Introduction to Computer Performance Evaluation

Dr. Jozo Dujmović
Department of Computer Science
San Francisco State University

Why to Study CPE?

Computer science job areas:

- **Software engineering** (general purpose software design, development, testing, installation, use, and maintenance)
- **Domain expertise** (software development based on expertise in a specific domain of applications; e.g. graphics, games, artificial intelligence, OS, DB, compilers, protocols, numerical methods, etc.)
- **Systems** (non-programming jobs)
  - System/network/DB administration
  - Security management
  - Performance management (measurement, modeling, tuning, optimization, capacity planning, benchmarking, and system evaluation, comparison, and selection)

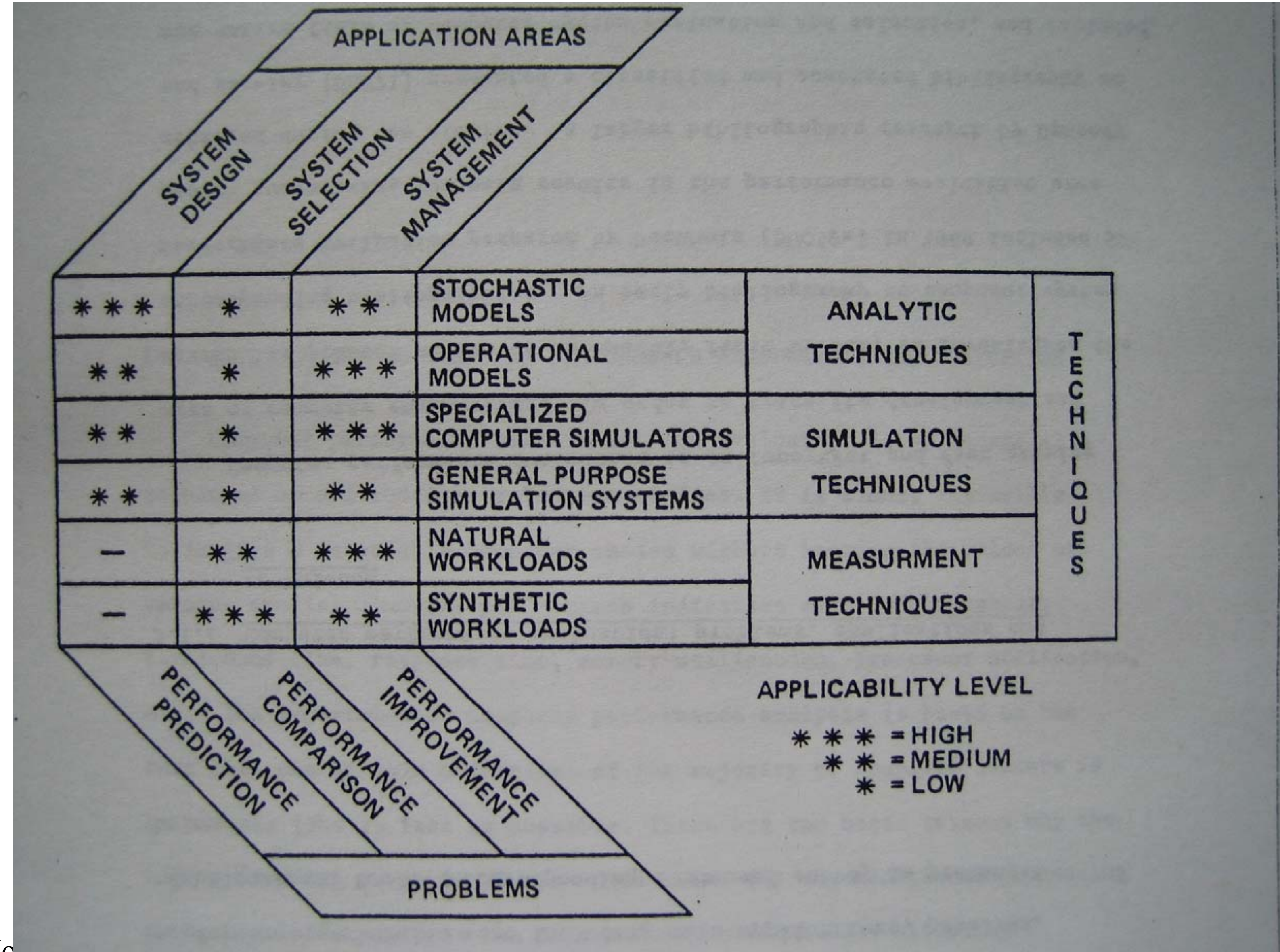
More Reasons to Study CPE

- All advances in computer architecture have the main reason to increase system performance
- All advances in the area of operating systems also have the main goal to increase system performance
- Understanding computer performance is indispensable for understanding computer architecture, operating systems, and the functioning of computer systems

Main CPE Areas

- Analytical performance models
  - Stochastic models ($T \to \infty$)
  - Operational models ($T < \infty$)
- Simulation models
  - General purpose simulation languages and systems
  - Specialized and home-made computer simulators
- Performance measurement (benchmarking)
  - Workload characterization
  - Benchmarking and system comparison with natural and synthetic workloads
- Performance management
  - System tuning
  - Capacity planning, design and sizing

CPE Application Areas



Selecting a CPE Method

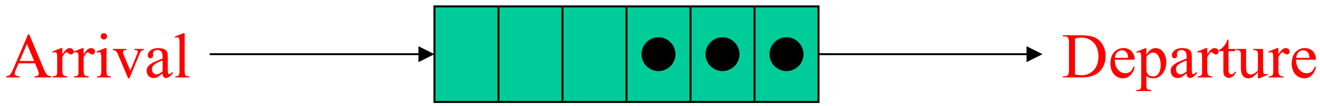
- **Analytic models**: inexpensive, good for experimenting and understanding dynamic phenomena, but sometimes have limited accuracy.
- **Simulation**: used when accurate analytic models are not available, too complex, or insufficiently accurate
- **Measurements**: reflect actual performance of a specific (measured) system for a specific workload. Based on a variety of software tools.

Performance Models

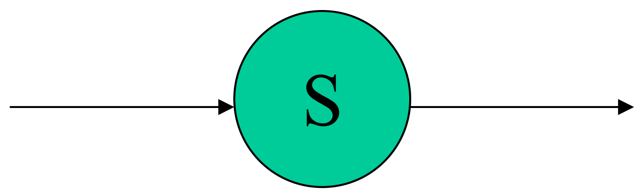
- Performance models of computer components:
  - Processors
  - Memory
  - Disks
  - Tapes
  - I/O devices
  - Compilers
  - Operating system
- Performance models of computer systems:
  - Batch processing
  - Interactive systems
  - Networks (servers, clients, communication links)

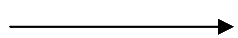
Basic Components of Performance Models

- Queue = memory element that can hold up to n service requests (n = queue capacity):



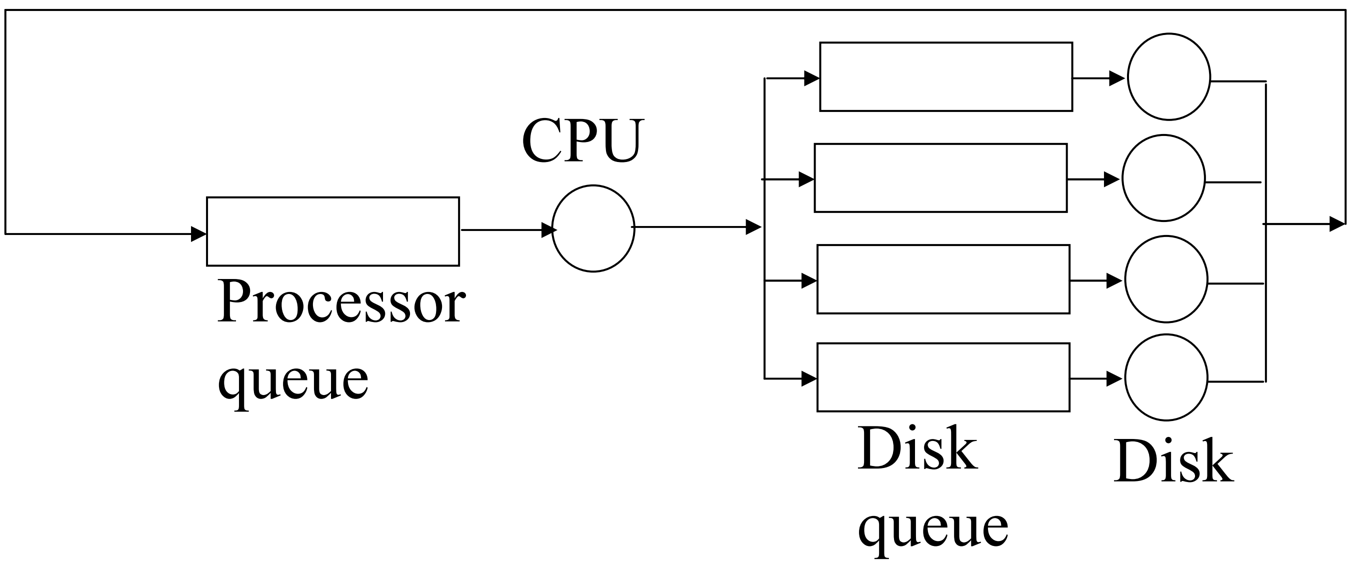
- Server = a processing unit that provides uninterrupted S time units of service to a service request:



- Link = connection between queues and servers: 

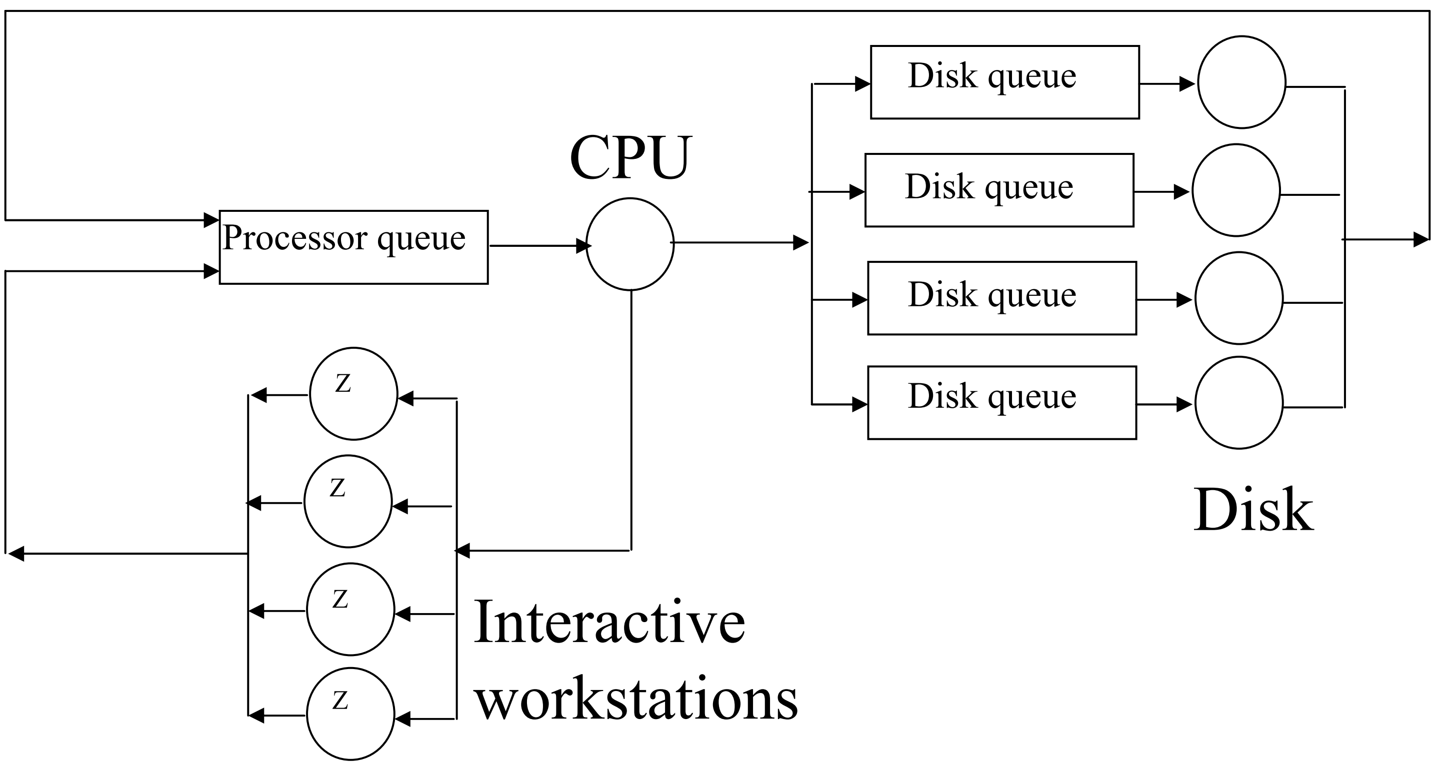
Queuing Network Models (1/2)

Workload = sequential file processing



Queuing Network Models (2/2)

Workload = interactive transaction processing



Basic Performance Indicators

- General indicators
  - Observation interval (T)
  - Think time (Z)
  - Arrival rate (λ)
  - Interarrival time (a)
  - Service time (S)
  - Service rate (μ)
  - Visits and Demand (V,D)
  - Number of jobs in a system (J)
  - Throughput (X)
  - Server utilization (U)
  - Queue length (Q)
  - Response time (R)
- Special indicators
  - Memory size (M)
  - Memory speed (v_{mem})
  - Instruction mix time (T_{mix})
  - Processor speed (v_p)
  - Disk seek time (T_{seek})
  - Disk latency time (T_{rd})
  - Data transfer time (T_{dt})
  - Disk access time (T_a)
  - Disk/tape transfer rate
  - Program size (LOC)
  - Compilation rate (ips)
  - Code density (m_1)

Observation Interval (T)

- Time from the beginning to the end of observation of dynamic behavior of a computer system. Analytic models describe the behavior of system during the observation interval.
- Operational models use finite T.
- Stochastic models use infinite T.
- Unit = second

Think time (Z)

- Time necessary for an interactive user to perform the following actions:
  - Read and understand data displayed on a screen
  - Decide what action to perform
  - Enter data/command necessary to specify the next action
  - Send request (press Return key or click mouse)
- Unit = second
- Typical range = [2sec, 60 sec]

Arrival rate (λ)

- Customers (transactions, jobs) arrive randomly to a service center
- Arrival rate is the average number of customers (service requests) that arrive per time unit (during the observation interval T)
- Unit = 1/sec

Interarrival time (*a*)

- The time between to successive customer arrivals is a random value (customers are assumed to arrive independently and randomly)
- a = mean value of the random interarrival time $a = 1/\lambda$
- Unit = second

Service Time (S)

- Average interval of time during which a server (e.g. processor, or disk) delivers an uninterrupted service to a customer.
- Customers frequently visit the same server multiple times; in such cases S denotes the average time quantum received per visit.
- Unit = second

Service rate (μ)

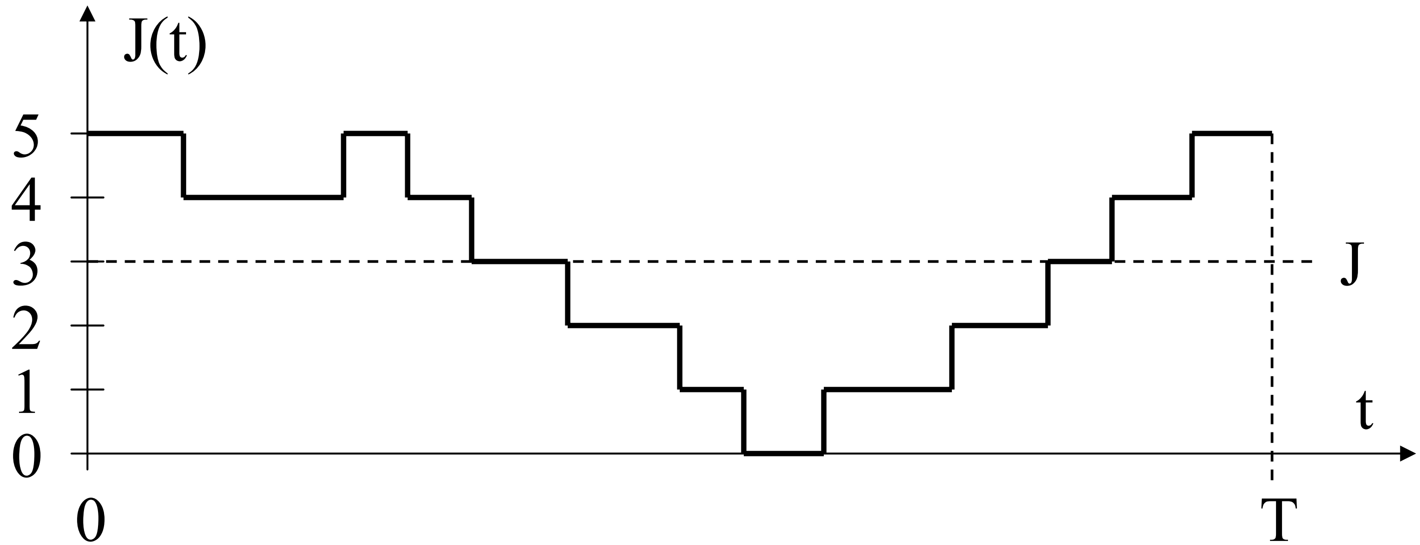
- The maximum number of (average) service requests a server can serve per time unit.
- Unit = 1/sec
- $\mu = 1/S$
- Actual service rate of a server is usually less than μ (the service rate μ is obtained if the server is permanently active, 100% in use)

Visits and Demand (V,D)

- Suppose that a job visits a server (e.g. disk) V times in order to complete processing (e.g. the job terminates after V disk accesses)
- Demand D is the total accumulated service time that a job receives from a server during V visits: $D = VS$

Number of jobs in a system (J)

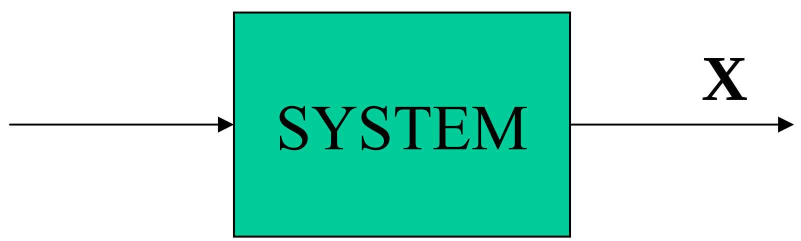
- Jobs come and go, and the number of jobs in a system/subsystem/queue is a function of time:



- J is the average number of jobs in a system during the observation interval T.

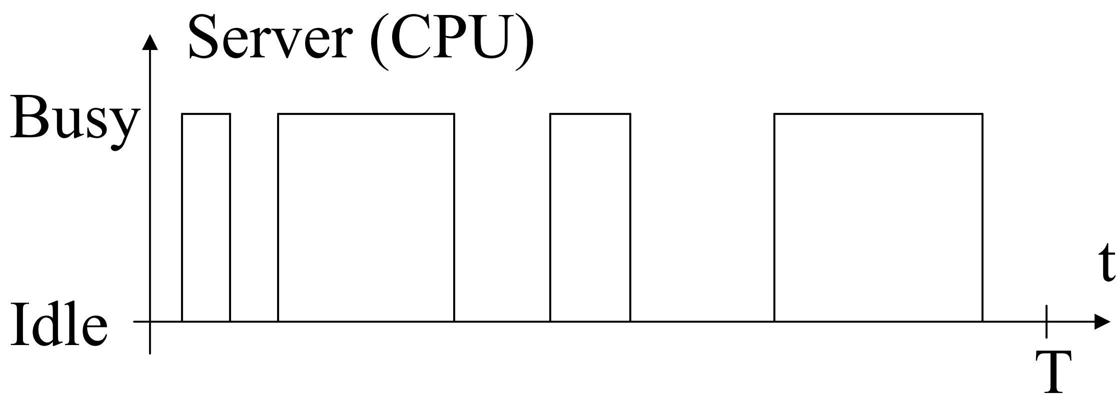
Throughput (X)

- Average number of service requests that a service center processes per second
- Unit = 1/sec
- Throughput is always measured at the output of a system:



Server utilization (U)

- During the observation interval T a server can be:
  - Busy (serving customers) during time B
  - Idle (waiting for customers) during time T-B
- Utilization is a fraction of observation time when the server was busy: **U = B/T**

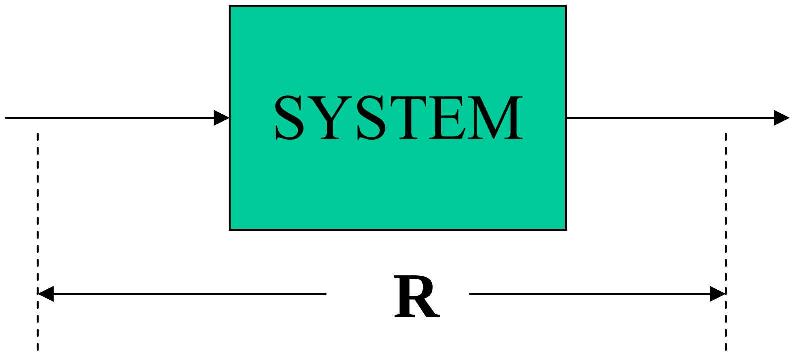


Queue length (Q)

- The number of customers in a queue is a random value and varies during the observation interval T.
- Queue length Q is the average number of customers in a queue during the observation interval T.

Response time (R)

- R is the average time a customer spends in a system (from arrival to departure).
- R includes waiting in queues and receiving service.
- Waiting time increases when the number of customers increases (example: barber shop)



Memory size (M)

- Total number of available bytes (words) in a memory (main memory, cache memory, disk memory, etc.)
- Units: B, KB, MB, GB, TB, PB
- K = kilo = 1024 bytes = $2^{10} \approx 10^3$
- M = mega = K*K = $K^2 = 2^{20} \approx 10^6$
- G = giga = K*M = $K^3 = 2^{30} \approx 10^9$
- T = tera = K*G = $K^4 = 2^{40} \approx 10^{12}$
- P = peta = K*T = $K^5 = 2^{50} \approx 10^{15}$

Address Space

- 32-bit address field restricts the address space
- $2^{32}=2^2\times 2^{10}\times 2^{10}\times 2^{10}=4\times K\times K\times K=4GB$
- To access larger memory we need wider address field (64-bit technology)

Memory Speed (v_{mem})

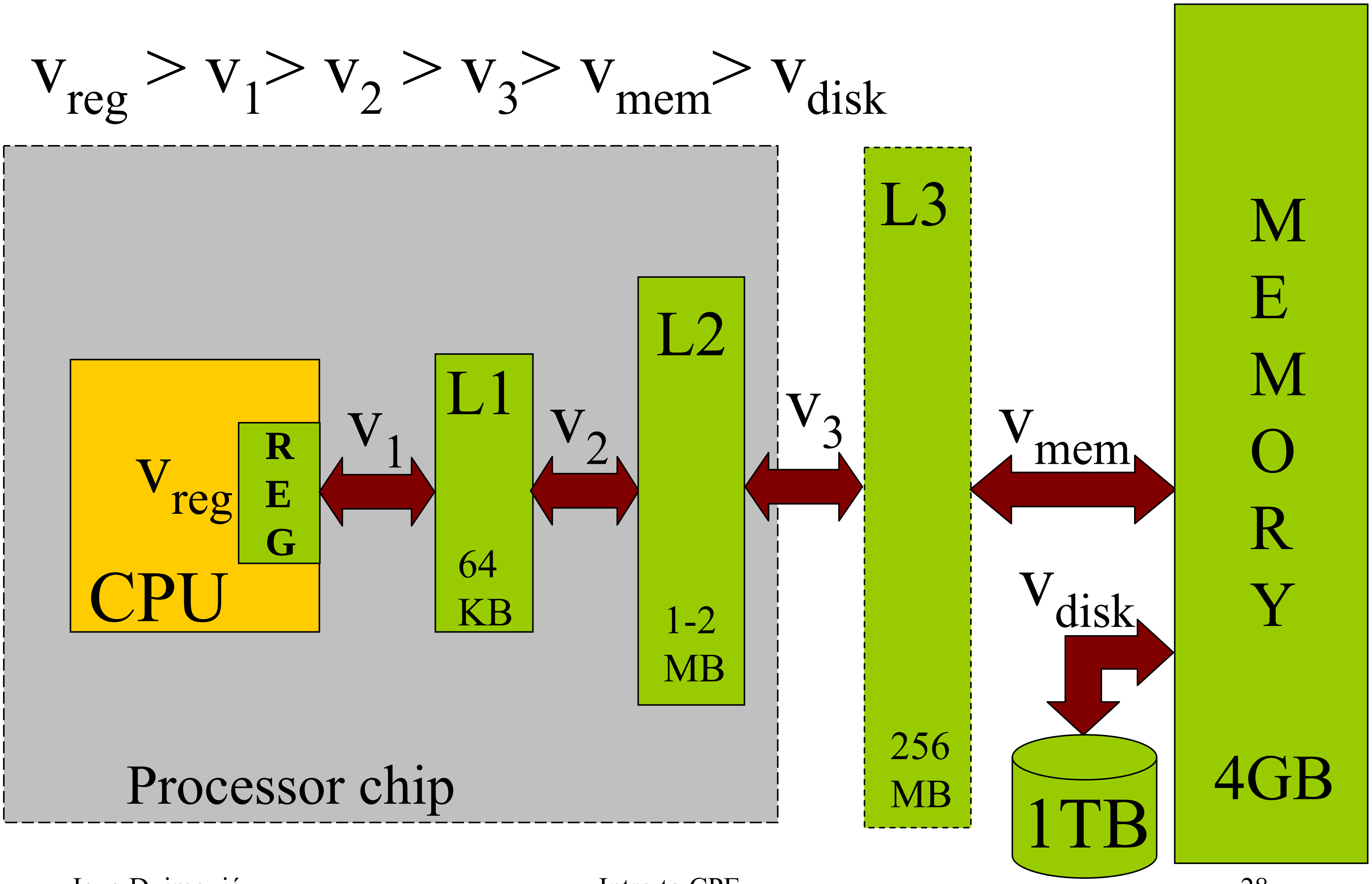
- Number of bytes that can be moved to/from memory in a time unit
- Affected by bus speed
- Unit = B/sec, MB/sec, B/μsec

Features That Affect Processor Performance

- Clock speed (determines the rate of execution of machine instructions)
- L2 cache size (on-chip cache; multiple processors can have their individual L2 cache memories (1-2MB)
- Front side bus speed (speed of bus that communicates with memory and graphics 500-800MHz)
- Hyper-threading (execution of two or more software threads in a multiplexed/parallel way)
- Dual-core technology (two parallel processors with separate caches sharing the same chip)

Levels of Cache Hierarchy

$$v_{reg} > v_1 > v_2 > v_3 > v_{mem} > v_{disk}$$



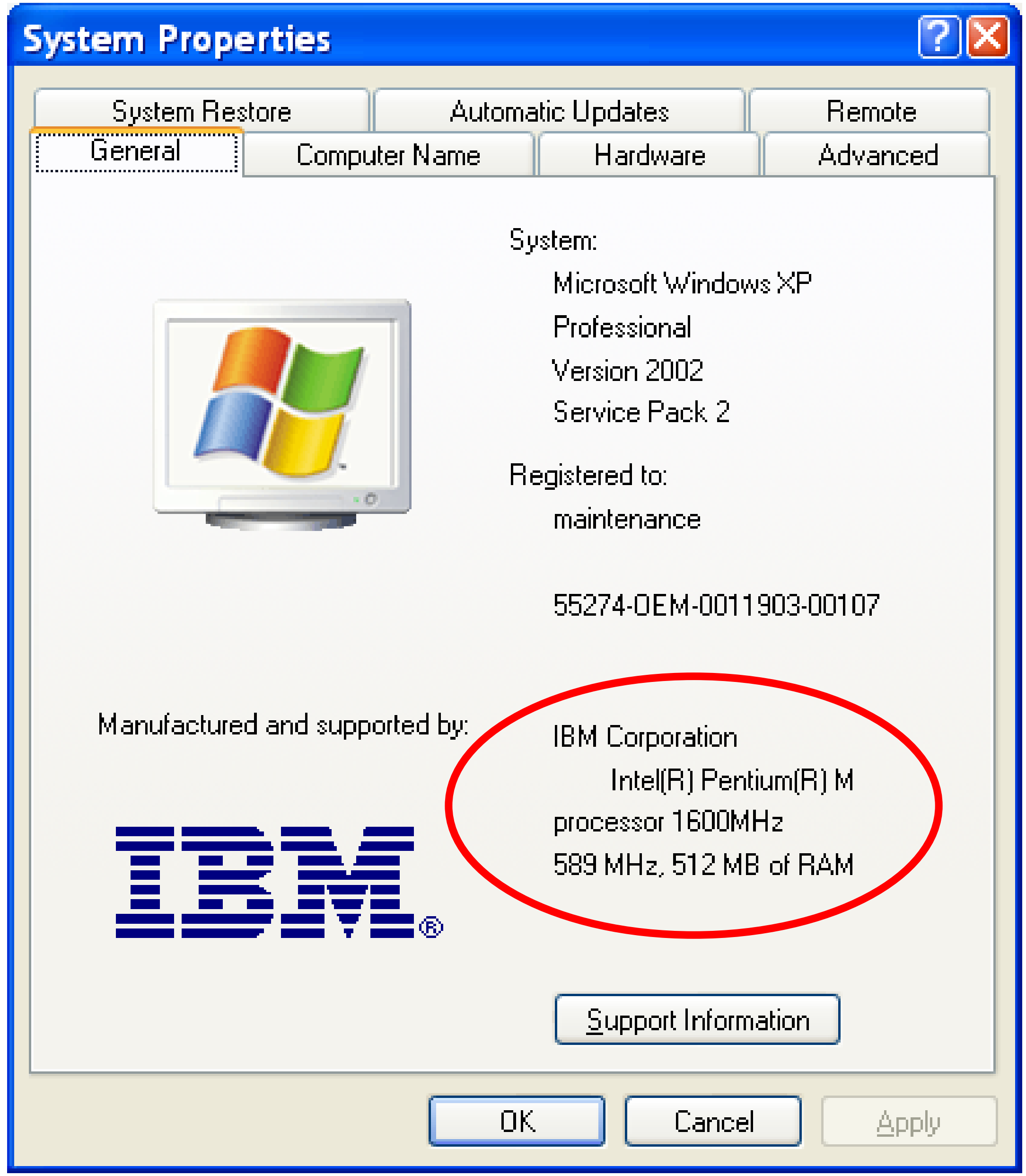
Instruction Mix Time (T_{mix})

- Time necessary for processor to process an average machine instruction, assuming basic model of sequential execution
- T_{mix} depends on
  - Instruction type
  - Addressing mode
  - Number and size of arguments

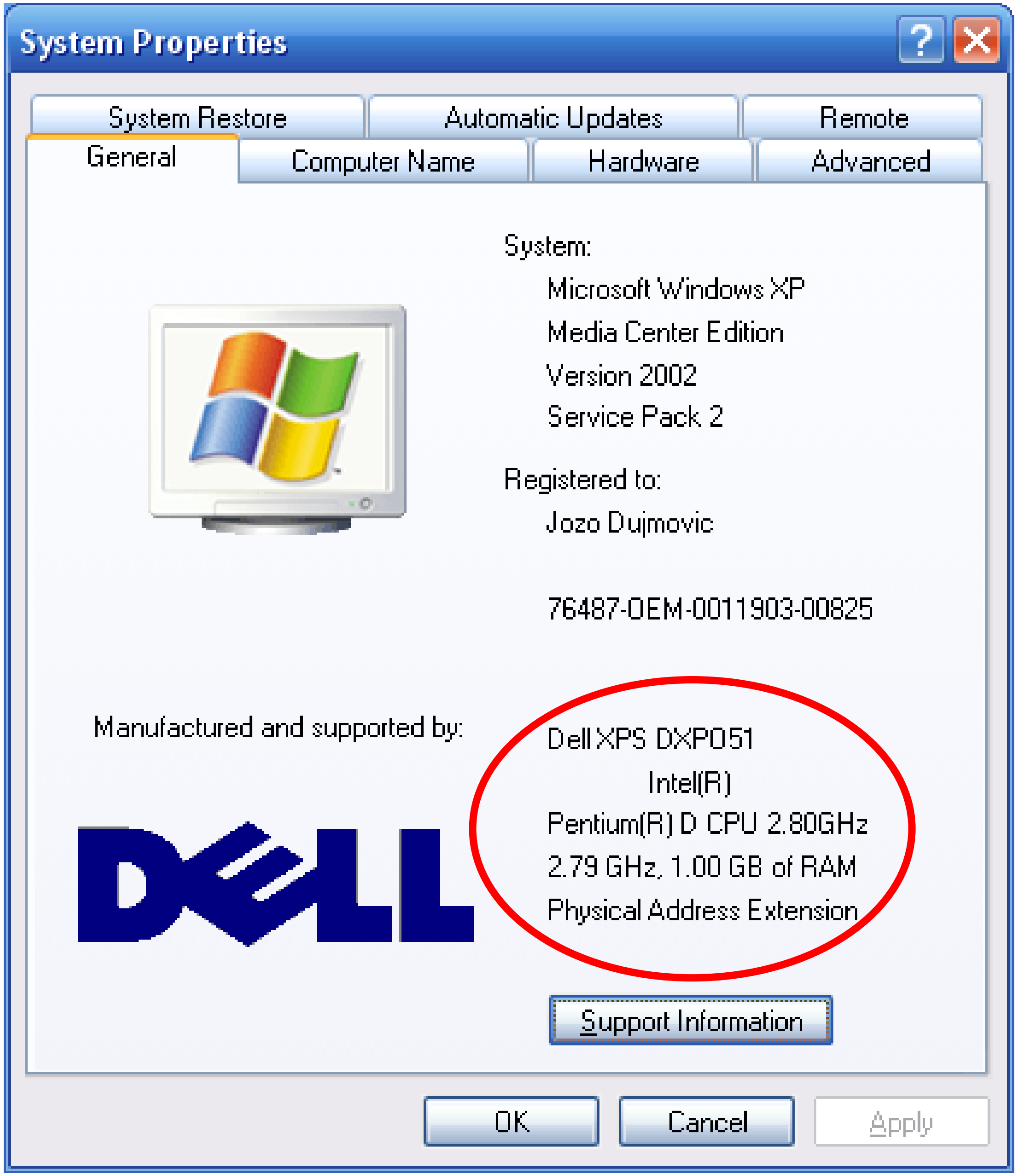
Processor Speed (v_p)

- Average number of machine instructions executed per second
- $v_p = 1/ T_{mix}$
- Units = ips (instruction per second), 1/sec, MIPS (million of instructions per second)
- Sometimes the unit is Mflops (million of floating point operations per second)

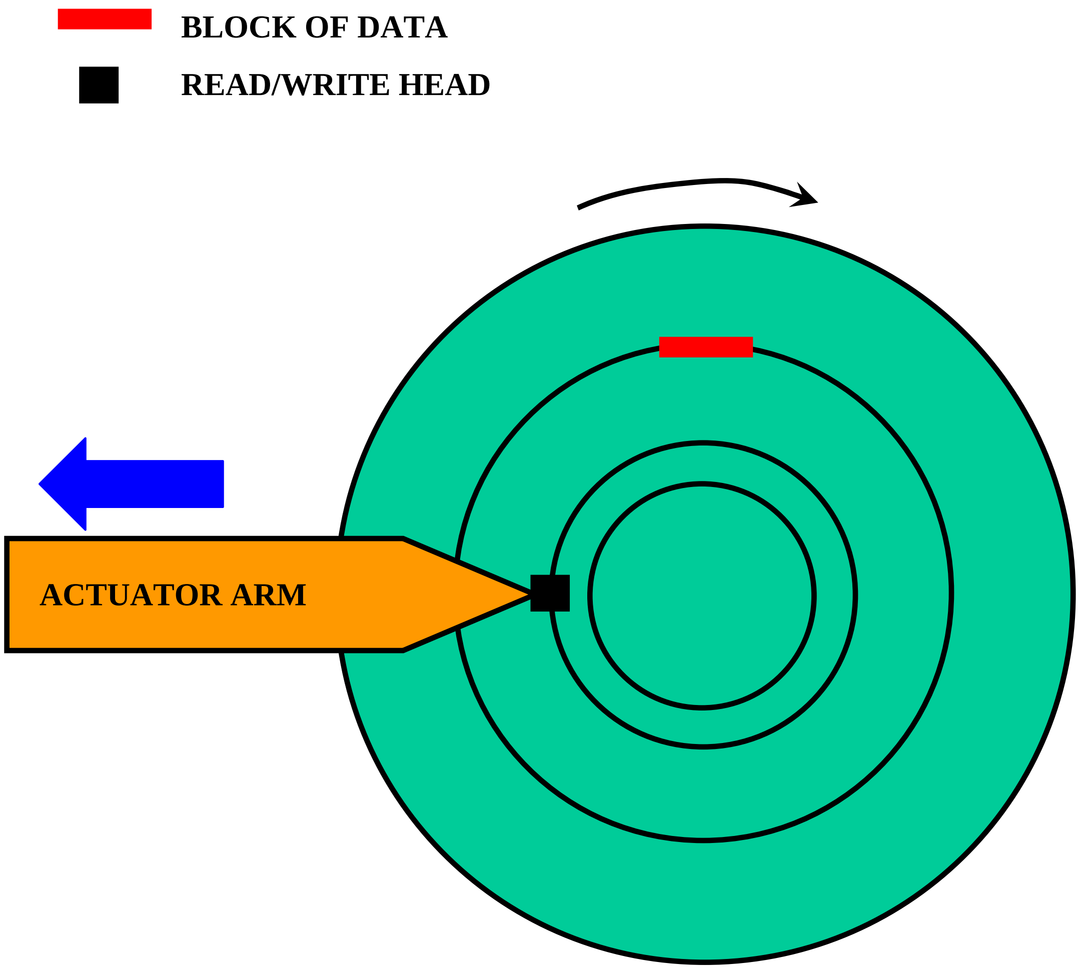
An example of system properties for a single processor machine



An example of system properties for a 2-processor machine



Disk operations



1. Seek: move head from the current track to the destination track
2. Latency: wait for data (rotational delay)
3. Transfer data

Disk seek time (T_{seek})

- Average time necessary to move the I/O mechanism from its current random position to a random destination position (cylinder)
- Minimum seek time is 0.
- Maximum seek time is the time necessary for moving over all cylinders
- Movement over large distances uses higher speed than the movement for short distances
- Typical mean values: 3ms – 10ms

Disk latency time (T_{rd})

- At the end of seek time the data can be anywhere on circular track. Latency or the rotational delay time of a disk unit is a time necessary for rotating data to come under the I/O head.
- The mean value of T_{rd} is ½ of the revolution time = 30/N (N = number of revolutions per minute = 2400, 3600, 5400, 7200, 10000, 15000…)
- Unit = second or ms
- Typical values: 2ms-5ms

Data transfer time (T_{dt})

- After the the seek and rotational delay time the I/O mechanism is properly positioned for data transfer to/from disk unit.
- Data transfer time depends on the size of the transferred block of data.
- A full revolution of disk is needed to transfer data that occupy the whole track.

Disk access time (T_a)

- Disk access time is the sum of seek time and the rotational delay time (from the beginning of seek to the beginning of data transfer)
- $T_a = T_{seek} + T_{rd}$
- Unit = second, ms

Disk access and transfer time (T_{at})

- Disk access and transfer time is the time from the beginning of seek to the end of data transfer
- $T_{at} = T_{seek} + T_{rd} + T_{dt} = T_a + T_{dt}$
- Unit = second, ms

Disk/tape transfer rate

- Data transfer rate = (mechanical speed of tape) * (density of data)
- Mechanical speed (=) inch/sec
- Data density (=) B/inch
- Data transfer rate (=) (inch/sec)(B/inch) (=) B/sec (or MB/sec)

Seagate ST936751SS Disk

- Track to track seek time = 0.2 ms
- Average seek time = 2.9 ms
- Full stroke seek time = 5 ms (estimated)
- RPM = 15000
- Read/Write transfer rate = 79-112 MB/sec (low rate is for inner tracks, and high rate is for outer tracks)

Solid State Disks

- Based on flash memory technology
- Typical performance parameters
  - Access time = 0.1 ms
  - Read rate = 40-70 MB/s
  - Write rate = 28-40 MB/s
- Read performance of SSD can be 20 times faster than the magnetic disk
- Random write performance of SSD can be 15 times slower than the magnetic disk

Program size (LOC)

- LOC = Lines Of Code (in high level language)
- LLOC = Logical Lines Of Code (number of language constructs, such as for, if, etc.)
- PLOC = Physical Lines Of Code (number of physical lines, i.e. the number of ‘\n’ characters)

Compilation Rate (ips)

- Average number of instructions translated by a compiler per second
- Unit = instructions per second (ips)
- Note: This indicator is meaningful only for linear compilation process, and for large programs. Each compiler needs an initialization time (typically < 1 second)
- It depends on the complexity of code, and the performance of hardware.

Code Density (m_1)

- Code density is the average number of memory bytes per one typical HLL instruction
- Unit = B/LLOC or B/PLOC
- Note: each program also needs a constant initial space for libraries and and program initialization code.
- $M = m_0 + m_1 N$ (N = LLOC or PLOC)

CPE Methods

- Analytic methods (fast, inexpensive, sometimes with modest accuracy)
- Simulation (more time consuming and more expensive)
- Measurement techniques (time consuming but accurate)

Conclusions

- The study of CPE complements the study of operating systems and computer architecture
- Main components of CPE curriculum:
  - Performance of hardware/software components
  - Analytic models of computer systems
  - Performance measurement: tools and benchmarks
  - Simulation: RNG's and discrete event simulators
  - Performance management (system tuning, sizing, comparing, selecting, and capacity planning)
- Performance analyst jobs are not outsourced