Using G/G/m-Models for Multi-Server and Mainframe Capacity Planning

Müller-Clostermann, Bruno

In: ICB Research Reports - Forschungsberichte des ICB / 2007

DOI: https://doi.org/10.17185/duepublico/47152

URN: urn:nbn:de:hbz:464-20180925-070747-3

Link: https://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet?id=47152

Source: ICB-Research Report No. 16, May 2007

ICB
Institut für Informatik und Wirtschaftsinformatik

Bruno Müller-Clostermann

Using G/G/m-Models for Multi-Server and Mainframe Capacity Planning

16

ICB-RESEARCH REPORT

UNIVERSITÄT DUISBURG ESSEN

ICB-Research Report No.16

May 2007

Die Forschungsberichte des Instituts für Informatik und Wirtschaftsinformatik dienen der Darstellung vorläufiger Ergebnisse, die i. d. R. noch für spätere Veröffentlichungen überarbeitet werden. Die Autoren sind deshalb für kritische Hinweise dankbar.

The ICB Research Reports comprise preliminary results which will usually be revised for subsequent publications. Critical comments would be appreciated by the authors.

Authors' Address:

Bruno Müller-Clostermann

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Schützenbahn 70
D-45117 Essen
Germany

bmc@icb.uni-due.de

Managing Assistant and Contact:

Jürgen Jung

Institut für Informatik und
Wirtschaftsinformatik (ICB)
Universität Duisburg-Essen
Universitätsstr. 9
45141 Essen
Germany

Email: icb@uni-duisburg-essen.de

ISSN 1860-2770

Abstract

Results from classic queueing theory are shown to be useful for capacity planning of large-scale multi-server systems. In the field of mainframe systems on-line monitoring and dynamic workload management care for the fulfilment of short-term performance goals with respect to service class specific execution velocity, wait time and wait time percentiles. The same performance measures are useful for long-term capacity management. On the basis of well-known approximate results for G/G/m queueing models we derive analytical formulas for the calculation of (global) steady state performance measures. We illustrate the usage of these formulas and show their application in the field of mainframe capacity planning.

Keywords: Capacity Planning, Mainframes, Performance, Queueing Models, Velocity, Wait Time

Contents

Terms and abbreviations

AC-formula	G/G/m-formula due to Allen/Cunneen
Contention	IBM term for percentage of waiting units (cf. velocity)
CEC	Central Electronic Complex
C-Formula (due to Erlang)	computes the probability that a system is full
COD	Capacity On Demand
CP	Central Processor
G/G/m	Kendall-Notion for general multi-server queueing models
KK-formula	G/G/m-formula due to Kingman/Köllerström
KLB-formula	G/G/m-formula due to Krämer/Langenbach-Belz
LPAR	Logical Partition
MIPS	Million Instructions Per Second
PU	Processing Unit
RMF	Resource Measurement Facility (Monitoring Software)
QOS	Quality of Service
Utilization	$\rho_m = \lambda / m\mu$
Velocity	IBM term for contention, also named “Excecution Velocity”
WLM	Workload Manager
wrt	with respect to
z/OS, z/VM, z/VSE	Mainframe operating systems

Formal notions

C_A	Coefficient of variation of inter arrival times
C_S	Coefficient of variation of service time
E[W]	Expected wait time, also denoted as $\overline{W}$
$E[W_m]$	Expected wait time, in case of m processors
E[S]	Expected service time
E[V]=100·(1-P[W>0]) %	Average Velocity
F	Slowdown factor
F_m	Slowdown factor, in case of m processors
m	Number of processors
μ	Processor speed, service rate
p, P[ ]	Probability
P[W>0]	Delay probability, also called wait probability
$\rho_m = \lambda / m\mu$	Utilization (in case of m processors)
π_0	Probability for an empty (idle) system
σ_X, σ_X^2	Standard deviation and variance of random variable *X*
S	Service time
V	Velocity
W	Wait time
$\overline{W}, \overline{W}_m$	Expected wait time
z	Overall workload variability, $z = C_A^2 + C_S^2$

1 Introduction

We contribute to a capacity planning methodology for multi-server systems using a macroscopic view of a complete IT installation. A macroscopic description consists of a rather coarse workload model obtained from global measurements of system utilization, and a system description, which is focussed on the number of processors and the processing capacity according to scaling tables for the system under consideration. Of course, the underlying assumptions lead to a very abstract model on system level. Moreover, the results hold for long-term “steady-state” behaviour and do not distinguish between service classes or even individual tasks. Such a simplified view on a complex and distributed system matches the high level view of a capacity planner.

Using queueing theoretic results for G/G/m-models it is possible to calculate execution velocity, average wait times and wait time percentiles. The developed formulas can be parameterized with low effort and solved easily; hence an integration of these models into the capacity planning process will allow the evaluation of many different model scenarios. In particular, it is possible to consider different workload forecasts and their allocation to a set of systems.

With respect to real world applications we focus on nowadays mainframe systems where the workload is given by a set of LPARs (logical partitions), which are running on several systems. As a consequence capacity planning has to cope with the management of multiple workloads (here: LPARs) and their allocation to a set of resources (here: CECs). Measurement data are normally available on a rather macroscopic level, e.g. the 15 minutes averages of consumed MIPS (Million Instructions Per Second) over a certain time interval of some weeks or months. Also the feasible system configurations are considered on a coarse level, i.e. we solely include the number and type of processors in combination with application specific benchmark results in form of a scaling table. For this macroscopic description of workload and systems we employ the class of G/G/m-Models, which are relatively simple, but not too simple! They provide a level of detail that matches the macroscopic view on the overall system.

For related work on capacity planning we refer to standard textbooks of Daniel Menascé with different co-authors, e.g. [MeAl02, MeAD04], the books of Neil J. Gunther for the practitioner [Gunt00, Gunt07], and the general source in capacity management [MüCl01].

The rest of the paper is organized as follows. Chapter 2 introduces briefly into the context and gives a short overview of mainframe systems, their workload and the established monitoring techniques. In Chapter 3 analytical G/G/m-formulas are used to derive expressions for execution velocity, average wait time and wait time percentiles. The meaning of these measures is illustrated by a set of diagrams. Chapter 4 finally demonstrates a possible application of these results to mainframe capacity planning. Chapter 5 concludes this report and gives an outlook.

2 Multi-server and mainframe computing

2.1 Mainframe architecture

Mainframes belong to the most advanced high performance systems. They provide processing power for business-critical, high volume online transaction processing and database applications. We briefly sketch some typical features of a current mainframe system, the z Series of IBM. The z9 servers allow up to 54 processors (called CPs or PUs) which are organized on 1 to 4 books in a single mainframe. The major components of such a system are the ability to add and activate processors, memory, and I/O connectivity while the server is running. There is a fault-tolerant communication ring to interconnect books; processors on other books are available for activation to maximize the recovery capability in the event of individual processor failures, and many other features. Similar redundancy techniques exist with respect to memory, I/O, secondary storage, power and cooling. Various operating systems are supported such as Linux, z/OS, z/VM and z/VSE**.** For an overview cf. [FCSM04].

2.2 Workload and logical partitions

Mainframe workload is given by transactional workload and batch workload, which are both generated by so-called logical partitions sometimes abbreviated as LPARs. A LPAR is a virtualized computing environment, which abstracts from all physical devices. LPARs are equivalent to physically separate servers with no connections. The typical number of logical partitions running on a single system is 10 to 60 depending on the type of mainframe and on the requested processing capacity of the partitions.

The processing capacity requested by a logical partition is described by the amount of MIPS (Million Instructions Per Second), which is necessary to run all applications with satisfactory service levels. The amount of MIPS to be provided is usually fixed by a contract between customer and provider. Since mainframe workload is relatively homogeneous, this simplified view of processing capacity quantified as MIPS is widely accepted as a useful performance measure. The installed (available) as well as the requested processing power are both described in MIPS.

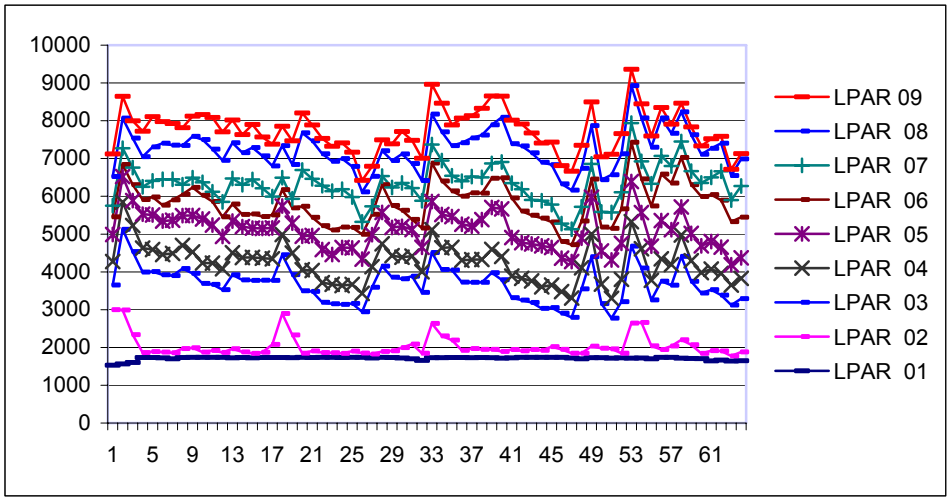

Fig. 1: MIPS consumptions of 9 logical partitions over two days (only busy hours)

Figure 1 displays the measured MIPS-rates consisting of average resource requests of 9 logical partitions over 64 time intervals each of 15 minutes length. These data cover the busy hours of two days (7.00 a.m. until 3.00 p.m.). As an example a system with 24 processing units and an overall processing power of approximately 9500 MIPS may be sufficient to process such a workload. This MIPS-value has been taken from vendor information for an IBM Z9-109-System with 24 PUs, where a single PU is declared to provide 580 MIPS and a high end system with 54 PUs has approximately 18000 MIPS.

Historical data of MIPS-consumptions are a rather coarse measure; nevertheless, they are an important resource for workload management and capacity planning.

2.3 Monitoring and measurements

Performance management metrics for mainframe systems include throughput, utilization, response times and execution velocity. Some of these metrics are measured online by monitoring systems, like the IBM Workload Manager (WLM), which is a component of the z/OS operating system. The workload processed on the z/OS operating system is classified by the WLM in distinct service classes. For each service class a set of performance goals – also called service levels – may be specified, which are used by the WLM to manage dynamically the complete workload. The WLM constantly monitors the system and adapts processing and allocation of resources such as CPU-time to meet the performance goals [IBM05].

Monitoring, accounting and reporting are essential activities for the operation of mainframe systems. The most important example is the Resource Measurement Facility (RMF), which supports a set of functionalities and capabilities to measure and store collected performance data. RMF collects data concerning workload, resource utilization and other system activities. Typically processor utilization, I/O device activity and response times are determined during measurement periods of a specified length and are aggregated into performance reports. The fulfilment of specified performance goals is also monitored for each service class. Workload management with the WLM includes the definition of performance goals and the setting of an „importance” (an IBM term) to each goal. The WLM decides how much resources, such as CPU and storage, should be provided to meet the performance goals.

RMF data sampled over many days and for many partitions define time series for each logical partition. Of course these time series exhibit a certain stochastic behaviour that can be investigated by different analysis techniques and may be used to forecast future workload requests.

Figure 2 shows workload behaviours derived from empirical data as an example for the available workload description. The available capacity for the logical partitions is displayed by the horizontal dotted line; the limit is defined by a contract and is here set to 1600 MIPS in both cases. The curves show the requested (and consumed) capacity for a set of partitions over one month. The variations due to daily and weekly seasonal effects are obvious.

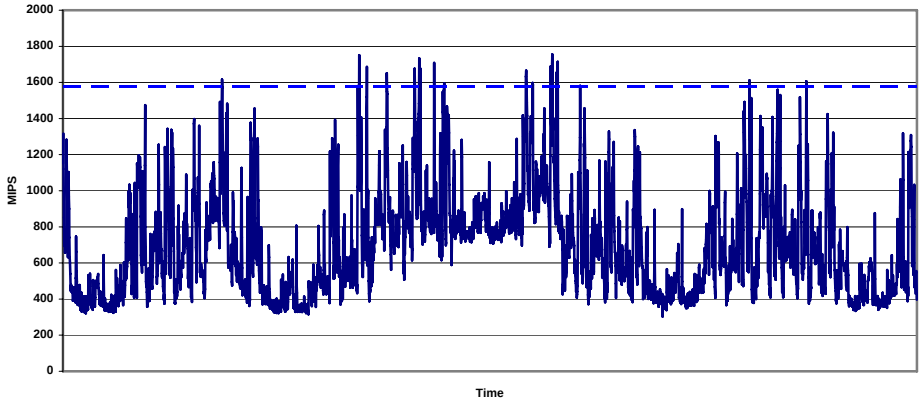

Fig. 2a: Requested versus installed capacity over 4 weeks (Example a)

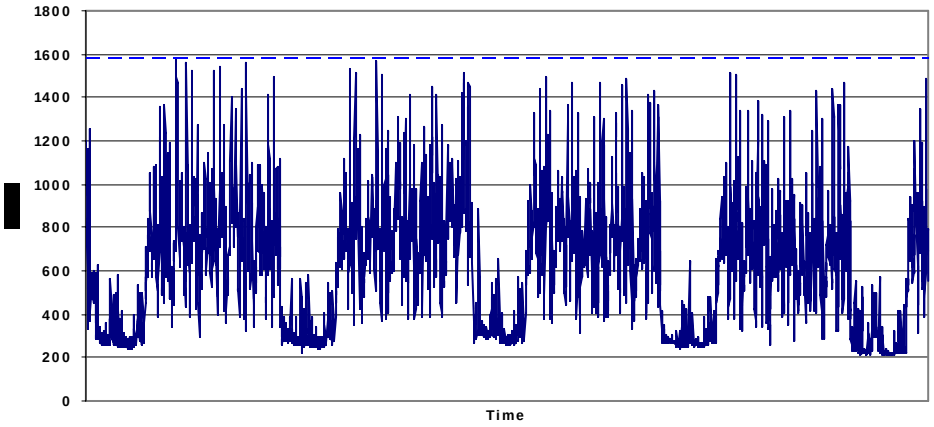

Fig. 2b: Requested versus installed capacity over 4 weeks (Example b)

Measurements of this type are usable for dimensioning and capacity planning and they are also useful to predict future QOS-violations, which may occur due to short-term fluctuations of the workload (work load peaks, bursts). As a consequence we have to be aware of the relation between workload fluctuations on small and large time scales, i.e. on a granularity of a few seconds and on 15 min intervals. Of course, measurements on a less coarse time scale would allow the building of better or more detailed performance models. Moreover, a better knowledge of the fluctuations within the 15 min intervals, say on an observation level of a few seconds (or even below one second) would be helpful to describe the workload more accurately.

2.4 Objectives of capacity planning

We focus on a scenario including a workload behaviour as sketched in Fig. 1 and 2, i.e. a set of LPARs is running on a m-way system (with m processors) providing a certain installed capacity (in MIPS). This assumed future workload is based on a forecast depending on past data and assumptions on shaping, capping, scheduling and priorization of online processing over batch.

Capacity planning includes the question how many processors are necessary to provide a sufficient quality of service. A reasonable approach is to define capacity levels C1 and C2,

which may only be violated with small probabilities. For example, C1 may be violated once per day (one interval out of 96), C2 once per month (one interval out of 2880), and C3 once per year. Of course it is desirable to transform capacity levels (defined as installed MIPS) into service levels like execution velocity, slowdown or wait time percentiles. There is another level C3, which defines a ceiling for the installed capacity.

In the following we derive some formulas that allow a mapping of the coarse measurement data (utilized versus installed capacity) to customer oriented service levels. As a result the allocation of workload to resources can be planned more systematically.

3 Calculation of performance measures

3.1 Notions from queueing theory

During the planning of new systems techniques like Markov chains, stochastic Petri nets, queueing networks and discrete event simulation are well developed modelling techniques, which have been implemented in many performance modelling tools. Here we use models and results from classic queueing theory and provide a brief introduction into some necessary notions.

In particular we use approximate formulas for the solution of G/G/m-models. According to the Kendall notion a G/G/m-model is a queueing model with a general arrival process, a general service process, m servers and an unlimited waiting buffer for arriving tasks. Hence G/G/m-models are rather abstract yet appropriate models for multi-server systems.

Arrival and service processes are characterized by the averages for inter arrival time $E[A]$ and service time $E[S]$ respectively, and moreover by the coefficients of variation $C_A = \sqrt{VAR[A]} / E[A]$ and $C_S = \sqrt{VAR[S]} / E[S]$ for inter arrival and service times.

In case of negative exponential distributions (abbreviated as M) it holds $c_A = 1$ and $c_S = 1$, the associated models are denoted as M/M/1 or M/M/m and the steady-state averages for queue length and response time are given by exact analytical formulas. Also for M/G/1 with $c_S \neq 1$ we have exact solutions for steady state measures. The more general cases G/G/m and G/M/m with approximate solutions are discussed in the next sections.

A simple but important measure is the utilization ρ, which occurs in the following formulas. In case of single servers system of type G/G/1 in steady state, it holds that $\rho = \lambda / \mu < 1$, where $\lambda = 1/E[A]$ and $\mu = 1/E[S]$. For multi-server systems the utilization is defined as $\rho_m = \lambda / m\mu$, where m is the number of processors. For stability and the existence of stationary solutions it is required that $\rho_m < 1$. Multi-Processor systems with non-linear scaling are modelled as multi-server stations with load-dependent service rates $\mu(n)$ and utilization is defined as $\rho_m = \lambda / \mu(m)$.

In the next sections we use known approximate formulas for G/G/m queueing models to derive expressions for the following performance measures

- average execution velocity *E[V]*,
- slowdown factor *F*,
- (normalized) average wait time $E[W_m] = E[W_{G/G/m}]$,
- percentiles of the wait time distribution.

Although simplifications and assumptions will be necessary to use such highly abstract models the consideration on this level of detail increases the insight into system behaviour.

3.2 Average execution velocity

In the context of mainframe terminology the term execution velocity (here abbreviated by V or *E[V]* for expectation value of *V*) is used to describe the ratio for the total amount of workload units that is accepted to be delayed due to waiting for system resources. Execution velocity is a number between 0 and 100, where the value 100 means that during the observation interval the workload did not encounter any wait delays, and the value 0 expresses that all work is delayed over the measurement interval, cf. the documentation of the workload manager [IBM 05].

Here we show how to calculate the average execution velocity as an overall long-term measure of congestion in multi-server systems. No service classes are distinguished, i.e. all tasks are aggregated into a single workload class.

The average velocity *E[V]* is a stationary (global) measure which is calculated from the basic parameters ρ and m, whereas during system operation the execution velocity is determined from measurements over rather short observation periods. *E[V]* is defined as *E[V]=100·(1-P[W>0])*%, where the waiting probability P_m=*P[W>0]* can be calculated according to different fomulas. *P[W>0]* is the probability that a task is delayed because it has to wait. Hence *P[W>0]* is also named delay probability or wait probability and is denoted equivalently (in percent) as *100*P[W>0]%*, Results for the calculation of *P[W>0]* for different types of models are available from the literature, cf. the next sections or [Whit92, Whit04]. A survey of approximate formulas for a variety of G/G-models, like G/G/m and the less general case G/M/m is summarized in [BGMT98].

The classic formula for computing the wait probability (at least in case of M/M-systems) is the Erlang C-formula or M/M/m/m-formula, well-known from its wide usage in telephony [BGMT98], cf. (6.28)] or [Klein78]. The Erlang C-formula provides the probability that the number of customers K reaches or exceeds the limit m.

$P[W > 0] = P_m = P(K \geq m) = \frac{(m\rho)^m}{m!(1-\rho)} \cdot \pi_0$, where the probability π_0 for the empty system is defined as

$$\pi_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \frac{(m\rho)^m}{m!} \frac{1}{1-\rho} \right]^{-1}.$$

The Erlang C-formula is needed to evaluate the average waiting time given by the approximations due to Allen/Cunneen and Krämer/Langenbach-Belz.

Figure 3 illustrates the average velocity *E[V]=100·(1-P[W>0])*% for m processors, m = 1, 4, 8, 16, 32, 64.

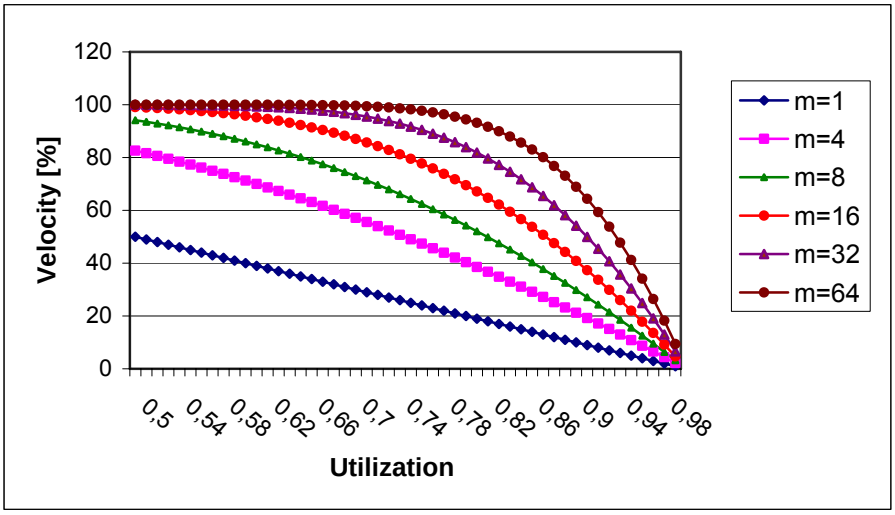

Fig. 3: Average execution velocity for m processors (m=1, …, 64)

Consider e.g. a value of 40% as an example for a possible service level. Figure 3 shows under which utilization the average velocity drops below 40%, e.g. for m=32 this is the case for utilizations beyond 92%. Note that the average execution velocity is not sensitive with respect to the variability of the arrival or service process, i.e. C_A and C_s do not influence the average velocity.

3.3 Normalized average wait time

Here we consider the calculation of the average wait time. Since we do not distinguish service classes or even individual tasks we calculate the average wait time and further on the slowdown factor on the basis of virtual transactions of average unit length *E[S]=1*. Normalization of service time to the length of one unit is a simple transformation known from queueing theory. Note that for *E[W]=0* the response time equals the service time *S*. Since tasks often suffer some wait time they are “slowed-down”. Hence we define the slowdown factor $F = (E[W] + E[S])) / E[S]$. Summarizing we obtain the simplified slowdown factor definition $F = E[W] + 1$. Due to the normalization *E[S] = 1* the expression for slowdown also provides the (normalized) average response time $E[R] = F = E[W] + 1$.

In case the slowdown factor is $F = 1$ we have no wait time at all, in case of $F >> 1$ there is heavy contention or high overload. Slowdown, say up to a factor of 2, may be considered still satisfying. Different quality classes for the quality of service may be defined, as an example we may assume that $1 \le F < 2$ is good, $2 \le F < 6$ is moderate and $F \ge 6$ stands for bad service. The values *F=2* and *F=6* may be considered as service levels, critical thresholds or performance goal definitions.

Another technique to deal with service levels is the usage of a “Performance Index” (*PI*) as known from IBM´s Workloadmanager. A performance index is scaled in such a way that values ≤1 indicate the meeting of the performance goal. Such a performance index calculates to

$$PI = Pg / Pa,$$

where *Pg* denotes a defined performance goal and *Pa* denotes the actual (measured) performance value. For details and examples see the WLM-manual (IBM redbooks, [IBM06]).

Now we calculate the performance measures average wait time *E[W]* and slowdown F by using the few parameters we have at hand. No exact formula is known for the G/G/m-model, hence we use the approximate formulas due to Kingman/Köllerström, Allen/Cunneen and Krämer/Langenbach-Belz. Like before, m denotes the number of processors, λ the arrival rate, C_A the coefficient of variation of the mean inter arrival time, *E[S]* the mean service time, and C_S is the coefficient of variation of the mean service time. Since the mean service time *E[S]* is normalized to 1, we have $\rho = \rho_m = \lambda E[S]/m = \lambda/m$ as the (normalized) utilization.

Since C_A^2 and C_S^2 occur as factors in the $E[W_{G/G/m}]$-formulas, the wait time is very sensitive with respect to the workload variability. For a comparison of the formula of Kingman/Köllerström versus the formula of Allen/Cunneen see e.g. [BGMT98].

3.3.1 KK: Kingman/Köllerström-formula

Firstly, we provide the full approximate formula due to Kingman/Köllerström for the G/G/m/∞-model [Klei78] and its normalized version using *E[S]=1,* $E[A] = 1/\rho m$ (due to the definition $\lambda = 1/E[A]$) and the notion $\sigma_A = C_A/\rho m$ (by definition of C_A)

$$E[W_{G/G/m}] \approx \frac{\sigma_A^2 + \sigma_S^2/m^2}{2E[A](1-\rho)} \qquad \text{(G/G/m, average wait time due to KK),}$$

$$E[W_{G/G/m}] \approx \frac{\rho}{(1-\rho)} \frac{C_A^2/\rho^2 + C_S^2}{2m} \qquad \text{(G/G/m, normalized by } E[S]=1\text{).}$$

In case of $\sigma_S = c_S = 1$ (neg. exp. distributed service time), the G/G/m/∞-formula further simplifies to the approximate G/M/m/∞-formula

$$E[W_{G/M/m}] \approx \frac{\rho}{(1-\rho)} \frac{C_A^2/\rho^2 + 1}{2m} \qquad \text{(G/M/m) .}$$

Throughout the formulas given above the utilization ρ is a function of m, and could be written also as ρ_m.

3.3.2 AC: Allen/Cunneen-formula

Instead of the Kingman/Köllerström-Formula alternatively the formula of Allen/Cunneen may be used. Here it is displayed in combination with the calculation P_m given above.

$$E[W_{G/G/m}] = \overline{W} \approx \frac{P_m / \mu}{1-\rho} \cdot \frac{C_A^2 + C_S^2}{2m}$$ (G/G/m, average wait time formula).

We again use the normalization *E[S]=1* to simplify this expression to

$$E[W_{G/G/m}] = \overline{W} \approx \frac{P_m}{1-\rho} \cdot \frac{C_A^2 + C_S^2}{2m}$$ (G/G/m, normalized by *E[S]=1).*

As an example for the usage of the AC-formula we show a diagram using the assumption that the workload variability is quantified by $z = C_A^2 + C_S^2 = 2$.

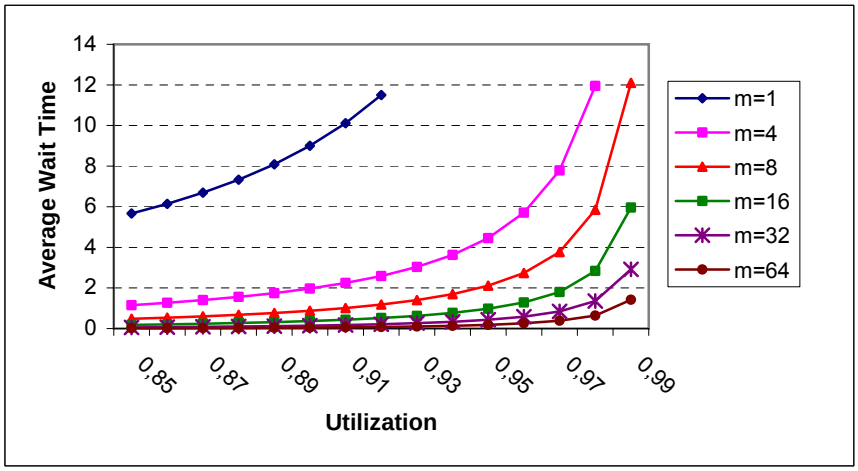

Fig. 4: Normalized average wait time

Figure 4 illustrates the decrease of wait time with increasing number of processors m.

3.3.3 KLB: Krämer/Langenbach-Belz-formula

The formula of Allen/Cunneen has been improved to the Krämer/Langenbach-Belz-formula, cf. [BGMT98, formula (6.91) und (6.92)] using a correction factor abbreviated as G_{KLB}.

$$\overline{W} \approx \frac{P_m}{1-\rho} \cdot \frac{C_A^2 + C_S^2}{2m} \cdot G_{KLB}$$ (G/G/m/∞-formula, KLB-formula)

$$G_{KLB} = \begin{cases} \exp(-\frac{2}{3}\frac{(1-\rho)}{P_m} \cdot \frac{(1-C_A^2)^2}{C_A^2 + C_S^2}), & 0 \le C_A \le 1, \\ \exp(-(1-\rho) \cdot \frac{C_A^2 - 1}{C_A^2 + 4C_S^2}), & C_A > 1. \end{cases}$$

3.3.4 Other formulas for the average wait time E[W]

There are more results on G/G/m-models, cf. again [BGMT98]

KB: Kingman/Brumelle/Marchal (upper and lower bounds)
Ku: Kulbatzki (another improvement)
Ja: Jaeckel (Improvement of Kulbatzkis´s improvement)
Ki: Kimura (improvement for the case $C_A^2 < 1$)

BCH: Boxma/Cohen/Huffels (mixture from formulas for G/D/m, …, G/M/m)
Ti: Tijms (by modification of a G/M/m-formula)

For the rest of the paper we use the formulas of Kingman/Köllerström and Allen/Cunneen. A comparison of the goodness of all these approximations is not within the scope of this paper.

3.4 Percentiles of the normalized wait time distribution

Performance monitoring often uses the performance metric "percentile" (also called quantile), which is the percentage of work in an observation period that should complete within the response time. For example, a 70%-percentile of 0.5 sec states that 70 percent of the transactions have a response time ≤ 0.5 sec. In the given context percentiles are defined as performance goals or service levels and are permanently monitored during normal system operation. Percentiles are calculated per service class and observation periods are usually rather short, e.g. 10 seconds, to allow the scheduler (which is part of the workload managing software) to take some action if performance goals are not met.

Here we consider the calculation of global steady state (≈ long term) percentiles under the assumptions discussed throughout the preceding sections. We simply postulate that the wait time is exponentially distributed; under this assumption the (inverse) mean wait time can be used as a parameter for the exponential wait time distribution. As a consequence the calculation of percentiles for wait time and response time is straightforward.

Summarizing, we obtain the cumulative distribution function for the normalized wait time distribution *FW* depending on m, ρ, C_A and C_S.

$$FW_{m,\rho,C_A,C_S}(w) = 1 - e^{-\gamma_m w}, \text{ where } \gamma_m = 1/E[W_{G/G/m}].$$

Hence the p-percentiles w_p of *FW* are defined as $w_p = -\frac{1}{\gamma_m}\ln(1-p)$.

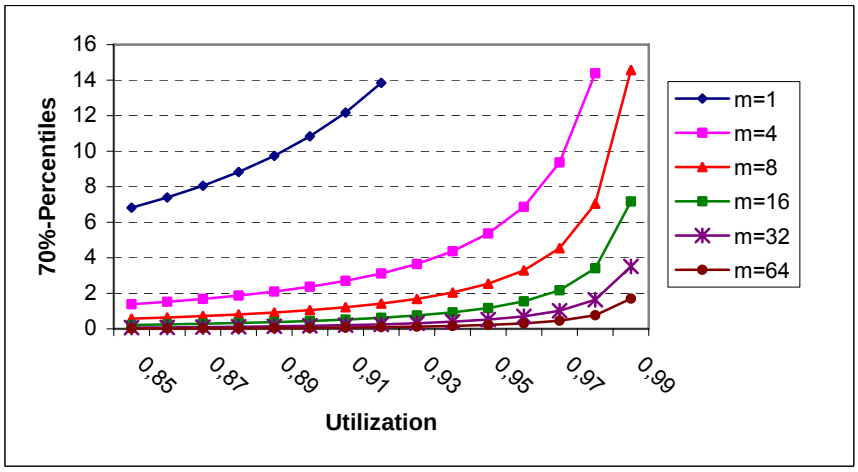

Fig. 5: 70%-percentiles of the wait time distribution

These percentiles determine boundaries w_p, i.e. the normalized wait time is smaller than w_p with probability p. The statistical interpretation is that $p \cdot 100\%$ of the tasks suffer a

normalized wait time that is less than w_p. In Figure 5 the 70%-percentiles of the (normalized) wait time $E[W_m]$ are displayed for different values of m and utilization ρ.

For an interpretation consider e.g. the curve for m=8 at utilization 0.93 yielding a value of 1.7 with the meaning that 70% percent of all normalized wait times are smaller than 1.7 time units. Note that this holds under a workload variability, which is expressed by $z = C_A^2 + C_S^2 = 2$.

4 Application to mainframe capacity planning

The discussed performance metrics are useful tools to define and calculate service levels or threshold values for different scenarios. We describe a scenario by three components: a workload description, a system configuration and service level requirements (also called performance goals). Using of the developed formulas we can quantify to which extent a workload allocated to a specified system can be executed without violating the desired service levels. As a consequence we are able to propose how to allocate workloads to systems and/or how to upgrade or downgrade a system.

This approach is in particular useful if we have many workloads (many LPARs) to be allocated to a number of mainframe computers (also called CECs, where CEC stands for Central Electronic Complex).

4.1 Definition of capacity planning scenarios

In this section we describe how to build a scenario, which comprises workload, system configuration and service levels. The level of detail and the input parameters are of course determined by the formulas introduced before. Furthermore there are other important issues with respect to workload management, which must be considered additionaly. In particular the intelligent allocation of LPARs to CECs is of course a key to efficient and cost saving operation, which has to consider many factors including accounting, contracting, costs for software licences and more.

Here, the main objective is to achieve a tradeoff between high utilization of the processor hardware and the prespecified service level requirements.

4.1.1 Workload model

The workload description consists of a mix of LPARs to be allocated on a system (= CEC). According to the given granularity of the workload description the processing time consumptions are assumed to be given in MIPS. We may distinguish several variants of workload characterization, e.g. the maximum of a 4 hours rolling average over all LPARs or a time series of 15 min values over all LPARs. In both cases the amount of requested service is quantified by an average value over a certain time interval.

Apart from the average value we include the workload variability, which is described by the values C_A and C_S, which account for the variability of the arrival and service process. The formulas for wait time in the preceeding section include the factor $z = C_A^2 + C_S^2$, whereas

velocity is robust with respect to C_A and C_S. In case of good-natured workloads we may use the assumption $C_A^2 + C_S^2$ = 1.0 which may be reasonable and not too optimistic in many cases. Even values less than 1.0 may be justified in case of a good balanced system where automatic workload management by scheduling, capping and shaping cares for homogeneous work load behaviour.

An essential part of capacity planning is the development of workload forecasts. Often a workload forecast is based on historical data that have been obtained from aggregated monitoring data (RMF). Seasonal effects and trends have to be considered as well as the evolution of the business processes.

Since the logical partitions hosted on mainframes often serve a closed user community and are regulated by contracts between provider and user the variation of the workload is rather low and develops slowly over time. Moreover, logical partitions can be moved between different mainframe systems to achieve appropriate workload profiles. Hence, workload forecasting is not a very hard task. For a further elaboration of this topic, we refer to workload forecasting techniques as e.g. described in the literature, cf. [MeAD04, ch. 12], [BeMe04], [Gunt07].

4.1.2 Configuration model

A system configuration (for a single CEC) consists of the number of processors m and the corresponding amount of installed MIPS. Typically for each system or machine type the manufacturer or independent institutions provide tables with scaling factors. Figure 6 shows an example, where the MIPS-increments initially slowly decrease and finally approach a nearly linear behaviour.

Of course system scalability of multi-processor systems depends on the machine architecture, the operating system, the kinds of applications, the scheduling and workload management, and more. For a thourough discussion of scaling of multiprpocessor systems see [Gunt00, Gunt07].

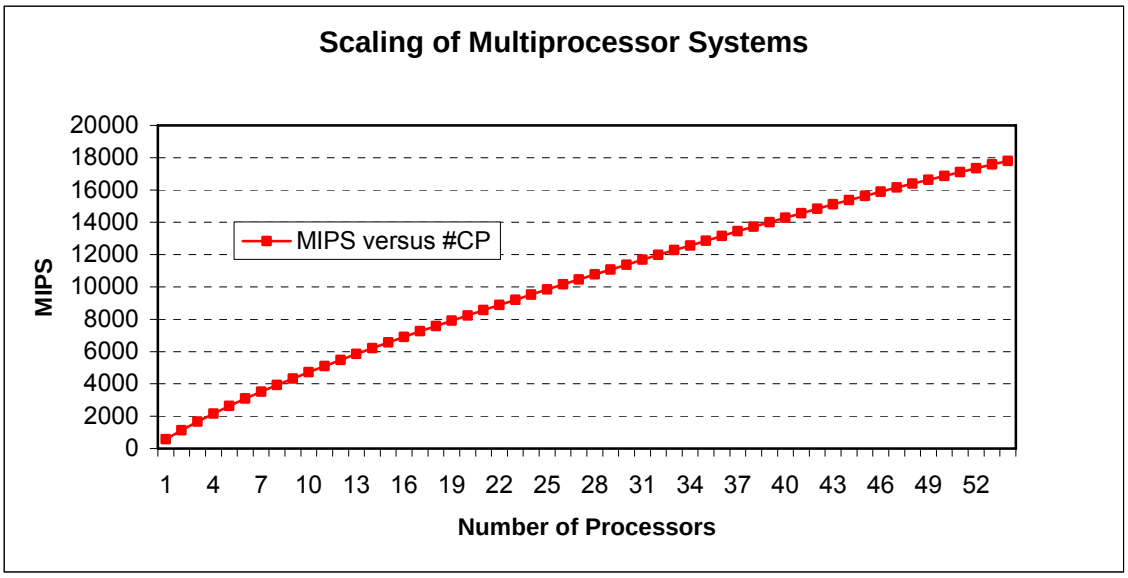

Fig. 6: Scaling factors (Example)

The scaling function displayed in Figure 6 is obtained from real benchmark data and shows a piecewise approximate linear behaviour with rather constant increments. Hence the usage of G/G/m formulas is justified for a restricted range of n ± 3 processors.

4.1.3 Definition of service levels

The definition of service levels requires numerical values for velocity V, slowdown F and percentiles for the wait time distribution. We can focus on one of these measures or on combinations of several measures. In any case we refer to averages under steady-state assumptions. For an illustration we give some examples of service levels.

Velocity V = 40%: The interpretation is that 40% of all tasks are served without any wait delay, i.e. 60% of all tasks suffer a certain wait time. Velocity V=40% is equivalent to wait probability P[W>0]=0.6.
Slowdown F < 2: It holds that E[R]/E[S] < 2, where E[R] denotes the average response time and E[S] the average service time. In other words, the mean wait time is less than the mean service time.
Percentiles (w_p =2, p=0.70) or equivalently (w_p =2, p=70%)**:** The pair (w_p , p) defines that the normalized wait time is less than w_p = 2 seconds with a probability of p=0.7 (= 70%).

Table 1: Service level definitions (numerical examples)

4.2 Example: A real world scenario

A typical real world scenario in mainframe capacity planning includes the economic allocation of a set of LPARs (workloads) to a number of CECs (servers) in such a way that performance goals will be met and the number of operative processors (and other resources) is as low as possible. An immediate consequence is to maximize utilization of the processors (per CEC) and provide enough spare capacity in order to cope with peaks in the workload.

4.2.1 Workload allocation strategies

One approach (as described in [CapS07]) comprises as a first step an analysis of all LPARs wrt the processing time consumption over a certain period of time followed by a grouping which is favourable with respect to utilization and available spare capacity. This LPAR allocation algorithm uses statistical data of each LPAR (mean, maximum, variance of consumed processing capacity), ordered capacity (i.e. amount of MIPS), type and number of processors, main memory, and more. As a result we obtain a value, which quantifies the inclination of a LPAR towards a certain CEC. The capacity planner may choose from different strategies to control the allocation of LPARs; an example is the “maximum headroom strategy” which minimizes the maximum overall utilization on the CEC under consideration. The allocation process of LPARs to CECs and a possible result is briefly sketched in Figure 7. The arrows indicate the strength of inclinations to a certain CEC.

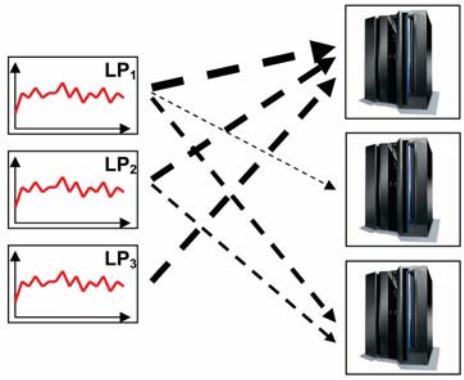

Fig. 7: Allocation of LPARs to CECs (basic outline)

In Figure 8 it is displayed how the allocation of a set of LPARs may evolve over a certain time period. The horizontal line labeled C1 indicates the capacity limit, which should normally suffice for a performant operation, line C2 indicates the installed capacity, i.e. the difference C2-C1 is the available headroom (fast spare capacity), which will suffice to deal with temporary workload peaks. In case the COD option (COD = Capacity On Demand) is available we may care for extra capacity (slow or quiet spare capacity), which defines another line C3. The specification of concrete values for C1, C2 and C3 depends on many factors. Of course a model based approach with the goal to quantify performance measures and to derive concrete values for C1, C2 and C3 is highly desirable.

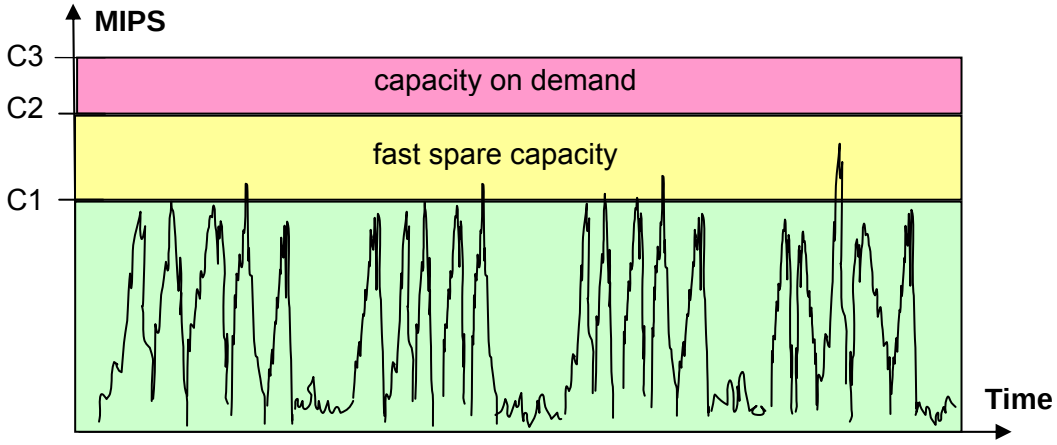

Fig. 8: Utilization over 4 weeks (basic outline)

Figure 8 illustrates an approach where the capacity limits C1 and C2 have been chosen in a way that a given hypothetical workload will be processed with just a few violations of the limit C1 and no violation of limit C2 at all. Of course we are well aware that a workload may evolve completely differently over time if more or less processing capacity is allocated; moreover the workload manager will take action in order to meet all performance goals. Nevertheless the capacity planner has to cope with his/her restricted macroscopic view on the overall system. As an example how real system behaviour is considered as part of the LPAR allocation strategy we refer to Figure 9, where workload utilization behaviour is displayed before and after the application of shaping [CapS07]. The violations of the capacity limit (left diagram) have disappeared after shaping (right diagram).

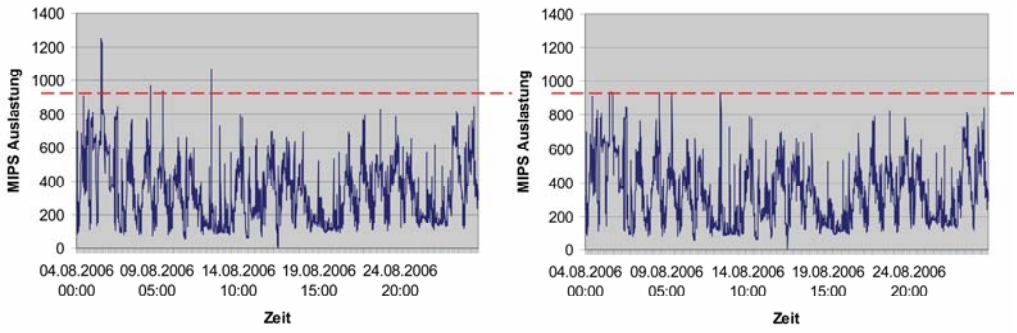

Fig. 9: Workload Shaping

Note that shaping is an approximate and heuristic technique that models the capping feature as applied during real mainframe operation, e.g. in order to restrict batch load to a specified maximum MIPS consumption.

4.2.2 Model based capacity planning

Here we map the scenario sketched above to G/G/m-formulas which will provide quantitative results for velocity and wait time. The concrete scenario assumes a single mainframe, which can be equipped with a certain number of processors. The range of interest is m = 16, 17, …, 22 processors, yielding the values for installed computing power as displayed in Table 2, cf. the column “Available MIPS”. Furthermore we have a set of LPARs, which are planned to be allocated to a single CEC and request a certain computing capacity, cf. the column “Requested MIPS”. The column “Utilzation” is given by the ratio 100*(Requested MIPS/Available MIPS) [%].

Processing Units	Available MIPS	Requested MIPS	Utilization [%]
16	6211	6208	99,9
17	6596	6208	94,1
18	6984	6208	88,9
19	7372	6208	84,2
20	7760	6208	80,0
21	8148	6208	76,2
22	8563	6208	72,5

Table 2: Available MIPS versus requested MIPS

Note that the average number of requested MIPS is the simplest possible characterization of a workload. The variable behaviour of this workload over time (more precisely over a busy hour) is assumed to be described sufficiently by the variability z.

4.2.3 Average velocity

The formula given in section 3.2 yields the values for average velocity, cf. Figure 10. If we define a service level for average execution velocity of E[V]=70%, the diagram shows that a 20-way configuration should be appropriate. From Table 2 we see that a value of E[V]=70% corresponds to a utilization of 80% providing a headroom of 20%.

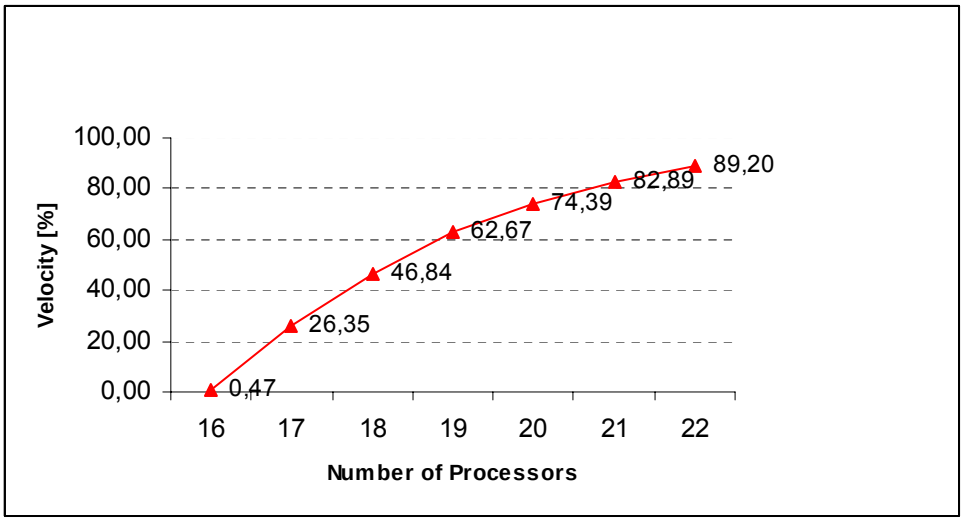

Fig. 10: Average velocity [%] versus number of processors

Note that the average velocity is robust with respect to workload variability, i.e. Figure 10 holds for all values of z.

4.2.4 Average wait time and wait time percentiles

The (normalized average) wait time has been evaluated using the AC-formula for different values of workload variability $z = C_A^2 + C_S^2$. Unlike velocity the wait time is sensitive with respect to workload variability. In particular in the case of 17-way and 18-way configurations where utilization exceeds 90% the wait time increases exponentially for a highly variable workload (z=5). As a result we obtain characteristic curves as function of processor speed, number of processors, utilization and workload variability.

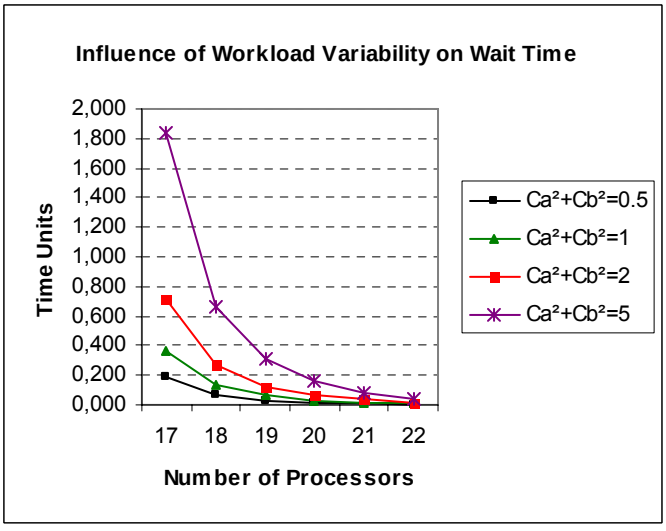

Fig. 11: Average wait time as function of workload variability z = 0.5, 1.0, 2.0, 5.0

The wait time percentiles shown in Figures 12 and 13 have been computed under the assumption of moderate and high workload variability using the formula in Sec. 3.4.

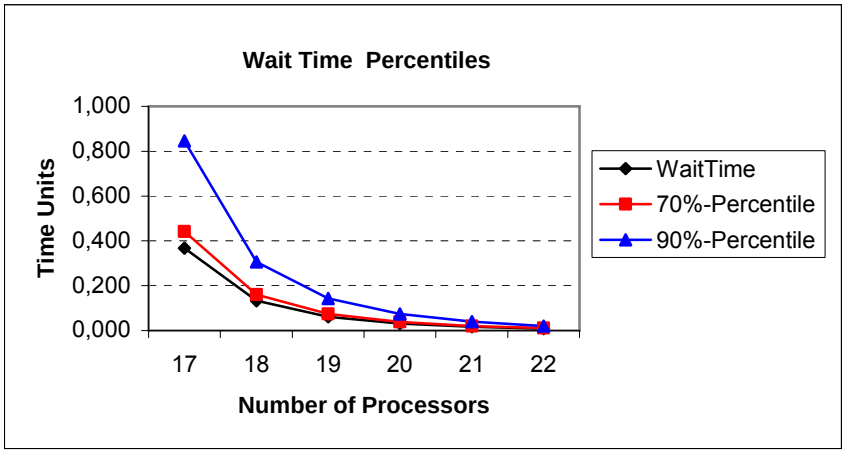

Fig. 12: Wait time percentiles for moderate workload variability (z=1.0)

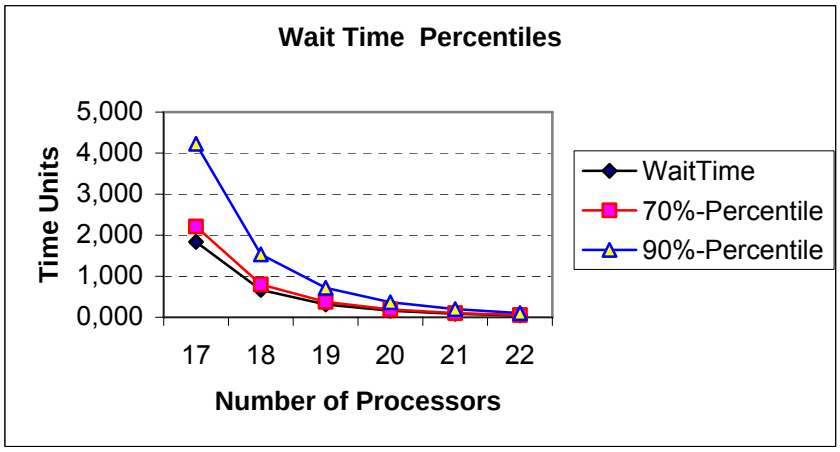

Fig. 13: Wait time percentiles for high workload variability (z =5.0)

The 19-way configuration performs well with both types of workloads, although the 18-way configuration will degrade in case of a highly variable workload. The 90%-percentile is greater than 1.5, i.e. wait time exceeds service time by 50%.

5 Conclusion and outlook

Approximation formulas from queueing theory for G/G/m-models have been used to derive useful measures like average execution velocity, average wait time, and wait time percentiles to support capacity and workload planning of multi-processor and mainframe systems. The application has been illustrated by an example taken from a real world scenario. Future work will integrate this approach into a capacity management framework of a large service provider [CapS07]. Practical applications and field studies must be performed to demonstrate the feasibility of this approach. Hence the research focus will be on the analysis of workload data in order to improve the processes of measurement, forecasting and model validation.

6 References

[BGMT98] Bolch, G., Greiner, S., de Meer, H., Trivedi, K.: Queueing Networks and Markov Chains – Modeling and Performance Evaluation with Computer Science Applications. John Wiley and Sons, 1998

[Bolc83] Bolch, G.: Approximation von Leistungsgrößen symmetrischer Mehrprozessorsysteme, Computing 31, 305-315, 1983

[CapS07] CapSys User Manual. University of Duisburg-Essen, ICB, Systems Modeling Group, Internal Document, 2007

[FCSM04] Fair, M.L., Corkin, C.R., Swaney, S.B., Meany, P.J., Clarke, W.J., Alves, L.C., Modi, I.N., Freier, F., Fischer, W., Weber, N.E.: Reliability, availability, and serviceability (RAS) of the IBM eServer z990. IBM Journal Res. & Dev. (2004), vol 48, no. 3/4, 519-534

[Gunt00] Gunther, N. J.: The Practical Performance Analyst. Author Choice Press, 2000

[Gunt07] Gunther, N. J.: Guerilla Capacity Planning – A Tactical Approach for Highly Scalable Applications and Services, Springer, 2007

[IBM05] IBM red books: System Programmer's Guide to: Workload Manager – Workload Manager Overview and Functionalities, Sept 2005

[IBM06] IBM red books: http://www.redbooks.ibm.com/, IBM red books, 2006

[IBM0b] IBM z series: www.ibm.com/systems/z/, IBM Mainframes and the new system z9, 2006

[Klei78] Kleinrock, L.: Queueing Systems, Vol. II: Computer Applications, John Wiley, 1978

[MeAl02] Menascé, D. A., Almeida, V. A.: Capacity Planning for Webservices – Metrics, Models, and Methods. Prentice Hall, 2002

[MeAD04] Menascé, D. A., Almeida, V. A., Dowdy, L.: Performance by Design: Computer Capacity Planning by Example. Prentice Hall, 2004

[MüCl01] Müller-Clostermann, B.: Kursbuch Kapazitätsmanagement – Kompendium für Planung, Analyse und Tuning von IT-Systemen. ISBN 3-8311-2823-5 (in German as pdf-version available under http://sysmod.icb.uni-due.de/), Essen, 2001

[Whit92] Whitt, W.: Understanding the Efficiency of Multi-Server Systems, Management Science (1992) 38, 708-723, 1992

[Whit04] Whitt, W.: A Diffusion Approximation for the G/GI/n/m Queue. Operations Research (2004), Vol. 52, No. 6, 922-941

7 Appendix: Tools

Although the approximate G/G/m-fomulas due to Allen/Cunneen, Kingman/Köllerström and Krämer/Langenbach-Belz may be evaluated by a spreadsheet calculation it is more convenient to use existing tools. A source of small tools is provided by the performance modelling toolbox which includes also some freeware programs to calculate the relevant performance measures for G/G/m-models. Various algorithms including AC, KK, and KLB; provide either textual or graphical output; also the graphical comparison results from different solution formulas or diverse model variants is possible. Additionally csv output to a file may be used to proceed with the analysis within a spreadsheet calculation program. The performance modelling toolbox can be accessed via http://sysmod.icb.uni-due.de/ .

The performance modelling toolbox is a permanently running project, which has the objective to provide basic algorithms from queueing theory and performance modelling for the public domain. The software is freely available from the above web page for purposes of teaching and research. No warranties are made that any of these programs is error-free, or will meet your requirements for any particular application. The authors disclaim all liabilities for direct or indirect damages resulting from the usage of this program.

Currently (in May 2007) the performance modelling toolbox contains contributions, which are due to Falko Hildebrand, Denis Hirsch, Tim Jonischkat, Martin Kaczmarczyk, Marc-Oliver Meyer, Milen Tilev, and Mykhailo Vilents.

Previously published ICB - Research Reports

2007

No 15 (April 2007)
Heise, David; Schauer, Carola; Strecker, Stefan: "Informationsquellen für IT-Professionals – Analyse und Bewertung der Fachpresse aus Sicht der Wirtschaftsinformatik"

No 14 (March 2007)
Eicker, Stefan; Hegmanns, Christian; Malich, Stefan: "Auswahl von Bewertungsmethoden für Softwarearchitekturen"

No 13 (February 2007)
Eicker, Stefan; Spies, Thorsten; Kahl, Christian: "Softwarevisualisierung im Kontext serviceorientierter Architekturen"

No 12 (February 2007)
Brenner, Freimut: "Cumulative Measures of Absorbing Joint Markov Chains and an Application to Markovian Process Algebras"

No 11 (February 2007)
Kirchner, Lutz: "Entwurf einer Modellierungssprache zur Unterstützung der Aufgaben des IT-Managements – Grundlagen, Anforderungen und Metamodell"

No 10 (February 2007)
Schauer, Carola; Strecker, Stefan: "Vergleichende Literaturstudie aktueller einführender Lehrbücher der Wirtschaftsinformatik: Bezugsrahmen und Auswertung"

No 9 (February 2007)
Strecker, Stefan; Kuckertz, Andreas; Pawlowski, Jan M.: "Überlegungen zur Qualifizierung des wissenschaftlichen Nachwuchses: Ein Diskussionsbeitrag zur (kumulativen) Habilitation"

No 8 (February 2007)
Frank, Ulrich; Strecker, Stefan; Koch, Stefan: "Open Model -- Ein Vorschlag für ein For-schungsprogramm der Wirtschaftsinformatik (Langfassung)"

2006

No 7 (December 2006)
Frank, Ulrich: "Towards a Pluralistic Conception of Research Methods in Information Systems Research"

No 6 (April 2006)
Frank, Ulrich: "Evaluation von Forschung und Lehre an Universitäten – Ein Diskussionsbeitrag"

No 5 (April 2006)
Jung, Jürgen: "Supply Chains in the Context of Resource Modelling"

No 4 (February 2006)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part III – Results Wirtschaftsinformatik Discipline"

2005

No 3 (December 2005)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part II – Results Information Systems Discipline"

No 2 (December 2005)
Lange, Carola: "Development and status of the Information Systems / Wirtschaftsinformatik discipline: An interpretive evaluation of interviews with renowned researchers, Part I – Research Objectives and Method"

No 1 (August 2005)
Lange, Carola: „Ein Bezugsrahmen zur Beschreibung von Forschungsgegenständen und -methoden in Wirtschaftsinformatik und Information Systems"

The Institute for Computer Science and Business Information Systems (ICB), located at the Essen Campus, is dedicated to research and teaching in Applied Computer Science, Information Systems as well as Information Management. The ICB research groups cover a wide range of expertise:

Research Group	Core Research Topics
Prof. Dr. H. H. Adelsberger Information Systems for Production and Operations Management	E-Learning, Knowledge Management, Skill-Management, Simulation, Artificial Intelligence
Prof. Dr. P. Chamoni MIS and Management Science / Operations Research	Information Systems and Operations Research, Business Intelligence, Data Warehousing
Prof. Dr. F.-D. Dorloff Procurement, Logistics and Information Management	E-Business, E-Procurement, E-Government
Prof. Dr. K. Echtle Dependability of Computing Systems	Dependability of Computing Systems
Prof. Dr. S. Eicker Information Systems and Software Engineering	Process Models, Software-Architectures
Prof. Dr. U. Frank Information Systems and Enterprise Modelling	Enterprise Modelling, Enterprise Application Integration, IT Management, Knowledge Management
Prof. Dr. M. Goedicke Specification of Software Systems	Distributed Systems, Software Components, CSCW
Prof. Dr. R. Jung Information Systems and Enterprise Communication Systems	Process, Data and Integration Management, Customer Relationship Management
Prof. Dr. T. Kollmann E-Business and E-Entrepreneurship	E-Business and Information Management,E-Entrepreneurship/ E-Venture, Virtual Marketplaces and Mobile Commerce, Online-Marketing
Prof. Dr. B. Müller-Clostermann Systems Modelling	Performance Evaluation, Modelling and Simulation, SAP Capacity Planning for R/3 and mySAP.com, Tools for Queueing Network Analysis and Capacity Planning, Communication Protocols and Distributed Systems, Mobile Systems
Prof. Dr. K. Pohl Software Systems Engineering	Requirements Engineering, Software Quality Assurance, Software-Architectures, Evaluation of COTS/Open Source-Components
Prof. Dr.-Ing. E. Rathgeb Computer Networking Technology	Computer Networking Technology
Prof. Dr. R. Unland Data Management Systems and Knowledge Representation	Data Management, Artificial Intelligence, Software Engineering, Internet Based Teaching
Prof. Dr. S. Zelewski Institute of Production and Industrial Information Mangement	Industrial Business Processes, Innovation Management, Information Management, Economic Analyses

For more information visit us on the Web: **http://www.icb.uni-due.de**

ISSN 1860-2770