



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Επίδοση Υπολογιστικών Συστημάτων

1η Εργασία

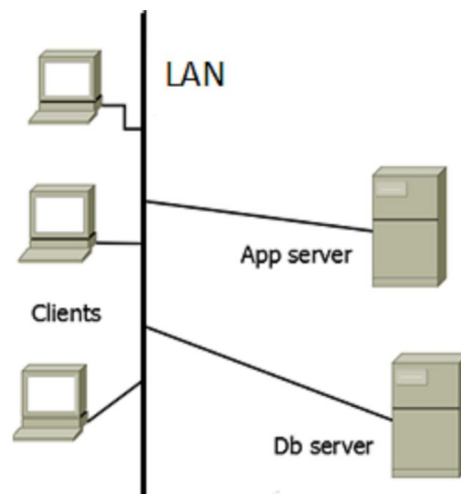
Αχλάτης Στέφανος-Σταμάτης (03116149)

<el16149@central.ntua.gr>

Μάιος 2020

1.Εισαγωγή

Στο παρόν Project μελετάται ένα σύστημα εξυπηρετητή-πελατών ως ανοιχτό δίκτυο. Πιο συγκεκριμένα μελετάμε το ιδιωτικό δίκτυο LAN ενός ειδησεογραφικού οργανισμού. Στην μελέτη μας θεωρούμε ότι ο οργανισμός έχει 480 χρήστες αυτού του δικτύου, και οι δυο λειτουργίες που μπορούν να εκτελέσουν είναι εγγραφές και ανανεώσεις. Το σύστημα ακολουθεί αρχιτεκτονική δύο βαθμίδων η οποία περιλαμβάνει εξυπηρετητή εφαρμογών (App Server) και εξυπηρετητή βάσης δεδομένων (Db Server). Οι εξυπηρετητές περιλαμβάνουν CPU και έναν δίσκο έκαστος.



Και μας έχουν δώσει τον εξής αρχικό πίνακα παραμέτρων του συστήματος:

α/α	Σταθμός	Τύπος	D_{ij} (msec)	
			Εγγραφές	Ανανεώσεις
1	Term	D	25000	19000
2	LAN	D	47	24
3	App Server CPU	LI	43	19
4	App Server Disk	LI	44	29
5	Db Server CPU	LI	45	59
6	Db Server Disk	LI	41	37

Αξίζει να σημειωθεί ότι ο σταθμός term δηλώνει τον χρόνο που δαπανά ο χρήστης του συστήματος για να σκεφτεί και να αρχικοποιήσει τις απαιτούμενες πληροφορίες στο σύστημα έτσι ώστε το σύστημα να εκτελέσει την απαιτούμενη ενέργεια. Τονίζεται ότι αυτός ο χρόνος δεν περιλαμβάνει τον χρόνο απόκρισης του συστήματος, δηλαδή πόσο χρόνο κάνει το σύστημα να αναζητήσει, τροποποιήσει κλπ και να δώσει στον χρήστη αυτές τις πληροφορίες.

Γι αυτόν τον λόγο αυτός ο χρόνος είναι της τάξεως των αρκετών δευτερολέπτων σε αντίθεση με τους υπόλοιπους χρόνους που είναι της τάξεως των msec.

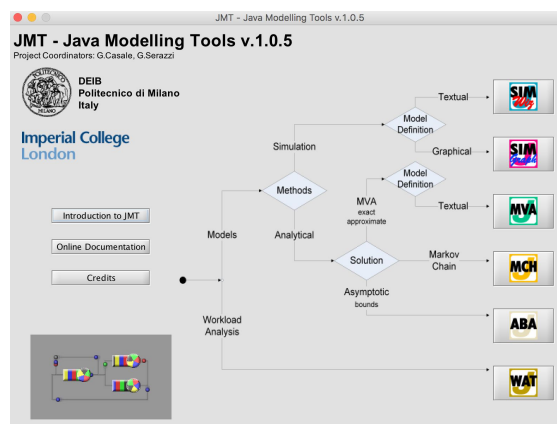
Έτσι, στην αρχή ψάχνουμε να βρούμε το μείγμα πληθυσμού που μεγιστοποιεί το throughput του συστήματος ως σύνολο μέσω του εργαλείου what-if του JMVA, και για το δεδομένο μείγμα μελετάμε ορισμένες μετρικές και καταλήγουμε στο συμπέρασμα ότι οι αρχικές προδιαγραφές της cpu του db server δημιουργεί στένωση και καθυστέρηση στο σύστημα. Γι

αυτό στην συνέχεια αντικαθιστούσε την cpu του db server με μια load dependent cpu και βλέπουμε πως επηρεάζει το σύστημα.

Στην συνέχεια, με στόχο την απλούστευση της προσομοίωσης, ομαδοποιούμε το προηγούμενο σύστημα σε μοντέλο μιας κατηγορίας όπου η τιμή της απαίτησης εξυπηρέτησης σε κάθε σταθμό του μοντέλου μιας κατηγορίας θα τεθεί ίση με τον μέσο όρο των επιμέρους τιμών σταθμισμένων με τον ρυθμό απόδοσης της αντίστοιχης κατηγορίας. Τέλος, γι αυτό το νέο απλοποιημένο σύστημα παρατηρούμε ότι η cpu του db server δημιουργεί στένωση και γι αυτό την αντικαθιστούμε με μια load dependent cpu και βλέπουμε πως αυτό επηρεάζει το σύστημα.

2. Το εργαλείο JMVA

Οι προσομοιώσεις έγιναν στο εργαλείο JMVA του πακέτου JMT.



Το εργαλείο αυτό εφαρμόζει τον Mean Value Algorithm σε ένα σύστημα το οποίο ορίζει ο χρήστης.

Αρχικά ο χρήστης μπορεί να ορίσει το πληθυσμό της προσομοίωσης, χωρίζοντας το σε κλάσεις αν χρειάζεται ορίζοντας την κάθε μια ως κλειστή ή ανοικτή.

Για το συγκεκριμένο πρόβλημα οι κλάσεις μας ήταν κλειστές αφού δεν επιτρέπεται να αφορμή στο σύστημα πέρα από του 480 χρήστες του.

*	Name	Type	No. of Customers	Arrival Rate (λ)
1	Class1	closed	42	

Μετά στην συνέχεια ο χρήστης μπορεί να ορίσει τους σταθμούς του συστήματος, και να τους ονοματοδοτήσει έτσι ώστε να μπορεί να τους ξεχωρίσει ευκολότερα.

Επίσης μπορεί να τους χαρακτηρίσει ως load dependent, load independent και ως delay station.

Stations characteristics Number: 3

Number, customized name and type of stations. Add stations one by one or define the total number at once. Load Dependent stations necessarily require the use of MVA.

New Station

*	Name	Type
1	Station1	Load Independent
2	Station2	Delay (Infinite Server)
3	Station3	Load Dependent

Στη συνέχεια μπορεί να ορίσει τους χρόνους κάθε σταθμού και για την περίπτωση των load dependent station να δώσει καθυστέρηση ανάλογα με το πλήθος χρηστών στο συγκεκριμένο station.

Classes Stations **Service Times** Visits Reference Station What-if Comment

Service Times
Input service times of each station for each class.
If the station is "Load Dependent" you can set the service times for each number of customers by double-click on "LD Settings..." button.
Press "Service Demands" button to enter service demands instead of service times and visits.
MULTICLASS MODELS: when for a station the per-class service times are different, the results are

*	Class1
Station1	0.0000
Station2	0.0000
Station3	LD Settings...

Μετά μπορεί να ορίσει σταθμό αναφοράς ,με βάση τον οποίον το jnna υπολογίζει την ρυθμαπόδοση και το system response time για κάθε κλειστή κλάση. Εμείς στην εργασία ορίσαμε ως σταθμό αναφοράς τον term.

Reference Station
The station is used to compute the system throughput and the system response time for each closed class.
Performance metrics of open classes are always computed with respect to the arrival process. Visits at the Reference station can not be Zero.
WARNING: the reference station for all closed classes is forced to be the same station.

Class	Station
Class1	Station1

Τέλος ο χρήστης μπορεί να τρέξει what-if analysis ως προς πολλές μετρικές. Εμείς τρέξαμε ως προς το πλήθος των χρηστών έτσι ώστε να βρούμε το μείγμα πληθυσμού που βελτιστοποιεί την ρυθμαπόδοση του συστήματος.

Classes Stations Service Times Visits Reference Station **What-if** Comment

What-if analysis
Select a control parameter if you want to solve several models with its values changing in the selected range. The performance indices will be shown in a graph.

Control Parameter : -- Disabled --

- Disabled --
- Number of Customers
- Arrival Rates (only open classes)
- Population Mix (only 2 closed classes)
- Service Demands

Τέλος ο χρήστης μπορεί να επιλέξει να τρέξει διάφορες παραλλαγές του mna αλλά εμείς επιλέξαμε να τρέξουμε τον default mna όπως αυτός διδάχτηκε στο μάθημα.

Κλείνοντας, αξίζει να τονιστεί ότι αντίστοιχα την προσομοίωση θα μπορούσαμε να την τρέχαμε γράφοντας κώδικα σε γλώσσες όπως η python3, η C++ ή ακόμη και η Java όπου

θα κάναμε πολύ συγκεκριμένη κάθε τεχνική λεπτομέρεια του συστήματος χωρίς να αναγκάζομαστε να προσαρμόσουμε το σύστημα στις ανάγκες του προσομοιωτή. Αλλά όπως μάθαμε και στο μάθημα αποτελεί καλή τεχνική πρώτα να τρέχουμε τις προσομοιώσεις μας σε ένα generic περιβάλλον όπως ο JMVA και μετά να χτίζουμε εμείς οι ίδιοι δικές μας προσομοιώσεις για να μελετήσουμε με **ακόμη μεγαλύτερη ακρίβεια** το σύστημα μας.

Έτσι, σε αυτή την εργασία θα τρέξουμε τις προσομοιώσεις μόνο μέσα από το JMVA.

3.Εκτέλεση Άσκησης

Ερώτημα Α

Με χρήση της επιλογής What-if του εργαλείου να προσδιοριστεί το μείγμα πληθυσμού των δύο κατηγοριών, για το οποίο επιτυγχάνεται μεγιστοποίηση του ρυθμού απόδοσης του συστήματος. Το μείγμα αυτό θα ισχύει στα επόμενα ερωτήματα.

Ζητούνται ο ρυθμός απόδοσης και ο χρόνος απόκρισης του συστήματος συνολικά και για καθεμία από τις κατηγορίες πελατών, καθώς και ο βαθμός χρησιμοποίησης των σταθμών του συστήματος. Ποιος σταθμός είναι η στένωση του συστήματος;

Αρχικά, δηλώνουμε τις 2 κατηγορίες χρηστών με τους αντίστοιχους πληθυσμούς.

Η κατηγορία 1 ανήκει στις εγγραφές ενώ η κατηγορία 2 ανήκει στις ανανεώσεις.

Δεδομένου ότι ο στόχος της προσομοίωσης είναι να μελετήσουμε το population mix βάζουμε δύο αριθμούς στον πληθυσμό της κάθε κατηγορίας, τέτοιοι ώστε **το άθροισμα** τους να είναι ίσο με το μέγιστο δυνατό πληθυσμό που ορίζεται στο σύστημα ως σύνολό, δηλαδή 480.

Επιλέγουμε να αρχικοποιήσουμε τον πληθυσμό σε 240 σε κάθε κατηγορία.

Classes characteristics Number: 2

Number, customized name, type of classes and number of customers (closed class) or arrival rate (open class). Add classes one by one or define total number at once.

New Class

*	Name	Type	No. of Customers	Arrival Rate (λ)	
1	Class1	closed	240		✗
2	Class2	closed	240		✗

Μετά σειρά έχει η δήλωση των σταθμών του συστήματος και ο χαρακτηρισμός τους ως Load Independent και Delay για τους σταθμούς αναμονής και καθυστέρησης αντίστοιχα.

Stations characteristics Number: 6

Number, customized name and type of stations. Add stations one by one or define the total number at once. Load Dependent stations necessarily require the use of MVA.

New Station

*	Name	Type	
1	LAN	Delay (Infinite Server)	✗
2	APP Server CPU	Load Independent	✗
3	APP Server Disk	Load Independent	✗
4	DB Server CPU	Load Independent	✗
5	DB Server Disk	Load Independent	✗
6	Users	Delay (Infinite Server)	✗

Στη συνέχεια ορίσαμε τις απαιτήσεις σε μέσο χρόνο χρήστη,δηλαδή σε μέσο χρόνο σκέψης, και σε μέση συνολική απαίτηση εξυπηρέτησης για κάθε σταθμό. Οι τιμές που φαίνονται στον πίνακα είναι της τάξεως των sec.

Service Times Input service times of each station for each class. If the station is "Load Dependent" you can set the service times for each number of customers by double-click on "LD Settings..." button. Press "Service Demands" button to enter service demands instead of service times and visits. MULTICLASS MODELS: when for a station the per-class service times are different, the results are correct ONLY IF its scheduling discipline is assumed Processor Sharing (PS) and not FCFS (See BCMP Theorem).	*	Class1	Class2
	LAN	0.0470	0.0240
	APP Server...	0.0430	0.0190
	APP Server...	0.0440	0.0290
	DB Server ...	0.0450	0.0590
	DB Server ...	0.0410	0.0370
	Users	25.0000	19.0000

Στην συνέχεια ορίσαμε Preference Station για τις 2 κατηγορίες.

Class	Station
Class1	Users
Class2	Users

Reference Station
The station is used to compute the system throughput and the system response time for each closed class.
Performance metrics of open classes are always computed with respect to the arrival process. Visits at the Reference station can not be Zero.
WARNING: the reference station for all closed classes is forced to be the same station.

Και στο τέλος ορίσαμε στην what-if analysis ότι θέλουμε να μελετήσουμε με παράμετρο το **Population Mix**.

Ένα ενδιαφέρον ερώτημα είναι το πόσα βήματα θα πρέπει να χρησιμοποιήσουμε για να έχουμε ακριβές αποτέλεσμα στη μελέτη μας.

Είναι προφανές ότι όσο μικρότερο είναι το n τόσο πιο γρήγορη θα είναι και η προσομοίωση αλλά όσο πιο μεγάλο είναι το n τόσο πιο ακριβής είναι η προσομοίωση, αυτό είναι ένα trade off που πρέπει να αντιμετωπίσουμε.

Το ερώτημα αυτό μπορεί να απαντηθεί μόνο αν ορίσουμε το πόσο ακρίβεια θέλουμε να έχουμε στην προσομοίωση μας. Σε αυτή την άσκηση θα μελετήσουμε το πρόβλημα με τη μέγιστη δυνατή ακρίβεια δηλαδή για $n=479$.

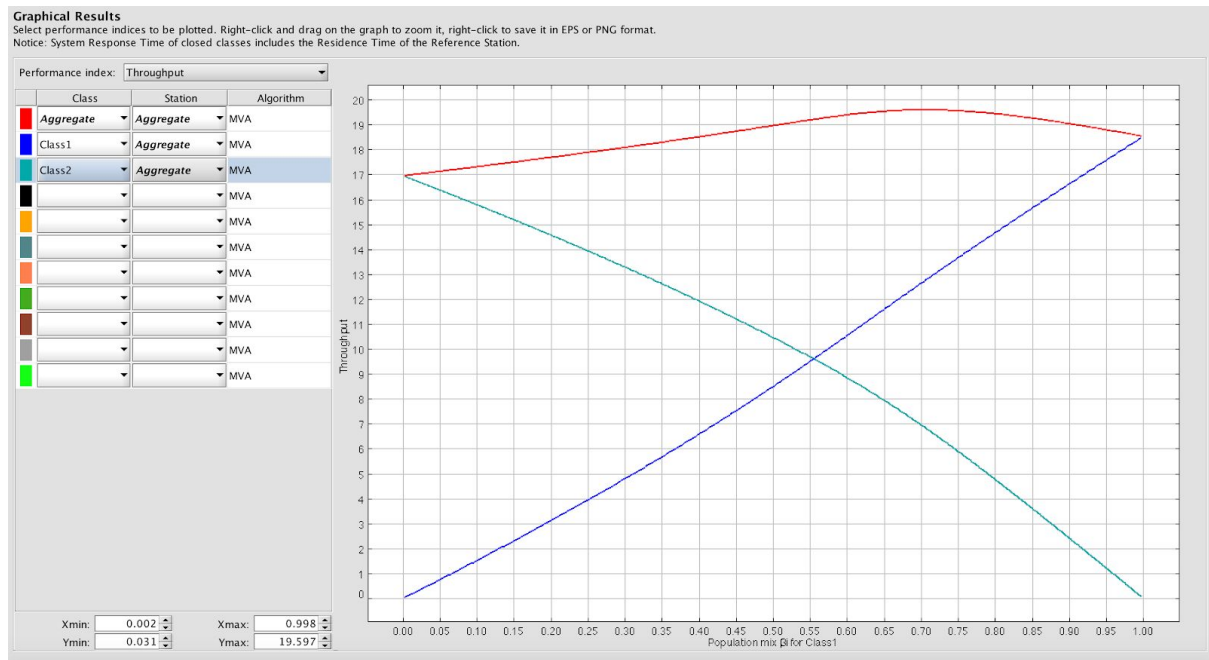
Ωστόσο, τρέξαμε την προσομοίωση και για την default τιμή των $n=11$ βημάτων και είδαμε ότι το αποτέλεσμα είναι σχεδόν ίδιο - διαφοροποίηση κατά 2 μόλις άτομα στο βέλτιστο μείγμα πληθυσμού όσον αφορά την συνολική ρυθμαπρόδοση του συστήματος.

What-if analysis	Control Parameter : Population Mix
Solve models with different proportion of jobs between two closed classes, keeping constant the total number of jobs N ($\beta_i = N_i / N$). It is required that $N_i > 0$. Since only integer N_i values are allowed, the number of models executed can be very small.	Class : Class1
	From (β_i) : 0.0021
	To (β_i) : 0.9979
	Steps (n. of executions) : 479

Τέλος, αφήσαμε τον default algorithm ανάλυσης, τον MVA.

Μετά τους υπολογισμούς λοιπόν προέκυψαν τα ακόλουθα αποτελέσματα:

Αρχικά όσον αφορά το μείγμα πληθυσμού σε σχέση με το throughput βλέπουμε την εξής γραφική:



Η κλάση 1 είναι οι χρήστες εγγραφής και χρωματίζεται με μπλε χρώμα.

Η κλάση 2 είναι οι χρήστες ανανεώσεις και χρωματίζεται με πράσινο χρώμα.

Ενώ με κόκκινο χρώμα χρωματίζεται η συνολική ρυθμαπόδοση του συστήματος.

Βλέπουμε λοιπόν ότι η συνολική ρυθμαπόδοση για την κλάση 1 βελτιώνεται όσο μεγαλώνει το πλήθος του πληθυσμού της και μειώνεται το πλήθος του πληθυσμού της κλάσης 2.

Ενώ, η συνολική ρυθμαπόδοση για την κλάση 2 χειροτερεύει όσο μικραίνει ο πληθυσμός της και μεγαλώνει ο πληθυσμός της κλάσης 2.

Η συνολική ρυθμαπόδοση αρχικά βελτιώνεται όσο αυξάνεται ο πληθυσμός της κλάσης 1 και μειώνεται ο πληθυσμός της κλάσης 2 αλλά μετά από ένα σημείο, ένα ακρότατο, χειροτερεύει όσο αυξάνεται ο πληθυσμός της κλάσης 1 και μειώνεται ο πληθυσμός της κλάσης 2.

Εμείς καλούμαστε να βρούμε αυτό το τοπικό ακρότατο, η γραφική χάνει ακρίβεια αλλά βλέπουμε ότι το ακρότατο είναι κοντά στο 0.7 δηλαδή για πληθυσμό κλάσης 1 ίσο με $0.7 \cdot 480 = 336$ και για πληθυσμό της κλάσης 2 ίσο με 144.

Ωστόσο επειδή θέλουμε ακρίβεια, γι' αυτό και βάλαμε το μέγιστο δυνατό n θα ελέγξουμε αυτή την τιμή από το textual result.

Βλέπουμε ότι η βέλτιστη ρυθμαπόδοση επιτυγχάνεται με δύο μείγματα πληθυσμού, δηλαδή η βέλτιστη συνολική ρυθμαπόδοση είναι ίση με 19.5972 και επιτυγχάνεται με

- $\beta_1 = 0.70417$ δηλαδή για το μείγμα με πληθυσμός κλάσης 1 ίσο με 338 και με πληθυσμός κλάσης 2 ίσο με 142.
- και $\beta_2 = 0.70625$ δηλαδή για το μείγμα με πληθυσμός κλάσης 1 ίσο με 339 και με πληθυσμός κλάσης 2 ίσο με 141.

Το πρώτο συμπέρασμα εξάγεται από τα εξείς αποτελέσματα:

Graphical Results Textual Results Synopsis			
Execution number: 338			
Population mix $\beta_i = 0.70417$ for Class1			
MVA			
Throughput Number of Customers Residence Times System Response Time Utilization System Power			
Throughput Throughput of each class for each station. System Throughput is the completion rate of the Reference Station.			
*	Aggregate	Class1	Class2
System	19.5972	12.7295	6.8678
LAN	19.5972	12.7295	6.8678
APP Serve...	19.5972	12.7295	6.8678
APP Serve...	19.5972	12.7295	6.8678
DB Server...	19.5972	12.7295	6.8678
DB Server...	19.5972	12.7295	6.8678
Users	19.5972	12.7295	6.8678

Graphical Results Textual Results Synopsis			
Execution number: B38			
Population mix $\beta_i = 0.70417$ for Class1			
MVA			
Throughput Number of Customers Residence Times System Response Time Utilization System Power			
Number of Customers Average number of customers for each class at each station.			
*	Aggregate	Class1	Class2
Aggregate	480.0000	338.0000	142.0000
LAN	0.7631	0.5983	0.1648
APP Serve...	2.0864	1.6837	0.4026
APP Serve...	3.1068	2.2904	0.8164
DB Server...	21.9130	12.9000	9.0129
DB Server...	3.4066	2.2907	1.1159
Users	448.7241	318.2369	130.4873

Ενώ το δεύτερο συμπέρασμα εξάγεται από τα εξείς αποτελέσματα:

Graphical Results Textual Results Synopsis			
Execution number: 339			
Population mix $\beta_i = 0.70625$ for Class1			
MVA			
Throughput Number of Customers Residence Times System Response Time Utilization System Power			
Throughput Throughput of each class for each station. System Throughput is the completion rate of the Reference Station.			
*	Aggregate	Class1	Class2
System	19.5972	12.7727	6.8245
LAN	19.5972	12.7727	6.8245
APP Serve...	19.5972	12.7727	6.8245
APP Serve...	19.5972	12.7727	6.8245
DB Server...	19.5972	12.7727	6.8245
DB Server...	19.5972	12.7727	6.8245
Users	19.5972	12.7727	6.8245

Graphical Results Textual Results Synopsis			
Execution number: 339			
Population mix $\beta_i = 0.70625$ for Class1			
MVA			
Throughput Number of Customers Residence Times System Response Time Utilization System Power			
Number of Customers Average number of customers for each class at each station.			
*	Aggregate	Class1	Class2
Aggregate	480.0000	339.0000	141.0000
LAN	0.7641	0.6003	0.1638
APP Serve...	2.0960	1.6947	0.4013
APP Serve...	3.1171	2.3038	0.8133
DB Server...	21.6310	12.7848	8.8462
DB Server...	3.4093	2.2998	1.1095
Users	448.9826	319.3167	129.6659

Έτσι, επιλέγουμε τυχαία έναν από τους δύο ως βέλτιστο συνδυασμό.

Άρα, η μέγιστη συνολική ρυθμαπόδοση επιτυγχάνεται για: 338 χρήστες εγγραφής (κλάσης 1) και 142 χρήστες ανάγνωσης (κλάσης 2).

Χρησιμοποιείτε αυτό το μείγμα για τα επόμενα ερωτήματα.

Ο ρυθμός απόδοσης του συστήματος συνολικά και για καθεμία από τις κατηγορίες πελατών είναι:

Throughput				
Throughput of each class for each station. System Throughput				
*	Aggregate	Class1	Class2	
System	19.5972	12.7295	6.8678	

Ο χρόνος απόκρισης του συστήματος συνολικά και για καθεμία από τις κατηγορίες πελατών είναι:

System Response Time				
The global aggregate is the "System Response Time" and is obtained weighting the aggregated values by the relative per-class throughput.				
A: This value of System Response Time includes the Residence Time of the Reference Station.				
B: This value of System Response Time does NOT include the Residence Time of the Reference Station.				
Notice: For open classes the Reference Station always coincides with the arrival process. Thus the B values are not computed.				
*	Aggregate	Class1	Class2	
A	24.4933	26.5525	20.6763	
B	1.5959	1.5525	1.6763	

Εμείς επικεντρώνουν την προσοχή μας στην περίπτωση A, όπου συνυπολογίζεται και ο χρόνος σκέψης και δράσης των χρηστών.

Υπενθυμίζουμε ότι ο χρόνος σκέψης και δράσης της κλάσης 1 είναι 25sec ενώ της κλάσης 2 19sec.

Άρα, συνολικά η ένας χρήστης της κλάσης 1 πρέπει να περιμένει 1.5525 sec για να εκτελεσει την εργασία του το συστημα, ενώ ένας χρήστης της κλάσης 1 πρέπει να περιμένει 1.6763 sec για να εκτελέσει την εργασία του το συστημα.

Χρόνος αρκετά μεγάλος επομένως θα πρέπει να προσπαθήσουμε να βελτιώσουμε το σύστημα μας.

Ο βαθμός χρησιμοποίησης των σταθμών του συστήματος είναι ο εξής:

Utilization				
Utilization of a customer class at the selected station. The utilization of a delay station is the average number of customers in the station (it may be greater than 1).				
*	Aggregate	Class1	Class2	
LAN	0.7631	0.5983	0.1648	
APP Serve...	0.6779	0.5474	0.1305	
APP Serve...	0.7593	0.5601	0.1992	
DB Server...	0.9780	0.5728	0.4052	
DB Server...	0.7760	0.5219	0.2541	
Users	448.7241	318.2369	130.4873	

όπου παρουσιάζονται με την σειρά η χρησιμοποίηση των σταθμών(τόσο συνολικά όσο και ανα κλάση) : LAN, APP Server CPU,APP Server Disk, DB Server CPU, DB Server Disk και Users.

Προσοχή στην ανάγνωση του πίνακα όπου η συνολική χρησιμοποίηση του σταθμού lan πρέπει να μεταφραστεί ως 76.31%. Το ίδιο ισχύει και για τους υπόλοιπους σταθμούς.

Γενικός Σχολιασμός Αποτελεσμάτων Α Ερωτήματος

Θα μπορούσε με μια πρώτη ανάγνωση να έκανε εντύπωση το αποτέλεσμα της μίξης πληθυσμών που βελτιστοποιεί την ρυθμαπόδοση δεδομένου ότι σχεδόν σε όλους του σταθμούς του δικτύου (εκτός από την cpu του db server) έχουμε ότι οι χρήστες κλάσης 2, δηλαδή η χρήστες ανανέωσης έχουν λιγότερο station time.

Ωστόσο, αυτό το αποτέλεσμα του περίπου 70-30 καθορίζεται από το γεγονός ότι έχουν διαφορετικούς term χρόνους, δηλαδή οι χρήστες κλάσης 1 έχουν μεγαλύτερο term χρόνο όπου δεν χρησιμοποιούν το δίκτυο, δηλαδή τους σταθμούς του συστήματος, αλλά χρησιμοποιούν το UI κομμάτι του συστήματος. Γι αυτό και η ρυθμαπόδοση μελετάται με βάση το term της κάθε κλάσης.

Παρατηρούμε από την συνολική χρησιμοποίηση των σταθμών (του συστήματος- εκτος του χρόνου σκέψης των χρηστών) την πιο μεγάλη τιμή παρουσιάζει ο **DB Server CPU** όποτε εκεί εντοπίζεται η **στένωση** του συστήματος!

Βέβαια το αποτέλεσμα αυτό δεν θα μπορούσε να μας προκαλέσει έκπληξη καθώς παρατηρούμε ότι και για τις δύο κλάσεις στην ΚΜΕ του db server σπαταλιέται ο περισσότερος χρόνος, από κάθε άλλο σταθμό του συστήματος (προφανώς δεν μετράμε το term σε αυτή την ανάλυση).

Επίσης το γεγονός ότι η κμε του db server χρησιμοποιείται κατά (περίπου) 98% θα δημιουργεί πολλά προβλήματα στην λειτουργία της όσο και στις ανάγκες ψύξης της.

Σχετικά με τον χρόνο αναμονής που είναι περίπου 1.5sec για την ανταπόκριση του συστήματος είναι κατανοητό ότι δεν θέλουμε οι χρήστες να περιμένουν τόσο πολύ για το σύστημα να ανταποκριθεί αλλά πρέπει να βελτιώσουμε την υποδομή του.

Ερώτημα Β

Προκειμένου να βελτιωθεί η επίδοση του συστήματος μελετάται η αντικατάσταση της CPU του εξυπηρετητή βάσης δεδομένων με άλλη υψηλότερων επιδόσεων. Σημειώνεται ότι, λόγω ειδικών απαιτήσεων ακεραιότητας και ασφάλειας, η νέα CPU θα λειτουργεί σε κατάσταση πολυεπεξεργασίας και θα υφίσταται πρόσθετη υπολογιστική επιβάρυνση. Συνεπώς, η CPU του εξυπηρετητή βάσης δεδομένων θα έχει ρυθμό εξυπηρέτησης εξαρτώμενο από το φορτίο (LD). Υπολογίζεται ότι η απαίτηση εξυπηρέτησης $D_{ij}(k)$, $i=5$, $j=1, 2$, όταν στον σταθμό βρίσκονται k εργασίες ανεξαρτήτως κατηγορίας, θα δίνεται από τη σχέση:

$D_{ij}(k) = D_{ij} / (a + b \cdot k)$ όπου $a=5$, $b=0,01$ και D_{ij} η απαίτηση εξυπηρέτησης που αντιστοιχεί στη CPU του Db Server, όπως δίνεται στον Πίνακα για την περίπτωση σταθμού LI. Ζητείται να υπολογιστούν οι δείκτες επίδοσης όπως στο ερώτημα (α) λαμβάνοντας υπόψη την αντικατάσταση της CPU του Db Server. Υπάρχει βελτίωση σε σχέση με την επίδοση του (α);

Παρατηρήσαμε στο προηγούμενο ερώτημα ότι η στένωση του συστήματος εντοπίζεται στην cpu του db server και τώρα προσπαθούμε να την βελτιώσουμε.

Αρχικά, ορίζουμε τον πληθυσμό του συστήματος σύμφωνα με το αποτέλεσμα της προηγούμενης ανάλυσης.

Classes characteristics
Number, customized name, type of classes and number of customers (closed class) or arrival rate (open class). Add classes one by one or define total number at once.

Number:

*	Name	Type	No. of Customers	Arrival Rate (λ)
1	Class1	closed	338	
2	Class2	closed	142	

Μετά σειρά έχει η δήλωση των σταθμών του συστήματος και ο χαρακτηρισμός τους ως Load Independent και Delay για τους σταθμούς αναμονής και καθυστέρησης αντίστοιχα, ωστόσο τώρα ρυθμίζουμε ως load dependent τον σταθμό DB Server CPU.

Stations characteristics
Number, customized name and type of stations. Add stations one by one or define the total number at once. Load Dependent stations necessarily require the use of MVA.

Number:

*	Name	Type
1	LAN	Delay (Infinite Server)
2	APP Server CPU	Load Independent
3	APP Server Disk	Load Independent
4	DB Server CPU	Load Dependent
5	DB Server Disk	Load Independent
6	Users	Delay (Infinite Server)

Στη συνέχεια ορίζουμε τις απαιτήσεις σε μέσο χρόνο χρήστη, δηλαδή σε μέσο χρόνο σκέψης, και σε μέση συνολική απαίτηση εξυπηρέτησης για κάθε σταθμό. Οι τιμές που φαίνονται στον πίνακα είναι της τάξεως των sec. Ωστόσο για τον DB Server CPU συνεχίζουμε την ανάλυση και εφαρμόζουμε τον τύπο της εκφώνησης.

Service Times
 Input service times of each station for each class.
 If the station is "Load Dependent" you can set the service times for each number of customers by double-click on "LD Settings..." button.
 Press "Service Demands" button to enter service demands instead of service times and visits.
 MULTICLASS MODELS: when for a station the per-class service times are different, the results are correct ONLY IF its scheduling discipline is assumed Processor Sharing (PS) and not FCFS (See BCMP Theorem).

*	Class1	Class2
LAN	0.0470	0.0240
APP Server...	0.0430	0.0190
APP Server...	0.0440	0.0290
DB Server ...	LD Settings...	LD Settings...
DB Server ...	0.0410	0.0370
Users	25.0000	19.0000

Επιλέγοντας το LD Setting για την κλάση 1 βάζουμε τον εξής τυπο επιλέγοντας όλα τα n και πατάμε evaluate.

Editing Service Times of DB Server CPU

*	Class1
473	0.0046
474	0.0046
475	0.0046
476	0.0046
477	0.0046
478	0.0046
479	0.0046
480	0.0046

Enter the expression to be evaluated, using **n** for the number of customers. For syntax details and a list of supported functions click on "Help"

$0.045/(5+0.01*n)$ Evaluate

Αντίστοιχα, επιλέγοντας το LD Setting για την κλάση 2 βάζουμε τον εξής τυπο επιλέγοντας όλα τα n και πατάμε evaluate.

Editing Service Times of DB Server CPU

*	Class2
473	0.0061
474	0.0061
475	0.0061
476	0.0060
477	0.0060
478	0.0060
479	0.0060
480	0.0000

Enter the expression to be evaluated, using **n** for the number of customers. For syntax details and a list of supported functions click on "Help"

$0.059/(5+0.01*n)$ Evaluate

OK Cancel Help

Στην συνέχεια ορίσαμε Preference Station για τις 2 κατηγορίες.

Reference Station The station is used to compute the system throughput and the system response time for each closed class. Performance metrics of open classes are always computed with respect to the arrival process. Visits at the Reference station can not be Zero. WARNING: the reference station for all closed classes is forced to be the same station.	Class	Station
	Class1	Users
	Class2	Users

Τέλος, αφήσαμε τον default algorithm ανάλυσης, τον MVA.

Έτσι τρέχοντας την προσομοίωση βλέπουμε ότι:

Ο ρυθμός απόδοσης του συστήματος συνολικά και για καθεμία από τις κατηγορίες πελατών είναι:

Throughput				
Throughput of each class for each station. System Throughput is t				
*	Aggregate	Class1	Class2	
System	20.5046	13.1948	7.3098	

Ο χρόνος απόκρισης του συστήματος συνολικά και για καθεμία από τις κατηγορίες πελατών είναι(σε sec):

System Response Time				
The global aggregate is the "System Response Time" and is obtained weighting the aggregated values by the relative per-class throughput. A: This value of System Response Time includes the Residence Time of the Reference Station. B: This value of System Response Time does NOT include the Residence Time of the Reference Station. Notice: For open classes the Reference Station always coincides with the arrival process. Thus the B values are not computed.				
*	Aggregate	Class1	Class2	
A	23.4094	25.6162	19.4260	
B	0.5484	0.6162	0.4260	

Ο βαθμός χρησιμοποίησης των σταθμών του συστήματος είναι ο εξής:

Utilization				
Utilization of a customer class at the selected station. 1).				
*	Aggregate	Class1	Class2	
LAN	0.7956	0.6202	0.1754	
APP Serve...	0.7063	0.5674	0.1389	
APP Serve...	0.7926	0.5806	0.2120	
DB Server...	0.2046	0.1185	0.0861	
DB Server...	0.8114	0.5410	0.2705	
Users	468.7559	329.8699	138.8860	

Προσοχή στην ανάγνωση του πίνακα όπου η συνολική χρησιμοποίηση του σταθμού lan πρέπει να μεταφραστεί ως 70.63%. Το ίδιο ισχύει και για τους υπόλοιπους σταθμούς.

Γενικός Σχολιασμός Αποτελεσμάτων Β Ερωτήματος

Αρχικά, η συνάρτηση είναι φθίνουσα και έτσι τα station time για την cpu του db server για την κλάση 1 κυμαίνεται μεταξύ των τιμών: 9msec - 4.6msec

Ενώ για τη κλάση 2 κυμαίνονται μεταξύ των τιμών: 11.8msec-6 msec

Οπότε αρχικά παρατηρούμε μια μεγάλη βελτίωση της επίδοσης της cpu.

Ας συγκρίνουμε τώρα τα αποτελέσματα ως προς τις τρεις μετρικές:

Όπου αριστερά είναι το αποτέλεσμα του ερωτήματος A και δεξιά το αποτέλεσμα του ερωτήματος B.

Throughput				
Throughput of each class for each station. System Throughput is the sum of the class throughputs.				
*	Aggregate	Class1	Class2	
System	19.5972	12.7295	6.8678	

Throughput				
Throughput of each class for each station. System Throughput is the sum of the class throughputs.				
*	Aggregate	Class1	Class2	
System	20.5046	13.1948	7.3098	

Αρχικά παρατηρούμε βελτίωση της ρυθμαπδόσης, τόσο της συνολικής όσο και της ρυθμαπδόσης της κάθε κλάσης και οφείλεται στην βελτίωση της επίδοσης της cpu της db. Η ρυθμαπδόση ωστόσο δεν είναι τόσο εντυπωσιακή όσο η εντυπωσιακή βελτίωση της βελτίωσης της cpu του db server καθώς οι υπόλοιποι σταθμοί και οι χρόνοι χρήσης παραμένουν σταθερή.

System Response Time				System Response Time			
The global aggregate is the "System Response Time".				The global aggregate is the "System Response Time".			
A: This value of System Response Time includes the response time of the reference station.				A: This value of System Response Time includes the response time of the reference station.			
B: This value of System Response Time does NOT include the response time of the reference station.				B: This value of System Response Time does NOT include the response time of the reference station.			
Notice: For open classes the Reference Station always has a response time of 0.				Notice: For open classes the Reference Station always has a response time of 0.			
*	Aggregate	Class1	Class2	*	Aggregate	Class1	Class2
A	24.4933	26.5525	20.6763	A	23.4094	25.6162	19.4260
B	1.5959	1.5525	1.6763	B	0.5484	0.6162	0.4260

Παρατηρούμε εντυπωσιακή βελτίωση στον χρόνο του συστήματος, όπου πριν ο χρήστης περίμενε περίπου 1.5 sec για το σύστημα να βρεί την απάντηση που αναζητούσε τώρα θέλουμε μόλις 0.5 sec, χρόνος ανεκτός για την δεδομένη εφαρμογή.

Utilization					Utilization				
Utilization of a customer class at the selected station. The utilization is the sum of the utilization of the classes.					Utilization of a customer class at the selected station. The utilization is the sum of the utilization of the classes.				
*	Aggregate	Class1	Class2		*	Aggregate	Class1	Class2	
LAN	0.7631	0.5983	0.1648		LAN	0.7956	0.6202	0.1754	
APP Serve...	0.6779	0.5474	0.1305		APP Serve...	0.7063	0.5674	0.1389	
APP Serve...	0.7593	0.5601	0.1992		APP Serve...	0.7926	0.5806	0.2120	
DB Server...	0.9780	0.5728	0.4052		DB Server...	0.2046	0.1185	0.0861	
DB Server...	0.7760	0.5219	0.2541		DB Server...	0.8114	0.5410	0.2705	
Users	448.7241	318.2369	130.4873		Users	468.7559	329.8699	138.8860	

Παρατηρούμε ότι αυξήθηκε η χρησιμοποίηση όλων των σταθμών για όλες τις κατηγορίες και για τις δυο κλάσεις εκτός από την cpu του db server όπου εντοπίζεται μεγάλη πτώση της χρησιμοποίησης της και σε αυτό συμβαίνει εξαιτίας της πολύ πιο καλής station time της. Επίσης έτσι αντιμετωπίσαμε και το πρόβλημα της υπερβολικά μεγάλης χρήσης της cpu του σερβερ της βάσης δεδομένων.

Οπότε συνολικά και ειδικά το σύστημα μας τρέχει πολύ καλύτερα.

Ερώτημα Γ

Προκειμένου να απλουστευθεί η ανάλυση, θα ορίσουμε με τα αρχικά δεδομένα αντίστοιχο συγκεντρωτικό μοντέλο μιας κατηγορίας, η οποία θα αποτελεί συνένωση των δύο κατηγοριών. Η τιμή της απαίτησης εξυπηρέτησης σε κάθε σταθμό του μοντέλου μιας κατηγορίας θα τεθεί ίση με τον μέσο όρο των επιμέρους τιμών σταθμισμένων με τον ρυθμό απόδοσης της αντίστοιχης κατηγορίας, όπως υπολογίστηκε στο (α). (Η CPU του Db Server θα ληφθεί ως σταθμός LI, όπως αρχικά.) Να υπολογιστούν οι δείκτες επίδοσης του μοντέλου μιας κατηγορίας και να συγκριθούν με τα αποτελέσματα του (α).

Σε αυτό το σημείο θα προσπαθήσουμε να κάνουμε μια **απλούστευση** στο μοντέλο μας χρησιμοποιώντας μόνο μια κλάση πληθυσμού η οποία θα συνδυάζει κατάλληλα τις δυο επιμέρους κλάσης του συστήματος μας.

Έτσι αυτή η εννοιά κλάση θα πρέπει να έχει πληθυσμό ίσο με το συνολικό πληθυσμό του συστήματος δηλαδή 480, όπως φαίνεται και στην είσοδο της προσομοίωσης παρακάτω:

Classes characteristics
Number, customized name, type of classes and number of customers (closed class) or arrival rate (open class). Add classes one by one or define total number at once.

Number: 1

New Class

*	Name	Type	No. of Customers	Arrival Rate (λ)
1	Class1	closed	480	

Μετά σειρά έχει η δήλωση των σταθμών του συστήματος και ο χαρακτηρισμός τους ως Load Independent και Delay για τους σταθμούς αναμονής και καθυστέρησης αντίστοιχα. Τα ορίζουμε όπως ακριβώς στο πρώτο ερώτημα.

Stations characteristics
Number, customized name and type of stations. Add stations one by one or define the total number at once. Load Dependent stations necessarily require the use of MVA.

Number: 6

New Station

*	Name	Type
1	LAN	Delay (Infinite Server)
2	APP Server CPU	Load Independent
3	APP Server Disk	Load Independent
4	DB Server CPU	Load Independent
5	DB Server Disk	Load Independent
6	Users	Delay (Infinite Server)

Στη συνέχεια ορίσαμε τις απαιτήσεις σε μέσο χρόνο χρήστη, δηλαδή σε μέσο χρόνο σκέψης, και σε μέση συνολική απαίτηση εξυπηρέτησης για κάθε σταθμό.

Η διαδικασία που ακολουθήθηκε για να βγουν αυτές οι τιμές είναι με χρήση του σταθμισμένου μέσου όρου όπου το w_i είναι το throughput της κλάσης i ($i=1,2$) όπως αυτό βρέθηκε στο ερώτημα Α, x_i είναι το service time ενός συγκεκριμένου station για τη κλάση i , όπως ακριβώς δίνεται στην εκφώνηση. Έτσι λοιπόν ο σταθμισμένος μ.ο δίνεται από τη σχέση:

$$\bar{x} = \frac{\sum_{i=1}^2 w_i \cdot x_i}{\sum_{i=1}^2 w_i} = \frac{w_1 \cdot x_1 + w_2 \cdot x_2}{w_1 + w_2}$$

π.χ για το πρώτο κελί έχουμε ότι: $\bar{x}_\alpha = \frac{47 \cdot 12.7295 + 24 \cdot 6.8678}{19.469} = 39.196 \text{ msec}$

Αντίστοιχα και για τα υπόλοιπα κελιά.

Έτσι, συνολικά έχουμε τον εξής πίνακα, όπου οι τιμές που φαίνονται στον πίνακα είναι της τάξεως των sec:

Service Times Input service times of each station for each class. If the station is "Load Dependent" you can set the service times for each number of customers by double-click on "LD Settings..." button. Press "Service Demands" button to enter service demands instead of service times and visits. MULTICLASS MODELS: when for a station the per-class service times are different, the results are correct ONLY IF its scheduling discipline is assumed Processor Sharing (PS) and not FCFS (See BCMP Theorem).	
*	Class1
LAN	0.0392
APP Server...	0.0348
APP Server...	0.0390
DB Server ...	0.0502
DB Server ...	0.0399
User	23.0482

Στην συνέχεια ορίσαμε Preference Station για την μοναδική κατηγορία μας:

Reference Station The station is used to compute the system throughput and the system response time for each closed class . Performance metrics of open classes are always computed with respect to the arrival process . Visits at the Reference station can not be Zero. WARNING: the reference station for all closed classes is forced to be the same station.	Class	Station
	Class1	User

Τέλος, αφήσαμε τον default algorithm ανάλυσης, τον MVA.

Έτσι βλέπουμε ότι:

Ο ρυθμός απόδοσης του συστήματος είναι:

Throughput

Throughput of each class for each station.

*	Aggregate	Class1
System	19.4604	19.4604

Ο χρόνος απόκρισης του συστήματος είναι:

System Response Time

The global aggregate is the "System Response Time" and is obtained weighting the aggregated values by the relative per-class throughput.

A: This value of System Response Time **includes** the Residence Time of the Reference Station.

B: This value of System Response Time **does NOT include** the Residence Time of the Reference Station.

Notice: For **open classes** the Reference Station always coincides with the arrival process. Thus the **B** values are not computed.

*	Aggregate	Class1
A	24.6655	24.6655
B	1.6172	1.6172

Ο βαθμός χρησιμοποίησης των σταθμών του συστήματος είναι ο εξής:

Utilization

Utilization of a customer class at the se

*	Aggregate	Class1
LAN	0.7628	0.7628
APP Serve...	0.6776	0.6776
APP Serve...	0.7588	0.7588
DB Server...	0.9776	0.9776
DB Server...	0.7757	0.7757
User	448.5277	448.5277

Γενικός Σχολιασμός Αποτελεσμάτων Γ Ερωτήματος

Αρχικά ας σχολιάσουμε ότι δεν πήραμε τον κλασσικό μέσο όρο αλλά τον σταθμισμένο ως προς την ρυθμαπόδοση καθώς θέλαμε να έχουμε ένα δίκαιο σύστημα που προσομοιάζει όσο πιο πολύ το αρχικό-αναλυτικό σύστημα. Έτσι του δίνουμε station time σταθμισμένο ως προς την συνολική ρυθμαπόδοση των κλάσεων, θα μπορούσαμε να ήμασταν ακόμη πιο συγκεκριμένη και με πιο καλή προσέγγιση δίνοντας όχι την συνολική ρυθμαπόδοση της κλάσης αλλά την ρυθμαπόδοση της κλάσης επι του σταθμού. Ωστόσο επειδή η συνολική ρυθμαπόδοση ανάγεται από τις επιμέρους είμαστε με μια μεγαλύτερη απόκλιση κοντά στο αρχικό μοντέλο.

Ας συγκρίνουμε το throughput:

Throughput				
Throughput of each class for each station. System Throughput				
*	Aggregate	Class1	Class2	
System	19.5972	12.7295	6.8678	

Throughput			
Throughput of each class for each station.			
*	Aggregate	Class1	
System	19.4604	19.4604	

Αριστερά είναι το throughput στο ερώτημα A και δεξιά στο ερώτημα B, όπου έχει γίνει η απλουστευση του συστήματος.

Νόημα έχει να συγκρίνουμε μόνο το κελί aggregate στο οποίο βλέπουμε ότι έχουμε σχεδόν ίδια τιμή σε μεγάλο βαθμό, το αν αυτή η απόκλιση γίνει ανεκτή ή όχι εξαρτάται από την ακρίβεια που θέλουμε για τη μελέτη μας.

Ας συγκρίνουμε τον χρόνο απόκρισης του συστήματος:

System Response Time				
The global aggregate is the "System Response Time" A: This value of System Response Time includes the B: This value of System Response Time does NOT include Notice: For open classes the Reference Station always				
*	Aggregate	Class1	Class2	
A	24.4933	26.5525	20.6763	
B	1.5959	1.5525	1.6763	

System Response Time			
The global aggregate is the "System Response Time" A: This value of System Response Time includes the B: This value of System Response Time does NOT include Notice: For open classes the Reference Station always			
*	Aggregate	Class1	
A	24.6655	24.6655	
B	1.6172	1.6172	

Αριστερά είναι ο χρόνος απόκρισης στο ερώτημα A και δεξιά στο ερώτημα B, όπου έχει γίνει η απλουστευση του συστήματος.

Νόημα έχει να συγκρίνουμε μόνο τα κελιά aggregate στα οποία βλέπουμε ότι έχουμε σχεδόν ίδιες τιμές σε μεγάλο βαθμό, το αν αυτή η απόκλιση γίνει ανεκτή ή όχι εξαρτάται από την ακρίβεια που θέλουμε για τη μελέτη μας.

Ας συγκρίνουμε τον βαθμό χρησιμοποίησης των σταθμών:

Utilization Utilization of a customer class at the selected station.			
*	Aggregate	Class1	Class2
LAN	0.7631	0.5983	0.1648
APP Serve...	0.6779	0.5474	0.1305
APP Serve...	0.7593	0.5601	0.1992
DB Server...	0.9780	0.5728	0.4052
DB Server...	0.7760	0.5219	0.2541
Users	448.7241	318.2369	130.4873

Utilization Utilization of a customer class at the selected station.		
*	Aggregate	Class1
LAN	0.7628	0.7628
APP Serve...	0.6776	0.6776
APP Serve...	0.7588	0.7588
DB Server...	0.9776	0.9776
DB Server...	0.7757	0.7757
User	448.5277	448.5277

Αριστερά είναι ο χρόνος απόκρισης στο ερώτημα A και δεξιά στο ερώτημα B, όπου έχει γίνει η απλουστευση του συστήματος.

Νόημα έχει να συγκρίνουμε μόνο τα κελιά aggregate στα οποία βλέπουμε ότι έχουμε σχεδόν ίδιες τιμές σε μεγάλο βαθμό, το αν αυτή η απόκλιση γίνει ανεκτή ή όχι εξαρτάται από την ακρίβεια που θέλουμε για τη μελέτη μας.

Παρατηρούμε από την συνολική χρησιμοποίηση των σταθμών (του συστήματος- εκτος του χρονου σκέψης των χρηστών) την πιο μεγάλη τιμή παρουσιάζει ο **DB Server CPU** όποτε εκεί εντοπίζεται η στένωση του συστήματος!

Την ίδια παρατήρηση κάναμε και στο ερώτημα A, όπου δεν είχαμε θέσει απλοποιήσεις στο σύστημα.

Ερώτημα Δ

Θεωρούμε το μοντέλο μιας κατηγορίας που ορίστηκε στο (γ) υποθέτοντας ότι η CPU του Db Server αντικαθίσταται από σταθμό LD, κατ' αναλογία με το ερώτημα (β). Στο μοντέλο μιας κατηγορίας, η απαίτηση εξυπηρέτησης $D_i(k)$, $i=5$, όταν στον σταθμό βρίσκονται k εργασίες, θα δίνεται από τη σχέση: $D_i(k) = D_i / (a + b \cdot k)$ όπου $a=5$, $b=0,01$ και D_i η απαίτηση εξυπηρέτησης που αντιστοιχεί στη CPU του Db Server, όπως υπολογίστηκε στο (γ) για την περίπτωση σταθμού LI. Ζητείται να υπολογιστούν οι δείκτες επίδοσης του μοντέλου μιας κατηγορίας λαμβάνοντας υπόψη την αντικατάσταση της CPU. Υπάρχει βελτίωση σε σχέση με το ερώτημα (γ);

Παρατηρήσαμε στο προηγούμενο ερώτημα ότι η στένωση του συστήματος εντοπίζεται στην cpu του db server και τώρα προσπαθούμε να την βελτιώσουμε.

Αρχικά, ορίζουμε τον πληθυσμό του συστήματος:

Classes characteristics
Number, customized name, type of classes and number of customers (closed class) or arrival rate (open class). Add classes one by one or define total number at once.

Number:

New Class

*	Name	Type	No. of Customers	Arrival Rate (λ)
1	Class1	closed	480	

Μετά σειρά έχει η δήλωση των σταθμών του συστήματος και ο χαρακτηρισμός τους ως Load Independent και Delay για τους σταθμούς αναμονής και καθυστέρησης αντίστοιχα, ωστόσο τώρα ρυθμίζουμε ως load dependent τον σταθμό DB Server CPU.

Stations characteristics
Number, customized name and type of stations. Add stations one by one or define the total number at once. Load Dependent stations necessarily require the use of MVA.

Number:

New Station

*	Name	Type
1	LAN	Delay (Infinite Server)
2	APP Server CPU	Load Independent
3	APP Server Disk	Load Independent
4	DB Server CPU	Load Dependent
5	DB Server Disk	Load Independent
6	Users	Delay (Infinite Server)

Στη συνέχεια ορίζουμε τις απαιτήσεις σε μέσο χρόνο χρήστη, δηλαδή σε μέσο χρόνο σκέψης, και σε μέση συνολική απαίτηση εξυπηρέτησης για κάθε σταθμό. Οι τιμές που φαίνονται στον πίνακα είναι της τάξεως των sec. Ωστόσο για τον DB Server CPU συνεχίζουμε την ανάλυση και εφαρμόζουμε τον τύπο της εκφώνησης.

Service Times
 Input service times of each station for each class.
 If the station is "Load Dependent" you can set the service times for each number of customers by double-click on "LD Settings..." button.
 Press "Service Demands" button to enter service demands instead of service times and visits.
 MULTICLASS MODELS: when for a station the per-class service times are different, the results are correct ONLY IF its scheduling discipline is assumed Processor Sharing (PS) and not FCFS (See BCMP Theorem).

*	Class1
LAN	0.0392
APP Server...	0.0348
APP Server...	0.0390
DB Server ...	LD Settings...
DB Server ...	0.0399
Users	23.0482

Επιλέγοντας το LD Setting για την κλάση 1 βάζουμε τον εξής τυπο επιλέγοντας όλα τα n και πατάμε evaluate.

Editing Service Times of DB Server CPU

*	Class1
473	0.0052
474	0.0052
475	0.0051
476	0.0051
477	0.0051
478	0.0051
479	0.0051
480	0.0051

Enter the expression to be evaluated, using *n* for the number of customers. For syntax details and a list of supported functions click on "Help"

0.0502/(5+0.01*n) Evaluate

OK Cancel Help

Στην συνέχεια ορίσαμε Preference Station για τις 2 κατηγορίες.

Reference Station
 The station is used to compute the system throughput and the system response time for each closed class.
 Performance metrics of open classes are always computed with respect to the arrival process. Visits at the Reference station can not be Zero.
WARNING: the reference station for all closed classes is forced to be the same station.

Class	Station
Class1	Users

Τέλος, αφήσαμε τον default algorithm ανάλυσης, τον MVA.

Έτσι βλέπουμε ότι:

Ο ρυθμός απόδοσης του συστήματος είναι:

Throughput

Throughput of each class for each station.

*	Aggregate	Class1
System	20.3367	20.3367

Ο χρόνος απόκρισης του συστήματος είναι:

System Response Time

The global aggregate is the "System Response Time" and is obtained weighting the aggregated values by the relative per-class throughput.

A: This value of System Response Time **includes** the Residence Time of the Reference Station.

B: This value of System Response Time **does NOT include** the Residence Time of the Reference Station.

Notice: For **open classes** the Reference Station always coincides with the arrival process. Thus the **B** values are not computed.

*	Aggregate	Class1
A	23.6027	23.6027
B	0.5545	0.5545

Ο βαθμός χρησιμοποίησης των σταθμών του συστήματος είναι ο εξής:

Utilization

Utilization of a customer class at the customers in the station (it may be greater than 1.0000)

*	Aggregate	Class1
LAN	0.5497	0.5497
APP Serve...	0.7077	0.7077
APP Serve...	0.7931	0.7931
DB Server...	0.2037	0.2037
DB Server...	0.8114	0.8114
Users	1.0000	1.0000

Γενικός Σχολιασμός Αποτελεσμάτων Δ Ερωτήματος

Αρχικά, η συνάρτηση είναι φθίνουσα και έτσι τα station time για την cpu του db server κυμαίνεται μεταξύ των τιμών: 10msec - 5.1msec

Οπότε αρχικά παρατηρούμε μια μεγάλη βελτίωση της επίδοσης της cpu.

Ας συγκρίνουμε τώρα τα αποτελέσματα ως προς τις τρεις μετρικές:

Όπου αριστερά είναι το αποτέλεσμα του ερωτήματος Δ και δεξιά το αποτέλεσμα του ερωτήματος Γ.

Throughput		
Throughput of each class for each station.		
*	Aggregate	Class1
System	20.3367	20.3367

Throughput		
Throughput of each class for each station.		
*	Aggregate	Class1
System	19.4604	19.4604

Αρχικά παρατηρούμε βελτίωση της ρυθμαπόδοσης, τόσο της συνολικής όσο και της ρυθμαπόδοσης της κάθε κλάσης και οφείλεται στην βελτίωση της επίδοσης της cpu της db. Η ρυθμαπόδοση ωστόσο δεν είναι τόσο εντυπωσιακή όσο η εντυπωσιακή βελτίωση της βελτίωσης της cpu του db server καθώς οι υπόλοιποι σταθμοί και οι χρόνοι χρήσης παραμένουν σταθερή.

Άρα παρατηρούμε όντως επίδοση τόσο στο απλοποιημένο όσο και στο πιο ρεαλιστικό μοντέλο.

Αριστερά είναι το αποτέλεσμα του ερωτήματος Δ και δεξιά το αποτέλεσμα του ερωτήματος Γ.

System Response Time		
The global aggregate is the "System Response Time".		
A: This value of System Response Time		
B: This value of System Response Time		
Notice: For open classes the Reference is not computed.		
*	Aggregate	Class1
A	23.6027	23.6027
B	0.5545	0.5545

System Response Time		
The global aggregate is the "System Response Time".		
A: This value of System Response Time		
B: This value of System Response Time		
Notice: For open classes the Reference is not computed.		
*	Aggregate	Class1
A	24.6655	24.6655
B	1.6172	1.6172

Παρατηρούμε εντυπωσιακή βελτίωση στον χρόνο του συστήματός, όπου πριν ο χρήστης περίμενε περίπου 1.6 sec για το σύστημα να βρεί την απάντηση που αναζητούσε τώρα θέλουμε μόλις 0.55 sec, χρόνος ανεκτός για την δεδομένη εφαρμογή.

Άρα παρατηρούμε όντως επίδοση τόσο στο απλοποιημένο όσο και στο πιο ρεαλιστικό μοντέλο.

Αριστερά είναι το αποτέλεσμα του ερωτήματος Δ και δεξιά το αποτέλεσμα του ερωτήματος Γ.

Utilization			Utilization		
Utilization of a customer class at the customers in the station (it may be gr			Utilization of a customer class at the sel		
*	Aggregate	Class1	*	Aggregate	Class1
LAN	0.5497	0.5497	LAN	0.7628	0.7628
APP Serve...	0.7077	0.7077	APP Serve...	0.6776	0.6776
APP Serve...	0.7931	0.7931	APP Serve...	0.7588	0.7588
DB Server...	0.2037	0.2037	DB Server...	0.9776	0.9776
DB Server...	0.8114	0.8114	DB Server...	0.7757	0.7757
Users	1.0000	1.0000	User	448.5277	448.5277

Παρατηρούμε ότι αυξήθηκε η χρησιμοποίηση όλων των σταθμών για όλες τις κατηγορίες και για τις δυο κλάσεις εκτός από την cpu του db server όπου εντοπίζεται μεγάλη πτώση της χρησιμοποίησης της και σε αυτό συμβαίνει εξαιτίας της πολύ πιο καλής station time της. Επίσης έτσι αντιμετωπίσαμε και το πρόβλημα της υπερβολικά μεγάλης χρήσης της cpu του σερβερ της βάσης δεδομένων.

Οπότε συνολικά και ειδικά το σύστημα μας τρέχει πολύ καλύτερα και **υπάρχει βελτίωση**. Επίσης παρατηρούμε ότι εξάγουμε τα ίδια ακριβώς συμπεράσματα στο ρεαλιστικό και στο απλοποιημένο μοντέλο. Επομένως η απλοποίηση ήταν εύστοχη.

Αξίζει να σημειωθεί ότι το γεγονός ότι το utilization για τους users (term) βγαίνει ίσο με την μονάδα το οποίο είναι λάθος και προκύπτει από πιθανό **bug** του συστήματος. Κανονικά πρέπει να είναι ίσο με το μέσο αριθμό εργασιών!