CS-417
COMPUTER SYSTEMS MODELING

Spring Semester 2020
Batch: 2016-17
(LECTURE # 2)

FAKHRA AFTAB
LECTURER
DEPARTMENT OF COMPUTER & INFORMATION SYSTEMS ENGINEERING
NED UNIVERSITY OF ENGINEERING & TECHNOLOGY

Recap of Lecture 1

Computer System Performance Evaluation

Common Goals of Performance Evaluation

Model

Modeling Tools (Solution Techniques)

Emulation

Chapter # 1 (Cont'd)

COMPUTER SYSTEMS PERFORMANCE MODELING AND EVALUATION

TYPES OF MODELS

DYNAMIC MODEL

- Systems that evolve with time. Example: A model to gauge performance of a search engine.

DETERMINISTIC MODEL

- There are no probabilistic components in the systems.
- Example: The worst case analysis of an algorithm.

CONTINUOUS MODEL

- System's state changes continuously. Example: A chemical process.

STATIC MODEL

- Systems that don't evolve with time. Example: DC voltage-current relationship.

STOCHASTIC MODEL

- At least one component has probabilistic behavior.
- Example: Queuing Systems.

DISCRETE MODEL

- State changes only at discrete points in time. Example: Inventory Model

Performance Metric

Metric is an actual value that is used to describe performance.

There are two types of performance metric:

- System Oriented
- User Oriented

A) System Oriented Measures

These are the metrics important from the system administrator's perspective.

- **Throughput**

✓ No. of jobs completed in a unit time.

✓ Nature of job depends on the context or application.

✓ This metric gives the productivity of the system.

Some representative examples are given below:

System	Metric
OLTP	Transactions per second (tps)
Web site	HTTP requests/second Page view per second Bytes per second
E-commerce Site	Web Interactions Per Seconds (WIPS) Sessions per second Searches per second
Router	Packets Per Second
CPU	Millions of Instructions Per Second (MIPS) Floating Point Operations Per Second (FLOPS)
Disk	I/Os per second KB transferred per second
E-mail Server	Messages sent per second

- **Resource Utilization (ρ)**

✓ It indicates the fraction of time the resource is busy serving requests.

✓ Resource utilization can be defined in different manner. For Example:

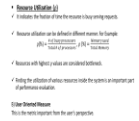
$$\rho(\%) = \frac{\#\ of\ busy\ processors}{Total\ \#\ of\ processors}\ ,\ \rho\ (\%) = \frac{Memory\ used}{Total\ Memory}$$

✓ Resources with highest ρ values are considered bottleneck.

✓ Finding the utilization of various resources inside the system is an important part of performance evaluation.

B) **User Oriented Measure**

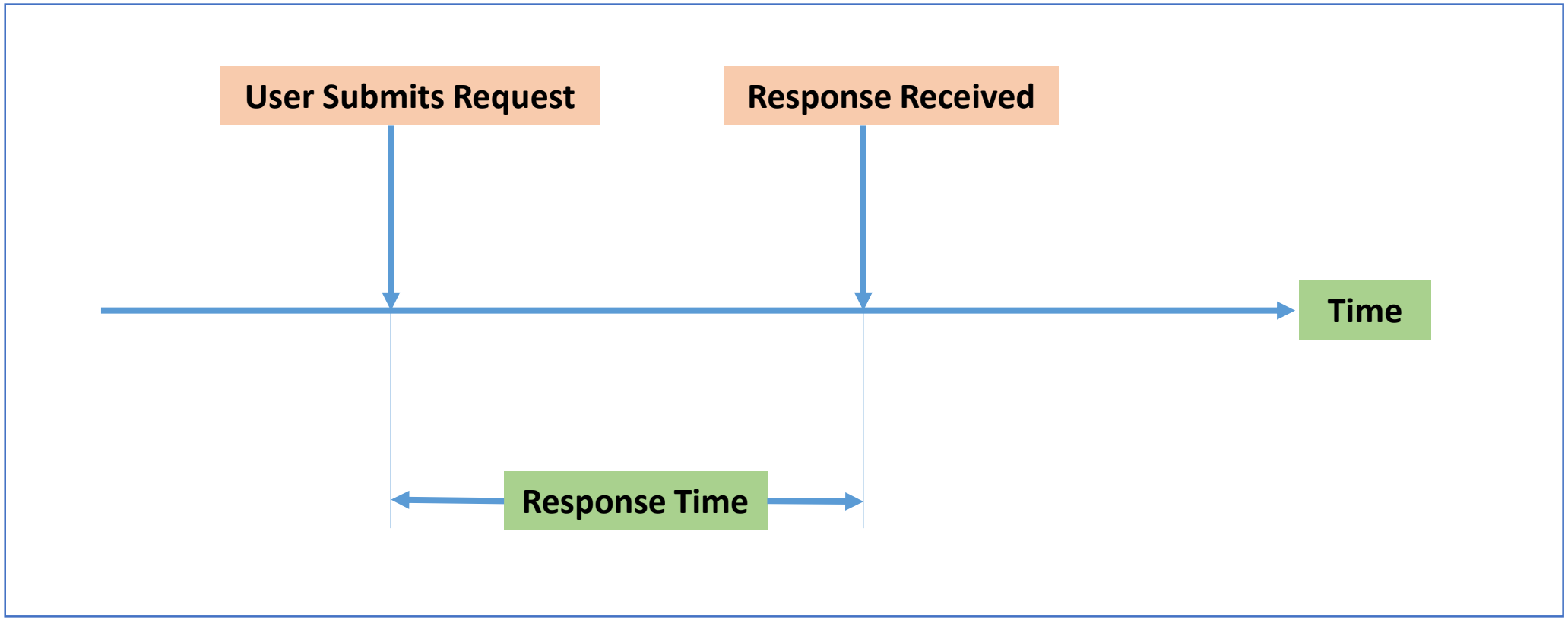
This is the metric important from the user's perspective.



▪ Response Time

✓The time elapsed between the request submitted & system's response received by the user.

✓For a batch stream, responsiveness is measured by *turnaround time*.



Workload (WL) or Load

- The *quantity* and *nature* of requests submitted to system during some given period of time to model or drive the system.

- e.g. the no. of requests submitted to database server per second i.e. the *intensity* of workload.

- WL also depends on nature of request.

- *No. of instructions per second* and the *mix of instruction types* presented for execution per second.

Duration of the Load

- Duration may be *all at once*, requiring the system to queue up the requests and perform them as resources become available.

- Duration could be *endless*, with the load continually refreshed to provide a constant saturation load to the system.

- Loads can be *periodic*, where the load is reentered after the prescribed period of time.

Controllable vs Uncontrollable parameters of Workload

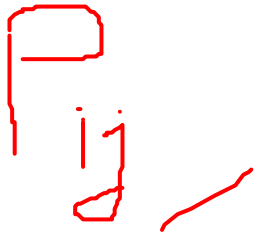
- Parameters that can be controlled by the system designer/ administrator, e.g., the scheduling disciplines, inter-connections between devices and resource allocation policies.

- Uncontrollable parameters, e.g., the inter-arrival times and service demands of incoming jobs.

- All these inputs are referred to as the *workload.*

Workload Characterization

- It involves deciding about
  - different *aspects* of the workload (controllable and uncontrollable parameters etc.),
  - level of *detail* for recording the workload, and
  - *representation* of the workload.



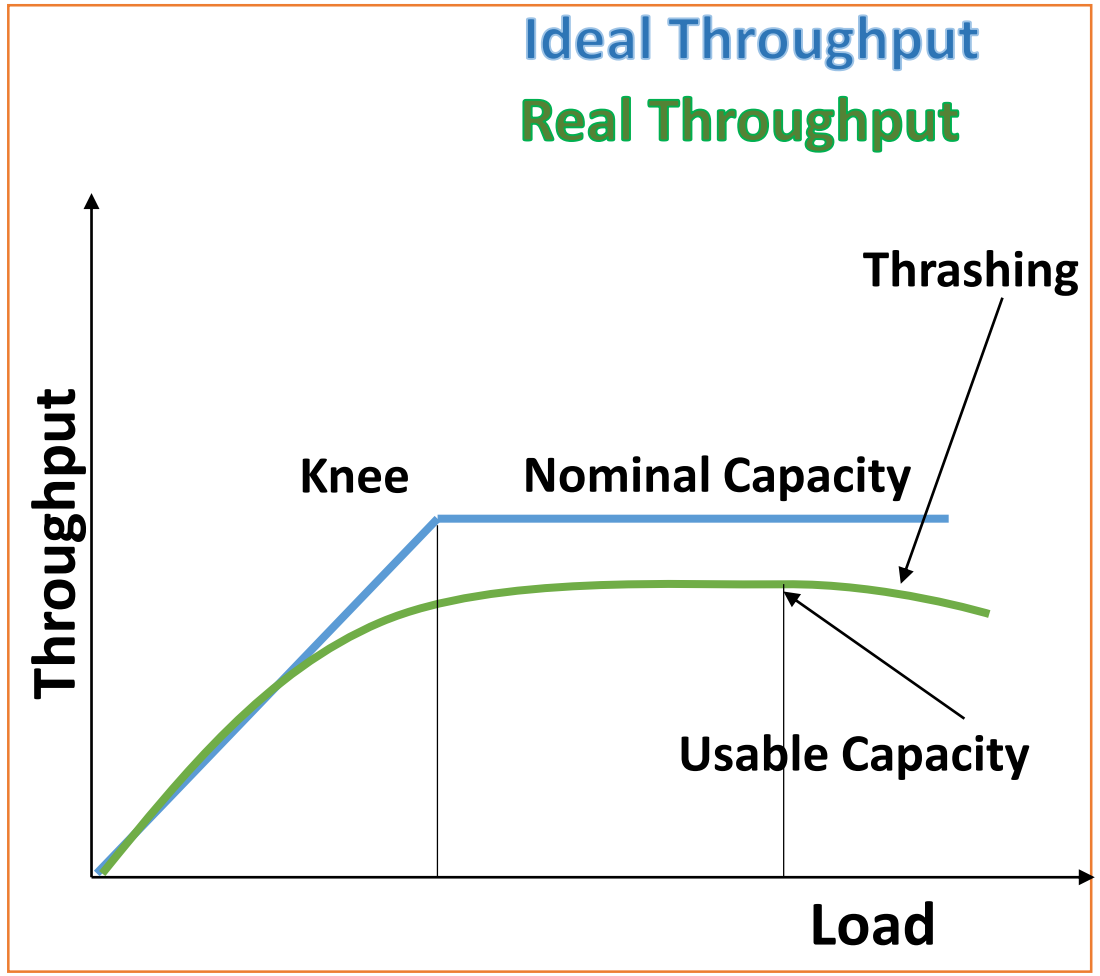
- WL characterization only builds a model of real workload, since not every aspect of real WL may be captured or is relevant.

- For reasons of cost-effectiveness and robustness, it is desirable to keep model (hence the WL) as abstract as possible.

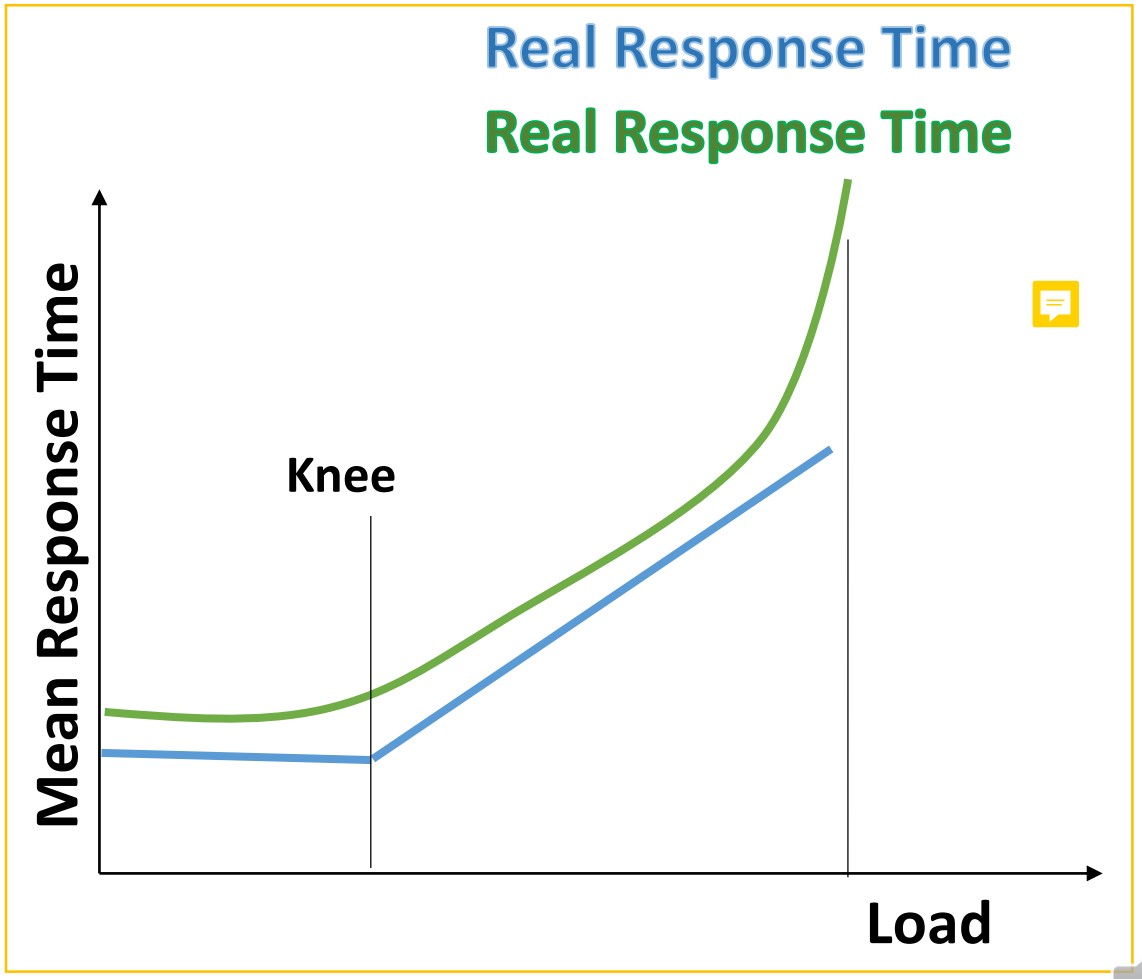
Workload Characterization

- A WL model may be executable or non-executable.
  - E.g. recording the arrival instants and service durations of jobs creates an executable model,
  - whereas only determining the distributions creates a non-executable model.
  - An executable model need not be a record of inputs, it can also be a program that generates the inputs.
  - Executable WLs are useful in direct measurements and trace-driven simulations.
  - Non-executable WLs are useful for analytic modeling and distribution-driven simulation.

Throughput vs. Workload



Mean Response Time vs. Workload



- **Efficiency:**

It is defined as the ratio of usable performance to maximum (theoretical/nominal) performance.

- **Stretch Factor:**

The ratio of response time at a particular load to that at the minimum load is called *stretch factor*.

For a time sharing system, for example, stretch factor is defined as the ratio of response time with multiprogramming to that without multi-programming

Execution Time

This is the ultimate performance measure.

I) Wall Clock Time:

The wall clock time measures the total time that a user would have to wait to obtain the results produced by the program. That is, the measurement includes the time spent waiting for input/ output operations to complete, memory paging, and other system operations performed on behalf of this application, all of which are integral components of the program's execution.

II) CPU Time:

This is the time CPU spends on process execution.

- Both are reported generally.
- Measure a program's total elapsed execution time several times and report at least the mean and variance of the times

Normalization Measures

- **Speed up:**

The speed up of System 2 w.r.t System 1 denoted as $S_{2,1}$ is defined as:

$$S_{2,1} = R_2/R_1$$

where 'R' is the rate metric.

If 'W' is the work then, R = W/T

Let $W_1 = W_2 = W$:

$$S_{2,1} = \frac{W/T_2}{W/T_1} = \frac{T_1}{T_2}$$

- If '2' is faster than '1', $T_2 < T_1$ and $S_{2,1} > 1$
- If '2' is slower than '1', $T_2 > T_1$ and $S_{2,1} < 1$

Normalization Measures (Cont'd)

- **Relative Change:**

$$\Delta_{2,1} = \frac{R_2 - R_1}{R_1}$$

$$\Delta_{2,1} = \frac{W/T_2 - W/T_1}{W/T_1}$$

$$\Delta_{2,1} = \frac{T_1 - T_2}{T_2}$$

$\Delta_{2,1}$ is positive if System 2 is faster

$\Delta_{2,1}$ is negative if System 2 is slower

Re-visiting Throughput Metrics

Millions of Instructions Per Second (MIPS)

- It is an attempt to develop a rate metric for computer systems that allows a direct comparison of their speeds.
- If a machine executes IC instructions in time ET then MIPS metric can be calculated as:

$$\text{MIPS} = \frac{IC}{ET * 106}$$

Millions of Floating-Point Operations Per Second (MFLOPS)

- It is an attempt to correct the primary shortcoming of the MIPS metric by more precisely defining the unit of 'computation' performed by the computer system when executing a program.
- If a machine executes OP floating point operations in time ET then MFLOPS metric can be calculated as:

$$\text{MFLOPS} = \frac{OP}{ET * 106}$$

Characteristics of a Good Performance Metric

i) Linearity – a metric is linear if its value is proportional to actual performance.

ii) Reliability – a metric is reliable if system A outperforms system B always when the corresponding values of the metric indicated so.

iii) Repeatability – reproducible and deterministic.

iv) Ease of measurement – if not easily measurable; chances are that it will be measured incorrectly & analyst will not use it.

v) Consistency – a metric is consistent if its definition is same across all system configuration.

vi) Independence – a metric should be insensitive to tweaking by vendors to be used in their benefit in order to befool the customer.