

# Benchmarking

Andraž Filipčič

January 2022

## 1 Uvod

Merjenje zmogljivosti je zaradi več razlogov postalo zelo razširjeno področje z veliko ponudbe, med glavnimi razlogi je za večino uporabnikov to pomoč pri primerjavi izdelkov med sabo, vsi smo že navajeni nakih visokih števil, ki nam brez razlage povedo, kateri kosi računalniške opreme so bolj zmogljivi.

Takšne meritve uporabljajo tudi pri snovanju novih procesorjev, preverijo namreč lahko, če so zamišljene izboljšave tudi dejansko izboljšave in kako bi lahko te izboljšave še izboljšali.

Pomembno je pa tudi za ugotavljanje, če oprema pravilno, oziroma optimalno, deluje. Lahko celo najdemo možnosti za izboljšavo (mogoče ne ravni pri procesorju, mogoče pa npr. pri razporeditvi diskovja v večjih sistemih).

## 2 Merjenje procesorske zmogljivosti

Merjenje zmogljivosti procesorskih enot je morda še najbolj razširjena. Za uporabnike je pomembna predvsem pri morebitnem nakupu novega procesorja, povsod pa najdemo veliko različnih meril in števil, s katerimi je zmogljivost procesorjev predstavljena. A kljub različnim vrednotenjem večina načinov merjenja zmogljivosti uporablja podobne teste:

### 2.1 Celoštevilske operacije

Eno najbolj pogostih meril so celoštevilske operacije, saj predstavljajo nekako najbolj osnovne procesorske ukaze, zato najbolj uspešno predstavi koliko operacij na sekundo zmora izvajati procesor.

### 2.2 Operacije števil s plavajočo vejico

Naslednje dokaj pomembno merilo so operacije števil s plavajočo vejico, saj večina znanstvenih (in pa tudi drugih) izračunov ne deluje samo s celimi števili. Tako na primer pri ocenah zmogljivosti superračunalnikov zmeraj govorimo o FLOPS-ih (floating point operations per second - število operacij s števili s plavajočo vejico na sekundo), saj je to bolj realna mera kot pa samo število operacij na sekundo.

### 2.3 Kompresijski testi

Pomembni testi so tudi testi kompresije, saj se izvajajo nad bolj kompleksnimi podatkovnimi strukturami kot pa samo števili in potrebujejo kompleksnejše operacije za izvedbo, ki pa so dobra predstavitev realnih operacij, ki jih uporablja veliko programske opreme. Večinoma se izvaja brezizgubna (lossless) kompresija, sami algoritmi pa so različni od testa do testa.

## 2.4 Iskanje praštevil

Eden pogosto uporabljenih testov je tudi iskanje praštevil, saj takšni algoritmi uporabljajo veliko zank, množenja in ostanka pri deljenju, kar so pogoste operacije programske opreme.

## 2.5 Primeri programske opreme za izvajanje merjenja

Programske opreme za merjenje zmogljivosti procesorjev je zelo veliko.

Windows ima recimo vgrajen program Performance Monitor, s katerim lahko merimo zmogljivost sistema (tako procesorja kot ostalih komponent, na primer disk, pomnilnik, itd).

Linux pa ima na primer na voljo ukaz 'stress', ki sicer samoopisno ni "benchmarking tool", saj je bolj mišljen temu, da se sistem maksimalno obremeni in preveri, kako se ob takih pogojih obnaša, a lahko tudi na tak način dobim nekaj podobnih informacij kot z bolj konvencionalnimi testi za merjenje zmogljivosti. Izvaja ukaze `sqrt()`, `sync()`, `malloc()`, itd, tako da obremeni več delov sistema kot pa samo procesor.

Poleg tega pa je seveda še cel kup zastoj in plačljive opreme, ki nudi razne funkcionalnosti.

## 3 Merjenje zmogljivosti glavnega pomnilnika

Operacije, ki jih lahko izvajamo za merjenje zmogljivosti pomnilnika so veliko bolj omejene kot pa pri procesorju. Tukaj se večinoma meri hitrost prenosa podatkov oziroma čase dostopa do podatkov. Programi uporabljajo dolge nize nad katerimi izvajajo preproste operacije, na primer seštevanje/odštevanje in prepisovanje podatkov. Tako najlažje izmerijo samo komunikacijo med procesorjem in pomnilnikom, brez da bi izgubljali čas s kompleksnim delovanjem procesorja.

Pomembno pri merjenju zmogljivosti je upoštevanje predpomnilnika, namreč če imamo podatkov premalo se nam lahko zgodi, da bodo dostopi (po prvem) hitrejši, saj bodo podatki na voljo v procesorskem predpomnilniku. Program "STREAM" zato vse svojim nizom nastavi dolžino na približno štirikrat velikost predpomnilnika.

Eden od načinov testiranja pomnilnika je sekvenčni sprehod čez celoten pomnilnik s spreminjujočim korakom, tako da ob vsakem sprehodu povečamo korak in dostopamo do vsakega  $n$ -tega podatka.

Zanimivo pri testiranju pomnilnika so večprocesorski sistemi s skupnim pomnilnikom (NUMA), kjer dostopni časi do različnih delov pomnilnika niso enaki, saj so fizično bolj oddaljeni. Tako je treba izmeriti vse različne čase dostopa do bolj oddaljenih delov pomnilnika, da dobimo bolj realno sliko.

## 4 Merjenje zmogljivosti omrežja

Cilj pri testiranju omrežja je doseči čim manj zahtevno delo za procesor, zato da se najlažje dobi realne številke zmogljivosti samega omrežja brez izgube časa na procesorju. Velikokrat se v ta namen uporabi sistem strežnika in klienta, v katerem en računalnik deluje kot strežnik, ki podatke samo sprejema, drug računalnik, klient, mu jih pa pošilja, v obdelavi pa so preprosti paketi, s katerimi procesor nima veliko opravka. Na ta način dobimo precej čisto sliko same hitrosti prenosa.

Pri meritvah je pomembnih več faktorjev:

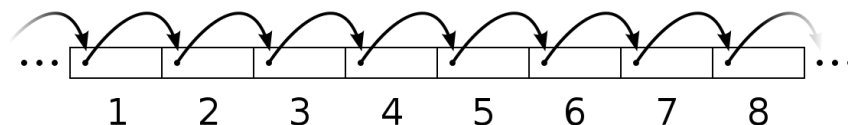
- Latenca, ki pove, koliko časa je potrebno, da en paket prepotuje od enega do drugega konca
- Količina podatkov, ki se prenese v časovni enoti
- Količina paketov, ki se je na poti izgubila

## 5 Merjenje zmogljivosti datotečnega sistema

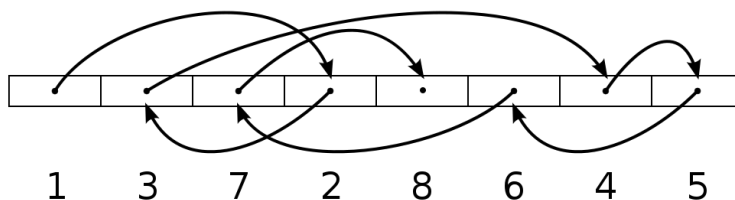
Pri merjenju zmogljivosti datotečnega sistema nas lahko zanima veliko različnih stvari, odvisno od namembnosti sistema. Pogosti testi so kombinacije sekvenčnega in naključnega branja in pisanja.

Pri sekvenčnem branju se datoteke iz sistema berejo v celoti, tako se dostopa do zaporednih podatkov na disku. Pri naključnem pa se dostopa samo do posameznih delov datotek, dostop je v tem primeru do nezaporednih podatkov.

### Sequential access



### Random access



Za osebne računalnike to merilo mogoče ni tako zanimivo kot ostala, je pa precej pomembno pri večjih računalniških sistemih, na primer superračunalnikih. Pri teh je pomembno, kako zmogljiv je datotečni sistem, saj do njega lahko dostopa več računalnikov hkrati, lahko zahtevajo različne količine podatkov (veliko zahtevkov za malo podatkov je na primer veliko bolj obremenjujoče kot malo zahtevkov za veliko podatkov), dostopajo do istih podatkov, uporabljajo ista dostopna mesta za dostop do podatkov, itd.

## 5.1 Primeri programske opreme za izvajanje merjenja

Različne programske opreme je veliko, za ogled bom prikazal dve bolj popularni, fio in ior:

- Operacije  
Same operacije, ki ju programa izvajata, so podobne, saj so namenjene merjenju istih stvari. Dopusčena je možnost spremembe količine podatkov, načina pisanja/branja, velikost blokov v katerih se podatki prenašajo, itd.
- Izvedba poizvedovanja  
Razlika med programa se pokaže pri paralelnem izvajanju I/O operacij. Fio za paralelizem uporablja sistemski klic `fork()`, ior pa za ta namen uporablja MPI. Na same rezultate to sicer ne vpliva, omogoči pa na primer to, da ior vse rezultate nabere in izpiše v eno samo datoteko, medtem ko fio svoje rezultate porazdeli med več datotek (ena za vsak proces).
- Generiranje datotek  
Še ena razlika v implementaciji je način generiranja datotek za branje. Fio je sposoben v trenutku generirate neke naključne datoteke, ki jih potem bere, ior pa rabi najprej izvesti pisanje vseh datotek, da jih potem lahko še prebere. Na samo meritev hitrosti branja to nima učinka, test pa vseeno traja dlje zaradi dodatnih potrebnih operacij pisanja (kar se lahko precej pozna pri velikem številu in velikosti datotek).

## 6 Merjenje zmogljivosti za določeno programsko opremo

Kot zanimivost se tukaj pojavijo testi, ki izmerijo kako dobro deluje specifična programska oprema na določenem sistemu.

Primer tega so nekatere računalniške igrice, ki ponujajo vnaprej pripravljene simulacije, s katerimi izmerijo frekvenco slik (frame rate), zasedenost in temperaturo procesorja in grafične kartice, itd. Takšno možnost je ponujal npr. že Quake 3, ki je vnaprej določeno simulacijo igre skušal izvesti čim hitreje.

Veliko programske opreme za znanstvene izračune tudi ponuja vnaprej pripravljene izračune/simulacije, kateri so med drugim namenjeni testiranju sistema in skalabilnosti.

## 7 Zaključen razmislek

Medtem ko so benchmarki lahko dober indikator tega, kako dobro deluje nek kos opreme, pa so lahko tudi zelo zavajajoči. V preteklosti so na primer nekateri proizvajalci računalniške opreme svojo opremo zasnovali tako, da je na določenih testih kazala odlično zmogljivost, v realnosti pa se je pokazalo, da so to dosegli z določenimi triki, tako da realna zmogljivost ni bila primerljiva s predstavljenim.

Prav tako ob različnih testih za merjenje ista oprema kaže drugačne rezultate. Torej so seveda uporabljeni tisti testi, ki bodo pokazali najboljše rezultate, in predstavljeni bodo samo takšni.

## 8 Viri

[https://en.wikipedia.org/wiki/Benchmark\\_\(computing\)](https://en.wikipedia.org/wiki/Benchmark_(computing))  
[https://en.wikipedia.org/wiki/Computer\\_performance](https://en.wikipedia.org/wiki/Computer_performance)  
<https://linux.die.net/man/1/stress>  
[https://www.cpubenchmark.net/cpu\\_test\\_info.html](https://www.cpubenchmark.net/cpu_test_info.html)  
<https://www.cs.virginia.edu/stream/FTP/Code/stream.c>  
[https://fio.readthedocs.io/en/latest/fio\\_doc.html](https://fio.readthedocs.io/en/latest/fio_doc.html)  
<https://ior.readthedocs.io/en/latest/>