CS5239 Computer System Performance Analysis

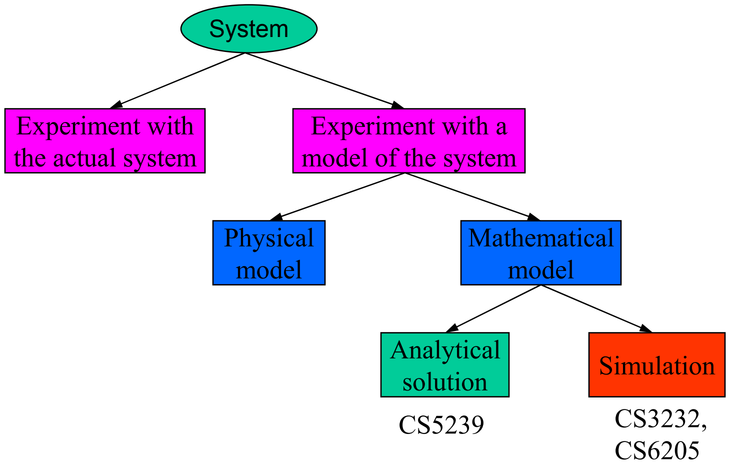
2004/05 – Semester 1

Assoc. Professor Teo Yong Meng
Room: S14, #06-12
Department of Computer Science
National University of Singapore
E-mail: teoym@comp.nus.edu.sg
www.comp.nus.edu.sg/~teoym/cs5239.htm

Overview of Performance Analysis

- Ways of studying a system
- Introduction
- Purpose of evaluation
- Applications of performance evaluation
- Performance evaluation techniques
- Criteria for selecting an evaluation technique
- Applicability of evaluation techniques
- Steps for a performance evaluation study
- Performance evaluation study example
- Performance evaluation metrics

Ways to Study a System



System

Experiment with the actual system

Experiment with a model of the system

Physical model

Mathematical model

Analytical solution

Simulation

CS5239

CS3232, CS6205

Introduction

Computer system users, administrators, and designers are all interested in *performance evaluation* since the goal is to obtain or to provide the highest performance at the lowest cost. Computer performance evaluation is of vital importance in the selection of computer systems, the design of applications and equipment, and the analysis of existing systems.

Purpose of Evaluation

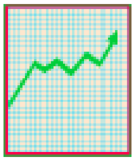
Three general purposes:

- selection evaluation - system exists elsewhere (procurement)
- performance projection - system does not yet exist (new system)
- performance monitoring - system in operation (capacity management)

Selection Evaluation (Procurement)

- evaluate plans to include performance as a major criterion in the decision to obtain a particular system from a vendor is the most frequent case
- to determine among the various **alternatives** which are available and suitable for a given application
- to choose according to some specified selection criteria
- requires at least one prototype of the proposed system must exist

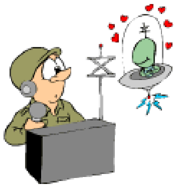
Performance Projection

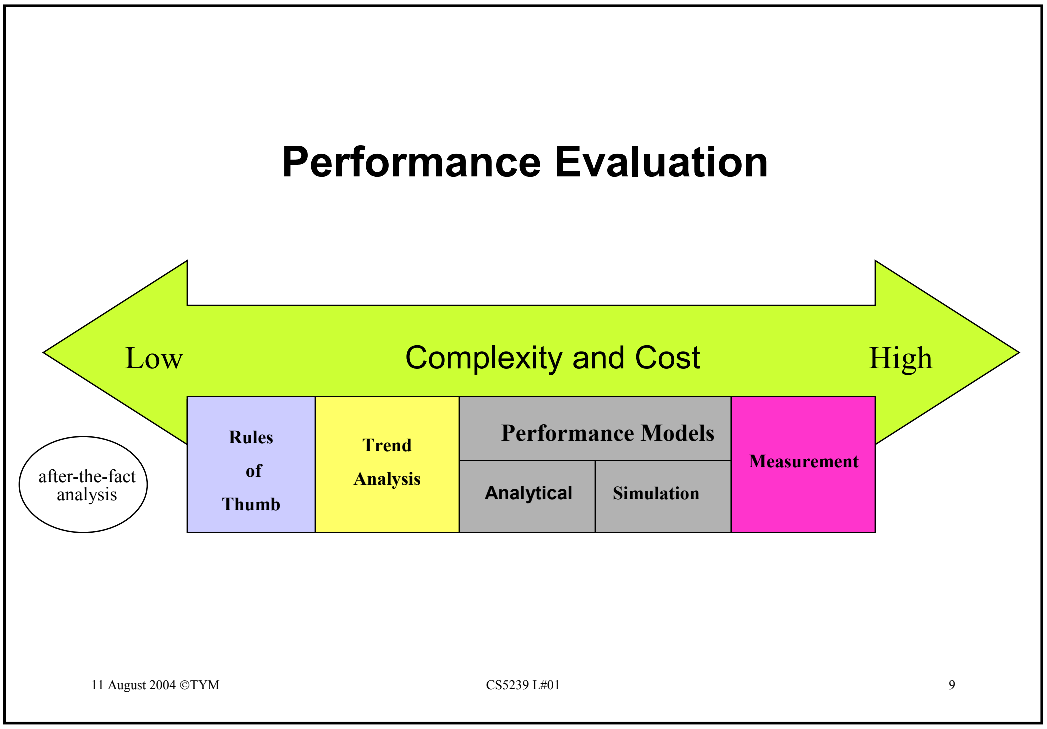


- orientated towards **designing** a new system
- to estimate the performance of a system that does not yet exist
- secondary goal - projection of a given system on a new workload, i.e. modifying existing system in order to increase its performance or decrease it costs or both (*tuning therapy*)
- upgrading of a system - replacement or addition of one or more hardware components

Performance Monitoring

- usually performed for a substantial portion of the lifetime of an existing running system
- Performance monitoring is done:
  - to detect *performance bottlenecks*
  - to predict future capacity shortcomings
  - to determine most cost-effective way to upgrade the system
    - to overcome performance problems, and
    - to cope with increasing workload demands





Performance Evaluation

Low

Complexity and Cost

High

Rules of Thumb

Trend Analysis

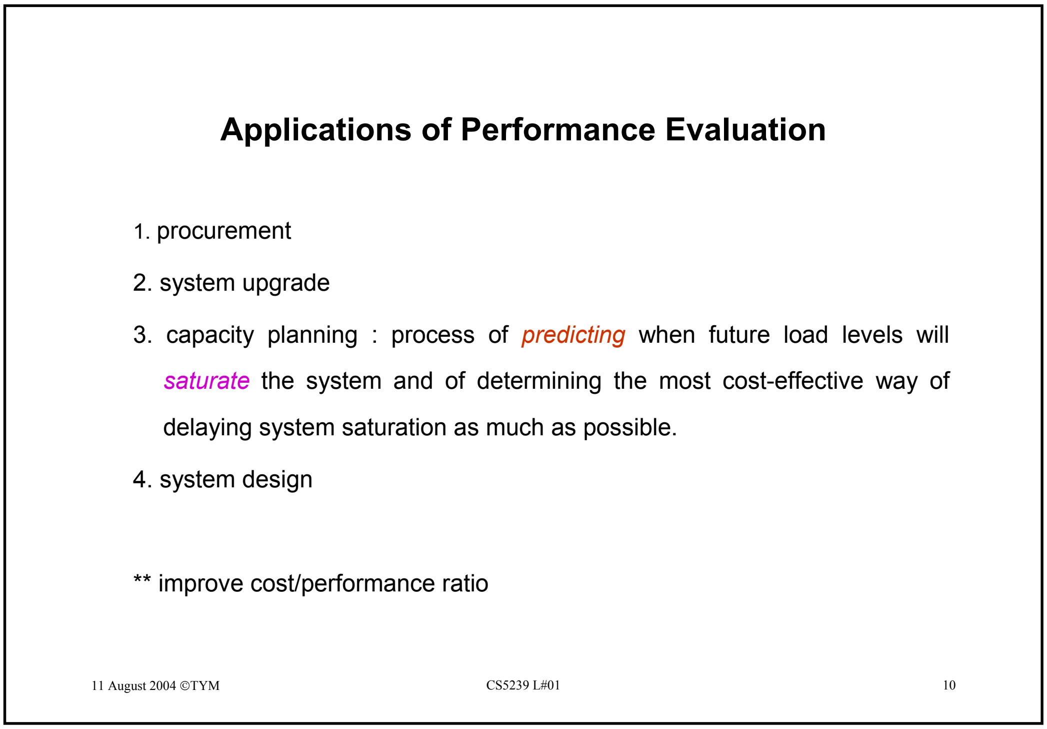
Performance Models

Analytical

Simulation

Measurement

after-the-fact analysis



Applications of Performance Evaluation

1. procurement
2. system upgrade
3. capacity planning : process of *predicting* when future load levels will *saturate* the system and of determining the most cost-effective way of delaying system saturation as much as possible.
4. system design

** improve cost/performance ratio

Example

A virtual car dealership provides users with a Web site they can visit to search and submit purchase requests. It has 1,300 affiliated car dealers that provide information about the vehicles available in their parking lots. A full description of each vehicle is stored in a database. The Web server receives three types of requests:

- requests for documents and images
- requests to search the data base
- purchase requests

Example

Let T denotes the *response time* for search transaction: if $T \leq 4$ seconds, then user will successfully complete search request; if $4 < T \leq 6$ seconds, then 60% of the users will abort the search; and if $T > 6$ seconds, then 95% of the users will abort the search.

Assume that 5% of search transactions generate a car sale, and that a sale generates US$18,000 on the average in revenues.

Example – what if?

Management wants to answer the following questions:

- Will the Web server support the load increase while keeping T below 4 seconds?
- If not, at which point will its capacity be saturated and why?
- How much money could be lost daily if the Web server saturates when the load increases?

Performance evaluation by the capacity planner produced the following results:

Example

	current load	+10%	+20%	+30%
search per day	92,448	101,693	110,938	120,182
response time (sec)	2.86	3.80	5.67	11.28
sales lost (%)	0	0	60	95
sales per day	4,622	5,085	2,219	300
daily revenue	83,203	91,524	39,938	5,408
potential daily revenue	83,203	91,524	99,844	108,164
lost daily revenue	-	-	59,906	102,756

last three row entries are in (in US$1,000)

Performance Evaluation Techniques

- measurement or empirical techniques
  - the system to be evaluated must exist and be available
- three types of modeling technique:
  - simulation modeling
  - analytic modeling
  - hybrid modeling

Models are inevitably partial and approximate representations of reality, accuracy of the models must be supported.

Criteria for Selecting an Evaluation Technique

Criterion	Analytical Modeling	Simulation	Measurement (exit)
1. Stage	Any	Any	Post-prototype
2. Time required	Small	Medium	Varies
3. Tools	Analysts	Computer languages	Instrumentation
4. Accuracy*	Low	Moderate	Varies
5. Trade-off evaluation	Easy	Moderate	Difficult
6. Cost	Small	Medium	High
7. Saleability	Low	Medium	High

* In all cases, result may be misleading or wrong.

Applicability of Evaluation Techniques

		Evaluation technique		
			Modeling	
Type of study	Object	Measurement	Simulation	Analytic
Design	System	I	A	A
	Program	I	A	I
Procurement	System	A	A	I
	Program	A	A	I
Capacity planning	System	I	A	A
Improvement	System	A	A	A
	Program	A	A	A

A, adequate; I, inadequate.

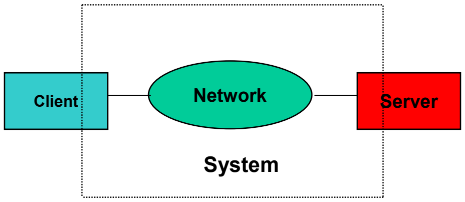
Steps for a Performance Evaluation Study

1. State the *goals* of the study and define the *system boundaries*.
2. List system services and possible outcomes.
3. Select performance metrics.
4. List system and workload parameters.
5. Select factors and their values.
6. Select evaluation techniques.
7. Select the workload.
8. Design the experiments.
9. Analyze and interpret the data.
10. Present the results. Start over, if necessary.

Performance Evaluation Study Example

1. **System Definition**:

   The goal of the case study is to compare the performance of applications using remote pipes (caller is not blocked) to those of similar applications using remote procedure calls (calling program is blocked).

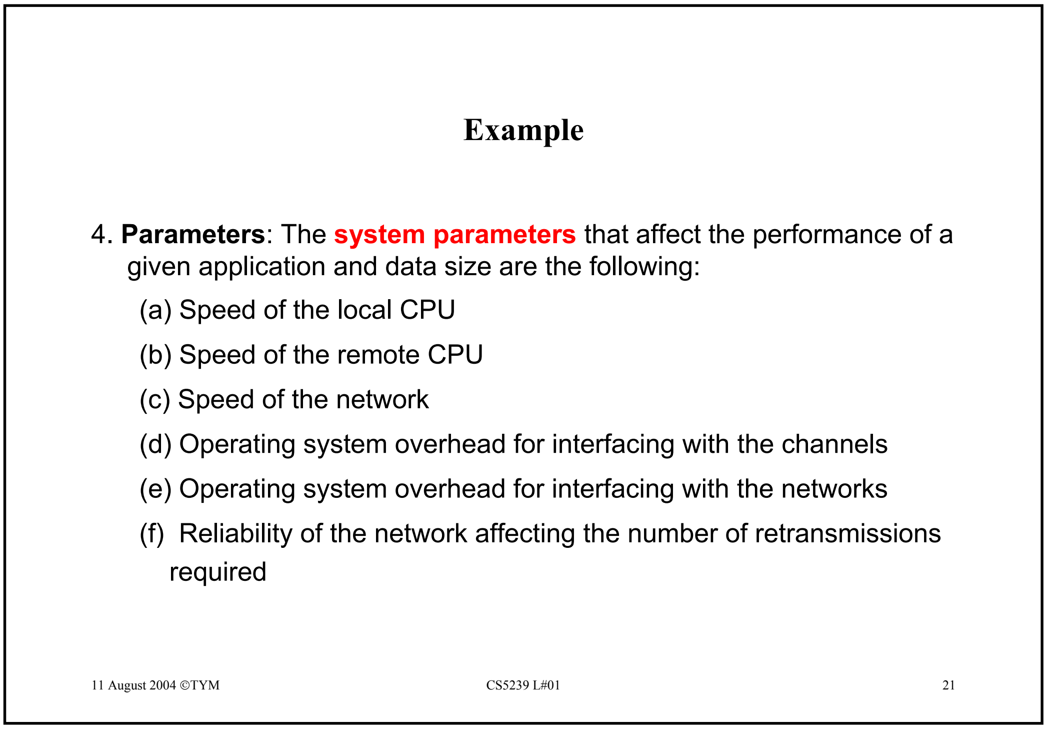


Example

2. **Service**:

   The services offered by the system are the two types of channel calls - remote procedure call and remote pipe.

3. **Metrics**: The resources are the local computer (client), the remote computer (server), and the network link. This leads to the following performance metrics:
   - (a) Elapsed time per call
   - (b) Maximum call rate per unit of time, or equivalently, the time required to complete an block of n successive calls
   - (c) Local CPU time per call
   - (d) Remote CPU time per call
   - (e) Number of bytes sent on the link per call



Example

4. **Parameters**: The **system parameters** that affect the performance of a given application and data size are the following:

   (a) Speed of the local CPU

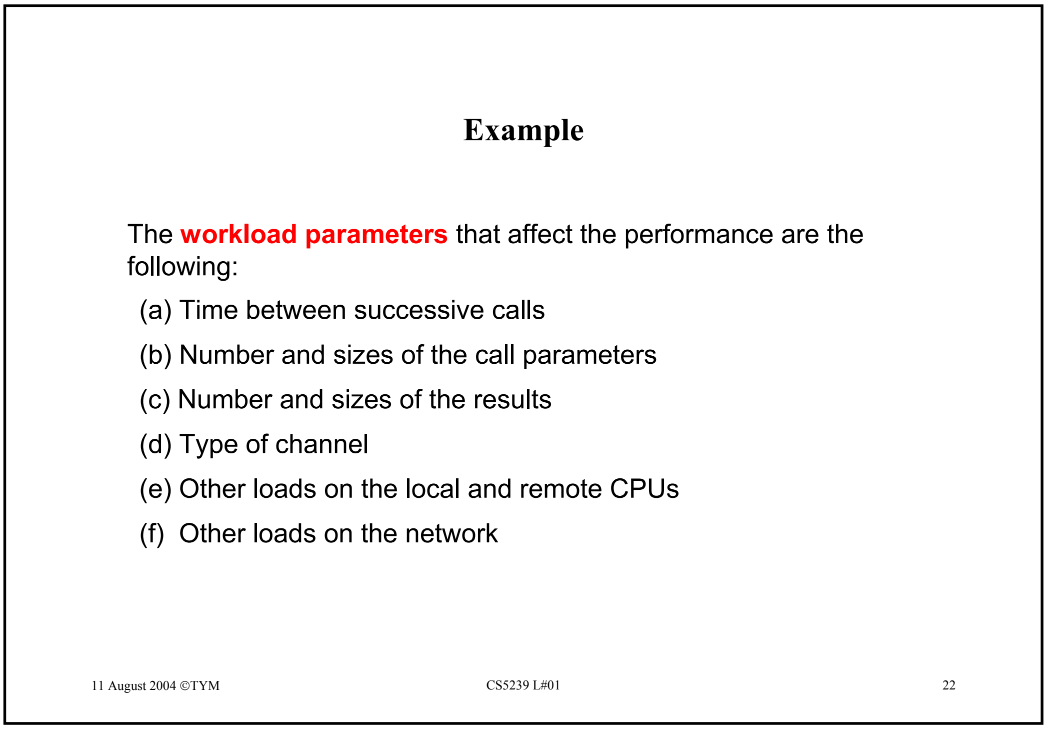
   (b) Speed of the remote CPU

   (c) Speed of the network

   (d) Operating system overhead for interfacing with the channels

   (e) Operating system overhead for interfacing with the networks

   (f) Reliability of the network affecting the number of retransmissions required



Example

The **workload parameters** that affect the performance are the following:

(a) Time between successive calls

(b) Number and sizes of the call parameters

(c) Number and sizes of the results

(d) Type of channel

(e) Other loads on the local and remote CPUs

(f) Other loads on the network

Example

5. **Factors**: The key factors chosen for this study are the following:
   - (a) Type of channel. Two types-remote pipes and remote procedure calls-will be compared.
   - (b) Speed of the network. Two locations of the remote hosts will be used-short distance (in the campus) and long distance (across the country).
   - (c) Sizes of the call parameters to be transferred. Two levels will be used - small and large.
   - (d) Number n of consecutive calls. Eleven different values of n -1, 2, 4, 8, 16, 32,…, 512, 1024, - will be used.

   The factors have been selected based on resource availability and the interest of the sponsors.

Example

6. **Evaluation Technique**: Since prototypes of both types of channels have already been implemented, measurements will be used for evaluation. Analytical modeling will be used to justify the consistency of measured values for different parameters.

7. **Workload**: The workload will consist of a synthetic program generating the specified types of channel requests. This program will also monitor the resources consumed and log the measured results. Null channel requests with no actual work but with monitoring and logging activated will be used to determine the resources consumed in monitoring and logging.

Example

8. **Experimental Design**: A full factorial experimental design with 2^3 x 11 = 88 experiments will be used for the initial study.

9. **Data Analysis**: Analysis of Variance will be used to quantify the effects of the first three factors and regression will be used to quantify the effects of the number n of successive calls.

10. **Data Presentation**: The final results will be plotted as a function of the block size *n*.

Performance Evaluation Metrics

- For each performance study, a set of performance criteria or metrics must be chosen.

- Some commonly used metrics are:
  - ***turnaround time*** - the time between the submission of a batch job and the completion of its output
  - ***response time*** - the interval between an interactive user's request and the system response
  - ***throughput (or productivity)*** - the rate (requests per unit time) at which requests are serviced by the system
  - ***utilization of a resource*** - is a measured of the fraction of time the resource is busy servicing requests

Performance Evaluation Metrics

- ***reliability*** - measured by the probability of errors or the meantime between errors (MTTF - mean time to failure)
- ***availability*** - the percentage of total time during which the system is at the disposal of the users

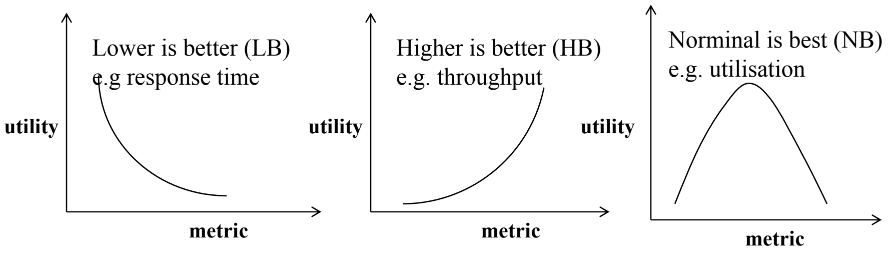
$$A = \frac{MTTF}{MTTF + MTTR}$$

MTTR - mean time to repair

- The relative importance of various measures depends on the application domains - general-purpose computing, high availability, real-time control, mission oriented, long-life, etc.

Performance Evaluation Metrics

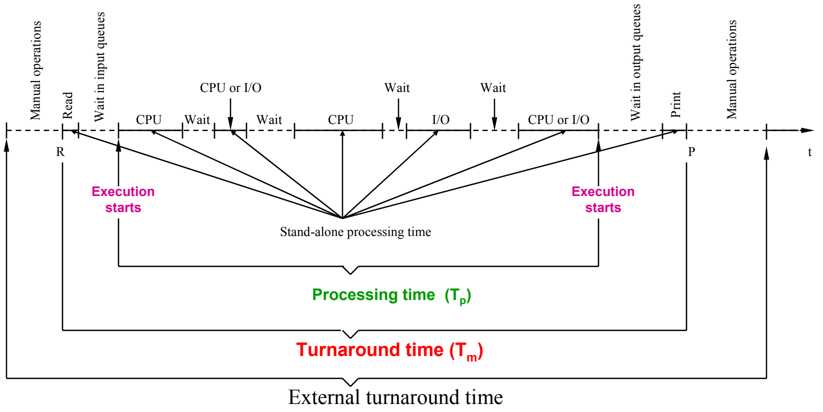
- Performance metrics can be categorised into three classes based on their utility function:
  - Lower is Better or LB
  - Higher is Better or HB
  - Nominal is Best or NB

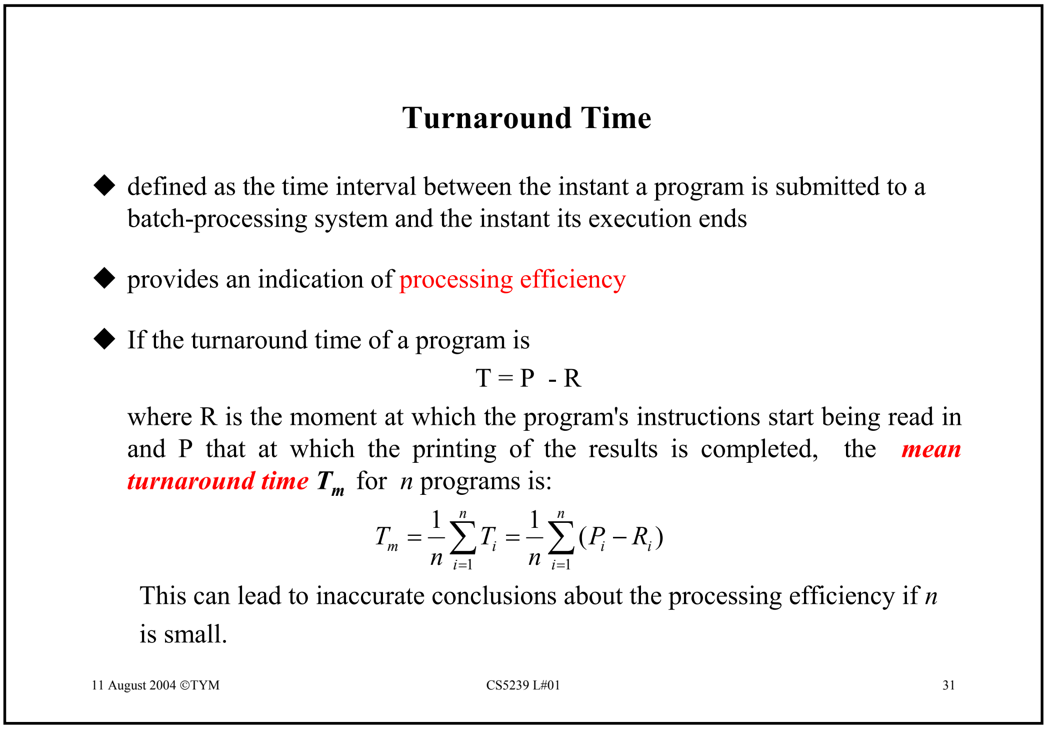


Cost/Performance Ratio

- A commonly used metric for comparing two or more systems in system procurement application
  - cost includes h/w and s/w licensing, installation and maintenance costs
  - performance is measured in terms of throughput under a given response time constraint

Turnaround Time





Turnaround Time

- defined as the time interval between the instant a program is submitted to a batch-processing system and the instant its execution ends

- provides an indication of processing efficiency

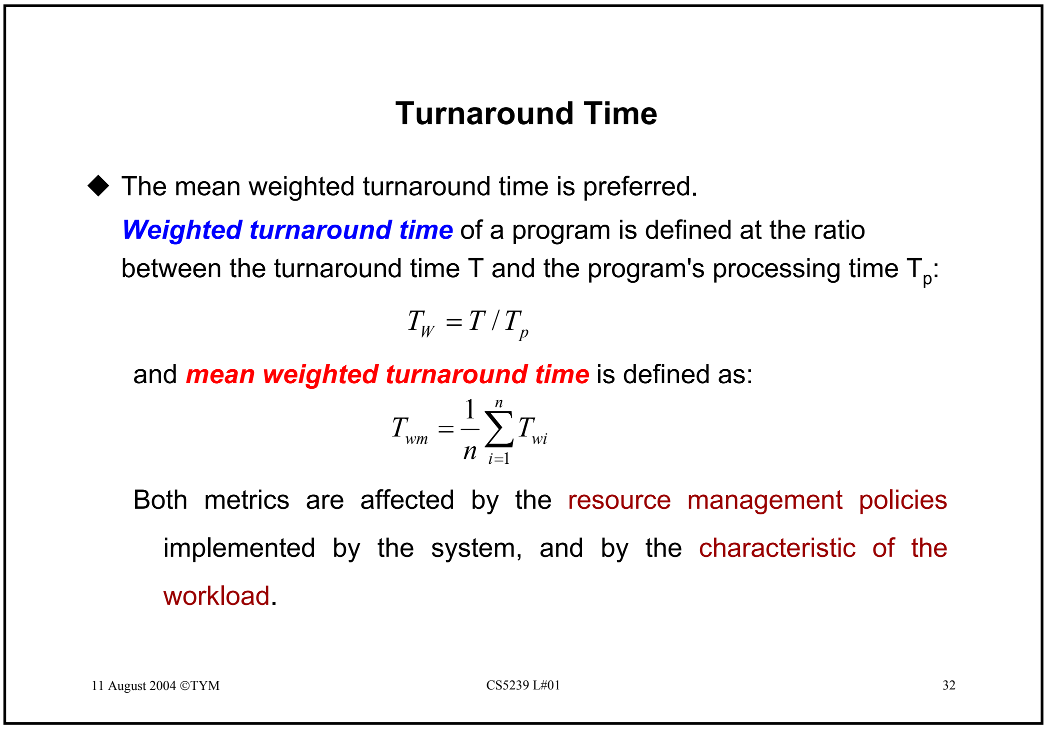
- If the turnaround time of a program is

  T = P - R

  where R is the moment at which the program's instructions start being read in and P that at which the printing of the results is completed, the ***mean turnaround time T_m*** for n programs is:

$$T_m = \frac{1}{n}\sum_{i=1}^{n} T_i = \frac{1}{n}\sum_{i=1}^{n}(P_i - R_i)$$

This can lead to inaccurate conclusions about the processing efficiency if n is small.



Turnaround Time

- The mean weighted turnaround time is preferred.

  Weighted turnaround time of a program is defined at the ratio between the turnaround time T and the program's processing time T_p:

$$T_W = T / T_p$$

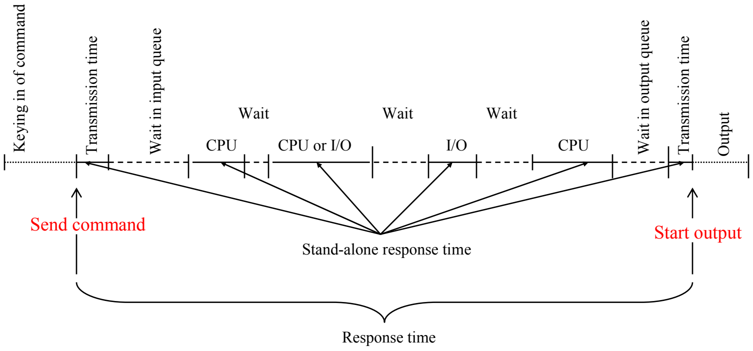
and ***mean weighted turnaround time*** is defined as:

$$T_{wm} = \frac{1}{n}\sum_{i=1}^{n} T_{wi}$$

Both metrics are affected by the resource management policies implemented by the system, and by the characteristic of the workload.

Response Time

- defined as the time interval between the instant the inputting command to an interactive system terminates and the corresponding reply begins to appear at the terminal



Keying in of command
Transmission time
Wait in input queue
Wait
CPU
CPU or I/O
Wait
I/O
Wait
CPU
Wait in output queue
Transmission time
Output
Send command
Start output
Stand-alone response time
Response time

Response Time

- response time depends very much on the type of the command the system executes:
  - ***light commands*** - requires less than one quantum of CPU time (e.g. text editing commands - insert, delete, modify; requests for information such as date and time, etc.
  - ***heavy commands*** - requires more than one quantum time to be executed (e.g. compilation, execution, sort, etc.)
- The mean response time R_m does not provide complete information about an interactive system's performance, i.e. variability of performance. The standard deviation of response time, σ, is commonly used:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (R_i - R_m)^2}{n}}$$

where R_i is the response time of the *ith* command.

Throughput (or Productivity)

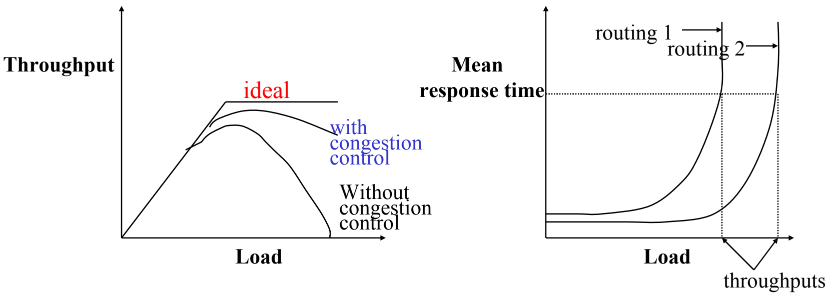
- defined as the amount of work performed by a system in a given unit of time

- examples:

batch systems	- jobs per second;
interactive systems	- requests per second;
CPUs	- MIPS or MFLOPS;
networks	- packets per second (pps) or bits per second;
transaction processing systems	- transactions per second.

Throughput

- Depending on the system, throughput is influenced by many factors:
  - e.g., in a computer network, good routing combined with congestion control schemes can improve throughput



Throughput (or Productivity)

- A general definition of throughput is:

  $$X = N_P / t_{tot}$$

  where N_P is the number of programs processed during the measurement interval t_{tot} . X gives an indication of the speed of execution for the set of N_P programs (workload).

- Throughput is influenced by the following factors:
  - the characteristic of the workload with which it is evaluated
  - speed of hardware and software components
  - degree of multiprogramming allowed by the hardware
  - configuration of the system
  - the resource allocation algorithm used

Everything should be made as simple as possible, but no simpler – attributed to Albert Einstein