TECHNISCHE UNIVERSITÄT DRESDEN

Performance Analysis of Computer Systems

Introduction to Queuing Theory

Holger Brunst (holger.brunst@tu-dresden.de)

Matthias S. Mueller (matthias.mueller@tu-dresden.de)

ZIH
Center for Information Services & High Performance Computing

TECHNISCHE UNIVERSITÄT DRESDEN

Summary of Previous Lecture

Simulation Validation and Verification

Holger Brunst (holger.brunst@tu-dresden.de)

Matthias S. Mueller (matthias.mueller@tu-dresden.de)

Center for Information Services & High Performance Computing

Validation and Verification

- Does the simulation model represents the real system?
  - Or: Are the assumptions about the real system (the system being modeled) reasonable?
  - $\Rightarrow$ validation
- Is the simulation model correctly implemented?
  - Or: Does the simulation implements these assumptions cerrect?

  $\Rightarrow$ verification

Model validation techniques

- Key aspects of a model:
  1. Assumptions
  2. Input parameter values and distributions
  3. Output values and conclusions
- Each of these aspects needs to be validated
- Three possible sources of validity tests:
  1. Expert intuition
  2. Real system measurements
  3. Theoretical results

Model verification techniques

- Also called “debugging”
- Modularity or top-down design
- Antibugging
- Structured Walk-Through
- Deterministic models
- Run simplified cases
- Trace or online monitoring
- Continuity test
- Degeneracy tests
- Consistency tests
- Seed independence

Operational Method

- Is based on a set of concepts that correspond naturally and directly to observed properties of real computer systems
- The computer system will be modeled with the help of a queuing network
- A queuing network has two types of nodes: wait and delay nodes
- A wait node consists of a input queue and a server
  - Jobs arrive at in the input queue
  - The server can only work one job at the time
  - A server is not idle when there are jobs in his queue
- Delay nodes can serve multiple jobs at once
  - Jobs stay until there are finished
- Jobs which have fulfilled they total work time demand leave the system

Constraints to queuing networks

1. Flow balance in each node
2. One step at the time
3. Routing homogeneity
4. Work time homogeneity
5. Arrival homogeneity

- Queuing networks which apply to these constraints are called *separable* networks
- Because each wait node can be analyzed independent of all others
- Combining the performance results of all nodes gives the performance of the total queuing network

Summary of Foundational Laws for Operational Methods

- Throughput Law

$$\frac{U_k}{V_k S_k} = X$$

- Forced Flow Law

$$X_k = V_k X$$

- Utilization Law

$$U_k = X_k S_k = D_k X$$

- Little's Law

$$N = XR$$

$$N_k = X_k W_k = XR_k$$

Open models

$$X = \lambda$$

$$X_k = \lambda V_k$$

$$U_k = \lambda D_k$$

- For delay nodes

$$R_k = D_k$$
$$N_k = XR_k = \lambda D_k = U_k$$

- For wait nodes

$$R_k = \frac{D_k}{1 - U_k}$$
$$N_k = XR_k = \lambda \frac{D_k}{1 - U_k} = \frac{U_k}{1 - U_k}$$

TECHNISCHE UNIVERSITÄT DRESDEN

Introduction to Queuing Theory

Holger Brunst (holger.brunst@tu-dresden.de)

Matthias S. Mueller (matthias.mueller@tu-dresden.de)

Center for Information Services & High Performance Computing

If the facts don't fit the theory, change the facts.

Albert Einstein

Outline

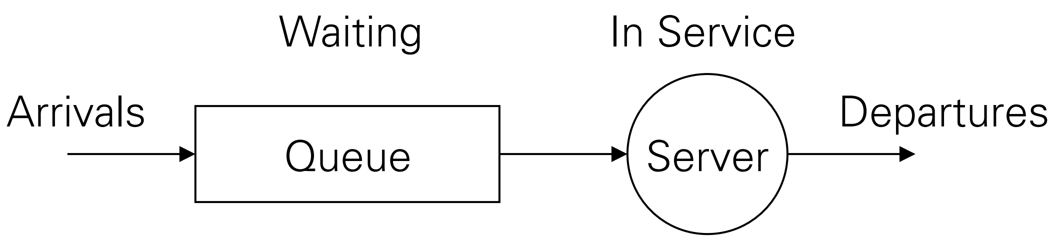
- Motivation
- Queuing/Kendall notation
- Queuing in daily life
- Exponential distribution and its memoryless property
- Little's law
- Stochastic processes, birth-death process
- M|M|1 queuing model

Motivation

- **Sharing of system resources** in computer systems:
  - CPU, Disk, Network, etc.
- Generally, only one job can use the resource at any time
- All other jobs using the same resource **wait in queues**
- **Queuing** or **queuing theory** helps in determining the time that jobs spend in various system queues.
- These times can be combined to predict the response time of jobs

Queuing Notation

- Imagine yourself at a supermarket checkout
- The checkout has a number of open cash points
- Usually, the cash points are busy and arriving customers have to wait
- In queuing theory terms you would be called "**customer**" or "**job**"
- In order to analyze such systems, the following system characteristics should be specified:
  1. Arrival Process (Interarrival Time Distribution)
  2. Service Time Distribution
  3. Number of Servers
  4. System Capacity
  5. Population Size
  6. Service Discipline



Queuing Notation

1. **Arrival Process (Ankunftsprozess)**
   - If customers arrive at $t_1, t_2, \ldots, t_j$, the random variables $\tau_j = t_j - t_{j-1}$ are called **interarrival times (Zwischenankunftszeiten)**.
   - General assumption: The τ_j form a sequence of **i**ndependent and **i**dentically **d**istributed (**IID**) random variables
   - Most common arrival process is the **Poisson Process** which has exponentially distributed inter-arrival times
   - Erlang- and hyper-exponential distributions are also used
2. **Service Time Distribution (Antwortzeitverteilung)**
   - The time a customer spends at the service station e.g. the cash points
   - This time is called the **service time (Antwortzeit)**
   - Commonly assumed to be IID random variables
   - Exponential distribution is often used

Queuing Notation

3. **Number of Servers (Anzahl der Bedienstationen)**
   - The number of service providing entities available to customers
   - If in the same queuing system, servers are assumed to be:
     - Identical
     - Available to all customers
4. **System Capacity**
   - The maximum number of customers who can stay in the system
   - In most systems the capacity is finite
   - However, if the number is large, infinite capacity is often assumed for simplicity
   - The number includes both waiting and served customers

Queuing Notation

5. **Population Size**
   - The total number of customers to be served
   - In most real systems the population is finite
   - If this size is large, once again, the size is assumed infinite for simplicity reasons
6. **Service Discipline** or **Scheduling**
   - The order in which customers are served:
     - First come first served (**FCFS**)
     - Last come first served (**LCFS**) with or without preemption
     - Round Robin (**RR**) with fixed size quantum
     - Shortest processing time **(SPT)**
     - Service in random order **(SIRO)**
     - System with fixed delay, e.g. satellite link
     - Prioritized scheduling (**PRIO**)

Kendall Notation

- These six parameters need to be specified in order to define a single **queuing station**
- To compactly describe the queuing station in an unambigous way, the so called **Kendall Notation** is often used:

  Arrivals | Services | Servers | Capacity | Population | Scheduling

  - Arrivals ➡ customer arrival process
  - Services ➡ customer service requirements
  - Servers ➡ number of service providing entities
  - Capacity ➡ maximum number of customers in queuing station
  - Population ➡ size of the customer population
  - Scheduling ➡ employed scheduling strategy
- Population and Scheduling are often omitted i.e. assumed to be infinitely and FCFS

Kendall Notation

- The specific values of the parameters, especially **Arrivals** and **Services**, are diverse. Some commonly used once are:
  - ***M*** (Markovian or Memory-less): whenever the interarrival or service times are (negative) exponentially distributed
  - ***G*** (General): whenever the times involved may be arbitrarily distributed
  - ***D*** (Deterministic): whenever the times involved are constant
  - E_r (*r*-stage Erlang): whenever the times involved are distributed according to an *r*-stage Erlang distribution
  - H_r: whenever the times involved are distributed according to an *r*-stage hyper-exponential distribution

Kendall Notation - Example

- **M|G|2|8||LCFS** denotes a queuing station with:
  - Negative exponentially distributed interarrival times
  - Generally distributed service times
  - 2 service providing entities
  - Maximal 8 customers present
  - No limitation on the total customer population
  - LCFS scheduling strategy
- Simple queuing stations as above can be used for many queuing phenomena in computer and communication systems
- However, just a single queue with single service entity is considered, only allowing performance evaluations of parts of a complex system
- Examples: Analysis of network access mechanisms, simple transmission links, or various disk and CPU scheduling mechanisms

Queuing in Daily Life

- Coin-operated coffee machines
  - Service time, i.e., the time for preparing the coffee, is deterministic
  - Waiting time occurs due to the stochastic in the arrival process
  - Kendall notation: G|D|1
- Visiting a doctor with appointment
  - Arrival times of patients is deterministic (if their appointments are accurate)
  - However, one often experiences long waiting times due to the stochastic service times (time the doctor talks to or examines patients)
  - Kendall notation: D|G|1
- Visiting a doctor without appointment
  - Things become get even worse during “walk-in” consulting hours
  - Both arrival and service process obeys only general characteristics and the perceived waiting time increases
  - Kendall notation: G|G|1

Exponential (Markov) Distribution

- The (negative) exponential distribution is used extensively in queuing models
- It is the only continuous distribution with the so-called **memoryless property** which strongly simplifies the analysis:
  - *Remembering the time since the last event does not help in predicting the time till the next event!*
- Commonly used to model random durations, e.g.:
  - Duration of a phone call, Time between two phone calls
  - Duration of services, reparations, maintenance
  - Lifetime of radiactive atoms
  - Lifetime of parts, machines, technical equipment (without decline!)

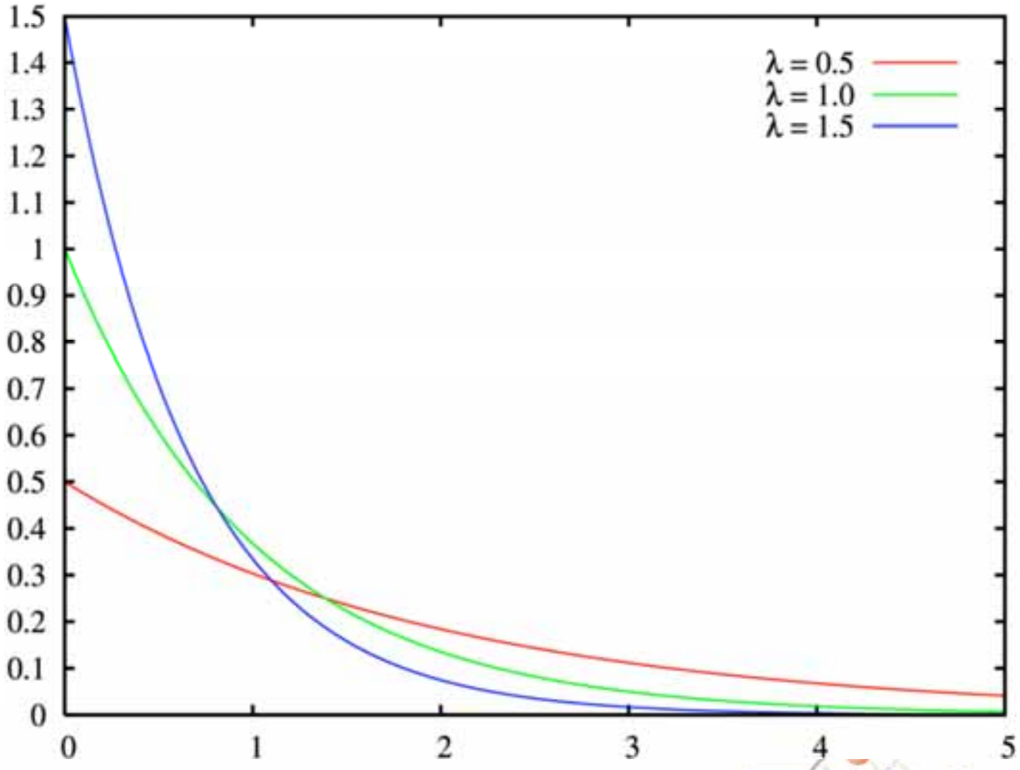
Exponential Distribution

Probability density function (Dichtefunktion), short: pdf

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & , \quad x \geq 0, \\ 0 & , \quad x < 0. \end{cases}$$

- Supported on interval [0,∞)
- λ > 0 is a parameter of the distribution
- Often called **rate parameter**
- Probability of continuous random variable X:

$$\mathrm{P}(a \leq X \leq b) = \int_a^b f(x)\mathrm{d}x$$



Exponential Distribution

Cumulative distribution function (Verteilungsfunktion), short CDF

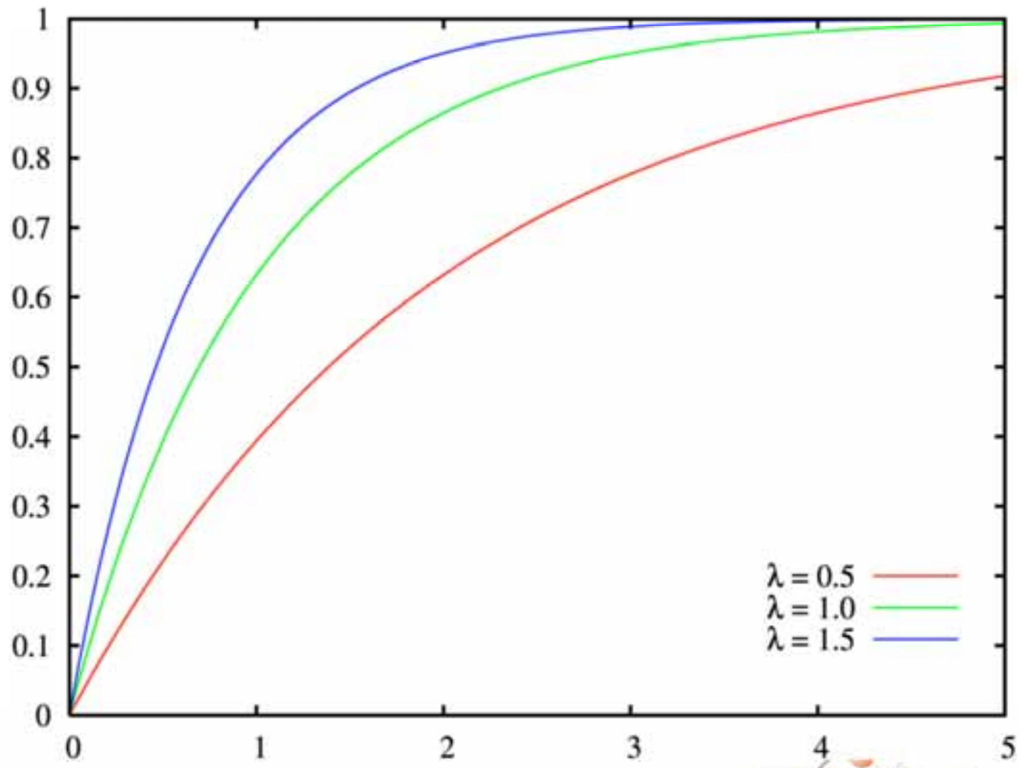
$$F(x,\lambda) = \begin{cases} 1 - e^{-\lambda x} & , \quad x \geq 0, \\ 0 & , \quad x < 0. \end{cases}$$

- Mean:

$$\mathrm{E}[X] = \frac{1}{\lambda}$$

- Variance:

$$\mathrm{Var}[X] = \frac{1}{\lambda^2}$$



Memoryless Property

- Stated earlier: Remembering the time since the last event does not help in predicting the time till the next event!
- Probability distribution of an exponentially distributed event *T* to occur within time *t:*

$$F(T) = P(T < t) = 1 - e^{-\lambda t}, t \geq 0$$

- We see an arrival event and start the clock at $t = 0$. The mean time to the next arrival event is $1/\lambda$.
- Suppose we do not see an arrival event until $t = x$. The distribution of the time remaining until the next arrival is:

$$\begin{aligned} P(T < x + t \mid T > x) &= \frac{P(x < T < x + t)}{P(T > x)} \\ &= \frac{P(T < x + t) - P(T < x)}{P(T > x)} \\ &= \frac{(1 - e^{-\lambda(x+t)}) - (1 - e^{-\lambda x})}{e^{-\lambda t}} \\ &= 1 - e^{-\lambda t} \end{aligned}$$

Memoryless Property

- A random variable T is said to be memoryless if:

$$P(T < x + t \mid T > x) = P(T < t) \quad \forall\ x, t \geq 0$$

- Example:
  - Give a real-life example whose lifetime can be modeled by a variable *T* such that $P(T > s + t \mid T > s)$ goes down as *s* goes up
  - Bus with exponentially distributed arrival times with λ=2/h
    - Average waiting time?
    - Expected waiting time when already waiting for 15 minutes?

$$P(T < 20 \mid T > 15) = P(T < 5)$$

    - It does not mean that

$$P(T < 20 \mid T > 15) = P(T < 20)$$

Little's Law

- Named after John Little (MIT) who proved the law in 1961
- One of the most general laws in performance analysis
- Can be applied almost unconditionally to all queuing models and at many levels of abstraction
- Interesting point of notice: Long used before actually proved

- Little's Law basically **relates** the **average number of jobs** *N* in queuing station **to** the **average number of arrivals** per time unit λ and the **average time** *R* spent in the queuing station

$$N = \lambda R$$

Little's Law - Understanding

- Consider a queuing station as a black box
- On average λ jobs arrive per time unit
- Upon its arrival, a job is either served or has to wait
- Denote $E[R]$ (**r**esidence time or **r**esponse time) as the average time spend in the queuing system
- Denote average number of jobs in the queuing system as $E[N]$
- Observe a single marked job which enters the system at $t=t_i$ leaves at $t=t_o$.
- On average $t_o - t_i$ will be equal to $E[R]$
- While this particular job passes the system, other jobs have arrived
- Since on average $E[R]$ time units elapsed, their average number is $\lambda \times E[R]$
- This number must be equal to the previously defined $E[N]$ as every job could be the marked job. Thus:

$$E[N] = \lambda E[R]$$

Little's Law - Remarks

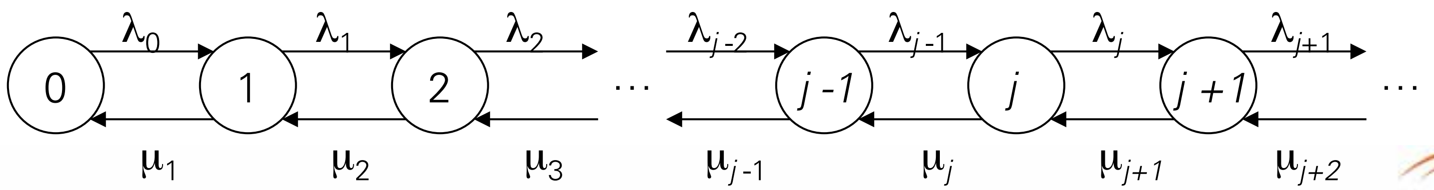
- We assumed that the queue throughput T equals the arrival rate λ
- Always the case if system is not overloaded and infinite buffers
- Otherwise customers will get lost and $E[N] = T\ E[R]$
- The relationship expressed by Little's law is valid independently of the form of the involved distributions
- This law is valid independently of the scheduling discipline and the number of servers
- $E[N]$ is easy to obtain and measures like $E[R]$ can be derived from it
- Applies also to networks of queuing stations

Stochastic Processes

- Analytical modeling uses several random variables but also **stochastic processes** which are sequences of random variables
- Collection of random variables $\{ X(t) \mid t \in T \}$, indexed by the parameter t *(usually time)* which can take values of set T
- Values that $X(t)$ assumes are called **states**. All possible states are called **state space *I***.
- The state space and the time parameter can be discrete or continuous
- Discrete-state stochastic processes are also called **chain**, often with $I = \{0,1,2,\ldots\}$
- Famous representatives: Markov Process, Birth-Death Process, and Poisson Process (form a hierarchy)

Birth-Death Process

- Future states of the process are independent of the past and depend only on the present
- Special case of the continuous time Markov chain
- State transitions are restricted to neighboring states
- States are represented by integers. State *n* can only change to state *n+1* or state *n-1*
- Example: The number of jobs in a queue with a single server and individual arrivals (no bulk arrivals)
- An arrival to the queue (birth) causes the state to change by +1. A departure (death) causes the state to change by -1
- Below: State transition diagram with *n* states, arrival rates λ_n and service rates μ_n. Arrival times and service times are exponentially distributed



Birth-Death Process

- The **steady-state probability** p_n of a birth-death process being in state n is given by the following theorem:

$$p_n = p_0 \frac{\lambda_0 \lambda_1 \cdots \lambda_{n-1}}{\mu_1 \mu_2 \cdots \mu_n}$$

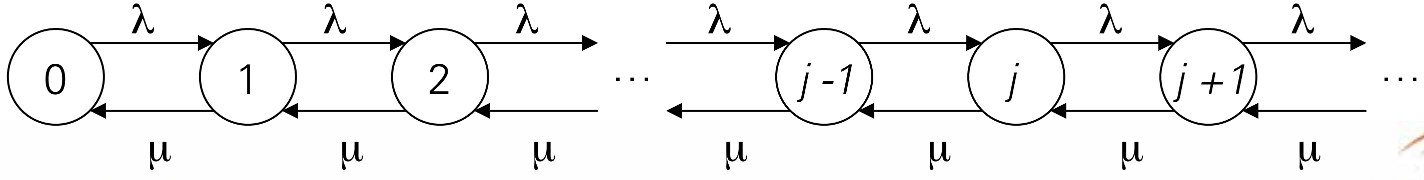
$$= p_0 \prod_{j=0}^{n-1} \frac{\lambda_j}{\mu_{j+1}}, \qquad n = 1, 2, \ldots, \infty$$

- p_0 is the probability of being in the **zero state**
- Can be proven (see book)
- Now that we have an expression for state probabilities we are able to analyze queues in the form of M/M/m/B/K
- Based on the state probabilities we can compute many other performance measures

M|M|1 Queuing Model

- **Most commonly used** type of queue
- Can be used to model single-processor system or individual devices in a computer system
- Interarrival and service times are exponentially distributed, one server
- No buffer or population size limits, FCFS service discipline
- Analysis: We need the **mean arrival rate λ** and the **mean service rate μ**
- State transition similar to birth-death process with $\lambda_n = \lambda$ *and* $\mu_n = \mu$
- The probability of *n* jobs in the system becomes:

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0, \qquad n = 1, 2, \ldots, \infty$$



M|M|1 Queuing Model

$$p_n = \left(\frac{\lambda}{\mu}\right)^n p_0, \qquad n = 1, 2, \ldots, \infty$$

- The quantity $\lambda/\mu = \rho$ is called **traffic intensity**
- Thus $p_n = \rho^n p_0$
- All probabilities should add to 1. Knowing this we can derive an equation for the probability of zero jobs (p_0) in the systems:

$$p_0 = \frac{1}{1 + \rho + \rho^2 + \ldots + \rho\infty} = 1 - \rho$$

- Substituting p_0 in p_n leads to:

$$p_n = (1-\rho)\rho^n, \qquad n = 0, 1, 2, \ldots, \infty$$

- Based on this expression, many other properties can be derived
- Utilization of the server: $U = 1 - p_0 = \rho$
- The mean number of jobs in the system:

$$E[n] = \sum_{n=1}^{\infty} n p_n = \sum_{n=1}^{\infty} n(1-\rho)\rho^n = \frac{\rho}{1-\rho}$$

M|M|1 Queuing Model

- The probability of n or more jobs in the system is:

$$P(\geq n \text{ jobs in the system}) = \sum_{j=n}^{\infty} p_j = \sum_{j=n}^{\infty} (1-\rho)\rho^j = \rho^n$$

- Using Little's law we can compute the mean response time:

$$E[n] = \lambda E[r]$$

$$E[r] = \frac{E[n]}{\lambda} = \left(\frac{\rho}{1-\rho}\right)\frac{1}{\lambda} = \frac{1/\mu}{1-\rho}$$

Thank You!

Holger Brunst (holger.brunst@tu-dresden.de)

Matthias S. Mueller (matthias.mueller@tu-dresden.de)