Κεφάλαιο 2 : Φορτίο εργασίας, μετρήσεις και πειράματα

Τεχνικές Εκτίμησης Υπολογιστικών συστημάτων

Γιάννης Γαροφαλάκης

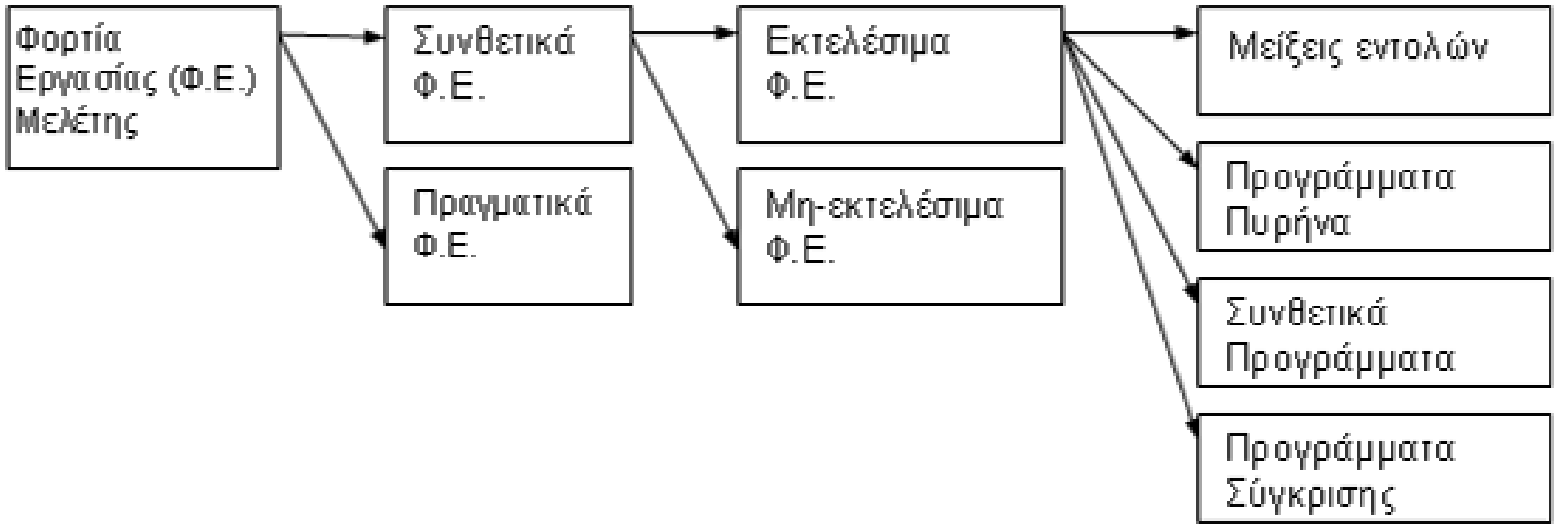
Καθηγητής

Ορισμός φορτίου εργασίας

- Φορτίο Εργασίας (ΦΕ) είναι το σύνολο των απαιτήσεων επεξεργασίας που επιβάλλονται σε ένα σύστημα από την κοινότητα των χρηστών του, στη διάρκεια κάποιας χρονικής περιόδου.

- ΦΕ Μελέτης: ΦΕ που χρησιμοποιείται σε μελέτη απόδοσης πληροφοριακού συστήματος.

Ιεραρχία κατηγοριών φορτίου εργασίας



- Πραγματικό φορτίο εργασίας: παρατηρείται σε σύστημα υπό πραγματικές συνθήκες
- Συνθετικό φορτίο εργασίας: τεχνητή κατασκευή ειδική για μελέτες, που χρησιμοποιείται με ελεγχόμενο τρόπο
  - *Μη-εκτελέσιμο* : δεν είναι δυνατή η χρησιμοποίησή του σε πειράματα απευθείας στο πραγματικό σύστημα(π.χ. μέση τιμή, διασπορά)
  - *Εκτελέσιμο :* χρησιμοποιείται και σε πραγματικό σύστημα

Πλεονεκτήματα συνθετικών φορτίων εργασίας

- Αποτελούν **αξιόπιστα** μοντέλα των πραγματικών φορτίων εργασίας
- **Δεν χρειάζονται συλλογές πραγματικών δεδομένων**, οι οποίες συνήθως είναι τεράστιες και είναι πιθανό να περιέχουν ευαίσθητα στοιχεία
- Μπορούν να **τροποποιηθούν** εύκολα χωρίς να επηρεάζεται η λειτουργία του συστήματος
- Μπορούν εύκολα να **μεταφερθούν** σε διάφορα συστήματα, κυρίως λόγω του μικρού μεγέθους τους
- Πολλές φορές έχουν ενσωματωμένους **μηχανισμούς μετρήσεων**

Μείξεις εντολών (1)

Μείξη εντολών : σύνολο διαφόρων εντολών χαμηλού επιπέδου του συστήματος, μαζί με τις σχετικές συχνότητες εμφάνισης των εντολών αυτών.

i	Τύπος Εντολής	f_i
1	Load and Store	31.2
2	Fixed-Point Add and Subtract	6.1
3	Compares	3.8
4	Branches	16.6
5	Floating Add and Subtract	6.9
6	Floating Multiply	3.8
7	Floating Divide	1.5
8	Fixed-Point Multiply	0.6
9	Fixed-Point Divide	0.2
10	Shifting	4.4
11	Logical, And, Or	1.6
12	Εντολές που δεν χρησιμοποιούν καταχωρητές	5.3
13	Indexing	18.0
		100.0

Μείξη εντολών Gibson (1959)

Μείξεις εντολών (2)

- Με την εκτέλεση ενός συνόλου αντιπροσωπευτικών προγραμμάτων σε κάθε ένα από τα συστήματα που θα μελετήσουμε και την καταγραφή του αριθμού εμφανίσεων των επιμέρους τύπων εντολών, μπορούμε να υπολογίσουμε τη σχετική συχνότητα f_i εμφάνισης του τύπου εντολών *i* (γενικά ή σε κάθε σύστημα)
- Γνωρίζοντας τους χρόνους εκτέλεσης t_i κάθε τύπου εντολών *i* για το συγκεκριμένο σύστημα, μπορούμε να υπολογίσουμε το μέσο χρόνο εκτέλεσης της μείξης εντολών για κάθε σύστημα, $\tau = \sum_i f_i t_i$

Μείξεις εντολών (3)

- Συγκρίνοντας τις τιμές του τ για τα διαφορετικά συστήματα μπορούμε να πάρουμε αποφάσεις σχετικά με τη σχετική απόδοσή τους. Αν διαλέξουμε ως μονάδα χρόνου το microsecond (μsec), τότε η ποσότητα $1/\tau$ αντιπροσωπεύει την ταχύτητα της CPU σε MIPS.

Μειονεκτήματα μείξεων εντολών

- Οι σημερινοί υπολογιστές υποστηρίζουν περισσότερους πολύπλοκους τύπους εντολών που δεν αντιπροσωπεύονται στις γνωστές μείξεις.
- Ο χρόνος εκτέλεσης μιας εντολής t_i δεν είναι πλέον σταθερός.
- Οι μείξεις δεν παίρνουν υπόψη τους νέες δυνατότητες όπως η ιδεατή διευθυνσιοδότηση κ.α.
- Οι μείξεις εντολών είναι χρήσιμες ουσιαστικά μόνο όταν πρόκειται να συγκρίνουμε την ταχύτητα επεξεργαστών σε συστήματα με παρόμοιες αρχιτεκτονικές.

Προγράμματα πυρήνα (1)

- **Προγράμματα πυρήνα (kernels)**: μικρά τμήματα προγράμματος που αντιπροσωπεύουν τον εσωτερικό βρόχο μιας συχνά χρησιμοποιούμενης εφαρμογής
  - Σύνολο εντολών ή μια λειτουργία που εμφανίζεται συχνά στο βασικό πυρήνα εκτέλεσης μιας ή περισσοτέρων κατηγοριών εφαρμογών του συστήματος
  - Γενίκευση των μείξεων εντολών
  - Π.χ. μία ρουτίνα αντιστροφής πίνακα, ένα πρόγραμμα επίλυσης διαφορικών εξισώσεων

Προγράμματα πυρήνα (2)

+	-
Χρήσιμα στα αρχικά στάδια σχεδιασμού των τμημάτων ενός συστήματος	Διατηρούν τα περισσότερα μειονεκτήματα των μείξεων εντολών
Χρήσιμα για αξιολόγηση τμημάτων λογισμικού	Δεν κάνουν χρήση των συσκευών εισόδου-εξόδου
Δεν χρησιμοποιούν πραγματικά δεδομένα του συστήματος	

Συνθετικά προγράμματα (1)

- Συνθετικά προγράμματα: “τεχνητά” προγράμματα που περιέχουν όλα τα σημαντικά συστατικά των προγραμμάτων, για να δοκιμαστούν διάφορα τμήματα των συστημάτων
  - Απλά προγράμματα δοκιμής τα οποία κάνουν κλήσεις των υπηρεσιών του λειτουργικού συστήματος και αιτήσεις για χρήση των συσκευών εισόδου/εξόδου
  - Γραμμένα συνήθως σε γλώσσα υψηλού επιπέδου όπως FORTRAN, Pascal, C
  - Περιλαμβάνουν μεταβλητές παραμέτρους ώστε να μπορούν να περιγράψουν ένα ευρύ φάσμα χαρακτηριστικών των προγραμμάτων (ύψος των απαιτήσεων σε CPU, οι απαιτήσεις σε χώρο αποθήκευσης, ο αριθμός προσπελάσεων στο δίσκο)

Συνθετικά προγράμματα (2)

+	-
Κατασκευάζονται και τροποποιούνται εύκολα	Δεν μπορούν να κάνουν αντιπροσωπευτικές αναφορές στη μνήμη και στους δίσκους του συστήματος
Είναι ανεξάρτητα από οποιοδήποτε υλικό	Δεν είναι κατάλληλα για πολυχρηστικά περιβάλλοντα
Δεν χρησιμοποιούν πραγματικά δεδομένα	

Προγράμματα σύγκρισης

- Προγράμματα σύγκρισης (**benchmarks**) : πλήρη προγράμματα γραμμένα σε γλώσσα υψηλού επιπέδου, τα οποία θεωρούνται αντιπροσωπευτικά κλάσεων προγραμμάτων εφαρμογών.
  - Αξιολογούν την απόδοση του συνολικού συστήματος σε ρεαλιστικές συνθήκες.
  - Είναι κατασκευασμένα έτσι ώστε να χρησιμοποιούν και να αξιολογούν τους περισσότερους πόρους του συστήματος.
  - Χρησιμοποιούνται συχνά στη διαδικασία επιλογής προμηθευτή για ένα σύστημα.
  - Γνωστά προγράμματα σύγκρισης
    - TPC benchmarks,
    - SPEC

TPC benchmarks

- Προδιαγράφηκαν από το TPC (Transaction Processing Performance Council), στο οποίο συμμετέχουν κατασκευαστές, χρήστες (π.χ. Τράπεζες), σύμβουλοι συστημάτων επεξεργασίας διεργασιών (transaction processing systems) και δημιουργήθηκε το 1988.
- Εκτιμούν την απόδοση επεξεργαστών, υποσυστήματος I/O, του ΛΣ και του υποσυστήματος διαχείρισης ΒΔ.
- Βασική *μετρική απόδοσης* : ρυθμαπόδοση μετρημένη σε TPS (διεργασίες ανά δευτερόλεπτο), υπό τον περιορισμό ότι το 90% των διεργασιών έχουν χρόνο απόκρισης το πολύ 2 sec
- www.tpc.org : *TPC-C*, version 5.11.0, *TPC-E*, version 1.14.0
- Νέα benchmarks: *TPCx-BB* και *TPCx-HS* για Hadoop based Big Data systems, τα *TPC-DS* και *TPC-H* για decision support περιβάλλοντα, το *TPCx-IoT* για IoT gateways, *TPCx-AI,* κ.α.,

SPEC (1)

- Το SPEC (Standard Performance Evaluation Corporation) δημιουργήθηκε, επίσης, το 1988 από εξέχουσες βιομηχανίες παραγωγής υπολογιστών, με σκοπό να αναπτύξει ένα πρότυπο σύνολο προγραμμάτων σύγκρισης
- Τα αρχικά προγράμματα σύγκρισης SPEC, χρησιμοποιούσαν κυρίως τη CPU, τη μονάδα κινητής υποδιαστολής και λιγότερο το υποσύστημα μνήμης.
- Χρησιμοποιήθηκαν αρχικά για τη σύγκριση των ταχυτήτων διαφορετικών CPU.

SPEC (2)

- Η έκδοση 1.0 του SPEC benchmark suite (1990), αποτελείται από 10 προγράμματα σύγκρισης που αντιπροσωπεύουν διάφορες επιστημονικές και τεχνολογικές εφαρμογές:
  1. ***GCC***: Μετρά το χρόνο που χρειάζεται ο μεταγλωττιστής GNU C για να μετατρέψει 19 προεπεξεργασμένα αρχεία πηγαίου κώδικα σε γλώσσα assembly. Το πρόγραμμα αυτό σύγκρισης, είναι αντιπροσωπευτικό ενός περιβάλλοντος τεχνολογίας λογισμικού εκτιμώντας την αποδοτικότητα της μεταγλώττισης σε ένα σύστημα.
  2. ***Espresso***: Είναι ένα εργαλείο αυτοματισμού ηλεκτρονικού σχεδιασμού (EDA – Electronic Design Automation), το οποίο κάνει ελαχιστοποίηση Boolean συναρτήσεων για προγραμματιζόμενους λογικούς πίνακες (PLA – Programmable Logic Arrays). Μετρά το χρόνο που απαιτείται για την εκτέλεση ενός συνόλου επτά μοντέλων εισόδου.
  3. ***Spice 2g6***: Ένας ακόμα εκπρόσωπος του EDA, το Spice 2g6 είναι ένα εργαλείο προσομοίωσης αναλογικών κυκλωμάτων. Μετρά το χρόνο προσομοίωσης ενός διπολικού κυκλώματος.
  4. ***Doduc***: Εκτελεί μία προσομοίωση Monte Carlo για συγκεκριμένες λειτουργίες πυρηνικού αντιδραστήρα. Δοκιμάζει ιδιαίτερα την αποτελεσματικότητα της cache μνήμης.
  5. ***NASA7***: Συλλογή από επτά προγράμματα πυρήνα που χρησιμοποιούν αριθμητική κινητής υποδιαστολής, τα οποία εκτελούν λειτουργίες πινάκων με δεδομένα διπλής ακρίβειας.

SPEC (3)

6. ***LI***: Μετρά το χρόνο επίλυσης του προβλήματος των 9 βασιλισσών με τη χρήση του διερμηνευτή της γλώσσας LISP.
7. ***Eqntott***: Μεταφράζει μία λογική αναπαράσταση μιας Boolean εξίσωσης, σε ένα πίνακα αληθείας.
8. ***Matrix300***: Εκτελεί διάφορες λειτουργίες πινάκων με τη χρήση ρουτινών LINPACK, σε πίνακες μεγέθους 300 x 300. Χρησιμοποιεί αριθμητική κινητής υποδιαστολής διπλής ακρίβειας.
9. ***Fpppp***: Είναι ένα πρόγραμμα σύγκρισης από την περιοχή της κβαντικής φυσικής, το οποίο είναι γραμμένο σε FORTRAN και χρησιμοποιεί αριθμητική κινητής υποδιαστολής διπλής ακρίβειας.
10. ***Tomcatv***: Είναι ένα πρόγραμμα παραγωγής πλεγμάτων με τα ίδια κατασκευαστικά χαρακτηριστικά όπως το Fpppp.

SPEC (4)

- Τώρα, είναι το *SPEC CPU2017*, που αποτελείται από 43 προγράμματα και μετρά:
  - CPU
  - Memory architecture
  - Compilers

  SPEC CPU subcommittee members: *AMD, ARM, Dell, Fujitsu, HPE, IBM, Inspur, Intel, Nvidia, Oracle*
- www.spec.org : Και άλλα benchmarks:
  - SPECworkstation 3.1 για Workstations (2018, *δωρεάν*)
  - SPEC Cloud_IaaS 2018 για Cloud Performance
  - SPECstorage Solution 2020 για file servers
  - SPECpower_ssj2008 για ενεργειακά χαρακτηριστικά
  - ….

SPEC (5)

ΜΕΤΡΙΚΕΣ

- **SPECthruput:** ο λόγος του χρόνου που απαιτεί η εκτέλεση *κάθε προγράμματος* στο υπό μελέτη σύστημα, προς το χρόνο που απαιτεί η εκτέλεση του ίδιου προγράμματος σε ένα *σύστημα αναφοράς*

  (στο *SPEC CPU2017* είναι το Sun Fire V490 with 2100 MHz UltraSPARC-IV+ chips)

SPEC (6)

- **SPECmark** : η *συνολική απόδοση* του συστήματος.

  Ορίζεται ως ο *Γεωμετρικός Μέσος Όρος* (ΓΜΟ) των SPECthruput των *n* προγραμμάτων σύγκρισης.

  Δηλαδή:

  SPECmark = $\sqrt[n]{\alpha_1 \cdot \alpha_2 \cdots \alpha_n}$

όπου α_i είναι το SPECthruput του προγράμματος i.

ΓΜΟ:

- μειώνει την επίδραση των ακραίων τιμών
- ≤ Αριθμητικού Μέσου Όρου
- = Αριθμητικού Μέσου Όρου όταν τα α_i είναι ίσα

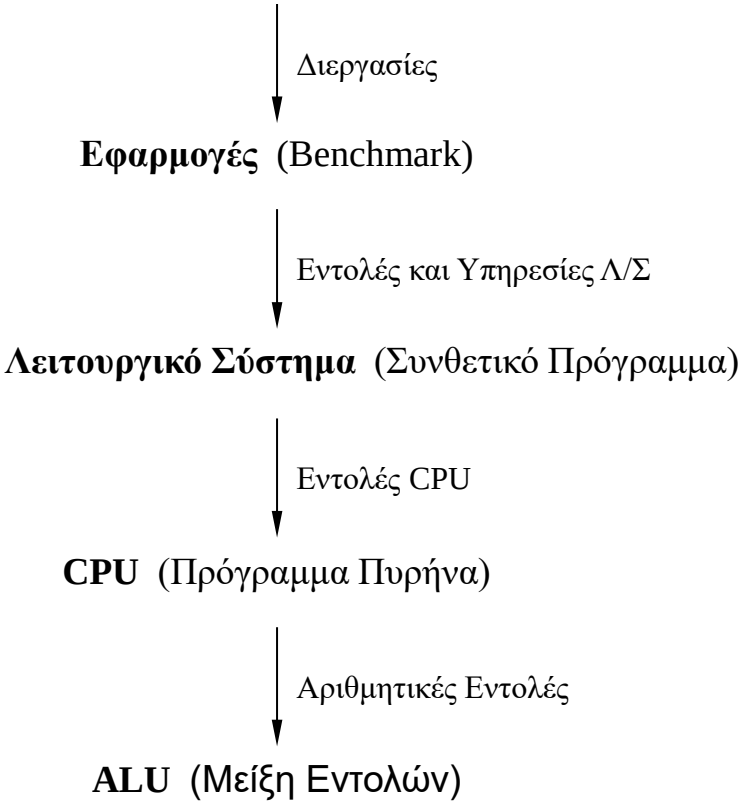
Επιλογή φορτίου εργασίας

- Βασικοί παράγοντες
  1. Υπηρεσίες που προσφέρει το υπό μελέτη σύστημα

     Καθορισμός στοιχείων μελέτης και σημαντικότερων υπηρεσιών που θα πρέπει να χρησιμοποιεί το φορτίο εργασίας
  2. Επίπεδο λεπτομέρειας στην περιγραφή των απαιτήσεων των χρηστών
  3. Αντιπροσωπευτικότητα του φορτίου εργασίας
  4. Παρακολούθηση αλλαγών στη συμπεριφορά των χρηστών με την εξέλιξη του χρόνου

1. Υπηρεσίες που προσφέρει το υπό μελέτη σύστημα

Παράδειγμα ιεραρχικής άποψης Συστήματος Επεξεργασίας Διεργασιών



2. Επίπεδο λεπτομέρειας

Πιθανές λύσεις:

- ***Περισσότερο χρησιμοποιούμενη αίτηση*:** Συχνά χρησιμοποιείται ως αρχικό φορτίο εργασίας.
  - Χρήσιμη όταν ο τύπος αυτός αιτήσεων κυριαρχεί σε σχέση με τις υπόλοιπες, ή καταναλώνει το μεγαλύτερο μέρος των πόρων του συστήματος.
- ***Συχνότητα τύπων αιτήσεων*:** Καταγραφή διαφόρων αιτήσεων, των χαρακτηριστικών τους και των συχνοτήτων εμφάνισής τους.
- ***Ακολουθίες αιτήσεων*:** Ακολουθία πραγματικών αιτήσεων μαζί με τους χρόνους υποβολής τους, την οποία ονομάζουμε *ίχνος* (trace).
  - Πολύ μεγάλο και συχνά ανεπιθύμητο επίπεδο λεπτομέρειας.
- ***Μέση απαίτηση χρήσης πόρων*:** Χρησιμοποιείται στις αναλυτικές τεχνικές
- ***Κατανομή απαιτήσεων χρήσης πόρων*:** Σε περιπτώσεις που έχουμε μεγάλη διασπορά στις απαιτήσεις, ή δεν έχουμε μια γνωστή θεωρητική κατανομή.

3. Αντιπροσωπευτικότητα

Το φορτίο εργασίας θα πρέπει να αντιπροσωπεύει τα παρακάτω χαρακτηριστικά της εφαρμογής

1. ***Ρυθμός Αφίξεων***
2. ***Απαιτήσεις Χρήσης Πόρων***
3. ***Τρόπος Χρήσης Πόρων*****:** Σχετίζεται με την ακολουθία και το μέγεθος χρήσης των διαφορετικών πόρων του συστήματος από μια εφαρμογή.

4. Παρακολούθηση αλλαγών

- Οι χρήστες αλλάζουν συμπεριφορά ανάλογα με τις υπηρεσίες που τους προσφέρονται από τα νέα συστήματα και έτσι τα φορτία εργασίας γίνονται πολλές φορές άχρηστα μόλις γίνουν κατανοητά.
- Οι χρήστες έχουν την τάση να προσαρμόζουν τις απαιτήσεις τους έτσι ώστε να βελτιστοποιούν την απόκριση από το σύστημα, προσανατολιζόμενοι στις υπηρεσίες που το σύστημα προσφέρει πιο αποδοτικά

Χαρακτηρισμός φορτίου εργασίας

Η διαδικασία μελέτης του πραγματικού περιβάλλοντος των χρηστών, παρατήρησης των βασικών χαρακτηριστικών του και κατασκευής ενός φορτίου εργασίας που μπορεί να χρησιμοποιηθεί πολλές φορές.

- *Χρήστες*: άνθρωποι, προγράμματα, σταθμοί εργασίας (ανάλογα με το σύστημα)
- *Παράμετροι φορτίου εργασίας*: ποσότητες που μετρώνται για να χαρακτηρίσουν το φορτίο εργασίας(αιτήσεις εξυπηρέτησης, απαιτήσεις πόρων συστήματος), όπως: τύποι διεργασιών, μεγέθη πακέτων, ρυθμός αφίξεων

Τεχνικές χαρακτηρισμού παραμέτρων φορτίου εργασίας (1)

1. Μέση τιμή $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ αν έχουμε παρατηρήσει *n* τιμές $\{x_1, x_2, \ldots, x_n\}$ μιας παραμέτρου

2. Διακύμανση $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$

3. Ιστογράμματα
   - Παρουσιάζουν τις σχετικές συχνότητες των διαφόρων τιμών μιας παραμέτρου
   - Και σε μορφή πίνακα

Τεχνικές χαρακτηρισμού παραμέτρων φορτίου εργασίας (2)

4. Ακολουθίες Markov

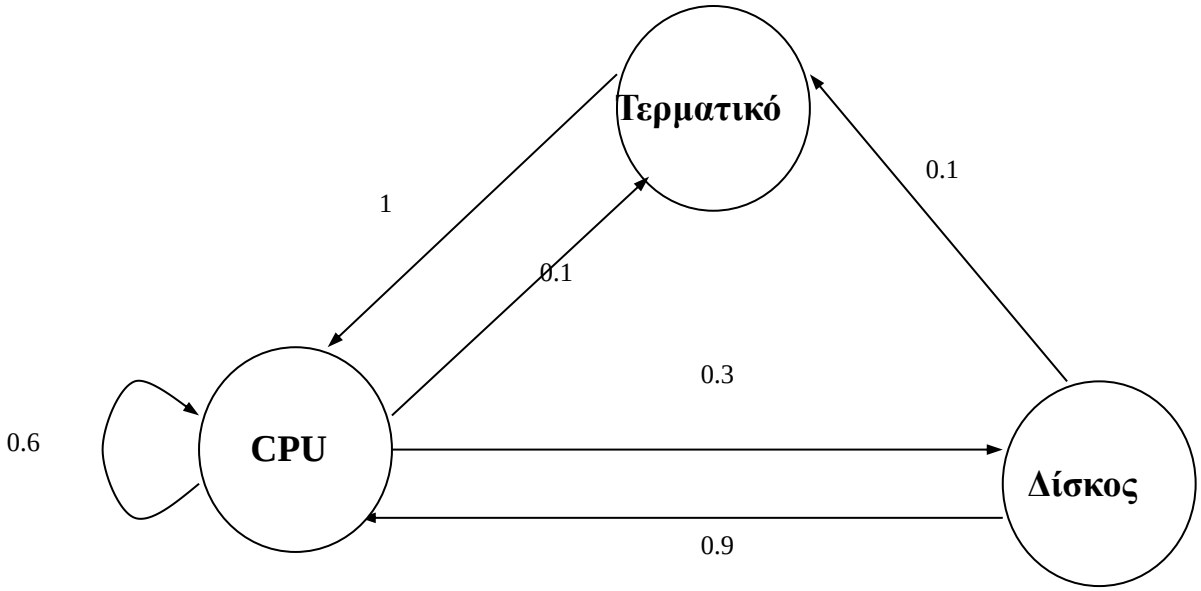
Περιγράφονται από ένα *πίνακα μεταβάσεων* ο οποίος περιέχει τις πιθανότητες να έχουμε επόμενη αίτηση ενός τύπου, δεδομένου του τύπου της τελευταίας αίτησης.

Παράδειγμα:

Από/Προς	CPU	Δίσκος	Τερματικό
CPU	0.6	0.3	0.1
Δίσκος	0.9	0	0.1
Τερματικό	1	0	0

Τεχνικές χαρακτηρισμού παραμέτρων φορτίου εργασίας (3)

- Αντίστοιχο *Διάγραμμα Καταστάσεων – Πιθανοτήτων Μεταβάσεων*

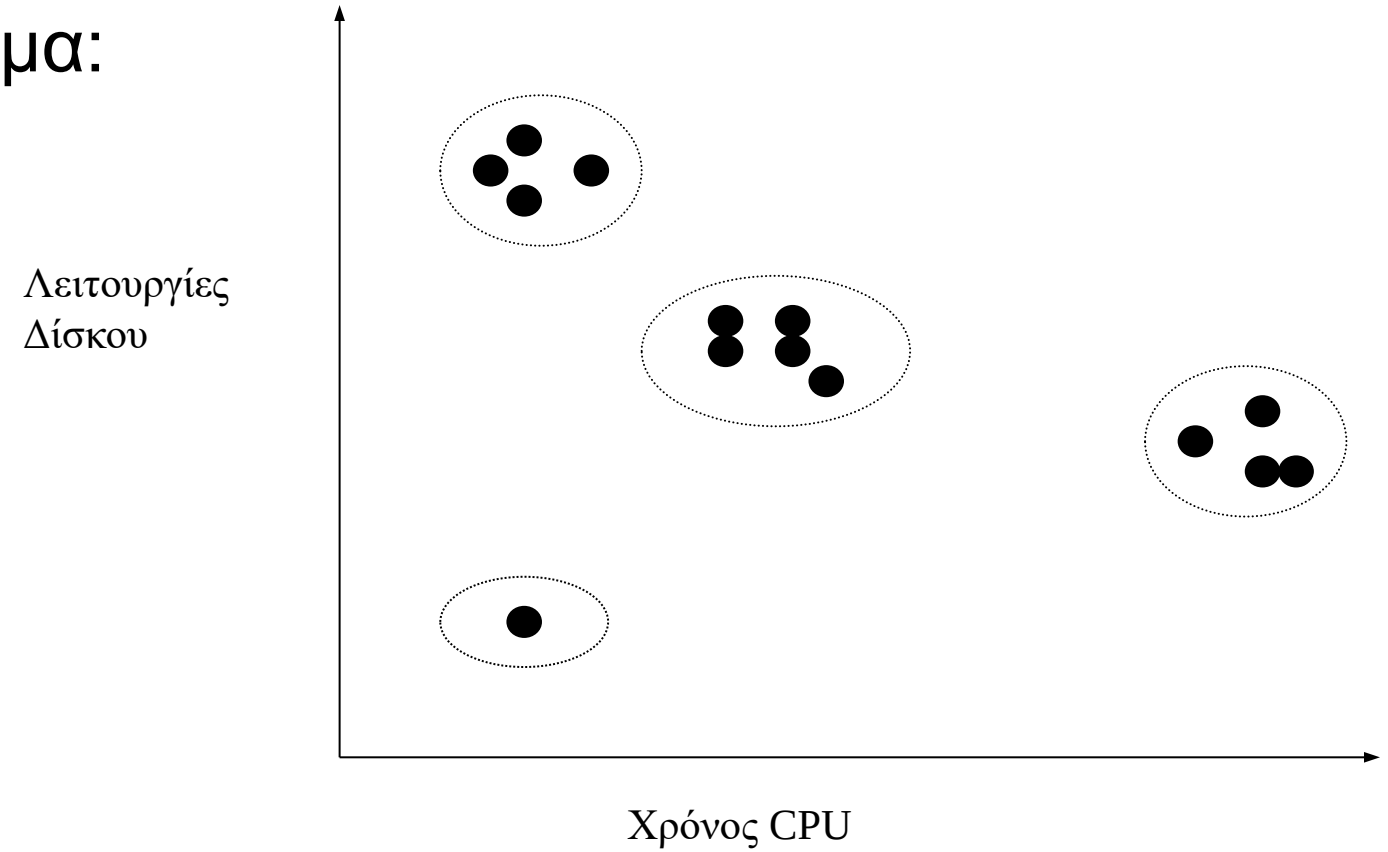


Τεχνικές χαρακτηρισμού παραμέτρων φορτίου εργασίας (4)

5. Ομαδοποίηση
   - Ομαδοποίηση παρόμοιων στοιχείων σε ομάδες. Ένα μέλος κάθε ομάδας μπορεί να επιλεγεί ώστε να αντιπροσωπεύει την ομάδα στη μελέτη ως φορτίο εργασίας

Τεχνικές χαρακτηρισμού παραμέτρων φορτίου εργασίας (5)

Παράδειγμα:



Ελεγκτές (Monitors)

- Εργαλεία που χρησιμοποιούνται για τη μέτρηση του επιπέδου δραστηριότητας σε ένα πληροφοριακό σύστημα.
- Πρέπει να κατασκευάζονται και να χρησιμοποιούνται με τρόπο ώστε να μην επηρεάζουν τη λειτουργία του συστήματος
- Δύο χαρακτηριστικά:
  - Τύπος
  - Τρόπος Λειτουργίας

Τύποι ελεγκτών (1)

- Υλικοί Ελεγκτές
  - Εργαλεία μέτρησης που εντοπίζουν γεγονότα μέσω της παρακολούθησης προκαθορισμένων σημάτων
  - Παρακολουθούν την κατάσταση των τμημάτων υλικού, όπως οι motherboards, καταχωρητές, οι θέσεις μνήμης και τα κανάλια εισόδου/εξόδου.
  - Σήμερα κυρίως για Μπαταρίες, Ανεμιστήρες, Θερμοκρασία.
  - Παραδείγματα: *HWMonitor*, *SpeedFan*

Τύποι ελεγκτών (2)

- Λογισμικοί
  - Αποτελούνται από ρουτίνες που έχουν εισαχθεί στο λογισμικό ενός πληροφοριακού συστήματος.
  - Ενεργοποιούνται με την εμφάνιση συγκεκριμένων γεγονότων, ή σε συγκεκριμένες χρονικές στιγμές.
  - Στην κατηγορία αυτή:
    - Accounting Systems (billing)
    - Program Analyzers
    - Log systems
  - Παραδείγματα: *ipMonitor, AlertBot, MS Performance Monitor*

Τύποι ελεγκτών (3)

- Υβριδικοί
  - Συνδυασμοί υλικών και λογισμικών ελεγκτών
  - *Ρουτίνες Λογισμικού*: Υπεύθυνες για τον εντοπισμό γεγονότων και μεταβίβαση της πληροφορίας σε ειδικούς καταχωρητές.
  - *Υλικό*: Υπεύθυνο για την καταγραφή των δεδομένων, αποφεύγοντας την επιβάρυνση των λειτουργιών I/O του συστήματος.
  - Χρήσιμοι, αν είναι ενσωματωμένοι στο σύστημα
  - Σήμερα, οι Υλικοί είναι ουσιαστικά Υβριδικοί

Πλεονεκτήματα – Μειονεκτήματα ελεγκτών

	+	-
Υλικοί	■ Δεν καταναλώνουν πόρους του συστήματος ■ Είναι διαθέσιμοι για χρήση σε περισσότερα από ένα συστήματα	■ Δεν έχουν φιλικότητα στη χρήση τους ■ Είναι δύσκολος ο σχεδιασμός πειραμάτων βασισμένων σε υλικούς ελεγκτές.
Λογισμικοί	Ευκολία εγκατάστασης και χρήσης	Επιβάρυνση στη λειτουργία του συστήματος και πιθανή εξάρτηση από ένα λειτουργικό σύστημα
Υβριδικοί	Συνδυάζουν τα καλύτερα χαρακτηριστικά των υλικών και λογισμικών	Απαίτηση εξειδικευμένου υλικού

Τρόποι λειτουργίας ελεγκτών

- **Καταγραφή γεγονότων**

Συλλογή πληροφοριών από το σύστημα τις χρονικές στιγμές εμφάνισης συγκεκριμένων γεγονότων

  - Όταν ο ρυθμός εμφάνισης γεγονότων γίνει πολύ μεγάλος, οι ρουτίνες του ελεγκτή εκτελούνται πολύ συχνά και η επιβάρυνση στο σύστημα φτάνει σε ανεπιθύμητα επίπεδα (ανεκτό έως 5%)

- **Δειγματοληψία**

Συλλογή πληροφοριών για το σύστημα σε συγκεκριμένες χρονικές στιγμές. Η επιβάρυνση εξαρτάται από:

  - τον αριθμό των μεταβλητών που μετρώνται και
  - το μέγεθος του διαστήματος δειγματοληψίας.

Δοσοληψία: Επιβάρυνση **vs** Ακρίβεια Μετρήσεων

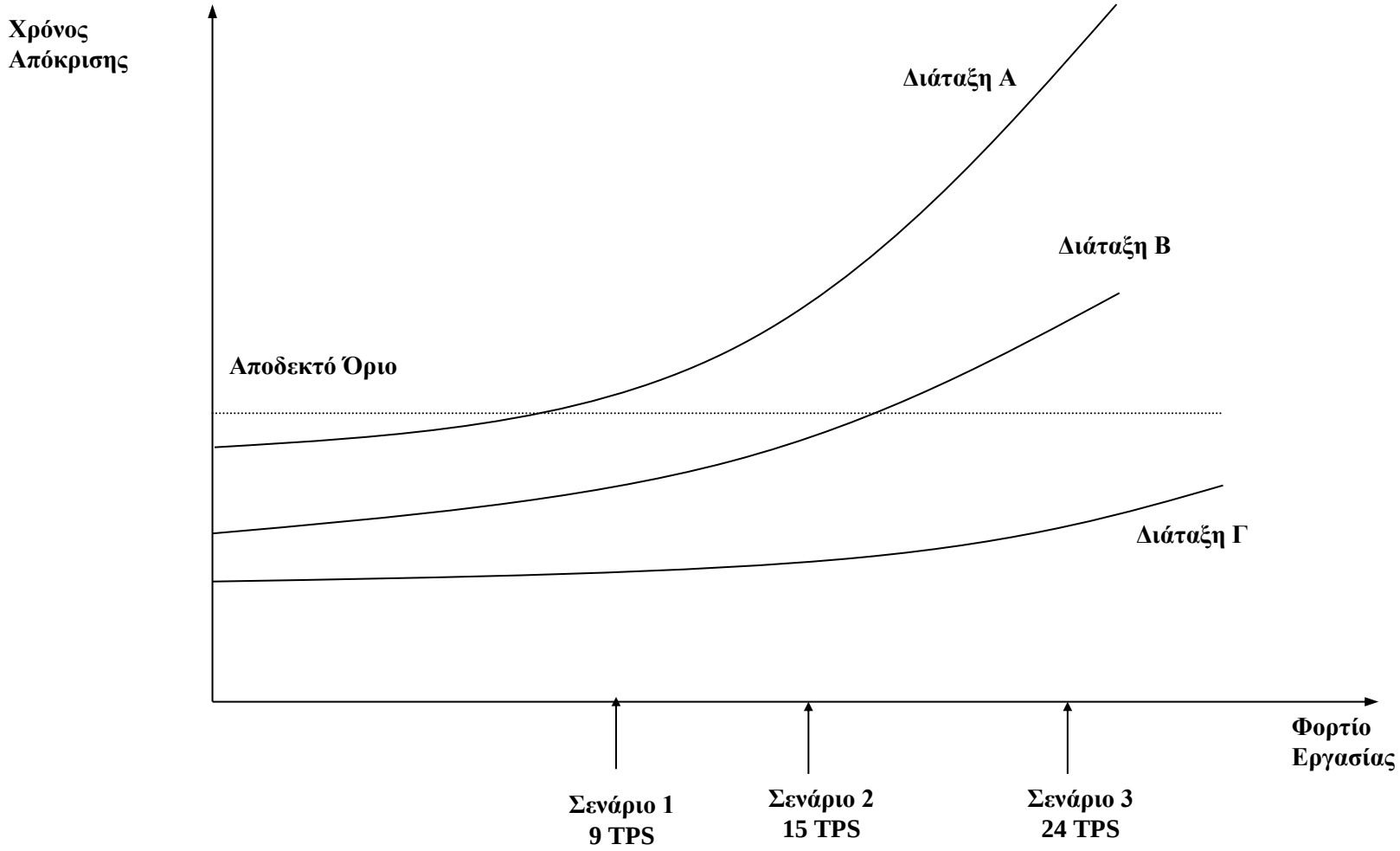
Σχεδιασμός και διαχείριση χωρητικότητας

- Ο **σχεδιασμός χωρητικότητας** (capacity planning) διασφαλίζει ότι θα υπάρχουν διαθέσιμοι αρκετοί πόροι, ώστε να εξυπηρετούνται με αποδοτικό τρόπο τα μελλοντικά φορτία εργασίας, ικανοποιώντας παράλληλα τις απαιτήσεις απόδοσης του συστήματος.
- Η **διαχείριση χωρητικότητας** (capacity management) αφορά τη μεγιστοποίηση της απόδοσης του υπάρχοντος συστήματος.

Βήματα στο σχεδιασμό και τη διαχείριση χωρητικότητας

1. Προετοιμασία του συστήματος για μετρήσεις.
2. Καταγραφή της χρήσης του συστήματος.
3. Χαρακτηρισμός του φορτίου εργασίας.
4. Εκτίμηση της απόδοσης για διαφορετικές εναλλακτικές επιλογές.
5. Υιοθέτηση της επιλογής με το μικρότερο λόγο κόστους/απόδοσης.

Παράδειγμα Σχεδιασμού Χωρητικότητας



Αποτελέσματα εκτίμησης της απόδοσης συστήματος επεξεργασίας διεργασιών.

Σχεδιασμός πειραμάτων (1)

- Ασχολείται με τον προσδιορισμό του **αριθμού** των διαφορετικών πειραμάτων, των συνδυασμών των **επιπέδων** των παραγόντων για κάθε πείραμα και των αριθμών **επαναλήψεων** για κάθε πείραμα.
- Παράδειγμα (Σχεδιασμός νέου σταθμού εργασίας):

  Παράγοντες με τα επίπεδά τους:

  1. *Τύπος CPU*: Pentium III, UltraSPARC III, R12000 RISC
  2. *Μέγεθος Μνήμης*: 16MB, 32MB, 64MB
  3. *Αριθμός Δίσκων*: 1, 2, 3, 4
  4. *Φορτίο Εργασίας*: Γραμματειακό, Διοικητικό, Επιστημονικό
  5. *Επίπεδο Χρηστών*: Λύκειο, ΑΕΙ, Μετ/κοί-Διδάκτορες

Σχεδιασμός πειραμάτων (2)

- Για δύο παράγοντες Α και Β λέμε ότι **αλληλεπιδρούν**, αν η επίδραση ενός από τους δύο εξαρτάται από το επίπεδο του άλλου.
- Στο παρακάτω παράδειγμα φαίνεται η απόδοση ενός συστήματος που εξαρτάται από 2 παράγοντες Α και Β:

Α. Οι παράγοντες δεν αλληλεπιδρούν			**Β. Οι παράγοντες αλληλεπιδρούν**	
	A_1	A_2	A_1	A_2
B_1	3	5	3	5
B_2	6	8	6	9

Πίνακας 2.3. Απόδοση συστήματος με και χωρίς αλληλεπίδραση των παραγόντων.

Μέθοδοι σχεδιασμού πειραμάτων (2)

- **Πλήρως Παραγοντικός Σχεδιασμός**

  Αξιοποιεί όλους τους δυνατούς συνδυασμούς επιπέδων των παραγόντων.

$$n = \prod_{i=1}^{k} n_i$$

- Παράδειγμα:

  n = (3 CPU) x (3 επίπεδα μνήμης) x (4 οδηγοί δίσκων) x (3 φορτία εργασίας) x (3 επίπεδα χρηστών) =

  = 324 πειράματα

Μέθοδοι σχεδιασμού πειραμάτων (1)

- **Απλός Σχεδιασμός**

  Αρχίζουμε με μια τυπική διάταξη του συστήματος και αλλάζουμε την τιμή ενός παράγοντα τη φορά, για να δούμε πως ο παράγοντας αυτός επηρεάζει την απόδοση.

  Με *k* παράγοντες, με τον *i*-οστό παράγοντα να έχει n_i επίπεδα, ο απλός σχεδιασμός απαιτεί *n* πειράματα:

$$n = 1 + \sum_{i=1}^{k} (n_i - 1)$$

  Παράδειγμα: Έστω μια τυπική διάταξη: R12000, 2 δίσκοι, 32 MB μνήμη, διοικητικό, AEI. Τότε:

  n = 1+2+2+3+2+2 = 12

  - Ο απλός σχεδιασμός δεν είναι αποδοτικός στατιστικά, ενώ δεν αντιμετωπίζει ικανοποιητικά την τυχόν αλληλεπίδραση παραγόντων

Μέθοδοι σχεδιασμού πειραμάτων (3)

- **Κλασματικός Παραγοντικός Σχεδιασμός**

  Χρησιμοποιούμε ένα κλάσμα μόνο του πλήρως παραγοντικού σχεδιασμού.

  Εναλλακτικά μπορούμε (ή και σε συνδυασμό):

  - Να μειώσουμε τον αριθμό των επιπέδων παραγόντων
  - Να μειώσουμε τον αριθμό των παραγόντων

Μέθοδοι σχεδιασμού πειραμάτων (4)

- Παράδειγμα (αγνοούμε τον παράγοντα «Αριθμός Δίσκων»)

Αριθμός Πειράματος	CPU	Επίπεδο Μνήμης	Τύπος Φορτίου Εργασίας	Επίπεδο Εκπαίδευσης
1	SPARC	16M	Διοικητικό	Λύκειο
2	SPARC	32M	Επιστημονικό	Μεταπτυχιακό
3	SPARC	64M	Γραμματειακό	ΑΕΙ – ΤΕΙ
4	R12000	16M	Επιστημονικό	ΑΕΙ – ΤΕΙ
5	R12000	32M	Γραμματειακό	Λύκειο
6	R12000	64M	Διοικητικό	Μεταπτυχιακό
7	Pentium	16M	Γραμματειακό	Μεταπτυχιακό
8	Pentium	32M	Διοικητικό	ΑΕΙ – ΤΕΙ
9	Pentium	64M	Επιστημονικό	Λύκειο