



Ενότητα 1^η: Η απόδοση ενός υπολογιστικού συστήματος

Σκοπός Ο σκοπός της ενότητας αυτής είναι να παρουσιάσει τον τρόπο με τον οποίο υπολογίζεται η απόδοση ενός υπολογιστικού συστήματος και συγκρίνονται οι αποδόσεις διαφορετικών υπολογιστών.

Προσδοκώμενα Αποτελέσματα Όταν θα έχετε μελετήσει την ενότητα, θα είστε σε θέση να:



συγκρίνετε την απόδοση δύο ή περισσότερων υπολογιστικών συστημάτων,



διαφοροποιείτε την απόδοση ενός υπολογιστή ανάλογα με το κριτήριο μέτρησης,



επιλέγετε προγράμματα για την αξιολόγηση της απόδοσης ενός υπολογιστή,



ορίζετε τον συνολικό χρόνο εκτέλεσης και τον σταθμισμένο χρόνο εκτέλεσης,



προσεγγίζετε την απόδοση με τον κανονικοποιημένο χρόνο εκτέλεσης,



απαριθμείτε τα υπέρ και τα κατά του Γεωμετρικού Μέσου,



απαριθμείτε και διαχωρίζετε τις πιο συνηθισμένες "πλάνες και παγίδες".



χρόνος απόκρισης, χρόνος εκτέλεσης, ρυθμαπόδοση, απόδοση, χρόνος αναμονής, χρόνος χρήσης της ΚΜΕ, χρόνος συστήματος ΚΜΕ, πραγματικά προγράμματα, πυρήνες, προγράμματα δοκιμής παιχνιδιών, σύνθετα προγράμματα δοκιμής, αριθμητικός μέσος, αρμονικός μέσος, σταθμισμένος αριθμητικός μέσος, σταθμισμένος αρμονικός μέσος, κανονικοποιημένος χρόνος εκτέλεσης, MIPS, MFLOPS



Μέτρηση και καταγραφή της απόδοσης

Στην ερώτηση «Πότε ένας υπολογιστής είναι “ταχύτερος” από έναν άλλο;», η απάντηση διαφοροποιείται ανάλογα με την ιδιότητα του ερωτούμενου. Ένας χρήστης H/Y ίσως απαντήσει : “όταν ο H/Y τρέχει ένα πρόγραμμα σε λιγότερο χρόνο”, ενώ ένας διαχειριστής συστήματος “όταν ολοκληρώνει περισσότερες εργασίες σε μία ώρα”. Ένας χρήστης ενδιαφέρεται να μειώσει το χρόνο απόκρισης ή χρόνο εκτέλεσης, ενώ ένας διαχειριστής μεγάλου κέντρου επεξεργασίας δεδομένων, επιδιώκει να αυξήσει τη ρυθμαπόδοση.



Ο χρόνος εκτέλεσης αφορά το χρονικό διάστημα που μεσολαβεί από την αρχή ως την ολοκλήρωση μιας εργασίας.



Η ρυθμαπόδοση (throughput) είναι το σύνολο δουλειάς που γίνεται σε συγκεκριμένο χρόνο.

Η **απόδοση** ενός υπολογιστικού συστήματος αναφέρεται στις επιδόσεις του, δηλαδή στα ιδιαίτερα χαρακτηριστικά και τις δυνατότητες που διαθέτει το σύστημα. Αντιστοιχεί κυρίως στην ταχύτητα, στην αξιοπιστία και στη λειτουργική ευελιξία του υπολογιστικού συστήματος.

Για τη σύγκριση δύο διαφορετικών υπολογιστικών συστημάτων, σχετίζουμε την απόδοση δύο διαφορετικών υπολογιστών, έστω X και Y. “Ο X είναι ταχύτερος από τον Y” σημαίνει ότι ο χρόνος απόκρισης ή εκτέλεσης στον X είναι μικρότερος από ότι στον Y, για μια συγκεκριμένη εργασία. Πιο συγκεκριμένα, “ο X είναι n φορές ταχύτερος από τον Y” σημαίνει ότι η απόδοση ισούται με (τύπος 1.1) :

$$n = \frac{(\text{χρόνος εκτέλεσης})_Y}{(\text{χρόνος εκτέλεσης})_X} \quad (\text{Τύπος 1.1})$$

Δεδομένου ότι ο χρόνος εκτέλεσης είναι αντιστρόφως ανάλογος της απόδοσης, η παραπάνω σχέση μετατρέπεται ως εξής :

$$n = \frac{\frac{1}{(\text{απόδοση})_Y}}{\frac{1}{(\text{απόδοση})_X}} = \frac{(\text{απόδοση})_X}{(\text{απόδοση})_Y} \quad (\text{Τύπος 1.2})$$



Σύμφωνα με τους τύπους 1.1 και 1.2, αυξάνοντας την απόδοση μειώνεται ο χρόνος εκτέλεσης και αντίστροφα.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1

Τι σημαίνει η φράση «Η ρυθμαπόδοση του υπολογιστή X είναι 1,3 φορές μεγαλύτερη από αυτή του υπολογιστή Y»;

Να επιλέξετε μία από τις παρακάτω απαντήσεις και να δικαιολογήσετε την απάντησή σας.

α) Ο χρόνος εκτέλεσης μίας εργασίας από τον X είναι 1,3 φορές μικρότερος από το χρόνο εκτέλεσης στον Y.

β) Ο αριθμός των εργασιών που εκτελούνται στον X, είναι 1,3 φορές περισσότερες από τις εργασίες που εκτελούνται στον Y, στη μονάδα του χρόνου.

γ) Ο X είναι ταχύτερος από τον Y κατά 1,3 φορές.



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 1

Η σωστή απάντηση είναι η (β) σύμφωνα με τον ορισμό της ρυθμαπόδοσης: η ρυθμαπόδοση είναι το σύνολο δουλειάς που γίνεται σε συγκεκριμένο χρόνο.

Τελικά μπορούμε να πούμε ότι είτε μας ενδιαφέρει η ρυθμαπόδοση είτε ο χρόνος απόκρισης το μέτρο κλειδί είναι ο χρόνος. Ο υπολογιστής που φέρει εις πέρας τον ίδιο φόρτο εργασίας σε λιγότερο χρόνο είναι ο ταχύτερος. Η διαφορά έγκειται στο αν μετράμε μια εργασία (χρόνος απόκρισης) ή πολλές (ρυθμαπόδοση).



Το μοναδικό αξιόπιστο και αμετάβλητο μέτρο της απόδοσης ενός υπολογιστικού συστήματος είναι ο χρόνος εκτέλεσης των πραγματικών προγραμμάτων, στα οποία έχουμε άμεση και χωρίς καθυστέρηση επεξεργασία δεδομένων. Οι όποιες άλλες εναλλακτικές επιλογές οδηγούν σε παρανοήσεις και λάθη.



Μέτρα του χρόνου εκτέλεσης

Ο **χρόνος εκτέλεσης** μπορεί να ορισθεί με διαφορετικούς τρόπους ανάλογα με το τι μετράμε. Ο πιο ορθός ορισμός του χρόνου είναι ο χρόνος “ρολογιού” ή χρόνος απόκρισης ή διανυθείς χρόνος, δηλαδή το άθροισμα του **χρόνου αναμονής** για την ολοκλήρωση μιας εργασίας, μαζί με τις προσβάσεις στο δίσκο, στη μνήμη, τις διεργασίες εισόδου/εξόδου, τις εντολές λειτουργικού συστήματος, κλπ.



Στον πολυπρογραμματισμό, η ΚΜΕ ενώ περιμένει για την εκτέλεση διεργασιών εισόδου-εξόδου, χωρίς απαραίτητα να περιορίζει τον διανυθέντα χρόνο ενός προγράμματος, δουλεύει σε κάποιο άλλο πρόγραμμα.

Άλλη μια έννοια που πρέπει να διασαφηνίσουμε είναι ο **χρόνος της ΚΜΕ**, που περιλαμβάνει το χρόνο που αυτή υπολογίζει, μη περιλαμβανομένου του χρόνου που περιμένει για είσοδο / έξοδο ή του χρόνου που τρέχει άλλα προγράμματα. Ο χρόνος της ΚΜΕ διαχωρίζεται στο **χρόνο χρήσης της ΚΜΕ**, αναφερόμενοι στο χρόνο που ξοδεύεται σε ένα πρόγραμμα και στο **χρόνο συστήματος ΚΜΕ**, δηλαδή το χρόνο που το λειτουργικό σύστημα πραγματοποιεί εργασίες κατόπιν αιτήσεως του προγράμματος.

Παραλείποντας περαιτέρω λεπτομέρειες σημειώνουμε ότι γίνεται διάκριση στην απόδοση που προκύπτει με τον χρόνο απόκρισης (elapsed time) και τον χρόνο της ΚΜΕ. Ο όρος απόδοση του συστήματος αναφέρεται στον διανυθέντα χρόνο σε ένα σύστημα που δεν είναι φορτωμένο, ενώ η απόδοση της ΚΜΕ, αναφέρεται στο χρόνο χρήσης της ΚΜΕ σε ένα σύστημα που δεν είναι φορτωμένο. Θα ασχοληθούμε με την 1^η περίπτωση.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2

Ποια είναι η διαφορά ανάμεσα στο χρόνο εκτέλεσης και το χρόνο της ΚΜΕ; Στην περίπτωση που δυσκολεύεστε να απαντήσετε στην ερώτηση αυτή, κρίνεται σκόπιμο να επαναλάβετε την ανάγνωση της προηγούμενης παραγράφου: «Μέτρα του χρόνου εκτέλεσης».



Προγράμματα δοκιμής

Ένας χρήστης υπολογιστή που τρέχει καθημερινά τα ίδια προγράμματα θα ήταν ο πιο κατάλληλος για την αξιολόγηση ενός νέου συστήματος, συγκρίνοντας απλώς τον χρόνο εκτέλεσης τού φόρτου εργασίας-σύνολο προγραμμάτων και των εντολών του λειτουργικού συστήματος που εκτελούνται σε ένα σύστημα.

Υπάρχουν 4 διαφορετικά επίπεδα προγραμμάτων που χρησιμοποιούνται για την εκτίμηση της απόδοσης ενός υπολογιστή, τα οποία παρατίθενται με σειρά φθίνουσας ακρίβειας πρόβλεψης.

- ♦ **Πραγματικά προγράμματα:** χρησιμοποιούνται για την επίλυση πραγματικών προβλημάτων. Μερικά παραδείγματα πραγματικών προγραμμάτων είναι οι μεταγλωττιστές της C, το λογισμικό επεξεργασίας κειμένου, όπως το TeX κλπ. Τα προγράμματα αυτά δέχονται είσοδο, δίνοντας κάποια έξοδο και παρέχουν στους χρήστες τη δυνατότητα επιλογής κατά την εκτέλεση του προγράμματος.
- ♦ **Πυρήνες:** Οι πυρήνες είναι μικρά τμήματα ή “κλειδιά” κώδικα που έχουν εξαχθεί από πραγματικά προγράμματα και χρησιμοποιούνται για τον υπολογισμό της απόδοσης ενός συστήματος. Σε αντίθεση με τα πραγματικά προγράμματα οι πυρήνες δεν εκτελούνται από τους χρήστες, αφού χρησιμοποιούνται μόνο για την εκτίμηση της απόδοσης ξεχωριστών χαρακτηριστικών ενός συστήματος και για να εξηγήσουν τις διαφορές στην απόδοση των προγραμμάτων. Μερικά παραδείγματα πυρήνων είναι το Linpack και τα Livermore loops.
- ♦ **Προγράμματα δοκιμής – παιχνίδια:** είναι συνήθως 10 έως 100 γραμμές κώδικα που παράγουν αποτελέσματα τα οποία είναι ήδη γνωστά στο χρήστη, πριν την εκτέλεσή τους. Αντιπροσωπευτικά είναι “**το κόσκινο του Ερατοσθένη**” και η “**Quicksort**”. Είναι ιδιαίτερα δημοφιλή γιατί είναι μικρά, εύκολα στη γραφή και τρέχουν σχεδόν σε όλους τους υπολογιστές.
- ♦ **Σύνθετα προγράμματα δοκιμής:** όπως και οι πυρήνες, τα προγράμματα αυτά προσπαθούν να εξομοιώσουν την μέση συχνότητα των λειτουργιών και τελεστών ενός μεγάλου συνόλου από προγράμματα. Τα πιο γνωστά από αυτά είναι το **Whetstone** και **Dhrystone**. Οι χρήστες αποφεύγουν συνήθως την εκτέλεσή τους, αφού δεν τους προσφέρουν τίποτα χρήσιμο. Σε αντίθεση με τους πυρήνες, δεν είναι τμήματα κώδικα από πραγματικά προγράμματα, αλλά είναι κώδικας ο οποίος έχει δημιουργηθεί ώστε να ανταποκρίνεται σε ένα μέσο προφίλ εκτέλεσης εντολών.



Επειδή η οικονομική ανάπτυξη και γενικά η καθιέρωση μίας εταιρίας υπολογιστών στην αγορά εξαρτάται από τον λόγο τιμή/απόδοση των προϊόντων της σε σχέση με εκείνο των ανταγωνιστών τους, έχουν βρεθεί τρόποι βελτίωσης της απόδοσης των προγραμμάτων που χρησιμοποιούνται ευρέως για την αξιολόγηση ενός υπολογιστικού συστήματος. Ειδικότερα, γίνονται βελτιώσεις τόσο στο λογισμικό όσο και στο υλικό με σκοπό την αύξηση της απόδοσης των σύνθετων προγραμμάτων δοκιμής,

Κεφάλαιο 1 / Ενότητα 1

των πυρήνων και των πραγματικών προγραμμάτων. Γι' αυτό το λόγο, οι προμηθευτές προγραμμάτων δοκιμής προσδιορίζουν τους κανόνες κάτω από τους οποίους λειτουργούν οι μεταγλωττιστές. Απώτερος σκοπός είναι να βελτιστοποιήσουν το προϊόν τους, ώστε να εμφανίζει μεγαλύτερη ταχύτητα από άλλα παρόμοια προϊόντα, για ένα συγκεκριμένο πρόγραμμα δοκιμής.



Εκτός από τα τέσσερα επίπεδα προγραμμάτων που εφαρμόζονται για τον υπολογισμό της απόδοσης των επεξεργαστών χρησιμοποιούνται και **ακολουθίες από προγράμματα δοκιμής**. Οι ακολουθίες προγραμμάτων δοκιμής πλεονεκτούν έναντι των μεμονωμένων προγραμμάτων δοκιμής, διότι η αδυναμία μίας μεμονωμένης δοκιμής απόδοσης μειώνεται με την παρουσία των υπολοίπων δοκιμών της ακολουθίας. Αυτό συμβαίνει γιατί οι μέθοδοι που χρησιμοποιούνται για τη σύνοψη της απόδοσης της ακολουθίας αναφέρονται στο χρόνο εκτέλεσης ολόκληρης της ακολουθίας. Οι ακολουθίες προγραμμάτων δοκιμής είναι συλλογές προγραμμάτων, μερικά από τα οποία μπορεί να είναι πυρήνες και άλλα να είναι πραγματικά προγράμματα. Ένα τέτοιο παράδειγμα είναι το SPEC92 .



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 3

Να απαριθμήσετε τα διαφορετικά επίπεδα προγραμμάτων που χρησιμοποιούνται για την εκτίμηση της απόδοσης ενός υπολογιστή και να αναφέρετε τουλάχιστον ένα παράδειγμα από τα προγράμματα που χρησιμοποιούνται για κάθε ένα από αυτά. Στην περίπτωση που δυσκολεύεστε να απαντήσετε στην ερώτηση αυτή, κρίνεται σκόπιμο να επαναλάβετε την ανάγνωση της προηγούμενης παραγράφου: «Προγράμματα δοκιμής».



Σύγκριση – αποτελέσματα απόδοσης

Η βασική αρχή που μας καθοδηγεί στην καταγραφή της απόδοσης ενός υπολογιστικού συστήματος, είναι η δυνατότητα αναπαραγωγής των ίδιων αποτελεσμάτων. Για να το πετύχουμε αυτό θα πρέπει να γίνει παράθεση όλων εκείνων των παραγόντων που θα οδηγήσουν στα ίδια αποτελέσματα κατά την επανάληψη ενός πειράματος.

Είναι σύνηθες φαινόμενο η ύπαρξη σημαντικών αποκλίσεων μεταξύ των διαφόρων μετρήσεων της απόδοσης ενός συστήματος. Το γεγονός αυτό οφείλεται στις διαφορετικές υποθέσεις που έχουν γίνει κατά την διάρκεια των μετρήσεων ή στην έλλειψη επαρκών στοιχείων για την εξαγωγή συμπεράσματος. Ακόμα όμως και αν υπάρξει κοινή συμφωνία για τα προγράμματα που θα χρησιμοποιηθούν, για το περιβάλλον στο οποίο θα εκτελεστεί η μέτρηση και για τον ορισμό του πιο γρήγορου συστήματος, η εικόνα που θα πάρουμε εξακολουθεί να είναι ασαφής. Αυτό μπορούμε να το διαπιστώσουμε με την βοήθεια του παρακάτω παραδείγματος.



Παράδειγμα

Ας θεωρήσουμε 2 προγράμματα, τα P1 και P2 τα οποία εκτελούνται σε 3 διαφορετικούς υπολογιστές: A,B,C.

	Computer A	Computer B	Computer C
Program P1	1	10	20
Program P2	1000	100	20
Ολικός χρόνος	1001	110	40

Σχήμα 1.1.1 - Χρόνοι εκτέλεσης 2 προγραμμάτων σε 3 διαφορετικούς υπολογιστές

Χρησιμοποιώντας τον ορισμό του **ταχύτερου υπολογιστή** έχουμε ότι:

- Ο A είναι 10 φορές γρηγορότερος από τον B για την εκτέλεση του P1.
- Ο B είναι 10 φορές γρηγορότερος από τον A για την εκτέλεση του P2.
- Ο A είναι 20 φορές γρηγορότερος από τον C για την εκτέλεση του P1.
- Ο C είναι 50 φορές γρηγορότερος από τον A για την εκτέλεση του P2.
- Ο B είναι 2 φορές γρηγορότερος από τον C για την εκτέλεση του P1.
- Ο C είναι 5 φορές γρηγορότερος από τον B για την εκτέλεση του P2.

Εξετάζοντας μεμονωμένα κάθε μία από τις παραπάνω δηλώσεις εκτιμούμε καλύτερα την απόδοση κάθε υπολογιστή. Όλες μαζί όμως παρουσιάζουν μια μπερδεμένη εικόνα. Η **σχετική απόδοση** καθενός από τους υπολογιστές A,B,C δεν είναι ξεκάθαρη. 🚩



Συνολικός χρόνος εκτέλεσης και Σταθμισμένος χρόνος εκτέλεσης



Συνολικός χρόνος εκτέλεσης: Ένα αξιόπιστο μέτρο

Η πιο απλή προσέγγιση για να συνοψίσουμε την σχετική απόδοση ενός συστήματος είναι να χρησιμοποιήσουμε τον συνολικό χρόνο εκτέλεσης και των 2 προγραμμάτων στον ίδιο υπολογιστή για το προηγούμενο παράδειγμα. Έτσι έχουμε ότι:

Ο Β είναι 9,1 φορές ταχύτερος από τον Α για τα προγράμματα P1 και P2.

Ο C είναι 25 φορές ταχύτερος από τον Α για τα προγράμματα P1 και P2.

Ο C είναι 2,75 φορές ταχύτερος από τον Β για τα προγράμματα P1 και P2.

Για τον υπολογισμό της απόδοσης χρησιμοποιήσαμε ως μέτρο τον συνολικό χρόνο εκτέλεσης. Εάν τα προγράμματα εκτελούνται ίσο αριθμό φορών η παραπάνω δήλωση προβλέπει τους **σχετικούς χρόνους εκτέλεσης** για το φόρτο που εκτελείται σε κάθε υπολογιστή.

Ο μέσος όρος των χρόνων εκτέλεσης που καθορίζει τον συνολικό χρόνο εκτέλεσης είναι ο **αριθμητικός μέσος** (τύπος 1.3):

$$\frac{1}{n} \sum_{i=1}^n \text{χρόνος εκτέλεσης}_i \quad (\text{Τύπος 1.3})$$



Χρόνος εκτέλεσης_i : ο χρόνος εκτέλεσης για το i-οστο πρόγραμμα ενός συνόλου από n προγράμματα που εκτελούνται για την μέτρηση της απόδοσης ενός συστήματος.

Εάν η απόδοση εκφράζεται ως λόγος τότε ο μέσος όρος που καθορίζει τον ολικό χρόνο εκτέλεσης είναι ο **αρμονικός μέσος** (τύπος 1.4):

$$\frac{n}{\sum_{i=1}^n \frac{1}{\text{λόγος}_i}} \quad (\text{Τύπος 1.4})$$



Λόγος_i : είναι ίσος με το λόγο 1/χρόνος εκτέλεσης_i



Σταθμισμένος χρόνος εκτέλεσης

Εάν τα προγράμματα P1 και P2 του προηγούμενου παραδείγματος δεν εκτελούνται ίσο αριθμό φορών όπως έχει εννοηθεί στον αριθμητικό μέσο, τότε για τον υπολογισμό της απόδοσης χρησιμοποιούμε δύο διαφορετικές προσεγγίσεις.

Όταν έχουμε άνισο αριθμό φορών εκτέλεσης των προγραμμάτων τότε αναθέτουμε έναν **παράγοντα βάρους** w_i σε κάθε πρόγραμμα για να δηλώσουμε την σχετική συχνότητα εκτέλεσης του προγράμματος στο φόρτο (workload) .

Εάν για παράδειγμα 20% των εργασιών που εκτελούνται είναι το πρόγραμμα P1 και το 80% το πρόγραμμα P2 τότε $w_1=0,2$ και $w_2=0,8$ ($w_i \leq 1$).

Αθροίζοντας τα γινόμενα των παραγόντων βάρους με τους χρόνους εκτέλεσης, προκύπτει μια πιο ξεκάθαρη εικόνα για την απόδοση του φόρτου, ο λεγόμενος **σταθμισμένος αριθμητικός μέσος** (τύπος 1.5):

$$\sum_{i=1}^n \text{βάρους}_i \times \text{χρόνος εκτέλεσης}_i \quad (\text{Τύπος 1.5})$$



Βάρος_i (w_i): η συχνότητα εκτέλεσης του i-οστού προγράμματος.



Χρόνος εκτέλεσης_i: ο χρόνος εκτέλεσης του i-οστού προγράμματος.

Ο **σταθμισμένος αρμονικός μέσος** λόγω εμφανίζει την ίδια σχετική απόδοση όπως ο σταθμισμένος αριθμητικός μέσος χρόνων εκτέλεσης (τύπος 1.6)

$$\frac{1}{\sum_{i=1}^n \frac{\text{βάρους}_i}{\text{λόγος}_i}} \quad (\text{Τύπος 1.6})$$

Όπου **βάρος_i (w_i)** η συχνότητα εμφάνισης του προγράμματος i στο φόρτο.

	A	B	C	W(1)	W(2)	W(3)
Πρόγραμμα P1 (sec)	1.00	10.00	20.00	0.50	0.909	0.999
Πρόγραμμα P2 (sec)	1000.00	100.00	20.00	0.50	0.091	0.001
Αριθμητικός μέσος : w(1)	500.50	55.00	20.00			
Αριθμητικός μέσος : w(2)	91.91	18.18	20.00			
Αριθμητικός μέσος : w(3)	2.00	10.09	20.00			

Σχήμα 1.1.2 - Σταθμισμένος αριθμητικός μέσος χρόνος εκτέλεσης για τρεις διαφορετικές σταθμίσεις

Πιο αναλυτικά για τα στοιχεία του σχήματος 1.1.2 μπορούμε να πούμε: Το w(1) σταθμίζει ισοδύναμα τα προγράμματα και δίνει τον αριθμητικό μέσο της γραμμής 3 ο οποίος είναι ίδιος με τον μη σταθμισμένο αριθμητικό μέσο. Το w(2) δίνει το συνδυασμό των προγραμμάτων που είναι αντιστρόφως ανάλογα με τους χρόνους εκτέλεσης του υπολογιστή B. Η γραμμή 4 δείχνει τον αριθμητικό μέσο για αυτή την στάθμιση. Το w(3) σταθμίζει τα προγράμματα με αντίστροφη αναλογία ως προς τους χρόνους εκτέλεσης των δύο προγραμμάτων στον υπολογιστή A. Ο αριθμητικός μέσος για την στάθμιση w(3) δίνεται από την τελευταία σειρά του πίνακα. Το αποτέλεσμα της δεύτερης και τρίτης περίπτωσης στάθμισης είναι η κανονικοποίηση των βαρών ως προς τους χρόνους

Κεφάλαιο 1 / Ενότητα 1

εκτέλεσης των προγραμμάτων που τρέχουν σε αυτό τον υπολογιστή, έτσι ώστε ο χρόνος εκτέλεσης για κάθε υπολογιστή να κατανέμεται ισάξια σε κάθε πρόγραμμα για αυτό τον υπολογιστή. Έτσι για ένα σύνολο n προγραμμάτων κάθε ένα από τα οποία διαρκεί χρόνο T_i σε έναν υπολογιστή, βρίσκουμε ότι το βάρος που αντιστοιχεί για τον συγκεκριμένο υπολογιστή είναι (τύπος 1.7):

$$w_i = \frac{1}{T_i \times \sum_{j=1}^n \left(\frac{1}{T_j} \right)} \quad (\text{Τύπος 1.7})$$



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 4

Εάν υποθέσουμε πως έχουμε ένα σύνολο προγραμμάτων, πως υπολογίζουμε με την απόδοση ενός συστήματος αν τα προγράμματα του φόρτου εργασίας, εκτελούνται ίσο αριθμό φορών;



ΑΠΑΝΤΗΣΗ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ 4

Ένας απλός τρόπος υπολογισμού της απόδοσης ενός συστήματος είναι να υπολογίσουμε τον αριθμητικό μέσο ο οποίος είναι ο μέσος όρος των χρόνων εκτέλεσης των προγραμμάτων του φόρτου.

$$\frac{1}{n} \sum_{i=1}^n \text{χρόνος εκτέλεσης}$$



Κανονικοποιημένος χρόνος εκτέλεσης – Συγκρίσεις

Μια δεύτερη προσέγγιση στην περίπτωση άνισου ποσοστού συμμετοχής των προγραμμάτων στο φόρτο εργασίας (workload), είναι η κανονικοποίηση των χρόνων εκτέλεσης ως προς έναν υπολογιστή αναφοράς και στη συνέχεια ο υπολογισμός του μέσου όρου των κανονικοποιημένων χρόνων εκτέλεσης (Αυτή η προσέγγιση χρησιμοποιείται από τα προγράμματα δοκιμής απόδοσης SPEC, με υπολογιστή αναφοράς τον VAX / 11/780)

Ο μέσος όρος των κανονικοποιημένων χρόνων εκτέλεσης μπορεί να εκφραστεί είτε ως αριθμητικός είτε ως γεωμετρικός μέσος.

Ο γεωμετρικός μέσος κανονικοποιημένων χρόνων εκτέλεσης είναι (τύπος 1.8):

$$\sqrt[n]{\prod_{i=1}^n \text{λόγος χρόνου εκτέλεσης}_i} \quad (\text{Τύπος 1.8})$$



Λόγος χρόνου εκτέλεσης: χρόνος εκτέλεσης κανονικοποιημένος ως προς μηχανή αναφοράς για το i-οστό πρόγραμμα ενός συνόλου από n προγράμματα του φόρτου.



Μια σημαντική ιδιότητα του γεωμετρικού μέσου για δύο δείγματα X_i και Y_i είναι:

$$\frac{\text{γεωμετρικός μέσος}(X_i)}{\text{γεωμετρικός μέσος}(Y_i)} = \text{γεωμετρικός μέσος}\left(\frac{X_i}{Y_i}\right) \quad (\text{Τύπος 1.9})$$



Σε αντίθεση με τον αριθμητικό μέσο, ο γεωμετρικός μέσος είναι συνεπής ανεξαρτήτως του υπολογιστή αναφοράς που χρησιμοποιείται, επομένως ο αριθμητικός μέσος δεν θα έπρεπε να χρησιμοποιείται για τον υπολογισμό μέσου όρου κανονικοποιημένων χρόνων εκτέλεσης.



Παράδειγμα

Έστω:

	Computer A	Computer B	Computer C
Program P1	1	10	20
Program P2	1000	100	20

Να υπολογιστεί ο κανονικοποιημένος αριθμητικός και γεωμετρικός μέσος καθώς και ο συνολικός κανονικοποιημένος χρόνος εκτέλεσης των προγραμμάτων.

Απάντηση:

Αρχικά κανονικοποιούμε τους χρόνους εκτέλεσης ως προς τον υπολογιστή A διαιρώντας τους χρόνους εκτέλεσης για κάθε πρόγραμμα με τους χρόνους εκτέλεσης στον υπολογιστή A. Επαναλαμβάνουμε τα ίδια για να κανονικοποιήσουμε ως προς τον υπολογιστή B μετά ως προς τον υπολογιστή C (οι τιμές φαίνονται στο σχήμα 1.1.3). Στην συνέχεια υπολογίζουμε τον αριθμητικό και γεωμετρικό μέσο με βάση τους αντίστοιχους τύπους.

	Κανονικοποίηση με A			κανονικοποίηση με B			κανονικοποίηση με C		
	A	B	C	A	B	C	A	B	C
P1	1	10	20	0,1	1	2	0,05	0,5	1
P2	1	0,1	0,02	10	1	0,2	50	5	1
Αριθμητικός μέσος	1	5,05	10,11	5,5	1	1,1	25,03	2,75	1
Γεωμετρικός μέσος	1	1	0,63	1	1	0,63	1,58	1,58	1
Ολικός χρόνος	1	0,11	0,04	9,1	1	0,36	25,03	2,75	1

Σχήμα 1.1.3 - Σταθμισμένος αριθμητικός μέσος χρόνος εκτέλεσης για τρεις διαφορετικές σταθμίσεις . Η απόδοση του αριθμητικού μέσου ποικίλει και εξαρτάται από τον υπολογιστή αναφοράς που χρησιμοποιούμε. Παρατηρούμε από το σχήμα 1.1.3 ότι σύμφωνα με τον κανονικοποιημένο αριθμητικό μέσο ως προς τον υπολογιστή A ο χρόνος εκτέλεσης του υπολογιστή B είναι 5 φορές μεγαλύτερος από αυτόν της A ενώ το αντίθετο συμβαίνει αν κανονικοποιήσουμε ως προς τον υπολογιστή B. Ο γεωμετρικός μέσος είναι συνεπής και ανεξάρτητος από την κανονικοποίηση. Οι υπολογιστές A, B έχουν την ίδια απόδοση και ο χρόνος εκτέλεσης της C είναι το 0,63 του χρόνου εκτέλεσης των A, B ($1/1,58=0.63$). Δυστυχώς όμως ο ολικός κανονικοποιημένος χρόνος εκτέλεσης του A είναι 10 φορές μεγαλύτερος από αυτόν του B και ο ολικός κανονικοποιημένος χρόνος εκτέλεσης του B είναι 3 μεγαλύτερος από αυτόν του C .

Αρχιτεκτονική Υπολογιστών Ι

Υπολογίζω τους κανονικοποιημένους αριθμητικούς μέσους ως προς τον υπολογιστή A με την βοήθεια του τύπου 1.3. Κατά συνέπεια :

$$\text{Κανονικοποιημένος Αριθμητικός μέσος } A/A = 1/2 * (1+1) = 1$$

$$\text{Κανονικοποιημένος Αριθμητικός μέσος } B/A = 1/2 * (10+0.1) = 5.05$$

$$\text{Κανονικοποιημένος Αριθμητικός μέσος } C/A = 1/2 * (20+0.02) = 10.11$$

Ομοίως και για τους κανονικοποιημένους αριθμητικούς μέσους ως προς τους υπολογιστές B και C .

Στη συνέχεια υπολογίζω τους κανονικοποιημένους γεωμετρικούς μέσους. Υπολογίζω πρώτα ως προς τον υπολογιστή A:

$$\text{Κανονικοποιημένος Γεωμετρικός μέσος } A/A = 1$$

$$\text{Κανονικοποιημένος Γεωμετρικός μέσος } B/A = \sqrt[2]{10+0.1} = 1$$

$$\text{Κανονικοποιημένος Γεωμετρικός μέσος } C/A = \sqrt[2]{20+0.002} = 0.63$$

Ομοίως και για τα υπόλοιπα.

Τέλος για τον υπολογισμό του κανονικοποιημένου ολικού χρόνου εκτέλεσης που ζητάει κάνουμε τα εξής:

Πρώτα αν ο υπολογιστής αναφοράς είναι ο A έχω:

$$\text{Ολικός κανονικοποιημένος χρόνος } A/A = (1+1000) / (1+1000) = 1$$

$$\text{Ολικός κανονικοποιημένος χρόνος } B/A = (10+100) / (1000+1) = 0.11$$

$$\text{Ολικός κανονικοποιημένος χρόνος } C/A = (20+20) / (1000+1) = 0.04$$

Ομοίως και για υπολογιστές αναφοράς τους B και C. 🌈



Η σχέση ανάμεσα στους μέσους (αρμονικός, γεωμετρικός, αριθμητικός) του ίδιου συνόλου αριθμών είναι πάντα:

$$\text{αρμονικός μέσος} \leq \text{γεωμετρικός μέσος} \leq \text{αριθμητικός μέσος}$$



ΔΡΑΣΤΗΡΙΟΤΗΤΑ 5

Μπορείτε να περιγράψετε τα πλεονεκτήματα και τα μειονεκτήματα που προκύπτουν από τη χρήση του γεωμετρικού μέσου για τον υπολογισμό της απόδοσης ενός υπολογιστικού συστήματος; Να συγκρίνετε την απάντησή σας με την υποενότητα που ακολουθεί: «Υπέρ και κατά του γεωμετρικού μέσου».



Υπέρ και κατά του γεωμετρικού μέσου

Τα βάρη στον σταθμισμένο αριθμητικό μέσο είναι ένα σύνολο ανάλογο των χρόνων εκτέλεσης. Κατά συνέπεια, για έναν υπολογιστή, τα βάρη επηρεάζονται από τη συχνότητα εκτέλεσης του κάθε προγράμματος, από τις ιδιαιτερότητες του κάθε υπολογιστή και από το μέγεθος του προγράμματος εισόδου. Ο γεωμετρικός μέσος κανονικοποιημένων χρόνων εκτέλεσης, είναι ανεξάρτητος από τους χρόνους εκτέλεσης των μεμονωμένων προγραμμάτων και τον υπολογιστή που έχει γίνει η κανονικοποίηση. Όταν πρέπει να εκτιμηθεί η σχετική απόδοση για σταθερά προγράμματα με απροσδιόριστες εισόδους, οι διαγωνιζόμενοι μπορούν να νοθεύσουν τα αποτελέσματα που δίνει ο σταθμισμένος αριθμητικός μέσος με την χρήση προγραμμάτων δοκιμής απόδοσης με μεγαλύτερες εισόδους, υπερέχοντας στον χρόνο εκτέλεσης. Σε τέτοιες περιπτώσεις, ο γεωμετρικός μέσος είναι λιγότερο παραπλανητικός από τον αριθμητικό μέσο.

Ένα σημαντικό μειονέκτημα του γεωμετρικού μέσου κανονικοποιημένων χρόνων εκτέλεσης είναι ότι παραβιάζει τη βασική αρχή της μέτρησης της απόδοσης, αφού δεν προβλέπει το χρόνο εκτέλεσης. Ο γεωμετρικός μέσος στο παράδειγμα της προηγούμενης υποενότητας (σχήμα 1.1.3), δηλώνει ότι η απόδοση των υπολογιστών A και B για τα προγράμματα P1 και P2 είναι η ίδια. Αυτό είναι σωστό για ένα φόρτο που εκτελεί το πρόγραμμα P1 100 φορές για κάθε εκτέλεση του προγράμματος P2. Ο συνολικός χρόνος εκτέλεσης για αυτό το φόρτο δηλώνει ότι οι υπολογιστές A και B είναι περίπου 50% πιο γρήγοροι από τον υπολογιστή C σε αντίθεση με τον γεωμετρικό μέσο ο οποίος δηλώνει ότι ο υπολογιστής C είναι πιο γρήγορος από τους A και B !



Γενικά, δεν υπάρχει κάποιος φόρτος για τρεις ή περισσότερους υπολογιστές ο οποίος να έχει την απόδοση που προβλέπεται από τον γεωμετρικό μέσο κανονικοποιημένων χρόνων εκτέλεσης.

Ένα επιπλέον μειονέκτημα της χρήσης γεωμετρικού μέσου ως μέθοδο σύνοψης της απόδοσης για μία ακολουθία προγραμμάτων δοκιμής, είναι ότι ενθαρρύνει τους σχεδιαστές λογισμικού και υλικού να επικεντρώσουν την προσοχή τους στις δοκιμές απόδοσης των οποίων η απόδοση βελτιώνεται πιο εύκολα, παρά σε αυτά που είναι πιο αργά.

Η ιδανική λύση είναι να μετρήσουμε ένα πραγματικό σύνολο εργασίας και να σταθμίσουμε τα προγράμματα σύμφωνα με την συχνότητα εκτέλεσης τους. Αν αυτό δεν είναι εφικτό, τότε κανονικοποιώντας με τέτοιο τρόπο ώστε να αναλογεί σε κάθε πρόγραμμα ίσος χρόνος για κάποιο υπολογιστή, βρίσκουμε τουλάχιστον μοναδικά σχετικά βάρη τα οποία υπολογίζουν τον χρόνο εκτέλεσης ενός συνόλου εργασίας με αυτή την αναλογία. Εάν τα αποτελέσματα πρέπει να κανονικοποιηθούν ως προς έναν συγκεκριμένο υπολογιστή αναφοράς, πρώτα συνοψίζουμε την απόδοση με την κατάλληλη σταθμισμένη μέτρηση και μετά κανονικοποιούμε.



Πλάνες και παγίδες

Σκοπός αυτής της υποενότητας είναι να εξηγήσει ορισμένες συχνές παρανοήσεις (πλάνες) που πρέπει να αποφεύγονται και λάθη (παγίδες) στα οποία υποπέφτουμε ίσως επειδή γενικεύουμε αρχές που ισχύουν κάτω από συγκεκριμένες συνθήκες. Ενδεικτικά έχουμε τις ακόλουθες περιπτώσεις:

- **MIPS (εκατομμύρια εντολών ανά δευτερόλεπτο)**
- **MFLOPS (εκατομμύρια πράξεις κινητής υποδιαστολής ανά δευτερόλεπτο)**
- **Τα σύνθετα προγράμματα δοκιμής**
- **Τα προγράμματα δοκιμής απόδοσης**
- **Κορυφαία και μέση απόδοση**

MIPS

ΠΛΑΝΗ: Το MIPS είναι ένα ακριβές μέτρο σύγκρισης απόδοσης μεταξύ υπολογιστών.

Όπου MIPS δίνεται από τον τύπο (1.10) :

$$\text{MIPS} = \frac{\text{πλήθος εντολών}}{\text{χρόνος εκτέλεσης} * 10^6} = \frac{\text{ρυθμός ρολογιού}}{\text{CPI} * 10^6} \quad (\text{Τύπος 1.10})$$

Σχετίζοντας το MIPS με το χρόνο εκτέλεσης παίρνουμε τον τύπο (1.10) :

$$\text{χρόνος εκτέλεσης} = \frac{\text{πλήθος εντολών}}{\text{MIPS} * 10^6} \quad (\text{Τύπος 1.11})$$

Μιας και το MIPS είναι ο ρυθμός λειτουργιών ανά μονάδα χρόνου συμπεραίνουμε ότι οι ταχύτεροι υπολογιστές έχουν υψηλότερο MIPS. Το βασικό πλεονέκτημα του MIPS είναι ότι είναι εύκολο στην κατανόηση ειδικά από έναν μη ειδικό, το πρόβλημα όμως που προκύπτει χρησιμοποιώντας το ως μέτρο σύγκρισης της απόδοσης είναι:

- Το MIPS εξαρτάται αποκλειστικά από το εκάστοτε σύνολο εντολών καθιστώντας έτσι δύσκολη τη σύγκριση MIPS υπολογιστών με διαφορετικά σύνολα εντολών.

- Το MIPS διαφέρει μεταξύ προγραμμάτων ακόμα και στον ίδιο υπολογιστή .
- Το MIPS μπορεί να μεταβάλλεται αντίστροφα της απόδοσης .

MFLOPS

ΠΛΑΝΗ : Το MFLOPS είναι ένα ακριβές και χρήσιμο μέτρο σύγκρισης της απόδοσης

Όπως το MIPS έτσι και το MFLOPS είναι μια δημοφιλής εναλλακτική μέθοδος σύγκρισης της απόδοσης αντί του χρόνου εκτέλεσης. Όπου MFLOPS είναι (τύπος 1.12):

$$\text{MFLOPS} = \frac{\text{πλήθος πράξεων κινητής υποδιαστολής σε ένα πρόγραμμα}}{\text{χρόνος εκτέλεσης σε δευτερόλεπτα} * 10^6} \quad (\text{Τύπος 1.12})$$

Όπως διαφαίνεται στον τύπο (1.12), η κατάταξη σε MFLOPS εξαρτάται τόσο από τον υπολογιστή όσο και από το ίδιο το πρόγραμμα. Το γεγονός όμως ότι χρησιμοποιείται για μέτρηση επίδοσης κινητής υποδιαστολής το κάνει μη εφαρμόσιμο εκτός του πεδίου αυτού. Χαρακτηριστικό παράδειγμα είναι αυτό των μεταγλωττιστών όπου έχουν μηδενικό MFLOPS όσο γρήγορος και αν είναι ο υπολογιστής κι αυτό γιατί σπάνια χρησιμοποιούν αριθμητική κινητής υποδιαστολής.

Βασισμένο σε πράξεις και όχι σε εντολές το MFLOPS υποτίθεται ότι αποτελεί ένα αντικειμενικό μέτρο σύγκρισης διαφορετικών μηχανών, δυστυχώς όμως δεν είναι αξιόπιστο γιατί το πλήθος των πράξεων κινητής υποδιαστολής διαφέρει από μηχανή σε μηχανή. Χαρακτηριστικό παράδειγμα ο Cray C90, ο οποίος δεν έχει την πράξη της διαίρεσης ενώ αντίθετα ο Intel Pentium έχει τις πράξεις της διαίρεσης, του ημιτόνου, του συνημιτόνου κ.α. Οι μετρήσεις με MFLOPS μεταβάλλονται και από πρόγραμμα σε πρόγραμμα ανάλογα με το είδος των πράξεων κινητής υποδιαστολής που αυτό χρησιμοποιεί. Για παράδειγμα ένα πρόγραμμα με 100% προσθέσεις κινητής υποδιαστολής δίνει καλύτερες μετρήσεις από ένα άλλο με 100% διαιρέσεις κινητής υποδιαστολής. Τέλος, θα μπορούσαμε να πούμε ότι η μέτρηση του MFLOPS για ένα μόνο πρόγραμμα δεν μπορεί να γενικευτεί έτσι ώστε να αποτελέσει καθολικό μέτρο της απόδοσης ενός H/Y.

Τα σύνθετα προγράμματα δοκιμής

ΠΛΑΝΗ : Τα σύνθετα προγράμματα δοκιμής προβλέπουν την απόδοση πραγματικών προγραμμάτων .

Τα περισσότερο γνωστά παραδείγματα τέτοιων προγραμμάτων δοκιμής είναι το Dhrystone και το Whetstone. Δεν είναι αληθινά προγράμματα και κατά συνέπεια δεν αντικατοπτρίζουν τη συμπεριφορά των προγραμμάτων για αστάθμητους παράγοντες. Η βελτιστοποίηση μεταγλωττιστή και υλικού μπορεί να διογκώσει τεχνητά την απόδοση των προγραμμάτων δοκιμής πράγμα που δεν ισχύει με τα πραγματικά. Η άλλη πλευρά

του νομίσματος είναι ότι επειδή τα προγράμματα δοκιμής δεν είναι πραγματικά προγράμματα, ο σχεδιαστής δεν ανταμείβεται για τις βελτιστοποιήσεις που κάνει .

Τα προγράμματα δοκιμής

ΠΛΑΝΗ : Τα προγράμματα δοκιμής απόδοσης παραμένουν απεριόριστα έγκυρα.

Αρκετοί παράγοντες επηρεάζουν τη δυνατότητα ενός προγράμματος δοκιμής απόδοσης να προβλέψει την πραγματική απόδοση, πολλοί απ' τους οποίους είναι χρονικά μεταβαλλόμενοι. Ο πιο αξιόλογος παράγοντας που επηρεάζει την χρησιμότητα ενός προγράμματος δοκιμής, είναι η ικανότητά του να αντιστέκεται στο "cracking", γνωστή και ως δοκιμασία μηχανικής απόδοσης. Μόλις ένα πρόγραμμα δοκιμής τυποποιηθεί και δοθεί στην αγορά, ασκούνται τρομερές πιέσεις να γίνει καλύτερη η απόδοση με αντικειμενικές βελτιώσεις ή με τη διερμηνεία των κανόνων για την εκτέλεση ενός προγράμματος δοκιμής απόδοσης.



Οι πωλητές βρήκαν μεθόδους να συντονίσουν τις αποδόσεις διαφορετικών προγραμμάτων δοκιμής με τη χρήση διαφορετικών μεταγλωττιστών ή προεπεξεργαστών, καθώς επίσης και ειδικών σημαιών για τα προγράμματα δοκιμής απόδοσης. Οι ειδικές σημαιές επιτρέπονται, παρόλο που είναι αντικανονικές και οδηγούν συνήθως σε λανθασμένη μεταγλώττιση! Η δοκιμασία μηχανικής απόδοσης μερικές φορές εφαρμόζεται και σε βιβλιοθήκες για εκτέλεση προγραμμάτων.

Οι αυξανόμενες βελτιώσεις στην τεχνολογία μπορούν επίσης να αλλάξουν το ενεργό πεδίο μέτρησης για ένα πρόγραμμα δοκιμής. Ας θεωρήσουμε το πρόγραμμα δοκιμής gcc, το οποίο θεωρείται ένα από τα πιο ρεαλιστικά και ενδιαφέροντα προγράμματα δοκιμής του SPEC92. Η επίδοσή του είναι ένας συνδυασμός χρόνου ΚΜΕ και επεξεργασίας σε πραγματικό χρόνο. Η είσοδος παραμένει σταθερή και η επεξεργασία σε πραγματικό χρόνο είναι περιορισμένη από κάποιους παράγοντες, συμπεριλαμβανομένου του χρόνου πρόσβασης στον δίσκο (ο οποίος βελτιώνεται αργά). Άρα, ένα αυξανόμενο ποσοστό του χρόνου εκτέλεσης είναι ένας μεγαλύτερος χρόνος συστήματος από το χρόνο της ΚΜΕ. Απ' την άλλη, μπορεί να αλλάξει η είσοδος επιβαρύνοντας τον χρόνο, όταν πρέπει να μεταγλωττιστούν μεγαλύτερα προγράμματα. Στην πραγματικότητα, η είσοδος του SPEC92 μεταβλήθηκε ώστε να περιλαμβάνει τέσσερα αντίγραφα από κάθε αρχείο εισόδου το οποίο χρησιμοποιούταν στο SPEC89. Ενώ αυτό αυξάνει το χρόνο εκτέλεσης, μπορεί να απεικονίζει ή όχι τον τρόπο με τον οποίο χρησιμοποιούνται οι μεταγλωττιστές. Μετά από μια μεγάλη χρονική περίοδο, αυτές οι αλλαγές μπορούν να αχρηστέψουν και το καλύτερο πρόγραμμα δοκιμής απόδοσης.

Η κορυφαία και μέση απόδοση

ΠΛΑΝΗ : Η κορυφαία απόδοση ακολουθεί τα ίχνη της μέσης απόδοσης.

Μία ερμηνεία του όρου "κορυφαία απόδοση" είναι εγγυημένα η μεγαλύτερη τιμή για την απόδοση που ωστόσο δεν πρόκειται να την ξεπεράσει το σύστημα. Το χάσμα μεταξύ της κορυφαίας απόδοσης και της μέσης απόδοσης (αυτή που παρατηρείται πειραματικά) είναι τυπικά ένας παράγοντας του 10 ή περισσότερο στους υπερ-υπολογιστές. Μιας και η διαφορά είναι τόσο μεγάλη, η κορυφαία απόδοση δεν είναι χρήσιμη στην πρόβλεψη της μέσης απόδοσης εκτός αν η εργασία που πρέπει να γίνει αποτελείται από μικρά προγράμματα τα οποία συνήθως λειτουργούν κοντά στην κορυφή.

Όσο η χρήση της κορυφαίας απόδοσης ήταν υπερβολική στους υπερ-υπολογιστές, η χρήση της στην επιχείρηση των μικροεπεξεργαστών είναι το ίδιο παραπλανητική.

Ανακεφαλαιώνοντας λοιπόν ...



Ο υπολογιστής που φέρει εις πέρας το ίδιο ποσό δουλείας σε λιγότερο χρόνο είναι ο ταχύτερος. Η διαφορά στη σύγκριση έγκειται στο αν μετράμε το χρόνο εκτέλεσης μιας εργασίας (χρόνος απόκρισης) ή πολλών (ρυθμαπόδοση).



Η απόδοση ενός υπολογιστή διαφέρει ανάλογα με το αν υπολογίζουμε βασιζόμενοι στο διανυθέντα χρόνο ή στον χρόνο χρήσης της ΚΜΕ.



Υπάρχουν 4 διαφορετικά επίπεδα προγραμμάτων που χρησιμοποιούνται για την εκτίμηση της απόδοσης ενός υπολογιστή: Πραγματικά προγράμματα, Πυρήνες, Προγράμματα Δοκιμής – Παιχνίδια, τα Σύνθετα προγράμματα δοκιμής αλλά και συλλογές από προγράμματα δοκιμής.



Ο συνολικός χρόνος εκτέλεσης είναι η πιο απλή προσέγγιση της σχετικής απόδοσης ενός συστήματος. Καθορίζεται από τον αριθμητικό μέσο (ο μέσος όρος των χρόνων εκτέλεσης). Ο σταθμισμένος χρόνος εκτέλεσης προσδιορίζεται με δύο μεγέθη: i) τον σταθμισμένο αριθμητικό μέσο και ii) τον σταθμισμένο αρμονικό μέσο.



Ο κανονικοποιημένος χρόνος εκτέλεσης αποτελεί μία 2^η προσέγγιση όταν έχω άνισο ποσοστό συμμετοχής των προγραμμάτων στο φόρτο. Περιλαμβάνει την κανονικοποίηση των χρόνων εκτέλεσης ως προς τον υπολογιστή αναφοράς και τον υπολογισμό του μέσου όρου των κανονικοποιημένων χρόνων εκτέλεσης.



Ο γεωμετρικός μέσος είναι λιγότερο παραπλανητικός από τον αριθμητικό μέσο.



Συχνές παρανοήσεις (πλάνες) και λάθη (παγίδες) που υποπέφτουμε γενικεύοντας αρχές σχεδιασμού αποτελούν οι: MIPS, MFLOPS, τα συνθετικά προγράμματα δοκιμής, τα προγράμματα δοκιμής απόδοσης, κορυφαία και μέση απόδοση.