Case Study I: A Database Service

Prof. Daniel A. Menascé
Department of Computer Science
George Mason University
www.cs.gmu.edu/faculty/menasce.html

1

Copyright Notice

- Most of the figures in this set of slides come from the book “Performance by Design: computer capacity planning by example,” by Menascé, Almeida, and Dowdy, Prentice Hall, 2004. It is strictly forbidden to copy, post on a Web site, or distribute electronically, in part or entirely, any of the slides in this file.

2

DB Server Log Sample

	(msec) CPU	(count) Disk 1	(count) Disk 2	TR ID
1	116.824	9	9	18
2	64.383	7	9	37
3	35.403	7	9	58
4	104.409	8	12	77
5	119.793	9	8	19
6	47.956	5	7	1

OS Performance Measurements

Resource	Utilization (%)
CPU	45
Disk 1	75
Disk 2	65

Basic Statistics for the DB Service Workload

$CV = \frac{\text{standard deviation}}{\text{mean}}.$

	CPU Time (msec)	No. I/Os Disk 1	No. I/Os Disk 2
Mean	238.2	51.38	44.85
Standard Deviation	165.9	27.0	26.4
Sample Variance	27510.4	728.7	698.1
Coeff. of Variation	0.696	0.525	0.677
Minimum	23.6	5	7
First Quartile (Q1)	104.4	33	26
Median (Q2)	151.6	63	39
Third Quartile (Q3)	418.1	72	68
Maximum	507.5	85	92
Range	483.9	80	85
Largest	507.5	85	92
Smallest	23.60	5	7
Sum	47640.8	10275	8969

Quantiles (quartiles, percentiles) and midhinge

- Quartiles: split the data into quarters.
  - First quartile (Q1): value of Xi such that 25% of the observations are smaller than Xi.
  - meadian – Second quartile (Q2): value of Xi such that 50% of the observations are smaller than Xi.
  - Third quartile (Q3): value of Xi such that 75% of the observations are smaller than Xi.
- Percentiles: split the data into hundredths.
- Midhinge: $Midhinge = \frac{Q_3 + Q_1}{2}$

Example of Quartiles

1.05
1.06
1.09
1.19
1.21
1.28
1.34
1.34
1.77
1.80
1.83
2.15
2.21
2.27
2.61
2.67
2.77
2.83
3.51
3.77
5.76
5.78
32.07
144.91

Q1	1.32
Q2	2.18
Q3	3.00
Midhinge	2.16

In Excel:
Q1=PERCENTILE(<array>,0.25)
Q2=PERCENTILE(<array>,0.5)
Q3=PERCENTILE(<array>,0.75)

Example of Percentile

1.05
1.06
1.09
1.19
1.21
1.28
1.34
1.34
1.77
1.80
1.83
2.15
2.21
2.27
2.61
2.67
2.77
2.83
3.51
3.77
5.76
5.78
32.07
144.91

80-percentile	3.613002

In Excel:
p-th percentile=PERCENTILE(<array>,p)
(0=p=1)

The 50th percentile is called the median.

Range, Interquartile Range, Variance, and Standard Deviation

- Range: $X_{\max} - X_{\min}$
- Interquartile Range: $Q_3 - Q_1$
  - not affected by extreme values.
- Variance:

$$s^2 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}$$

In Excel:
s^2=VAR(<array>)

- Standard Deviation:

In Excel:
s=STDEV(<array>)

$$s = \sqrt{\frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}{n-1}}$$

9

Meanings of the Variance and Standard Deviation

- The larger the spread of the data around the mean, the larger the variance and standard deviation.
- If all observations are the same, the variance and standard deviation are zero.
- The variance and standard deviation cannot be negative.
- Variance is measured in the square of the units of the data.
- Standard deviation is measured in the same units as the data.

10

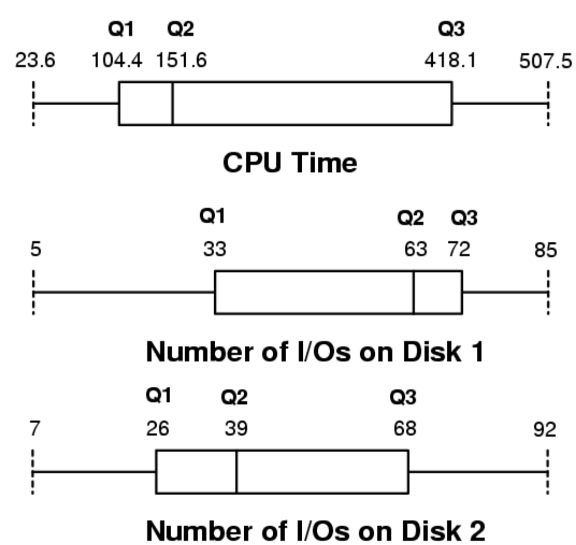
Coefficient of Variation

- Coefficient of variation (COV) : $s / \overline{X}$
  – no units

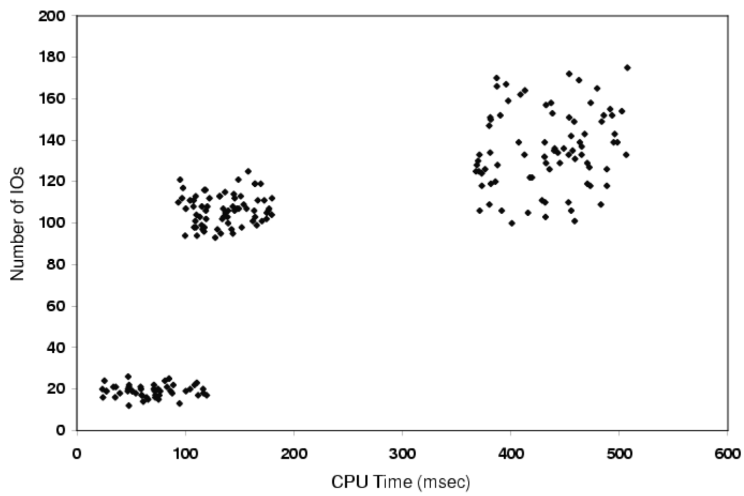
1.05
1.06
1.09
1.19
1.21
1.28
1.34
1.34
1.77
1.80
1.83
2.15
2.21
2.27
2.61
2.67
2.77
2.83
3.51
3.77
5.76
5.78
32.07
144.91

S	29.50
Average	9.51
COV	3.10

Box and Whisker Plots



Total No. of I/Os vs CPU Time



13

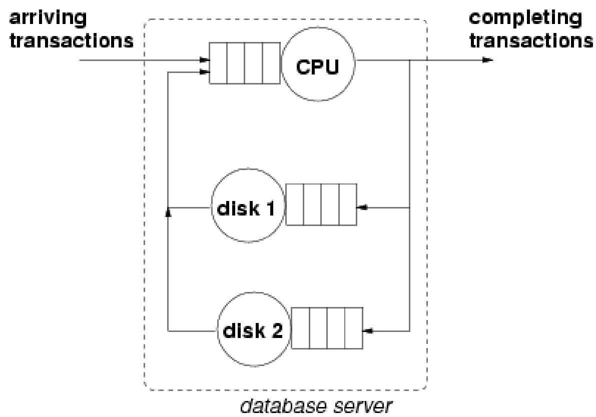
Result of Clustering Process

Table 5.3. Workload Clusters

Cluster Number	CPU Time (msec)	I/Os disk 1	I/Os disk 2	Npoints
1	67.5	8.0	11.0	50
2	434.2	62.4	73.1	80
3	136.1	69.8	36.7	70

14

QN for the DB Server



15

Building a Performance Model

- Need to apportion total resource utilizations to individual classes:

$$U_{i,r} = U_i \times f_{i,r}$$

The apportionment factor depends on the type of resource.

16

CPU Apportionment Factor

$$f_{CPU,r} = \frac{\text{Total CPU Time for class r}}{\text{Total CPU Time for all classes}}$$

$$f_{CPU,1} = \frac{67.5 \times 50}{47640.8} = 0.071$$

$$f_{CPU,2} = \frac{434.2 \times 80}{47640.8} = 0.729$$

$$f_{CPU,3} = \frac{136.1 \times 70}{47640.8} = 0.200$$

Disk 1 Apportionment Factor

$$f_{disk-i,r} = \frac{\text{Total no. I/Os on disk i by class r}}{\text{Total no. I/Os on disk i for all classes}}$$

$$f_{disk-1,1} = \frac{8 \times 50}{10275} = 0.039$$

$$f_{disk-1,2} = \frac{62.4 \times 80}{10275} = 0.486$$

$$f_{disk-1,3} = \frac{69.8 \times 70}{10275} = 0.475$$

Disk 2 Apportionment Factor

$$f_{disk-i,r} = \frac{\text{Total no. I/Os on disk i by class r}}{\text{Total no. I/Os on disk i for all classes}}$$

$$f_{disk-2,1} = \frac{11 \times 50}{8969} = 0.061$$

$$f_{disk-2,2} = \frac{73.1 \times 80}{8969} = 0.652$$

$$f_{disk-2,3} = \frac{36.7 \times 70}{8969} = 0.287$$

19

Model Parameters

Log total time 150 sec Total Throughput 1.33

Apportionment Factors	Class 1	Class 2	Class 3	Total
CPU	0.071	0.729	0.2	1.00
Disk 1	0.039	0.486	0.475	1.00
Disk 2	0.061	0.652	0.287	1.00

Utilization Values	Class 1	Class 2	Class 3	Total
CPU	0.032	0.328	0.090	0.45
Disk 1	0.029	0.365	0.356	0.75
Disk 2	0.040	0.424	0.187	0.65
Class throughput	0.33	0.53	0.47	

Service Demands	Class 1	Class 2	Class 3
CPU	0.096	0.615	0.193
Disk 1	0.088	0.683	0.763
Disk 2	0.119	0.795	0.400

20

5.3.2 Computing Service Demands

- The model requires the per device, per class service demands, $D_{i,r}$.
- The Service Demand Law (see Chapter 3) states that $D_i = U_i/X_0$.
- Applied to the current multi-class example, $D_{i,r} = U_{i,r}/X_{0,r}$.
- The utilizations, $U_{i,r}$, are available from Table 5.4.
- The throughput for each class, $X_{0,r}$, is obtained by dividing the number of transactions processed for each class by the duration of the measurement interval.
- The number of transactions processed per class is given in the last column of Table 5.3. Thus, recalling the measurement interval of 150 seconds,

$$X_{0,1} = 50/150 = 0.33 \ \ tps$$
$$X_{0,2} = 80/150 = 0.53 \ \ tps$$
$$X_{0,3} = 70/150 = 0.47 \ \ tps.$$

Equation 5.3.9

- Using the multi-class device utilizations from Table 5.4 and the multi-class throughputs from Eq. (5.3.9), the multi-class service demands at each device are easily computed from the Service Demand Law, $D_{i,r} = U_{i,r}/X_{0,r}$.
- The results are shown in Table 5.5.

Table 5.5. Service Demand Values for All Classes (in sec)

	Class1	Class2	Class3
CPU	0.096	0.615	0.193
Disk1	0.088	0.683	0.763
Disk2	0.119	0.795	0.400

5.4 Using the Model

- The queuing network model of Fig. 5.4 is now completely parameterized using the service demands of Table 5.5 and the class arrival rates from Eq. (5.3.9).
- Note: The transaction arrival rates must equal the transaction completion rates (i.e., $X_{0,1}$, $X_{0,2}$, and $X_{0,3}$) since this is a lossless system and the flow into the database server must equal the flow out of the database server.
- The parameterized model is solved using OpenQN-Chap5.xls.
- The residence times for all classes and queues as well as the response time per class are shown in Table 5.6.

Table 5.6. Residence Times and Response Times for All Classes (in sec)

	Class1	Class2	Class3
CPU	0.174	1.118	0.351
Disk 1	0.351	2.734	3.054
Disk 2	0.340	2.270	1.142
Response Time	0.865	6.122	4.546

- As indicated in Table 5.6, response times for classes 2 and 3 are relatively high. Moreover, disk 1 is the resource where all classes spend most of their time.
- This is not surprising since disk 1 has the highest overall utilization (see Table 5.2). Thus disk 1 is the system bottleneck.
- From looking at Table 5.6, transactions from classes 1, 2, and 3 spend over 40%, 44%, and 67% of their time using or waiting to use disk 1, respectively.
- Note also that the ratios of the residence times at disk 1 to the corresponding service demand are quite high.
- These ratios are 4.0 ($\frac{0.351}{0.088}, \frac{2.734}{0.683}, \frac{3.054}{0.763}$) for each class on disk 1.

- In other words, the time a transaction spends on disk 1 is four times its total service time on that disk. Stated equivalently, a transaction's total waiting time at disk 1 is three times its total service time.
- In comparison, the corresponding waiting times at the CPU and disk 2 are only 0.8 and 1.9 times the class service times, respectively. Therefore, to improve performance (i.e., to reduce response times), the most effective strategy is to reduce the time spent on I/O and particularly at disk 1.

What-If Scenarios

- This baseline model can be used to evaluate relevant "what-if" scenarios.
- It has already been noted that performance would improve by upgrading the disk subsystem.
- Also, as the predicted workload intensity changes over time, an understanding of the predicted resulting performance is also important.
- For example, suppose that the workload intensity is predicted to change over the next ten months as indicated in Table 5.7. (Note that January's workload (i.e., X_0 = 1.33 tps) is the baseline workload intensity assumed up to this point.)

Table 5.7. Workload Intensity Prediction

Month	Arrival Rate
January	1.33
February	1.45
March	1.68
April	2.26
May	2.68
June	3.25
July	3.98
August	4.78
September	5.74
October	6.76

5.4.1 Adding a Third Disk

- To cope with the increasing expected workload (i.e., Table 5.7) and to address the bottleneck, a new disk equivalent to the two existing disks is proposed to be added to the database server. The performance analyst also decides that the I/O activity should be balanced on the three disks to further improve I/O performance. Balancing the load is achieved by distributing the file requests across the three disks so that the service demand of any class is the same at all three disks. Thus, the new values of the service demands for disks 1 through 3 are computed as

$$\boxed{D_{disk\ i,r}^{new} = \frac{D_{disk\ 1,r}^{old} + D_{disk\ 2,r}^{old}}{3} \quad for\ i = 1,2,3.}$$

Equation 5.4.10

Table 5.8. Service Demand Values With a Third Disk Added (in sec)

	Class1	Class2	Class3
CPU	0.096	0.615	0.193
Disk 1	0.069	0.493	0.388
Disk 2	0.069	0.493	0.388
Disk 3	0.069	0.493	0.388

- The new response time values for the three classes are obtained with the help of OpenQN-Chap5.xls
- (Note: Remember to reinitialize the model to indicate that there are now four queues).
- The results are shown in the top part of Table 5.9 for the first five months (January-May) of the predicted new workload intensity levels.
- By looking at the first line in Table 5.9 (i.e., the January intensity of 1.33 tps) and comparing the class response times against the baseline metrics in Table 5.6, the effects of simply adding the third disk and balancing the I/O load across the disks results in over a 35% performance improvement (i.e., response time reduction). However, by May (i.e., workload intensity of 2.68 tps) the predicted performance is unacceptably poor, with class 2 anticipated response time exceeding 30 seconds.

Table 5.9. Response Times (in sec) for Various Scenarios

Arrival Rate (tps)	Response Time (sec)		
	Class 1	Class 2	Class 3
Scenario: third disk added.			
1.33	0.56	3.89	2.53
1.45	0.61	4.21	2.75
1.68	0.73	5.02	3.28
2.26	1.39	9.65	6.37
2.68	4.34	30.28	20.78
Scenario: 3 disks and a dual CPU.			
2.26	1.11	7.86	5.81
2.68	3.47	24.71	19.03
Scenario: 3 disks 3x faster and a dual CPU.			
2.68	0.24	1.59	0.84
3.25	0.27	1.78	0.93
3.98	0.32	2.16	1.11
4.78	0.45	3.01	1.47
5.74	1.78	11.60	4.34
Scenario: 3 disks 3x faster and a quad CPU.			
5.74	0.33	2.25	1.41
6.76	0.45	3.15	2.09

- By looking at the ResidenceTimes worksheet of OpenQN-Chap5.xls, it is seen that with the third disk, the CPU is the resource where class 1 and class 2 transactions spend most of their time.
- That is, for these classes, the disks are no longer the bottleneck, but rather it is the CPU that is the most limiting the performance.
- To maintain an acceptable response time from, say April (i.e., 2.26 tps) on, it is necessary to reduce contention on the CPU. One alternative is to replace the current CPU with a faster processor. Another alternative is to upgrade the system to a multiprocessor by adding a second CPU.
- This second scenario is considered in the next subsection.

5.4.2 Using a Dual CPU System

- In order to maintain an acceptable QoS level past April, in addition to the extra disk and a balanced I/O load, an additional CPU is proposed in a dual CPU configuration.

  In order to analyze the effects of this change using OpenQN-Chap5.xls, the CPU queue is specified as MP2 (i.e., a multiprocessor with two CPUs). The results are shown in the middle of Table 5.9 for the April and May workloads (i.e., 2.26 tps and 2.68 tps).
- The largest reduction in response time, as expected, is for classes 1 and 2, the ones that spend more time at the CPU. However, the improvements are relatively minor and the response times are still very high for classes 2 and 3 for an arrival rate of 2.68 tps (i.e., May's workload).
- An analysis of the residence time breakdown indicates that with the dual CPU, all three classes spend most of their time at the disks. That is, adding the second CPU shifted the system bottleneck back to the disks. The next step is to improve disk access performance further.

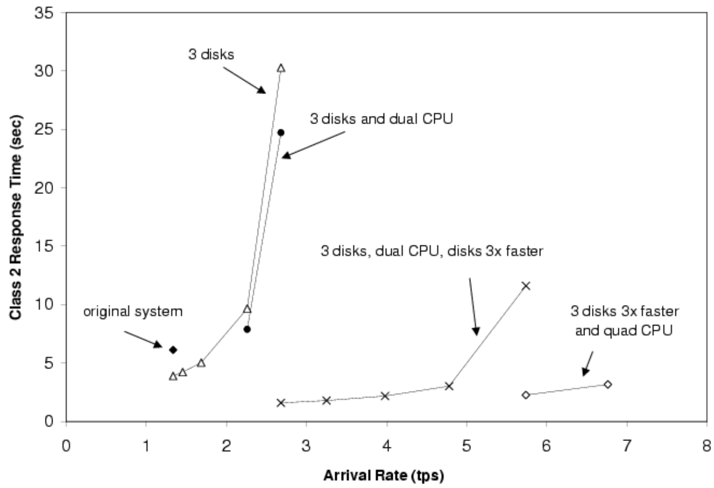
5.4.3 Using Faster Disks

- A more dramatic upgrade is considered here to hopefully cope with the increasing workload intensities. Each disk is replaced by one that is three times faster. This is reflected in the model parameters by simply dividing the service demands on all disks by a factor of three. Solving the model with these new disk speeds yields the results shown in the middle lower half of Table 5.9.
- The results indicate that acceptable response times are obtained for an arrival rate up to 4.78 tps (i.e., through August). However, for 5.74 tps (i.e., September's workload), the response time for class 2 exceeds 10 seconds. A look at the residence times for class 2 reveals that 87% of its response time is being spent at the CPU. This indicates the need to further enhance the CPU.

5.4.4 Moving to a 4-CPU System

- In order to reduce the response time of class 2 transactions at high arrival rates, the dual CPU is replaced by a quad CPU. This change is reflected in the model by specifying the type of the CPU as MP4.
- The results are shown in the lower portion of Table 5.9. With this final upgrade, the model indicates that the response times are at acceptable service levels for all classes throughout the ten months period of concern.
- Figure 5.5 illustrates how the response time varies for class 2 transactions (i.e., those transactions that have the highest CPU and I/O demands resulting in the highest response times) for each of the scenarios.
- If a 10-second service level is deemed "acceptable" then an appropriate capacity planning strategy becomes apparent: 1) in January purchase an additional disk and load balance the load across the three disks by moving files, 2) in May exchange the disks for ones that are three times faster and purchase an additional CPU, 3) in September, purchase two additional CPUs. Though the parameters and time frames may change, this case study illustrates the usefulness of a quantitative analysis using queuing network modeling approach.

Class 2 Response Time for Various Scenarios



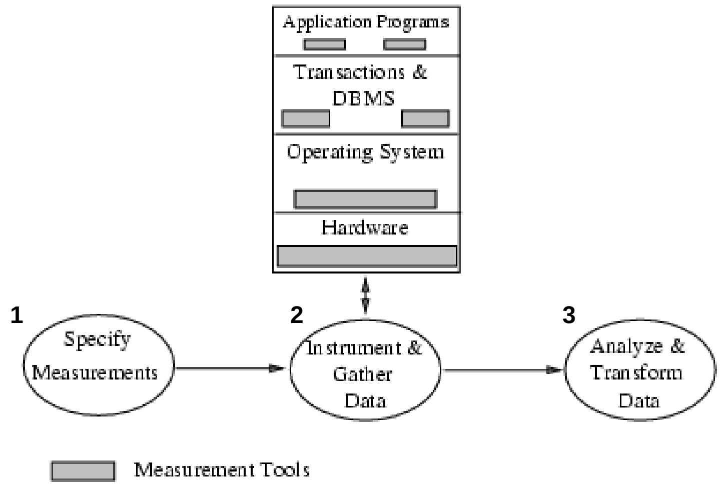
23

Monitoring Tools

- Hardware monitors
- Software monitors
  - Accounting systems
  - Program analyzers
- Hybrid Monitors
- Event-trace monitoring
- Sample monitoring

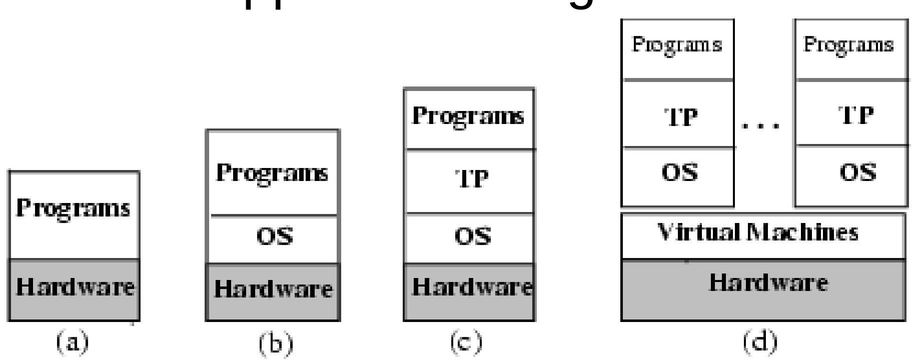
24

The Measurement Process



25

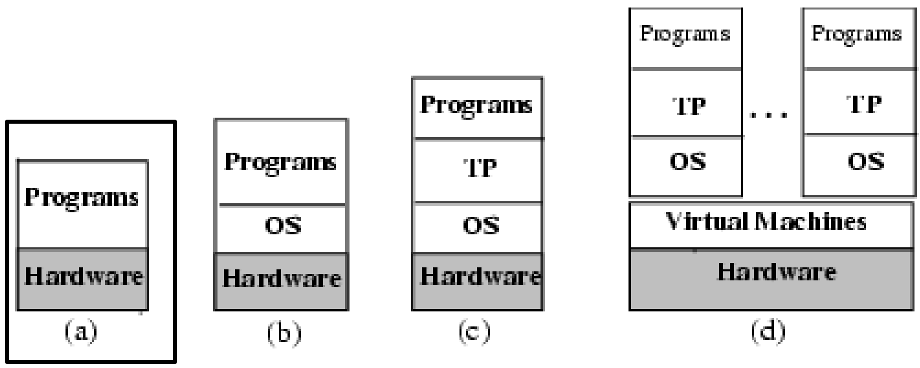
Execution Environments for Application Programs



The measurement of CPU utilization depends on the various software layers between the program and the bare machine.

26

Execution Environments for Application Programs

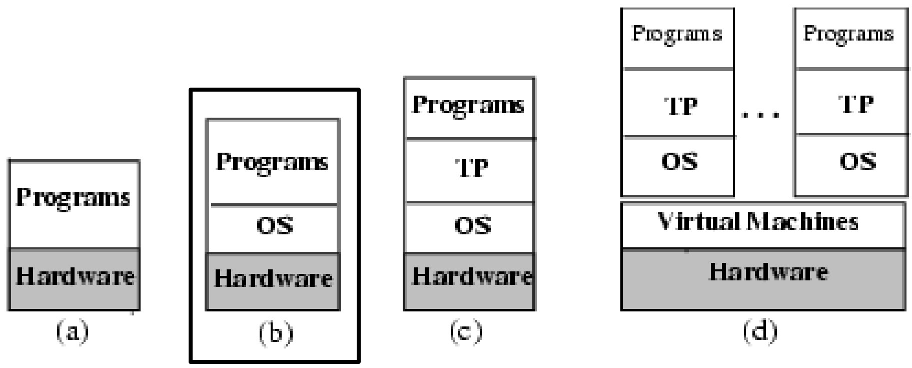


Bare Machine Example

- Consider an early computer with no OS that executes one program at a time. During 1,800 sec, a hardware monitor measures a utilization of 40% for the CPU and 100 batch jobs are recorded. The average CPU demand for each job is:

  0.4 x 1800 / 100 = 7.2 seconds

Execution Environments for Application Programs



29

OS Example

- Consider a computer system running batch programs and interactive commands. The system is monitored for 1,800 sec and a software monitor measures the CPU utilization as 60%. The accounting log of the OS records CPU times for batch and for the 1,200 executed interactive commands separately. From this data, the class utilizations are batch = 40% and interactive = 12%.

30

OS Example (cont'd)

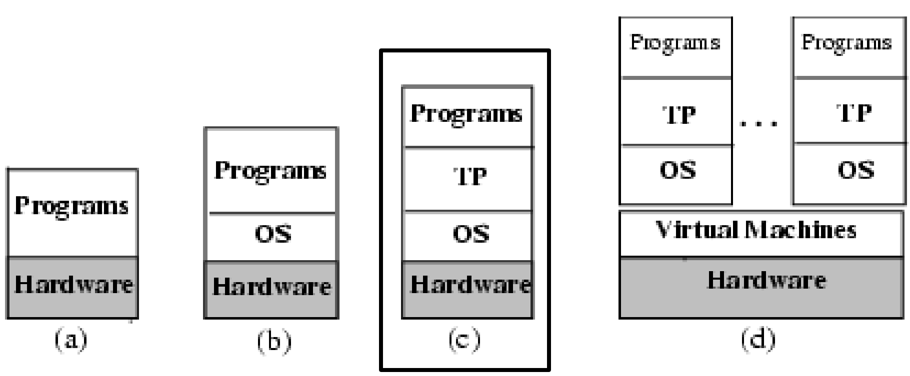
- The CPU demand for the interactive class is given by

$$D_{cpu,\text{interactive}} = \frac{U^t_{cpu} \times f_{cpu,\text{interactive}}}{X_{0,\text{interactive}}}$$

$$= \frac{0.6 \times [0.12/(0.12 + 0.40)]}{1200/1800} = 0.208\text{sec}$$

31

Execution Environments for Application Programs

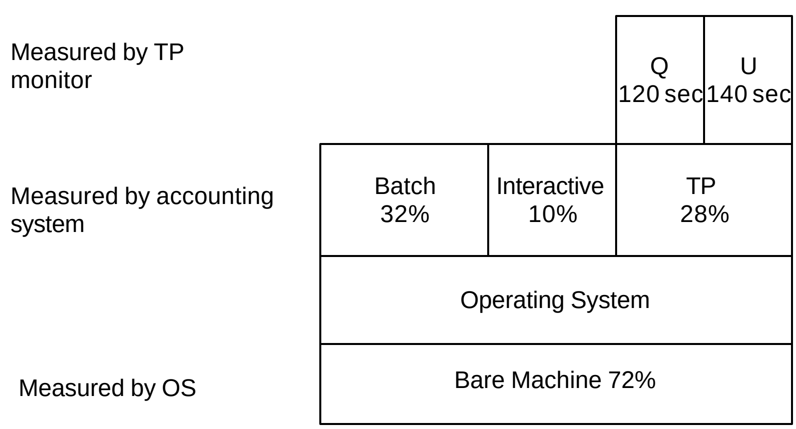


32

TP Example

- A mainframe processes 3 workload classes: batch (B), interactive (I), and transactions (T). Classes B and I run on top of the OS and class T runs on top of the TP monitor. There are two types of transactions: query (Q) and update (U). What is the service demand of update transactions.

TP Monitor Example



TP Example (cont'd)

- Measurements from the OS monitor during 1800 sec: CPU utilization is 72%.
- Measurements from the accounting facility:

$$U^{os}_{cpu,B} = 32\%$$

$$U^{os}_{cpu,I} = 10\%$$

$$U^{os}_{cpu,T} = 28\%$$

35

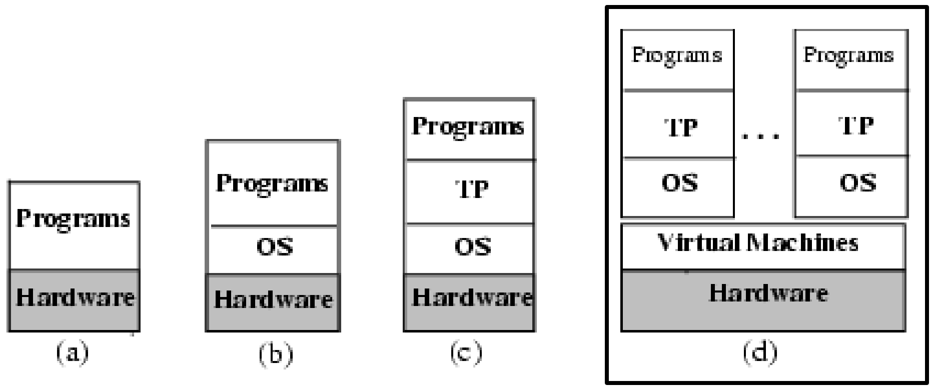
TP Example (cont'd)

- Measurements from the program analyzer of the TP monitor:
  - 1200 query transactions, which consumed 120 sec of CPU.
  - 400 query transactions, which consumed 140 sec of CPU.

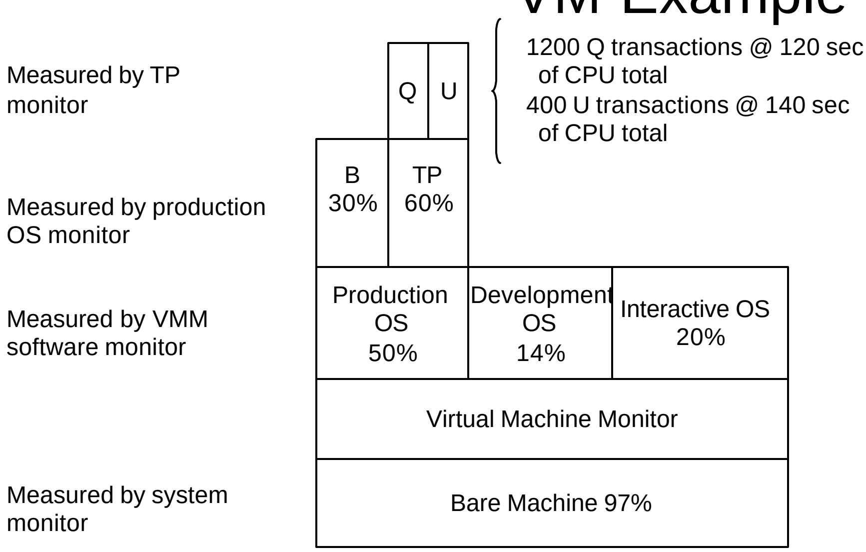
$$D_{cpu,U} = \frac{0.72 \times \frac{0.28}{0.32+0.10+0.28} \times \frac{140}{120+140}}{400/1800} = 0.698\text{sec}$$

36

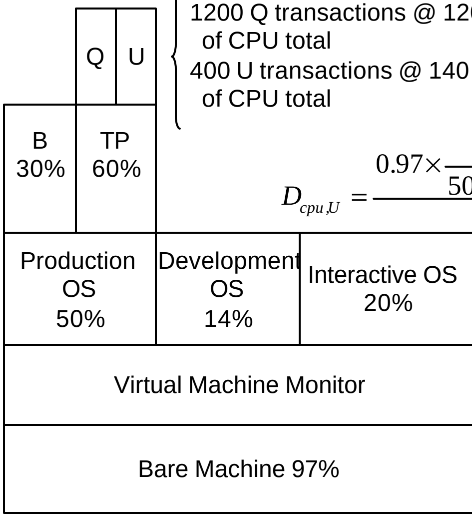
Execution Environments for Application Programs



VM Example



VM Example (cont'd)

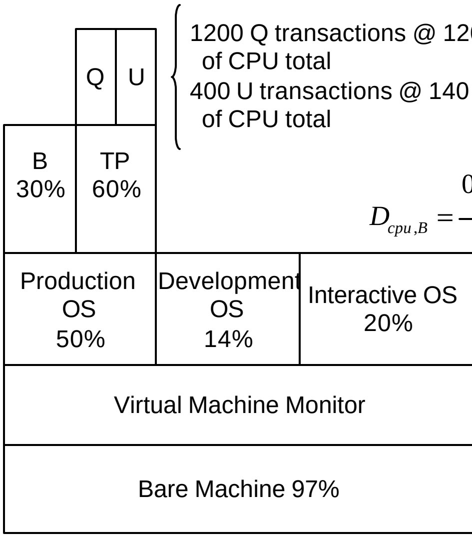


1200 Q transactions @ 120 sec of CPU total
400 U transactions @ 140 sec of CPU total

$$D_{cpu,U} = \frac{0.97 \times \frac{50}{50+14+20} \times \frac{60}{30+60} \times \frac{140}{120+140}}{400/1800} = 0.933\,\mathrm{sec}$$

39

VM Example (cont'd)



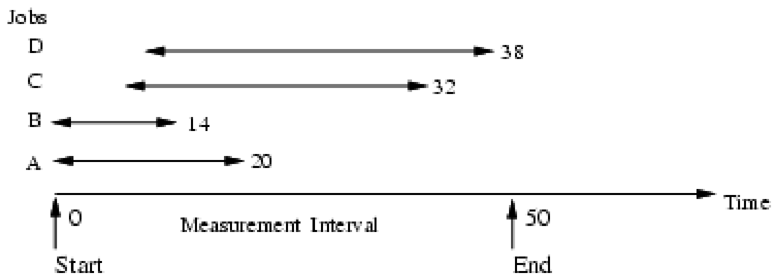
1200 Q transactions @ 120 sec of CPU total
400 U transactions @ 140 sec of CPU total

$$D_{cpu,B} = \frac{0.97 \times \frac{50}{50+14+20} \times \frac{30}{30+60}}{80/1800} = 4.33\,\mathrm{sec}$$

assuming that 80 batch jobs complete during the period.

40

Computing the average concurrency level



$$\overline{N} = \frac{20 + 14 + 32 + 38}{50} = 2.08$$