for software defined networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 12, pp. 2595–2602, Dec. 2018.

[11] M. Karakus and A. Durresi, "A scalability metric for control planes in software defined networks (SDNs)," in *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2016, pp. 282–289.

[12] R. Steinmetz, I. Stavrakakis, C. E. Rothenberg, and B. Koldehofe, "Adaptive and scalable communication networks [scanning the issue]," *Proc. IEEE*, vol. 107, no. 4, pp. 635–638, Apr. 2019.

[13] Y. Li, D. Ou, X. Zhou, C. Jiang, and C. Cérin, "Scalability and performance analysis of BDPS in clouds," *Computing*, vol. 104, pp. 1–36, Feb. 2022.

[14] X.-H. Sun and D. T. Rover, "Scalability of parallel algorithm-machine combinations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 5, no. 6, pp. 599–613, Jun. 1994.

[15] P. Tran-Gia and A. Binzenhöfer, "On the stochastic scalability of information sharing platforms," in *Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*. Cham, Switzerland: Springer, 2006, pp. 11–27.

[16] E. J. Weyuker and A. Avritzer, "A metric to predict software scalability," in *Proc. 8th IEEE Symp. Softw. Metrics*, Jul. 2002, pp. 152–158.

[17] A. Y. Grama, A. Gupta, and V. Kumar, "Isoefficiency: Measuring the scalability of parallel algorithms and architectures," *IEEE Parallel Distrib. Technol., Syst. Appl.*, vol. 1, no. 3, pp. 12–21, Aug. 1993.

[18] J. L. Bosque, O. D. Robles, P. Toharia, and L. Pastor, "H-isoefficiency: Scalability metric for heterogeneous systems," in *Proc. 10th Int. Conf. Comput. Math. Methods Sci. Eng.*, 2010, pp. 1–11.

[19] S. Henning and W. Hasselbring, "How to measure scalability of distributed stream processing engines?" in *Proc. Companion ACM/SPEC Int. Conf. Perform. Eng.*, Apr. 2021, pp. 85–88.

[20] W.-T. Tsai, Y. Huang, and Q. Shao, "Testing the scalability of SaaS applications," in *Proc. IEEE Int. Conf. Service-Oriented Comput. Appl. (SOCA)*, Dec. 2011, pp. 1–4.

[21] A. Avritzer, V. Ferme, A. Janes, B. Russo, A. V. Hoorn, H. Schulz, D. Menasché, and V. Rufino, "Scalability assessment of microservice architecture deployment configurations: A domain-based approach leveraging operational profiles and load tests," *J. Syst. Softw.*, vol. 165, Jul. 2020, Art. no. 110564.

[22] S. Henning and W. Hasselbring, "A configurable method for benchmarking scalability of cloud-native applications," *Empirical Softw. Eng.*, vol. 27, no. 6, p. 143, Nov. 2022.

[23] P. Jogalekar and C. Woodside, "A scalability metric for distributed computing applications in telecommunications," *Teletraffic Sci. Eng.*, vol. 2, pp. 101–110, 1997. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1388343797800169

[24] L. Kleinrock, "Internet congestion control using the power metric: Keep the pipe just full, but no fuller," *Ad Hoc Netw.*, vol. 80, pp. 142–157, Nov. 2018.

[25] F. Metzger, T. Hoßfeld, A. Bauer, S. Kounev, and P. E. Heegaard, "Modeling of aggregated IoT traffic and its application to an IoT cloud," *Proc. IEEE*, vol. 107, no. 4, pp. 679–694, Apr. 2019.

[26] M. Ibrar, L. Wang, G.-M. Muntean, N. Shah, A. Akbar, and K. I. Qureshi, "SOSW: Scalable and optimal nearsighted location selection for fog node deployment and routing in SDN-based wireless networks for IoT systems," *Ann. Telecommun.*, vol. 76, pp. 331–341, Apr. 2021.

[27] M. Chen and Y. Hao, "Task offloading for mobile edge computing in software defined ultra-dense network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 587–597, Mar. 2018.

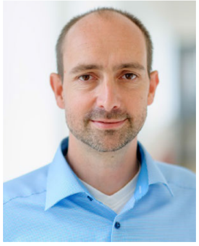
[28] *IEEE Standard for Local and Metropolitan Area Network—Bridges and Bridged Networks*, Standard IEEE 802.1Q-2018, IEEE 802.1 Working Group, IEEE Standards Association, 2018.

[29] *Draft Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment: Asynchronous Traffic Shaping*, Standard IEEE P802.1Qcr/D2.0, IEEE Standards Association, 2019.

[30] A. Grigorjew, F. Metzger, T. Hoßfeld, J. Specht, F.-J. Götz, F. Chen, and J. Schmitt, "Bounded latency with bridge-local stream reservation and strict priority queuing," in *Proc. 11th Int. Conf. Netw. Future (NoF)*, Oct. 2020, pp. 55–63.

[31] *IEEE 802.1Qdd: Draft Standard for Local and Metropolitan Area Networks—Bridges and Bridged Networks—Amendment: Resource Allocation Protocol*, IEEE 802.1 Working Group, IEEE Standards Association, Piscataway, NJ, USA, 2023.

[32] J. Specht and S. Samii, "Urgency-based scheduler for time-sensitive switched Ethernet networks," in *Proc. 28th Euromicro Conf. Real-Time Syst. (ECRTS)*, Jul. 2016, pp. 75–85.



TOBIAS HOSSFELD (Senior Member, IEEE) was the Head of the Chair Modeling of Adaptive Systems, University of Duisburg-Essen, Germany, from 2014 to 2018. He has been a Full Professor and the Head of the Chair of Communication Networks, University of Würzburg, Germany, since 2018. He is a member of the editorial board of IEEE Communications Surveys and Tutorials, ACM SIGMM Records, and *Quality and User Experience* (Springer).

POUL E. HEEGAARD (Senior Member, IEEE) was a Senior Scientist with SINTEF Digital, from 1989 to 1999, and Telenor R&I, from 1999 to 2009. He is currently a Full Professor with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology (NTNU), where he is also the Head of the Department of Information Security and Communication Technology and the Head of the Networking Research Group.

WOLFGANG KELLERER (Senior Member, IEEE) is currently a Full Professor with the Technical University of Munich (TUM), where he is also heading the Chair of Communication Networks, Department of Electrical and Computer Engineering. Before that, he was with the NTT DOCOMO's European Research Laboratories, for over ten years. He currently serves as an Associate Editor for IEEE Transactions on Network and Service Management and an Area Editor for Network Virtualization and IEEE Communications Surveys and Tutorials.